ODA-based modeling for document analysis[1]

Rainer Bleisinger*, Rainer Hoch†, Andreas Dengel ††

German Research Center for Artificial Intelligence, D - 6750 Kaiserslautern, Germany
* bleising@dfki.uni-kl.de
† hoch@dfki.uni-kl.de
†† dengel@dfki.uni-kl.de

ABSTRACT: This article proposes the document model of a hybrid knowledge-based document analysis system for business letters. The model combines requirements of object-oriented representation of both, documents as well as knowledge necessary for analysis tasks, and is based on the ODA platform. Model-driven document analysis increases the flexibility of a system because several analysis specialists can be used in co-operation to assist each other and to improve the results of analysis. The inherent modularity of the system allows for a reuse of knowledge sources and integral constituents of the architecture in other document classes such as forms or cheques.

KEYWORDS: Document Representation, Model-Driven Analysis, ODA, SGML, EDIFACT.

CONTENTS:

1 Introduction and system overview..... 2
2 Representation standards for structured documents..... 3
3 Document architecture model of ODA..... 5
4 Document model for analysis in P_{ODA}..... 7
4.1 Object description conventions..... 7
4.2 Generic model layer..... 8
4.3 Knowledge portions for analysis..... 10
5 Model-based document image analysis..... 11
6 Conclusion..... 13
References..... 14

[1] This work has been supported by the Germany Ministry for Research and Technology BMFT under contract ITW 9003 0.

1 Introduction and system overview

In the last years many people have predicted the *paperless office* where paper documents would be obsolete. Despite all efforts in office automation, offices produce more paper than ever before. One reason for this observation is the weakness of commercial office information systems in supporting an international standard representation to facilitate the exchange of documents between heterogeneous systems. In consequence, paper documents will further remain the most popular and dominating medium for exchanging information. By this way, tools for getting existing paper documents into an equivalent symbolic representation on the computer become increasingly important.

In this paper, the underlying document model of our image analysis system P_{ODA} (**P**aper **I**nterface to **ODA)** is presented. P_{ODA} tackles the problems described above and is one result of our current research within the ALV project. ALV is the German acronym for *Automatic Reading and Understanding*. The intention of P_{ODA} is to bridge the gap between paper and computer. P_{ODA} is a model-driven system based on the ODA platform. To support the analysis process, various knowledge sources such as typesetting knowledge, spatial knowledge, geometric and lexical knowledge, as well as syntactic knowledge are involved. Figure 1 gives an impression of our system and the underlying document model.

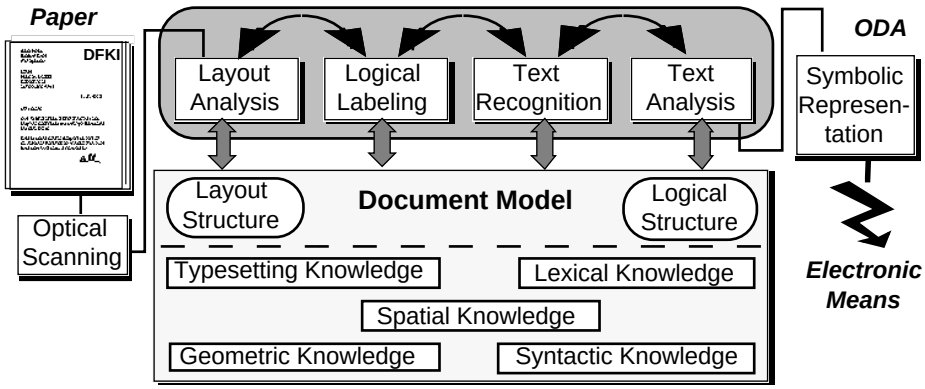

Figure 1: System Architecture of P_{ODA}.

The entire system includes several interwoven phases of analysis: *Layout extraction* comprises all low-level processing routines like image capturing, skew angle adjustment and segmentation to investigate the layout structure of the document. *Logical labeling* is used to “hypothesize & test” the logical meaning of layout objects [Dengel91a]. *Text recognition* explores the captured text of logical objects. By this way, word hypotheses are generated, verified and

redundant word candidates eliminated [Hönes90]. Finally, a partial *text analysis* of selected objects (subject, body) is initiated to classify the document (invoice, order, offer, etc.) and furthermore, to retrieve conceptual information. As output, P_{ODA} produces a symbolic representation of a business letter that conforms to ODA.

All analysis tasks are driven by the object-oriented document model which provides facilities to associate additional knowledge with document objects. Knowledge sources contain information used from several algorithms for analyzing particular document objects. Hence, an object-based incremental analysis strategy can be performed.

To put it in a nutshell, the design of P_{ODA} reveals many advantages. Our model-based approach fundamentally separates between the representation of a document associated with knowledge and the analysis algorithms. For this reason the system is neither hard-coded nor restricted to business letters. It is modular, extensible and our analysis methods, we call them specialists, can be reused in another application domain, e.g. bank cheques. Existing systems for document image analysis are more rigid and their design is tailored to particular document types or special parts of a document, for example, forms or address readers [Srihari86, Schürmann92]. Therefore, changing to another document type or including new analysis algorithms (e.g. segmentation routines) often lead to a complete reimplementation of the analysis system from scratch.

Having given a short overview of our overall system architecture and processing tasks, Chapter 2 considers existing standards for representing and exchanging structured documents as far as they have had an influence on our system design. While Chapter 3 explains the document architecture model of ODA and introduces basic concepts, Chapter 4 discusses extensions with respect to the requirements of document image analysis. In Chapter 5 the distinct analysis phases of our system are exposed in short according to the architecture model. Chapter 6 concludes this article and points to our current research topics.

2 Representation standards for structured documents

In the last years the exchange and processing of electronic documents have taken a central role in the field of office information systems. Primarily, two international standards in this domain have been developed and published in the meantime, namely the *Office Document Architecture and Office Document Interchange Format* (ODA/ODIF)[2] [ISO8613] and the *Standard Generalized Markup Language* (SGML) [ISO8879, Bryan89].

[2] ODIF is a convention how ODA structures are mapped into a corresponding data stream for electronic exchange. Because this article is concerned with aspects of document modeling, we neglect ODIF in the following.

SGML as well as ODA provide formalisms for document structure representation. In both standards, the aspect of logical document organization is central. It is used to divide the contents of a document (text, graphics, images) into logical entities that are associated with an author‘s intellectual meaning. For example, a business letter may be divided into logical objects such as sender, recipient, subject, and body. In SGML, this formalism is described through so-called document type definitions (DTD‘s), while in ODA it is designated by a generic logical structure of a document.

Another possibility to consider the organization of a document is by its layout. The layout structure is determined by hierarchically nested rectangular blocks. These may be entire pages, graphic frames, image frames, and text frames, while the latter ones may be further subdivided into lines, words and characters. Both structures, layout as well as logical, are strictly hierarchical and express two different but complementary views to the contents of a document.

In contrast to ODA, however, SGML does not support a description of the layout structure of a document for reasons of simplicity and universality. SGML is designed for the representation of any kind of structured text. For instance, SGML is typically used in a publishing environment, where an author logically marks up a document‘s components and the publisher performs all future processing such as copy-editing, proof-reading and production, including the final distribution. In this closed application area, standardized layout characteristics are less important. In contrast, the scope of ODA covers office documents (business letters, reports, forms) in particular. An office environment requires that documents may be sent to arbitrary recipients allowing for an automatic reproduction and interactive modifications of the document at the receiving end.

To complete the discussion about standards, a third and more commercial standard, named *EDIFACT* (Electronic Data Interchange for Administration Commerce and Transport) [ISO9735, Frank91] should be considered here. EDIFACT specifies the structure and formal semantics of a data stream for exchanging fixed and predefined types of business letters, called *message types*, and enables further processing of the message content. Each message type description includes optional or mandatory segments (records), data element groups, or data elements respectively. Such elements represent logical components of a document; any layout information is taken away. So far only two message types for business letters, invoice and order, have been standardized [Frank91].

Many in-house styles have been developed (e.g. Interscript, Scribe, DCA), but a discussion is beyond the scope of this paper (cf. [Joloboff89, Quint89]).

Since low-level routines of document image analysis are mainly oriented on layout aspects (e.g. block segmentation), we base the document model of our analysis system on the ODA platform, but we enhance the standard to the requirements of document image analysis as needed. Moreover, logical elements

identified and captured by EDIFACT message types have a strong influence on the design of our logical model of business letters.

In the next chapter a short introduction to the concepts of ODA is given emphasizing the crucial points with respect to the modeling character of our paper.

3 Document architecture model of ODA

One of the distinguishing features of the document architecture model of ODA [ISO8613] is a strict separation between the contents of a document and its structural representation. Consequently, the notion of *structure* is a key concept of ODA (see also [Brown89]).

There are two distinct, but complementary structures of a document, the *layout structure* and the *logical structure*. Both structures are represented by a tree whose nodes correspond to document components (objects). The leaves of each tree are associated with specific *content portions* of a document. An object that is not subdivided into smaller objects, i.e. a leaf, is called a *basic object*. All other objects are called *composite objects*.

ODA defines five types of layout objects in the document architecture:

- block: a basic layout object corresponding to a rectangular area on the presentation medium containing a portion of the document content;
- frame: a composite layout object corresponding to a rectangular area containing one or more frames or blocks;
- page: a basic or composite layout object corresponding to a rectangular area and containing one or more frames or blocks respectively;
- page set: a set of one or more page sets or pages;
- document layout root: the highest level object in the layout hierarchy.

Because logical objects of a document are strictly application-dependent (e.g. sender, recipient, body, ...), the classification in ODA is less concrete comprising the types basic, composite and document logical root.

In a document, both, the logical objects and layout objects can often be classified into groups of similar objects, the so-called *object classes*. An object class is comparable to the well-known class concept in object-oriented programming paradigm. Such a class can best be thought of as a specification of the set of characteristics that is common to its members and associated methods. In this way, ODA provides a hierarchical and object-oriented document model.

Using such object classes, the logical structure as well as the layout structure of similar documents can be modeled by a set of logical object classes. This is called the *generic structure concept* of ODA. Generic structures provide a means for defining document classes or “styles” that define the types and combinations of objects allowed. The structures that are particular to a concrete document instance are called *specific logical structure* and *specific layout structure*. Figure 2 illustrates the specific structures of a business letter.

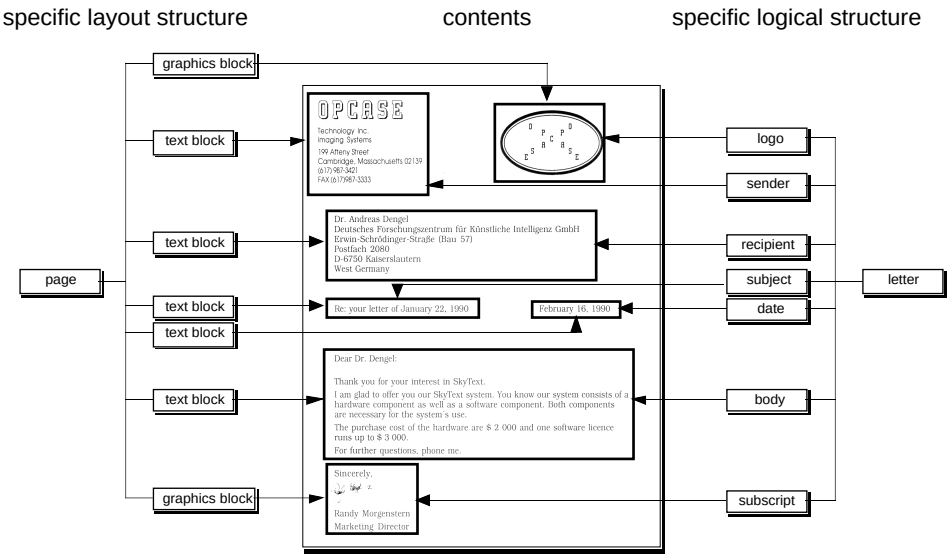

Figure 2: A specific business letter represented in ODA (simplified).

All objects of a document are supplied with specific characteristics, known as *attributes* in ODA. The set of attributes associated with document components can be categorized into layout attributes (e.g. “position”, “dimension”), logical attributes (e.g. “layout style”) and shared attributes (e.g. “object identifier”).

One important attribute should be considered here in more detail, namely the *generator-for-subordinates*. This attribute defines how an object of a document is composed of subordinate objects (optional=OPT, required=REQ, repetitive =REP), e.g. a text line may be built up from several words, or the recipient is built up from name, street, city, and country. In addition, this attribute specifies an ordering among these subordinates (sequential=SEQ, aggregate=AGG).

Only basic objects (logical as well as layout) can be associated with content portions of a real document. These content portions may have a more detailed internal structure depending on the type of content. The rules for processing different kinds of document contents are known as *content architectures*. Currently, ODA defines three types of architectures: character content (ASCII code), raster graphics content (images) and geometric graphics content (graphics primitives). Content portions related to basic objects belong to exactly one content architecture.

After this short introduction of the ODA architecture model, the next chapter motivates how the powerful concepts of ODA can advantageously be used and transferred to a model-driven document analysis system.

4 Document model for analysis in P_{ODA}

For transforming a document image into a corresponding symbolic representation, different knowledge layers are used. The final representation should be oriented on existing standards for reasons of exchange and transmission. Therefore, the underlying model is used for representation of both, the analysis results as well as the knowledge supporting analysis tasks.

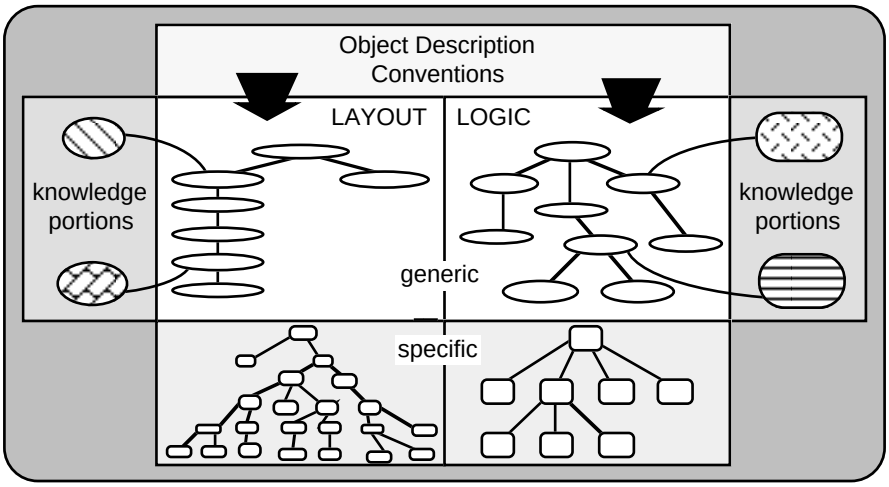

Figure 3**:** Overview of ALV document model.

In this way, we are following the object-oriented formalism provided by ODA. Necessary extensions and modifications mainly concern knowledge to support analysis (cf. Figure 3). Like ODA, our document model utilizes a layout view and a logical view but consists of three layers, namely object definition conventions, generic layer and specific layer.

During analysis the document model is used for directing the generation of a symbolic representation of an actual document. This one is composed of the specific layout and the specific logical structures that are related by content portions of type text and image.

Within the generic layer of the model, abstract descriptions for document types are specified. Exemplarily, the type “business letter“ is defined in P_{ODA}.

To guarantee flexible definition of new or adaptation of existing document types, we have developed object definition conventions. To this end, a document model editor can be built for a user support in model specification.

4.1 Object description conventions

In Chapter 3 the ODA representation formalism has been discussed. The object description conventions in P_{ODA} provide the same mechanisms for defin-

ing and relating object classes. In addition, possibilities for attaching knowledge only required for the analysis are provided.

With respect to the views, two intersecting sets of attributes are obtained, for specifying layout aspects as well as logical aspects. The attributes are chosen according to the requirements of once, document image analysis and second, document representation in an ODA conforming manner. Consequently, a further classification is useful.

The closed set of *ODA standard attributes* contains mandatory attributes, e.g. object-type (layout, logic), as well as optional attributes, like generator-for-subordinates (both). Details are explained in the ISO standard 8613. Note, all mandatory attributes of ODA are also mandatory in our document model.

The open set of P_{ODA} *extensions* consists of attributes defined in extension to ODA. Actually, these are attributes for storing intermediate results of the analysis steps. In this way, all object related intermediate results are available for the whole analysis task. For example, hypotheses about relations from logical to layout objects are explicitly stored in logical objects. Additionally, attributes for attaching knowledge portions assisting the analysis process are incorporated. Their values are explained in Section 4.3 in more detail.

All attributes of a class are typified according to their different usage. *Class attributes* describe features of a class itself, e.g. object-class-identifier. They have to be filled while class definition. *Instance attributes* contain information needed for the individual instances of classes, e.g. object-identifier. They are filled during or after instantiation. The third group comprises attributes for information similar for all instances, e.g. object-type or content-type, and is called *class-instance attributes*.

When defining a class, any constraints for a certain attribute combination have to be considered. Primarily, attributes only for logical objects or only for layout objects can be used. Moreover, attributes are mutually exclusive or strictly coupled (generator-for-subordinates—subordinates). Additionally, the specified value of an attribute can force or forbid other attributes (object-type basic—generator-for-subordinates) in the same class.

The introduced description conventions enables an object-oriented definition of document types by specifying the involved object classes. Especially, requirements for document analysis are taken into account.

4.2 Generic model layer

For establishing a model for a particular document type (e.g. business letter, scientific paper), object class descriptions have to be defined on the basis of the object description conventions. The *generic structure* of a document type is specified by appropriate values of the generator-for-subordinates attribute. As mentioned in Chapter 3, this attribute provides a framework for a structural

combination and aggregation of object classes. Both, the layout and the logical view are separately considered in the following.

For the document type business letter, the layout object classes page, non-text-block, text-block, line (cf. Figure 4), and word are defined. With a close look to the specified values of the generator-for-subordinates attribute, the following abstract generic layout structure for business letters can be attained (see also Figure 4).

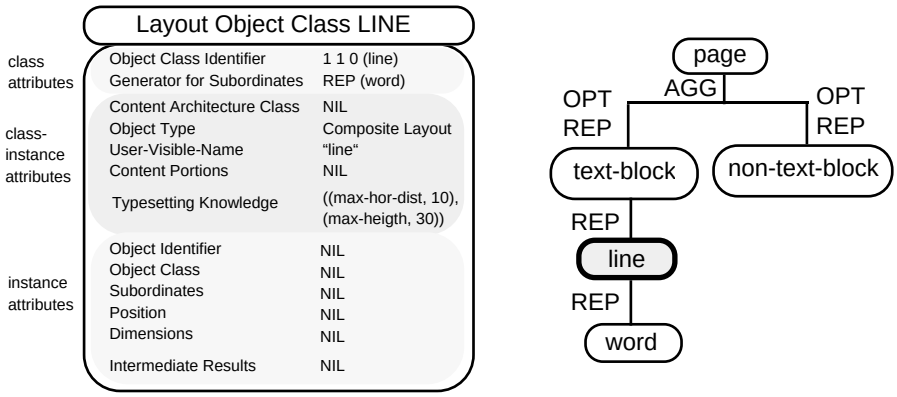

Figure 4: Generic layout structure for document class business letter.

To complete the model for business letters, logical object classes are specified for representing the generic logical structure (cf. Figure 5). First, a grouping in three major parts is done: the *letter thematic part* contains the subject and the letter body; in the *sender specific part* objects like sender, company logo, and company data are incorporated; the *procedure relevant part* is composed of references like "your sign", "our sign" or the date, enclosures and recipient. Some of these, such as letter body or recipient, may further be subdivided.

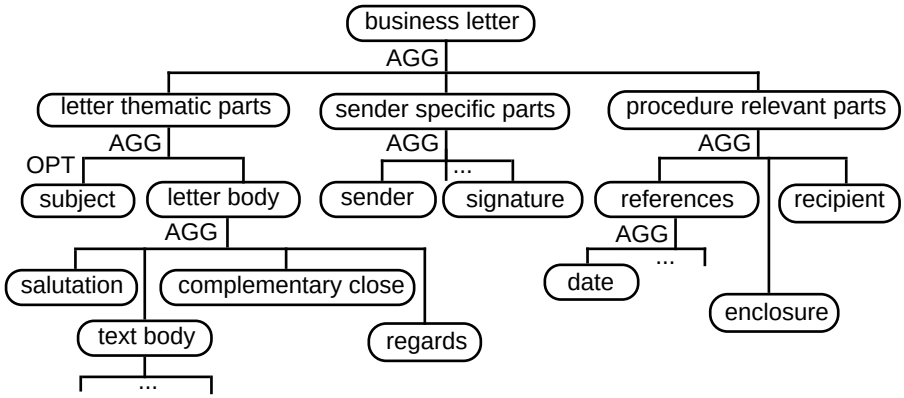

Figure 5: Generic logical structure for document type business letter (partial).

The resulting generic structures serve as a basis for a model-driven analysis and causes an ODA like document representation.

4.3 Knowledge portions for analysis

For assisting document analysis, the P_{ODA} model provides attributes to associate knowledge portions with object classes of the generic layer. Knowledge portions contain information which is used from several specialists for analyzing particular document objects. In this way, an object-based incremental analysis strategy can be performed. The knowledge types actually used are:

•*Typesetting Knowledge:* The attribute *typesetting knowledge* comprises features such as character, word and line spacing, block distances, font heights etc. This attribute is only attached to layout object classes and the specified knowledge portions are used as parameter values for instantiation of the specific layout structure.

•*Spatial Knowledge:* For a specific document type (e.g. business letter), presentation conventions can be formalized by expressing spatial dependencies among logical objects with respect to the presentation medium document. Because the occurrence and position of one logical object restricts the position of all others (e.g. the existence of a footer restricts the length of text on a page), in general different alternatives for document presentation exist. For expressing dependencies among logical object arrangements, a so-called *spatial dependency tree* (SDT)—also designated as geometric tree [Dengel89]—is introduced.

The SDT is a collection of different presentation alternatives of a specific document type describing them on different abstraction levels. Common spatial arrangements of logical objects are specified in spatial classes. Refined arrangements result in appropriate but concurrent subclasses. The SDT is attached to the logical root of a document type model by the attribute *spatial knowledge.* It is the key knowledge to relate layout objects of a document to logical objects which capture their semantics.

•*Geometric Knowledge:* The SDT is used to generate working hypotheses about logical object location. This is done while matching a document at hand with the spatial classes in the tree. For hypotheses verification, it is necessary to provide additional local descriptions of logical objects. Therefore, geometric knowledge, i.e. position, extensions, number of lines, etc., is collected in sets of rules which are associated by the *geometric knowledge* attribute. In the action parts of the rules, measures of belief or disbelief are stored, corresponding to probability values that are obtained by evaluating a few hundred business letters under geometric aspects [Dengel91a].

•*Lexical Knowledge:* Concerning text in logical objects, typical words may be determined, e.g. the names of possible recipients. Such groups of words are stored in so-called logical vocabularies. They are attached to logical objects by the *lexical knowledge* attribute which is mainly accessed for improving text

recognition results. Additionally, typical phrases are collected according to logical objects. The most obviously ones in a letter are salutation (Dear Sir/Madam, Dear Mr.) and regards (sincerely yours, yours truly). But also standard phrases like “according to your offer“, “we refer to“, etc. are useful.

•*Syntactic Knowledge:* The syntactic knowledge is concerned with word order of text in logical objects. For instance, text within the logical object recipient can be described by a context-free grammar and attached by the attribute *syntactic knowledge* [Hönes 90]. This knowledge assists text recognition and improves hypotheses verification of logical labeling. Moreover, it enables a refinement of the specific logical structure, e.g. the refinement of recipient into street, city and name, while the latter one may further be split in first name and last name.

These knowledge sources in combination with the entire architecture model are used to initiate a model-based document image analysis and understanding.

5 Model-based document image analysis

Because this paper focuses on the aspects of document modeling, the following section only gives a brief overview of processing document images. Details can be found in [Dengel91b].

One necessary and early step of analysis, however not model-driven, is the *detection of skew* within the scanned document image. After *skew correction*, we are applying a derivation [Dengel91a] of the RLSA approach [Wang89] for model-driven *segmentation.* Our segmentation routine follows a top-down strategy to establish the specific layout structure of a document by considering typesetting knowledge (e.g. pixel distances of the original distance) and the definition of the generic layout structure. Beside the particular structural representation of layout objects, specific slots are associated with each layout object capturing data about their position as well as dimensions.

In a next step, spatial knowledge allows for model-driven *logical labeling* of layout objects [Dengel88]. Thereby, the labels of the spatial dependency tree indicate hypotheses about the meaning of layout objects. To verify these hypotheses, features of layout objects are compared to geometric knowledge of the related logical object. For example, layout objects may be recognized as belonging to the recipient of a letter, because they fulfil several characteristics, in especially, they are suited at a certain position, being left justified, being composed of lines capturing five words as a maximum, and matching a certain line number criteria (see also [Dengel91a]).

According to this labeling, logical objects are generated with respect to the generic logical structure of the document model, providing a top-level logical view to a given document, which can further be refined.

Up to now, no text recognition has been applied. The basic layout objects representing words are related to content portions capturing corresponding word

images. These images are the input for a subsequent *text recognition* for which we are using a commercial OCR system. Because each logical object provides a focus on a logical context, specific logical vocabularies can be accessed for improving text recognition [Hönes90].

For several objects of the logical structure, i.e. those for which syntactic conventions are typical (e.g. recipient, sender, date), an additional *text analysis* phase ascertains refinements in logical document structuring. For that purpose, a grammar is used as input for a parser that makes predictions about the meaning of particular words according to their position and context. For verification, restricted vocabularies (i.e. street/month names, names of persons, zip codes) are accessed [Hönes90]. In this way, appropriate logical objects are generated to expand the specific logical structure of the document.

Figure 6 schemes an example of symbolic document representation after having finished all phases of document analysis.

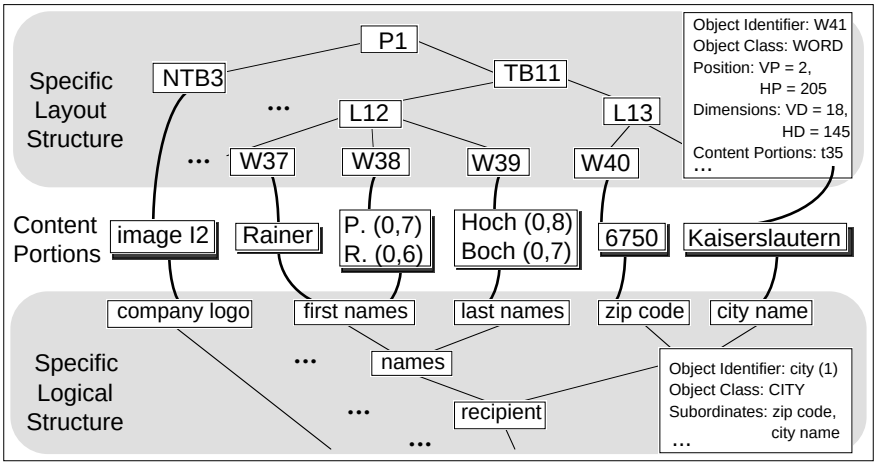

Figure 6: Part of our P_{ODA} model including text recognition results.

6 Conclusion

This article has presented a document model developed for the hybrid knowledge-based analysis system P_{ODA} which is capable to transform existing paper documents, actually business letters, into an equivalent symbolic representation. To support and to simplify the exchange and processing of electronic documents resulting from document analysis, we focus our interest on the international standard ODA for office documents. Consequently, our system is based on a hierarchical document architecture that has been extended by corresponding knowledge sources. In this sense the model allows for the reusability of components and easy adaptation to other document types as well as

domains. Because document image analysis is also driven by the model, the whole system profits from an increased flexibility whereby analysis specialists can be reused or even replaced.

Our future work will also concentrate on a expectation-driven partial text analysis of logical objects, especially subject and letter body, to enhance the syntactic and semantic knowledge of business letters. In a first step, we apply AI techniques and classical results of information retrieval to create a conceptual structure of a business letter (e.g. offer, invoice) to provide for a restricted context of content-based analysis.

Our system has been implemented for the analysis of single-paged business letters in German and currently runs on a Macintosh IIfx computer connected to an Apple Scanner. All implementations have been done in Common Lisp, except the scanner interface which is partially written in MPW Pascal. An enhanced Unix implementation on Sun SPARCstation systems will soon be available.

References

[Brown89] H. Brown. Standards for Structured Documents. *The Computer Journal*, vol. 32, no. 6, 1989, pp. 505–514.

[Bryan89] M. Bryan. *SGML: An Author‘s Guide to the Standard Generalized Markup Language*. Addison-Wesley, England; Reading, MA et al., 1988.

[Dengel89] A. Dengel, 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.

[Dengel91a] A. Dengel. ANASTASIL: A System for Low-Level and High-Level Geometric Analysis of Printed Documents. In H. Baird, H. Bunke, K. Yamamoto (eds.), *Structured Document Image Analysis*, Springer Verlag, Heidelberg 1991.

[Dengel91b] A. Dengel, R. Bleisinger, R. Hoch, F. Fein, F. Hönes. From Paper to an Office Document Standard Representation. Submitted to *IEEE Computer*, Special Issue on Document Image Analysis Systems (to appear).

[Frank91] U. Frank. Anwendungsnahe Standards der Datenverarbeitung: Anforderungen und Potentiale. *Wirtschaftsinformatik*, vol. 2, no. 2, April 1991, pp. 100-111 (in German).

[Hönes90] F. Hönes, R. Bleisinger and A. Dengel. Intelligent Word-based Text Recognition. In M. J. Chen (ed.), *High-Speed Inspection, Architectures, Barcoding and Character Recognition*, Proceedings OE-90, vol. 1384, Boston, MA, Nov. 1990, pp. 305-316.

[ISO8613] ISO 8613, Text and Office Systems, ODA, 1988.

[ISO8879] ISO 8879, Text and Office Systems, SGML, 1986.

[ISO9735] ISO 9735, Trade Data Interchange, EDIFACT, 1988.

[Joloboff89] V. Joloboff. Document Represenation: Concepts and standards. In J. André, R. Furuta, V. Quint (eds.), *Structured Documents*, Cambridge University Press, 1989, pp. 75-105.

[Quint89] V. Quint. Systems for the Manipulation of Structured Documents. In J. André, R. Furuta, V. Quint (eds.), *Structured Documents*, Cambridge University Press, 1989, pp. 39-74.

[Srihari86] S. N. Srihari, C.-H. Wang, P. W. Palumbo, J. Hull. Recognizing Address Blocks on Mail Pieces: Specialized Tools and Problem-Solving Architecture, *AI Magazine*, vol. 8 No. 4, winter 1987.

[Schürmann92] J. Schürmann, N. Bartneck, T. Bayer, J. Franke, E. Mandler, T. Oberländer. Document Analysis: From Pixels to Contents, *IEEE Proceedings, Special Issue on OCR and Document Analysis*, scheduled for May 92.

[Wang89] D. Wang, S.N. Srihari. Classification of Newspaper Image Blocks Using Texture Analysis. *Computer Vision, Graphics and Image Processing*, vol. 47, 1989, pp. 327-352.