

ACSEのLOTOSによる記述の試み

内山光一 (株) 東芝 情報通信システム技術研究所)
藤田朋生 (日本電気 (株) 基本ソフトウェア開発本部)
小野昌秀, 佐藤嘉一 (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, 辻宏郷 (三菱電機 (株) 情報電子研究所)
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 (電子総合技術研究所 情報アーキテクチャ部)

OSIのプロトコルやサービスの仕様を記述するために、ISOで仕様記述言語LOTOSが開発され、LOTOSを使用した仕様記述が行われるようになってきた。しかしながら、OSIの応用層を記述した例は、データ構造や記述対象が複雑であるために、まだ少ない。そこで、我々LOTOS研究会では実際にOSI応用層の内、プロトコル的には簡単であるが、ある程度データ構造が複雑であり、またプロトコルを実現する環境が比較的複雑である、ACSEの仕様をLOTOSを使用して実験的に記述してみた。本稿では、この現実に使用されているOSI応用層プロトコルの記述実験に関して報告する。

Experimental Descriptions of OSI ACSE Protocol Specification

M. Uchiyama (Toshiba Corporation)
T. Fujita (NEC Corporation)
M. Ono, Y. Sato (Oki Electronic Industry Co., Ltd.)
T. Gono (Fujitsu Limited)
K. Tanaka, H. Tsuji (Mitsubishi Electric Corporation)
K. Yamanaka (Nippon Telegraph and Telephone Corporation)
K. Ohmaki (Electrotechnical Laboratory)

Formal specification language LOTOS was developed in ISO in order to describe the specifications of OSI protocols and services. Several specifications have been described in LOTOS. For OSI Application Layers, however, there are few LOTOS descriptions developed, since their data structures and entities are very complex. We, members of LOTOS reserach group, have described OSI ACSE as an OSI Application Layer example. In this paper, this experimental description is reported.

1. はじめに

通信プロトコルOSIのための形式的仕様記述言語LOTOSがISO 8807として制定された[1]。我々はLOTOSの妥当性を探ることおよびLOTOSの処理系としてはどんなものが考えられるかを調べることを目的とし、(財)情報処理相互運用技術協会の場合を借りて、LOTOS研究会を行ってきた。その活動の1つとして、OSIの現実の規格をLOTOSにより記述することを試みていたが、作業が一段落したので、今までの成果を報告する。

最初の試みなので、仕様があまり複雑でないこと、およびLOTOSによる応用層の形式記述化がまだされていないことを理由に、ACSEプロトコル[2]を対象とした。

ACSEプロトコルは、アソシエーション(コネクション相当のもの)の確立と解放の処理だけを規定した、OSIの応用層の中では比較的簡単なプロトコルである。しかし、プロトコルデータ単位のフォーマットをASN.1[3]を用いて記述しているため、その部分がLOTOSの抽象データ型でどう記述されるかということ、またACSEの記述とはアソシエーション管理という観点からみたときに応用層のモデルの中でどの部分を記述することになるのか、という2点に特に興味を持たれた。

本稿では、ACSEを簡単に説明した後で、LOTOSで記述するに当たっての基本的な取り組み方、特に意識した点を述べ、その後、形式記述の構成、プロセスの説明を簡単に行い、最後にこの活動を通しての結論/感想/課題を述べる。

LOTOSの構文や意味については[1]を参照されたい。

2. ACSEの説明

実開放型システムにおいて、業務を実行するアプリケーションプロセスの一部で、通信に関する部分をアプリケーションエン

ティティ(AE)と呼ぶ。AEはOSIの中の応用層に位置し、プレゼンテーション層から提供されるサービスを利用して通信を行う。AE間の協調関係はアソシエーションと呼ばれる。これはプレゼンテーション層以下の層のコネクションに相当する。通信を行うAEは通信を行うに先立ち、必ず相手AEとの間にアソシエーションを確立しなければならない。1つのAEは複数の応用サービス要素(ASE)といくつかの調整機能から構成される。ASEは特定の通信機能の集合であり、ACSEはアソシエーションを制御するためのASEである。

ACSEは以下の4つのサービスプリミティブを利用者に提供する。

A-ASSOCIATE 要求/指示/応答/確認

A-RELEASE 要求/指示/応答/確認

A-ABORT 要求/指示

A-P-ABORT 指示

A-ASSOCIATEはアソシエーションを確立するために用いられ、必ず確認がとられる。A-RELEASEはアソシエーションを解放するために用いられ、両方のAEの合意に基づきアソシエーションは解放される。A-ABORTとA-P-ABORTはアソシエーションを強制的に解放するためのものであり、前者はACSEサービスの利用者または提供者から起動される場合に使用され、後者はプレゼンテーション層以下の層から起動された場合に使用される。

サービスプリミティブはそれぞれ以下のようにPDU(プロトコルデータ単位)にマッピングされる。

A-ASSOCIATE 要求/指示 -- AARQ APDU

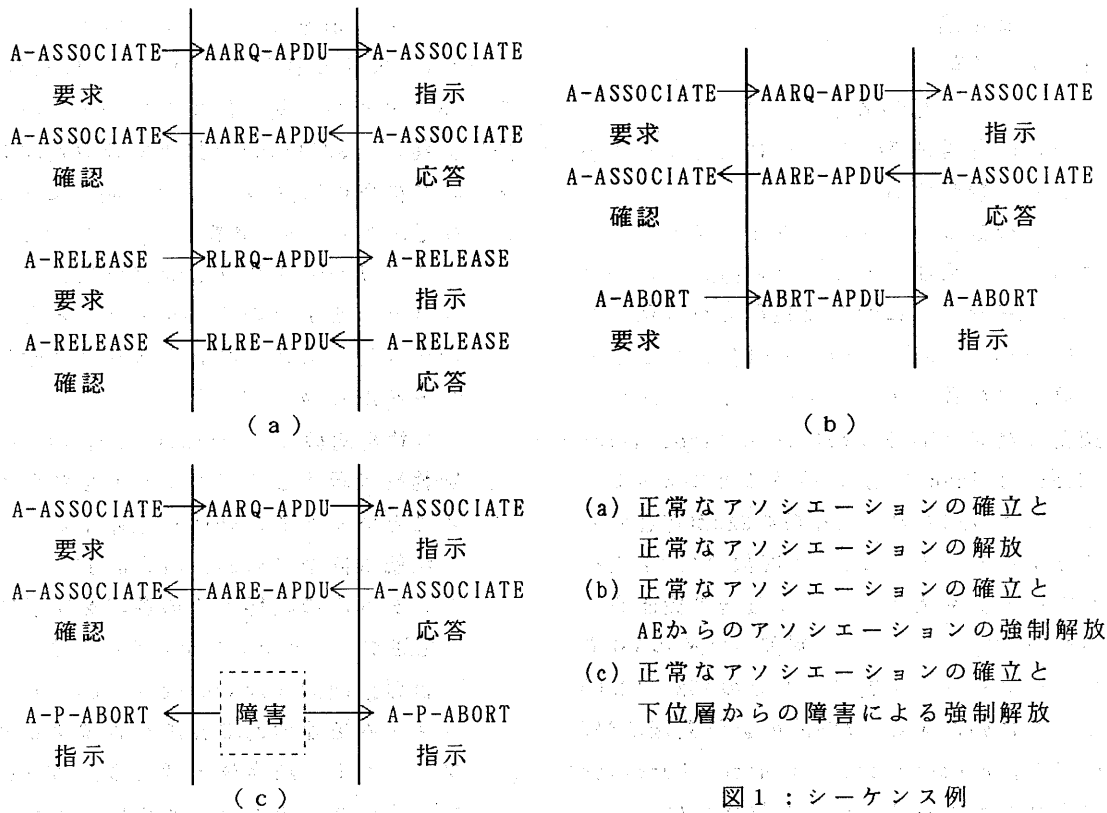
A-ASSOCIATE 応答/確認 -- AARE APDU

A-RELEASE 要求/指示 ---- RLRQ APDU

A-RELEASE 応答/確認 ---- RLRE APDU

A-ABORT 要求/指示 ----- ABRT APDU

A-P-ABORT 指示 ---- PDUは存在しない
典型的なシーケンスを図1に示す。



A E

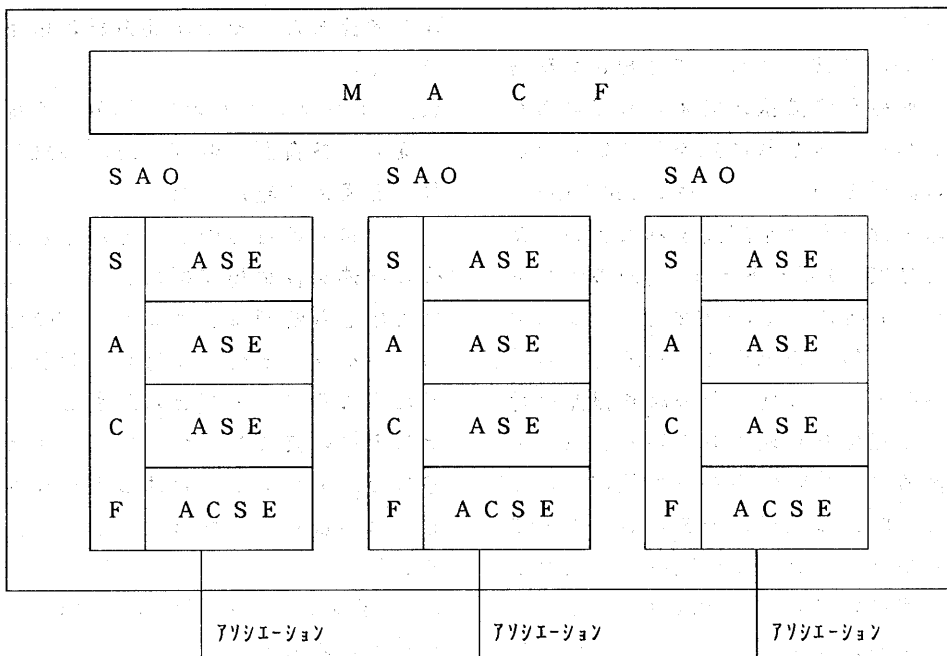


図2 : A E の内部構造

3. 形式記述

3. 1. 方針

3. 1. 1. 記述部分について

AEの内部構造を、ALS[4]に従ってもう少し詳しく説明する。

AEは複数のアソシエーションをサポートすることが可能である。その際、1本のアソシエーションに閉じた処理を行う部分をSAO (Single Association Object) として定義し、複数のアソシエーションに関する調整の機能はMACF (Multiple Association Control Function) が行うことになっている。1つのAEの中には複数のSAOが存在することがあり、1つのSAOは複数のASEを内包している。例えばファイル転送の機能を持つASEはコミットメント制御のためのASEやアソシエーション制御のためのASE (ACSE) とともに動作する。ASE間の調整機能はSACF (Single Association Control Function) により制御される。ACSEの規格はACSEというASEの振舞いを規定したものである。

以上のことを図にして示すと図2のように示される。

MACFやSACFの機能については曖昧な部分があり、明確化や拡張が考えられているが、我々はこのモデルを基に記述していった。

ACSEの記述ということでは、SAOの中のACSEの箱の部分だけを記述すればよいのだが、例えばFDTのガイドライン[5]の中のトランスポート層のLOTOS記述ではアドレスや接続ノードの割当て/管理といった、MACFやSACFのような何らかの調整機能が行うであろう処理も記述されている。我々はそれを参考にしてMACFやSACFの記述も試みた。MACFやSACFの記述はモデルから入って来るためのガイドとしての役割りを果たす程度に記述されている。ACSE自体の記述はプロトコル部分に限られている。サービスの記述も可能なら今後追加していきたい。

3. 1. 2. 記述スタイルについて

LOTOSに関する資料において、LOTOS記述の記述スタイルに関し言及しているものが多い。特に、SC21/WG7の開放型分散処理システムの標準化[6]においては、明確に次のような記述スタイルの考察が行われている。

- モノリシック型 (monolithic)
システムを1つのオブジェクトと考え、内部構造は考えず観測可能な外部動作のみを取り扱う。
- 制約指向型 (constraint-oriented)
制約を表現するオブジェクトを結合することでシステム全体を記述する。
- 状態指向型 (state-oriented)
システム内の状態とその遷移を明示的に記述することでシステムを記述する。
- 資源指向型 (resource-oriented)
システムを論理的な資源を表現するオブジェクトの結合体と考える。各資源を上述の3つのスタイルのいずれかで記述する。

我々はこうした状況を踏まえ記述スタイルを考慮した。その記述方針は以下の通りである。

(1) トップレベルのALS部分は資源指向型に従い、各資源をMACF、SAO、SACF、各ASEなどと考えて記述した。

(2) (1)の各資源の1つであるACSE ASE全体の記述も資源指向型に従った。ACSE ASEにおける各資源は、プロトコル動作のうちのシーケンス部分に注目し、明かに違うと考えられるフェーズと対応させた。そして、それぞれの資源を1つのプロセスとして記述した。またエラー処理のような例外処理も1つの資源 (プロセス) として分離し記述した。

(3) 各フェーズに対応するプロセスでは、シーケンス部とサービスプリミティブ処理部とに分けて各部分を資源と考え、資源指向型で記述した。

(4) ACSEのISOのドキュメントには既に状態遷移表が存在している。そこで、遷移表の検討も兼ねて、シーケンス部は遷移表を参考にできるところはこれをベースに記述を行った。ただし記述上は明示的に状態名等是用いていないので、状態指向型というよりはモノリシック型になっている。

(5) 一方サービスプリミティブ処理部は、汎用の処理プロセスで記述し、さらにこの汎用プロセスを詳細化するにあたっては、制約指向型記述スタイルを繰り返し使用した。

このような混合型の記述スタイルを採った利点としては、

- シーケンス部分、プリミティブ処理部分、例外処理などの各記述が独立に行える。
- 既存の遷移表との対応がとりやすい。
- 遷移表の無駄な空欄の記述に余分な労力を取られることがない。

などが考えられる。

3. 1. 3. ASN.1のライブラリ化

ACSEのPDUのフォーマットは、ASN.1[3]と呼ばれる抽象構文記法で定義されている。ASN.1は、それ自体の仕様と基本符号化規則の仕様の2つの仕様を持つ。抽象構文記法を使うと符号化したイメージを意識することなくPDUのフォーマットを規定することが可能となる。

例えばACSEの中で、ABRT APDU は次のように規定されている。

ABRT-apdu

```
::= [APPLICATION 4]IMPLICIT SEQUENCE
{abort-source [0] IMPLICIT ABRT-source,
 user-information [30] IMPLICIT
     Association-information
     OPTIONAL }
```

ABRT-source

```
::= INTEGER { acse-service-user (0),
```

```
acse-service-provider (1) }
Association-information
 ::= SEQUENCE OF EXTERNAL
```

これを見ると、ABRT APDU は abort-source と user-information の2つのパラメタの並びになっていることがわかる。ACSEはこのフォーマットに従って処理を行わなければならない。これらはACSEの規定であり、例えばファイル転送の機能を持つASEはこのPDUを受けても処理できない。しかしながら、SEQUENCE, INTEGER, SEQUENCE OF, EXTERNAL というのは、ASN.1の中で規定してある型であり、ASN.1を用いて通信を行う全てのASEはこれらの型を認識できる。

我々は、ACSEをLOTOSで記述するに当たり、これらのASN.1の中で規定されている型をライブラリ化して定義し、他の規格を記述するときにもそのライブラリを利用することを考えた。これらの型を、LOTOSの抽象データ型で定義しておけば、actualize または rename することにより汎用的に使用することができる考えたのである。

ASN.1からACT-ONEへの変換を検討した文献[7]もあるが、ASN.1の詳細な部分、例えば省略や既定値の属性等を表そうとすると、なかなかうまく記述できないことがわかった。ライブラリ化は現在まだ検討中の段階であり、本稿ではこれ以上触れない。文献[9]を参照して欲しい。

3. 2. 記述の説明

以下では実際の記述について解説する。紙面の関係もあるので特徴的な部分のみ解説をする。

3. 2. 1. ALSモデルに関する記述

この節ではプロセス記述のトップの部分から、ACSEのASEの動作を記述したプロセス ACSEASEまで、プロセスがどのようにつなが

っているかを簡単に説明する。

プロセス記述のトップは次のように記述した。

```
behaviour
  SetOfAEI[a, p]
  |[a, p]|
  ( AEIIdentification[a, p](Aempty)
  |[a, p]|
  ( AEIAcceptance[a](Ainsert(2,
    Ainsert(1, Ainsert(0, Aempty))))
  |||
  PLCAcceptance[p](Pinsert(2,
    Pinsert(1, Pinsert(0, Pempty))))
  ) )
```

プロセスSetOfAEIはAEIの機能を実行する主体である。プロセスAEIIdentificationはAEIの識別子の管理を行う。プロセスAEIAcceptanceとPLAcceptanceはAEIが同一のAE内で動作することの管理を行うためのプロセスである。Aempty、Ainsert、Aempty、Pinsert、Pemptyは、AEI識別子を表す抽象データ型に関する演算子である。

トップからプロセスACSEASEまでの記述は、図3で示されたゲートと各モジュールMACF、SAO、SACF、ASE、ACSEASEなどをLOTOSのプロセスに対応させて行なった。各モジュール内の記述は、複雑なものは小さな資源に分割して制約指向型で記述し、単純なものはモノリシック型で記述した。詳細は省略するが、複数アソシエーションの管理に関してのみ記述してある。

3. 2. 2. 主なプロセスの説明

前節ではトップからプロセスACSEASEまでのプロセスの概略を説明したが、ここではその先のプロセスの主なものを説明する。

プロセスACSEASEは次のように記述した。

```
process ACSEASE[u, l] : exit :=
  hide e, r in
  (((
```

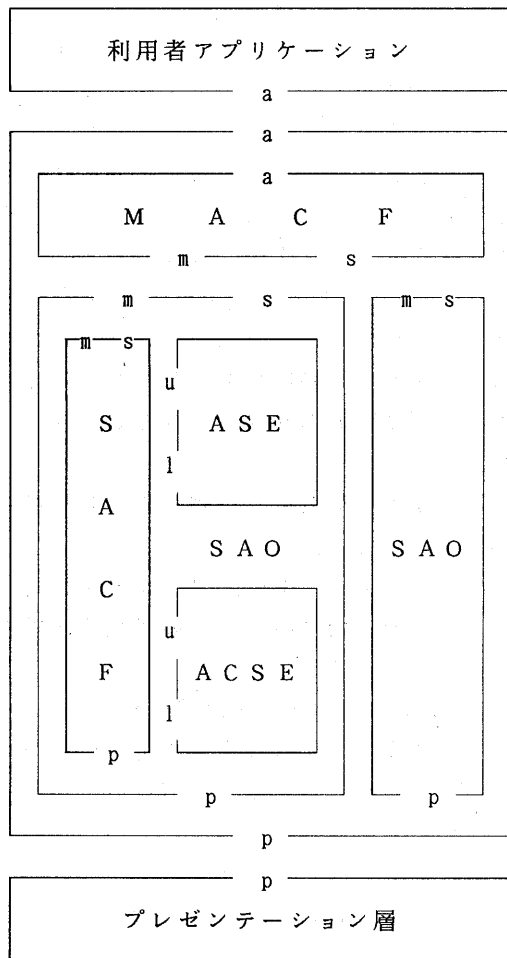


図3：ゲートとモジュールの関係

```
ACSEInvoke[u, l, e, r]
>> accept acseid:ACSEId,
      ascrole:AssociationRole,
      m:SPMode in
ACSEProcess[u, l, e, r]
  (acseid, ascrole, m) [> exit]
|[r]|
AssociationRejectCase[u, l, e, r]
  [> exit]
|[e]|
```

ErrorCase[u, l, e] [> exit]

プロセスACSEInvokeは上位からA-ASSOCIATE要求が来たときにはAssociationRoleとしてAsclnit(起動側)を、下位から

P-CONNECT指示が来たときには AscResp (応答側) を設定する。この起動側/応答側の違いにより A-RELEASE要求が衝突したときのシーケンスが異なる。また SPMode はノーマルモード/X.410モードの区別を示す。エラー時、例えば最初に A-ASSOCIATE 以外のサービスプリミティブが入ってきたような場合にはゲート e が発火される。ただしその場合の処理はローカルマターであり、プロセス ErrorCase の中では内部イベント i でそれを表している。ゲート r はアソシエーションの確立が拒否される際に発火するゲートである。メインとなるプロセスは ACSEProcess である。その中にはプロセス NormalCase と AbnormalCase があり、それぞれ正常系と異常系の処理を行い、disabling の操作子 “[>” で関係付けられている。プロセス NormalCase の中では、アソシエーション確立に関するプロセス AssociationEstablishPhase と アソシエーション開放に関するプロセス AssociationReleasePhase が enabling の操作子 “>>” で関係付けられている。これらのプロセス内は、遷移表との対応を考えながら、制約指向型とモノリシック型で記述した。

3. 2. 3. 現在の状況

本記述のポイントは既に述べたように、ALSの記述、記述スタイルの使い分けによる遷移表に対応づいた記述、ASN.1ライブラリによるデータ記述の簡易化、の3つである。

このうち、ASN.1ライブラリは先に述べたように現在開発中である。また、それ以外の部分は主にデータ記述とは独立に議論可能なので、ASN.1記述を簡易化したシーケンス部分の記述を中心に、ヨーロッパのESPRIT/SEDOSで開発されたLOTOSシミュレータ、HIPPO[8]での検証を行った。

現在ACSE ASEについては異常系(アボート、コネクション拒否など)も含めて遷移

表に対応づいた一通りの動作を確認している。

ALS部分に関しては、既に述べたように、ALS規格[4]に明確に述べられていない部分や、アソシエーションの管理やプロセスの起動の仕方など実装に密接に関係すると思われる部分もあり、一般に記述が難しい。従って、現在の記述は1つの解釈を与える感じになっている。そこで、このような解釈にあまり依存しない1本のアソシエーションを確立し解放する単純な正常系に関してのみ動作を確認している。複数本のアソシエーションの確立や異常系に関しても動作は可能であるが、ALSのモデルに従っているかどうかに関しては、今後の検討が必要である。

4. 結論/感想/課題

今回の作業を通しての結論/感想/課題を以下にまとめる。

ASN.1のライブラリ化の作業を分けて、今回の記述ではデータ型に関する処理の記述のレベルを落としたことにより、データ型の定義の記述は全体の20%程度になっている。これはHIPPOのシミュレータで動作を確認する範囲のものであり、今後データ型の定義とその処理の部分の記述は詳細化していく必要がある。

シーケンスについてはLOTOS記述の方が自然言語の記述や状態遷移表よりも理解しやすいのではないだろうか。ただし、ACSEはシーケンスが比較的簡単であり、他のより複雑なプロトコルについても検討してみる必要がある。

記述を進める上で問題となったのは、どこまで詳細化して記述すればよいかという点である。規格の範囲を越えてしまわないよう常に意識しておく必要があった。またALSの記述の部分においては、本来規格の中で規定されていなければならないもの(MA

CFやSACFの機能など)もあると考えられ、規格そのものを見直すいい機会となった。

LOTOS研究会は1989年1月に開始し1990年3月に終了した。得られた成果は、1990年度から始まった(財)情報処理相互運用技術協会内のプロトコル形式記述WGにおいて利用される。その中で処理系の作成を行う予定だが、今回の記述をその処理系に付けるサンプルとして用い、さらに変更を加えてよりよい記述にしていきたい。また他の規格の記述についても検討していきたい。

LOTOS研究会の成果の内、ASN.1のライブラリ化の作業とLOTOSの処理系の設計については論文[9]、[10]、[11]を参照されたい。

謝辞

LOTOS研究会は多くの方々の御協力により開催できた。特に、各社の関係者の方々には、LOTOS研究会に対して快くメンバを出していただいた。また、(財)情報処理相互運用技術協会には開催に対して多大の便宜をはかっていただいた。

ここに記して感謝の意を表したい。

参考文献

- [1] ISO 8807: "Information processing systems - Open Systems Interconnection - LOTOS - A formal description technique based on the temporal ordering of observational behaviour", 1989-02-15
- [2] ISO 8650: "Information processing systems - Open Systems Interconnection - Protocol specification for the Association Control Service Element", 1988-12-15
- [3] ISO 8824: "Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)", 1987-12-15
- [4] ISO/IEC 9545: "Information technology - Open Systems Interconnection - Application Layer Structure", 1989-12-15
- [5] ISO/IEC Proposed Draft Technical Report 10167: "Information Processing Systems - Open Systems Interconnection - Proposed Draft Technical Report on Guidelines for the Application of Estelle, LOTOS and SDL", 1989-02-22
- [6] ISO/IEC JTC1/SC21 N4024: "Working Document on Topic 6.2 - Formalisms and Specifications", 1989-12-11
- [7] A. Azcorra: "Conversion Rules From ASN.1 To ACT-ONE", SEDOS/C1/WP/23/M 1987-10-15
- [8] Peter H. J. VAN EIJK, Chris A. VISSERS, Michel DIAZ: "THE FORMAL DESCRIPTION TECHNIQUE LOTOS Results of the ESPRIT/SEDOS Project", 1989
- [9] 五ノ井他: "ASN.1からLOTOS ADTへの変換方法", 情報処理学会マルチメディア通信と分散処理研究会(1990) 発表予定
- [10] 辻, 佐藤他: "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", 情報処理学会マルチメディア通信と分散処理研究会(1990) 発表予定
- [11] 佐藤, 辻他: "LOTOSの汎用的な中間言語", 情報処理学会マルチメディア通信と分散処理学会(1990) 発表予定