

## 6 Q-5

## 文書処理統合環境D<sub>I</sub>E<sub>T</sub>における 外部データ参照記述形式

佐々木 貴幸 山川 正

キャノン(株) 情報システム研究所

### 1. はじめに

ハードウェア技術の発達に伴い、ワークステーション上で文字ばかりでなく図形や画像、音声といった様々なメディアのデータが取り扱えるようになりつつある。また、それらのデータの格納管理形態も様々である。

本稿では、論理構造付き文書データ中に、格納形態に依存しない形式で様々な外部データを混在させるための記述形式を定めたので報告する。

### 2. D<sub>I</sub>E<sub>T</sub>形式での外部データ参照

我々が開発を進めている文書処理統合環境D<sub>I</sub>E<sub>T</sub>(Document Integration Environment and Tools)<sup>1),2)</sup>では、SGML<sup>3)</sup>に基づいて、テキストベースで論理構造情報を付加して文書を表示する文書記述形式(D<sub>I</sub>E<sub>T</sub>形式)を定めた<sup>2)</sup>。

D<sub>I</sub>E<sub>T</sub>形式で記述された文書データ(D<sub>I</sub>E<sub>T</sub>ary)に、図形・画像データといったタイプの異なるデータを混在させるための一つのアプローチとして、それらのデータファイルをD<sub>I</sub>E<sub>T</sub>aryに結合する拡張D<sub>I</sub>E<sub>T</sub>aryの概念を先に提案した<sup>3)</sup>。この提案では、文書要素の格納先をファイルに限定していた。また、SGMLにおいても、参照可能な外部データはSGMLを利用するシステムのデータ格納形態に依存している。

そこで、データの格納先に依存することなく、文書データ中に様々な外部データを混在させるための記述形式として、SGMLで用いられているエンティティ参照のエンティティ定義に関して、D<sub>I</sub>E<sub>T</sub>形式独自のエンティティ定義を導入した。

この記述形式を採用することにより、格納先に依存せず、テキストベースで外部データ参照が可能な記述を実現できる。また、こういった記述の文書データを用いることによって、文書の表示・印刷以外の処理段階では、外部データとは仮想的な結合として取り扱うことができる。

### 3. D<sub>I</sub>E<sub>T</sub>形式のエンティティ参照記述形式

エンティティ参照は、「エンティティ」と「エンティティ定義」の対によって記述される。(図1参照)

```
<論文>
<# section1
  <#TYPE dietary>
  <#AGENT shell>
  <#REQUEST /bin/cat /usr/ssk/doc/section1>
>
<章> &section1; </章>
</論文>
```

図1 エンティティ参照の記述例

#### 3.1 エンティティ

「エンティティ」は、外部データが参照されるべきポイントに、エンティティ名を用いて以下のような形式で記述する。

例えば、エンティティ名が“section1”のエンティティは、

```
&section1;
```

と記述する。

#### 3.2 エンティティ定義

「エンティティ定義」は、エンティティ名を指標にして、次の三つ組の要素で構成される。

**TYPE** : 参照するデータの型。

(dietary, tiff, ezps, etc)

**AGENT** : 参照するデータの格納管理形態。

(shell, db\_server, オンラインデータ入力装置, etc)

**REQUEST** : データを参照するためのコマンド。

(このコマンドは、AGENTに依存する。)

例えば、エンティティ名“section1”のエンティティ定義は、図1の例では以下のように記述されている。

```
<# section1
  <#TYPE dietary>
  <#AGENT shell>
  <#REQUEST /bin/cat /usr/ssk/doc/section1>
>
```

これは、エンティティ名“section1”によって参照されている外部データが、

- D<sub>I</sub>E<sub>T</sub>ary型のデータであること、
- shellによって格納管理されたデータであること、
- /bin/cat /usr/ssk/doc/section1というコマンドをshellに送ることによってデータの実体を参照できること、

External Data Reference Notation in the Document Integration Environment and Tools (D<sub>I</sub>E<sub>T</sub>)  
Takayuki SASAKI and Tadashi YAMAKAWA  
Information Systems Research Center, Canon Inc.

を表わしている。

つまり、上記のエンティティ定義によって様々な形態に格納された、様々なタイプのデータについての情報をDfTary中に記述することができ、かつ、それによって、エンティティをデータと仮想的に結びつけることができる。

#### 4. エンティティ参照に対応した処理系

エンティティ参照を含むDfTaryを表示・印刷処理するためには、処理の前段階で、エンティティの展開、すなわち、エンティティによって参照される外部データそのものをDfTaryに結合する機能を設ける必要がある。この機能を実現するためには、以下の3つの機能が必要である。(図2参照)

**エンティティ定義解析機能**：エンティティ定義を解析し、「AGENT」と「REQUEST」の内容を外部データ参照機能へ、「TYPE」の内容を参照データ展開機能へ伝達する。

**外部データ参照機能**：エンティティ定義解析機能から得られた「AGENT」と「REQUEST」の内容をキーにして外部データを参照する。参照の結果、得られたデータを参照データ展開機能へ伝達する。

**参照データ展開機能**：エンティティ定義解析機能から得られた「TYPE」の内容をキーにして、外部データ参照機能で参照したデータを、DfT形式に沿った形式で展開する。

この機能は、DfTaryを対象とした文書処理プロセスにおけるプリプロセッサとして位置付けられる。

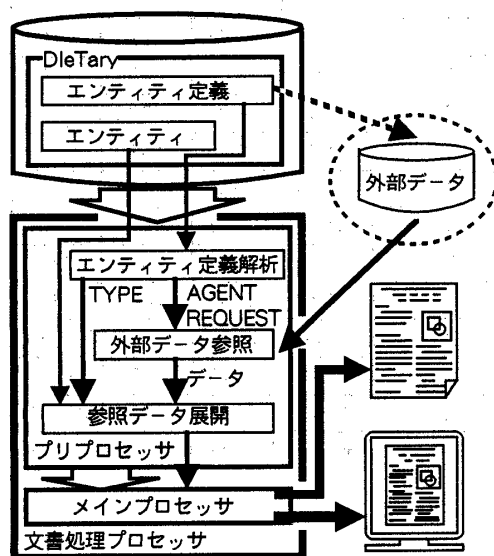


図2 エンティティ参照に対応した処理系

```

<月報>
<# kongetsu_yotei
  <#TYPE dietary>
  <#AGENT db_server>
  <#REQUEST db_refer geppou_db 'month=3' 'yotei'>
>
<# raigetsu_yotei
  <#TYPE dietary>
  <#AGENT db_server>
  <#REQUEST db_refer geppou_db 'month=4' 'yotei'>
>
<# sinten
  <#TYPE dietary>
  <#AGENT db_server>
  <#REQUEST db_refer geppou_db 'month=3' 'sinten'>
>
<月>>3</月>
<今月の予定> &kongetsu_yotei; </今月の予定>
<今月の進展> &sinten; </今月の進展>
<来月の予定> &raigetsu_yotei; </来月の予定>
</月報>

```

図3 エンティティ参照による月報DfTaryの記述例

#### 5. 適用例

エンティティ参照記述形式の適用例として、月間進行状況報告書(月報)の例を示す。

図3は、3月の月報DfTaryである。図3のように記述することによって、データベース参照関数db\_referからなる一連のコマンドを用いて、db\_serverに管理された月報データベース(月報DB)中のデータを、<今月の予定>、<今月の進展>、<来月の予定>の内容として参照することができる。ちなみに、月報DBには、各月ごとに「予定」と「進展」が格納されている。

#### 6. むすび

論理構造付き文書データ中に、格納形態に依存しない形式で外部データを混在させるために、DfT形式を拡張し、エンティティ参照に関するDfT形式独自のエンティティ定義を導入した。これにより、様々な形態に格納された、様々なタイプのデータを論理構造付き文書データの文書要素として取り扱うことができるようになった。

今後は、エンティティ参照を含むDfTaryの処理を行なう文書処理プロセッサ、および、これに付随する処理ツールの開発を進める予定である。

#### 参考文献

- 1) 山川, 川端, 田村: “キャノンFODs計画とその統合文書処理環境”, 情報処理学会 卓上出版シンポジウム報告集, pp. 223-232 (1988.7).
- 2) 川端, 山川, 出井, 田村: “文書処理ワークステーションと文書アーキテクチャ”, 電子通信学会 ワークショップ — 電子出版の現状と課題, pp. 53-59 (1989.4).
- 3) 山川, 清水, 杉岡, 佐々木: “文書処理統合環境DfTにおける図形・画像データの取扱い”, 情報処理学会全国大会第40回, 6P-10 (1990.3).
- 4) ISO 8879: Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML) (1986).