4th World Multiconference on Systemics, Cybernetics and Informatics and 6th International Conference on Information Systems, Analysis and Synthesis (SCI/ISAS 2000), Orlando, Florida (USA), July 23 - 26, 2000 (Invited Talk)

Profile-Based Information Supply from Text Sources

Andreas Dengel, Claudia Wenzel, Markus Junker
German Research Center for Artificial Intelligence (DFKI) GmbH
P.O. Box 2080, D-67608 Kaiserslautern, Germany

Abstract

It is the common goal of today's knowledge management systems to bring the right piece of knowledge to the right person at the right time. As soon as documents are involved in this process of information supply, intelligent techniques for information supply from text sources have to be employed. To this end, we propose a profile-based approach. Profiles describe the generic information need of individual persons according to their tasks and interests. Attached to these information needs, declarative analysis knowledge exhibits the textual properties of information satisfying these profiles. Such patterns are used by intelligent information assistants and allow them a very efficient and goal-directed analysis. Whenever the current context of a user is available, it can be used as a dynamic extension of a profile. In this case, information assistants can act more specific, thus receiving a better result quality. Our approach currently distinguishes three information assistants: one for text categorization, one for information extraction, and one for process identification. To make profile construction as easy as possible, distinct acquisition mechanisms have been developed for each assistant.

Keywords: Information Extraction, Text Categorization, Document Analysis, User Profiles, Process Context

1. INTRODUCTION

Information is one of the key factors for the competitiveness of an enterprise. The exploitation of data and information in order to support the roles and tasks of individuals is indispensable for success of the daily business. *Knowledge Management* may be seen as the task to understand information with respect to the actual need, to acquire the appropriate bits of knowledge and to convert and organize them for an effective use in an enterprise.

Information however in most cases is informal, heterogeneous and distant in place and time. The accessibility of information requires tools which are aware of specific needs capable to capture the relevant parts of information and relate it to an enterprise‘s workflow and its employees. To tackle this problem, a promising approach is the employment of *user profiles* describing categories of information as well as formal messages representing attribute-value slots of individual information needs to be satisfied. Thus, profiles represent the persistent part of a user‘s information need which also should adapt to changing requirements. Beneath this persistent part, the dynamic *context* of the user's environment may also be valuable analysis information.

Intelligent information assistants (IA) [1] for e.g., text categorization, information extraction, or information retrieval are techniques which help converting textual data into user-specific information. To achieve this, they must on the one hand be able to integrate profiles and contexts during analysis while on the other hand they should also provide facilities for easing the task of profile and context generation.

A typical approach is presented by Budzik and Hammond [2] who describe *Information Management Assistants* which observe users while they interact with everyday applications and then anticipate their information needs using a model of the task at hand. The idea of assessing the user's work context enhances the quality of information retrieval.

Unfortunately, the module which shall anticipate the user's future information needs on the basis of the actual work context and the stored task model, is not discussed in depth in their published work. However, it seems that they mainly build upon rather shallow, static context models which roughly determine the area of work in order to dissolve linguistic ambiguities in text retrieval. Natural-language aspects play an important role in their work: The system [3] observes interactions with everyday applications like word processors and web browsers, and mainly tries to find out the linguistic context of occupation. This approach is focused on the personal context rather than the dynamic task or process context and is thus related to user profiling and personal information agents.

This paper describes our own approach to profile-based information supply from electronic and paper documents. Therefore, the next section reveals the overall scenario of our solution while section 3 explains the special treatment necessary for a successful analysis of paper documents. Afterwards, section 4 and 5 explain the behavior of the two assistants for text categorization and information extraction in a context-free situation in more detail. The usage of context information for two similar purposes is subject to section 6. We conclude with some final remarks in section 7.

2. PROFILE-BASED INFORMATION SUPPLY

The usage of user profiles for textual analysis within the scenario of a networked company requires a coherent connection between profiles, the company's information systems and the company's organizational memory to derive its full benefits. The resulting static scenario is subject of figure 1.

In the center of the scenario considered, we find the user with his information needs. They are expressed by profiles. The problem, however, is how to get appropriate profiles which could serve as context for measuring the relevancy of information for individual´s tasks, roles, interests.

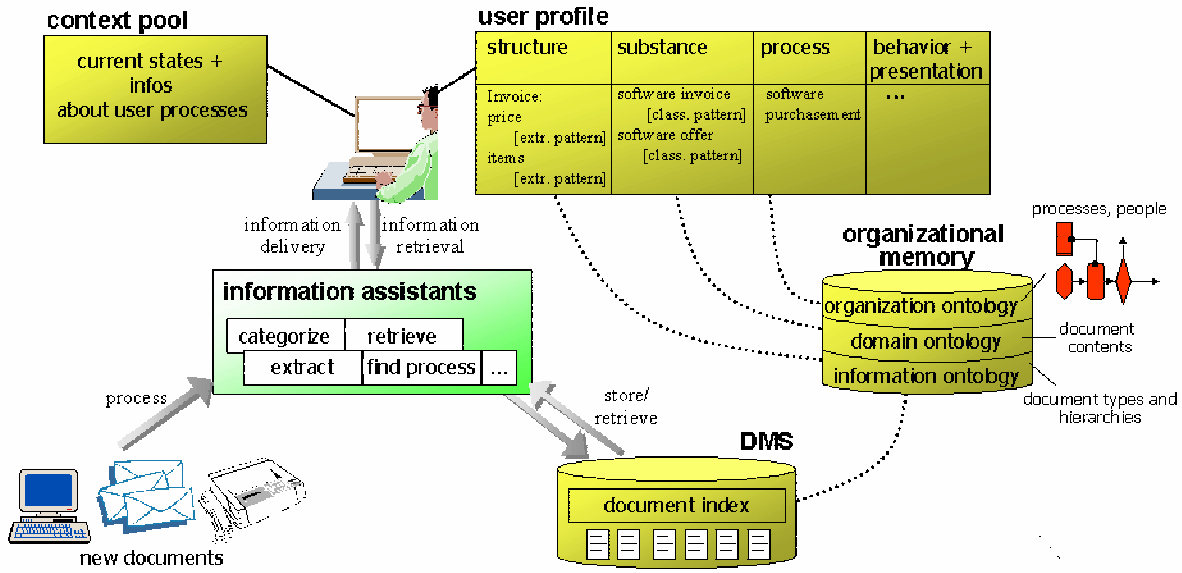
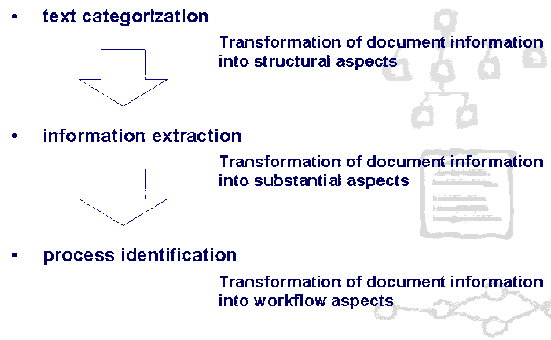


Figure 1: A scenario for profile-based information supply

In our scientific consideration, a profile may be seen as a threefold entity consisting of a structural, a substantial as well as a workflow-related aspect. A *structural profile* may be seen as a taxonomy of placeholders representing an implicit aggregation or generalization of information categories which in general correspond to a individualized classification scheme. A *substantial profile* may be seen as an abstraction of the subjective understanding of a set of documents with respect to a pregiven set of information categories. In other words each set of documents belonging to one and the same category is described by a set of representative textual features, i.e. terms, describing common information extracts of the captured topics. A *workflow-related profile* reflects the given context of an individual need expressed by process states, a process history as well as process goals, like the order information in a purchasing process. Concerning these three aspects there are three major tasks with respect to satisfy information needs:



These three tasks consequently build on each other by successively using the existing profiles in order to transform given text sources into user-requested information.

However, profiles also include a user's typical behavior or her presentation likings. Further considerations in this direction are subject of the research community for user modeling (e.g., [4]) and not of this paper.

Profiles can be seen as a special, subjective view on the company's ontology, typically called *organizational memory*. Such an ontology defines the important types of knowledge occuring in the company in an objective matter. It is divided into an *organization ontology* describing the company's processes and its organization structure, an *information ontology* defining which information carriers exist (e.g., all document types in a hierarchical structure) and a *domain ontology*, describing e.g. the typical contents of certain document types [5].

Profiles denote in which parts of the organizational memory a specific user is interested in and also provide the necessary textual knowledge to derive new instances from texts. This information is private and therefore not accessible for other users.

The process part of the user profile is especially helpful when context about the user's actual processes is electronically available. Imagine, e.g., the company's processes are enacted as workflows within a workflow management system. In this case, context can be collected within the workflow management system, stored in a *context pool* and provided to the information assistant. This enables the assistent to accomplish a very dynamic and actual analysis.

The information assistent may accomplish different analysis tasks. Therefore, figure 1 shows a toolbox of IAs which may categorize new incoming documents (concerning the structure part of a user profile), extract relevant information items (dealing with the substance part of the profile), or attach the document to the corresponding process (dealing with the process part). There may as well be other assistants, e.g., for information retrieval or document filtering, but this goes beyond the scope of this paper.

Document sources for information assistants are either new incoming documents (arriving electronically such as e-mail/ html documents or arriving via paper mail or fax) or documents already stored within the company's document management system (DMS). In the letter case, documents have already been registered and their contents are partly accessible by a certain document index. This index relates the documents to the objective organizational memory (e.g., states that a certain document is an invoice of a specific company). In such a case, an informa-

tion retrieval query from a user might be answered without any further analysis just by using the index information.

Within this scenario, a very critical success factor is the effort a user has to put in the generation and modification of her profiles and contexts. Therefore,

- profiles are kept slim, the user is not burdened with defining relationships between her different terms since such relations are already available within the company's ontology,
- IAs must be equipped with means to ease the acquisition of the necessary textual knowledge,
- and context collection must be unintrusive.

As already mentioned, the usage of process context can have a high impact on an IA. First, it allows a more specific assignment of a new document: Without context, a document can only be attached to the corresponding document class denoted in the structure part of a profile (or it can be rejected if the user is not interested). This allows a static assignment where the real interest of the user might not be matched properly. Just imagine, that there is more than one clerk in a company dealing with software invoices. In such a case, a new software invoice is assigned to all of these clerks although only one of them is really interested in processing this document. With context, a document can be assigned to the correct user by relying on the correct class and the correct process. That means, the user only gets the software invoice if she is really the person dealing with the appropriate process.

However, there are situations where either context is not electronically accessible or the costs of context collection do not count. So, both kinds of information supply make sense. Therefore, we will first explain in section 4 and 5 how information extraction and text categorization is accomplished when being based just on profiles. Afterwards, section 6 explains the different scenario of a context-based assignment and extraction.

3. PREPROCESSING OF PAPER TEXT SOURCES

The treatment of paper documents requires some necessary preprocessing steps before a semantic interpretation of the contents of a document can take place. Therefore, a sequence of document analysis techniques is employed:

The analysis of a scanned document starts with low-level image preprocessing such as skew angle adjustment and upside-down detection. Afterwards, segmentation divides the document into geometrically connected components and identifies segments of characters, words, lines, and blocks. Then, text recognition explores the captured text segments, generates character hypotheses by using OCR systems, and merges character hypotheses into word hypotheses. Structure classification takes this given geometric structure to hypothesize the so-called logical objects of a document, e.g. title, author, chapter, etc. Afterwards, the generated word hypotheses are validated by an error-tolerant dictionary look-up. For more detail, see e.g., [6], [7], [8].

4. TEXT CATEGORIZATION

For automated information delivery it is important to know the type of documents a specific user is interested in. The personal view of a user on a document is captured by the structure part of the users profile. The *structure* relates documents to content-based categories which may by itself relate to each other. The mapping of documents to content-based categories is generally done by using text patterns: If a category-dependent text pattern matches in a document, this document category is assigned to that document.

In our scenario, text categorization allows to filter documents of those categories which are interesting for a speficic user. When retrieving information, personal categories can be used as an additional construct of the query langauge: Only documents that in addition to the traditional query have a specific category label assigned are returned to the user.

A well known system for text categorization is the TCS-system which is used with great success in categorizing financial news [9]. A typical pattern in TCS is

(**and** gold
(**and** (**not** medal) (**not** jewelry)))

This pattern matches in every documents that contains "gold", but none of the words "medal" or "jewelry". It can be used, e.g., to find articles dealing with gold in which gold is not a good. More elaborated pattern constructs generally also allow tests on words sequences and word properties.

A problem of category assignments by patterns is the manual effort needed in order to define the patterns. Learning approaches for text categorization try to solve this problem by replacing the manual edited patterns by example documents [10]. The input of such learning systems are documents with correct category assignments. Using these examples, new documents can be assigned to the respective categories automatically. Most learning approaches for text categorization rely on statistical techniques with linear classifiers [11]. A very attractive alternative are approaches that learn rules relying on text patterns. Until now, only a few rule learners have been applied to text categorization (examples are SWAP-1 [12] and Ripper [13]). Nevertheless, rule learners offer some practical advantages:

- they produce *very compact classifiers*,
- which *are easy to understand and to modify* by humans (if, e.g., a manual fine tuning is needed), and
- the classifiers are *portable* in the sense that they can be used to query nearly any IR search engine.

Because of these advantages we recently focused on learning algorithmus that use text patterns as classifiers. Learning of such patterns can be seen as a search problem within the space given by the pattern language. The optimization criterion is the effectiveness of a pattern with respect to a specific category on unseen documents. The fact that effectiveness should be optimal on *unseen* documents makes this task a non-trivial one since reliable estimates for future performance of patterns have to be found based on the examples. Patterns in the language we use are all of the form (or k1.. kn). The ki (also called complexes) may be single word tests, but they may also test on more complex document properties such as the occurrence of words with specific properies within some distance and order in the documents.

Learning of patterns is done using a separate-and-conquer algorithm, which can be described roughly as follows:

```
INPUT: positive and negative example
       documents for target category
OUTPUT: pattern for describing positive
        documents

set pattern to the empty disjunction
repeat
  1) find complex which describes a subset of
     the positive examples with high precision
  2) add this complex to the pattern as new
     argument of the disjunction
  3) remove all positive documents covered
     by this complex from examples
until (no sufficently good complex can be found
     anymore or no more positive example)
```

Finding the complexes in the above algorithm is done using sophisticated search heuristics which we describe by a declarative strategy language.

5. INFORMATION EXTRACTION

Text categories are used in order to identify documents relevant for a user within a large document collection. Often, the user is only interested in some information or *substance* represented within the documents, e.g., in the case of invoices the price to be paid. Which information he is interested in depends on his personal profile: Some users might be interested in the price, others also want to know the items, etc. The extraction of information from documents is accomplished with pattern languages similar to those used in text categorization. In contrast to text categorization, pattern languages in information extraction have to provide variables for binding extracted text parts and the order of the words is crucial: In order to detect the boundaries of information to be extracted, properties of the words within the information and within the surrounding text are important. For the domain of office automation, more details on information extraction can be found in [14]. Simple patterns for extracting prices and discount rates can be written as:

```
"total amount:" ?<NUMBER> currency
?days "days:" ?<NUMBER> "%"
```

The first pattern extracts every number which is enclosed between the string "total amount:" and some currency (currency counts as a hypernym). The second pattern extracts discount rates and the corresponding maximum number of days the payment should be done.

As in text categorization it is very interesting to learn patterns capturing the substance automatically by examples [15]. Examples in information extraction are documents together with the information to be extracted. In parallel to text categorization, extraction patterns can also be learned using a separate-and-conquer algorithm which learns disjunctions of complexes. Learning complexes for information extraction can also be described as a search in the space of valid pattern expression. Due to the variables in extraction patterns, the learning of complexes for information extraction is more complicated than in text categorization. Nevertheless, it can also be described using a strategy language.

6. IMPACTS OF CONTEXT: PROCESS IDENTIFICATION AND SUBSEQUENT INFORMATION EXTRACTION

We will now explain shortly how information supply works when process contexts can be used for the textual analysis of new documents. More details can be found in [16] and to the best of our knowledge, no other approaches exist within that area.

In this scenario, the first task is to assign the document to the correct process (that means, choosing one out of a set of "waiting" workflows) and the subsequent, second task is to extract exactly the information which is required by the user within this specific process.

Looking a little bit closer, such workflows (as enactments of processes) are characterized by heterogenous documents which belong to one common process and arrive in a chronological order. A typical example is an insurance process with initial applications for contracts, changes in the policies, annual invoices, and damage claims. Another example are business trips where the traveller has to fill out an application, the application must be confirmed, some invoices, e.g., for plane tickets, have to be payed in advance and, finally, several receipts must be accounted for. Certainly, there are a lot of similar examples but we will use a purchasing process within a company as a basis for our examples.

In order to support IAs, the workflow instance transfers context information to the IA. Having finished its task, the IA hands over the data requested and consequently satisfies the user's information need within the workflow instance.

Information Assistent for Process Identification

Workflow activities within a running process are sometimes triggered by events occuring outside of the workflow. In such a case, the process is in a waiting state until this certain event occurs. Imagine for example that an order has been written to a supplier. Afterwards, the corresponding workflow is waiting for a confirmation of this order by the supplier. Thus, this workflow can be satisfied by observing incoming documents and by assigning the appropriate one to the waiting workflow.

The IA for process identification basically calculates the best match of information contained in documents with the corresponding data available in workflow instances. To achieve this, the relevant data in workflow instances have to be collected for two different reasons:

- First of all, they build one kind of input for the instance match. Imagine there are a lot of open instances and a new document has to be assigned to them. In this case, the more data we have about the instances and the expected document (e.g, document type, sender, products mentioned and even possible references to other events within the same instance) the more accurate the match will be accomplished.
- On the other hand, these same data specify which information should be derived from the document.

In this scenario, several activities within different workflow instances state the current context of these instances whenever new context is available.

This context subsumes two different kinds of information: The first kind of context is delivered by the workflow itself and therefore called workflow context. It consists of workflow-rele-

vant data (e.g., the process number), data coming from the workflow's audit trail (e.g., reference data to preceeding documents) and data from the workflow models (e.g., the name of the next event which provides the return address for the workflow). The second kind of context stems from applications triggered by the workflow (e.g., from the text processing application for writing an order) and is called task context (e.g., the supplier's address).

All these context data are transferred to the *context pool*, a database available outside the WfMS. There, the context is stored until an event occurs in a workflow instance which implies a response by a document, i.e., the user within the workflow has an information need which will be satisfied by an incoming document related to the event or workflow. This information need is stated by informing the IA for process identification and by handing over its context. This whole collection is called an *expectation*.

It describes content and meaning of the expected document, additional information need such as a list of data which has to be extracted, and some administrative data in order to identify the workflow after a process determination was successful. In order to allow the IA to interpret the expectation, it has to be connected to the user profiles and the company's information and domain ontology which describes the structure and content of the documents of the domain under consideration.

Therefore, such an expectation is generated by using the context pool as input for an inference engine. The engine uses predefined transformation rules which relate the context within the context pool to possible contents of an expected document. These rules have been defined once at the workflow's construction time and can be used for every user profile.

Thus, these rules transform the data scheme used in the workflow and attached applications into the domain and information ontology and by that, to the user profile as well. For instance, a rule states that the receiver of the (already known) order is the sender of the expected corresponding invoice.

The IA for process identification is triggered by any incoming document. Then, the identification of the corresponding process takes all expectations of all workflow instances into account.

Therefore, the IA for process identification first starts the IA for text categorization who categorizes the document according to all structure profiles for all users. Afterwards, the hypothesis for a certain document type is used to restrict the number of appropriate expectations. By that, a certain set of expectations remains as possible process candidates for the document. In the next step, the IA for process identification invokes the IA for information extraction. It hands over all information items mentioned in the remaining expectations. For these items, the information extraction IA retrieves the corresponding text patterns from the substance part of a profile and instantiates these patterns with the expectations contents. To clarify this, imagine that the substance pattern is a simple pattern which determines how to extract a price (`"total amount:" ?<NUMBER> currency`). The price expectation now gives a concrete price (e.g., 105 USD), so the specialization of the pattern is `"total amount:" ?105 "USD"`. As a consequence, the information extraction IA tries to extract the specialized pattern.

In this way, certainties are determined for each possible information item in each expectation. Finally, these certainties are accumulated to determine the best expectation and thus, the best process match is calculated. The result of process identification is a unique process identifier and, of course, the name of the document which is assigned to the process. The expectation of the correponding instance is deleted after the process identification has been verified within the instance.

This ends the process identification scenario. Data kept within the context pool are deleted when the corresponding process instance has come to an end.

Information Assistent for Information Extraction

In this context, only one process is of further interest. Within this process, the context pool has still been updated after a process identification request whenever new context is available. Further information needs are now stated by subsequent activities within the workflow. In such a case, a new expectation is generated as described above. The request now includes a list of identifiers according to the domain ontology which have to be extracted from the attached business letter.

In contrast to our process identification scenario, the information extraction IA now analyses a document already available within the company's DMS. If a certain information item has already been extracted, it can be directly retrieved from the document index. Otherwise, extraction is accomplished, the appropriate result is handed over to the user and it may as well be entered into the document index.

One Remark concerning Context Collection

Note that the collection and the delivery of context is done in a way that it is not noticed by the user. Only at the construction time of a process, the workflow has to be extended to achieve the context collection. The only thing a user has to do is to integrate the corresponding process within his profile if this is not supported by the workflow management system.

7. CONCLUSION

We have proposed a solution for the individual information supply of users within a company environment. This solution uses user profiles, a company's organziation memory and process contexts as declarative analysis knowledge sources and establishes intelligent information assistants for text categorization, information extraction and process identification. The system is able to process paper and electronic documents.

Profiles are a very helpful means to express individual and subjective views of humans on the company's information world. Thus, they allow an individual information supply. Whenever the personal context is available as well, this can further improve and guide the analysis to allow an even more specific information supply.

Our concept can be easily extended for other information assistants. It also allows an easy adaptation to new tasks and document types since all IAs include acquisition mechanisms.

Our future work deals with the extension of context usage for retrieval processes. When looking at the information retrieval task, we also have to deal with queries which must somehow be related to the information system environment.

Acknowledgements

We thank our colleague Andreas Abecker for a fruitful discussion and some valuable hints concerning the contents of this

paper. This work has partly been funded by the German federal ministery for education and research under the grant 01 IW 807.

8. REFERENCES

[1] A. Abecker, A. Bernardi, H. Maus, M. Sintek, C. Wenzel. Coupling workflow technology with intelligent information supply. Elsevier's "Knowledge Based Systems" Journal, Special Issue on AI in Knowledge Management, June 2000.

[2] J. Budzik, K. J. Hammond. User Interactions with Everyday Applications as Context for Just-in-time Information Access, Intelligent User Interfaces 2000, ACM Press.

[3] J. Budzik, K. J. Hammond. Watson: Anticipating and Contextualizing Information Need, Sixty-second Annual Meeting of the American Society for Information Science,1999, Information Today, Inc., Medford, NJ.

[4] http://www.um.org/conferences.html.

[5] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, M. Sintek. Toward a technology for organizational memories. IEEE Intelligent Systems 13(3).

[6] S. Baumann, M. Ben Hadj Ali, A. Dengel, T. Jäger, M. Malburg, A. Weigel, C. Wenzel. Message Extraction from Printed Documents – A Complete Solution. Fourth Int. Conference on Document Analysis and Recognition (ICDAR 97), Ulm, Germany, August 18-20, 1997.

[7] A. Dengel and K. Hinkelmann. The Specialist Board – A Technology Workbench for Document Analysis and Understanding, 2nd World Conference on Integrated Design and Process Technology (IDPT '96), Austin, TX, USA, December 1996.

[8] S. Baumann, M. Malburg, C. Wenzel. May document analysis tools bridge the gap between paper and workflow? A critical survey. First IFCIS Int'l Conference on Cooperative Information Systems, Brussels, Belgium, June 1996.

[9] P.J. Hayes, S.P. Weinstein. Construe-TIS: A System for Content-Based Indexing of a Database of News Stories, Innovative Applications of Artificial Intelligence 2, AAAI Press / MIT Press, 1991.

[10] M. Junker, R. Hoch. An Experimental Evaluation of OCR Text Representations for Learning Document {C}lassifiers, International Journal on Document Analysis and Recognition, Vol. 1, Number 2, 1998.

[11] D.D. Lewis, R.E. Schapire, J.P. Callan, R. Papke. Training Algorithms for Linear Text Classifiers, 19th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval (SIGIR 96), Zurich, Switzerland, August 18 - 22, 1996.

[12] C. Apte and F. Damerau and S. Weiss. Towards Language Independent Automated Learning of Text Categorization Models, 17th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval (SIGIR 94), Dublin, Ireland, July 3-6, 1994.

[13] W.W. Cohen. Learning to Classify English Text with ILP Methods. Advances in Inductive Logic Programming, IOS Press, 1996.

[14] C. Wenzel. Integrating Information Extraction into Workflow Management Systems. Natural Language and Information Systems Workshop (NLIS/DEXA 98), Vienna, Austria, August 24-28, 1998.

[15] S. Soderland. Learning Information Extraction Rules for Semi-Structured and Free Text. Machine Learning, 34(1-3), 1999.

[16] C. Wenzel, H. Maus. An approach to context-driven document analysis and understanding. Submitted to: Fourth IAPR International Workshop on Document Analysis Systems (DAS 2000).