

4C-3

構造付文書の簡易入力方法

佐々木 貴幸 山川 正

キヤノン(株) 情報システム研究所

1. はじめに

文書情報を有効に活用することを目的として、文書を構造化してデータ化する技術が実用化されてきた。これに伴い、構造化文書を作成・編集するために、構造化エディタ¹⁾やSGML²⁾等の文書記述言語を用いて文書入力を行なうツールが開発されている。

ところが、従来の文書作成・編集ツールは、特に最終形態を想定せずに、平易な形態で日常の雑多な情報を蓄積する場合や、文書原稿の素案を作成するような場合には向いていない。これは、従来のツールが最終形態である印刷文書を想定して作成されていることに起因すると考えられる。

本稿では、構造化の意識を最小限にとどめながらも、暗に付けられる構造情報を後の処理に活用できる、文書記述形式(常用文書記述形式)について述べ、その適用例を紹介する。

2. 文書処理フェーズ

一般に、構造化文書を取り扱う文書処理システムの文書処理フェーズ(論理構造付文書処理フェーズ)は、論理構造付き文書作成フェーズ、レイアウトフェーズ、プレゼンテーションフェーズの3つのフェーズに分類される。しかし、基本的にこれらのフェーズ分けは、印刷文書の作成過程をフェーズ分けしたものであり、文書データ入力時点で念頭に置くべき文書の論理構造も、印刷文書の形態に影響されていることが多い。これにより、最終形態を意識せずに文書による情報蓄積を行ないたい場合でも、所定の論理構造に適合するように入力することを強いられることになる。

そこで、テキストデータとして文書入力を行なう際に、特に文書構造を意識することなく、構造付けを実現する文書記述形式(常用文書記述形式)を定めた。

常用文書記述形式の導入により、文書データ作成フェーズは、常用文書データ作成フェーズと論理構造付文書データ編集フェーズの2つに分けられる。(図1参照)

常用文書データ作成フェーズでは、文書の最終形態を意識せずに文書データを作成することができ、しかも、常用文書構造で構造化された文書データとして、情報の蓄積が可能である。

一方、論理構造付文書データ編集フェーズでは、この常用文書データを、組版対象となる論理構造に適合するように加工・編集を行なう。これによって、常用文書

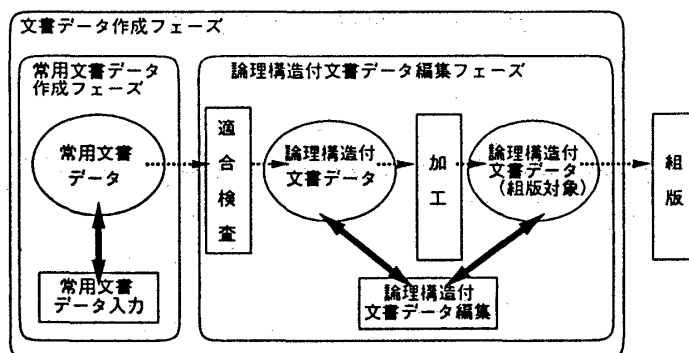


図1 文書処理フロー

データからレイアウトされた文書を作成することが可能になる。また、この論理構造適合検査をとった常用文書データは、論理構造付文書データとして、自動加工の対象になりうる。

3. 常用文書記述形式

文書として、「見出し」と「段落」を区別して表現できるようにすることによって、大方の文書情報を記述可能である。そこで、常用文書記述形式は、「見出し」と「段落」の並びを基本とする。見出しは、「【】と【】」で囲むことによって区切り、段落は、一行以上の空行で区切る。これにより、メモ書きのような、タイトル(見出し)とそれに付随する内容(段落)を、要素名を伴ったタグを明示することなしに記述可能となる。

また、文書内容中に多用される箇条書きに対しては、箇条書き項目開始記号「・」を用いることによって、段落内に記述できるようにした。

これに加え、見出しと箇条書きに深さ(レベル)の情報を付加するために、レベル記述子を「:」を導入した。見出し区切りと箇条書き項目開始記号の直前に置かれたレベル記述子の数によって、それぞれの深さを記述する。また、箇条書きの終了は段落の終了と一致させるような限定を加えることにより、箇条書きの入れ子を、箇条書きのスタート・エンドの明示なしで記述可能である。

文書内容中のプログラム等の記述に関しては、改行を伴う場合(逐次行)と伴わない場合(逐次語)に分け、前者は、「『」のみからなる行、後者は、「『」一字、を区切り記号とし、区切り記号で囲んで表すようにした。なお、この区切り記号自体を、データとしてその内容に含め

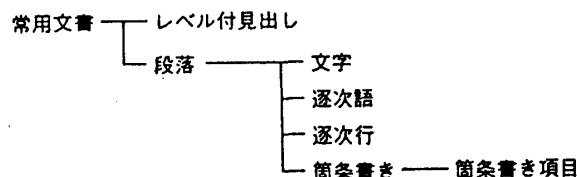


図2 常用文書データの要素の関係

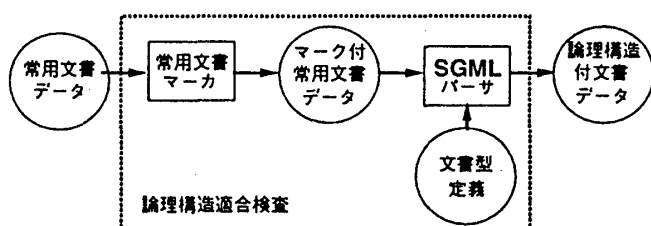


図3 常用文書の処理系

る場合には、区切り記号を二つ連続して記述することによって表現する。

また、常用文書データに対して、更に詳細な構造付けを行なう必要がある場合には、SGMLの標準的な記号付けによって対処する。

常用文書記述形式で記述された文書データ(常用文書データ)の文書要素の関係を、図2に示す。常用文書は、レベル付見出しと段落を任意個並べた構造であり、段落は、一般の文字、逐次語、逐次行、箇条書きの任意個の並びを内容とする要素である。また、箇条書きは、任意個の箇条書き項目の並びを内容とする要素である。なお、箇条書き項目には、段落内の要素を含むことができる。

4. 処理系

常用文書データをプログラムによる加工対象にするためには、特定の型や論理構造に適合した文書データに変換する必要がある。

そこで、論理構造適合検査による処理によって、常用文書データを論理構造付文書データに変換する。ここでの論理構造適合検査では、yaccを用いた常用文書マーカーによって、常用文書データの構造検査とSGMLパーサが処理可能な形式のデータ(マーク付常用文書データ)への変換を行い、次に、SGMLパーサを用いて、マーク付常用文書データを論文型の論理構造付文書データに変換することを行なっている。(図3参照)

これによって、常用文書データによる基本的な文書構造から、指定のDTD(ここでは、論文型のDTD)で用いられる文書要素名の対応関係を変えることが可能である。

【ABCDEF】

:【uvwxyz】

いろはにほへと、ちりぬるを。

わかよたれそつねならむ。

・【あいうえお】 有為の奥山今日越えて。

: ・【カキクケコ】 アサキユメミシ。

・【さしすせそ】 酔るもせず。

(a) 常用文書データの例

<h0><hc>ABCDEF

<h1><hc>uvwxyz

<p>いろはにほへと、ちりぬるを。

<p>わかよたれそつねならむ。

<l><i><ih>あいうえお<ip>有為の奥山今日越えて。

<l><i><ih>カキクケコ<ip>アサキユメミシ。

</l><i><ih>さしすせそ<ip>酔るもせず。

</l>

(b) マーク付常用文書データの例

図4 常用文書の処理例

なお、常用文書データを論理構造付文書データに変換すれば、型変換や内容変換を行なうことが可能になる。この変換に要するツールは、DPL³⁾の処理系によって容易に作成可能である。

5. 適用例

本稿の文書型として論文型を定め、その型の論理構造付き文書データから $\text{\LaTeX}$ への変換プログラムを作成し、本稿の作成に適用した。本処理系の処理例として、常用文書データの例とマーク付常用文書データの例を図4に示す。

6. まとめ

暗に構造化される文書データを用いて情報の蓄積を行なうための、常用文書記述形式とその処理系について述べた。これにより、文書作成時には最終形態を意識することなく情報を文書として作成可能となり、後の活用時には蓄積した情報を文書として利用可能となる。

参考文献

- 1) 上林憲行: “文書エディタの現状と将来展望”, 情報処理, Vol.31, No.11, pp. 1535-1542 (1990).
- 2) ISO 8879: Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML) (1986).
- 3) 長島, 山川: “文書処理統合環境DLETにおける文書データ処理言語”, 情報処理学会全国大会第42回, 6Q-4 (1991.3).