ANASTASIL:

A Hybrid Knowledge-based System for Document Layout Analysis

Andreas DENGEL & Gerhard BARTH

German Research Center for Artificial Intelligence (DFKI)
P O Box 20 80
Erwin-Schrödinger-Straße, D-6750 Kaiserslautern, Fed. Rep. of Germany
Phone (01149) 631-205-3213

Abstract

This paper describes a knowledge-based system for the identification of the different regions of a document image. It uses a hybrid, modular knowledge representation, a so called geometric tree being its essential part. This tree is used to perform a best-first search in combination with a "hypothesize & test"-strategy. It produces an internal, editable description of the entire document and its constituents. The system has been implemented for the analysis of single-sided business letters in Common Lisp on a SUN 3/60 Workstation. It is running for a large population of different business letters. The results obtained have been very encouraging and have convincingly confirmed the soundness of the approach.

1 INTRODUCTION

The issue of a paper-free office has recently enjoyed vivid considerations, but its full realization is still far from being accomplished. The development of suitable storage devices with direct access, coupled with high speed data networks is still being investigated. A standardization of external comunication protocols has yet to be established. Moreover, paper consumption is increasing by 10-15 % every year [Schäfer and Fröschle, 1984]. Thus, the replacement of paper as an information conveying medium seems to be difficult to achieve. Therefore, the need to automatically read and transmit large volumes of information that are contained in paper documents becomes increasingly important.

The intention is to design systems which embody knowledge about the basic structures of different kinds of documents, as well as a set of characteristics of their components and the special relations among them. The resulting knowledge sources are used to analyze and identify the different components of a paper document and transmit them into an internal, electronical representation.

Relevant work in this field includes the use of classification and segmentation methods to establish a formal representation of the whole document and the different layout objects within it. Different techniques have been proposed and used, to varying degrees and success. The resulting formal representation of the document page is the input for a highlevel control structure, that interprets the different layout objects, thereby using different knowledge sources.

To automatically "read & understand" a document, classical approaches of pattern recognition, concepts for a suitable knowledge representation and several AI-techniques can be fruitfully combined. Many applications using knowledge-based systems have been developed in the last years. [Woehl, 1984] i.e., illustrates the use of relational data bases coupled with a PROLOG expert system. The applications of production systems for document understanding have been proposed by [Kubota, Iwata and Arakawa, 1984] and by [Niyogy and Srihari, 1986]. For the analysis of business letters, [Bergengrün, Luhn, Maderlechner and Ueberreiter, 1986] used ATN´s (Augmented Transition Networks) in combination with fuzzy relationships. To model syntactical knowledge about paper forms [Domke, Günther & Scherl, 1986] proposed the application of Petri-Nets and finally the use of X-Y trees for the representation of information about a document image has been described by [Nagy and Seth, 1984].

This paper proposes a hybrid knowledge-based system called ANASTASIL, which means: *Ana*lysis *S*ystem *t*o Interpret *A*reas in *Si*ngle-sided *L*etters. It is based on a tree search. The foundamental tree structure represents knowledge at different layout abstraction levels. The tree is called *geometric tree*. The nodes of the tree contain hypotheses for different logical objects, like *date* or *receiver* of the letter. Thus, the system generates working hypotheses about the semantic meaning of layout blocks in a document, by comparing its individual layout structure with the nodes in the geometric tree. To verify the hypotheses, a statistical data base (SDB) is used. It contains a set of local features of all possible entities in business letters. Branching in the tree is directed by different measures of similarity. Thus, we perform a best-first search, which represents a kind of the uniform-cost search, proposed by [Barr and Feigenbaum, 1981].

This paper first describes the overall architecture of the system (Section 2), and then gives details of the various components of the system. In Section 3 a special kind of page layout description is introduced and used to establish the geometric tree. Furthermore the Section describes the design of the statistical data base (SDB). Section 4 illustrates the use of the geometric tree and the SDB to identify several layout objects of a document page. Section 5 shows the principles how to extend the two kowledge sources, especialy the geometric tree. Finally, experimental results are discussed in Section 6.

2 ARCHITECTURE OF THE SYSTEM

The knowledge-based system that we are developing is composed of four basic parts:

- The input of the system consists of the digitized document image data. A **Document Preprocessing Modul** takes clusters of black pixels, which have been segmented as basic and composite layout-objects (characters, words, lines, textblocks) to obtain data about the various printed blocks in the document. The data are represented in a hierachical data structure [Dengel and Barth, 1988]. It includes for all layout objects intrinsic properties, as well as spatial relationships between them.
- The **Knowledge Base** contains the structural knowledge of the geometric tree and the SDB.
- The **Control Structure** (Inference Engine) tries to successively refine the layout of a concrete document, using the knowledge from the two sources. It uses different tools. The most important are: a consistency check, an agenda and several evaluation functions.
- Additionally the system contains a **Knowledge Acquisition Modul** for collecting new knowledge and automatically modify the different knowledge sources.

Figure 1 shows the overall system and the interaction of the four parts

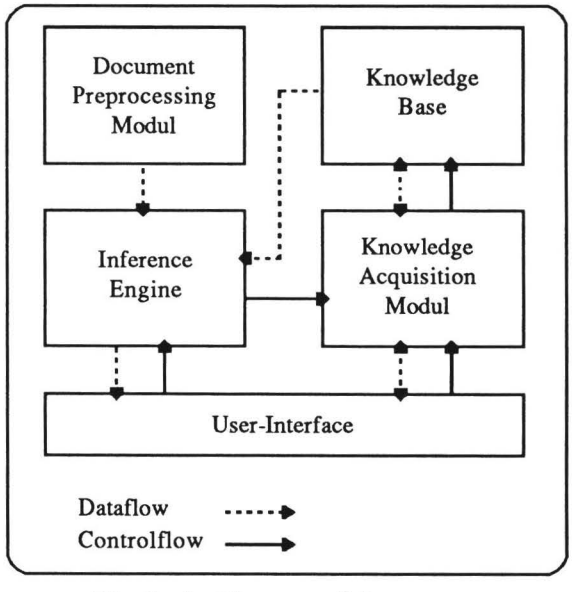

Fig. 1: Architecture of the system.

3 KNOWLEDGE REPRESENTATION

Document image analysis is a search problem, whereby the search space is the entire image. A digitized document page forms a binary two-dimensional space.

The effectiveness of model-based reasoning depends on the certainty and completeness of the underlying model. Any document is characterized by its content and its internal organization. Thus the electronic representation of a document must capture both the representation of its contents, as well as the document's layout and logical structure. To describe structural knowledge, we have developed our own formalisms for document page representation, because we believe that documents form a very special class of images for which data structures and algorithms capable of dealing with arbitrary pictures would be inefficient.

The structural elements of a document page, like columns, paragraphs, titles, lines and words of text are generally laid out as rectangular blocks. The orientation of the text information is along horizontal and vertical directions. Furthermore, these orientations coincide with the boundaries of a page.

A document page is considered as a rectangle, having a characteristic width and height. To describe its spatial structure, the page is divided into smaller rectangles by vertical and horizontal cuts. Cuts are placed in such a way that they do not intersect with textual or graphical areas. The subrectangles can recursively be divided in the same way, until the layout of the page is described in sufficient detail. To furthermore describe the logical structure, different rectangles are assigned a label, which describes their semantic content. We therefore use the following definition:

Rule:
For each refinement step in document layout description, choose one of the following possibilities:

1) The rectangle is left unchanged.
2) The rectangle is assigned a semantic label, which represents a hypothesis for the parts, it contain.
3) The rectangle is cut along one direction (horizontal, vertical) by one or more cuts and 1) or 2) are executed.

Consequently, most document pages can be partitioned into nested rectangular areas by order, position and orientation of cuts and by assignment with logical labels. Figure 2 shows an example of a partitioned and labeled letter.

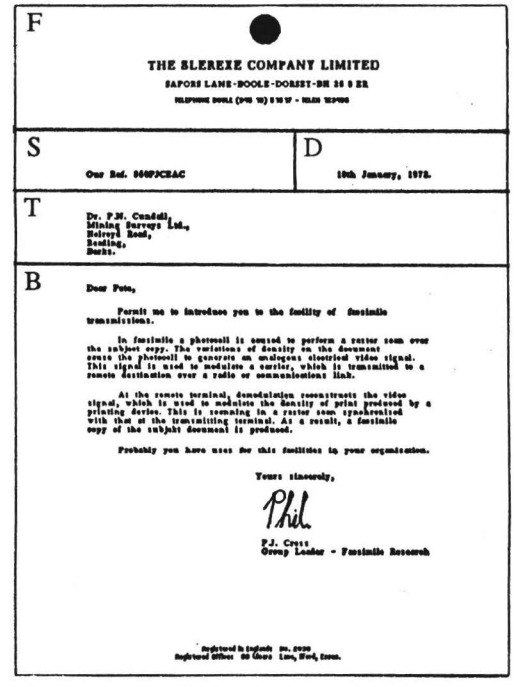

Fig. 2: Structure-representation of a letter [sender (designated as 'F'), receiver ('T'), subject ('S'), date ('D'), body of letter ('B') and white space ('W').

To transform the definition, we use a special notation with identifiers :H (horizontal), :V (vertical) to the orientation of a cut and :L to name a logical label. Additionally, we use real numbers, so that the letter shown in Figure 2 can be represented by the following list:

```
(:H 0.2 0.3 0.45  (:L 'F)
                  (:V 0.6  (:L 'S)
                           (:L 'D))
                  (:L 'T)
                  (:L 'B))
```

We use this description procedure to establish a hierarchical model, realized as a binary tree. We call it geometric tree. It describes by means of cuts and labels the individual layout structures of a document class by stepwise refinement. An example for a simple geometric tree is shown in Figure 3.

The advantages of the tree structure include:

- a guided search from an abstract towards a more concrete layout is possible,
- redundant layout information is avoided,
- a document layout can be described at different levels of specificity, dynamically adapting to the amount of information available.

Each node in the tree represents a layout class, whereby terminal nodes correspond to concrete layouts of a given document. We take the notation described above and transform this knowledge representation in a combined *list-in-list* implementation. One level represents the different alternatives, whereas at the other level complete structures are described.

Another major advantage of this model is the fact that at each level no area of the document page is left unaccounted for. As a consequence thereof, the model is fault-tolerant with respect to preprocessing errors. Should the address of a letter, which usually consists of one single block, be erroneously split into different lines by the segmentation procedure, the lines are still contained within the area hypothesized to contain the address. This is quite different from other models that attempt to classify each single block in a document. However, completeness and certainty of model knowledge is responsible for the effectiveness of model based reasoning. We do not want to propose our approach as the universal solution, but it behaves pretty well for most practical applications.

In addition to the structure model, we use a statistic data base (SDB), where we have stored the examination results of a few hundred business letters. The statistical validation for each possible logical object is transformed into a set of rules, one for each possible object. The rules have the form:

if *<predicate>* **then** *<confidence-value>*

Lateron, during the analysis, the SDB is examined to pinpoint those predicates that help identify different logical objects.

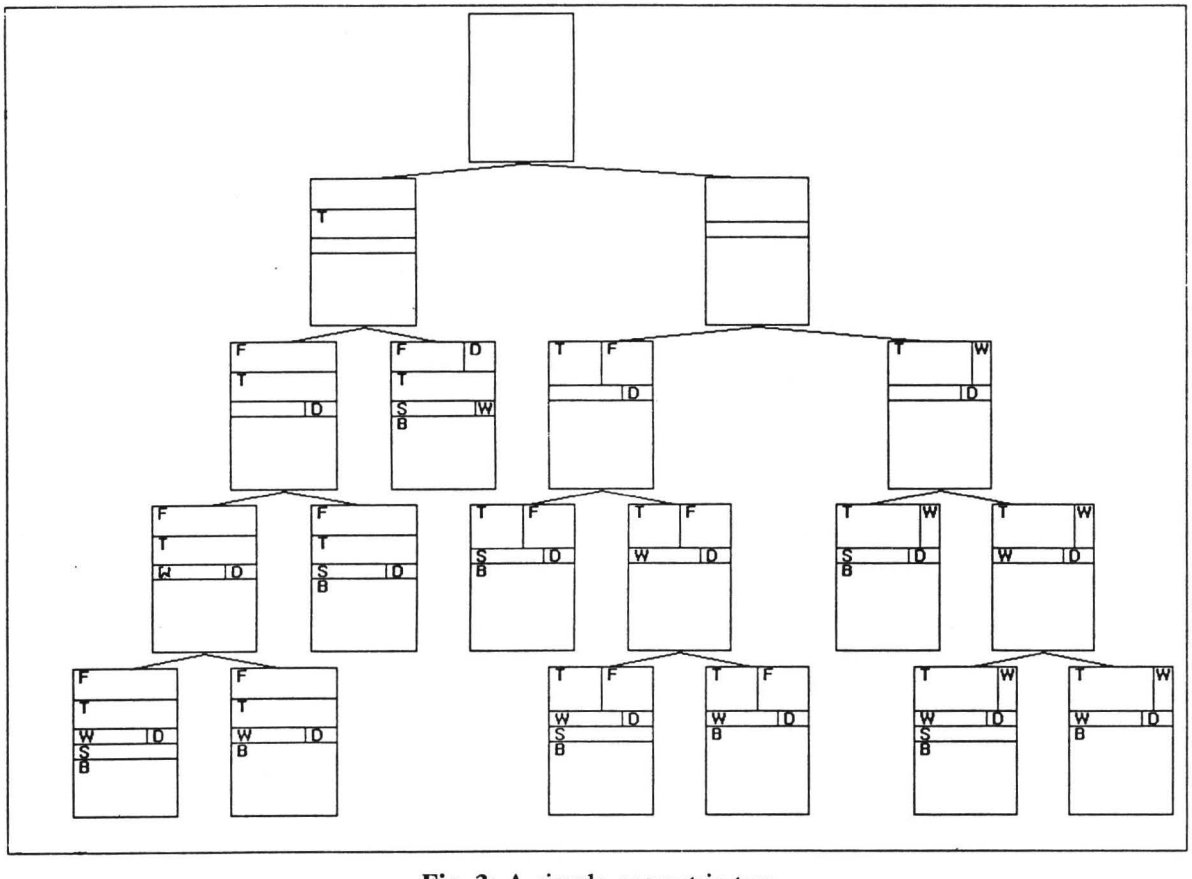

Fig. 3: A simple geometric tree.

4 CONTROL STRUCTURE

The input for ANASTASIL is a document whose constituents are represented as rectangular blocks [Dengel, Luhn and Ueberreiter, 1987]. We use a goal-driven (top-down) approach for our system, which employs a "hypothesize & test" strategy for deriving conclusions. Figure 4 sketches the strategy.

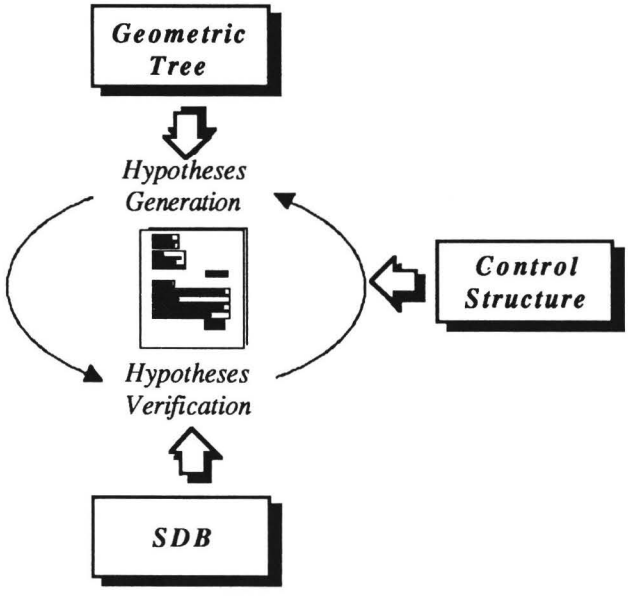

Fig. 4: "hypothezise & test"-strategie of ANASTASIL.

Interpreting a given document amounts to finding a path in the geometric tree from the root to one of its leaves. Starting at the root, in each step the actual document is matched with the two sublayout-classes of the actual node. The degree of similarity between the layout of the given document and the nodes in the model have to be quantified. Semantic labels in distinct nodes represent hypotheses about the semantic meaning of the contained parts. To verify a hypothesis, the features defined in the SDB have to be compared with the blocks of the area to be examined. Each inspected node gets a measure of belief (MB) for its similarity with the actual document. This measure of belief is composed of a confidence value for its quality of cut matching as well as evidence for hypotheses verification. We use an agenda to store the different intermediate conclusions and conduct a best-first search in the geometric tree. Thus, several "hypothesize & test" processes are performed, and the system reaches a satisfactory conclusion no sooner than a leaf in the geometric tree is reached and all areas of the document are labeled.

While matching a given document with a layout class of the geometric tree, we try to take into account small variations within a document´s layout. Therefore, we allow small shifting of the cuts with respect to their original positions in the layout classes. If a cut position intersects any textual or graphical block of the document, the control mechanism searches for some alternative positions. The validation function for cuts (see Formula 1) works in such a way that the amount of shifting the original position is computed. When looking for an different alternative position x, small shifts should not count as much as large shifts. Thus, the quality of each cut is based on a measure of belief $v(x)$, which is calculated by the following validation function.

$$v(x) = \begin{cases} f_1(x) * r_1(x) & l_1 \le x \le c \\ f_2(x) * r_2(x) & c < x \le l_2 \end{cases} \qquad (1)$$

Thereby, c denotes the original position and l_1 and l_2 describe the boundaries of the area to be partitionated. f_i and r_i are defined by the Formulas 2 and 3.

$$f_i(x) = 1 - \left[\frac{(x-c)}{(l_i - c)} \right]^2 \qquad (2)$$

$$r_i(x) = \left| 1 - \frac{(x-c)}{(l_i - c)} \right|^n \qquad (3)$$

$f_i(x)$ defines the entire validation function for cut shifting within the interval $[c, l_i]$, whereas $r_i(x)$ denotes a factor which determines the entire curve of the function. The curve of the function can be altered depending on the degree of layout-standardization of the underlying document class.

Therefore, the variable n is assigned a non-negative value, whereby a lower value indicates less standardization and a higher one more standardization. Figure 5 shows the semantics of the formulas with $n = 0,..,3$.

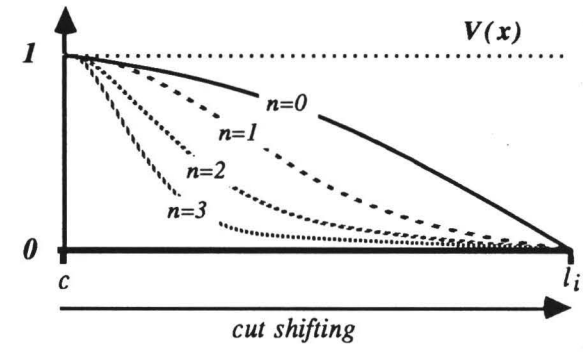

Fig. 5: Curve of the function depending on n.

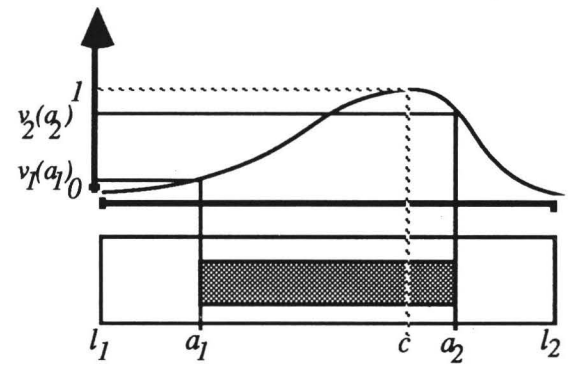

Fig. 6: Example for cut-validation.

In case a cut like the one in Figure 6 intersects a block (shaddowed area), the alternatives a_1 and a_2, are the closest possible positions for a cut with respect to its original position. The measure of belief for the two alternatives is calculated by means of a partially defined function (see Formula 1). The cut has to be placed somewhere between the delimiters l_1 and l_2 of the area under consideration. They confine the domain of the functions. The functions are converging in point c. The maximum of $v(x)$ is 1, in which case no alternative position must be searched. The minimum 0 is attained at the boundaries l_1 and l_2.

Usually there is more than one possible cut for each node in the geometric tree. Hence, for every match with a layout class we fold the values of v for each of the cuts into a single confidence factor. The weighting of the single cut values is done according to the relative lengths of the cuts. Since we require the function values to be less than 1, we normalize them with respect to the total length of the cuts from the root of the geometric tree to the respective node. That means:

The contribution of each cut to the total measure of belief for the layout equals its share of the total cut-length in the document.

This yields the following expression for V_i:

$$V_i = \frac{1}{C_i}\left(V_{i-1} * C_{i-1} + \sum_{j=1}^{k_i} v_{ij} * c_{ij}\right) \qquad (4)$$

$$V_0 = v_{00} = 1$$

In (4), i denotes the level of a tree node (0 being the root), P_{i-1} is the parent node of node P_i. The number of cuts in a node is k_i.

The terms v_{ij} denote measures for cuts j of length c_{ij}, $1 \leq j \leq k_i$. The normalization factor is the total length C_i of all previous cuts:

$$C_i = C_{i-1} + \sum_{j=1}^{k_i} v_{ij} * c_{ij} \qquad (5)$$

$$C_0 = c_{00} = 0$$

The result of the validation process only denotes the quality of cut matching. All patterns obtained as plausible combinations of cut positions are used for further examinations. In other words: all areas getting labels (hypotheses) during this pattern matching step, have to be verified. Thus, evidence is gathered to confirm or refute a hypothesis. Therefore, all layout segments being contained within the labeled area are considered as common parts of one and the same logical object (indicated by the label). Validation of a hypothesis will be achieved by inspection of the corresponding characteristics and the appropriate confidence value in the SDB. The relevant parameters for the application of the rules are calculated. Depending on the specific feature, we distinguish between measures of belief and measures of disbelief.

Normally several rules can be applied. So, it is necessary to combine the different confidence values. Each resulting confidence value represents the degree of supporting or refuting the labeling hypothesis. Whereas probability theories, like the Bayesian formulas refers to conditional probabilities, the SDB is based upon the combination of completely independent events with their respective probabilities. To this end, we use Dempster-Shafer´s rules of combination [Shafer, 1976].

Consequently, the measure of belief for the similarity with a document class in the geometric tree (Formulas 1 to 5) and the one for the hypotheses verification are combined and the result is used to guide the best-first search.

5 KNOWLEDGE ACQUISITION

However, in some cases the system is not able to classify the actual given document (see Fig. 7.1).

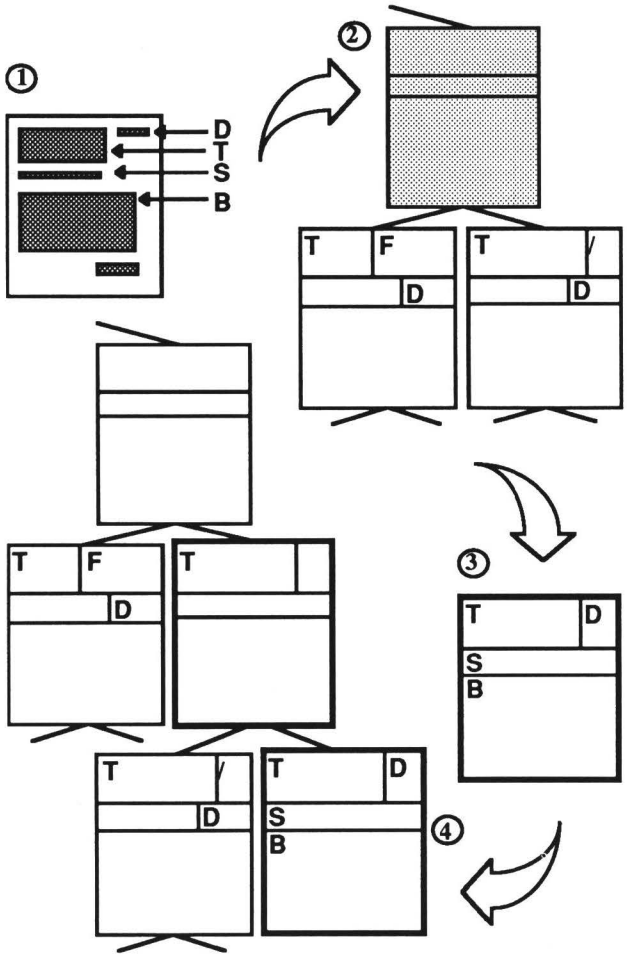

Fig. 7: Example for knowledge acquisition in the geometric tree.

Therefore, we have constructed a knowledge acquisition component which allows the modification and extension of the knowledge base. It is activated as soon as no satisfactory conclusion can be reached. In such a case, the control structure provides the actual best intermediate conclusion (see Fig. 7. 2, the shaddowed pattern) and presents it in a specific window on the screen. The user himself is now able to employ different graphical facilities and thus complete the

layout description of the pattern by setting cuts and labels. The result is a new layout class (Fig. 7. 3).

Subsequently, the graphical pattern is converted into the internal knowledge representation. This is done by a model generator, which automatically extends the knowledge sources by the new knowledge. In particular, the geometric tree is extended by modifying the appropriate subtree. Therefore, a common super-layout-class of the old subtree and the new layout-class forms the root of the new subtree (see Fig. 7.4).

6 EXPERIMENTAL RESULTS & FUTURE WORK

On the next page, see Figure 8, we illustrate some examples for interpretation results of typical business letters that have been analyzed and classified by ANASTASIL. The system performs a search and finally brought up the interpretations shown in the examples. To classify a conventional business letter, our system needs between 1 and 7 cpu seconds on the average, depending on the complexity of the document layout. For the analysis of very complicated letters, which differ greatly from the layout model given in the geometric tree, we are capable to trigger backtracking in order to solve the appropriate classification problem. Then, cpu time increased up to 10 seconds. If no classification is possible, the knowledge acquisition component will be activated. Thus, our system is capable to learn by example and for that reason was able to classify all letters we have considered.

The work reported about in this paper will be the starting point for a research project to be conducted at the newly founded German AI Research Center. The project will be titeled by "Automatic Reading & Understanding" and tries to understand the abstract semantics of multi-media docu-ments.

REFERENCES

[Bergengrün, Maderlechner, Luhn and Ueberreiter, 1986] O.Bergengrün, A.Luhn, G. Maderlechner and B. Ueberreiter, *Dokumentanalyse mit Hilfe von ATN's und unscharfen Relationen*, Proc. of 9. DAGM-Symposium, Braunschweig 1987, p. 78

[Dengel, Luhn and Ueberreiter, 1987] A. Dengel, A. Luhn and B. Ueberreiter, *Model Based Segmentation and Hypothesis Generation for the Recognition of Printed Documents*, Proceedings of the SPIE'87, Vol. 860, Real Time Image Processing: Concepts and Technologies, Cannes 1987, p. 89

[Dengel and Barth, 1988] A. Dengel and G. Barth, *Document Description and Analysis by Cuts*, Proceedings of the RIAO'88: User-Oriented Content-Based Text and Image Handling, Vol 2, Cambridge, MA, March 1988, p. 940

[Domke, Günther and Scherl, 1986] L. Domke, A. Günther and W. Scherl, *Wissensgesteeurte Formularinterpretation mit Hilfe von Petrinetzen*, Proceedings 8. DAGM-Symposium, Paderborn 1986, p. 29

[Kubota, Iwata and Arakawa, 1984] K. Kubota, O. Iwata and H. Arakawa, *Document Understanding System*, Proc. 7th Int. Conf. on Pattern Recognition, Montreal 1984, p. 612

[Nagy and Seth, 1984] G. Nagy and S. Seth, *Hierarchical Representation of Optically Scanned Documents*, Proc. of the 7th Int. Conf. on Pattern Recognition, Montreal 1984, p. 347

[Niyogy and Srihari, 1986] D. Niyogy and S. Srihari, *A Rule-based System for Document Understanding*, Proc. AAAI'86, p. 789

[Shafer, 1976] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976

[Schäfer and Fröschle, 1984] M. Schäfer and H. P. Fröschle, *Die Vision vom papierlosen Büro*, Funkschau 19, 1986, p. 46

[Woehl, 1984] K. Woehl, *Automatic Classification of Documents by Coupling Relational Data Bases and Prolog Expert Systems*, Proceedings 2nd Conf. on Very Large Data Bases, Singapore 1984, p. 529

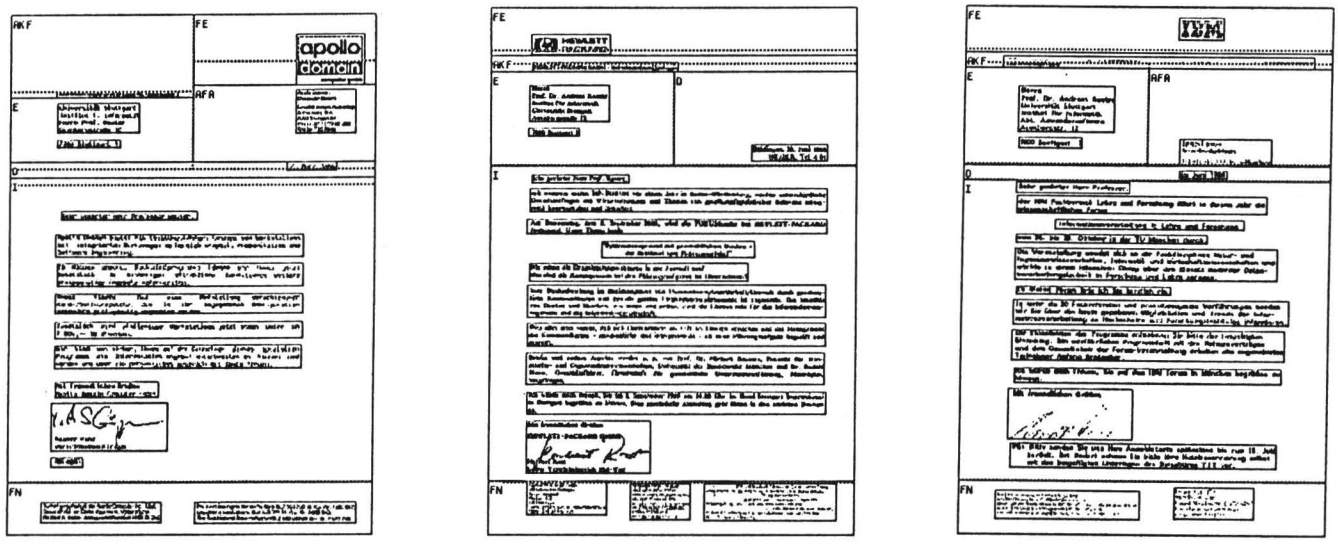

Fig. 8: Analysis Results of ANASTASIL (The two examples have been scanned with a density of 75 dpi. The labels denote different abbreviations of logical objects).