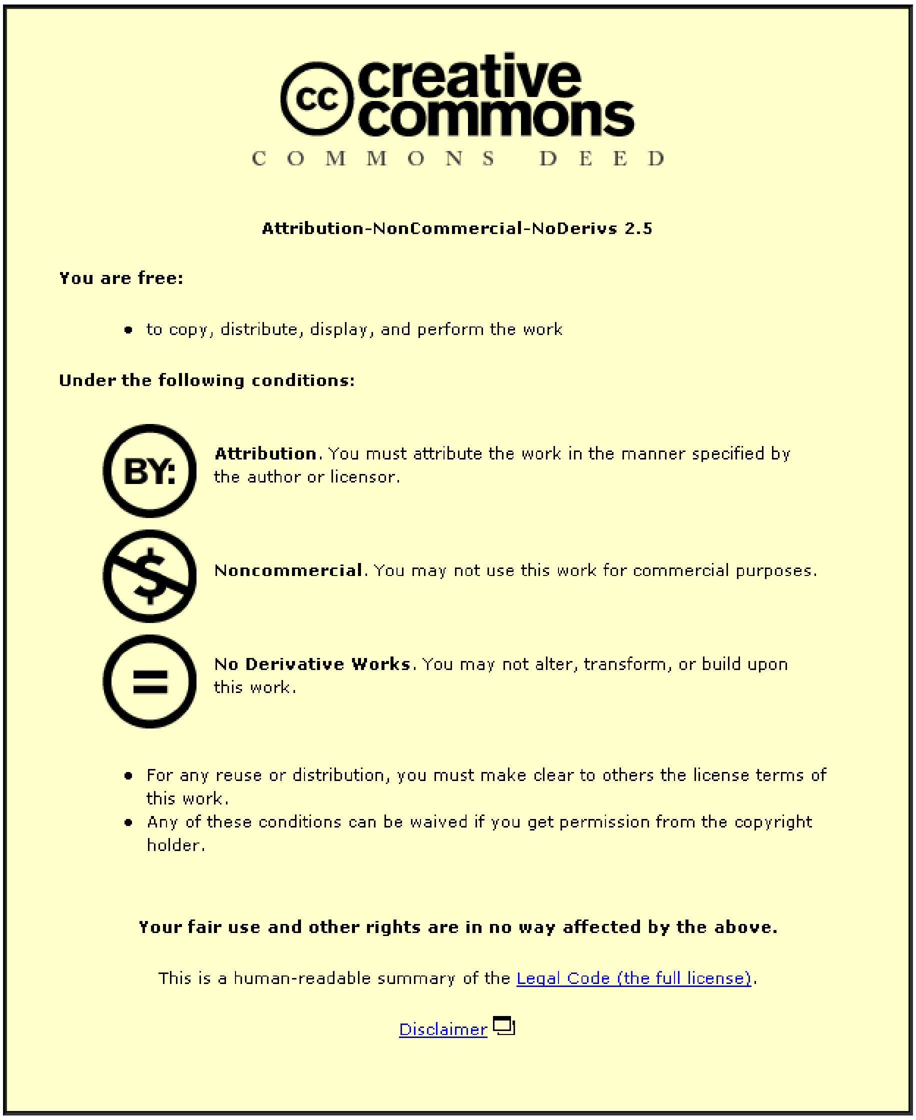

Cross-Organisation Dataspace (COD) - Architecture and Implementation

Paul W.H. Chung, Zhining Liao
Computer Science Department
Loughborough University
Loughborough, UK
p.w.h.chung@lboro.ac.uk, z.liao@lboro.ac.uk

Abstract

With the rapid development of information and communication technologies, the need to share information to improve efficiency in large enterprises is also increasing rapidly. For a large enterprise the information can come from many different sources and in different formats. There is a real requirement to manage the vast amount and diverse sources of data in a convenient and integrated way so that repositories of information can be built up with little additional effort and the information can be easily accessed globally. This paper presents the design and implementation of a prototype, called COD (Cross-Organisation Dataspace), that addresses the above challenges. COD, in the context of an enterprise involving multiple organisations, allows users from different geographical locations to contribute information and to search and access information easily. The information can be contained in many different forms, e.g. text files, reports, drawings and databases.

Keywords- Dataspace; Search engine; Information retrieval; Keyword search.

I. Introduction

With the advancement of the Internet, the development of technologies to search multiple data sources stored in different geographical locations for information has become a major area of work. Queries are posed over multiple information sources that are distributed across a wide area network. Levy [1] identified a number of characteristics that relate to searching for information from multiple information sources over the Internet:

- The information exists in a dynamic environment where it can be easily changed as the information sources may be acting autonomously.
- Information may be stored in different formats.
- Because of the heterogeneity of information sources, some information may be presented in different information sources (i.e., different in structure, query languages and the naming of the data element).
- Data is often required to be transferred over a wide area network.

To deal with the problems of different formats and how the information should be retrieved and displayed, there are mainly two types of search. One is a search for information stored in different file formats. The other is a search for information stored in highly structured databases. Well known examples of the first type are the facilities provided by Google and Yahoo. The other type requires searching distributed databases. Examples are Garlic [2], KOLA [3], and DIOM system [4]. The main disadvantage of structured database searches is that users need to have some knowledge of the information sources in order to retrieve the required information. This paper proposes an architecture that supports both kinds of search in an integrated manner and describes a prototype, called COD, which is based on the proposed architecture. Dataspace is the term that is used to describe a searchable repository that stores both structured and unstructured information [5, 6].

The rest of the paper is organized as follows. Section 2 provides the background of dataspace systems and highlights the problems that need to be addressed to make them practical. Section 3 presents the architecture of COD. Section 4 gives an overview of the prototype design and describes the function of each of the sub-module in COD. Summary and conclusions are given in section 5.

II. Background

Distributed information management systems can be classified into three types: distributed databases, data integration systems and dataspace systems [5].

A distributed database system stores data in different locations on a computer network and data allocation is managed by a database administrator (for example, R* [7], SDD-1[8], INGRES[9]). A distinct feature of distributed database system is that there is a dedicated central site that keeps the schema of the databases and the location of the information [1].

Data integration is the process of combining data residing at different sources and providing the user with a unified view of these data. Data integration systems

978-0-7695-3336-0/08 $25.00 © 2008 IEEE
DOI 10.1109/CSSE.2008.1638

require semantic integration before any services can be provided. Although the schemas are not unified, the terms used in the system and the relationships between them are specified. Therefore, significant upfront effort is required to set up a data integration system.

The goal of the development of a "dataspace" system is to provide the basic functionalities for accessing distributed data sources regardless of the achieved level of integration [5]. For example, a dataspace system can provide keyword search over all of its data sources, similar to what is provided by existing desktop search systems [10, 11]. Other features such as relational-style queries and data mining can be added incrementally with additional effort. Some distinguishing features of a dataspace system are the abilities to:

- handle data in different formats. For example, structured information (database), semi-structure information (HTML/XML) and unstructured information (text file).
- access and update information through different appropriate interfaces;
- integrate the information in the dataspace as necessary;
- rank the relevance of the information returned in response to a query. In this respect a dataspace is more like an information retrieval system than a data retrieval system.

In a dataspace system, there are two main problems to be solved. The first is how to dynamically add and delete information resources and how to collect metadata of new information resources. The second is how to make it easy for users to retrieve relevant and accurate information. To deal with these challenges, a prototype, called COD (Cross-Organisation Dataspace), is proposed in the context of an enterprise involving multiple organizations.

III. Architecture of COD (Cross-Organisation Dataspace)

COD (Cross-Organisationbal Dataspace) is a prototype dataspace system being developed at Loughborough University. This section describes the underlying architecture of COD, which is a general architecture that can be used as a basis for developing dataspace systems for information sharing for large enterprises or collaborating organizations.

This architecture identifies the basic components, defines the purpose and functions of the components, and indicates how these components interact with one another. The main focus of the architecture is to allow information providers and users to share and access information easily through agreed protocols at different layers of the architectural model. The architecture has four tiers (Figure 1):

- Resource layer — the resource layer defines the interface to local resources, which may be shared. The resource layer calls functions to access and control local resources such as databases and other kinds of data. This layer only handles information at the local level.
- Connectivity layer — the connectivity layer defines the communication and authentication protocols required for network-service transactions.
- Collective layer — the collective layer manages the global resources by interacting with the individual resources through the connectivity layer.
- Application layer — the application layer enables the use of resources that are collected through the collective layer in a wide area network environment

Security components work across all four tiers to ensure network security and information access security.

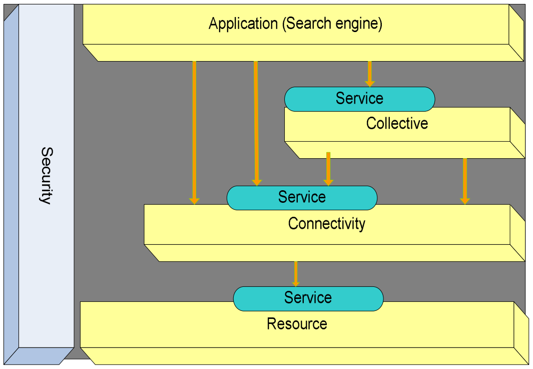

Figure 1. Four-layer Architecture of COD

The application layer has two tiers: mediator tier and wrapper tier (Figure 2). Mediators are software modules that handle application specific service requests. The main task is to utilize the metadata obtained from information sources for efficient processing of user queries. Wrappers are software modules that build around the information sources in order to make them accessible by the mediators. Wrappers provide the following services:

- Translate a user query into different query language expressions that are appropriate for the different information sources. The information sources may have different formats due to the way they are organized and stored. They may be structured information (database), semi-structured information (HTML/XML) and unstructured information (Text file). Each information source is autonomous.
- Submit the translated queries to the target information sources.
- Package the results of the queries.

In the mediator tier, the *Repository* stores the metadata of information sources, i.e. table names, field names, document names, and file directory etc. This information is extracted automatically and helps in global query optimization. The *Capability collection and description* module collects and manages metadata of each information source and monitors any changes. The *Query mediation manager* provides query processing services including source selection, query decomposition, query generation, and result gathering. The *Wrapper manager*, as mention above, provides services including submitting queries and packaging results from different sources.

The function for collecting and monitoring information consists of the following steps: information source registration, metadata collection of the registered information sources and monitoring of any changes to the information source. The function for information retrieval consists of the following steps: query parsing, source selection, query translation, query execution, and packaging of the results. This function makes use of both the mediator and wrapper tiers.

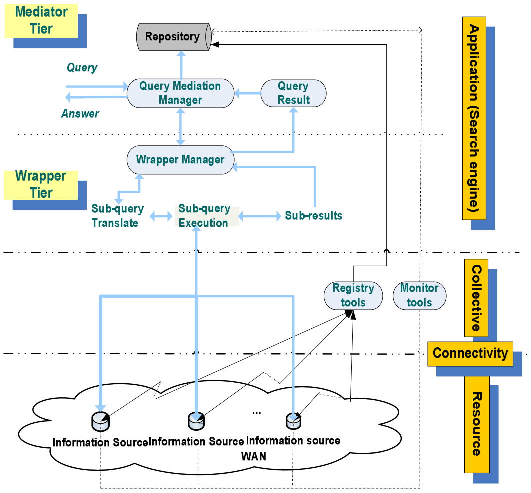

Figure 2. Expanded View of the Architecture of COD

IV. COD Functions

A. Collect and monitor information

In an enterprise-wide application, there are lots of information sources that can contribute to a dataspace system and it is important that users can contribute information to the dataspace easily. Within such a dynamic environment, the system should also be able to monitor changes made to the information sources. COD has four components that provide the intended functions: *Registry Interface*, *Collectivity Tool, Repository* and *MonitorTtool.*

- ○ The Registry Interface allows a user to register information sources to COD.
- ○ The Collectivity Tool is to access individual information source and collect metadata of the source.
- ○ The Repository is to store the schemata and the mappings currently known to the system, and provides a query interface over this collection for use by the other components of the system.
- ○ The Monitor Tool is to monitor individual information source. For example, if the metadata of an information source is changed then the Collectivity Tool will be called to update the metadata information.

In COD, there are two ways to register information sources: email-based and web-based. In the first way a user can just send an email to the email server which contains information such as host name, user name, password and a description of an information source. The data mining tool will analyze the email and extract the information for assessing the information source. The other way is to register an information source is through filing in a form on the web site. Authorised users can register new information sources and delete existing ones.

B. Information retrieval processing

In COD, there are three methods for accessing information. The first is for retrieving structured information. The second is for retrieving unstructured information. COD also provides an integrated keyword-based search tool where a user can specify a string of keywords and information is retrieved from both the structured and unstructured information sources, without any knowledge of the databases and database query languages such as SQL. The details are discussed in the following sub-sections.

1) Structured information retrieval

For users who have knowledge about information sources, such as the names of the information sources, data tables and attributes, an interface is provided for them to choose what they want to search and specify the conditions for the query. For this method, the query forming procedure is similar to most data integration systems. A query to COD is specified in some subset of SQL or XQuery, and is formulated in terms of the mediated schema. When a query is posed, it is reformulated to access the data sources. The reformulation process uses the semantic mappings to determine which data sources are relevant to the query, then formulates appropriate queries for the individual data sources. It also specify how the results from the different sources are to be combined (e.g., via join or union) to produce the final answer to the query.

The other way is for users who have no or little knowledge about the information sources. The vast majority of queries on the web are keyword queries; thus, COD also supports this way of accessing information. COD takes a keyword query and reformulates it as structured queries that are appropriate for the different databases. The query system works in the following way:

1. Classify keywords: The keywords in the query are divided into those corresponding to data values, attribute names or table names. Of course, the keywords may be data values, attribute names or table names in different sources. To make this classification, COD looks up the keywords in two indexes provided by the Repository: the structure

index, that indexes schema elements of data sources, and the value index, that indexes data values.

2. Generate structured queries: structured queries are generated for each relevant information sources and then submitted.
3. Results integration: the results returned from the different databases from different sources are combined into a single result page. The relevant rows from the databases are shown together with information indicating which data source they originated from.

2) Unstructured information retrieval

In an enterprise-wide application, some level of search capability to allow users to find what they need on the intranet quickly and easily is required. To support this, COD integrates the Google desktop search engine for searching unstructured information. The procedure for searching unstructured information by using Google desktop is explained below:

1. The user submits a query via a client browser
2. COD translates the query then sends the translated query to Google desktop.
3. Google Desktop retrieves the information from the local source that meets the search requirements.
4. Google desktop returns the search results to COD.
5. COD screens the search results according to the user access permission.
6. After a series of processing of the screened results (for example, format adjustment, file name transformation, hyperlinks re-positioning, etc.) the search results are returned to the client browser.
7. The user can then select any links on the returned page to access the documents they are interested in.

3) Integrated keyword-based search tool

COD also provides an integrated keyword-based search tool for structured and unstructured information, which has a single user interface for inputting keywords for submitting a query. The search procedure of the integrated tool is:

1. Parse keywords: the inputted keywords are analysed and some words or symbols, for example, “is”, “of” are removed.
2. Call the keyword-based search tools for structured information and unstructured information giving the extracted keywords.
3. Integrate the search results from different information sources from both tools.
4. Present the search results to the user.

V. Conclusion

Being able to share information is a crucial function of a network enabled environment. To enable information sharing, an information system must allow users from different geographical locations to *contribute* information and to *access* information easily. To deal with these challenges, we have presented the architecture, and the functions of a prototype implementation of COD in this paper. COD provides a higher level of query capability compared to web search engine and traditional data integration system. Future work will focus on semantics-enhanced keywords based search.

Acknowledgments

The work reported in this paper has been supported by the NECTISE project jointly funded by BAE Systems and the UK Engineering and Physical Sciences Research Council Grant EP/D505461/1. The authors would also like to thank Dr David Gunton for helpful comments.

REFERENCES

[1] A.Y. Levy, Combining Artificial Intelligence and Databases for Data Integration, Artificial Intelligence Today: Recent Trends and Developments, pp 249-272, 1999.

[2] R.M. Tork, F. Ozcan and L.M.Haas, Cost Model Do Matter: Providing Cost Information for Diverse Data Sources in a Federated System, Proc. of the 25th VLDB conference, pp 599-610, 1999.

[3] M. Cherniack and S.B. Zdonik., Changing the Rules: Transformations for Rule-Based Optimizers, Proc. Of the ACM SIMMOD International Conference on Management of Data, pp 61-72, June 1998.

[4] L. Ling and C. Pu, Distributed Query Scheduling Service: an Architecture and its Implementation, International Journal of Cooperative Information Systems, Vol. 7 Nos. 2&3, pp 123-166, 1998.

[5] M. Franklin, A. Halevy and D. Maier, From databases to dataspaces: a new abstraction for information management, ACM SIGMOD Record, Vol 34(4), pp 27 – 33, 2005.

[6] Halevy, A., Franklin, M., and Maier, D. 2006. Principles of dataspace systems. In *Proceedings of the Twenty-Fifth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (Chicago, IL, USA, June 26 - 28, 2006). PODS '06. ACM, New York, NY, pp 1-9.

[7]B.G. Lindsay, L. Hass, C. Mohan, P. Wilms, and R. Yost. Computation and communication in R: A distributed database Manager. ACM Trans. Computer systems, Vol 2(1), pp 24 - 38 , 1984.

[8] P.A. Bernstein, N. Goodman, E. Wong, C. L. Reeve, and J.J. B. Rothnie. Query processing in a system for distributed database (sdd-1). ACM Trans on Database systems, Vol 6(4): pp 602-626, 1981.

[9] M. Stonebraker. The design and implementation of distributed Ingress. The INGRES papers, Addison-Wesley, MA, 1988.

[10] Vu, Q., Ooi, B., Papadias, D., and Tung, A. K. 2008. A graph method for keyword-based selection of the top-K databases. In *Proceedings of the 2008 ACM SIGMOD international Conference on Management of Data* (Vancouver, Canada, June 09 - 12, 2008). SIGMOD '08. ACM, New York, NY, pp 915-926.

[11] Markowetz, A., Yang, Y., and Papadias, D. 2007. Keyword search on relational data streams. In *Proceedings of the 2007 ACM SIGMOD international Conference on Management of Data* (Beijing, China, June 11 - 14, 2007). SIGMOD '07. ACM, New York, NY, pp 605-616.