Search of Medical Resources using Semantic Descriptions of Web Services

Jose M. Almellones, M. T. Arredondo

Life Supporting Technologies, ETSI Telecomunicación, UPM, Madrid, Spain

jmalmellones@lst.tfo.upm.es

Abstract

In the latest years there has been an attempt in European healthcare to establish various collaborative association forms that help the healthcare professional in sharing the risk in diagnosis and treatment of patients. Since these associations are not supported by an IT risk management solution, the COCOON [1] project borns to support these collaborative activities. This paper presents a Semantic Search engine for discovering Medical Resources published as Web Services.

Service Oriented Architectures are the best known solution to solve differences between heterogeneous systems. They can encapsulate internal behaviours within interfaces easily accessible. They can overcome most of the structural and the syntactic problems between applications owned by different medical associations but can not deal with semantic incompatibilities. COCOON Glue has been developed to add semantic information to the description of Web Services used in these architectures. Following WSMO [6] guidelines, this information is backgrounded by a series of Ontologies describing certain domains of interest and permits to semantically search for services containing information of interest, no matter the differences existing between these services.

1. Introduction

The risk in the processes of diagnosis and treatment of patients in European healthcare is high for the professional to take a decision that could cost a lot to the healthcare system and to the patient. This cost is high if errors mean making unnecessary operations or maintaining wrong treatments. To prevent medical errors and their costs to European healthcare systems and their patients, the COCOON [1] system borns, with the aim to be an IT solution supporting the collaborative work between European healthcare associations.

With the use of new Service Oriented Architectures, it is possible to use the information from various heterogeneous systems, no matter the way they are implemented. This way we can achieve a complete information infrastructure easy accessible by professionals of each one of the communities taking part in the initiative. Each diagnosis and treatment is guaranteed by the knowledge provided by the whole system, not only by the professional in charge of doing it but also his community.

The objective of the COCOON [1] project is to manage the information provided by these European communities, usually published by means of Web Services. By an efficient and semantic use of the resources' information, help on a concrete issue can be found easily within the information provided by the communities taking part in the project. This is possible thanks to the COCOON Glue, the semantic Web Service search engine of the COCOON [1] project.

The primary objective of the COCOON Glue is to facilitate the semantic publication of existing and new information Web Services provided by the European communities of practice. The COCOON Glue also has to ease the searching for medical information Web Services within these communities.

The second objective is to permit the dynamic addition of functionality to the COCOON [1] solution, making it easily expandable by adding new elements to the architecture. These elements should not have any fixed characteristics or technology requirements. These elements only have to be published in the form of Web Services and they will be used by the system. When a component of the system needs a functionality, it searches for it in the COCOON Glue and uses the web service it returns.

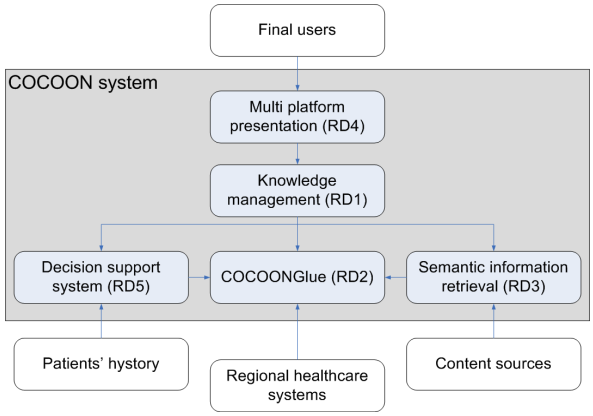

Figure 1: COCOON [1] Structure

As can be seen in Figure 1 the architecture of the COCOON [1] project is dependent on the COCOON Glue that is in the middle providing search capabilities to the other components. This way, the other components can find the functionality required, and more functionality can be added in any moment. It can be seen also in Figure 1, how the communities of practice can also publish their information Web Services directly in the COCOON Glue.

Also, Figure 1 shows how the whole COCOON [1] project depends on the COCOON Glue to use all the functionality offered by all the rest of the system components.

2. Methods

The architecture of the COCOON Glue has to be based on Web Services, so there has to be a web interface exposing COCOON Glue's functionality and a tier for the business logic. The web interface has to permit the publishing and deletion of semantic Web Service Descriptions (WSD). This interface has also to permit searching semantically the repository for WSDs previously published and to get the results.

The publication of a WSD should be based on the concept of Web Service Description classes, a form of abstraction aimed to generalize the characteristics of WSDs. So a class in conjunction with the necessary information will permit to create a concrete WSD.

The discovery of a Web Service starts after a Goal instance is submitted through the search functionality of the interface. To create a Goal instance, the external client needs to provide the Goal class name and information necessary, like in the case of WSD instances.

The discovery is performed by matching the information contained in the Goal instance with the capabilities described in the previously published WSD instances that are stored in the repository.

Once the discovery is finished, a list of discovered Web Services, meeting the requirements in the Goal, is returned. This list is parsed and returned to the client containing the URL of each one of the services discovered in the matching.

3. Results

The system developed has a Web Service architecture. It has two different Web Services. The first is for publication and permits the publishing and deletion of semantic Web Service Descriptions (WSDs), that are semantic enriched Web Service descriptions. The second is for discovery and it allows a client to search the repository for WSDs previously published and to get the results. This is possible thanks to the submission of a query, called Goal in the terms of WSMO [6] that contains all the characteristics that are searched for.

As can be seen in Figure 2, the system contains a central repository in charge of storing the WSMO [6] key elements: ontologies, Goals, WSDs and mediators. This central database-driven repository is accessed by the corresponding managers that store and retrieve the necessary elements from the repository through the repository manager. This manager can be seen as a database abstraction layer.

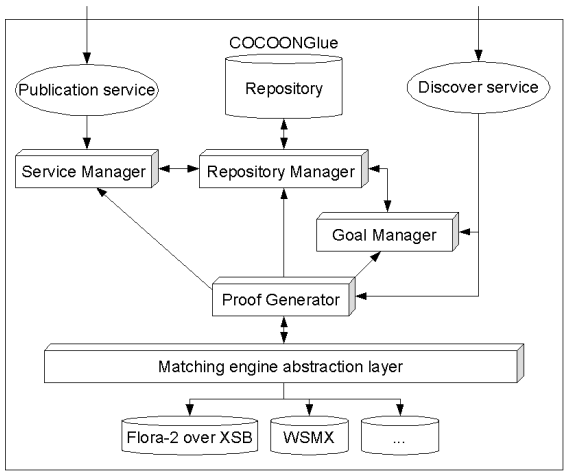

Figure 2: COCOON Glue Architecture

A series of requirements are necessary in order to publish a WSD or submit a Goal for semantically search for Web Services:

- To publish a WSD instance, the system receives the name of a WSD class and an XML [2] fragment with the parameters.
- To submit a Goal instance, the system needs a Goal class name and the XML [2] document with the parameters.

There is a template system illustrated in Figure 3 to generate a complete WSD or Goal from a reduced set of parameters included in the XML [2] fragment of the web method call. These parameters are converted in a valid WSD or Goal in the language understood by the matching engine.

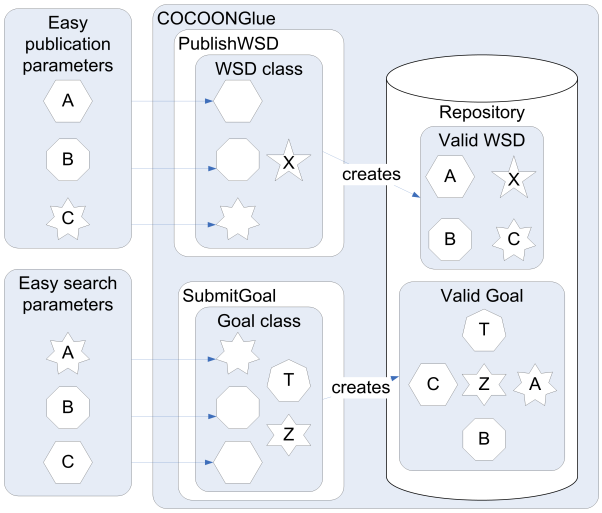

Figure 3: Template system, easiness of creation of Goals and WSDs

The transformations needed to convert the parameters given in the web method call are done by means of XSL transformations of the XML [2] depending on the WSD or Goal class used. When a class name is submitted, the corresponding XSL style sheet is grabbed from the repository and applied to the XML to make an instance in F-logic. This result is stored in the database for later use in the matching process.

In the current stage of development of the COCOON Glue, this language is Frame Logic, the language understood by Flora-2 over XSB.

The discovery is performed by this engine, Flora-2 over XSB, which needs the information contained in the Goal instance and the published WSD instances that are stored in the repository and related to the Goal. The information contained in the Goal is used to match the information previously stored in the repository. This process is called “the matching”.

In order to perform the matching we need the Goal its class and all the ontologies referenced by the class of the Goal. We also need all the wg-mediators related to the class of the Goal and their associated ontologies. Each wg-mediator associates one Goal class with one WSD class. This way, before the matching there is a set of associations that tells wich WSD can be an answer to a Goal's requirements. It is important to see how much impact in the performance could have this if all the WSD were included in each matching session of the matching engine.

4. Discussion and Conclusions

The system has been completely developed following a Service Oriented Architecture having Apache Axis [3] in the base and running on top of Apache Tomcat [5] as the application server. It is working well inside the COCOON [1] project and only some minor issues have to be solved in order to achieve the final version of the COCOON Glue.

The use of Apache Axis [3] speeds up the development and the deployment of the Web Services. As can be seen in Figure 4, the process is clear and easy, and can be automated with the use of scripting languages such as Apache Ant for more easiness. Starting from server classes, a Web Service can be created and mapped by using an Apache Axis [3] deploy file. This way the WSDL description of the service will be available for any user that wants to use the service. In order to do the tests, Apache Axis [3] is used to generate the stub source code to access the service from the WSDL description. Then the test client source code is told to use the stub code and test the service functionality.

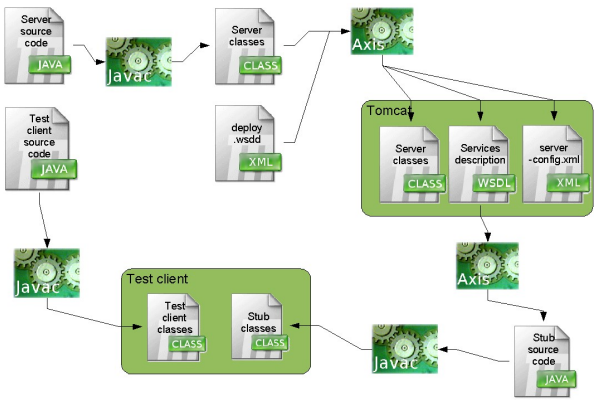

Figure 4: Web Service development with Apache Axis

The interfaces make use of XML [2] for the transmission of the parameters necessary for the publishing and searching.

Currently the repository is database and file driven, making use of MySQL [4]. The repository has the necessary relations to know the ontologies that each WSD class, Goal class and wg-mediator use. In addition to this, there are relations to know the class that corresponds to each instance of WSD or Goal and the hierarchy between WSD classes and the hierarchy existing between Goal classes. So in the database there are WSDs stored, being each an instance of a WSD class, that can have a parent WSD class or not. The same occurs to the Goals and Goal classes.

Currently the system supports two ways of making the discovery:

- There is a synchronous mode in which the matching is triggered by the submission of a Goal. In this mode the client has to wait for the matching to be done for getting the results. Also in this mode saturation in the call of the service can occur because the queue system used to store petitions is the one provided by the application server.
- The other way is a asynchronous mode in which the matching is done automatically when there is any Goal to be satisfied. The client publishes the Goal and periodically ask for the results. When the results are ready the Web Service returns them. In this case there are no queue problems since the petitions are served almost atomically.

It has to be taken in account that in the first mode there is no need to store the Goal in the repository. But in the second mode, all Goals are stored in the repository and so they are the corresponding results until they are taken by the client. There is space in the database reserved to the results of the searches when the system is working in the second mode.

The semantic execution environment is now implemented with Flora-2 [8] over XSB, a Frame Logic interpreter but it will soon be replaced by WSMX [7], the execution environment of WSMO [6].

The functionality of the COCOON Glue is near completion but the changes in technology and the migration to WSMO [6] standards could require some refinements in the implementation, like the completely migration of the repository to the database. This change will require the development of ontology, mediator and classes editors in order to make changes to the repository directly modifying the database.

5. Acknowledgements

The "COCOON Glue" work is a complete integrated system part of the COCOON [1] project, "Building Knowledge driven and dynamically networked communities within European healthcare systems" (IST-2002-507126). COCOON [1] is a 6th Framework Project of the IST programme.

6. References

[1] COCOON Network, the medical network for the EU. URL: http://www.cocoon-health.com

[2] XML, eXtensible Mark-up Language. URL: http://www.w3.org/XML

[3] Apache AXIS. URL: http://ws.apache.org/axis

[4] Mysql Database. URL: http://www.mysql.com

[5] Apache Tomcat. URL: http://tomcat.apache.org

[6] Web Service Modeling Ontology. URL: http://www.wsmo.org

[7] Web Service Execution Environment. URL: http://www.wsmx.org

[8] Flora 2. URL: http://flora.sourceforge.net

Address for correspondence

Jose Miguel Almellones Cabello
Life Supporting Technologies
ETSI Telecomunicación. B-321
Ciudad Universitaria s/n
Madrid 28040
e-mail: jmalmellones@lst.tfo.upm.es
Tel. +34913366834