

ASN.1 ツール導入事例

インフォネクス株式会社 / ロジック株式会社
技術部 ソフトウェアグループ 課長

船橋 健

1. はじめに

当社では、H.323 を使用した VoIP ゲートウェイや VoIP ルータ等の音声通信システムの開発を行ってきており、現在いくつかの製品を市場に投入している。

音声通信システムを構成する H.323 通信プロトコル部分は、ASN.1 という抽象構文記法で記述されたメッセージを使用して装置間の通信を行う。この ASN.1 で定義された抽象構文は、実際にネットワーク上に転送する場合にバイナリイメージである転送構文に変換する必要がある。この変換を行うロジックの開発には多くの開発コストがかかる。

新たに H.323 によるソフトウェアベースの IP-PBX 機能を開発するにあたり、変換ロジックの開発コストを削減させるために ASN.1 ツールの採用を決定した。

本稿は、当社が ASN.1 ツールを導入した経緯から ASN.1 に関わる記事の執筆を依頼されたものである。よって、簡単ではあるが「ASN.1 ツールの役割、その導入効果、ツール適用結果」について紹介することとする。

2. ASN.1 ツールの役割

ASN.1 とは、Abstract Syntax Notation One: 抽象構文記法 1 の略語であり、データ構造を記述するための一種のプログラミング言語である。

ASN.1 は、コンポーネントと呼ばれる基本的な単位を組み合わせることで、複雑なデータ構造を表現することができる。コンポーネントを組み合わせたデータ構造をオブジェクトと呼ぶ。

ASN.1 を使用することで複雑なデータ構造を持った情報の交換に、相互運用性を持たせることが可能となる。

ASN.1 コンポーネントの目的は情報の定義であり、情報の型と情報の値から定義される。

情報の型には、INTEGER、OCTET STRING、BOOLEAN といったコンピュータの内部表現に近いものと、SEQUENCE、CHOICE、OPTION といったコンポーネントの集合を表記する型と（各コンポーネントの出現順序、選択、有無に意味を持たせたい時に使用する）および定義済みの他の ASN.1 オブジェクトを使用することもできる。

これらは C 言語の構造体をイメージす

ると分かりやすい。ASN.1 オブジェクトの構造はユーザーが独自に定義することができる。

ASN.1 は抽象的なデータ構造を表現する言語なので、ASN.1 で記述された構文（抽象構文と呼ぶ）は、ネットワークで転送するためのバイナリデータ（転送構文と呼ぶ）に変換する必要がある。

抽象構文と転送構文間の変換規則は、ASN.1 で符合化規則として定義されており、BER (Basic Encoding Rules)、PER(Packed Encoding Rules)、および

その他いくつかの符合化規則が存在する。

なお、ASN.1 処理系の範疇は、抽象構文の定義と抽象構文と転送構文間の変換が責任範囲となる。

この符合化規則の説明は省かさせていただくが、この符合化規則の変換ロジックの開発こそが開発コストから見た場合に大きな負荷となる。ASN.1 ツールは、この変換ロジックを実装した関数のソースコードを生成するものである。

【例：手紙の情報を ASN.1 で記述した場合】

```
Letter ::= SEQUENCE {
    opening      OCTET STRING (SIZE(0..80)),
    body         OCTET STRING (SIZE(1..4294967295)),
    closing      OCTET STRING (SIZE(0..80)),
    receiverAddr AddressType OPTIONAL,
    senderAddr   AddressType OPTIONAL
}
```

【例：手紙のアドレス情報を ASN.1 で記述した場合】

```
AddressType ::= SEQUENCE {
    name         OCTET STRING (SIZE(1..20)),
    number       INTEGER (0..60000),
    street       OCTET STRING (SIZE(1..256)),
    apartNumber  INTEGER (0..100) OPTIONAL,
    postOffice   OCTET STRING (SIZE(1..80)),
    state        OCTET STRING (SIZE(1..80)),
    zipCode      INTEGER (1..65535)
}
```

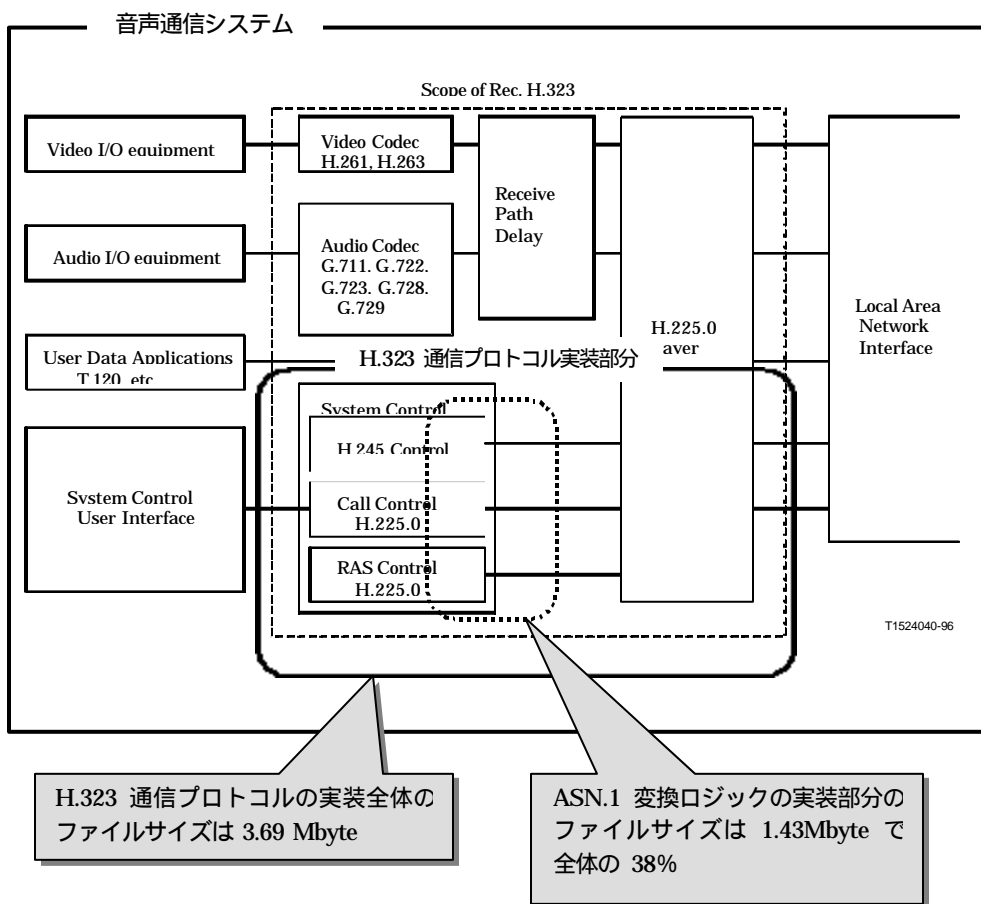
3 .ASN.1 ツールの導入効果

ASN.1 ツールの導入効果を示すために、ASN.1 変換ロジック実装部分に費やしたソースコードのファイルサイズを示す。

音声通信システムの中から H.323 通信プロトコル実装部分を選択し、その

H.323 通信プロトコル実装部分のファイルサイズと、その中の ASN.1 変換ロジック実装部分のファイルサイズを示す。

これらのファイルサイズと割合から、開発コストの削減の効果を推測できることと思う。



4 .ASN.1 ツール適用結果

(1) ツールの不具合

ASN.1 ツールの不具合の傾向は、Extention Marker 以降のメンバ（拡張メンバ）の取り扱いが勧告書に準拠していないものであった。実際発生した不具合は下記の2件である。

1 件目：CHOICE の拡張メンバ選択時の符号化処理

CHOICE の拡張メンバ選択時の不具合とは、本来拡張メンバが選択された場合、Extention Marker が設定され、さらに Extention Marker 以降、何番目のメンバ設定されたかを指定しなければならない。（例 Extention Marker 後の2番目のメンバが選択された場合、1 と符号化されなければならない。）しかしながら、ASN.1 ツールでは CHOICE 全体の通算で符号化されていた。

2 件目：拡張メンバの符号化処理

拡張メンバの不具合とは、本来、拡張メンバが使用される場合、レングス + メンバの値という形で符号化されなければならない。しかしながら、ASN.1 ツールではメンバの値のみ設定され、レングスが設定されていなかった。

注意すべき点としては、このような不具合は符号化と復号化の対で同一の不具合を持つため、ASN.1 ツールで作成したプログラム間の対向通信では発見できないこと。また、開発上 ASN.1 の符合化規則についての知識を持っている必要が

あることを付け足しておく。

(2) ユーザサポート

質問や不具合修正の依頼に対するユーザサポートについては、すべて2日以内に迅速に回答、不具合修正に応じていただいたことに感謝している。

マニュアルについては、ツールが生成したコードの入出力仕様の使用方法がわかり難いといった感想を持っている。

(3) ツール使用上の注意事項

- ASN.1 ツールが生成したコードのメモリ消費量

ASN.1 ツールが生成したコードのスタック消費量、および符号化/復号化に使用する一時的なメモリ使用量に注意しなければならない。当社事例における評価結果では、スタック消費量は18KBytes、メモリ使用量は10KBytesであった。

- 一時的なメモリを使用することによるオーバヘッド

符号化/復号化を行う都度、転送構文で表現されたメッセージと抽象構文で表現されたメッセージ間で一時的なメモリを経由しなければならないため、転送効率やCPU使用率にクリティカルなシステムを開発する場合は注意しなければならない。

- ASN.1 オブジェクトの再帰定義
ASN.1 オブジェクトが再帰定義さ

れた抽象構文は、ASN.1 ツールにてコードの生成ができない。対処方法としては、再帰定義されたデータ構造を使用する必要がない場合は、抽象構文を書き直すことで対処できる。

(4) まとめ

ASN.1 の符合化規則の変換ロジックの開発は開発コストから見た場合に大きな負荷となる。よって「ASN.1 ツール」等の活用を検討することを強く推奨する。