In: Verbraeck, A. (ed.): Proc. Int. Conf. EuroMedia/WEBTEC, London U. K., December 19-21, 1996. San Diego: The Society for Computer Simulation, pp. 5-12

ADAPTIVE REFINEMENT OF SEARCH PATTERNS FOR DISTRIBUTED INFORMATION GATHERING

Uwe M. Borghoff, Pierre-Yves Chevalier, Jutta Willamowski

Rank Xerox Research Centre, Grenoble Laboratory
6, chemin de Maupertuis. F-38240 Meylan, France
E-mail: {borghoff,chevalier,willamowski}@grenoble.rxrc.xerox.com

KEYWORDS

adaptive query formulation, attribute-based search, constraint-based knowledge brokers, information gathering.

ABSTRACT

Information brokerage systems support distributed information gathering for data collections stored within the World Wide Web or for other on-line data repositories. These systems present a unified face of the underlying heterogeneity, both in terms of the access/search protocols and in terms of the data schemata. Typically, the user may input some values (e.g., interpreted as strings within a fulltext search), and may refine/relax them once she gets too many/little results for her query. Adaptiveness with respect to a refinement of *attributes* can improve the precision of the searches dramatically.

This paper discusses adaptive refinement of attribute-patterns together with its main implications in the context of the *Constraint-Based Knowledge Broker* system. It presents an implementation for distributed information gathering, and motivates design choices through concrete examples.

INTRODUCTION

The last few years have seen a growing interest in information brokerage systems.

Systems like the *Constraint-Based Knowledge Broker* system [Andreoli et al. 1996] or *Tsimmis* [Chawathe et al. 1994] are able to gather data from collections available on the World Wide Web or other on-line repositories. Despite the distributed and heterogeneous nature of the underlying environment, they present to the user a unified face of the information sources they rely on. In practice this means unifying both the access and search protocols (like http, gopher, Z39.50), and the data schemata.

Given the vast amount of information these systems allow to access, the quality and the precision of the search results become critical factors. The goal is two-folds: first to prevent the user from being overloaded by too many results, and second not to miss pertinent data. Main issues that arise in this context are:

- *finding and selecting information sources*. This is not an easy task within a fast evolving context such as the Web. A support for yellow pages grouping information sources by domain becomes necessary. Besides, in general, without preindexing or preprocessing there is no statistical information about the information sources available, i.e., methods applied in traditional information retrieval (e.g., methods based on term frequency or document frequency) cannot be applied to select relevant information sources for a given query.
- *formulating complex queries*. This involves query decomposition [Andreoli et al. 1995], planning, projections and joins [Borghoff and Schlichter 1996].
- *gathering information*. As explained in [Borghoff et al. 1996] for a network publication system, the system must be able to formulate a query in different formats, i.e. syntactically as well as semantically, and to select the right keywords for the search. This means defining a general purpose interface integrating the full poten-

tial of the external sources in terms of search capabilities.

- *extracting information*. This problem arises because results are received in different formats, the information is often unstructured (e.g., a simple html-page), and further processing (e.g., using natural language techniques) has to be applied to extract higher-level meta-information such as the attribute/value pairs for author, title, and date.
- *processing the extracted information and generating the results*. Information sources provide different coverage, i.e., there is a strong need for post-processing such as filtering. Attribute-based ranking techniques for the result presentation provide a fine-grained result structure, increase the degree of relevance feedback [Rocchio 1971], and support a sound view over the search semantics. Furthermore, they allow avoiding the collection fusion problem [Callan et al. 1994] that arises when putting together ranked results from different information sources.

In the context of the Constraint-Based Knowledge Broker system, we are exploring the use of attribute-based searches. We combine three major techniques: refinement/relaxation of constraint attributes, dynamic ranking, and adaptive selection of relevant attributes. This combination gives birth to a simple, yet efficient and powerful search paradigm: adaptive refinement.

Adaptive refinement of a query allows the system to help its user refine a search pattern. In usual systems, the user can refine or relax the search keywords when a query was too general or too specific. Adaptive refinement provides further support: it lets the system propose new, promising attributes to the user, attributes it learned during previous searches. The user can also use these new attributes to dynamically rank the search results. Information reuse among intersecting queries and sharing of search results among users with similar interests are other aspects that influence the degree of adaptivness.

Adaptive refinement in general is also explored in other contexts, e.g., in the context of mail-browsers or news-readers. *MessageWorld* [Rose et al. 1995], a prototype system at Apple Computer, provides a rendez-vous mechanism for readers of messages stored in several on-line repositories. It implements adaptive algorithms (using a voting approach) that select preferred messages and re-arrange the presentation of these messages to the readers, e.g., by a ranking policy according to the expressed opinion on the interest of already read messages.

The remainder of this paper concentrates on the problem of query refinement and the use of search attributes. Section 2 introduces traditional query formulation and shows some of the drawbacks. Section 3 illustrates the adaptive refinement approach in detail and shows how the refinement of search attributes helps to increase the search precision (see Figure 1). In order to illustrate problems and solutions in a homogeneous way, the search for *books/articles about "distributed systems"* serves as an example throughout the paper. Section 4 concludes by discussing some relevant features for future work.

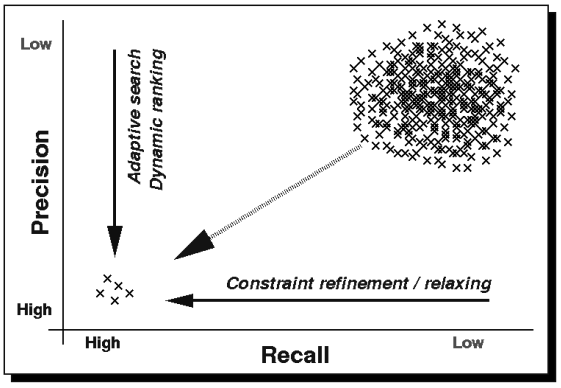

Fig. 1: Shifts within the precision/recall matrix through adaptive refinement.

QUERY FORMULATION

Formulating a query consists in general in selecting a set of search keywords and in linking them with the boolean operators and, and or. Often, some keywords can be excluded with not. Several problems arise in the context of query formulation that are also relevant for query refinement.

One essential problem concerns the selection of the search keywords. The user has to select appropriate search keywords in order to optimize the retrieval *precision* and *recall*: if the selected keywords are too general too many irrelevant documents will be returned; if the keywords are too specific, too many relevant documents will be missed. In order to support the keyword selection process, linguistic techniques (e.g., thesaurus-based [Güntzer et al. 1989]) allow to preprocess the users search keywords and to

propose an extension with related keywords before submitting the query, e.g., suggest to narrow "distributed system" to "distributed file system" or to complement "computer science" with "informatics".

The keyword selection problem is even more important when refining a query in consequence to a bad precision or a bad recall. Yet most information retrieval systems on the Web do not propose any functionalities specific to query refinement. Linguistic techniques can be useful when the recall is bad: proposing search keywords with a more general meaning than the initial ones will lead to better results. In reaction to bad precision adapting the search keywords by adding new, related words found in the results can improve the quality of the results obtained. This technique, called relevance feedback [Allan 1996], is well known in traditional information retrieval, but not yet integrated with information retrieval on the World Wide Web. *Discover* [Discover 1996] uses this possibility to allow the user to extend her search keywords from a list of words found in matching documents located next to the keywords initially chosen by the user.

Another problem related to query formulation is that most information retrieval systems are based on fulltext search. This means that the system returns as a result all documents containing the search keywords anywhere in the text. Fulltext search does not allow to efficiently focus a query, nor afterwards to appropriately rank the results: a document containing the search keyword in the title should have higher ranking than one containing it in the abstract, which in turn should have higher ranking than one containing it in the body of the document. Here, the definition of search attributes a) giving meta-information, e.g., title, abstract, or fulltext, and b) supporting the possibility to formulate more sophisticated queries (i.e., not only selecting a set of search keywords but also assigning them to some search attribute) can dramatically increase the quality of the information retrieved. With regard to the huge amount of information accessible on the Web, here the use of search attributes is essential.

Yet, most information retrieval systems on the Web provide only fulltext search within the html-documents they index. This is still true, even if they return results structured in title, date and other attributes. Some retrieval systems already start to integrate the possibility to use such attributes also for query formulation in their advanced user interfaces [AltaVista 1996, Open Text 1996]. The problem is that html is poorly structured. Thus it provides only poor information about the documents, usually the only meta-information explicitly given is the title. Therefore it is impossible to use other attributes without further processing: either the authors of html-documents must introduce themselves some additional hidden meta-information, or further linguistic techniques must be applied when indexing the html-documents (or, after searching - what is rather resource consuming -, to extract meta-information from the fulltext).

Beside html-documents, the Web gives also access to a large set of on-line data repositories. These data repositories (e.g., NCSTRL [NCSTRL 1996] and Library of Congress [Library of Congress 1996]) provide information often structured in a large set of attributes. They are searchable via html-forms and cgi-scripts but cannot be indexed by robots; thus their content cannot be retrieved with usual Web search engines!

One major goal of the Constraint Based Knowledge Broker system, is to provide a unified interface to both, raw html documents and on-line data repositories. The key for building such a system is the use of an attribute/value representation that homogenizes poorly and highly structured results in a single yet powerful scheme.

ADAPTIVE REFINEMENT OF SEARCH PATTERNS

Definition of Search Attributes

Formulating a query in the Constraint-Based Knowledge Broker system consists in the following three steps:

1. *selecting the query domain*, e.g., "operas" or "books/articles". The system will forward the query to the sources relevant for that domain, e.g., among others to ACM [ACM 1996] and NCSTRL [NCSTRL 1996] for queries concerning "books/articles".
2. *defining the search pattern*. The search pattern is defined by constraints expressed via attribute/value pairs, e.g., one might search for books/articles written by a specific author such as "Lamport". Depending on the query domain different attributes are relevant and searchable. Initially, the query panel gives access to a predefined set of search attributes relevant for the selected domain: e.g., author, title and keywords

for “books/articles”. The user can indicate alternative values for each attribute, e.g., search for an article written by Lamport or by Lampson. The conjunction of attribute/value pairs represents the global search pattern.

3. *submitting the query*, along with its options, allowing for example a) to search general purpose repositories concurrently, and b) to submit the search also to usual Web search engines.

Once submitted, the query is sent to the broker system and is processed further. The result panel appears on the screen and results are shown in order of arrival. Results which contradict any of the user’s constraints are filtered out by the brokers local sifting processes.

The user can post-process the results by deleting uninteresting hits, or by reordering the hits dynamically, either alphabethically (e.g., by author) or according to a selected ranking strategy. Different possible ranking strategies are defined within the system; the user can dynamically switch from one to another, depending on which she considers appropriate.

If the user is not satisfied with the results received, she can modify the query. Figure 2 shows an example for bad precision: the search for books/articles containing “distributed system” in the title returned more than one hundred answers. These results were returned from only five connected information sources: the Glimpse Bibliography server [Computer Science Glimpse 1996], the NCSTRL service [NCSTL 1996], the ACM Journal Abstracts server [ACM 1996], the Library of Congress service [Library of Congress 1996], and a server concerned with bibliographic information on the topics of database and logic programming [Database and Logic Programming 1996].

One can easily imagine what the result would be with several dozens of attached information sources. NCSTRL alone returned more than three hundred raw results: only sixty contained “distributed system” in the title; the others were filtered out by the broker system.

Relaxing / Refining the Query

To modify a query the user tunes either the domain, the search pattern, or the search options. The modification depends on the characteristics of the results obtained. If the system retrieved not enough results, some of the constraints can simply be relaxed or suppressed. If it returned too many and inappropriate results, the query has to be refined.

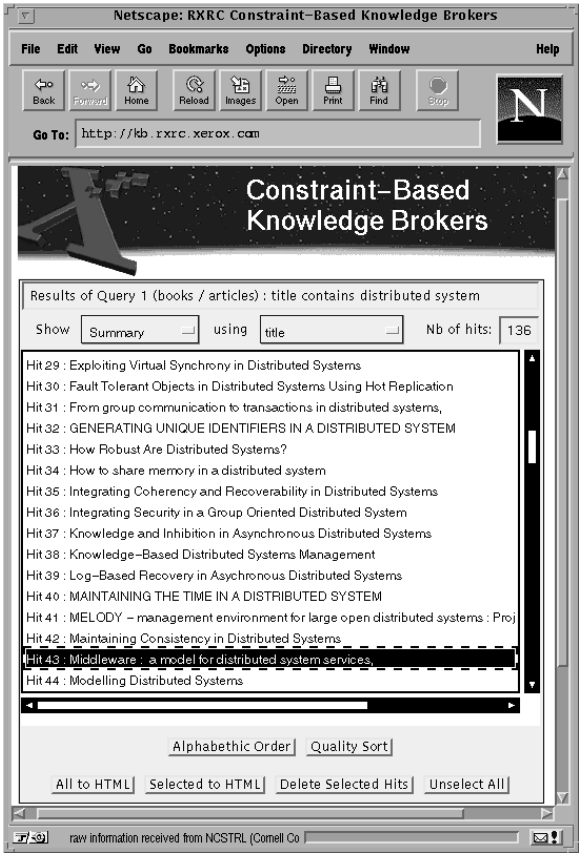

Fig. 2: Bad precision: results obtained with the initial search for books/articles with a title containing “distributed system”.

Bad retrieval precision is caused by insufficient search patterns. The Constraint Based Knowledge Broker system considers each result not contradicting the search pattern as correct (i.e., logically sound) and forwards it to the user. This allows integrating heterogeneous information sources, but leads sometimes to a large number of results: e.g., when searching for books/articles by author “Lamport” and published by “ACM”, an information source which only contains information about authors but not about publishers will return all the books/articles it contains written by an author named “Lamport” and not give any information about the publisher. All these results are logically sound (as far as the constraint solvers are concerned - there is no contradiction!) and will be forwarded to the user.

In order to improve precision, the user can refine the query using the following strategies:

- specifying that the publisher attribute is obligatory in the result, which excludes information sources from the search which do not know this attribute,
- constraining other attributes known by the information source which were not constraint initially. The names of these attributes can be deduced from the results it sent back,
- extracting related attributes / keywords from the results the user considers appropriate. Related keywords can be extracted manually by the user or with support form the system through natural language summarizing techniques.
- defining a sub-query that delivers new search keywords (e.g., refine a query for books about operas to a query for books about operas composed by Wagner)

Most of these refinement strategies depend on the results obtained previously.

Figure 3 shows a result in detail that was ranked high; it shows all its attribute/value pairs. Some of these attributes, such as the title, were initially known to the system and the user; others are new, e.g., the bibtype or the publisher.

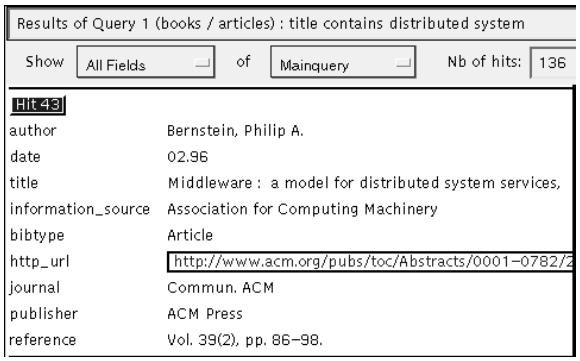

Fig. 3: A good result in detail: some of the attributes are new to the system. They are learned and can thus be used for query refinement.

Thus, from the results of our initial example we have learned several new attributes. Among these, publisher and journal are particularly interesting[1].

We will use the publisher attribute to refine the query and focus the search on ACM journals where we expect to find high-quality documents (see Figure 4).

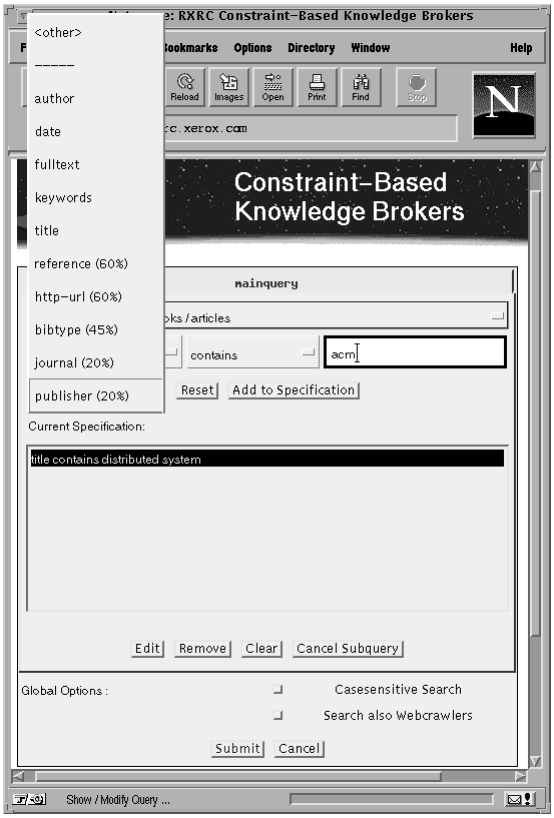

Fig. 4: Refinement of the search using the learned publisher-attribute, which is now accessible in the attribute list.

Figure 5 shows the final list of results to the "distributed system" query limited to publisher "ACM".

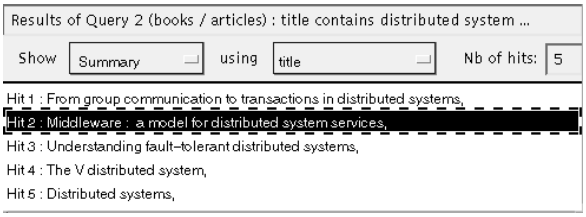

Fig. 5: Good precision: results obained with the refined search for books/articles with a title containing "distributed system" and publisher "ACM".

[1] For the sake of clarity, we chose to let the system discover common-sense attributes within the books/articles domain. Of course, discovered attributes like the Dewey-number or the bibtype are not so obvious for the average user, but still helpful to track specific objects covered by search.

From about one hundred results to the initial query, the number of results has been reduced to five articles through a simple attribute-based refinement.

How Ranking Strategies Influence Adaptive Refinement

Apart from the well-known techniques of result ranking within an information retrieval process [Harman 1992], the information gathering from different Web-repositories brings new challenges.

In the following, we list some of our findings that have partly been implemented within the Constraint-Based Knowledge Broker system:

- a search result that is obtained from a repository with high trustworthiness with respect to the query, i.e. being well maintained, regularly updated and specialized in the query domain (e.g., the ACM Journal server for a query about computer science articles), gets a higher ranking than a result from a less reliable source, for instance, a privately maintained repository or a general information source such as AltaVista. Attributes found within such a result are preferred to build refined search patterns.
- a search result that is retrieved in an identical form from different repositories (but not mirror sites of the repository) gets a higher ranking than a result that is found at only one site with low trustworthiness. Where appropriate, a so-called *majority consensus* may be applied if search results differ in some attributes; that is, if some sites say, e.g., "11-17" for the page numbers of a particular paper, and a majority of independent sites say, e.g., "11-27", we tend to believe the latter. This tendency can be combined with the trustworthiness value of a site, i.e., we give higher *votes* to a reliable site. Attributes whose values are verified in this way (that is, majority consensus or some voting approach) are strong candidates to refine the search patterns.
- a search result that is retrieved in overlapping parts from different repositories may be *fused*, i.e., we combine the result attributes. For attributes obtained from several sites, we use the ranking policy described above, for attributes obtained from single sites, we use their trustworthiness value. A heuristic may be implemented to install an overall ranking value: results for a bibliographical search may give preference to the author and title attributes; the attributes for the publication month or the publisher's address are less influential. These preferred attributes are also preferred to build refined search patterns.
- a search result providing values for a large number of constraint search attributes gets higher ranking than search results which fill only few of these attributes, e.g., in the refined example query results with the title and the publisher-attribute filled get higher ranking than those only filling the title-attribute. Again, depending on the search domain, more significant attributes can be privileged. Again, attributes found within such a result are preferred to build refined search patterns.

In general, the attributes proposed to the user for refining his query are ranked according to the ranking strategy applied to the results. Each of these ranking strategies follows a different criterion.

Some of them are already available. Others are being implemented. As it is not clear which criteria are the best or how these criteria - and the associated strategies for the selection of the refinement attributes - are to be combined in an optimal way, the user has the possibility to select/change the strategy interactively. Her choice can be saved between sessions (along with other preferences) in a user profile.

CONCLUSION AND FUTURE WORK

Information brokerage systems gather data from collections stored around the world, and present to their users a unified image of the underlying data repositories. These systems dramatically increase the number of information sources a given user is able to access and greatly simplifies the interaction with these sources.

One of the major pit-falls these systems have to face is information overload. From the user point of view, returning too many answers for a given query is as useless as returning no answer at all. Classical methods such as fulltext search fail to overcome this limitation.

This paper shows that "adaptive refinement of attribute patterns" based on the use of search attributes provides many ways by which the system helps the user refine or relax her query. This refinement method is strongly related to result ranking strategies. Indeed the selection of an attribute for refinement (e.g., publisher in the article example) follows a ranking policy.

A stable version of the Constraint Based Knowledge Broker system which demonstrates the principle of adapative refinement in action, is available on-line (see [CBKB 1996]). We invite the interested reader to visit our site, try the system, and send in feedback.

A future version, currently under development, will provide more powerful ranking strategies, a sophisticated domain hierarchy and support for natural language analysis.

ACKNOWLEDGEMENTS

We would like to thank Jean-Marc Andreoli and Boris Chidlovskii, members of the Constraint Based Knowledge Brokers project, for their contributions to the design and implementation of the system.

REFERENCES

J. M. Andreoli, U. M. Borghoff, R. Pareschi, J. H. Schlichter: Constraint Agents for the Information Age. In: *J. Universal Computer Science* 1:12 (1995), 762-789.

J. M. Andreoli, U. M. Borghoff, R. Pareschi: The Constraint-Based Knowledge Broker Model: Semantics, Implementation and Analysis. In: *J. Symbolic Computation* (1996), in press.

ACM: *http://www.acm.org/pubs/toc/Search.html*, status september 1996.

J. Allan: Incremental Relevance Feedback for Information Filtering. In: *Proc SIGIR*, ACM. 1996, pp. 270-278.

AltaVista: *http://altavista.digital.com*, status september 1996.

U. M. Borghoff, R. Pareschi, H. Karch, M. Nöhmeier, J. H. Schlichter: Constraint-Based Information Gathering for a Network Publication System. In: *Proc. PAAM '96*, London. 1996, pp. 45-59.

U. M. Borghoff, J. H. Schlichter: On Combining the Knowledge of Heterogeneous Information Repositories. In: *J. Universal Computer Science* 2:7 (1996), pp. 512-532.

CBKB accessible prototype: *http://www.xerox.fr*, and follow links *grenoble/ct/cbkb*, status september 1996.

S. Chawathe et al.: The Tsimmis Project: Integration of Heterogeneous Information Sources. In: *Proc. IPSJ '94*, Tokyo. 1994.

J. P. Callan, Z. Lu, W. B. Croft: Searching Distributed Collections with Inference Networks. In: *Proc. SIGIR*, ACM. 1995.

Database Systems and Logic Programming server: *http://www.informatik.uni-trier.de/~ley/db/index.html*, status september 1996.

Discover: *http://discover.imag.fr*, status september 1996.

Computer Science Bibliography Glimpse Server: *http://glimpse.cs.arizona.edu/bib/,* status september 1996.

U. Güntzer, G. Juttner, G. Seegmüller, F. Sarre: Automatic Thesaurus Construction by Machine Learning from Retrieval Sessions. In: *Information Processing and Management* 25:3 (1989), 265-273.

D. Harman: Ranking Algorithms. In W. B. Frakes, R. Baeza-Yates (eds.): *Information Retrieval: Data Structures and Algorithms*, Prentice Hall. 1992, pp. 363-392.

Library of Congress: *http://lcweb.loc.gov/z3950/mums2.html*, status september 1996.

NCSTRL: *http://www.ncstrl.org/Dienst/UI/2.0/Search*, status september 1996.

Opentext: *http://index.opentext.net/main/powersearch.html*, status september 1996.

J. J. Rocchio: Relevance Feedback in Information Retrieval. In G. Salton (ed.): *The Smart retrieval System, Prentice Hall*, 1971, pp. 313-323.

D. E. Rose, J. J. Bornstein, K. Tiene: MessageWorld: A New Approach to Facilitating Asynchronous Group Communication. In: *Proc. CIKM '95*, Baltimore MD, ACM. 1996, pp. 266-273.

BIOGRAPHIES

Uwe Borghoff holds B.S., M.S. and Ph.D. degrees in Computer Science from the Technische Universität München, Munich, Germany. In 1993, he was awarded the Venia Legendi in Computer Science. He worked for IBM and freelance with IBBG for three years before joining the faculty of Computer Science at the Technische Universität München as Assistant and Research Scientist in 1986. Since February 1994, he has been Senior Scientist at the Rank Xerox Research Centre in Grenoble, France. His research interests include language support and protocols for computer-supported collaborative work, and distributed document processing. As Project Leader, he has devoted his interests to interworking using constraint-based programming techniques, especially to the problem of knowledge brokerage in the world wide web. Uwe Borghoff is a coauthor of "Rechnergestützte Gruppenarbeit" (Springer-Verlag 1995), a standard German textbook on the subject, and author of "Catalogue of Distributed File/Operating Systems" (Springer-Verlag 1992). He is an editor of the Journal of Universal Computer Science (J.UCS), and is a member of the ACM, the IEEE, and the German Association of Computer Scientists.

Pierre-Yves Chevalier holds B.S., M.S. and Ph.D. degrees in Computer Science from the Université Joseph Fourier (Grenoble-I), Grenoble, France. His

dissertation addresses the problem of persistence and fault-tolerance support in distributed object-oriented systems. He worked for Bull in the framework of the COMANDOS project for three years before joining the European Computer-Industry Research Centre (ECRC) in Munich, Germany, as Research Scientist in 1994. Since March 1996, he is a Research and Development Engineer at the Rank Xerox Research Centre in Grenoble, France. His research interests encompass the broad topic of distributed computing with a particular emphasis on mobile computing, persistence support, fault tolerance, operating systems and distributed multi-media applications.

Jutta Willamowski holds B.S., and M.S. degrees in Medical Computer Science from the Universität Heidelberg, Germany, a diploma in Artificial Intelligence from the Université de Savoie, Chambéry, France, and a Ph.D. in Computer Science from the Université Joseph Fourier (Grenoble-I), Grenoble, France. Her Ph.D. addresses Problem Solving in System-User-Cooperation. She was then member of the Sherpa project at INRIA, the French National Institute for Research in Computer Science and Control. In 1995 she held an industrial post doctoral fellowship from INRIA and the company ILOG in Paris for transferring the system developped during her Ph.D. into an industrial product. Since January 1996 she is member of the Constraint Based Knowledge Broker project at the Rank Xerox Research Centre in Grenoble, France. In this context, she is especially concerned with information retrieval on the World Wide Web and with related user interface issues.