SEARCHY: A Metasearch Engine for Heterogeneous Sources in Distributed Environments

David F. Barrero
RedIRIS
Tel: + 34 91 212 76 - 20
Ext. 5545
david.barrero@rediris.es

M. Dolores R-Moreno
Dept. de Automática
Univer sidad de Alcalá
Tel: + 34 91 885 6607
mdolores@aut.uah.es

Oscar García
Dept. de Automá tica
Universidad de Alcalá
Tel: + 34 91 885 6601
oscar@aut.uah.es

Angel Moreno
Dept. de Automática
universidad de f Alcalá
Tel: + 34 91 885 6953
angel@aut.uah.es

Abstract:

In this paper we present a federated solution to the problem of resource searching across several organisations. It is a cooperative distributed multiagent system that locates and semantically integrates the access to heterogeneous distributed data sources, using Dublin Core as the metadata model.

Keywords:

Metasearch engine, semantic web, web services, RDF, multiagent systems.

1.Introduction

The adoption of such information technologies has produced a huge increment of information generation in all its forms. This information is stored in documents of many natures, from single text or web pages to multimedia documents. Giving an integrated access to all this information is a mayor problem.

This problem increases the complexity when information systems are distributed in different organisations, each one with its own technological solution and business culture. In this context, using a common technology may not be a valid approach: it is needed a solution that respects technological independency meanwhile grants interoperability.

Some solutions have been developed to solve this problem such as creating management information systems or reusing the existent ones. The disadvantages of these approaches are that they impose working procedures, they are aggressive methods with the existent infrastructure technology or they are too specific to some information models.

In this paper we present a program called ***Se a rch y*** (1), a metasearch engine that solves the problem described above using legacy backend and it is aimed to be used in a very specific context: several organisations with some documental management systems that must interoperate. ***Se a rch y*** is a multiagent (11) metasearch engine that uses Dublin Core as the metadata model to search and describe documents. It is a non intrusive, cooperative, extensible and information model independent solution.

The paper is structured as follows: section 2 describes the motivations of our work. Next, we give an overview of the ***Se a rch y*** metasearch engine. Then, we describe in detail its architecture, the metadata model and the mechanisms used for the integration and recovery of the information. Finally, future work and conclusions are outlined.

2.Motivations

Our main motivations emerged from the scenario analysis described in the last section. That is, searching and location of documents across heterogeneous information systems hosted by different organisations.

In order to obtain successful real world solutions in that context, we need a system that can have a simple deployment and that can avoid redundancies using documental management systems that may be already working organizations. It just has to grant interoperability in documental searches across different organisations. In the context of multiorganisational interoperation, a distributed approach may be more suitable for many reasons:

- It will not need any strong central authority.
- It implies entities that participate in the application deployment from an equality position (collaboration).

https://doi.org/10.23106/dcmi.952108304

- The previous information systems are reused, minimizing redundancies and maximizing efficiency (federation).
- It is loosely coupling with the search engines (non intrusive).

Due to the project nature, the application must be standards based and platform independent. Previous work has been done in information integration. Tthis field has focused in ontology based integration systems (11) and the semantic web technologies. With few exceptions (6), they are centralized approaches, limited to some sort of data sources, typically databases or too application specific. Most of them do not address this problem in the context that we do it (9).

3. *Searchy*: a Metasearch Engine

Our proposal complies to the objectives described in the previous section thanks to a metasearch engine called ***Searchy*** (8). From the user point of view it is just a document search engine, he or she can submit queries referred against some term and the system returns a description of different documents that satisfy the query. But Searchy does more than that.

Searchy is a general purpose search federation facility. It uses existing search engines, integrating and showing them as a whole uniform entity. Thanks to its modular design, it allows an easy extension to new information systems.

The users interact with one single monolithic search engine instead of querying against different distributed search engines as it is nowadays done. All the process will be kept transparent for the users.

Searchy is a multiagent system that once it is deployed, is able to:

- Get abstract queries independent from the calling system.
- Translate and submit the queries to different information systems.
- Extract the metainformation from the responses.
- Map the metainformation to Dublin Core metadata.
- Return the results than can be graphically presented through a user friendly interface or saved them in a file.

Searchy is not aimed to be used by ***end user***s but rather by other applications, i.e., it is just a middleware, an abstraction layer that integrates search systems.

Since ***Searchy*** interface is web services based, it allows using it from simple web applications to heavy ones. ***Searchy*** clients can work as a simple graphical interface that collects the query and visualizes the data, to data sources for other not direct-related applications with the final user.

4. The *Searchy* Architecture

Most of the Searchy features are a consequence of its particular distributed architecture based on the concept of agent, i.e., an autonomous software piece with social skills. A ***Searchy*** agent is composed of three well-defined elements as Figure shows. Their main features are:

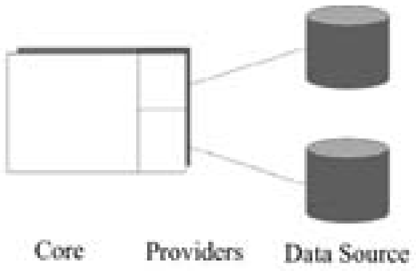

Figure 1: Agent architecture

1 *Core*. It is the common part for all the agents and it is in charge of the message processing, network related tasks, the agent configuration setting, the basic services to the client and the supplier agent, and any task non-related to the data sources. 2 *Provider*. It manages the access to a single data support system and builds the metadata from it. It is the interface between the core agent and the data source. An agent can contain several providers. 3 *Data Source*. It is the information system that stores the document or contains an index of accessible documents. There is an important flexibility about the data sources that may be accessed, although given the approach of our system, usually the data source will be a database or an index.

The agents are not specialized in any task since they carry out the same activities with an exception: each agent can access to different information systems.

A key feature in ***Searchy*** is its collaborative nature; it may interchange queries with other agents or create metadata search networks. A typical scenario where ***Searchy*** can be deployed consists of different agents, each one accessing the information contained in one or several information sources.

The criteria of “which” agent can do “what” is of very different kind, and non-necessarily technical. Usually, the responsibility assignment follows an administrative criterion that tries to guaranty the entity

https://doi.org/10.23106/dcmi.952108304

and integrity of a specific unit, department, branch office, etc in a company.

Each data source requires a specific provider what makes it extensible to any new data sources. This implementation can be done relatively easily. Actually, ***Searchy*** covers four different providers: SQL, LDAP, Google and Harvest.

5. The Metainformation Model

We need to recover any type of information under a heterogeneous framework, and simplicity is a desirable objective. Given the generality of uses that it can have, the information format should be flexible, self-content and platform independent: this kind of technology that we need for service federation and information retrieval can be found in the Semantic Web.

The semantic web is built on the Resource Description Framework, RDF. It defines a grammar to express triples of the form <Resource, Property, Value> to characterize resources and relationships between them.

The semantics are set throw ontologies (4), which defines formally sets of terms, with well-defined semantics and the relationships between them. The more recent approach to ontologies definition languages is the OWL (Ontology Web Language), a W3C's recommendation based on RDF.

The DCMI defined the Dublin Core Metadata Element Set (DCMES), it is a metadata model created from an interdisciplinary point of view suitable to describe a wide range of resources.

Searchy has been designed to locate several different types of documents stored in arbitrary backend, thus it needs a general metadata model that abstracts local information formats, representing some properties about the documents it locates. The system should be a general propose document metasearch engine, so, the range of targeted documents has to be wide.

The metadata model must be flexible enough to be able to describe documents of different natures and supports. DCMES fulfils the requirement described above. All data in ***Searchy*** is based on it, queries are expressed using Dublin Core and the response with the document description also uses Dublin Core.

6. How is the Information Retrieved in *Searchy*?

The mapping of the metainformation available in the data source to Dublin Core is a main aspect of ***Searchy***. This is a well-known problem (ontology mappings) in the information integration field and it is a focus of research.

To solve the problem of metainformation mapping in ***Searchy***, we have adopted a conservative approach: offering an interface to the agent administrator to define manually the mapping between ontologies. In practical terms, extracting metainformation from the information provider is, by far, the most difficult task for the system administrator.

For this goal, an easy and simple string substitution mechanism has been developed. Part of the complexity remains hidden with this procedure and the mapping rules can be establish with less effort. Each metainformation field may be composed by none, one or more information fields.

Getting metainformation from the information stored in the data support, is a task done by the provider and it has a strong dependency on the data properties of the support system. There are some support systems, for example some text formats, that have some metainformation integrated in the document, and ***Searchy*** is able to use. But the general case is when we have to obtain the metainformation from the stored information, directly mapping the information into the metainformation.

The agents have been designed to be highly extensible, therefore, adding new information supports may be quite easy, and the flexibility of the system facilitates the implementation of a wide variety of Providers but there are very few limitations. If data can be read, Searchy can support it.

7.A Simple Example

To clarify how Searchy can be integrated in any organization or company, we take as example our university and in order to reduce the example, we will consider our Computer Sciences School. Figure 3 shows a typical structure of our University Faculty.

The objective is to provide an integrated view of all those resources and facilitate the location of the documentation. In this scenario imposing any intrusive technology is not possible because each department has its own idiosyncrasy and particularities.

Searchy provides a satisfactory solution, each department only has to set up a ***Searchy*** agent and establish a criterion to map their databases structure, directory schema and Google metainformation into Dublin Core. Once it has been done their search facilities may be integrated within the rest of Departments of the School.

The departments are composed by different areas, like the Automatic Department. This department can delegate the integration in any of the areas (in the example, Language and OOSS). The agent in the department will provide an interface to the two agents of the two areas.

https://doi.org/10.23106/dcmi.952108304

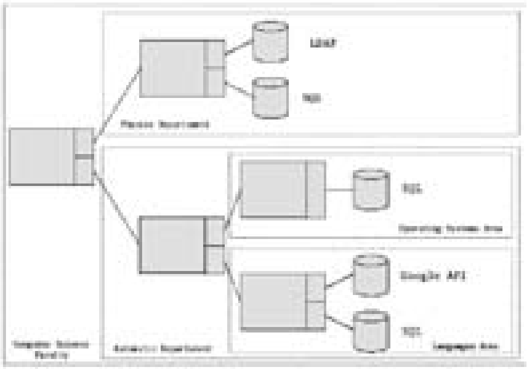

Figure 2: The Computer Science School layout in the *Universidad de Alcalá*

8.Future work

The distributed static nature of ***Searchy*** has some intrinsic disadvantages. The main one is the strong dependence of time response in relation with the number of agents. There are two ways to overcome this situation: dynamic agents discovery and alternative transport mechanisms.

An increment in the number of agents in a ***Searchy*** network, may increase the number of found documents at a point that the query result might get not useful because of it size. More intelligent search mechanisms are needed to avoid this situation, like introducing machine learning techniques in the agents to adapt the query to the user preferences.

Experience shows a widely demanded feature: to control who can access to what document. Thus, ***Searchy*** is going to use SAML (Security Assertion Markup Language) to grant autentification as well as rights management.

A key point to the success of ***Searchy*** is its ability to manage any sort of data source by developing new providers. In this way, providers for documental management systems, like Google Desktop and Beagle support will be developed.

9.Conclusions

Searchy is a scalable, modular and highly distributed metasearch system that provides document searching over different information systems as well as a framework for distributed information retrieval.

This system is especially suitable for environments where several entities must interoperate with different search systems. Its strongest points are: the generality of data sources that it can integrate, and the limited coupling with the information systems that addresses. The cost of implementing a ***Searchy*** network is quite reduced: there is no need to modify any information infrastructure, it is quite simple to manage and is freely distributed with the GPL license.

Acknowledgments

We would like to thank specially Diego R. López, José Manuel Macías and Javier Masa Marín, researchers of RedIRIS, for their support all along the development of ***Searchy***.

This work has been funded by the Universidad de Alcalá project UAH PI2005/084 and the PTYOC program of RedIRIS.

References

1. D. F. Barrero, D. R. López and O. García. Distributed metainformation searching: an approach to information retrieval in the age of the semantic web. In *VII$^{t h}$ TERENA Net wor king Conference*. Rhodes, Greece. June 2004.

2. T. Berneers-Lee, J. Hendler and O. Lassila. The semantic web. *Scientific A m erica n* , 2001, vol. 5, n. 285, pp. 28-31.

3. DCMI Usage Board. DCMI Metadata Terms.

4. A. Gómez Pérez, M. Fernández-López and O. Corcho. Ontological Engineering. Springer-Verlag, 2003.

5. Searchy Project Web Site. http://jsearchy.sourceforge.net

6. M. Michalowski, J.L. Ambite, S. Thakkar, R. Tuchinda, C.A. Knoblock and S. Minton. Retrieving and Semantically Integrating Heterogeneous Data from the Web. In IEEE Intelligent Systems, Vol. 19, No. 3, pp. 72-79, May-June, 2004.

7. H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hübner. Ontology-Based Integration of Information: A survey of Existing Approaches. In *Proceedings of Intern ational Joint Conference on A rtificial Intelligence (IJCAI-0 1),* pp. 108-117, Seattle, 2001.

https://doi.org/10.23106/dcmi.952108304