THE UNICATS APPROACH -
NEW MANAGEMENT FOR BOOKS IN THE INFORMATION MARKET

Michael Christoffel[1], Sebastian Pulkowski[2], Bethina Schmitt[1], Peter Lockemann[1], Christoph Schütte[2]

[1]Universität Karlsruhe, Fakultät für Informatik
D-76128 Karlsruhe, Germany
email {christof, schmitt, lockeman}@informatik.uni-karlsruhe.de

[2]Universität Karlsruhe, Universitätsbibliothek
D-76128 Karlsruhe, Germany
email {pulkowski, schuette}@ubka.uni-karlsruhe.de

Abstract

The WWW offers an increasing number of services for searching and acquiring scientific literature. New kinds of information providers are available online, traditional libraries follow and offer their services via the WWW. However, this development will not come without a price: Literature supply is changing into an open and competitive information market bringing new challenges to the user. To derive the whole benefit from the new situation in the electronic literature market, the user needs assistance from an integration environment. This environment regains the transparency in the information market so that the user receives comparable offers, and integrates different services of scientific literature search and delivery under a uniform user interface. To provide such an environment is the aim of the UniCats project.

1 Introduction

The WWW offers an increasing number of services for searching and acquiring scientific literature, with a major impact on users in the university environment:

- The traditional libraries have entered the Internet and offer their catalogues online. These catalogues contain bibliographic information and loan possibilities, sometimes also include electronic order and delivery.
- There are bibliographic databases on special topics such as INSPEC [FIZ], which contain a huge amount of documents with their complete bibliographic references or even full texts.
- Technical report servers (such as NCSTRL [NCST]) contain full texts of gray literature.
- Publishing houses (such as Springer [SPRI]) and online bookstores (such as Amazon [AMAZ]) provide electronic ordering, and offer additional information on the documents such as abstracts, cover arts, table of contents, or reviews.
- Electronic document delivery services, such as Subito [SUBI], load down any article of a journal in electronic format.

Undoubtedly, this development has fundamentally improved the availability of literature. The user can find, order and obtain the desired documents without leaving his/her desk.

But this development will not come without a price: We observe the commercialization of the Internet, and this will lead to a commercialization of the literature supply. Internet providers demand a prize for their services. And even public and university libraries will not continue to offer their services for free. The information “buffet” changes into an *information market* where the value of a piece of information is determined by the cost of producing it and the law of supply and demand.

In this environment, a user must become much more market-conscious:

- First of all, he/she has to know the available services. However, it is not easy to be up to date and stay up to date in this booming area.
- Second, the user must evaluate the different services: he/she has to initiate search requests in different providers, manually and sequen-

tially. This is often an annoying and time-consuming business.

- Third, he/she has to be able to deal with the services. But every Internet provider has its own kind of user interaction with its own user interfaces and input formats. The user is confronted with several dialog languages and different result formats. Further, to increase the complications, the payment methods differ from one provider to the next: the user has to leave the credit card number, has to use a special kind of electronic cash, or has to open an account at the provider.
- Fourth and most important, the user must compare different offers directly with each other in order to be able to take a decision. There is no help for detecting the differences between two services, and comparing delivery times and delivery procedures, or the subtleties of the pricing structure. There is a tendency for the user to take one of the first offers he/she finds rather than the optimal offer.

To derive the whole benefit from the new situation in the electronic literature market, the user needs assistance from an integration environment. This environment regains the transparency in the information market so that the user receives comparable offers, and integrates different services of scientific literature search and delivery under a uniform user interface. To provide such an environment is the objective of the UniCats project.

The UniCats project (a UNiform Integration of Catalogues based on an Agent-supported Trading and wrapping System) [UNIC] has started in the year 1998 at the University of Karlsruhe in cooperation between the Institute for Program Structures and Data Organization and the University Library. It is supported by the Deutsche Forschungsgemeinschaft (DFG) as a part of the strategic research initiative V^3D^2 (Verteilte Vermittlung und Verarbeitung Digitaler Dokumente) [V3D2].

The paper is organized as follows. The next section gives an introduction to the market for scientific literature and determines the requirements for the architecture of the integration environment which is then discussed in section 3. The architecture includes three main system components, user agent, trader, and wrapper, which are considered in sections 4, 5 and 6. Section 7 reviews some other projects dealing with the integration of library services. The final section concludes with a summary and suggestions for further work.

2 An Electronic Market for Literature

As described in [Chr98], the situation of supplying scientists with literature is expected to develop into an open market of services, regulated by the law of supply and demand. Each service provider is allowed to take part in the market and to leave it at any time.

It is also a very heterogeneous environment, because the providers and the customers in this market are distributed by locality and, as a consequence, offer diverging data access and exchange services, query and result formats.

The market can be called non-transparent, too. Neither customers nor providers can currently gain a view of the total environment.

In spite of the problems listed in the first section, we believe that this new market economy will bring benefits to both users and libraries. Competition will force providers to improve their services continuously on behalf of the user. Decreasing budgets will force traditional libraries to follow suit and charge for their services in the near future. This means that traditional libraries will have to learn to survive within such a market; they must get a feeling for the value of information and information services which a user is willing to pay for.

The idea of this open, heterogeneous and non-transparent market marks the conditions of the integration environment. We need a flexible architecture, open for modifications and extensions, and independent from concrete computer platforms. We need tailored interfaces towards customers and providers. The customer-side interface should be based on an understanding of the user's intention and present itself to him/her such that he/she does not notice that he/she is actually working with a variety of services. The provider-side interface must not set excessive conditions to the source in order to guarantee the open character of the market. And we need a component that surveys the market and produces the desired transparency.

The *UniCats environment* provides a technical platform that supports the structure and the dynamics of an information market. Providing the desired transparency by knowing the available services is the job of our traders, adapting to the user and integrating the different services is the task of user agents, and easing the connection of providers is the objective of the wrappers. But only by cooperation of all system components will the user profit from the potential of an open market.

3 Architecture

In this section, we describe the architecture of the UniCats system and the interaction between the system components, as illustrated in figure 1.

3.1 Components

According to the previous section, an integration environment inside an information market is managed by three main components: an interface component on the customer side, an interface compo-

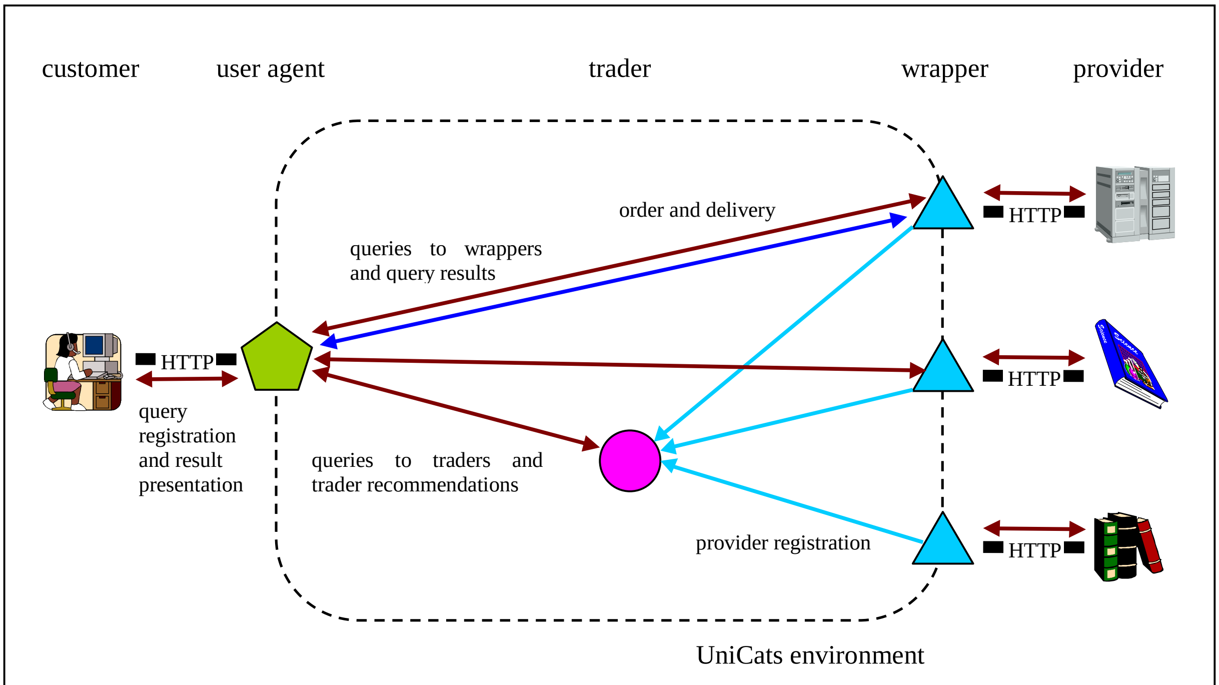

customer
user agent
trader
wrapper
provider
order and delivery
queries to wrappers and query results
HTTP
HTTP
HTTP
HTTP
query registration and result presentation
queries to traders and trader recommendations
provider registration
UniCats environment

Figure 1: The UniCats environment

nent on the provider side, and an intermediary component establishing the contact between customer and provider.

The three main components of the UniCats environment are the following:

- At the one end there are *user agents* which offer the customer a uniform access point with an appropriate user interface. But user agents are much more than the interface towards the customer. They develop plans to transform the customer's demand into different queries representing the interests of the customer, e.g., minimizing the costs, and they contain algorithms to integrate results from different sources.
- At the other end there are *wrappers* as the proxies for the service providers in our system. Their task is to translate queries into the special format of an information source and results back from the format of the source into the result format of the UniCats environment. They are, however, more than simple translators, since they contain modules for cost estimations, query organizing, and access-control.
- In between there are *traders*. Traders know the available service providers and match incoming user demands with existing service offers to find those providers most appropriate to the user query.

Obviously, user agents and wrappers work best if they are adapted to their individual customers and service providers. To obtain a truly open market by placing as few obstacles to the providers as possible, we provide a wrapper generator to tailor-make wrappers. This is not possible for user agents due to their larger individuality. We overcome some of the costs by using group profiles including special characteristics of the users, e.g., students or junior research assistants.

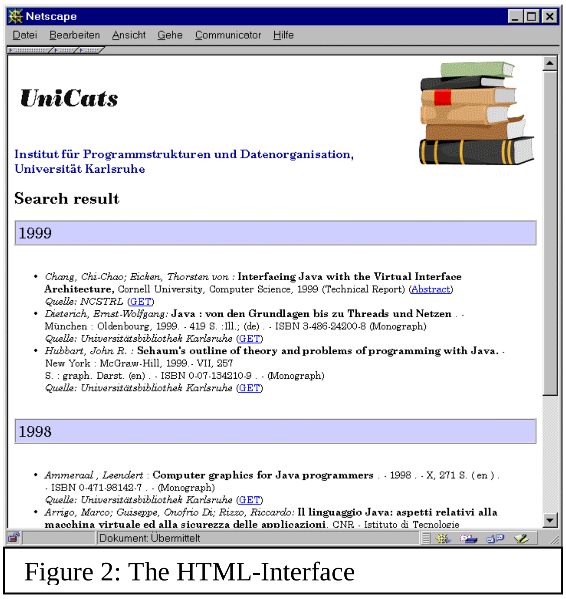
An important consideration for our environment is that we expect an open market of user agents, traders and wrappers as well, i.e., there may be any number of them which compete with each other.

To place user agents, traders and wrappers, we need a uniform framework, where user agents, traders and wrappers are all instances of *UniCats agents* [Nim99]. The framework overcomes the heterogeneity of the market and establishes communication and data transfer between the system components. The capability for electronic commerce is achieved by use of an extension of the Open Trading Protocol [OPT]. The open market forces us to a rather low-level exchange of XML documents based on TCP/IP.

The communication between user agent and customer, and wrapper and service provider, respectively, is based on HTTP. This is the protocol available to practically all, customers and providers alike, since customers can use their favorite web browser to access the UniCats system, and providers can use a standard HTML site they usually already have.

3.2 Cooperation among UniCats Components

Figure 1 illustrates the communication principle and information flow inside the UniCats environment.



UniCats

Institut für Programmstrukturen und Datenorganisation,
Universität Karlsruhe

Search result

1999

- *Chang, Chi-Chao; Eicken, Thorsten von :* **Interfacing Java with the Virtual Interface Architecture,** Cornell University, Computer Science, 1999 (Technical Report) (Abstract)
Quelle: NCSTRL (GET)
- *Dieterich, Ernst-Wolfgang:* **Java : von den Grundlagen bis zu Threads und Netzen** . - München : Oldenbourg, 1999. - 419 S. :Ill.; (de) . - ISBN 3-486-24200-8 (Monograph)
Quelle: Universitätsbibliothek Karlsruhe (GET)
- *Hubbart, John R. :* **Schaum's outline of theory and problems of programming with Java.** - New York : McGraw-Hill, 1999.- VII, 257
S. : graph. Darst. (en) . - ISBN 0-07-134210-9 . - (Monograph)
Quelle: Universitätsbibliothek Karlsruhe (GET)

1998

- *Ammeraal , Leendert* : **Computer graphics for Java programmers** . - 1998 . - X, 271 S. (en) . - ISBN 0-471-98142-7 . - (Monograph)
Quelle: Universitätsbibliothek Karlsruhe (GET)
- *Arrigo, Marco; Guiseppe, Onofrio Di; Rizzo, Riccardo:* **Il linguaggio Java: aspetti relativi alla macchina virtuale ed alla sicurezza delle applicazioni**. CNR - Istituto di Tecnologie

Figure 2: The HTML-Interface

A wrapper has to register with a trader (or more than one trader) to become known within the system and to announce its offers. While doing so, the wrapper gives some metadata about its source, such as number of documents, languages, cost model, or attributes the user can search for.

When given a user query, the user agent contacts a trader asking for services appropriate for this special query. The trader answers with a list of recommendations for service providers together with available additional information, e.g., the estimated cost for an access to this source.

Thereafter, the user agent addresses the suggested services, actually the wrappers. These transform the query into the native query format of the information source and send the re-transformed query results back to the user agent. The user agent collects the results from the different sources, integrates them, and presents a final result collection to the user.

When the user decides to order a document, this order will be transmitted by the user agent to the selected document source. Documents which are available online will be delivered immediately, otherwise delivery (or loaning in case of a library) will be initiated.

Of course, interaction between UniCats components may be much more complicated. A user request may result in a sequence of trader and wrapper queries. For example, only a short version of the bibliographic data may be presented to the user in a first step, and the full version will be provided in a second step after selecting titles. To improve the quality of the traders, we add a feedback mechanism. The user agent analyses the user's behavior and sends the results of this analysis to the trader, e.g., the number of visited documents, or the timed needed for an access.

In the long run, an open market must include financial transactions. Consequently, we currently prepare our platform to include, in the near future, functions for accounting, billing and electronic payment. UniCats is supposed to act as an intermediary for payment transactions, providing more flexibility in the choice of payment systems both to customers and providers. Additionally, providers profit from dealing with a trusted organization, and customers benefit from better prizes the UniCats system can negotiate.

4 User Agents

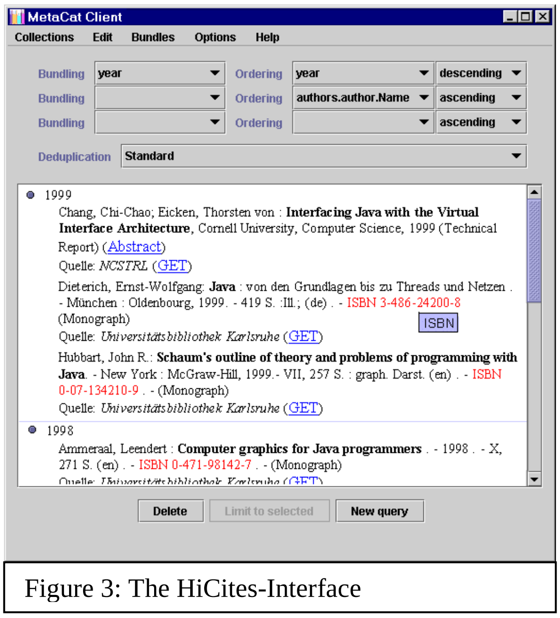
In this section, we take a more detailed look at the user agent.

The user agent coordinates the execution of the search process. To integrate sources with divergent semantics expressed by different source structure schemas, the user agent, after collecting the results from the various providers, must bring them into a homogenized form before presenting them to the user. Homogenization is based on a unified "global" schema.

4.1 Result Integration

The parallel search in different sources has lots of advantages but also causes problems: Results must be combined, duplicates recognized and merged.

Simple duplicate detection algorithms are based on checking the ISBN or the URL of the title. Since these are not universally available or not necessarily unique, duplicate detection must be tailored to the specific situation. Therefore, we use a generic detection algorithm. The user may influence this process. Usually, two documents from different editions are treated as equal. But some users rely on a special edition, so a duplicate detection algorithm



Bundling year Ordering year descending
Bundling Ordering authors.author.Name ascending
Bundling Ordering ascending

Deduplication Standard

● 1999

Chang, Chi-Chao; Eicken, Thorsten von : **Interfacing Java with the Virtual Interface Architecture**, Cornell University, Computer Science, 1999 (Technical Report) (Abstract)
Quelle: *NCSTRL* (GET)

Dieterich, Ernst-Wolfgang: **Java** : von den Grundlagen bis zu Threads und Netzen . - München : Oldenbourg, 1999. - 419 S. :Ill.; (de) . - ISBN 3-486-24200-8 (Monograph)
ISBN
Quelle: *Universitätsbibliothek Karlsruhe* (GET)

Hubbart, John R.: **Schaum's outline of theory and problems of programming with Java**. - New York : McGraw-Hill, 1999.- VII, 257 S. : graph. Darst. (en) . - ISBN 0-07-134210-9 . - (Monograph)
Quelle: *Universitätsbibliothek Karlsruhe* (GET)

● 1998

Ammeraal, Leendert : **Computer graphics for Java programmers** . - 1998 . - X, 271 S. (en) . - ISBN 0-471-98142-7 . - (Monograph)
Quelle: *Universitätsbibliothek Karlsruhe* (GET)

Figure 3: The HiCites-Interface

that does not pay attention to the number of the edition would be contrary to the user's intention.

On duplicate elimination, one cannot simply delete one of the duplicates but must check whether a provider may have some information about a document which is missing in other sources.

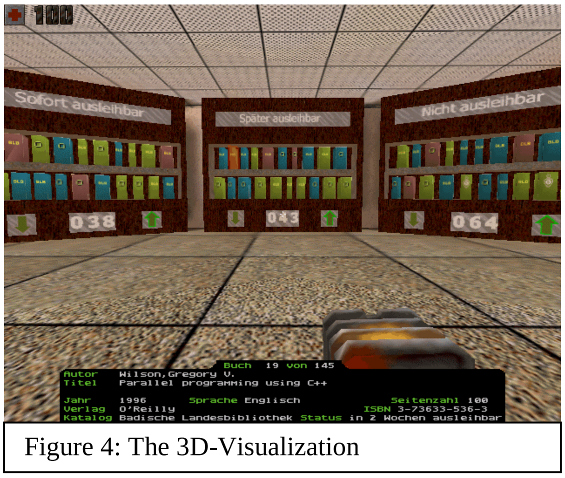
To offer post-processing functionality, the user agent contains flexible algorithms for grouping and sorting the final result collection. Figure 3 gives a simple example where the results are grouped by year and ordered by the name of the first author.

4.2 Result Presentation

The presentation of the query results should be independent from the internal representation of the data. The MVC design pattern meets this requirement. It allows us to use different kinds of user interfaces and presentations, which may be tailored to different kinds of user groups, or may be preselected by an individual user.

Developing and analyzing different user interfaces is the aim of our planned studies. To do so, we implemented three kinds of user interfaces:

- The simplest, most common and also the fastest one is the presentation by a structured HTML-text (figure 2). The complete information is presented to the user in one though organized piece. Of course, for large numbers of results, this approach will be difficult to inspect.
- A different idea is a tree-representation with Hi-Cite functionality: In figure 3, the documents are displayed as leaves in a tree and sorted, for example, by year.
- A third kind of results presentation uses a physical metaphor for visualization of the result set. We implemented a 3D-model of a library using Quake II (figure 4). Although this kind of user interface tends to be slower than the interfaces mentioned before, we believe that for a beginner it offers an attractive first contact to get the flavor of the system.



Buch 19 von 145
Autor Wilson,Gregory V.
Titel Parallel programming using C++

Jahr 1996 Sprache Englisch Seitenzahl 100
Verlag O'Reilly ISBN 3-73633-536-3
Katalog Badische Landesbibliothek Status in 2 Wochen ausleihbar

Figure 4: The 3D-Visualization

5 Traders

This section deals with the traders and explains the way they establish market transparency. A trader holds the connection to the providers which have been registered with it. It may be in contact with other traders, too. We foresee traders to be joined in a hierarchically *organized trader federation*. The principal architecture of a trader is shown in figure

5.1 Provider Selection

The task of the UniCats trader is to mediate providers that deliver the desired piece of information according to the conditions preferred by the user for a fair price. Traders perform a uniform query handling considering various attributes: content-based attributes such as the kind of the provider and the covered thematic areas, financial aspects such as an estimation of the costs for an access to a source, and quality of service issues such as the delivery time. We also use restricting attributes interpreted by the trader itself; as an example, the user can set a limit for the trader charge. Traders need not necessarily recognize the same set of attributes. As a consequence, traders may be confronted with unknown attributes. The process of query handling is described in [Chr99]. An important aspect of query handling is that traders keep profiles of providers and other traders.

We see a need that traders have some learning capacity. It should make a use of the experience the trader acquires when handling a query, especially any feedback by the user agent. Additionally, a trader may gain in experience by sending test queries to the registered wrappers.

5.2 Trader Federation

Traders may differ in their capabilities. Hence, traders should cooperate in order to improve the overall trading capabilities. To do so, they may join in a trader federation. A trader may then send subsidiary queries to their federated traders, and construct the final recommendations from the returned query results.

The main reason for a trader to join a *trader federation* is that traders can specialize in a provider class (such as libraries or providers that primary deal with documents about computer science). As a consequence, the trader must receive assistance in query handling from the whole federation, including help in finding providers to recommend. To obtain the desired service, a user will then address a trader federation (e.g., by addressing any federation member).

Traders should be able to administrate their federation on their own. To establish this, we have developed centralized and decentralized approaches.

The *centralized* administration requires a special facility inside the federation, called a *central trader*. This may be the root of the hierarchy, but quite often it is not a trader at all. Knowing the profiles of every federated trader, a central trader can organize the structure of the hierarchy according to the specialization of the traders. It can also perform the additional task of routing provider and user contacts to the members of the federation. The visual weakness of the centralized approach is the bottleneck characteristics of the central trader.

In a *decentralized* federation, there is no administrating facility other than the members of the federation themselves. Each trader autonomously decides to join or leave a federation, to specialize by exchanging providers or to change its position in the trader hierarchy. In our approach, no trader has an complete picture of the structure of the federation; rather only knows its neighbors. A trader should be able to generate a new trader, too.

5.3 Trader Finances

Since the use of an information source will cost money even to the trader, the trader must find a way of getting the money back. Moreover, the author of the trader wants to make some profit from installing the trader into the net. Hence, a trader must be able to charge for its services as well.

Charges may be billed to the customer, the provider (which benefits from being recommended), or both of them. To determine which cost models perform best in an information market of university users, we plan to do some experiments after completing the system. Our initial hypothesis is that a service-rendered model is fair to the customer: He/she has to pay for each recommended information source. For the provider, we tend to a flat rate model. The provider pays for being made known by the trader during a period of time. After expiration of this period, the provider may decide to extend or to cancel the contract.

Alternatively, financing of a trader may be by advertisement. For instance, a customer receives the services of the trader for free, but he/she has to accept that each query result contains a commercial message.

5.4 Quality of the trader results

The quality of the results of two traders may differ. A reason for this is that the address or the profile of a provider has altered, and the trader was not notified. Moreover, the introduced concepts are based on an ideal market model where all system components are well-meaning and cooperative, and the traders set aside their own financial interests for the user's benefit. But a real market will bring some additional complications:

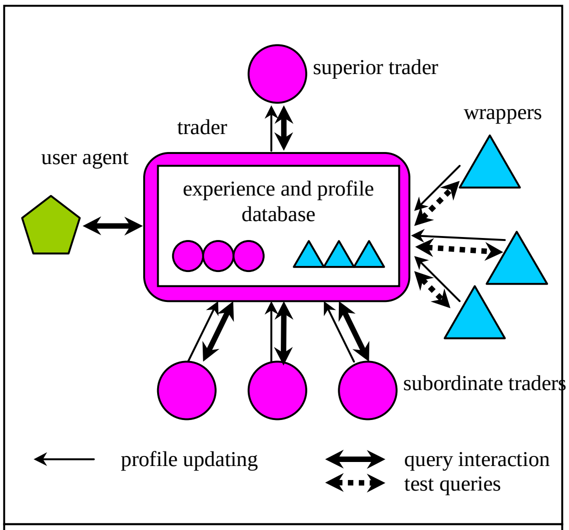


Figure 5: Trader architecture

- Traders can be corrupted by the providers. So provider selection may be everything but fair.
- The traders might try to maximize their profit when handling a query.
- Competitive traders could try to harm each other instead of working together.

One solution for these problems is to install a certification facility into the system. Traders may register with this facility to obtain the certification, and they will lose the certification if the quality of their recommendations falls below a limit. Alternatively, an evaluation facility may be introduced, which periodically tests the traders of the system and publishes the results of these tests on a web site, accessible by both customers and providers.

6 Wrappers

The main problem with the information sources is their wide variety regarding the content and the presentation of data. If we wish to install the system for an open market, we cannot assume that each information source is willing or even capable of creating a wrapper for its site on its own. Instead, a tool should be provided to generate wrappers. We refer to the tool as a wrapper generator.

One particular difficulty with wrapper generation is that information about documents is usually not on a single page but distributed across a sequence of pages. Therefore, we need a wrapper which is capable of also navigating in the information source.

6.1 Wrapper Generation

Creating wrappers manually is hard work. We have to know a lot about programming languages, regular expressions, or security mechanisms. A wrapper tailored to one site is hardly reusable at another site.

A wrapper generator should ease the task for a particular source, but should nonetheless be receptive to the idiosyncrasies of the source. We base our approach to the concept of wrapping by example. A person controlling the wrapper visits all relevant pages of a site, and marks the important parts. With this information the generator produces the wrapper.

Parallel with this “teach-in”, metadata about the source and its content are extracted. Metadata include, for example, the attributes used, the number of documents or the included languages.

6.2 The Architecture of a UniCats Wrapper

The generated wrapper includes dynamically several modules which can be configured into it at runtime. These modules have specific functions which are not common to all wrappers, such as the administration of logins, encoding of data, or consideration of costs. This enables the wrapper to be flexible without incurring too much overhead.

The wrapper itself has the capability of planning a search before the final execution. This allows the wrapper warn a user that a given cost limit may be insufficient to execute the query. Thus, the user can increase the limit or cancel the request before incurring the fees. Of course, this requires a wrapper to be fair-minded to the user.

Another relevant feature of our wrapper is location independence: With the same wrapper generator both, the source and the creator of the user agent can build their own wrappers. The only requirement is the existence of an HTML-Interface at the source site.

7 Related Work

The Karlsruher Virtueller Katalog [KVK] is a very successful meta-engine for literature search. It combines multiple university catalogues with a uniform user interface. However, it can only offer a limited presentation of bibliographic data such as author, title and year. Also, it neither offers duplicate detection nor post-processing operations. Results are displayed in a standard text-based HTML interface only. To obtain further details about a document, the user has to manually follow additional links, leading to the providers of this information. Cost-based information search is not available, and there is no support either for acquiring documents or direct document comparison.

One of the most important approaches in the integration of heterogeneous bibliographic information sources is the Stanford Digital Library Project [SDLP]. This project extends search an delivery to areas such as multimedia documents and graphical data, whereas our project is restricted to text-based documents. The Stanford project offers a large toolkit of different utilities including multiple user interfaces. Proxies translate between the native protocol of the installed databases and the internal protocol DLIOP. So, including a database requires the cooperation of the operator of this source to install a special interface. The tool gGLOSS [Gra95] offers a trading service that is based on automatically generated glossaries for selected databases. UniCats traders, on the other hand, use characterizations for the service providers that are gained by metadata obtained by the providers themselves or by experience, and consider a large range of criteria for selecting providers, that is not limited to content-based aspects.

The digital library project at the University of Michigan [Wel96] is based on agents, too. It uses a three-tier architecture consisting of user interface agents, mediator agents, and collector agents. The considered data may contain pictures, audio files and geographical data - among others. This makes the Michigan project much broader than our project. However, the integration of a data source forces the cooperation of the provider in installing a special interface.

Likewise, a three-parted architecture is used by Medoc [Bar98, MEDO]. User agent, provider agent and brokers enable search and delivery from literature concerning computer science, which in some cases is available online on a special document server. The aim of the UniCats project is not to set up a new provider, and it is also not limited to a special scientific area. Provider selection in the Medoc brokers is based on estimated costs, whereas our traders may be advised to consider the location of the provider and to prefer local providers, for instance. Also, the Medoc project needs the assistance of the providers to integrate a service. In Medoc, a business model has been developed to control financial and legal aspects between the providers and the Medoc service.

8 Conclusion and Future Work

In this paper, we presented a platform for literature search in an open and heterogeneous market with different information providers.

Our initial experiences suggest that UniCats provides a viable approach. For one, it appears that work of companion projects in the V^3D^2 special initiative can easily be integrated, and, hence, in the longer term, a full range of services around digital libraries can be offered to the users. For another, we have found that the architecture is flexible enough to divide the necessary work into small tasks that can be processed independently and thus in parallel.

We consider UniCats to be a platform into which to integrate other services provided by our partners in the collaborative DFG program, and with which to conduct extensive experiments on the acceptance of cost models, trading functions, dynamic evolving query scripts, ease of provider attachment, user interfaces, selectivity of results, and the like.

References

[AMAZ] *Amazon*. http://www.amazon.com

[Bar98] A. Barth, M. Breu, A. Brüggemann-Klein, A. Endres, A. de Kemp: *The MeDoc Digital Library Project: Its Goals and Major Achievemants.* In: Digital Libraries in Computer Science: The MeDoc Approach, Lecture Notes in Computer Science, 1-9, 1998

[Chr98] M. Christoffel, S. Pulkowski, B. Schmitt, P. Lockemann: *Electronic Commerce: The roadmap for university libraries and their customers to survive in the information jungle*. In ACM Sigmod Record, 68-73, December 1998

[Chr99] M. Christoffel: *A Trader for Services in a Scientific Literature Market*. In: Proceedings of the 2nd International Workshop on Engineering Federated Information Systems, Kühlungsborn, 1999

[FIZ] *Fachinformationszentrum Karlsruhe.* http://www.fiz-karlsruhe.de

[Gra95] L. Gravano, H, Garcia-Molina: *Generalizing GLOSS to Vector-Space Databases and Broker Hierarchies.* In: Proceedings of the 21st VLDB Conference, 78-89, Zürich, 1995

[KVK] *Karlsruher Virtueller Katalog*. http://www.ubka.uni-karlsruhe.de/kvk.html

[MEDO] *MeDoc - Die elektronische Informatikbibliothek.* http://medoc.informatik.tu-muenchen.de/deutsch/medoc.html

[NCST] *NCSTRL.* http://www.ncstrl.ubka.uni-karlsruhe.de:8080/

[Nim99] J. Nimis: *Konzeption und Implementierung eines agentenbasierten Electronic Commerce Frameworks für digitale Bibliotheken*. Diplomarbeit, University of Karlsruhe, 1999

[OTP] *Open Trading Protocol.* http://www.otp.org

[SDLP] *The Stanford Digital Library Project.* http://www-diglib.stanford.edu/diglib/

[SPRI] *Springer Online Catalogue.* http://www.springer.de/cgi-bin/search_main.pl?bookdealer=Springer

[SUBI] *Subito.* http://www.subito-doc.de

[UNIC] *Homepage of the UniCats project.* http://wwwipd.ira.uka.de/~unicats

[V3D2] *Homepage of the strategic research initiative V^3D^2.* http://www.cg.cs.tu-bs.de/dfgspp.VVVDD

[Wel96] M. Wellman, E. Durfee, W. Birmingham: *Trends & Controversies - The role of AI in digital libraries.* In: IEEE Expert - Intelligent Systems and Their Applications, 8-13, 11(3), 1996