再帰的SGML文書整形システム

4L−7

山川正　長島正明
キヤノン(株) 情報システム研究所

1 はじめに

文書記述言語 Standard Generalized Markup Language (SGML[1]) による文書の構造化記述を応用して，文書のデータベース化が実現されるようになってきた[2]．こういった文書データベースでは，文書を登録する時点で，文書の構造化を図る必要がある．そこで我々は，構造付文書の簡易入力方法として，常用文書記述形式(第0版)を提唱した[3]．

しかし，第0版では，多重の入れ子になった文書要素に関する適合検査はできない．論理構造付き文書加工フェーズに移る段階でSGML文書として構造解析し，初めて適合検査を適用できる．この問題の解決を図るため，常用文書記述形式を，SGMLに準拠した簡易記述形式(常用文書記述形式第1版)に発展させた．そして，このSGMLで記述した文書を入力対象とし，整形結果もまたSGML文書データとして出力するツール「再帰的文書整形システム」を開発した．

本稿では，再帰的SGML文書整形を実現するするときの問題点を挙げ，この解決方法について述べる．

2 再帰的SGML文書整形における問題点

常用文書記述形式の目的は，文書構造のためのマーク付けの意識を最小限にとどめながら，構造付き文書を記述する仕組みを提供することである．第0版において,「見出し」と「段落」の並びを基本に置き，これらと多用される入れ子の箇条書きに関して，明示的なスタートタグ・エンドタグを用いない記述形式を定めた．

この記述形式の基本を変えることなくSGML化するためには，短縮参照(Short Reference)の機能を用いる．すなわち，文書要素に応じて，その内容記述における特定の文字列をタグに置換する機能を用いて，暗に示される構造を解析する．たとえば,「空行を段落のスタートタグに置換する」という指定を行う．しかし，以下の問題が生じる．

入れ子の箇条書き記述 箇条書きという要素の中に箇条書きが含まれる場合，入れ子を検出するには開始と終了のタグを記述しなければならなくなる．

短縮参照との混同 文書整形前では短縮参照の置換対象とならない文字が，文書整形した後に置換対象となる場合がありうる．たとえば，行の中ほどに「【」が記述されていても，文書内容の文字とみなされるが，整形後，行の先頭に置かれると,「見出し」の開始とみなされてしまう．

代替文字の処理 エンティティ参照による文字の代替を，そのまま，整形してしまうと，整形後はその文字がタグと解釈されてしまうことがある．たとえば,「＜」の変わりにエンティティ参照「<」を用いることがあるが，整形後「＜」と出力し，この直後に英字が続いた場合，スタートタグと解釈されてしまう．文字「＜」を「<」として出力するようにして回避すると，行末の「＜」等タグとしてではなく文字と扱われるときに，整形後「<」に置き換えられてしまうという問題が発生する．

表1: 基本文書要素とその記述形式

要素	GI	要素の説明	記述形式
段落	P	段落内要素の並びを内容として持つもっとも基本的な文書要素.	次の段落との間は，空行で区切る.
見出し	H0	見出しタイトル(HT0)の並び．最初の見出しタイトルを見出しの本体として取り扱い，以降は，副見出しとして取り扱う.	行頭の「【」に続いて見出しタイトルを置く．見出しタイトル間は空行で区切り，最後に「】」を置き改行する．先頭の「【」の直前の「:」の数で深さを表す.
箇条書き	L0	箇条書き項目(I0)の並び．箇条書き項目は箇条書き見出し(IH0)に続くひとつ以上の箇条書き段落(IP0)を内容として持つ．箇条書き見出しは省略可能.	各箇条書き項目ごとに，行頭に「・」を置く．箇条書き見出しを内容として持つ場合は，その内容を「[」と「]」とで囲んでこの直後に置く．続いて箇条書き段落の内容を記述する．先頭の「・」の直前の「:」の数で深さを表す.

3 再帰的SGML文書整形の実現方法

3.1 入れ子の箇条書き記述における問題解決

箇条書きの入れ子の深さに応じて，別の文書要素とし，箇条書きの項目の内容を一般の段落の並びとせず，深さに応じた箇条書き用の段落とする文書構造を採用した．そして，深さに応じた要素ごとに，短縮参照の定義を切り替えることによりエンドタグを明示することなく，箇条書きの入れ子を検知することを可能にした．基本文書要素とその記述形式を表1に示し，文書型定義を図1に示す．

たとえば，一般の段落では，行頭の「・」に対して，第0レベルの箇条書き(L0)，項目(I0)，箇条書き段落(IP0)が開始するように定義した．この置換が行われると，文脈は箇条書き段落(IP0)になる．ここでは，行頭の「・」は，第0レベルの項目(I0)，箇条書き段落(IP0)が開始するように定義した．すなわち，箇条書き(L0)の開始は指定しない．また，行頭の「:・」は，第1レベルの箇条書き(L1)，項目(I1)，箇条書き段落(IP1)が開始するように定義した．一方，第1レベルの箇条書き段落(IP1)では，第1レベルの箇条書き(L1)を終了させ，直後に第0レベルの項目(I0)，箇条書き段落(IP0)が開始するように定義した．

Recursive SGML Document Formatting System
Tadashi YAMAKAWA, Masaaki NAGASHIMA
Information Systems Research Center, Canon Inc.

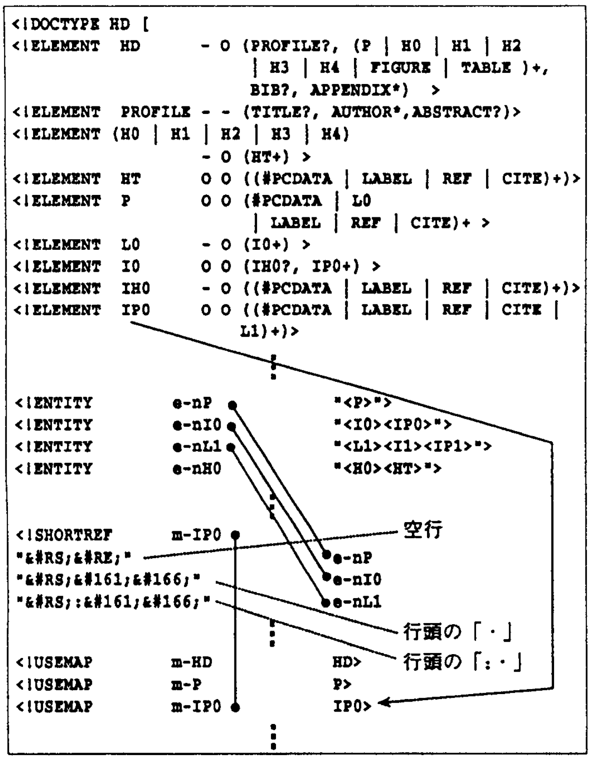

図 1: 再帰的 SGML 文書整形用文書型定義

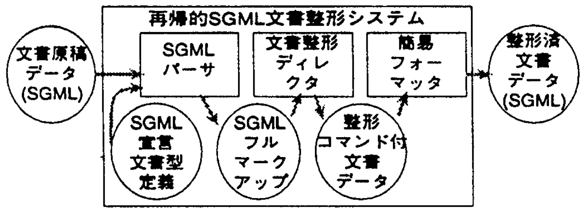

図 2: 再帰的 SGML 文書整形システムの構成

3.2 短縮参照との混同における問題解決

行中に含まれ，行頭・行末に置くと制御記号として取り扱われる文字は，整形プログラムにおいて，それぞれ行頭禁則文字・行末禁則文字とすることにより，行の中ほどに配置されるようにした.

3.3 代替文字の処理における問題解決

データとしての文字と等価のエンティティの定義を，文字としてではなく SDATA として定義した．そして文書整形時に，SDATA として定義されたエンティティを，アンパーサンド記号・エンティティ名称・セミコロン記号で構成されるワードとして扱うことにした.

3.4 システム構成

再帰的SGML 文書整形システムの構成を図 2に示す.

SGML パーサは，フリーソウトウェアである「sgmls」を用いた．なお，標準外の短縮参照用のデリミタを用いるため，SGML 宣言は図 3のようにデリミタ定義を加えた.

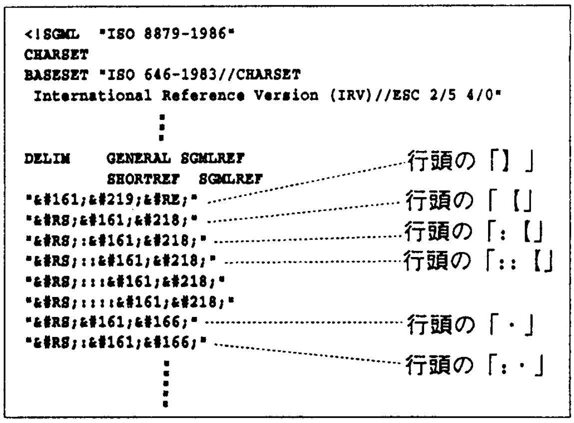

図 3: 短縮参照のためのデリミタ定義の例

ただし,「sgmls」は，デリミタ定義の追加機能を実現していないため，これらのデリミタを基本的にサポートするように改造を行った.

文書整形ディレクタは，文書要素の種類に応じて，その整形方法を整形コマンドの形で文書データに付加するもので，DIeT Processing Language (DPL) [5] を用いて作成した． DPL もまた，DIeT の基本ツールである．文書整形ディレクタのプログラムサイズは，約 200 行である.

簡易フォーマッタは，文字端末を対象にして，字下げ，左右マージン設定し，ワードラップ，禁則処理等を適用した整形を行うプログラムである．これは，文書処理統合環境 DIeT[4] の基本ツールとして，すでに開発されたものを利用した.

4 おわりに

SGML に準拠した構造付文書の簡易記述形式及び，本形式で記述された文書を本形式を守って整形する「再帰的文書整形システム」の実現方法について述べた.

本システムにより，既存の文書エディタを利用しながら，監修者などをふくめ複数の担当者で，文書の校正を進める段階から，構造化文書を取り扱うことが可能になる．また，SGML の基本機能に基づくため，さまざまなシステムとの情報交換にも対応することができる．本稿も本システムにて作成し，TeX への変換ツールを利用してカメラレディコピーを自動作成した.

参考文献

[1] ISO 8879 : Information Processing – Text and Office Systems – Standard Generalized Markup Language (SGML) (1986).

[2] L. R. Reynolds and S. J. Derose : Electronic Books, Byte, Vol. 17, No. 6, pp. 263-268 (1992).

[3] 佐々木, 山川 : 構造付文書の簡易入力方法, 情報処理学会第 44 回全国大会論文集, No.3, 4C-3, pp.281-282 (1992).

[4] 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).

[5] 長島, 山川 : 文書データ処理言語 DPL(2) — 記述形式の簡素化—, 情報処理学会第 44 回全国大会論文集, No.3, 4C-2, pp.279-280 (1992).