ＯＳＩ応用層プロトコル処理ソフト自動生成のためのＡＳＮ．１処理系

5E-4

大原康博(*)　菅沼知久(*)　千田昇一(**)
(*)ＮＴＴ情報通信処理研究所
(**)ＮＴＴネットワークシステム開発センタ

1．まえがき

近年OSI応用層ﾌﾟﾛﾄｺﾙの標準化とそれに伴う製品開発が進展している．ASN.1（Abstract Syntax Notation-1：抽象構文記法1）はOSIﾌﾟﾛﾄｺﾙ仕様規定で扱うPDUの構造を厳密に定義する言語であり各応用層ﾌﾟﾛﾄｺﾙで広く使われている．製品開発の観点から見たときASN.1の処理は，普及度の点で重要であると同時に，その形式性を利用して仕様からｿﾌﾄの自動生成により開発の効率化が図れる点で実用的効果が高い．本稿は，この度開発した『ﾊﾟｿｺﾝ等上で既存の下位層を利用した応用層処理ｿﾌﾄを半自動的に実装するためのASN.1処理系』について，その特徴および適用例（ROS）を示している．

2．ＡＳＮ．１処理の概要

OSIでは各応用ｴﾝﾃｨﾃｨ（AE）から抽象構文で渡されたPDUをﾌﾟﾚｾﾞﾝﾃｰｼｮﾝ層で転送構文に変換する構造を規定している．しかしPDU構造は各応用層ﾌﾟﾛﾄｺﾙにより異なるため，ASN.1処理（符号化/復号化）は，各AE毎に異ならざるを得ない．従って処理方法として次の代替案が考えられる．

（案１）各AEで個別処理を行う．

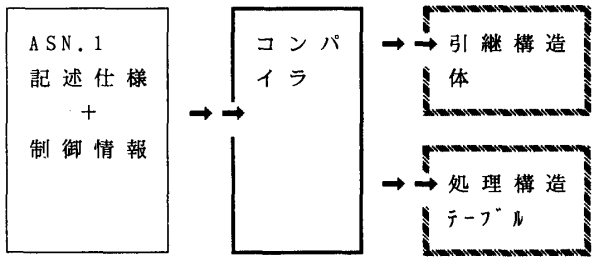
（案２）ASN.1処理を汎用的に行い，各AEでそれを共通に利用できる構造にする．

案１は特定のPDU構造にﾁｭｰﾝした最適な内部ﾃﾞｰﾀ表現が可能であるが，AE数だけ類似のASN.1処理を行うのでｿﾌﾄ規模が増大する欠点がある．案２は汎用的に処理するためのｲﾝﾀﾌｪｰｽが必要となりﾃﾞｰﾀ表現のためのﾒﾓﾘ使用効率が若干落ちる可能性があるが，今後のﾌﾟﾛﾄｺﾙ追加への柔軟性・発展性がある．従って案２を採用する．

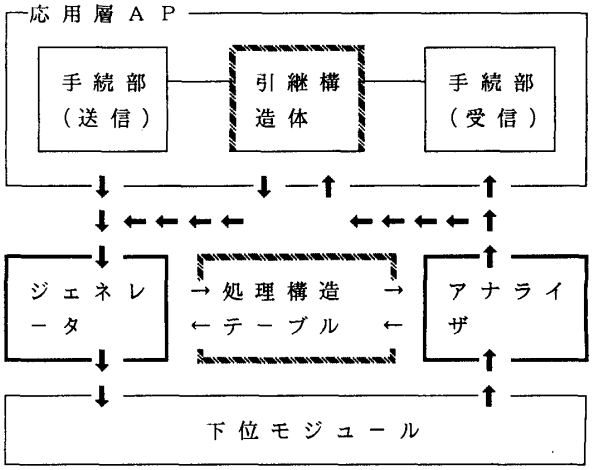
案２の処理は，①各AEに対応するASN.1記述内容を反映したﾃｰﾌﾞﾙ形式に変換する，②またAEの手続き部で送信PDU値/受信PDU値を設定でき，符号器／復号器に対して処理要求を渡すｲﾝﾀﾌｪｰｽを用意する，③汎用的符号器/復号器がこのﾃｰﾌﾞﾙを参照して符号／復号化を行う，形に詳細化できる．今回開発した処理系は，①②を「ASN.1ｺﾝﾊﾟｲﾗ」，③を「ｼﾞｪﾈﾚｰﾀ/ｱﾅﾗｲｻﾞ」部分で実現している（図１）．

ｺﾝﾊﾟｲﾗは，「各AEのASN.1によるPDU記述＋ｼﾞｪﾈﾚｰﾀ/ｱﾅﾗｲｻﾞに対する制御指示を与えるための情報」を入力とし，「PDUのトリー構造情報と付加制御情報を含む制御表（処理構造ﾃｰﾌﾞﾙ）」，および「送受信PDUの値を設定しｼﾞｪﾈﾚｰﾀ/ｱﾅﾗｲｻﾞに渡す領域（引継構造体）」を出力する．

ｼﾞｪﾈﾚｰﾀは引継構造体に設定されたPDU値を，処理構造ﾃｰﾌﾞﾙを参照しながら転送構文に符号化して，下位ﾓｼﾞｭｰﾙに渡す．ｱﾅﾗｲｻﾞは逆に下位ﾓｼﾞｭｰﾙから転送構文形式の受信PDU値を引継ぎ，処理構造ﾃｰﾌﾞﾙを参照して復号化し引継構造体に変換して上位ﾕｰｻﾞに渡す．



（ａ）コンパイラ



（ｂ）ジェネレータ／アナライザ

図１　ＡＳＮ．１処理系の構成

3．ＡＳＮ．１処理の問題点

応用層で扱うﾃﾞｰﾀ構造はｾｼｮﾝ層以下のﾃﾞｰﾀ構造に比べ複雑であり単純なｵｸﾃｯﾄ列のみでは表しきれない．このためASN.1では，SET型（順序づけのないﾃﾞｰﾀ集合），CHOICE型（ﾃﾞｰﾀ集合の中からの１要素選択）等抽

ASN.1 Tools for automatic-implementation of OSI application layer protocols
Yashhiro OHARA, Tomohisa SUGANUMA, Shoichi SENDA
NTT

象性の高い構造記述を可能としているが反面，処理上次のようないくつかの難しさを生じる．

（１）ASN.1仕様の不確定性に関するもの

［問題1］符号化/復号化処理をﾕｰｻﾞから引継ぐPDUの単位が規定されていない．

［問題2］要素数が不定である集合的ﾃﾞｰﾀを処理する場合，処理可能な最大数を決める必要がある（SET OF型，SEQUENCE OF型）．

［問題3］構造型PDUの時不定長形式の符号化対象PDUをﾕｰｻﾞが選択する必要がある．

（２）符号化/復号化処理効率に関するもの

［問題4］ﾚｲﾔ間で処理の独立性を確保する処理ｿﾌﾄでは，各ﾚｲﾔの処理が逐次的に行われ（例：PDU全体を符号化し終わって下位に渡すなど）処理時間が増大する傾向がある．特にASN.1処理は応用層の長大ﾃﾞｰﾀを扱うため性能低下が起こり易い，など．

4．コンパイラ制御情報

上記の問題点を解決するためには，いくつかの方法が考えられる．

（案１）処理の不確定性が起こった箇所でｲﾝﾀﾗｸﾃｨﾌﾞにﾕｰｻﾞに問い合わせて確定させる．

（案２）付加情報を入力記述の中で与える．

案1は実用的な処理性能が得られないこと，から案2を採用する．本来の仕様記述に対する影響を最小限に抑えるため，ASN.1ｺﾒﾝﾄ文を拡張した形式(--*)で与える（表１）

表１　主な制御情報

分類	制御情報と使用例
処理の確定化	--* ENCODE　符号化対象範囲の指定 例：--*ENCODE A::=INTEGER
	--* INDEFINITE　指定PDUを不定長符号化する 例：--*INDEFINITE A::=SEQUENCE {INTEGER,IA5String}
処理の効率化	--* FIXED VALUE=xx　ﾃﾞﾌｫﾙﾄ値を用いた符号化 例：--*FIXED VALUE=5 A::=INTEGER

5．ジェネレータ／アナライザの処理方式

引継構造体は木構造ﾃﾞｰﾀをC言語の構造体に変換したもので転送ﾃﾞｰﾀを入れるnode域を指すﾎﾟｲﾝﾀの集合である．処理構造ﾃｰﾌﾞﾙは型ﾃｰﾌﾞﾙ，制御情報ﾃｰﾌﾞﾙ等から成る制御表（Cの2次元配列）である．ｼﾞｪﾈﾚｰﾀの符号化処理の概要を次に示す．

［ｽﾃｯﾌﾟ1］Identifierの作成

符号化する型に対応する型ﾃｰﾌﾞﾙに登録されているid,classからIdentifierを求める．構造型の場合型ﾃｰﾌﾞﾙの子供/兄弟指標によりﾁｪｰﾝをたどりその要素のIdentifierを作成する，等

［ｽﾃｯﾌﾟ2］Lengthの作成

①ﾌﾟﾘﾐﾃｨﾌﾞ型の場合引継構造体が指すnodeのﾃﾞｰﾀ長を求める（固定長），

②構造型で不定長符号化が指定された場合，Length値をX'80'とし，引継構造体が指す各nodeに対応する型のIdentifier，Lengthを求めnodeﾁｪｰﾝする．終わりにX'0000'を作成しnodeﾁｪｰﾝする，等

［ｽﾃｯﾌﾟ3］Contentsの作成

①引継構造体が指すnodeをﾁｪｰﾝすることにより，転送ﾃﾞｰﾀをﾊﾞｯﾌｧに作成する，

②FIXED VALUE指定がある場合，引継構造体中の対応する要素にﾃﾞｰﾀがない時は処理構造ﾃｰﾌﾞﾙ（定数ﾃｰﾌﾞﾙ）の値を用いる，等

また，本ｼﾞｪﾈﾚｰﾀは符号化だけでなく，上下位ﾓｼﾞｭｰﾙとの標準的通信ｲﾝﾀﾌｪｰｽ(send/recv)を用意しており，既存のｾｼｮﾝ層処理ｿﾌﾄとﾘﾝｸして応用層処理ｿﾌﾄを開発できる特徴をもつ．ﾕｰｻﾞはｼﾞｪﾈﾚｰﾀを呼び出すことにより符号化を指示することができる．ｼﾞｪﾈﾚｰﾀは，指示により上に示した符号化を行い，下位ﾓｼﾞｭｰﾙにﾃﾞｰﾀを転送し，制御をﾕｰｻﾞに戻す．ｱﾅﾗｲｻﾞの復号化処理も，ｼﾞｪﾈﾚｰﾀと同様である．

6．具体的適用例および評価

本処理系はCで作成しUNIX及びMS-DOS上で動作する．規模はｺﾝﾊﾟｲﾗが約4K，ｼﾞｪﾈﾚｰﾀ/ｱﾅﾗｲｻﾞが各約2Kｽﾃｯﾌﾟである．具体的にROS（ﾘﾓｰﾄｵﾍﾟﾚｰｼｮﾝｻｰﾋﾞｽ）に適用した結果，250行のASN.1定義文から170個のC構造体（400行のｿｰｽｽﾃｰﾄﾒﾝﾄ）を生成した．これは人手作成に比べ規模/ﾒﾓﾘ域が2-3割増加する程度で，開発の迅速化の効果の方が大きい．

7．あとがき

本処理系は応用層ﾌﾟﾛﾄｺﾙのﾃﾞｰﾀ記述部分のみ処理可能であり人手作成の手続き部とリンクする必要がある．手続き部を含む完全な処理の自動化は今後の課題である．

参考文献

(1)姉崎，"ASN.1ﾊﾝﾄﾞﾗの設計と実装"，ﾏﾙﾁﾒﾃﾞｨｱ通信と分散処理研資36-1，1988.2

(2)中川路他，"ASN.1ﾂｰﾙAPRICOTの設計"，情処62年後期全大，533