Collaboration and the Knowledge Economy: Issues, Applications, Case Studies
Paul Cunningham and Miriam Cunningham (Eds)
IOS Press, 2008 Amsterdam
ISBN 978–1–58603–924-0

eGovernment Interoperability with Open Standards

Wout HOFMAN
TNO, Brassersplein 2, Delft, 2600 GB, the Netherlands
Tel: +31 15 2857129, Fax: + 31 15 2857349, Email: wout.hofman@tno.nl

Abstract: Governments have the objective of using the Internet as one of the channels for communicating with citizens and companies. They launch quite a number of initiatives to do so, both nationally and in international (EU) projects. On the one hand, many of these initiatives lack a clear architecture in themselves and coherence with other initiatives. On the other hand, high investments are made in developing those incoherent solutions. This paper presents a government interoperability framework based on an existing architectural framework. The framework semantically specifies information requirements as the basis for electronic interfaces supported by open standards. This paper will show how an architectural approach will lead to a consistent and clear specification of a relatively minimal set of services. We will show how application semantics can be implemented by various technological solutions such as natural language systems for service discovery and web services. The paper refers to solutions and laws that are implemented in the Netherlands.

1. Introduction

It is the objective of many governments to increase customer satisfaction by implementing multi channel service delivery [1]. Government services should not only be accessible via paper, but also via the Internet [2]. It is further stated that public information and services of all government organizations must be accessible via each government portal ('no wrong door', also stated in the EU Service Directive [13]). Since each government organization will have its own portal, a complete distributed environment has to be realized based on a service-oriented architecture [3]. This paper presents a government interoperability framework specifying application semantics utilising concepts of the architectural framework Archimate [7]. We will illustrate how this framework is the basis for specifying electronic interfaces by separating data and process aspects of those interfaces.

First, we will present government information and government services in this chapter. Government is also known as public. Therefore, public service is identical to government service like public information is the same as government information. Secondly, we will discuss international developments in the context of government services and standardization.

1.1 Government information and government services

Government information is for instance laws and regulations, policy documents, zoning scheme, and building permits. A government distinguishes the following information:

- Policy information. All information that is output of internal, policy-making processes. This information is the basis for government service specification, e.g. government services are based on a particular law or local regulation, including any restrictions to their usage.
- Operational information. This type of information is in general the input or output of government services initiated by citizens or companies. A number of government

organizations offer identical services, e.g. municipalities all offer permit services. A citizen or company has to select the proper service provider for a specific region of authority of that provider.

- Reference data, which is relative stable for the operation of a government organization, e.g. its organizational structure, offices and their opening times, and buildings, companies and citizens living in their area of control. The latter is especially of relevance to local municipalities.

Currently, the Dutch government considers public information to be documents that should also be accessible for citizens and companies via the Internet, irrespective of their relation to a particular government service. These documents are resources that need tags to improve discovery. Dublin Core standards are being applied to add metadata to html pages and documents and a search engine is installed for disclosure of the information to citizens and companies [2]. Public information is organized in different collections, such as permits and products, each with different metadata specifications. Taxonomies are implemented for information access via the Dutch Government portal (www.overheid.nl). The Dublin Core Metadata Initiative (DCMI) basically focuses on describing documents, web pages, and records, but offers also a loosely defined framework for service tagging.

Government services are for instance 'passport', 'building permit', and 'social security support'. Government services relate to the public tasks assigned to a government organization and are directly related to the applicable laws and (local) regulations. With respect to government services, the following issues arise:

- Implementation issue: not all government organizations have specified their services explicit, stored and disclosed these services in a structured way, e.g. the organizations for social security only have services specified in words using HTML format.
- Service discovery based on discrete events: discrete events in the life of a person [4] or for a company [5] are being specified and implemented. These events group services accordingly. The specification of all possible events is time consuming and does not cover all possible (combination of) events.
- Service composition: services of different government organizations are related to each other. Service composition should be possible per user request or on the basis of intergovernmental agreements. In most cases, it is currently only supported by hyperlinks. It implies that a citizen or company has to enter the same information more than once.

One could argue that documents describing government services should themselves not be seen as resources, but as the basis for finding a service. Thus, documents would only need a limited set of tags, e.g. document type and author. But there is more information relevant to a government service. Because a clear information model is missing, in most occasions more tags are added to the document than necessary, e.g. tags identifying related government services. The contents of documents can also be used for discovery of an appropriate government service, since a number of documents specify a service in more detail. Once the appropriate government service is found, one should have the ability to initiate application services. We will elaborate further on this approach in this paper. We will show that government services and documents describing those services are the basis for application services. An information model associates application services to each other.

1.2 International developments

On the EU community level, there has been a number of projects that deal with government services. The most recent one is still in progress: Access-eGov (www.accessegov.org). The state of the art report [11] lists a number of similar initiatives, e.g. Terregov and OntoGov. The Terregov project (www.terregov.eupm.net) focussed on the publication of web services and the use of these web services by government organizations. The OntoGov project

(www.ontogov.com) developed a prototype for specifying government services based on government service ontology. The service ontology specifies both static and dynamic aspects of government services. The static aspects consist of concepts like ‘language’, ‘format’, ‘source’, ‘type’, ‘date’, ‘title’, ‘description’, ‘is of a topic’, and ‘has contributor’, ‘creator’, and ‘publisher’. These concepts are similar to tagging documents according the DCMI, like we discussed before. Additionally, a government service refers to a law and/or article of a law. The dynamic parts refer to the process, e.g. a pre condition, a post condition, and the structure of a process.

In this respect, OntoGov models in- and outputs conceptually. These can be documents, messages, etc. In addition, Access-eGov adds aspects like fees and non-functional properties like spatial and temporal availability. They have not yet incorporated these aspects in an ontology framework. Other developments such as CityGML model structures in cities with spatial and temporal aspects with UML class diagrams [16]. These structures include all relevant information objects of cities like buildings, streets, and addresses.

On the level of Member States and the EU organization itself, interoperability frameworks are (being) developed. These frameworks define generic concepts for interoperability and choices for open standards to be used. They also include aspects like management and versioning of standards; see for instance the eGIF of the UK [15]. These aspects are all relevant, but the frameworks lack application specific semantics that could be common to many government organizations and is technology independent (see for instance http://europa.eu.int/idabc/3761 for the EU Interoperability Framework).

2. Objectives

It is our objective to offer a government interoperability framework based on a (high level) specification of government semantics as the basis for development of application services of government organizations. The semantic model presented in this paper will be constructed by combining the results of existing Dutch and international projects such as Dutch transparency projects, the construction registers for reference data, and EU projects like OntoGov for defining public services. By separating semantics and process aspects of these application services, and implementing the process aspects with web services, it will be possible to offer an evolutionary scenario for implementing those web services. Evolutionary means that semantics can gradually be enriched, whereas the web services structures remain unchanged and the message structures will change.

3. Methodology

We will apply the Archimate framework as a reference for developing electronic interfaces. Since the Archimate Foundation and the Open Group joined efforts, Archimate is probably going to be adopted as an international standard. Whereas we want to focus on applying architectural concepts, we do not endorse any architectural framework for its usefulness in the context of eGovernment. This section discusses how we apply the Archimate concepts to develop a functional specification to identify all possible electronic interfaces. The next chapter discusses possible technology to support these interfaces.

3.1 Archimate framework

Archimate defines generic concepts (Figure 1) that have been made specific to different views. Each view is represented as a layer. Archimate [7] distinguishes three layers: business layer, application layer, and technical layer. Applying the concept of layering, three different service types are defined (figure 1). Archimate defines a *business service* as the external visible (“logical”) functionality, which is meaningful to the environment and is realized by business behaviour (business function, business process or business interaction).

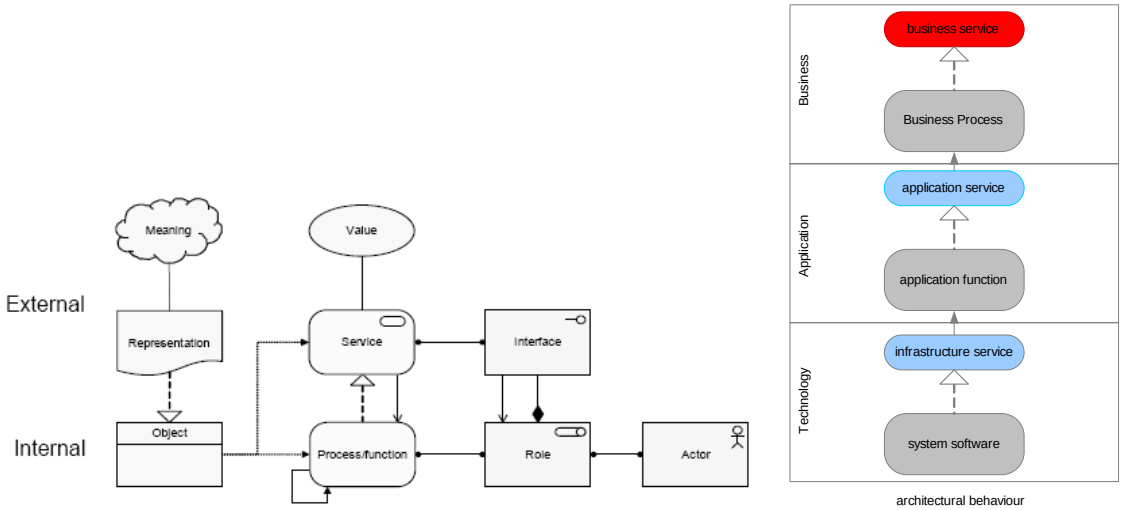

Figure 1. Generic Archimate concepts and service types

A ‘business process’ is a business function as a unit of behaviour that groups behaviour according to required skills and is performed by a single role within an organization. A business process uses application services and supports business services. An application service is defined as an externally visible unit of functionality as provided by one or more components, is exposed through well-defined interfaces, and is meaningful to the environment [7]. The concept of application service fits with the concept of web service. For the purpose of our discussion, we will consider them as equivalent. The concept of infrastructure service is not particular relevant to this article, since this concept is not specific to government.

Besides these notions of different service types, we need notions of interaction between different organizations. Together with business collaboration, business interaction can be used to model a business transaction (see further). Business collaboration is a collective of roles within an organization that perform collaborative behaviour. A business interaction refers to behaviour performed in a collaboration of two or more specific business roles.

3.2 Applying the Archimate framework to eGovernment

The mapping of government concepts to Archimate concepts is relatively simple. A government service is identical to a business services. All other Archimate concepts can be applied directly to government organizations, e.g. a government service is supported by one or more application services.

To develop electronic interfaces, we have to identify business interactions as specified by Archimate. A generic data structure for case management by government organizations that has been developed by the Dutch government [14] is the basis for specifying government semantics. As architectural frameworks like Archimate are applied to structure internal ICT, they have a weak support for defining ‘data’ (see also [7], where a superficial concept for data is given). The case management data structure will support the missing data aspects and identify the appropriate web services.

Figure 2 shows a high level view of government semantics.

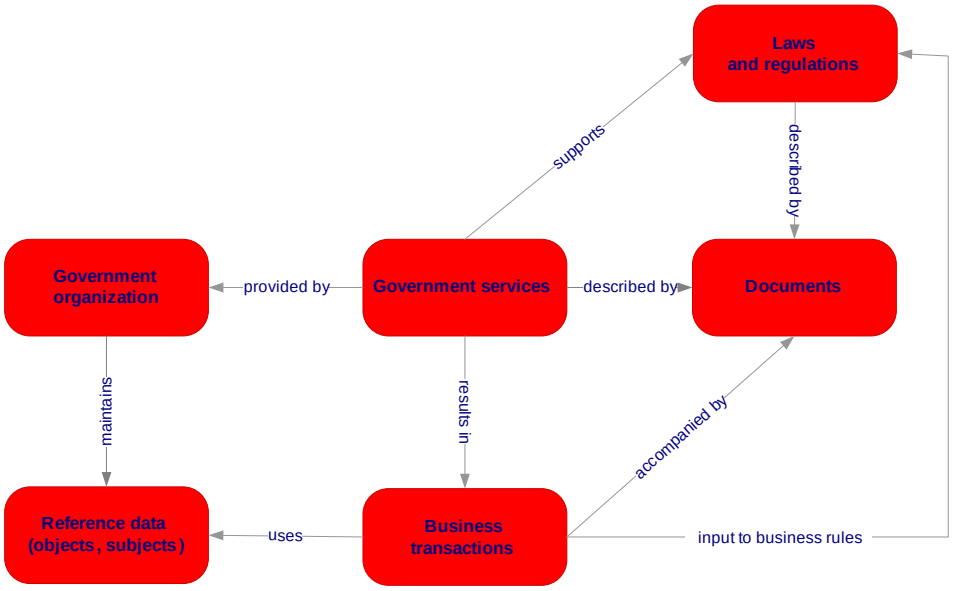

Figure 2. High level government semantics

The concepts shown in Figure 2 can be specialised further and mapped to separate application component. Associations between concepts reflect the associations between those components. The specialisation and components are:

- Reference data can be decomposed into company data, person data, addresses, buildings, geo-graphical data, etc. This type of data is already specified in for instance CityGML (see before). Furthermore, existing application components are in place, e.g. person data is stored in a specific component in the Netherlands (GBA: Gemeenschappelijke BevolkingsAdministratie). Application services are being developed for these components.
- Government organizations can be decomposed into the structure of that organization, personnel, contact information, etc. Not many government organizations currently have an application component in which this type of information is all stored.
- Laws and regulations are most often only stored in documents. Additionally, these documents are disclosed via specialised application components based on text retrieval and structuring (see for instance www.wetten.nl).
- Government services can be decomposed in attributes like service name, service identification, and process aspects like duration and objection period. The information to specify a government service might be identical to that specified by for instance OntoGov (see before). Furthermore, a government service can be composed of other government services, e.g. like the composition of one permit for all current environmental permits currently implemented in the Netherlands. Government services are stored in a so-called product catalogue of a government organization that interfaces with all other product catalogues of other government organizations.
- Documents are most often (or should be) stored in a document management system. Their tags contain data that is also stored in other application components; see the discussion on DCMI tagging.
- We introduce the concept of ‘business transaction’ [6] as the aggregation of data exchanged in all business interactions between a service supplier and service requestor for the delivery of one particular government service. For example, it consists of a permit request, the final permit, and all interactions between those two for the delivery of one specific instance of a government service. A business transaction consists of elements that are required to validate the business rules of a government service. For instance, in case a drivers licence can only be given to persons over 18 years old, the

business transaction should have ‘birth date’ and ‘business transaction date’ as input to calculate the age at the time a drivers licence is requested (see also OntoGov).

We have thus seen that a government service corresponds to a business service, a high level specification of government semantics is feasible and can be mapped to existing application components. The final step in the methodology is the identification of application services. The approach is quite simple: each concept of government semantics needs functionality for entering new data (C: create), viewing existing data (R: read), modifying existing data (U: update), and deleting data (D: delete). Thus, each application service should contain this particular functionality, which is also known as ‘CRUD-actions’. We can thus identify the following high-level application services (see Figure 3):

- Government service description and discovery services.
- Organization structure services, e.g. retrieving the office opening times and locations of a physical front office.
- Law and regulations services for maintenance of a database with laws and regulations, including the possible retrieval of documents describing the laws and regulation by calling the appropriate document services.
- Reference data services. As reference data is probably implemented by more than one application component, these data aspects of those application services will be specific to each specialised data object maintained by that component.
- Business transaction services. These include all business interactions for the request and delivery of a specific government service. It also includes the interactions for data retrieval by others than the service requester to support transparency.

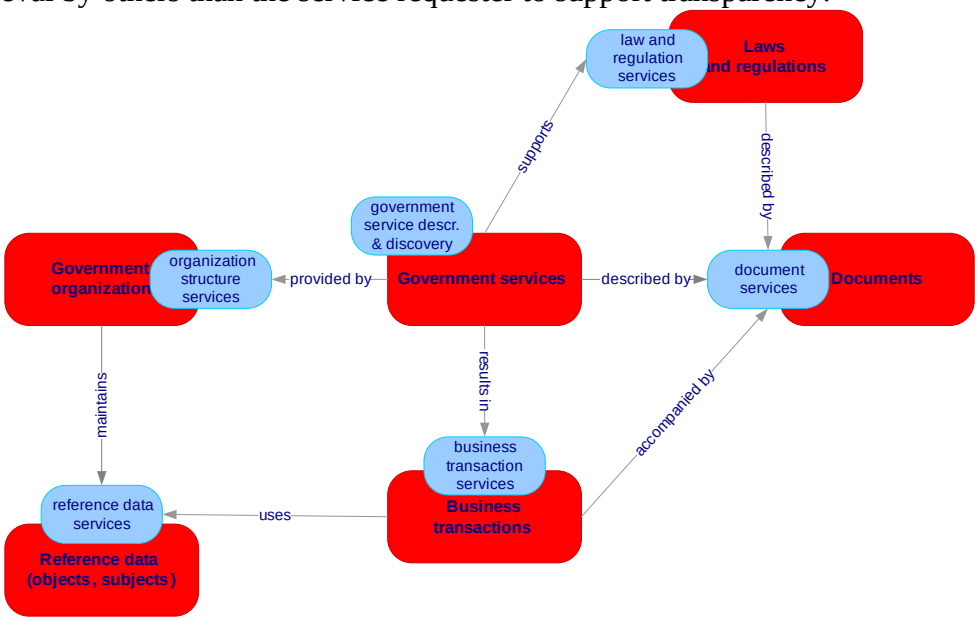

Figure 3. High level application services for eGovernment

There is one important aspect that we need to discuss with respect to application services. As we have stated, concepts of the government semantics need to be specialised. This specialisation defines refined concepts contained by the application services. We have already stated that probably application services will be defined for a specific specialisation, in case the instances of concepts are stored by a separate application component. Refined concepts of business transactions services are however different, since a business transaction service relates to requirements of the associated government service. For instance, a permit business transaction service requires other data than a passport business transaction service. Also a permit business transaction service for building requires different data than one for a restaurant or hotel. By specifying the business transaction

services with CRUD actions and separating concepts of those actions, an evolutionary implementation approach can be taken. One can specify the refined concepts for each government service and refer to those refined concepts in a business transaction service by identifying the related government service. For instance, by relating to the 'building permit' government service in a business transaction service, the refined concepts of that government service are known.

4. Technology considerations

We have identified application services to support government services and have addressed aspects with respect to refinement of concepts for these application services. This section briefly addresses the open standards and technology that could be used to implement the approach:

- Standards and technology for government service description and discovery are for instance WSMO [8] and OWL-S [12]. We have not yet shown any technique for modelling a government service, but have implicitly used terminology of ontology. In this respect, a particular government service like 'permit' is an instance of the concept 'government service' with related concepts like service delivery, etc. WSMO contains other aspects than those required for ontology. It specifies for instance various mediation functions, e.g. for service discovery, service composition and orchestration, etc. The function for service discovery can be implemented by for instance natural language systems [9] with functionality like spell checking, syntax support, etc. The next figure shows a mock-up of a user interface for government service discovery.

  OWL-S specifies a service profile and process. A service profile contains process aspects like delivery of a service, etc. We consider a service profile part of a government service, although an application service also requires a service profile. Possibly, elements of the OWL-S service profile need to be considered to properly specify government services in an eGovernment environment.

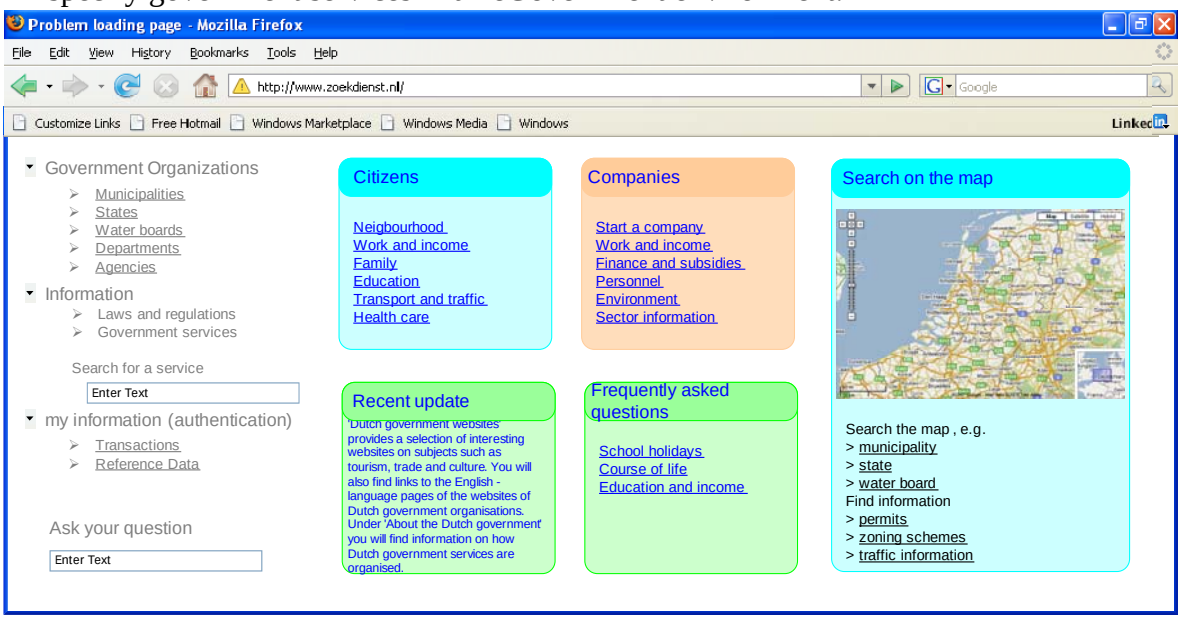

Fig.4. Mock-up of a user interface for government service discovery

- Application services are generic and can be structured with a WSDL [8]. Such a WSDL only specifies the possible operations, not the sequencing of the operations. Process orchestration specifies the internal processing of an application service. Such an internal process orchestration is not required in this context. It is only required to specify the

choreography as the external sequencing of business interactions in a business transaction. Interaction choreography is probably going to be supported by BPM 2.0[i], according to specifications given in the Business Process Definition Metamodel [10]. Choreography is especially of importance to citizens and organizations for business transaction services. This type of service is their only instrument to manage data related to a government service. The business interactions within a business transaction service are basically asynchronous, e.g. entering a request or updating a request. The underlying application services may be synchronous, e.g. requesting data that needs to be updated. One particular set of business interactions will be the retrieval of the status of a business transaction service, which is synchronous. Business interactions from government organizations to citizens and organizations might be implemented by alerts based on for instance RSS or email notifications. They inform a citizen or organization on a status change, in which case the status retrieval part of a business transaction service can be used to retrieve the actual status.

- The concepts of government semantics can be represented by for instance an XML Schema [8], although also other technology is possible for exchanging instances of these concepts (e.g. OWL or RDF Schema). An XML schema, however, can be imported in a WSDL document. By refining and enriching government semantics, the schema will support more functionality. By (automatically) deriving XML Schema('s) from an ontology representing government semantics, the schema('s) will be upwards compatible and, in case there is more than one, coherent.
- Query-response type of application services for data retrieval can be specified differently. In most cases, application developers specify the allowed types of queries and offer an intelligent tool to execute specific queries. In terms of application services, it would be worthwhile to implement queries with for instance XQuery or SparQL [8]. The latter consists of a query language and a protocol for querying implemented by a WSDL. SparQL is specifically developed to query RDF data.
  The query-response pattern is mainly applicable for application services offered to citizens and organizations to support transparency. This pattern does not allow those citizens to change the data they are able to retrieve.

5. Results and business benefits

We have already identified one of the main business benefits, namely evolutionary development and implementation based on separation of semantics and process. Whereas process aspects are specified by WSDL documents with their choreography, and/or other mechanisms like RSS, these WSDL documents import semantics as XML Schema.

Other relevant results and business benefits are:

- The specification of government and application services is independent of a technology. Currently, XML Schema and SOAP/WSDL are applicable means, but it is foreseeable that in the future other mechanisms will be used. RDF and OWL are already means to implement the same concepts. A technology independent specification implies that a technology change can easily be implemented.
- Specifying government semantics offers a means to develop a coherent set of application services. This set of application services only relates to the semantics and is not dependent on any particular government service or government organization. The set of external visible application services can be decomposed internally to application services supported by application components, in which case process orchestration is required. A service requestor is also able to orchestrate several external visible application services to get the required result, e.g. first a retrieval of a government

service, secondly a retrieval of the applicable law and finally the retrieval of the documents describing the law.

- Different technologies can be applied in parallel, e.g. service discovery can be based on the mock-up shown before or with a SparQL query according to the protocol. A service requestor can choose the technology that best fits its state of the art with respect to IT.
- Results of projects have proved the feasibility to develop and implement one generic application service to support transparency, whereas in the past an application service per government service has been developed. One instead of many application services will certainly shorten the implementation time of transparency. Implementation of application services is always an issue; better to improve implementation by offering less application services.
- Whereas other projects like OntoGov and SemanticGov specify a public service with one input, one output, and elements required for validating business rules of a law, our approach allows a set of more than two interactions combined to a business transaction supporting the business process delivering a public service. A business transaction must contain the elements that are required to validate a business rule defined by the law of a government service. A business process supporting a government service normally consists of two phases: checking the validity of a business transaction for a particular government service and allocating resources for decision making during the first phase and making and communicating the decision during the second phase. Interaction is possible during both phases. By supporting multiple interactions, our model is quite practical.
- Although it could not be shown in the context of this paper, we have a number of concepts of the semantic model specified in more detail. Like we have already indicated, a semantic model of reference data exists. Furthermore, there is a semantic model for government services, including metadata of documents. Our semantic model integrates existing semantic models.

6. Conclusions and further research

This paper shows that separation of semantics and process is the way forward for specification of interfaces, utilising concepts of an existing architectural framework. Various technologies can be used in parallel to implement these interfaces, depending on the state of the art of a service requestor and/or service supplier. The results presented in this paper are in line with international projects like Access-eGov and OntoGov.

Issues like identification, authentication, authorisation, and representation in a federated (international) environment have not (yet) been addressed. These issues are especially of importance for government organizations with respect to companies. Identification of employees of those companies should be based on a federation mechanism, by which companies are trusted domains with their (auditable) policies. A federated infrastructure for identification needs to be solved to be able to realize virtualisation at for instance EU level.

Besides a number of issues for further research, we recommend on the one hand the separation of semantics and process for external visible application services, and on the other hand a technology independent specification of semantics and process. We recommend further to incorporate these two guidelines in government interoperability frameworks. Issues for further research are:

- The basic issue is to develop government semantics. This semantics needs to be applicable for all government organizations that offer government services, but will probably need to be specialised for government organizations that offer particular government services like social security or municipalities.

- In case it is feasible to develop a generic government semantics to support business transaction services for government services, it might be feasible to develop a generic (set of) application component(s) to support the identified application services and the semantics.
- We have not addressed the issue of process orchestration, but would it be feasible to specify generic business processes to support business transaction services? In principle different services have the same structure, e.g. from validating a business transaction for decision making, preparing a decision to making the final decision. For particular types of government services these processes could be identical. A related issue is: will this lead to standard interaction patterns and how can the choreography for these patterns be specified? What standards are available?
- A government and application services ontology can be used to generate a service engineering workbench. Such an integrated workbench supporting various standards is not yet available. Possibly results of the aforementioned EU projects can be applied.
- The approach specifies a push of existing or composite government services to citizens and companies. However, citizens and companies may have other questions that can be potentially be met by more than one government service, possibly of different service providers. New composite services must be specified dynamically, based on an external question. Overlapping concepts of those dynamically composed business transactions services need to be resolved.
- Besides personal access to ones services, government services need to be personalized. Each citizen and company must be able to compose ones service based on existing services. Information specification for the individual government services need to be combined to offer one personal service. Is there a role for intermediates like specified in WSMO?
- Dynamically defining composite services implies that the underlying process must also be generated dynamically. Since the composite services are not known in advance, possibly a mechanism based on business rules must be implemented to dynamically configure a business process.
- Is virtualization of a government and/or EU, i.e. meaning a citizen or company may choose its own entry point to all government/EU services, in the public or in the private domain? What can existing technology offer to support personalization in the private domain? Can each citizen or company define its personal government services? Can these personal services be published and re-used by others?
- Issues like data aspects of services need to be specified. One can envisage that data aspects and semantics of a permission service differs from data aspects of a passport. Data and its semantics can also be visualized differently, e.g. textually or graphically in a 3D/4D environment (4D includes time).
- Finally, implementation is still an important issue. Quite a number of government organizations need to implement application services to support their business services. First of all, they have to agree on a common approach, which has to be detailed, secondly, a functional specification has to be made to fit all requirements, and, finally, the functional specification needs to be implemented. It is our opinion that a structured approach will shorten implementation time. One organization that supports and facilitates all processes will be a prerequisite. That particular organization has to prove the approach is feasible, i.e. by means of a proof of concept and pilots.

References

[1] Centrum voor Good Governance, Ministerie van Binnenlandse Zaken en Koninkrijksrelaties (http://www.minbzk.nl).

[2] E- overheid: Service improvement by egovernment (www.e-overheid.nl).

[3] Knowledge Centre e-government (www.e-overheid.nl): Nederlandse Overheids Referentie Architectuur, versie 2.0, April 2007.Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.
[4] Overheid.nl: De wegwijzer naar informatie en diensten van alle overheden (www.overheid.nl).
[5] www.bedrijvenloket.nl: Composite government services to companies.
[6] Dr.ir. W.J. Hofman, A conceptual model of a Business Transaction Management System, Uitgeverij Tutein Nolthenius, 1994.
[7] Marc Lankhorst et al.: Enterprise Architecture at work, Springer-Verlag, 2005.
[8] www.w3c.org: Technical standards like BPEL (Business Process Execution Language), WSDL (Web Services Definition Language), UDDI (Universal Description, Discovery Interface), SWSF (Semantic Web Services Framework), WSMO (Web Services Modelling Ontology), OWL (Web Ontology Language), RDF (Resource Description Framework), XML Schema (eXtensible Markup Language), XQuery (XML Query Language), SparQL (SparQL Protocol and RDF Query Language).
[9] John Davies, Rudi Studer, Paul Warren, Semantic Web Technologies - trands and research in ontology-based systems, John Wiley & Sons Ltd., 2006.
[10] www.omg.org: BPMN (Business Process Modelling Notation), BPDM (Business Process Definition Meta model), MDA (Model Driven Architecture), and UML (Unified Modelling Language).
[11] Kolter J, Duerbeck S., Mach M, Bednar P., Hreno J., and Skokan M., Access eGov - State of the art report, version 1.6, may 2006, www.accessegov.org.
[12] Ontology Web Language for Services, the OWL Service Coalition.
[13] Directive 2006/123/EC of the European Parliament and of the Council of 12 December 2006 on services in the internal market.
[14] ICTU/EGEM, Generic Functional Design for case management data (GFO-Zaken), www.egem-iteams.nl/system/files/gfozakeninzicht.pdf, 7-10-2004.
[15] E-Government Interoperability Framework, version 6.1, 18 March 2005, www.govtalk.gov.uk.
[16] City Geography Markup Language, version 0.40, Open Geospatial Consortium, www.citygml.org, 2007.

[i] BPMN version 1.2 of February 2008 states that choreography will be incorporated after the work of the Choreography working group of W3C is finished.