

EDI トランスレータの実装方式の提案

杉山 敬三[†] 小花 貞夫[†]
鈴木 健二^{††} 浦野 義頼^{†††}

利用形態やデータフォーマットが異なる EDI (電子データ交換) トランスレータの体系的な実装方式を提案する。この方式では、データフォーマット変換においてシンタックスとセマンティクスを分離して扱う。そのために、セマンティクスを表すメッセージのプログラム内部表現として、EDI の各種標準データフォーマットに共通に適用可能である汎用的なデータ構造を定義し、そのデータ構造と EDI の特定のデータフォーマット間で対応するシンタックスルールに基づき符号化・復号を行う。これにより、符号化・復号を行うモジュールを交換することで、標準データフォーマット間の変換や、端末でのプログラム内部のデータ構造と標準データフォーマット間の変換といった利用形態への対応を可能とする。また、異種データフォーマット間でのセマンティクスの対応付け規則を示す。最後に、本方式を適用した CII (産業情報化推進センター) 標準対応のトランスレータの実装を通じて、提案方式の有効性を示す。

A Proposal on Implementation Method of Electronic Data Interchange (EDI) Translator

KEIZO SUGIYAMA,[†] SADA OOBANA,[†] KENJI SUZUKI^{††}
and YOSHIYORI URANO^{†††}

This paper proposes a systematic method for implementing Electronic Data Interchange (EDI) translators. It allows EDI translators to support different usage and data formats flexibly. The key of this method is to separate syntax from semantics of EDI data formats. To realize it, a data structure in translator programs is defined as a common internal representation of EDI message, which corresponds to the semantics of a data format. Based on each syntax rule, encoding and decoding functions are conducted between the data structure and the EDI data format. The rules for mapping between different syntaxes are also defined. Finally, we show the effectiveness of the proposed method through the implementation of a translator for CII (Center for the Informatization of Industry) standard.

1. はじめに

近年、設計・製造や調達といったビジネスの様々な局面において情報を電子化し、ビジネスシステム全体の効率化を図る CALS (Commerce At Light Speed) が注目されている。CALS においては、異企業間の取引活動に関する情報を、標準的な規約を用いて電子化し、ネットワークを介して情報通信システム間で直接交換する EDI (電子データ交換)¹⁾が重要な要素となる。

EDI のデータフォーマットの標準には、国際標準に限らず、国内標準や業界標準など現状では様々なものが存在する。また、企業内のローカルなデータフォーマットと企業間で扱う標準データフォーマットも異なる。このため、EDI ではデータフォーマット変換を行うトランスレータが必要となる。しかもトランスレータは、企業内と企業間のデータフォーマット間の変換、標準データフォーマット間の変換、ならびに端末におけるプログラム内部のデータ構造と標準データフォーマットとの変換という 3 つの利用形態で必要となる。

従来のトランスレータは、上記ローカルデータフォーマットと標準データフォーマット間の変換の利用形態のみを対象としているうえに、扱うデータフォーマットごとに異なった実装を行っていた^{2),3)}。したがって、利用形態やデータフォーマットが異なるトランスレータのプログラムを効率的に開発する体系的な実装方式

[†] 国際電信電話株式会社研究所ネットワーク管理グループ
KDD R&D Laboratories

^{††} 国際電信電話株式会社研究所研究企画グループ
KDD R&D Laboratories

^{†††} 早稲田大学理工学総合研究センター
Advanced Research Institute for Science and Engineering, Waseda University

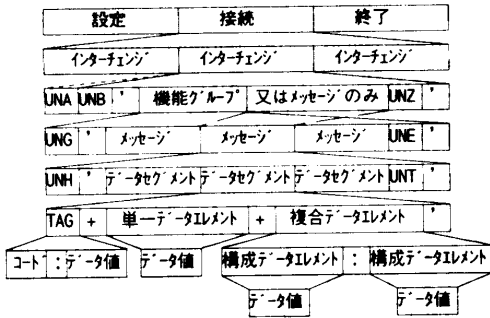


図1 EDIFACT シンタックスルール
Fig. 1 EDIFACT syntax rule.

が望まれている。

本論文では、種々の利用形態やデータフォーマットに柔軟に対応可能な EDI トランスレータの実装方式を提案する。まず 2 章で EDI におけるデータフォーマットを概説し、3 章でトランスレータの 3 つの利用形態と従来のトランスレータの問題点について述べる。この問題点を解決するために、4 章で汎用データフォーマット変換方式を提案する。5 章では提案方式に基づく CII (産業情報化推進センター) の標準データフォーマットに対応するトランスレータの実装概要を述べ、この実装を通じた提案方式の評価と考察を 6 章で行う。

2. EDI におけるデータフォーマットの概要

EDI の標準データフォーマットは、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レクトリの 3 つの要素で構成される¹⁾。シンタックスルールは、日付や注文番号などの帳票のデータ項目に対応するデータエレメントからシステム間で交換するデータを組み立てるための文法規則である。これは、通常図 1 のように階層的に規定され、階層ごとにヘッダ (UNA, UNG, etc.) とトレーラ (UNZ, UNE, etc.) が付く場合が多い。図 1 で、インターチェンジはシステム間で交換される単位であり、機能グループは同種の帳票の集合に、メッセージは帳票に、またデータセグメントは帳票に含まれる同種のデータ項目の集合に各々相当する。標準メッセージは、注文書や送り状等の各種帳票に含まれる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レクトリは、すべての標準メッセージで使用されるデータエレメントを定義した辞書である。

現在、データフォーマットの国際標準として ISO の EDIFACT⁴⁾、米国標準として ANSI の X.12⁵⁾、また日本国内の標準として、業界標準である EIAJ (日

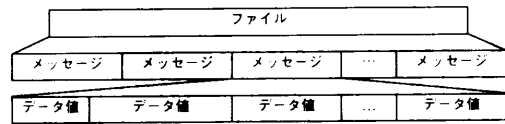


図 2 固定長データフォーマットのシンタックスルール
Fig. 2 Syntax rule for flat file data format with fixed record length.

本電子機械工業会) 標準⁶⁾やそれをベースにした CII 標準⁷⁾、さらには銀行やチェーンストアの業界標準など、各種の標準データフォーマットが存在している。このうち CII はシンタックスルールのみを規定しており、標準メッセージとデータエレメントディレクトリは EIAJ 等の各業界で規定したものを使用する。

一方、企業内では、固定長データフォーマットがローカルデータフォーマットとして一般に使用される。固定長データフォーマットでは、システム間で交換される単位のファイルにメッセージを含み、メッセージ内の各データ項目が出現する順番やその長さが固定される。データ項目の値が規定された長さに満たない場合、スペースやゼロをパディングする (図 2)。

3. トランスレータの 3 つの利用形態と従来のトランスレータの問題点

3.1 トランスレータの 3 つの利用形態

EDI トランスレータの利用形態には、以下の 3 種類が考えられる。

形態 1: ローカルデータフォーマットと標準データフォーマットの変換

固定長データフォーマットと、X.12 や CII 等の標準データフォーマットとの変換を行う。従来のトランスレータは、この形態に属する。

形態 2: 標準データフォーマットどうしの変換

X.12 と EDIFACT といったように、異なる標準データフォーマット間の変換を行う。現に EIAJ では、外国系企業との取引のために EIAJ の既存の伝票に対応する EDIFACT の標準メッセージを定義しており、その場合は EDIFACT と EIAJ 間での標準データフォーマット間の変換が必要になる。

形態 3: 端末でのプログラム内部のデータ構造と標準データフォーマット間の変換

上記 2 つの利用形態は異なるデータフォーマット間の変換であるが、これは端末での入力値をプログラム内部のデータ構造に保持して標準データフォーマットを作成したり、受信した標準データフォーマットの解析後プログラム内部のデータ構造に格納して、データを加工したり、ディスプレイ上に表示する形態で

ある。

3.2 従来のトランスレータの問題点

従来のトランスレータは、利用形態として上記の形態1であるローカルデータフォーマットと標準データフォーマット間の変換しか対象としていない¹⁾。さらに、EIAJ標準用トランスレータ³⁾のように特定の標準データフォーマットに固有のプログラムとなっており、異なる標準間などの種々のデータフォーマット間の変換に対応できる体系的な実装方式を用いたものではなかった。これは、以下の理由による。

従来のトランスレータが特定のEDI標準に固有の実装となっていた理由:EDIのデータフォーマットは、以下のように定義する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2つの側面を持つ。

(1) セマンティクスの定義

データフォーマットに含まれるメッセージ(帳票)の持つ情報の内容。たとえば、メッセージの種類は発注伝票であり、企業Aから企業Bに対し商品Cを96年7月7日までに10個納入するといった情報の意味に相当する。

(2) シンタックスの定義

上記のメッセージを特定のシンタックスルールに従って符号化したビット列のデータ。たとえば、(1)の納入日のデータエレメントをCII標準のシンタックスルールで符号化すると、値はYYMMDD形式(960707)で表し、納入日を表すタグ(40)と値の長さ(6)をその前に付与するため、16進数で“2806393630373037”となる。

情報のセマンティクスとシンタックスを明確に区別して扱う例としては、OSI(開放型システム間相互接続)参照モデル⁸⁾が存在する。OSI参照モデルでは、多様化する応用業務に対応するため、各応用業務で用いる情報のセマンティクスを扱う機能と、情報のビット列による表現、すなわちシンタックスを扱う機能を、各々応用層(第7層)とプレゼンテーション層(第6層)に割り当てている。

そのために、応用層で扱う情報がどのような要素から構成されるかを記述するASN.1⁹⁾(抽象構文記法1)と、ASN.1に対する符号化規則^{10),11)}を独立に規定している。これにより、応用層の情報に対し、異なる符号化規則が容易に適用できる。

一方、EDIでは、OSI参照モデルにおけるASN.1とその符号化規則の場合と異なり、標準メッセージの定義がシンタックスルールと独立していない。そのため、標準メッセージを定義する際には、表1に示す

表1 シンタックスルールにおける階層の対応

Table 1 Hierarchical correspondence among syntax rules.

EDIFACT	X.12	CII
インターチェンジ	インターチェンジ	ファイル
機能グループ	機能グループ	メッセージグループ
メッセージ	トランザクションセット	メッセージ
データセグメント	データセグメント	-
複合データエレメント	-	-
単一データエレメント/ 構成データエレメント	データエレメント	TFD (Transfer Form Data)

注: -は対応する階層がないことを示す。

ような各種データフォーマットにおけるシンタックスルールの差異を認識する必要がある。たとえば、CIIの標準メッセージは住所や宛名等のデータエレメントの集合として定義されるが、EDIFACTの標準メッセージはNAD(Name and Address)等のデータセグメントの集合として定義され、NADの中に名前や住所を表すデータエレメントが含まれる。

このような理由から、従来のトランスレータは、特定の利用形態やデータフォーマットに固有の実装となっている。このため、異なる利用形態やデータフォーマットに柔軟に対応するトランスレータを効率的に開発する体系的な実装方式が必要となる。

4. 汎用データフォーマット変換方式の提案

ここでは、3.2節で述べた従来のトランスレータの問題点を解決するためのEDIトランスレータの構成法およびデータ構造と、異種データフォーマット間でのセマンティクスの対応付け規則について提案する。

4.1 EDIトランスレータの構成法

EDIのデータフォーマット変換において、セマンティクスを表すメッセージのプログラム内部表現として、各種のデータフォーマットに共通に適用可能な汎用的なデータ構造を定義し、そのデータ構造とEDIの特定のデータフォーマット間で対応するシンタックスルールに基づき符号化・復号を行う方式を提案する(図3)¹²⁾。これは、OSI参照モデルの応用層プロトコルプログラム実装¹³⁾の際に、一般にASN.1に対応してデータ構造を規定し、そのデータ構造に含まれる応用層の情報をASN.1符号化規則に基づき符号化・復号するのと同様な方式をEDIに適用したものである。しかしながら、EDIでは各シンタックスルールに依存した標準メッセージが既に定義されているため、ASN.1のようなセマンティクスのための記法を新たに導入するのではなく、ここではトランスレータのプ

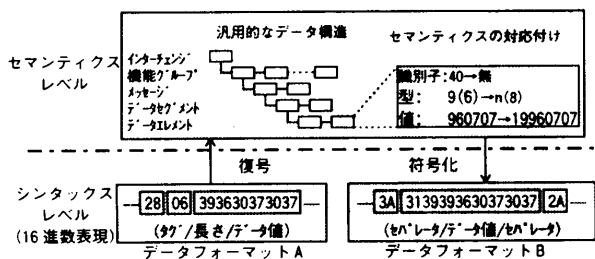


図3 汎用データフォーマット変換方式

Fig. 3 Universal data format conversion method.

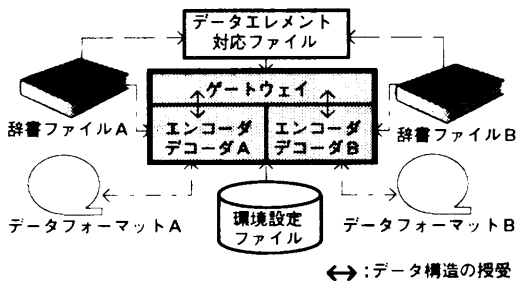


図4 EDIトランスレータのソフトウェア構成

Fig. 4 Software configuration of EDI translator.

プログラム内部におけるメッセージの表現方法の共通化を図ることで同じ効果を得る。具体的には、C言語の構造体をメッセージ内部表現のためのデータ構造として用いる。

本方式におけるデータフォーマット変換は、まずデータフォーマットAから、Aのシンタックスルールに従って、上記の汎用的なデータ構造上にメッセージを抽出する(復号処理)。次に、抽出されたメッセージに含まれる情報に対し、そのデータ構造上でセマンティクスを保持しながら、データフォーマットAからデータフォーマットBへの型(整数、文字列等)および長さの変換や、Aのメッセージには存在しないがBのメッセージに必須なデータエレメントの補充等を行う(セマンティクスの対応付けと呼ぶ)。最後に、セマンティクスの対応付けを行ったデータ構造から、Bのシンタックスルールに従ってデータフォーマットへエンコードする(符号化処理)。これに基づき、EDIトランスレータは、図4のように、符号化・復号処理を行うエンコーダ/デコーダ、セマンティクスの対応付けを行うゲートウェイの各モジュールおよび変換に必要なファイル群から構成する。

これにより、エンコーダ/デコーダやゲートウェイの組合せを変えることで、利用形態やデータフォーマットが異なるトランスレータを効率的に実現できる(図5)。つまり、利用形態2は、単に扱うデータフォー

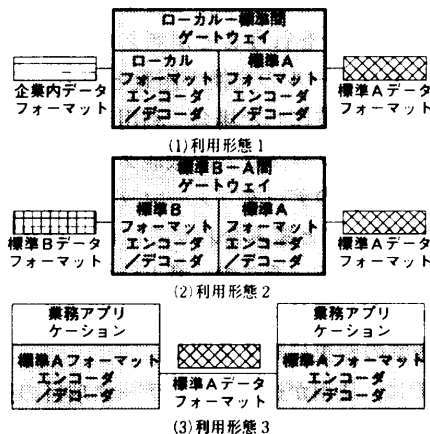


図5 利用形態やデータフォーマットが異なるトランスレータの効率的な実現

Fig. 5 Efficient realization of different usage and data formats for EDI translators.

マットが利用形態1と異なるだけと見なすことができ、ローカルデータフォーマットのエンコーダ/デコーダを標準データフォーマットのエンコーダ/デコーダに交換し、ゲートウェイにおいて標準データフォーマット間でのセマンティクスの対応付けを行うことで実現できる。また、利用形態3は、業務アプリケーションから直接エンコーダ/デコーダのモジュールを呼び出すことで実現できる。この場合、エンコーダ/デコーダのモジュールは、利用形態1と2と同じモジュールを流用できる。

4.2 セマンティクスの内部表現のための汎用的なデータ構造

セマンティクスの内部表現のための汎用的なデータ構造は、以下の3条件を満たす必要がある。

(1) 種々のデータフォーマットへの対応

4.1節で述べたように、このデータ構造は、各種のデータフォーマットに共通に適用可能でなければならない。現在複数のデータフォーマットが存在しているが、そのシンタックスルールは類似しており、基本的には表1に示したような階層構造をとる。これらデータフォーマットのうちEDIFACTは他のデータフォーマットの階層構造を包含しているため、このデータ構造はEDIFACTの階層構造に基づくものとする。

(2) 3つの利用形態への対応

3.1節の3つの利用形態に対応するには、エンコーダ/デコーダモジュールの交換が容易でなければならない。交換の容易性は、上記(1)で示した各種データフォーマットに対するデータ構造の共通化に加え、エンコーダ/デコーダの扱うデータフォーマットの階層に依存する。具体的には、帳票に相当するメッセージ

の単位や、ファイルに相当するインターチェンジの単位で符号化/復号を行うことが考えられるが、階層が低いほどモジュールの交換が困難となる。たとえば、メッセージの単位にした場合、利用形態3では業務アプリケーションプログラムが機能グループを作成することになり、他のデータフォーマットに対応するエンコーダ/デコーダと組み合わせるにはプログラムを変更する必要がある。そこで、最上位の階層であるインターチェンジの単位でデータ構造を規定する。

(3) 帳票の変更に対する対応

EDIでは、運用が進むにつれデータ項目の追加等の既存の標準メッセージへの変更や新たな標準メッセージの定義が行われることがあり、その場合でもデータ構造の変更を不要としなければならない。

そこで、特定の標準メッセージに固有の変数名等をデータ構造に定義せず、シンタックスルールの各階層の要素に対応するデータ構造はその要素の識別子・型・値から構成し、そのデータ構造をリストで連結してその階層を構成する。たとえば、データエレメントを表すデータ構造には、CII標準におけるデータタグ等のデータエレメントの識別子、9属性(数字)等の型並びにデータ値を含むフィールドを持たせる。

図6に、上記の3条件を満足するデータ構造としてC言語の構造体を定義した例を示す。ここでは、図1に示したEDIFACTのシンタックスルールのインターチェンジ以下の各階層の要素ごとに構造体を定義しており、下位の要素をポインタで連結している。また、データエレメントの識別子はElemTagに、データエレメントの型はAttrbFlagに、データエレメントの値は型に応じてAttrbValの共用体のメンバに格納される。

4.3 異種データフォーマット間でのセマンティクスの対応付け規則

ゲートウェイモジュールにおける異種データフォーマット間でのセマンティクスの対応付けでは、機能グループやメッセージ等のシンタックスルールにおける要素間での対応付けと、型や桁数等のデータエレメントにおける属性の変換が重要となる。ここではデータフォーマットとして代表的なEDIFACT, X.12およびCIIを対象とした対応付け規則を以下に示す(表2)¹⁴⁾。また、他のデータフォーマットも基本的には同様の階層構造をとっており、これらの規則が適用可能である。

4.3.1 シンタックスルールにおける要素の対応付け

EDIFACT, X.12, CIIデータフォーマット間のシンタックスルールの要素間の対応関係は表1に示さ

```
typedef struct {
    ELEM_TAG      ElemTag;
    ATTRB_FLAG   AttrbFlag;
    union { EDI_STRING  Str;          /* 文字列 */
            EDI_INT     IntVal;       /* 整数 */
            EDI_REAL    RealVal;     /* 実数 */
            ! AttrbVal;
    } 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;        /* セグメントタグ */
        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6 セマンティクスの内部表現のための汎用的なデータ構造
Fig. 6 Data structure of common internal representation for semantics.

れる。

● インターチェンジ

ファイル転送や電子メールにおけるファイルやボディに相当するため、ファイルやボディを対応付ける。

● 機能グループ

EDIFACTおよびCIIでは同一のメッセージの集合であるが、X.12では“purchase order”と“planning schedule”といった関連するトランザクションセットの集合として定義される。このため、X.12の1つの機能グループに1種類のトランザクションセットしか含まれていない場合には、EDIFACTやCIIの機能グループに1対1に対応付ける。また、異なる種類のトランザクションセットが含まれる場合には、CIIにおける複数のメッセージグループやEDIFACTにおける複数の機能グループに対応付ける。EDIFACTでは機能グループを構成しないことも許しているため、その場合は他のデータフォーマットの機能グループに含

表2 異種データフォーマット間でのセマンティクスの対応付け規則
Table 2 Semantics mapping rules among different data formats.

要素の 対応付け	インターチェンジ	ファイルやボディを対応付ける。
	機能グループ	1機能グループに同種のメッセージが含まれる場合、1対1に対応付ける。異種のメッセージが含まれる場合、複数の機能グループに対応付ける。機能グループが存在しない場合、メッセージに展開する。
	メッセージ	同一の意味を持つメッセージの場合、メッセージどうしを対応付ける。同一の意味を持つメッセージが存在しない場合には、新たにメッセージを定義する。
	データセグメント/ 複合データエレメント	データエレメントに展開してから対応付けを行う。
属性の 対応付け	データエレメント	対応する要素が存在する場合属性の対応付けを行うが、1対多の対応や動的な対応を処理する場合がある。 対応する要素が存在しない場合には、デフォルト値や導出値を割り当てる。
	型・桁数	プログラミング言語の型変換の機能で実現、有効桁数の異なる場合、削除やパディングを行う。
	コード値	同じ意味を持つコード値が存在する場合にはそれとの対応付けを行い、存在しない場合には適切なコード値を割り当てる。
	繰返し数 キャラクタセット	繰返し数 標準で規定される回数を越えた場合には、複数のメッセージに分割する。 キャラクタセット セパレータが含まれる場合には、他の文字に置き換える。

まれるメッセージと EDIFACT のメッセージを対応付ける。

- メッセージ

同じ意味を持つメッセージが存在する場合には、メッセージどうしを対応付ける。しかしながら、EDIFACT における “MEDREQ (Medical Service Request)” のように特定の EDI 標準にしか存在しない場合、どちらかの商慣行に合わせ、新たにメッセージを定義する。

- データセグメント/複合データエレメント

データセグメントは CII には存在せず、EDIFACT と X.12 でもデータセグメントの意味や構成するデータエレメントが異なる。また、複合データエレメントは EDIFACT にしか存在しない。このため、データセグメントや複合データエレメントの対応付けを行うには、これらに含まれるデータエレメントを対応付ける。

- データエレメント

(1) 対応する要素が存在する場合

基本的には 4.3.2 項のデータエレメントにおける属性の変換を行う。ただし、以下の処理が必要な場合がある。

① 1対多の対応

たとえば、EIAJ の “注文年月日” というデータエレメントは、EDIFACT では “date qualifier”+“date”+“date format qualifier” から構成される複合データエレメントに対応する。このような場合、データエレメントを 1対多に対応付ける。

② 動的な対応関係の決定

事前にデータエレメント識別子間の対応付けを登録することで静的に決まる場合以外に、動的に対応関係が決定される場合がある。たとえば、上述した

EDIFACT の複合データエレメントは、“date qualifier” の値が “purchase order” である場合に初めて EIAJ の “注文年月日” に対応付けられるため、データエレメントの値を精査して対応関係を決定する。

(2) 対応する要素が存在しない場合

変換元に存在するが変換先に存在しない場合には、省略する。変換元に存在せず変換先で必須項目の場合、基本的には事前に登録したデフォルト値を設定し、“注文金額” のように他の要素から計算等により導出できる場合はその値を設定する。

4.3.2 データエレメントにおける属性の変換

対応するデータエレメント間では、以下に示す属性の変換を行う。

- 型・桁数

基本的に各標準とも文字列あるいは数字列で表現するため、プログラミング言語が提供する型変換の機能で変換可能である。ただし、有効桁数が異なる場合、文字や数字の一部削除やパディングを行う。

- コード値

データエレメントの値が標準によってコード化されている場合、同じ意味を持つコード値との対応付けを行い、それが存在しない場合には事前に登録したコード値を割り当てる。たとえば単位コードに関して X.12 から EIAJ に変換する場合、“PC” (Piece) は両方とも存在するが、“CU” (Cup) は EIAJ に存在しないため、X.12 の “CU” に対しては EIAJ の “CN” (Can) 等を割り当てる。

- 繰返し数

データエレメントの出現回数が規定回数を越えた場合、複数のメッセージに分割する。たとえば、EIAJ

表 3 実装の範囲
Table 3 Supported range in implementation.

利用形態	・標準-ローカルフォーマット間変換 ・端末での標準フォーマットの直接変換
データフォーマット	標準側：CII 標準 (Ver1.11, 全オプションサポート) ローカル側：固定長フォーマット
標準メッセージ	EIAJ の「注文情報」と「注文請け情報」
データエレメント ディレクトリ	EIAJ の項目 No.1 (データ処理 No.) から No.158 (指定メーカー名) まで (他の情報も登録可能)
CII 標準のデータ型	X : 8 bit 文字 K : 16 bit 文字 (JIS コード, ESC 文字無) 9 : 固定小数点正数 N : 浮動小数点数 Y : 日付
固定長フォーマットの データ型	X : 8 bit 文字 K : 16 bit 文字 (JIS コード, ESC 文字有) KJ : 8/16 bit 文字混合 (JIS コード, ESC 文字有) KE : 8/16 bit 文字混合 (EUC コード) KJ : 8/16 bit 文字混合 (シフト JIS コード) 9 : 固定小数点正数 N : 浮動小数点数 Y : 日付
通信手順	F 手順

の注文情報のメッセージには 1 種類の品名しか含まれないが、X.12 では複数種類可能であるため、複数の品名が含まれる場合にはメッセージを分割する。

- キャラクタセット

X.12 における “*” のように、データエレメントを区切るためのデータエレメントセパレータが CII のデータエレメントに含まれる場合、他の文字に置き換える。また、EDIFACT では、リリース文字 “?” を挿入する。

5. 汎用データフォーマット変換方式に基づく EDI トランスレータの実装

提案方式の有効性を実証するため、4 章の方式に基づく CII 標準対応の EDI トランスレータを Windows パソコン上に C 言語を用いて実装した¹⁵⁾。

5.1 実装範囲

表 3 に今回の実装範囲を示す。ここでは、3.1 節の利用形態 3 を実証するため、Windows の表計算ソフト (Excel Ver5.0) から直接エンコーダ/デコーダを呼び出すようにした。固定長側は CII 側と同じ意味のメッセージを定義し、データエレメントが一方にしか存在しない場合や異なる型の指定を許すようにした。また、CII と同様にファイルに複数種類のメッセージを含めることを可能とした。端末間での通信手順として、EDI 用のファイル転送手順である F 手順¹⁶⁾を用いた。

5.2 変換に必要なファイル

変換の際には、エンコーダ/デコーダにおける属性変換のための情報やゲートウェイモジュールにおけるデータエレメントの対応関係を格納したファイルが必要となる。ここでは、Windows における初期化ファイルの形式である INI ファイル形式を採用した。

5.2.1 辞書ファイル

エンコーダ/デコーダで使用する辞書ファイルとして、以下に示す 3 種類のファイルを設けた。

- 標準項目辞書ファイル

データエレメントの値を精査するため、個々のデータエレメントの属性 (データタグ、項目名、型、最小・最大長、整数部・小数部桁数および値が登録・コード化されているかを示すフラグ) を定義する。

- ID 選択項目ファイル

データエレメントの値がコード化されている場合に、その値を精査するためのコード値の一覧表である。

- 帳票別項目一覧ファイル

メッセージ (帳票) を構成するデータエレメントが正しく含まれているかを確認するため、標準メッセージ中のデータエレメントの集合を定義する。

5.2.2 データエレメント対応ファイル

ゲートウェイモジュールでのセマンティクスの対応付けのために、2 つのデータフォーマットのデータエレメント間の変換テーブルを定義する。

本ファイルの記述例を図 7 に示す。セクション名は、標準の版が EIAJ011D で、メッセージの種類が 0502

```
[EIAJ011D_0502] ;セクション名
LINECNT=6
No.001=項目 No.=00001,必須=Y,属性(桁数)=9(5)
No.002=MDH=H,繰返数 1=031 ;マルチ明細ヘッダ
No.003=項目 No.=00031,必須=Y,属性(桁数)=9(6),値="123456"
No.004=ISID=I ;インターナルセグメント指示子
No.005=項目 No.=00032,属性(桁数)=9(9)V(3)
No.006=MDT=H ;マルチ明細トレーラ
```

図 7 データエレメント対応ファイルの記述例

Fig. 7 Description example for data element mapping file.

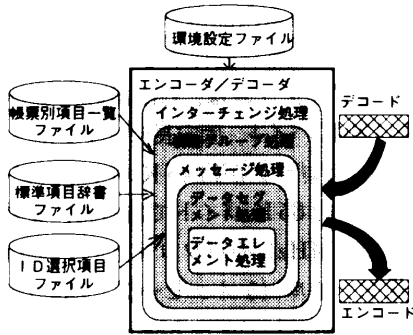


図 8 エンコード/アコード処理の概念

Fig. 8 Concept of encoding/decoding processing.

(注文情報)である変換テーブルを示している。図 7 では CII 側の項目に対応する固定長側の項目を示しており、たとえば No.003 では CII の項目 No.00031 に対応する固定長データフォーマットの属性が 9 (固定小数点正数) で桁数が 6 である。また、No.002, 004, 006 は各々 CII におけるマルチ明細ヘッダ、インターナルセグメント指示子、マルチ明細トレーラに対応しており、No.003 から No.005 が 31 回繰り返され、明細番号が H であることを示している。CII では必須項目(「必須=Y」と記述)であるが固定長側に対応する項目がない場合、No.003 の「値="123456"」のようにデフォルト値を設定する。

5.2.3 環境設定ファイル

トランスレータの動作を決定するためのファイルであり、各種オプション、たとえば CII の場合、可変長レコードが取り扱えない通信回線のための分割モード、通信制御文字と同一のオクテットが使用できない通信回線のため非透過モード、EDIFACT と並行使用するための TYPE-E モードを指定したり、発信者コードや文字コードなどメッセージグループヘッダに固定的に設定する情報を格納した。

5.3 エンコーダ/デコーダモジュールの処理

以下では、エンコーダ/デコーダモジュールにおける処理を述べる(図 8)。

```
USHORT encode_XXX( file, len, Inter, envinfo, err )
CHAR *file; /* エンコード結果のファイル名 */
ULONG *len; /* エンコード結果のファイル長 */
INTERCHANGE *Inter; /* インターチェンジ構造体 */
ENV_INFO *envinfo; /* オプション情報 */
USHORT *err; /* エラーコード */
```

(1)エンコーダ

```
INTERCHANGE *decode_XXX( file, envinfo, err )
CHAR *file; /* デコードするファイル名 */
ENV_INFO **envinfo; /* オプション情報 */
USHORT *err; /* エラーコード */
```

(2)デコーダ

図 9 エンコーダ/デコーダのインタフェース仕様

Fig. 9 Interface specification for encoder/decoder.

5.3.1 インタフェース仕様

図 9 にエンコーダ/デコーダの各モジュールのインタフェース仕様を示す。4.2 節で述べたセマンティクスのプログラム内部表現のための汎用データ構造 (INTERCHANGE 型) は、エンコーダでは関数の引数であり、デコーダでは戻り値としている。また、どちらの関数にもエラー情報を出力パラメタとした。関数名の中で XXX になっている部分には、データフォーマットを示す文字列 (CII/FIX) が入る。

5.3.2 エンコーダ

エンコーダでは、初期化処理として環境設定ファイルを読み込み分割モード等の動作モードを決定した後、シンタックスルールの階層に従って、以下のように上位から順に 1 つ下の階層の処理を呼び出す。

(1) インターチェンジ処理

エンコード結果を格納するファイルをオープンし、構造体が存在する数だけ機能グループ処理を呼び出す。

(2) 機能グループ処理

使用する帳票別項目一覧辞書や標準項目辞書を決定し、機能グループヘッダを書き出す。その後メッセージの構造体が存在する数だけメッセージ処理を呼び出し、最後に機能グループトレーラを書き出す。

(3) メッセージ処理

メッセージヘッダを書き出し、構造体の存在する数だけデータセグメント処理を呼び出しながら、エンコードされた結果をファイルに書き出し、最後にメッセージトレーラを書く。

(4) データセグメント/データエレメント処理

標準項目辞書ファイルや ID 選択項目定義ファイルを参照して、正しい値が構造体に設定されているかどうか確認し、値をエンコードする。

CII のメッセージにはヘッダにメッセージ長のフィールドが含まれており、メッセージを組み立てるまでは

メッセージ長が決定できないため、メモリ上でメッセージを組み立てた後でファイルに書き出すようにした。また、メッセージの組立て処理をどのモードでも共通に使えるようにするため、メッセージの組立て時でなくファイルへの書出し時に、分割モードや非透過モードの処理を行うようにした。

固定長ファイルでは、1ファイルに異なる種類のメッセージを含めることを可能にするために、各メッセージの直前にその種類を示す情報区分コードを含むヘッダフィールドをオプションとして設けた。

5.3.3 デコーダ

デコーダもエンコーダと同様、図8に示すようにシンタックスルールの階層に従って処理を行うが、下位の階層の処理に移る前にあらかじめ下位層の構造体の領域、たとえば機能グループ処理ならメッセージの構造体を割り当てる点で異なる。また、デコード時には、受信したインターチェンジがどのモードでエンコードされているか分からないため、メッセージグループヘッダの先頭を1バイト先読みし、この値によりTYPE-Eモードなどモードの処理を振り分けるようにした。

5.4 ゲートウェイモジュールの処理

基本的には4.3節の規則を適用する。

5.4.1 シンタックスルールにおける要素の対応付け

固定長データフォーマットには図2に示すように機能グループが存在しないため、CIIから固定長に変換する場合には、CIIのメッセージグループに含まれるメッセージを固定長のメッセージに対応付け、逆の場合には連続して格納された同種のメッセージ群をメッセージグループに組み立てるようにした。

5.4.2 データエレメントにおける属性の変換

データエレメント対応ファイルでは基本的にCIIと固定長で同一の型を設定するが、数字のみを含むX属性と9属性といったセマンティクスが保持できるものは異なる設定を可能とした。また、表3で示した漢字に関する型（固定長側KJ,KE,KS,K属性とCII側K属性）は任意の組合せを可能とし、そのためのコード変換を行う機能を実装した。ただし、8/16bit文字を混合したデータの送受はCII標準では規定されないため、EDIを行う両者間で合意した場合のみ、CIIのX属性を使用して行うこととした。

5.5 表計算ソフトとの連携

EDIのデータ入出力端末として利用可能とするため、ユーザがExcelで定義した帳票画面（ワークシート）上からVBA（Visual Basic for Applications）を用いてエンコーダ/デコーダを呼び出すようにした。ここでは、事前にユーザがワークシートのセルと項目No.

の対応を登録し、VBAがこの情報を参照してセルとの間でデータを授受することで、新たにプログラムを作成せずに任意のワークシートに対応できるようにした。

6. 評価と考察

6.1 EDIトランスレータの構成法について

5章で述べた実装を行った結果、ローカルデータフォーマットと標準データフォーマット間の変換および端末でのプログラム内部のデータ構造と標準フォーマット間の変換が正常に動作することを確認した。開発したソースプログラムは、CII用のエンコーダ/デコーダが約190Kバイト、固定長データフォーマット用のエンコーダ/デコーダが約45Kバイト、ゲートウェイモジュールが約65Kバイトであった。また、CPUが486DX2（66MHz）のパソコン上で、約5.6Kバイトの固定長ファイル（5メッセージ、250データエレメント）をデコードしてゲートウェイでセマンティクスの対応付けを行い、約3KバイトのCII標準ファイルへの変換時間およびその逆変換に要する時間は各々約2秒であった。この値は、単一の利用形態および特定のデータフォーマットにのみ対応する市販のトランスレータと同等であり、実用的な時間で変換できることが確認できた。今回は標準データフォーマット間の変換は実装しなかったが、X.12等の他の標準データフォーマットのエンコーダ/デコーダを4.2節のデータ構造に基づいて作成し、データエレメントの動的対応の決定のような標準データフォーマット間変換で生じる処理をゲートウェイモジュールに追加することで容易に実現できる。

本方式は、データフォーマットごとのエンコーダ/デコーダをライブラリとして用意し、ゲートウェイモジュールと組み合わせさえすれば、任意の利用形態やデータフォーマットに対応でき、EDIトランスレータを効率的に開発する有効な手法であるといえる。

6.2 セマンティクスの内部表現のための汎用的なデータ構造について

4.2節で述べたセマンティクスのプログラム内部表現のための汎用的なデータ構造をCIIと固定長データフォーマット両方に適用し、異なるデータフォーマットや利用形態に対応可能であるとともに、標準メッセージの変更にも対応可能であることが確認できた。このデータ構造はEDIFACTのシンタックスルールに基づいており、今回の実装のようにCIIや固定長データフォーマットを扱う場合には使用しないフィールドがいくつか存在するため、注意深くプログラミングする

必要がある。これは、プログラマに対し CII の要素だけを見せるマクロ等を提供することで容易に対処できる。

6.3 異種データフォーマット間でのセマンティクスの対応付け規則について

シンタックスルールにおける要素の対応付けと、データエレメントにおける属性の変換の2つの観点から分類することにより、異種データフォーマット間でのセマンティクスの体系的な対応付け規則を示すことができた。各データフォーマットにおいて同一種類の標準メッセージに含まれる情報に本質的な差異はなく、使用されるデータエレメントの型も文字列や数字列がほとんどであり、本規則が幅広く適用できると考えられる。

ただし、ゲートウェイモジュールのプログラムには、一度プログラムを作成すれば他のデータエレメントの変換にも適用できる部分と、特定のデータエレメントの変換にしか適用できない部分が存在する。たとえば、型変換のモジュールは汎用的に使用できるが、4.3節で述べたデータエレメントが1対多に対応する場合にはプログラム内にこのための変換ロジックを組み込む必要がある。また、有効桁数の差違による桁落ちを容認できない場合や、日本語と英語のように扱う言語が異なる場合など、EDIを行う両者の事前の合意により運用で避けるしかない場合も存在する。

しかしながら、このような例外処理はそれほど多くなく、4.3節で示した対応付け規則は各種のゲートウェイモジュールの開発に有効である。

7. おわりに

本論文では、利用形態やデータフォーマットが異なるEDI用トランスレータを体系的に実装するための汎用データフォーマット変換方式を提案した。

この方式では、EDIのデータフォーマット変換において、セマンティクスを表すメッセージのプログラム内部表現として、各種のデータフォーマットに共通に適用可能である汎用的なデータ構造を定義し、そのデータ構造と特定のデータフォーマット間に対応するシンタックスルールに基づき符号化・復号を行う。これにより、符号化・復号を行うモジュールを交換することで種々のデータフォーマット間の変換や、端末でのプログラム内部のデータ構造と標準フォーマット間の変換といった利用形態への対応を可能とした。また、異種データフォーマット間でのセマンティクスの対応付け規則を示した。さらに、提案方式に基づいたトランスレータの実装を通じて本方式の有効性を実証した。

今後とも様々なEDI標準が並用される状況においては、複数のデータフォーマットへの対応を容易とする本方式は、異なるデータフォーマットで構築されたEDIシステム間接続を実現する有効な手法になると考えられる。

謝辞 日ごろご指導いただくKDD研究所村上仁己所長に感謝します。

参考文献

- 1) 通商産業省(編):最新EDI事情,工業調査会(1990).
- 2) Sugiyama, K., et al.: New System Architecture for PC Based Electronic Data Interchange (EDI), ICC'92 (1992).
- 3) 商品化が進むEDIパソコンソフト,コミュニケーションテクノロジー,2月号(1991).
- 4) ISO9735: Electronic Data Interchange for Administration, Commerce and Transport (EDIFACT)-Application Level Syntax Rules (1988).
- 5) ANSI X.12: American National Standard for Electronic Business Data Interchange (1986).
- 6) EIAJ取引情報化対応標準1C,日本電子機械工業会(1991).
- 7) CIIシンタックスルール1.11,産業情報化推進センター(1993).
- 8) ITU-T Recommendation X.200: Reference Model of Open Systems Interconnection for CCITT Applications (1988).
- 9) ITU-T Recommendation X.680: Information Technology-Abstract Syntax Notation One(ASN.1): Specification of Basic Notation (1994).
- 10) ITU-T Recommendation X.690: Information Technology-ASN.1 Encoding Rules: Specification of Basic Encoding Rules(BER), Canonical Encoding Rules (CER) and Distinguished Encoding rules (DER) (1994).
- 11) ITU-T Recommendation X.691: Information Technology-ASN.1 Encoding Rules: Specification of Packed Encoding Rules (PER) (1995).
- 12) 杉山,小花,鈴木:EDI用汎用トランスレータの実装,マルチメディア通信と分散処理ワークショップ,情報処理学会,3月(1993).
- 13) Huitema, Christian and Ghislain: Measuring the Performance of an ASN.1 Compiler, Upper Layer Protocols, Architectures and Applications, IFIP (1992).
- 14) 杉山,小花,浅見,鈴木:EDI用汎用トランスレータにおける異種データフォーマット間の対応付けに関する一考察,第47回情報処理学会全国大会,7F-4 (1993).

- 15) 杉山, 小花, 鈴木: Windows 対応パソコン EDI
トランスレータの実装, 第 50 回情報処理学会全
国大会, 1T-1 (1995).
- 16) 杉山, 小花, 鈴木: パソコン EDI 端末における
F 手順の設計と評価, 電子情報通信学会情報ネッ
トワーク研究会, IN93-120 (1994).

(平成 7 年 9 月 29 日受付)

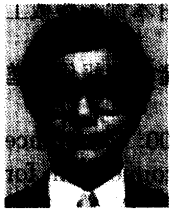
(平成 8 年 9 月 12 日採録)



杉山 敬三 (正会員)

昭和 37 年生。昭和 60 年京都大学
工学部情報工学科卒業。昭和 62 年
同大学院修士課程修了。同年国際電
信電話 (株) 入社。現在, 同社研究
所ネットワーク管理グループ主査。

この間, OSI プロトコル実装, ネットワークアーキテ
クチャ, 分散処理, ネットワーク管理, EDI の研究に
従事。平成 6 年度電子情報通信学会学術奨励賞受賞。
電子情報通信学会会員。



小花 貞夫 (正会員)

昭和 28 年生。昭和 51 年慶応義塾
大学工学部電気工学科卒業。昭和 53
年同大学院修士課程修了。同年国際
電信電話 (株) 入社。現在, 同社研
究所ネットワーク管理グループリー
ダ。工学博士。この間, パケット交換方式, ネットワ
ークアーキテクチャ, OSI プロトコル実装, データベー
ス, ビデオテックス, 分散処理, ネットワーク管理の
研究に従事。電子情報通信学会会員。

この間, パケット交換方式, ネットワ
ークアーキテクチャ, OSI プロトコル実装, データベー
ス, ビデオテックス, 分散処理, ネットワーク管理の
研究に従事。電子情報通信学会会員。



鈴木 健二 (正会員)

昭和 20 年生。昭和 44 年早稲田大
学理工学部電気通信科卒業。昭和 44
~45 年オランダのフィリップス国際
工科大学に招待留学。昭和 51 年早
稲田大学大学院博士課程修了。工学
博士。同年国際電信電話 (株) 入社。現在同社研究所
研究企画グループリーダー。この間, 磁気記録, パケッ
ト交換方式, ネットワークアーキテクチャ, 高速・分
散処理の研究に従事。平成 4 年度電子情報通信学会業
績賞, 平成 7 年度科学技術庁長官賞を各受賞。平成 5
年度より電気通信大学大学院情報システム学研究科客
員教授。電子情報通信学会, IEEE 各会員。



浦野 義頼 (正会員)

昭和 17 年生。昭和 40 年早稲田大
学理工学部電気通信科卒業。昭和 45
年同大学院博士課程修了。工学博士。
同年国際電信電話 (株) 入社。同社
研究所所長を経て, 平成 8 年早稲田
大学理工学総合研究センター教授, 現在に至る。この
間, フェイル・セーフ論理システム, パケット交換方
式, ネットワークアーキテクチャ, OSI, ビデオテッ
クス, ソフトウェア信頼性, AI などの研究・開発に
従事。昭和 63 年科学技術庁長官賞, 平成 4 年電子情
報通信学会業績賞各受賞。電子情報通信学会, IEEE
各会員。