Daffodil: An Integrated Desktop for Supporting High-Level Search Activities in Federated Digital Libraries*

Norbert Fuhr[1], Claus-Peter Klas[1], André Schaefer[2], and Peter Mutschke[2]

[1] University of Dortmund
{fuhr,klas}@ls6.cs.uni-dortmund.de
[2] Social Science Information Centre, Bonn
{schaefer,mutschke}@bonn.iz-soz.de

Abstract. Daffodil is a digital library system targeting at *strategic support* during the information search process. For the user, mainly high-level search functions, so-called stratagems, implement this strategic support, which provide functionality beyond today's digital libraries. Through the tight integration of stratagems and with the federation of heterogeneous digital libraries, Daffodil reaches a high synergy effect for information and services. These effects provide high-quality metadata for the searcher through an intuitively controllable user interface. The visualisation of stratagems is based on a strictly object-oriented tool-based model. This paper presents the graphical user interface with a particular view on the integration of stratagems to enable *strategic support*.

1 Introduction

Today's digital library (DL) systems offer a large variety of functions for accessing their content. However, most of these systems are restricted to a single database, whereas federated DL systems provide only minimum searching and browsing capabilities. The high-level structure of the set of functions offered is either missing or more system-oriented. Finally, the interaction style of these systems is rather restricted (i.e. by using forms and menus only). Most of them do not support direct manipulation operations (like e.g. Drag&Drop), which would make the interaction more effective. In order to overcome these deficiencies, we have developed the Daffodil[1] system which provides solutions to the three problems mentioned above.

Daffodil is a federated DL system that offers a rich set of functions across a heterogeneous set of DLs (see [1] for a description of the architecture). The current prototype integrates 10 DLs in the area of computer science. Since different DLs may contain various pieces of information about the same publications, the federation yields important synergies for the user.

* Funded by the German Science Foundation (DFG) as part of the research initiative "Distributed Processing and Delivery of Digital Documents".

[1] **D**istributed **A**gents **f**or User-**F**riendly Access **o**f **Di**gital **L**ibraries

For structuring the functionality, we employ the concept of high-level search activities for strategic support as proposed by Bates [2]. Based on empirical studies of the information seeking behaviour of experienced library users, Bates distinguishes four levels of search activities. Whereas typical information systems only support low-level search functions (so-called moves), Bates introduced three additional levels of strategic search functions:

- A *tactic* is one or a handful of moves made to further a search. For example, breaking down a complex information need into subproblems, broadening or narrowing a query are tactics applied frequently.
- A *stratagem* is a complex set of actions (comprising different moves and/or tactics) exercised on a single domain (e. g. citation database, tables of contents of journals). Examples for stratagems are subject search (searching for all documents referring to this subject), citation search (find all documents citing / cited by a given article) or journal run (browse through issues or complete volumes of a relevant journal).
- A *strategy* comprises a complete plan for satisfying an information need. Thus, it typically consists of more than one stratagem (e. g. perform a subject search, browse through relevant journals and then find the documents cited by the most important articles).

Strategic support during the information search process is the fundamental concept implemented within Daffodil. High-level search functions, based on the stratagem level, implement this strategic support for the user and provide functionality beyond today's digital libraries. To our knowledge, Daffodil is the first implementation of Bates' ideas.

For visualising the strategic support, we use an object-oriented tool-based model with direct manipulation. On the desktop, the set of available stratagems is represented as a set of tools. A tool can be invoked in several ways, e. g. by dragging a DL object onto it.

The design of Daffodil offers a wide range of synergies, starting from the information sources up to the visualisation, whereby an optimal, strategy-supported information search process is presented to the user. Furthermore, the synergies are extended through a tight integration of stratagems, e. g. by using Drag&Drop mechanisms or links to external information sources.

In the remainder of this paper, we describe the user interface, along with the main features of the Daffodil desktop. Furthermore, we give a survey over the set of stratagems currently available, by describing typical use cases. We present two evaluations, performed during the Daffodil project, before related work is discussed in section 4. The paper ends with a conclusion and outlook.

2 Daffodil Integrated Desktop Design

In this section we describe the Daffodil desktop, the conceptual design model and the tools we integrated into the workplace.

2.1 High-level search activities

The Daffodil architecture is structured according to the levels of search activities named above. Each agent fulfills a function at a certain level and can invoke functions on its own or lower levels.

1. On the *move* level, *wrappers* connect to various DLs or services like thesauri or spell-checkers. The heterogeneity problem is addressed, by mapping the external data into a homogeneous XML metadata format.
2. The *tactic* level provides simple strategic actions by combining appropriate moves. For example, co-author search performs a search for all publications of an author and extracts the co-authors; in parallel, the corresponding function of Homepage-Search[2] is employed, and then the results of the two steps are merged.
3. *Stratagems* provide domain specific depth-search-functionality, by applying tactics to a set of similar items, like e. g. journals (*journal run*). The available stratagems are fully described in section 2.4.

Strategies are not supported by Daffodil automatically, yet. Instead the user is enabled to work much more strategy-oriented, by applying the high level functions of the *stratagems* and *tactics*. To offer these services in a user-friendly way, we needed to fulfill the following requirements:

- integrate distributed services and software agents in a consistent manner,
- provide tool chains to enable users to combine different services,
- ensure a flexible work flow,
- hide complexity.

For addressing these issues, we used the WOB[3] model, which is described below. Section 2.3 outlines Daffodil's user interface tools. Section 2.4 discusses how central stratagems are provided by its desktop.

2.2 The WOB Model

The *WOB* model for user interface design is based on the tool metaphor [3]. It attempts to solve the inherent contradictions in the interface design process — like that between flexible dialog control and conversational prompting — using a set of co-ordinated ergonomic techniques. It tries to fill the conceptual gap between interface style guides[4] and generic international standards[5]. The general software ergonomic principles of the *WOB* model are:

[2] http://hpsearch.uni-trier.de/

[3] German acronym for "object oriented directly manipulative graphical user interface based on the tool metaphor"

[4] like the Java Look and Feel Guidelines [4]

[5] like e. g. ISO 13407: "Human-centred design processes for interactive systems"

- *Strict Object Orientation and Interpretability of Tools*
  Strongly related functionality of the system is encapsulated in tools that are displayed as icons (not as menus). The tools open *views*, which are 'normal' dialog windows. Due to well-defined *dialog guidelines*, the chain of views a user is working on can be interpreted as a set of forms to be filled. In contrast, experienced users will prefer the tool view, which enables them to perform tasks more quickly; however, this view is cognitively more complex, and it is not required for interpretation.
  The user can manipulate objects on the surface in a direct manipulative manner. It is essential that consistency is guaranteed for the direction of the manipulation. Thus, the model requires object-on-object interaction style with a clear direction and semantics. The generally recommended interaction style is as follows: To apply a function on an item, the latter has to be dragged to a *tool*.
- *Dynamic Adaptivity*
  The interface adapts its layout and content always to the actual state and context. This is mostly used for a reduction of complexity in non-trivial domains, like browsing simultaneously in several relevant hierarchies at once. For example, the user may set the relevant context by choosing a classification entry; when activating the journal catalogue as the next step, the journals are filtered according to the valid classification context, to reduce complexity.
- *Context Sensitive Permeability*
  When known information is reusable in other contexts, it will automatically be reused.
- *Dialog Guidelines*
  The views of the tools are functionally connected e. g. by means of action buttons, hypertext links or rules which are triggered by plan recognition. A tool can also open its view proactively if the user needs its function in a given situation.
- *Intelligent Components*
  Tools and controls in the interface have access to context and state, in order to decide, if their function is valuable for the user. If applicable, they shall interact pro-actively with the user or the shared environment (the desktop), respectively.

Two principles of the model are information system-specific:

- *Status Display with Edit Mode*
  The system shall always display a paraphrase of the current state for the user. It can be shown as a natural or formal language string or even by using some visual formalism (like a table). The most obvious use case is query formulation. With a form-based interface some aspects (e. g. boolean operators) are always hidden. Thus, DAFFODIL also displays the paraphrase (i.e. the formal query) in order to prevent the user from forgetting parts of his/her query (re-)formulation. It enables easy access to all aspects of the systems state, e. g. for iterative query formulation. Novice users can learn

details from the paraphrase they would otherwise have to guess. They also can see if the system interprets their input in the way they expect it to.

- *Iterative Retrieval and Query Transformation*
  Initial query formulations tend to be inadequate for the user's intentions, due to uncertainty or unconscious goals in the search process. Therefore applications shall simplify iterative query formulation for the user. This can be achieved e. g. by summarising the query when displaying results. Furthermore, methods for automatic transformation have to be provided, in order to address the 'zero result' problem and to allow for the handling of semantic or syntactical heterogeneity of underlying data sources.

In accordance with the *Dialog Guidelines* principle, a particular feature of Daffodil's interface is *Multi-Level-Hypertext* [5] interaction that allows for switching the level of information, e. g. from a document to the journal or to the authors institution or homepage. External links are provided for giving strategic support when Daffodil's services supply no results. In these cases, queries for external search engines like Google[6] or HPSearch[7] are generated dynamically and executed from within Daffodil. This results in an external browser being invoked, where interaction may continue.

2.3 Daffodil's tools

Daffodil's high-level search activities, as outlined above, have been designed in close accordance with the *WOB* model as a range of tools that are integrated into a common workspace environment. The goal of Daffodil's desktop is to provide an environment to allow for retrieval, search and browse tasks, as well as collation, organisation and reuse of the retrieved information in a user–friendly way.

When the user first sees the desktop, the most frequently used tool is open. The default setting opens the search tool, but this setting is user–specific and will be made a personal choice or part of the user's profile. The user has free choice which tool to use first. It is possible to start browsing journals before using the search tool to commit a fielded search. Typical desktop states can be seen in Figures 1 to 3.

The tools built so far include:

- *Search tool*, to specify the search domain, set filters and compose queries. The queries are broadcasted to a set of distributed information services (via agents and wrappers). Integrated result lists are displayed for navigation and detail inspection.
- *Reference Browser*, which can be invoked by dropping document items on it. Citation indexes (like e. g. NEC ResearchIndex) are consulted to find references to and from the given item.

[6] `http://www.google.de`

[7] `http://hpsearch.uni-trier.de`

- *Classification Browser*, to allow hierarchical topic-driven access to the information space. It enables browsing of classification schemes like e. g. the ACM Computing Classification System.
- *Thesaurus Browser*, to transform search terms to broader or narrower terms. Subject-specific or Web-based thesauri, like e. g. *WordNet*[8], are used for finding related terms. Items can be used via Drag&Drop to another tool.
- *Author Network Browser*, to compute and browse co-author networks for a list of given authors. The list can be either typed or given by dropping a document item on the tool.
- *Journal Browser*, to search for a journal title and browse many journal directories, often with direct access to meta-data or the full-text of articles.
- *Conference Browser*, to search for a conference title and browse conference proceedings. The full-texts are directly accessible from within the tool, provided they are available in any of the DLs connected.
- *Personal Library* which stores DL objects in personal or group folders, along with the possibility of enabling awareness for these items.

This list is not comprehensive. However, the desktop has been designed to be easily extensible such that further tools and services can be added in the future. Due to limitation of resources, we implemented the most popular functionality so far. The focus of Daffodil lies on the integration of services (tools) and high-level functionality as described in the following section. This integration supports free choice of the search strategy and helps to provide the right functionality at the right time. Note that all tools can be used as starting points for search activities.

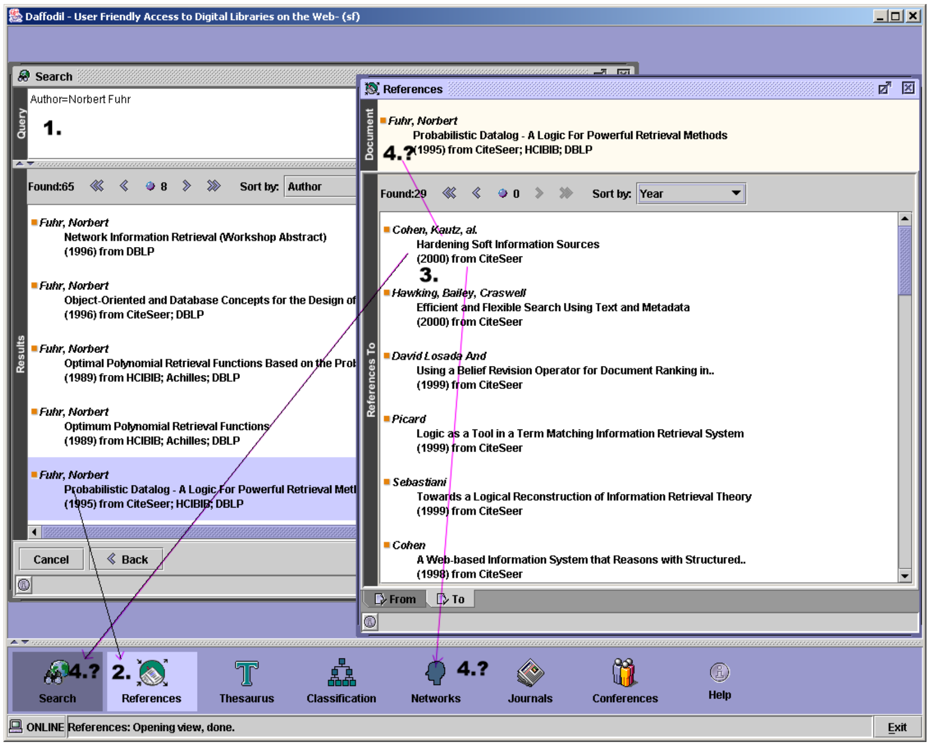
2.4 Stratagems on the Desktop

A major focus of Daffodil is to provide the user with high-level search services, called *stratagems* (sec. 2.1). Stratagems in Daffodil provide a segmentation of functionality (e.g. functions useful for searching and browsing journal articles). They provide depth searches, which are frequently needed and empirically observed in information seeking behaviour (e. g. [6, 7]), and exhaustively exploit the data structures of a single domain.

For the interface, we put particular emphasis on the fact that browsing and searching stratagems can be combined in a natural way. Furthermore, domain-specific properties are preserved which otherwise would have been abstracted due to heterogeneity and integration needs. In the following, we show how major stratagems of Daffodil can be used at its desktop.

The interactions between the Daffodil tools are based upon the observation that users frequently tend to initiate their strategic work with a generic or vague initial query; for this purpose, they use standard search methods, e. g. by *subject search* or *area scan*. The initial result set produced this way is usually not satisfactory.

[8] `http://www.cogsci.princeton.edu/~wn/`



Daffodil - User Friendly Access to Digital Libraries on the Web- (sf)

Search

Author=Norbert Fuhr

Found:65 8 Sort by: Author

- *Fuhr, Norbert*
  Network Information Retrieval (Workshop Abstract)
  (1996) from DBLP
- *Fuhr, Norbert*
  Object-Oriented and Database Concepts for the Design of
  (1996) from CiteSeer; DBLP
- *Fuhr, Norbert*
  Optimal Polynomial Retrieval Functions Based on the Prob
  (1989) from HCIBIB; Achilles; DBLP
- *Fuhr, Norbert*
  Optimum Polynomial Retrieval Functions
  (1989) from HCIBIB; Achilles; DBLP
- *Fuhr, Norbert*
  Probabilistic Datalog - A Logic For Powerful Retrieval Metl
  (1995) from CiteSeer; HCIBIB; DBLP

References

- *Fuhr, Norbert*
  Probabilistic Datalog - A Logic For Powerful Retrieval Methods
  (1995) from CiteSeer; HCIBIB; DBLP

Found:29 0 Sort by: Year

- *Cohen, Kautz, al.*
  Hardening Soft Information Sources
  (2000) from CiteSeer
- *Hawking, Bailey, Craswell*
  Efficient and Flexible Search Using Text and Metadata
  (2000) from CiteSeer
- *David Losada And*
  Using a Belief Revision Operator for Document Ranking in..
  (1999) from CiteSeer
- *Picard*
  Logic as a Tool in a Term Matching Information Retrieval System
  (1999) from CiteSeer
- *Sebastiani*
  Towards a Logical Reconstruction of Information Retrieval Theory
  (1999) from CiteSeer
- *Cohen*
  A Web-based Information System that Reasons with Structured..
  (1998) from CiteSeer

ONLINE References: Opening view, done.

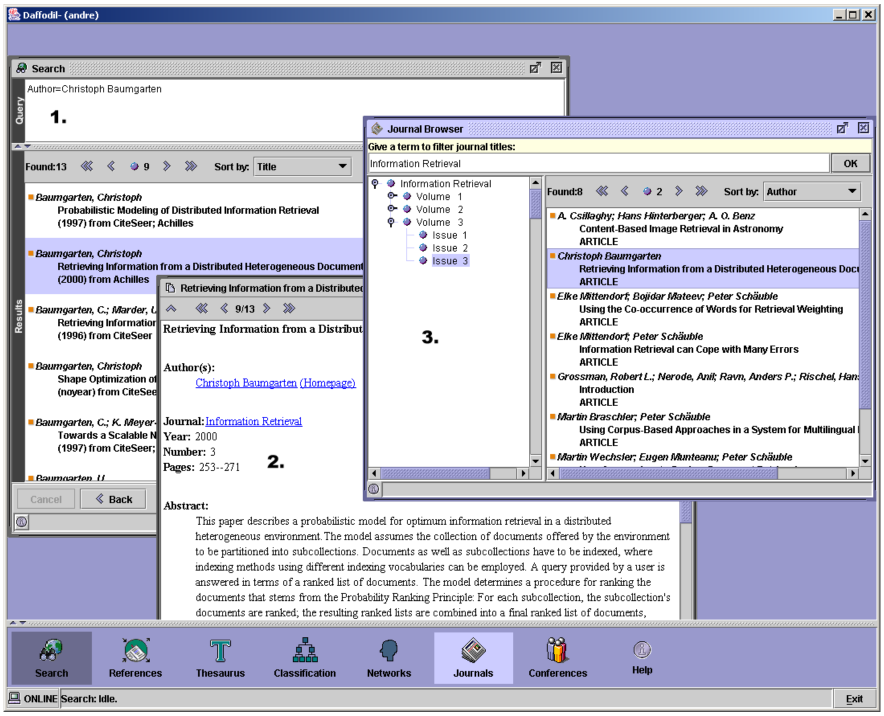
Fig. 1. Using the Citation Search for Known Item Instantiation strategies.

In Figure 1, the user started with a simple fielded search, giving a known author's name. Seeing the results he chose one entry and dragged it to the citation browse tool. He is presented the citing and cited documents, sourced from a citation index. From here he has again the free choice to continue iteratively inside the citation tool, formulate a new fielded query with the search tool, or compute a coauthors network for the authors in one of the documents. The choice is illustrated by the question mark in the Figure. Transition between the tools can be done by a drag and drop action. Accepting tools signal their acceptance by showing a green border on *drag-over*. In future the system may be enabled to give active help regarding the most promising choice for the next step. The system will learn about the most promising path by individual observation of chosen paths and the yielded results.

So at any point in the work flow, users may want to to initiate further searches on the basis of the current result. For this, two scenarios are provided by DAFFODIL:

- *Citation Search.* A typical example is to invoke a KNOWN ITEM INSTANTIATION strategy [8]: An interesting item was recognised to be relevant, and is being dropped on another tool to bring up further results. As illustrated in Fig. 1, a relevant document is dragged to the tool for citation tracking. Documents that cite or are cited by a given document are searched in several underlying data sources. Since stratagems can be performed iteratively, documents retrieved by *citation search* can be used as starting points for further citation searches, as well as other search activities (e. g. author network browsing or distributed search, as indicated in Fig. 1).

- *Cross Reference Linking.* Result set items are annotated by hyper-links that link to other services. As an example, journal titles occurring in metadata records are highlighted as hyper-links; clicking on such a link triggers an activation event to all tools dealing with journals, e. g. the journal browser that supplies the user with all other articles in the corresponding issue and enables a complete journal run from there (see Fig. 2).



Author=Christoph Baumgarten

Found:13 9 Sort by: Title

Baumgarten, Christoph
Probabilistic Modeling of Distributed Information Retrieval
(1997) from CiteSeer; Achilles

Baumgarten, Christoph
Retrieving Information from a Distributed Heterogeneous Document
(2000) from Achilles

Baumgarten, C.; Marder, U
Retrieving Information
(1996) from CiteSeer

Baumgarten, Christoph
Shape Optimization of
(noyear) from CiteSee

Baumgarten, C.; K. Meyer-
Towards a Scalable N
(1997) from CiteSeer;

Baumgarten, U

Retrieving Information from a Distributed

9/13

Retrieving Information from a Distribut

Author(s):
Christoph Baumgarten (Homepage)

Journal: Information Retrieval
Year: 2000
Number: 3
Pages: 253--271

Abstract:
This paper describes a probabilistic model for optimum information retrieval in a distributed heterogeneous environment. The model assumes the collection of documents offered by the environment to be partitioned into subcollections. Documents as well as subcollections have to be indexed, where indexing methods using different indexing vocabularies can be employed. A query provided by a user is answered in terms of a ranked list of documents. The model determines a procedure for ranking the documents that stems from the Probability Ranking Principle: For each subcollection, the subcollection's documents are ranked; the resulting ranked lists are combined into a final ranked list of documents,

Give a term to filter journal titles:
Information Retrieval

Information Retrieval
Volume 1
Volume 2
Volume 3
Issue 1
Issue 2
Issue 3

Found:8 2 Sort by: Author

A. Csillaghy; Hans Hinterberger; A. O. Benz
Content-Based Image Retrieval in Astronomy
ARTICLE

Christoph Baumgarten
Retrieving Information from a Distributed Heterogeneous Docu
ARTICLE

Elke Mittendorf; Bojidar Mateev; Peter Schäuble
Using the Co-occurrence of Words for Retrieval Weighting
ARTICLE

Elke Mittendorf; Peter Schäuble
Information Retrieval can Cope with Many Errors
ARTICLE

Grossman, Robert L.; Nerode, Anil; Ravn, Anders P.; Rischel, Han
Introduction
ARTICLE

Martin Braschler; Peter Schäuble
Using Corpus-Based Approaches in a System for Multilingual I
ARTICLE

Martin Wechsler; Eugen Munteanu; Peter Schäuble

ONLINE Search: Idle.

Fig. 2. Browsing for journal articles, following a link from an items detail view.

- *Author network based stratagems*, which are discussed in more detail in [9], address the interaction among the actors of scientific work and their strategic position in scientific collaboration networks. The central scenarios provided by DAFFODIL's interface are:
(1) *Author network browse:* When a user detects relevant documents in a result set, s/he can drop their authors onto the author network browser to

explore their collaboration network. The tool computes the co–author network of these authors and displays a list of authors in the structural neighbourhood of the starting authors, where relationships among the authors are represented as hyper-links. This allows the user to find further relevant documents of co–authors, resp. co–co–authors, whose documents have not been found before e. g. because of a mismatch of indexing and search terms. Since the list is ranked by the author's centrality in the network, the user may also find authors who are more central than the starting authors and can take them for further search activities.
(2) *Ranking a Document Result Set Using Author Centrality:* A specific problem of Daffodil is that most of the underlying services provide no ranking of the documents found due to the user's information needs. Thus, a given result set is re–ranked by descending centrality of the authors in co–author networks that are propagated on the basis of the documents in the result set.

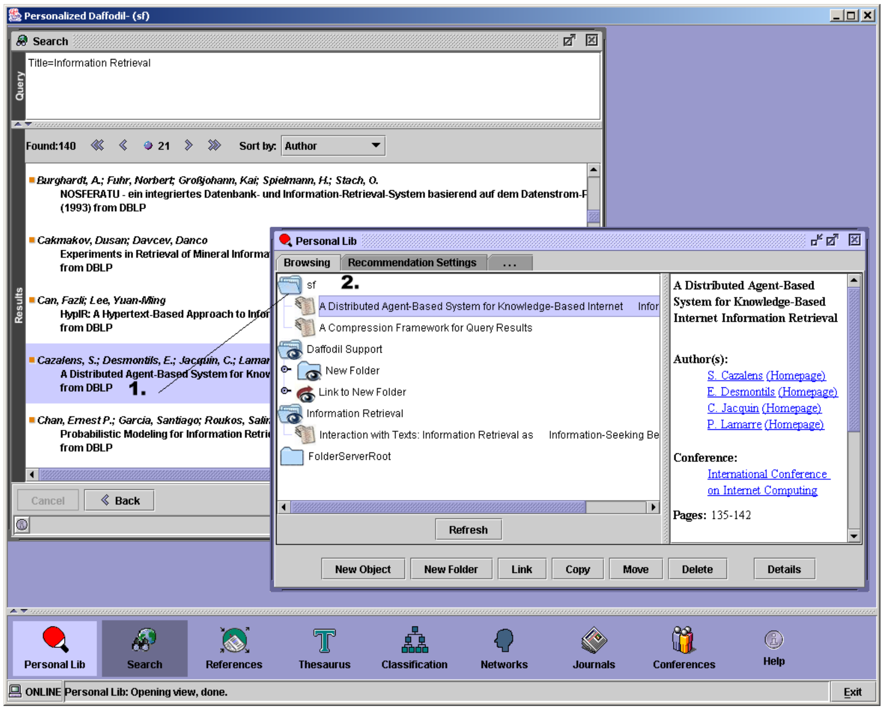
The quality of the ranked document sets was evaluated by a small retrieval test, which was conducted for ten queries on the basis of the German social science database SOLIS[9]. Each of the ten result sets was re-ranked by both *closeness*[10] and *betweenness*[11] centrality in co-author networks that were derived from the specific result set. For precision after the top 20 documents, this method produced a value of 0.58 on average. In contrast, average precision at 20 documents for the standard output of SOLIS was 0.28. This result demonstrates the strengths of ranking document sets by author centrality in co–author networks. Central authors seem to provide more relevant documents because of their key position in the network.

As a first step towards *personalisation*, we have integrated a personal library in the desktop, which supports individuals as well as groups (see Fig. 3). This tool allows for storing DL objects — documents, authors, journals, conferences as well as query formulations — in folders (via the standard Drag&Drop interaction). For any of these objects, alerting (awareness) can be activated; in this case, the user will be informed when the system has new information concerning this object (e. g. new publications by an author, a new issue of a journal, new references to a document, new answers to a query). So alerting is similar to an SDI service, but its invocation is very simple. For group folders, awareness will highlight objects which have been added, modified or annotated by other users.

[9] `http://www.social-science-gesis.de/en/information/SOLIS`.
SOLIS was chosen because of availability of qualified relevance judgements.

[10] *Closeness* relates to the number of shortest paths of an actor to all other vertices in the graph. This stresses the level of an actor's efficiency in scientific communication and collaboration networks

[11] *Betweenness* centrality focuses on the ratio of shortest paths an actor lies on. It indicates an actor's degree of control or influence of communication and collaboration processes.



Title=Information Retrieval

Found:140 21 Sort by: Author

Burghardt, A.; Fuhr, Norbert; Großjohann, Kai; Spielmann, H.; Stach, O.
NOSFERATU - ein integriertes Datenbank- und Information-Retrieval-System basierend auf dem Datenstrom-F
(1993) from DBLP

Cakmakov, Dusan; Davcev, Danco
Experiments in Retrieval of Mineral Informa
from DBLP

Can, Fazli; Lee, Yuan-Ming
HyplR: A Hypertext-Based Approach to Infor
from DBLP

Cazalens, S.; Desmontils, E.; Jacquin, C.; Lamar
A Distributed Agent-Based System for Know
from DBLP

Chan, Ernest P.; Garcia, Santiago; Roukos, Salin
Probabilistic Modeling for Information Retrie
from DBLP

sf
A Distributed Agent-Based System for Knowledge-Based Internet Infor
A Compression Framework for Query Results
Daffodil Support
New Folder
Link to New Folder
Information Retrieval
Interaction with Texts: Information Retrieval as Information-Seeking Be
FolderServerRoot

A Distributed Agent-Based System for Knowledge-Based Internet Information Retrieval

Author(s):
S. Cazalens (Homepage)
E. Desmontils (Homepage)
C. Jacquin (Homepage)
P. Lamarre (Homepage)

Conference:
International Conference on Internet Computing

Pages: 135-142

ONLINE Personal Lib: Opening view, done.

Fig. 3. Moving a document to the personal library

3 Evaluation

Two kinds of evaluation have been conducted during the project: A *heuristic evaluation* of the user interface and *questionnaire interviews* regarding system functionality. The results are described in the following.

After the first design and implementation phase, as soon as the SEARCH TOOL and the AUTHOR NETWORK TOOL were fairly stable, the interface was tested according to the *heuristic evaluation* method, as proposed by Nielsen [10]. Eight persons[12] explored the interface, while the experimenter took notes. Each problem reported by the participants was categorised according to a given list of usability heuristics.[13] The task was to search with the search tool and to apply the known item instantiation scenario, by using the AUTHOR NETWORK TOOL. This study produced the following findings:

a) *Irritations due to long waiting times*
While it was clear to the participants that a distributed search takes longer than a single search (besides waiting for the last response, re-integration and duplicate-elimination tasks in the middle layer took additional time), they

[12] A group of 5 computer science students and two academic assistants.

[13] e. g. `http://www.useit.com/papers/heuristic/heuristic_list.html`

became nervous after about 90 seconds and often interrupted the process. The need for a more transparent system state was expressed. The redesign addressed this need by cutting waiting time according to a user-defined option. The result list then contains only the documents retrieved so far. Profiling for the different wrappers will be used for dynamic adaption of filter settings in future. When a shorter maximum waiting time is specified, only those data sources are selected which are likely to respond within this time span.

b) *Interpretability of error cases*
Empty result sets posed a problem. It was unclear why the result set remains empty, as the middle layer was an additional possible source of problems. An agenda or log where one could look for possible explanations was requested. Visualisation of error states and counts for parts of the query or single data sources are needed to ease the interpretation.

c) *Acquaintance with new concepts*
The idea of using a high level tool, like the author network browser, was new to the participants. They needed time to explore the possibilities and asked for help regarding the interpretation of the results. However, when they learned how to use the tool, they expressed delight. The concept was quickly generalised and applied to the other browsers, which unfortunately had not been finished at that time. Using the rest of the GUI as mock–up, the participants followed the *tool-chain-idea* quite naturally, by applying intermediary results via drag and drop exchange or by following links.

d) *Author-networks: Computational effort and cut-off depth*
The cut-off depth of coauthor network computation had to be given by the users. It was not easy to decide, how to set this option. When choosing low values, computation was quick, but important authors were missing. Therefore the further development of a *main path analysis*, as described in [9] has been chosen to improve the situation: Instead of cutting the search at a given depth, local maxima for centrality values are computed and a hill climbing algorithm proceeds to the nearest local maxima from the starting point. This method reduces computing times while it allows to step much deeper into the social structure such that central authors who are located above a low cut-off point can be found.

After the second design phase, 14 computer science master students and six research assistants filled out a questionnaire after using Daffodil. The questionnaire was about ease of installation and usability of the prototype. It consisted of four parts: First, they were asked about their acquaintance with digital libraries. The second part concerned the installation of Daffodil via Webstart[14], an Internet installation tool for Java applications. For the third part participants had to perform small tasks with each tool. The last part was a complex task, with the intention to combine several tools in a strategic way. The goal was to find an overview article about “spatial data structures for multimedia data”. Other

[14] `http://java.sun.com`

information sources like the DBLP database and Google should be used and compared to Daffodil.

The results of the questionnaire are as follows:

a) *Library usage:* Usually, the participants were acquainted with the OPAC[15] of the local library and general web search engines like Google. Digital libraries, like DBLP[16] were only known to the research assistants, but usually not to the students.
b) *Installation:* The installation with WebStart on the operating systems Solaris, Windows and Linux was straight forward and caused no further problems.
c) *Tool testing:* 85% of all participants were able to answer the questions. The others complained about problems like faulty German umlaut handling in the author network tool. The problem of semantic identity of author names was discussed, as the service was sometimes not able to detect a person as unique due to differences in abbreviations and spelling.
d) *Complex task:* Only 50% of the participants found relevant articles. This was mainly due to insufficient query formulation and wrong keywords. In comparison, Google or DBLP appeared to provide an easier search, but resulted in less precise and very large result sets; so the identification of relevant documents in the result list was much harder.

One major problem was caused by the heterogeneity of query languages, where no correct translation was possible in some cases; post-processing of results will be a method for resolving this issue. Also, semantic heterogeneity caused unsatisfactory results (e. g. searching for a category of the classification scheme currently is supported by a single DL only); integration of appropriate transfer modules (like the ones described in [11]) could solve this problem. Missing functions and tools were requested, like a language translator, printing of results and a better help system (tool-tips). A personal library, which was not present at the time, was requested, to store results for later work. Also, additional pro-active integration of the tools was demanded, e. g. to combine the search tool with the thesaurus or a spell checker, if the query led to an empty answer. Nearly all participants said that, if a document is found, the quality of the detail view is very good.

The test results have been the basis for many improvements to the GUI. Most of the issues not explicitly named here are of technical nature. Thus, the tests helped substantially to improve stability and performance of the system.

4 Related Work

Several areas contributing to the digital library and agent-based information retrieval field are connected with Daffodil. In this section we give an overview on related work for integration of DL systems, agent based information retrieval and usability issues related to agent–user interaction.

[15] Online Public Access Catalog

[16] Digital Bibliography & Library Project: `http://dblp.uni-trier.de`

Integrated Digital Library Systems: The *Stanford Info-Bus*[17] is an early approach for integrating digital library catalogues and web sources into a federated system. With *DLITE* [12] it has an interface with several interesting design choices. It supports the search process and result (re-)presentation. Direct Manipulation is the main interaction style.

The *SketchTrieve* interface [13] enables the user to extract parts of information items. The collected items can be kept on a clipboard-like desktop for reuse in future work. Strategic support or high level services have not been integrated.

The new portal of the *ACM digital library*[18] offers an indexed collection with static links for searching, classification browsing, journal and conference browsing; however, searching on a subset of the collection selected by browsing is not directly supported. A personalised bookshelf and awareness on queries and group folders are also offered. However, DAFFODIL extends these features to federated information sources and guides the user with tools for query reformulation. Also the personalisation in DAFFODIL is based on all objects, not only queries.

The *Digital Work Environment* (DWE) [14] organises and facilitates various information sources in a tree-based interface to guide the staff and students through predefined tasks, like an exam paper preparation. Although tasks are similar to strategies, they are implemented in DWE as fixed schedules, allowing for no variations.

The *Ariadne* framework handles inconsistency between web sources, to enable information extraction, and to support wrapper re-induction for adapting to changes in dynamic sources [15].

Agent based information retrieval: The idea to employ agents for information retrieval tasks has a long history. Common approaches are WebBots or Spiders which collect information on the back end side, respective mediators and assistants on the front-end side. An example for the latest autonomous agent approach to support web searching is *InfoSpiders* [16]. Services like NECs *ResearchIndex* [17]also employ robots to search for publications and to automatically index them for building high level services, like citation tracking. The agents in *Margin Notes* [18] tries to recommend related information while traversing web pages.

Agent–User usability issues: Agents have been proposed as a means to reduce work and information overload [19]. Critics mentioned, that mediating agents reduce user control [20] and lead to lower predictability of user interfaces. However, current consensus is, that agents can provide helpful service, when acting in cooperation with the user within a common environment (for a discussion see [21]). The *Letitia* system offers helpful agents to support navigation in complicated hyper media spaces [22]. Agent based *just in time information retrieval*, is an unobtrusive interaction style [23], which is needed, as required by [24]; the latter paper also points out that trustworthiness of agents is a problem for users. *User autonomy* is crucial for work success and user satisfaction [25].

[17] For an overview refer to `http://www-diglib.stanford.edu/diglib`

[18] `http://portal.acm.org`

5 Conclusion and Outlook

In this paper we have presented the research and implementation state of the Daffodil user-interface along with evaluation results. The current version integrates search and browse of federated digital libraries within a graphical user interface, based on the WOB model. The tight integration of stratagems through the graphical tools enables efficient information searching with high-quality results. The key concept of *strategic support*, as proposed by Bates, is realized from the agent-framework up to the GUI. The prototype is freely accessible at `http://www.daffodil.de`.

In contrast to federal DLs based on standard protocols like e. g. Z39.50, Daffodil is able to integrate any DL that is accessible on the Internet. However, due to frequent changes of Web-based interfaces, continuous maintenance is required to keep the wrappers up-to-date.

Currently, we are working on the integration of intelligent components that help to further a search. A recommender–agent will analyse the result set and identify frequently occurring objects like e. g. conferences or journals. These objects will be offered to the user as links to the corresponding dedicated browsers. On link activation an optimised journal or conference run will give a specific relevance-driven view on the collections in question.

The personal library component implements personalisation and group support. Thus, we are following Paepcke's [26] vision of supporting the whole information search process consisting of the phases *Discover – Retrieve – Collate – Interpret – Re-Present*. Furthermore, the personal library will form the basis for implementing advanced features like pro-activeness and adaptivity. With pro-active agents, the system will be able to fill the personal library with more relevant data over time. Awareness-based services act upon insertion of new entries: When an entry is added, pro-active agents will become active and apply stratagems with the new entry as input, to add related items automatically (e. g. recommended readings). For pro-active search, the user can explicitly specify a depth search criterion, like the name of a known author, a topic of interest or a temporal range. Then an agent will search through all available journals and conferences (i.e. collections) for these criteria and return a list of matching articles, as well as branches of browsing trees, which contain data for the criteria. The agent will be able to monitor changes in the domain and notify the user about them. Then the user will only have to decide about acceptance or rejection of items on the recommendation list, rather than explicitly searching for them.

References

1. Fuhr, N., Gövert, N., Klas, C.P.: An agent-based architecture for supporting high-level search activities in federated digital libraries. In: Proceedings 3rd International Conference of Asian Digital Library, Taejon, Korea, KAIST (2000) 247–254
2. Bates, M.J.: Where should the person stop and the information search interface start? Information Processing and Management **26** (1990) 575–591

3. Krause, J.: Das WOB-Modell. In: Vages Information Retrieval und graphische Benutzeroberflächen: Beispiel Werkstoffinformation. Konstanz: Universitätsverlag, Konstanz (1997) 59–88
4. Sun Microsystems, Inc.: Java Look and Feel Design Guidelines. Addison Wesley Longman, Inc. (1999)
5. Fuhr, N.: Information retrieval in digitalen bibliotheken. In: 21. DGI-Online-Tagung – Aufbruch ins Wissensmanagement., Frankfurt (1999)
6. Baldonado, M.Q.W.: Interfaces for information exploration: Seeing the forest (1998)
7. Ellis, D.: Modeling the information-seeking patterns of academic researchers: a grounded theory approach. Library Quaterly **63** (1993) 469–486
8. Chen, H., Dhar, V.: Cognitive Process as a Basis for Intelligent Retrieval Systems Design. Information Processing and Management (1991) 405–432
9. Mutschke, P.: Enhancing information retrieval in federated bibliographic data sources using author network based stratagems. In Constantopoulos, P., Sölvberg, I.T., eds.: Reserach and Advanced Technology for Digital Libraries: 5th European Conference, ECDL 2001, Darmstadt, Germany, September 4-9, 2001; Proceedings. Lecture Notes in Computer Science; 2163. Springer, Berlin (2001) 287–299
10. Nielsen, J.: How to conduct a heuristic evaluation (1998)
11. Hellweg, H., Krause, J., Mandl, T., Marx, J., Müller, M.N., Mutschke, P., Strötgen, R.: Treatment of Semantic Heterogeneity in Information Retrieval. IZ-Arbeitsbericht; Nr. 23. IZ Sozialwissenschaften, Bonn (2001)
12. Cousins, S., Paepke, A., Winograd, T., Bier, E., Pier, K.: The digital library integrated task environment(dlite). In: 2nd ACM International Conference on Digital Libraries. (1997) 142–151
13. Hendry, D.G., Harper, D.J.: An informal information - seeking environment. Journal Of The American Society For Information Science **48** (1997) 1036–1048
14. Meyyappan, N., Al-Hawamdeh, S., Foo, S.: Digital work environment (dwe): Using tasks to organize digital resources. In Constantopoulos, P., Slvberg, I.T., eds.: Reserach and Advanced Technology for Digital Libraries: 5th European Conference, ECDL 2001, Darmstadt, Germany, September 4-9, 2001; Proceedings. Lecture Notes in Computer Science; 2163, Berlin, Springer (2001) 239–250
15. Knoblock, C.A., Minton, S.: The ariadne approach to web-based information integration. IEEE Intelligent Systems **13** (1998)
16. Menczer, F., Monge, A.: Scalable web search by adaptive online agents: An infospiders case study. In Klusch, M., ed.: Intelligent Information Agents. Springer (1999)
17. Lawrence, S., Giles, C.L., Bollacker, K.: Digital libraries and Autonomous Citation Indexing. IEEE Computer **32** (1999) 67–71
18. Rhodes, B.J.: Margin notes: building a contextually aware associative memory. In: Proceedings of the 5th international conference on Intelligent user interfaces, ACM Press (2000) 219–224
19. Maes, P.: Agents that reduce work and information overload. Communications of the ACM **37** (1994) 30–40
20. Shneiderman, B.: Direct manipulation for comprehensible, predictable and controllable user interfaces. In: 1997 international conference on Intelligent user interfaces, Orlando, FL United States (1997) 33–39
21. Shneiderman, B., Maes, P.: Direct manipulation vs interface agents. ACM Interactions (1997) 42–61
22. Lieberman, H.: Personal assistants for the web: An mit perspective. In Klusch, M., ed.: Intelligent Information Agents. Springer (1999)

23. Rhodes, B.J., Maes, P.: Just-in-time information retrieval agents. IBM Systems Journal **39** (2000) 685–??
24. Norman, D.: How might people interact with agents. Communications of the ACM **37** (1994) 68–71
25. Friedman, B., Nissenbaum, H.: Software agents and user autonomy. In Johnson, W.L., Hayes-Roth, B., eds.: Proceedings of the 1st International Conference on Autonomous Agents, Marina del Rey, CA USA, ACM Press (1997) 466–469
26. Paepcke, A.: Digital libraries: Searching is not enough. D-Lib Magazine **May 1996** (1996)