# MeDoc Information Broker—Harnessing the Information in Literature and Full Text Databases

Dietrich Boles*
Markus Dreger†
Kai Großjohann‡

September 16, 1996

**Introduction.** MeDoc is a two-year project sponsored by the German Secretary for Education and Research which has begun in September 1995. The project is being led by the *Springer Verlag*, the *Fachinformationszentrum Karlsruhe*, and the *Gesellschaft für Informatik*, its participants are six German universities. Further, there are about twenty pilot user institutions so that the system being developed can be tested for usability right away. The goal of MeDoc is to build the MeDoc system which is to help Computer Science researchers and students (primarily in Germany, but also all over the world) to find information relevant to them. This aim is going to be achieved by two means. Useful information will be provided to the researchers and students by building a full-text database of "critical mass"; on the other hand, the MeDoc Information Brokering System (IBS) will provide transparent access to existing bibliographic and full-text databases. The paper presented focuses on the latter part, the IBS.

**Problems addressed.** Creating the IBS involves several kinds of problems:

- Considering the number of existing full-text and bibliographic databases it will not be sufficient just to broadcast queries to all connected provider systems (databases). Therefore, a mechanism for provider selection has to be implemented.
- As the connected provider systems cannot be expected to have a homogeneous schema, a facility for schema transformation has to be provided. This facility should properly take into account the vagueness and uncertainty inherent to IR applications.

*OFFIS Oldenburg, `Dietrich.Boles@OFFIS.uni-oldenburg.de`
†FU Berlin, `dreger@inf.fu-berlin.de`
‡Uni Dortmund, `grossjohann@informatik.uni-dortmund.de`

- The distribution of the databases directly implies that a mechanism for merging the result sets produced by the different providers has to be developed.

**Problems not addressed.** The presented paper does not address the problems of accounting and security which are nevertheless recognized as being very important.

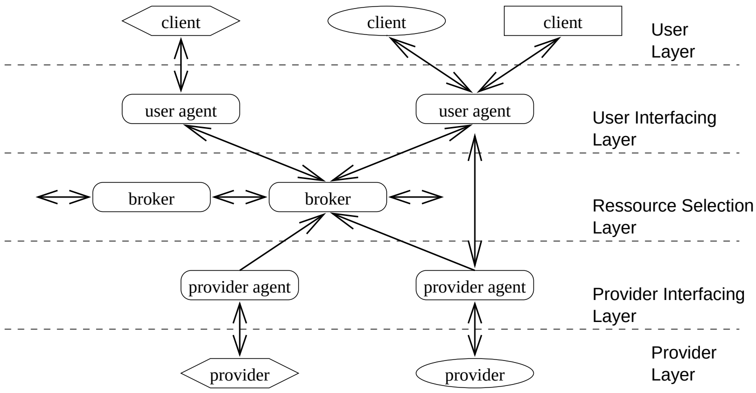


Figure 1: The Architecture of the MeDoc IBS

**Architecture.** Figure 1 shows the architecture of the IBS. It comprises several layers. The uppermost layer is the *User Layer* which contains the software that is being used by humans to interact with the system. This layer is really outside the IBS. Standard WWW, Hyper-G or email clients are used to access the system. The *User Interfacing Layer* accepts requests from a client system, transforms them into the internal format used by the MeDoc system and communicates them to the other layers of the IBS. It also accepts responses from the other parts of the system and transforms them into a user readable format. The *Ressource Selection Layer* is optionally contacted by the User Interfacing Layer to determine what ressources to contact in order to fulfill a given request. The *Provider Interfacing Layer* is responsible for the transformation of requests from other layers into the particular formats needed by the database systems; it also transforms the responses from the database systems back into an IBS internal format. The *Provider Layer* contains the full text and bibliographic databases themselves (and lies outside the IBS, just like the User Layer).

**Components of the Architecture.** Every layer will be implemented in a distributed way. The User Interfacing Layer contains a number of *User Agents*, where each User Agent services several users but a user is expected to contact one User Agent only. Likewise, each database is associated with a *Provider Agent*. All Provider Agents comprise the *Provider Interfacing Layer*. Finally, the Ressource Selection Layer will also be implemented in a distributed way and contains so-called *Brokers*.

Provider Agents are responsible for schema transformations and meta data extraction. Users formulate their queries with respect to a global schema. The queries are transformed into the schema of each provider system by the Provider Agents. Likewise, the User Agents deal with documents and document references with respect to the global schema; the Provider Agents are responsible for transforming them into that global schema. The meta data the Provider Agents extract from the provider systems are used by the Brokers to determine how many documents to request from each provider for any given query. In addition to transforming queries and results between the user readable and the internal formats the User Agents store queries and results for perusal by the user. One particular application of this is the processing of periodic queries (profiles).

It is possible to use the Provider Agents to increase the capabilities of the provider systems. For example, it is possible to get some kind of ranking from a Boolean System by issuing several queries. Simply put, when a Provider Agent receives the query $ab$ (linear query formulation) it can issue the queries $a \wedge b$, $a$, and $b$ to produce a ranking (with three ranks).

**Query Processing.** The way of processing a query to the IBS is based on the "trader" concept as explained in the Open Distributed Processing standard (see also [ISO95]). A User Agent corresponds to an *importer* in Open Distributed Processing whereas a Broker corresponds to a *trader*. A query is sent by a User Agent to a Broker which determines which databases to query (and how many documents to request from each, see below). The Broker optionally contacts other Brokers for this, so the Ressource Selection Layer is also implemented in a distributed way. When the Broker is done with this, a list of providers to query and how many documents to request from each is sent back to the originating User Agent which then queries each Provider Agent involved, collects and merges the query results and presents them to the user. When the user requests a document (from a full text database) the User Agent directly contacts the appropriate Provider Agent.

**Provider Selection.** [Fuh96] explains what kind of metadata is to be used and how it is to be used by the Brokers to determine how many documents to request from each database. It is shown how to estimate the number of documents relevant to a query in each database and how to use that estimate to determine what number of documents requested from each database minimizes the costs of processing the query. The theoretical approach presented there is limited to a linear query formulation, however, as well as textual fields only. Therefore, in order to fully utilize the schemata of the databases connected an ad-hoc approach to deal with disjunction and conjunction as well as different data types has been developed. (This ad-hoc approach assumes that all query conditions are stochastically independent. Dealing with stochastic dependence must remain a problem for future work.)

**Data Fusion.** Another interesting problem is the issue of merging the results (ranking lists) returned from each Provider Agent into a homogeneous result presented to the user. This merging will be done in the User Agent. The problem can be divided up into the two subproblems of correctly ordering (sorting) the result list and transforming the schemata. The latter problem will be dealt with on a rather simple basis: the result list presented to the user contains the attribute/value pairs that each Provider Agent returns. The former problem of sorting will be dealt with on the assumption that each Provider Agent produces Retrieval Status Values that reflect the probability of relevance, implying the merging of the ranking lists is trivial.

**Status of our work and outlook.** Currently, a first version of the IBS prototype is being implemented, due to be finished in October 1996. It contains only one Broker. Also, a few databases have been looked at to determine the schemata used by different database providers. All of the databases accessible via the "Services" entry in Ariadne [Ari] use the standard BIBTEX set of attributes; therefore that will be the schema used in the first version of the prototype, though it will be extended with a *keywords* and an *ACM classification* attribute as well as a few attributes needed for bookkeeping (such as document id). The database selection will at first be based on the keywords and ACM classification attributes only.

The MeDoc participants intend to gain experiences with the first version of the IBS prototype when it is finished. That experience will be used to determine what kinds of improvements and changes are necessary for the second, final, prototype version. The final version is due in the second quarter 1997. The participants (in particular the pilot users) are then going to evaluate the final prototype for the project report.

**Acknowledgements.** We thank all of the MeDoc participants for their work on the MeDoc system. Without them, this paper obviously would not have been possible!

# References

[Ari] The ARIADNE Home Page. http://ariadne.inf.fu-berlin.de:8000/.

[Fuh96] N. Fuhr. A Decision-Theoretic Approach to Database Selection in Networked IR. http://ls6-www.uni-dortmund.de/reports/95/Fuhr-96a.html, 1996.

[ISO95] Final Draft—ISO/IECDIS 13235—ODP Trading Function. http://www.dstc.edu.au/AU/research_news/odp/trader/standards.html, 1995.