*MVA '92* IAPR Workshop on Machine Vision Applications Dec. 7-9,1992, Tokyo

# A SYNTACTICAL APPROACH TO THE DATABASE CONSTRUCTION METHOD FROM DOCUMENT IMAGES

Atsuhiro Takasu, Shin'ichi Satoh and Eishi Katsura

National Center for Science Information Systems
3-29-1, Otsuka, Bunkyo-ku, Tokyo, 112 Japan

## ABSTRACT

This paper presents a database construction method from journal content images focusing on the logical structure construction. In this paper, we propose a variant of the regular expression to represent a logical structure and layout of the document. This expression is extended in order to analyze tokens located in a plane. We also discuss the systematic definition of the extended regular expression from the schema of nested relation that stores the contents of journals.

## 1 INTRODUCTION

Recently many studies have been done on the information extraction from documents. As a result, many document processing methods are proposed and applied to various kinds of documents such as office documents [1], newspapers [2], business cards [3] and so on. There seems to be two approaches for the document processing. One is based on the numerical-valued features of the document images. In this approach, the logical structure and layout of documents are implicitly defined in the algorithms [4, 5]. The other is the knowledge-directed approach in which several kinds of rules are used with numerical-value based algorithms [6, 7]. We discuss a database construction method from journal content images focusing on the logical structure construction. Knowledge is indispensable to analyze journal contents because the layout of the contents are much different depending on the journals. Therefore, we adopt the knowledge-directed approach.

A database schema represents a logical structure of data. Generally, database models have sufficient power to define the structure of various kinds of data. We use database schema as the logical structure of the objective documents. In our approach, knowledge is represented in a kind of regular expression that is constructed from the database schema by adding the layout information. This approach has the following advantages.

- The structure definition ability of the expression is expected to be as powerful as the adopted database model. Therefore, the expression can be applied to various kinds of documents.
- Since the logical structure has already been defined as the database schema, the users have only to add the layout information to the schema. This reduces the user's work on knowledge definition.
- The generated structure is directly mapped to the database data.

As for the journal content, we use the nested relational database model [8]. In this paper, we first overviews our numerical-value based processes, then shows the extended regular expression and parser for it.

## 2 PRECEDENT PROCESSES

Main roles of document processing systems are to

1. extract segments from document images that correspond to logical units of the documents,
2. classify the segments into classes and
3. construct the logical structure of the documents.

From the database construction point of view, the segment, the class and the logical structure correspond to attribute value, attribute and schema, respectively.

Ordinal relation of the relational model is viewed as a simple table in which the attribute value must be atomic. The schema of relations is defined as a set of attribute names. The nested relational database has recursive schema in which the attribute value can be any of atomic value, tuple and relation. The recursive schema of the nested relation is represented with a tree structure like fig.1 that represents a logical structure of contents of a journal. A content corresponds to a tuple of the nested relation.

For the purpose of the database construction, we need to decompose content images in the level of atomic attribute values and classify them into the class of the attributes. This section overviews our approach to the segmentation and the classification of journal contents.

**Segmentation** In this process, rectangular segments shown in fig.2 are extracted. Then these segments are ordered as much as possible. In the ordinal documents, logical units of the document are naturally ordered in the left-upper to right-bottom direction. However, some journals have complicated layouts and it is difficult to order the segments uniquely. We represent a document

This research was supported in part by a Scientific Research Grant-in-Aid from the Ministry of Education, Science and Culture of Japan under Grant No.04229225.

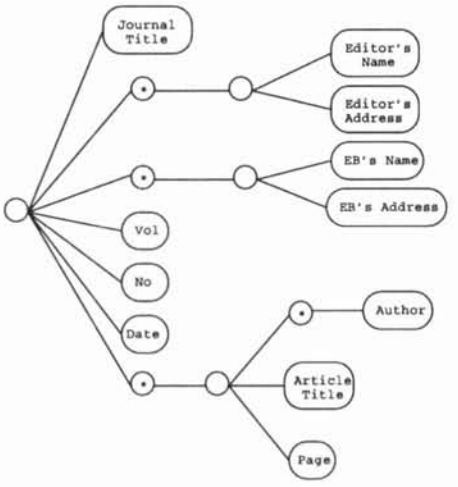


Figure 1: Schema of a Nested Relation

$$\left(\left(\begin{array}{c} Head_1 \\ \begin{pmatrix} Name & Name \\ Address & Address \\ Name & Name \\ Address & Address \\ Name & Name \\ Address & Address \end{pmatrix} \\ Head_2 \\ \begin{pmatrix} Name & Name \\ Address & Address \\ Name & Name \\ Address & Address \\ Name & Name \\ Address & Address \end{pmatrix} \end{array}\right)\left(\begin{array}{c} Jtitle \\ (\ Vnd \quad Vnd\ ) \\ Vnd \\ Head_3 \\ \begin{pmatrix} Author & Atitle & Page \\ Author & & \\ Author & Atitle & Page \\ Author & & \\ Author & Atitle & Page \\ Author & & \\ Author & & \\ Author & Atitle & Page \\ Author & & \end{pmatrix} \end{array}\right)\right)$$

Figure 3: An Output of the Precedent Processes

page with a recursive matrix whose components are a segment or a recursive matrix and whose columns and rows are separated by separators such as horizontal, line vertical line and spaces. For example, the content of fig.2 is represented with a matrix ( **left** **right** ) where **left** and **right** are recursive matrices corresponding to the left and the right halves of the content respectively. This process is realized by meshing a content and applying the projection profile method to each mesh.

**Classification** In this process, segments are mapped to tokens which correspond to attributes according to the features of the segments such as the segment size, type of fonts and so on.

In the database construction from documents, the content may include unnecessary information for the database such as headers "Editors", "Editorial board", "Articles" in fig.2. In order to handle these additional segments, we add several tokens corresponding to these segments. For the journal of fig.2, the added tokens are $Head_1$, $Head_2$, and $Head_3$ that correspond to the headers "Editors", "Editorial board" and "Articles" respectively.

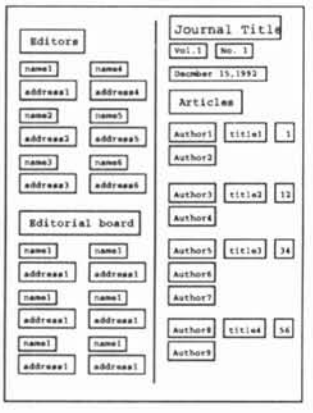


Figure 2: Segements of a Content

It is difficult to map each segment to an attribute completely by using only the numerical-valued features. Therefore, we define the token as a set of attributes. We assume that for any two tokens, their associated attribute sets are disjoint. Tokens for the journal in fig.2 are, for example,

| **token** | **attributes** |
|---|---|
| $Vnd$ | $\{Vol, No, Date\}$ |
| $Jtitle$ | $\{Journal\ Title\}$ |
| $Name$ | $\{Editor's\ Name, EB's\ Name\}$ |
| $Address$ | $\{Editor's\ Address, EB's\ Address\}$ |
| $Author$ | $\{Author\}$ |
| $Atitle$ | $\{Page\}$ |
| $Head_1$ | $\{Head_1\}$ |
| $Head_2$ | $\{Head_2\}$ |
| $Head_3$ | $\{Head_3\}$. |

After the segmentation and the classification processes, character codes of each segments are extracted by OCR and the strings are associated to each token as its value. Some segments are further divided by delimiters such as commas, colons and so on.

After these processes, the output data is a recursive matrix that consists of tokens. The matrix for fig.2 is shown in fig.3.

## 3 RULE DESCRIPTION

For the logical structure construction, the following functions are required:

- constructing a database tuple that consists of segments,
- identifying the class of tokens.

In order to realize these functions, we use a kind of regular expression that represents the layout of segments. The expression is constructed from the schema tree of the nested relation through the following four operations.

**Permutation** For the database tuple construction, the rule keeps the structure of the schema as much as possible. On the other hand, for the complete classification, the parser needs some information about the planar relationship of segments. For example, in the content of fig.2, *Vol*, *No* and *Date* are classified to the same class *Vnd*. In order to classify these segments completely, the parser need the information that these segments are horizontally located in the order of *Vol,No,Date*.

We assume that contents have multi-column layouts which has the recursive structure like the matrix of fig.3. We make an order of component in the top-left to bottom-right direction, that is,

for $(n_1, m_1)$ component $C_1$ and $(n_2, m_2)$ component $C_2$ of a matrix, if $(n_1 = n_2 \wedge m_1 < m_2) \vee (m_1 = m_2 \wedge n_1 < n_2)$ holds, then $C_1$ is located before $C_2$ in the order. (1)

We regard the schema tree as ordered tree in which the child nodes of each internal node are ordered in the top-to-bottom direction. The ordering information is given to the schema tree by permuting the nodes.

**Planar Relation** Ambiguity often occurs when an internal nodes of schema tree is matched with a $n$-by-$m$ matrix $(n, m \neq 1)$. In order to reduce the ambiguity, we assign the vertical/horizontal relationship to the internal nodes of schema tree. Four kinds of labels are attached to the internal nodes. Label $h$ and $v$ means that segments of its child nodes are located in horizontal and vertical direction respectively. Label $h^*$ and $v^*$ means that its child node is repeatedly located in horizontal and vertical direction respectively.

**Additional Nodes** As described in the classification, the contents have more data than required by database. These data are very useful in the analysis of the contents. For example, if the header "Editorial Board" does not exist in fig.2, it is difficult to separate the data for editors and editorial boards. We add some nodes corresponding to these segments to the schema tree.

**Space Segment** In journal contents, data is sometimes represented in a table-like format like editors, editorial boards and articles of fig.2. In the table format, null components are very important to extract logical components. For example, if we does not consider the null components of articles in fig.3, it is difficult to associate authors with articles in the table. Therefore we add the null component information to the leaf of the schema tree. The rule for the journal of fig.1 is shown in fig 4.

Note that if we ignore the labels of the internal node, this rule can be transformed into the normal regular expression, by unfolding it in the depth-first, top-to-bottom direction. For example the rule of fig.4 is transformed to

$$Head_1\ (Name\ Address)^*\ Head_2\ (Name\ Address)^*\ Jtitle\ Vnd\ Vnd\ Vnd\ Head_3\ (Author^*\ Atitle\ Page)^*. \quad (2)$$

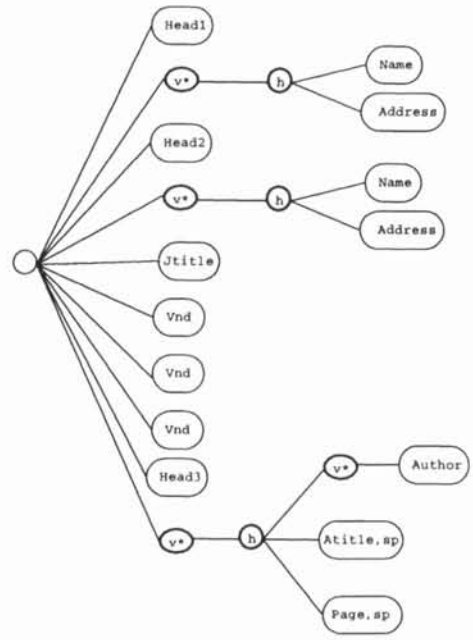


Figure 4: A Rule

## 4 Parser

The parser first unfolds all 1-by-n and m-by-1 matrices (vectors) according to the condition (1). For example the matrix of fig.3 is unfolded to be

$$Head_1\ Editor\ Head_2\ EB\ Jtitle\ Vnd\ Vnd\ Vnd\ Head_3\ Article$$

where *Editor*, *EB* and *Article* stand for matrices for editors, editorial boards and articles respectively. If the matrix does not include non-vector matrix, the string of tokens are obtained. In this case, the string can be analyzed by the regular grammar (2). In general, the derived data consists of token strings and matricies. For the part of token string, the parser analyzes it as the ordinal regular expression parser. For the matrix part, the parser begins to analyze with the $(1, 1)$ component. In this parsing, if the matrix is analyzed with the Kleenean closure part of the rule, it takes much time for analysis. Hereafter we show the efficient parsing for this case.

For a matching of a subtree $T$ and a simple n-by-m matrix $M$, we first consider the way to decompose the $M$ vertically (horizontally). In the following discussion, we assume the root of $T$ is labeled with $h^*$ without loss of generality. In this case, the matrix is decomposed vertically. Let consider the pattern of the first row of the matrix. The pattern is obtained by the following procedure.

```
Procedure first(T)
  begin
    if T is leaf then
      return label(T);
    if the label of T's child is h then
      E := ε
      for all grandchild node N of T do
        E := E first(N);
      return E;
    else
      return first(leftmost grandchild of T);
    end if
  end
```

In this procedure, label($N$) stands for the token assigned to the node $N$. This procedure is easily generalized to generate the pattern for $n$-th row. The parser calls this procedure to get the regular expression for the $k$-th row. If it gets the regular expression that are not ambiguous, then applies it to the corresponding rows of the matrix and decompose it. For example, let consider the matching of the subtree corresponding to the articles in fig.4 and the matrix corresponding to the same part in fig.3. In this case, since the root of the subtree is labeled with $v^*$, the matrix is decomposed horizontally. The regular expression for the first column is $(Author^*)^*$. Since this is ambiguous, the parser obtains the regular expression of the second column. The expression of the next column is $(Atitle\ sp^*)^*$. Since this expression is not ambiguous, the parser analyze the second column of the matrix with this regular expression. The result is the following three matrices.

$$\begin{pmatrix} Author & Article & Page \\ Author & & \\ Author & Article & Page \\ Author & & \end{pmatrix} \quad \begin{pmatrix} Author & Article & Page \\ Author & & \\ Author & & \end{pmatrix}$$

If there is an internal node in the grandchildren of $T$, the parser apply the same procedure to the decomposed submatricies recursively. In this example, the first column matches with $Author^*$. Second and third rows also match with $Atitle$ and $Page$ respectively. Thus the matrix of articles matches with the sub regular expression $(Author^*\ Atitle\ Page)^*$.

## 5 CONCLUSION

This paper discusses the method to extract the logical structure of journal contents from its image data. This method is unique in presenting the manner of rule construction from database schema systematically.

Currently we have been implementing each function of our document processing system. The accuracy and the performance of our methods will be reported near future. For the database construction, we need to avoid making ambiguous rules. We have not fully studied on the ambiguity of our extended regular expression yet. The ambiguity deeply depends on the accuracy of the classification process. We are now discussing the relationship between the classification accuracy and the ambiguity. In the syntactical approach, the error correcting is an important problem. Document processing requires numerical-value based analysis where the error is unavoidable. Therefore the error correcting ability is especially important in the document processing. We are now trying to extend the error correcting technique of the regular expression parser.

## References


[1] H. Eirund, D. Malerba, and G. Semeraro. "An Experimental Page Layout Recognition System for Office Document Automatic Classification: An Integrated Approach for Induscrive Generation". In *Proceedings of 10th International Conference on Pattern Recognition*, pages 557–562, 1990.

[2] S. W. Lam, D. Wang, and S. G. Srihari. "Reading Newspaper Text". In *Proceedings of 10th International Conference on Pattern Recognition*, pages 703–705, 1990.

[3] K. KIse, J. Sugiyama, N. Babaguchi, and K. Tezuka. "Layout Model Based Analysis of Document Structure". *The Transaction of the Institute of Electronics, Information and Communication Engineers (in Japanese)*, J72-D-II(7):1029–1039, 1989.

[4] K. Y. Wong, R . G. Casey, and F. M. Wahl. "Document Analysis System". *IBM journal Res. Develop.*, 26(6):647–656, 1982.

[5] T. Akiyama and I. Masuda. "A Segmentation Method for Document Images Without the Knowledge of Document Formats". *The Transaction of the Institute of Electronics, Information and Communication Engineers (in Japanese)*, J66-D(1):111–118, 1983.

[6] Q. Luo, T. Watanabe, U. Yoshida, Y. Inagaki, and T. Saito. "Understanding of Library Cataloging Cards on the Basis of a Knowledge-Based Approach". *The Transactions of Information Processing Society of Japan (in Japanese)*, 31(12):1755–1767, 1990.

[7] S. Tsujimoto and H. Asada. "Understanding Multi-articled Documents". In *Proceedings of 10th International Conference on Pattern Recognition*, pages 551–556, 1990.

[8] H. J. Schek and M. H. Scholl. "The Two Roles of Nested Relations in the DASDBS Project". In *Lecture Notes in Computer Science - Relations and Complex Objects in Databases*, pages 50–68. Springer-Verlag, 1987.