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.
Universidad de Alcalá
Tel: +34 91 885 6607
mdolores@aut.uah.es

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

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

Abstract:

Intensive use of information technologies has produced the need of managing important amounts of information in digital format. Depending on particular circumstances, the information may be stored in single documents, data bases, web pages or in document management systems what may lead a potential source of interoperatibility problems.

In this paper we present a federated solution to this problem. It is a cooperative distributed multiagent system that locates and 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, metadata model, Dublin Core.

1. Introduction

In the early seventies when the preliminary connections of the ARPAnet were established, anyone possibly could not imagine the deep impact of these ideas in our actual information society.

The evolution of these computer networks, especially once the Internet has been fully deployed, represents the most successful example, in the brief history of computer sciences, about the benefits that can be obtained when research, development and investment are applied in the society.

Thanks to Internet, new information technologies have appeared what have revolutionized the way that traditional (human) concepts such as business, commerce, or education were developed.

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.

On the one hand, those documents are managed by different information systems, depending on many factors. They may be document management systems, web servers or database managers; each one is a suitable solution for some specific context. There are some factors that complicates the information management, one of them is that the information is non well structure-stored as in the case of the Web, with very low metainformation associated and with non very well defined semantic. Quite often these systems have to work together and then, interoperatibility becomes a main 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 interoperatibility.

Some solutions have been developed to solve this problem such as creating management information systems or reusing of 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.

The Dublin Core Metadata Initiative (DCMI) was born for the purpose of developing interoperable metadata and it has in the Web a natural place. The idea is to maintain a set of metadata terms that can be used to describe a wide spectrum of documents. Such metadata standards are necessary in order to ensure that

different kinds of descriptive metadata are able to interoperate with each other and with metadata from other systems (bibliographic and non-bibliographic).

Then, for applications on the Web, we need a semantic definition language not only to define concepts but also to express them. Constraints and rules is a plus. Different language belongs to what is called the Semantic Web (2): RDF[1], RDF Schema[2] or the more recent approach OWL[3].

In this paper we present a program called ***Searchy*** (1), a metasearch engine that solves the problem described above using previous information systems and aimed to be used in a very specific context that involves several organisations with some documental management system working. ***Searchy*** 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 our ***Searchy*** 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 onganisatoin. It just has to grant interoperatibility 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 a equality position (collaboration).
- The previous information systems are reused, minimizing redundancies and maximizing efficiency (federation).
- It is loosely coupling with the search engines (non intrusive).

The main objective described above involves two global tasks: integration of information and integration of processes:

1. Integration of the information requires a global model of information, i.e., the definition of a metadata set, able to describe the wide range of potential documents that we can address with such a search system. All the local metadata have to be mapped into this shared general model..
2. Integration of processes involves the definition of new access interfaces to the search system . Any search system must publish its services in some form; they used to be specific to the search system. The search engines federation has to provide an integrated interface to the service.

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

3. *Searchy*: a Metasearch Engine

Our proposal complies the objectives described in the previous section thanks to a metasearch engine called ***Searchy*** (8). From the user's 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

[1]http://www.w3.org/RDF/
[2]http://www.w3.org/TR/2000/CR-rdf-schema-20000327/
[3]http://www.w3.org/2004/OWL/

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 users but rather by other applications, i.e., it is just a middleware, an abstraction layer that integrates search systems.

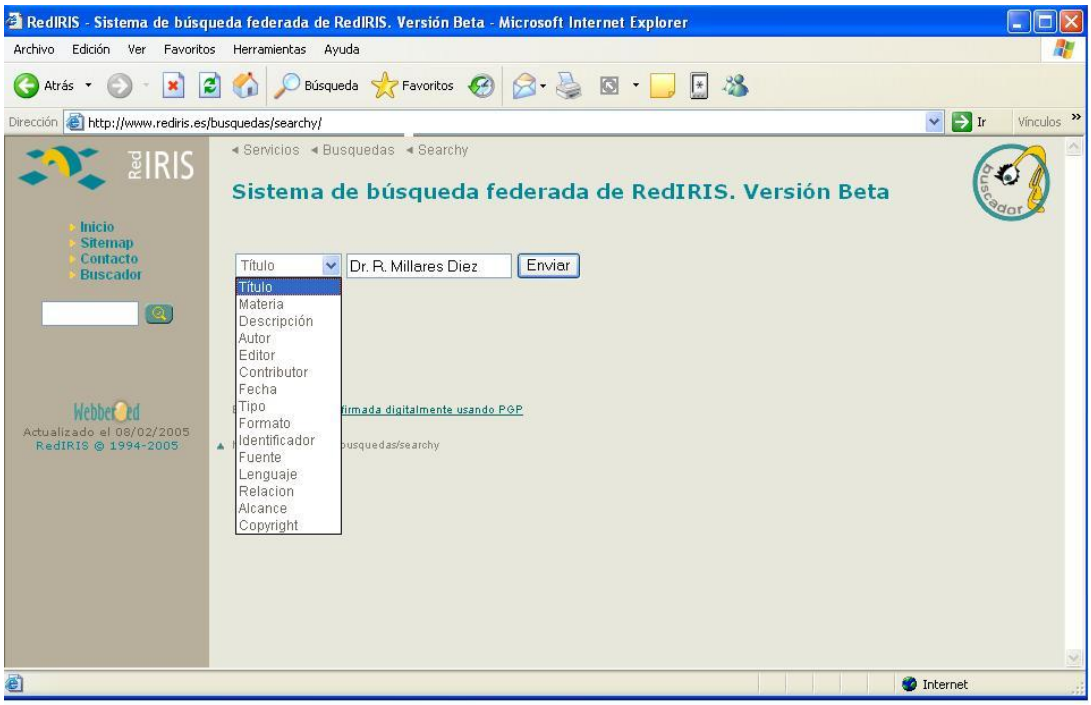

Figure 1 *Searchy* graphical interface

Figure 1shows the graphical interface where ***Searchy*** is actually used. This interface is actually written in Spanish (and very soon will be translated into English) and all the information fields to perform the search (i.e. contributor, identifier, title or date, among others) are DCMES terms.

Since ***Searchy*** interface is web services based, it allows using it from simple web applications to heavy ones. The ***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.,. the minimal functional unit with complete autonomy in Searchy.

A ***Searchy*** agent is composed of three well-defined elements as Figure 2 shows. Their main

features are:

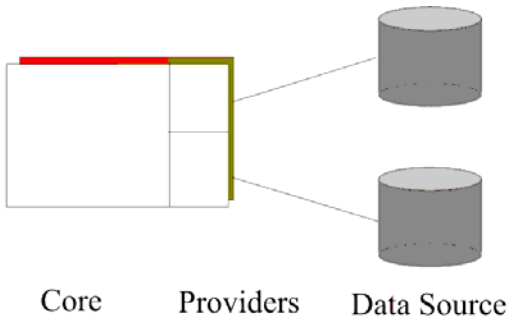

Figure **2**: Agent architecture

1 The 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 The 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 may contain several providers.
3 The 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 and integrity of a specific unit, department, branch office, etc in a company.

The underlying philosophy of the ***Searchy*** architecture makes it very flexible and it can be used within a wide range of situations and aims. The concept of provider and its implementation allows easily adding new information supports.; the possibility of communication between different agents is the key point to distribute the application, and it is a powerful feature for the administrator. Like that, we can easily comply with organization policies designing the appropriate agents' deployment.

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. Thanks to SQL and LDAP providers we can access to structured data in relational database and directories that are part of a system information backend. The Google provider uses the SOAP API to allow interaction with its search engine for searching in Internet or a web site. Finally, the Harvest provider allows locating resources (i.e., Web pages or LATEX documents) from any intranet.

When a query arrives into an agent, it is processed by each one of its providers. It is translated into the query format managed by the data source. If it is an SQL database, the query is translated into SQL, or in the case the data source is another Searchy agent, query is just forwarded. The effect is a query propagation mechanism.

The results follow the inverse path. The providers translate results from the local ontology used by the data source into the shared ontology (this process is described in section 6).

5. Metainformation model

We need to recover any type of information under an 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 eXtended Markup Language, XML, provides a universal storage and interchange format for such Web-distributed knowledge representation and it has been accepted as the emerging standard for data interchange on the Web. XML allows authors to create their own

markup (e.g. <NAME>), which seems to carry some semantics. However, from a computational perspective tags like <NAME> carry as much semantics as a tag like <N1>.

But applications on the Web require much richer data and the XML data model is too low-level and hieratical. A new proposal is the Resource Description Framework, RDF, and the RDF Schema. The first one uses triples of the form <Resource, Property, Value> to characterize resources and relationships between them. The second one defines classes and properties so it can be considered as an ontology language. RDF makes possible to use multiple different pieces of software to process the same metadata, and to use a single piece of software to process (at least in part) many different vocabularies.

RDF is an abstract model and by it self does not have syntax neither semantics. There are defined different syntaxes for RDF, the most used one is XML, but there are others like N3 or N-Triples; they are needed to transfer RDF model and store 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.

The DCMI defined the Dublin Core Metadata Element Set (DCMES), it is a metadata model created from a 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 medata model must be flexible enough to be able to describe documents of different natures and supports. DCMES fulfills the requirement described. 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 is a focus of research. When there is no metainformation in the data source, the information is usually obtained from any field or property of the resource.

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.

Access to the specific information source interface must be, clearly, hard coded in the Provider. Then, a new provider must be developed in order to access a different data support.

One of ***Searchy*** main design objectives was to simplify the development of the Providers. For this goal the system provides a set of facilities and services in order to the developer can focus in his/her own business.

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. At the moment four types of information Providers have been successfully implemented: SQL, LDAP, Harvest and Google. But one of the strongest features of ***Searchy*** is

its ability to deal with any kind of information format. If the information can be read, it can be supported by ***Searchy***.

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 an University Faculty.

The School is composed by a few Departments, each one with its own technological infrastructure. Some departments have research databases, others have directory services with information about its staff and some have web sites containing web documents about academic and research subjects in many formats: web pages, PDF files, etc.

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 schemas 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.

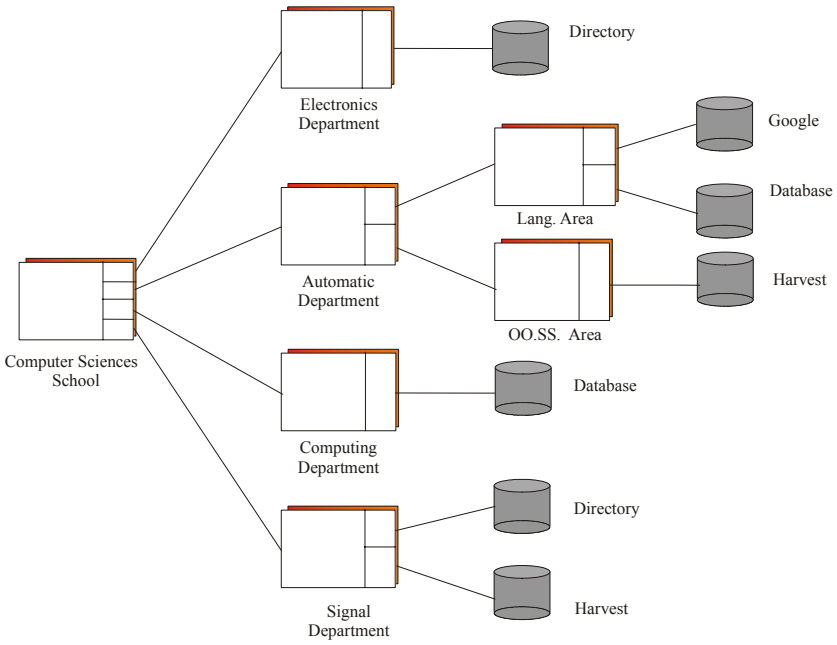

Figure 3: The Computer Sciences School layout in the Universidad de Alcalá

8. Future work

The distributed nature of ***Searchy*** has some intrinsic disadvantages. The main one is the strong dependence of time response in relation with the number of agents. Each agent queries concurrently to all its providers, but query propagation across the agents is serial. This type of propagation has a strong impact on the performance and is a critical point for scalability.

There are two ways to overcome this situation: dynamic agents discovery and alternative transport mechanisms.

If an agent could discover dynamically all the agents that compose the network, it would be able to invoke them concurrently, and performance will be increased. Service discovery is a field that has been focus of high attention in the web services community and

there are many solutions available, may UDDI[4] (Universal Description, Discovery, and Integration) be the best known.

Usually, web services are used with one-to-one protocols, like HTTP, but there are better alternatives for this context where many entities are implied in the communication. A paradigm with remarkable success is the given by peer-to-peer networks, then JXTA[5] may be a good solution. A different approach quite extended in multimedia applications is the given by IP multicast; it is used for one-to-much communications and would be added to ***Searchy*** in a near future.

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.

The 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 (8) with the GPL[6] license.

Acknowledgments

We would like to thank specially Diego R. López, José Manuel Macías and Javier Masa Marín, researchers of RedIRIS, 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[7] 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 *VIIth TERENA Networking Conference*. Rhodes, Greece. June 2004.
2. T. Berneers-Lee, J. Hendler and O. Lassila. The semantic web. *Scientific American*, 2001, vol. 5, n. 285, pp. 28-31.
3. DCMI Usage Board. DCMI Metadata Terms. http://dublincore.org/documents/dcmi-terms
4. A. Gómez Pérez, M. Fernández-López and O. Corcho. Ontological Engineering. Springer-Verlag, 2003.
5. James Hendler. Agents and the semantic web. *IEEE Intelligent Systems Journal*, 16(2):30-37, March-April 2001.
6. L. Kerschberg, M. Chowdhury, A. Damiano, S. Jeong, S. Mitchell, J. Si and S. Smith. Knowledge *Sifter*: Ontology-Driven Search over Heterogeneous Databases. In the 16th International Conference on Scientific and Statistical Management (*SSBDM 04)*. Place: Santoriny Island. IEEE Computer Society. 2004.
7. S. Kokkelink and R. Schwänzl. Expressing Qualified Dublin Core in RDF/XML. http://dublincore.org/documents/dcq-rdf-xml/.
8. Searchy Project Web Site. http://jsearchy.sourceforge.net
9. T. Son S. McIlraith and H. Zeng. Semantic web services. *IEEE Intelligent Systems*, 16(2):46-53, March/April, 2001.
10. M. Michalowski, J.L. Ambite, S. Thakkar, R. Tuchinda, C.A. Knoblock and S. Minton.

[4] http://www.uddi.org
[5] http://www.jxta.org
[6] http://www.gnu.org/licenses/gpl.html
[7] http://www.rediris.es/app/ptyoc

Retrieving and Semantically Integrating Heterogeneous Data from the Web. In IEEE Intelligent Systems, Vol. 19, No. 3, pp. 72-79, May-June, 2004.

11. 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 International Joint Conference on Artificial Intelligence (IJCAI-01),* pp. 108-117, Seattle, 2001.
12. F. Wan and M. P. Singh. Commitments and Causality for Multiagent Design. In *2nd International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, Melbourne, ACM Press, July 2003.