Implementation of Crawling and Locating Services in the eGovernment Domain

Yashwant Singh, Ila Kumar

1CSE Dept, BRCMCET, MD University, Haryana, India
2Gautam Budh University, Noida, U.P., India

Abstract: As we know there is an increasing number of services available for citizen in eGovernment domain. This can introduce some level of confusion for citizens that may encounter difficulties to locate the desired service. Thus, mechanisms to search and locate a particular service must be provided. In this task, the use of semantic technologies is expected to play a paramount role. This papers aims at presenting a two-fold mechanism to fight back this issue. The proposed solution will be able to make semantic annotations on services and to crawl eGovernment web pages in search for services. Finally, these services will be put at the disposal of citizen by means of a semantic search engine. All details regarding the semantic support and the technical details are fully discussed in the paper that also lays some conclusions.

INTRODUCTION

Searchers in the domain of eGovernment are currently benefiting from a multi-disciplinary effort towards a more mature state of the art. The evolution on the quality of eGovernment services comes along with an increasing number of possibilities for citizens. And this may turn out to be a burden for citizen. Citizens may get confused or lost when looking a particular service among the overwhelming amount of possibilities and platforms for different agencies at different administrational levels. To rectify this problem, solutions aimed at helping citizens to locate the proper service across different Public Administrations or Agencies (here after PAs) must be developed. And, in the current state of the art, this is a not solved yet issue.

It must be kept in mind that most of eGovernment services are delivered by means of web portals. Thus, addressing this task involves tackling the exploration and classification of web pages from PAs. Currently, a citizen must explore through all possible involved PAs in advance to discover which one is the most suitable PA; and then, he/she must look for which is the actual service most suitable for his/her needs.

In order to tackle this situation this papers presents a contribution that faces these problems. The path towards the solutions begins defining which the business model we must tackle is (see Section II). In order to face that environment, semantics can play a paramount role. Therefore, a review of current state on semantic technologies is provided (see Section III). By means of this technology, it is provided a set of semantics annotations using OWL and later on transformed in a microformat support (see Section IV). Using this semantic annotation as the cornerstone for the system, an entire software system is developed (see Figure 1). This platform includes a tool to make annotations on web pages and also to store that annotations on a remote DBMS (see Section V). Later on, those contents semantically annotated are indexed by means of a crawler (see Section VI). The last step is the provision of a web tool to put at the disposal of the citizen the contents already gathered and stored with in the system (see Section VII). Finally on Section VIII, some conclusions are presented.

THE BUSINESS MODEL

The very first step in the provision of the solution is the identification of the business model we must deal with. Many different PAs are providing services under different models and based on different concepts. So, we would need to identify some sort of pattern for the description of those services that are actually being provided by PAs.
Currently, it is not uncommon to find services modeled under the term of Life Events in different approaches from both public and academic sources.

In the scope of this work, we are prone to use a lower level definition of service. At this point we bring into focus the proposal presented in conclusion part. This work introduces its own definition of Life Event, in a quite similar manner to already deployed solutions, but it also proposes the use of the artifact Administrative Service, a concept used to model services in the context of PAs.

Administrative Services are those services that take place just in a single office and generate, as output, some documents for the citizen. These last ones, the Administrative Services (ASs hereafter), actually fit in the aim of the present proposal. Thus ASs, as they were defined in [8], are reused in this proposal. To describe an AS, it is required to know about the following aspect:

- **Title**: Brief name for the AS.
- **Description**: A brief textual description about the service for the citizens.
- **Max Span Life**: The maximum span of time for the response from the PA before the operation is considered approved/dismissed.
- **Public Administration**: Information about the PA that it is responsible for the execution of the AS. Therefore, it can be used to decide about the scope of the operation
- **Input Documents**: The documents the citizen needs to be in possession of to be able to invoke the AS.
- **Output Documents**: These are the documents that will be generated as output in case the AS is completed as expected.
- **Area**: The group of services this AS can fit at.
- **Location**: The URL where the service is hosted.

At this point, it must be brought into focus the key importance of documents. These are the legal contract linking the PA and the citizen in the fulfillment of an operation or in declaring a status or circumstance and they are considered as the input for the invocation of an AS and also its output, i.e., the legal proof for a new status or the requirement for another further operation, i.e., AS.

**SEMANTICS**

The Semantic Web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users.

The gist of this idea is to make machines capable of understanding the information within the web. This feature will allow them to make more complex interactions without the need of human support. To accomplish this ambitious goal a long evolution on the technological side has taken place during these last years.

According to the previous article, the semantic web is: “an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation” Therefore, semantics is compelled to provide information, not just data. In other words, semantics introduces meaning in to the data in order to allow computers to deal with this information in a more interoperable manner. The objective of this discipline is the provision of information understandable by machines. To achieve this ambitious goal, ontologies are a key element.

An Ontology is a formal, explicit specification of a shared conceptualization of a domain of interest.

To express ontology in a formal manner, different languages are at our disposal. OWL (Ontology Web Language) is the W3C Recommendation intended to provide a fully functional way to express ontologies. To make different levels of complexity possible, OWL provides different sublanguages with increasing expressivity: OWL Lite, OWL DL and OWL Full. By using OWL, we are addressing a W3C recommendation that can be considered as solid and interoperable support for the provision of this solution.

Regrettfully, all the power within this tool, OWL, may be not be needed and, on the contrary, it may introduce too much trouble to develop simple-to-use solutions. And this must be avoided, specially, in those cases where the semantic features of the system are quite close the human user. To deal with these cases, there are other options available such the use of microformats. According to its creators.

So, we can consider those as light-weighted semantic tools to represent information. The price to pay for this simplicity to use and possibilities for quick adoptions is related to the Loss of power to express complex rules, relations among concepts and inference possibilities. Despite all those problems, microformats could be considered as a successful technology in the scope of the Semantic Web with a quite broad use.

Figure. 2: Ontological Model
Describing Knowledge

In order to describe the knowledge in the system, the use of an ontological support is the most convenient tool. In our case, we will take advantage of former works in the area.

This information is provided using an ontology (check figure 2) supported by OWL, an official recommendation from the W3C to express semantic knowledge.

This semantic information involves a large effort in the modeling of the business model implicit in the problem and is supposed to take into account all relevant features in the system.

- Modeling the artifacts LE and AS.
- Support for annotating the legal framework for ASs.
- Management of legal documents from citizens.
- Relations of locate which PA is in charge for each AS.
- Connections between required and generated documents in each AS.

Nevertheless, in our approach this level of semantic information could exceed actual requirements of the system. Besides, it could be difficult for normal users or even civil servants to generate the full semantic description. So, in the context of this solution it was decided to take advantage of a microformat specification. To keep a balance between the complexity of the system and its simplicity of use, some restrictions were put on the semantic model to make it fit the proposal of microformats. Therefore, the solutions is just concern on modeling ASs and for its definition, a number of relevant fields were identified taking the ontological model in Figure 2 as the ground.

- Name: name of the service itself.
- Description: brief description of the service.
- PA: public entity in charge for the service.
- Input Document: name of the document required to invoke the service.
- Output Document: name of the document generated as output for the service.
- MaxSpan: maximum delay in the execution of the service, expressed in days.
- URL: address of the web page where the service is accessible.
- Areas: Areas of interest for the AS.

This way a owl instance is transformed into a simpler RDF representation.

In order to allow its inclusion in an actual Web page, it is also provided a XHTML compliant code:

```xml
div class="AdministrativeService">
  <span class="name">Request Birth Certification</span>
  <p class="description">Service to gather a signed document stating the date and place of birth of a citizen</p>
  <span class="PA">Vigo City Hall</span>
  <span class="InputDocument">IDCard</span>
  <span class="OutputDocument">Birth Certification</span>
  <span class="MaxSpan">5 days</span>
  <a class="Service">http://palocal.det.uvigo.es/service.jsp?id=23</a>
</div>
```

Of course, a generic Cascading Style Sheets (CSS) [16] is also provided. This CSS makes possible that the contents are properly displayed on a web page in case the corresponding web master decides to use it.

ANNOTATING WEB PAGES

It must be kept in mind that the effectively of this approach lays in the amount of users we can drag into our proposal.
Asking PAs to update their web contents may turn out in a low rate of successful implantation of the solution. To fight back this issue, it is provided a simple to use software tool that will perform two different tasks: creating the microformat content and storing in a remote database this information.

The software tool will provide with a simple user interface see Figure 3). The user can fill up a form about a certain service with all data required to its full characterization. Upon he completion of this information, two different options are available. The first option is to generate the microformat code itself (check button "Generate Code" on Fig. 3). As usually the person generating these contents will not be on charge for updating he web, this platform offers an alternative approach to take Full advantage of mining tools and semantic data recovery.

Once the information is introduced, the user can submit the information (check button "Submit" on Fig. 3). This button will invoke a Web Service that will introduce the information on a remote server where all these pieces of data are stored. Thus, this information will not be lost and, as shown in further sections, users can still benefit from this effort.

**CRAWLING THE WEB**

The presented system has also the ability to explore the World Wide Web in search for new services. Those services, once located, are analyzed and all available information is Inferred and stored in the system. Actually, as previously mentioned, the information with in the system is managed by a database populated with semantic annotations. Besides of the method based on annotations made directly by users (check Section V), the system will acquire new contents by means of its own exploration of the web for contents annotated by the introduced micro formats.

This task is performed by a crawler. Developed in Java, this software agent is implemented using a number of software libraries to download HTTP content, scan HTML contents, and manage information using libraries such as Jaxen [17] and Jena [18].

To perform this task, the system is fed with an initial set of known URLs corresponding to certain web pages with contents relevant for the application.

Once the target web is located, and downloaded, the crawler performs two different operations:

- Explores the web in search for the information introduced in the web page itself using the micro format annotations.
- Scrutinize the HTML code of the page in search for further links that may content new pages to feed again the process.

The process to locate new suitable web pages to increase the pool of pages known by the system is quite simple. The system just scans the HTML code downloaded in search for tags a and frame. Those HTML entities include the attribute src that points to web pages of potential interest, in the same web site or in external ones. All of them are used for future iterations of the crawler.

Nevertheless, the most relevant operation for our purposes is the second one: obtaining the information from the web pages under analysis. This operation is undertaken on the HTML content already downloaded for each target. The crawler must look for the micro format information included there. This operation is performed using Xpath [19] expressions designed to extract the information compliant with the proposal presented and Xerces [20]. Therefore, it was only possible to perform evaluations on ad-hoc contents.

**GATHERING SERVICES**

The final goal we are looking for involves the provision of a simple-to-use mechanism to store and recollect information on the web regarding services provided by PAs to citizens. In order to make that possible it was introduced a semantic support (check Section IV) that was used to annotate web pages (check Section V) that would be indexed later on by a crawler (check Section VI). The next and ultimate step is gathering that information from the common place where it was stored by all the software agents involved in this solution. Therefore, citizens, the final users of the proposed solution, can take full advantage of the system.

In this approach, the searches are conducted by mean of a web interface (see Figure 4) where the citizen introduces information about the desired AS, a normal service from his/
her perspective. With this information, the semantic engine will present as response all the data matching the expressed conditions.

Figure. 4: Web Interface for Searching ASs

Once the web form is filled up, the system will generate the SPARQL [22] query containing both the information from the web form and the information from the profile of the citizen. Generated queries are similar to the following example:

```
SELECT ?ASid WHERE
{
?ASid rdfs: typ e as: AS.
?ASid a: isSupportedBy: PA X.
?ASid a: requires: Doc X.
?ASid a: generates: Doc Y
}
```

So this query will return those ASs that are assigned to the PA PA X, that require Doc X to be invoked and that generate Doc Y as output. Queries introduced on the system are executed on the server that manages the DBMS using the support provided by Jena. The result of the execution of this query is presented to the user and the direct navigation to the service provided along with a detailed explanation of the AS.

CONCLUSION

This paper proposes a solution to facilitate the searching of the desired service. The first step is the proper definition of what an eGovernment services is and how this can be modeled. In this approach, we use the introduced artifact identified as Administrative Service. This is used as the ground for the entire system and taking advantage of the use of semantics, it is provided a simple solution to locate the desired service.

Services potentially at the disposal of the citizen are stored in a pool. This pool of services is populated using two different sources. The first option is the information gathered for a crawler, developed along this proposal, that explores web site and analyzing information included those web pages additionally, it is provided a software tool that supports the semantic annotation of an eGovernment Web site that offers an actual service. This tool also allow the user to submit this information to the stock of available services. Thus, the repository of available services is supposed to be updated and filled with proper contents.

This approach is based on the collaboration of users or even providers, public administrations or agencies, willing to take part in the system. Actually, this is a feature of the utmost importance to succeed in this case. We need the commitment of all parts implied in this environment to come up with an actual solution that may overcome the current status and unneeded burdens. Therefore, to make that possible, it was released an open definition of Administrative Services and the tools for both, creating the microformat-based annotation and toring the contents as they are defined in the context of this work. Achieving the critic mass for this project will be the key to its final success.

As the prototype is intended to be used as a proof-of-concept, some issues related to security have not been taken into account. Further solutions aimed to provide a regular service must deal with features such as legitimacy of information, privacy of citizen, local legal regulation, etc.

REFERENCES