Graph Grammar Based Analysis System of Complex Table Form Document

Akira Amano
Graduate School of Informatics
Kyoto University
Yoshida Honmachi, Sakyo-ku, Kyoto, Japan, 606-8501
amano@i.kyoto-u.ac.jp

Naoki Asada
Faculty of Information Science
Hiroshima City University
Hiroshima, Japan
asada@its.hiroshima-cu.ac.jp

Abstract

Structure analysis of table form document is important because printed documents and also electronical documents only provide geometrical layout and lexical information explicitly. To handle these documents automatically, logical structure information is necessary. In this paper, we first propose a general representation of table form document based on XML, which contains both structure and layout information. Next, we present structure analysis system based on graph grammar which represents document structure knowledge. As the relation between adjacent fields in table form documents become two dimensional, two dimensional notation is necessary to denote structural knowledge. Therefore, we adopt two dimensional graph grammar to denote them. By using grammar notation, we can easily modify and keep consistency of it, as the rules are relatively simple. Another advantage of using grammar notation is that, it can be used for generating documents only from logical structure. Experimental results have shown that the system successfully analyzed several kinds of table forms.

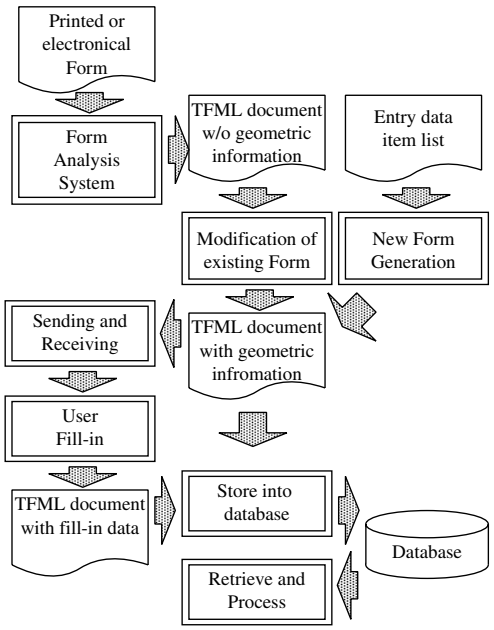


Figure 1. Form Processing and TFML.

1 Introduction

Various kinds of form documents are in circulation around us such as research grant application sheets to which we need to fill in appropriate data to send them others. Among them, one popular type of form document is table form document which are widely used in Japanese public documents. For the table form documents, document analysis system have been researched such as detection of the rules by multiresolutional wavelets[1], extraction of the filled-in writings by block adjacency graph[2]. As there exist number of printed forms, these image processing researches are still very important, however, many forms are becoming electronical in these days.

Considering life cycle of form documents, they are generated and sometimes modified to fit new regulations, and each field is filled-in, then read or analyzed by others and sometimes stored into databases (Fig.1). Although many forms are produced electronically in these days, it is difficult to process them automatically as they do not hold their structural information explicitly. Therefore, it is very important to retrieve structural information from existing forms and define representation of them as well.

Most famous representation of structured document is SGML[3], however it only defines attributes or simple structure of documents and thus it is difficult to represent form like structures. Other industrial representations such as Document Management Alliance (DMA)[4] or Open Document Architecture (ODA)[5] also scopes on document handling and do not offer capability of complex structural representation.

For the structural information, table form structure analysis have been studied which analyzes the connectivities of each fields with production rules[6][7]. However, as table forms has much variety in their structure, it is difficult to

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 IEEE

Table 1. Box types and labels.

category	type	label	description
entry	blank	BLK	empty box to be filled-in
	insertion	INS	box to be inserted or pasted between or on the preprinted letters
description	indication	IND	box that indicates the entry of BLK or INS box
	explanation	EXP	box that includes general explanation

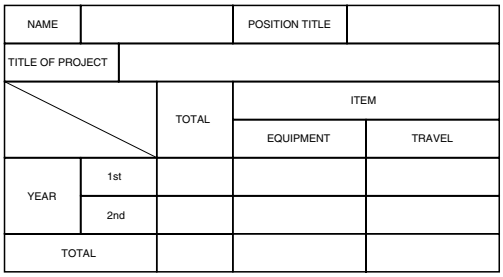


Figure 2. An example of table form document.

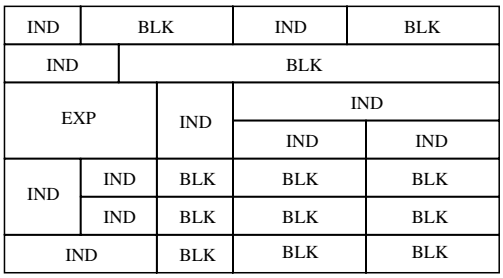


Figure 3. Box classification result of Fig.1.

adopt these system to each document structure as they are constructed on production rules.

To cope with these problems, in this paper, we first propose a representation of table form documents TFML, and then propose a method to retrieve structural information from exsisting documents based on document structure grammar written by graph grammar representation. We used planar graph grammar to analyze the documents, whose production rules can also be used to generate new forms and modify existing documents.

2 Table Form Document and its Representation

2.1 Box Type

The primitive element of the table form document is rectangular fields formed by horizontal and vertical rules as shown in Fig.2. In this paper, the field is called *box*. In the table form documents, each box has the following information.

Identifier is the unique number in the document.

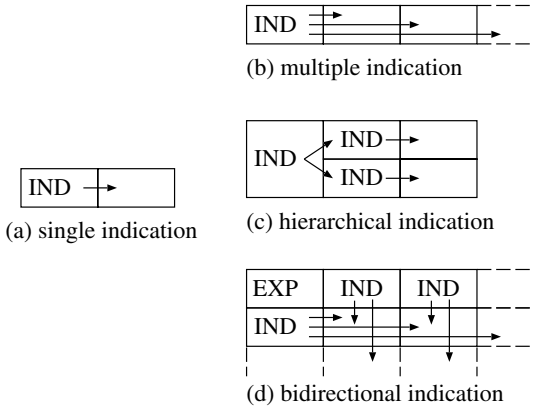


Figure 4. Box indication patterns.

Label is one of the four box types BLK, INS, IND and EXP which represents blank, insertion, indication and explanation box respectively(Tab.1). The blank and insertion boxes which are called entry boxes, are reserved to be filled with the data given by user, whereas the description one represents the indication to entry box or the general explanation. Figure 3 shows the box types of Fig.2.

Position and **Size** represent the coordinates of left-upper corner and width and height, respectively, of the box.

2.2 Box Indication Patterns

The indication box plays an important role in the document; that is, the function of the blank and insertion boxes are determined by the left or upper adjacent indication box, and such a horizontal or vertical relation is always established when both boxes have the same height or width, respectively. This means that the unification of indication box and its associated entry one forms a rectangular block like a box, so we call it a *compound box*.

We define four patterns of compound box according to the relation between indication box and entry one: single, multiple, hierarchical and bidirectional indications as shown in Fig.4. The single indication is the basic pattern of one-to-one box combination horizontally or vertically. Substituting a compound box for a pair of single indication boxes recursively, the multiple indication is represented as a list structure, the hierarchical one is a tree structure, and the

```
<!ELEMENT document ((single | multiple |
          hierarchical | table | INS | EXP)+)>
<!ELEMENT single (IND , (BLK | INS))>
<!ELEMENT multiple (IND , (BLK | INS)+)>
<!ELEMENT hierarchical (IND
          ,(single | multiple | hierarchical)
          ,(single | multiple | hierarchical)+)>
<!ELEMENT table ((EXP | BLK)
          ,col_indication , row_indication , en-
try)>
<!ELEMENT col_indication (indication+)>
<!ELEMENT row_indication (indication+)>
<!ELEMENT indication (IND+)>
<!ELEMENT entry (row+)>
<!ELEMENT row (col+)>
<!ELEMENT col (BLK | INS)>
<!ELEMENT BLK EMPTY>
<!ELEMENT INS (#PCDATA)>
<!ELEMENT IND (#PCDATA)>
<!ELEMENT EXP (#PCDATA)>
<!ATTLIST BLK box_num CDATA #IMPLIED
          str_width CDATA #IMPLIED
          str_height CDATA #IMPLIED
          position CDATA #IMPLIED>
          relation CDATA #IMPLIED>
<!ATTLIST INS box_num CDATA #IMPLIED
          str_width CDATA #IMPLIED
          str_height CDATA #IMPLIED
          position CDATA #IMPLIED
          relation CDATA #IMPLIED
          xml:space (de-
fault|preserve) 'preserve'>
<!ATTLIST IND box_num CDATA #IMPLIED
          position CDATA #IMPLIED>
<!ATTLIST EXP box_num CDATA #IMPLIED
          position CDATA #IMPLIED>
```

Figure 5. DTD of TFML

bidirectional one is a table structure where the indications in row and column directions meet at each entry box.

2.3 TFML: Representation of Table Form

To support handling of table form documents in the process of generation, modification, fill-in and in many other processes, we need general representation of table form documents. Here, we propose XML based representation of table form which can handle geometrical information, indication structures, and also meta-structure of table forms. Meta-structure means logical relation between non-adjacent boxes. For example, the data of box A is summation of box B and box C where these boxes are not adjacent to each other.

DTD of TFML is shown in Fig.5. Sample TFML file is shown in Fig.6. Basic structure of the file reflects indication pattern of the document where geometrical information is embedded as optional information. Meta-information, such as arithmetical relations are also embedded as optional information to each box.

As TFML's representation is general, it can be used as resulting format of document structure analysis. Furthermore, it can be used as input definition of table form generation system, and entry data format of database systems.

```
<?xml version="1.0" encoding="EUC-JP" stan-
dalone="no"?>
<!DOCUTYPE document
          SYSTEM "file:document.dtd">
<document>
  <single>
    <IND>NAME</IND>
    <BLK/>
  <\single>
                        ⋮
  <table>
    <EXP box_num = "7"></EXP>
    <col_indication>
      <indication>
        <IND box_num = "8">TOTAL</IND>
      </indication>
      <indication>
        <IND box_num = "9">ITEM</IND>
        <IND box_num = "10">EQUIPMENT</IND>
      </indication>
      <indication>
        <IND box_num = "9">ITEM</IND>
        <IND box_num = "11">TRAVEL</IND>
      </indication>
    </col_indication>
    <row_indication>
      <indication>
        <IND box_num = "12">YEAR</IND>
        <IND box_num = "13">1st</IND>
      </indication>
      <indication>
        <IND box_num = "12">YEAR</IND>
        <IND box_num = "17">2nd</IND>
      </indication>
      <indication>
        <IND box_num = "21">TOTAL</IND>
      </indication>
    </row_indication>
    <entry>
      <row>
        <col><BLK box_num = "14" rela-
tion="15+16"/></col>
        <col><BLK box_num = "15"/></col>
        <col><BLK box_num = "16"/></col>
      </row>
                        ⋮
```

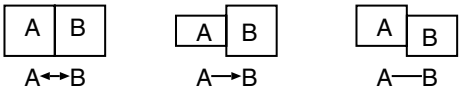
Figure 6. TFML representation of Fig.2(part)

3 Document Structure Analysis Based on Graph Grammar

Since most of the table form documents are decomposed into several combinations of the box indication patterns, we describe the production rules to generate the box elements from the document by a grammar called *document structure grammar*. The advantage of the grammar based approach is that we can manage the structure knowledge and the analysis procedure separately. This scheme enhances the system extensibility because it is easy to extend the document classes by modifying the production rules without changing the syntax analyzer, i.e. parser program. There is previous research which use graph grammar to analyze document layout[8], however, the structure of our target document is much more complex. Therefore, we need to design completely new rules.

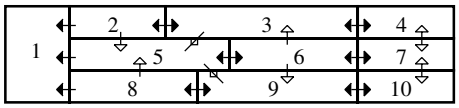
3.1 Graph Representation

As we use graph grammar to represent structural knowledge of table form, we need to define graph representation of them. Althogh we can think of many representation, we used box and adjacency representation. A node is a box which has box types for its label, and a edge is adjacency of two boxes. Each edge has edge label which represents adjacency type illustrated in Fig.7. For example, graph representation of table form document in Fig.8(a) becomes (b).

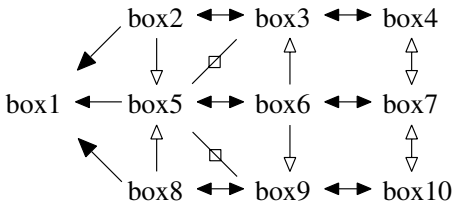


(a)same height (b)included (c)don't connect

Figure 7. Symbols for adjacent box connectivity.



(a)Box adjacencies of table form document.



(b)Graph representation of (a).

Figure 8. Graph representation of table form document.

3.2 Document Structure Grammar

As the distribution of each box has two dimensional information, document structure grammar naturally becomes two dimensional graph grammar.

A garaph grammar is represented with four tuple (Σ, Δ, S, P), where Σ is a set of node label, and Δ is a set of edge label, and S is a starting symbol which is **document** here, and P is a set of production rule. Each rule of P is denoted by $p = (L, R, E)$, where L and R denotes lhs and rhs of the rule, and E represents embedding rule which tells

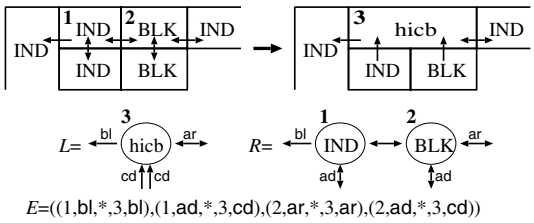


Figure 9. Production rule of graph grammar.

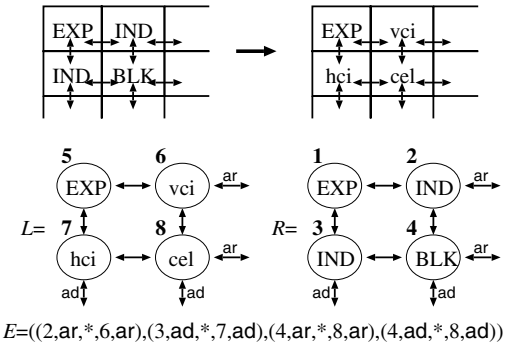


Figure 10. Production rule for bidirectional indication part.

edge label conversion from rhs to lhs. An example of production rule for single indication is shown in Fig.9. For the analysis of single, multiple and hierarchical indications, we can simply extend one dimensional grammar rules which we have proposed previously[9].

On the other hand, bidirectional indication has two dimensional information in its nature, thus we need graph notation of rules to analyze them. One example rule for bidirectional indication is shown in Fig.10. Other rules for analysis of bidirectional indication is shown in Fig.11 which is represented in simplyfied form. Here we used four terminal symbols "BLK", "INS", "IND", "EXP", and six nonterminal symbols <table>, <Ev>, <hc>, <vci>, <hci> and <cel> whose meanings are as follows.

<**table**> denotes whole bidirectional indication structure.

<**Ev**> denotes EXP box which appears on top left corner of the bidirectional indication structure and vci.

<**hc**> denotes all hci and cel boxes in bidirectional indication structure.

<**vci**> denotes indication boxes in bidirectional indication structure that indicates the entry data in vertically adjacent entry boxes.

<**hci**> is same as <vci> except the function is horizontal.

<**cel**> represents the entry boxes in bidirectional indication structures.

With these rules, we can analyze the structure of table form documents, that is, indication relations of every entry box is analyzed.

3.3 Document Structure Analysis

With the graph grammar described above, document structure analysis is performed. As the knowledge of document structure is fully denoted in the grammar, we can use general graph grammar parser for document structure analysis. However, as the grammar is context sensitive, it is difficult to parse the input in realistic computation time.

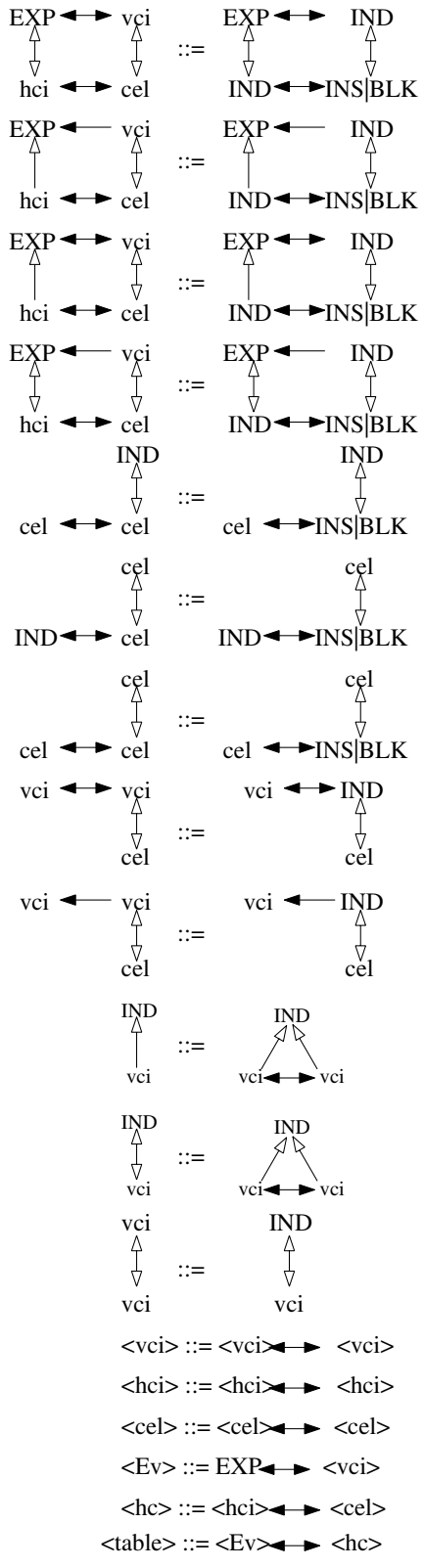


Figure 11. Logical document structure grammar of bidirectional indication.

Therefore, we assign priority to each rule in the grammar. Priority of the rules is as below.

1. Rules for bidirectional indications.
2. Rules for single indications.
3. Rules for multiple indications.
4. Rules for hierarchical indications.

With our document structure graph grammar, we could successfully analyze 31 types of table forms with two document structure grammar.

3.4 Document Generation and Modification

With box indication information, we can generate table form document by using document structure grammar as generation rules. This aspect of the grammar is very important in the sense that the knowledge of the document handling system is completely denoted with both document structure grammar and TFML format.

Typical application will be as follows: first, preprinted form is analyzed by the system using the grammar, second, some modification such as adding or modifying some indication boxes are performed to the TFML document, finally, with the grammar used in reverse order, modified document is generated by table form generation system.

4 Conclusion

In this paper, we proposed table form document representation TFML and structure analysis algorithm based on graph grammar. The effectiveness of the structure analysis is shown for the example table form document.

References

[1] Y.Y Tang, H. Ma, J. Liu, B.F. Li, D. Xi: "Multiresolution Analysis in Extraction of Reference Lines from Documents with Gray Level Background," IEEE PAMI, 19, 8, pp.921-925, 1997.

[2] Bin Yu, Anil K. Jain: "A Generic System for Form Dropout," IEEE PAMI, 18, 11, pp.1127-1134, 1996.

[3] "Information processing – Text and office systems – Standard Generalized Markup Language (SGML)," ISO 8879:1986.

[4] "DMA The Document Management Alliance," http://www.infonuovo.com/dma/.

[5] "Information technology – Open Document Architecture (ODA) and interchange format: Document structures," ISO/IEC 8613-2:1995.

[6] T. Watanabe, Q. Luo, N. Sugie: "Layout Recognition of Multi-Kinds of Table-Form Documents," IEEE PAMI, 17, 4, pp.432-445, 1995.

[7] F. Cesarini, M. Gori, S. Marinai, G. Soda: "INFORMys: A Flexible Invoice-Like Form-Reader System," IEEE PAMI, 20, 7, 730–745, 1998.

[8] Rahgozar, M., Cooperman, R.: A Graph-based Table Recognition System. SPIE Proc. **2660**, pp.192–203, 1996.

[9] Amano, A., Asada, N., Motoyama, T., Sumiyoshi, T., Suzuki, K.: Table Form Document Synthesis by Grammar-Based Structure Analysis. 6th ICDAR, pp.533–537, 2001.