PROFILE - A Multi-Disciplinary Approach to Information Discovery

J. Simons[†], A.T. Arampatzis[⋆], B.C.M. Wondergem[⋆],
L.R.B. Schomaker[†], P. van Bommel[⋆], Th.P. van der Weide[⋆],
C.H.A. Koster[⋆]

[†]
Nijmegen Institute for Cognition and Information
University of Nijmegen, Nijmegen,
The Netherlands

[⋆]
Computing Science Institute
University of Nijmegen, Nijmegen,
The Netherlands

Abstract

This document describes the PROFILE project, a multi-disciplinary project aiming at the development of a proactive information filter for dynamic information environments. The goal of this paper is threefold. First, it evaluates the suitability of an agent-based architecture for the PROFILE project. Second, this article provides an overview of the research done in the PROFILE project. Third, it describes the integration of this research, which has led to the implementation of a prototype. The architecture and workings of the prototype are illustrated.

1 Introduction

The PROFILE project, started in 1996, aimed to decrease the problems caused by information overload. Its main goal was to develop an effective agent as intermediary between information sources and information users in the context of the World Wide Web. The effectiveness of the PROFILE system was to be enlarged by going beyond keyword-based approaches. The term Information Discovery ([35]) was introduced to describe the synthesis of information retrieval and information filtering.

Two research groups of the University of Nijmegen participate in the PROFILE project: The cognitive ergonomics group of the NICI (Nijmegen Institute for Cognition and Information) and the IRIS (Information Retrieval and Information Systems) group of the subfaculty of computer science. The researchers in the two groups each have a different background. The PROFILE project thus integrates several viewpoints from multiple disciplines and organisations.

This article sets out to do three things. First, it evaluates the suitability of an agent-based architecture for the PROFILE project. The organisation of the PROFILE project - decentralised research done by groups with different styles of working and cultures - calls for a flexible architecture. We argue that an agent-based architecture supports the required extensibility. However, within PRO-

FILE, agents are not viewed as a central research issue but as a tool to implement and rationalise our work.

Second, this article provides an overview of the research done in the different components of the PROFILE project. The PROFILE project divided the research themes over several components. The aims of these components are described. Their results reveal that different levels of progress were realised.

Third, it describes the integration of the research that has led to the design of a prototype system. Issues concerning the implementation of the prototype are elaborated. In addition, the types of agents used in the PROFILE system are described. Furthermore, the functionality of the prototype is illustrated.

This article has the following structure. Section 2 advocates an agent-based architecture for the PROFILE system. It also describes the basic setup of the PROFILE project as a cooperation between four components. Section 3 describes the research done in each of the components. Section 4 describes the prototype, integrating the individual findings. Section 5 compares the PROFILE project with several related projects. Section 6 provides concluding remarks and suggestions for further research.

2 The PROFILE Multi-Agent System

In this section, we first examine constraints on the design of the PROFILE system. Then, we advocate that these constraints suggest an agent-based approach. Finally, we show that functionally decomposing the PROFILE system results in a multi-agent system.

2.1 Constraints on PROFILE

Several constraints arise from the nature and goals of the project and the intended use of the system. These constraints can partly be derived from the organisational context of the project and partly from the desired functionality of the system.

The organisational context in which four independent groups from two different institutes had to participate in one project posed the following limitations:

- The project is essentially a research project. Therefore, provisions should be available to cope with differences in progress of the project parts. It also implies a wish of every part to be constrained as little as possible by other parts.
- Different parts of the project are used to different implementations tools. This calls for special attention on integration.
- The multi-disciplinary project is carried out by different organisations with different styles of working and different research cultures.

Next to these constraints, other constraints are imposed by the desired characteristics of the PROFILE prototype:

- The general required proactiveness of the PROFILE system – It should be able to react on both information sources and information users. – implies that every part of the system has to be able to take the initiative.
- The system is intended to work partly on the user's work station and partly on a central service which proactively searches for different users. PROFILE

thus aims at a flexible distributed system that goes beyond a simple client-server model.

- The system should be easily extensible, enabling different instantiations of some functionality to be readily incorporated.

These constraints call for a flexible architecture consisting of several independent modules that cooperatively implement an Information Discovery (ID) system.

2.2 Properties of Agents

An agent-based approach addresses many of the constraints described in the previous subsection. Especially, the following characteristics of an agent-oriented paradigm are useful:

Autonomy. Autonomous agents have control over their internal state and planning of their actions. Implementing the project components as autonomous agents alleviates problems of brittleness caused by interdependencies. This prevents general delay of a project development if a single part encounters problems.

Elaborate Communication. Communication between components is directed toward a flexible form of cooperation. Agents can negotiate and form teams to adjust to the needs of a situation. This suggests a more powerful approach to communication than simple function invocation. Because the precise functionality of each component was not known at the start of the project, the communication protocol between the different components had to be sufficiently general. This includes several possible ways of representing information needs, situational factors, and general characteristics. The protocol also prescribes how components should react to the different kinds of messages.

Proactiveness. This property enables an agent to initiate actions on its own. This property is valuable when designing a information discovery agent because its proactivity can be achieved by the proactivity of the sub-agents.

Concurrent Processing. The dynamic and distributed nature of the PROFILE system introduces asynchronic events, e.g., query input or document filter output. Synchronous flow of control thus is impractical.

Many other properties have been assigned to agents (see e.g. [38]). However, the goal of this project is not to elaborate on the notion of agency, but merely to use valuable properties of agents in an information seeking environment.

2.3 Multi-Agent System

The multi-disciplinary nature of the system calls for a functional decomposition. The original design identified four different components for a proactive information filtering task. Each component was implemented as an agent, having the abovementioned characteristics. Therefore, the PROFILE system essentially is designed as a multi-agent system. As an extra advantage, a multi-agent system goes beyond the client-server model by allowing to postpone the decision about where to do the actual processing. In addition, multi-agent systems are easily extensible. One can easily create different agents for the same task and experiment with it. Extensibility is of great importance in open environments like the Web. The following four components were identified:

User Interaction. This part delivers the tools for the users to formulate their information need and to react upon the delivery of documents.

User Modelling. This part derives representations of information needs and users, and constructs optimised queries or profiles.

Matching. This part of the project compares documents with user interests. In the matching component, the set of rendered documents is identified.

Language Processing. This part of the project analyses documents in order to deliver a representation of their contents.

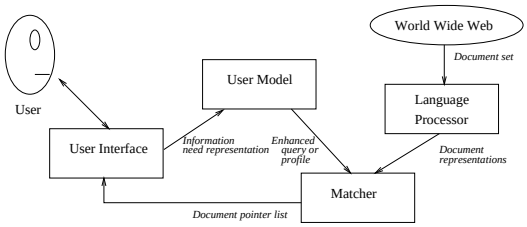

Figure 1: Conceptual PROFILE setup. Arrows denote the flow of information.

Every part of the project aimed at delivering an agent for the specific subtask. The flow of information between the components is depicted in Figure 1. The user interacts with the PROFILE system in order to specify his/her information needs. Relevant information concerning the formulation process is sent to the user modelling module. The user modelling component maintains a representation of the user's interests, their situational factors, and general characteristics. This is done by observing the actions of the user or by direct interaction. The user module delivers optimised queries to the matching component. The parsing module indexes available documents. Representations of user interests and document contents are passed on to the retrieval component, which establishes relevance estimates in order to distinguish the set of documents to be rendered. Rendered documents are presented to the user, who optionally gives feedback on the results. The feedback is used to adjust the user model and starts a new information cycle in the PROFILE system. The four major components of the PROFILE project are discussed more elaborately in the next section.

3 Research in PROFILE

Each component of the PROFILE project has its own field of interest, focussing on a subtask of the system. It should be noted that the topics of interests are not always strictly divided over the components.

3.1 User Modelling

The user modelling component focuses on an investigation of the use of domain knowledge in the formulation of a query. This is not a new idea. Domain knowledge is applied in document indexing (e.g. [7] and [28]), matching (e.g. [27]), and query formulation (e.g. [31], [14]). Another knowledge based approach is using a semantic network in which nodes stand for words ([1]).

The research in this component deviates from other applications of domain knowledge in query augmentation in that it investigates whether the richness of knowledge representation can influence the success of query expansions. The idea is to use domain knowledge represented in so called *ontologies* ([15]). These knowledge structures usually represent their domain in a taxonomy of concepts with each concept represented as a frame with slots, slot values, and restrictions on these

slot values. More elaborate knowledge structures allow for more different types of expansions. More different expansions enable more possible ways to improve a query. Therefore, our hypothesis is that elaborate knowledge structures can improve ID performance.

ID with domain knowledge represented in an elaborate knowledge representation language poses two separate problems. A first question is how an agent that uses this kind of domain knowledge can be realised using current technology. The second question is how domain knowledge should be applied to improve a query. The following two sections sketch the research that is done to answer these questions.

3.1.1 A Knowledge-Based Query Formulation Agent

A common-sense criticism against a knowledge-based approach to ID is that it would cost too much resources to provide an agent with a new knowledge base in order to do information filtering or retrieval. Therefore, in order to turn this idea into a useful technology, the agent should be able to re-use existing knowledge bases. One attempt to facilitate the re-use of knowledge was made by [24]. Their solution comprises an editor which can be accessed by an internet browser. A library of functions and objects enables programs to access these ontologies in a uniform way. An OKBC[1] client can access OKBC-servers on the Internet to query ontologies about concepts, slots, and values or facets on those slots ([11]).

Figure 2 shows the design of knowledge based agents in the PROFILE approach. The idea is implemented in an applet in which a user can select an information need represented in ONTOLINGUA ([15]), expand

[1]Open Knowledge Base Connectivity Protocol

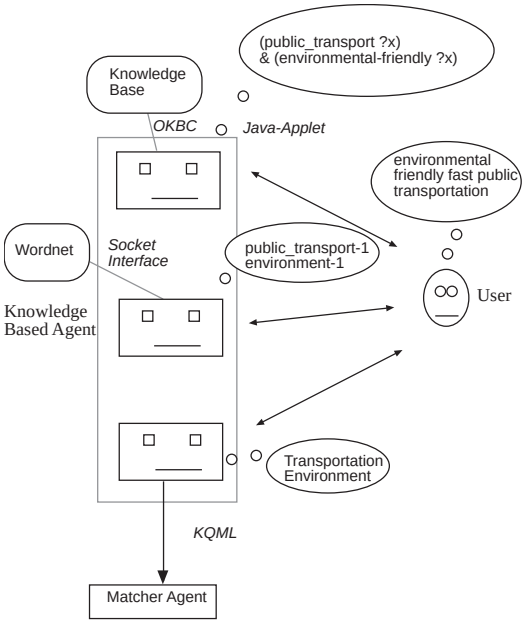

Figure 2: Overview of a prototypical knowledge based agent. Arrows denote flow of information. Italics indicate how it is implemented. The ovals give an example of what is represented in each component.

it to add concepts, translate these concepts to synonym sets in WORDNET ([20]), and finally, use expansions in WORDNET to create the final query. The user can interact with the agent at three levels. The current implementation of the knowledge-based agent only expands when the user asks for it. The applet extracts its domain knowledge from a central server using OKBC. A socket interface to WORDNET was developed to acquire lexical knowledge.

3.1.2 Query Expansion Research

The prototype described in the previous section turned out to have some disadvantages. First, the communication layer between the knowledge base server and the user agent was quite complex. This made the construction of new strategies quite tedious. Second, ONTOLINGUA is not written with the purpose to do inferences. Therefore, no good inference system has been written in ONTOLINGUA. However, inferences are necessary in order to make use of the more elaborate knowledge represented in this formalism.

For these reasons the research upon domain based query augmentation was done using LOOM ([19]) and a socket based interface to SMART ([25]). The JFACC ontology[2] ([29]) was used as domain knowledge for an experiment. Representations of fifteen information needs in JFACC concepts with differring degrees of complexity were constructed for this experiment. A collection of 1500 documents was gathered from websites that contained information specific to that domain. The non-expanded query was used to collect 100 documents for each query. The documents were scored manually. Then, each query was expanded using different strategies that used different aspects of the knowledge representation. E.g, a strategy that adds hyponyms of a concept views the ontology as a pure taxonomy while a strategy that expands on slots and slot values uses more of the power of the knowledge representation formalism.

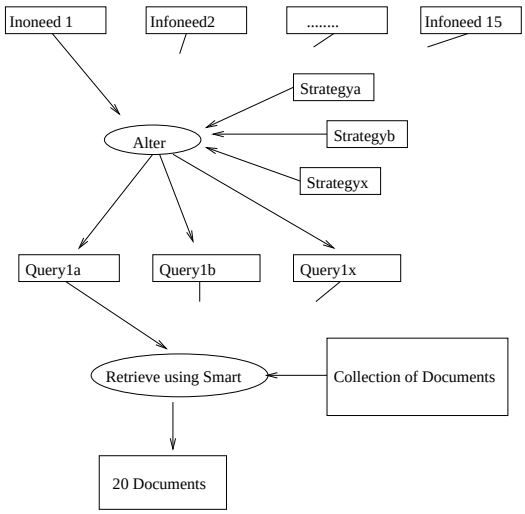

Figure 3: A general sketch of how a run is carried out for a single query and a single information need. Each possible route though the figure delivers one result.

Figure 3 shows how each run was carried out. Every information need was altered by each strategy to produce a set of queries. Each of these queries was then used to retrieve twenty documents.

The results ([26]) suggest that the answer to when and how domain knowledge can help in expanding a query is not an easy one. For example, the concepts in the ontology turned out not to be as richly interconnected as was originally assumed. A lot of the concepts – especially the more complex ones – could not be expanded by strategies that depended on connections beyond a taxonomy. A set of experiments that try to use more of the structure available in an ontology is being prepared to

[2]Joint Force Air Campaign Commander

shed more light on this issue. Equally interesting is the use of learning techniques to adjust the expansions in a filtering context.

3.2 Language Processing

The Language Processing component is responsible for providing representations for textual information objects (documents), a process widely known as *indexing*. Indexing in PROFILE goes beyond the use of simple keywords to characterise documents. In an attempt to capture more of documents' content, we employ natural language processors and linguistic resources.

Initial small-scale experiments with the IRENA system ([4, 3]) showed the promising aspects of lexical and morphological expansion of keywords in improving the effectiveness of retrieval systems. In the same study, the Noun Phrase Co-occurrence criterion, i.e., co-occurrence of keywords or their synonyms or morphological variants of them in the same noun phrase, was applied successfully in determining whether keywords are semantically related in a more beneficial way for precision than proximity search. Although low recall was obtained, at any rate, the Noun Phrase Co-occurrence criterion can be used for relevance feedback.

In a further investigation, we extended the traditional Keyword Retrieval Hypothesis to a Phrase Retrieval Hypothesis, upon which we have built a linguistically-motivated indexing scheme ([5, 6]). Two types of phrases have been considered for indexing, these are: the noun phrase including its modifiers, and the verb phrase including its subject, object and other complements.

We defined an abstract representation of these phrases suitable for indexing. Full linguistic parse-trees contain too much linguistic detail, most of which is unnecessary for indexing, as such details reflect mostly the syntactic description of the natural language used rather than the intended meaning. Therefore, we settled for less than full linguistic parsing, eliminating structures which can be assumed not to be beneficial to indexing. Figure 4 gives an example of the level of detail of our syntax analysis. Arguably, this is a form of lighter analysis than full parsing while a reasonable amount of structural information is still retained.

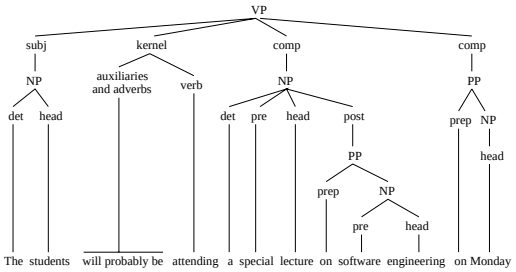

Figure 4: Light parsing for indexing purposes.

Although phrases can be used in their literal form as terms, the performance will be inferior to that of keywords. On the one hand, phrases achieve better precision, but on the other hand recall will be too low, because the probability of a phrase re-occurring literally is too low. To deal with this sparsity of phrasal terms, we introduce *linguistic normalisation*. The goal of normalisation is to map alternative formulations of meaning to a normalised form called *phrase frame* (Figure 5). We distinguish between three types of normalisation: *morphological, syntactical*, and *lexico-semantical* normalisation.

Morphological normalisation has traditionally been performed by means of stemming (non-linguistic suffix stripping). Taking into account the linguistic context, we follow a more conservative approach called *lemmati-*

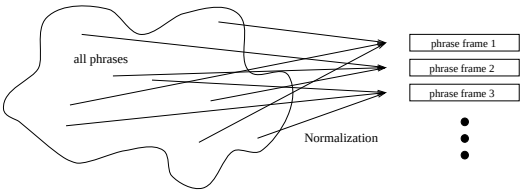

Figure 5: Linguistic normalisation

sation which prevents many errors of stemming. Lemmatisation can be seen as part-of-speech-directed stemming. It reduces verb forms to the infinitive, inflected forms of nouns to the nominative singular, comparatives and superlatives of gradable adjectives to their absolute form.

Syntactical normalisation is based on the linguistic principle of *headedness*: any phrase has a single head. Thus any phrase can be transformed to a canonical form (an ordered relation between its components): head first, followed by its modifiers. The head gives the central concept of the phrase and modifiers serve to make it more precise. Conversely, the head may be used as an abstraction of the phrase, loosing precision but gaining recall. Heads and modifiers may recursively contain phrases. A number of such syntactical transformations from the phrase domain to head-modifier domain have been investigated.

Lexico-semantical normalisation is based on certain relations which can be found between the meaning of individual words, such as *synonymy*, *antonymy*, *is-a*, and *part-of*. Our approach combines thesaural information from WordNet with statistical word co-occurrence data to establish such word relations. Two possibilities are explored for lexico-semantical normalisation. The first is *lexico-semantical clustering* which reduces closely related words to one word cluster, and the second is *fuzzy matching* which introduces a semantical similarity function between words into the retrieval function.

Parts of this linguistically-motivated indexing model are still under investigation, tuning, and evaluation. However, initial experiments have yielded promising results [2], suggesting that we are not far from finalising a model which should help to overcome the known and long-survived problems of bag-of-words representations.

3.3 Matching

The matching component, also called retrieval component, is responsible for a comparison between document contents and user interests. Therefore, an important focus in the retrieval component are metrices for expressing the similarity between descriptors. This yields relevance estimates for documents with respect to user queries and profiles, allowing the distinction between more and less relevant documents.

The language of index expressions ([9, 10]) is considered in the retrieval module. Index expressions are constructed from terms (e.g. keywords, concept names, or denotations of attribute values) and connectors, representing relations between terms in the form of prepositions and gerunds. Index expressions feature a simple linguistically motivated refinement mechanism, sometimes referred to as headedness or concept refinement. An advantage of index expressions is that they support the construction of networks that are suitable for navigational formulation techniques. This allows users to formulate their information need by stepwise refinement in a navigational network. A thorough formal basis for index expressions is provided ([33]). In addition, several similarity measures for index expressions were devised, focussing on different properties relating to subexpressions ([32]). These were evaluated in the context of

navigational networks for index expressions. A dynamic retrieval system was built, supporting navigational query formulation for searches on the WWW ([37]).

Special attention is given to a tractable approximation of noun phrases, coined Boolean index expressions (BIEs) ([34]). BIEs can be constructed by the refinement mechanisms from index expressions combined with logical concept building, by inclusion of Boolean operators. An example BIE is given in figure 6. Next to sufficiently expressive, BIEs are tractable and compact.

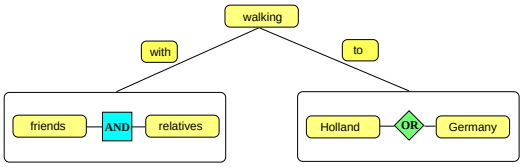

Figure 6: Example BIE.

From a user oriented point of view, compactness is beneficial in formulating the information need. Compactness of BIEs is yielded by allowing nested logical constructors. This effectuates subexpression sharing. If an information need involves several related concepts, subexpression sharing saves space (for representing the need) and time (for users to read the representation). The nature of the compactness of BIEs was studied, revealing that exponential compactness can be reached ([36]).

A normalisation function for BIEs was devised to reduce the syntactical variety yielded by Boolean operators. Normalisation consists of zipping up the diadic operators, providing BIEs in so called propositional form (see figure 7). This form consists of a logical part, containing the diadic operators, and an atomic part. Advantages of BIEs in such form are that operations on them can be specified in terms of (slightly modified) functions on classical index expressions.

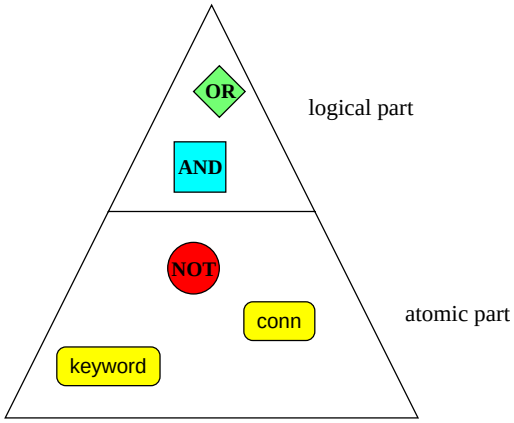

Figure 7: General setup of propositional form.

In order to assist information searchers during formulation of their information need, two tools for constructing and adaptating BIEs were described. The first tool combines direct manipulation of BIEs with navigational formulation for classical index expressions. Index expressions that are encountered during navigation may be included in the query or profile at hand and can be manipulated by direct actions. The second tool shows how relevance feedback can be incorporated with BIEs. This tool guarantees good control over the form of the constructed BIE by separately modelling positive and negative feedback.

Combining filtering and retrieval in a single paradigm, as aimed at in the PROFILE project, may result in an imbalanced setting. In order to guarantee fairness in ID, information brokers need to be designed carefully. In the retrieval module, the imbalance was found to stem from the dual nature of retrieval and filtering ([35]). Since this duality

is examined within cumulative communication, i.e., communication that is required to go via brokers, it is coined cumulative duality. An instrument was designed to explicitly state the influence of criteria concerning cumulative duality on the required role of information brokers. Example criteria concern privacy of user interests, partial knowledge about broker services, and the dynamics of user interests and offered information.

3.4 User Interface

The user interface of an information filtering system is arguably its most complex component. This is due to (a) the real-time character of such an interface, (b) the convergence of input and output streams at a single point in time and space, (c) the required degree of user control over the internal computational modules of the information filtering system, and (d) the large design space as regards the information-rendering methods for a wide range of target platforms. However, most of the effort in the PROFILE project has been invested in the development of the information retrieval and knowledge representation techniques.

The user interface will eventually be realised taking a number of requirements into account. For an easy development, interface components will be designed according to the characteristics of the chosen multiple-agent architecture. The complexity of interface requires, e.g., that a distinction be made between **rendering** functions and **query**-related functions.

Ideally, in an information filtering context, the user interface will also contain a dedicated 'intelligent user agent', which is capable of maintaining dialogs with the user for the specification of user profiles. By necessity, such a user agent requires knowledge representations which are focused on (a) real-world semantics, (b) self-capability descriptors concerning the system's beliefs as regards its intrinsic information filtering functionality, and (c) pragmatics of search in large real-life digital information sources.

What has been realised until today is of a far more modest nature. A collection of JAVA routines has been developed which allow for the basic functionality required for information filtering: user enrollment, specification of interest profiles using keywords, and the presentation of dynamic HTML pages on the basis of JAVA servlets. This package uses JATLITE conventions as described before. However, this preliminary setup of a user interface is rather basic and is currently being evaluated from the point of view of current user-interface guidelines in cognitive ergonomics.

Preliminary results ([30]) indicate that the user interface in an information filtering context introduces a number of new concepts which will be unfamiliar to the users of the common interactive search engines as these are available on the World Wide Web today. A number of user-interface metaphors have been proposed in other groups, such as 'sending out a dog on a hunt for information' or 'the personalized and dynamic newspaper'. However, a generally acceptable solution remains to be realized at the user interface level of any information filtering system.

4 Current Prototype

This section describes the current implementation which is based upon the research done in the previous section. This implementation uses very basic versions of the developed techniques and focuses on the communication that occurs between the different modules. We describe the different agents in the PRO-

FILE system and sketch an example to illustrate a possible run with the current prototype.

The JATLITE ([16]) architecture was chosen because it meets the constraints described in Section 2. Agents are implemented as JAVA threads, located in different virtual machines. Communication exploits KQML ([13]) strings sent through TCP sockets. A central router receives messages from every agent and redirects them to the target agent. Because KQML does not specify content, the PROFILE Negotiation Language (PNL) was used. PNL extends KQML in order to specify information needs, documents representations, and user evaluations.

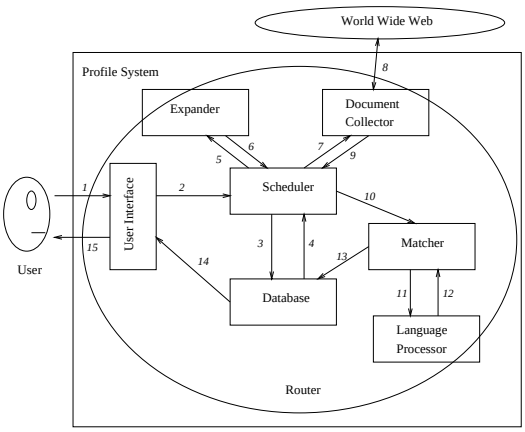

Figure 8: Intended cooperation between the agents in PROFILE for the case of interactive query processing. Arrows denote possible communication.

4.1 Agents in the Implementation

For the implementation, our original division of four different agents had to be refined. The implementation involved seven different agent types. First, the user model part was divided into a part that stores the user preferences, and a part that actually delivers the optimised query. Second, proactive searching on the Web suggested a distinction between a module that actively searches the Web to deliver a preselection of potentially relevant documents and a module that indexes these documents. Table 1 summarises the different agent types. They correspond to the ones in Figure 8. Different instantiations exist for the Document Collector. Each instantiation creates an initial document stream from a broad query to a search engine. Currently the prototype may contact HOTBOT or ALTAVISTA. All agents are embedded in the JATLITE framework.

4.2 Functionality of the Implementation

The agents in the PROFILE system cooperatively realise the three main tasks:

Query Processing. When a user specifies a query in the user interface, the resulting documents are retrieved by the PROFILE system. The flow of control is shown in Figure 8. The Scheduler gets the initial request (1, 2) via the User Interface. It asks additional user information (3, 4) to the Database, optionally expands (5, 6), and sends the query to the Document Collector (7). This agent then retrieves an initial set of documents by doing an http-request to a publicly available search engine (8). The resulting documents are then sent back to the Scheduler (9) who sends them to the Matcher for filtering (10). The Matcher forwards documents to the Language Processor (11), who parses them

and returns the corresponding characterisations (12). The Matcher then performs the matching and delivers the document to the database if the similarity value is above the threshold defined in the information need (13). After putting it in the Database, the document can be rendered to the user (14, 15).

Document Filtering. Upon arrival of new documents, the Scheduler contacts the Database to obtain information need representations. It then forwards these to the matcher. Filtering then proceeds as query processing.

Proactive Filtering. A stream of documents is generated upon an initiative of the Scheduler who sends an initial broad query. Proactive filtering then proceeds as document filtering.

The PROFILE prototype forms a testbed for new results and techniques of the components in our project. Therefore, the prototype is not a final product, but is constantly under (re)construction when new material becomes available.

5 Related Work

Information discovery receives a lot of research attention. In this section, we provide pointers to several related projects. We acknowledge that this overview is far from complete. However, it provides insight in the diversity of aspects of information discovery the projects concentrate on.

5.1 Medoc

The Medoc (see e.g. [12]) project provides a multi-agent architecture for disemmination of scientific documents. Medoc is a cooperation between several universities and pilot institutions in Germany. As implementation, the Medoc Service, a distributed information service for Computer Science literature, was built.

Medoc identifies three major areas of work. First, a shift from printed material toward electronically available documents, in particular scientific journals, is paid attention to. Second, Medoc builds effective bibliographic databases and delivery services. In addition, access and usage structures for protected information are enhanced. Third, a common user interface unifies the search in distributed and heterogeneous information providers.

Like the PROFILE system, a multi-agent system forms the distributed architecture. However, the initial Medoc system is domain specific, considering documents from the area of Computer Science. Furthermore, Medoc does not concentrate on the use of linguistic technology for the disemmination of information.

5.2 Doro

The DORO (see e.g. [18, 17]) project focusses on the routing of human-readable documents which are in electronic form. DORO aims at assuring that they arrive at the proper destination with the smallest possible time delay. Since manual mail routing is error prone and time consuming, DORO aims at automating this process.

The purpose of the DORO project is to develop a scaleable and platform-independent system performing automatic routing of human-readable documents in electronic form on the basis of an analysis of the contents of the documents and knowledge of the characteristics of the possible destinations. The system ought to be capable of handling

the increasing demands of big users with respect to throughput and complexity. The system will be embedded into standard workflow environments and will appear as a "black box component" of such systems.

The DORO project thus concerns only one side of information discovery, namely document routing which can be seen as multi-way filtering. DORO shares with PROFILE the use of linguistic analysis of documents. Unlike PROFILE, DORO aims at the development of a single module of a system instead of a complete architecture.

5.3 EuroSearch

The EuroSearch project (see e.g. [22]) aims at forming a distributed and multi-lingual federation of European search and categorisation services. The system supports cross-language retrieval, permitting users to query all of the federation's national sites in his own language.

The EuroSearch consortium offers an open and extensible framework and exploits existing search facilities. A common interface, instantiated for each participating local language, provides access to the functionality of the system. This comprises, for example, redirecting queries, profile adaptation, and subject-based retrieval using classification of documents.

Like the PROFILE project, organisational as well as technical issues play an important role in the EuroSearch consortium. Unlike the PROFILE project, the EuroSearch project explicitly focusses on cross-language retrieval. The PROFILE project, however, focusses more on the integration of multiple research disciplines.

5.4 Condorcet

The Condorcet project (see e.g. [7]) focusses on automatic indexing of scientific documents in several domains. Assigning concepts to documents takes place on the basis of intensive analysis using sophisticated linguistic technology. Details about the techniques used may be found in [21]. A prototype system has been implemented.

An important feature of the Condorcet project is the use of structured concepts for indexing. By using concepts like "cures(aspirin, headache)" and "causes(aspirin, headache)", document characterisations can be made more subtle. By exploiting this, the precision of retrieval systems will be enhanced.

Like the PROFILE project, linguistic analysis of documents plays an important role in the Condorcet project. The Condorcet project, however, does not consider an agent-based architecture and is domain specific.

6 Discussion

In this article, we presented the work done in the PROFILE project. The agent-based architecture provided a means to capitalise on any overlap between the components. That is, although the attunement between the components was not optimal, the flexible framework allowed the agents to cooperate based on their shared capabilities. If we had used a more rigid architecture, the project as a whole would have suffered more from the delay in a single component.

The user modelling, language processing, and matching components conducted research in descriptors that go beyond the keyword-based approach.

This means that PROFILE is capable of extracting structured descriptors from text,

normalising these descriptors with respect to morphology, syntax, and lexico-semantical issues. In addition, descriptors are created based on knowledge representations and the user model. Furthermore, similarity functions to match structured descriptors were designed.

The PROFILE prototype performs proactive filtering. The Scheduler adjusts the flow of control to the type of agent showing proactive behaviour. Currently, the Scheduler may start an information cycle proactively itself or react to proactive calls from a Document Collector or the Expander.

Every research line in the PROFILE system offers opportunities for further research. An interface that accepts information needs in different forms would greatly enhance the appeal of the PROFILE prototype. Expansions on information need representations can be more elaborate and focus on words that should *not* occur in a document. The expanded result should yield Boolean index expressions. More complex document characterisations may be the result of more sophisticated parsing techniques.

An interesting consequence of these investigations for the PROFILE system would be that different instantiations will appear for every module. This necessitates a need for negotiation strategies and learning of capabilities, something which was not marked as a research issue for PROFILE in the original line of investigation.

Acknowledgements

The authors would like to thank Eduard Hoenkamp and the former PROFILE team members Debbie Tarenskeen, Janek Mackowiak, and Theo Huibers. The first author would like to thank Eduard Hovy for his supervision on a part of his research.

References

[1] L. Ambrosini, V. Cirillo, and A. Micarelli. A hybrid architecture for user-adapted information filtering on the World Wide Web. In A. Jameson, C. Paris, and C. Tasso, editors, *Proceedings of the Sixth International Conference on User Modeling, UM'97*, pages 59–61. Springer Wien New York, Vienna, New York, 1997.

[2] A. T. Arampatzis, Th. P. van der Weide, C. H. A. Koster, and P. van Bommel. An evaluation of linguistically-motivated indexing schemes. In *Proceedings of the BCS-IRSG'2000*, 2000. To appear.

[3] A.T. Arampatzis and T. Tsoris. A linguistic approach to information retrieval. Master's thesis, Department of Computer Engineering and Informatics, University of Patras, Patras, Greece, June 1996.

[4] A.T. Arampatzis, T. Tsoris, and C.H.A. Koster. IRENA: Information retrieval engine based on natural language analysis. In *Proceedings of RIAO'97 Computer-Assisted Information Searching on Internet*, pages 159–175, McGill University, Montreal, Canada, 1997.

[5] A.T. Arampatzis, T. Tsoris, C.H.A. Koster, and Th.P. van der Weide. Phrase-based information retrieval. *Information Processing & Management*, 34(6):693–707, December 1998.

[6] A.T. Arampatzis, Th.P. van der Weide, C.H.A. Koster, and P. van Bommel. Linguistically-motivated information retrieval. In *Encyclopedia of Library and Information Science*. Marcel Dekker, Inc., New York, Basel, 2000. To appear.

[7] B. van Bakel, R.T. Boon, N.J.I. Mars, J. Nijhuis, E. Oltmans, and P.E. van der Vet. Condorcet annual report. Technical report, Knowledge Based System Group, University of Twente, The Netherlands, 1997.

[8] E. Brill. Some advances in rule-based part of speech tagging. In *Proceedings of the*

Twelfth National Conference on Artificial Intelligence (AAAI-94), Seattle, Wa., 1994.

[9] P.D. Bruza. Hyperindices: A novel aid for searching in hypermedia. In A. Rizk, N. Streitz, and J. Andre, editors, *Proceedings of the European Conference on Hypertext - ECHT 90*, pages 109–122, Cambridge, United Kingdom, 1990. Cambridge University Press.

[10] P.D. Bruza and Th.P. van der Weide. The modelling and retrieval of documents using index expressions. *ACM SIGIR FORUM (Refereed Section)*, 25(2), 1991.

[11] V.K. Chaudri, A. Farquhar, R. Fikes, P.D. Karp, and J.P. Rice. OKBC: A programmatic foundation for knowledge base interoperability. Technical Report KSL-98-08, Knowledge Systems Laboratory, Stanford University, 1998.

[12] M. Dreger, N. Fuhr, K. Grossjohan, and S. Lohrum. Provider selection - design and implementation of the medoc broker. In *Proceedings of the Dagstuhl workshop*, July 1997.

[13] T. Finin and R. Fritzon. KQML - a language and protocol for knowledge and information exchange. Technical Report CS-94-02, Computer Science Department, University of Maryland, 1994.

[14] David Gardiner, John Riedle, and James Slagle. TREC-3: Experience with conceptual relations in information retrieval. In Donna K. Harman, editor, *NIST Special Publication 500-226: Overview of the Third Text REtrieval Conference (TREC-3)*. Department of Commerce and National Institute of Standards and Technology, 1994.

[15] T.R. Gruber. A translation approach to portable ontology specification. *Knowledge Acquisition*, 5(2):199–220, 1993.

[16] H. Jeon. Jatlite. ¨`http://java.stanford.edu/java_agent/`¨, 1997.

[17] C.H.A. Koster. Fuzzy Matching using WordNet. Addendum to [18], Department of Computer Science, University of Nijmegen, Nijmegen, The Netherlands, 1998.

[18] C.H.A. Koster. Normalization and matching in the DORO system. In *Proceedings of the BCS-IRSG*, 1999.

[19] R.M. MacGregor. Inside the LOOM classifier. *SIGART Bulletin*, 2(3):70–76, 1991.

[20] G.A. Miller. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[21] E. Oltmans. *A Knowledge-Based Approach to Robust Parsing*. PhD thesis, University of Twente, Enschede, The Netherlands, 2000.

[22] E. Picchi and C. Peters. Exploiting lexical resources and linguistic tools in cross-language information retrieval: the eurosearch approach. In *Proceedings of the First International Conference on Language Resource and Evaluation*, Granada, Spain, 1998.

[23] M.J. Plasmeijer and M.C.J.D. van Eekelen. Language report concurrent Clean. Technical Report CSI-R9816, Computing Science Institute, University of Nijmegen, Nijmegen, The Netherlands, June 1998.

[24] J. Rice, A. Farquhar, P. Piernot, and T. Gruber. Using the Web instead of a window system. In *Proceedings of CHI'96 Conference on Human Factors in Computing Systems*, pages 103–110, Vancouver, Canada, 1996. Addison Wesley.

[25] G. Salton. *The Smart System – Experiments in Automatic Document Processing*. Prentice Hall Inc., 1971.

[26] J. Simons. An evaluation of semantic inference to enhance information retrieval. ¨`http://www.nici.kun.nl/~simons/Publications/sem-inference.pdf`¨, 2000.

[27] A.F. Smeaton and I. Quigley. Experiments on using semantic distances between words in image caption retrieval. In H. Frei, D. Harman, P. Schäuble, and R. Wilkinson,

editors, *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 174–180. ACM Press, 1996.

[28] M.A. Stairmand. *A Computational Analysis of Lexical Cohesion with Applications in Information Retrieval*. PhD thesis, University of Manchester Institute of Science and Technology, October 1996.

[29] A. Valente, T. Russ, R. MacGregor, and W. Swartout. Building and (re)using an ontology of air campaign planning. *IEEE Intelligent Systems*, 1:27–36, 1999.

[30] Hardeveld M. van. Evaluation and redesign of the Profile user interface. Technical report, Nijmegen Institute of Cognition and Information/University of Twente, 1999.

[31] E.M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, July 1994.

[32] B.C.M. Wondergem, P. van Bommel, and Th.P. van der Weide. Matching index expressions for information retrieval. *Information Retrieval Journal*. To appear.

[33] B.C.M. Wondergem, P. van Bommel, and Th.P. van der Weide. Nesting and defoliation of index expressions for information retrieval. *Knowledge and Information Systems*. To appear.

[34] B.C.M. Wondergem, P. van Bommel, and Th.P. van der Weide. Boolean Index Expressions for Information Retrieval. Technical Report CSI-R9827, University of Nijmegen, Nijmegen, The Netherlands, December 1998.

[35] B.C.M. Wondergem, P. van Bommel, and Th.P. van der Weide. Cumulative duality in designing information brokers. In *Proceedings of the 9th International Conference on Database and Expert Systems Applications (DEXA)*, Vienna, Austria, August 1998.

[36] B.C.M. Wondergem, P. van Bommel, and Th.P. van der Weide. Compactness of Boolean Index Expressions. Technical Report CSI-R9911, University of Nijmegen, Nijmegen, The Netherlands, 1999.

[37] B.C.M. Wondergem, M. van Uden, P. van Bommel, and Th.P. van der Weide. INdex Navigator for Searching and Exploring the WWW. In *Proceedings of the Conferentie Informatiewetenschap (CIW'2000)*, Rotterdam, The Netherlands, April 2000. To appear.

[38] M. Wooldridge and N.R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

Agent	Function	Implementation
User Interface	Interacts with user.	HTML with JAVA
Database	Stores user characteristics & interests.	JAVA
Document Collector	Generates a stream of initial documents.	JAVA that sends a query to several search engines and retrieves the results.
Matcher	Accepts an indexed document and a query in the form of index expressions and compares each of them.	JAVA, equipped with a C wrapper, that starts up a CLEAN[23] program.
Language Processor	Creates index expressions from documents using part of speech tagging and shallow parsing.	JAVA that starts Brill's POS tagger [8] and a PERL program.
Expander	Accepts an information need and expands it when possible.	JAVA
Scheduler	Controls the proactivity of the system.	JAVA

Table 1: PROFILE agents with their capabilities and tasks.