差分DTD生成型の構造化文書差分抽出方式

3S－6

青山 ゆき　高橋 亨　東野 純一†
(株)日立製作所 情報・通信開発本部‡

1. はじめに

構造化文書記述言語SGMLが、国際標準として普及し始めた[1][2]。このSGMLは、文書の共同執筆や、再利用に適している。このため、マニュアル等の作成に用いられるが、編集を行い版を管理する際、文書データベースの容量の巨大化を抑え、また、編集効率を上げるために、構造情報を区別して変更箇所を抽出し表示したいという要求がある。しかし、従来の方式では、構造を区別した差分の抽出や変更箇所の表示は困難であった。

本稿では、この問題に対する解決方法として、変更箇所を論理構造を考慮して抽出し、DTD(Document Type Definition)に基づき構造化した差分情報を出力する差分DTD生成型の構造化文書差分抽出方式を提案する。

2. 構造化文書差分抽出方式

従来の差分抽出方式をSGML文書に適用すると、SGML文書中に埋め込まれた論理的な構造を表すタグも、他の文字列と同様に比較され、適切な結果が得られない。例えば、図1のように(1)論理構造の意味が異なるもの同士の対応付け、(2)構造間に跨った対応付けが生じてしまう。

この問題を回避するために、文書の論理構造に基づいた比較を行う、構造化文書差分抽出方式を開発した。本方式では、構造化文書から文書の論理構造を表す文書木を作成し、文書木のノードを単位に差分を抽出することにより、構造比較を実現する。また、論理構造上意味のない比較をしないために、予め比較する際のルールを定義しておき、文書木の作成時に、このルールを参照し文書木を変形することにより、論理構造を考慮した比較を行う。以下、本方式の処理手順を示す。

[Step1] 比較対象であるSGML文書のDTDに対応した比較基準テーブルを作成する。この比較基準テーブルとは、差分を抽出する際のルールを記述するもので、今回は、次の四つの比較基準を設けた。

(1) 恒等タグ：タグ同士が一致したときだけその中身を比較するタグ
(2) 無視タグ：文書全体を比較する際、タグの中身の差異を無視するタグ
(3) 同等タグ：論理的な意味が同じタグの組(比較の際、同じタグとみなす)
(4) 比較禁止タグ：中身を比較しないタグの組

図1のSGML文書に対して、例えば“＜著者名＞および＜所属＞は恒等タグ”、“＜章番号＞は無視タグ”、“＜章＞と＜初章＞は同等タグ”というように、テーブルを定義しておく。

[Step2] 比較基準テーブルを参照しながら、構造化文書から文書木を作成する。この時、文書木の各ノードへの要素の割当ては、次のルールを用いて行う。

(ルール1)：タグは一つのノードに割り当てる。
(ルール2)：開始タグと終了タグの間の文字列は開始タグの子ノードに割り当てる。
(ルール3)：終了タグは、開始タグの子ノードに割り当てる。
(ルール4)：恒等タグで挟まれた文字列は、開始タグと終了タグを含めて一つのノードに割り当てる。
(ルール5)：無視タグおよび無視タグで挟まれた文字列は、ノードに割り当てない。
(ルール6)：同等タグは、同じタグ名に変換してノードに割り当てる。

このルールを適用すると、上述した比較基準を参照することによって、図1の文書a, bから、図2の文書木a, bが生成できる。

[Step3] 文書木のノードを単位に差分抽出を行う。ノードを単位に比較を行うため、恒等タグである＜著者名＞および＜所属＞は、タグと中身の文字列が両者とも一致しない限り、対応付けられることはない。

[Step4] 一致しなかったノードのみ、今度は文字列間の差分抽出を行う。ただし、恒等タグのノードはノードの先頭文字であるタグが一致した場合のみ、文字単位の比較を行う。無視タグはこの時点で比較を行う。

図1の文書aと文書bの差分抽出を行った結果を図3に示す。図1に示したような論理構造の異なる“日立”同士の対応付けや、文書の構造間にまたがった文字列の対応付けは行われていないことが分かる。

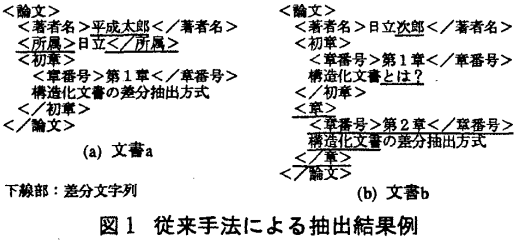


図1 従来手法による抽出結果例

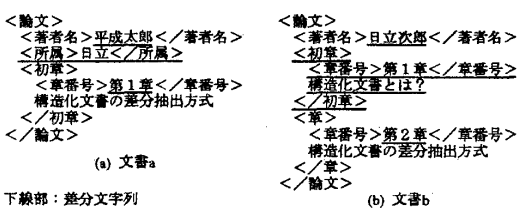


図3 構造化文書差分抽出方式による抽出結果例

An Improved Method for Extracting Differences between Structured Documents
†Yuki AOYAMA, Tooru TAKAHASHI and Jun'ichi HIGASHINO
‡Information Systems R & D Division, Hitachi, Ltd.

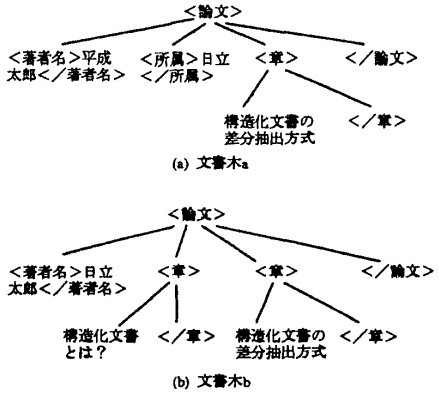


図2 構造化文書木の例

3. 差分DTD生成型のSGML差分抽出方式

さらに、差分抽出結果の構造情報を表現するため、SGML形式で結果を出力する差分DTD生成型のSGML差分抽出方式を開発した。

3.1 SGML形式での差分抽出結果出力方式

編集前後の構造化文書間の変更箇所を抽出した差分データとしては、構造を持たない通常の文書を比較した場合と異なり、次のような特徴がある。（1）構造自体の変更と構造中の文字列の変更がある。（2）差分情報にも論理的な構造がある。

この特徴を表現するため、本方式では、差分結果(図3)を図4の例のように構造化して出力する。構造自体の変更は、その構造を示すタグにdiffflagという属性を持たせて構造の変更を表現し、構造中の文字列の変更は、差分を表すタグでその文字列を挟んで表現し、両者を区別する。

```
<論文>
  <著者名><変更前>平成太郎</変更前>
          <変更後>日立次郎</変更後></著者名>
  <所属 diffflag=削除><削除>日立</削除></所属>
  <初章 diffflag=挿入>
    <章番号 diffflag=挿入><挿入>第1章</挿入></章番号>
    <挿入>構造化文書とは？</挿入>
  </初章>
  <章>
    <章番号><変更前>第1章</変更前>
            <変更後>第2章</変更後></章番号>
    構造化文書の差分抽出方式
  </章>
</論文>
```

図4 差分抽出のSGML形式出力例

3.2 差分DTDの生成方式

このSGML形式で出力した差分データを利用するには、差分データ用のDTDが必要である。

図4の差分データを表現する、差分DTDを考える。図3の構造化文書は、例えば図5(a)のようなDTDを持つ。この比較対象文書のDTDから、差分DTDを生成する。差分DTDは、元のDTDに対して、(1)文字データが差分を表すタグを含むことを可能にし、(2)最上位の構造以外はdiffflagという属性を持たせ、(3)変更のなかった構造は差分データに含めないことが可能となるよう出現指標子を変更することで生成される。

図5(a)のDTDをこの様に変更すると、図5(b)の差分DTDが生成される。生成されたDTDは、図4の差分データを表すDTDとなっている。

```
<!ELEMENT 論文    − − (著者名, 所属?, 初章, 章*)>
<!ELEMENT 著者名  − − (#PCDATA)>
<!ELEMENT 所属    − − (#PCDATA)>
<!ELEMENT 初章    − − (章番号, #PCDATA)>
<!ELEMENT 章      − − (章番号, #PCDATA)>
<!ELEMENT 章番号  − − (#PCDATA)>
```

(a) DTD

⇓ 生成

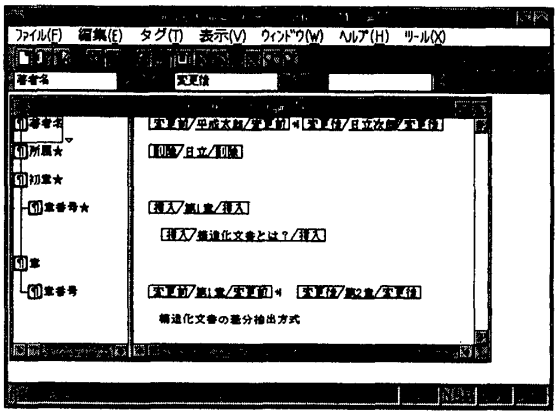
```
<!ENTITY % DiffElement  "挿入 | 削除 | 変更前 | 変更後">
<!ELEMENT 挿入    − − (#PCDATA)>
<!ELEMENT 削除    − − (#PCDATA)>
<!ELEMENT 変更前  − − (#PCDATA)>
<!ELEMENT 変更後  − − (#PCDATA)>

<!ELEMENT 論文    − − (著者名?, 所属?, 初章?, 章*)>
<!ELEMENT 著者名  − − (#PCDATA | %DiffElement;)* >
<!ELEMENT 所属    − − (#PCDATA | %DiffElement;)* >
<!ELEMENT 初章    − − (章番号, (#PCDATA | %DiffElement;)*)>
<!ELEMENT 章      − − (章番号, (#PCDATA | %DiffElement;)* >
<!ELEMENT 章番号  − − (#PCDATA | %DiffElement;)* >

<!ATTLIST 著者名 diffflag
          (NULL | 挿入 | 削除 | 変更前 | 変更後)  NULL>
<!ATTLIST 所属   diffflag
          (NULL | 挿入 | 削除 | 変更前 | 変更後)  NULL>
<!ATTLIST 初章   diffflag
          (NULL | 挿入 | 削除 | 変更前 | 変更後)  NULL>
<!ATTLIST 章     diffflag
          (NULL | 挿入 | 削除 | 変更前 | 変更後)  NULL>
<!ATTLIST 章番号 diffflag
          (NULL | 挿入 | 削除 | 変更前 | 変更後)  NULL>
```

(b) 差分DTD

図5 差分DTDの生成例

4. 応用例

上記、SGML差分抽出方式により出力された差分データは、差分DTDに基づき、下図の様に、SGMLエディタを用いて構造化表示することができる。



また、差分データを、SGMLコンバータを使って任意の改訂履歴の形式に変換することができる。これにより、マニュアル等の変更書を自動作成することが容易になる。

5. おわりに

本稿では、SGML文書間の変更箇所を論理構造を考慮して抽出し、DTDに基づき構造化した差分情報を出力する差分DTD生成型の構造化文書差分抽出方式を提案した。本方式により、SGML文書の構造を考慮した版管理機能や変更箇所表示機能を容易に実現することが可能となった。

参考文献

[1] 吉岡, “SGMLのススメ”, オーム社, p.160(1993)
[2] E.Herwijnen, “Practical SGML”, Kluwer Academic Publishers, p.307(1992)