EDI用汎用トランスレータの設計

1M−1

杉山敬三　小花貞夫　鈴木健二

KDD研究所

1. はじめに

近年，帳票等の商取引に関する情報を異企業のコンピュータ間で直接交換するEDI(電子データ交換)が注目されており，筆者らもパソコンをベースとし，独自プロトコルによる企業内EDIシステムとEDI標準による企業間EDIシステムを相互に接続するパソコンEDIシステムを開発している[1]．このようなEDIシステムの実現には，企業内で用いる独自のデータフォーマットと企業間で用いる標準のデータフォーマット間の変換を行うトランスレータが必要となる．しかしながら，EDI標準は，国際標準に限らず国内標準や業界標準など様々なものが存在するため，トランスレータは種々の標準に柔軟に対応できる汎用性が要求される．そこで本稿では，種々のEDI標準への対応が容易なEDI用汎用トランスレータの設計について報告する．

2. EDIにおける標準とトランスレータの概要

EDIのデータフォーマットの標準は，一般にシンタックスルール，標準メッセージ，データエレメントディクショナリの3つの要素で構成される[2]．シンタックスルールは，日付や注文番号などのデータエレメントからEDIメッセージを組み立てるための文法規則であり，通常図1に示すように階層的に規定される．標準メッセージは，注文書やインボイスなどのメッセージが含むデータエレメントの集合を定義する．データエレメントディクショナリは，全ての標準メッセージで使用されるデータエレメントの集合である．現在，国際標準としてISOのEDIFACT[3]，北米の標準としてANSIのX.12[4]，また国内の標準としてCII[5]，さらには銀行やチェーンストアなどの業界標準など，種々のEDI標準が存在している．

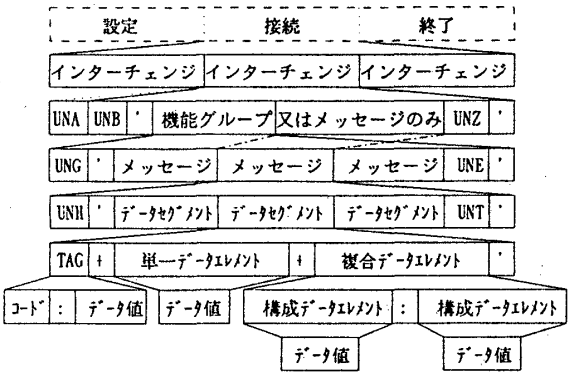

図1　EDIFACTシンタックスルール

Design of Multiple Data Format Translator for Electronic Data Interchange
Keizo SUGIYAMA, Sadao OBANA and Kenji SUZUKI
KDD R&D Laboratories

このような標準のデータフォーマットと各企業のローカルなデータフォーマットとの変換を行うソフトウェアをトランスレータと呼ぶ(図2)．通常トランスレータでは，整数や文字列といった型や長さ等の属性をデータエレメント毎に対応させて変換する．

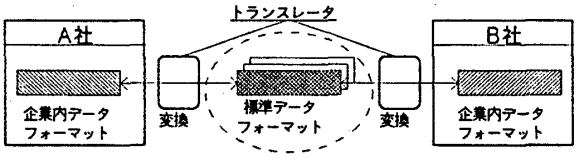

図2　EDIにおけるトランスレータの役割

3. 汎用データフォーマット変換方式の提案

これまで企業内フォーマットと特定標準フォーマットに対応したトランスレータは存在するが，複数のデータフォーマットや異なる標準間のトランスレータの実現は困難となっている．これは，トランスレータでは転送される情報の意味内容(セマンティクス)と表現形式(シンタックス)の両方の変換が必要であるが，これらを明確に分離せずにデータフォーマットを規定しているためであると考えられる．そこで，図3に示すように，データフォーマットのシンタックスとセマンティクスを分離し，セマンティクスに対する共通な表現としてトランスレータのプログラム内部にデータ構造を定義して，各データフォーマットのシンタックスルールはその共通表現に対する符号化規則として捉えることとする．

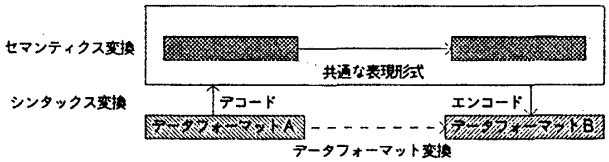

図3　EDI汎用フォーマット変換方式

シンタックス変換はEDIのデータフォーマットとプログラム内部のデータ構造間の変換であり，セマンティクス変換はデータエレメントの対応や属性の変換，例えば一方では発注日という項目が他方では日付と発注という2項目に別れているような場合や，またシンタックスルールの階層が異なる場合の変換に相当する．したがって，データフォーマット変換は，一旦データフォーマットをデコードした結果をセマンティクスに対応するデータ構造に設定し，セマンティクスレベルでの変換後に異なるデータフォーマットへエンコードすることで実現できる．この方式により，X.12とEDIFACT間といった種々のEDI標準間のトランスレータも容易に実現できる．

4. EDI用汎用トランスレータの設計

ここでは，３章の変換方式に従ったEDIトランスレータのソフトウェア構成，データ構造並びに変換に必要なファイルについて述べる．

4.1 ソフトウェア構成

種々の標準に対応するには，トランスレータのプログラム内で特定の標準に依存する部分をモジュール化し，モジュールを容易に交換できる必要がある．３章で述べた変換方式の場合，シンタックス変換のためのエンコーダ／デコーダ及びセマンティクス変換を行うゲートウェイの機能モジュールの単位は，モジュール間で授受するデータ構造の単位，すなわち図１のようなシンタックスルールのどの階層に対応するデータ構造を授受するかに依存する．データ構造の単位が小さいほど変換に必要なメモリ容量は少ないが，特定の標準に依存した部分を抽出するのが困難となる．そこで，本トランスレータでは，最上位の階層であるインターチェンジ単位で各モジュール間のデータ授受を行うこととする．図４に，ソフトウェア構成を示す．本トランスレータは，プログラムを変更せずに特定の標準の帳票やデータ項目の追加に柔軟に対応するため，実行時に帳票に関する情報をファイルから取り込んで動作させる．

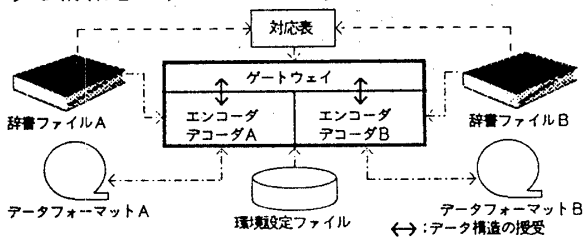

図４　トランスレータのソフトウェア構成

4.2 データ構造

複数の標準に対応するには，異なるデータフォーマットにも同一のデータ構造を使用できる必要がある．そこで，インターチェンジのデータ構造は，種々のシンタックスルールのスーパーセットであると考えられるEDIFACTに従って定義する．また，特定の帳票に依存しないデータ構造にするため，データエレメントに対応するデータ構造はデータエレメントの識別子，型を示す識別子並びにその値から構成し，それをリストで連結することでメッセージを表わす．機能グループなど他の階層も同様の構造とし，それらを上位の階層から順に連結することでインターチェンジのデータ構造とする．図５に，EDIFACTに対応したＣ言語による構造体の定義を示す．CII標準にこの構造体を適用した場合，CIIのファイルとメッセージグループは，各々インターチェンジと機能グループに対応付けられる．また，CIIにはデータセグメントの概念はないため，CIIのTFD(Transfer Form Data)のデータタグはセグメントタグに，データ値は単一データエレメントに対応付けられる．

4.3 ファイルの概要

(1)環境設定ファイル

トランスレータの各種オプション，例えばCIIの場合可変長レコードが扱えない通信システムに適合するための分割モードやEDIFACTと並行使用するためのTYPE-Eモード，を指定したり，発信者コードなどヘッダ等に固定的に設定する情報を格納する．

```
typedef struct { ATRB_FLAG        AtrbFlag;
                 union    {EDI_STRING      Str;     /*  文字列  */
                           EDI_INT         IntVal;  /*  整数  */
                           EDI_REAL        RealVal; /*  実数  */
                 } AtrbVal;
} SINGLE_DATA_ELEM_VAL;   /*  単一データエレメント  */
typedef struct ComponentDataElem { struct ComponentDataElem  *Next;
                                   SINGLE_DATA_ELEM_VAL ComponentVal;
} COMPONENT_DATA_ELEM;    /*  構成データエレメント  */
typedef struct CompositDataElem {  USHORT ElemNo;
                           COMPONENT_DATA_ELEM ComponentDataElemList;
} COMPOSIT_DATA_ELEM;     /*  複合データエレメント  */
typedef struct DataElem { struct DataElem  *Next;
                          BOOL             Simple;
                          union {  SINGLE_DATA_ELEM_VAL SingleVal;
                                   COMPOSIT_DATA_ELEM CompositVal;
                          } ElemVal;
} DATA_ELEM;      /*  データエレメントの選択(単純/複合)  */
typedef struct DataSeg {  struct DataSeg   *Next;
                          SEG_TAG          SegTag;
                          USHORT           ElemNo;
                          DATA_ELEM        *DataElemList;
} DATA_SEG;       /*  データセグメント  */
typedef struct Msg {      struct Msg       *Next;
                          USHORT           KindOfMsg;
                          USHORT           ElemNo;
                          DATA_SEG         *DataSegList;
} MSG;            /*  メッセージ  */
typedef struct FuncGroup { struct FuncGroup *Next;
                           USHORT   KindOfFuncGroup;
                           USHORT   ElemNo;
                           MSG      *MsgList;
} FUNC_GROUP;     /*  機能グループ  */
typedef struct Interchange { BOOL  Group;
                           USHORT   ElemNo;
                           union {  FUNC_GROUP        *FuncGroupList;
                                    MSG               *MsgList;
                           } ElemVal;
} INTERCHANGE;    /*  インターチェンジ  */
```

図５　モジュール間で授受するデータ構造

(2)辞書ファイル

辞書ファイルとして，個々のデータエレメントの属性を格納するファイル，データエレメントの値がコード化されている場合のコードの一覧表並びに帳票に含まれるデータエレメントの集合を定義したファイルをデータフォーマット毎に設ける．

(3)データエレメント対応表

２つのデータフォーマットのデータエレメント間の対応表をメッセージ毎に保持する．

5. おわりに

本稿では，トランスレータの機能をセマンティクス変換とシンタックス変換に分離し，シンタックス変換の機能モジュールを交換することで種々の標準への対応が容易なEDI用汎用トランスレータの設計概要について報告した．また，データ構造を汎用的に定義し，実行時に帳票に関する情報を取り込むことで，帳票等の追加にもプログラムを変更せずに対応可能とした．現在，本設計に従って，ローカルフォーマットとCII標準間のトランスレータの開発を行っている．最後に，日頃ご指導頂く小野所長，浦野次長及びご討論頂いた浅見通信網支援ソフトウェアグループリーダに感謝します．

参考文献

[1]:Keizo SUGIYAMA et al."New System Architecture for PC based Electronic Data Interchange(EDI)", ICCC'92
[2]:「最新EDI事情」通商産業省編，工業調査会
[3]:ISO9735,"Electronic Data Interchange for administration, commerce and transport(EDIFACT)- Application level syntax rules"
[4]:ANSI X.12,"American National Standard for electronic business data interchange"
[5]:「CIIシンタックスルール1.10」産業情報化推進センター