

X. 409コーディング処理法の検討

2T-5

花木 三良 本村 公太
(NTT 電気通信研究所)

1. まえがき

CCITT X. 409は、アプリケーションレイヤのエンティティ間で交換されるデータの記述法、およびコーディング方法(転送構文)に関する勧告であり[1]、MHSのプロトコルやミクストモードの文書構造などの記述に使用されている。

X. 409コーディング処理法に関して従来、幾つかの方式[2-4]が検討されているが、本稿ではアプリケーションのデータ項目毎に処理ロジックを作成する方式、および汎用的なロジックとポインタテーブルによる方式について検討する。

2. X. 409のコーディング規則

X. 409のコーディング規則では、各データ要素を次の3フィールドからなるオクテット列で表現する。

- ①タイプ：データ要素を識別する識別子
- ②長さ：データ要素の長さ(オクテット数)
- ③値：データ要素の内容

1つのデータ要素の値フィールドに他データ要素のタイプ、長さ、値のフィールドをネストさせることも許される。

ネスト構造をもつデータは木構造で表現することができ、X. 409コーディングでは、木のノードはデータ要素を、木の枝はデータ要素間のネスト関係を示す。データ要素の記述において“::=”の左項を親ノード、右項をその子ノードとする。また枝は順序型、集合型あるいは選択型などのX. 409の型の属性をもつ。木構造で末端にあるデータ要素(葉に相当する)の値がアプリケーションのデータ項目の値となる。木構造はアプリケーション対応に存在する。図1に例を示す。

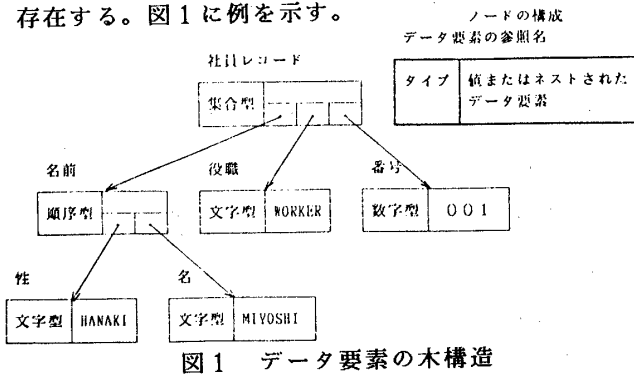


図1 データ要素の木構造

3. X. 409コーディング処理法

X. 409処理概念を図2に示す。アプリケーションの実用性を重視したX. 409処理部の構成では拡張性、処理速度などが要求される。

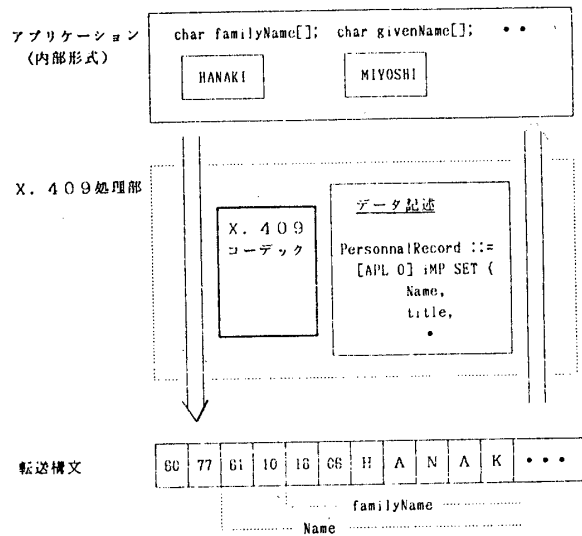


図2 X. 409の処理概念

3.1 処理法の検討

X. 409コーディング処理は、X. 409データ記述法に基づくアプリケーションデータの記述から構築される木構造を、枝の属性に基づいて探索し、アプリケーションのデータ項目の値を符号化・復号化する処理である。木構造の探索法において、

方式①：木構造ロジック方式

探索対象となる木構造の構造をもつ探索ロジックとする方式

方式②：木構造テーブル方式

探索ロジックは枝の属性(X. 409の型)による処理とし、探索対象となる木構造はデータとしてデータ要素テーブルにもつ方式

について検討する。

方式①は、アプリケーションのデータ要素毎に処理サブロジックを作成し、データ要素の木構造に応じて処理を分岐する方法である。ルート of データ要素から下向きに探索していく。探索ロジックの例を図3に示す。

```

PersonnalRecord()
/* 社員レコードの処理プログラム (以下同様) */
{
    set ( Name(), title(), no() );
}
Name() /* 名前 */
{
    seq ( familyName(), givenName() );
}
title() { /* 役職 */ }
no() { /* 番号 */ }

familyName() { /* 姓 */ }
givenName() { /* 名 */ }

set( argument_list ){ /* 集合型の処理 */ }
seq( argument_list ){ /* 順序型の処理 */ }
    
```

図3 探索ロジック (方式①)

方式②は、アプリケーション対応のデータ要素テーブルを参照しながら探索処理を行う方式である。データ要素テーブルの基本構成を図4に示す。データ要素テーブルのポインタによるリンク構造が、アプリケーションのデータ要素の木構造に対応する。テーブルの項目には必要に応じて不定長サポートのフラグ、データ項目の最大長などの制限条件を追加する。データ要素テーブルは、符号化・復号化で同一のテーブルを共用することができる。

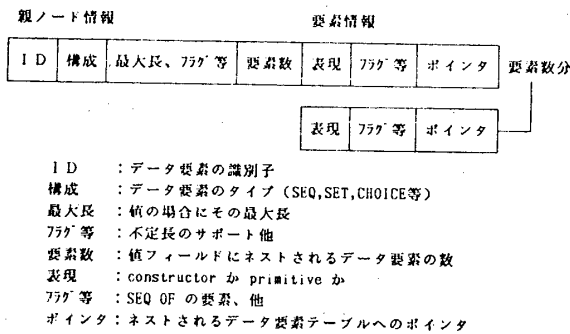


図4 データ要素テーブルの構成 (方式②)

探索ロジックは、構成フィールドの値に応じて処理を分岐する。探索ロジックには、
 (7)スタックを利用する再帰的なロジック、
 (イ)データ要素テーブルに親ノードへのポインタをもつ2重リンク構造として順次的に処理するロジック
 などが考えられる。
 テーブルサイズは親ノードへのポインタの分だけ(イ)が大きく、探索ロジックのプログラム規模はスタック操作の分だけ(7)が大きい。

3.2 評価

(a)拡張性

方式①は探索ロジックそのものがアプリケーションに依存するものであり、拡張性は小さい。方式②では、アプリケーションに依存する部分はデータ要素テーブルに限られ、探索ロジックは任意のアプリケーションに適用可能であり拡張性に優れる。

通信プロトコルのインプリメントにおけるサービスサポート範囲の拡大などデータ項目の追加・変更が多いと考えられるアプリケーションでは、方式②が適する。

(b)プログラム規模および処理速度

各処理方式のプログラム規模、処理速度などを評価するために方式①、方式②(7)の処理プログラム(復号化部分:但し、パラメータの抽出のみ)をC言語により試作した。表1に試作結果を示す。

表1 処理方式の試作結果

| | 方式① | 方式② |
|---------|---------|-------------------------|
| プログラム規模 | 探索部 1.5 | テーブル部 1 探索部は一定規模 |
| 処理時間 | 探索部 1 | 探索部 0.7 テーブル初期化部 0.7 |

(数字は比を示す)

- プログラム規模: 方式①の探索ロジック部および方式②のデータ要素テーブル部は、アプリケーションのデータ項目数に比例する。方式②の探索ロジック部分は、アプリケーションには依存せず約200ステップである。

アプリケーションのデータ項目数が少ない時には方式①が、多い時には方式②が適する。

- 処理時間: 方式②のデータ要素テーブルの初期化に要する時間を除き、探索時間のみを比較すれば、方式②のほうが小さい。

4. あとがき

X.409コーディング処理法について木構造データの探索法の検討により、データ要素テーブルの参照による方式が拡張性、処理時間などの点で優れていることが分かった。今後は、処理方式の詳細化を進める。

[参考文献]

- [1]CCITT X.409 MHS:Presentation Transfer Syntax and Notation
- [2]昭61信学総全大 No.1773 8-3
- [3]昭60信学総全大 No.1779 8-15
- [4]昭59信学通信全大 No.205 1-205