Conceptualisation and Semantic Annotation of eGovernment Services in WSMO

Karol Furdik[1], Ján Hreňo[2], Tomas Sabol[2]

[1]InterSoft, a.s., Florianska 19, 040 01 Kosice, Slovakia
karol.furdik@intersoft.sk
[2]Technical University of Kosice, Faculty of Economics,
Letna 9, 042 00 Kosice, Slovakia
{jan.hreno, tomas.sabol}@tuke.sk

Abstract. The paper presents an approach to semantic annotation of eGovernment services by means of ontologies using WSMO conceptual model, as it was applied in the FP6 IST Access-eGov project. The user-oriented life event approach was used to adapt the WSMO conceptualisation for modelling and integration of both electronic and traditional governmental services on a semantic level. Ontologies were created and formalised by reusing available knowledge resources as well as by systematic collection of the requirements formulated by public administrations and implemented in WSML notation by system developers. Finally, the Annotation tool is described as a client application for semantic annotation and publishing of the governmental services.

1 Introduction

In the field of eGovernment, the *interoperability* of existing governmental services seems to be a key issue nowadays. In particular, interoperability was recognised as a precondition for the implementation of European eGovernment services in the eEurope Action Plan [4] and is explicitly addressed as one of the four main challenges in the new EU strategy "i2010 - A European Information Society for growth and employment" [5]. It is recommended to build the solutions supporting interoperability of governmental services on standards, open specifications and open interfaces [6].

Semantic technologies are one of the most promising and most frequently used approaches to achieve the interoperability of eGovernment services. Availability of formal, i.e. machine-readable, description of meaning and context of the services, without necessity to modify the services themselves, belongs to the main advantages of this approach.

The "semantic web" paradigm since its beginning has been considered as a promising vision also for eGovernment. Indeed, the eGovernment is an important research and application domain for Semantic Web technologies for many years, as it can be illustrated by a number of EU R&D projects (e.g. SemanticGov, TerreGov, OntoGov, SmartGov, eGOV, EU-PUBLI.com). However, practical outcomes and effects of this

© Václav Snášel (Ed.): Znalosti 2008, pp. 66–77, ISBN 978-80-227-2827-0.
FIIT STU Bratislava, Ústav informatiky a softvérového inžinierstva, 2008.

intensive research are still somehow behind expectations. One of the reasons may be a lack of methodology, tools, and guidelines describing how to design, develop, implement, and employ semantic eGovernment services (i.e. electronic governmental services enhanced by semantic annotations and/or mark-up) [7]. In other words, complexity of the semantic modelling and annotation process needs to be dealt with a user-oriented framework that is easy to use and understandable for public administrations.

An important issue that affects the representation of governmental services, their semantic descriptions, and consequently the whole annotation process is a selection of proper semantic technology. Among a variety of available frameworks, the most popular are RDF-S, WSDL-S, OWL-S, and WSMO technologies.

In this paper, we will focus on the WSMO (Web Service Modelling Ontology) framework [9], applied to semantic description of governmental services within the IST FP6 project Access-eGov [1], [2], [10]. Next section presents an overview of WSMO conceptual model and its adaptation within the Access-eGov project. Section 3 describes knowledge-driven and requirement-driven approaches to the process of semantic annotation and ontology development. User roles and activities are specified by means of the use-case diagram of the semantic annotation. Process of creation and formalisation of ontologies from the existing knowledge resources as well as from requirements systematically formulated by service providers is presented in sections 3.2 and 3.3. Section 4 gives an overview of design principles and features of the Annotation tool developed within the Access-eGov project.

2 Conceptual Model for eGovernment Services

Interoperability of governmental services on the semantic level requires a specification of common semantic basis, i.e. a *conceptual model*. Such model contains a set of relevant entities - concepts, relations, properties, constraints, etc. that can serve as building blocks for annotation, i.e. the formal representation of potentially very complex governmental services and their relationships. The conceptual model is significantly determined by a technology applied for the semantic annotation.

In the Access-eGov project, we decided to apply the WSMO as a basic conceptual framework and implementation platform. This decision was made after detailed survey and analysis of existing approaches, investigating already mentioned RDF-S, WSDL-S, and OWL-S ontologies for semantic description of web services, as well as BPEL4WS for modelling the web services in a business process interaction [1]. The main reason for selecting the WSMO as the most suitable candidate was that WSMO, as technically more advanced alternative, provides a consistent conceptual model for semantic description of web services, with the inclusion of mediators and the distinction between goals and capabilities. In addition, the WSMO conceptual model fits best the proposed architecture and functionality of Access-eGov system [1], [10].

WSMO package also offers the WSMX execution environment that enables discovery, selection, mediation, and invocation of semantic web services. WSMX is based on the conceptual model provided by WSMO, so it serves also as a reference

implementation of it. The WSML language, i.e. the internal formal language of WSMX, provides means to formal description of all the elements defined in WSMO that may be used for semantic description of services in an eGovernment application. Internal data representation of WSMO elements can then be obtained through parsing the WSML descriptions into the WSMO4j data object.

The WSMO conceptual model [9] provides the *Ontologies, Web services, Goals*, and *Mediators* as the top-level elements for semantic description of general web services. For modelling the governmental services, the conceptual model of WSMO needs to be enhanced according to the specific characteristics that can be identified for governmental services [12]. In the Access-eGov, we distinguish *traditional* (non-electronic, face-to-face) services, mostly based on paper forms, and *electronic* services (available directly via web service interfaces or web forms)[1]. Semantic interoperability of these services can be achieved by annotation and consequent composition of the services to the coherent units that are understandable for service consumers - citizens and businesses.

A progressive method for modelling governmental services is the *life event approach* [3], where the life event concept plays a central role[2], being a formal representation of user's point of view, his/her needs and requirements. Based on the life event approach, the WSMO conceptual model was reused and adapted for eGovernment application [1]. The top-level elements were modified as follows:

- *Life Events* were added as formal models of users' needs, consisting from multiple goals and services organised to the *generic scenario* and expressed by the orchestration construction consisting from workflow, control-flow, and data-flow sequences.
- *Goals* were reused from the WSMO model, as it was specified in [10]. They are organised in hierarchical manner, consisting from multiple sub-goals. Besides the specification of outputs and effects, the goals contain a capability statement that is matched against a service's capability.
- *Services* were added as generalisations of the *Web service* concepts. This solution enables to describe both electronic and traditional governmental services by means of a *service profile*, containing functional and non-functional properties, capabilities, and interfaces. In the case there is no executable service available for traditional service, the textual description of required inputs (e.g. some documents, forms, etc.) and requested actions (e.g. to visit particular office, etc.) is specified as the non-functional property.
- *Mediator* and *Ontology* elements were reused from the WSMO model without changes. System ontologies, that provide a set of generic concepts used by other elements of the conceptual model, were designed as implementations of the Ontology element. Life events, service profiles, and domain ontology types

[1] In practice we often find *hybrid* services, e.g. a form can be downloaded or an application may be filed, but the citizen still has to go to an office to complete the procedure [7].

[2] Contrary to the life events, a service-oriented approach was adopted in several older eGovernment solutions, where the service provided by PAs was used as a central concept. For example, in the eGov and OntoGov projects, the user's point of view is represented by a taxonomy that determines a distribution of the services in the web portal [3].

were defined in [10] to describe the concepts that are specific for eGovernment domain and/or required by particular application of the system.

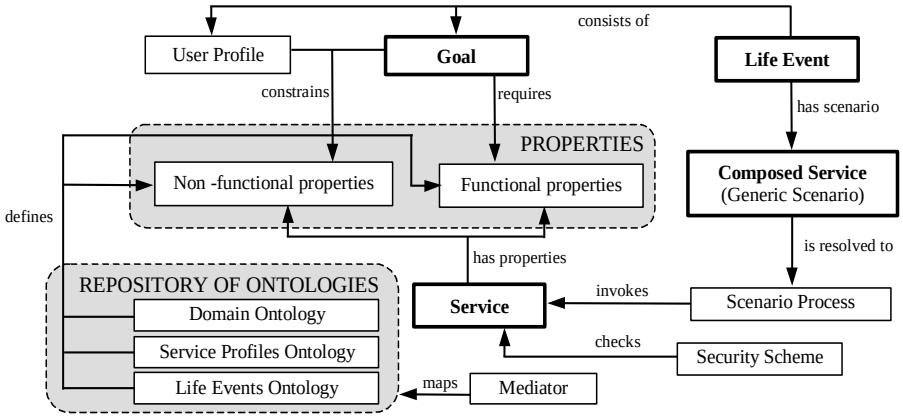

Fig. 1. Conceptual model adapted for eGovernment application

Structural relations between the elements in the proposed conceptual model are depicted in Figure 1. The WSMX interfaces were reused in the implementation by mapping the Access-eGov system components for discovery, reasoning, selection, invoking, mediation and execution of modelled services on the WSMX components [10]. The semantic descriptions of particular services, goals, and life events were created in cooperation with user partners, as it is described in the following section.

3 Approaches to the Semantic Annotation

The conceptual model provides a background framework for describing the governmental services on a semantic level. However, consequent semantic annotation, i.e. actual specification of a particular service, its functional and non-functional properties as preconditions, inputs and outputs, mutual dependencies between services and design of workflow sequences should be provided by particular PAs. It is obvious that the process of semantic annotation is very complex and requires proper *methodology* as well as *annotation tools* that are easy to use for PA employees, without any expert-level knowledge of semantic technologies.

3.1 Use-case Diagram of Semantic Annotation

Proposed use-case diagram for the overall process of semantic annotation [11], as presented in Figure 2, introduces on the side of PA the role of Annotator and specifies four activities for this role:

- *Introduction and annotation of governmental services* (both electronic and traditional ones) to the eGovernment system requires the semantic description,

selection of process model, and consequent registration of the service in the Access-eGov system.

- *Ontology browsing and management*. To semantically describe a service, the Annotator browses the space of available ontologies. Then, he/she chooses concepts and relations from the selected ontology to mark-up important aspects (non-functional properties) of the currently annotated service.
- *Goals management*. In addition to the services, there is a possibility to create and manage goals and life events. The goals and life events are workflow-like constructs that could be considered as outputs provided by eGovernment system for users (citizens or businesses). The Annotator can define new or modify existing goals by means of semantic description, similarly as it is done for services.
- *Life events management*. The goals and services can be combined into more complex workflow models - life events, which are then exposed to users.

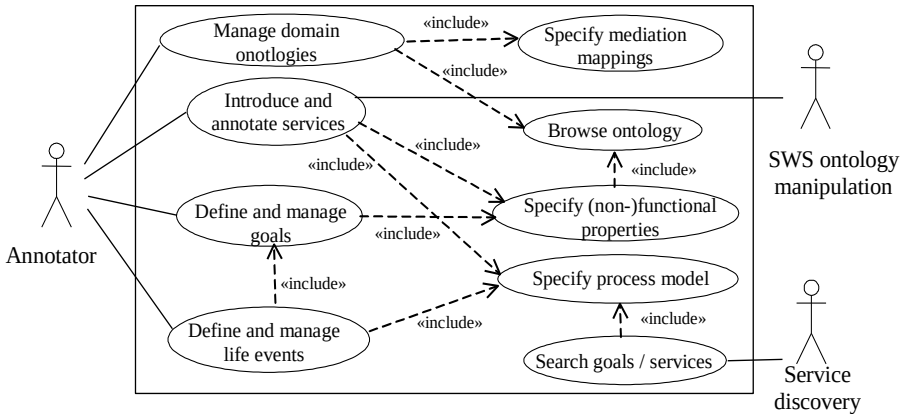

Fig. 2. Use-case diagram for the process of semantic annotation

To accomplish the described process of the semantic annotation of governmental services in practice, it is necessary to solve the following issues:

a) How to create the ontologies that are capable to model the provided services;
b) How to obtain a semantic description of services from PAs;
c) How to verify that the semantic annotation is correct and matches the requirements of both PAs and citizens.

Design and formalisation of ontologies, from existing knowledge resources as well as from requirements systematically formulated by PAs, will be discussed in the next two sections. Concrete application of semantic annotation and verification of results is provided by a specialised annotation tool, which will be presented in section 4.

3.2 Construction of Domain Ontologies

Semantic technologies were used for description of governmental services in significant number of projects and approaches, some of them were already mentioned in

section 1 above. Thus, a natural way how to design an ontology suitable for our purposes is to reuse the existing ontologies, standards, and knowledge resources.

Wide range of available resources was analysed within the Access-eGov project [8] and the following ontologies were chosen as best candidates for reusing:

- WSMO ontologies (wsmo.org) for description of date, time, and location;
- vCard ontology (www.w3.org/2006/vcard/) for addresses and personal data;
- DublinCore (dublincore.org) for metadata and document types;
- Terregov, DIP, DAML, GEA, GOVML, AGLS metadata set, and IPSV ontologies [7], [8] for description of specific eGovernment concepts.

Existing ontology resources were used in the Access-eGov project to produce some fragments of the whole ontology structure, mostly the definitions of non-functional properties for services. The example below presents an implementation of vCard ontology for WSML representation of the ontology concept 'Organization':

```
namespace{ _"http://www.accessegov.org/ontologies/core/",
  dc _"http://purl.org/dc/elements/1.1/",
  v _"http://www.w3.org/2006/vcard/ns#" }
concept Organization
  v#relation ofType Link
  v#organizationName ofType _string
  v#organizationUnit ofType _string
  v#addr ofType (1 1) v#Address
```

The process of modelling and formalisation of a domain from known conceptual descriptions by knowledge engineers is sometimes referenced as the knowledge-driven design approach [7]. The advantage of this approach, especially if the available ontology resources are reused, is that it ensures a standardisation of the output model. However, such model can easily be artificial and useless in practice, since it does not reflect the requirements of its usage. A solution can be to combine the existing ontology resources and modelling capabilities of knowledge engineers with a systematic description of requirements given by users of the ontology model.

3.3 Requirement-driven Approach

The statement that "the design of the semantic structures should systematically follow requirements concerning the use of semantic eGovernment services (in addition to a knowledge-driven design approach)" was defined in [7] as the starting point and the basic underlying assumption for the requirement-driven approach. This approach provides a generic method how to collect, relate, and formally express the information needs of the service providers, i.e. PAs, to design the specific semantic structures and descriptions of provided governmental services.

The proposed method [7], designed by the German University in Cairo (project partner), consists of the following seven steps that should be fulfilled by PAs in cooperation with system developers:

1. *Identification of information needs*, i.e. specification of life events. It requires an analysis of prior knowledge of users (citizens and businesses) and the diversity of informational needs of different user groups. The output is a list of user group's in-

formation needs, created and provided by PAs in a free-text format. System developers will transform it to the list of life events, formally expressed by the WSML notation, and store it to the Life events ontology. Within the Access-eGov project, this step has been accomplished through requirement analysis and user scenarios.

2. *Identification of required information quality*. The information needs from previous step are analysed with respect to the required properties as scope, relevance, etc. A list of proposed services together with related laws and regulations, documents needed to negotiate between users and PA, and other requirements concerning information provision will be provided in a free text format or possibly in the UML notation. Within the Access-eGov project, the required information quality has been identified within the trial descriptions.

3. *Creation of a glossary of topics and terms*. The glossary should contain all relevant topics and terms needed for describing the services identified in step 2. In the Access-eGov project, the table format with columns as {*term* (in English), *term translations* (to other languages used in pilot applications), *description*, *notes*} was used for the glossary.

4. *Creation of controlled vocabulary*. The controlled vocabulary contains a hierarchy of categories and subcategories created from the glossary by grouping the terms into hierarchical subgroups. The hierarchy is given by a generalisation of terms, from most general terms (root nodes) to the most specific ones (leaf nodes). The formal notation of a category includes subcategories, attributes, and a free text description [8], as it is presented in Table 1:

Table 1. Category *Organisational Entity* from the Access-eGov controlled vocabulary.

Category: **Document**	
Subcategories:	Certificate, Form, Notification, Payment Receipt
Attributes:	Title, Description of purpose
Description:	This category is used for concepts that refer to artefacts which have a specific structure and which play a certain role with respect to government services, e.g. *certificates* provided by the administration, *forms* to be filled in by citizens, *notifications* issued by an administration in order to inform a service consumer about certain changes in status, *payment receipts* which a citizen receives after having paid a fee.

In the Access-eGov project, the categories for the controlled vocabulary were specified by grouping the glossary terms and then the category names were included into the table of glossary as a new column. The categories from the controlled vocabulary can then serve as basis for the design of classes in the domain ontologies of the Access-eGov system.

5. *Grouping and relating the identified items*. After step 4, the categories are organised in the controlled vocabulary only by *is_a* relation in the hierarchy. In step 5, a set of other relations and mutual dependencies will be identified between the categories. New categories can also be defined here, if it is needed for the consistence of the whole structure. An ontology-like structure is provided as the output of this step.

Fragment of such structure, identified for the Access-eGov system, is depicted in Figure 3. The grey background identifies so-called boundary concepts [8] that will be annotated as non-functional properties of the services.

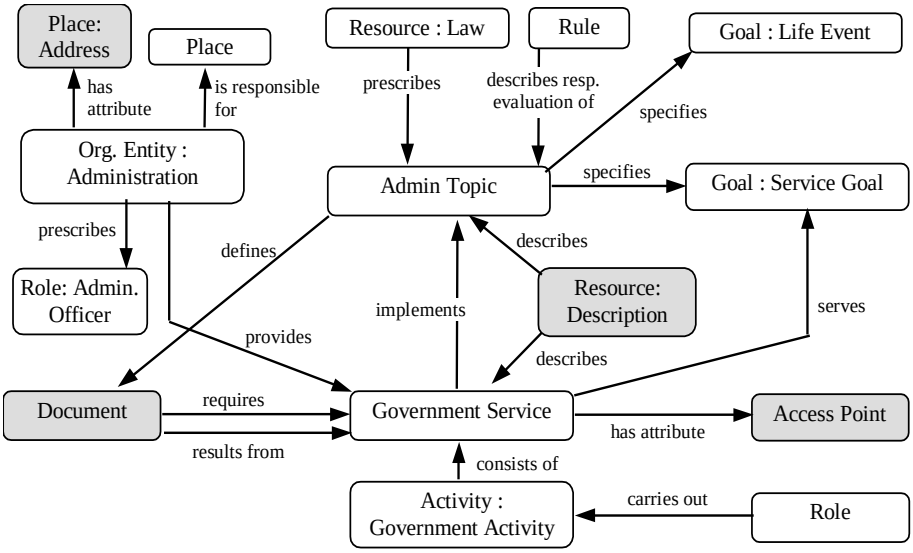

Fig. 3. Fragment of ontology-like structure, as it was identified for the Access-eGov system

6. *Design of ontology*. The ontology-like structure created in the previous step needs to be formalised and expressed by WSML statements. It requires fixing the meaning of the terms and relations defined in the controlled vocabulary, as well as verifying that formal meaning reflects informal description in the glossary. For example, a hierarchy of certificates can be expressed in WSML notation as follows:

```
concept certificate
  subConceptOf document
  concept birth_certificate
    subConceptOf certificate
```

7. *Implementation of semantics*. The formal ontology specified in the step 6 is rather static, consisting of declarative statements that express the concepts, their attributes, and mutual relations. In many cases, the conceptualisation needs to be enriched by "business rules" that can be, for example, conditional if-then-else expressions, loops, and workflow sequences. The WSMO framework provides means to semantically describe the life events, goals, and services in a dynamic manner. The example below presents the WSML formalisation of the life event for marriage (expressed as complex goal):

```
namespace {_"http://www.accessegov.org/ontologies/shg/",
  dc _"http://purl.org/dc/elements/1.1#",
  aeg _"http://www.accessegov.org/ontologies/core/"}
goal MarriageLifeEvent
```

```
nfp dc#title hasValue "Marriage" endnfp
interface MarriageLifeEventInterface
 orchestration
  workflow
    perform n1_1 receive ?x memberOf Q1.
    perform n1_2 achieveGoal ApplyForMarriageGoal
    perform n1_3 achieveGoal WeddingPlaceReservationGoal
    perform n1_4 achieveGoal WeddingCeremonyGoal
  controlFlow
    source n1_1 target n1_2
    source n1_2 target n1_3
    source n1_3 target n1_4
  dataFlow
    source n1_1{?x} target n1_2{?x}
```

A formalised WSML representation of the ontology containing all the definitions (concepts, classes) of services, goals, and life events can be produced as a result of this 7-step procedure. To use this ontology in a real eGovernment application, a tool is needed for creation and maintenance of instances of the services actually provided by PAs in particular applications.

4 Annotation Tool

It is assumed that the annotation will be managed by PA employees with very limited or no knowledge of semantic technologies. So the annotation tool should be easy-to-use and should require only standard PC skills; certainly no WSML statements should be displayed on the screen during the annotation. To fulfil this requirement, all the WSML "business rules" specifying workflow structures in the service processing (i.e. definitions of life events, elementary and complex goals, and functional properties of services) were implemented in advance and stored into the system repository. The service definitions were clustered according to their scenarios (workflow models) and grouped to so-called *service templates*. The service template then specifies a particular *type of a service*, since it contains definitions of functional properties and the workflow structures that should be applied during the processing of the service.

Introduction of service templates simplifies significantly the process of semantic annotation for PA employees. The annotation of services then requires a specification of concrete values for non-functional properties and a selection of a service type from the list of available templates.

The annotation tool was designed in the Access-eGov project as a standard web application, in its *first version* as a client-server application using the JSP and mySql DB technologies. This first version served as a mock-up for user partners of the project and influenced significantly their inputs to the 7-step procedure described above, especially within steps 2 and 3. As advantages of this tool, we can mention the possibility to check and verify the results of annotation immediately after a modification. The results of annotation process are available in the HTML preview (as they will be provided to citizens) as well as in the WSML notation (as they will be provided to other client applications via web service interfaces).

However, the solution based on a relational database had a significant disadvantage: the database structure and related forms on the web interface did not reflect the WSML definitions stored in the ontology. It was especially obvious in the phase of ontology development, when every change of non-functional properties required modifications in both data structure and user interface. So the *second version* of the annotation tool was designed and implemented using the WSMO object model and JSF technology (Figure 4).

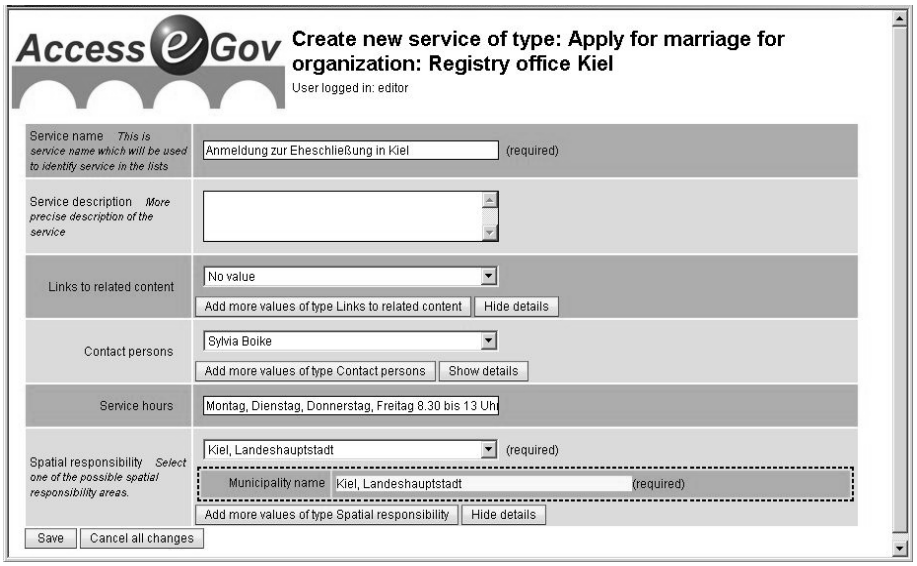

Fig. 4. Annotation tool, 2nd version. Annotation of a service in the German pilot application

Some new features were added to this version, namely:

Simple user access control was added to the annotation tool. Every user of the annotation tool can have four different access rights levels, namely *administrator*, *editor*, *publisher*, and *viewer*. First one is reserved for modifications on the system level, namely adding / deleting users, organisations, templates for services, and organisation types. This level also allows users to modify particular organisation properties as address, list of contact persons, etc. With editor privileges user can add and edit services for particular organisation. Publisher privileges provide means to publish services after editor entered them into the system. Finally, viewer privilege can be granted to users only for previewing of the existing organisation and services information.

Templates mechanism was introduced to ease the maintenance of workflow sequences for the annotated services. Annotators just select a proper type of the service by clicking proper button on screen to assign a particular scenario for this service.

Non-functional properties were updated and fixed for the first trial, according to the feedback obtained from project partners. Enumeration lists for the properties with pre-defined values (e.g. list of towns in a region, etc.) were formalised in WSML and were implemented into the interface. The web interface is dynamically generated according the structure of non-functional properties formalised in WSML. This ap-

proach enables to customise the interface easily, just by proper modifications of the structure of non-functional properties stored in the ontology.

Multilingual support on the data level enables to formulate services and describe their properties in several languages. For pilots of the Access-eGov project, the primary language is the official language of the country (German, Polish, and Slovak, respectively), and the second language is English.

A result of the annotation process is an instance of the service with its properties specified. The following example presents such instance in the WSML notation.

```
namespace {_"http://www.accessegov.org/ontologies/shg/",
  dc _"http://purl.org/dc/elements/1.1#",
  aeg _"http://www.accessegov.org/ontologies/core/"}
instance aeg#RespDept memberOf aeg#Department
     dc#identifier hasValue "RegOfPloenID"
     dc#title hasValue "Register office Ploen"
     aeg#spatial_responsibility hasValue "Ascheberg, Ploen"
     aeg#contact_information" hasValue aeg#RespDeptContact
```

The WSML instances of the annotated services are accessible for various client-side tools via a web service interface. The *Personal Assistant* client [10] is envisioned to be one of such tools, providing browsing, discovery, and execution capabilities of proper services for citizens / businesses according to the specified life event or goal.

5 Conclusions and Future Work

Semantic annotation of governmental services is presented in this paper as an effective method how to achieve the semantic interoperability in heterogeneous environment of public administrations. In the context of the Access-eGov project, the WSMO conceptual model was adapted to provide a basic set of concepts for modelling the governmental services according to the user-oriented life event approach. A process of ontology design and formalisation by reusing available knowledge resources and by collecting information needs from service providers was described as a step-by-step procedure. The developed ontology was then used for semantic description of governmental services by means of the Annotation tool. The annotated services are envisioned to be stored in a repository and are accessible by standard web service interface. Semantic annotation and underlying Access-eGov framework enables to combine the services into complex workflow structures and supports discovery, execution, and mediation of the services. Citizens and businesses can then use various client-side applications or interfaces, for example the Access-eGov Personal Assistant client, to access, browse, retrieve, and actively use the integrated governmental services.

Currently (September 2007), the implementation of the first prototype of the Access-eGov system and its components approaches its final phase. According to the work plan, the prototype should be ready by the end of 2007. The system will be tested on three pilot applications in Germany, Poland, and Slovakia. The trials will be evaluated and implementation of the system will be adapted accordingly in early 2008. More information on the Access-eGov project can be found at www.accessegov.org.

Acknowledgements

The Access-eGov IST project No. FP6-2004-27020 is co-funded by European Commission. The work presented in the paper was also supported by the Slovak Grant Agency of the Ministry of Education and Academy of Science of the Slovak Republic within the 1/4074/07 Project "Methods for annotation, search, creation, and accessing knowledge employing metadata for semantic description of knowledge".

References

1. Bednar, P. et al: D3.1 Access-eGov Platform Architecture. Technical report of the Access-eGov project, FP6-2004-27020. University of Regensburg, Regensburg, Germany (2006)
2. Furdik, K., Sabol, T., Bednar, P.: Framework for integration of e-Government Services on a Semantic Basis. In: Grönlund, A., Scholl, H. J., Wimmer, M. A. (eds.), Electronic Government, 6th International EGOV Conference. Proceedings of ongoing research, project contributions and workshops (September 3-6, 2007, Regensburg, Germany). Trauner Verlag, Linz, Austria (2007), 71-78, ISBN 978-3-85499-255-4.
3. Gugliotta, A., Cabral, L., Domingue, J.: Knowledge Modelling for Integrating E-Government Applications and Semantic Web Services. AAAI Spring Symposium Series "Semantic Web meets E-Government", Stanford University, California, USA (2006)
4. eEurope 2005: An information society for all. COM(2002) 263 final of 28 May 2002, http://europa.eu.int/information_society/eeurope/2005/all_about/action_plan/index_en.htm
5. i2010 - A European Information Society for growth and employment, COM (2005) 229 final of 1 June 2005, http://europa.eu.int/i2010
6. Interoperability for Pan-European eGovernment Services. COM(2006) 45 final, Brussels, 13.2.2006, http://europa.eu.int/idabc/servlets/Doc?id=24117
7. Klischewski, R., Ukena, S.: Designing Semantic e-Government Services Driven by User Requirements. In: Grönlund, A., Scholl, H. J., Wimmer, M. A. (eds.), Electronic Government, 6th International EGOV Conference. Proceedings of ongoing research, project contributions and workshops (September 3-6, 2007, Regensburg, Germany). Trauner Verlag, Linz, Austria (2007), 133-140, ISBN 978-3-85499-255-4.
8. Klischewski, R. et al: D7.1 Public administration resource ontologies. Technical report of the Access-eGov project, FP6-2004-27020. InterSoft, a.s., Kosice, Slovakia (2007)
9. Roman, D., Lausen, H., Keller, U. (eds.): Web Service Modelling Ontology (WSMO). Technical report, WSMO Final Draft, http://www.wsmo.org/TR/d2/v1.2/ (2005)
10. Skokan, M., Bednar, P.: Access-eGov architecture. In: Mikulecký, P., Dvorský, J., Krátký, M. (eds.), Proceedings of the 6th annual conference Znalosti (Knowledge) 2007 (Ostrava, Czech Republic, February 2007). Faculty of Electrical Engineering and Computer Science, VŠB - Technical University of Ostrava, Czech Republic (2007), 384-390, ISBN 978-80-248-1279-3.
11. Tomasek, M. et al: D3.2 Access-eGov Components, Functional Descriptions. Technical report of the Access-eGov project, FP6-2004-27020. InterSoft, a.s., Kosice, Slovakia (2007)
12. Wang, X., Goudos, S., Peristeras, V., Vitvar, T., Mocan, A., Tarabanis, K.: WSMO-PA - Formal Specification of Public Administration Service Model on Semantic Web Service Ontology. Proceedings of the the 40th Hawaii International Conference on System Sciences, IEEE Computer Society, Hawaii, USA (2007)