International Journal of Computer Science and Applications

Vol. 7 No. 2, pp. 14 - 29, 2010

INFORMATION INTEGRATION IN SEARCHY: AN ONTOLOGY AND WEB SERVICES BASED APPROACH

DAVID F. BARRERO, MARÍA D. R-MORENO
Computer Engineering Department, University of Alcalá
Edificio Politécnico
Crta. Madrid-Barcelona Km. 33,6
Alcalá de Henares, Madrid 28871,Spain
(david|mdolores)@aut.uah.es
http://atc1.aut.uah.es/(david|mdolores)

DIEGO R. LOPEZ
RedIRIS
Edificio CICA
Avenida Reina Mercedes s/n
Seville, Spain
diego.lopez@rediris.es

Organisations need from heterogeneous information systems to deal with the complexity of information management. The result is the creation of isolated views of the information and therefore an inherent difficulty to obtain a global view. Being able to provide such an unified and global view of the -likely heterogeneus- information available in an organisation is a goal that provides added-value to the information systems. To address this problem several Enterprise Information Integration (EII) solutions have been proposed. In this paper, we present an EII solution named Searchy. Searchy is a agent-based platform which, through the utilization of heterogeneous semantic wrappers, integrates information from arbitrary sources and translates them into semantic terms. Its distributed nature and non intrusive operation enables it to operate in a B2B environments where several entities share their information systems.

Keywords: Information Integration; metasearch; MAS.

1. Introduction

There are many factors that can explain the success of web technologies. One of them is the fact that the Web has been able to integrate the information and services that were available on the Internet. Several classical services on Internet like Gopher, Archie, FTP and even the email have found on the Web a way to unify their services or to give a coherent integrated user interface. The users only have to deal with a tool, the browser, that is easy to use and provides uniform access to information as well as services. The way in which the Web eased the use of Internet contributed to its socialization in the mid-nineties.

The success that the Web has achieved integrating information is remarkable. However, from the user point of view, this success is not enough to deal with the rapid-growing of the volume of information and services contained on the Web. Useful exploitation of information requires several steps: locating, accessing, filtering and compositing information. Similar steps must be done to access to the services. All those tasks must be typically performed manually and are usually quite tedious for the user. Then, automating this process can save many time, but the actual Web architecture makes it a difficult task.

The success of the Web as an integrator of information and services for human users contrasts with the difficulty of integrating information and services from an application point of view. Overcoming this problem is one of the major challenges that the Web has to face. Several data and web mining techniques [Frakes (1992)] have been developed with more or less success, but the results are far away from the actual needs. The Web is evolving from being the web of users to the web of applications. A new generation of web technologies such as Web Services and the Semantic Web are being developed to overcome those problems, giving also new possibilities for the enterprise requirements.

Any modern enterprise has many of its processes automated. Usually enterprises depends on legacy systems, specific programs that were not usually designed with interoperability as an objective. Thus, it originates the creation of information islands within the enterprise. The concept of EAI (Enterprise Application Integration) [Linthicum (1999)] has grown in this context with the aim of avoiding the isolation of systems. In some circumstances we may be more interested in integrating the information than integrating logics, arising new specific problems. This has originated a new paradigm derived from EAI, named EII (Enterprise Information Integration) [Halevy (2005)]. EII deals with on-demand integration of information when it is stored in heterogeneous information sources. The complexity of information integration within an organisation can be increased when the integration is extended to several organisations.

When a bunch of organizations are involved in an integration process, the problems associated to the integration are increased. Integration problems, such as interoperability, are amplified, and new problems arises. One of the most interesting problems in such context is the need to respect the information systems available in each partner of the integration process. Being able to develop a solution that does not require to modify legacy systems in such context is a plus. Within this environment we have developed an EII solution called Searchy.

Searchy is a joint project between RedIRIS -the Spanish Educational and Research Network- and the Universidad de Alcalá (UAH). It is a distributed platform that extracts information from heterogeneous sources, translates them into semantic terms using different ontologies, and integrates them on demand. It is a wrapper container that eases wrapper development by providing a platform with several services that the wrapper developer can use. It is based on Web Standards like SOAP

[Mitra (2003)], RDF (Resource Description Framework) [Manola (2004)] and OWL (Web Ontology Language) [W3C (2004)]. Thus, it can be easily integrated in other platforms and systems based on Semantic Web and in a SOA (Service Oriented Architecture) [Endrei (2004)]. We have focused on developing a simple design of Searchy, so its deployment is easy and can be performed by non specialized staff.

This article is structured as follows. First, we describe the related work and technologies our work is based on. Section 3 presents Searchy from a general view, and section 4 is focused on the description of the system architecture. The description of the application finishes in section 5 with a discussion of the information retrieval and information integration mechanism used in Searchy. A real deployment example of our solution is described in section 6. Then, some future steps are outlined and finally conclusions are presented.

2. Related Work

Initially the technologies behind the Web were (human) user oriented. This approach for human-machine iteration on the Web was satisfactory at least in the beginning. Although the exponential expansion of the Web in the mid-nighties (see: news.netcraft.com/archives/2006/10/06/ october_2006_web_server_survey.html, for a further description) has proofed its success, new challenges emerge with the increasing volume of information and services published on the Web. From another point of view, the Web is used not only to access information, but also to access services. Despite the Web has been able to adapt to those changes, it is obvious that the Web is information oriented in an human-machine communication. An extension of the Web architecture includes two new technologies to face the previous problems: the Semantic Web (SW) and Web Services (WS).

The tendency in information integration is to add intelligence, moving from the concept of information integration (located in single documents) towards knowledge integration (located across several documents) [Zaine (1998)]. The use of ontologies [Grubber (1993)] has had a big impact in information integration. With the development of the Semantic Web [Hendler (2001)], the integration of information has found a new framework by means of standards. The impact that the Semantic Web may have in the information retrieval field might be complemented by the Web Services.

2.1. *Semantic Web and Web Services Technologies*

An example of how the actual Web may be improved can be found in search engines. The most popular search engines are based on indexers that retrieve a set of parameters from documents on the Web and stores them in an index. Information kept by a typical index may contain a list of words with the number of hits in each document [Frakes (1992)]. The search engine uses this index to relate the word that the user is interested in, and the document in which it appears. Thus, a sequence of letters with no meaning are used as the basis of the search, i.e., it is a lexical search.

One word with different meaning gives the same results, no matter what the user was looking for.

The experience of users with search engines should be more satisfactory. This is the goal of the Semantic Web: transform the Web of information into a Web of knowledge [Hendler (2001)]. Human users of the SW will not manually search, select and aggregate information on the Web using a web browser. Instead of that, search, selection and aggregation of information will be done automatically by personal agents [Hendler (2001)] to fullfill some task given by the user.

All this transformation is supposed to be based on three technologies supported by the W3C: RDF (Resource Description Framework) [Manola (2004)], RDF schema (RDFS) [Brickley (2004)] and OWL (Web Ontology Language) [W3C (2004)]. RDF is a language used to express knowledge. RDF defines a conceptual model -with theoretical foundations in description logics-, and a set of rules to serialize the model. Serialization of RDF may be performed using several syntaxes, but XML is the most extended serialization. RDFS is a basic ontology language built on the top of RDF. It is a simple language with a limited scope, and thus RDFS is used to express simple ontologies. OWL is a complete full featured ontology language, whose definition is done using RDFS, that allows definition of complex ontologies, using cardinalities, disjoint classes and a long list of features.

Web Services provides a set of that enable the publication, description and consumption of services between machines using XML based protocols. WS were designed to be used in a open environment such as Internet. So they are independent of the architecture and the programming language used by the publisher and consumer. WS can be fully described using a description language, thus, they have a loose coupled nature, enabling them to be used in Service Oriented Architectures [Endrei (2004)].

WS are founded on three technologies: SOAP for services invocation, WSDL (Web Services Description Language) for services description and UDDI (Universal Description, Discovery and Integration) for services discovery.

SOAP is a service access protocol used to invoke a WS, in its simplest form SOAP is just a XML document container. However, SOAP is not a transport protocol. HTTP, SMTP, FTP and other transport protocols may be used with SOAP, and each one provides different behavior to the communication, from request-response iterations to one way messaging. A WSDL document is a contract between the service provider and the service client that states how the client can access to the service. Information included in WSDL contains the service interface, data types used within the WS, binding information about the transport protocol and the address in which the WS may be located. Finally, UDDI is used to discover WS throw a directory similar to the yellow pages. UDDI is itself a web service with a SOAP interface. UDDI has two groups of services: one used to publish new web services and another used by clients to search services across the directory.

SOAP, WSDL and UDDI are technologies that provide the minimal set of fea-

tures needed to build the Web Services Architecture. They were not designed to include advanced features, instread they include an extension mechanism that adds advanced features without adding complexity to the basic standards. Thus, a whole set of complementary technologies that extends basic WS capabilities, known as WS-* standards, have been developed.

2.2. *Information Integration*

The work done in the field of information integration in the last years [Wache (2001)] has focused on the use of the ontologies. The main ontologies application in information integration is the definition of the shared information models. An interesting ontology based solution for distributed on demand information integration is InfoSleuth [Nodine (2000)]. It is an agent based system that creates three types of agents: user agent, for user interaction; resource agent that acts as mediator with the information system and finally the core agent that is the kernel of the system. InfoSleuth shows a set of interesting features such as notification or complex query processing. It is a pre-Semantic Web solution, and uses ontologies described in OKBC, and thus, is difficult to integrate within the Semantic Web. Searchy has been designed with the goal of integrating it in the Semantic Web.

With the development of the SW, ontology research has made an intense use of SW standards. Semantic integration of information has been influenced by this evolution and many solutions have been created using SW technologies. Buildind Finder [Michalowski (2004)] is a domain specific tool aimed to retrieve and integrate information about streets and buildings from heterogeneous sources and presents them in satellite images. Information is gathered from Microsoft TerraService, US Census Tigerline files, and Yahoo White Pages, thought a mediator and it is expressed in RDF. User agents can extract RDF graphs using RDQL, a RDF query language. Building Finder is domain specific, and thus is not suiteable as a general information integration that the context considered in this paper requires.

Vdovjak and Houben [Vdovjak (2001)] present a semantic interface for querying heterogeneous information sources. This application translates XML documents into RDF. Thus, it is not able to access to information systems that do not generate XML output, and it depends on a wrapper to get the XML. This wrapper is quite similar to Searchy wrappers. In our opinion, the described approach lacks of distribution to fulfil the requirements of horizontal integration, as Searchy does.

New information integration systems have been created using WS. A web services based solution is SODIA (Service-Oriented Data Integration Architecture) [Zhu (2004)] and, as its name shows, it has a SOA architecture. By using a SOA approach, SODIA has many of the benefits of using an agent technology. However, this is a process centric solution and has limited semantic support. Searchy is a solution focused in information, instread of processes, and it is fully based on a semantic model to share information.

Finally, not much work has been done joining Web Services with Semantic Web

for information integration. The solution most aligned with Searchy is Knowledge Sifter. It is an agent based approach that uses OWL to describe ontologies and WS as interface to the agents' services. The location of agents is done by using the most common discovery mechanism in WS, the UDDI directory. Knowledge Sifter is composed of several agents that perform some tasks such as ontology mappings, complex query and results ranking. The focus in Searchy has been to create a platform in which new arbitrary wrappers can be integrated meanwhile Knowledge Sifter is a complete solution focused in a domain area.

3. Searchy overview

The Searchy project was born with an objective: localize and describe heterogeneous resources located in different administrative domains [Barrero (2005)]. A distributed approach for this solution has been chosen in which the application is located in several cooperative agents that wrap different information sources. The result is the creation of a virtual information source that encapsulates the access to several heterogeneous information sources. A weak point in this kind of application is the lack of generality. Typically integration solutions are created to operate in a specific domain, and it cannot be easily moved to another application domain. Searchy is a general solution suiteable for a wide range of applications.

On one hand Searchy can be used as a metasearch engine that receives a query from the user and distributes queries across the network of agents. These agents map an abstract query into a local query format understood by a local information system, submit it and map the response into a shared data model, for instance, Dublin Core, integrating the received responses. On the other hand Search can be used as a complete information integration system, in which a piece of information can be retrieved from several information sources, integrating the results to obtain a piece of information that is the sum of the fetched parts. The information model is based on ontologies described with OWL or RDFS. It performs structural as well as semantic integration of arbitrary information sources that includes -but it is not limited to- databases, directories, indexers and web search engines.

Searchy performs several tasks to fulfill the integration. From a functional perspective, it receives an abstract query, not bounded to any information system. The query is expressed using an abstract syntax, so Searchy translates it into a syntax understood by the local information system and sends the query. The response is processed to map it into a generic information model and syntax. If more than one information system is accessed by the agent, the responses are integrated using the previous general information model.

Web Services based agents allow Searchy to be used in a wide range of environments, from web applications to heavy desktop applications. Searchy clients may be simple interfaces between users and Searchy to get queries and visualize results. But it also may be used by other applications as data source to perform all kind of tasks, from sensor integration to grid resource planning.

The proposed solution considers the intrinsic information integration problem, as well as it analyzes the context in which this type of application is used, and these considerations have an important weight. Information integration is, by itself, a first order problem that may increase its complexity in some magnitude order if we consider an environment in which there are no control over the information sources. This is a common situation when some horizontal integration must be done, i.e, when several independent organisations have to integrate their information systems and there is no central authority able to impose a technology inside each organisation.

To give an adequate solution to this problem, a technical and as well as a non-technical analysis must be done. In the context we are dealing with, technological requirements are close to the ones used in a B2B scene. Interoperability and portability are two first order problems that must be considered. They are well known issues and may be over run using an adequate technology. We also should must consider that when several different actors in different organizations have to cooperate, each one with its own idiosyncrasy, using a centralized, intrusive and complex solution may find obstacles that might difficult its implementation.

In such a context, we consider that a distributed, simple and federated solution is more adequate. With this approach we may obtain several advantages:

- Application execution does not need any strong central authority (distribution);
- All the agents work from an equality position for the success of the application (collaboration);
- Reuse of legacy systems in the organization, making them working as a whole (federation);
- Loose coupling between integration systems and legacy systems (no intrusion).

Searchy has been designed with two design objectives in mind. On the one hand, simplify the system as much as possible. In this way the deployment and the administration of the application shall be performed by non specialized staff. On the other hand, Searchy has been designed to be easily extensible to almost all types of information sources, with a minimum coding need. These design objectives determined the architecture of Searchy and an agent based architecture. This architecture is described in the next section.

4. Searchy Architecture

Many properties of Searchy are direct consecuences of two design decisions: the agent based architecture [Wan (2003)] and the Web standards compliance. The agent architecture gives Searchy a distributed and decentraliced nature. Web Services are used by Searchy agents as interface to access their functionalities meanwhile the Semantic Web standards are used to provide an information model for

semantic and structural integration.

The basic element in the Searchy architecture is the agent. The agent is the minimal unit with a complete funcionality, from an agent technology perspective Searchy agents have a reactive behaviour and a limited autonomy. The platform uses the agent-based methodology for two main reasons; on one hand to allow a distributed and reliable problem solving computation, on the other hand agents do not have a complete knowledge of the whole system, so the agent perspective is the most appropriate. Thus, coordination among the agents is needed in order to be a complete problem solving system, conforming a MAS. Agents in a Searchy system have similar capabilities, they integrate information, however the information sources it is able to integrate may (and likely will) be different. How agents in a MAS are coordinated is a critical issue for the success of the solution.

Coordination is based on an organisational structuring model [Nwana (2006)] with two different agent discovery mechanisms. In the first mechanism, the data flow among the agents is defined a priory, each agent has a static knowledge about where and how access to the set of assigned agents. The result is a static hierarchical structure. It is useful in order to adapt a Searchy deployment to the hierarchy of a organisation, however it cannot take benefice of parallelism agent access, the reliability of the whole system is reduced and it is difficult to integrate in very dynamic environments -to add a new agent, another agent has to be restarted.

To overcome some of these disadvantages, a more dynamic coordination mechanism has been designed. Using our previous organisational structuring model, relationship among the agents are not stored within the agents, but externally in a WSDL document that can be fetched by any agent from a HTTP or FTP server. This agent discovery mechanism is simpler than using an UDDI directory or a Directory Facilitator (DF) in a FIPA platform. To add a new agent to the system, a WSDL document should be modified manually.

Agents can have different responsibilities and different relationships, depending on the context in which they are deployed. High level policies may be defined, and each agent (human) administrator can assign responsibilities to the agents based on policies. Policies and responsibilities may or may not obey to technical reasons. For instance, one criteria can be set from an administrative point of view. Each agent may integrate the information systems of an administrative unit like an university or an enterprise department. From this point of view, agent responsibilities are fixed once the agent has been deployed, and they can not be modified.

Each agent is composed of four architectural elements, as can be seen in Figure 1. Some of the key properties of Searchy are directly derived from this architecture. Those elements are: the communication layer, the core, the wrappers and the information source. The next lines describe these components related with the FIPA Agent Management Reference Model [FIPA (2004)].

Communication layer It provides the communication related features such as SOAP message processing, access control and message transport. The Com-

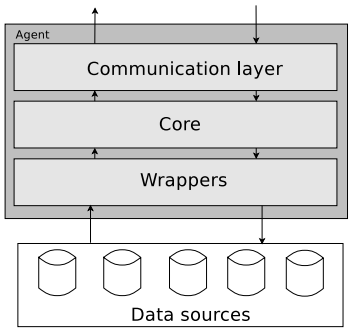

Fig. 1. Searchy agent architecture

munication layer is equivalent to the Message Transport System (MTS) in the FIPA model.

Core It contains the basic skills used by all the agents, including configuration management, mapping facilities or agent identification. Any feature required by all the agents is contained in the core. It may be considered as a wrapper container as long as its main purpose is to facilitate the development of the wrappers. A Searchy agent core contains some of the features defined by FIPA for the Agent Management System (AMS), however they are not equivalents. AMS are supposed to control the access of agents to the Agent Platform (AP) and their life cycle, meanwhile the agent core supports the operation of the wrappers.

Wrapper The wrapper is the interface between the core agent and the data source. It implements the algorithm that accesses the information in the local information system. Thus, each type of information support requires a specific wrapper. The wrapper are a key point in order to achieve generality and extensibility to new information sources with a relative ease of development. Agents in the FIPA model have some similarities with Searchy wrappers from an architectural point of view. An AP in the FIPA model may contain several agents meanwhile each agent in Searchy may contain several wrappers. Both of them, FIPA's AP and Searchy's core are containers for some software asset, agent in case of FIPA and wrappers in case of Searchy. From this point of view, Searchy can be defined as a wrapper platform. FIPA's Directory Facilitators (DF) are implemented in Searchy as wrappers since the capability of agent discovery and access are included in specialized wrappers.

Data source It is where information that is the object of the integration process is retrieved. Almost any digital information source might be used as data source. The only condition is that the information must be readable. Due to the nature of Searchy, data sources are usually some kind of information

system such as a web server or an index. However any source of digital information is a potential Searchy data source. There is no equivalent in the FIPA model to data sources.

The agent is accessed through a Searchy client, that may be any piece of software able to invoke a web service. It can be a web page, a heavy application or other agents. From the client point of view, a Searchy agent is an entity that provides a RDF document in response to a request, i.e., it perceive a single information source that in that is the federation of several information systems.

The support of new data sources is done by the development of new wrappers. There is no restriction on the algorithm and data source that the wrapper might implement. It may be a direct access to a data base, a data mining algorithm or data obtained from a sensor. This feature enables Searchy to integrate information from almost any digital information source.

At present Searchy includes four wrappers: SQL, LDAP, Google (Google API is deprecated since December 2006) and Harvest. By means of SQL and LDAP wrappers, structured data in data bases supported by JDBC and LDAP directories may be accessed. The Google wrapper uses the Google API to access its search engine. In this way Searchy can integrate information about resources on the Internet or reduce the search space into a single web site. Finally, using the Harvest wrapper, Searchy can integrate resources available in an intranet like HTML, LaTeX, Word, PDF documents and other formats.

There are two special wrappers responsable of the discovery and communication between Searchy agents: the Searchy and WSDL wrappers. These wrappers implement the coordination mechanism in Searchy, as described in section 4. Using FIPA terminology, Searchy and WSDL wrappers contain a simplified version of the white pages of the DF.

Searchy receives the HTTP request that has been sent by the Searchy client and extracts the SOAP message. In order to provide a first layer of security, the HTTP subsystem filters the request using the Access Control (AC) Module. This Module is an IP based filter that enables the definition of allowed and forbidden clients based on its IP address. The HTTP server has responsibilities with the SOAP messages transport, but the processing of these messages is done by its own module, the SOAP Processing Module. It processes the SOAP messages and it transfers the operation to the Control Module or returns an error message. Once the message has been successfully processed, the Control Module begins to operate.

The Control Module sets the flow of operations that the different elements involved in the integration must perform, including the wrappers, the Mapping Module, and the Integration Module. The Mapping Module is composed of three subsystems with different responsibilities in the mapping process. The query mapping subsystem translates the query from a general syntax into the local format, for example, SQL. Meanwhile, the response mapping subsystem translates the response from a local syntax like SQL, into RDF following a shared ontology. Both, query

and response mapping subsystems use the mapping subsystem, that provides common services. The way in which the integration process operates is described in the next section.

5. Information integration in Searchy

Integrating information means dealing with heterogeneity in several dimensions [Busse (1999); Wache (2001)]. Technical heterogeneity can be overcomed by selecting the proper implementation technology. In our work it has been done using WS as interface to access the service. To address information heterogeneity there is a need to define a shared common information model among all the entities involved in the integration process, and a mapping mechanism to perform a mapping between the different local information models and the global information model. Defining this model is a critical subject in an information integration system.

Searchy uses semantic technologies standardized by the WWW -RDF, RDFS and OWL- to represent the integrated information. RDF is basically an abstract data model that can be represented using several syntaxes. Searchy uses RDF with XML to represent the information, this combination of RDF and XML grants interoperability in a structural level. Semantic integration requires an agreement about the meaning of the information to deal with semantic heterogeneity. This agreement is performed by using shared ontologies expressed in RDFS or OWL. Then, there must be an explicit agreement among all the actors involved in a Searchy deployment to establish at least one shared ontology. The mapping between the shared data model and the local model is done by each agent.

Ontology mappings is a key subject that itself is an active area of research [Kalfoglou (2005)]. The practical profile of Searchy motivates the adoption of a conservative (and simple) approach to this problem by giving an interface in which the Searchy administrator configures manually the mappings. It is based on a string substitution mechanism very similar to the traditional printf() function in C. This mechanism is enough to satisfy the needs in almost all cases, meanwhile it is not complex for the administrator.

Query format is a tuple ¡attribute, query¿ of strings. The first element in the tuple is an URI that represents the concept to which the query is referred, meanwhile the query is a word with the content of the concept that is being queried. The query model is simple but enough to fulfil the requirements of the application. Complex querying models might be adopted, but the complexity of the management in the application might become a unuseful solution. The translation of the query to the local format is performed using the Mapping Module.

Once a query has been translated, the response of the local information source must be extracted, mapped to a shared ontology and integrated. This operation is done in two stages:

(1) The response is mapped semantically conforming with a shared ontology. It is done using the same mechanism than the query mapping. A critical aspect is

to provide a URI identifier for each resource, just like RDF requires to identify any resource. There is no unifed way to do this task: each type of wrapper and user policy define a way to name resources.

(2) Each response of each wrapper is integrated in the agent core Integration Module. Integration is based on the URI of the resource returned by the wrappers. When two wrappers return two resources identified by the same URI, the agent interprets that they are referred to the same object and thus they are joined.

A better understanding about our information integration approach can been reached by analyzing a real study case, as next section does.

6. A case study: deploying Searchy in RedIRIS network

Searchy is a joint project between the Universidad de Alcalá and RedIRIS. RedIRIS is the Spanish National Research and Educational Network (NREN) and thus it works closely with the Spanish and European academic community. The accumulated experience in some projects suggested the necessity of some type of information integration system. RedIRIS offers a complex scenario for information integration, with heterogeneous information systems, services, hardware, operating systems and strict security constrains. The heterogeneity of the information systems is complemented with a relative complex environment. In this context a first testbed of Searchy was developed.

The nature of the RedIRIS community -autonomous organizations with legacy systems of many technologies- and the need to provide an unified access to their information systems, suggested the use of a distributed solution. Modification of legacy information systems is not a chose due to the cost in terms of human resources, stability of the service and complexity. Thus, a federated non-intrusive solution such as Searchy was preferred. A first deployment within RedIRIS' own systems has been done to test Searchy in a real environment.

RedIRIS has resources stored that must be of public access through an access point. Sometimes resources are associated with a search engine published with any network protocol and/or located in a web page. The access point through the Web to those services is not unified, there are several interfaces, one for each service. Thus, it does not contribute to a better user experience. An unified access point to the resources stored in RedIRIS is a better option; a single point in which any user could search any resource provided by RedIRIS.

The information systems in RedIRIS involves a bunch of technologies and resources. Some of the resources are:

- A web site (www.rediris.es) that includes HTML, as well as documents in PDF and doc.
- LDAP directory with OpenLDAP.
- Distribution lists, indexed in a Postgres relational data base.
- Software repositories, accesible thought HTTP and FTP.

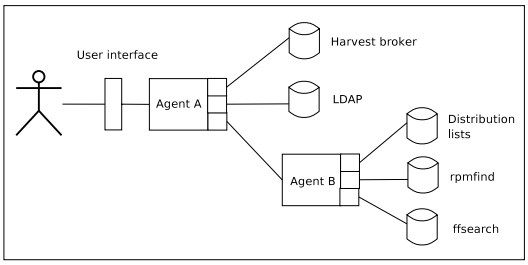

Fig. 2. Searchy testbed in RedIRIS

- RPM packages repositories, accesible thought HTTP and FTP.

To facilitate the location of resources to the user, some search systems with web interfaces have been adopted. Web pages are indexed by `Harvest` (that uses a Glimpse text oriented database). `LDAP` directory can be browsed and queried using a web based software developed by RedIRIS (http://www.rediris.es/ldap/ldap-es/navega). Meanwhile, software repositories use two different search systems: rpmfind (http://rpmfind.rediris.es) to search RPM packages and ffsearch that indexes all the FTP site. The indexes of rpmfind and ffsearch are stored in a Mysql database. More information systems are available in RedIRIS and will be integrated in the future.

The deployment of Searchy in this case study consists of two different kind of agents. The first agents (A) are the responsible to communicate with users, to access to `Harvest Broker` and `LDAP` sources, and to integrate the information retrieved from the second kind of agents (B), these agents can access to `distribution lists`, `rpmfind` and `ffsearch` sources (see Figure 2 for an schematic representation). This deployment of Searchy corresponds to security reasons, Agents B are responsible to access to secure DataBase sources which are restricted by a Firewall, so the access must be a local (secure) agent. Therefore, the agents A can manage the user queries and other (non-restricted) data source.

The goal in the practical case study is the integration of the information from the resources in order to ease location and access to RedIRIS resources. A Dublin Core vocabulary was estimated to be a suitable vocabulary since its purpose is resources description. However, other approaches were tested, such as location of individuals, where a FOAF ontology were used as shared data model. With FOAF, Searchy was able to describe people as well as resources, providing more accurate information.

The final deployment of Searchy in RedIRIS allow to integrate six different search interfaces (each corresponds to Postfix and MySQL databases, LDAP directories, Harvest brokers and Google API) into an unique one. Therefore, the user interaction with our Agent A was really simplified (see Figure 3). On the other hand, a new SOAP based search interface has been provided. Thus, any member of the RedIRIS

community can integrate Searchy service in RedIRIS within its information systems.

7. Future work and conclusions

Along this paper we have briefly presented the problem of information integration and we have proposed a partial solution called Searchy, based on the concept of service federation and wrapping. It is an agent based solution in which agents contain several wrappers that fetch information in local formats, translate them into a semantic standard format and integrates the information spread across several information systems. This agent is a wrapper platform that eases wrapper development providing an execution environment and several features.

The distributed nature of the platform that we have described in this paper has an intrinsic problem: the strong dependence of the response time with the transport mechanism. This fact, together to the agent discovery and agent communication in the implementation, limits the scalability. An advantage of using web services is the possibility of using alternative transport mechanisms suitable for different environments. There is a strong coupling between the transport mechanism and the behavior of the application. Thus, it is a promising topic that will be explored. In a near future Searchy may use SOAP multicast over IP or peer-to-peer protocols like JXTA.

The creation of a wide network of agents implies the management of huge amount of information. Then, the physical scalability of the system should be done in parallel with the intelligence system improvement. Some mechanims such as the intelligent filtering with a case based reasoning or collaborative filtering [Nowlan (2007)] are considered to provide some intelligence to the system that will produce better user satisfation. One fact that shall be considered is that information can be of a confidential nature, so there should be some mechanisms to autenticate and autorize

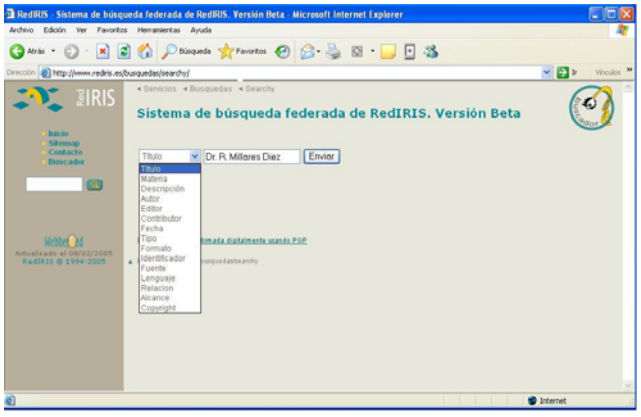

Fig. 3. Searchy user interface for RedIRIS

who accesses to the information. For this purpose Searchy is going to incorporate SAML (*Security Assertion Markup Language*).

In order to archive success in production environments, one needs to provide access to a higher number of information system types, thus the development of new wrappers is a key objetive. We also consider that the understanding of the environment in which Searchy may be theoretically used is an important step to find new application areas.

Acknowledgments

This work has been partially founded by the UAH proyect PI2005/084, in collaboration with the PTYOC program of RedIRIS.

References

Barrero, D., R-Moreno, M., López, D., and García, O. Searchy: A metasearch engine for heterogeneus sources in distributed environments. In *Proceedings of the International Conference on Dublin core and Metadata Applications* (Madrid, Spain, Sept. 2005), pp. 261–265.

Brickley, R., and Guha, R. V. *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation, Feb. 2004.

Busse, S., Kutsche, R.-D., Leser, U., and Weber, H. Federated Information Systems: Concepts, Terminology and Architectures. Tech. Rep. Forschungsberichte des Fachbereichs Informatik 99-9, Technische Universitat Berlin, 1999.

Endrei, M., Ang, J., Arsanjani, A., Compte, P., Krogdahl, P., Luo, M., and Newling, T. *Patterns: Service-Oriented Architecture and Web services*. Redbooks. IBM, Apr. 2004.

Frakes, W. B., and Baeza-Yates, R. *Information Retrieval*. Prestince Hall, Upper Saddle, New Jersey, 1992.

Grubber, T. A translation approach to portable ontology specifications. *Knowledge Acquisition 5*, 2 (1993), 199–220.

Halevy, A., Ashish, N., Bitton, D., Carey, M., Draper, D., Pollock, J., Rosenthal, A., and Sikka, V. Enterprise Information Integration: successes, challenges and controversies. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (Baltimore, Maryland, USA, 2005), pp. 778–787.

Hendler, J. Agents and the semantic web. *IEEE Intelligent Systems 16*, 2 (2001), 30–37.

Hendler, J., Berners-Lee, T., and Lassila, O. The semantic web. *Scientific American 284*, 5 (May 2001), 28–31.

Kalfoglou, Y., and Schorlemmer, M. Ontology mapping: The state of the art. In *Semantic Interoperability and Integration* (2005), no. 04391 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.

Linthicum, D. S. *Enterprise Application Integration*. Information Technologies Series. Addison-Wesley, 1999.

Management, F. T. A. *FIPA Agent Management Specification*. Foundation For Intelligent Physical Agents, March 2004.

Manola, F., and Miller, E. *RDF Primer*. W3C Recommendation, Feb. 2004.

Michalowski, M., Ambite, J., Thakkar, S., Tuchinda, R., Knoblock, C., and Minton, S. Retrieving and semantically integrating heterogeneous data from the web. *IEEE Intelligent Systems 19*, 3 (2004), 72–79.

Mitra, N. *SOAP Version 1.2 Part 0: Primer*. W3C Recommendation, June 2003. http://www.w3.org/TR/2003/REC-soap12-part0-20030624/.

Nodine, M. H., Fowler, J., Ksiezyk, T., Perry, T., Taylor, M., and Unruh, A. Active information gathering in infosleuth. *International Journal of Cooperative Information Systems 9*, 1-2 (2000), 3–28.

Nowlan, M. F., and Blake, M. B. Agent-mediated knowledge sharing for intelligent services management. *Information Systems Frontiers 9*, 4 (2007), 411–421.

Nwana, H. S., Lee, L., and Jennings, N. Co-ordination in software agent systems. *British Telecom Journal 14*, 4 (October 1996), 79–88.

Vdovjak, R., and Houben, G. Rdf based architecture for semantic integration of heterogeneous information sources. In *Proceedings of the International Workshop on Information Integration on the Web* (Rio de Janeiro, Brazil, Apr. 2001), E. Simon and A. Tanaka, Eds., pp. 51–57.

W3C Recommendation. *OWL Web Ontology Language Overview*, Feb. 2004. http://www.w3.org/TR/owl-features/.

Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hübner, S. Ontology-based integration of information — a survey of existing approaches. In *IJCAI–01 Workshop: Ontologies and Information Sharing* (Seattle, Washington, USA, 2001), pp. 108–117.

Wan, F., and Singh, M. P. Commitments and causality for multiagent design. In *2nd International Joint Conference on Autonomous Agents and multiagent Systems (AAMAS)* (Melbourne, Australia, July 2003), A. Pres, Ed.

Zaine, O. R. From Resource Discovery to Knowledge Discovery on the Internet. Tech. Rep. TR 1998-13, Simon Fraser University, 1998.

Zhu, F., Turner, M., Kotsiopoulos, I. A., Bennett, K. H., Russell, M., Budgen, D., Brereton, P., Keane, J., Rigby, M., and Xu, J. Dynamic data integration using web services. In *IEEE International Conference on Web Services (ICWS'04)* (San Diego, California, USA, June 2004), pp. 262–269.