デジタル・ドキュメント　6－4
（1997．3．28）

SGML 文書の内容検証方式の検討

今村 誠　　森口 修　　鈴木 克志
E-mail: {imamura, mog, suzuki}@isl.melco.co.jp

三菱電機（株）情報技術総合研究所 音声・言語インタフェース技術部

SGML 文書を入出力データとする応用システムでは、入出力データの正当性の保証やエラー処理のために、DTD による構文チェックだけでなく、アプリケーションに応じた意味的な文書内容の検証が必要になる。しかし、既存の SGML 文書変換ツールでは、 内容検証処理の記述で必要とされる「文書の部分構造間の比較」や「複数文書にまたがる文書内容の制約関係の判定」といったグローバルな文書構造操作の記述能力が弱いという問題があった。本稿では、SGML 文書の部分構造を取り出す基本演算式のネスト構造を許す問い合せ言語を用いた内容検証方式を提案する。本方式によれば、内容検証に必要な知識を検証対象文書と同じ SGML 形式で表現することにより、SGML 文書の内容検証処理を統一的かつ簡潔に記述できる。

A Study on a Method of Content test of SGML Documents

Makoto IMAMUMA　　Osamu MORIGUCHI　　Katsushi SUZUKI
E-mail: {imamura, mog, suzuki}@isl.melco.co.jp

Mitsubishi Electric Corporation
Information Techonology R & D Center
Human Media Technology Dept.

Not only consistency test with DTD but also application dependent content test is necessary for systems using SGML documents as input/output data to confirm validity of the input/output data or to handle errors. However, existing SGML conversion tools are not expressive enough to describe global operations on document structure for content test of SGML documents, such as comparison between parts of the document structure and to check satisfiability of semantic constraints among parts of documents. We present a method of content test using a query language which supports nesting of primitive expressions accessing parts of the document strcture. This method enables us to describe content test processing of SGML documents uniformly and concisely by expressing knowledge for content test by the same SGML as tested documents.

1．はじめに

CALS(Commerce At Light Speed)の進展に伴い、形式を標準化した文書を統合管理することにより、文書情報を既存の情報システムから利用しやすくするための技術が要求されている。この要求に応えるための文書形式が SGML[1](Standard Generalized Markup Language)である。SGML は、文書型定義(DTD:Document Type Definition)によって、業務に応じて、文書の論理構造を厳密に規定できるので、文書中から必要な情報を機械的に抽出しデータベースに自動登録したり、仕様書から必要な項目を抽出しEDI(Electronic Data Interchange)メッセージに変換するといった処理が容易になる[2]。

しかし、SGML 文書による機械処理を円滑に進めるためには、DTD による構文的な文書構造のチェックだけでなく、アプリケーションに応じた意味的な文書内容のチェック(文書の内容検証)が必要になる。例えば、SGML 文書のデータベース自動登録の際には、格納すべき情報の属性の名称、データ型、データ長、及びデータ単位等がデータ辞書の条件を満たしていることをチェックする必要がある。

また、SGML 文書を内容検証するプログラムでは、文書構造に応じて検証に必要とされる処理を記述しなければならない。SGML の文書構造に応じた変換処理を記述する従来ツールとして OmniMark[3]や AEsop[4]がある。これらは SGML 文書を TEX や HTML といった印刷・表示用の形式へと変換する処理を主な応用としている。そのため、文書の部分構造同士を比較したり、複数文書にまたがる文書内容の制約関係をチェックするといったグローバルな文書構造の操作を記述する能力が弱いという問題があった。

本稿では、SGML 文書の文書構造中の必要な部分構造を取り出すための問い合せ言語により、SGML 文書の内容検証処理を簡潔に記述する方式について述べる。本方式は、以下の特徴をもつ。

(1) SGML 文書中の部分構造の取り出し・比較等の基本演算式から構成される文書構造問い合せ言語を導入することにより、グローバルな文書構造の操作を簡潔に記述できる。

(2) 内容検証に必要とされる知識(内容検証用知識)を検証対象の文書形式と同じ SGML で記述することによって、内容検証用知識の保守性を高めることができる。

本稿の構成は以下のとおり。2.では、本稿を理解するのに必要な SGML に関する知識について簡単に説明する。3.では、内容検証の必要性と内容検証に要求される機能を例に基づいて説明する。4.では、SGML 文書構造の問い合わせ言語を用いた内容検証処理方式の実現方針を示し、5.では、その問い合せ言語の構文と意味を定義する。6.では，内容検証処理が5.の問い合せ言語により簡潔に記述できることを示す。7.では、本研究と既存の SGML 文書変換ツールと構造化文書データベースの問い合せ言語との差異について考察する。

2．SGML(準備)

SGML は、文書形式の ISO 規格(ISO8879)であり、文書の論理構造の記述を表現するものである。個々の SGML 文書は**文書インスタンス**と **DTD** からなる。

図 1は文書インスタンスの例であり、図中の<申請者>や<名称 分類="計算機">等のタグと呼ばれる印により論理構造を表現した文書である。

```
<購入仕様書>
<申請者><氏名>森口 太郎</氏名>
        <社員番号>31415</社員番号></申請者>
<購入品><名称 分類="計算機">○○パソコン</名称>
        <価格>150000</価格>
        <特記>タワータイプにしてください</特記>
</購入品>
<購入品><名称 分類="ソフトウェア">××ワープロ</名称>
        <価格>20000</価格>
        <仕様><項目><項目名>対象計算機</項目名>
                    <項目値>パソコン</項目値></項目>
              <項目><項目名>OS</項目名>
                    <項目値>Windows95</項目値></項目>
          </仕様>
</購入品>
. . . . . .
<価格合計> 325000</価格合計>
<検印>あり</検印>
</ 購入仕様書>
```

図 1　購入仕様書(SGML 文書インスタンス)

DTD は文書インスタンスが満たすべき文書の論理構造を規定する文法規則の集合である(図 2)。

```
1. <!DOCTYPE 購入仕様書 [
2. <!ELEMENT 購入仕様書 -o (申請者, 購入品*, 価格合計, 検印) >
3. <!ELEMENT 申請者 -o (氏名,社員番号)>
4. <!ELEMENT 氏名 -o (#PCDATA)>
5. <!ELEMENT 社員番号 -o (#PCDATA)>
6. <!ELEMENT 購入品 -o (名称,価格,仕様?,特記?)>
7. <!ELEMENT 名称 -o (#CDATA)>
8. <!ATTLIST 名称  分類 CDATA #REQUIRED>
9. <!ELEMENT 仕様 -o (項目+)>
10.<!ELEMENT 項目 -o (項目名,項目値)>
11.<!ELEMENT 項目名 -o (#PCDATA)>
12.<!ELEMENT 項目値 -o (#PCDATA)>
13.<!ELEMENT 価格合計 -o (#PCDATA)>
14.<!ELEMENT 検印 -o (#PCDATA)>                ]>
```

図 2 購入仕様書の DTD

DTD の主要部分は、**エレメント宣言**と**アトリビュート宣言**である。**エレメント**とは SGML 文書における論理的な構成成分である。具体的には、ある特定の開始タグと終了タグで挟まれた文書の部分を指す。例えば、図 1中のタグ<申請者>と</申請者>で挟まれた部分のテキストが "申請者"エレメントである(本稿では、エレメントの開始タグと終了タグを除いたテキスト部分を**エレメントの内容**と呼ぶ)。エレメント宣言では、エレメントの名称や下位エレメントの出現順序を規定する。例えば、図 2の 2 行目では、"購入仕様書"エレメントは、下位エレメントとして、"申請者" エレメント,"購入品"エレメント,"価格合計" エレメント,そして"検印"エレメントがこの順に出現し、"購入品"エレメントは 0 回以上繰り返して出現することを規定する。

アトリビュートとは、<名称 分類="計算機">中の分類のようにタグを修飾する印である。タグ中の"計算機"は**アトリビュートの値**と呼ばれる。アトリビュート宣言では、アトリビュートの名前と値に対する条件を規定する。例えば、図 2の 8 行目は、アトリビュート"分類"は、"名称"エレメントの開始タグ中に必ず出現し、値は文字列(CDATA)であることを規定する。

3. SGML 文書の内容検証とは

2. で説明したように、DTD で規定できるのは、エレメントの出現順序や出現回数などの SGML 文書インスタンスに対する構文的な制約だけである。しかし、SGML 文書を CALS などの応用システムで利用するためには、エレメント中の文書内容に関する制約やエレメント間の制約関係を規定する必要が生じる。これらの制約をチェックすることを、DTD による構文の検証と対比する意味で、**SGML 文書の内容検証**と呼ぶことにする。以下、SGML 文書の内容検証の必要性と要求機能を示し、そして内容検証機能を組み込んだ SGML 文書処理ツールについて述べる。

3.1 SGML 文書内容検証の必要性

(1) データ内容の標準化による情報共有の促進

CALS では、企業内の部門間や企業間で製品情報や受発注情報を交換する。これらの情報交換を円滑に行うためには、交換する文書の論理構造だけでなく、文書中で用いる用語や機械処理用に挿入する数値情報や識別記号の使用に関する標準が必要になる。例えば、電子商取り引きのための製品カタログを SGML 化する際には、製造メーカ間の比較検索を容易にするために、製品分類体系や製品の性能を特徴付ける属性名(パソコンならディスク容量や搭載 CPU など)の標準化が必要である。内容検証機能は、交換する文書情報がこれらの標準に従っているかをチェックする役割を担っており、標準が適切に運用されていることを保証するために必要な機能である。

(2) プログラムの入出力/中間データとしての文書利用の促進

SGML では、DTD によりアプリケーションに応じた文書構造を厳密に規定できるので、SGML 文書をプログラムの入出力あるいは中間データとして利用することができる。この種の応用としては、SGML 文書データベース、EDI におけるメッセージ記述言語、ハイパーテキストの記述言語、そして、SGML 文書情報のデータベースへの自動登録などがあげられる[2]。これらを利用したシステムの入出力の正当性を保証したり、エラー処理を記述するためには、DTD による構文的な文書構造のチェックだけでなく、アプリケーションに応じた文書の内容検証が必要になる。例えば、SGML 文書のデータベース自動登録の際には、格納すべき情報の属性の名称、データ型、データ長、

及びデータ単位等がデータ辞書の条件を満たしていることをチェックする必要がある。

3.2 SGML 文書の内容検証の要求機能

以下では、図 1の購入仕様書の例を基に、内容検証ではSGML 文書の部分構造の取り出し・比較等の操作を組み合わせた処理を実現する機能が必要であることを示す。

(1) 申請者と社員名簿との整合性チェック

購入仕様書(図 1)の"申請者"エレメントの内容が、社員名簿(図 3)の"社員"エレメントのいずれかと等しくなければならない。すなわち、"氏名"と"社員番号"の両エレメントとも一致しなければならない。

```
<社員名簿>
<社員><氏名>森口 太郎</氏名>
        <社員番号>31415</社員番号></社員>
<社員><氏名>今村 次郎</氏名>
        <社員番号>26535</社員番号></社員>
        .....
</社員名簿>
```

図 3 社員名簿

(2) 個々の価格と価格合計との整合性チェック

購入仕様書(図 1)の個々の"購入品"エレメントの内容中の"価格"エレメントの内容の合計が、"価格合計"エレメントの内容に等しくなければならない。

(3) 価格合計と検印との整合性チェック

購入仕様書(図 1)の"合計価格"エレメントの内容が 200000 以上の場合は、"検印"エレメントの内容が「あり」でなければならない。

(4) 製品分類と必須仕様項目の整合性チェック

購入品の分類ごとに定まる物品購入に必要な仕様項目(例えば、ソフトウェアの場合は、対象計算機やOS 等)が購入仕様書中に記載されているかどうかをチェックする。すなわち、購入仕様書 (図 1)の個々の"購入品"エレメントの内容に対して、"名称"タグ中の"分類"アトリビュートの値を必須仕様項目リスト(図 4)中の"分類名"エレメントの内容としてもつ"必須項目名"エレメントの内容の各々は、ある "項目名"エレメントの内容と等しくなければならない。

```
<必須仕様>
<項目><分類名>ソフトウェア</分類名>
        <必須項目名>対象計算機</必須項目名>
        <必須項目名>OS</必須項目名></項目>
<項目><分類名>計算機</計算機>
.....
</必須仕様>
```

図 4 必須仕様項目リスト

(5) 仕様項目の値と必須項目制約リストの整合性チェック

購入仕様書仕様項目の値が、必須仕様項目制約リストの条件(対象計算機としては、ワークステーション、パソコンから選択する等)を満たしているかどうかをチェックする。すなわち、購入仕様書(図 1)の各々の"項目"エレメントにおいて、"項目値"エレメントの内容が 、"項目名"エレメントの内容を仕様項目制約リスト(図 5)の"必須項目名"の内容としてもつ"項目"エレメントの"列挙"エレメントの内容のいずれかと等しくなければならない。

```
<仕様項目制約>
<項目><必須項目名>対象計算機</必須項目名>
        <値の範囲><列挙>ワークステーション</列挙>
                <列挙>パソコン</列挙>
                .....
                <列挙>その他</列挙></値の範囲>
</項目>
<項目><必須項目名>OS</必須項目名>
        <値の範囲><列挙>Windows3.1</列挙>
                <列挙>Windows95</列挙>
                .....</値の範囲></項目>
.....
<仕様項目制約>
```

図 5 仕様項目制約リスト

3.3 内容検証機能のSGML文書処理ツールへの組み込み

以下に内容検証機能を組み込んだツール例を示す。

(1) 内容検証機能付き SGML 文書自動登録ツール

SGML 文書のエレメントの内容をデータベースへ登録する前に、データ型、データ長、そしてデータ間の制約関係などのデータ辞書が規定する制約条件をチェックする。

(2) 内容検証機能付き SGML 文書エディタ

SGML 文書の編集時に、検証機能を呼び出すことで、検証用知識と矛盾した入力をした場合には、エラーメッセージまたはガイダンスを提示する。

4. 内容検証処理の実現方針

3.2 で述べたように、内容検証処理では、検証で

必要とされるSGML文書中のエレメントを抽出し比較する処理が中心となる。本章では、文書中からある条件を満たすエレメントの内容を抽出し比較するための演算を表現する文書構造問い合せ言語を導入することにより、内容検証処理を簡潔に記述する方式の実現方針を述べる。

4.1 文書構造の問い合せ言語によるアプローチ

基本方針は、内容検証用知識（3.2では社員名簿や必須仕様項目リスト等に相当）を検証対象の文書形式と同じSGMLで表現することにより、SGML文書構造問い合せ言語によって、内容検証処理を統一的かつ簡潔に記述することである。以下に文書構造の問い合せ言語の設計方針を示す。

(1) ラベル付き木のリストを基本データ構造とする。

図6に示すように、SGML文書をDTDに従って構文解析した結果は、タグをラベルとしてもつ木構造として表現できる（**ラベル付き木**と呼ぶ）。ラベル付き木の部分木（リーフも含む）もラベル付き木として扱う。但し、同じタグが繰り返し出現する場合には出現順序もラベルに含めることにより、ラベル名の一意性を保つものとする。

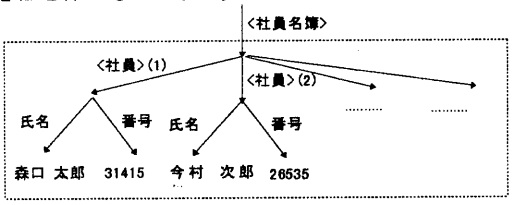

図 6 SGML文書（社員名簿）の解析結果の木構造表現

(2) 問い合せ言語はラベル付き木リストを操作・比較する演算式から構成される。

問い合せ言語を構成する基本演算の結果もラベル付き木リストとすることで、基本演算をネストさせて新たな演算を定義できるようにする。すなわち、関係型データベースの問い合せ言語が表（タプルの集合）を基本データ構造とし、基本演算の結果を表とすることで、問い合せのネストを可能にしたように、ラベル付き木のリストを基本データ構造とし、問い合せのネストを許す問い合せ言語を設計する。

4.2 内容検証機能の組み込みシステムの構成

内容検証を組み込んだシステムでは、SGML文書のエレメント内容を直接利用するなど、内容検証機能が扱うデータの中身を直接参照する必要がある。そのため、問い合せ言語の関数ライブラリ（C言語ライブラリを想定）を提供し、図7に示す構成により外部プログラムから利用できるようにする。

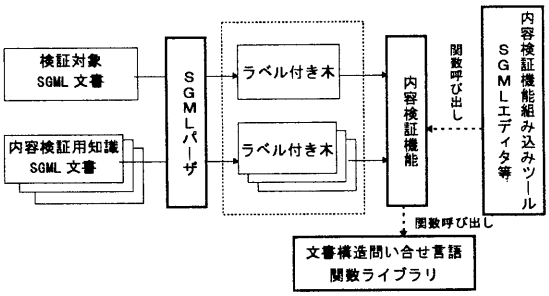

図 7 内容検証機能の呼び出し方式

また、本方式では、3.2で述べたように内容検証用知識をSGML文書で表現することで、内容検証用知識を内容検証処理の記述と分離しているので、以下の点で内容検証用知識の保守性を高めることができる。

・内容検証用知識のDTDに変更がない限り、内容検証用知識の変更があっても、内容検証処理の記述を変更する必要はない。

・問い合せ言語ではDTD中のエレメント名やアトリビュート名を用いて内容検証用知識にアクセスするので、内容検証用知識のDTDに変更があった場合にも、それに伴って変更すべき内容検証処理の記述部分を限定しやすい。

5. SGML文書構造の問い合せ言語

SGML文書構造の問い合せ言語の構文と意味を示す。

5.1 問い合せ言語の構文

問い合せ言語は、以下のBNFで定義される式QLの集合である。

```
QL ::= QLVal | QL . ElName | QL.*
     | QL ! ElName.AttrName | QL where Cond
     | QL ⊂ QL | QL ¬⊂ QL
Cond ::= QL = QL | QL contain "STR"
     | Cond ∧ Cond | Cond ∨ Cond | ¬Cond
/* QLVal はラベル付き木リストを指し示す変数。
   ElName はエレメント名。AttrName はアトリビュート名。
   STR は文字列。*/
```

5.2 問い合せ言語の意味

問い合せ言語は、エレメントアクセス演算子(.)、アトリビュートアクセス演算子(!)、条件指定演算子(where)、そして二つの包含演算子(⊂, ¬⊂)をもつ式から構成される。各々の演算子は、ラベル付き木リストを引数としラベル付き木リストを返す関数である。

(1) **エレメントアクセス演算式(QL . ElName)**

QLの意味するラベル付き木リストの各々の要素に対して、ルートからラベルElNameをたどることにより得られるラベル付き木をとりだし(複数の場合もある)、それらをつないだリストを返す。QLにラベル付き木が代入された場合は、1要素からなるラベル付き木リストと解釈する(他の演算式も同様)。

例：xを図 6に示すラベル付き木とするとき、

- 式 x.<社員名簿>は図 6の点線で囲まれたラベル付き木を要素とするリストを意味する。
- 式(x.<社員名簿>).<社員> は、図 8に示すラベル付き木リストを意味する。

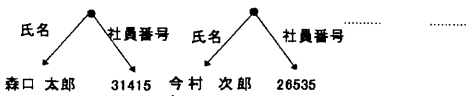

図 8 式(x.<社員名簿>).<社員> の意味

ElNameが*の場合(式QL.*)は、ラベル付き木リスト中の各々の要素のすべての部分木からなるリストを返す。

(2) **アトリビュートアクセス演算式(QL ! ElName.AttrName)**

式 QL の意味するラベル付き木リストの各々の要素に対して、ルートを始点とするラベル名 ElName を持つタグ中でアトリビュート名 AttrName を持つものの値を取り出し、それらをつないだリストを返す。

(3) **条件指定演算式(QL where Cond)**

式QLが意味するラベル付き木リスト中の要素で条件式Cond(TRUEかFALSEを返す関数)を満たすもののリストを返す。但し、whereの左辺のQLがcond中に出現する場合は、 cond中の QLには whereの左辺の QLの意味するリストの要素がバインドされるものとする。QL1=QL2は、式QL1と式QL2が意味するラベル付き木が等しい時にTRUEを返し、そうでない場合はFALSEを返す。QL contain "STR"は、式QLが意味するラベル付き木が文字列STRを含む時にTRUEを返し、そうでない場合はFALSEを返す。演算子∧∨¬の意味はブール代数の対応する演算に同じである。

(4) 包含演算式

(4-1) **⊂包含演算式 (QL1 ⊂ QL2)**

QL1 の意味するラベル付き木リスト中の要素で式QL2 の意味するラベル付き木リストに含まれるものを取り出し、それらをつないだリストを返す。

(4-2) **¬⊂包含演算式 (QL1 ¬⊂ QL2)**

式QL1の意味するラベル付き木リスト中の要素で式QL2の意味するラベル付き木リストに含まれていないものを取り出し、それらをつないだリストを返す。

6. 問い合せ言語による内容検証処理の記述詳細

3.2 で例示した内容検証処理に対する 5. の問い合せ言語による記述を示す。

(1) 申請者と社員名簿の整合性チェック

申請者と社員名簿の整合性は、以下の破線部分の式がTRUEであることをチェックすればよい。

```
1. x := sgml_parse("購入仕様書");
2. y := sgml_parse("社員名簿");
3. x.<購入仕様書>.<申請者> =
4. (x.<購入仕様書>.<申請者> ⊂ y.<社員名簿>.<社員>)
/* sgml_parseはsgml文書を引数として、sgml文書の構
   文解析結果得られるラベル付き木を返す関数。
   =は等号演算子。:=は代入演算子。*/
```

4行目では、購入仕様書の申請者の社員名簿での登録の有無を判定する。その判定中では、木としての等しさを用いるので、記述中には氏名や社員番号といった木の内部構造を参照する必要がなく、簡潔な記述ができる。

(2) 個々の価格と価格合計との整合性チェック

申請者と社員名簿の整合性は、以下の記述の破線部分の式がTRUEであることをチェックすればよい。

```
1. x := sgml_parse("購入仕様書");
2. sum(x.<購入仕様書>.<購入品>.<価格>)
     = x.<購入仕様書>.<合計価格>
/* sumは、ユーザが定義したリストの要素の合計を返す関数 */
```

ラベル付き木のデータ構造をプログラムライブラリとしてユーザに開放することにより、sumのようにユーザが定義した関数と併用して、応用に即した

検証処理を記述することができる。

(3) 価格合計と検印との整合性チェック

申請者と社員名簿の整合性は、以下の記述の破線部分の式がTRUEであることをチェックすればよい。

```
1. x := sgml_parse("購入仕様書");
2. if(x.<購入仕様書>.<購入品>.<合計価格> > 200000){
       x.<購入仕様書>.<検印> = "あり"}
```

(4) 製品分類と必須仕様項目の整合性チェック

製品分類と必須仕様項目の整合性は、以下の破線部分の式がTRUEであることをチェックすればよい。

```
1. x := sgml_parse("購入仕様書");
2. x1 := (x.<購入仕様書>).<購入品>;
3. y := sgml_parse("必須仕様項目リスト");
4. y1 := y.<必須仕様>.<項目>;
5. while ( (z := car(x1)) ≠ NULL ) {
6.   (y1 where y1.<分類名> = z!<名称>."分類").<必須項目名>) =
7.   ((y1 where y1.<分類名> = z!<名称>."分類").<必須項目名>
8.       ⊂ z.<仕様>.<項目>.<項目名>)
9.     x1 := cdr(x1);}
/* carはリストの先頭の要素を返す関数。
   cdrはリストから先頭の要素を除いたリストを返す関数。
   NULLは空リスト。*/
```

基本演算式をネストさせて新たな演算式を構成できるので、一つの演算式で取り出した情報を用いて、ある条件を満たす部分構造を取り出す処理を簡潔に記述できる。

(5) 仕様項目の値と必須項目制約リストの整合性

仕様項目の値と必須項目制約リストの整合性は、以下の破線部分の式が TRUE であることをチェックすればよい。

```
1. x := sgml_parse("購入仕様書");
2. x1 := (((x.<購入仕様書>).<購入品>).<仕様>).<項目>;
3. y := sgml_parse("仕様項目制約リスト");
4. y1 := y.<仕様項目制約>.<項目>;
5. while ( (z := car(x1)) ≠ NULL ) {
6.   z.<項目値> =
7.   (z. <項目値> ⊂
8.   ((y1 where y1.<必須項目名> = z.<項目名>).<値の範囲>).<列挙>)
9.   x1 := cdr(x1);}
/* 2行目は、 x1 := x.*.<項目>; でも同じ。*/
```

7. 考察

本研究と既存技術との差異について考察する。

7.1 SGML 文書変換技術

(1) SGML 変換ツール([3][4])

OmniMark[3]と AEsop[4]は、変換のための命令をエレメント名毎に指定する変換言語を提供する SGML 文書変換ツールである。変換処理はこの変換言語で書かれた命令を解釈実行することによりなされる。エレメント毎の変換命令の記述は、本稿の問い合わせ言語では、式 QL.*.ElmName で得られたラベル付き木リストの各要素に対する操作命令の記述として表現できる。

OmniMark の特徴は、文字列のパターンマッチ処理であり、AEsop の特徴は、SGML 文書の木構造を操作する機能と複数の変換処理をパイプで連結させる機能を提供することにより変換命令の記述能力を向上させたことにある。

本稿では、検証処理を簡潔に記述するために、従来の SGML 文書変換ツールでは困難であった以下の点を解決した。

・文書の部分構造間のグローバルな制約の記述

従来ツールでは、エレメント毎に命令を記述するので、エレメント間の比較のように複数の部分構造を対等に扱う処理を簡潔に記述できなかった。

・複数文書にまたがる処理の記述

従来ツールでは、1 文書から 1 文書の変換を想定しており、複数文書の文書構造を扱う機能がなかった。

(2) DSSSL([5])

ISO 規格である DSSSL(Document Style Semantics and Specification) [5]は, SGML 文書変換のための変換言語と SGML 文書の印刷用の割付けのためのスタイル言語を規定している。変換言語は Lisp の一方言である Scheme で記述されており、SGML 文書を構文解析した結果得られる木構造(grove と呼ぶ)を操作する問い合わせ言語として、Standard Document Query Language(SDQL)を規定している。SDQL を用いれば、SGML 文書構造の操作に必要な基本関数とリスト処理機能を用いて、割付け処理で必要とされる複

雑な変換処理を記述できる。しかし、内容検証処理の記述言語としては以下の問題点がある。

・SDQL が提供する基本関数はプリミティブなものに限定されている。
・応用システムで通常用いられる C 言語から容易に呼び出せない。

本稿では、文書構造の問い合せ機能を関数ライブラリとして、C や DSSSL から利用できるように、問い合せ言語を特定のプログラミング言語に依存しないように、抽象的なデータタイプ上の演算として定義することを試みた。実際、ラベル付き木は grove で表現できるので、本稿の問い合せ言語の関数ライブラリを SDQL のコンセプトと矛盾することなく組み込むことができると思われる。

7.2 構造化文書データベースの問い合せ言語

データベースの分野では、構造化文書に対する問い合わせ言語の研究[6][7]がなされている。

[6]では、オブジェク指向データベースの問い合わせ言語(O_2SQL)を拡張することにより SGML 文書の問い合わせを表現する方法を提案している。本稿は、内容検証処理を簡潔に記述するために新たな演算子⊂や¬⊂を導入した点が[6]と異なる。

[7]では、文書構造検索と文字列検索を組み合わせた検索式を表現する問い合わせ言語とその意味論である領域代数について述べている。領域代数では、その代数の基本データ構造としては、テキストの領域(テキストの部分文字列)の集合を用いる。すなわち、問い合わせ言語の意味は、問い合わせの条件を満たす文書の部分構造や文字列の領域の集合になる。本稿は、基本データ構造をラベル付き木のリストとする点が[7]と異なる。この差異は両者の目的の差異によるところが大きい。すなわち、内容検証のための問い合せ言語では、文書の部分構造がラベル付き木として等しいかどうかを問題にするので、問い合せ言語を解釈する基本データ構造としては木としての構造を持たせる必要がある。

8. まとめ

本講では、SGML 文書の内容検証の要求機能の検討に基づいて、SGML 文書の部分構造を取り出す基本演算式のネスト構造を許す問い合せ言語を提案した。そして、本問い合せ言語によれば、内容検証用知識を検証対象文書と同じ SGML 形式で表現することにより、内容検証処理を統一的かつ簡潔に記述できることを示した。今後の課題は以下の通りである。

(1) SGML 文書構造の問い合せ言語の評価

本稿で考察した内容検証は、データベースの分野では、インテグリティ制約([8])として扱われているものに相当する。しかし、文書の応用システムとデータベースの応用システムでは、必要とされるデータに対する制約が異なることが予想される。複数の SGML 応用システムで必要とされる内容検証処理を分析・パターン化し、今回提案した問い合わせ言語の記述能力を評価することは今後の課題である。

(2) 高速な SGML 文書構造操作アルゴリズムの開発

本稿では、文書構造の問い合わせ言語の構文と意味を定義することを中心とした。しかし、検証用知識が膨大になれば、リアルタイムでの応答が要求されるシステムに適用する際には、高速な SGML 文書の構造操作処理が必須となる。

参考文献

[1] 文書記述言語 SGML, JISX4151 (1992).

[2] Eric van Herfwijnen: 実践 SGML, p203-216, 日本規格協会 (1992)(原著は 1990).

[3] OmniMark Programmer's Guide Version 2, Exoterica Corporation (1993).

[4] 高橋 亨, 松本 正義 : SGML 文書を対象とする文書構造操作言語の提案, 49 回情処全大, 3 分冊, pp265-266(1994).

[5] DSSSL:Document Style Semantics and Specification Language, ISO/IEC 10179 (1996).

[6] V. Christophides, S. Abiteboul, S. Cluet and M. Scholl : From Structured Documents to Novel Query Facilities, SIGMOD`94, pp313-324, ACM (1994).

[7] M. Consens and T. Milo: Algebras for Querying Text Regions, PODS`95, pp11-22, ACM (1995).

[8] 滝沢 誠: データベースシステム入門技術解説, pp60-84, SRC (1991).