Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Technical Memo
TM-94-01

Text Skimming as a Part in Paper Document Understanding

Rainer Bleisinger, Klaus-Peter Gores

March 1994

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
67608 Kaiserslautern, FRG
Tel.: (+49 631) 205-3211/13
Fax: (+49 631) 205-3210

Stuhlsatzenhausweg 3
66123 Saarbrücken, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Deutsches Forschungszentrum für Künstliche Intelligenz

The German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) with sites in Kaiserslautern and Saarbrücken is a non-profit organization which was founded in 1988. The shareholder companies are Atlas Elektronik, Daimler-Benz, Fraunhofer Gesellschaft, GMD, IBM, Insiders, Mannesmann-Kienzle, SEMA Group, and Siemens. Research projects conducted at the DFKI are funded by the German Ministry for Research and Technology, by the shareholder companies, or by other industrial contracts.

The DFKI conducts application-oriented basic research in the field of artificial intelligence and other related subfields of computer science. The overall goal is to construct systems with technical knowledge and common sense which - by using AI methods - implement a problem solution for a selected application area. Currently, there are the following research areas at the DFKI:

- Intelligent Engineering Systems
- Intelligent User Interfaces
- Computer Linguistics
- Programming Systems
- Deduction and Multiagent Systems
- Document Analysis and Office Automation.

The DFKI strives at making its research results available to the scientific community. There exist many contacts to domestic and foreign research institutions, both in academy and industry. The DFKI hosts technology transfer workshops for shareholders and other interested groups in order to inform about the current state of research.

From its beginning, the DFKI has provided an attractive working environment for AI researchers from Germany and from all over the world. The goal is to have a staff of about 100 researchers at the end of the building-up phase.

Friedrich J. Wendl
Director

Text Skimming as a Part in Paper Document Understanding

Rainer Bleisinger, Klaus-Peter Gores

DFKI-TM-94-01

This work has been supported by a grant from The Federal Ministry for Research and Technology (FKZ ITW-9003 0).

ISSN 0946-0071

Text Skimming as a Part in Paper Document Understanding

Rainer Bleisinger, Klaus-Peter Gores

German Research Center for Artificial Intelligence (DFKI)
P.O. Box 2080
D-67608 Kaiserslautern, Germany
Phone: (+49) 631-205-3216, Fax: (+49) 631-205-3210
e-mail: bleising@dfki.uni-kl.de

Abstract: In our document understanding project ALV we analyse incoming paper mail in the domain of single-sided German business letters. These letters are scanned and after several analysis steps the text is recognized. The result may contain gaps, word alternatives, and even illegal words. The subject of this paper is the subsequent phase which concerns the extraction of important information predefined in our "message type model". An expectation driven partial text skimming analysis is proposed focussing on the kernel module, the so-called "predictor".
In contrast to traditional text skimming the following aspects are important in our approach. Basically, the input data are fragmentary texts. Rather than having one text analysis module ("substantiator") only, our predictor controls a set of different and partially alternative substantiators.
With respect to the usually proposed three working phases of a predictor — start, discrimination, and instantiation — the following differences are remarkable. The starting problem of text skimming is solved by applying specialized substantiators for classifying a business letter into message types. In order to select appropriate expectations within the message type hypotheses a twofold discrimination is performed. A coarse discrimination reduces the number of message type alternatives, and a fine discrimination chooses one expectation within one or a few previously selected message types. According to the expectation selected substantiators are activated. Several rules are applied both for the verification of the substantiator results and for error recovery if the results are insufficient.

Keywords: natural language analysis, expectation-driven analysis, text skimming, document analysis, document understanding

Contents:

1. Introduction

Document understanding is a process of interpretation whereby the extraction of information of distinct levels is performed. Two main levels can be distinguished. Some analysis phases concentrate on the document image data (pixels) and extract layout as well as structural information, e.g., segmentation, text recognition, and image-based logical labeling. Other phases focus on complex graphical objects and the recognized text of a document and extract contentual information, e.g., text-based logical labeling, analysis of text and graphic.

In many existing document analysis systems the primary goal is the recognition of text within scanned paper documents (e.g., [Baird et al 86], [Hull et al 92]), or sometimes the identification of logical portions of the document (e.g., [Nagy et al 92], [Story et al 92]), such as the recipient of a letter or the title of a paper. Moreover, applications in mail distribution systems also require an understanding of addresses (e.g., [Palumbo & Srihari 86], [Prussak & Hull 91]).

Our document understanding project ALV (German acronym for "Automatic Reading and Understanding") goes far beyond this level of information and builds a bridge between recognition and comprehension of text ([Dengel et al 92a], [Dengel et al 92b]). ALV takes as input single-sided paper documents of machine-printed business letters written in German. The overall goal of document understanding in ALV is the extraction of important information. One information is the class of a business letter in terms of message types, such as "order" or "complaint". Furtheron, important informations are the sender and the recipient, and in case of an order, the articles ordered, their prices and quantities. Based on such information we do a notification via e-mail. Other potentially applications are intelligent filing and retrieval or support of workflow management.

Because much of the important information is transmitted as text in a business letter, approaches for text understanding have to be integrated. In the research field of document understanding the use of existing natural language processing approaches for "semantic" text analysis is problematic due to the following two facts. First, today there exists no complete semantic analysis system which can handle real text as found in business letters. Existing systems mostly operate only on small text parts, like a few sentences, or are often restricted to a very small application context. Recently, this problem is attacked by enhancing such systems with shallow analysis features resulting in a combined bottom-up and top-down approach ([Rau & Jacobs 88], [Lehnert et al 91], [Hobbs et al 92], [Hu et al 93]). Second, existing text analysis systems usually require complete and correct text input (one exception: [Jackson et al 91]). But these approaches do not have to take care of problems which are caused by text recognition, such as gaps, word alternatives or even illegal words. A robust system must contend with such input or at least tolerate it.

Because of the explained problems of a complete understanding we apply a partial text analysis which is called "text skimming". Ideas on expectation driven text skimming have already been proposed by [DeJong 82], [Lebowitz 85], [Mauldin 91], [Hayes 92]. But the respective systems extract important information from complete and correct electronic documents, only. However, this shallow analysis of the text seems as a reasonable approach tackling the problems caused by poor text recognition results.

The principle of text skimming within ALV is introduced in the next chapter. Afterwards the central control component, called "predictor", is described whereby its working phases are explained in detail. In the following the usage of the analysis results is mentioned. Finally, a conclusion shows the main differences between our text skimming approach and others.

2. Text skimming in ALV

Our partial text analysis for the extraction of important information operates on text recognition results, character and word hypotheses, which may contain alternatives, gaps, and illegal words. Some recognition errors are tackled in a preprocessing step by text-based segmentation. Thus hypothesized words are resegmented, e.g., “bestellen:” into “bestellen” and “:”, or “Haus-” and “boot” into “Hausboot”. This results in additional word hypotheses.

The main part of partial text analysis is done incrementally starting with previously instantiated logical objects like “recipient” or “letter body”. Since the text in logical objects varies from well-structured text (e.g. the addresses) to free natural language text (e.g. in the letter body) specific requirements arise and necessitate several analysis approaches (substantiators). The skimming approach in ALV serves as control mechanism for a shallow analysis of the whole business letter, especially the letter’s body. Fig. 1 shows the components of text skimming in ALV.

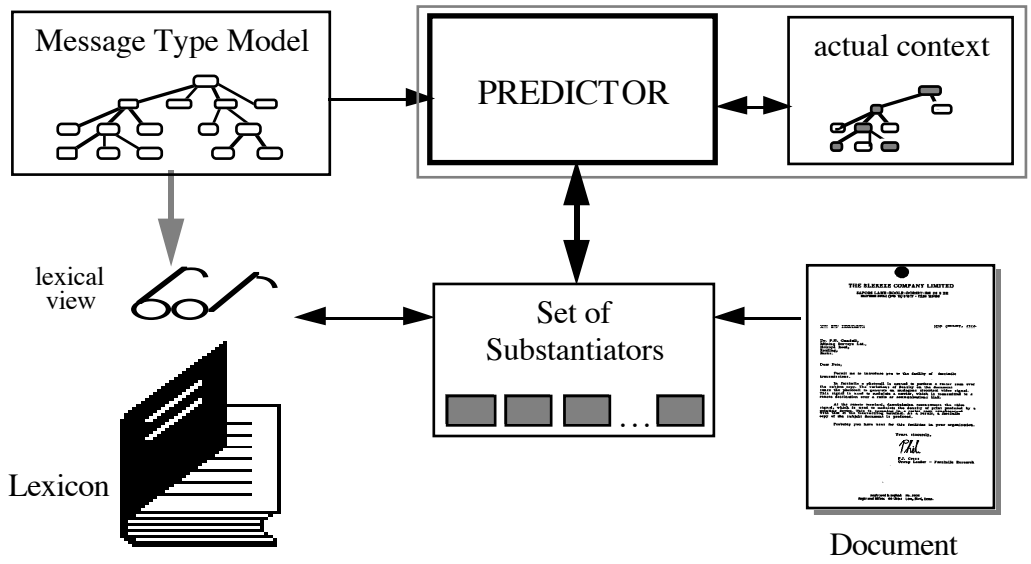

Fig. 1: Concept of Text Skimming in ALV.

The lexicon ([Hoch & Kieninger 93]) contains full forms and stem forms of words. Full forms are necessary for the requirements of text recognition. Some semantic information is given in form of clustered words, such as all names of customers, or all words oftenly contained in specific logical objects. This grouping is a partitioning (not necessarily a disjoint one) of the whole lexicon and each group of words is called a “lexical view”. Also the words are supplied with a basic meaning by lexical views referring to conceptual dependency-forms (CD-form, see [Schank 72]), such as agent, object or action. In addition, specific CD-forms are used for defining lexical views containing synonyms, such as the view “bestellen” (German for “to order”) which comprises words like “bestellen”, “ordern”, “anfordern”, etc.

The substantiators are the analysis components actually operating on the document text. They cover a wide range of different analysis techniques, such as statistically based information retrieval methods ([Hoch & Dengel 93]), keyword and pattern (surrounding context of key words) matching techniques ([Schmidt 93]), and island parsing approaches ([Kirchmann 93], [Malburg & Dengel 93]). Each substantiator is appropriate for special tasks of analysis, according to the text structure of logical objects, and to the kind and quality of extracted information. During their analysis all substantiators use the partitioned lexicon to get necessary

word information, e.g. to check lexical views. If necessary for the work of the substantiators, the words are analyzed morphologically by MORPHIX ([Finkler & Neumann 88]) to obtain the word stem and the morphological information.
In the message type model ([Gores & Bleisinger 92]), knowledge about different types of business letters is hierarchically organized. On the one hand, this knowledge comprises the important part of information searched for which can be either obligatory or optional. Information items are, for example in an order, the products or the prices of the products, respectively. On the other hand, semantic and syntactic constraints as well as control information for the predictor are included. Common control items are priority factors for the importance of the targeted information, an ordering of the targeted information within a document, or names of substantiator classes which are applicable. Basis of the expectation´s representation are enhanced CD-forms. Message elements are build upon these; message types are a set of message elements. In Fig. 2 a short example of the message type model is shown.

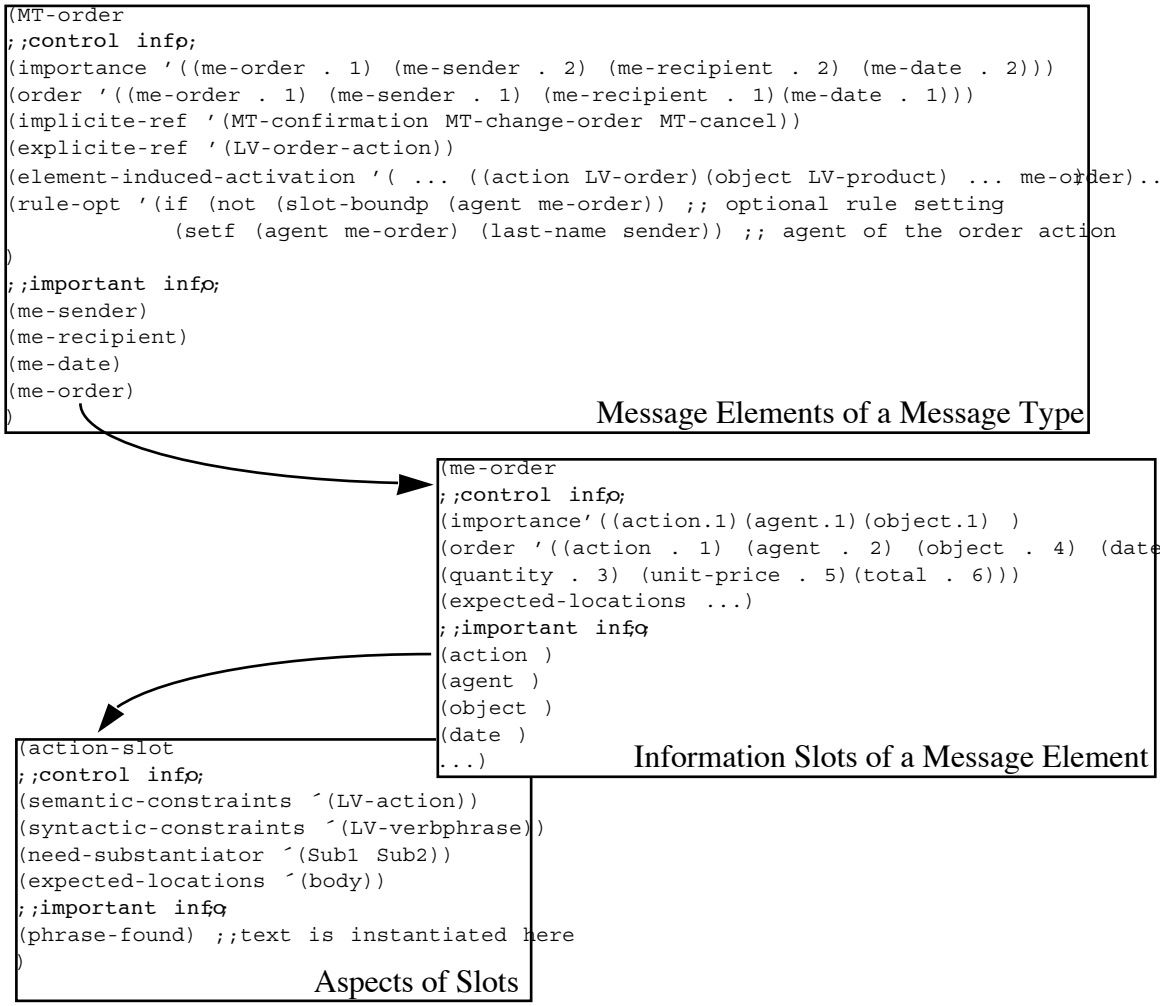

Fig. 2: Example of message elements and types.

The control component predictor uses the predefined message type model as a knowledge base describing what information is important (expectations). It creates the internal actual context containing all instantiated message elements and message types which are derived from analysis results yielded by the substantiators (text context) as well as pending expectations (expectation

context). The predictor first predicts the information searched for and second, the place searched in (e.g., logical object, specific position in a logical object). This is done according to both the specifications in the message type model and the actual context. The predictor performs a goal-oriented activation of substantiators, based on the predictions, thereby giving hints about the information searched for, and the place searched in. The substantiator results are returned to the predictor for further consistency tests and for the integration into the actual context.

3. The Predictor

Up to now, we gave an explanation of the text skimming concept in ALV as well as the lexicon component, the set of substantiators and the message type model. In the following, we discuss the behavior of the predictor and its actual context in more detail.

3.1 Predictor Concept

One major problem of expectation driven text analysis is the search for the starting point. The sole information which the system has at the beginning, is the set of all possible expectations given by all message types of the model. This initial set is too large to be of any help; therefore, no helpful expectations can be generated and the predictor has to overcome the so called starting problem. The way this is done in ALV is explained in Section 3.2.

After the predictor has set up a starting point for the analysis, it is able to generate several expectations about which kind of information might appear in the text. Because there are still many alternatives, the predictor has to discriminate among them in an efficient way until one expectation is chosen. This expectation corresponds to one message type — and in rare cases to a few. The discrimination phase of the ALV-predictor is discussed fully in Section 3.3.

After the discrimination is done, one substantiator is activated and the information found is passed to and examined by the predictor. When the predictor has detected text elements matching expectations, the data structure representing the outcome of the analysis — the text context — is build or enlarged. If the text elements found do not satisfy the expectation, the predictor performs an error recovery. These tasks are done in the instantiation phase which is proposed in Section 3.4.

Through consecutive calls to one or more substantiators all expectations of one message type are being tried to be instantiated. Thus, repeating discrimination and instantiation, the document is analyzed.

In the following the three phases of the predictor are explained in more detail and their interaction is shown in Fig. 3.

3.2 Start phase

In the previous section we have introduced the starting problem which has to be solved by cutting down the possible alternative expectations. This section describes different approaches to attack this problem: first the predictor activates special tools (substantiators) and secondly it performs, if necessary, a start discrimination using advanced data structures.

Being embedded into the document analysis system in ALV, it is possible to use previously extracted information for the classification of documents. With the substantiator InFoClas ([Hoch & Dengel 93]) working on textual information in combination with logical labeling results, documents are classified based on statistics. This is done according to message type specific word lists defined in the message type model, a letter database, and word frequency statistics for German texts.

Another possible solution for the classification task is the use of a keyword search and an advanced pattern search. (For details of the pattern match idea see for example [Hayes et al 88]). Therefore, the appropriate patterns have to be defined for each message type in the model. The classification leads to a recommendation about what kind of message type the analyzed document is. The result, a list of pairs of a message type and the probability of its occurrence in the document, is passed to the predictor. The message type with the highest peak value above a threshold value is used first by the predictor to build proper expectations. In this way, only the expectations which are part of the hypothesized message type are relevant for the ongoing analysis. The predictor stores the probability list for later use, since the next best classified message type is used in case of a faulty classification.
Having done this preliminary work, the set of expectations about what type the document is and what its contents are, can be reduced drastically. For most documents in our test set the message type with the highest probability is really the type of message in the analyzed document. This enables the predictor to start the analysis with the expectations constrained to the right message type.
If the probabilities of all message type hypotheses are lower than a given threshold value, the predictor cannot benefit from the classification. Then a repeated start discrimination and start instantiation is applied up to a level of restricting the expectations to some or even one message type. Thereby advanced data structures like discrimination trees are used (proposed in the FRUMP system [DeJong 79]). The concept of the start phase is shown in Fig. 3.

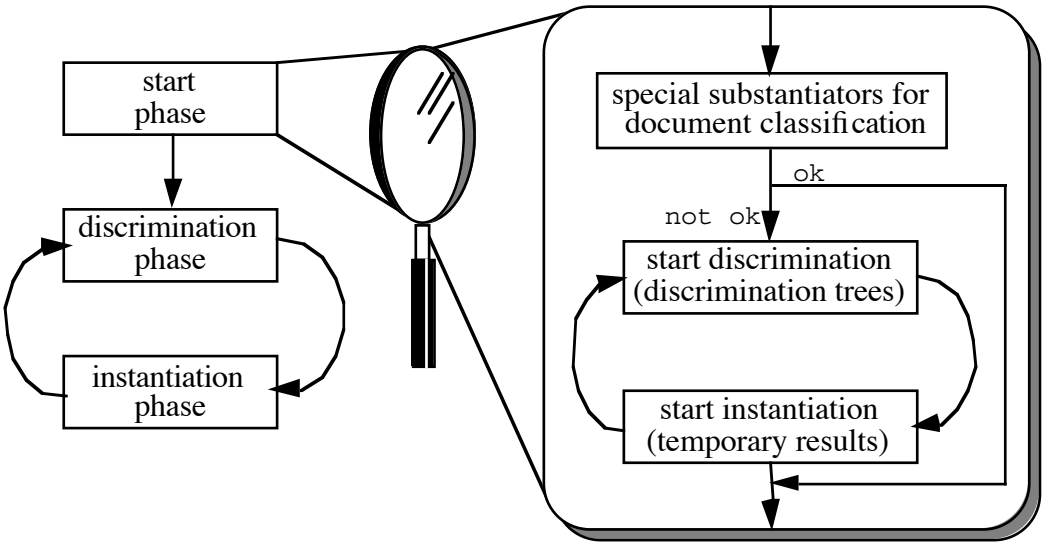

Fig. 3: Predictor concept and start phase in detail.

Our discrimination trees are constructed via control information given in the message types. The discrimination trees are constructed to set up significant expectations, so that by their occurrence in the text a document can be classified. All expectations not useful for this strong classification are ignored. An example of a special structural view of the message type model is shown in Fig. 4.

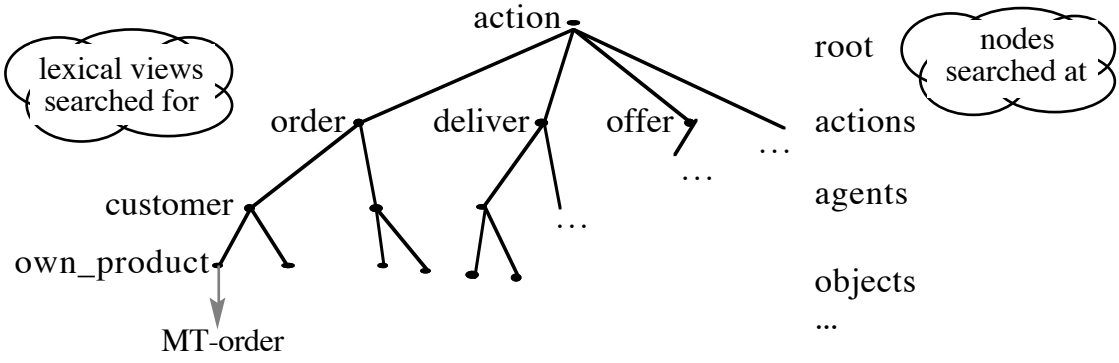

Fig. 4: Discrimination Tree for Actions.

The initial step of start discrimination begins at the root nodes of the discrimination trees, where the predictor has to decide which tree to use. This is done empirically using the most encouraging discrimination tree first. Heuristics included are to prefer textual over layout information, to prefer well recognized text, to prefer special words (e.g. verbs) and so on.
In the case of using the discrimination tree for actions, the first prediction generated asks the substantiators to search an action, i.e. a verbal phrase with the lexical view "action". This prediction is a disjunctive combination of the expectations given at the subnodes of the root. For example, given order, deliver, and offer as subnode expectations, all subsets of the lexical view "action", a verbal phrase with the lexical view "order", "deliver" or "offer" is searched.
The result of the substantiator called with this combined expectation is matched by the predictor with any expectation at the action nodes. Afterwards, matched results are temporarily instantiated in the start instantiation. In the next step the subnodes of the matching node are used to generate an expectation about the agent of the indicated action. Thus start discrimination and start instantiation descend the tree, predicting the expected actions, agents, objects, and so on.
Result of the start discrimination, whether done with the special classification tools or the conventional way, is always the restriction from the possible expectations of all message types to the expectations of one or a few message types. At this point the predictor can start the process of instantiating expectations of selected message types performing first a discrimination between these expectations. This is described in Section 3.3 and Section 3.4.

3.3 Discrimination phase

After the start discrimination the expectation context is cut down to one or a few message types, corresponding to several expectations per message type activated. The text context is empty if the start discrimination was done by classification, or consists of the text in the document leading to the classification if done using the discrimination trees.
Start discrimination works on the high level of the whole message type model to reduce the expectations to one or a few message types. Now the tasks of discrimination is done on basis of the actual context, i.e. the text already found and the pending expectations generated. It is splitted in a coarse discrimination and a fine discrimination. The goal of coarse discrimination is to identify one message type being relevant for further document analysis. The goal of fine discrimination, done at the level of message elements, i.e. usually "inside" the activated message type(s), is to identify one message element for extracting information.
Coarse discrimination is performed if expectations of more than one message type are still in question or the activated message type is not appropriate. Remember, the discrimination phase

alternates with the instantiation phase until the whole document is analyzed. Therefore coarse discrimination is performed either after the start phase or after the instantiation phase. If during analysis the message type activated cannot be verified, or additional message types are discovered in the document the coarse discrimination must start again.
The activation of a message type happens in either of two ways: explicitly by very important items (phrases, layout or logical objects) or implicitly by such special items found in the context of an already activated message type. Explicit activation is performed during the coarse discrimination if significant items are found in the document. Since this activation is based only on a small analyzed portion of the document, it may be only partially correct. With the mechanism of implicit activation it is possible not only to correct misactivations, but also to recognize documents containing more than one message type by activating the related ones. The implicit activation is started by the predictor using the activated message types´ control information pointing to related message types.
Fine discrimination is performed immediately after the start phase if one message type is already identified. Also it is done after coarse discrimination if more than one message type was hypothesized, or after the instantiation phase if another expectation of the actual message type has to be selected for instantiation.
In our model the message elements of each message type have an associated order value concerning their arrangement in a typical document as control information. By using this value similarly structured documents can be analyzed efficiently, calling the substantiator with the expectations of the message type according to the order values. But this is only possible during fine discrimination. Document structures differing from this given order require more computational work to be analyzed by the predictor.
Alternatively, the predictor can build expectations, so called discriminants, according to the sequence set up by the tree-like representation of the message type model, the so-called message tree. Especially relevant to prune the message tree is the message element's control information importance. The disadvantage of the discrimination based on the message tree pruned is the disregard of the order control information.
To benefit from both order and importance control information without giving up the goal of a robust document analysis, our predictor uses a combination of both approaches. Thus the predictor is able to analyze documents with structures varying from the one set up in and preferred by the message type model through the order control information. The discriminant build is constructed using the combination of the expectations in question yet.
The predictor finishes the loop of discrimination and instantiation if no discriminant can be built anymore. This happens first if a successful analysis results in an instantiated message type containing the document´s important information. Secondly, this happens if the activated message type cannot be verified and no new one is predictable, thus the analysis fails.
If a discriminant is constructed it is used in the instantiation phase as an expectation for guiding substantiators and filling the actual context. This is described in the following section.

3.4 Instantiation phase

The starting point for an instantiation phase is the expectation discriminated before. According to the control information of this expectation in the message type model, a substantiator is selected for extracting information from the document's text. Also on the basis of control information, the predictor generates predictions both for the required information and for the search area within the document. Then, the predictor activates the chosen substantiator with the predictions as parameters.

The activated substantiator analyzes some text of the document at hand and returns the result to the predictor. The answer produced by the substantiator is examined by the predictor with regard to restrictions in the initial expectation which are syntactic and semantic constraints as well as special rules. Such rules are categorized either default, optional or obligatory. After the substantiator's call, mainly two different situations are possible: first the substantiator's analysis was successful and second the substantiator has created no result or the answer returned was unacceptable.
If the analysis was successful and the extracted information matches fully with the constraints of the predefined expectation, the predictor enlarges the text context of the actual context. This is done by simply entering text fragments found or corresponding CD-forms in the appropriate place of message elements. If the answer comprises more than the expected information, e.g. expecting "action:order" and receiving "action:order" and "agent:customer_x", the predictor attempts to enter all this information found. If the answer does not match fully, the predictor tries to generate an entry of the analysis result as a partial instantiation of the expectation. This can be problematic because the prediction was not really fulfilled.
Following the instantiation the obligatory rules must be applied and are used to check for sound instantiation and consistency.
After a successful instantiation, a new prediction of one expectation is build by the discrimination phase, if possible, and instantiation begins again. The skimming task stops if no new predictions can be established.
If the analysis of the substantiator fails, that means a partial match is too bad or no answer is returned, an error recovery has to start. First the predictor tries to activate alternative substantiators, if possible, in order to obtain an acceptable result.
If these substantiator calls also cannot produce the expected result, the predictor tries the application of rules defined for the expected information. Here all three classes of rules are relevant. At first, default rules are used to serve unsatisfied expectations with a default value, e.g. the city name of the recipient in our case defaults to "Kaiserslautern".

```
(rule-def  '((if  (not  (slot-boundp  (city  me-recipient)))
             (setf  (city  me-recipient)  'kaiserslautern))))
```

If no default rule is defined, the predictor can assign values to expectations from related message elements and message types using optional rules. For example, if the customer of an order is not identified within the letter body, then the sender of the letter will be used as the value for the customer using the optional rule

```
(rule-opt  '((if  (not  (slot-boundp  (agent  me-order)))
             (setf  (agent  me-order)  (last-name  me-sender)))))
```

To check the values for consistency, obligatory rules are applied. Our example rule tests, whether the agent of the order equals the last name of the sender or the subscriber.

```
(rule-oblig  ´((or  (eq  (agent  me-order)  (last-name  me-sender))
             (eq  (agent  me-order)  (last-name  me-subscriber)))))
```

As an abbreviation, obligatory rules can be defined for both testing the values and calculating new values for unbound slots. The test performed produces an error, if none of the conditions are satisfied.
The instantiation of any expectation with rules is attached a value of uncertainty. This expresses the fact, that instantiations yielded by rules are less certain than instantiations yielded by substantiator calls.
If neither the instantiation with substantiators nor the application of rules produced the awaited results, the predictor checks this failure's consequences. If the unsatisfied expectation was a necessary expectation according to its control information, the analysis cannot proceed with the current expectation context or has to be cancelled. Otherwise a new discrimination phase is activated, leaving the previous expectation unsatisfied. The result of the new discrimination is a new expectation for instantiation, either contained in the same or in a modified actual context. The modification is due to changing activated message types. If no new expectation can be generated, the skimming process stops without any completely instantiated message type.

4. Usage of the analysis results

The results of the predictor's analysis are in case of success one or more (partial) instantiated message types classifying the analyzed letter (see Fig. 5). All important information expected is instantiated, either directly by substantiator calls or indirectly using rules. If the analysis process has stopped because essential information could not be found, the result consists of partial instantiated expectations only, which are usually too uncertain for any postprocessing.

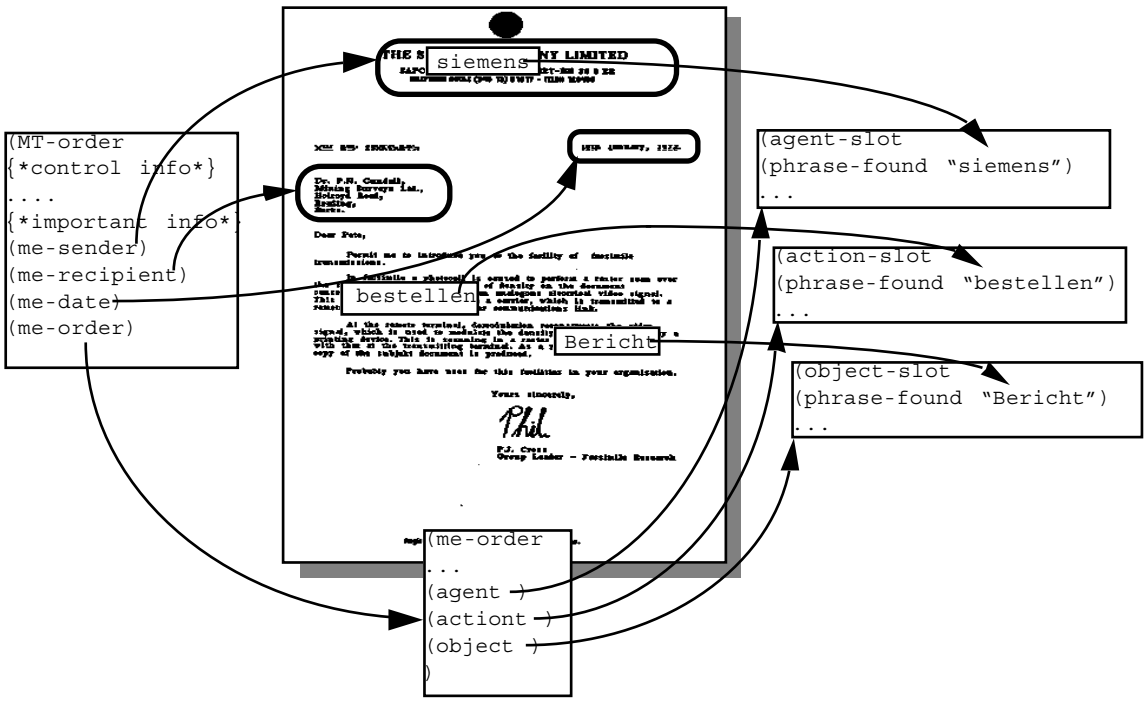

Fig. 5: Instantiated message type.

With successfully instantiated message types, that means with the extracted information of business letters, a further postprocessing is performed. Using the analysis results of the recipient address an electronic notification of incoming business letters is done, including the

forwarding of the identified message type and the extracted information in the sense of a letter abstract. The classification results and the extracted information can also be used for an automatic indexing of documents filed. Thus the working amount for mail distribution and archiving is reduced.
Moreover, additional analysis results can be utilized, for example in case of an order, the products, their prices and quantities can be automatically checked against a product database. Thus, the generation of partial answers for some types of business letters is possible, for example the confirmation of an order.

5. Conclusion

Text analysis as a part of paper document analysis has to meet specific requirements. On the one hand, text analysis is performed on real natural language texts, and, on the other hand, the input text contains gaps, word alternatives, and even illegal words. In order to handle these problems it seems reasonable to perform an expectation driven partial analysis in order to extract predefined important information.
Obviously, a partial analysis has the drawback of loosing information in the text under analysis. But a shallow text analysis or skimming is an excellent way to overcome the imperfections of the text recognition results and to identify important information.
In general, the architecture of our expectation driven partial text analysis comprises the message type model specifying the important information as well as control information, the set of substantiators for different analysis tasks, and the control module predictor. This architecture shows a strict separation between the predictor and the substantiators. The architectures of other proposed systems, such as FRUMP ([DeJong 82]) or RESEARCHER ([Lebowitz 85]), involve only one substantiator which is strongly connected with the predictor. In this way, our analysis flow is more flexible and our system is easily expandable with additional substantiators applying different methods or doing new analysis tasks.
Any of the three working phases of our predictor comprises new features. The starting problem is solved by applying specialized substantiators, namely a statistics-based classifier and, in near future, a pattern matcher. This is a new and effective way to choose from the whole model a message type as a starting point for further expectation driven analysis. Additionally, the solution of the start problem as done in the FRUMP system, which has been realized on the basis of discrimination trees, is incorporated in our system.
Our discrimination phase is structured into distinct steps, the coarse discrimination among message types and the fine discrimination among message elements of one or a few message types. During discrimination we use control information of both order and importance of message elements within a message type. In this way, we combine two approaches.
In the instantiation phase, the error recovery is done in a skilful way by activating alternative substantiators or by using rules specified in the message type model. On the basis of rules the predictor can assign default entries or can calculate new entries from other results for the expectation.
Up today, all parts of the ALV skimming concept are prototypically implemented. The message type model specified comprises five message types, the lexicon developed contains more than 8000 entries, whereby only for a set of 50 very important words semantic information is attached. Substantiators fully implemented are the statistically based classifier, the pattern matcher as well as the address parser based on a semantic grammar and an island parsing strategy. Additionally, island parsing for the subject part and the letter body are currently under investigation and these implementations are nearly completed.

All phases of the predictor are implemented, but only in a rudimentary form. In such a manner, we can do first tests for the applicability of the whole skimming concept. Momentarily, the predictor is being extended and its intended performance (for a detailed description see [Gores & Bleisinger 93]) will be achieved soon.

6. References

[Baird et al 86] H. S. Baird, S. Kahan, and T. Pavlidis. Components of an Omnifont Page Reader. Proc. of 8th Intern. Conf. on Pattern Recognition, Paris, 1986.

[DeJong 79] Gerald Francis DeJong. Skimming Stories in Real Time: An Experiment in Integrated Understanding. Dissertation (Ph.D.), Faculty of the Graduate School of Yale University, 1979.

[DeJong 82] Gerald F. DeJong. An Overview of the FRUMP System. in W. G. Lehnert, M. H. Ringle (eds.): Strategies for NL Processing, Lawrence Erlbaum Assoc., Hillsdale, pp. 149-175, 1982.

[Dengel et al 92a] A. Dengel, R. Bleisinger, R. Hoch, F. Hönes, F. Fein, and M. Malburg. POda: The Paper Interface to ODA. DFKI Research Report RR-92-02, Febr. 1992.

[Dengel et al 92b] A. Dengel, R. Bleisinger, R. Hoch, F. Hönes, and F. Fein. From Paper to Office Document Standard Representation. IEEE Computer, vol. 25, no. 7, pp. 63-67, July 1992.

[Finkler & Neumann 88] W. Finkler and G. Neumann. MORPHIX - A Fast Realization of a Classification-based Approach to Morphology. Proc. of 4th Österreichische AI-Tagung, pp. 11-19, 1988.

[Gores & Bleisinger 92] K.-P. Gores and R. Bleisinger. The Message Type Model Representation (in German). DFKI Document D-92-28, Nov. 1992.

[Gores & Bleisinger 93] K.-P. Gores and R. Bleisinger. An Expectation-driven Coordinator for a Partial Text Analysis (in German). DFKI Document D-93-07, May 1993.

[Hayes 92] P. J. Hayes. Intelligent High-Volume Text Processing Using Shallow, Domain-Specific Techniques. in P. S. Jacobs (ed.): Text-Based Intelligent Systems, Lawrence Erlbaum Association, Hillsdale, pp. 227-243, 1992.

[Hayes et al 88] P. J. Hayes, L. E. Knecht, and M. J. Cellio. A News Story Categorization System. Proc. of 2nd Conf. on Applied NL Processing, Austin, Texas, pp. 9-17, February 1988.

[Hobbs et al 92] J. R. Hobbs, D. E. Appelt, J. Bear, M. Tyson, and D. Magerman. Robust Processing of Real-World Natural-Language Texts. in P. S. Jacobs (ed.): Text-Based Intelligent Systems, Lawrence Erlbaum Association, Hillsdale, pp. 13- 33, 1992.

[Hoch & Dengel 93] R. Hoch and A. Dengel. INFOCLAS: Classifying the Message in Printed Business Letters. Proc. of 2nd Symp. on Document Analysis and Information Retrieval, Las Vegas, Nevada, pp. 443-456, April 1993.

[Hoch & Kieninger 93] R. Hoch and T. Kieninger. On virtual partitioning of large dictionaries for contextual post-processing to improve character recognition. will be published in: Proc. of 2nd Conf. on Document Analysis and Recognition, Tsukuba Science City, Oct. 1993.

[Hu et al 93]) P. Hu, C. Shi, K. Wang, and X. Zhang. An Integrated Approach to Text Understanding. Proc. of 9th IEEE Conf. on AI for Applications, Orlando, Fl, pp. 79-85, March 1993.

[Hull et al 92] J. J. Hull, S. Khoubyari, and T. K. Ho. Word Image Matching as a Technique for Degraded Text Recognition. Proc. of 11th Intern. Conf. on Pattern Recognition, The Hague, Aug./Sept. 1992.

[Jackson et al 91] E. Jackson, D. Appelt, J. Bear, R. Moore, and A. Podlozny. A Template Matcher for Robust NL Interpretation. Proc. of 2nd Speech and NL Workshop, pp. 190-194, Febr. 1991.

[Kirchmann 93] H. Kirchmann. Usage of Island-Parsing Strategies in Document Analysis (in German). Master thesis, University of Kaiserslautern, March 1993.

[Lebowitz 85] M. Lebowitz. RESEARCHER: An Experiment in Intelligent Information Systems. Proc. of 9th Intern. Joint Conf. on AI, Los Angeles, CA, pp. 858-862, 1985.

[Lehnert et al 91] W. Lehnert, C. Cardie, D. Fisher, E. Riloff, and R. Williams. University of Massachusetts: Description of the CIRCUS System as Used for MUC-3. Proc. of 3rd Message Understanding Conf. (MUC-3), San Diego, Ca, May 1991.

[Malburg & Dengel 93] M. Malburg and A. Dengel. Address Verification in Structured Documents for Automatic Mail Delivery. Proc. of 1st Intern. Conf. on Postal Technologies, Nantes, pp. 447-454, June 1993.

[Mauldin 91] M. L. Mauldin. Retrieval Performance in FERRET - A Conceptual Information Retrieval System. SIGIR Forum, Special Issue of 14th Annual Intern. ACM/SIGIR Conf. on Research and Development in Information Retrieval, pp. 347-355, 1991.

[Nagy et al 92] G.Nagy, S. Seth, and M. Viswanathan. A Prototyp Document Image Analysis System for Technical Journals. IEEE Computer, vol. 25, no. 7, pp. 10-24, July 1992.

[Palumbo & Srihari 86] P. W. Palumbo and S. N. Srihari. Text Parsing using Spatial Information for Recognizing Addresses in Mail Pieces. Proc. of 8th Intern. Conf. on Pattern Recognition, Paris, 1986.

[Prussak & Hull 91] M. Prussak and J.J. Hull. A Multi-level Pattern Matching Method for Text Image Parsing. Proc. of 7th IEEE Conf. on AI for Applications, Miami, FL, Feb. 1991.

[Rau & Jacobs 88] L. F. Rau and P. S. Jacobs. Integrating Top-down and Bottom-up Strategies in a Text Processing System. Proc. of 2nd Conf. on Applied NL Processing, Austin, Texas, pp. 129-135, Febr.1988.

[Schank 72] R. C. Schank. Conceptual□ Dependency: A theory of natural language understanding. Cognitive Psychology, 3 (4), pp. 552-631, 1972.

[Schmidt 93] M. Schmidt. Key Word Substantiator for Document Analysis (in German). Bachelor thesis, University of Kaiserslautern, May 1993.

[Story et al 92] G. A. Story, L. O´Gorman, D. Fox, L. L.Schaper, and H. V. Jagadish. The RightPages: An Electronic Library for Alerting and Browsing. IEEE Computer, vol. 25, no. 9, 1992.

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

DFKI
-Bibliothek-
PF 2080
67608 Kaiserslautern
FRG

DFKI Publikationen

Die folgenden DFKI Veröffentlichungen sowie die aktuelle Liste von allen bisher erschienenen Publikationen können von der oben angegebenen Adresse oder per anonymem ftp von ftp.dfki.uni-kl.de (131.246.241.100) unter pub/Publications bezogen werden.
Die Berichte werden, wenn nicht anders gekennzeichnet, kostenlos abgegeben.

DFKI Publications

The following DFKI publications or the list of all published papers so far are obtainable from the above address or via anonymous ftp from ftp.dfki.uni-kl.de (131.246.241.100) under pub/Publications.
The reports are distributed free of charge except if otherwise indicated.

DFKI Research Reports

RR-93-10
Martin Buchheit, Francesco M. Donini, Andrea Schaerf: Decidable Reasoning in Terminological Knowledge Representation Systems
35 pages

RR-93-11
Bernhard Nebel, Hans-Jürgen Bürckert:
Reasoning about Temporal Relations:
A Maximal Tractable Subclass of Allen's Interval Algebra
28 pages

RR-93-12
Pierre Sablayrolles: A Two-Level Semantics for French Expressions of Motion
51 pages

RR-93-13
Franz Baader, Karl Schlechta:
A Semantics for Open Normal Defaults via a Modified Preferential Approach
25 pages

RR-93-14
Joachim Niehren, Andreas Podelski, Ralf Treinen:
Equational and Membership Constraints for Infinite Trees
33 pages

RR-93-15
Frank Berger, Thomas Fehrle, Kristof Klöckner, Volker Schölles, Markus A. Thies, Wolfgang Wahlster: PLUS - Plan-based User Support
Final Project Report
33 pages

RR-93-16
Gert Smolka, Martin Henz, Jörg Würtz: Object-Oriented Concurrent Constraint Programming in Oz
17 pages

RR-93-17
Rolf Backofen:
Regular Path Expressions in Feature Logic
37 pages

RR-93-18
Klaus Schild: Terminological Cycles and the Propositional μ-Calculus
32 pages

RR-93-20
Franz Baader, Bernhard Hollunder:
Embedding Defaults into Terminological Knowledge Representation Formalisms
34 pages

RR-93-22
Manfred Meyer, Jörg Müller:
Weak Looking-Ahead and its Application in Computer-Aided Process Planning
17 pages

RR-93-23
Andreas Dengel, Ottmar Lutzy:
Comparative Study of Connectionist Simulators
20 pages

RR-93-24
Rainer Hoch, Andreas Dengel:
Document Highlighting —
Message Classification in Printed Business Letters
17 pages

RR-93-25
Klaus Fischer, Norbert Kuhn: A DAI Approach to Modeling the Transportation Domain
93 pages

RR-93-26
Jörg P. Müller, Markus Pischel: The Agent Architecture InteRRaP: Concept and Application
99 pages

RR-93-27
Hans-Ulrich Krieger:
Derivation Without Lexical Rules
33 pages

RR-93-28
Hans-Ulrich Krieger, John Nerbonne, Hannes Pirker: Feature-Based Allomorphy
8 pages

RR-93-29
Armin Laux: Representing Belief in Multi-Agent Worlds viaTerminological Logics
35 pages

RR-93-30
Stephen P. Spackman, Elizabeth A. Hinkelman:
Corporate Agents
14 pages

RR-93-31
Elizabeth A. Hinkelman, Stephen P. Spackman:
Abductive Speech Act Recognition, Corporate Agents and the COSMA System
34 pages

RR-93-32
David R. Traum, Elizabeth A. Hinkelman:
Conversation Acts in Task-Oriented Spoken Dialogue
28 pages

RR-93-33
Bernhard Nebel, Jana Koehler:
Plan Reuse versus Plan Generation: A Theoretical and Empirical Analysis
33 pages

RR-93-34
Wolfgang Wahlster:
Verbmobil Translation of Face-To-Face Dialogs
10 pages

RR-93-35
Harold Boley, François Bry, Ulrich Geske (Eds.):
Neuere Entwicklungen der deklarativen KI-Programmierung — *Proceedings*
150 Seiten
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

RR-93-36
Michael M. Richter, Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Ralf Legleitner, Gabriele Schmidt: Von IDA bis IMCOD: Expertensysteme im CIM-Umfeld
13 Seiten

RR-93-38
Stephan Baumann: Document Recognition of Printed Scores and Transformation into MIDI
24 pages

RR-93-40
Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Werner Nutt, Andrea Schaerf:
Queries, Rules and Definitions as Epistemic Statements in Concept Languages
23 pages

RR-93-41
Winfried H. Graf: LAYLAB: A Constraint-Based Layout Manager for Multimedia Presentations
9 pages

RR-93-42
Hubert Comon, Ralf Treinen:
The First-Order Theory of Lexicographic Path Orderings is Undecidable
9 pages

RR-93-43
M. Bauer, G. Paul: Logic-based Plan Recognition for Intelligent Help Systems
15 pages

RR-93-44
Martin Buchheit, Manfred A. Jeusfeld, Werner Nutt, Martin Staudt: Subsumption between Queries to Object-Oriented Databases
36 pages

RR-93-45
Rainer Hoch: On Virtual Partitioning of Large Dictionaries for Contextual Post-Processing to Improve Character Recognition
21 pages

RR-93-46
Philipp Hanschke: A Declarative Integration of Terminological, Constraint-based, Data-driven, and Goal-directed Reasoning
81 pages

RR-93-48
Franz Baader, Martin Buchheit, Bernhard Hollunder:
Cardinality Restrictions on Concepts
20 pages

RR-94-01
Elisabeth André, Thomas Rist:
Multimedia Presentations:
The Support of Passive and Active Viewing
15 pages

RR-94-02
Elisabeth André, Thomas Rist:
Von Textgeneratoren zu Intellimedia-Präsentationssystemen
22 Seiten

RR-94-03
Gert Smolka:
A Calculus for Higher-Order Concurrent Constraint Programming with Deep Guards
34 pages

RR-94-05
Franz Schmalhofer, J.Stuart Aitken, Lyle E. Bourne jr.:
Beyond the Knowledge Level: Descriptions of Rational Behavior for Sharing and Reuse
81 pages

RR-94-06
Dietmar Dengler:
An Adaptive Deductive Planning System
17 pages

RR-94-07
Harold Boley: Finite Domains and Exclusions as First-Class Citizens
25 pages

RR-94-08
Otto Kühn, Björn Höfling: Conserving Corporate Knowledge for Crankshaft Design
17 pages

RR-94-10
Knut Hinkelmann, Helge Hintze:
Computing Cost Estimates for Proof Strategies
22 pages

RR-94-11
Knut Hinkelmann: A Consequence Finding Approach for Feature Recognition in CAPP
18 pages

RR-94-12
Hubert Comon, Ralf Treinen:
Ordering Constraints on Trees
34 pages

RR-94-13
Jana Koehler: Planning from Second Principles —A Logic-based Approach
49 pages

RR-94-14
Harold Boley, Ulrich Buhrmann, Christof Kremer:
Towards a Sharable Knowledge Base on Recyclable Plastics
14 pages

RR-94-15
Winfried H. Graf, Stefan Neurohr: Using Graphical Style and Visibility Constraints for a Meaningful Layout in Visual Programming Interfaces
20 pages

RR-94-16
Gert Smolka: A Foundation for Higher-order Concurrent Constraint Programming
26 pages

DFKI Technical Memos

TM-92-04
Jürgen Müller, Jörg Müller, Markus Pischel, Ralf Scheidhauer:
On the Representation of Temporal Knowledge
61 pages

TM-92-05
Franz Schmalhofer, Christoph Globig, Jörg Thoben:
The refitting of plans by a human expert
10 pages

TM-92-06
Otto Kühn, Franz Schmalhofer: Hierarchical skeletal plan refinement: Task- and inference structures
14 pages

TM-92-08
Anne Kilger: Realization of Tree Adjoining Grammars with Unification
27 pages

TM-93-01
Otto Kühn, Andreas Birk: Reconstructive Integrated Explanation of Lathe Production Plans
20 pages

TM-93-02
Pierre Sablayrolles, Achim Schupeta:
Conlfict Resolving Negotiation for COoperative Schedule Management
21 pages

TM-93-03
Harold Boley, Ulrich Buhrmann, Christof Kremer:
Konzeption einer deklarativen Wissensbasis über recyclingrelevante Materialien
11 pages

TM-93-04
Hans-Günther Hein:
Propagation Techniques in WAM-based Architectures — The FIDO-III Approach
105 pages

TM-93-05
Michael Sintek: Indexing PROLOG Procedures into DAGs by Heuristic Classification
64 pages

TM-94-01
Rainer Bleisinger, Klaus-Peter Gores:
Text Skimming as a Part in Paper Document Understanding
14 pages

TM-94-02
Rainer Bleisinger, Berthold Kröll:
Representation of Non-Convex Time Intervals and Propagation of Non-Convex Relations
11 pages

DFKI Documents

D-93-10
Elizabeth Hinkelman, Markus Vonerden, Christoph Jung: Natural Language Software Registry
(Second Edition)
174 pages

D-93-11
Knut Hinkelmann, Armin Laux (Eds.):
DFKI Workshop on Knowledge Representation Techniques — Proceedings
88 pages

D-93-12
Harold Boley, Klaus Elsbernd, Michael Herfert, Michael Sintek, Werner Stein:
RELFUN Guide: Programming with Relations and Functions Made Easy
86 pages

D-93-14
Manfred Meyer (Ed.): Constraint Processing – Proceedings of the International Workshop at CSAM'93, July 20-21, 1993
264 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-93-15
Robert Laux:
Untersuchung maschineller Lernverfahren und heuristischer Methoden im Hinblick auf deren Kombination zur Unterstützung eines Chart-Parsers
86 Seiten

D-93-16
Bernd Bachmann, Ansgar Bernardi, Christoph Klauck, Gabriele Schmidt: Design & KI
74 Seiten

D-93-20
Bernhard Herbig:
Eine homogene Implementierungsebene für einen hybriden Wissensrepräsentationsformalismus
97 Seiten

D-93-21
Dennis Drollinger:
Intelligentes Backtracking in Inferenzsystemen am Beispiel Terminologischer Logiken
53 Seiten

D-93-22
Andreas Abecker:
Implementierung graphischer Benutzungsoberflächen mit Tcl/Tk und Common Lisp
44 Seiten

D-93-24
Brigitte Krenn, Martin Volk:
DiTo-Datenbank: Datendokumentation zu Funktionsverbgefügen und Relativsätzen
66 Seiten

D-93-25
Hans-Jürgen Bürckert, Werner Nutt (Eds.):
Modeling Epistemic Propositions
118 pages
Note: This document is available only for a nominal charge of 25 DM (or 15 US-$).

D-93-26
Frank Peters: Unterstützung des Experten bei der Formalisierung von Textwissen
INFOCOM:
Eine interaktive Formalisierungskomponente
58 Seiten

D-93-27
Rolf Backofen, Hans-Ulrich Krieger, Stephen P. Spackman, Hans Uszkoreit (Eds.):
Report of theEAGLES Workshop on Implemented Formalisms at DFKI, Saarbrücken
110 pages

D-94-01
Josua Boon (Ed.):
DFKI-Publications: The First Four Years
1990 - 1993
75 pages

D-94-02
Markus Steffens: Wissenserhebung und Analyse zum Entwicklungsprozeß eines Druckbehälters aus Faserverbundstoff
90 pages

D-94-03
Franz Schmalhofer: Maschinelles Lernen:
Eine kognitionswissenschaftliche Betrachtung
54 pages

D-94-04
Franz Schmalhofer, Ludger van Elst:
Entwicklung von Expertensystemen:
Prototypen, Tiefenmodellierung und kooperative Wissensevolution
22 pages

D-94-06
Ulrich Buhrmann:
Erstellung einer deklarativen Wissensbasis über recyclingrelevante Materialien
117 pages

D-94-08
Harald Feibel: IGLOO 1.0 - Eine grafikunterstützte Beweisentwicklungsumgebung
58 Seiten

D-94-07
Claudia Wenzel, Rainer Hoch:
Eine Übersicht über Information Retrieval (IR) und NLP-Verfahren zur Klassifikation von Texten
25 Seiten