An Integrated Document Management System

ANDREAS DENGEL
German Research Center for Artificial Intelligence (DFKI)
P. O. Box 20 80, Erwin-Schrödinger-Straße, D-6750 Kaiserslautern, Fed. Rep. of Germany
Mail Address: dengel@uklirb.uucp., Phone: (++49) 631-205-3215, Fax: (++49) 631-205-3210

NELSON M. MATTOS & BERNHARD MITSCHANG
University of Kaiserslautern
CS Department, Erwin-Schrödinger-Straße, D-6750 Kaiserslautern, Fed. Rep. of Germany

1 INTRODUCTION

All activities in an organisation require or produce information. Therefore, a document is not only the main information carrier but also the central aid for the integration of office functions /1/. The office of the future will be characterized by new fundamental tendencies which lie not only in a more comprehensive possibility for information interchange but also in an altered processing of the information /3/. As a part of communication, documents play the central role in todays office domain. A document can be described as a structured amount of text that can be interchanged between an originator and a receiver /2/. A given document is characterized by its contents and its internal organization, where the organization is defined by a logical and a layout structure. The elements of the logical structure of a document are constituents like receiver, sender, author or signature. Layout objects are titles, text blocks, words or single characters.

In today´s office, software tools on personal workstations facilitate the handling of electronic documents. An unavoidable consequence thereof is an increasing amount of information to be processed and despite some advances, paper consumption increases 10-15% every year /4/. This continuing dependence on paper documents as an important information medium and the simultaneous thrust in the direction of electronic media require the development of systems, which allow information (structural, content-based) to be exchanged without loss between paper and electronic carrier media. As a result, it will become possible to manage both electronic and paper documents by using a common electronic archive. The framework of a document processing and management system, which we are developing has the following characteristics:

- *Reception of printed information.* Using pattern recognition methods, it is possible to automatically transform printed information into a symbolic representation without any loss of information.
- *Document evaluation.* Applying AI techniques, a document interpretation procedure is initiated. It attempts to identify several layout objects of a document at hand by their logical meaning, thus creating a conceptual structure. Moreover, it provides a restricted context for further content-based analysis /5/. Consequently, a OCR-system is used for partial recognition of textual information within the logical objects. The resulting ASCII-Code is employed to initiate a full text search with keywords in connection with morphological analysis. As a result of the reception and evaluation phases, we obtain different perceptions of a document, namely a layout, a conceptual, and a semantic view.
- *Document management.* To support the processes of reception and evaluation and their corresponding hybrid document representations in an appropriate manner, knowledge representation concepts will be applied. Furtheron, a persistent and efficient document management is a prerequisite to the integration of subsequent services, e.g., document archiving and retrieval, document manipulation (i.e., DTP), and mailing.

2 OVERALL ARCHITECTURE

In order to accomplish these objectives, our architectural approach towards an integrated document management system (reflected in Fig. 1) is consequently based on a

- *knowledge base management system (KBMS)* /6/ responsible for effective document management and service integration, as well as on a
- *document analysis system* /7/ capable of document reception and evaluation.

Due to the KBMS's flexibility in document representation, this layered architecture allows for easy extensibility of further services (e.g., archive browser, response generator) under a user-friendly interface.

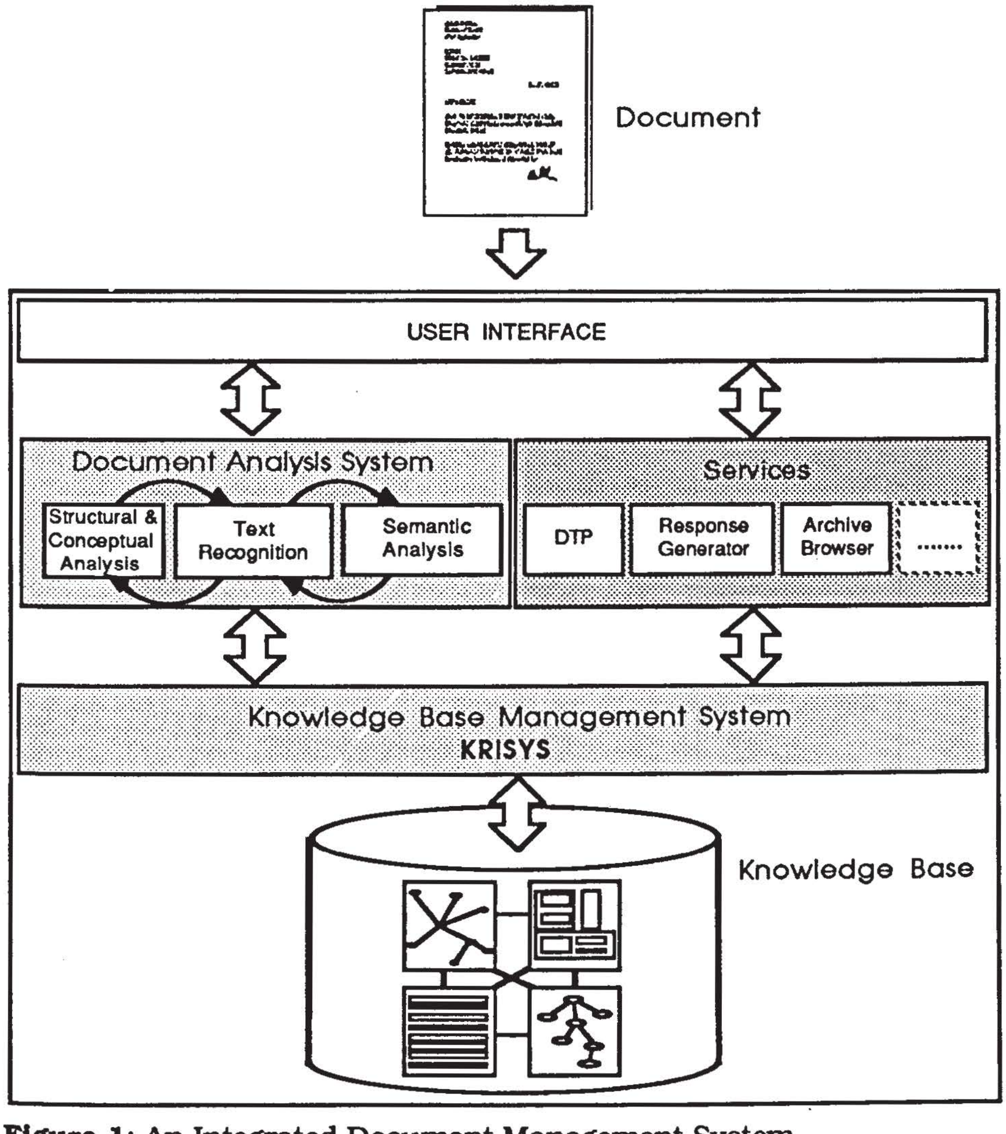

Figure 1: An Integrated Document Management System

The ANASTASIL-system /7/ being developed at the University of Stuttgart and the German Research Center for AI uses a hybrid knowledge base to support structural and conceptual analysis of paper documents. The goal of the analysis is to provide a multitude of information possibilities. Firstly, the layout of the whole document and its parts is completely described in a treelike structure that offers easy modification and efficient access down to the smallest layout objects /8/. Secondly, logical labels (e.g., the sender or the date of a letter) are assigned to all layout objects recognized. In addition, an Optical Character Recognition (OCR) procedure is applied for analyzing each layout object and finally assigning every basic textual layout object (characters) specific ASCII codes. Moreover, text recognition can be supported by appropriate views on a structured dictionary depending on the conceptual meaning assigned to the layout object under consideration. The obtained word descriptions are furthermore used as key words to access associated conceptual models. The results of the analysis offer different views on documents with respect to different abstraction levels. Figure 2 schemes the central information structures encountered during document analysis. The bidirectional arrow represents relations between the different structures.

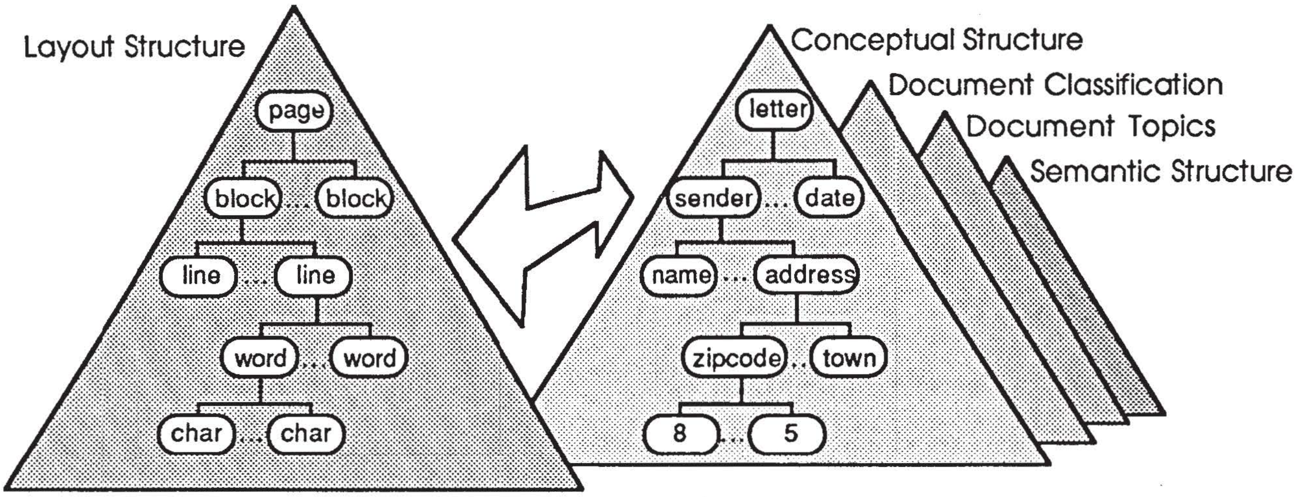

Figure 2: Central Information Structures encountered during Document Analysis.

The task of the underlying document management component comprises efficient and reliable management of the various information structures used or generated by ANASTASIL, or other services. The compliance with these constraints is crucial to the overall system behavior and efficiency. Therefore, we rely on the KBMS KRISYS /1/ (see Fig. 1) developed at the University of Kaiserslautern. KRISYS offers a rich set of powerful and flexible constructs for object modeling and manipulation /9/, as well as object management, e.g.:

- *Structural object orientation* (i.e., support for complex objects) is provided by the abstraction concept of aggregation.
- *Behavioral object orientation* is yielded by integrating procedural informatiom, i.e., methods, into the object description.
- *Sematic structuring of information* is provided by means of an integrated view of abstraction concepts (classification, generalization, association, and aggregation), allowing the same object to act in different roles at the same time /10/.
- *Active system behavior* is achieved by means of demons (allowing flexible reaction on certain events) and rules (used to represent intentional information being dynamically derived).
- Expressive *query and manipulation language.*
- Efficient knowledge base management and support of application programming.

Based on these facilities, KRISYS provides for an accurate representation and management of all information structures encountered during document processing.

In this paper, we describe the architecture of our integrated document management system in more depth focussing on ANASTASIL and KRISYS. Especially, we discuss the main issues of this integration and present solutions, which provides the basis for an efficient overall system behavior.

3 DOCUMENT ANALYSIS

The task of document analysis requires the scanning of a given document and the examination of the resulting binary image. The symbolic representation of a document has to capture information about contents as well as about logical and layout structure. The first step in the recognition process is scanning of the paper document as well as filtering and binarization of the internal document image. Subsequently, a segmentation method is used to establish a layout representation of the whole document and the different layout objects within it. Different techniques have been proposed and used, to varying degrees and success /7/. An overview of the different techniques is given in /21/. The resulting formal representation of the document page is the input for a highlevel control structure, that classifies the different layout objects as logical objects, like the *subject* and the *date* of a letter, or a specific *company logo*.

In the following, we introduce the main steps in document analysis in more detail thereby focusing on the data structure needed by ANASTASIL and provided by KRISYS.

3.1 DOCUMENT LAYOUT EXTRACTION

To automatically extract the layout structure of a given document, different phases have to be passed through. They are mainly based on methods of pattern recognition and are more or less supported by knowledge and AI techniques. The phases contain the classification of textual and graphical information, its segmentation in basic and composite layout objects, and their mapping into a data structure, which represents the appropriate layout of the given document.

Layout objects which are text and graphics elements, e.g., characters, words, text-lines, text blocks, bussines graphs, diagramms, company logos are hierarchically nested. The different objects can be described by rectangular regions of text or graphics information. Different results of the preprocessing have to be stored with each object:

- Position and size of layout objects resulting from segmentation processes /11/, /12/.
- ASCII codes provided by an optical character recognition procedure /13/. This representation must also take into account different alternative recognition results.
- Recognition results obtained by the analysis of graphical objects /14/, /15/.

3.2 REPRESENTATION OF DOCUMENT STRUCTURE

So far, we have described how documents are manipulated by ANASTASIL. But how can the knowledge base management system KRISYS be employed to model such documents?

KOBRA, the knowledge model of KRISYS, provides an object-centered representation of the real world /10/. That is, every entity existing in the application domain is expressed as an object of the KOBRA model, the so-called *schema*, in which descriptive, operational, and organizational aspects of the real world are integrated. In other words, a schema (not to be confused with a DB-schema!) is the symbolic representation of a real world entity (roughly analogous to *frame* or *unit* in other representation systems). It is always identifiable by a unique schema name and is composed of a set of attributes. The attributes may again be further described by aspects in order to characterize an object in more detail. Attributes are of different kinds. A schema may possess declarative attributes (slots) describing descriptive aspects of an object, procedural attributes (methods) that describe operational aspects, and structural attributes (abstraction relationships) used for expressing organizational relationships of the application domain.

The object-centered approach supported by KRISYS fits well the purposes of the representation of ODA (Office Document Architecture) /2/.

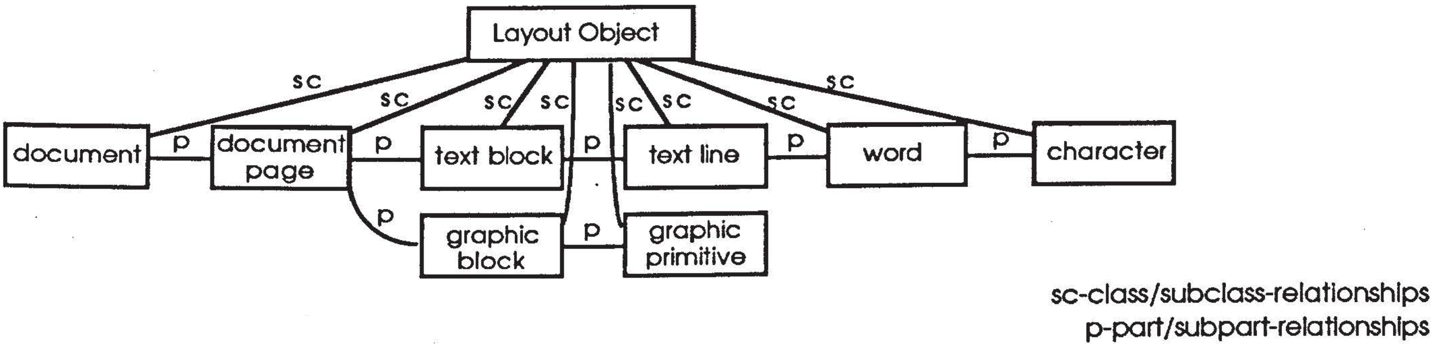

Figure 3a: Hierarchy of Layout Objects.

Layout objects are complex structures composed by other layout objects. For example, a document page is composed by several text blocks, which, in turn, contain several text lines. The latter ones are built of words that are composed of characters (c.f. Fig. 3a)

```
Layout Object
has-subclasses (document, document page, ... , character) terminal ownslot
x-origin                          terminal instanceslot
          possible-values (integer)
          cardinality [ 1 1 ]
y-origin                          terminal instanceslot
          possible-values (integer)
          cardinality [ 1 1 ]
width                             terminal instanceslot
          possible-values (integer)
          cardinality [ 1 1 ]
height                            terminal instanceslot
          possible-values (integer)
          cardinality [ 1 1 ]

document page
subclass-of (layout-object)       terminal ownslot
in-document                       nonterminal instanceslot
          possible-values (instance-of (document))
          cardinality [ 1 1 ]
has-text-block                    nonterminal instanceslot
          possible-values (instance-of (text block))
          cardinality [ 1 ∞ ]
has-graphic-block                 nonterminal instanceslot
          possible-values (instance-of (graphic block))
          cardinality [ 0 4 ]
```

Figure 3b: Representation as KOBRA Schemas.

Therefore, all layout objects hold the same information, however, with partially different semantics, i.e., all of them are complex objects built of distinct components. This is represented by means of the KOBRA model as shown in Figure 3b. The class *Layout Object* describes the aspects which every character, word, text line, etc. has in common. That is:

x-origin, y-Origin position of left-upper corner of circumscribing rectangle

width, height size of circumscribing rectangle

The different semantics of their components are then specified in the corresponding subclass by means of particular attributes and the aspects *possible-values* and *cardinality*. In addition, instances of object class *Character* and *Graphic-Block* have additional slot variables

ASCII ASCII code as resulting of the OCR-procedure

chain-code internal representation of the original binary image /17/.

It is important to point out the different types of attributes supported by KOBRA:

- Ownattributes (i.e., ownslots and ownmethods), as subclass-of in Figure 3b, are used to describe properties of the object itself, and as such may have values.

- Instanceslots and instancemethods, on the other hand, describe properties of the object´s instances, and have, therefore, no values (e.g., x-origin, y-origin, width, and heigth).
- Ownslots and instanceslots are further classified in nonterminals or terminals. Nonterminal slots indicate part-of properties (i.e., the components) of objects since their values correspond to other objects of the knowledge base (e.g., in-document, has-text-block, and has-graphic-block). Terminal slots, on the other hand, describe either characteristics of the objects themselves (terminal ownslots) or of their instances (terminal instanceslots).

Therefore, the abstraction concept of aggregation (/10/, /19/) is represented in KRISYS by means of user-defined attributes, allowing for the specification of several kinds of relationships, each of with very fine semantics (observe, for example, the distinct integrity constraint associated to has-text-block, has-graphic-block, which can not be expressed by systems supporting aggregation by means of one single part-of relationship).

3.3 REPRESENTATION OF DOCUMENTS

Documents are represented in the knowledge base as instantiations of the structure previously described. During preprocessing of a document, its different layout objects are extracted and represented as instances of corresponding classes, as illustrated in Figure 4. By means of generalization (i.e., class/subclass) and classification (i.e., class/instance) relationships, inheritance is automatically applied by KRISYS, exactly defining the properties with associated integrity constraints of every recognized layout object.

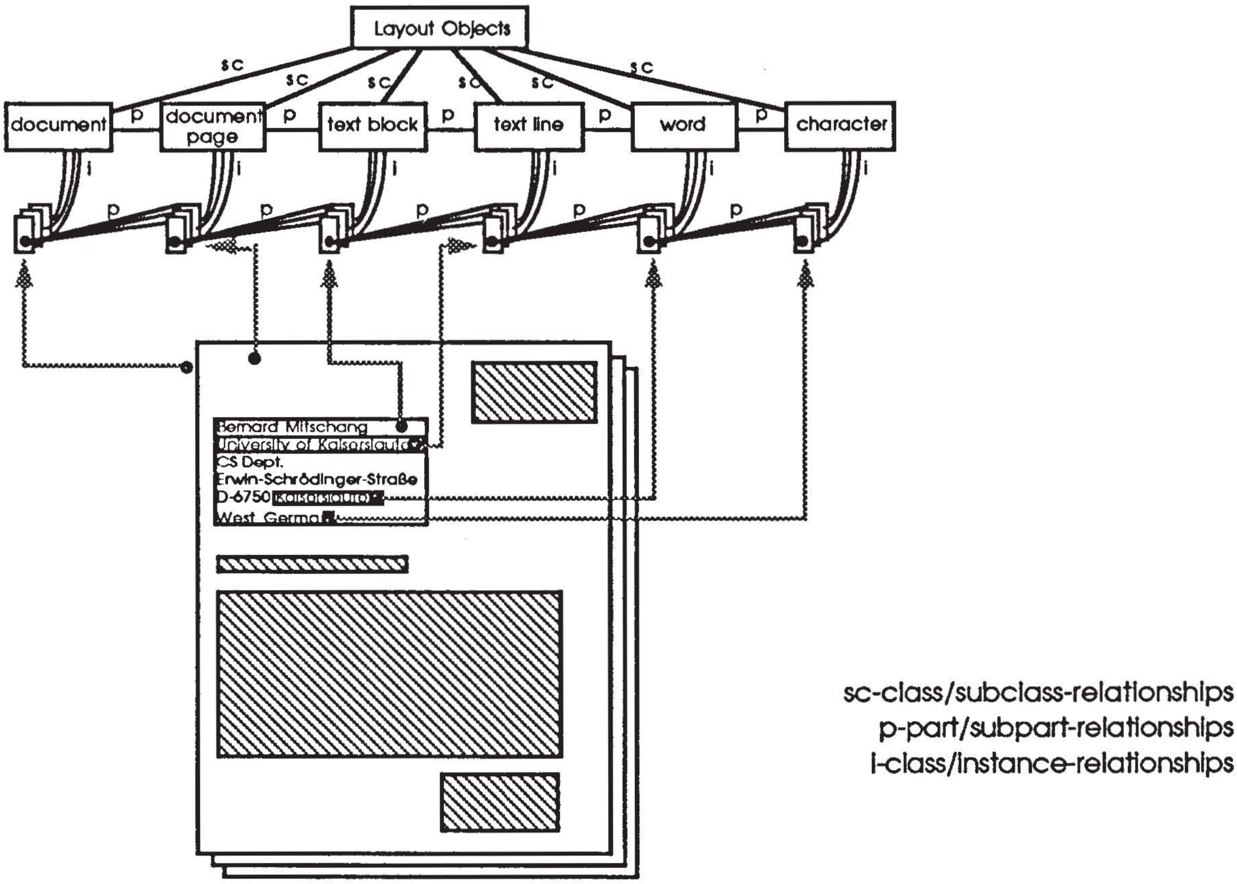

Figure 4: Representation of Document Layout.

Since the extraction process is based on hierarchical analysis of the document structure, it also provides the existing relationships between the several layout objects which are expressed in the part-of attributes of each introduced instance.

The KOBRA model provides several built-in reasoning facilities on specified abstraction relationships between objects (/9/, 10/). Inheritance, as mentioned above, is the reasoning as to the structure of an object apllied on generalization/classification hierarchies.

Aggregation-relationships are the basis for reasoning with so-called implied predicates (/10/, /20/). For example, the width and height of layout objects must grow upwards. Therefore, the knowledge about the size of a text line may be used by KRISYS either to infer minimum sizes for text blocks, pages, etc., or to control whether specified widths and heigths of layout objects are in accordance to each other.

3.4 DOCUMENT LAYOUT CLASSIFICATION

The goal of the classification phase comprises the assignment of semantics to parts of the layout structure such that essential logical objects of the document, like the sender, the receiver, or the footnote are determined.

A starting point for this process comprises formal attributes. Therefore, knowledge as to the possible layout and the composition of information in the document is used. Existing knowledge about document classes, which has been obtained by experience and from empirical tests, serves as a basis for the execution of this rough analysis. In a sense, we have developed our own data structures for document page representation /5/, because we believe that documents form a very special class of images. General data structures and algorithms for arbitrary picture representation would be inefficient.

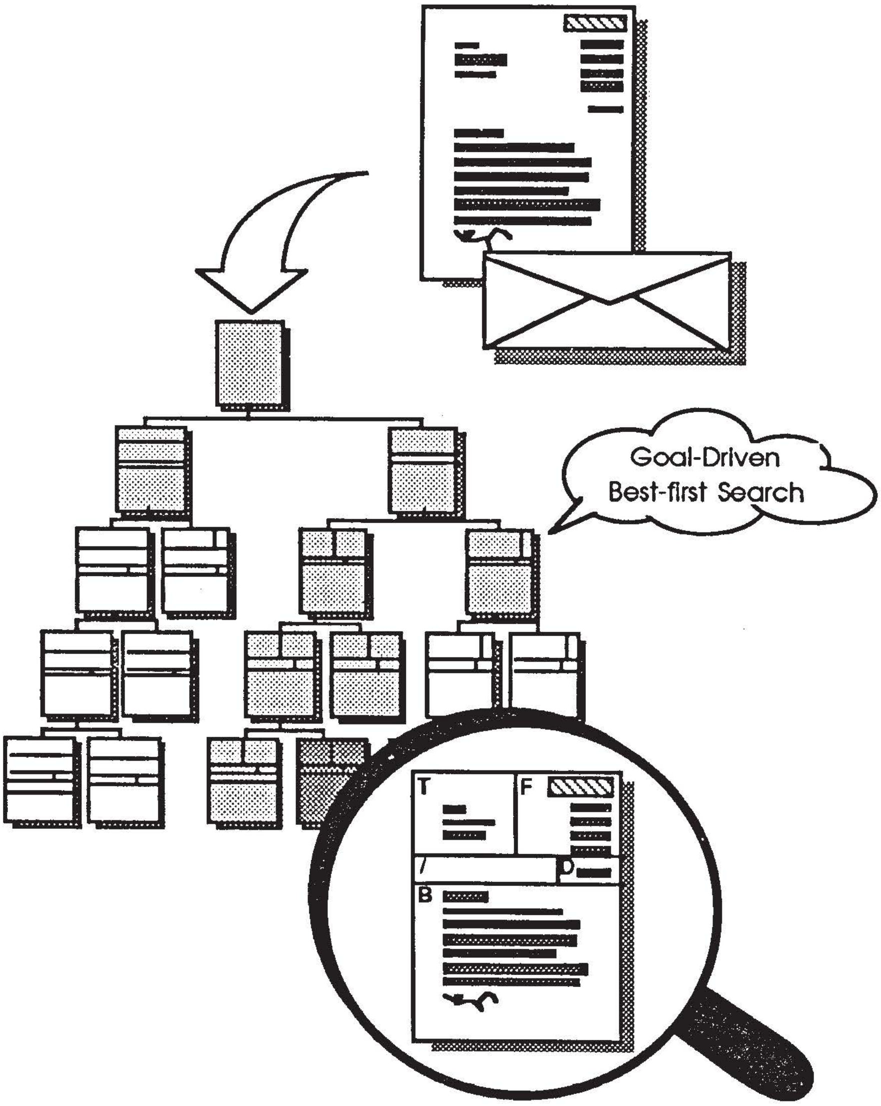

Figure 5: A Simple Geometric Tree and Principle of Best-First Search.

To obtain a document classification, we use a hybrid, modular knowledge base. It is composed of a document layout model and a Statistical Data Base (SDB). The ANASTASIL system, which have been used for document analysis, is based on a tree search. The fundamental tree structure represents knowledge at different layout abstraction levels. The tree is called *geometric tree* (see Fig. 5).

The nodes of the tree contain hypotheses for different logical objects, like *date* or *receiver* of the letter. Using this knowledge source during analysis, ANASTASIL generates working hypotheses about the logical meaning of layout objects 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 is composed of different packages of rules for the description of different possible logical objects /5/. During the classification process, the SDB is used to pinpoint these predicates that help identify different logical objects. Branching in the tree is directed by different measures of similarity. Thus, we perform a best-first search, which is a variant of the uniform-cost search, proposed by Barr and Feigenbaum /18/.

Classification is achieved by inspection of corresponding layout characteristics of the global document in combination with layout features of possible logical objects which are compared to those of given layout objects.

As a result of the classification phase, we obtain a document image in which all important constituents have assigned a logical label. The label indicates a common logical meaning for one or more layout objects grouped together.

In KRISYS, the geometric tree is represented by a generalization hierarchy. Every schema in this hierarchy contains slots corresponding to the labels, they indicate different logical objects of a document (such as sender, receiver, subject, date and body in the case of a letter) as well as their respective layout features within the document. Since a node in the tree is a specialization of its parent node (superclass), as well as a generalization of its more specialized children nodes (subclasses), ANASTASIL exploits the inheritance mechanism provided by KRISYS to support the classification of documents. Therefore, only these layout features are stored within the slots of a specific subclass which distinguishes it from its "sibling" classes. Common aspects are stored in the corresponding superclass. Once layout features of a particular node have been determined, they are described in corresponding slots automatically inherited by all children nodes.

The document itself is represented as an instance of one terminal node of the layout hierarchy. Each of the document´s logical objects (e.g. receiver, sender, date, etc.) is represented as a slot expressing a different aggregation relationship between the document and the existing text blocks. In other words, several text blocks have to be aggregated to build the information about one logical object, as illustrated in Figure 6.

3.5 SUPPORT OF DOCUMENT EVALUATION

Based on the determined layout structure, a semantic analysis of the document is performed. Every slot representing a logical object possesses a demon (i.e. an attached procedure) which is automatically triggered if a value (i.e., a corresponding text block) is assigned to it. The purpose of such a demon is to activate a special package of rules which then will interpret the contents of the text blocks and establish its "semantical" meaning.

In KRISYS, demons are represented by special schemas in which code for the respective procedure is stored. The linkage between slots and corresponding demons is done by storing the name of the schema containing the specified procedure in a special aspect. If a slot is accessed, KRISYS checks if there is a demon attached to this slot and activates it by sending a message to the corresponding schema, demanding the evaluation of its procedure /6/. By not storing the demon code directly in the aspect of the slot, as is done in many systems, KRISYS allows many slots to use the same demons without having to introduce redundancy in the representation. This is particularly important if slots representing different positions of the same logical object in the documents request the activation of the same demon.

The method employed by the interpretation rules is mainly based on a full text search with key words in connection with morphological analysis. Such rules take the conceptual meaning of the logical objects into consideration in order to perform a content-based analysis. They considerably restrict the context of search to a plausible set of patterns. Thus, the results of the interpretation process is remarkable. With a relatively simple model, it has been possible to identify important contents in business documents (particulary letters) such as client´s name, what is the document about, its date, etc.

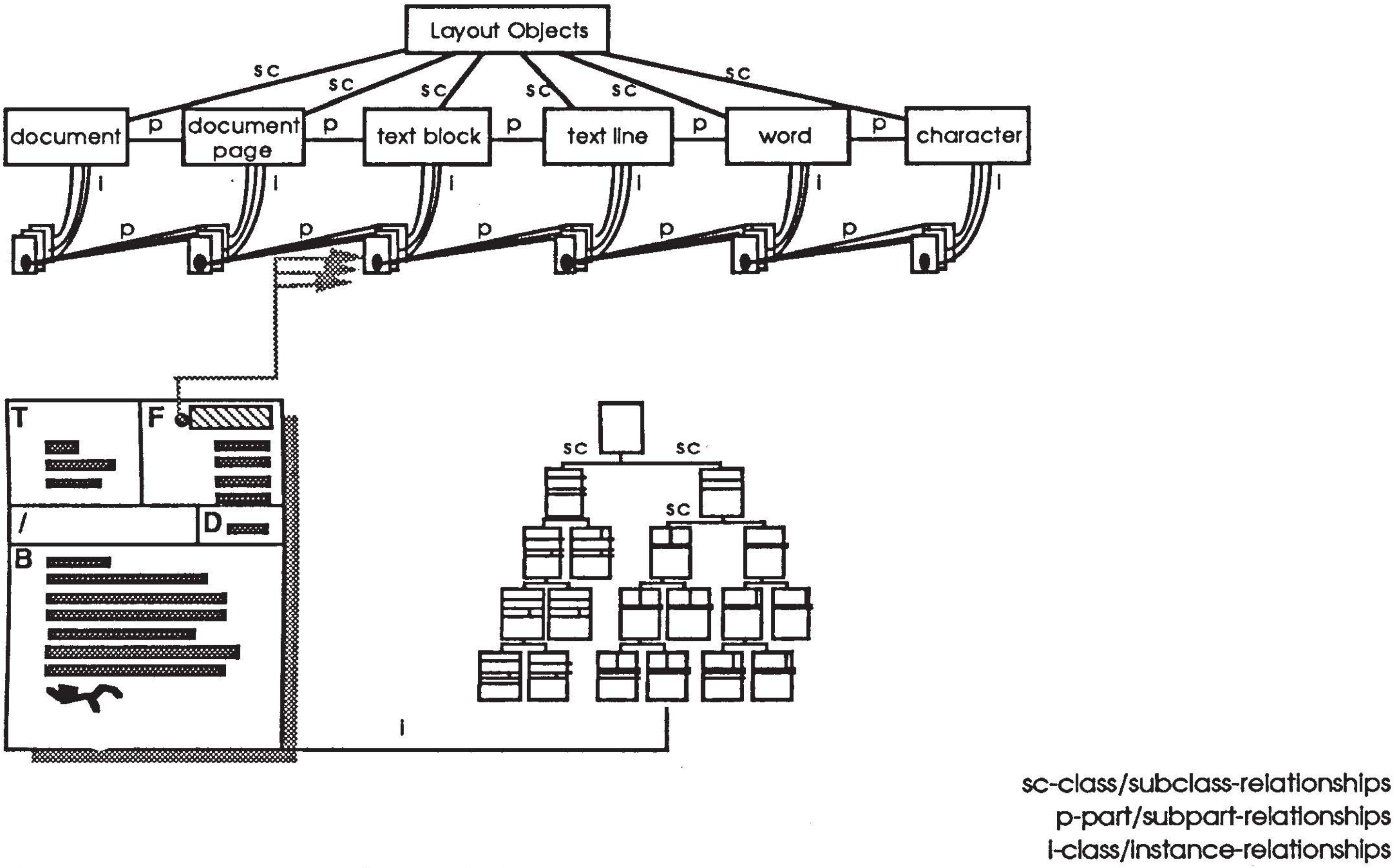
Layout Objects
sc
document
document page
text block
text line
word
character
p
i
T
F
/
D
B
sc-class/subclass-relationships
p-part/subpart-relationships
i-class/instance-relationships

Figure 6: Representation of Logical Objects.

Based on the information extracted from the document, another set of rules is then employed to classify it according to different criteria. In KRISYS, objects fulfilling common conditions are grouped into sets by means of association relationships (/9/, /10/). Thus, classification criteria are represented by hierarchies of abstraction concepts of association. Figure 7 shows a letter (document Y) which was classified by the system as one of the several documents received from client X, as being top-secret, and which has to be processed by Mary until 10th of September.

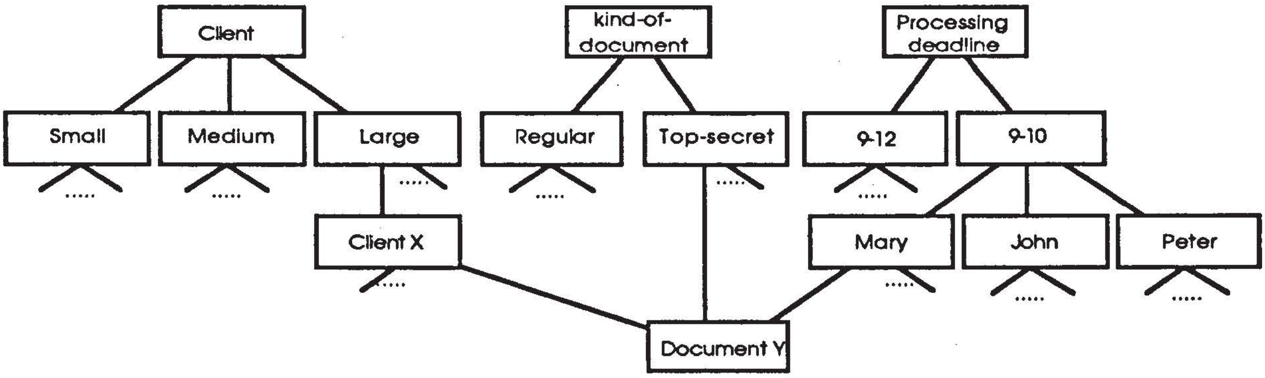

Figure 7: Example of an Association Hierarchy.

Again, the advantage of representing such criteria as association hierarchies is the built-in reasoning facilities supported by KRISYS (/9/, /10/). In the case of association, which sets possess some properties that describe the characteristics of the group of their elements as a whole. As such, the set represented by the documents to be processed by Mary has, for example, properties defining the average of her processing time, the number of documents to be analysed, etc. Since such set properties are typically based on characteristics of each individual element, conclusions about the values of set properties are automatically drawn by KRISYS when a new document is connected to a particular set. Based on this reasoning facility of the association concept, KRISYS can, for example, keep track of documents to be processed by each person, or generate warnings if the amount of work is too large to be done until the desired date.

4 DOCUMENT MANAGEMENT

After having described the analysis of documents as well as its representation using KRISYS, we now want to focus on the management of documents.

The system architecture of KRISYS is divided into three hierarchically ordered layers which control the stepwise abstraction process and the realization of the corresponding tasks within each layer (Figure 8). In the previous section, we have pointed out some features of the knowledge model KOBRA. The application interface of KRISYS is achieved by the query language KOALA /22/, which supports flexible and powerful operations for document retrieval and processing. The goal of the lowest layer is to efficiently cope with storage of the knowledge structures of KOBRA and its supply to the other layers. At this level, most of the issues are related to traditional DB problems applied to large KB, possibly shared by multiple users: storage structures, access techniques, efficiency, integrity features, transaction support, etc. Therefore, this layer is realized by a non-standard database system (NDBS) which seems to be quite advantageous in a KBMS architecture for a number of reasons /23/ and by the Working-Memory System (WMS).

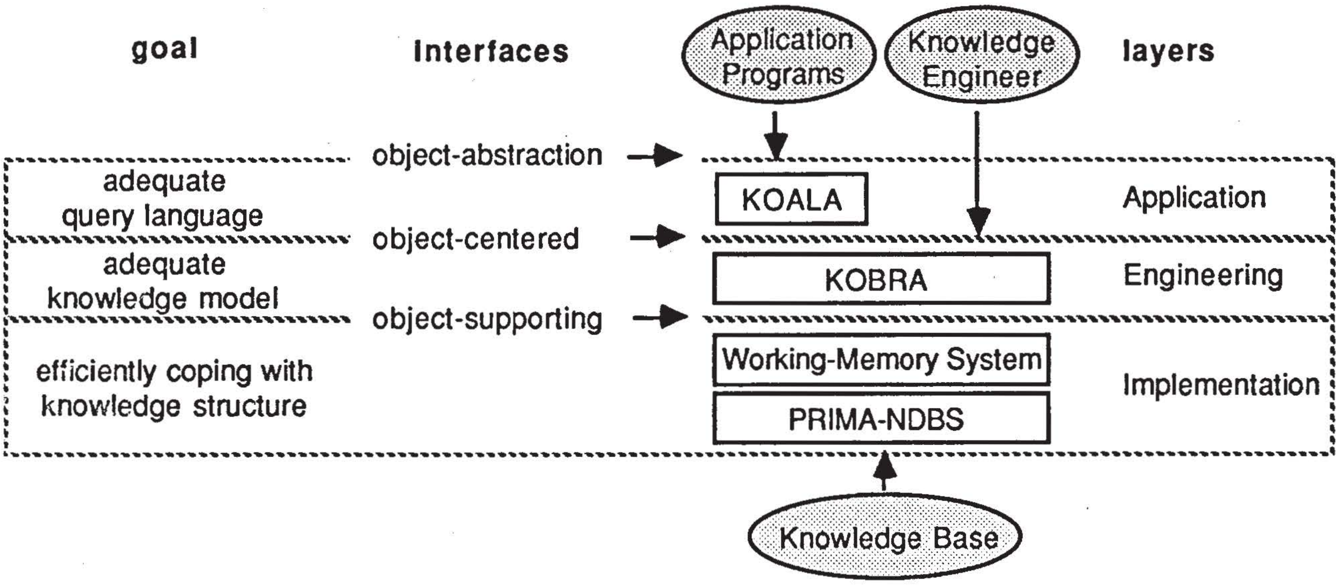

Figure 8: Overall System Architecture of KRISYS

Due to space limitations, it is not possible to discuss the aspects of the NDBS kernel and the mapping of documents . The kernel chosen for KRISYS, named PRIMA (/24/,/25/), offers powerful mechanisms for managing the KB efficiently; among them are storage techniques for a variety of object sizes, flexible representation and access techniques, basic integrity features, locking and recovery mechanisms.

From the point of view of higher levels, there is another very important issue (besides efficient storage management) when working with large KB: to efficiently cope with long execution paths of KB accesses, and time consuming requests to secondary storage. Thus, the tasks of WMS are to considerably reduce the path length and secondly to minimize the number of kernel calls when accessing KB objects. This is achieved by temporarily storing needed objects in a main-memory structure, called working memory, that offers almost direct access at costs comparable to a pointer-like access. The working-memory is a form of application buffer with very fast access to stored objects. WMS supports a processing model aiming at high locality of object references, by drastically reducing the path length when accessing the KB. To reduce the number of NDBS calls, WMS offers the concept of *contexts* being a collection of objects which are needed during a specific processing phase. Based on this concept, WMS fetches and discards contexts if notified by specific control calls from the application during changes of processing phases. These calls are then transformed into set-oriented kernel operations (complex queries) to extract specified objects from the DB or to discard them from the working memory /26/.

5 CONCLUSION

We propose an approach towards an integrated document processing and management system that has the intention to capture essentially freely structured documents, like those typically used in the office domain. The ANASTASIL component is capable to reveal the structure as well as the contents of complex office documents. Moreover, it facilitates the handling of the containing information. Analyzed documents are stored in a management system KRISYS that is connected to several different subsequent services.

The described system is an ideal extension of the human clerk, making his tasks in infor-mation processing easier. The symbolic representation of the analysis results allow an easy transformation in a given international standard, e.g., ODA/ODIF or SGML, and to interchange it via global network.

For further use of the results, one can imagine the following szenario:

In larger organisations incoming electronic and printed information is sorted automatically according to tasks, subjects or receiver. Consequently, it is possible to send it to the corresponding departments via electronic mail. At the same time, an intelligent, multimedia filing system administrates all incoming documents in a document database which constitutes the central part of an office information system. By using a text or publishing system, it will be possible to reproduce an individual information from the document database again on Din_A4 sized screens, or further, to print it in original form, edit or mail it.

REFERENCES

/1/ H. Donner, *Normen schaffen Freiheit im Büro, com* Siemens-Magazine for: Computer & Communications, 4 (1985), pp. 8-13

/2/ W. Horak, *Office Document Architecture and Office Document Interchange Formats: Current Status of International Standardization*, Computer, Oct. 1985, pp 50-60

/3/ H. Balzert, *Entwicklungstendenzen integrierter Bürosysteme: Ökonomie durch neue Qualität & Quantität*, Proceedings ONLINE'85, Europ. Kongreßmesse für technische Kommunikation, Düsseldorf 1985, pp. 2j-1

/4/ M. Schäfer and H. P. Fröschle, *Die Vision vom papierlosen Büro*, Funkschau 19, 1986, p. 45

/5/ A. Dengel and G. Barth, *High Level Document Analysis Guided by Geometric Aspects*, Internat. Journal on Pattern Recognition and AI, Vol. 2, No. 4, Dec. 1988, pp. 641-656

/6/ N. Mattos, *KRISYS - A Multi-layered Prototype Supporting Knowledge Independence;* Proceedings of the Internatl. Comp. Science Conf.-Artificial Intelligence: Theory and Applications, Hong Kong, Dec. 1988, pp. 31-38.

/7/ A. Dengel and G. Barth, *ANASTASIL: Hybrid Knowledge-based System for Document Image Analysis*, Proceedings of the IJCAI'89, Vol. 2, Detroit, MI, Aug. 1989, pp. 1249-1254

/8/ 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, Nov. 1987, pp. 89-100

/9/ N. Mattos and M. Michels, *Modeling with KRISYS - The design Process of DB-Applications Reviewed*, Proceedings of the 8th Interntl. Conf. on Entity-Relationship Approach, Toronto, Canada, Oct. 1989. pp 159-172

/10/ N. Mattos, *Abstraction Concepts - The Basis for Data and Knowledge Modeling*, Proceedings of the 7th Interntl. Conf. on Entity-Relationship Approach, Roma, Italy, Nov. 1988, pp. 331- 350

/11/ G. Nagy and S. Seth, *Hierarchical Representation of Optically Scanned Documents*, Proceedings of the 7th. ICPR, Montreal 1984, p. 347

/12/ E. Schweizer, *Erfassung, Justierung und Segmentierung von Dokumentstrukturen*, Diploma Thesis, CS Department, University of Stuttgart, 1989

/13/ F. Hönes, *Möglichkeiten der visuellen Erkennung von Worten mit Hilfe von geometrischen Eigenschaften der enthaltenen Zusammenhangskomponenten*, B.Sc. Thesis, CS Depart-ment, University of Stuttgart, 1988

/14/ E. Egeli, F. Klein and G. Maderlechner, *Model-Based Instantiation of Symbols from Structurally Image Primitives*, Proceedings of the SPIE'85: Image Processing, Cannes 1985

/15/ P. Kuner and B. Ueberreiter, *Knowledge-Based Pattern Recognition in Disturbed Line Image Using Graph Theory, Optimization, and Predicat Calculus*, Proceedings of the 8th ICPR, Paris 1986, p. 240

/16/ H. Samet, *Region Representation: Quadtrees from Boundary Codes*, Communications of the ACM, Vol. 23, No. 3, March 1980, pp. 163-170

/17/ A. Dengel, A. Luhn and B. Ueberreiter, *Data and Model Representation and Hypohesis Generation in Document Recognition*; Proc. of the 5th. SCIA, Stockholm, June 1987, pp. 57-64

/18/ A. Barr and E. A. Feigenbaum, *The Handbook of Artificial Intelligence*, Vol. 1, The William Kaufmann Inc., Los Altos 1981

/19/ J.M. Smith and P.C.P. Smith, *Database Abstractions: Aggregation and Generalization*, ACM Transcations on Database Systems, Vol. 2, No. 2, June 1977, pp. 105-133

/20/ A. Rosenthal et al, *Query Facilities for Part Hierarchies: Graph Traversal, Spatial Data and Knowledge-based Detail Supression*, Research Report, CCA, Cambridge, MA, 1987

/21/ S. N. Srihari and G. W. Zack, *Document Image Analysis*, Proceedings of the 8th ICPR, Paris, 1986, p. 434

/22/ N. Mattos, S. Deßloch, *KOALA - An Interface to Knowledge Base Management Systems*, internal report, University of Kaiserslautern, 1989, submitted for publication

/23/ N. Mattos, *An Approach to Knowledge Base Management - requirements, knowledge representation, and design issues -*, Doctoral Thesis, University of Kaiserslautern, Computer-Science Department, Kaiserslautern, 1989.

/24/ T. Härder (ed.), *The PRIMA Project: Design and Implementation of a Non-Standard Database System*, SFB 124 Research Report No. 26/88, University of Kaiserslautern, Kaiserslautern, 1988

/25/ T. Härder, K. Meyer-Wegener, B. Mitschang, A. Sikeler, *PRIMA - A DBMS Prototype Supporting Engineering Applications*, in: Proceedings of the 13th Conf. Brighton, UK, 1987, pp. 433-442

/26/ F.-J. Leick, N. Mattos, *A Framework for an Efficient DB-Support of Knowledge-Based Systems*, in: Proc.eedings of the 4th Brazilian Symposium on Data Bases, Campinas, Brazil, April 1989