情 報 学 基 礎 44－4
（1996. 11. 21）

既存文書のレイアウト情報付き構造化とその利用

石田 和生　　神谷 俊之　　市山俊治

{ishidakz,kamiya,ichiyama}@obp.cl.nec.co.jp

NEC　関西 C&C 研究所

〒 540 大阪市中央区城見 1-4-24

近年、既存文書を電子化する手法として文献のページイメージから論理構造を抽出して構造化テキストの形式で蓄積する方法がさかんに研究されている。しかし既存文書を電子化する場合には論理構造だけでは不十分で、レイアウト情報も同時に保存しておく必要がある。

本発表では SGML 形式の構造化文書データにレイアウト情報をタグとして埋め込むレイアウト付き構造化テキスト形式を提案し、既存文書をレイアウト付き構造化テキストに自動変換するシステムについて述べる。さらに、構造化されたデータの利用法のひとつとして、文献のレイアウト情報に基づいた検索を行うシステムについて述べる。

Generating Structured Text with Layout from Printed Document and Layout Retrieval System

Kazuo Ishida, Toshiyuki Kamiya and Shunji Ichiyama

Kansai C&C Research Laboratories, NEC Corporation

1-4-24 Shiromi, Chuo-Ku, Osaka 540, JAPAN

Recently it is developed to extract logical structure from the page images of a printed document and to accumulate it in the form of the structured text as the technique to electrolyze an existence document. Insufficiency to extract only logical structure and necessity to store layout information simultaneously is pointed out.

In this paper, SGML text format with layout that is embedded layout information as SGML's tags into SGML text is proposed. The system which translates a printed document into SGML text with layout automatically is described. The retrieval system using layout information as one of the applications using SGML text with layout is described.

1 はじめに

インターネットの普及や計算機の発達にともない WWW やネットニュースなどを利用した情報の収集や発信が個人レベルでも手軽に行えるようになってきた。しかし、流通させるコンテンツの作成には非常に多くの手間がかかっているのが現状である。特に電子図書館のようなシステムの場合、すでに電子的に存在しているデータだけでなく、紙ベースで存在している既存文書を電子化して入力する必要があり、そのコンテンツ作成にはさらに多くの労力が必要となる。

既存文書を電子化する方法として、文書をスキャナと OCR を用いてテキストデータに変換し、変換されたテキストデータから文書の持つ論理構造 (「タイトル」や「段落」といった情報) を抽出して論理構造情報を含んだ構造化テキストの形式で格納するやり方がある [1,2,3,4]。文書を単なるべたテキストではなく構造化されたテキスト形式で格納するのは、「セクションタイトルに『電話』というキーワードが含まれている文献」のようなきめの細かい検索や、検索結果のしぼりこみ等に論理構造情報が有効であるためである。しかし既存文書を電子化する場合、論理構造だけを抽出したのでは

- 電子化されたデータを表示するときに、「右図参照」のような文書のレイアウトに基づいた記述が正しく再現できない
- 「右上に図のあった論文」のようなレイアウト情報をもとにした検索が行えない

といった問題が生じる。これらの問題はデータを格納する際に、もとの文書が持っていたレイアウト情報が失われてしまっていることが原因である。そこで本研究では既存文書を電子化するときに文書のレイアウト情報も保存出来るような情報構造化方式を提案し、既存文書の電子化システムの試作を行った [5]。本報告では、このうち特に文書の構造化部分について詳しく述べる。さらに本システムで構造化されたデータの利用法のひとつとして、レイアウト情報に基づいて文献検索を行うシステムのプロトタイプを作成したので、これについても説明する。

本構造化手法は、データ量の増加をおさえつつ論理構造情報記述手段の枠組の中でレイアウト情報の記述を行うため、従来の構造化テキストを対象とした検索手法やエディタ等のツールをそのまま用いて、レイアウト情報の付加された構造化テキストを扱うことが出来るという特徴を持っている。

電話が変わる

1．はじめに
グラハム・ベルによって電話が発明されたのは１８７６年だった。日本では明治９年である。以来まだ１００年あまりしかたっていない。

2．おわりに
電話会社など通信業者が考えてつくった通信網の与えてくれるサービスを使うのみでは満足出来なくなるだろう。

(a) テキスト文書

```
<Book>
 <Title> 電話が変わる </Title>
 <Section> <SecTitle> はじめに </SecTitle>
  <Para>
   グラハム・ベルによって電話が
   発明されたのは１８７６年だった。
   日本では明治９年である。
   以来まだ１００年あまりしか
   たっていない。
  </Para>
 </Section>
 <Section> <SecTitle> おわりに </SecTitle>
                    .
                    .
                    .
```

(b) SGML 形式データ

図 1: SGML データの例

2 レイアウト情報付き構造化テキスト

2.1 SGML

構造化テキストの文書フォーマットとして最近広く用いられているものに SGML (Standard Generalized Markup Language) がある。SGML は ISO で規定されたドキュメント

等の格納形式の統一規格であり、現在では構造化テキスト記述用の標準フォーマットになりつつある。このため、本研究でも構造化テキストフォーマットとしてこの SGML を採用することにする。

SGML はフォントの種類や行のセンタリング等、文書のレイアウトにかかわる情報を可能な限り排除し、文書の持つ論理構造だけに注目して文書を記述する。文書の論理構造は、タグと呼ばれる特定の文字列で表される。図 1 (b) は図 1 (a) の文書を SGML 形式のテキストに変換したものの一部を表している。図 1 (b) の中に出てくる <...> と </...> という部分がタグであり、これらのタグで囲まれた部分の論理構造を表している。例えば、<Para> タグはそこから段落が始まり、次の </Para> でその段落が終わっていることを表している。同様にして図 1 (b) の文書を見ていくと、この文書は、タイトルが「電話が変わる」で、本文は「はじめに」と「おわりに」の 2 つのセクションからなっており、各セクションはひとつの段落で出来ていることが分かる。

2.2 レイアウト情報の重要性

SGML のような構造化テキストは論理構造だけを記述しており、文書のレイアウトに関する情報は持っていない。このためデータを画面に表示したり紙に印刷したりするときには、表示アプリケーションが文書データのタグ情報 (論理構造) を利用して文書の割り付け (レイアウト) 処理を行わなければならない。従って、既存文書を構造化テキストに変換し、そのデータを表示アプリケーションが表示した場合、必ずしももとの文書のレイアウトが再現されるとは限らない。ところが、文書の中には「右図参照」等のようにレイアウトが変わってしまうと正しく読み手に意味が通じなくなる記述が存在することがある。また、レイアウト情報は人間の記憶の中でも比較的大きな割合をしめており、以前見た文書を、「右上にグラフのあった論文」のようにレイアウトをもとにして覚えていることも多い [6]。このようなレイアウトに関する記憶をもとに検索を行う場合、レイアウト情報の抜けた構造化テキストデータでは対応することが非常に困難である。

以上のようなことから、既存の文書を構造化テキストに変換する場合に、文書のレイアウト情報の持つ意味がたいへん重要であると言うことが出来る。

2.3 レイアウト情報付き SGML

既存文書を電子化する場合に、レイアウト情報を保存しておく方法には次のようなものが考えられる。

1. 文書をイメージデータとして保存
2. 文書のおおまかなレイアウト (「右上が図で、左下が文章」といったもの) をインデックスとして文書に付加する

しかし、1 の方法だとデータサイズが大きくなり、さらに、検索を行うためには別途インデックスを作成しなければならないという問題がある。2 の方法だと、データサイズをおさえながらレイアウト情報をもとにした検索がある程度実現できるが、特殊なインデックスを付加したデータになるため、従来の検索システムやデータ編集システムでは扱えなくなるという問題点がある。そこで本研究では、文書のレイアウトに関する情報を SGML のタグとして文書中に埋め込むことにした。

SGML のタグにレイアウト情報を埋め込む方法としては、大きく分けて

1. 論理構造を表すタグの属性に埋め込む
2. レイアウト情報を埋め込むタグを新たに設ける

の 2 通りが考えられる。前者は例えば、

```
<Para TOP=0.2 LEFT=0.3
      WIDTH=0.5 HEIGHT=0.2>
  グラハム・ベルによって電話が
  発明されたのは・・・
</Para>
```

のように、段落を表す <Para> タグの属性 (TOP, LEFT など) にその段落のレイアウトに関する情報を埋め込む方法である。一方、後者は

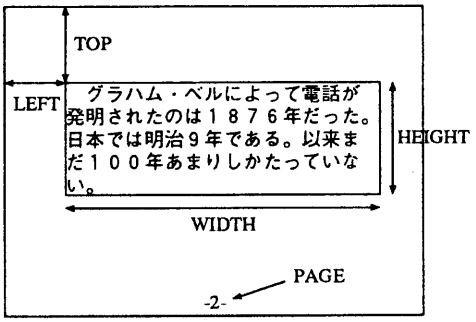

図 2: 保存するレイアウト情報

```
<Para>
<Layout TOP=0.2 LEFT=0.3
        WIDTH=0.5 HEIGHT=0.2>
  グラハム・ベルによって電話が
  発明されたのは・・・
</Layout>
</Para>
```

のように、レイアウト情報を表す <Layout> タグを新たに設けて、その <Layout> タグの属性にレイアウト情報を埋め込むというものである。前者のほうがデータ量が少なく、SGML で文書の論理構造を規定する DTD (Document Type Definition: SGML 文書で、文書の論理構造を定義するもの) の構造が簡単になるという利点があるが、ひとつの段落が複数ページにまたがっているときのように、論理構造から見たときのブロックとレイアウト構造から見たときのブロックが 1 対 1 に対応していないような文書を表現することが困難になるという欠点がある。そこで本研究では後者の <Layout> タグを用いる方法をとることにする。

埋め込むレイアウト情報には、文字の位置情報だけでなく、フォントの種類やサイズ、色など様々なものが考えられるが、今回は、文書全体のおおまかな位置情報を保存しつつデータ量を最小限におさえるために、タイトルや段落といったひとかたまりの文章ごとにその文章を囲む矩形を考え、矩形の左上の座標と横幅、高さ、及び、その矩形が含まれているページ番号を保存しておくことにした (図 2)。

```
<Section>
 <SecTitle>
  <Layout LEFT=0.3 TOP=0.1 WIDTH=0.2
          HEIGHT=0.05 PAGE=2>
    はじめに
  </Layout>
 </SecTitle>
  <Para>
   <Layout LEFT=0.3 TOP=0.2 WIDTH=0.5
           HEIGHT=0.2 PAGE=2>
     グラハム・ベルによって電話が
     発明されたのは１８７６年だった。
     日本では明治９年である。
     以来まだ１００年あまりしか
     たっていない。
   </Layout>
  </Para>
```

図 3: レイアウト情報つき SGML テキストの例

図 3 にレイアウト情報付き SGML テキストの例を示す。図 3 は図 1 の文章からセクション「はじめに」の部分を抜き出したものであるが、含まれている <Layout> タグがレイアウト情報を表している。例えば、セクションタイトル「はじめに」が「2 ページ目の紙の左端から 0.3、上から 0.1 の場所に、横幅 0.2、高さ 0.05 の大きさ」で書かれていた、ことを表している。ここで、場所や大きさを表す数値は、TOP, HEIGHT についてはそのページの縦の長さを 1 とした相対値で、LEFT, WIDTH についてはページの横幅を 1 とした相対値で表している。このようにしたのは、後の章で説明するレイアウト検索を実現する際に、本のページの大きさによる影響を受けないようにするためである。また、これまでの説明では段落等の文字部分について、レイアウト情報の埋め込み方法を述べてきたが、図表等についても同様に、その図表がページ上で占めている領域の位置と大きさを計算し、<Layout> タグに埋め込んでいく。

以上のように SGML のタグとしてレイアウト情報を保存する手法を用いることによって

1. 表示するときに、もとの文書のレイアウトを再現することが出来る

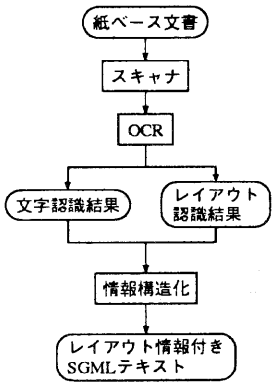

図 4: 処理の流れ

2. レイアウト情報をもとにした検索が容易に行える

3. SGML の規格を拡張する必要がない

4. 既存の SGML 関連ツールがそのまま使用できる

5. <Layout> タグを無視すれば、論理構造のみを含んだSGML テキストとして扱える

といった利点が出てくる。2 番目の「レイアウト情報をもとにした検索」というのは「右上にグラフのあった論文」のような検索のことで、<Layout> タグの属性 LEFT, TOP, WIDTH, HEIGHT の値を利用して実現することが出来る。詳しくは第 4 章で述べる。

3 情報構造化システム

3.1 システム概要

次に、今回試作した情報構造化システムの概要について説明する。このシステムは、紙ベースの既存文書から前節で説明したレイアウト付き SGML テキストを自動生成するシステムである。システムの入力対象となる文書は、電子図書館での利用を考え、文庫本などの一般書とした。

図 4 に本システムの処理手順のおおまかな流れを示す。まず、紙ベースの文書をスキャナで読み込み、読み込んだ画像のレイアウト認識、及び文字認識を行う。この認識の結果として、文書中に含まれる文字とその文字の位置情報が出力される。次に、それらの結果から文書の論理構造を抽出する。論理構造の抽出方式には大きく分けて、文書のレイアウト情報 (フォントの大きさや種類、あるいはセンタリングされているといった情報) をもとに行う方法 (文献 [1, 3] など) と、文章の意味的な情報 (「結論としては…」といった特定の言い回しなど) をもとに行う方法 (文献 [2,4] など) の 2 つがある。今回は入力対象とする文献が一般書であるため特定の言い回しによる構造化は困難であると判断し、前者のレイアウト情報にもとづく抽出方法をとることにした。具体的には、

- ひとつ前の行との行間とひとつ後ろの行との行間がある基準値以上空いている
  ⟶ その行はセクションタイトル

- 行の先頭が、ひとつ後ろの行の先頭よりも下がっていて行末はそろっており、かつ、行間がある基準値より小さい
  ⟶ その行は段落の開始行

のようなルールに基づいて論理構造の抽出を行う。抽出する論理構造であるが、今回は一般書を対象とするため出来るだけ広い範囲の文献をカバーするように、文書の基本的な要素だけで構成した。図 5 に今回我々が定義した文書の階層構造を表す。この図は、ひとつの文献 (BOOK) は前部 (Fm) と本文 (Body) からなっており、Fm はタイトル (Title) と著者 (Author) からなっていることなどを表している。また、図中に含まれる + はその要素の 1 回以上の繰り返し、* はその要素の 0 回以上の繰り返し、? はその要素が「ない」か「ひとつだけある」をそれぞれ意味している。参考までに、この構造に基づいて作成した DTD の一部を図 6 に示す。ただし現時点では、これらの論理構造のうち実際に認識している要素は、文庫本等の文書が持つ基本的な論理構造であるセクション (セクションタイトルを含む) と段落、図、表のみである。

論理構造の抽出が終わったら、その結果に基づいて入力文書を SGML 形式の構造化テキストとして出力する。このとき、文字の位置情報をレイアウト情報として埋め込むが、前章で

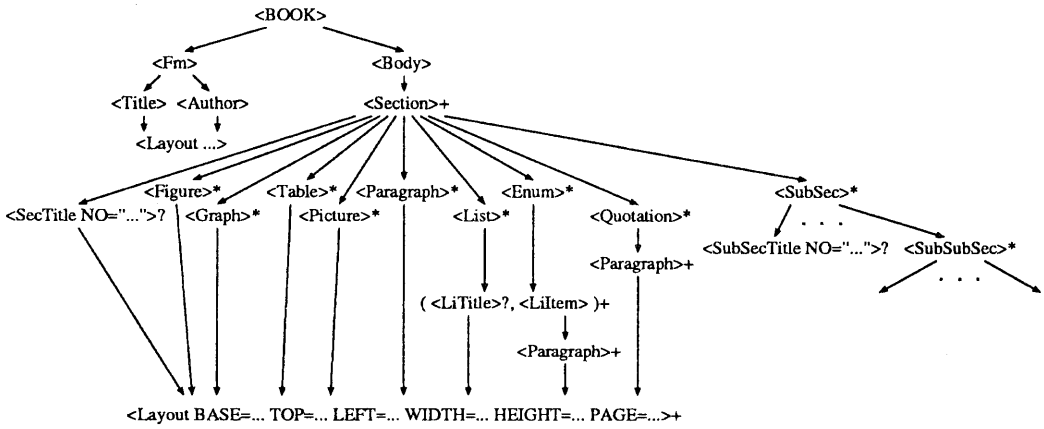

図 5: 文書の階層構造

```
<!DOCTYPE Book [
<!-- Entity -->
<!ENTITY  % Para  "Figure|Graph|Table|Picture|List|Enum|Quotation|Paragraph" >
<!-- Element -->
<!ELEMENT factory o o (Book)>
<!ELEMENT Book - - (Fm?,Body) >
<!ELEMENT Fm   - o (Title? & Author*)>

<!ELEMENT Title - - (Layout*)>
<!ELEMENT Author - - (Layout*)>
<!ELEMENT Body   - o (Section+) >
<!ELEMENT Section - o (SecTitle?,(SubSec|%Para;)*)>
<!ELEMENT SecTitle  - o (Layout+)>
<!ELEMENT Paragraph - o (Layout+)>
<!ELEMENT Layout - o (#PCDATA)>
<!-- Attribute -->
```

（以下、省略）

図 6: 文書の DTD

述べたように 1 文字単位でレイアウト情報を埋め込むのではなく、段落等のブロック単位で埋め込むため、レイアウト情報のまとめあげを行う。具体的には、レイアウト情報を埋め込むブロックに含まれる全ての文字を囲むような最小の大きさの矩形を考え、その矩形の左上の座標と横幅、高さを計算し (この計算は各文字の位置情報などから容易に計算可能である)、<Layout> タグに埋め込む。

3.2 実行例

次に、今回試作したシステムの動作例を示す。図 7 のような入力文書に対し、本システムによって構造化を行った結果の一部を抜き出したものを図 8 に示す。この結果には、ひとつのセクションと、そのセクションタイトル、段落、図が含まれており、それぞれのレイアウト情報が <Layout> タグに埋め込まれていることが分かる。

4 レイアウト検索システム

今回試作した情報構造化システムが出力するレイアウト付き構造化テキストの利用アプリケーションのひとつとして、レイアウト情報に基づく文献検索システムのプロトタイプを作成した。これは、例えば、「右上に図のあった論文」というような情報をもとに検索を行うシステムである。実際の検索は、文書中に埋め込まれた <Layout> タグを利用して行う。

具体的に説明すると、例えば「右上に図のあった論文」の場合、蓄積されている各文書中から <Figure> タグを探索し、その <Figure> タグに含まれている <Layout> タグからその図が存在している場所の位置情報を抜き出す。抜き出した位置情報から図の重心を

$$\left(\mathtt{TOP} + \frac{\mathtt{HEIGHT}}{2}, \mathtt{LEFT} + \frac{\mathtt{WIDTH}}{2}\right) \quad (1)$$

のように計算し、その図が右上にあるかどうかを判断する。今回の検索システムは、図の重心が紙を 3 × 3 に分割した領域のどの位置に存在するかで判断しているので、重心の x 座標が (紙の横幅 ×2/3) よりも大きく、y 座標が (紙の縦の長さ ×1/3) よりも小さければ右上の存在していると判断される。

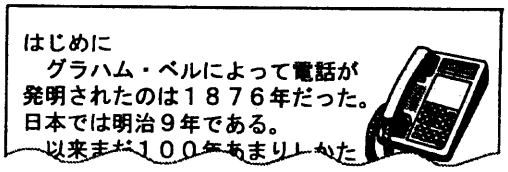

図 7: 紙ベース文書の例

```
<Section>
 <SecTitle>
  <Layout LEFT=0.02 TOP=0.1 WIDTH=0.2
          HEIGHT=0.07 PAGE=2>
    はじめに
  </Layout>
 </SecTitle>
 <Figure>
  <Layout LEFT=0.8 TOP=0.15 WIDTH=0.18
          HEIGHT=0.4 PAGE=2>
  </Layout>
 </Figure>
 <Para>
  <Layout LEFT=0.02 TOP=0.2 WIDTH=0.7
          HEIGHT=0.3 PAGE=2>
    グラハム・ベルによって電話が発明されたのは
    １８７６年だった。日本では明治９年である。
  </Layout>
 </Para>
```

図 8: レイアウト付き SGML 変換結果

検索システムは、ネットワーク上での利用を考え、WWW サーバを介してアクセスするように開発した。これにより、WWW のクライアントが動く任意のマシンからレイアウト検索を行うことが出来る。

システムの条件入力画面を図 9 に、それに対する検索結果を図 10 に示す。条件入力画面 (図 9) の上半分は、通常の書誌事項による検索条件入力部分で、下半分がレイアウト情報の条件入力部分となっている。そこでは、レイアウト検索の対象とする要素 (図、表、写真など) の選択と、その要素の位置情報を指定するようになっている。図 9 では、上中央に表のある文献を条件として検索しているが、その結果 (図 10) を見れば正しく目的の文献が得られていることが分かる。

5 おわりに

本研究では、電子図書館等のコンテンツ情報作成を目的として、SGML テキストにレイアウト情報を埋め込む手法の提案と、紙ベースの既存文書からレイアウト情報付き構造化テキストを生成するシステムについて述べた。また、生成されたレイアウト付き SGML テキストの利用法のひとつとして、レイアウト検索システムの試作を行った。これらのシステムにより、コンテンツ作成手間の軽減と、レイアウト情報に基づく文献検索が可能となった。

今後は、構造化システムの入力対象文書の種類の拡大と、保存するレイアウト情報についての検討などを行うとともに、レイアウト情報付き SGML テキストの検索以外の利用法についても検討を行っていく予定である。

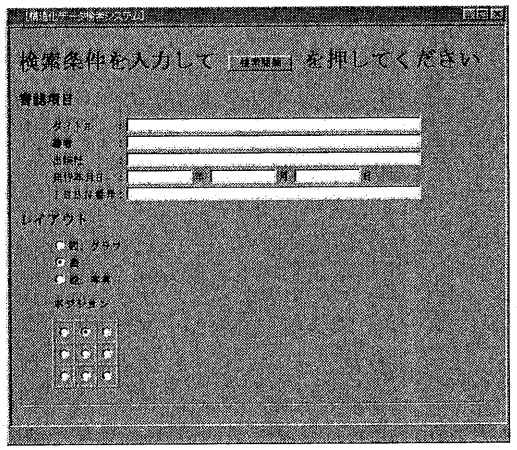

図 9: 条件入力画面

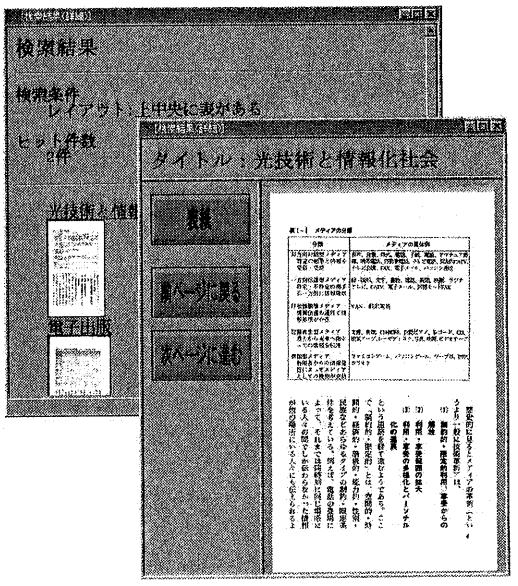

図 10: 検索結果

参考文献

[1] 山田 満: 文書画像の ODA 論理構造化文書への変換方式, 信学論 D-II, Vol. J76-D-II, No. 11, pp. 2274–2284, 1993.

[2] 西村 他: 特定表現の重点解析による科学技術論文構造化手法, 情処研報 93-FI-29, pp. 35–42, 1993.

[3] M. Yamaoka, M. Sato, K. Iwane and O. Iwaki, A Document Understanding System for Converting Printed Documents to SGML Instances, Proc. ISDL '95, pp. 287–288, 1995.

[4] 成田 他: 科学技術論文プレーンテキストへの SGML タグ付けの自動化, 自然言語処理の応用に関するシンポジウム, pp. 49–56, 1995.

[5] 神谷 他: ユニバーサル図書館に向けての図書入力システム「情報ファクトリ」の試作, 「ディジタル図書館」ワークショップ第 8 回, pp. 44–58, 1996.

[6] P. Herrmann, G. Schlageter, Retrieval of Document Images Using Layout Knowledge, Proc. 2nd ICDAR, pp. 537–540, 1993.