情　報　学　基　礎 33－5
（1994．5．18）

SGML文書構造の文法を用いた変換処理

酒井　乃里子
東京大学工学部

高須　淳宏　安達　淳
学術情報センター研究開発部

SGMLにより記述された科学技術論文は一般に多様な論理構造を持つので、このような文書群を対象とする全文データベースでは、多様性への対処が課題となる。本研究では、各文書を固有のDTDによる多様な形式のまま蓄積し、ユーザには統一されたビューを提示する全文データベースの構成を検討している。本稿では核心として、固有の論理構造を標準的なビューの形式に変換する、文法を用いた手法を提案する。併せて、作成したプロトタイプも紹介する。この手法はまず、データベース側で統一したビューとなる項目を設定し、DTDごとに各項目に対応する論理要素を定めておく。ユーザの処理要求に基づいて、前述の対応から処理に必要な論理要素が決定し、これらを字句解析により文書中で特定し、構文解析により受理し、結果とする。

A Grammatical Method for Transformation of Document Structures in SGML

Noriko SAKAI
Graduate school of Engineering, University of Tokyo

Atsuhiro TAKASU　Jun ADACHI
Research and Development Department, NACSIS
(National Center for Science Information Systems)

Scientific documents described in SGML (Standard Generalized Markup Language) tend to have diversified structures according to their own DTDs (Document Type Definition). It causes difficulty on implementing a large-scale full-text databases of those documents. We propose a processing method by which various structures of DTDs are transformed into a canonical structure. In the proposed method, a set of data items as a canonical view for users is fixed, and elements of each DTD are assigned to each item. When a user requires some items, elements of each DTD are identified according to a predefined assignment. Thereafter, through lexical and syntactical analysis, elements are specified and strings are extracted form each document.

1 はじめに

近年の計算機の性能向上により、文書を電子媒体の形で作成し、マークアップした上、計算機上で扱うことが普及してきた。これに伴い、さらに執筆から編集、印刷など一連の処理を統一して行えるように、マークアップの規格が求められるようになり、Standard Generalized Markup Language (SGML, ISO 8679) が定められた [2]。

SGML では文書の論理構造・論理要素の定め方（定義の記述方法）のみを規定しており、具体的な構造は各著者に委ねられている。つまり、各自が自分の要求に合う構造を設定し、それを DTD (Document Type Definition) に記述することになる。従って、一般に構造は多様になる可能性がある。

一方現在データベースとしては、書誌情報や要旨などを扱った「二次情報データベース」が主流となっているが、電子媒体の形態の文書データが増えてくるにつれて、文書全体を対象とする「全文データベース」が望まれるようになってきた。

SGML により記述された論文を対象とした全文データベースの実現にあたっては、文書の論理構造の多様性により困難が生じる。著者にとってこの多様性は、著述の自由度の点で不可欠なものであるが、一方ユーザとしては、多数の文書を扱う際に、各々の論理構造を考慮することは不可能であることから、統一的な取り扱い法が必要となる。

このような現状を踏まえて、本研究では、

- SGML により記述された多様な論理構造を持つ文書群を用いて、
- ユーザには統一されたビューを提供する

ようなデータベースを目指し、核心となる、文書を固有の論理構造から標準的なビューへの変換処理の手法を提案する。

本稿では、まず前提とするデータについて触れ、これらを前述のように変換する困難さを具体的に述べる。続いて本研究の手法を、処理の流れに沿って説明する。この中で、書誌情報を対象として作成したプロトタイプも適宜紹介する。また最後に、この手法の課題も挙げる。

2 変換の困難さと処理の概要

2.1 文書構造の多様性

本研究では SGML により記述された科学技術論文を対象としてデータベースを実現することを目的としている。これらの文書については、DTD は学会などに相当する単位で設定されると想定できるが、説明の便宜上、以下では DTD を扱う者と文書自体の著者をまとめて著者と呼ぶ。例えば、「学術情報センター紀要第 5 号」[1] 用に設定された DTD と、そのインスタンスは図 1、2に示す通りである。

ここで、DTD に記述される文書の論理構造は、木構造として表されるものと本質的に等価であり、木構造の方が人間が論理構造として持つ概念に近く、直観的に把握しやすいため、本稿では、文書の論理構造や DTD での記述内容をしばしば木で表す[1]。

```
論文 --- 論文情報 --- 資料区分
      |          +- 英題
      |          +- 英副題
      |          +- 著者情報 --- Fネーム
      |          |          +- Lネーム
      |          |          +- 英所属
      |          +- 英キーワード
      +- 要旨    --- 段落
      +- 本文    --- 章 --- タイトル --- 主題
      |          |    |          +- 副題
      |          |    +- 段落
```

図 1: DTD の例（部分）

```
<論文 ID=''RO3'' LANG=''JA''>
<論文情報>
<資料区分>研究論文</資料区分>
<英題>Co-authoring of scholarly papers</英題>
<英副題>A comparative study on Japanese ...</英副題>
<著者情報 ROLE=''AUTHOR''>
<Fネーム>Hisao</Fネーム>
<Lネーム>YAMADA</Lネーム>
<英所属>National Center for Science ...</英所属>
</著者情報>
<英キーワード>Co-authored papers</英キーワード>
</論文情報>
<要旨>
<段落>わが国の学術論文では、外国論文に比べて、一般に共著者の数が多いといわれることがある。この背景には、わが国における
...

の発達が共同研究の実施を一層容易にする反面、成果の発表においては個別化・個性化をもたらす可能性を考える。</段落>
</要旨>
<本文>
<章 ID=''RO3CO1''>
<タイトル><主題>はじめに</主題></タイトル>
<段落>わが国の学術論文では、共著者数が、外国の論文に比べて全般的に多い傾向にあるということが、経験的直感としてかなり
...
づいてわが国における研究のあり方について議論を進めることは関連施策の適確な立案に資することにもなるであろう。</段落>
```

図 2: SGML インスタンスの例（部分）

[1] ただし、DTD の方は、回数や順序について規定を記述でき、この点で木構造より記述能力は高い。

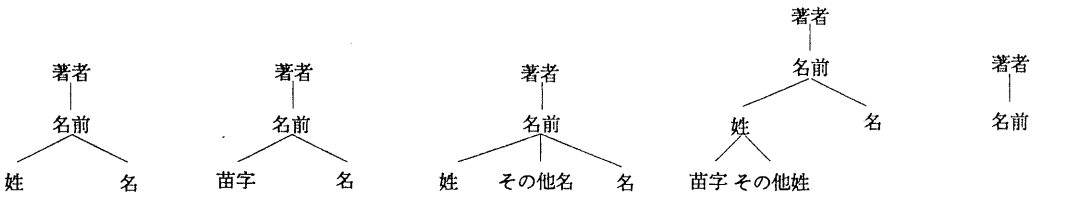

図 3: 文書の様々な構造

SGML では、文書の論理構造は、著者が定めることになっているので、多様な論理構造が存在する可能性がある。例えば著者名について、図 3のような様々な例が考えられる。実際の文字列データは、構造の末端である葉にポインタなどを介して割り当てられるので、論理構造の調和を図る場合は、末端の構成の差異について対応を考慮しなくてはならない。

構造 1 と 2 とは、名称のみの違いで、構造そのものは同等である。構造 3 は、それらに比べて、末端の階層は同じ深さであるものの、より細かく論理要素が区分されている。構造 4 では、より深い階層を設けることにより、構造 1 や 2 より細かい論理要素区分がなされている。構造 5 では逆に、それらに比べて浅い階層までしかなく、より粗い要素区分となっている。

これらに対して、マッチングなどを行うために著者の姓を抽出する場合には、各構造でそれぞれ、「姓」、「苗字」、「姓」、「苗字」と「その他姓」の組、「名前」の各要素を対象とすればよい。

このように、論理構造の差異は、扱う文書数が少なければ一件一件検討することによりどのような変換や対応づけも可能である。また、DTD は学会などの単位で設けられるという前提によれば、一つの論理構造を扱えるだけでそれなりのデータ数は確保できる。しかし、データベースとして最大限の意義を引き出すためには、複数の論理構造を扱う必要があり、個別に考慮する方法では間に合わない。

2.2 システムの概要

このような困難に対処するために、本研究では、図 4のような変換処理を提案する。

(1) データベース側では、ユーザに提示する統一的なビューとして、表示・検索用の項目を設定しておく。また (2) 著者は、各自の DTD のどの要素が各々の項目に対応するかを定める。

(3) ユーザが、データベース側で定めたビューの提示する項目に基づいて、検索処理での項目指定を行うと、(4) 指定された項目と先ほど定めた対応とによって、処理に必要な各 DTD での論理要素が決定する。

続いて、(5) 必要な要素を字句解析により各文書中で特定し、要素に含まれる情報を抽出する。その結果をもとに構文解析を行って、文法的に受理した上で、整形して、(6) 結果として返す。

データとしては、固有の DTD により記述されたインスタンスを、元の形式のまま保持する。予め変換した形でなく、オリジナルな形式で蓄積することにより、その DTD に付随する処理系を用いて、著者の意図した本来の全文情報を再現できる。また、そのためと情報データベース用とに別途データを用意しなくてよいという利点が得られる。

なお、ここで処理されるインスタンスは、既に「正しく完全な」状態にあるものとして、この変換処理には従来の SGML パーザに見られた、DTD に記述された論理構造の検討や、インスタンスに対するマークアップ誤りの検出あるいは訂正の過程は含まれないものとする。

3 準備

以下に処理の各過程を詳細に説明する。

ユーザとしては、多様な文書を多数扱うためには、統一的な操作環境が望ましい。そこで、データベース側では、様々な文書構造の標準と位置付けることのできる項目群を、ユーザへの検索処理用のビューとして設定する。この項目群は、ユーザが常識的に論理構造として意識しているものに近付けることが

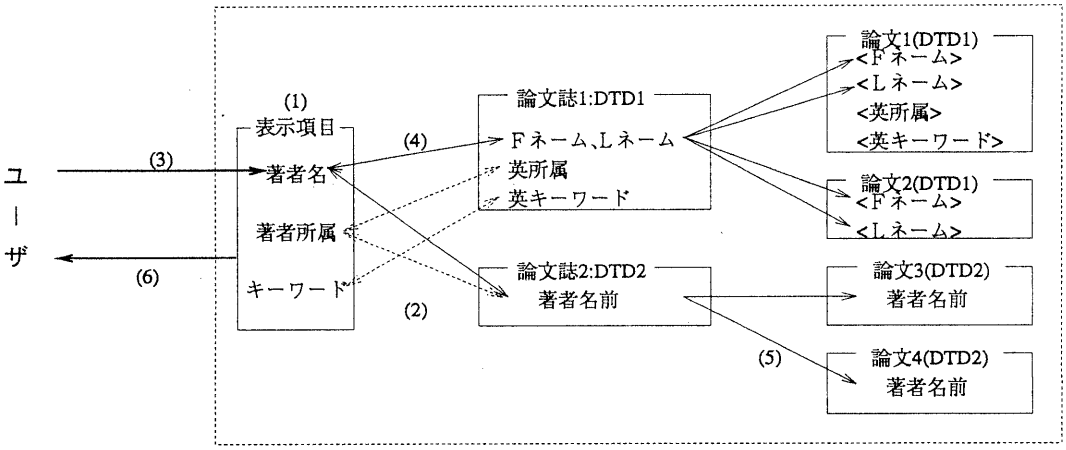

図 4: システムの概要

表 1: 設定した表示項目とその表示形式

表示項目	表示形式
著者名	姓, 名前の頭文字.
著者所属	(特になし)
タイトル	main title(sub title)
収録誌名	(特になし)
発行年月	月名略称., 西暦年
巻号	vol. 巻番号, no. 号番号.
発行者	(特になし)
ページ	開始ページ-終了ページ, 総ページ.
ISSN	(特になし)
キーワード	(特になし)
要旨	(特になし)

できれば、操作性のの向上が期待できる。

また一般的手法として、単純なビューとしてのみならず、普遍的な文書を表現するために十分な論理構造を持たせれば、「論理構造から論理構造への」変換処理フィルタの役割を果たすことも可能である。

今回のプロトタイプでは、現存の二次情報データベースでの項目設定に倣って、表1の 11 項目を設けた。ここで、データとして表示する際の形式も定めることができる。これも併せて表1の通りである。

本研究では、DTD は学会程度の規模を単位として設けられることを前提としているので、この単位につき一度ずつ、DTD でどの要素が各項目に対応するのか定める。

この対応づけでは、複数の要素を一つの項目に割り当てることも可能とする。これにより、論理要素の方が項目よりも細かく区分されている場合に対処できる。例えば、図 3では、項目「著者名」について、構造 1 では「姓」と「名」の要素を組み合わせることになる。この組み合わせ処理は、最終的なデータの表示形式への整形と同時に行われる。

逆に、各文書の方が標準的な項目よりも粗い場合、

- 厳密な方針を採り、該当する項目はないとする
- ユーザが求めるものと合致する「かもしれないもの」があるとする

という二通りの方針が考えられる。例えば図 3の構造 5 で、「著者の姓」を要求されている場合、前者では「姓は特定できない」とし、後者では要素「名前」を示して、「合致するかもしれないものはこの中に含まれる（しかし、違う意味を持つ文字列であるかもしれない）」との結果を示す。この方針の選択は、データベースシステム設計者に委ねられる。

また、同一名称の要素が、異る位置づけで用いられることも考えられる。例えば「年」「月」「日」などは、「受理年月日」「発行年月日」など様々な要素の下位要素となることが予想される。このような場合に、想定する要素を正しく指定するためには、論

表 2: 表示項目と対応する要素

表示項目	対応する論理要素
著者名	/論文/論文情報/著者情報/Fネーム /論文/論文情報/著者情報/Lネーム
著者所属	/論文/論文情報/著者情報/英所属
タイトル	/論文/論文情報/英題 /論文/論文情報/英副題
収録誌名	/論文/論文情報/英論文誌名
発行年月	/論文/論文情報/発行年月日/年 /論文/論文情報/発行年月日/月
巻号	/論文/論文情報/巻 /論文/論文情報/号
発行者	/論文/論文情報/発行者
ページ	/論文/論文情報/開始頁 /論文/論文情報/終了頁 /論文/論文情報/総頁
ISSN	/論文/論文情報/ISSN
キ-ワ-ド	/論文/論文情報/英キーワード
要旨	/論文/英要旨/段落

理構造の根からの「絶対パス表現」で表す必要がある。反対に、それらの「年」をすべて含む項目を設けた場合は、論理構造の任意の点からの「相対パス表現」を、あるいはそれらのパス表現の一部を省略する「ワイルドカード」を含む表現も必要である[2]。

これらの方針や表現方法により、互いにどのように異る構造を持つ文書間の対応づけや、どのような位置関係にある要素の組み合わせにも柔軟に対処できる。今回のプロトタイプでは、表 2のような対応を定めた。

4 字句解析と構文解析

ユーザは、データベースの提示する標準的な項目に基づいて検索などを行い、その際「……の項目（が……であるもの）」という条件を付加する。この条件から、前節の対応表によって、処理に必要な各 DTD 中の要素名が決定される。この要素を、各文書中で特定し、情報を抽出するのが字句解析ルーチンであり、字句解析の結果を文法的に判断し、受理するのが構文解析ルーチンである。この二者間の受渡しに用いるデータをトークンと呼ぶ。ここでは説明の便宜上、要素名をそのままトークン名として用いる。

字句解析ルーチンは、文書をはじめから終りへシーケンシャルに走査し、必要な要素が出現したところで、出現した要素のトークンを返し、同時に要素に含まれる文字列情報をトークンに付随する値として返す。そのトークンに関する構文解析の処理が終了し次第、さらにその先を引き続き解析する。

構文解析では、字句解析の結果を受けて、返されたトークンを文法的に受理する。その役割は、具体的には

- 項目と要素の対応づけの際に許容した「複数の要素を組み合わせて構成される」項目を認識し、
- 一つの項目を構成する要素（または要素の組）の複数回にわたる出現を認識する

ことである。さらに、データの表示形式としての指定や、複数要素の組み合わせに従って、整形を行ったのち、結果をユーザに提示する。

この構文解析ルーチンには、作成の簡易化のために、UNIX 上での yacc を利用した。以下に yacc 仕様書の作成方針を述べる。

まず、組み合わせに関しては、例えば要素「姓」と「名」の組で項目「著者名」を認識したい場合には、

```
名前 : 姓　名
     ;
```

となる。繰り返しは、yacc の一般的技法である左再帰構造を利用して、例えば項目「著者名」を構成する要素「名前」が繰り返し表れる場合、

```
著者名 : 名前
       | 著者名　名前
       ;
```

となる。実際には、組み合わせられたものが繰り返し表れることが多いので、上記の手法を応用する。繰り返し単位となる要素の組み合わせに相当する中間トークンを設ける。例えば、項目「著者名」が要素「姓」と「名」の組の繰り返しで構成される場合、「姓」と「名」の組を表す中間トークンとして「名前」を設けて、

[2]ただし相対パス表現やワイルドカードを含む表現では、論理構造の木上で、根から葉への方向のみを許し、逆方向が存在し得ることによる範囲の不用意な拡張を許さないものとする

```
著者名　：名前
　　| 著者名　名前
　　;
名前　　：姓　名
　　;
```

とすればよい。

また、データの整形はyaccのアクションとして記述できる。今回はこれを、C言語の関数で記述した。「複数のデータを持つ項目（文書内で同一項目が複数回出現した）は、カンマを挟んで並べる」など頻繁に用いられる一般的な処理は、汎用のライブラリとして整え、仕様書の作成負荷の軽減を図っている。

一方、構文解析ルーチンの動作では、全項目を一度の解析で受理するためには、論理構造の木の任意の位置の要素群が組み合わさって一つの項目を構成する場合などを考慮して、結局文書全体の論理構造を念頭において仕様書を作成しなくてはならない。

このようにして仕様書が不必要に繁雑になるのを避けるため、字句・構文解析の動作は、一度の解析で一つの項目の構成処理を行うものとし、求める項目が複数の時は、それに相当する回数、解析を繰り返すようにしている。

5　使用例

本研究で作成したプロトタイプの使用例を紹介する。2.1項に述べたように、本研究では「学術情報センター紀要」第5号用に設定されたDTDと、同誌原稿として作成された電子媒体による文書に、構造の独自さを損なわない範囲で手を加えて、サンプルデータとして用いている。そのDTDによって表される論理構造やSGMLインスタンスは図1、2（2.1項）の通りである。

またこのDTDに基づいて、データベース側では表1のように標準的なビューとなる項目を定めてある。さらに、上述のDTDについて、表2のような対応を定めた（表はいずれも3節）。

このようなDTDによるインスタンスの一つ（計算機上では、kiyou03.sgmlというファイル名になっているとする。）から、項目の「著者名」「タイトル」「キーワード」にあたる情報を得る場合、図5のような動作になる。

```
% parse_inst 著者名 タイトル キーワード kiyou03.sgml

著者名　　: NEGISHI, M., YAMADA, H.
タイトル : Co-authoring of scholarly papers
(A comparative study on Japanese and Western papers.)
キーワード : Co-authored papers, Abstracting
databases, Bibliometrics, Originality

　　「インスタンス "kiyou03.sgml" から
　　著者名、タイトル、キーワードを抽出する」
```

図5: プロトタイプの実行結果例

6　まとめと今後の課題

本稿では、多様な論理構造を持つ文書を対象として、ユーザに統一的なビューを提供する全文データベースのための核心となる機構として、文法を用いて、必要な要素を各文書中で特定し、情報を抽出する手法を提案した。

本稿では、この手法を書誌情報に対して適用した場合の有効性を示したが、今後は文書の本体に用いることが大きな課題の一つである。これを実現するためには、ユーザが文書を計算機上で利用する際に、どのような形で論理構造を活用するのか、またその潜在的な要求があるのか、について検討しなくてはならない。

さらに、現状ではyaccを用いて本手法の実現性を示したが、これを、字句・構文解析を一貫して行うルーチンを設計するか、あるいはyacc仕様書作成処理の負荷を軽減し、さらにはDTDから自動的に処理して作成する方法も含めた方向での改良などを今後の課題として検討したい。

参考文献

[1] 学術情報センター紀要編集委員会編: “学術情報センター紀要,” 学術情報センター (1992).

[2] van Herwijnen,E. 著, SGML懇談会実用化WG監訳: “実践SGML,” 日本規格協会 (1992).

[3] Lesk,M.: “The CORE Electronic Chemistry Library,” Bellcore社内部資料 (1992).

[4] Warmer,J. and van Vlet,H.: “Processing SGML documents,” Electronic Publishing, Vol.4, No.1(Mar.1991).