Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol I,
IMECS 2010, March 17 - 19, 2010, Hong Kong

A RESTful Approach For Platforms In The eGovernment Domain

Luis Álvarez Sabucedo and Luis Anido Rifón *

Abstract— **The provision of support for operations in the domain of eGovernment is a complex task that has been driving the efforts of the community for the last years. This pursuit aims to achieve not just a good support for present service but also for the close future. Scalability, traceability, and security of the final solution are features of the utmost importance in this context. In this line, the paper presents a holistic approach based on the principles of RESTful to support services within the domain. Authors also take full advantage of semantic resources that are used to boost the capacities of the system due to its possibilities to provide a complete and explicit formalization of the knowledge involved. Thus, advanced features will be possible, as shown in the paper. Also, some conclusions are presented.**

Keywords: eGovernment, semantics, software architecture, interoperability, Web-based solutions

1 Introduction

Providing a software architecture to host domain services is far from being a simple task. On the contrary, the definition of a suitable architecture for a particular environment is a never-ending task towards, not just a good technical solution, but also adoptable for all parties involved. One particularly complex scenario is related to eGovernment. In the frame of this domain, it is required to comply with additional concerns regarding security and privacy beyond the constrains present in other related domain (check Section 2). And that is the frame of this work whose aim is to propose a software platform that allows the deployment of services in a simple and straightforward manner.

First of all, it must be considered a solid definition of the environment. This definition must be able to address all relevant features with in the solution. This task is not actually a simple goal. As a matter of fact, several projects made their try on this issue and the results obtained, even of high quality, did not achieve a general acceptance. This is due to a lack of maturity in the domain that does not allow to evolve the current state of the art. This paper presents its own suggested model of the domain taking advantage of previous works of the authors (check Section 3). In this definition, there is a technological tool that will play a paramount role: semantics. This technology is briefly described on Section 4.

The next step is the use of the RESTful principles to accomplish this project (check Section 5). The application of this model to our solution is shown on Section 6. As the reader would note, this proposal combines the advantages of the above mentioned architectural style and power to refer to knowledge provided by semantic tools. Finally, upon the achievement of the result, conclusions and final considerations about this work are presented in Section 7.

2 The digital administration

The digital administration or eGovernment is experiencing during these last years a huge development boosted by both the demands from the citizens for better services and the constrains imposed by the Public Administrations (hereafter PAs) themselves. These efforts are oriented to drive the transformation required for the actual implantation of actual eGovernment solutions. Developers and stakeholders should not consider eGovernment as a simple replacement of a paper-based administration with electronically-driven services. On the contrary, the arising of eGovernment solutions should involve a deep transformation of the horizon for services. In this context, it should be considered the change from process-based applications into new paradigms of services focussed on the actual needs of the citizens. This aspect of the problem is clear from the definition of eGovernment provided by the World Bank[1]:

> the use by government agencies of information technologies (such as Wide Area Networks,the Internet, and mobile computing) that have the ability to transform relations with citizens, businesses, and other arms

In order to accomplish this ambitious goal a deep reengineering process should be undertaken. In this long-term process some issue may not be overlooked:

*Universidade de Vigo (Spain)
Telematics Engineering Department
Email: {lsabucedo,lanido}@det.uvigo.es

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- Interoperability. Addressed by all major institutions, specially by EU[2], must be considered at all phases of the project. Final solutions derived from current initiatives must properly address this feature at different levels, namely: technical, application and semantical.

- Accessibility. The fight agains digital gap is clearly present in this domain. PAs can not afford to prevent users from using their own solutions for problems related to accessibility or availability.

- Maintainability. Solutions in PAs are expected to have long life-cycle. Applications will be in use for a long time and task related with maintenance must be taken into account during the design and deployment phase.

The final goal for this sort of platforms is the holistic support for services in the context of PAs. This ambitious goal have been tackled in past. We would like to outline the contributions from some of them, namely:

- The SemanticGov project[3]. This project supported by the 6th Framework Program aims at developing a software infrastructure intended to provide support for PAs. Semantic technologies play a main role as it is based on the use of WSMO to provide interoperability mechanisms.

- The ALIAL initiative[4]. An Open Source initiative aimed to provide with software for Local Public Administrations in Spain. These projects will be based on open source, open platforms and interoperability.

3 Modeling the objective

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 LifeEvents in different approaches from both public and academic sources such as [5, 6, 3, 7, 8, 9, 10]. Nevertheless, this concept is not always used to refer to the very same thing. Also, it is used most of times at very high level of abstraction.

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 [11]. This work introduces its own definition of LifeEvent, in a quite similar manner to already deployed solutions, but it also proposes the use of the artifact AdministrativeService, a concept used to model services in the context of a particular PAs.

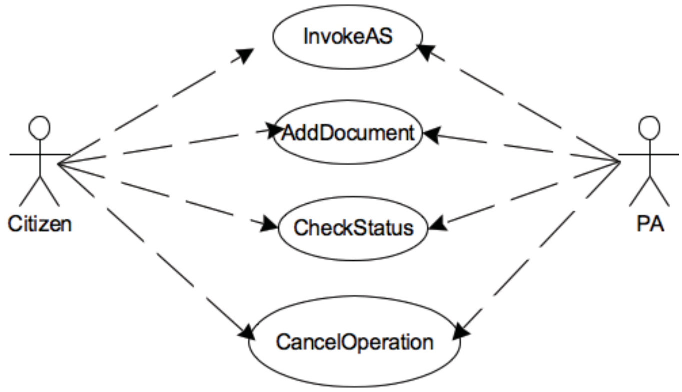

Figure 1: Uses cases supported

The first artifact, LifeEvent (hereafter LE), is used to speak about high level services that may involve a number of PAs, but the latter refers to those services provided by a single PA in a straightforward manner. Actually, AdministrativeServices (hereafter AS) are those services that take place just in a single office and generate, as output, some documents for the citizen. Therefore, ASs can be considered as a second level of articulation to deploy LEs as they fulfill needs at a local level and the coordination of several may be needed to compose a whole LE.

As the intended goal is to provide with support for invoking services from a single PA in a simple and straightforward manner, they actually can be considered to be basic tool in the frame of this work. Thus ASs, as they were defined in [11], are re-used 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.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Citizens, final users of the solution, will need to able to perform a set of operations regarding this artifacts within this proposal in order to make possible the fulfillment of their needs. These required operations are namely (check Fig. 1) :

- Invoke a operation. Given the desired AS, the citizen can request its invocation.
- Add a document to an in-progress operation. After its invocation, it could be required to add a new document to complete the portfolio of a service or just in response to an update on the status.
- Check the status of an invocation. This use case allows the citizen to be informed about the current status of the operation requested.
- Cancel an operation. In case of any circumstance arises, the citizen may decide the stop the requested service.

4 Applying semantics

The semantic web has emerged as a new promising technology aimed at addressing information instead of data, i.e., it enables software agents to treat data in a meaningful manner. Making this possible would allow new mechanisms to operate on a higher level of abstraction. Also, by means of this technology, it is possible to express knowledge in a formal and interoperable way.

The "semantics", as an IT researching field, was born in the early 2000's. In May 2001, Sir Tim Berners-Lee published the foundational article presenting the semantic web to the world[12]. According to this article,

> 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. 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."

A key tool in this path towards a *well-defined meaning* is the use of ontologies. They can be considered as a knowledge representation expressed in a formal manner with different languages are at our disposal. OWL (Ontology Web Language)[13], the W3C Recommendation, is the chosen option in our case. In particular, we will take advantage of OWL DL. Using the slots of information identified in the previous section and making use the methodology METHONTOLOGY[14], a complete an ontology is derived. Therefore, all agents in the system will share this definition of knowledge and will be in position to know what the exchanged information actually means.

The ontology developed in this context fits on the criteria shown on Section 3. Thus, it is centered in the definition of AdministrativeServices (AS) as the definition of the service that is intended to be used. Each one of the realizations of this artifact is called as "invocation of AS"" (hereafter iAS). While the former is the definition of capacities and abilities of general service, the latter is linked with a particular invocation of an AS from a particular citizen.

In the definition of the properties of a AS, fields identified in the previous Section are used. Therefore, a AS will include information such as Name, Description, Legal Framework, Required input documents, etc. Correspondingly , the definition of the iAS involves properties identified to keep the link between the definition of the service and required data to track the operation. A review of its main properties is presented in Table 1. These ones, AS and iAs, are the resources to be used under the proposed RESTful approach in next section.

5 The RESTful paradigm

As mentioned before, in order to deploy a convenient software platform, it is required to make a wise election of an architectural style, i.e., the guidelines and the principles that will rule the actual software architecture for the prototype. In the frame of this work it was decided to make use of the REST (*REpresentational State Transfer*) philosophy[15]. System fitting in this principle are called to be "RESTful" and, taking the client-server model as the basis, it poses some principles:

- Resources to be used always must be able to be identified from the outside of the solution. One of the aims of this sort of architectures lays on managing data, i.e., resources, regardless of their storage, origin or status. Thus, it must be provided in a unique manner to establish a link with them. The preferred way is by means of URIs (Uniform Resource Indicator) and IRIs (Internationalized Resource Identifiers). As a result, it is possible to get an unlimited number of resources.
- Messages, i.e., commands, must be a set of clear and simple verbs. Contrary to what happen to resources, actions must be limited and well-known. Usually, these systems are supposed to be based

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table 1: Properties for iASs

Name	Multiplicity	Description
ASOriginator	1	Reference to the AS corresponding to the invocation
Invoker	1	Citizen invoking the AS
LaunchingDate	1	Date of the invocation of the AS
Deadline	1	Latest date for the execution of the AS
DocumentsAttached	0..n	Documents provided within the current invocation
DocumentsGenerated	0..n	Documents generated as result of the invocation

on CRUD[1] systems. Invoking these commands on resource within the system, all required operations should be possible.

On the light of these ideas, it is clear that HTTP are the perfect example for RESTful system. It is possible to use a limited set of commands (addressed in the HTTP1.0[16] or HTTP1.1[17] specification) on any URL provided by the software agent in use. Anyhow, RESTful systems do not have to be HTTP or even based on HTTP.

The purpose for a RESTful system is to manage all resources in the system by means of this set of limited and well-know *verbs*. Reader can compare this feature with RPC-based system where the set of operation is freely upgradable and parameters attached does not identify the resource but the properties to be changed.

The change of the status for resources in the RESTful approach is driven by these verbs for actions and never stored on the server that will not have memory about former invocations. This feature drives the design of the servers and makes possible a higher scalability and larger simplicity for these solutions.

In our case, as mentioned before, we will take full advantage of semantics. Thus, instead of providing links to "simple" resources, semantically defined items will be used in the frame of this contribution. Thus, the actual content to be managed by the system belongs to an ontology. This feature enable the system to provide with additional features not present in non-semantic environments such as reasoning about the outcome of an operation, fully-autonomous discovery of next iterations, or smart searches for additional services.

6 Providing the solution

As mentioned in previous section, the use of a RESTful approach poses some restrains about how operations can be deployed and which kind of information can be used. From Section 4, a characterization of resources, i.e., ASs and iASs, is provided. Besides, these resources, as they are defined in terms of an ontological support, can be addressed by means of a link. This is due to the fact that all instances in a ontology are provided with an identifier under the form of a URL.

In order to fully characterize the architecture itself of the system, it must be identified how interactions are conducted and which operations are possible. These operations will be mapped into *verbs*, using a RESTful terminology, and used with the corresponding resources. In the case of this proposal, we will stick with the HTTP protocol and take advantage of the provided commands. These operations will used with a concrete meaning (check Table 2). Upon each invocation of the former commands, the server will respond with information expressed in the terms of the ontology included in the system, i.e., a rdf-like file will be generated and sent back to the client. Thus, the agent receiving this response will be able to understand the knowledge implicit in this response and, if the case, take the correspondent decision.

Under these assumptions, the behavior of a citizen invoking a service is shown on Fig. 2.

- On first step, the citizen, aware of the AS he is willing to invoke, launch his/her request. As a result, the server will return de iAS individual, defined according to the ontology. This new resource in the system is identified by a unique URI that will used in future interactions. Let us say, that the citizen want to just request a license for building a house. In this case, the request will be something like this:

  ```
  GET /ASAvailable/GetLicense HTTP/1.1
  Host: pa1.government.uvigo.es
  Accept: application/rdf+xml
  ```

  After the corresponding operations in the back office of the PA, it will be generated the response, i.e., the corresponding iAS. This instance of the ontology will be sent back to the citizen and used in further interactions.

- On phase 2, once the operation is launched, the citizen can add some additional documentation to this request. In our example, the citizen will, for instance, add some relevant document such the plans of the building or the receipt of the correspondent taxes.

[1]Create, Read, Update and Delete

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table 2: Verbs used and meaning

HTTP command	Resource	Meaning
GET	AS	Invoke the execution a particular AS and return a iAS
POST	iAS	Recover the result of an operation from a PA
PUT	iAS	Add a document to the iAS mentioned
DELETE	iAS	Cancel the invocation of a iAS

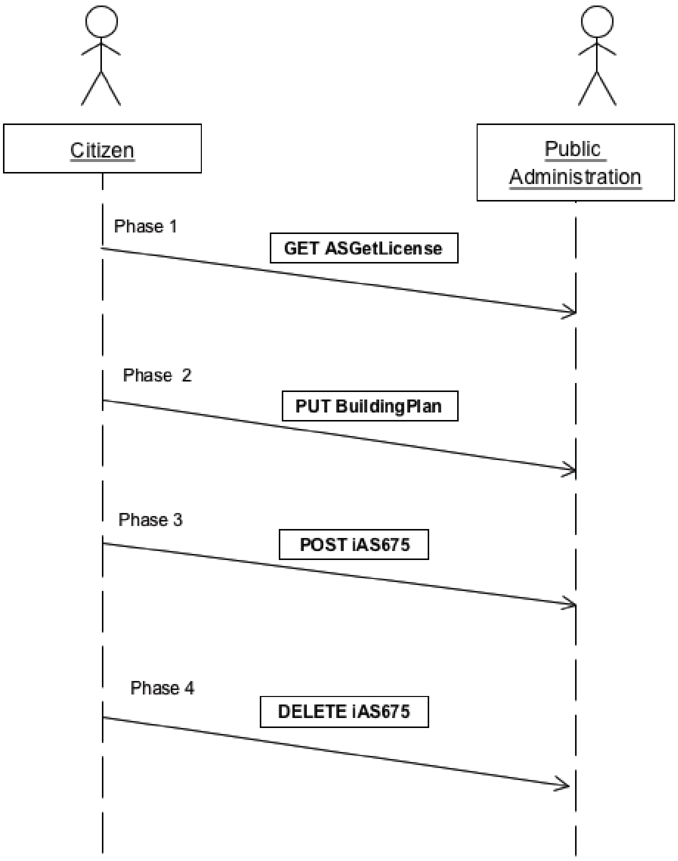

Figure 2: Diagram of sequence

- At any time the citizen can check the status of the requested operation. As shown on the third phase, the citizen can make use of the verb POST to check the current stage of the operation and gather all the information that may be of his interest.
- Eventually, as shown on phase 4, the citizen can cancel the invoked operation making use of the command delete.

The use of a RESTful system is possible mainly due the business model designed. As all the operations possible are defined in terms of ASs and iASs it is not required to negotiate about the interface of the service or decide and discover how a particular WS must be addressed. As requested by the RESTful approach, the state of the operations invoked does not need to be stored and also the use of cookies is not need, a hardly acceptable feature in RESTful systems. In order to simplify the system as much as possible, all the operations are designed to be idempotent.

These operations are actually a front-end for services in the domain of eGov and further operations inside the PAs should be performed to actually complete de service. Also, note that with this brief set of services it is possible to conduct complex operations as the definition of the service is already provided by the knowledge inserted in the AS itself.

7 Conclusion

In the current state of the art, it is required for governmental institutions to face long-term projects aimed to fulfill needs of the citizens from a holistic perspective. PAs must tackle these solutions taking into account a large number of considerations in a complex and interoperable scenario. And this is the context for projects mentioned in previous sections and this work itself.

The work presented is aimed to contribute in this area to grant a support for solutions focussing on features such as scalability, simplicity in deployment and reuse in different frames. To make that possible, two main tools or technological supports are used: semantics and the RESTful paradigm. Semantics is used as formal tool to characterize the environment. From its use, resources addressed within the proposal are clearly identified, including all related information. The use of the RESTful architectural style simplifies both the design of the platform and its development. The combination of both provides us with quite flexible and powerful environment to provide with solutions in the domain.

The model presented in this paper makes little assumptions about the technological support and final details for implementation can be adapted to different scenarios of application. Therefore, further implementations and new developers can provide with new versions according to additional circumstances including mobile environments, web-based solutions or special legal constrains.

Also, in the development of this solution, a large ontological model was considered and a reduction of it was considered for this case. Later adoptions of this model, can include more concepts semantically described and applied to the RESTful model of the system presented.

Anyhow, developers of this solution must borne in mind that this approach is not similar to those based on Semantic Web Services[18]. In this context is not possible to discover on runtime the interfaces for services or to compose services using techniques such as is done in WSMO[19] environments. Therefore, all these operations must be performed in programming time.

Acknowledgment

We want to thank Ministerio de Educación y Ciencia for their partial support to this work under grants “Servicios adaptativos para e-learning basados en estándares” (TIN2007-68125-C02-02) and Xunta de Galicia, Consellería de Innovación e Industria through project “SEGREL: Semántica para un eGov Reutilizable en Entornos Locais” (08SIN006322PR).

References

[1] World Bank, “About e-government,” Disponible en la web, 2007, http://www.worldbank.org/egov.

[2] D. Enterprise and I. I. Unit, “European interoperability framework for pan-european egovernment services,” Web available, 2005, http://europa.eu.int/idabc/en/document/2319/5644.

[3] “The SemanticGov Project,” Web available, 2006, http://www.semantic-gov.org.

[4] “Alial,” Web available, 2009, http://www.asolif.org/?page=firma_convenio_alial.

[5] “egov: Online one-stop government,” Web available, 2005, http://www.egov-project.org/.

[6] M. Wimmer, “Implementing a knowledge portal for egovernment based on semantic modelling: the e-government intelligent portal (eip.at),” in *Proceedings of the 39th Hawaii International Conference on System Sciences*, 2006.

[7] “Archiving Web Resources: A Policy for Keeping Records of Web-based Activity in the Commonwealth Government,” Web available, http://www.naa.gov.au/recordkeeping/er/web_records/intro.html, 2009.

[8] “State Portal of the Republic of Slovenia,” Web available, http://e-uprava.gov.si/e-uprava/en/portal.euprava, 2009.

[9] Irish Government, “Ipsms,” Web available, 2005, http://www.gov.ie/webstandards/metastandards/index.html.

[10] suomi.fi editorial team, “suomi.fi,” Web available, http://www.suomi.fi/suomi, 2007.

[11] L. Álvarez Sabucedo, L. A. Rifón, R. M. Pérez, and J. M. S. Gago, “Providing standard-oriented data models and interfaces to egovernment services. a semantic-driven approach,” *Computer Standards & Interfaces*, 2008.

[12] T. Berners-Lee, J. Hendler, and O. Lassila, “The semantic web- a new form of web content that is meaningful to computers will unleash a revolution of new possibilities,” Web available, 2005, http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21\&catID=2.

[13] W3C, “Web ontology language,” Web available, 2007, http://www.w3.org/2004/OWL/.

[14] Fernández-López, M., Gómez-Pérez, A, and Juristo, N., “Methontology: From ontological art towards ontological engineering.” *Symposium on Ontological Art Towards Ontological Engineering of AAAI.*, pp. 33–40, 1997.

[15] R. T. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” Irvine, 2000.

[16] IETF, “Hypertext Transfer Protocol – HTTP/1.0,” Web available, 2009, http://www.ietf.org/rfc/rfc1945.txt.

[17] Network Working Group (IETF), “Hypertext Transfer Protocol – HTTP/1.1,” Web available, 2009, http://www.ietf.org/rfc/rfc2616.txt.

[18] “Semantic Web Services Interest Group,” Web available, 2007, http://www.w3.org/2002/ws/swsig/.

[19] SDK WSMO working group, “Wsmo,” Web available, 2005, http://www.wsmo.org/TR/d2/v1.1/.