

ASN. 1 サポートツールの一方式

5U-6

P A S T E L の構成

西本 一志、堀坂 七美、桧垣 伸俊、山下 邦彦

松下電器産業(株) 情報システム研究所

1. はじめに

近年OSIの標準化が急速に進み、多種多様なアプリケーションが規格化されつつある。OSI応用層プロトコルのPDU定義にはASN. 1⁽¹⁾抽象構文記法を用いる。しかし、ASN. 1抽象構文記法は複雑な構造を持つのでPDU処理プログラムの作成が困難である。そこで従来からさまざまなASN. 1サポートツールが提案されている⁽²⁾⁽³⁾。従来のツールは一般にPDU構造をC言語の構造体で記述し、この構造体内の領域あるいは構造体内のポインタが指し示す領域に実際の値データを格納したものをを用いてPDU処理を行っている。だが、複雑でネスト構造の深いPDU構造を構造体で記述すると、特に工夫をしなければ構造体のサイズが非常に大きくなる。実際に郵政省電子メール通信端末アクセス推奨通信方式(JUST-MHS)のPDUを構造体で記述したところ94Mbyteにもなった。また構造体による記述は、リンクされるデータ領域がメモリ上に分散して一括的な扱いができず、メモリ管理が困難で障害の原因となりやすい。

本論文では、C言語の構造体を使用せず、動的なデータ構造を用いることでアプリケーションをコンパクトかつ容易に開発することができるASN. 1サポートツールPASTEL (Panasonic ASN.1 Support Tool for Effort-Less implementation of OSI applications)の実現方式について報告する。

2. PASTELの実現方式

今回のASN. 1ツールPASTELの開発にあたり、

- ・アプリケーションをコンパクトに作れること。
- ・アプリケーションを簡単に作れること。

の2つを目標に、次のような実現方式を採用した。

- (1) PDU構造情報から値データ領域を切り離してPDU構造表として独立させる。
- (2) 値データ領域は必要に応じて必要なだけ確保し、かつ必要に応じて互いに関連付ける動的データ構造とする。
- (3) 複数の独立したメモリ環境を操作できるメモリ管理機能を実現してPDU構造表と値データ領域を管理する。
- (4) 各値データのPDU構造に沿った関連付けを自動的に行う。

方式(1)によりPDU構造記述の重複を避けられる。PDU構造情報に値データ領域がリンクされていると、同じ構造を持つプロトコル要素の出現箇所全てにその構造を記述しなければならない。しかも、そのPDUに対する一連の処理が終わるまでPDU構造情報が占有されるので、同じ構造を持つPDUを同時に複数処理するにはPDU構造情報を複数持つなどの工夫が必要であった。

PASTELにおけるPDU構造表の例を図1に示す。

```
A ::= SET{
    B [0] CHOICE{
        c OCTETSTRING,
        e BITSTRING,
        ...}
    d INTEGER OPTIONAL}
```

表 1

A	S E T
B	C H O I C E
c	O C T E T S T R I N G
d	I N T E G E R
e	B I T S T R I N G

表 2

B	[0] 必須
d	OPTIONAL
c	
e	

図1 PDU構造表の例

図1において表1はA、Bなどの各プロトコル要素の属性情報 (SETやCHOICEなどのタイプなど) を記述している。表2は構文型のプロトコル要素の構成要素とその属性情報 (タグやOPTIONAL, IMPLICIT指定など) を記述している。

(2)により値データのための領域は最小限にできる。PDU構造情報と値データが一体の場合、全プロトコル要素の値領域、あるいは値領域へのポインタ領域をあらかじめ静的に確保する必要があり、使用されない領域が多く存在した。本方式では値データ領域は動的に確保・関連付けが行われ、必要最小限のPDU構造をもつ値データ表として格納されるので、無駄な領域が発生しない。

(3)は、従来メモリ上に分散していたデータ領域をひとまとめにしてひとつのメモリ環境として扱い、この環境単位で一括操作することでメモリ管理に起因する障害を抑制するものである。さらに、環境単位での受渡しやファイルへの退避、ファイルからの読み出しも行えるのでアプリケーションのマルチプロセス化を容易に行える。またメモリ環境は複数設定することができるので、PDU構造情報と値データの分割管理や複数の値データ表の操作を容易に行える。

A Design of ASN.1 Support tool "PASTEL".
K.NISHIMOTO, N.HORISAKA, N.HIGAKI, K.YAMASHITA.
Matsushita Electric Industrial Co., Ltd.

(4)は、必要な末端プロトコル要素(すなわちその配下に構造を持たないプロトコル要素)の値データだけを登録すれば、PDU構造表を参照しながら上位構造情報を付加して値データ表をPDU構造に則って組み上げる機能を提供することによって、アプリケーションのプログラミングを簡素化するものである。図2は図1で例示したプロトコルを使用してデータを転送構文に変換する処理の流れを示したものである。プロトコル要素cとdを

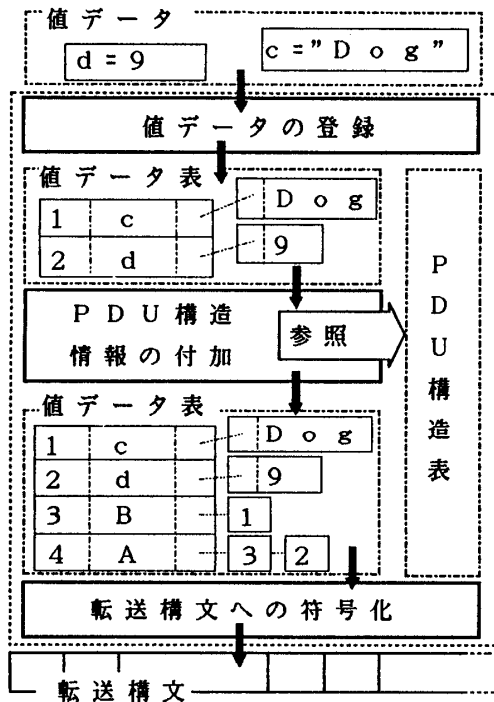


図2 PDU処理の流れ

使用する場合、プログラマは要素cがA下のB下にあるというPDU構造を認識せずに、

c = "Dog"

というようにcの値データを登録するだけでよい。dについても同様である。必要な値データの登録が終わると、PDU構造表を参照して値データ表にPDU構造情報を付加する。つまり値データ表上に要素BおよびAの項目が追加され、かつ各要素の関連付けが行われる。このようにしてPDU構造に則って構成された値データ表を符号化し、転送構文を出力する。

より一般的なPDU構造例として、上記プロトコルに

X ::= SET { ..., d, ... }

Y ::= SEQUENCE { A, X }

のようなPDU定義を追加した場合、dの値が登録されてもコーデックはそれがAの配下のもかXの配下のもかを明示的に指定されない限り区別できない。そこで、この解決のためにドメイン分割の概念を導入した。これ

はPDUを構成プロトコル要素に重複が無いようにいくつかのドメインに分割し、各ドメインに値データを登録してドメイン単位でPDU処理を行うことにより前述のような混乱を避けるものである。上記の例ならばYをAとXの2ドメインに分割すればよい。これによってPDU構造を意識したプログラミングを最小限に抑えられる。

3. PASTELの構成と機能概要

PASTELはASN.1コンパイラと汎用コーデックの2つからなる一般的構成を持つ。

(1) ASN.1コンパイラ

ASN.1記法で記述したPDU定義テキストファイルをC言語define文で記述されたプロトコル要素識別子一覧とバイナリ形式のPDU構造表ファイルに変換する。

(2) 汎用コーデック

PDU構造表のロード、必要なプロトコル要素の値データの登録、登録した値データへのPDU構造情報の付加、値データ表から転送構文への符号化、受信した転送構文の値データ表への変換、値データ表のPDU適合性検査、値データ表からの値データ取り出し、値データ表の一部または全部の外部記憶域への退避、外部記憶域からの復元、削除機能を実現している。

また、値データ表やPDU構造表内の各要素へのアクセスは、ポインタを使わず、値データ表上の各値データに割り振られた値識別子やASN.1コンパイラが出力したプロトコル要素識別子によって行われる。

4. 評価

PASTELを利用してMS-DOS上にJUST-MHSを実装した。PDU構造表は約8Kbyteであった。また一つの宛先に対する通常の発信処理の場合、値データ表は末端プロトコル要素の値データ領域をのぞいて約4Kbyteであった。両者を合わせて約12Kbyteであり、従来の方法に比べて必要となるメモリは格段に削減された。さらにメッセージ管理・メッセージ分配・端末アクセスの3プロセスで構成するメールボックス機能の実現にあたり、プロセス間のPDU構造表の共用と値データ表の受渡しを容易に行うことができた。

5. おわりに

汎用性、携帯性、プログラミングの容易性を狙った応用層プロトコルの開発サポートツールの実現方法を示した。今後は各種プロトコルへの適用性評価とASN.1フルセットのサポートを行っていく予定である。

参考文献

- (1)ISO:"ASN.1",IS 8824/8825 (1988).
- (2)T.Nakakawaji,etal:"DEVELOPMENT AND EVALUATION OF APRICOT ...",ISIIS'88,pp.55-62(Nov.1988).
- (3)Y.Ohara,etal:"ASN.1 Tools for Semi-automatic Implementation ...",ISIIS'88,pp.63-70(Nov.1988).