SEMANTIC LIFE EVENT ENVIRONMENT FOR EGOVERNMENT SERVICES

Towards a Holistic Support to Host eGovernment Services in a Scalar Manner

Luis Álvarez Sabucedo and Luis Anido Rifón
Depto. of Telematics, University of Vigo, Spain

Keywords: Interoperability, semantics, eGovernment, reference architecture.

Abstract: The huge amount of resources that are being devoted to eGovernment projects makes possible the soaring of new solutions. This onrush of contributions leads us to some shortcomings such as lack of interoperability or poor scalability. To overcome this situation, the use of Life Events is proposed. Introduced in the paper, Life Events are the cornerstone for a semantic proposal to provide eGovernmentservices in an organized manner. A methodology for the use of Life Events and, also, the modelling of the entire system are presented.

1 INTRODUCTION

In the present moment, we are being witnesses of a huge effort in eGovernment development. As ICTs (Information and Communication Technologies) are getting a broad use, Public Administrations (here after PAs) are getting into this tendency. But it is important to bear in mind that eGovernment is not solely a simple replacement of technology to provide a 24/7 service. Indeed, provision of eGovernment solutions involves a huge effort in reengineering all processes involved in public services. As a matter of fact, this technology forces PAs to orient services toward a better service by putting the citizen in the centre of all provided operations.

A drawback of this trend is that interoperability is reduced as a result of the data based, independent approach used to fulfil solutions. In order to overcome this situation, this paper presents a platform where eGovernment services can be provided using a semantic approach based on Life Events. Life Events, as discussed later on, are those events that play a role a in the normal life of citizen and stimulate the needs that compel citizens to interact with the administration.

The proposal presented in this paper takes care of the provision of intelligent framework that using a holistic approach can perform operations in an open and interoperable platform. This one will be developed using a semantic support to provide an integral approach where solutions from different PAs may be integrated.

The rest of the paper is organized as follows. Firstly, we will present the eGovernment state of the art. Secondly, we introduce the problem we are dealing with. Later on, major technological features we are going to deal with are shown: semantics and Semantic SOA, the architectural style used in the system. The next section introduces the concept of Life Event as it is going to be considered in the frame of our proposal. Later on, the proposal for a software design and the methodology to address the provision of the system is provided. Finally, some conclusions are yielded.

2 STATE OF THE ART

A large amount of national projects oriented toward the provision of eGovernment services is available. In the scope of the different countries, several initiatives must be underlined: SAGA (Standards und Architekturen in eGovernment Anwendungen)(KBSt, 2005) in Germany, e-GIF (eGovernment Interoperability Framework)(UK GovTalk, 2004) in United Kingdom, ADEA (l'Agence pour le dèveloppement de l'administration électronique)(French Gov-

Álvarez Sabucedo L. and Anido Rifón L. (2007).
SEMANTIC LIFE EVENT ENVIRONMENT FOR EGOVERNMENT SERVICES - Towards a Holistic Support to Host eGovernment Services in a Scalar Manner.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Society, e-Business and e-Government / e-Learning*, pages 163-168
DOI: 10.5220/0001283701630168

ernment, 2004) in France, EIF (European Interoperability Framework)(Enterprise and Unit, 2005) for European solutions, FEAF (Federal Enterprise Architecture Framework)(POPKIN Software, 2004) in USA,...

There also a large amount of projects from non national entities: OntoGov(OntoGov, 2007), Terregov(TERREGOV, 2005), EPRI(Epr, 2005), EU-Publi(Mariangela Contenti and Baldoni, 2004),QUALEG(QUALEG, 2005), SmartGov(SmartGov, 2005), eGovernment Good Practice Framework, FASME, eGoia,...

3 THE PROBLEM

In order to overcome this drawback, a new formulation for the problem is proposed. As, actually, all interactions between the citizen are driven by the exercise of right or the fulfilling of a demand from the administration, we propose the expression of services provided of the administration in those terms. Thus, it is possible to focus on what the citizen is requesting and not on the PA itself as it happens many times.

To undertake this approach, instead of developing services in a data layer from use cases expressed in natural terms, the provision of a semantic description of services is proposed in a scalar and reusable manner. To model that idea, we define the so-called *Life Event*(here after LE). This one is the concept used to refer to any particular situation a citizen must deal with and requires assistance, support or license from a PA. We can considered as “life events” situations such getting certifications, paying a fine, getting married, moving,...

As LEs have being modelled in the semantic layer we are dealing the problem in higher layer of abstraction. This feature allows us to deal with the problem in a more scalable way as we can use technologies that provide us interesting features in that area. Besides, it is possible to develop additional services to provide added value services: advanced data mining, automatic service composition,...

4 SEMANTIC WEB

Speaking about semantic data, we are addressing the problem of dealing with not just data but information. The aim for this discipline is the provision of information understandable by machines. In the literature, several definitions or approximations to the concept of ontology are provided. A quite suitable definition for ontology may be(Gruber, 1993): “ An ontology is a formal, explicit specification of a shared conceptualization of a domain of interest. ”

By means of this definition we are addressing an ontology as a way to put in a concrete way abstract information about a certain domain by means of machine-understandable data format.

We would also like to outline a key aspect of this technology: the agreement. In fact, it does not matter too much how good or rich is your definition of the medium, but how general is the agreement you can get around your proposal. To be able to work as intended, we need a general agreement among our ontology.

To undertake the provision of an ontology different languages(Gómez-Pérez et al., 2003) are possible. OWL (Ontology Web Language)(Web Ontology Working Group, 2004) is the W3C Recommendation that covers most of DAML+OIL and it is intended to provide a fully functional way to express ontologies. This technology is the chosen one for our proposal. By using OWL, we are addressing a standard, solid and interoperable platform for the provision of this solution.

A different problem is the provision of a semantic description of services. In this case, a solid framework accepted for a general use is not available for the community. We can outline several alternatives: OWL-S (Ontology Web Language - Services)(OWL-S Coalition, 2005), WSMO (Web Service Modeling Object)(SDK WSMO working group, 2005), WSDL-S (Web Service Description Language - Semantic)(Akkiraju et al., 2007), ...

5 SEMANTIC SERVICE ORIENTED ARCHITECTURE

At first glance, we can consider Service Oriented Architecture, here after SOA, as an approach/methodology to the problem of providing an Architecture Model. The gist of this proposal lays on the definition and modelling of Services themselves. This philosophy leads us to not pay attention to other details such as the network or the format for the messages. From this point of view, a service is a software resource accessible and defined by means of some key concepts: advertisement, service, contract, data model and policy. All of them together define and model the service in a holistic manner.

This approach will provide interesting features such as loosely coupled, platform independent and protocol independent. Nowadays, the most common support for system modelled under this style is Web Services. However, any other middleware can be used instead of the former.

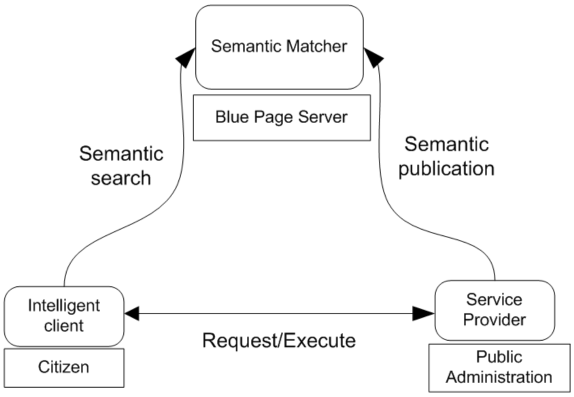

Figure 1: SSOA Services.

Besides, we are prone to use Semantic Service Oriented Architecture, here after, SSOA. So, SSOA, in our proposal, is considered as an architectural style, see (Bass et al., 2003), that uses advantages of semantic applied to the concepts shown for SOA. In this context, a semantic version of the usual elements of the SOA architecture is used, see Figure 1.

6 OUR PROPOSAL

Taking into account the former considerations, a semantic based definition for LE is established. These elements are going to play a main role in our case and they are expressed using semantic terms shared by the whole system. The definition of a LE includes the following items:

Task Title for the considered operation. Folksonomies can play an interesting role as they provide support for semi-automatic enhancements of discovering services.

Description High level description of the desired operation expressed in natural terms from the point of view of the citizen.

Input documents As previously stated, all operations carried out by the administration require some input document. As minimum, citizen is requested to provide a signed form in order to invoke the operation. This element plays a role similar to *preconditions* in some environments.

Scope We must identify the scope of the operation (local, national, international,...) where it must to be recognized.

Output document Of course, as result of any performed operation, the PA in charge must provide an output expressed in terms of the ontology. This information will be put together into one or several documents. This output will vary its content from the expected document (i.e., a certification, a license,...) to information about the failure to get the expected document.

Security Conditions This is intended to express the conditions for the security mechanism involved during the whole process. This includes the identification of both parties, citizen and PAs, and also the way is stored by any agent involved that could be able to use it.

Cost This will express the amount you have to pay for the requested operation and/or also the time it will take for the completion of the operation.

These elements will be defined using the power of semantic expression that will allow us to provide advanced services for discovering and orchestrating them. Life Events can also be tagged using well-know metadata standard already proposed and endorsed by relevant organizations.

So, we propose the transformation of final services as they are requested into new LEs expressed in terms of the semantic definition using the former items presented.

6.1 Methodology

In order to transform natural or normal services to LE expressed in the proposed terms, a simple methodology is yielded:

1. Identify the particular problem you are dealing with in terms of features and PAs involved.
2. Decompose the problem into several different problems that may be resolved in a single step, i.e., each step must produce as output a document meaningful for the citizen.
3. For each procedure identified in previous step, look for the input documents, scope and cost. These ones must be expressed in terms of the ontology of the system. If required, the ontology will be expanded.
4. Identify internal partial steps that the citizen could be interested in. These steps usually involve internal documents that may carry no meaning for the citizen but are relevant for the administration.
5. Identify all possible documents created as possible final steps of the operation.
6. Update all services and agents that may be aware of the new service.

7 DEVELOPING THE SOFTWARE

The whole design of the system will be driven by LEs as they were presented in former sections. Through them, all components will be sketched and the system built up. In the process of defining a software architecture a solid methodology must be undertaken. In our case, we will take advantage of: Unified Modelling Process (UMP)(I. Jacobson et al., 1999) and Bass contributions(Bass et al., 2003); Model Driven Architecture (MDA) (OMG, 2007) and Semantic Service Oriented Architecture (SSOA).

Eventually, a solid methodology is derived based on the phases of MDA and using the interactive proposed by the UMP to design a SSOA-based system. Also, UML(OMG, 2005) is used to provide a formal description of the outputs/inputs of each phase.

The process of designing the system is based on the definition of a Reference Model that will drive the entire proposal. This Reference Model, obtained from the domain experience, will be transformed into a Reference Architecture by mean of the UMP. The former is made of use cases that include the requirements for the system. Applying an individual study to each use case, a Reference Architecture, using the Bass terminology, can be derived. This fits with the idea of PIM (Platform Independent Model) from the MDA terminology and the Model of Analysis in the UMP. Final implementations in a particular software support, also known as PSM (Platform Specific Model) in MDA would turn out to be the functional solution for real world according to the proposed methodology. Similar schemes have also been tested successfully in other fields(Luis Anido et al., 2002).

7.1 Model of Reference

Model of Reference involves the definition of several very high level features and properties that establish the main lines of the project. One of the most important features in that sense is the definition of agents. In our case, the following ones are included:

Client Agent. This agent acts on the behalf of the citizen that is actually requesting an operation. A wide range of agents may be involved in the interchange of information. Agents will be responsible for tasks related to composition and orchestration. This element fits in the profile of the component intelligent client in Figure 1.

Service providers. These elements are the final responsible for the service provision. This role is assigned to PAs. This component plays mainly the role of the Service Provider in Figure 1.

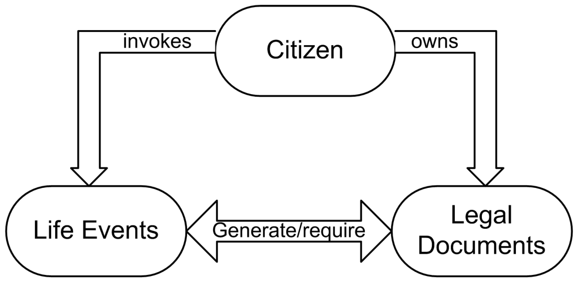

Figure 4: Model of relations in the system.

Blue Page Server. In this architecture an element to solve the problem of locating the server and the proper service is mandatory. This component plays the role of the Semantic Matcher in Figure 1.

Security agent. A third part agent responsible for auditing and supervising operations that may require additional security considerations.

Note that the formers perfectly fit in the SSOA philosophy. The review of the requirement from the citizen and the administration leads us to the identification of up to 10 general uses cases as shown on Fig 2 and 3 that also are part of the Model of Reference.

The system will evolve and provide support to operations according to the schema shown on Fig. 4. At last, the final aim of the citizen is to achieve the document suitable in each case as documents are the legal prove of a status or the fulfilling of an operation when dealing with Public Administrations.

7.2 Defining a PIM

In the context of our methodology, PIM is defined mainly in terms of an Analysis Model. The former is the result of applying to each use case a process to generate the UML design for Analysis Cases. Analysis Cases are defined in terms of classifiers classes (boundary, control and entities classes) and grouped in packages that work cooperatively to achieve a major goal, a use case.

This leads us to the identification of several UML diagrams containing a more in detail description of the problem. One of them is included to illustrate how this methodology works, see Fig. 5. Actually, each Use Cases generates its own diagram. All of them together sketch the final result in an independent way of the chosen software architecture. The entities identified in this process (boundary, control and entities classes) can be reused for several initial cases and lead to a simpler implementation of the platform. Actually, each entity can be assigned to an indepen-

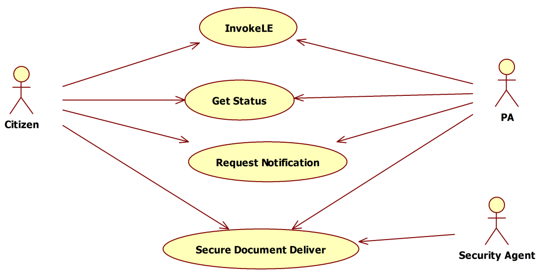

Figure 2: Use case related to the citizen.

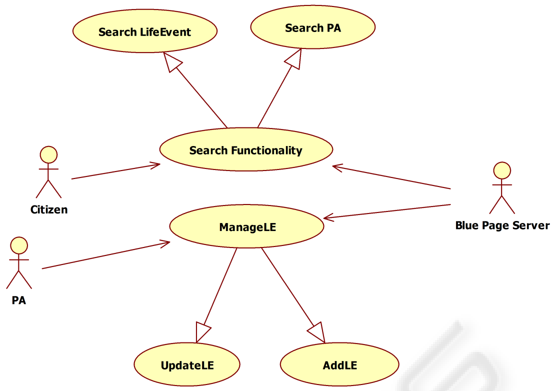

Figure 3: Use Cases related to the blue page server.

dent group of experts for its implementation in further steps.

In our case, up to 25 different classes were identified to perform in a completed manner the use cases identified in the former phase of our methodology.

7.3 Developing PSMs

To provide a real software solution that can successfully fulfill actual services, the actual software must be developed. In order to achieve this final goal, we use as input the PIM from the previous step and map them into new software components. These new components are the so called PSM: Platform Specific Model. The process to obtain them is the next step according to the MDA methodology. Thus, in case of need to deploy another software platform, the same PIM can be used and applied to other constrains or circumstances.

In the present case, to develop the system, a Java based environment was used. To implement the semantic features of the system, the chosen option was Jena(Hewlett-Packard, 2005) due to the advantages in managing for semantic queries. Services are implemented under Web Services defined by mean of OWL-S files. One of the most complex features to implement due to the current state of art in semantics is the provision of the entity SearchEngine in the use case SearchLifeEvent. In our case, to develop this component we take advantage of ARQ for Jena(Hewlett-Packard, 2006). This is an engine that implements support for SPARQL(W3C, 2006), a protocol to execute semantic query against an ontology. Finally, as result, an implementation for each identified entity in the former step is provided.

8 CONCLUSION

This paper presents an approach to model and fulfill services in the eGovernment domain. The main aim is the presentation of a methodology based on the use of semantics and SOA to drive the design and development of a whole system. The result of the complete process must a system that can provide support in an open, scalar and adaptable manner to solutions in the domain.

This approach fits perfectly on a proposal based on Life Events as main concept where the entire system is based on. The use of Life Events, something not new in the area, has been reviewed in this paper and re-elaborated under a different conception to include semantic meaning and citizen centered view in the context of this solution.

Also, as no proprietary solutions are used and only open technologies are taken into consideration, this environment allows the interchange of data and operation among different agents that may be developed under the provided basis by third parties.

The final status for the proposed architecture is a common and global place where heterogeneous agents may interact using the provided support.

ACKNOWLEDGEMENTS

We want to thank “Ministerio de Educación y Ciencia” for their partial support to this work under grant (TIN2004-08367-C02-01).

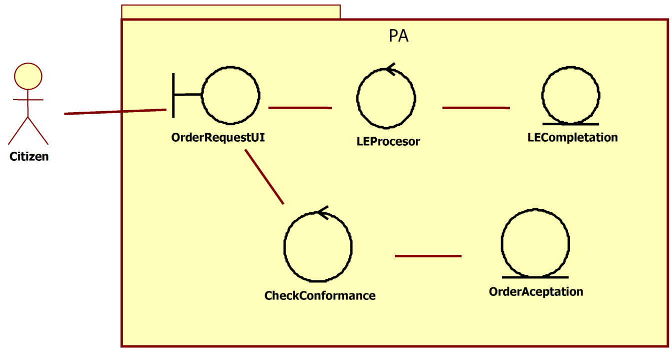

Figure 5: Analysis model for ExecuteLE.

REFERENCES

(2005). Epri. Web available. http://www.epri.org/.

Akkiraju, R. et al. (2007). Web service semantics - wsdl-s. Web available. http://www.w3.org/Submission/WSDL-S/.

Bass, L., Clements, P., and RickKazman (2003). *Software Architecture in Practice*. Addison Wesley.

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

French Government (2004). Adea. Web available. http://www.adae.gouv.fr/adele/.

Gómez-Pérez, A., Fernández-López, M., and Corcho, O. (2003). *Ontological Engineering*. Springer.

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, pages 199–220.

Hewlett-Packard (2005). Jena. http://www.hpl.hp.com/semweb/.

Hewlett-Packard (2006). ARQ for Jena. Web available. http://jena.sourceforge.net/ARQ/.

I. Jacobson, G. Book, and J. Rumbaugh (1999). *The Unified Software Development Process*.

KBSt (2005). Saga. Web available. http://www.kbst.bund.de/-,182/SAGA.htm.

Luis Anido, Juan Santos, Judith Rodríguez, and Manuel Caeiro (2002). An MDA-based model for vertical application task forces. An experience Report. *Procededings of the Sixth International Enterprise Distributed Object Computing Conference'02*.

Mariangela Contenti, Alessandro Termini, M. M. and Baldoni, R. (2004). Supporting inter-administratin cooperation: the eu-publi.com approach. *EGOV*, pages 226–233. LNCS.

OMG (2005). Unified modeling language. Web available. http://www.uml.org/.

OMG (2007). Omg model driven architectures. Web available. http://www.omg.org/mda/.

OntoGov (2007). Ontogov. Web available. http://www.ontogov.com/.

OWL-S Coalition (2005). OWL-S: Semantic Markup for Web Services. Web available. http://www.daml.org/services/owl-s/1.1/.

POPKIN Software (2004). FEAF. Web available. http://government.popkin.com/frameworks/feaf.htm.

QUALEG (2005). Qualeg. Web available. http://www.qualeg.eupm.net/.

SDK WSMO working group (2005). Wsmo. Web available. http://www.wsmo.org/TR/d2/v1.1/.

SmartGov (2005). Smartgov. Web available. http://www.smartgov-project.org/.

TERREGOV (2005). Terregov. Web available. http://www.terregov.eupm.net/my_spip/index.php.

UK GovTalk (2004). e-GIF. Web available. http://www.govtalk.gov.uk/.

W3C (2006). SPARQL Query Language for RDF. Web available. http://www.w3.org/TR/rdf-sparql-query/.

Web Ontology Working Group (2004). Web ontology language. Web available. http://www.w3.org/2004/OWL/.