

Distributed agents for user-friendly access of Digital Libraries

Claus-Peter Klas Norbert Gövert Norbert Fuhr

University of Dortmund

{klas,goevert,fuhr}@ls6.cs.uni-dortmund.de

Despite the fact that many Digital Libraries (DLs) are available on the Internet, users cannot effectively use them because of inadequate functionality, deficient visualisation and insufficient integration of different DLs. In the framework of this project we develop a user-oriented access system for DLs which overcomes these drawbacks. Based on experiences from the librarian area, higher functions to assist proved search strategies will be implemented. Different DLs will be tightly integrated, so that system-wide search and navigation is possible. The system will be adaptive towards different user wishes, regarding preferences concerning content and system involvement.

1 Introduction

The Internet makes searching for literature in Digital Libraries (DLs) feasible. However there are certain problems: In extensive searches the user needs to contact several DLs. For this users need to know about their respective access points. Furthermore each of the DLs provide their own functionality, of course presented by different user interfaces. Examining functionality offered by todays DLs leads to the conclusion that only elementary search and browsing facilities are provided.

We propose an agent-based system which overcomes these drawbacks: The systems aims at providing access to functionality and data of arbitrary DLs through a single access point with a uniform user interface. Furthermore it implements functionality which gives users *strategic* support for satisfying their information needs. The nature of such functionality is described in Section 2. Section 3 describes the architecture of the system, based on the concept of agents.

2 Information search activities

In [Bates 90] a comprehensive systematisation of information search activities on four abstraction levels is given:

Moves are understood as basic units in the information search process. They correspond to the commands of information systems. Examples are adding a term or a condition to a given query or following a link.

Tactics comprise one or more moves with the purpose of improving or speeding a search.

One tactic is query refinement, e. g. generalising a search term in order to push recall or specialising a search term in order to improve precision.

Stratagems exploit data-structures of a given search domain, e. g. in a journal run (browsing a journal which is central to a given information need), in a footnote search (following links in a relevant document), or in an author search (searching for other publications of an author, who occurred earlier in one or more relevant documents).

Strategies are plans which comprise moves, tactics, and stratagems, in order to process a complete information search, i. e. cover an information need. A simple strategy for literature search in a novel area would be an area scan in a first step, followed by an author search with authors found in relevant documents resulting from the first step.

In our system we will implement functionality which strategically supports the user during her / his information search activities, i. e. we will provide search function on the level of tactics, stratagems, and strategies.

3 Architecture based on agents

In order to implement the concepts described above, we use an agent-based architecture. Due to the fact that we have a dynamic area in digital libraries with their singularities and heterogeneous, changing attributes we need a flexible agent architecture to cover all digital libraries and to be able to add new services to the system.

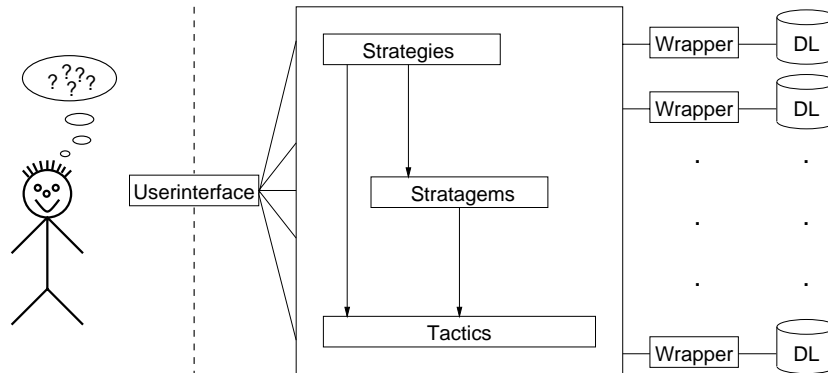


Figure 1: System-architecture

The user interaction with digital libraries via the agent system consists of four different levels (Figure 1). On the lowest level are the digital libraries which we will access via the WWW and which provide basic functions. By using the wrapper agent level transparent access to functions and data local to a given DL is possible. Communication of the user with the system is provided by an user-interface agent. The task of the user-interface agent is twofold: On the one hand it must present the functionality of the agent system to the user. On the other hand, it has to visualise the results adduced by the system.

Between user interface and wrapper reside the service agents, which offer the search functionality as described in Section 2. Every agent can subdivide his task in different tasks and use other service agents and the wrapper agents in order to fulfill them.

Cooperation problem To solve the cooperation problems of agents and to provide a flexible and extensible system a contract net protocol [Davis & Smith 83] will be used: A *manager*, responsible for a tactic or stratagem, subdivides a task into subtasks and announces them to a communication bus. All services which can solve the tasks, called potential *contractors*, *locally* evaluate their capability to respond and therefore bid or decline the task. The manager will evaluate the bids and award the task to one or more contractors, which will perform the tasks and report the result. The evaluation of the bids are *context sensitive*: in some cases the user is asked to decide, whereas in other cases the manager might decide on his own.

One example for a managers task is query refinement. The manager announce the task on the communication bus. All different services which implement tactics for query refinement, like relevance feedback or thesaurus to generalise or specialise terms, can bid based on there statistical quality to solve the task. The manager can then evaluate the bids with its known background and award the task to one or several of the bidding agents.

A different example is database selection, where locality and information / knowledge of the database is important. Centralised systems cannot reflect all local attributes and functions available in a database, since all off them can be different, like citation and full-text databases. So the manager can announce a task and only the databases which know how to provide the needed function will bit.

Pro-activity Another aspect of agents is that they can behave pro-active. *Pro-activity* means the ability of agents to trigger tactics or stratagems automatically, if certain preconditions hold. Suppose a user marked one author name twice, then the stratagem *author run* could be proposed (interactively or automaticly) and evaluated. The pro-activity is needed for novice users but also for experienced users. It can help novice users implicit to use all strategies available without knowing them. And experienced users can use them explicit to do the needed work and so save time in the search task.

4 Summary

The characteristics of agents like adaptiveness, the ability to communicate and autonomy [Nwana 96] allow in a flexible way to develop a system providing higher search functions for digital libraries. Such functionality enables the system to support the user *strategically* in their information search activities.

Some of the aspects of the system presented here are currently implemented by a student project group.

References

- Bates, M.** (1990). Where Should the Person Stop and the Information Search Interface Start? *Information Processing and Management* 26(5), pages 575–591.
- Davis, R.; Smith, R. G.** (1983). Negotiation as a Methaphor for Distributed Problem solving. *Artificial Intelligence* 20(1), pages 63–109.
- Nwana, H. S.** (1996). Software Agents: An Overview. *The Knowledge Engineering Review* 11(3). <http://www.cs.umbc.edu/agents/introduction/ao/>.