ISBN 978-83-60810-14-9
ISSN 1896-7094

Access-eGov–Personal Assistant of Public Services

Magdalena Sroga
Cities on Internet Association, ul. Prądnicka 4/15, 30-002 Kraków, Poland
Email: m.sroga@mwi.pl

***Abstract*—Aim of this article is to present targeted research project Access-eGov, founded from the Sixth Framework Programme , priority 2 - Information Society Technologies. The article describes Access-eGov architectural and logical design and used technologies which enable to achieve the Project's aims. The Project introduces new, user-centric approach which uses predefined life events to solve citizens' problems which require from users implementation of public services. Project's approach enables users to perform complex executive scenarios instead of separated public services. This approach concentrates on user's needs and helps to build semantic interoperability among e-government services.**

I. Introduction

Access-eGov (Access to e-Government Services Employing Semantic Technologies) is the acronym of a project pursued by a consortium of European organizations. The project started in January 2006 and is going to be finished in December 2008. The aim of the Access-eGov results from the action plan eEurope which formulated specific aims in the area of eGovernment to make public administration open, transparent, inclusive and productive. In this area the specific objective of the Project is to support semantic interoperability among e-government services across organisational, regional and linguistic borders what is being achieved by employing semantic technologies.

In “real life” situations citizens, as well as businesses usually do not need an atomic (singular) government service, but a (often non-linear) sequence (including if-then-else branches) – it means “scenario” of atomic services. Since we are still far away from the situation where 100% of government services are available on-line (and the level of the availability of e-services varies quite significantly across the EU member states), users usually have to deal with a combination of traditional services and e-services. Due to that fact Access e-Gov’s approach considers both electronic and traditional services from which the user scenario (usually “hybrid” one) is composed. Before identifying and executing a scenario, users are sometimes facing a more trivial problem – which public administration institutions are providing service(s) which they need in the given situation (context) and what inputs are required to execute this service independently on the way of provision.

To solve the first users’ problem using Access-eGov system, currently existing public services may be annotated using Annotation Tool and afterwards registered in the Personal Assistant platform. All accessible services are enhanced by the detailed description, links to relevant documents and guidelines on service providers, thus the user is provided with the complete information related to the particular service. The second problem is solved by composing a user scenario from available services relevant to user’s life situation. A significant issue is that the user is provided with complete, customised information relevant to her life case. Customisation is achieved by employing semantic technologies, particularly WSMO ontologies.

II. Background

In the last years a number of significant developments have occurred that motivate the use of Semantic Technology in eGovernment. It is well-known that Semantic Technology enables federation, aggregation and inferencing over information, as well as helps to solve interoperability, integration, reusability, and accountability issues in and across different institutions. There are created new systems supporting public administrations by Semantic Technologies and applications based on ontologies. But not many systems adopt the approach of user’s life events, which is going to be the core Access-eGov strategy – to simplify users’ interactions with public administration as much as possible.

Web Services are supposed to be the technology which supports realisation of cross-governmental, integrated services. They constitute the common technology to fulfil application-to-application interoperability, based on XML message exchange that is capable of dynamically invoking remote software components with a minimum effort in interface description and customizing.

In semantically-enriched systems, ontologies or controlled vocabularies are used as conceptual support for providing information about resources and for accessing them. Therefore public servants shall be empowered by Access e-Gov technology to annotate their institutions’ services on their own, being provided with intuitive software and straight-forward reference manuals.

Modern e-Government landscapes can be categorized in many ways according to solutions offered to their customers or according to their technical implementations. Criteria that Access-eGov takes into consideration are notably the easy accessibility of government services for the customers and the extent to which information systems can interact with each other in modern e-Government landscapes. The criteria of interoperability mainly involve dedicated interconnection

978-83-60810-14-9/08/$25.00 © 2008 IEEE

of information systems between several agencies on the same administrative level of government. Only in a few scenarios in the UK and Australia cross-governmental information links on a mutual basis and on different levels of government can be observed. Openness to external partners also includes the ability to interact on a technical layer with non-governmental or private organizations. Most advanced examples of modern service-oriented architectures and usage of message-based information exchange services in order to communicate between back-end-systems are: open service interfaces with OIOXML in Denmark, Public Service Infrastructure (PSi) in Singapore, Government eLink in Sweden, and Government Gateway in the UK.

The evolution towards integrated IT-based public services shows the necessity to adopt new ways of interaction between administration institutions. Some EU projects have been already developing practicable solutions to problems deriving from interoperability issues. They mainly focus upon semantic enrichment of electronic services and their aggregation and orchestration towards combined "complex e-services". Nevertheless, most of these projects focuses on the technical side and still lacks a citizen-centred point of view that could be taken by implementing software components tailor-made for the citizen's attendance when applying for a public service. In these projects, citizens' needs take a background position compared to technical aspects. Therefore, new approaches in e-Government have to put the emphasis on easy service accessibility for customers, what is the main difference between Access-eGov and those projects and also an noticeable value added of the project.

III. Access-eGov Functionality and Components

An Access-eGov software component is any piece of software that performs a specific technical task – i.e. it does not fulfil a user requirement by itself, but it solves a low-level problem specific to a domain.

An Access-eGov software module is any piece of software that performs a specific functional task – i.e. a task that fulfils a certain user requirement, thus solving a high-level problem specific to the user domain.

Whereas a component cannot operate individually without the tasks performed by other software components, a module should be able to work independently of other modules and when the module stops working, other modules are not affected.

A. Access-eGov Logical Architectures

The figure 1 illustrates general architecture of Access-eGov platform which may be sub-divided into the three major parts: the Access-eGov Infrastructure, the Access-eGov Personal Assistant client, and Access-eGov Annotation services (not an integral, but an affiliated part of the Access-eGov Infrastructure).

The services are owned by public administration and thus located on its premises. They are simply made available via the Access-eGov system and thus they are not an integral part of the overall system. These can be both electronic services (provided directly via web service interfaces or web forms) and traditional ones (i.e. provided face-to-face). Executable services will dispose of an electronic XML-interface to the Access-eGov Infrastructure, whereas traditional ones are only described and registered in the Access-eGov platform. They are supposed to be annotated by public agencies which are responsible for them and want to expose them to the public.

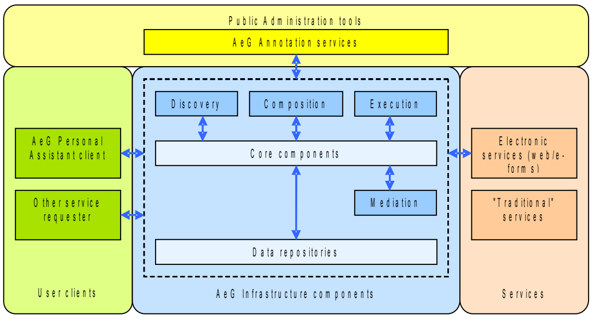

Fig 1: Overall architecture of the Access-eGov platform

Access-eGov Infrastructure components may be chosen by public agencies and installed on their premises or data centres dependently on which of them they wish to have. Such a "local" installation of the Access-eGov Infrastructure components is supposed to interact as a peer in the peer-to-peer overlay network that Access-eGov is likely to consist of.

B. Access-eGov Data View

This section describes the relations between several data entities in order to provide a more detailed view on the platform data structures. The figure 2 presents structural correlations of data.

Life event denotes specific situation in user's life which requires performance of public services series. Life events can be categorized into groups and organized in multiple hierarchies. Using the Personal Assistant portal site, user may "browse" or navigate through life event categories in order to select an appropriate life event.

A life event may be assigned multiple goals, which will formalize user needs. Life event's goals can have specified optional preconditions, which allow users to customize their specific life event. Preconditions are specified as logical expressions with input variables provided either explicitly by the user or from the user profile (preconditions are not dependent on service invocation).

Goal specifies objectives of the user who wants to perform a particular service, including functionalities that the service should provide from the user's perspective. Goals formalize user needs by specifying the requested outputs and effects. This is declared in the same way as service functional properties.

Service profile specifies what the service does provide from user's perspective and is used by the public administration to advertise services. Service profile consists of non-functional and functional properties.

Functional properties describe inputs, outputs, preconditions and effects of the service (IOPEs). They are specified as logical expressions, which consist of terms constraining type and property values of various resources required for or

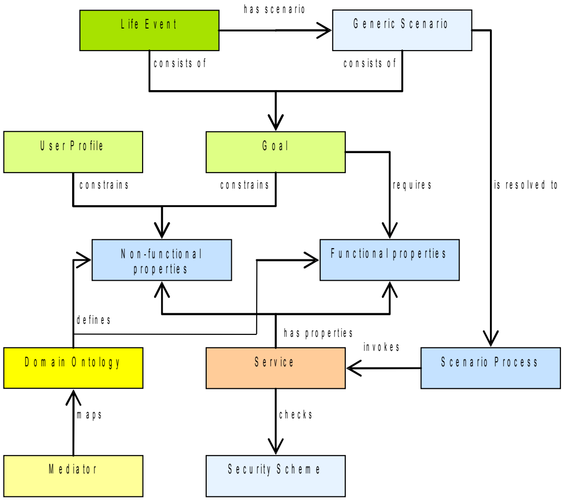

Fig 2: Logical data view on Access-eGov

provided by services. Types used to specify functional properties are defined in the domain specific resource ontologies.

Non-functional properties describe semi-structured information intended for requesters for service discovery, e.g. service name, description, information about the service provider and properties which incorporate further requirements for service capability (e.g. traditional office hours and office location, quality-of-service, security, trust, etc.). Structured non-functional properties are specified by domain specific ontologies.

The process model guides users to achieve specific goals and coordinates activities of users (citizens or businesses), traditional public administration services and web services. The process model is represented as a collection of activities designed to produce a specific output for a particular requester, based on a specific input. In that sense, activity is a function or a task that is carried out over time and has recognizable results.

C. *Access-eGov Modules*

Most of the Access-eGov modules are derived from the needs implied by the use cases and they may be divided into three main groups: information provider related modules, information consumer related modules, and system management related modules.

1) Information Provider Related Modules

Among the information provider related modules there are three separated modules: service annotation module, service discovery module, and service composition module.

The Annotation service module consists of a web-based application not being an integral part of the Access-eGov Infrastructure. Its main purpose is to enable domain experts to semantically describe their electronic or traditional services, by using relevant public service ontology. The web application provides also capabilities to allow annotation of traditional websites as well as an easy inspection of the existing content, which concurrently can be annotated.

Via web service interfaces, the Access-eGov Annotation module is able to access Ontology and Service Repositories within the Persistence Layer in order to register services and publish their descriptions. The creation, modification and editing of these semantic descriptions is controlled by the security subsystem.

Access-eGov system requires from the services to be semantically described and registered. For semantic description the Annotation service module uses the concepts and relationships from an appropriate ontology and to mark-up important aspects of the service or website.

Additionally that module is responsible for creating and managing goals and life events which are key idea of Access-eGov approach. Goals describe what user wants to achieve (similarly to WSMO vocabulary) and life events are situations happening to users which directly cause the need of executing some public services. Moreover goals and life events are workflow-like constructs that could be considered as outputs or interfaces provided by AeG system for users.

Service discovery module semantically matches functional properties of goals and services in the process of service discovery, in order to select services which are able to achieve the goals. Non-functional properties specified by the requester are then used to additionally filter or reorder the discovered services according to the requester preferences.

Service discovery task can be divided into two cases: “full-text search” and “semantic search”. The first case is a functionality provided in form of an interface to already existing full-text search engines in order to retrieve services and life events or goals from simply comparing the set of annotated properties. The second case is a functionality which allows to go beyond mere full-text matching for a semantic on-the-fly computation of an appropriate chain of services. The user input is then used to check preconditions and effects of registered services without invoking an already existing workflow.

Service composition module is used in case the service discovery module cannot find an atomic service which is able to achieve required goal. This module tries to orchestrate existing services to the new scenario to solve this goal. Although there are many initiatives to define industry standard languages for web service orchestration like BPEL, they have restricted capability to support only static service composition. Access-eGov Composition module provides support for dynamic composition of the services, which is not based on the static workflow pre-defined for the life event.

Automatic composition of services, which will solve these problems, is the subject of the current and future research. For this reason, current specification of the Access-eGov Composition component includes a semi-automatic approach based on the generic scenarios defined for the life event categories.

2) Information Consumer Related Modules

Among the information consumer related modules there are two modules: scenario execution module and personal assistant module.

Scenario execution module is responsible for executing user's scenario which is done via Personal Assistant. Service execution is invoked in case the user wants to achieve the goal, and in order to do it lets Personal Assistant start the execution of the retrieved service or workflow. The main activities of the scenario execution are as follows: invoke web service, invoke traditional service (special type of invocation, where activity is led back from the user - to the Personal Assistant. Executing of the scenario waits then for user to input the output of the traditional service.), resolve subgoal, and check timeout.

The personal assistant module is responsible for interaction with the user of the Access-eGov platform, but the user does not work directly with the module but with the Personal Assistant Client which is a web application which communicates to the Access-eGov platform and invokes its services through the module. Significant issue in this context is user's authentication through the user name and password which supports customisation of e-government services according to citizen's profile which may contain sensitive data.

3) System Management Related Modules

System management related modules cope with the core functionality of the Access-eGov platform and the only one module belongs to this category - system core module which is responsible for the interaction with the user of the Access-eGov platform and tasks relevant to the core platform functionality. That means manipulation with ontologies, connections management and security issues.

D. Access-eGov Components

Each Access-eGov module consists of several components which perform its specific task and provide necessary functionality of particular parts of the modules. For example there is a group of components responsible for service annotation itself, for maintenance of life events and goals, for managing ontologies, for selecting services according to particular properties, or they provide e.g. interface to the full-text search of services.

E. Conceptual Architecture

The basic functionality of Access-eGov is annotating services and store them in efficient way. Those services have to be retrieved according to certain citizen requirements and the administrators in public authorities ought to have a possibility to string such annotated services together to form new “meta services” - user scenarios. Therefore two different views may be considered: point of view of an administrator in a public institution as well as citizen’s perspective.

An information provider has three main tasks, namely registering new services, annotating services and building generic workflows out of already defined services. A service consumer has two main possibilities of interaction with the platform: goal specification and request for executing the retrieved services. Communication with the platform always occurs through the personal assistant (or an equivalent user client interface).

IV. Technologies Used

There is a variety of Semantic Technologies used for ontology creation. And then ontologies combined with semantic description are used for defining semantic web services. Access-eGov's approach required services platform to achieve semantic interoperability on different administration levels. Therefore, semantic technologies are mainly used to overcome language barriers in service description and annotation terms. In order to fulfil the goal, Access-eGov applies Semantic Web Services formalisms to create ontologies and semantically describe services. Four different formalisms were considered, namely: Web Ontology Language for Services (OWL-S), Web Service Modeling Ontology (WSMO), Web Service Semantics (WSDL-S), and Business Process Executions Language for Web Services (BPEL4WS). Before making the decision their advantages and disadvantages were considered.

A. OWL-S

The structure of the OWL-S consists of a service profile for service discovering, a process model which supports composition of services, and a service grounding, which associates profile and process concepts with the underlying service interfaces. Moreover OWL-S distinguishes between atomic, simple, and composite processes. Atomic processes can be invoked, have no sub-processes, and are executed in a single step from the requester's point of view.

Two main OWL-S disadvantages are: usage of single modelling element (Service Profile) for requester and provider, and the problem with rule languages, which in spite of being recommended in the combination may lead to undecidability or leave semantics open (SWRL, DRS). There are also problems with ways of interaction between OWL and rule languages (e.g. KIF). The consortium saw a little apprehension before using OWL-S due to the fact the language must have been extended for traditional services.

B. WSDL-S

WSDL-S is a small set of proposed extensions to Web Service Description Language (WSDL) by which semantic annotations may be associated with WSDL elements. WSDL-S defines URI reference mechanisms to the interface, operation and message constructs to point to the semantic annotations defined in the externalized domain models. WSDL-S defines following extensibility elements: *modelReference*, *precondition*, *effect*, *category*, and attribute – *schemaMapping*.

Significant advantage of WSDL-S is its independence on the semantic model languages. However, it has several weaknesses: new, being still in research phase approach and lacking explicit support of orchestration and choreography.

C. BPEL4WS

BPEL4WS is a specification that models the behaviour of Web Services in a business process interaction. It is based on the XML grammar which describes the control logic required to coordinate Web Services participating in a process flow. An orchestration engine can interpret this grammar so it can coordinate activities in the process. BPEL4WS is a layer on the top of WSDL, which defines the specific operations and BPEL4WS defines how the operations can be sequenced.

Although BPEL4WS is very suitable for representing workflow, it needs to employ semantic into WSDL, what is a new approach. Also, there might have been problem with dealing with implementation of semantics.

D. WSMO

The Web Service Modeling Ontology (WSMO) is a conceptual model for describing semantic Web Services. WSMO consists of four major components: ontologies, goals, Web Services and mediators.

Ontologies provide the formal semantics to the information used by all other components. WSMO specifies the following constituents as part of the description of ontology: concepts, relations, functions, axioms, and instances of concepts and relations, as well as non-functional properties, imported ontologies, and used mediators. The latter allows the interconnection of different ontologies by using mediators that solve terminology mismatches.

Goals specify objectives that a client might have when consulting a Web Service, i.e. functionalities that a Web Service should provide from the user's perspective. Goals are characterized by a set of non-functional properties, imported ontologies, used mediators, the requested capability and the requested interface.

A Web Service description in WSMO consists of five subcomponents: non-functional properties, imported ontologies, used mediators, a capability and interfaces.

A choreography description consists of the states represented by ontology, and the if-then rules that specify (guarded) transitions between states. The ontology that represents the states provides the vocabulary of the transition rules ant contains the set of instances that change their values from one state to the other. Concepts of an ontology used for representing a state may have specified the grounding mechanism which binds service description to the concrete message specification (e.g. WSDL). Like for the choreography, an orchestration description consists of the sates and guarded transitions. In extension to the choreography, in an orchestration can also appear transition rules that have the invocation of a mediator as post-condition. The mediator links the orchestration with the choreography of a required Web Service.

Mediators describe elements that aim to overcome structural, semantic or conceptual mismatches that appear between the different components building up a WSMO description.

After analysis WSMO proved to be formalism having the biggest range of advantages, as relying on loose coupling with strong mediation, combining conceptual modelling and rules, and providing opportunity for sophisticated goal-oriented discovery.

One of WSMO advantages crucial for Access-eGov is very clear distinction between functional and non-functional properties which are used within Access-eGov project very widely. WSMO enumerates all the relevant functional properties (for example Web service has Capability and Interface as its functional properties) and it allows flexible and easy to extend sets of NFPs everywhere. In WSMO conceptual model it is possible to enumerate all the parameters that are functional properties for the aim of WSMO. It is also possible to add different NFPs according to the own needs of the user.

WSMO provides a consistent conceptual model for the semantic description of web services, with the inclusion of mediators and the distinction between goals and services. In addition, the WSMO conceptual model fits best the proposed architecture and functionality of Access-eGov system.

However, some parts of the specification haven't been finished yet, what was considered as disadvantage, but consequently it offers possibility of specification development and conducting research. Moreover, orchestration and choreography is based on the abstract state machine where workflow is encoded and it also must be extended for traditional services.

Despite these inconveniences, consortium has decided to use WSMO technology as best suitable for Access-eGov's approach.

V. Access-eGov Ontologies

A. Conceptual View of Ontologies

Access-eGov ontologies are utilized to semantically express real-world concepts in a way defined and agreed upon by communities of users. "Ontology" in technical terms constitutes a formal specification of a shared conceptualization. Ontologies define an agreed common terminology by providing concepts and relationships between these concepts. In order to capture semantic properties of relations and concepts, ontology also provides a set of axioms (i.e. logical expressions in some structured language). Access-eGov uses three basic ontologies: life events ontology, service profiles ontology, and Acces-eGov domain ontology. Structure of ontologies is illustrated in the figure 3.

The domain ontologies are considered lower level ontologies within the system. They describe all the relevant pieces of domain information related to user's scenarios. That means they describe functional and non-functional properties of a particular service. They are web based ontologies that are not necessarily relevant to the web services. The domain ontology describes the lower (i.e. technical) level of the Access-eGov system.

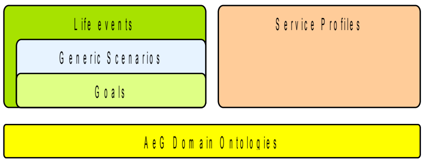

Fig 3: Conceptual data view in Access-eGov

The domain ontologies contain conceptual descriptions of domain-specific information for the pilot applications. It includes the concepts describing various forms, documents, certificates, location constraints, fees, questions, notification messages, etc., that are necessary to model the inputs and outputs of the provided governmental services.

The other two ontologies are "Life events" and "Service Profiles" ontologies which describe more abstract data. They are not simple web ontologies, but extended with semantic descriptions of possible life events (Life Events ontology) or (web) services (Services Profiles ontology). They denote more abstract system levels just as service description.

The Life events ontologies contain conceptual descriptions of life events, complex goals (also referenced as generic scenarios), and elementary goals for the pilot applications. The elements of the ontology are expressed by the WSMO choreography and orchestration interfaces.

The life events and goals described in the Life events ontology are used in the Personal Assistant client tool. The life events and goals of all the Access-eGov pilot applications specify a process model that will be composed and executed by the inner components of the Personal Assistant client according to the interactions with users.

All three Access-eGov ontologies describe several aspects and levels of the same real world data. All of them denote services (electronic or traditional) and the way of their usage.

B. Formalisms Used

The WSMO conceptual model was adapted and modified to meet the requirements of the life event approach to modelling governmental applications. Thus the resource ontologies were formalised and implemented using the WSML (Web Service Modelling Language) representation. More precisely toolchain for ontology manipulation was designed, consisting of the specialised Annotation Tool and of the third-party WSMO Studio environment. The Annotation Tool was developed as a web application for user-friendly semantic annotation of governmental services. This tool, together with the resource ontologies, was tested by all the public administrations involved in the Access-eGov project.

The conceptual model contains a set of relevant entities - concepts, relations, properties, constraints, etc., that can serve as building blocks for the implementation of the system components as well as for the semantic annotation, i.e. the formal representation of potentially very complex governmental services and their relationships.

The WSMO conceptual model provides following top-level elements: ontologies which provide terminology used by other elements, web services which represent computational entities able to provide access to services, goals which describe aspects related to user's requirements, and mediators which describe elements handle semantic interoperability problems between WSMO elements.

Structural relations between the elements in the proposed conceptual model are depicted in figure 4. The parts reused from the original WSMO model are marked with grey colour.

C. Design of Ontologies

The ontology-like structure needed to be formalised and expressed by WSML statements. It required fixing the meaning of the terms and relations defined in the controlled vocabulary, as well as verifying that the formal meaning reflects the informal description in the glossary.

The concepts and their relations were modelled by the following expressions:

```
concept ConceptName
  relationName RelaedConcept
```

For example, a hierarchy of certificates can be expressed in WSML notation as follows:

```
concept Certificate
  subConceptOf Document
concept Birth_certificate
  subConceptOf Certificate
concept Marriage_certificate
  subConceptOf Certificate
```

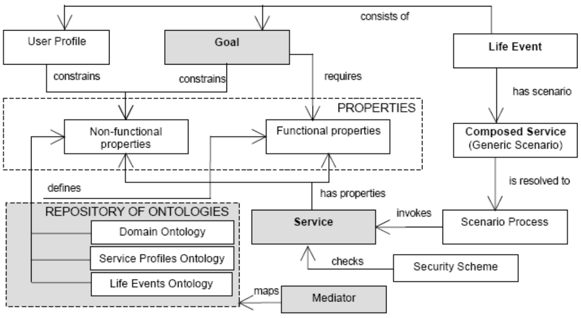

Fig 4: WSMO conceptual model adapted for the Access-eGov system

In addition, the external ontology resources, identified as relevant for the given domain, were used to standardise the ontology structure and to achieve a consistency between the semantic descriptions. The attributes of concepts were modelled as NFPs. Since the attributes are displayed in the client-side tools, they need to be localised to the proper languages. The localised values are modelled by Dublin Core's dc#title statements.

VI. Access-eGov most important facts

Access-eGov provides a specific and new approach to the administrative issues – namely user's point of view. Realisation of such a perspective uses user's goals and life events, which exactly describe user's needs and take into consideration specific user's conditions deriving from user's characteristic context.

Executive scenario is dynamically composed on the basis of user's requirements as well as service capabilities. That is first of all user-friendly, and does not use administration-centric point of view which is not understandable by common users. Moreover, dynamic composition helps to create particular user's path consisted of separated atomic services, which are to be executed via Access-eGov platform in the meaning of web service execution or waiting for user input after realising traditional services has been completed.

Such an approach is directed towards the users – citizens or businesses, to facilitate their interactions with public administration and to make government services interoperable.

Administration-centric approach is comfortable only for institutions, because considers realisation of particular services, and user-centric approach constructs whole, unified processes built from atomic services. Such a process meets the whole path of realisation of particular life event, complying also tasks not related to public administration, but essential to complete user's goal.

The second important Access-eGov value added is usage of WSMO technologies, which are less popular among EU projects. It turns out that WSMO approach of user goals and service capabilities is strictly tailor-made for user-centric approach.

Acknowledgment

The author expresses thanks to Access-eGov consortium for being involved in the Project and consequently having possibility of writing this article using Access-eGov deliverables as sources of knowledge.

References

[1] Access-eGov, “State-of-the-art Report” D2.1, May 2006.

[2] Access-eGov, “Access-eGov Platform Architecture” D3.1, February 2007.

[3] Access-eGov, “Access-eGov Components Functional Descriptions” D3.2, March 2007.

[4] Access-eGov, “Public administration resource ontologies” D7.1, November 2007.

[5] Peristeras, V., Tarabanis, K.: Reengineering the public administration modus operandi through the use of reference domain models and Semantic Web Service technologies. In: *Proceedings of the 2006 AAAI Spring Symposium on The Semantic Web meets eGovernment* (Stanford University, March 27-29, 2006), Technical Report SS-06-06, AAAI Press, Menlo Park, CA, 2006,

[6] Roman, D. et al: D2v1.0. Web Service Modeling Ontology (WSMO). WSMO Working Draft, 20 September 2004. Accessible at http://www.wsmo.org/2004/d2/v1.0/ [Last accessed in October 2007].

[7] Wang, X., Vitvar, T., Peristeras, V., Mocan, A., Goudos S., Tarabanis, K.: WSMO-PA: Formal Specification of Public Administration Service Model on Semantic Web Service Ontology, Hawaii International Conference on System Sciences (HICSS), Waikoloa, Big Island, Hawaii, 2007.