A WSMO-based Framework Enabling Semantic Interoperability in e-Government Solutions

Karol Furdík*, Martin Tomášek*, Ján Hreňo**

* Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovakia, e-mail: karol.furdik@tuke.sk, martin.tomasek@tuke.sk

** Faculty of Economics, Technical University of Košice, Letná 9, 042 00 Košice, Slovakia, e-mail: jan.hreno@tuke.sk

Abstract: The paper presents a software framework that was designed and implemented within the FP6 IST EU project Access-eGov to integrate governmental services of various types, i.e. on-line (electronic and web services), as well as off-line (i.e. traditional, face-to-face) services by means of enhancing the service description with semantic information. The user-centric approach of life events is employed for presenting complex workflow sequences to the end users. The system architecture, functionality, and structure of underlying ontologies is described together with a mechanism for the orchestration and choreography of semantic web services in WSMO. The framework includes tools for the maintenance of semantically enriched services and for presenting the services to the citizens via customisable web interfaces. The paper concludes with an outline of the results obtained from the testing and evaluation of the implemented Access-eGov platform in real settings within public administrations in Slovakia, Poland, and Germany.

Keywords: semantic interoperability; e-Govenrment; web services; ontologies; WSMO

1 Introduction

Interoperability in the field of information software systems stands for an ability of the seamless interoperation of the possibly heterogeneous services which may be provided and consumed by various independent actors in a networked environment. The increasing demand on interoperable frameworks and solutions in the last five years is invoked by adopting the advancements of service-oriented architectures (SOA) and web services. It is particularly notable in the areas of e-Business [20], e-Health [13], or e-Government [18], where there is pressure on data and information exchange between the services, data resources, and applications distributed among a wide community of stakeholders.

The aspects of interoperability as a general concept or approach cover technical, syntactic, semantic, and organisational issues, usually referenced as interoperability layers [11]. These layers, which are related and mutually interconnected, deal with the following objects:

- *Technical interoperability level*: signals, low-level services and data transfer protocols;
- *Syntactic interoperability level*: data in standardised exchange formats, mostly based on XML or similar formalisms;
- *Semantic interoperability level*: information in various shared knowledge representation structures such as taxonomies, ontologies, or topic maps;
- *Organisational interoperability level*: processes, defined as workflow sequences of tasks, integrated in a service-oriented environment.

The main focus of this paper is on semantic interoperability; however, other levels are addressed as well. The framework presented in the next sections is built on established and widely accepted standards for data transfer and exchange (TCP/IP, XML), web services (WSDL, SA-WSDL) and process models (BPMN, BPEL). The integration platform is based on the WSMO framework (Web Services Modelling Ontology[1]), which provides an environment for the creation and development of underlying semantic knowledge structures - ontologies and semantically annotated web services that may be organised into a dynamic process workflow [17].

The outlined approach was adopted in the IST FP6 project Access-eGov (Access to e-Government Services Employing Semantic Technologies[2]) and will be described in more detail in the next sections of this paper. The consortium of this project consisted of 11 partners from five countries (Slovakia, Poland, Germany, Greece, Egypt) and was coordinated by the Technical University of Košice, which was responsible for the majority of design and implementation works as well. The main objective of the project was to develop a software platform that will be capable of providing support for citizens and businesses in their life event situations and business episodes related to various governmental services. The solution combines the user-centric paradigm of life events (on the side of user interface) with semantically interoperable service-oriented architecture (on the side of back-office) [15].

1 http://www.wsmo.org

2 http://www.access-egov.org

1.1 Related Research

In the field of e-Government, interoperability was recognised as a precondition for the implementation of European e-Government services already in the eEurope Action Plan [4] and was explicitly addressed as one of the four main challenges in the i2010 EU strategy [7]. This is important especially for the integration and co-operation of existing services - employing solutions based on existing standards, open specifications and open interfaces [8], [11].

One of the most promising approaches to interoperability is the employment of semantic technologies [18], [1]. Semantics provides a capability to model and represent knowledge within a domain by means of explicit formalisation of key domain concepts, their attributes and relations, as well as workflow sequences and structures. Considering the heterogeneous and distributed nature of the e-Government domain, semantics can be very effectively used as a common background platform for describing the processes and services provided by governmental institutions on various levels. The common platform then allows for integrating the services, making them interoperable and transparent for the end users, citizens and businesses.

Intensive research in the application of semantics in the e-Government field is going on, mostly focused on the integration of back-offices, employing SOA and web services enriched by a semantic description [5]. This research can be documented, for example, by projects supported by the European Commission within its 6th and 7th Framework Programme. Most of the solutions apply semantic technologies to ease the system design by modelling the citizen's behaviour, to enable or enhance interoperability of services, to provide a platform for creation of semantically described web services, etc. The provision of better and more integrated public services to citizens and businesses can be recognised as a common goal of all the research efforts. In the following paragraphs, we will briefly mention some of the R&D projects which can be considered as examples of existing solutions and approaches.

The *Terregov* project[3] is focused on the semantic requirements of governments at local and regional levels for building flexible and interoperable tools to support the change towards e-Government services. The Terregov solution provides a specialised ontology as well as a platform for enhancing existing government web services with a semantic description. Such semantically enhanced web services can then be detected, accessed, and orchestrated in an interoperable way. However, the Terregov solution only operates on a regional level of administration and, as such, it lacks a more global point of view. In addition, the Terregov solution requires a suite of already existing web services on the side of public administrations. The support for transforming other types of services (such

[3] http://www.terregov.eupm.net

as traditional face-to-face services, or electronic services provided by web forms) into required web services is rather limited.

The *SemanticGov* project[4] is aimed at supporting the provision of pan-European services to resolve semantic incompatibilities amongst public administration systems. The focus is put on the discovery, composition, mediation, and execution of services within complex scenarios, and the global ontology of semantic components needed for web service description is provided. Again, this approach requires an existence of web services on the side of public administrations. Contrary to the Terregov project, the global level of government services is covered, but the application of the solution on the level of local public authorities is not directly supported.

The *OntoGov* project[5] provides a semantics-based platform for the consistent composition, reconfiguration, and evolution of e-Government services. The solution includes a set of ontologies to describe and support the lifecycle of eGovernment services. The OntoGov approach mainly focuses on the software engineering side rather than on detection and orchestration of the government services; as a consequence, the interpretation on how the ontologies can be used in practical scenarios can be rather vague. In addition, the maintenance and usage of the OntoGov solution requires expert knowledge and lacks a certain degree of transparency for public servants when using the system.

In addition to the outlined projects, we can also mention some e-Government interoperability frameworks such as e-GIF in the UK[6], SAGA in Germany[7], European EIF IDABC[8] or SEMIC.EU[9]. These frameworks provide detailed information and guidelines about central government systems (on the national or European level). However, they fail to introduce specific information and rules for building eGovernment solutions for local administration [10].

To sum it up, there exist quite a wide range of approaches, proposals, frameworks, and projects in the area of semantic interoperability in e-Government domain, especially in creation and maintenance of semantic web services. However, the practical outcomes of the current research in this area (see e.g. [18]) are lagging behind expectations. The lack of supporting methodology, specialised tools, and guidelines describing how to create and maintain formal semantic descriptions of the services in practice may be one of the reasons. Another reason may be weak support for existing types of governmental services, and necessity to change

4 http://www.semantic-gov.org

5 http://www.ontogov.com

6 http://www.govtalk.gov.uk

7 http://www.cio.bund.de/DE/Standards/SAGA/saga_node.html

8 European Interoperability Framework for pan-European eGovernment services, http://ec.europa.eu/idabc/en/document/2319/5938.html

9 The Semantic Interoperability Centre Europe, http://www.semic.eu/semic/

(reengineer) dramatically how governmental services are provided, e.g. by implementing them as semantically described web services.

One of the main advantages of the semantic enhancement of government services is the capability of formally describing the meaning and context of government services, both traditional (i.e. face-to-face, "paper-based") as well as electronic ones (provided as electronic forms or web services), without the necessity of modifying the services themselves. In Access-eGov, this issue was targetted by developing tools as well as a methodology enabling the semantic interoperability of government services in practical applications [9], [2].

2 Approach, Methodology, and Technology

Following the main objective of the Access-eGov project, the access to government services (either traditional or electronic) needs to be provided in an "integrated" manner, in accordance with the pre-defined life event situations and business episodes of system users, citizens and businesses. From this perspective, the central position of life events (business episodes) as expressions of a user's needs is in correspondence with the life event approach [12], an effective and frequently used method in the user-oriented e-Government solutions. The life event is a situation in the life of the citizen (similarly, a business episode is a situation in the life cycle of the business organisation), which requires the provision of government services and should be semantically described within the system. Life events are usually complex and can be decomposed into several mutually dependent sub-goals. The fulfilment of the sub-goals leads to the solution of the given situation. Each sub-goal can be resolved to (i.e. fulfilled by) a set of government services that are provided either in a traditional way (requiring face-to-face communication and mostly based on some paper forms) or in an electronic way (available on-line via web service interfaces or web forms).

Sub-goals can be conditioned, organised in workflow structures using if-then-else constructs, cycles, and dependencies on outputs of other services – according to the specific case of the citizen or the organisation. During execution, the list of sub-goals for a life event is customised (e.g. by information provided by the user to specify his/her case) and then dynamically evaluated [14]. Services, which resolve sub-goals, may require some additional inputs provided by other services, so sub-goals can be further decomposed to sub-sub-goals and so on. During the service resolution process, the system may dynamically create a user scenario of the life event by evaluating the conditions of sub-goals. The user is then navigated to the proper services that are capable of fulfilling the goals and solving the life event situation.

Ontologies, as powerful knowledge representation formalism for modelling real-world concepts, were chosen as a basic mechanism for semantic modelling and the annotation of life events, goals, sub-goals, services, and other specific concepts from the public administration (PA) domain. This approach allows for the integration of existing (and future) systems and government services, as well as their functional interconnection on the technical, syntactic, semantic, as well as organisational level. To design a concrete ontology structure and content according to the purposes of the Access-eGov project, three basic resources were identified, namely:

- a *conceptual model* provided by selected semantic framework,
- existing and available *ontology resources*,
- formalised *requirements* that were provided by user partners of the project and were systematically organised into an ontology-like structure.

2.1 Conceptual Framework

After a detailed survey and analysis of existing and most used approaches (RDF-S, WSDL-S, WSMO, and OWL-S for ontologies; BPEL4WS for modelling web services in a business process interaction, etc.), we decided to apply the WSMO as a basic conceptual framework and implementation platform. The WSMO framework provides a consistent conceptual model for the semantic description of web services, with the inclusion of mediators and the distinction between goals and capabilities. It also provides the WSMX execution environment, WSML language specification for ontology formalisation, as well as the WSMO Studio visual development environment [3]. Based on this technological framework and the specified functional description, the architecture of the Access-eGov system was proposed [14].

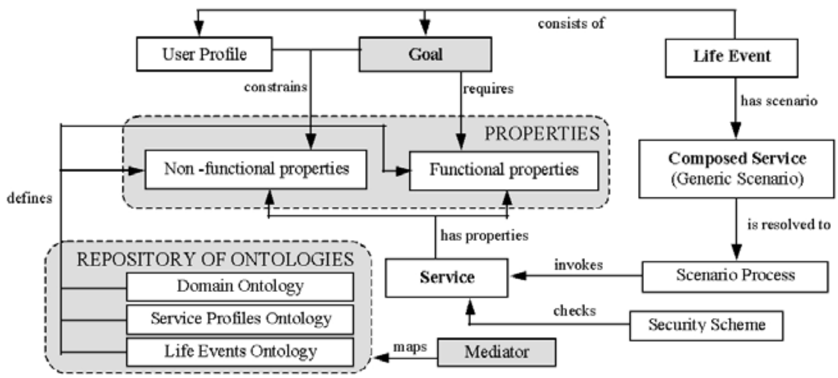

Figure 1
The WSMO-based conceptual model adapted for the life-event approach

In accordance with the life event approach, the conceptual framework of WSMO was extended, as it is depicted in Figure 1. The parts reused from the original WSMO model, such as Goal, Mediator, and Ontology top-level elements and the descriptive properties, are marked with a gray background. Two new top-level WSMO elements were added:

- *Life Event* element as a formal model of user's needs, consisting from multiple goals and services organised into a generic scenario and expressed by orchestration construction consisting from workflow, control-flow and data-flow sequences.
- *Service* element as a generalisation of the web service concept, already provided by WSMO. This extension enables the description of both electronic and traditional government services by means of a service profile, containing functional and non-functional properties, capabilities, and interfaces.

In addition, the WSMO conceptual model was enriched by the workflow extensions that are capable of representing a process model of the interactions with human actors in the e-Government domain [19]. The current WSMO specification provides the process model based on abstract state machines and is not structured in a way suitable for interaction with human actors, which is required for e-Government applications. The extended Access-eGov process model is based on the workflow CASheW-s model [16]. The state signature is reused from the WSMO specification and replaces the transition rules with the workflow constructs. The shared ontology state signature allows for reusing the grounding of input and output concepts to the communication protocols via WSDL for the invocation of web services. The workflow model consists of activity nodes connected with the control-flow or data-flow links. The nodes can be divided into atomic nodes (*Send*, *Receive*, *AchieveGoal* and *InvokeService*) and control nodes (*Decision*, *Fork* and *Join*).

An example of WSML statements for the orchestration interface that represents the high level process connected with the life event “Establish an enterprise” is presented in the following listing:

```
interface EstablishEnterpriseLifeEventInterface
orchestration
   workflow
     perform n1_1 receive ?x memberOf Q1.
     perform n1_2 achieveGoal RegisterInLocalGovernmentGoal
     perform n1_3 achieveGoal RegisterInStatisticalOfficeGoal
     perform n1_4 achieveGoal RegisterInTaxOfficeGoal
     perform n1_5 achieveGoal RegisterInSocialInsuranceAgencyGoal
   controlFlow
     source n1_1 target n1_2
     source n1_2 target n1_3
     source n1_3 target n1_4
     source n1_4 target n1_5
   dataFlow /**/
```

By interpreting this formal description, first the batch of answers to the pre-defined questions (*Q1*, invoked by the *Receive* node) needs to be received from the user by the process. Then other sub-goals need to be achieved in the right order. As can be seen, one of these goals is *RegisterInStatisticalOfficeGoal*. Transitions in the *controlFlow* part express that all nodes are executed in a sequence. The *dataFlow* part is empty in this case, since there is no direct use of some variable between these workflow nodes.

An example of the choreography interface, which composes a set of external services from a perspective of upper service, can be defined as follows:

```
interface RegisterInStatisticalOfficeInterface
choreography
   workflow
        perform n2_1 receive ?x memberOf Q3.
        perform n2_3 receive ?x memberOf FormRG_1.
        perform n2_4 decision
        perform n2_5 receive ?x memberOf FormRG_RD.
        perform n2_6 send ?x memberOf REGON.
   controlFlow
        source n2_1 target n2_3
        source n2_3 target n2_4
        source n2_4 target n2_5 guard ?x[q1 hasValue moreThanThree].
        source n2_5 target n2_6
        source n2_4 target n2_6
   dataFlow
        source n2_1{?x} target n2_4{?x}
```

This formal description can be interpreted as follows. At the beginning, the batch of answers to the pre-defined questions (*Q3*) needs to be received from the user. Then the process needs to receive a given form, which is represented by the *FormRG_1* variable. The *Decision* node in the next statement means that some of the following nodes are optional – in this case, only the node *n2_6* is optional, which is determined by the controlFlow part. Next, the process might need to receive another form *FormRG_RD*. Finally, the process sends the *REGON* number. The controlFlow part contains one conditional transition. The transition between the *Decision* workflow node and the following *Receive* workflow node depends on the answer to the question about the number of business activity types (i.e. if q1 is more that 3). The process can thus reach the final node right after this *Decision* node or from the last *Receive* node depending on the decision result. The dataFlow part specifies that the variable from the first node (*n2_1* - the batch of question) is equivalent with the variable from the *Decision* node (*n2_4*).

2.2 Ontology Construction

Semantic structures of the Access-eGov platform were created upon the specified and extended conceptual model of WSMO. The second resource used for the design of resource ontologies resulted from our survey of the ontology resources available worldwide. Using already existing ontologies assures consistency with the widely accepted standards and avoids unnecessary double work. After a detailed analysis of about 25 ontology resources and standards, we finally reused the following ontologies:

- WSMO ontologies[10] for description of date, time, and location;
- vCard ontology[11] for addresses and personal data;
- Dublin Core[12] for metadata and document types;
- Terregov, DIP, DAML, GEA, GOVML, AGLS metadata set, and IPSV ontologies for description of specific e-Government concepts.

Existing ontology resources were used 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
```

Finally, as the third resource of the ontology design procedure, requirements from the Access-eGov user partners were collected in a systematic way to produce ontology models of life events, sub-goals, and provided services for the pilot applications to be carried out within the project (cf. Section 4). The so-called *requirement-driven approach* [9], a method originally designed and developed within the Access-eGov project by one of the project partners (the German University of Cairo), was used as the main resource for ontology creation. This 7-step procedure starts with the identification of users' information needs for a particular case, which are provided by public administrations in a free-text format, e.g. as user scenarios. The information needs are then analysed with respect to the required properties, such as scope, relevance, etc. A list of proposed services together with related laws and regulations, documents needed to negotiate between users and public administrations, and other requirements concerning information quality are provided.

The descriptions and background materials are processed and a glossary of topics and terms is generated. The terms are organised into the controlled vocabulary, which contains a hierarchy of categories and subcategories created from the glossary by grouping the terms into hierarchical subgroups. This means that the

10 http://www.wsmo.org/WSMO_ontologies.html

11 http://www.w3.org/2006/vcard/

12 http://dublincore.org

terms in the controlled vocabulary are connected by *is_a* relations. In the next step, a set of other relations and mutual dependencies is identified between the terms. New categories (terms, concepts) 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. The core fragment of such a structure, which served in Access-eGov as a "seed" of Life events ontology, is presented in Figure 2. The grey background identifies so-called boundary concepts that will be annotated as non-functional properties of the services.

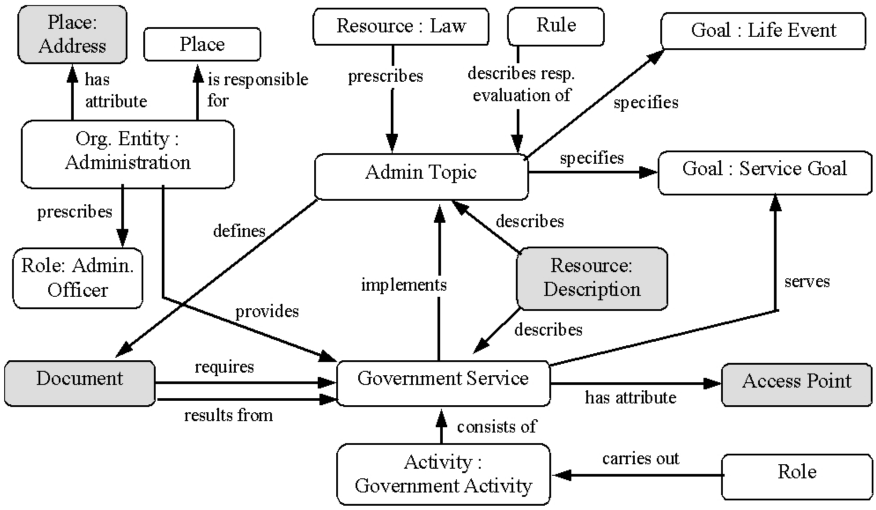

Figure 2
The ontology-like structure of identified concepts and relations

The ontology-like structure is then 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
```

The produced ontology is still 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 enhanced process model of WSMO, described in Section 2.1 above, provides the 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) by means of the orchestration interface:

```
namespace {_"http://www.access-egov.org/ontologies/shg/",
  dc _"http://purl.org/dc/elements/1.1#",
  aeg _"http://www.access-egov.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}
```

By interpreting this formal description, first a batch of answers to the pre-defined questions (*Q1*) needs to be provided by the user. Then other sub-goals (*ApplyForMarriageGoal*, etc.) need to be achieved in a proper order. Transitions in the controlFlow part express that all the nodes are executed in a sequence. The dataFlow part specifies that the variable from the first node (*n1_1*, the batch of questions) is equivalent to the variable from the decision node (*n1_2*).

As a result of the 7-step procedure applied in Access-eGov, the following ontologies were created and formalised by WSML language:

- The *Core ontology* containing definitions of basic elements (concepts, attributes, relations) that are shared among the pilot applications and used for the annotation of the atomic services.
- The *Life Events ontology* containing conceptual descriptions of life events, complex goals (also referenced as generic scenarios), and elementary sub-goals for the pilots. Separate Life-Events ontologies were produced for each of the Access-eGov pilot applications.
- *Domain ontologies*, providing domain-specific information for the pilots. The ontologies are fully localized (concepts have labels in several languages – in this case the labels are in the English, German, Polish, and Slovak languages) and include concepts for description of forms, documents, certificates, location constraints, fees, questions, notification messages, etc. that are necessary to model the inputs and outputs of the provided government services. Separate domain ontologies were produced for each of the pilots.

The produced ontologies were uploaded as an asset on the SEMIC.EU portal[13] and are, after the required registration, freely available for further reuse, exploitation, or customisation.

3 System Architecture and Functionality

The Access-eGov system architecture, schematically depicted in Figure 3, consists of four main functional modules:

- The *AeG resource ontology,* a persistent data repository and a knowledge base that contains WSML representations of the life events and goals. In addition, it contains generic service concepts and service templates that enable the service annotation, as well as the instances of already annotated services.
- The *AeG core components* module, which includes the inner business logic of the system. The components are responsible for the decomposition of a given life event or goal into sub-goals, for the orchestration, composition, and mediation of the sub-goals within a workflow thread, for the semantic matching and discovery of the services for a given goal, as well as for the execution of the retrieved and resolved services.
- The *Annotation tool* (AT) for the semantic description (i.e. annotation) of the services that are to be integrated by the Access-eGov system. The web-based interface allows information providers to specify the non-functional properties for various service types, including traditional face-to-face services (in this case, the service is described by an explanatory HTML text that is presented to citizens), electronic, and web services. Capability interfaces, required inputs and provided outputs, and related workflow sequences are determined by a service template used during the annotation. The resulting WSML representations of the annotated service instances are stored in the resource ontology.
- The *Personal Assistant client* (PAC), a tool that enables the citizens to browse and navigate through the life event and corresponding sub-goals. This web-based tool is implemented as a kind of wizard that enables the personalisation and customisation of the thread of sub-goals by the answering of a set of customisation questions, which can be defined in a process model of the semantic representation of corresponding life events and sub-goals.

[13] http://www.semic.eu/semic/view/Asset/Asset.SingleView.xhtml?id=270

Operations performed during the design time on the side of information provider (i.e. a public administration) are represented in Figure 3 by thin arrows and are labelled by capital letters. An ontology designer uses the WSMO Studio tool [3] to create the resource ontology and customise it according to a given application case (A). The steps of the requirement-driven approach can be employed to specify the life events and goals, as well as the services and service types (templates). The structure of life events and goals is then automatically populated to the PAC to be presented for information consumers (B). However, the services that should correspond to particular goals need to be created separately, by means of semantic annotation (steps C-F).

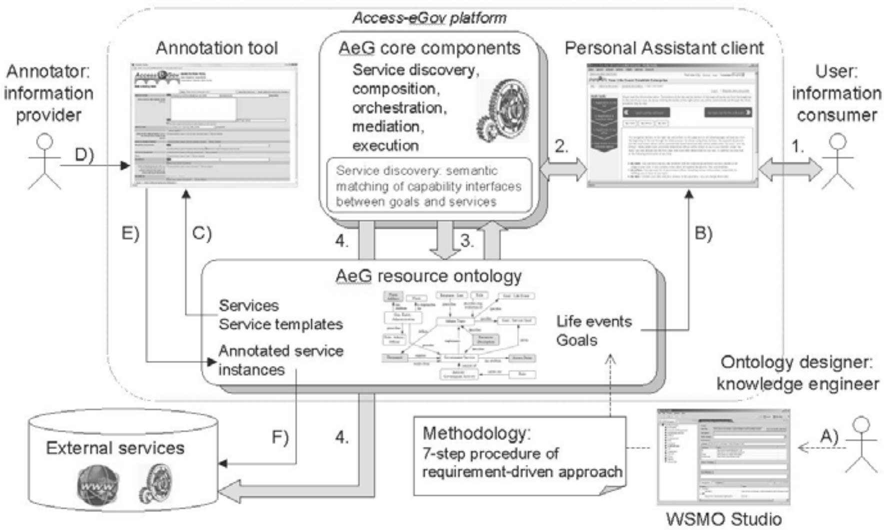

Figure 3
The architecture and control flow within the Access-eGov platform

The structure of generic services and service templates, created in the resource ontology, is automatically populated to the AT (C). An annotator then uses the AT to semantically describe the services, i.e. to specify concrete values for particular non-functional properties, defined by the employed service template (D). A WSML representation of the annotated services is created automatically and is uploaded into the resource ontology as a set of service instances (E). The service instances may contain a reference to an external web service or to an existing web content (i.e. a portion of a web page). This reference is specified as a non-functional property during the step D. After uploading the service instance to the ontology, the reference is evaluated, the external resource is validated, invoked, and the returning data are set as default value for the service instance (F).

Operations performed by the information consumer during the run time are numbered and represented in Figure 3 by thick arrows. A citizen uses the PAC to browse the life events and goals (1). Some of the goals may require an additional input that concretises the citizen's needs – then the citizen provides answers to the customisation questions. The core system evaluates the responses obtained from the citizen for a given goal and dynamically creates a new thread of sub-goals, which is then returned back to the PAC (2). The process model of the goal is modified by the provided answers and its evaluation includes the procedures as service discovery, composition, orchestration, mediation, and execution. The core system communicates with the ontology to decompose a complex goal to sub-goals, to orchestrate, mediate, and compose the sub-goals into a workflow thread (3). For atomic goals that cannot be decomposed to sub-goals, the semantic matching procedure is used to discover and dynamically resolve a set of proper services. The core system then transforms the resolved services, according to their type, to an executable form and invokes the referenced external services (4). The input values for the external services are populated from the input provided by users and/or calculated during the evaluation of the goal's process model (step 2). The output values provided by the invoked external services are returned to the process model, which is then modified accordingly and is presented to the citizen in the PAC (step1).

The Access-eGov system is built on the WSMO framework, using the WSMX execution environment for discovery, selection, mediation, and invocation of semantic web services. It is implemented in Java; the WSMO4j API[14] is employed for parsing the WSML ontologies and obtaining the respective WSMO elements. The architecture of the system combines principles of SOA-based web services and service-oriented peer-to-peer architecture, which brings the advancements of modularity, possibility of local or remote accessibility from any platform, fault tolerance, scalability, and ease of deployment.

3.1 The Annotation Tool

The Annotation Tool (Figure 4) was implemented as a standard web application, using the extended WSMO object model and JSF technology. The tool provides public administration personnel a set of forms for the specification of preconditions and non-functional properties as parameters of the government services. A template mechanism was implemented to ease the maintenance of predefined workflow sequences for the annotated services. A simple user access control and multilingual support, on both the interface and data level, is also included in the AT. In addition, a simple "content grabber" functionality enables to link particular a field in the form (i.e. the value of a service parameter, e.g. service hours of an office) with an element on an existing web site of the public administration. This solution enables the annotation of the external web pages and the semantic integration of their content into a unified e-Government application.

[14] http://wsmo4j.sourceforge.net.

Figure 4
The user interface of the Annotation tool

3.2 The Personal Assistant Client

On the side of citizens, the Personal Assistant client (Figure 5) was developed as a tool that provides browsing, discovery, and execution capabilities of proper services for citizens and businesses according to a specified life event or goal.

Again, PAC was implemented as a web application using the JSF technology. The layout, structure, and ordering of tabs in the interface are dynamically created from the annotated services and are customised by the answers provided by the given user. After selecting a life event, a corresponding navigation structure of sub-goals and services is displayed for users in a form of textual information, hyperlink, a field for inserting a specified input value, or an interface for invocation of a web service. Users can browse sub-goals and provide their answers when customisation input is requested. Then the system automatically resolves the sub-goals and navigates the user to a new set of sub-goals and services inferred from the conceptual model. The Access-eGov system can also directly invoke electronic services provided via a standardised web service interface. Finally, the user obtains all available information on the life event customised to his/her case, and has also the possibility to execute the actions required for particular services needed for the accomplishing of the life event.

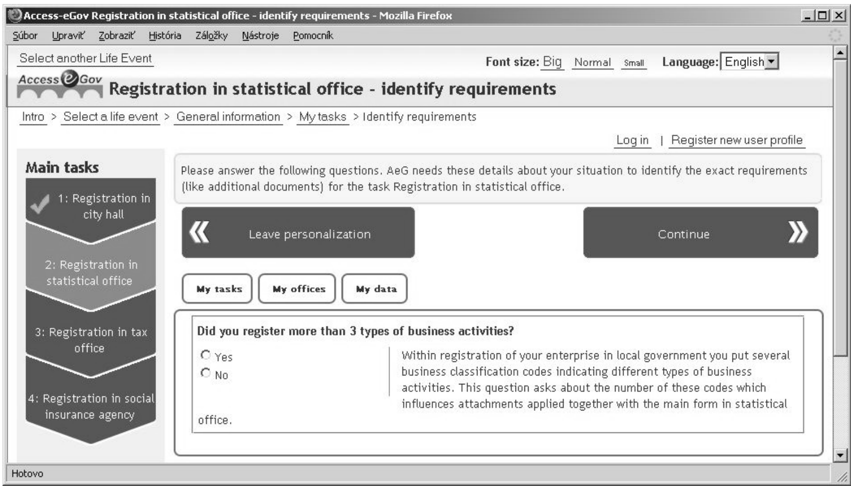

Figure 5

The user interface of the Personal Assistant client

4 Platform Testing and Evaluation

The Access-eGov platform in the scope of the described functionality was tested on three pilot applications in Germany, Poland, and Slovakia.

The *German pilot application* took place in the federal state of Schleswig-Holstein and dealt with the scenario “Getting married”. This application case was chosen as a prototype example [2], but the goal was to integrate the services of the different administrations of 1,120 municipalities located in the federal state.

The *Polish pilot application*, focusing on the "Establishing a new enterprise" scenario, ran in the Silesian region around the city of Gliwice. The main aim of this pilot application was to provide a portal-like interface that would integrate all the relevant information (provided in Poland mostly in a form of traditional services) of a rather complex process in one place, and would guide the citizens and businesses through the life event and related sub-goals.

The *pilot application in Slovakia* covered the area of Kosice Self-governing Region and the municipality of Michalovce as its part. The goal of this pilot was to provide a personalised guidance for citizens during the process of obtaining permits for building a house, including services related to the land-use planning and final approval proceedings.

The testing was carried out in two trials within each of three pilot applications. The first trial resulted in a set of requirements that were taken by system developers as a basis for system enhancements. The evaluation of the achieved results [6] demonstrated the feasibility of the proposed solution as a platform for front-office service integration. However, several issues were identified and requested as important for further improvement, specifically related to the graphical user interface of the PAC, navigation and browsing between the goals (tasks). The enhancements of the PAC invoked the necessity of adapting the core components accordingly, though the data structures and general functionality were not fundamentally changed. In addition, invocations of semantically annotated web services and related XSLT data transformations were fully integrated into the Access-eGov platform, namely into the PAC for presentation of the service results on the web user interface.

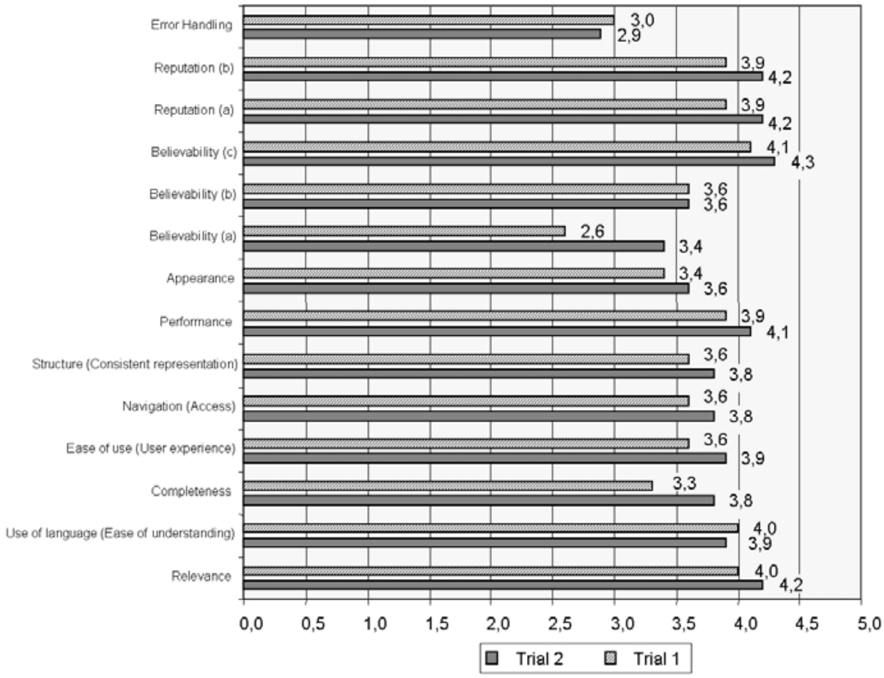

Figure 6

Overall results of the user evaluation after the second trial

The updated system was tested in the second trial on the same pilot applications as for the first trial. The testing was aimed at the ability of the system to integrate the external web services and data resources, and globally at proving the functionality and behaviour in real-world settings, especially focusing on the interoperability of heterogeneous distributed services and information resources. Technical testing, which was focused on the correctness, speed, and overall performance of the platform, resulted in the identification of significant improvements, in the service discovery and the overall speed of system response.

The method of online questionnaire was adopted for collecting the feedback from involved public. The chart presenting the responses obtained on the usability of the updated PAC is depicted in Figure 6. The levels of believability include a) ability to identify a provider (author) of the presented information, b) ability to determine which of the presented links lead to an external web site, and c) overall correctness of the provided information. The reputation covers a) a conviction to take the Access-eGov as a good (relevant, significant) source of information, and b) trustfulness of the information provided by the PAC. The usability evaluation of the second trial was quite successful, since average results of all the investigated aspects were better than those achieved in the first trial. The largest improvement was achieved in believability (+15%) and in completeness of presented information (+10%). Moreover, the people involved in the testing expressed rather positive feedback, stating that the solution is easy-to-use and provides very useful information for the modelled life events.

Acknowledgement

The work on this paper is the result of the project implementation: Development of the Center of Information and Communication Technologies for Knowledge Systems (ITMS project code: 26220120030) supported by the Research & Development Operational Program funded by the ERDF.

References

[1] Abecker, A., Sheth, A., Mentzas, G., Stojanovich, L. (eds.): Proceedings of AAAI Spring Symposium "Semantic Web Meets eGovernment", Technical Report SS-06-06, AAAI Press, Menlo Park, CA, 2006

[2] Bednar, P. et al: Semantic Integration of eGovernment Services in Schleswig-Holstein. In: Electronic Government, 7th International EGOV Conference, Springer, LNCS 5184, 2008, pp. 315-327

[3] Dimitrov, M. et al: WSMO Studio Users Guide, v. 1.27. July 25, 2007. Available at http://www.wsmostudio.org/doc/wsmo-studio-ug.pdf

[4] eEurope 2005: An information society for all. COM (2002) 263, 2002

[5] eGovernment RTD 2020 project. Visions and Conceptions of European Citizens. Deliverable D 4.1: Final Roadmapping Workshop Report, 2007

[6] Furdik, K., Mach, M., Sabol, T.: Practical Experiences with Enhancing Semantic Interoperability in eGovernment using WSMO. In: Proceedings of MeTTeG 2008, Halley Editrice, 2008, pp. 23-35

[7] i2010 - A European Information Society for growth and employment, COM (2005) 229, 1 June 2005

[8] Interoperability for Pan-European eGovernment Services. COM (2006) 45 final, Brussels, 13.2.2006

[9] Klischewski, R., Ukena, S.: Designing Semantic e-Government Services Driven by User Requirements. In: Electronic Government, 6th International EGOV Conference, Proceedings of ongoing research, project contributions and workshops, Trauner Verlag, Linz, 2007, pp. 133-140

[10] Koussouris, S. et al: Building a Local Administration Services Portal for Citizens and Businesses: Service Composition, Architecture and Back-Office Interoperability Issues. In: Electronic Government, Springer LNCS 4656, 2007, pp. 80-91

[11] Kubicek, H., Cimander, R.: Three Dimensions of Organizational Interoperability. Insights from Recent Studies for Improving Interoperability Frame-Works. In: European Journal of ePractice, www.epracticejournal.eu, No. 6, January 2009

[12] Leben, A., Vintar, M.: Life-Event Approach: Comparison between Countries. In: Electronic Government, LNCS 2739, 2003, pp. 434-437

[13] Lopez, D. M., Blobel, B.: A Development Framework for Semantically Interoperable Health Information Systems. In: International Journal of Medical Informatics 78(2), 2009, pp. 83-103

[14] Mach, M., Bednár, P., Hreno, J.: Execution and Composition of Government Services. In: Proceedings of MeTTeG 2007, Halley Editrice, 2007, pp. 139-153

[15] Mach, M., Sabol, T., Paralic, J.: Integration of eGov Services: Back-Office versus Front-Office Integration. In: Proceedings of ESCW 2006, Budva, Serbia - Monte Negro, June 2006, pp. 48-52

[16] Norton, B., Pedrinaci, C.: 3-Level Service Composition and Cashew: A Model for Orchestration and Choreography in Semantic Web Services. In: Springer LNCS 4277, 2006, pp. 58-67

[17] Roman, D., Scicluna, J., Fensel, D., Polleres, A., de Bruijn, J.: D14: Ontology-based Choreography of WSMO Services. WSMO working draft, DERI, 2006. Available at http://www.wsmo.org/TR/d14/v0.4/

[18] Scholl, J., Klischewski, R: E-Government Integration and Interoperability: Framing the Research Agenda. In: International Journal of Public Administration (IJPA), Vol. 30, Issue 8-9, 2007, pp. 889-920

[19] Skokan, M., Bednar, P.: Semantic Orchestration of Services in eGovernment. In: Proceedings of Znalosti (Knowledge) 2008, STU Bratislava, Slovakia, 2008, pp. 215-223

[20] Vernadat, F. B.: Technical, Semantic and Organizational Issues of Enterprise Interoperability and Networking. In: Annual Reviews in Control, Volume 34, Issue 1, April 2010, pp. 139-144