

ASN.1からLOTOS ADTへの変換法

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)
 内山 光一 (株東芝 情報通信システム技術研究所)
 佐藤 嘉一 (沖電気工業(株) コンピュータシステム開発本部)
 小野 昌秀 (沖電気工業(株) コンピュータシステム開発本部)
 藤田 朋生 (日本電気(株) 基本ソフトウェア研究所)
 山中 顕次郎 (日本電信電話(株) NTTソフトウェア研究所)
 辻 宏郷 (三菱電機(株) 情報電子研究所)
 大蔭 和仁 (電子技術総合研究所 情報7-キテチャ7部)

ASN.1 言語で記述された仕様を, LOTOS 言語の ADT 記述に変換する手法を提案する. この手法は, 2つの部分から成っている. 一つは, ASN.1 から LOTOS ADT への変換規則. もう一つは, ASN.1 ライブラリである. 本論文では, その両方を組み合わせて使用する. それにより, 人間に読みやすい変換結果を生成する. これにより, 従来の自然言語と ASN.1 で書かれていたプロトコル定義とサービス定義の ASN.1 部分を機械的に LOTOS へ変換することができるようになる.

A conversion method from ASN.1 to LOTOS ADT

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

We propose a conversion method from ASN.1 to LOTOS ADT. The method consists of two parts. The one is a set of conversion rules from ASN.1 to LOTOS ADT. The other is an ASN.1 library. In this paper, we use those two parts together. And, our method generates conversion results which has good readability. Therefore, ASN.1 part in protocol definition and in service definition are converted mechanically to LOTOS ADT.

1. はじめに

ISO (国際標準化機構) では, OSI (開放型システム間相互接続) のサービス/プロトコル仕様の記述から曖昧さ等を排除するために, 幾つかのFDT (形式記述技法) を制定した. その中の一つに ISO 8807 LOTOS [1] がある. そして, 現在は, 様々なレイヤーのサービス及びプロトコルをFDTで記述する試みが, ISO等で活発に行われている.

INTAPのLOTOS研究会でも, ACSEプロトコルの記述実験を行っていた [3]. このような上位層の記述を行う時に問題となるのは, ASN. 1 [2] で記述された部分をいかにLOTOSで表現するか, ということである.

殆どの上位層のデータ単位記述は, ASN. 1を使用して行われている. この部分をLOTOSで記述できれば, 通信サービス/プロトコルを形式記述で表現する作業が, 簡略化される. しかし, 現在のLOTOSの機能では, ASN. 1を直接記述するのは不可能である. そこで, ASN. 1記述をLOTOS ADT^{*)}で記述するためのサポート (手順+ライブラリ) が渴望されていた.

^{*)} ADT (Abstract Data Type) : 抽象データ型. LOTOSのデータ定義部である.

2. 考察範囲

本論文で議論の対象とするのは, ISO 8824とISO 8825で規定されているASN. 1とISO 8807で規定されているLOTOSである.

ASN. 1は, 抽象構文記法と符号化規則に分かれる. 本論文では, 抽象構文記法のみを考察する. また, 抽象構文記法のうちでも, ACSEで使用しない「集合型/単一集合型/マクロ」も範囲外とした.

さらに, 本方式の最終形としては, 総て機械的に変換するコンパイラのようなものをイメージしている. 本論文ではそのプロトタイプとし, 変換手順を示す.

3. 基本方針

本章では, ASN. 1をLOTOSで記述するための, 基本的な考え方を述べる.

3.1 何に主眼を置くか

ASN. 1を利用するメリットは, 規約の一部が計算機で理解できるテキストで表現されていることである. この論文で提示する方式は, 後々のコンパイラ化を考慮した方式であり, 総て機械化できる手法を取るものとする.

ASN. 1をLOTOSに変換する時に考慮しなくてはならないのは, 可読性である. いくら機械化可能でも, 人が読めなくては意味がない. ASN. 1の変換結果がLOTOSのプロセス部で使いにくい結果では意味がない. 本論文では, 機械化可能な見易い変換方式を大前提として議論する.

3.2 基本的形態

本論文の手法によって、最終的に示されるのは、ASN.1をLOTOSのデータ部に変換する方式とASN.1の共通的な概念をまとめたASN.1ライブラリである。これらを用いることによって、ASN.1で記述された部分をLOTOSに変換していく。これらの関係を図3.1に示す。



図3.1 ASN.1の変換方式

3.3 識別子の取り扱い

ASN.1とLOTOSで大きく異なる点の一つに、識別子がある。ASN.1では、英大文字と英小文字は区別され、さらにハイフン(-)を使用する。ところが、LOTOSでは、英大文字と英小文字は区別されず、さらにハイフンは使用しない。その代わりにアンダースコア(_)を使用する。

ASN.1で書かれた識別子を機械的にLOTOSの名標に変換するのは可能である。例えば、アンダースコアをエスケープコードとして使用し、英大文字/英小文字/ハイフンを識別することは可能である。しかし、ここではこのような機械的な変換は行わない。このような変換は、数学的には美しいかも知れないが、人間の目には奇異なものとなり、可読性を阻害するだけである。

本論文では、ASN.1のハイフンをアンダースコアに変換するのみで、英大文字と英小文字はそのまま使用する。その結果、名標が重複する可能性がある。重複したら、その名標についてだけ、名標変更をするようにする。そうすることにより、可読性の低下を防ぐようにする。

また、名標が予約語と重複した時も、同様に対処する。

この手法は、ASN.1の記述者が紛らわしい名標を使用しないであろう、という推論に基づいている。

ACSE-APDU	⇒	ACSE_APDU
Name	⇒	Name
NAME	⇒	NAME

ASN.1での識別子 LOTOSでの識別子

図3.2 識別子の変換例

図3.2では、NameとNAMEが重複してしまう。このような時の対処法は後で示す。

3. 4 値域の扱いについて

ASN. 1では、取りうる値に制限を設けることができる。しかし、LOTOSでは汎用的にこの機構を表現するのは不可能である。取りうる要素の数が少なければ、列挙して表現することは可能だが、数が多くなれば不可能になってしまう。

本論文では値域に制限を設けず、値が値域内か判定する関数を設けることにより、解決する。したがって、使用する時にLOTOSのガード等と併用する必要がある。

3. 5 提供される演算子

ASN. 1からLOTOS ADTを生成しただけでは、実用にはならない。なぜなら、演算子がまったく定義されていないからである。本論文では、以下に示す演算子を標準的に定義する。

- `__eq__` : 両者が等しい時, `true of Bool`を, 等しくない時, `false of Bool`を与える。
- `__ne__` : `__eq__`の否定(not)を与える。
- `is__valid__`要素名 : 正当な内容ならば, `true of Bool`を, 正当でなければ, `false of Bool`を与える。
- `is__`要素名 : 列挙型の時のみ与えられる演算子である。その要素ならば, `true of Bool`を, そうでなければ, `false of Bool`を与える。

この他にも、要素を取り出す演算子が提供される。

3. 6 任意型の実現方式

任意型はASN. 1の型で表現されるので、ASN. 1の型を総て定義できれば、任意型を表現可能である。しかし、このような手法は可読性の点で優れているとは言えない。本論文では、無構造な型を与え、内部構造に立ち入らないアプローチを取ることとする。

4. 変換方式

本章では、ASN. 1のそれぞれの型をどのように変換していくか、個々に議論する。

4. 1 識別子と定数 (identifiers and constants)

変換の前準備を行う。対象は、識別子と定数である。定数とは、数字項目/2進列項目/16進列項目/文字列項目である。これらを、LOTOSで扱える形式に変換する。

- 識別子に含まれるハイフン (-) をアンダースコア (_) に変換する。
- 識別子が LOTOS の予約後と重複する場合 a) の変換を行う。
- LOTOS の静的意味に照らし合わせて重複する識別子は a) の変換をする。
- 数字項目を ASN. 1 ライブラリの Integer の書式に変換する。
- 2 進列項目を LOTOS 標準ライブラリの BitString の書式に変換する。
- 16 進列項目を LOTOS 標準ライブラリの HexString の書式に変換する。

この方式では、重複しないような変換が必要である。その例を以下に示す。

a) 「識別子」を「識別子__通番」に変換する。

ASN. 1 では、ハイフンが連続することはない。そのため、通番を付けることにより、一意の識別が可能になる。

4. 2 論理型 (boolean type)

LOTOS 標準ライブラリの「type Boolean」に写像する。

4. 3 整数型 (integer type)

ASN. 1 ライブラリの「type Integer」に写像する。

4. 4 列挙型 (enumerated type)

列挙型が現れる度に、新しい type を作成する。列挙値は、無項演算子 (定数値) として定義される。各列挙値を整数型にマッピングする関数を用い、Bool 関数の定義は整数値のものを流用する。

4. 5 実数型 (real type)

ASN. 1 ライブラリの「type Real」に写像する。

4. 6 ビット列型 (bitstring type)

LOTOS 標準ライブラリの「type BitString」に写像する。

ASN. 1 では、ビット列値は 16 進列を使用することができる。しかし、LOTOS で、この場合 HexString を使用すると、扱いが難しくなってしまう。そこで、総て 2 進列に変換して、対応する。

4. 7 オクテット列型 (octetstring type)

LOTOS 標準ライブラリの「type OctetString」に写像する。

4. 8 ヌル型 (null type)

ASN. 1 ライブラリの「type Null」に写像する。

4. 9 順序列型 (sequence types)

順序列型が現れる度に、新しい `type` を作成する。要素型のリストとして表現される。リストを構成する演算子とリストを分解する演算子を定義する。

順序列型を扱う時に問題となるのは、「OPTIONAL」と「DEFAULT」扱いである。「OPTIONAL」は、オリジナルの型に省略を認めた新しい型を作成することにより対処する。新しい型には、省略を示す無項演算子 (`absent`) が追加して対処する。

4. 10 単一型順序列型 (sequence-of types)

「SEQUENCE OF」は、LOTOS標準ライブラリの「`type String`」に写像する。

4. 11 選択型 (choice types)

選択型が現れる度に、新しい `type` を作成する。選択項目を引き数とする演算子のオーバーローディングとして表現する。

選択項目を結合する演算子、選択項目の種類を調べる演算子、取り出す演算子を定義する。

4. 12 タグ付き型 (tagged types)

タグ値は、演算子の値に影響する。

4. 13 任意型 (any type)

ASN. 1ライブラリの「`type Any`」に写像する。

4. 14 外部型

任意型と同様の手法でサポートする。

5. ASN. 1ライブラリ

LOTOS標準ライブラリにはないが、良く使用されるタイプをASN. 1ライブラリとしてまとめてある。その一部を5. 5に示す。

5. 1 Integer

LOTOS標準ライブラリの `NaturalNumber` の整数版を定義する。ただし、巾乗は部分的にしか定義されない。

5. 2 Real

整数のトリプレットとして定義される。抽象構文記法に基づき、基数は2と10である。なお、比較演算子等は、現在のところ不必要と思われるため用意されていない。

5. 3 Null

ただ, Null というタイプを定義する.

5. 4 Any

無構造の型を定義する.

5. 5 ライブラリ本体

ASN. 1 ライブラリの一部を示す.

```
type Integer is Boolean
sorts Integer
opns 0 : -> Integer
      Succ, Prec : Integer -> Integer
      +, -, *, **: Integer, Integer -> Integer
      =, <, <=, >, >= : Integer, Integer -> Bool
eqns forall m, n : Integer
      ofsort Integer
      Succ (Prec (m)) = m ;
      Prec (Succ (m)) = m ;
      m + 0 = m ;
      m + Succ (n) = Succ (m) + n ;
      m + Prec (n) = Prec (m) + n ;
      :

type Real is Integer
sorts Real, Base
opns 0 : -> Real
      PLUS_INFINITY, MINUS_INFINITY : -> Real
      Base2, Base10 : -> Base
      Real: Base, Integer, Integer -> Real
      Mantissa, Exponent : Real -> Integer
      Base : Real -> Base
eqns forall m, e : Integer, b:Base
      ofsort Integer
      Mantisse ( Real ( m, b, e ) ) = m ;
      Exponent ( Real ( m, b, e ) ) = e ;
      :

type Any is Boolean
sorts Any
opns some : -> Any
      another : Any -> Any
      =, <, <=, >, >= : Any, Any -> Bool
eqns forall x, y : Any
      ofsort Bool
      some eq some = true ;
      some eq another (x) = false ;
      another (x) eq some = false ;
      another (x) eq another (y) = x eq y ;
      x ne y = not ( x eq y ) ;
end type
```

6. おわりに

6. 1 結論

現在のところ、ASN. 1にはセマンティクスが与えられておらず、非形式的な言語である。ところが、LOTOSはセマンティクスが与えられている形式的な言語である。本論文を適用すると、プロトコル記述等のASN. 1部分をLOTOSに機械的に変換できる。このことは、ASN. 1にセマンティクスを与えたのと同等の意味を持つ。すなわち、この論文によって得られた変換結果は、既に形式的なものなので、数学的に扱うことができる。例えば、形式的理論に基づくプロトコル/サービスの検証やテストスイートの自動生成などに活用することが可能となる。そして、その適用範囲が、自然言語とは別個に作成された形式的記述(FD)ではなく、自然言語で記述された規約であることに、多大な価値があると考えている。

今後は、この論文で示した変換を行うプログラムを作成する予定である。

6. 2 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頂いた。また、(財)情報処理相互運用技術協会には開催に対し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。

7. 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
- [2] ISO 8824: 'Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One(ASN.1)', 1987
- [3] 内山, 藤田, 他 : "ACSEのLOTOSによる記述の試み", 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)
- [4] 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)
- [5] 佐藤, 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研究会, 1990 (発表予定)
- [6] A. Azcorra: 'CONVERSION RULES FROM ASN.1 TO ACT-ONE', 1987
- [7] Gregor v. Bochmann, Michel Deslauriers: 'Combining ASN1 support with the Lotos language', 1989