

## Outline of the Access-eGov Architecture

### Marek Skokan, Peter Bednár

Technical University of Kosice  
Letna 9, 042 00 Kosice, Slovakia  
Marek.Skokan@tuke.sk, Peter.Bednar@tuke.sk

### Martin Tomášek

Intersoft, a.s.  
Floriánska 19, 040 01 Košice, Slovakia  
Martin.Tomasek@intersoft.sk

*Abstract: 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). And since we are still far away (especially in the New Member States) from the situation that all needed government services for the given life event are available on-line, it means that users usually have to deal with a combination of traditional services and e-services. The aim of the Access-eGov architecture is to cover both types of services and to provide hybrid scenarios to deal with the real life events (LE). The architecture will also provide an alternative solution by creating a distributed system and by delegating the responsibility for registering a new e-service into the decentralised system and updating information on existing services to local providers (Public Administration institutions).*

*Keywords: Semantic Web Services (SWS), Service Oriented Architecture (SOA), Peer to Peer (P2P), Life event models*

## 1 Introduction

The usability of the Access-eGov system will be proven by providing three scenarios. The Slovakian activity scenario [1] is based on the intention of building a new family house in a municipality of the Košice region. The service described in the Polish scenario is the establishment of an enterprise (starting one’s own business) by the user. The third scenario assumes that a German citizen is currently living in a municipality of Schleswig-Holstein wants to be married with a Slovak citizen.

The general architecture of the Access-eGov system will be based on concepts and methods proposed and implemented in various fields of research. This paper provides an overview of the overall Access-eGov architecture.

## 2 Conceptual View to Architecture

Although the user requirements are very diverse, the basic functionality of the Access-eGov platform is manageable [2]. We need to annotate services and store them in efficient ways. Those services have to be retrieved according to certain citizen requests and the administrators in public authorities ought to have a possibility to string such annotated services together to form new ‘meta services’, our so called scenarios.

### 2.1 User Groups

The first group consists of Information providers. The Information provider view will look at the functionality from the point of view of an administrator in a public authority, while the Information consumer view will provide a citizen’s perspective.

An information provider [2] has three main tasks, namely registering new services, annotating services and building generic scenarios (general goals) relevant to the typical life events of already defined goals:

- **Annotate Service**

The public administration official, who in any, except the most trivial case, also will have to be a domain expert, chooses from a set of available ontologies. He uses the therein contained concepts and relationships to mark up the important aspects of the service or website he is currently describing. The annotating person will only have limited choice of which ontology to use at a given point in time, since this choice is usually imposed by other, for example institutional, constraints.

Apart from this semantic annotation, the annotator also has to describe the security surroundings of a certain service. This will include information about who is eligible to actually use that service, what form of identification is required up to the point of the privacy policies that the service itself can offer.

Finally, the annotator has to add some general descriptions, to facilitate easier administration of the platform. These properties could include contact details for persons that are responsible for the service’s operations and could be part of the non-functional properties.

- **Register Service**

After successful annotation of the service, all generated data is stored; the service can now be retrieved within the whole Access-eGov P2P based infrastructure.

- **Create Generic Scenario**

A far more challenging task for the public administration official is, to build generic scenario based on already pre-defined goals or already existing scenarios. First, the annotator has to either find suitable goals or already existing, suitable generic scenarios. This is done through either searching or browsing the goal and generic scenario repositories.

After having found suitable goals and/or generic scenarios, the annotator has to order them in a reasonable succession that is imposed by internal procedural guidelines and/or legal conventions.

Note that goal is an entity which can be resolved by some elementary service and there might be more services which can fulfill the goal, but each of them can do this.

The second group consists of citizens. The Access-eGov platform appears a black box for this group. A citizen has two main possibilities of interaction with the platform [2]; he has to specify his goal and he has to command the platform to execute the retrieved services:

- **Specify Goal**

The citizen has to articulate wishes to the system and has to tell the actual task that he wants to accomplish. Therefore the personal assistant presents a list of life events or and/or services for browsing and the possibility to search for life events. Once the citizen has chosen one of the offered life events, the personal assistant will translate the selection into a pre-defined goal or LE generic scenario (more such goals composed together into Complex Goal).

It has to be noticed that the life event could either be connected to a pre-defined goal, or to a predefined, generic scenario.

- **Execute Goal/Scenario**

When the user wants to achieve his/her goal, s/he lets the personal assistant start the execution of the retrieved service or workflow. The current progress of this run is always visible to the user through the personal assistant client.

## 2.2 Ontologies

The Access-eGov system will use three basic ontologies in several parts of the Access-eGov system [2]. These are:

- Life events ontology
- Service profiles ontology
- Access-eGov Domain ontology

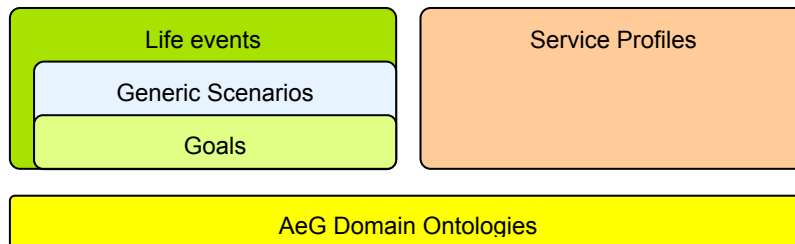


Figure 1

Conceptual data view in Access-eGov

AeG Domain Ontologies are considered lower level ontologies within the overall system. They are used to describe all the relevant domain information related to a user's scenarios. That means they will describe functional and non-functional properties of a particular service. We propose web based ontologies that are not necessarily relevant to the web services. When choosing the other two ontologies mentioned in the figure above, it would be favourable to describe them in a standardized way so that all system partners will be able to process them. A more detailed description will be known according to the Access-eGov project plan in the near future, but the assumed domain ontologies can be found in the following text (ct. 2 Logical architecture).

The other proposed two ontologies in Access-eGov system will be the 'Life events' and 'Service Profiles' ontologies and are used to 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).

Life events denote specific situations (in the life of a citizen or a life cycle of an organization) that require a set of public services to be performed. It contains information about particular 'Goals', which can be grouped into several 'Generic scenarios'.

In general, these three ontologies describe several aspects and levels of the same real world data. All of them denote services (web or traditional) and the way how they are used. Thus we propose to use these three ontologies in order to describe all relevant data inside the Access-eGov system. The domain ontology will be used to describe the lower (i.e. technical) level of the Access-eGov system, whereas the other ontologies will be utilized to denote more abstract system levels just as service description. On the upper most abstract level, sequences of goals -> scenarios -> events can be described. In order to allow interaction and deduction

between the different data models that underlie these ontologies, several layers of mediation will need to be introduced in Access-eGov.

### 3 Logical View to Architecture

#### 3.1 Data Elements

A *Life event* (LE) denotes a specific situation (i.e. event) in the life of a citizen or a life cycle of an organization that requires a set of public services to be performed. Life events can be categorized in groups and may be organized in multiple hierarchies.

A *Generic scenario* specifies generic composition of the goals relevant to the typical complex LE.

A *Goal* specifies those objectives that a client might have when consulting a service, including functionalities that a 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 the service functional properties.

Goals are logically matched against a service's capability, but the data structures are completely independent from one another. They are not directly connected to service- and user-related data.

A *Workflow* specifies composed activities which fulfil Goals in the Generic scenarios. The term activity in the workflow means either electronic service or traditional service. Thus, it is also used the term 'Composed Services' in this paper.

A *Service profile* specifies what the service does provide from a user's perspective and is used by the public administration to advertise services. A 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 the terms constraining type and property values of the various resources required for or provided by the services. Types used to specify functional properties are defined in the domain specific resource ontologies.

Non-functional properties describe the semi-structured information intended for the 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 (i.e. organization structure of the service provider) and general (e-government independent) ontologies (i.e. types used to describe quality-of-service, security or trust).

In the process of the service discovery, functional properties of goals and services are semantically matched by the AeG Discovery module to select services which are able to achieve these goals. Non-functional properties specified by the requester are then used to additionally filter or reorder the discovered services according to a requester's preferences.

There is a mapping between Generic scenarios and Workflows (Composed Services) as well as between Goals and Services. The following figure outline relations between these elements.

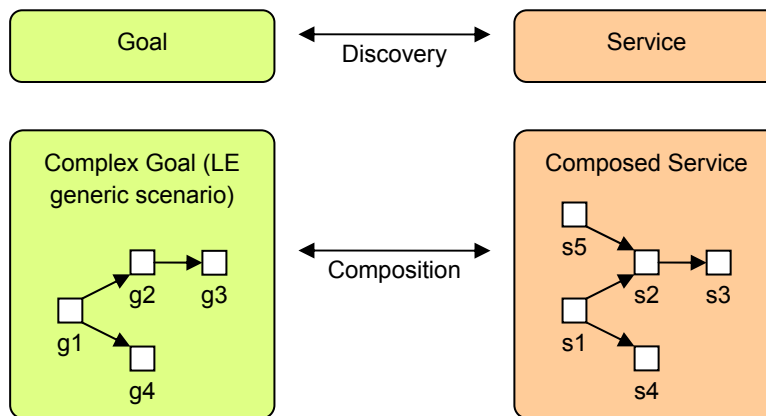


Figure 2  
Mapping between elements

### 3.2 AeG Domain Ontologies

At this stage it is assumed that the following domain ontologies will be needed to describe the concepts for non-functional properties of services for AeG [2]:

- *Fees*: Describes the fee that a citizen has to pay in order to use a service.
- *Forms*: Services may require information and / or they might provide information in the form of documents or forms. The *Forms* ontology will be used to describe these kinds of (both mandatory and optional) input and output of a service.
- *Input and output artefacts*: For inputs and outputs that cannot be described with the *Forms* ontology (for example, an artefact like a license plate), AeG will provide a special ontology that can be used to describe these special kinds of input and output.

- *Administration*: Every service is provided by one or more administrations. At least the following information related to service provision of an administration must be described:
  - *Responsibility*: Two administrations may offer the same service, but only one administration is responsible for providing the service to a particular citizen. The responsibility of an administration can be divided into at least three different kinds: *spatial responsibility* describes the geographical area within which an administration is responsible for providing a certain service, *temporal responsibility* describes the time during which the administration is responsible to provide a service (this may not be confused with the office hours, which is the time the service is provided to citizens), *subject-matter responsibility* describes the professional area of responsibility of an administration (for example, the civil office in Germany is *professionally responsible* for marriages, but not for income taxes). Depending on the legal requirements in effect, which administration is responsible for providing a particular service to a particular citizen can depend on any of the three kinds of responsibility or a combination thereof.
  - *Office hours / availability*: Describes the time when citizens can request the service. This is especially important for the description of traditional (non-electronic services).
  - *Address and contact information*: Describes the address, telephone number and other contact information of the service provider (for electronic services) or the location where the service can be requested (for non-electronic services).
  - *Physical accessibility constraints*: Describes any physical accessibility restrictions (for example, accessible by wheel-chair) that apply to a certain administration (service provision location). Only needed for non-electronic services.

### 3.3 Access-eGov Infrastructure

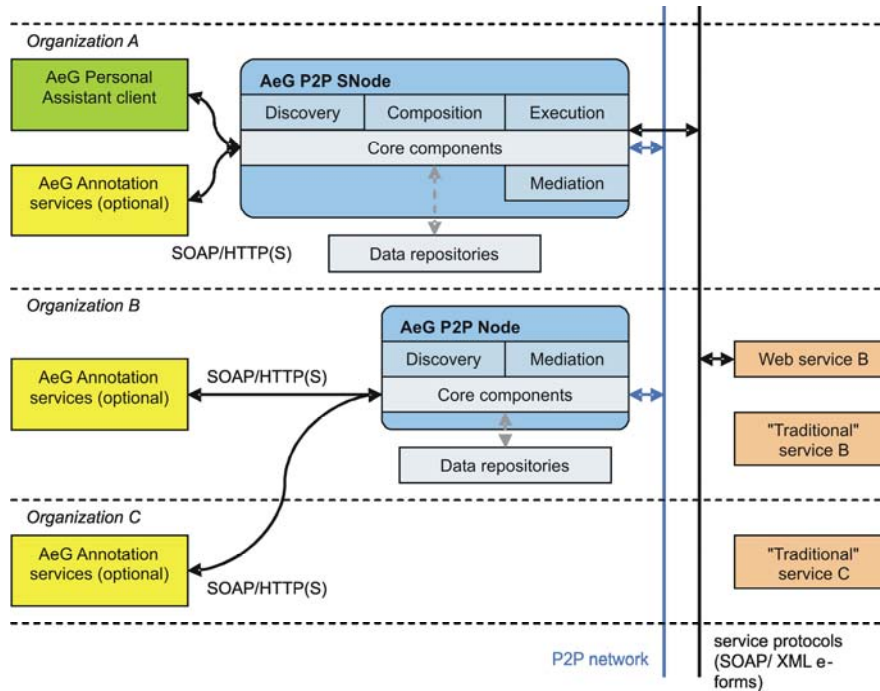


Figure 3  
 Structural view on Access-eGov

The above picture reflects the physical division of the system into several levels [2]. On the one hand, there is the vertical level division. Three vertical levels are depicted:

- user level (on the left side of the scheme),
- system level (in the middle of the scheme) and
- web and traditional services level (on the right side of the scheme).

On the other hand, the horizontal levels, divided by the horizontal dashed lines in the scheme above, represent several possible AeG installations on the public administration premises to access the AeG peer-to-peer-network. There are only three example Organizations, but practically there can be several horizontal levels in the real world applications.

Finally four different types of communication are depicted between the several levels and are represented by arrows in different colours. Communication in this chapter means physical data and information interchange.



### Vertical Levels

Let's explain the vertical levels first. On the left side in the above given scheme, the user level is situated. Two colours are used on this vertical level - green and yellow – to distinguish between the two types of users and user interfaces that are depicted in that level:

- user citizen uses the system via the AeG Personal Assistant client (green boxes).
- public administration, the AeG administrators on the Public Administration site, who annotates the services to be used in the AeG Node (see below, what AeG node means) using AeG Annotation Services interface.

There is another user type that is not shown in the scheme, namely the administrator of the AeG system. But this user will interact with the system only during the installation phase, not within the lifecycle of the system.

The AeG system level is depicted as the blue-coloured middle vertical level of the scheme. The AeG system will be represented by nodes called AeG P2P Nodes. These nodes are connected via a peer-to-peer-network. Each AeG Node itself consists of the modules and components as shown in chapter 4.1.5. According to the functionality needed in a particular AeG P2P node, the actual number of installed optional components may vary. More information on the AeG node functionality will be unveiled in the description of the horizontal levels of the scheme below.

On the right side of the scheme there are the Web and Traditional services. This is the level, where web and traditional services from the provider side are situated.

### Horizontal Levels

As mentioned before, horizontal levels in the scheme above represent particular organizations. Note, that the term particular organisation does not necessarily mean a different organization. It can also represent a department within one organisation. In such a particular organization one AeG P2P node will be set up, which can be administered by this organization, but should cooperate over the P2P network with other AeG nodes installed in different organizations. Every AeG node may lack some functionality components according to organizational needs or restrictions. There are three possible installation types depicted in the scheme.

- 1 Full version of the AeG node installation - all the AeG modules and components are installed.
- 2 Installation without the AeG Personal Assistant. Citizens can use electronic or traditional services described in this AeG node via other (full featured) node (using P2P).

- 3 Only the Personal Assistant is installed (without access to the AeG node). Service provider provided with the AeG Annotation service for annotation of its web site (with description of its traditional services).

Note that there are more possible configurations of AeG platform.

Finally let's describe different communication types proposed in the scheme. Black arrows on the left side between user level and system level are to represent HTTP(S)/SOAP communication. On the right side, blue arrows represent the P2P network used to communicate between the Nodes. A particular AeG Node can act independently of other nodes, but if there is the need of communication between two AeG Nodes (a user connected to one AeG Node needs a web service described at another AeG Node) the network will handle this communication. Black arrows on the right side represent communication between particular nodes and the services (web or traditional). There is another type of communication proposed in the scheme, represented by grey arrows. It is namely the communication between one AeG node and its Data repositories, as the data repositories will represent data in several formats that do not have to be stored on the same physical device as the AeG Node itself.

### Conclusions

Nowadays the situation in e-Government still requires to somehow support traditional services. The solution of life event of citizens as well as businesses can lead to complex scenarios of services. The Access-eGov architecture is proposed with consideration of both of these aspects. In addition, the architecture supports the registration and maintenance of both class of services (traditional and electronic) in the decentralised manner. The general architecture of the Access-eGov system is based on concepts and methods proposed and implemented in various fields of research.

### Acknowledgements

This paper is supported by the project FP6-2004-27020 Access-eGov (Access to e-Government Services Employing Semantic Technologies) and by project Nr. 1/4074/07 Methods for annotation, search, creation, and accessing knowledge employing metadata for semantic description of knowledge.

### References

- [1] User Requirement Analysis & Development/Test Recommendations. Deliverable D2.2, Access-eGov Project, 2006
- [2] Access-eGov Platform Architecture. Deliverable D3.1, Access-eGov Project, 2006