

FP6-2004-27020

Access-eGov

Access to e-Government Services
Employing Semantic Technologies

Instrument: STREP

Thematic Priority:

SO 2.4.13 Strengthening the integration
of the ICT research effort in an enlarged Europe

D7.1: Public administration resource ontologies

Start date of project: January 1, 2006 **Duration:** 36 months

Date of submission: November 30, 2007

Lead contractor for this deliverable: InterSoft, a. s.

Revision: Final 1.0

Dissemination level: PU

Acknowledgement: The Project is funded by European Commission DG INFSO under the IST programme, contract No. FP6-2004-27020.

Disclaimer: The content of this publication is the sole responsibility of the authors, and in no way represents the view of the European Commission or its services.

D7.1: Public administration resource ontologies

Work package: WP7

Task: T7.1, T7.3

Date of submission: November 30, 2007

Lead contractor for this deliverable: Intersoft (IS)

Authors: Ralf Klischewski (GUC)
Stefan Ukena (GUC)
Karol Furdik (IS)
Andrzej Marciniak (EMA)
Jan Hreno (TUK)
Marek Skokan (TUK)
Contributions from all partners

Version: 1.0

Revision: Final

Dissemination level: PU

Project partners:

Technical University of Kosice (TUK), Slovakia (Coordinator); University of Regensburg (UR), Germany; German University in Cairo (GUC), Egypt; Intersoft, a.s. (IS), Slovakia; EMAX S.A. (EMA), Poland; Kosice Self-Governing Region (KSR), Slovakia; Cities on Internet Association (COI), Poland; e-ISOTIS (ISO), Greece; Municipality of Michalovce (MI), Slovakia; City Hall of Gliwice (GLI), Poland; State Government of Schleswig-Holstein (SHG), Germany.

Abstract:

This deliverable reports on the process and results of modelling public administration resource ontologies. The ontologies were designed for semantic annotation of the governmental services provided within the scope of Access-eGov pilot applications. The requirement-driven approach was formulated as a 7-step procedure to obtain the ontology description from user partners. Existing ontology resources, publicly available from other projects, were reused after a detailed analysis. The Access-eGov resource ontologies were formalised and implemented using the WSMML representation. In order to utilise the developed resource ontologies, an annotation tool was designed and implemented as a web application for semantic description of provided governmental services. The tool was then successfully used by user partners to test its functionality as well as to provide the data for implementation of other Access-eGov system components and tools.

Document Sign-off

Nature of Sign-off	Name	Role	Participant short name	Date
Reviewed	Karol Furdik	TL	IS	28/11/2007
Approved	Karol Furdik	WPL	IS	28/11/2007
Submitted	Tomas Sabol	PM	TUK	30/11/2007

Document Change Record

Date	Version	Contributor(s)	Change Details
July 15, 2007	0.1	Stefan Ukena (GUC)	The first draft
July 17, 2007	0.2	Stefan Ukena (GUC)	Glossary updated
July 31, 2007	0.3	Ralf Klischewski, Stefan Ukena (GUC)	Chapters “Motivation and methodology” and “Ontology modelling within AeG” updated
Sept 19, 2007	0.4	Andrzej Marciniak (EMA)	Section “Use of classes in PAC by Citizen” updated
Oct 29, 2007	0.5	Karol Furdik (IS)	Renumbering of chapters. Chapters 2, 3, 4 reviewed and updated
Nov 2, 2007	0.6	Karol Furdik (IS), Jan Hreno (TUK)	Chapter “Semantic annotation of services” updated, Execution summary and Introduction added.
Nov 18, 2007	0.7	Karol Furdik (IS)	Chapter 4 updated, all the chapters reviewed, updated, and finalised for internal review.
Nov 23, 2007	0.8	Marian Mach (TUK), Maren Kleimann (SHG)	Comments and updates, internal review
Nov 26, 2007	0.9	Stefan Ukena (GUC), Marek Skokan (TUK)	Inputs and updates in sections 2.1, 2.4, 3.1, and 3.7.
Nov 30, 2007	1.0	Karol Furdik (IS)	Modifications according to the comments from internal review, finalisation.

Files

Software Products	User files / URL
Microsoft WORD	D7.1_PA_ontologies.doc
Microsoft EXCEL	D7.1_PA_ontoGlossary.xls
WSMO Studio http://www.wsmostudio.org	D7.1_PA_ontologies_impl.zip

Table of contents

EXECUTIVE SUMMARY	6
1 INTRODUCTION.....	7
1.1 OBJECTIVES AND SCOPE	7
1.2 DOCUMENT STRUCTURE.....	7
1.3 CHANGES TO PREVIOUS WORK	8
1.4 USED ABBREVIATIONS	8
2 MOTIVATION AND METHODOLOGY.....	10
2.1 NEED FOR RESOURCE ONTOLOGIES WITHIN ACCESS-eGOV	10
2.2 ADAPTATION OF THE WSMO CONCEPTUAL MODEL.....	11
2.3 STATE OF THE ART. EXISTING ONTOLOGY RESOURCES	14
2.3.1 <i>Metadata and document standards</i>	14
2.3.2 <i>Resource ontologies for eGovernment</i>	14
2.3.3 <i>Reference ontologies accompanying the tools</i>	15
2.3.4 <i>Reusing available resources within the Access-eGov system</i>	16
2.4 REQUIREMENT-DRIVEN APPROACH TOWARDS ONTOLOGY MODELLING	19
3 ONTOLOGY MODELLING WITHIN ACCESS-EGOV	21
3.1 IDENTIFICATION OF INFORMATION NEEDS	21
3.2 IDENTIFICATION OF REQUIRED INFORMATION QUALITY.....	22
3.3 CREATION OF A GLOSSARY OF TOPICS AND TERMS	22
3.4 CREATION OF CONTROLLED VOCABULARY	23
3.5 GROUPING AND RELATING THE IDENTIFIED ITEMS.....	26
3.6 DESIGN OF ONTOLOGY	28
3.7 IMPLEMENTATION OF SEMANTICS	29
3.8 VERIFICATION OF THE DESIGNED ONTOLOGY ON REAL DATA	32
4 ONTOLOGY IMPLEMENTATION.....	35
4.1 AEG CORE ONTOLOGY: SERVICE PROFILES	35
4.1.1 <i>SKOS ontology</i>	36
4.1.2 <i>Non-functional properties for visualisation</i>	36
4.1.3 <i>Service Profiles ontology: CoreAegConcepts</i>	37
4.1.4 <i>Modified Service Profiles ontology for SHG-GE pilot application</i>	38
4.2 LIFE EVENTS ONTOLOGY	40
4.2.1 <i>SHG-GE pilot: Marriage life event</i>	40
4.2.2 <i>GLI-PL pilot: Establish Enterprise life event</i>	41
4.2.3 <i>KSR-SK pilot: Build a House live event</i>	42
4.3 DOMAIN ONTOLOGIES	46
4.3.1 <i>SHG-GE pilot: SHGOntology</i>	46
4.3.2 <i>GLI-PL pilot: GLIOntology</i>	48
4.3.3 <i>KSR-SK pilot: KSROntology</i>	52
4.4 CREATING INSTANCES	54
4.4.1 <i>Class hierarchy view</i>	54
4.4.2 <i>Class view by usage</i>	56
4.5 APPLICATION OF RESOURCE ONTOLOGIES	58
4.5.1 <i>Use of classes in Annotation tool by Annotation Author</i>	58
4.5.2 <i>Use of classes in Personal Assistant by Citizen</i>	59

5	SEMANTIC ANNOTATION OF SERVICES.....	61
5.1	FUNCTIONALITY AND USER ROLES FOR THE SEMANTIC ANNOTATION.....	61
5.2	ANNOTATION TOOL.....	62
6	CONCLUSION AND OUTLOOK.....	65
	REFERENCES.....	67

Executive Summary

This deliverable contains a detailed description of resource ontologies for the public administration domain, as they were designed for semantic annotation of the governmental services provided within the scope of the Access-eGov pilot applications. Based on the guidelines for semantic mark-up [D7.2], where the semantic annotation process was described from the perspective of information providers (i.e. the public administrations of Access-eGov user partners), the process of ontology design, development, and implementation is presented here from the information consumers' perspective. Following the design of the overall architecture [D3.1] and particular components [D3.2], the ontologies are proposed as the specification of system data for the services and workflow structures within the Access-eGov system.

The WSMO (Web Service Modelling Ontology) conceptual model was adapted and modified to meet the requirements of the life event approach to modelling governmental applications. Available ontology resources from other projects were investigated and analysed to reuse them for the purposes of the Access-eGov project. The requirement-driven approach was formulated and adopted to collect, relate, and formally express the information needs of the service providers, i.e. particular public administrations, to design the specific semantic structures and descriptions of provided governmental services. The identification of informational needs has been accomplished through requirement analysis and user partner scenarios. The core ontology structure was designed in a 7-step procedure and the resulting model was verified on the real-world data provided by user partners. In a tight co-operation of developer and user partners, the resource ontologies were formalised and implemented using the WSML (Web Service Modelling Language) representation.

An annotation tool was designed and implemented as a web-based application that enables to semantically annotate governmental services in a user-friendly way. The first version of the annotation tool was used also as a mock-up to achieve a common understanding of particular properties of the services and to obtain comments and suggestions from user partners concerning the provided functionality. Based on these negotiations, the ontology was refined and the second version of the annotation tool was implemented. This version of the annotation tool was then successfully used for the training of annotation authors from the public administrations of the Access-eGov user partners. During the training, the concrete data (i.e. semantically annotated governmental services) were provided by user partners and these data were stored into the repository as instances of the designed resource ontologies. These resource ontologies together with the instances will be used as a basic data structure for the testing and enhancements of other Access-eGov components [D4.1] as well as for the implementation and testing of the Personal Assistant client [D5.1].

1 Introduction

This deliverable covers the work done within the tasks 7.1 and 7.3 of the workpackage WP7. It describes the process of design and implementation of the Access-eGov resource ontologies, i.e. the conceptual knowledge models required by platform components, as well as by pilot applications. Based on detail analysis and reusing of existing ontology resources, public administration resource ontologies were created in co-operation of user and developer project partners, employing the requirement-driven approach of ontology design. Toolchain for ontology manipulation was designed, consisting of the specialised Anntotation tool and of the third-party WSMO Studio environment¹. The Anntotation 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 complexity of the process of ontology creation, but mainly the amount of implementation work done within the tasks 7.1 and 7.3 are the reasons for the delayed release of this deliverable.

1.1 Objectives and scope

This deliverable can be seen as an interface between the technical infrastructure design and the usage of the whole Access-eGov system within the pilot applications. By means of implementation of the Access-eGov resource ontologies, this document provides a specification of the inner data structure for client-side tools (i.e. the Annotation tool and Personal Assistant client [D5.1]) as well as for the system components [D3.2]. It also provides a framework for semantic annotation of the governmental services, described from the perspective of public administrations in [D7.2].

The specification of resource ontologies (chapter 4) is especially useful for developers of the Personal Assistant client. It should enable to implement and test the tool on the conceptual ontology models designed and produced in co-operation with user partners according to the scenarios defined for pilot applications.

The Anntotation tool was designed and implemented within the scope of this deliverable (section 5.2). This part can be useful for user partners (public administrations), since they will use this tool to annotate and publish the provided governmental services.

1.2 Document structure

The document shortly motivates the needs for ontologies and identifies their purpose within the Access-eGov system. The semantic interoperability is emphasised in chapter 2 as the main advantage of the ontology-driven approach. The methodological issues and various resources for ontology design are also discussed in this chapter. The WSMO conceptual model [WSMO] is adapted according to the requirements given by the architecture design [D3.1], namely by the life event approach taken for modelling the governmental services. Existing standards and ontology resources are investigated and analysed for reuse within Access-eGov. Finally, the requirement-driven approach is described as a main methodology and method for collecting the information needs from user partners and using them as input for formalising semantic structures which are then implemented as ontologies.

¹ <http://www.wsmostudio.org>

Chapter 3 describes how the modelling of public administration resource ontologies has been carried out within the Access-eGov project. Following the requirement-driven approach [Klischewski & Ukena 2007], the seven steps of the modelling process are described in detail. The designed ontology structure is verified and modified, based on real-world data provided by one of the user partners.

The ontology implementation into the WSML representation is described in chapter 4. The creation of instances as well as a proposal for applying the resource ontologies within the Access-eGov components and tools are also presented here.

The semantic annotation of governmental services by means of the Annotation tool is described in chapter 5. The user interface and the provided functionality of the web-based annotation tool is presented within this chapter.

A short conclusion and proposal of future work finishes this report in chapter 6.

The document is accompanied with the two attachments, namely:

1. Access-eGov resource ontologies, implemented in the WSML formalism;
2. Glossary of topics and terms (in Microsoft Excel format).

1.3 Changes to previous work

No significant changes to the information published in previous deliverables are presented in this document. The syntax for describing the activities in the orchestration is slightly different from the one published in [D5.1]. The syntax was modified due to the enhancements of the structure of the Access-eGov orchestration and choreography process model, as it is described in the section 3.7.

Some changes were made in comparison to the process of semantic annotation, as it was described in [D7.2]. In this document, the template mechanism is proposed for enriching the annotated service by its process model (conf. a full control and specification of all the elements of process model during the annotation, as it was proposed in [D7.2]). Another modification of the proposal in [D7.2] is a reduction of user roles for annotation, described in the section 5.1. This approach corresponds to the proposal for functionality of Ontology management component [D4.1], where the ontology manipulation is handled by experts, using a third-party ontology editor (e.g. WSMO Studio environment). To semantically describe particular services, which means to specify non-functional properties and to select proper service template as a pre-defined identifier of a process model, the Annotation tool is described in the section 5.2 as a web-based client application.

1.4 Used abbreviations

API	–	Application Programming Interface
BNF	–	Backus - Naur form
DAML	–	DARPA Agent Markup Language
DARPA	–	Defense Advanced Research Projects Agency
DERI	–	Digital Enterprise Research Institute (www.deri.at)

GEA	–	Governance Enterprise Architecture
GLI-PL	–	Gliwice (Poland) pilot application
GUC	–	German University in Cairo
KSR-SK	–	Kosice self-governing region (Slovakia) pilot application
NFP	–	Non-functional property
OWL	–	Web Ontology Language
OWL-S	–	Web Ontology Language Schema; ontology built on top of OWL
PAC	–	Personal Assistant client tool
SOA	–	Service Oriented Architecture
SHG-GE	–	Schleswig-Holstein (Germany) pilot application
SKOS	–	Simple Knowledge Organisation Systems
SGS	–	Semantic governmental services
TOC	–	Table of contents
UML	–	Unified Modelling Language
URL	–	Uniform Resource Locator
WSML	–	Web Service Modelling Language
WSMO	–	Web Service Modelling Ontology
XHTML	–	Extended HyperText Markup Language

2 Motivation and methodology

This chapter describes how the modelling of the public administration resource ontologies has been carried out within the project. Section 2.1 identifies the needs for resource ontologies within the Access-eGov system and presents the requirements towards the ontology design. Three sources were employed in the process of resource ontology construction. The first source is the WSMO as the implementation platform selected for the Access-eGov system. Adaptation of the WSMO conceptual model according to the life-event approach for eGovernment applications is described in section 2.2. The second source is a suite of existing publicly available ontologies developed within other projects; section 2.3 reviews the state of the art and outlines the ontologies identified as candidates for reuse. The third source, described in section 2.4, is the requirement-driven approach – a systematic way of building ontologies from explicitly stated information needs of users.

2.1 Need for resource ontologies within Access-eGov

Ontologies are powerful knowledge representation formalism for modelling real-world concepts (e.g. objects, procedures) together with their mutual relationships. An ontology-driven approach enables to formally represent the concepts within a domain of interest in a way that is defined and agreed upon by communities of users, e.g. within collaborating organisations [Furdik et al 2007]. By means of semantic annotation, i.e. a description of the resources (documents, services, web sites, etc.) by proper concepts from shared ontologies, this approach allows an integration of existing (and future) systems and services, its functional interconnection on technical, semantic, as well as organisational levels [EC 2003]. This is a promising way in which ontologies, used for semantic annotation of resources, can increase the semantic interoperability of governmental services provided by public administrations [EC 2006].

The Access-eGov system is aiming to provide a way to integrate the existing governmental services – both electronic as well as the traditional non-electronic ones – by means of their semantic description. This requires the development of resource ontologies consisting of a set of common concepts that are general enough for the description of the full functionality of these services. Because governmental services will be modelled as a workflow within Access-eGov [D3.1], the ontologies must additionally allow the formal description of such a workflow, i.e. they should be used as building blocks in sequences, loops, conditions, and other workflow constructs.

For the Access-eGov system, ontologies were chosen as the knowledge representation mechanism for semantic description of the identified concepts: life events, goals, services, etc., as well as for the repository of data structures for particular components, modules, and tools [D3.2].

To summarise, the resource ontologies are needed within the Access-eGov system for:

- providing a common *conceptual model* as a common semantic basis for formalising key concepts for implementing the Access-eGov components [D3.2];
- providing a controlled vocabulary for the *annotation* of services and resources [D7.2];
- enabling *semantic computation* (matching, composition, retrieval, execution) for the software tools on a client side (e.g. the AeG Personal Assistant client [D3.1], [D5.1]).

When designing an ontology structure according to the specified needs, the following constraints and requirements should be considered:

- Already existing ontology resources should be reused to follow the widely accepted standards and to avoid unnecessary double work.
- Requirements for particular pilot applications should be collected from user partners in a systematic way; the ontology design should be a collaborative work of developer partners and user partners of the Access-eGov project. The design and implementation of the resource ontologies should be verified on real data provided by the user partners.
- To a lesser extend the technical constraints of the chosen technology needs to be considered as well. For the Access-eGov system WSMO [WSMO] was chosen for this purpose.

These requirements will be discussed in more detail in the following subsections to give an overview of the rather complex process of resource ontology design and development.

2.2 Adaptation of the WSMO conceptual model

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 conceptual model is significantly determined by a technology applied for the semantic annotation.

In the Access-eGov project, we decided, after detailed survey and analysis of existing approaches [D3.1], to apply the WSMO as a basic conceptual framework and implementation platform. One of the reasons for selecting the WSMO as the most suitable candidate was that 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 [D3.1], [Skokan & Bednar 2006].

The WSMO conceptual model provides the following top-level elements [Wang et al 2007]:

1. *Ontologies* provide terminology used by other elements of the conceptual model to describe the relevant aspects of the domains of discourse. In other words, ontology is a formal, explicit specification of a shared conceptualisation for a particular domain [D3.1].
2. *Web services* represent computational entities able to provide access to services that, in turn, provide some value in a domain. Properties of a web service are described using the terminology defined by ontologies.
3. *Goals* describe aspects related to user requirements with respect to the requested functionality. Again, ontologies are employed to define the domain terminology, which is then used for the description of the relevant aspects of the goals.
4. *Mediators* describe elements that handle semantic interoperability problems between WSMO elements, especially resolving mismatches between different terminologies used (data level), in communicating between web services (data and process level), and on the level of combining web services (data, process and functional level).

However, these elements serve for the semantic description of general web services only. The governmental services can also be modelled as web services, especially if the eGovernment application, as e.g. the Access-eGov system, adopts principles of service-oriented architecture [D3.1]. In this case, the conceptual model of WSMO needs to be enhanced according to the specific characteristics that can be identified for governmental services, both traditional and electronic ones.

Specific characteristics of governmental services that make the application of semantic technologies in eGovernment challenging were outlined in [Wang et al 2007]. The huge, diverged and distributed environment of public administrations layered in hierarchically organised levels causes difficulties when applying and testing semantic technologies and solutions in practice. Public administrations currently offer many heterogeneous services distributed in a large amount of partially independent agencies. Semantic interoperability of these services can be achieved by annotation and consequent composition of the services to the coherent units that are understandable for service consumers - citizens and businesses.

One solution can be a *life event approach* [Gugliotta et al 2006], where the life event concept plays a central role in the conceptual model², being a formal representation of the user's point of view, of his/her needs and requirements. The life event approach enables to define and formally describe complex scenarios of the execution of governmental services. The life events are formal models of the user's needs, consisting of multiple goals and particular services organised into workflow structures by means of preconditions, required inputs and outputs.

Based on the life event approach, the WSMO conceptual model updated for eGovernment applications will contain the following top-level elements [D3.1], [Skokan & Bednar 2006]:

- *Life Events* are the basic objects from the service consumer's perspective. A life event denotes a specific situation in the life of a citizen or a business episode of an organisation (e.g. establishing an enterprise, building a house, etc.) that requires a set of governmental services to be performed to achieve some goals (e.g. to obtain a permission for building a house in the given territory, etc.). A life event may have assigned multiple goals, formalising user needs. For complex and typical life events, the goals can be composed to *generic scenarios* (complex goals). The goal of a life event can be defined by specific optional preconditions, which allow the customisation of this life event. The preconditions are specified as logical expressions with input variables defined either explicitly by the user or retrieved from the user profile (see description of the User and profile management component in [D5.1]).
- *Services* are the basic objects from the service provider's perspective. Properties of a service are formally described in the corresponding *Service Profile*. A service profile consists of *non-functional* and *functional* properties [D3.1].

Functional properties specify input preconditions, output postconditions and effects of a service. They are expressed in WSMML logical expressions. The types used for the specification of functional properties are defined in the domain-specific resource ontologies.

Non-functional (N-F) properties are semantic descriptions for a particular instance of a service. They describe semi-structured information intended for citizens for service

² Contrary to the life events, a service-oriented approach was adopted in several older eGovernment solutions, where the service provided by public administrations was used as a central concept. For example, in the eGOV and OntoGov projects, the user's point of view is represented by a taxonomy that determines a distribution of the services in the web portal [Gugliotta et al 2006].

discovery, e.g. service name, description, information about the service provider and properties which incorporate further requirements for service capability (e.g. office hours, office location, accessibility, etc.). A set of available N-F properties can be obtained from the domain ontology.

- *Goals* are descriptions of the requirements that a user (service consumer – citizen or business) might have when invoking a service, including the requested outputs, effects, and functionality that a service should provide from the user's perspective. The formal expression of a goal consists of the functional properties required to achieve this goal and the non-functional properties that additionally constrain the candidate services.
- *Ontologies* provide a set of generic concepts used by other elements of the conceptual model. The following types of ontologies were defined in [D3.1]:
 - *Life events ontology* contains semantic descriptions of possible life events, including its goals and generic scenarios;
 - *Service profiles ontology* describes atomic services by means of functional and non-functional properties for a particular service. The ontology contains information on fees, forms, input and output artefacts, responsibility for the service, availability (e.g. opening hours of the office), address and contact information, and physical accessibility constraints.
 - *Domain ontology* is used to represent all the relevant information related to the domain of government, including eGovernment concepts. It covers such non-functional properties as the general and organisational structure of public administrations, concepts describing the quality of a service, security or trust, as well as concepts related to user management and profiling.

Structural relations between the elements in the proposed conceptual model are depicted in Figure 1. The parts reused from the original WSMO model are marked with gray background.

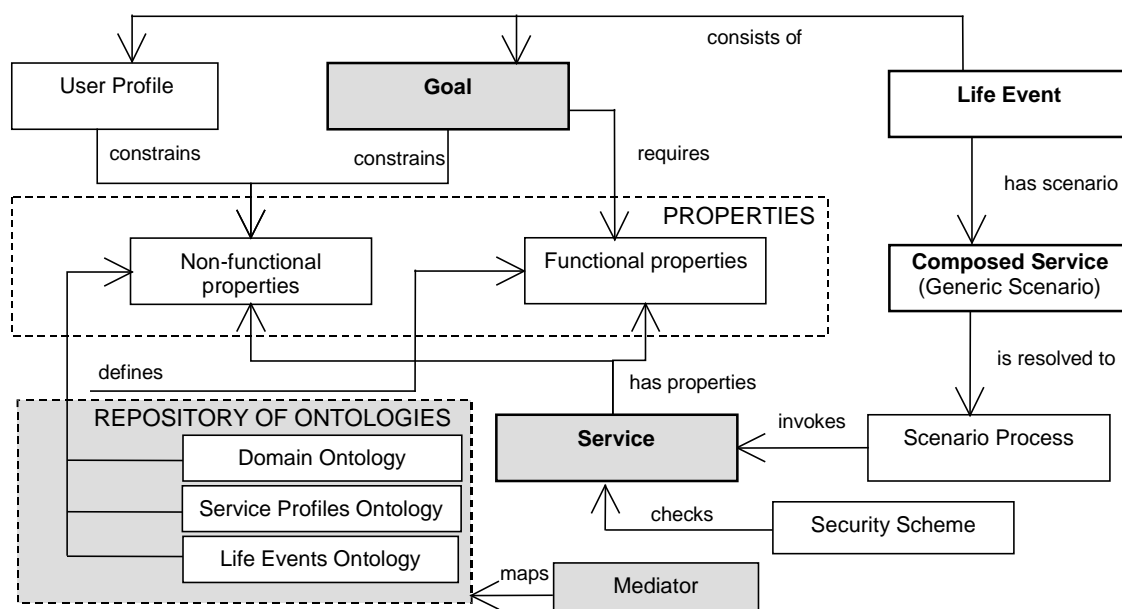


Figure 1. WSMO conceptual model adapted for the Access-eGov system

2.3 State of the art. Existing ontology resources

The number of available ontology resources for the eGovernment domain increased significantly the last three years³. Nowadays, a quite rich set of standards and tools, as well as specific ontologies developed for the eGovernment domain is available and was applied in a number of solutions. Substantial effort of modelling public administration resource ontologies has been made within the frame of the R&D projects as e.g. SemanticGov, TerreGov, OntoGov, SmartGov, eGOV, etc. For the purposes of the Access-eGov project, we have analysed several representative projects, resources, and approaches that provide potentially reusable ontology resources.

The following three subsections, divided according to the types of resources, contain a list of existing ontology resources that have been analysed for reuse within the Access-eGov project. The subsections are divided according to the types of resources to a) widely accepted metadata standards, b) resource ontologies developed within R&D projects, and c) the ontologies that accompany various tool for ontology editing (as e.g. reference implementations of given technology). Finally, the fourth subsection contains a table describing which of these ontologies were (at least partly) reused for the construction of the public administration resource ontologies within the Access-eGov project.

2.3.1 Metadata and document standards

DublinCore metadata element set (<http://dublincore.org/documents/dces/>) is a vocabulary of fifteen basic properties (e.g. as *creator*, *date*, *format*, *identifier*, *title*, etc.) for use in the semantic description of various resources.

SKOS (Simple Knowledge Organisation Systems) Core Vocabulary (<http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20051102/>) is a vocabulary for expressing the basic structure and content of concept schemes as controlled vocabularies, taxonomies, ontologies, etc.

vCard ontology (<http://www.w3.org/2006/vcard/>) contains definitions of personal and organizational data, specifically how they are used in electronic business cards.

XHTML namespace (<http://www.w3.org/1999/xhtml/>) is the W3C standard for XHTML (Extended HyperText Markup Language) document types. It contains specifications of modules (as e.g. *body*, *head*, *html*, *form*, *table*, *td*, etc.) and their properties that define the obligatory structure of the XHTML document types.

2.3.2 Resource ontologies for eGovernment

SemanticGov (<http://www.semantic-gov.org>) provides the ontologies based on the Governance Enterprise Architecture (GEA) [Peristeras & Tarabanis 2004]. The service model ontology defined by WSMO was combined with public administration domain descriptions, models, and concepts based on the GEA [Peristeras & Tarabanis 2006], [Wang et al 2007].

TerreGov (<http://www.terregov.eupm.net>) provides a single multipurpose ontology, formalised in OWL, for modelling the domain of the public administration activities

³ cf. [DIP D9.3], where the application of ontologies in the eGovernment environment is described as „immature field“ with „only a few isolated examples of eGovernment ontologies“ available in the year 2004.

[TerreGov D1.7]. The ontology contains, for example, the concepts for description of location and structure of public administrations, addresses and contact information, services of social care, etc.

OntoGov (<http://www.ontogov.com>) provides a set of ontologies for modelling eGovernment services [Stojanovic et al 2004a], [Stojanovic et al 2004b], [Apostolou et al 2005]. *Meta Ontology* contains entities needed to describe services, *Legal Ontology* describes the structure of the legal documents, *Domain Ontology* contains domain specific knowledge, *Service Ontology* describes a concrete service, *Lifecycle ontology* describes the information flow and the decision making process in the public administration, and *Web Service Orchestration Ontology* allows binding of services during execution [OntoGov D2].

SmartGov (<http://www.smartgov-project.org>) provides the *eGovernment Services ontology* for modelling and conceptual description of governmental services [Fraser et al 2003], [OntoGov D2]. The SmartGov eGovernment Services ontology contains 150 terms that describe e-government services in general, as e.g. *Citizen*, *Contact*, *Form*, *Letter*, etc. These terms were enhanced by the concepts such as *Activity*, *Actor*, *Legislation*, *Needs*, etc., taken from the Enterprise ontology [Uschold et al 1998].

eGOV (<http://www.egov-project.org>) does not provide the ontology itself, however, it defines the metadata standard that can be extended into an ontology for the semantic description of the eGovernment services [OntoGov D2]. In particular, the eGOV approach provides the *GovML* language [Kavadias & Tambouris 2003] as a common format for the data flow between the eGovernment portal and the service repositories in different public authorities.

DIP (<http://dip.semanticweb.org>) provides the ontology created from the *seamlessUK* taxonomy [DIP D9.3]. The ontology contains the concepts for the description of the structure and types of public administrations.

IPSV (http://www.esd.org.uk/standards/ipsv_internalvocabulary/), i.e. the Integrated Public Sector Vocabulary is an encoding scheme (taxonomy) for populating the Subject element of the eGovernment Metadata Standard⁴.

Other projects surveyed within the Access-eGov can be found on the Access-eGov web site, in the section Resources → Related projects⁵.

2.3.3 Reference ontologies accompanying the tools

WSMO ontologies (http://www.wsmo.org/WSMO_ontologies.html):

- *Date and Time Ontology* - defines a general model for specifying time and dates and relationships of them.
- *Location Ontology* - describes locations (such as continents, countries and cities and their interrelation).

WSMO provides also other ontologies (as e.g. *Purchase Ontology*, *Train Connection Ontology*, *Amazon ECS*, etc.), that seem to be of less importance for the purposes of designing the Access-eGov resource ontologies.

Protege ontology library

(<http://protege.cim3.net/cgi-bin/wiki.pl?ProtegeOntologiesLibrary>):

⁴ <http://www.esd.org.uk/standards/egms/>

⁵ <http://www.accessegov.org/acegov/web/uk/index.jsp?id=50031>

- *IPTC* Subject Reference System (<http://nets.ii.uam.es/neptuno/iptc/>); suitable for documents as news / newspapers.
- *REA* Resource-Event-Agent-Enterprise for eBusiness frameworks (<http://www.getopt.org/ecimf/contrib/onto/REA/index.html>).
- *Countries.owl*. The ISO 3166 Code List of countries (<http://www.bpiresearch.com/BPMO/2004/03/03/cdl/Countries>).
- *Family.swrl.owl*. A SWRL/OWL demo ontology about family relationships (<http://protege.cim3.net/file/pub/ontologies/family.swrl.owl/family.swrl.owl>).
- *generations.owl*. An ontology about family relationships that demonstrates classification (<http://protege.cim3.net/file/pub/ontologies/generations/generations.owl>).
- *Geographic Information Metadata*. ISO 19155 (<http://loki.cae.drexel.edu/~wbs/ontology/iso-19115.htm>).
- *OntoTool*. An ontology that is used to model Task and Tool features for project realization. This ontology contains essential concepts about the relationship between task and tool concepts and frequently asked questions about tool identification. (<http://www.ecolleg.org/trms/ontology.html>).

DAML ontology library (<http://www.daml.org/ontologies/>):

The two DAML ontologies refer to governmental concepts:

- *Government R&D* ontology (<http://www.daml.org/projects/integration/projects-20010811>) for the description of organisations and individuals participating in a government R&D program.
- *Government type* ontology (<http://reliant.teknowledge.com/DAML/Government.owl>) of governmental concepts used in the CIA World Fact Book 2002⁶.

DERI ontologies (<http://www.deri.at/digital-library/browse/ontologies/>):

- *e-Tourism Ontology* (<http://e-tourism.deri.at/>) describes the domain of tourism, including concepts for geographic location, properties of various organizations, etc.
- *MarcOnt Ontology* (<http://www.marcont.org/>) for description of library resources.
- *Semantic Web Portal Ontology* (<http://sw-portal.deri.org/>) is aimed to serve as the conceptual backbone for community portals driven by Semantic Web technologies.
- *X12 EDI Ontology* (<http://www.wsmo.org/TR/d27/v0.1/20050621/>) describes the Electronic Data Interchange standard.

2.3.4 Reusing available resources within the Access-eGov system

After the detailed analysis of the ontology resources mentioned above, the following ontologies were identified as best candidates for reuse:

- *Dublin Core*, *SKOS*, *vCard*, and *XHTML* as standards for the definition of concepts, for the description of their properties, for addresses and personal data.
- *SemanticGov*, *TerreGov*, and *OntoGov* ontologies for the modelling the process model of governmental services, for description of properties and structure of public administrations.
- *WSMO* and *Protege* ontologies for description of the properties of governmental services.

⁶ <https://www.cia.gov/library/publications/the-world-factbook/index.html>

The following table gives an overview of the ontology resources reused within the Access-eGov during the design and creation of the public administration resource ontologies. The reused resources are marked by bold font.

	Ontology resource	Reusing in the Access-eGov what was/was not reused and why
REUSED	Dublin Core	General types for <i>title</i> , <i>description</i> , <i>relation</i> were reused for attributes of concepts, especially for non-functional properties.
	SKOS	The SKOS Core Vocabulary was reused for definitions of <i>concepts</i> and <i>properties</i> as the top-most components of the Access-eGov ontology.
	vCard	General types for <i>address</i> , <i>contact information</i> , and <i>personal attributes</i> were reused. The concept <i>v#Address</i> was derived from the vCard ontology.
	XHTML	The <i>href</i> attribute was reused for the <i>Link</i> concept.
	SemanticGov	The GEA <i>Service model</i> for public administrations was adapted and reused for the purposes of the Access-eGov.
	TerreGov	The concepts for description of contact information, location, and structure of public administrations were adapted and reused.
REUSED	OntoGov	Parts of the <i>Service Ontology</i> and <i>Web Service Orchestration Ontology</i> were reused within the Access-eGov for design of entities describing governmental services and their process model.
	SmartGov	The <i>eGovernment Services ontology</i> was not reused within the Access-eGov directly, since it (more or less) duplicates the <i>GEA Service model</i> provided by SemanticGov approach.
	eGOV	The <i>GovML</i> language was not used within the Access-eGov, since the WSMO/WSML approach was taken as the implementation platform. However, the eGOV metadata standard was taken into consideration during the design of annotation components.
	DIP	The ontology is organisation-centred, and this design does not match with the Life Event approach adapted for the Access-eGov. This is the reason why the DIP eGovernment ontology was not reused.
	IPSV	The taxonomy contains terms that are specific to the governmental system in the United Kingdom. This was the reason why the IPSV was not reused within the Access-eGov; however, some terms were taken as naming conventions for English class names in the Access-eGov resource ontologies.
REUSED	WSMO	The <i>Date and Time</i> and <i>Location</i> ontologies were reused to describe the non-functional properties of services within the Access-eGov.
	Protege	The <i>Countries</i> and <i>Geographic Information Metadata</i> ontologies were partly reused for modeling the location properties of the services within the Access-eGov.

DAML	The DAML ontologies were not used within the Access-eGov. Few concepts from the <i>Government R&D</i> ontology were considered for modeling the properties of public administrations and the location properties of services within the Access-eGov.
DERI	The DERI ontologies were not used within the Access-eGov. Fragments of the <i>MarcOnt Ontology</i> were taken into consideration during the design of concepts for documents (inputs/outputs of governmental services) and their properties.

The listed ontology resources were used in Access-eGov to produce some fragments of the whole ontology structure, mostly the definitions of non-functional properties for services. The following example presents an implementation of the vCard ontology for WSMML 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
```

To summarise, the existing ontology resources were reused for two main purposes: 1) to reduce an amount of work needed for ontology development and 2) to keep the Access-eGov solution compatible with standards and approaches adopted in other similar projects and applications. In the *AeG Core ontology* (see section 3.1), about 60% of concepts and attributes were reused from the existing resources and standards. The *Life events ontology* (see section 3.2) contains application-specific descriptions of life events and goals expressed in WSMML formalisation. These descriptions were not available within the scope of existing resources and were developed for the purposes of the Access-eGov project from scratch. However, about 30% of the building blocks (classes, relations, elements of process model) for the descriptions of life events and goals were reused, mostly from the OntoGov and WSMO ontologies. The most specific parts of the Access-eGov ontology structure are the *Domain ontologies* (section 3.3). The concepts for these ontologies were defined from scratch, according to the needs and requirements of pilot applications. However, attributes of these concepts are highly standardised - approximately 80% of attribute types was taken from standards as Dublin Core, vCard, etc.

A design approach for ontology design where the knowledge engineers start by looking at the domain concepts and begin formalizing them is sometimes referred to as the *knowledge-driven* design approach [Klischewski & Ukena 2007]. The advantage of this approach, especially if there are available ontology resources that can be reused, is that it ensures a standardisation of the resulting ontology model. However, such model can easily be artificial and useless in practice, since it does not reflect the requirements of its usage. A solution can be to combine the existing ontology resources and modelling capabilities of knowledge engineers with a systematic description of requirements given by users of the ontology model.

2.4 Requirement-driven approach towards ontology modelling

Within the Access-eGov a user-driven approach to requirement analysis was chosen. This user-driven approach was also applied to the design of the ontologies. A step-by-step process was suggested (cf. [Klischewski & Ukena 2007]) that took the users' needs as the starting point. Each step of the process is intended to transform the informal description of these user needs towards a formal description in the shape of a conceptual, which is implemented as an ontology in the final step. This is referred to as the *requirement-driven* approach, which is guided by the following basic assumption: "The design of the semantic structures should systematically follow requirements concerning the use of semantic eGovernment services (in addition to a knowledge-driven design approach)" ([Klischewski & Ukena 2007]). This approach provides a generic method how to collect, relate, and formally express the information needs of the service consumers (i.e. citizens and business users) to design the semantic structures and descriptions of the governmental services that are provided by the service providers, i.e. the public administrations.

Information quality and *information architecture* concepts are introduced for the purpose of understanding (web) user requirements and informing the design and evaluation of web applications. The range of principle criteria applicable to measure the information quality do find some agreement among scholars (with differences mainly in emphasis): (1) intrinsic quality (for example, accuracy, objectivity, credibility, and authority), (2) accessibility quality (for example, security and ease of access), (3) contextual quality (for example, relevancy, currency, and completeness), and (4) representational quality (for example, consistency, ease of understanding, etc.) [Wand & Wang 1996], [Lillrank 2003], [Lee et al 2002]. However, those criteria are often difficult to apply, since the information quality is a context-sensitive concept. That is, the information quality can further only be defined relative to its actual use, taking into account the specific informational needs.

The question in focus is how to relate attributes of information quality to the design of specific semantic structures related to the semantic governmental services (SGS) in order to trace this relation from the requirement analysis to the design. For the information quality assessment it is crucial to understand to what extent the actors involved do share or can develop shared meanings. A distinction (introduced in [Lillrank 2003]) between (a) information as an artefact and (b) information as deliverable can be employed. In case of (a), the pre-defined and shared understanding about the domain and the intended use of the information is the basis for defining a common semantic framework incorporating semantic structures as the basis for any application or service design. In case of (b), the rules of negotiation actually frame the common understanding, including the acknowledgement of specific case-based resources, the reasoning and capabilities and the scope of information possibly to be delivered [Klischewski & Ukena 2007]. In both cases the design reflects a commitment of the information provider to a (assumably) common understanding of the task-related domain and/or to the delivery of information "in given situations equipped with certain resources and capabilities", i.e. taking into account the specific situation of the information consumer. The concept most relevant for fixing and maintaining such commitments to information delivery is information architecture.

Information architecture is, in simple words, the structuring of information for a certain purpose; it serves as a link between the technical management of distributed data and strategic business objectives [Klischewski & Ukena 2007]. Thus, it is meant to serve as a bridge between business strategy, end user requirements and technical implementation. Rosenfeld and Morville in [Rosenfeld & Morville 2006] mention categorisation of information, labelling (including choice of terminology), navigation, and search as the main information architecture

components; additionally they mention invisible information architecture components in the background including controlled vocabularies and thesauri, retrieval algorithms, and best bets.

Similar to information quality, developing information architecture presupposes a use situation to which the information architecture is supposed to make an important contribution. Expectations and assumptions of what users want to see, what kind of information they will find, and how users will act in order to find what they need, are guiding the structuring of the information in its static and dynamic aspects. For example, specific information architecture components such as TOC/site map/guide or search system/search wizard are meeting information needs such as retrieval of known items, orientation, and selective or comprehensive research, respectively.

As a result, substantial parts of the information architecture can be considered semantic structures: controlled vocabulary, categories and relations of informational elements are essential ingredients also of ontologies. But often these semantic structures are embedded in the site design; in information architecture they do not necessarily have to be made explicit and isolated in separate formalised components [Klischewski & Ukena 2007].

However, developing information architecture is an approach that, among other aspects, takes in informational needs as user requirements and delivers semantic structures incorporated in the technical implementation. The design decision, which of the user requirements will be served by the resulting information architecture (or not), and thus what information structure will be implemented, depends largely on the (business) objectives of the website operator, i.e. on the commitments towards information delivery.

The next chapter contains a description of applying the above constructs to the ontology modelling process within the Access-eGov project. This approach should demonstrate how the information quality analysis and information architecture development helps creating the semantic structures and how this approach may contribute to a new general method for designing SGS driven by user requirements [Klischewski & Ukena 2007].

3 Ontology modelling within Access-eGov

According to the principles of the requirement-driven approach [Klischewski & Ukena 2007], the ontology modelling has been carried out within the Access-eGov project in the following steps:

No.	Step / Task	Output
1.	Identify informational needs	User scenarios [D2.2]
2.	Identify required information quality	Trial descriptions [D8.2]
3.	Create glossary of topics & terms	Glossary
4.	Create controlled vocabulary	Controlled vocabulary, hierarchy of terms
5.	Group & relate terms	Ontology-like structure with the relations and dependencies specified
6.	Design an ontology	Formally expressed ontology (in Access-eGov, the ontology is represented in the WSML notation)
7.	Implement the semantics	Formal representation of ontology, enhanced by workflow structures.

These seven steps were originally proposed in [Klischewski & Ukena 2007] for the requirement-driven approach towards the ontology modelling. However, the following additional step was added from practical reasons during the process of ontology creation within the Access-eGov project:

8.	Verify the ontology on real data	Ontology with instances (possibly the structure of ontology updated)
----	----------------------------------	--

The following sections contain a detailed description of these particular steps, as well as samples of outputs produced within these steps during the modelling of Access-eGov resource ontologies.

3.1 Identification of information needs

The identification of information needs in the context of eGovernment application can be seen as a specification of life events. This requires an analysis of prior knowledge of users (citizens and business users) and the diversity of informational needs of different user groups. The output was a list of user group's information needs, created and provided by the public administrations in a free-text format. System developers will transform it to the list of life events, formally expressed by the WSML notation, and store it to the *Life events ontology*.

Within the project this took place during requirement analysis (i.e. Task T2.2), which resulted in the outline of three citizen / business user scenarios and an extensive analysis of their requirements in [D2.2]. The requirement analysis ([D2.2], chapter 2) has been elaborated collaboratively by user partners and developers, guided by the user partners' vision of future IT application as well as by the project's discourse seeking to achieve consensus.

The starting point for requirement elicitation has been the activity scenarios provided by the user partners ([D2.2], section 2.1), where three activity scenarios were defined: 1) Building Permission scenario for KSR-SK pilot, 2) Establishing an Enterprise scenario for GLI-PL pilot, and 3) Marriage / Responsibility Finding scenario for SHG-GE pilot application. In the next steps of the analysis ([D2.2], 2.2 - 2.4), regional differences of the pilots were compared to understand similarities and differences, general e-government requirements (based on the state-of-the-art-analysis) were discussed, and finally the results of the requirement analysis as recommendations to the subsequent work packages were produced ([D2.2], section 2.5).

3.2 Identification of required information quality

The information needs from the previous step are analysed with respect to the required properties as scope, relevance, etc. A list of proposed services together with related laws and regulations, documents needed to negotiate between users and PA, and other requirements concerning information provision are provided in a free text format or possibly in the UML notation.

Within the Access-eGov project, the required information quality has been identified within the trial descriptions [D8.2]. The trial portrait was specified for each of pilot applications ([D8.2], sections 2.1, 3.1, and 4.1), containing lists and descriptions of all the governmental services (both traditional and electronic ones) provided within the scope of the pilots. Relevant inputs and outputs, as necessary documents, forms, databases, properties, and other types of artefacts were specified as parameters of the life events and governmental services. For example, the output for KSR-SK pilot application was specified as follows [D8.2], sect. 2.1:

Life event	<ul style="list-style-type: none">- Land-use proceedings (ending with statement on location of a building)- Building proceedings (ending with building permit)- Final approval proceedings (ending with statement on final approval)
Traditional services	Regular mail, phone, personal visit of an office and piece of land.
Electronic services	<ul style="list-style-type: none">- On-line forms and documents – all the relevant forms and documents required for an application submission will be made available on the web site so a user can easily download, fill in or read- On-line database of land-use plans/maps (also information whether existing or not)- On-line guidance of a citizen through the whole process of all permits obtaining- Land registry office is to be accessible online in 2007- Integration of traditional and electronic public services in one place- If feasible, sending copies of required documents via email (once eSignature will be employed by the relevant offices, a citizen will be enabled to send all the documents electronically)- If feasible, administration fees online payment (internet banking)

3.3 Creation of a glossary of topics and terms

The glossary should contain all relevant topics and terms needed for describing the services identified in step one and two. In the Access-eGov project, the three detailed scenarios were

analyzed and an initial glossary was created by the GUC in English, which consisted of three parts, one for each scenario.

The table format with columns such as *term* (in English), *term translations* (to other languages used in pilot applications), *description* and *notes* was used for the glossary. A sample fragment of the glossary is presented here (the entries are for GLI-PL pilot application):

ID	Category	Term (PL/EN)	Description (EN)
2015	Administrative Topic	<i>Spółka cywilna</i> EN: Civil law partnership	Civil law partnership is a partnership of several (minimum two) natural persons who are aiming at common economic goal and the whole civil law partnership is an enterprise which isn't a legal entity (juristic person). Partnership agreement (contract) should be written. One of two [Ways of running business]
2040	Administrative Topic	<i>Przedsiębiorstwo</i> EN: Enterprise	Is established and registered in different authorities during [Establishing a new enterprise] Enterprise is legally, economically and organizationally defined entity running business activity which has no separate existence from its owner. In AeG case both private person and civil law partnership are entrepreneurs running their enterprises.
2024	Administrative Topic	<i>Miejsce zamieszkania</i> EN: Place of residence	Address of place where user permanently lives (place of permanent residence where citizen is registered as legal tenant). In some special cases it can be address of true (real) place of citizen's living where he is not registered. Precondition to identify [Responsible city hall]. One of preconditions to identify [Responsible statistical office]. One of preconditions to identify [Responsible tax office].
2026	Administrative Topic	<i>Adres korespondencyjny</i> EN: Correspondence address	Place where correspondence of citizen should be delivered (which is distinguishable from place of residence, because citizen lives there temporarily and will be able to collect correspondence there)

The main design guideline for the glossary was to include only those terms that appear in practice, i.e. to avoid inventing new terms. The glossary was submitted for review, extension and translation to all partners, especially the user partners. Each user partner refined, extended and translated their respective part of the glossary. The complete glossary is attached to this deliverable, and can also be found at the Access-eGov web site:

http://www.accessegov.org/acegov/uploadedFiles/webfiles/cffile_11_18_07_4_47_51_PM.xls.

Please note that the sheet contains three pages, one for each pilot / field test, which correspond to the three scenarios.

3.4 Creation of controlled vocabulary

The controlled vocabulary contains a hierarchy of categories and subcategories created from the glossary by grouping the terms into hierarchical subgroups. The hierarchy is given by a generalisation of terms, from most general terms (root nodes) to the most specific ones (leaf nodes). The formal notation of a category includes subcategories, attributes, and a free text description.

The following tables contain descriptions of the basic categories, which were used to classify the glossary terms, and which also served as the basis for the design of the Access-eGov classes. The categories were created by grouping the Access-eGov glossary terms and assigning category names. Then the category names were included into the glossary as a new column.

The category descriptions were initially created based on the glossary and later on updated during the creation of the Access-eGov domain model classes (cf. next section).

Administrative Topic

Subcategories:	Administrative Subtopic
Description:	<p>The category A. T. is used for concepts that are specific to the government and administration domain, e.g. marriage, enterprise establishment, place of residence. A. T. are usually defined (or prescribed) in some law and can be used to specify the topic of a life event. It should also have an easy-to-understand description.</p> <p>This is the main source of administrative perspective making. From these topics all boundary objects are derived.</p>

Goal

Subcategories:	Life event, Service goal, User need
Description:	<p>The category G. is used for concepts that constitute some kind of target or objective. It is abstract (i.e. has no direct instances) but has three concrete sub-classes:</p> <p>Service Goal: Every Government Service serves at least one Service Goal. This is especially important because the Personal Assistant will provide access to service initially via the search for Service Goals [D5.1].</p> <p>Life Event⁷: A Life event is solved by a Good Practice Process Pattern. The Good Practice Process Pattern describes how the Life event is solved by combining different Service Goals. The Good Practice Process Pattern may require user input in order to select the appropriate Service Goals.</p> <p>User Need: A User Need can be fulfilled by an appropriate Service Goal.</p>

Client Process

Attributes:	Start, End
Description:	<p><i>Client</i> refers to citizens or businesses. <i>Process</i> refers to the ordered set of activities but does not include process description and process plans (which are of the category <i>Resource</i>). It denotes the process from the point of view of the citizen or business users and how this process relates to the administrative activities.</p> <p><i>This could be seen as a sub-category of the more general category “Process” (which is not yet needed).</i></p>

Government Service

Attributes:	Access Point
Description:	<p>The category G. S. is used for concepts that refer to government services. A G. S. serves a certain goal, requires certain inputs and outputs (often in the form of documents), has a certain point of access (e.g. a particular office or a URL), and consists of a number of activities that are conducted by an administration employee. A G. S. is always provided by a specific administration.</p> <p><i>This could be seen as a sub-category of the more general category “Service” (which is not yet needed).</i></p>

⁷ “Life event” as it is meant in this document should **not** be confused with the planned WSMO-extension “complex goals”.

Document

Subcategories:	Certificate, Form, Notification, Payment Receipt
Description:	<p>The category D. is used for concepts that refer to artefacts which have a specific structure and which play a certain role with respect to government services, e.g. <i>certificates</i>, which are provided by the administration, <i>forms</i>, which must be filled in by citizens in order to use a government service, or <i>notifications</i>, which are issued by an administration in order to inform a service consumer about certain changes in status, <i>payment receipts</i> which a user receives after having paid a fee.</p> <p>This category does not cover artefacts which are of a general and “informational” nature, like a leaflet which gives some general information about types of businesses in Poland. This kind of artefact should be categorized as Resource: Information.</p>

Resource

Subcategories:	Address, Description, Case-based process plan, Case-based list of documents, Good practice process plan, Information
Description:	The category <i>Resource</i> is used for concepts that refer to informational resources but also includes other resources like a case-based process plan or an address.

Activity

Subcategories:	Client activity, Government Activity
Description:	This category is used for concepts that denote a small action, i.e. an activity, but not a service or a process (for example, provision of a document by public administration, provision of data entries by citizens, etc.).

Person

Attributes:	Birth date, Nationality, First name, Last name. All attributes of a person will be represented by using the class User Profile.
Description:	<p>This category is used for human actors. A super-concept (Actor) could later be added to include concepts such as legal actors (like companies) etc.</p> <p>A person will usually assume one or more roles. For Access-eGov we will only focus on the roles and not represent the person.</p>

Role

Subcategories:	Citizen, Administration Officer
Description:	This category is used for all roles a person may assume, e.g. “citizen”, “administration officer”.

Organizational Entity

Subcategories:	Administration
Attributes:	Address
Description:	This category is used for government organisations or parts thereof.

Place

Subcategories:	Country, State, District, Zip-Code, City, Street, House number, Address
Description:	<p>This category is used for any kind of area, i.e. something that can be located by using geo-spatial coordinates.</p> <p>This category does not include small areas, such as a single room or an office. If an office needs to be denoted for service provision, then it should be an attribute of a service access point. However, the names of the concepts in the Place category can be used to create addresses that could also be supplemented with “additional address information” (like “apartment no. 10”).</p>

Relation

Description:	This concept will be used to relate other concepts in order to enable reasoning. Except for the relation “is responsible for” (of an organisational entity / administration) all relations have a definitive domain and range (as depicted in section 4). However, the responsibility relation is central to resolving the responsibility and here the range is not (or not yet) defined. The range will at least include Places and Times.
---------------------	--

Rule

Description:	This is the category for every kind of complex constraint that needs to be explicitly stated as a business rule, e.g. to evaluate the responsible service provider for a certain citizen.
---------------------	---

Time


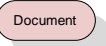

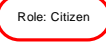
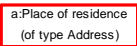
Subcategories:	Date, Time of day, Time interval
Description:	This is the category for different measures of time. Attributes like opening hours etc. will be of this type.

3.5 Grouping and relating the identified items

By creating a controlled vocabulary in step four, the categories are only organised in a hierarchy by *is_a* relation. In step five, a set of other relations and mutual dependencies between the categories will be identified. New categories can also be defined here, if it will be needed for the consistency of the whole structure. An ontology-like structure is provided as the result of this step.

Figure 2 provides an overview of the Access-eGov domain model classes and the relations between the classes. The domain model classes were designed on the basis of the Access-eGov trial glossary structured by the categories provided (cf. previous sections). Initially, each category was represented by one class with relations to other classes. New classes were added in order to account for organizational and/or technical necessities: For example, the class “user profile” was introduced as the system’s repository which accounts for all those attributes of a user that may be stored between sessions (the attribute “place of residence” is given as an example in Figure 2).

Legend for Figure 2:

 Concept or attribute	Represents a concept category. Categories, subcategories and instances are delimited by “:”, i.e. “Role: Administration Officer”. Important attributes are also represented with this symbol, but they are marked using the letter ‘a’, e.g. “a: Access Point” means that “Access Point” is an attribute.
 Boundary Object	Boundary objects are marked in light-red. (Note that there is also a boundary object which is represented as a relation, i.e. the “provides” relation between org. entity and government service.
 Relation	Represents a relation between two concepts.
 PA Concept	A red line around a concept signifies that this concept is relevant to the personal assistant.
 Attribute	Represents an attribute of a class.

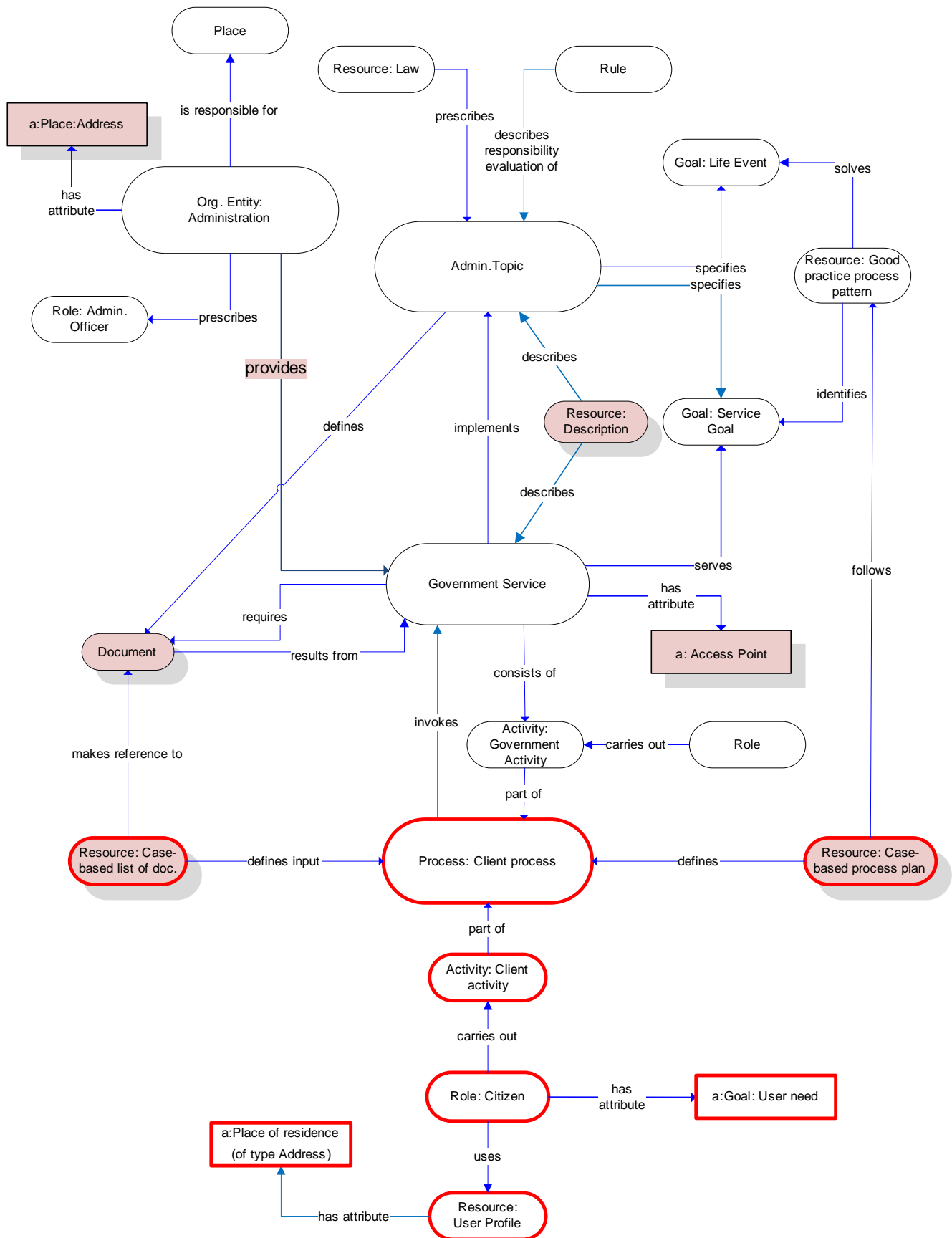


Figure 2: Access-eGov Domain Model classes and their relations

3.6 Design of ontology

The ontology-like structure created in the previous step needs to be formalised and expressed by WSMML statements. It requires 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 are modelled by the following expressions:

```
concept ConceptName
relationName RelaedConcept
```

For example, a hierarchy of certificates needed for the SHG-GE pilot application can be expressed in WSMML notation as follows:

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

In addition, the external ontology resources, identified as relevant for the given domain (see section 2.3), are used in this step to standardise the ontology structure and to achieve a consistency between the semantic descriptions. The attributes of concepts are modelled as non-functional properties. 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. An example of WSMO formalisation of the concept *Person* with its localised attributes is presented here:

```
concept Person
  nonFunctionalProperties
    notVisibleInConceptList hasValue "true"
    dc:title hasValue {"Person", "PL_pl#Osoba", "SK_sk#Osoba",
"DE_de#Person"}
  endNonFunctionalProperties
  v#familyName ofType (1 1) _string
  nonFunctionalProperties
    dc:title hasValue {"Family name", "PL_pl#Nazwisko", "SK_sk#Priezvisko",
"DE_de#Familiennamen"}
  endNonFunctionalProperties
  v#givenName ofType (0 1) _string
  nonFunctionalProperties
    dc:title hasValue {"Given name", "PL_pl#Imie", "SK_sk#Krstné meno",
"DE_de#Vorname"}
  endNonFunctionalProperties
  v#additionalName ofType (0 1) _string
  nonFunctionalProperties
    dc:title hasValue {"Additional name", "PL_pl#Drugie imie", "SK_sk#Ďalšie
mená", "DE_de#Namenszusatz"}
  endNonFunctionalProperties
  v#honorificPrefix ofType (0 1) _string
  nonFunctionalProperties
    dc:title hasValue {"Honorific prefixes", "PL_pl#Tytuł naukowy",
"SK_sk#Tituly pred menom", "DE_de#Ehrentitel"}
  endNonFunctionalProperties
  v#role ofType (0 1) _string
  nonFunctionalProperties
    dc:title hasValue {"Role in organization", "PL_pl#Stanowisko",
"SK_sk#Rola v organizácii", "DE_de#Rolle innerhalb der Organisation"}
  endNonFunctionalProperties
  v#workEmail ofType (0 1) _string
```

```

nonFunctionalProperties
  dc#title hasValue {"Work email", "PL_pl#Adres email", "SK_sk#Pracovný e-
mail", "DE_de#Email-Adresse (Büro)}
endNonFunctionalProperties
v#workTel ofType (0 1) _string
nonFunctionalProperties
  dc#title hasValue {"Work telephone", "PL_pl#Telefon", "SK_sk#Tel. číslo
do práce", "DE_de#Telefon (Büro)}
endNonFunctionalProperties
v#fax ofType (0 1) _string
nonFunctionalProperties
  dc#title hasValue {"Work fax", "PL_pl#Fax", "SK_sk#Fax do práce",
"DE_de#Telefax (Büro)}
endNonFunctionalProperties

```

The whole process of ontology formalisation and implementation within the Access-eGov project is described in chapter 4 below.

3.7 Implementation of semantics

The formal ontology specified in step six is rather static, consisting of declarative statements that express the concepts, their attributes, and mutual relations. In many cases, the conceptualisation needs to be enriched by “business rules” that can be, for example, conditional if-then-else expressions, loops, and workflow sequences. The WSMO framework provides basic choreography and orchestration models for the semantic Web Services that intended to use for semantic descriptions of the life events, generic scenarios, goals, and service profiles in a dynamic manner. However, we have found during the ontology design process that the current proposal of WSMO specification is not completely suitable for our purposes and needs to be extended.

The choreography and orchestration process model within the Access-eGov system will be used to guide citizens to achieve specific goals, and to coordinate activities performed by all actors - citizens, traditional public administration services and web services. A workflow mechanism [Skokan & Bednar, 2007] was designed and implemented as the WSMO extension to provide the required functionality. Our model is based on the workflow CASheW-s model [Norton et al, 2005] originally proposed for the OWL-S specification with the dataflow and WSMO mediation extensions. The general structure of the model is depicted on the diagram in Figure 3.

Access-eGov model reuses the state signature from the WSMO specification and replaces the ASMs transition rules with the workflow constructs. Workflow model consists of activity nodes. Node can be an atomic node (Send, Receive, AchieveGoal and InvokeService), or a control node (Decision, Fork and Join).

Nodes are connected with control links, which represent control flow. Each node can have at least one arriving control link, only the root node which represent starting activity of the workflow doesn't have any arriving links. Number of outgoing links depends on the type of the node. Fork and decision nodes have many outgoing links; other nodes can have only one outgoing link, which determine subsequent activity in the workflow. End nodes do not have any outgoing links (it is possible to have many end nodes, depending on branches in workflow).

Branching is defined with the Decision nodes. Cycles (while or do-while) are created with the decision node and backward control links which points to activity within the cycle (i.e. every activity can have many arriving control links). Decision node represents internal decision,

which is evaluated by the execution environment (service requester cannot directly select branch (i.e. deferred decision pattern), but decision can be depended on the data received from the user). Decision node can contain many outgoing control links, which are guarded by logical conditions. It can contain one un-guarded link, which represent *else* branch. All conditions should be disjoint, i.e. only one condition can be evaluated to true and subsequent node should be selected deterministically.

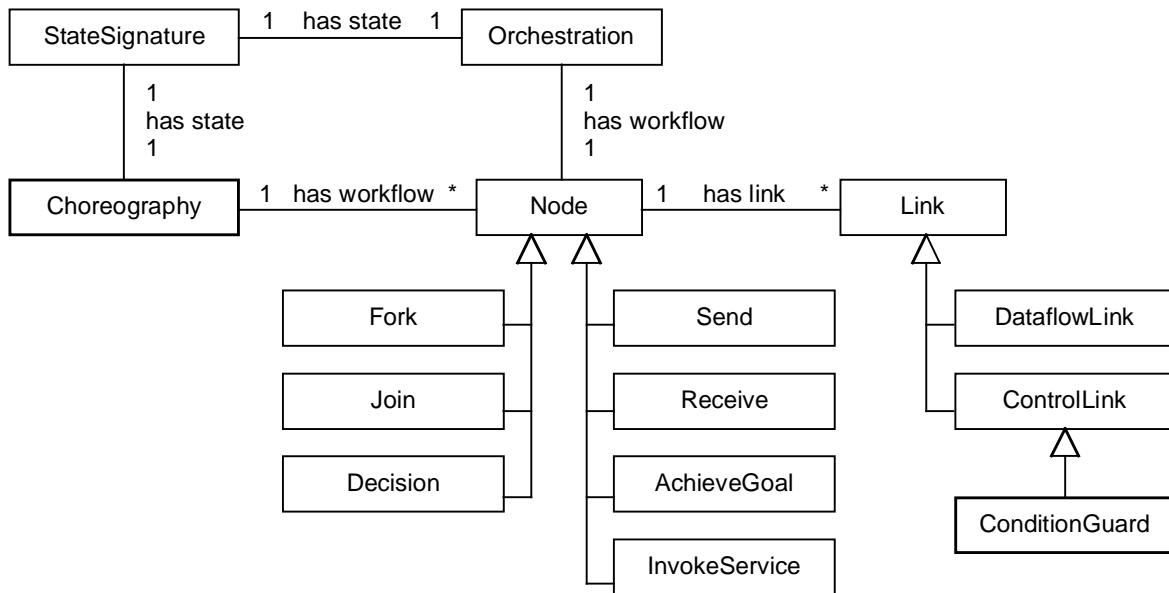


Figure 3. The structure of the Access-eGov orchestration and choreography process model.

Workflow can have parallel threads created with Fork/Join nodes. Fork node has at least two leaving control links and each subsequent activity is executed in the parallel thread. Join synchronizes parallel threads arriving to the node (i.e. it waits until all activities from arriving links will be finished).

The grammar of reference syntax is specified using a dialect of Extended BNF (*Backus - Naur form*, [ISO 14977]) which can be used directly in the SableCC compiler⁸. Terminals are quoted; non-terminals are underlined and refer to tokens and productions. Alternatives are separated using vertical bar '|', and are labelled with labels enclosed in curly braces. Optional elements are appended with a question mark '?'; elements that may occur zero or more times are appended with an asterisk '*'; and elements that may occur one or more times are appended with a plus '+'.
'+'.

```

aeg_workflow = t_workflow aeg_node* aeg_controlflow? aeg_dataflow?;
aeg_node =
  {send} t_perform id? t_send log_expr nfp? |
  {receive} t_perform id? t_receive log_expr nfp? |
  {achievegoal} t_perform id? t_achievegoal [goal]:id nfp? |
  {decision} t_perform id? t_decision nfp? |
  {fork} t_perform id? t_fork nfp? |
  {join} t_perform id? t_join nfp?

aeg_controlflow = t_controlflow aeg_control_links;
aeg_dataflow = t_dataflow aeg_dataflow_links;

aeg_control_links = aeg_control_link aeg_control_links?;
  
```

⁸ <http://sablecc.org/>

```

aeg_control_link = t_source [source]:id t_target [target]:id aeg_guard?;
aeg_guard = t_guard log_expr;

aeg_dataflow_links = aeg_dataflow_link aeg_dataflow_links?;
aeg_dataflow_link =
    t_source [source]:aeg_pin_reference
    t_target [target]:aeg_pin_reference;

aeg_pin_reference = id lbrace variable rbrace;

```

Following example presents the WSMML formalisation of the orchestration interface of the high level process that is connected with the life event “Establish an enterprise” in GLI-PL pilot.

```

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 /**/

```

The batch of answers to the pre-defined questions (Q1) needs to be received from the user by the process. Then other sub-goals need to be achieved in the right order. One of these goals is the *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.

The example bellow describes the choreography interface of *Registration in the Statistical Office* - the second step in the overall process.

```

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}

```

The batch of answers to the pre-defined questions (Q3) needs to be received from the user by this process (number of business activity types). Then the process needs to receive certain form

(variable of a certain type - FormRG_1). The decision node means that some of the following nodes are optional (only node n2_6 in this case - see the *controlFlow* part). Next, the process (might) need(s) to receive another form again. 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. 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).

3.8 Verification of the designed ontology on real data

Since the 7-step procedure is initially based on the user-defined scenarios (which are expressions of the required functionalities and as such does not need necessarily match the reality), it is essential to verify the designed ontology formalism by applying it on the pattern of real data already existing on the user side. This verification was performed in Access-eGov in tight co-operation of developers and user partners. A spreadsheet form was designed by developers according to the designed ontology structure. User partners were then asked to fill in sample real-world data, and possibly modify the form if it does not match the needs. This was especially the case of the SHG-GE pilot application [D8.2], where the sophisticated structure of spatial responsibilities of the public administrations in Schleswig-Holstein required significant changes. Consequently, it resulted in the design of separate *Service Profiles ontology* for SHG-GE pilot application (see section 4.1.4). A fragment⁹ of the spreadsheet form containing the data from SHG-GE pilot application is presented here:

1. Registry office: Standesamt Kiel								
Name:	Standesamt Kiel							
Short name:								
Type:	Registry office							
URL:	http://www.kiel.de/Aemter_01_bis_20/12/Amtsverzeichnis_12/Standesamt.htm							
Postal Address:				Contact:				
Landeshauptstadt Kiel Standesamt Postfach 1152 24099 Kiel				Tel: (0431) 901 - 1112 Fax: (0431) 901-62360 E-Mail: Gisela.Reuter@kiel.de				
Additions:								
Spatial Responsibility:		Kiel + 1002000						
Access Points:								
1. Name:		Kiel Marketing						
Address:		Street	House No	Postal code	Town	Part of Town	Additions	PO BOX
		Andreas-Gayk-Straße	31B	24103	Kiel			
Contact:		fax	phone	mobile	email		further info	
		49 (431) 679 1099	00 49 (431) 679 1015		info@kiel-marketing.de			
URL:		www.kiel-marketing.de						
Link to a map:				url				
Directions to the building:				description				
Access via public transport.:				description				
Num. of car parks:				number/description				
Wheelchair accessible:				yes/no	comments			
Walkway contours for blind people:				yes/no	comments			

⁹ The complete spreadsheet form in the MS Excel format is available on the Access-eGov web site, at: http://www.accessegov.org/acegov/uploadedFiles/webfiles/cffile_11_18_07_5_00_11_PM.xls

Service hours:									
Contact Persons: (for this Access Point)									
1.	role	name+titles		room No.	working hours	additions			
	organization of maritime marriages	Franciska Schmitz							
	contact:	fax	phone	mobile	email	further info			
			(04 31) 679 10 13		f.schmitz@kiel-marketing.de				
2. Name: Standesamt Kiel									
Address:	Street	House No	Postal code	Town	Part of Town	Additions	PO BOX		
	Fleethörn	26	24103	Kiel					
Contact:	fax	phone	mobile	email		further info			
	(0431) 901-62360	(0431) 901 - 1112		Gisela.Reuter@kiel.de		Head of office: Gisela Reuter			
URL:	<i>url</i>								
Link to a map:		<i>url</i>							
Directions to the building:		<i>description</i>							
Access via public transport.:		<i>description</i>							
Num. of car parks:		<i>number/description</i>							
Wheelchair accessible:		yes/no	comments						
Walkway contours for blind people:		yes/no	comments						
Service hours:									
Contact Persons: (for this Access Point)									
1.	role	name+titles		room No.	working hours	additions			
	Standesbeamtin	Gisela Reuter							
	contact:	fax	phone	mobile	email	further info			
			(0431) 901 - 1112		gisela.reuter@kiel.de				
2.	role	name + titles		room No.	working hours	additions			
	Standesbeamtin	Gesine Musch							
	contact:	fax	phone	mobile	email	further info			
			(0431) 901 - 2365		Gesine.Musch@kiel.de				
Services:									
1.	name	<i>Reservation of a the sailing boat Encarnacion</i>							
	type	Reservation of Marriage Location							
	Access Points for the service:								
		Office's Access Point:		Kiel Marketing					
		Phone - hotline Acc.Point:		<i>list of phone numbers</i>					
		Fax - hotline Acc.Point:		<i>list of fax numbers</i>					
		URL (electronic service):		<i>(single?) url address of electronic service</i>					
	spatial responsibility		<i>Kiel + 1002000</i>						
	description	Gleiten Sie auf der 21 m langen Segelyacht Encarnacion ins Glück. 250 Quadratmeter Segelfläche lassen die "Lady" selbst bei 3-4 Windstärken mit lockeren 7 Knoten durch's Wasser gleiten. In dem stilvollen Salon mit einer Fläche von 20 Quadratmetern findet sich für eine Hochzeitsgesellschaft von 15 Personen ausreichend Platz für eine romantische Trauung auf See.							
	responsible persons:		<i>Franziska Schmitz</i>						
	2.	name	<i>Reservation of the lighthouse Kiel</i>						
		type	Reservation of Marriage Location						
		Access Points for the service:							
			Office's Access Point:		Kiel Marketing				
		Phone - hotline Acc.Point:		<i>list of phone numbers</i>					
		Fax - hotline Acc.Point:		<i>list of fax numbers</i>					
		URL (electronic service):		<i>(single?) url address of electronic service</i>					
spatial responsibility		<i>Kiel + 1002000</i>							
description		Das Leuchtturm des Kieler Leuchtturms weist seit 1895 der Schifffahrt den rechten Weg. Der perfekte Ort, um hier den Kurs für ein gemeinsames Leben zu bestimmen							
responsible person:		<i>Franziska Schmitz</i>							

The verification showed that further attributes and concepts had to be added and the relations between the different concepts had to be remodelled in few cases. For example, properties had to be added to access points in order to describe them sufficiently and the structure of addresses of administrations were specified. Furthermore, the concept of *Contact person* was added; the contact persons were assigned with further properties and were related to specific access points. Furthermore the spatial responsibility of an administration was specified for each service of an administration individually and not for the administration as a whole.

As a result of this verification, the Service Profiles ontology was updated, and the separate version of this ontology for the purposes of the SHG-GE pilot application was created (see sections 4.1.3 and 4.1.4 for details). In particular, the classes for expressing spatial responsibilities of the public administrations located in the Schleswig-Holstein were enhanced. New classes for *AccessPoint*, *AccessPointSpatialResponsibilitiesResponsiblePersons*, and *Municipality* were created, and the attributes for the classes *Area*, *Organization*, *Person*, and *ServiceNFP* were modified.

In addition, the domain ontologies for all the pilot applications were updated and enhanced on the enumeration types (*Location* ontologies). The details of resource ontology implementation are described in the following chapter; the resulting ontology in the WSML formalisation is attached to this deliverable.

4 Ontology implementation

This chapter describes the technical implementation of the Access-eGov ontologies. The WSMO Studio¹⁰ was used as a tool for ontology creation (as depicted on Figure 4). The ontologies are formally represented as expressions of the WSMML language.

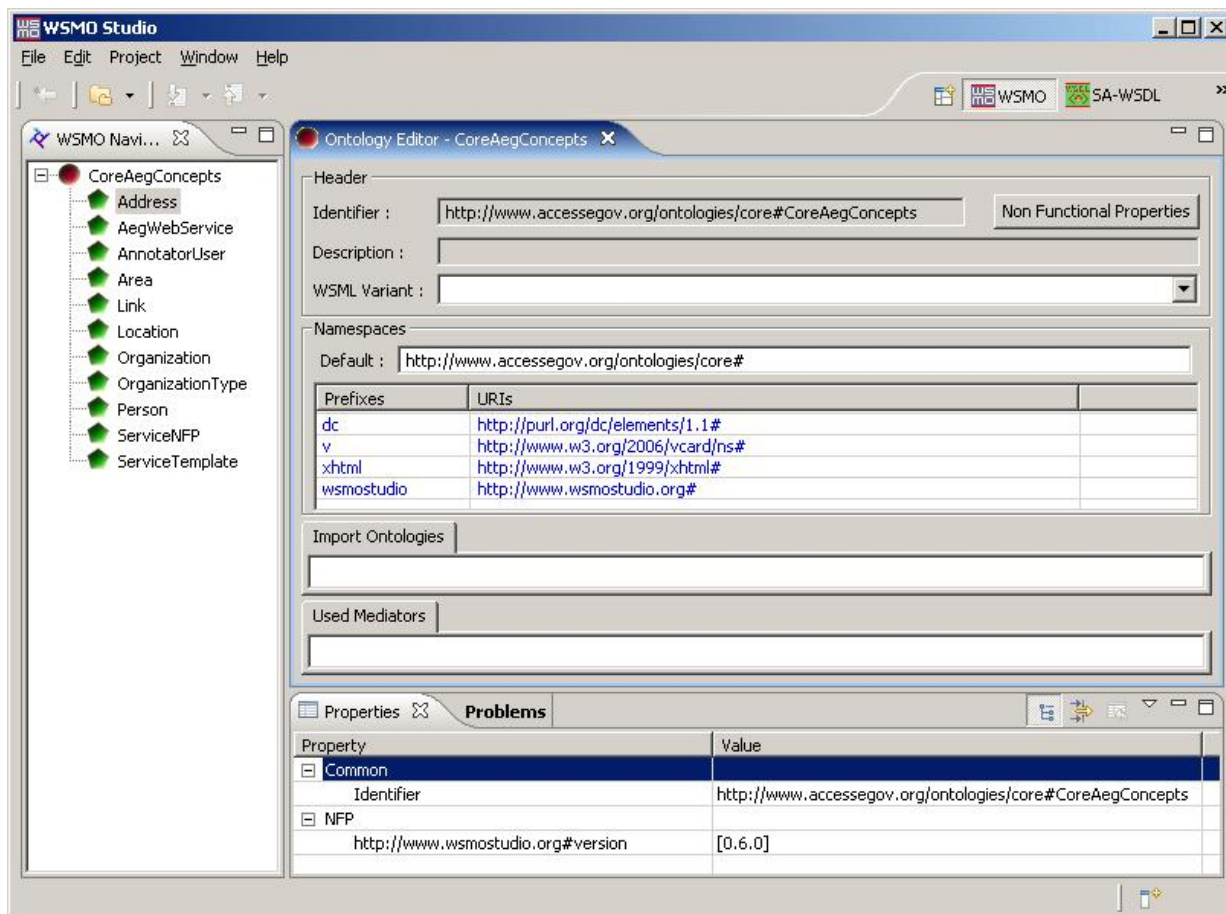


Figure 4. AeG Core - Service Profiles ontology in the WSMO Studio tool

4.1 AeG Core ontology: Service Profiles

The AeG Core ontology contains definitions of basic elements (concepts, attributes, relations) that are shared among the Access-eGov pilot applications and used for modelling and semantic annotation of the atomic governmental services provided by public administrations. In particular, the AeG Core ontology consists of:

- the *SKOS ontology* that defines basic elements for description of concepts and their relations;
- the *non-functional properties* for concepts and attributes, used for visualising forms and lists in the Annotation tool;
- the *Service Profiles ontology*, containing description of non-functional properties of the services identified for particular pilot applications.

¹⁰ <http://www.wsmstudio.org>

The concepts and attributes designed for these parts of the AeG Core ontology will be presented in the following subsections.

4.1.1 SKOS ontology

The *SKOS ontology*, as a part of the AeG Core ontology, defines basic elements for description of structure (hierarchy) of concepts. SKOS Core¹¹ is a model and a vocabulary for expressing the basic structure and content of concept schemes such as controlled vocabularies, taxonomies, ontologies, etc. The two SKOS concepts were reused in the AeG ontology for definitions of concepts, their properties, and relations between the concepts.

Identifier: <http://www.w3.org/2004/02/skos/core#SKOSOntology>

Namespaces / prefixes used: dc _"<http://purl.org/dc/elements/1.1#>"

Concepts	Attributes <attribute_name> <cardinality> <type>	Description
Concept	prefLabel [0..*] _string altLabel [0..*] _string scopeNote [0..*] _string narrower [0..*] Concept broader [0..*] Concept related [0..*] Concept isSubjectOf [0..*] _iri	An abstract idea or notion; a unit of thought.
ConceptScheme	hasTopConcept [0..*] Concept	A set of concepts, optionally including statements about semantic relationships between those concepts.

4.1.2 Non-functional properties for visualisation

The AeG Core ontology also contains the non-functional properties of concepts and attributes which are used for visualising forms and lists in the Annotation tool (see chapter 5). The following non-functional properties for concepts and attributes are defined in the AeG core ontology:

N-F properties for Concepts	Description
<i>notVisibleInConceptList</i>	Switch if a concept should be visible in the list of concepts or not. If a concept is hidden, then its instances can not be directly edited (e.g. the concepts <i>Link</i> or <i>Person</i>).
<i>dc#title</i>	Used for displaying the name of the concept in Annotation tool. Internationalised property.

N-F properties for Attributes	Description
<i>dc#title</i>	Used as a label for the attribute (e.g. in the Annotation tool, in a form for creation of instances of a given concept).

¹¹ <http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20051102/Overview.html>

<i>dc#description</i>	Used as a label for the attribute in a form for creation of instances of a given concept.
<i>isTextArea</i>	If “true”, then the attribute is displayed as a multilined input field in the form.
<i>isAlwaysNew</i>	If “true”, then only a new instance can be created as a value of this attribute (i.e., it is not possible to select an existing instance).
<i>notEditableInForms</i>	If “true”, then the attribute can not be changed directly in the form for editing of an instance.
<i>canOnlySelect</i>	If “true”, then the values of the attribute can be used from a set of predefined values only.
<i>possibleValues</i>	Predefined values for an attribute.

4.1.3 Service Profiles ontology: **CoreAegConcepts**

The *Service Profiles ontology* contains definitions of all the concepts needed for description of non-functional properties of atomic governmental services identified for particular pilot applications within the Access-eGov project. The service itself is represented by the *AegWebService* concept.

Identifier: <http://www.accessegov.org/ontologies/core#CoreAegConcepts>

Namespaces / prefixes used:

_ "<http://www.accessegov.org/ontologies/core#>"

dc _ "<http://purl.org/dc/elements/1.1#>"

← DublinCore ontology reused

v _ "<http://www.w3.org/2006/vcard/ns#>"

← vCard ontology reused

xhtml _ "<http://www.w3.org/1999/xhtml#>"

← XHTML metadata standard reused

wsmostudio _ "<http://www.wsmostudio.org#>"

← WSMO ontology reused

Concepts	Attributes <attribute_name> <cardinality> <type>
<i>Address</i>	postOfficeBox [0..1] _string extendedAddress [0..1] _string streetAddress [0..1] _string locality [0..1] _string region [0..1] _string postalCode [0..1] _string countryName [0..1] _string
<i>AegWebService</i>	serviceNonFP [1..1] ServiceNFP serviceTemplateUsed [1..1] ServiceTemplate
<i>AnnotationUser</i>	title [1..1] _string password [1..1] _string canEdit [0..*] Organization canAdmin [0..*] Organization canPublish [0..*] Organization canView [0..*] Organization isSuperAdmin [1..1] _string
<i>Area</i>	title [1..1] _string belongingLocation [1..*] Location

<i>Link</i>	title [1..1] _string description [0..1] _string href [1..1] _string
<i>Location</i>	title [1..1] _string
<i>Organization</i>	organizationName [1..1] _string organizationUnit [0..1] _string organizationType [1..1] OrganizationType relation [0..*] Link addr [1..1] Address member [0..*] Person
<i>OrganizationType</i>	title [1..1] _string provideService [1..*] ServiceTemplate
<i>Person</i>	familyName [0..1] _string givenName [1..1] _string additionalName [0..1] _string honorificPrefix [0..1] _string honorificSuffix [0..1] _string role [0..1] _string workEmail [0..1] _string workTel [0..1] _string fax [0..1] _string
<i>ServiceNFP</i>	title [1..1] _string description [0..1] _string relation [0..*] Link responsibleOrganization [1..1] Organization contactPerson [0..*] Person serviceHours [0..1] _string spatialResponsibility [1..1] Area
<i>ServiceTemplate</i>	title [1..1] _string description [0..1] _string templateNFP [1..1] ServiceNFP templateCapability [0..1] _iri templateInterface [0..*] _iri

This Service Profiles ontology is used for the GLI-PL and KSR-SK pilot applications.

However, for the SHG-GE pilot application the Service Profiles ontology was modified, since this pilot requires a more sophisticated structure of locations and spatial responsibilities of the public administrations located in the Schleswig-Holstein.

4.1.4 Modified Service Profiles ontology for SHG-GE pilot application

Identifier: <http://www.accessegov.org/ontologies/shg#CoreAegConcepts>

Concepts	Attributes <attribute_name> <cardinality> <type>
<i>AccessPoint</i>	title [1..1] _string streetAddress [0..1] _string streetNumber [0..1] _string

	postalCode [0..1] _string postOfficeBox [0..1] _string poBoxPostalCode [0..1] _string locality [0..1] _string partOfTown [0..1] _string addressAdditions [0..1] _string region [0..1] _string countryName [0..1] _string fax [0..*] _string workTel [0..*] _string mobileTel [0..*] _string workEmail [0..*] _string furtherInfo [0..1] _string url [0..*] Link linkToMap [0..1] _string directionsToBuilding [0..1] _string accessViaPublicTransport [0..1] _string numOfCarParks [0..1] _string wheelchairAccessible [0..1] _string wheelchairAccessibleComments [0..1] _string walkwayContoursForBlindPeople [0..1] _string walkwayContoursForBlindPeopleComments [0..1] _string serviceHours [0..1] _string contactPerson [0..*] Person
<i>AccessPointSpatialResponsibilitiesResponsiblePersons</i>	accessPoint [1..1] AccessPoint spatialResponsibility [0..*] Area responsiblePersonServiceAccessPoint [0..*] Person
<i>AegWebService</i>	<i>no changes in the attributes</i>
<i>AnnotationUser</i>	<i>no changes in the attributes</i>
<i>Area</i> Attributes were modified.	hasState [1..1] State hasRegion [1..1] Region hasMunicipality [1..1] Municipality
<i>Link</i>	<i>no changes in the attributes</i>
<i>Municipality</i> The concept is used instead of the <i>Location</i> concept.	title [1..1] _string
<i>Organization</i> Attributes were modified.	organizationName [1..1] _string organizationShortName [0..1] _string organizationType [1..1] OrganizationType url [0..*] Link streetAddress [0..1] _string streetNumber [0..1] _string postalCode [0..1] _string postOfficeBox [0..1] _string poBoxPostalCode [0..1] _string locality [1..1] _string partOfTown [0..1] _string

	addressAdditions [0..1] _string region [0..1] _string countryName [1..1] _string additions [0..1] _string accessPoint [0..*] AccessPoint organizationServices [0..*] AegWebService
<i>OrganizationType</i>	<i>no changes in the attributes</i>
<i>Person</i> Attributes were modified.	role [0..*] _string formOfAddress [0..1] _string familyName [1..1] _string givenName [0..1] _string additionalName [0..1] _string honorificPrefix [0..1] _string honorificSuffix [0..1] _string roomNumber [0..1] _string workingHours [0..1] _string personAdditions [0..1] _string fax [0..*] _string workTel [0..*] _string mobileTel [0..*] _string workEmail [0..*] _string url [0..*] Link furtherInfo [0..1] _string
<i>Region</i>	title [1..1] _string
<i>ServiceNFP</i> Attributes were modified.	title [1..1] _string accessPointWithResponsibilities [1..*] AccessPointSpatialResponsibilitiesResponsiblePersons description [0..1] _string relation [0..*] Link responsibleOrganization [1..1] Organization
<i>ServiceTemplate</i>	<i>no changes in the attributes</i>
<i>State</i>	title [1..1] _string

4.2 Life events ontology

The *Life events ontology* contains 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 [D5.1].

4.2.1 SHG-GE pilot: *Marriage life event*

The *Marriage life event* is implemented as a complex goal, consisting from the following subgoals:

- *ApplyForMarriageGoal* (applying for marriage)
- *WeddingPlaceReservationGoal* (reserving a wedding location and date)
- *WeddingCeremonyGoal* (getting married in a marriage ceremony)

The WSML implementation of the *Marriage life event* is presented in section 3.7 above.

The *WeddingPlaceReservationGoal* and *WeddingCeremonyGoal* are elementary goals. The *ApplyForMarriageGoal* is a complex goal, consisting of one or more of the following subgoals:

- *get Proof of the acquisition of the German citizenship*
- *get Certified copy of the family register of a previous marriage*
- *get Certified copy of the family register of the parents*
- *get Birth certificate indicating natural parents*
- *get a Birth certificate*
- *get a Birth certificate of the children*
- *get a Marriage certificate*
- *get a Marriage certificate of a previous marriage*
- *get a Certificate of death of a previous spouse*
- *get Decree Absolute of a divorce*
- *get Certificate No. 1347/2000*
- *get Slovak birth certificate with an Apostille*

As an example, the *get a Marriage certificate* goal was implemented in WSMML as follows:

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

goal MarriageCertificateGoal
  nfp
    dc#title hasValue "get a Marriage certificate"
    dc#description hasValue "The marriage certificate is issued by the register
    office at the place of marriage. The marriage certificate can be obtained by the
    spouses recorded on it or by their ancestors or descendants."
  endnfp
  interface MarriageCertificateInterface
    choreography
      workflow
        perform n10_lo send ?a1 memberOf MarriageCertificate.
```

4.2.2 GLI-PL pilot: *Establish Enterprise life event*

The *Establish Enterprise life event* is implemented as a complex goal, consisting of the following subgoals:

- *RegisterInLocalGovernmentGoal*
- *RegisterInStatisticalOfficeGoal*
- *RegisterInTaxOfficeGoal*
- *RegisterInSocialInsuranceAgencyGoal*

The WSMML implementation of the *Establish Enterprise life event* is presented here:

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

goal EstablishEnterpriseLifeEvent
  nfp
    dc#title hasValue "Establish Enterprise Life Event"
  endnfp
  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

```

The *RegisterInTaxOfficeGoal* is the complex goal, which contains the *CreateBankAccountGoal* as its subgoal. The *RegisterInLocalGovernmentGoal*, *RegisterInStatisticalOfficeGoal*, and *RegisterInSocialInsuranceAgencyGoal* are elementary goals.

As an example, the *RegisterInLocalGovernmentGoal* was implemented in WSMML as follows:

```

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

goal RegisterInLocalGovernmentGoal
  nfp
    dc#title hasValue "Register in city hall."
    dc#description hasValue "All issues related to registration an enterprise are
  completed in the city hall (local government) appropriate for user's place of
  residence."
  endnfp
  capability RegisterInLocalGovernmentCapability
  interface RegisterInLocalGovernmentInterface
  choreography
    workflow
      perform n0_1 decision
      perform n0_2 send ?x memberOf Licence.
      perform n0_3 receive ?x memberOf Fee.
      perform n0_4 send ?x memberOf CityHallRegistrationCertificate.
    controlFlow
      source n0_1 target n0_2 guard exists ?x (?y[q2 hasValue ?x] memberOf Q1 and
      ?x[licence hasValue required] and ?x memberOf TypeOfBusiness).
      source n0_2 target n0_3
      source n0_3 target n0_4
    dataFlow
      source n1_1{?x} target n0_1{?y}

```

4.2.3 KSR-SK pilot: *Build a House live event*

The *Build a House live event* is implemented as a complex goal, consisting from the following subgoals:

- *RegisterInLocalGovernmentGoal*
- *RegisterInStatisticalOfficeGoal*
- *RegisterInTaxOfficeGoal*

The WSMML implementation of the *Build a House live event* is presented here:

```

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

```

```
goal BuildAHouseLiveEvent
  nfp
  dc#title hasValue "Obtaining A Building Permit"
  endnfp
  interface BuildAHouseLiveEventInterface
    orchestration
    workflow
      perform nl_q1 receive ?q1 memberOf Q1.
      nfp aeg#configuration hasValue _boolean("true") endnfp
      perform nl_ld decision
      perform nl_lg achieveGoal SelectSiteGoal
      nfp dc#title hasValue "Select a site for new building." endnfp
      perform nl_q3 receive ?q3 memberOf Q3. //owner of the site
      nfp aeg#configuration hasValue _boolean("true") endnfp
      perform nl_20g achieveGoal
      ObtainStatementFromSlovakianGasIndustryCompanyGoal
      nfp dc#title hasValue "Obtain statement from Slovakian gas industry
company - distribution dept. to your project documentation" endnfp
      perform nl_21g achieveGoal ObtainStatementFromEast_SlovakianWaterCompGoal
      nfp dc#title hasValue "Obtain statement from East - Slovakian Water comp.
to your project documentation" endnfp
      perform nl_22g achieveGoal
      ObtainStatementFromSlovakianWater_ManagementCompanyGoal
      nfp dc#title hasValue "Obtain statement from Slovakian water-management
company to your project documentation" endnfp
      perform nl_23g achieveGoal ObtainStatementFromEast_SlovakianEnergyCompGoal
      nfp dc#title hasValue "Obtain statement from East - Slovakian Energy
comp. to your project documentation" endnfp
      perform nl_24g achieveGoal ObtainStatementFromOrangeSlovenskoASGoal
      nfp dc#title hasValue "Obtain statement from Orange Slovensko a.s. to
your project documentation" endnfp
      perform nl_25g achieveGoal ObtainStatementFromT_MobileSlovenskoASGoal
      nfp dc#title hasValue "Obtain statement from T-Mobile Slovensko, a.s. to
your project documentation" endnfp
      perform nl_30 achieveGoal GetListOfDocumentsFromTechInfrastructureGoal
      nfp dc#title hasValue "Get list of the other relevant Technical
Infrastructure bodies for your case." endnfp
      perform nl_31 achieveGoal ObtainDocumentsFromTechInfrastructureGoal
      nfp dc#title hasValue "Obtain relevant statemets needed from Technical
Infrastructure bodies to your project documentation." endnfp
      perform nl_4q2 receive ?q2 memberOf Q2.
      nfp aeg#configuration hasValue _boolean("true")
      dc#description hasValue "The type of your building might be as
expected from the zone plan or not."
      endnfp
      perform nl_40d decision
      perform nl_41g achieveGoal LandUseProceedingsGoal
      nfp dc#title hasValue "Land-Use Proceedings for your building" endnfp

      perform nl_42g achieveGoal BuildingProceedingsGoal
      nfp dc#title hasValue "Building Proceedings for your building" endnfp
      perform nl_50d decision
      perform nl_51g achieveGoal MergedProceedingsGoal
      nfp dc#title hasValue "Merged Proceedings for your building" endnfp
      perform nl_6g achieveGoal ApprovalProceedingsGoal
      nfp dc#title hasValue "Approval Proceedings for your building" endnfp

  controlFlow
    source nl_q1 target nl_ld
    source nl_ld target nl_lg guard ?x[q1 hasValue false] and ?x memberOf Q1.
    source nl_lg target nl_20g
    source nl_ld target nl_20g

    source nl_20g target nl_21g
    source nl_21g target nl_22g
    source nl_22g target nl_23g
    source nl_23g target nl_24g
    source nl_24g target nl_25g
```

```
source n1_25g target n1_30g
source n1_30g target n1_31g
source n1_31g target n1_4q2
```

```
source n1_40d target n1_41g guard ?q1[q1 hasValue false] and ?q1 memberOf Q2.
source n1_41g target n1_42g
source n1_42g target n1_50d
source n1_40d target n1_50d
```

```
source n1_50d target n1_51 guard ?q2[q1 hasValue true] and ?q2 memberOf Q2.
source n1_51g target n1_6g
source n1_50d target n1_6g
```

```
dataFlow
source n1_q1{?q1} target n1_1d{?q1}
source n1_q1{?q1} target n1_41g{?q1}
source n1_q1{?q1} target n1_42g{?q1}
source n1_q1{?q1} target n1_51g{?q1}
source n1_4q2{?q2} target n1_40d{?q2}
```

Complex goals:

- *ApprovalProceedingsGoal*; contains sub-goals:
 - *ObtainStatementFromSlovakianGasIndustryCompanyGoal*,
 - *ObtainStatementFromEast_SlovakianWaterCompGoal*,
 - *ObtainStatementFromSlovakianWater_ManagementCompanyGoal*.
- *BuildingProceedingsGoal*; contains sub-goals:
 - *ObtainProofOfLandOwnershipGoal*,
 - *VerbalTreatGoal*.
- *LandUseProceedingsGoal*; contains sub-goals:
 - *ObtainProofOfLandOwnershipGoal*,
 - *ObtainProjectDocumentationGoal*.
- *MergedProceedingsGoal*; contains sub-goals:
 - *ObtainProofOfLandOwnershipGoal*,
 - *ObtainProjectDocumentationGoal*,
 - *VerbalTreatGoal*.

Elementary goals:

- *GetListOfDocumentsFromTechInfrastructureGoal*
- *ObtainDocumentsFromTechInfrastructureGoal*
- *ObtainProjectDocumentationGoal*
- *ObtainProofOfLandOwnershipGoal*
- *ObtainStatementFromCountyFireProtectAndRescueHeadGoal*
- *ObtainStatementFromDistrictEnvironmentOfficeGoal*
- *ObtainStatementFromDistrictEnvironmentOffice_CivilEmergencyPlanningGoal*
- *ObtainStatementFromDistrictLandOfficeGoal*
- *ObtainStatementFromDistrictMiningOfficeGoal*
- *ObtainStatementFromDistrictOfficeOfTransportationAndSurfaceRoadsGoal*
- *ObtainStatementFromDistrictWoodlandOfficeGoal*
- *ObtainStatementFromEast_SlovakianEnergyCompGoal*
- *ObtainStatementFromEast_SlovakianWaterCompGoal*
- *ObtainStatementFromLandRegisterOfficeGoal*
- *ObtainStatementFromMechanicalAndGardenServicesOfCityGoal*

- *ObtainStatementFromMinistryOfDefence_AdministrationOfReal_EstateAndConstructionGoal*
- *ObtainStatementFromMinistryOfEnvironmentOfSlovakRepublicGoal*
- *ObtainStatementFromMunicipalityGoal*
- *ObtainStatementFromNaftaASGoal*
- *ObtainStatementFromOrangeSlovenskoASGoal*
- *ObtainStatementFromPoliceCountyHeadOfficeGoal*
- *ObtainStatementFromRegionalMonumentOfficeGoal*
- *ObtainStatementFromRegionalOfficeForTransportationAndSurfaceRoadsGoal*
- *ObtainStatementFromRegionalOfficeOfPublicHealthcareGoal*
- *ObtainStatementFromRegionalVeterinaryAndFoodAdministrationGoal*
- *ObtainStatementFromRoadAdministrationOfficeOfSelf_GoverningRegionGoal*
- *ObtainStatementFromSlovakianGasIndustryCompanyGoal*
- *ObtainStatementFromSlovakianRoadAdministrationOfficeGoal*
- *ObtainStatementFromSlovakianWater_ManagementCompanyGoal*
- *ObtainStatementFromSlovakTelecomASGoal*
- *ObtainStatementFromT_MobileSlovenskoASGoal*
- *ObtainStatementFromTechnicalInspectionOfficeGoal*
- *ObtainStatementFromTranspetrolASGoal*
- *ObtainStatementFromUPCSlovenskoSpolSROGoal*
- *SelectSiteGoal*
- *VerbalTreatGoal*

As an example, the *RegisterInLocalGovernmentGoal* was implemented in WSMML as follows:

```

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

goal ApprovalProceedingsGoal
  nfp
    dc#title hasValue "Approval Proceedings."
    dc#description hasValue "."
  endnfp

  capability ApprovalProceedingsCapability

  interface ApprovalProceedingsInterface
    choreography
      workflow
        perform n5_10 achieveGoal ObtainStatementFromSlovakianGasIndustryCompanyGoal
        perform n5_11 achieveGoal ObtainStatementFromEast_SlovakianWaterCompGoal
        perform n5_12 achieveGoal ObtainStatementFromSlovakianWater_ManagementCompanyGoal

  controlFlow
    source n5_10 target n5_11
    source n5_11 target n5_12

```

The life events and goals described in the *Life events ontology* are used in the Personal Assistant client tool (namely, in its GSE and GS modules, see [D5.1] for details). 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 - citizens or businesses.

4.3 Domain ontologies

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 following sections contain listings of the concepts and attributes identified as elements of the domain ontologies for all the three pilot applications.

4.3.1 SHG-GE pilot: *SHG*Ontology

The domain ontology for SHG-GE pilot application contains definitions of application forms, documents, notification messages, questions, locations, etc., needed to apply for marriage in Germany. The following table contains all the domain-specific concepts that were identified for the SHG-GE pilot. The concepts are related to each other by their attributes; type of the attribute denotes the related concepts. These domain-specific concepts are used for annotating the services of the SHG-GE pilot – both for specification of non-functional properties (as instances of these concepts) and functional properties (these concepts are referenced and used in the workflow structures, process models).

Identifier: <http://www.accessegov.org/ontologies/shg#SHGOntology>

Namespaces / prefixes used:

_ "<http://www.accessegov.org/ontologies/shg/>"

dc _ "<http://purl.org/dc/elements/1.1/>"

aeg _ "<http://www.accessegov.org/ontologies/core/>"

Concepts	Attributes name [cardinality] type
<i>ApplyForMarriage</i>	title [1..1] _string description [0..1] _string
<i>BirthCertificate</i>	title [1..1] _string description [0..1] _string
<i>BirthCertificateIndicatingNaturalParents</i>	title [1..1] _string description [0..1] _string
<i>Certificate1347</i>	title [1..1] _string description [0..1] _string
<i>CertificateOfMarriageApplication</i>	title [1..1] _string description [0..1] _string
<i>CertificateOfRegistration</i>	title [1..1] _string description [0..1] _string
<i>DeadCertificate</i>	title [1..1] _string description [0..1] _string
<i>DecreeAbsolute</i>	title [1..1] _string description [0..1] _string
<i>Location</i>	hasState [1..1] State hasRegion [0..1] Region hasMunicipality [0..1] Municipality
<i>MarriageDissolution</i>	title [1..1] _string Instances: - widow

	<ul style="list-style-type: none"> - divorcedInGermany - divorcedInEU - divorcedAbroad
<i>Nationality</i>	title [1..1] _string Instances: <ul style="list-style-type: none"> - iso_3166_deu - iso_3166_svk
<i>ProofOfCitizenShipAcquisition</i>	title [1..1] _string description [0..1] _string
<i>Q1</i> Description: The concept for predefined queries. Questions: “Are you 18 years or older?”, “What is your nationality?”, “Where is your place of residence in Germany?”, “Where do you like to have your wedding ceremony?”.	title [1..1] _string description [0..1] _string q1 [0..*] _boolean q2 [0..*] Nationality q3 [0..*] Municipality q4 [0..*] Region
<i>Q2</i> Description: The concept for predefined queries. Questions: “Have you been married or lived in a life partnership before?”, “Have you got child custody for any children?”.	title [1..1] _string q1 [0..*] _boolean q2 [0..*] _boolean
<i>Q3</i> Description: The concept for predefined queries. Question: “Are you a German citizen since birth?”	title [1..1] _string q1 [0..*] _boolean
<i>Q4</i> Description: The concept for predefined queries. Question: “Did your parents get married in Germany?”	title [1..1] _string q1 [0..*] _boolean
<i>Q5</i> Description: The concept for predefined queries. Question: “Have you been adopted as a child?”	title [1..1] _string q1 [0..*] _boolean
<i>Q6</i> Description: The concept for predefined queries. Question: “Did your last marriage ceremony take place in Germany?”	title [1..1] _string q1 [0..*] _boolean
<i>Q7</i> Description: The concept for predefined queries. Question: “Is the dissolution of your previous marriage indicated in the family register or marriage certificate?”	title [1..1] _string q1 [0..*] _boolean
<i>Q8</i> Description: The concept for predefined queries. Question: “What is the reason of the dissolution of your previous marriage?”	title [1..1] _string q1 [0..*] MarriageDissolution
<i>Q9</i> Description: The concept for predefined queries. Question: “Have you got any academic degree and do you want this degree to be recorded in the marriage documents?”	title [1..1] _string q1 [0..*] _boolean
<i>SlovakBirthCertificateWithApostille</i>	title [1..1] _string description [0..1] _string

In addition, the SHG-GE pilot has the locations defined in the *SHGLocations ontology*. This ontology contains the instances of particular locations that are relevant for the pilot application. The locations are instances of the *State*, *Region*, and *Municipality* concepts which are in a hierarchical relation to each other. The following example presents a fragment of the SHGLocations ontology:

```
namespace {_"http://www.accessegov.org/ontologies/shg/locations#",
  dc _"http://purl.org/dc/elements/1.1#",
  aeg _"http://www.accessegov.org/ontologies/core#"
}
ontology SHGLocations
importsOntology
  _"http://www.accessegov.org/ontologies/shg#CoreAegConcepts"
instance s010 memberOf aeg#State
  dc#title hasValue "Schleswig Holstein"
instance r01002 memberOf aeg#Region
  dc#title hasValue "Kiel"
instance r01003 memberOf aeg#Region
  dc#title hasValue "Lübeck"
instance r01004 memberOf aeg#Region
  dc#title hasValue "Neumünster"
instance r01051 memberOf aeg#Region
  dc#title hasValue "Dithmarschen"
instance r01053 memberOf aeg#Region
  dc#title hasValue "Herzogtum Lauenburg"
instance r01001 memberOf aeg#Region
  dc#title hasValue "Flensburg"
instance m01001000 memberOf aeg#Municipality
  dc#title hasValue "Flensburg, Stadt"
instance l01001000 memberOf aeg#Area
  aeg#hasState hasValue s010
  aeg#hasRegion hasValue r01001
  aeg#hasMunicipality hasValue m01001000
... etc.
```

4.3.2 GLI-PL pilot: *GLIOntology*

The domain ontology for GLI-PL pilot application contains definitions of application forms, documents, notification messages, questions, locations, etc., needed in the process of establishing an enterprise.

Identifier: <http://www.accessegov.org/ontologies/gli#GLIOntology>

Namespaces / prefixes used:

```
_ "http://www.accessegov.org/ontologies/gli/"
dc _"http://purl.org/dc/elements/1.1#"
aeg _"http://www.accessegov.org/ontologies/core/"
```

Concepts	Attributes
<i>EnumNumber</i> Description: Interval of numbers.	title [1..1] _string description [0..1] _string Instances: - one - twoOrThree - moreThanThree
<i>Fee</i>	title [1..1] _string

	amount [0..*] _decimal currency [0..*] _string
<i>FormNIP_1</i>	title [1..1] _string description [0..1] _string
<i>FormNIP_2</i>	title [1..1] _string description [0..1] _string
<i>FormNIP_B</i>	title [1..1] _string description [0..1] _string
<i>FormNIP_C</i>	title [1..1] _string description [0..1] _string
<i>FormNIP_D</i>	title [1..1] _string description [0..1] _string
<i>FormRG_1</i>	title [1..1] _string description [0..1] _string
<i>FormRG_RD</i>	title [1..1] _string description [0..1] _string
<i>FormVAT_R</i>	title [1..1] _string description [0..1] _string
<i>FormVAT_R_UE</i>	title [1..1] _string description [0..1] _string
<i>FormZUS_ZAA</i>	title [1..1] _string description [0..1] _string
<i>FormZUS_ZBA</i>	title [1..1] _string description [0..1] _string
<i>FormZUS_ZCNA</i>	title [1..1] _string description [0..1] _string
<i>FormZUS_ZCZA</i>	title [1..1] _string description [0..1] _string
<i>FormZUS_ZFA</i>	title [1..1] _string description [0..1] _string
<i>FormZUS_ZPA</i>	title [1..1] _string description [0..1] _string
<i>FormZUS_ZUA</i>	title [1..1] _string description [0..1] _string
<i>FormZUS_ZZA</i>	title [1..1] _string description [0..1] _string
<i>InsurancePayer</i> Description: Types of the subjects paying insurance premium.	title [1..1] _string description [0..1] _string Instances: - civilLawPartnership - partners - both
<i>InsuranceType</i>	title [1..1] _string description [0..1] _string Instances: - socialORHealth - health
<i>Location</i>	description [0..1] _string
<i>NIPnumber</i>	title [1..1] _string

<i>NumberOfBankAccount</i>	title [1..1] _string
<i>NumberOfPremises</i>	title [1..1] _string
<i>PayerVAT</i>	title [1..1] _string description [0..1] _string Instances: - yes - no - subjectivelyReleased
<i>Q1</i> Description: The concept for predefined queries. Questions: “Which way of running business do you want to establish?”, “Which type of business activity are you going to establish?”, “What is your place of residence?”, “What is the address of main premises of your enterprise?”.	title [1..1] _string q1 [0..*] WayOfRunningBusiness q2 [0..*] TypeOfBusiness q3 [0..*] Location q4 [0..*] Address
<i>Q2</i> Description: The concept for predefined queries. Question: “Do you want to apply for this licence/permission?”	title [1..1] _string q1 [0..*] _boolean
<i>Q3</i> Description: The concept for predefined queries. Question: “How many types of business activities did you register?”	title [1..1] _string q1 [0..*] EnumNumber
<i>Q4</i> Description: The concept for predefined queries. Question: “What is the way of taxation for your enterprise?”	title [1..1] _string q1 [0..*] TaxationWay
<i>Q5</i> Description: The concept for predefined queries. Questions: “How many different premises are for your enterprise?”, etc.	title [1..1] _string q1 [0..*] EnumNumber q2 [0..*] EnumNumber
<i>Q6</i> Description: The concept for predefined queries. Question: “Are you going to register as VAT payer?”	title [1..1] _string q1 [0..*] PayerVAT
<i>Q7</i> Description: The concept for predefined queries. Questions: “Who is going to pay insurance premiums for partners and employees?”, “Which type of insurance do you need?”, “Do you have any family members of your family who need health insurance?”, “Do your family members have the same place of residence as you?”.	title [1..1] _string q1 [0..*] InsurancePayer q2 [0..*] InsuranceType q3 [0..*] _boolean q4 [0..*] _boolean
<i>Q8</i> Description: The concept for predefined queries. Questions: “Are you going to employ people or have collaborators?”, “Does your employee have an employment contract?”, “Does your employee have an order or any other contract about providing services?”, “Is your employee pensioner, retiree or student?”.	title [1..1] _string q1 [0..*] _boolean q2 [0..*] _boolean q3 [0..*] _boolean q4 [0..*] _boolean
<i>RegistrationFormInCityHall</i>	title [1..1] _string

	description [0..1] _string
<i>RegistrationInSocialInsuranceAgency</i>	title [1..1] _string description [0..1] _string
<i>RegistrationInTaxOffice</i>	title [1..1] _string description [0..1] _string
<i>REGON</i> Description: A number that is issued by the statistical office.	title [1..1] _string description [0..1] _string
<i>TaxationWay</i> Description: The way of taxation for the enterprise.	title [1..1] _string Instances: - byTaxCard - byLumpSum - byBookOfIncAndExpenditures - byFullBookkeeping
<i>TypeOfBusiness</i>	title [1..1] _string description [0..1] _string abbreviation [0..*] _string
<i>VATcertificate</i>	title [1..1] _string description [0..1] _string
<i>WayOfRunningBusiness</i>	title [1..1] _string description [0..1] _string
<i>PrivatePerson</i>	title [1..1] _string description [0..1] _string

In addition, the GLI-PL pilot has the locations defined in the *GLILocations ontology*. This ontology contains the instances of particular locations that are relevant for the pilot application. The locations are instances of the *City* and *CityPartition* concepts, which are also defined in this ontology. Following example presents a fragment of the GLILocations ontology:

```
namespace {_ "http://www.accessegov.org/ontologies/gli/locations#",
  dc _ "http://purl.org/dc/elements/1.1#",
  aeg _ "http://www.accessegov.org/ontologies/core#"
}
ontology GLILocations
concept City //city
concept CityPartition //this reffers to city wards, streets, etc.
relation location(ofType CityPartition, ofType City)

instance pAll memberOf CityPartition
  nfp dc#title hasValue "All cities belonging to county/Whole city/all streets"
endnfp

instance c0940000 memberOf City
  nfp dc#title hasValue "Gliwice" endnfp
instance c0940849 memberOf City
  nfp dc#title hasValue "Knurów" endnfp
instance c0942417 memberOf City
  nfp dc#title hasValue "Pyskowice" endnfp
instance p0942423 memberOf CityPartition
  nfp dc#title hasValue "Czerwionka" endnfp
instance p0942446 memberOf CityPartition
  nfp dc#title hasValue "Mikuszowina" endnfp
instance p0942452 memberOf CityPartition
  nfp dc#title hasValue "Zaolszany" endnfp
```

4.3.3 KSR-SK pilot: *KSROntology*

The domain ontology for KSR-SK pilot application contains definitions of application forms, documents, notification messages, questions, fees, locations, etc., needed in the process of obtaining a building permit.

Identifier: <http://www.accessegov.org/ontologies/ksr#KSROntology>

Namespaces / prefixes used:

_ "<http://www.accessegov.org/ontologies/ksr/>"

dc _ "<http://purl.org/dc/elements/1.1/>"

aeg _ "<http://www.accessegov.org/ontologies/core/>"

Concepts	Attributes
<i>ApplicationFormBuildingPermit</i>	title [1..1] _string description [0..1] _string
<i>ApplicationFormBuildingProceedings</i>	title [1..1] _string description [0..1] _string
<i>ApplicationFormLand_useProceedings</i>	title [1..1] _string description [0..1] _string
<i>ApplicationFormStatementOnLocation</i>	title [1..1] _string description [0..1] _string
<i>ApplicationFormStatementOnFinal Approval</i>	title [1..1] _string description [0..1] _string
<i>BuildingPermission</i>	title [1..1] _string description [0..1] _string
<i>CompliantAttitudeOfOwnerOfSite</i>	title [1..1] _string description [0..1] _string
<i>CopyOfLandRegisterMap</i>	title [1..1] _string description [0..1] _string
<i>DeveloperOfProjectDocumentation</i>	title [1..1] _string description [0..1] _string
<i>Fee</i>	title [1..1] _string amount [0..*] _decimal currency [0..*] _string
<i>GeometricalPlanForRe_fixingStructure</i>	title [1..1] _string description [0..1] _string
<i>LandCertificate</i>	title [1..1] _string description [0..1] _string
<i>Location</i>	title [1..1] _string description [0..1] _string
<i>NecessaryVerificationChecks</i>	title [1..1] _string description [0..1] _string
<i>NecessaryVerificationChecksResults</i>	title [1..1] _string description [0..1] _string
<i>NotificationOfBeginningOfBuildingProceedings</i>	title [1..1] _string description [0..1] _string
<i>NotificationOfBeginningOfLand_usePlanning</i>	title [1..1] _string description [0..1] _string
<i>NotificationOfSuccessfulApplicationForLand_useProceedings</i>	title [1..1] _string

	description [0..1] _string
<i>PermissionMergedProcedure</i>	title [1..1] _string description [0..1] _string
<i>PermissionOfSmallBuilding</i>	title [1..1] _string description [0..1] _string
<i>ProjectDocumentation</i>	title [1..1] _string description [0..1] _string
<i>ProofOfLandOwnership</i>	title [1..1] _string description [0..1] _string
<i>Q1</i> Description: The concept for predefined queries. Questions: “Do you already have a site for your house?”, “Do you have some certificate proving your disability or is your case considered as an exemption?”, “Are you legal entity?”, “What type of building do you want to build?”	title [1..1] _string q1 [0..*] _boolean q2 [0..*] SubjectType q3 [0..*] _boolean q4 [0..*] TypeOfBuilding
<i>Q2</i> Description: The concept for predefined queries. Question: “Is your building of the same type as it is expected from the zone plan?”	title [1..1] _string q1 [0..*] _boolean
<i>Q3</i> Description: The concept for predefined queries. Questions: “Are you owner of the site for your new house?”, “Do you already have a project documentation for your new house?”	title [1..1] _string q1 [0..*] _boolean q2 [0..*] _boolean
<i>StatementOnApproval</i>	title [1..1] _string description [0..1] _string
<i>StatementOnLocalisation</i>	title [1..1] _string description [0..1] _string
<i>StatementOnLocatingNewBuilding</i>	title [1..1] _string description [0..1] _string
<i>SubjectType</i>	title [1..1] _string Instances: - corporateBody - naturalPerson
<i>TenancyAgreement</i>	title [1..1] _string description [0..1] _string
<i>TypeOfBuilding</i>	title [1..1] _string description [0..1] _string
<i>VerdictsOrStatementsOrAgreementsOrReviewsOfRelevantActors</i>	title [1..1] _string description [0..1] _string

The domain-specific information described in the *Domain ontologies* are used in both Access-eGov client-side tools – in the Annotation tool (see section 5.2), as well as in the Personal Assistant client tool [D5.1]. The conceptual descriptions of application forms, documents, questions, etc., for all three pilot applications are predefined (in the case of enumeration types with known values, as e.g. a list of locations in the *SHGLocations ontology*), or instantiated in the Annotation tool (as it is described in the next section). The concepts from the *Domain ontologies* are then referenced in the process model of life events and goals (see sections 3.7 and 4.1), as specifications of inputs and outputs for particular workflow elements. Finally, the

instanciated data of these concepts are presented to users (citizens and businesses) in the Personal Assistant client tool, during the execution of particular process models.

4.4 Creating instances

The resource ontologies define the classes (concepts) that are used as a framework for semantic annotation of governmental services [D7.2]. Annotation authors can use a specialised annotation tool (described in the chapter 5 below) to semantically describe a service, i.e., to create the Service Profile. The resulting semantic annotation is represented as a set of instances of the classes defined in the resource ontologies. So the instances contain the data values that are used (consumed) by Access-eGov system components [D3.2] and are provided to the users (citizens) via the interface of the Personal Assistant client [D5.1]. The following subsections of this chapter describe the process how instances are created and applied in the tools, based on the ontology implementation.

4.4.1 Class hierarchy view

This section provides a structural overview of the Access-eGov domain model classes. The classes are grouped by their super-class, i.e. “*Activity*” has two sub-classes “*Client Activity*” and “*Government Activity*”. Some top-level classes are not expected to have instances and are therefore marked as abstract indicated by “(***)”.

The column labels have the following meaning:

- **OA Creates Subclasses:** A mark in this column means that Ontology Authors (OA) (as defined in [D7.2]) may create subclasses of this class. For example, OA may decide to create subclasses of *Certificate* in order to create a class hierarchy of Certificates.
- **OA Creates Instances:** A mark in this column means that Ontology Authors may create instances of this class. For example, OA may create “*Passport*” as an instance of “*Photo ID Document*”.
- **AA Creates Instances:** A mark in this column means that Annotation Authors (AA) (as defined in [D7.2]) may create instances of this class. For example, AA will create instances of the *Place* class by specifying the address of particular public administration office.
- **AA Uses Instances:** A mark in this column means that Annotation Authors use instances of this class. For example, AA use *Service Goals* by selecting a *Service Goal* when a new *Government Service* is created.
- **AA Edits Instances:** A mark in this column means that Annotation Authors edit instances of this class. In general, Annotation Authors should be restricted (by the Annotation tool) to edit only the instances that were created by themselves (not by other Annotation Authors).
- **PAC Creates Instances:** A mark in this column means that the Personal Assistant client (PAC) will create instances of this class. Most of the classes with a mark in this column should be managed by the PAC on a per-session basis.

Table 1. Class hierarchy

Class	OA Creates Subclasses	OA Creates Instances	AA Creates Instances	AA Uses Instances	AA Edits Instances	PAC Creates Instances
Activitiy (***)						
• Activity: Client Activity	(***)					
• Activity: Government Activity	(***)					
Administrative Topic	X	X		X		
Case-Based Client Process						X
Document	X	X				
• Document: Certificate	X	X				
• Certificate: Photo ID Document	X	X				
• Document: Form	X	X				
• Document: Notification	X	X				
• Document: Payment Receipt	X	X				
Goal (***)						
• Goal: Life event	X	X				
• Goal: Service Goal	X	X		X		
• Goal: User need	X	X				
Government Service			X	X	X	
Law	X	X				
Organizational Entity (***)						
• Org. Entity: Administration	X	(***)	X	X	X	
Role (***)						
• Role: Administration Officer	X	(***)	X	X	X	
• Role: Citizen						X
Place	X		X	X	X	
• Place: Country	X		X	X	X	
• Place: State	X		X	X	X	
• Place: City	X		X	X	X	
• Place: Street	X		X	X	X	
• Place: House Number	X		X	X	X	
Resource (***)						
• Resource: Address	X		X	X	X	
• Resource: Access point	X		X	X	X	
• Resource: Description	X	X	X	X	X	

Class	OA Creates Subclasses	OA Creates Instances	AA Creates Instances	AA Uses Instances	AA Edits Instances	PAC Creates Instances
• Resource: Case-based list of documents	X					X
• Resource: Case-based process plan	X					X
• Resource: Good practice process pattern	X	X				
Rule	X	X				
Time	X		X	X	X	
User profile						X

4.4.2 Class view by usage

This section presents a different view of the Access-eGov domain model, as described in sections 4.1-4.4 above. Top-level abstract classes have been omitted (since they are not expected to have instances) and the classes are grouped by their usage. There are seven different “usage groups” of classes, as indicated by the different colours.

The additional column labelled “List of instances available?” indicates whether a list of instances is available from the user partners, whether it is at least in principle possible to create a complete list or whether this is not possible.

Table 2. Classes by usage

Class	OA Creates Subclasses	OA Creates Instances	AA Creates Instances	AA Uses Instances	AA Edits Instances	PAC Creates Instances	List of instances available ?
Government Service			X	X	X		Yes, partially
Resource: Description	X	X	X	X	X		Yes, partially
Org. Entity: Administration	X	(**)	X (**)	X	X		In principle, yes
Role: Administration Officer	X	(**)	X (**)	X	X		No
Time	X		X	X	X		in principle, yes
Place: Country	X		X	X	X		In principle, yes
Place: State	X		X	X	X		In principle, yes
Place: City	X		X	X	X		In principle, yes

Class	OA Creates Subclasses	OA Creates Instances	AA Creates Instances	AA Uses Instances	AA Edits Instances	PAC Creates Instances	List of instances available ?
Place: Street	X		X	X	X		In principle, yes
Place: House Number	X		X	X	X		In principle, yes
Resource: Address	X		X	X	X		In principle, yes
Resource: Access point	X		X	X	X		No
Administrative Topic	X	X		X			No
Goal: Service Goal	X	X		X			No
Rule	X	X					No
Document	X	X					No
Document: Certificate	X	X					No
Certificate: Photo ID Document	X	X					No
Document: Form	X	X					No
Document: Notification	X	X					No
Document: Payment Receipt	X	X					No
Goal: Life event	X	X					Yes, partially
Goal: User need	X	X					No
Resource: Good practice process pattern	X	X					No
Law	X	X					In principle, yes
Resource: Case-based list of documents	X					X	No
Resource: Case-based process plan	X					X	No
Role: Citizen						X	No
User profile						X	No
Case-Based Client Process						X	No
Activity: Client Activity	(*)						No
Activity: Government Activity	(*)						No

4.5 Application of resource ontologies

Annotation tool (see chapter 5) and Personal Assistant client [D5.1] are the main Access-eGov applications with a user interface. These two tools provide the means for usage and manipulation of the ontology resources. The following two subsections describe how these tools will use the classes and instances of the implemented ontologies.

4.5.1 Use of classes in Annotation tool by Annotation Author

An overview of the classes and instances that will be available to Annotation Authors from within the Annotation tool is described in Table 3. The column “Instances available?” indicates if the Annotation Tool should provide a *predefined* list of instances for this class. “Yes” indicates that a list needs to be made available to the Annotation Author through the Annotation tool. For example, a list of Service Goals must be available because an Annotation Author should only be able to create Government Services that match one or more Service Goals. Thus, the Annotation Author must be able to choose the appropriate Service Goal from a list.

Table 3: Class usage in Annotation Tool

Class	AA Creates Instances	AA Uses Instances	AA Edits Instances	Instances available?
Government Service	X	X	X	No (*)
Resource: Description	X	X	X	Partially
Org. Entity: Administration	X (*)	X	X	In principle, yes
Role: Administration Officer	X (*)	X	X	In principle, yes
Time	X	X	X	In principle, yes
Place: Country	X	X	X	In principle, yes
Place: State	X	X	X	In principle, yes
Place: City	X	X	X	In principle, yes
Place: Street	X	X	X	In principle, yes
Place: House Number	X	X	X	In principle, yes
Resource: Address	X	X	X	In principle, yes
Resource: Access point	X	X	X	No
Administrative Topic		X		Yes
Goal: Service Goal		X		Yes

(*) Services are created by choosing a Service Goal.

From Table 3, three different types of usage modes of the Annotation Tool can be derived, based on the availability of instances and their use:

1. **Instances available? Yes. Mode: Using existing instances only.** In this case the Annotation Author should be able to select one or more instances from a set of existing instances. The existing instances must have been created beforehand outside the Annotation Tool. Applies to: Administrative Topic and Service Goal.
2. **Instances available? In principle, yes or partially. Mode: Using existing instances and/or creating new instances.** In this case the Annotation Author should be able to choose existing instances from a set and / or create a new instance. For example, when assigning an Organizational Entity as a service provider to a Government Service, the Annotation Author can either choose an existing Administration instance or create a new one.

3. **Instances available? No. Mode: Creating instances:** In this case there are no instances predefined outside the Annotation Tool and the Annotation Author has to create all instances her- or himself.

4.5.2 Use of classes in Personal Assistant by Citizen

The classes and instances that will be available to user through Personal Assistant Client (PAC) are listed in Table 4. The column “Instances available?” indicates if the PAC should provide a *predefined* list of instances for this class.

Table 4: Class usage in Personal Assistant Client

Class	PAC Creates Instances	PAC Uses Instances	PAC Edits Instances	View Name (VN)	List of instances available?
Government Service		X		Service selection, Goal Description	No
Resource: Description		X		VN Component, Goal selection	No
Org. Entity: Administration		X		Service Details Panel, Service Selection	In principle, yes
Time		X		Service Details Panel,	No
Place: Country		X		VN Component, User Profile Edition	Yes
Place: State		X		VN Component, User Profile Edition	Yes
Place: City		X		VN Component, User Profile Edition	Yes
Place: Street		X		VN Component, User Profile Edition	Yes
Resource: Address, Place: Address		X		VN Component, User Profile Edition	Partially
Resource: Access point		X		Service Details Panel	No
Resource: Case-based list of documents	X	X	X	VN Component	No
Resource: Case-based process plan	X	X	X	VN Component	No
Administrative Topic		X		VN Component	No
Goal		X		VN Component	In principle, yes
Document		X		VN Component	No
Document: Certificate		X		Requirement Panel, Outcome Panel	No
Certificate: Photo ID Document		X		Requirement Panel	No
Document: Form		X		Customization Panel, Requirement Panel	No
Document: Notification		X			No
Document: Payment Receipt		X		Requirement Panel	No

Class	PAC Creates Instances	PAC Uses Instances	PAC Edits Instances	View Name (VN)	List of instances available?
Resource: Good practice process pattern		X		Description Panel	No
Law		X		Description Panel	In principle, yes
Person	X	X	X	User Profile Edition, Description Panel	Partially
User profile	X	X	X	User Profile Edition	Partially
Case-Based Client Process	X	X	X	User Profile Edition, VN Component	No
Activity: Client Activity	X	X	X	VN Component	No

From Table 4, three different types of usage modes of the Personal Assistant Client can be derived, based on the availability of instances and their use:

1. **Instances available? Yes. Mode: Using existing instances only.** In this case the user should be able to select one or more instances from a set of existing instances.
2. **Instances available? In principle, yes or partially. Mode: Using existing instances only. Using existing instances and/or creating new instances.** In this case the user should be able to choose existing instances from a set and / or create a new instance.
3. **Instances available? No.** In this case there are no instances predefined outside the PAC and the User has to create all instances her- or himself.

Following table presents an estimation of quantitative data – a number of instances being created within the pilot applications:

	SHG-GE pilot	GLI-PL pilot	KSR-SK pilot
Service Profiles ontology	<ul style="list-style-type: none"> - about 10 instances of service types - for each of service type, about 30-40 instances are expected 	<ul style="list-style-type: none"> - about 20 instances of service types - for each of service type, about 10-20 instances are expected 	<ul style="list-style-type: none"> - about 50 instances of service types - for each of service type, about 20-30 instances are expected
Life events ontology	<ul style="list-style-type: none"> - 1 life event consisting of 11 subgoals was specified. 	<ul style="list-style-type: none"> - 1 life event consisting of 5 subgoals was specified. 	<ul style="list-style-type: none"> - 1 life event consisting of 43 subgoals was specified.
Domain ontology	<ul style="list-style-type: none"> - about 500 instances in SHG ontology - about 2500 instances in SHGLocations ontology, about 1500 instance relations 	<ul style="list-style-type: none"> - about 150 instances in GLIOntology - about 900 instances in GLILocations ontology, 1325 instance relations 	<ul style="list-style-type: none"> - about 250 instances in KSR ontology

The *Life events ontology* and *Domain ontology* were already implemented (conf. also the WSM ontology implementation that is attached to this deliverable) and tested for all the pilots, so the numbers are quite exact. The estimations for *Service Profiles ontology* are based on the description of trials [D8.2] as well as on the experiments held during the testing of the Annotation tool (see section 5.2 and chapter 6).

5 Semantic annotation of services

The conceptual model, i.e. the set of resource ontologies, provide a background framework for describing the governmental services on a semantic level. The semantic annotation means then the actual usage of the resource ontologies for specification of a particular service, its functional and non-functional properties as preconditions, inputs and outputs, and selection of workflow sequence for the annotated service. From the technical point of view, the semantic annotation can also be seen as a creation of instances of the classes taken from the conceptual model. It is obvious that the process of semantic annotation is very complex and requires proper *methodology* as well as *annotation tools* that are easy to use for public administration employees, without any expert-level knowledge of semantic technologies.

5.1 Functionality and user roles for the semantic annotation

The use-case diagram for the overall process of semantic annotation, as it was presented in [D3.2], introduces on the side of public administration the role of an Annotator and specifies four activities for this role:

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

Activities related to the semantic annotation were deeply analysed, elaborated and described in detail in [D7.2], where the role of Annotation Author was specified as a person responsible for creating and maintaining a Service Profile. In addition, nine other roles (Annotation Manager, Annotation Reviewer, Ontology Manager, etc.) were identified for particular levels of the semantic annotation process. The roles for the annotation process itself, i.e. the *Annotation Author*, *Annotation Manager*, and *Annotation Reviewer*, were implemented directly into the annotation tool as the roles *editor*, *publisher*, and *viewer* (see next section). The rest of the roles proposed in [D7.2] are responsible for ontology creation and manipulation; this functionality is currently not supported by any Access-eGov tool and needs to be handled by using the WSMO Studio software package only. The schematic representation of the whole annotation process from the side of information providers was presented in [D7.2], including supporting methods and tools.

Besides this methodological framework, the 7-step procedure of collecting requirements and information needs from users, described in chapter 3, can also be seen as a part of methodology

for semantic annotation, especially in the part of ontology modelling and management. It gives a possibility to involve the public administrations, as service providers, into the process of creation of the resource ontologies; so the employees of public administrations are able to understand the ontological structure and use it properly for semantic description of their services. Other such methods, also employed during the design of Access-eGov ontologies, were the verification of ontology suitability on real data (see section 3.8) as well as development of mock-up version of the Annotation tool, described in the next section.

It is assumed that the semantic annotation of services will be managed by public administration employees with very limited or no knowledge of semantic technologies. So the annotation tool should be easy-to-use and should require only standard PC skills; certainly no WSMML statements should be displayed on the screen during the annotation.

To fulfil this requirement and to follow the methodology approaches mentioned above, the ontology was implemented (see chapter 4) with regard to ease the semantic annotation process. Functional properties of the services, inputs, outputs, preconditions, and effects, as well as workflow sequences of life events and goals, were predefined and stored as instances into the *Life events ontology* and cannot yet be edited by public administration workers in the developed annotation tool. According to the workflow scenarios, the service types (templates) were identified and implemented as instances of the *ServiceTemplate* class, stored in the *Service Profiles ontology*. The annotation of a service then lays on a selection of proper template from a list of available templates and a specification of non-functional properties for this service (as an instance of the *ServiceNFP* class from *Service Profiles ontology*).

5.2 Annotation tool

The annotation tool was designed in the Access-eGov project as a standard web application. The *first version* was a client-server application using the JSP and MySQL DB technologies (user interface is depicted on Figure 5). This first version served as a mock-up for user partners of the project and influenced significantly their inputs to the 7-step procedure described above, especially within steps 2 and 3. As advantages of this tool, we can mention the possibility to check and verify the results of annotation immediately after a modification. The results of the annotation process are available in an HTML preview (as they will be provided to citizens) as well as in WSMML notation (as they will be provided to other client applications via web service interfaces).



Figure 5. User interface of the annotation tool, the first version. List of annotated services for the SHG-GE pilot application

However, the solution based on a relational database had a significant disadvantage: the database structure and related forms on the web interface did not reflect the WSMML definitions stored in the ontology. It was especially obvious in the phase of ontology development, when every change of non-functional properties required modifications in both data structure and user interface. So the *second version* of the annotation tool was designed and implemented using the WSMO object model and JSF technology [AT AeG] (Figure 6).

Figure 6. Annotation tool, the second version. Annotation of a service in the SHG-GE pilot application

Some new features were added to this version, namely:

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

A templates mechanism was introduced to ease the maintenance of workflow sequences for the annotated services. Annotators just select the proper type of service by clicking the provided button (“add a new service of this type”) to assign a particular type to this service and to make it available in given scenario requiring this type of service.

Non-functional properties were updated and fixed for the first trial, according to the feedback obtained from project partners. The possible values for those properties with pre-defined values (e.g. list of towns in a region, etc., see section 4.3) were formalised in WSML and were implemented into the interface. Also once created instances, like contact persons, can be reused for the annotation of different services of the same organization by selecting them from a list (e.g. of contact persons) which can be created using the tool. The web interface is dynamically generated according to the structure of non-functional properties formalised in WSML. This approach allows an easy customization of the interface, by modifying the structure of the non-functional properties stored in the ontology.

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

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

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

The WSML instances of the annotated services are accessible for various client-side tools via a web service interface. The *Personal Assistant* client [D5.1] is envisioned to be one of such tools, providing browsing, discovery, and execution capabilities of proper services for citizens / businesses according to the specified life event or goal.

6 Conclusion and outlook

The process of modelling and creation of resource ontologies for the public administration domain was presented in this deliverable. The conceptual model and consequent ontology implementation, as described in chapter 4, can be considered as an ontological foundation for the Access-eGov platform, components and applications. The development of this foundation has followed a user-oriented and requirement driven methodology (chapters 2 and 3). However, the utilization of this foundation will be monitored during the trials, and it is expected that improvements will be necessary. From today's view, the following issues are expected to be on the agenda for further ontological development:

- **Service versus Process:** Process refers to the ordered set of activities and does not include process description and process plans (which are defined as Resource). However, currently the client process can also invoke services, i.e. there is a conceptual mixture of the process and service paradigm. Up to now, no problems are foreseen, but further development should define a clear relation between processes and services.
- **Collaborative management of process planning and control:** the *client process* is a central entity in the Access-eGov model. It denotes the process from the point of view of the citizen or business users (client) and how this process relates to the administrative activities. The challenge is that for individual (i.e. non-standard) processes the activities administration officer in charge – based on accepted good practice process patterns – and the client share in planning and controlling the process. While the Personal Assistant is designed to support this for the client, it is not yet clear how to involve any responsible administration officer in the computer-based client process management.

The Annotation tool as a web-based application for semantic annotation of governmental services was described in the chapter 5. More detailed description of functionality and proposed usage of the Annotation tool is published in the user manual, released as a separate deliverable [AT AeG]. The Annotation tool was developed in a tight co-operation between developers and user partners, and was successfully tested in October 2007 by all three public administrations involved in the Access-eGov project. Following issues were identified as possible enhancements in the next period:

- **Data binding** between the web resources existing on the side of service providers (public administrations) and particular properties of semantically annotated services. This feature is required to avoid data duplicity and necessity to manage several resources with the same content (on the side of public administration's web site as well as in the Access-eGov resource ontologies). The data-binding feature of the Annotation tool will enable to reuse the existing web content by means of semantic annotation thereof.
- **Visual maintenance and editing of functional properties** of the annotated services, including the workflow sequences, inputs, outputs, preconditions and effects of the services, scenarios of complex goals and life events, etc. To solve this functionality properly is a challenging task, since the complexity and variability of the workflow data structures should be harmonised with the requirement to keep the user interface of the Annotation tool simple and user-friendly.
- **HTML preview of the contents** and simple HTML formatting of values like service descriptions, which contain long texts, should be made available in the Annotation tool..

The data binding functionality is expected already for the first trial, while the visual administration of functional properties and workflow sequences will be investigated within the second trial of Access-eGov pilot applications.

References

- [Apostolou et al 2005] Apostolou, D., Stojanovic, L., Pariente Lobo, T., Thoenssen, B.: Towards a Semantically-Driven Software Engineering Environment for eGovernment. In: M. Böhlen et al. (Eds.): TCGOV 2005, LNAI 3416, pp. 157 –168, 2005.
- [AT AeG] Hreno, J.: Annotation Tool Trial 1 Help Template. Additional deliverable of the Access-eGov IST project (Draft, version 1.0). Technical University of Kosice, November 8, 2007. Available at http://www.accessegov.org/acegov/uploadedFiles/webfiles/cffile_11_8_07_4_45_13_PM.doc.
- [D2.2] Klischewski, R. et al.: Deliverable D2.2: User Requirement Analysis & Development / Test Recommendations. Public deliverable of the Access-eGov IST project. German University in Cairo, July 17, 2006.
- [D3.1] Bednar, P. et al.: Deliverable D3.1: Access-eGov Platform Architecture. Public deliverable of the Access-eGov IST project. Technical University of Kosice, October 4, 2006.
- [D3.2] Tomasek, M. et al.: Deliverable D3.2: Access-eGov Components Functional Descriptions. Public deliverable of the Access-eGov IST project. InterSoft, a.s., March 30, 2007.
- [D4.1] Schillinger, R. et al.: Deliverable D4.1: Specification of components for markup services. Deliverable of the Access-eGov IST project. University of Regensburg, April 5, 2007.
- [D5.1] Mach, M. et al.: Deliverable D5.1: Specification of components for "Personal Assistant". Deliverable of the Access-eGov IST project. Technical University of Kosice, March 30, 2007.
- [D7.2] Ukena, S., Klischewski, R.: Deliverable D7.2: Guidelines for Semantic Mark-Up of e-Government Resources. Deliverable of the Access-eGov IST project. German University in Cairo, December 5, 2006.
- [D8.2] Klischewski, R. et al.: D8.2 Specification of Pilot Applications and Design of Trials. Deliverable of the Access-eGov IST project. German University in Cairo, April 19, 2006.
- [DIP D9.3] Gutierrez, L. et al: D9.3: e-Government ontology. Public deliverable of the DIP project, FP6-507483. Essex County Council, 2004.
- [EC 2003] European Commission, Linking up Europe: the Importance of Interoperability for eGovernment Services, Commission Staff Working Paper, 2003. Accessible at: <http://ec.europa.eu/idabc/servlets/Doc?id=1675> [Last accessed in October 2007].
- [EC 2006] European Commission, Communication from the Commission to the Council and the European Parliament. Interoperability for Pan-European eGovernment Services. COM (2006) 45 final, Brussels, 13.2.2006, Accessible at: <http://europa.eu.int/idabc/servlets/Doc?id=24117> [Last accessed in October 2007].
- [Fraser et al 2003] Fraser, J., Adams, N., Macintosh A., McKay-Hubbard, A.: Knowledge Management Applied to e-Government Services: the Use of an Ontology. In: Proceedings of the KMGov2003 Workshop, Rhodes, Greece, 2003.
- [Furdik et al 2007] Furdik, K., Sabol, T., Bednar, P.: Framework for integration of e-Government Services on a Semantic Basis. In: Grönlund, A., Scholl, H. J., Wimmer, M.

- A. (eds.), Electronic Government, 6th International EGOV Conference. Proceedings of ongoing research, project contributions and workshops (September 3-6, 2007, Regensburg, Germany). Trauner Verlag, Linz, Austria, 2007, pp. 71-78, ISBN 978-3-85499-255-4.
- [**Gugliotta et al 2006**] Gugliotta, A., Cabral, L., Domingue, J.: Knowledge Modelling for Integrating E-Government Applications and Semantic Web Services. AAAI Spring Symposium Series "Semantic Web meets E-Government", Stanford University, California, USA, 2006.
- [**ISO 14977**] R. Scowen: Extended BNF - A generic base standard (EBNF). ISO/IEC 14977:1996.
- [**Kavadias & Tambouris 2003**] Kavadias, G., Tambouris, E.: GovML: A Markup Language for Describing Public Services and Life Events. 4th working conference on Knowledge Management in Electronic Government, Rhodes Island, Greece, 2003.
- [**Klischewski & Ukena 2007**] Klischewski, R., Ukena, S.: Designing Semantic e-Government Services Driven by User Requirements. In: Grönlund, A., Scholl, H. J., Wimmer, M. A. (eds.), Electronic Government, 6th International EGOV Conference. Proceedings of ongoing research, project contributions and workshops (September 3-6, 2007, Regensburg, Germany). Trauner Verlag, Linz, Austria, 2007, pp. 133-140, ISBN 978-3-85499-255-4.
- [**Lee et al 2002**] Lee, W., Strong D. M., Kahn, B., Wang R. Y.: Aimq: A Methodology for Information Quality Assessment. Information & Management. Vol. 40, 2002, pp. 133-146.
- [**Lillrank 2003**] Lillrank, P.: The Quality of Information. International Journal of Quality and Reliability Management. Vol. 20, 2003, pp. 691-703.
- [**Norton et al, 2005**] B. Norton, S. Foster, and A. Hughes.: A compositional operational semantics for OWL-S. In: Proc. of 2nd Intl. Workshop on Web Services and Formal Methods (WS-FM 2005), September 2005.
- [**OntoGov D2**] Stojanovic, L., Kavadias, G., Apostolou, D., Probst, F., Hinkelmann, K.: Deliverable D2: E-Gov Service Lifecycle Ontology. Public deliverable of the OntoGov IST project, June 2004.
- [**Peristeras & Tarabanis 2004**] Peristeras, V., Tarabanis, K.: Governance Enterprise Architecture (GEA) - Domain Models for eGovernance. 6th ACM International Conference on Electronic Commerce (ICEC'04), Delft, The Netherlands, 2004.
- [**Peristeras & Tarabanis 2006**] 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, pp. 56-63.
- [**Rosenfeld & Morville 2006**] Rosenfeld, L.; Morville, P.: Information Architecture for the World Wide Web. O'Reilly & Associates. Sebastopol, CA, USA, 2006.
- [**Skokan & Bednar 2006**] Skokan, M., Bednar, P.: Access-eGov architecture. In: Mikulecký, P., Dvorský, J., Krátký, M. (eds.), Proceedings of the 6th annual conference Znalosti (Knowledge) 2007 (Ostrava, Czech Republic, February 2007). Faculty of Electrical Engineering and Computer Science, VŠB - Technical University of Ostrava, Czech Republic, 2007, pp. 384-390, ISBN 978-80-248-1279-3.

- [**Skokan & Bednar, 2007**] Skokan, M., Bednár, P.: Semantic Orchestration of Services in E-Government. 2007 (submitted to the SAMI 2008 - The 6th Symposium on Applied Machine Intelligence, Herľany, Slovakia, <http://www.sami.tuke.sk>).
- [**Stojanovic et al 2004a**] Stojanovic, L., Abecker, A., Stojanovic, N., Studer, R.: An Approach for the Change Management in the EGovernment Domain. In: Proceedings of Second International Conference on Knowledge Economy and Development of Science and Technology (KEST'04), Beijing, China, 2004.
- [**Stojanovic et al 2004b**] Stojanovic, L., Abecker, A., Stojanovic, N., Studer, R.: On Managing Changes in the ontology-based EGovernment. In: Proceedings of the 3rd International Conference on Ontologies, Databases and Application of Semantics (ODBASE 2004), Larnaca, Cyprus, October 2004.
- [**TerreGov D1.7**] Barthès, J.P., Moulin, C.: TERREGOV Research Conclusions for 2006. Public deliverable D1.7 of the TerreGov project, 2007.
- [**Uschold et al 1998**] Uschold, M., King, M., Moralee, S., Zorgios, Y.: The Enterprise Ontology. The Knowledge Engineering Review , Vol. 13, Special Issue on Putting Ontologies to Use (eds. Mike Uschold and Austin Tate), 1998.
- [**Wand & Wang 1996**] Wand, Y., Wang R. Y.: Anchoring Data Quality Dimensions in Ontological Foundations. Communications of the ACM. Vol. 39, 1996, pp. 86–95.
- [**Wang et al 2007**] 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.
- [**WSMO**] 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].