4C−2

文書データ処理言語DPL(2)

— 記述形式の簡素化 —

長島 正明　山川 正
キヤノン(株) 情報システム研究所

1. はじめに

我々は，様々な文書処理システムを作成できる環境を構築することを目的とし，文書処理統合環境DIET(Document Integration Environment and Tools)の研究を行なっている[1][2][3]．そして，SGML[4]を基本にして，論理構造を付加した文書データの記述形式(DIET形式)を提唱した[2][3]．また，DIET形式で記述された文書データ(DIETary)を加工するためのメタツールである文書データ処理言語DPL(DIET Processing Language)を考案した[5]．

DPLは，ある型のDIETaryから，別な型のDIETaryやフォーマッタへの入力データに変換したり，特定文書要素の内容を抽出する等の処理が，容易に行なえることから，その有効性は高い．

今回は，文書データ処理言語DPLの記述に関する簡素化について述べる．

2. DPL記述の検討

2.1 従来の記述

DPLでは，DIETaryの各文書要素ごとに，処理の記述を行なう．その記述は，従来，

/文書要素名/{開始処理}{内容処理}{終了処理}

であった．すなわち，文書要素を指定するための記述と，その文書要素に対する処理(開始処理，内容処理，終了処理)の記述から構成されている．文書要素の指定の記述には，文書要素の階層構造も記述でき，「～の下にある～」，「～のすぐ下にある～」，「最上位にある～」等が表現できる．各処理の記述には，'if'，'while'等の制御文，変数への代入文や四則演算等の式の文，標準出力への出力文が用意されており，プログラミングが可能である．

しかし，従来の記述には，以下のような問題点があった．

- タグのつけ替え処理の記述が煩雑である．
- 変数の取り扱いに注意が必要である．

前者は，出力文の記述が煩雑であることに起因する．DIETaryの加工処理のうち，他のDIETaryに変更したり，あるいはLaTeXのデータに変換する等の処理がよく行なわれる．このような処理は，DIETary中のタグを，単に他の文字列に置き換えることで可能なため，その記述は，出力文を多く利用する．従来，出力文は，'write_str(式)'と記述し，多用するには適していない構文であった．

また，後者は，DPLの変数の取り扱いに起因する．DPLでは，変数を用いて，プログラミングが可能であるが，大域変数のみを取り扱う従来の方式では，上位の文書要素の処理で扱っている変数の値を，下位の文書要素の処理でこわすことがある．そのため，変数の取り扱いには，十分な注意を要していた．

2.2 記述の簡素化

上に述べた問題点を解決し，より簡潔に記述が行なえるようにするため，以下の方針のもと，DPL記述の簡素化を行なった．

- DPLの構文を変更し，タグのつけ替え等の単純処理の記述の簡素化を図る．
- 局所変数を導入することにより，変数の取り扱いを容易にする．

2.2.1 DPL構文の改良

DPL構文の一部をバッカス記法で記述したものを，図1に示す．今回，変更したのは，次の5つの構文である．

<出力文>
<処理部>
<開始処理>　<内容処理>　<終了処理>

出力文が多用される処理記述の手間を軽減するために，出力文は，入力が容易な'<<'で記述する．また，開始，内容，終了の各処理の記述は，各処理が単文で構成されている時は，'{'，'}'を記述する必要をなくした．

2.2.2 局所変数の導入

上に述べたように，階層構造を持った文書データに対し，変数に値を代入する等の処理を記述するには，大域変数だけでは不十分である．そこで，一つの文書要素の処理記述の範囲だけに有効な局所変数を導入し，名前によって，大域/局所変数を使い分ける．以下に，各変数名を正規表現を用いて表す．

<大域変数> → [a-zA-Z][a-zA-Z0-9_]*
<局所変数> → '$' [a-zA-Z0-9_]+

A Document Processing Language DPL(2) — Simplification of notation —
Masaaki NAGASHIMA, Tadashi YAMAKAWA
Information Systems Research Center, Canon Inc.

```
<DPL構文> → <指定部><処理部> |
            <DPL構文><指定部><処理部>
<指定部> → '/' <文書要素名> '/'
<処理部> → <開始処理><内容処理><終了処理>
<開始処理> → <文>
<内容処理> → <文>
<終了処理> → <文>
<文並び> → <文> | <文並び><文>
<文> → <式> ';' |
       'if' '(' <式> ')' <文> |
       .....
       <出力文> |
       <複合文> |
       <空文>
.....
<出力文> → '<<' <式> ';'
<複合文> → '{' <文並び> '}' |
           '{' '}'
<空文> → ';'
.....
```

図 1 DPLの構文(一部)

ただし，'$' の後に数字が来るものは，予約語として取り扱われ，特に，'$0'は，各処理で取り扱うデータチャンクをさす．ここでいうデータチャンクとは，文書要素で区切られる文字列，もしくは，文書要素に対する処理のリターン文字列である．

3. DPLの記述例

本稿で述べたDPL記述仕様での記述例を図2に示す．図2(3)は，(1)のDIeTaryを，(2)のLaTeXへの入力データに変更するためのDPL記述の一例である．DPLの記述の中で，変数Itemizeは，局所的な変数でなければならない．比較のために，従来のDPL記述を(4)に示す．

従来の記述に比べ，出力構文が簡素化され，，また，開始，終了，内容処理の記述が単文で記述される時は，'{'，'}' を記述する必要もなくなり，記述性が向上した．

また，局所的な変数を使用する場合，従来は，変数を配列にすることにより，各階層の文書要素だけに有効な変数を設けて対処していたため，階層レベルの管理を行なわなければならず，その記述は煩雑なものになっていた．今回は，局所変数の導入により，その記述が簡素化された．

4. おわりに

今回は，DPL記述の簡素化について報告した．処理部，特に出力文の記述の改良を行ない，また，大域/局所変数を使い分けることにより，記述性の向上を図った．

今後は，本記述仕様に基づいたDPL処理系を作成し，実用化について検討していく．

```
<section>AAA</>
<content>
  <itemize>
    <item>bbb bbb</>
    <itemize>
      <item> ccc ccc</>
    </>
  </>
</>
```

(1) 対象データ (DIeTary)

```
\section{AAA}
\begin{itemize}
\item bbb bbb
\begin{itemize}
\item ccc ccc
\end{itemize}
\end{itemize}
```

(2) LaTeX への入力データ

```
/section/ << "\\section{";
          << $0;
          << "}\n";
/content/ ; << $0; ;
/itemize/ $Itemize="\\begin{itemize}\n";
          $Itemize=$Itemize $0;
          return($Itemize
                 "\\end{itemize}\n");
/item/ $Item="\\item ";
       $Item=$Item $0;
       return($Item "\n");
```

(3) DPLによる文書データ変換記述例

```
/section/{write_str("\\section{");}
         {write_str($0);}
         {write_str("}\n");}
/content/{level=0;}
         {write_str($0);}{}
/itemize/{level++;
          Itemize[level]=
            "\\begin{itemize}\n";}
         Itemize[level]
           =Itemize[level] $0;
         {level--;
          return(Itemize[level+1]
                 "\\end{itemize}\n");}
/item/ Item[level]="\\item ";
       Item[level]=Item[level] $0;
       return(Item[level] "\n");
```

(4) DPL記述例 (従来例)

図 2 DPLの記述例

参考文献

1) 山川,川端,田村: "キヤノンEoDS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,川端,田村: "OA業界から見たDTP", 情報処理, Vol.31, No.11, pp.1508-1517 (1990).
4) ISO 8879: Information Processing - Text and Office Systems - Standard Generalized Markup Language(SGML) (1986).
5) 長島,山川: "文書処理統合環境DIeTにおける文書データ処理言語", 第42回全国大会講演論文集(3), pp308-309 (1991).