Reference Modeling and Lifecycle Management for e-Government Services

Knut Hinkelmann, Barbara Thönssen, Fabian Probst

University of Applied Sciences Northwestern Switzerland (FHNW)
Institute for Management and Information Systems (IMI)
Riggenbachstrasse 16; 4600 Olten, Switzerland
{knut.hinkelmann | barbara.thoenssen | fabian.probst}@fhnw.ch

Abstract

The realisation of e-Government services, in particular in Switzerland, has made limited progress. This is due – on the one hand – to the federal constitution of the Swiss Confederation. On the other hand, public administrations are concerned about the high initial effort and the corresponding financial investments. With the OntoGov system, reference process models are provided at various abstraction layers. Public administrations can adapt these reference models to their specific needs and make them available for citizens. Decisions for designing and adapting reference processes are explicitly modelled using ontologies, thereby making the decision process traceable. Thus, subsequent changes in the reference models can be transferred to all the depending models.

1 Motivation

Although municipalities provide virtually identical services, implementation of these takes place individually and is continually repeated. The relatively high expenditure, often linked with substantial investment in building up the required IT infrastructure necessary to realise e-Government services, and, at the same time, the benefits that are (still) considered minimal, have led to a situation where the implementation of e-Government at least in Switzerland - is only progressing slowly.

One approach, besides the development of standards (e.g. to promote interoperability), is the propagation of reference models for e-government services which can be used directly by the municipalities and cantons or as a basis for individual adaptations. Thanks to the specification of these reference models, as well as the provision of transaction-orientated web services, the expenditure of the municipalities on the IT implementation of individual web applications specific to the particular municipality should decrease markedly.

In order to adapt reference models and the individual implementations to changing laws and regulations, it is important to explicitly model the reasons for design decisions. In the following we present a method that makes possible the adaptation of processes in that changes to the reference models can be automatically updated in the municipalities

In Chapter 2 the EU OntoGov project is presented, within the framework of which the development and practical application of reference models for e-Government services takes place. Then, Chapter 3 gives an overview of the status of reference modelling and e-Government initiatives in Germany and Switzerland. Chapter 4 describes the procedure that has been selected for the development of reference models. In Chapter 5, it is demonstrated how the dependencies between the reference models and the life cycle of processes are recorded and Chapter 6 explains how this information can be used for the identification and evaluation of changes. In Chapter 7, the technical implementation of the solution approach is put forward. Chapter 8 gives an overview of future work.

2 OntoGov

OntoGov (Ontology-enabled e-Gov Service Configuration) is a research project funded by the EU within the context of the Information Society Technologies (IST) programme (IST-507237, http://www.ontogov.com). This project has as its aim the creation of a framework for the development, adaptation, operation and propagation of e-Government services and its implementation within the context of a pilot application [AAHP04]:

- **Development:** design decisions that influence the development of e-Government services should be displayed transparently and explicitly. In this way, the entire life cycle of an e-Government service should be documented from its conception, right up to its replacement by a new service.
- **Adaptation:** thanks to the explicit documentation of all design decisions, it should be possible to identify the processes and process steps affected in the event of a change to the law.
- **Continuity:** services from different public authorities (and third party companies) should be provided transparently for e-Government users ("One Stop Shop").

- **Operation:** the services developed should be compiled and carried out on the basis of semantic descriptions (ontologies) for the duration of processes.
- **Propagation:** the formal description and management of services should permit the propagation and reuse of services and the knowledge that forms their basis.

The Swiss Federal Chancellery aims to improve the offering of e-Government services at all levels of public administration in terms of both quantity and quality, in addition to reducing set-up and maintenance costs. This should be achieved by the Federal Chancellery

- creating and offering reference models for e-Government services,
- developing the reusable web services necessary for them,
- making these services available for the benefit of the Municipalities,
- guaranteeing the continuity of services covering all administration organisations (e.g. by the implementation of "One Stop Services")
- and the inclusion of non-administrative organisations.

Reference models form the basis for the model-based development, propagation and adaptation of e-Government services by the Federal Chancellery. They should be used to make processes and the associated knowledge explicit and available to the various administrative organisations. The following remarks concentrate on this point and show the method selected in OntoGov for the creation of reference process models, for maintaining dependencies between models and for the identification and evaluation of changes. This article does not go into the implementation of other goals in any depth.

3 Initial situation

By "reference modelling", we understand the design and application of models that can be reused (reference models) [BrBu04; FeLo04]. As administrative organisations – as opposed to companies – do not have to protect a competitive advantage and their processes are based on the same legal foundations, e-Government services are particularly well suited to reference models. However, there are relatively few initiatives for the concrete application of reference models for administrative processes.

3.1 Reference modelling of administrative processes

Within the framework of "eLoGo", a project of the Institute of Local Government Studies at the University of Potsdam, three independent reference models were developed for the process, demand and architecture levels, which describe the dependencies between administrative processes and information technology at various levels [OfHo04]. Thomas Off further developed the results of "eLoGo" and submitted a standardisation process to the German Standards Institute [Deutsches Institut für Normung - DIN]. In relation to the approach discussed in this article, above all the correlation between the ideal type of eLoGo reference process model and the firm establishment of the process view of the reference demand model is interesting. The method worked out to perform this includes instructions and rules for the "tailoring" of abstract reference models into concrete application models.

The aim of the "RAfEG" project (this project promoted by the BMBF in the context of the software initiative 2006 runs initially until the end of 2005, see: http://www.rafeg.de) is the development of an "e-Government reference architecture", with the aid of whose application it is possible to implement software transaction-orientated e-Government services. The object of the pilot project is the illustration of the processes for the planning procedure. The process models developed in the context of this project will be modelled and documented with the aid of the ARIS toolset [SMRV04].

However, neither "eLoGo" nor "RAfEG" are capable of automatically evaluating and actively using the documented knowledge about the process modelling and the derivation of the reference models.

In Switzerland, efforts are underway at state level to drive e-Government forward on the basis of standards and best practices. In the "eVanti" initiative (http://www.evanti.ch) reusing knowledge gained in the implementation of e-Government services is the most important issue. However, this knowledge is neither generalised nor represented as a formal model. In addition, the eCH organisation (http://www.ech.ch) defines non-mandatory guidelines for improving co-operation on the basis of sample processes and data standards. The approach of eCH certainly comes close to the idea of reference modelling. However, while data models are formally represented as XML schemas which can be instantiated [Broc03], sample processes are simply documented and cannoc be used for active derivation.

3.3 Contribution of OntoGov

With OntoGov, reference process models for e-Government services independent of Municipalities should be created which can be adapted to suit the individual municipalities. At the forefront here is the documentation of decisions relevant to modelling and the reasons for them, the traceability of derivations of new versions of models, as well as the partially automated identification and evaluation of changes.

Because of the pronounced federalist structure, the adaptability of reference process models is essential for their acceptance by the federal states and municipalities. At the same time however, these adaptations must be documented in such a way that they can be automatically identified and updated in the event of changes to the original reference process models.

4 Creation of reference process models

The services to be provided by the administrative organisations can be generalised. On the one hand, all of the services provided by the public authorities are based on statutory regulations, whereby a standardisation of the processes is given at a specific abstraction level. Furthermore, the general sequence of processes, e.g. that of a move, is essentially the same everywhere: de-registration in a Municipality, if necessary, informing the relevant Canton (e.g. if dealing with a foreign resident), information about the move to the new Municipality, registration with the new Municipality and again, if necessary, informing the Canton.

Apart from the reference models themselves, knowledge of the procedure for process development is managed with OntoGov. Here a differentiation is made between *functional knowledge* which is necessary for process handling and *process knowledge* which describes the information via the process development [HiKT02; NäSc02]. This process knowledge and the decisions associated with it are frequently documented inadequately or even not at all.

Process knowledge has particular significance with regard to the creation of reference models. Decisions that lead to the specialisation of a reference model should be capable of being understood at all times and the reasons behind decisions (e.g. statutory regulations) should be entirely transparent. In order to achieve this, with OntoGov functional and process knowledge is made explicit in that decisions are assigned to process elements that form their basis. As a result, in the case of changes, the processes affected can be automatically identified and put forward for checking. This in turn offers the opportunity to reduce the time required to implement the change.

4.1 Modelling reference processes

Throughout the rest of the paper we demonstrate our approach with the example of move process, i.e. registions and deregistration procedures when movin from one municipality to another.

Fig. 1 shows the reference process model as an activity diagram in UML2 notation. This was chosen because UML is widely known. In OntoGov itself a separate, project-specific notation is used (cf. Section 7).

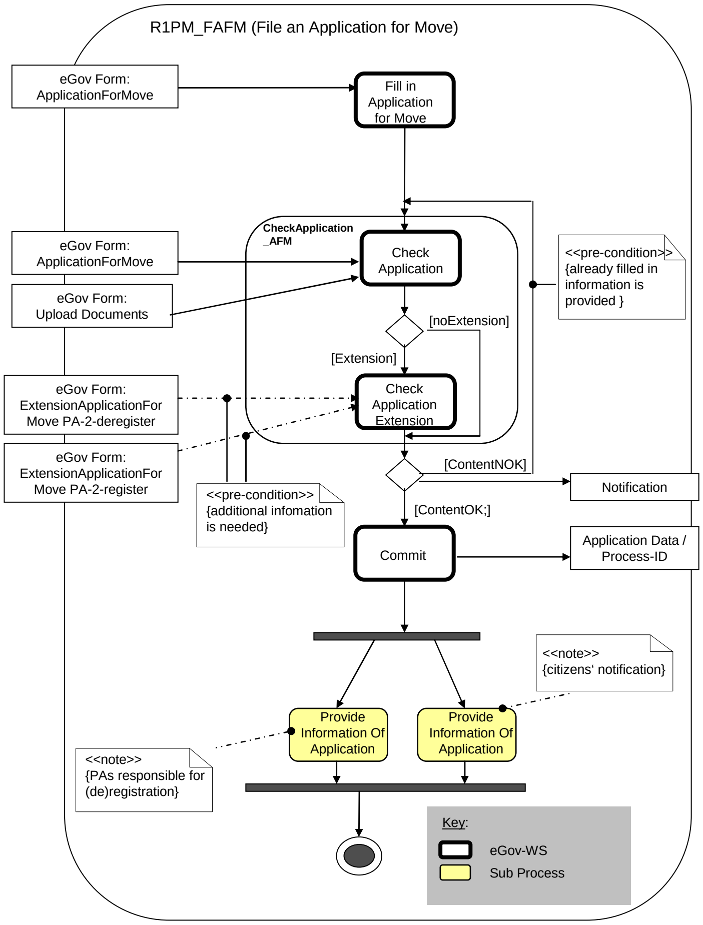

R1PM_FAFM (File an Application for Move)
eGov Form: ApplicationForMove
Fill in Application for Move
CheckApplication _AFM
eGov Form: ApplicationForMove
eGov Form: Upload Documents
Check Application
<<pre-condition>> {already filled in information is provided }
[noExtension]
[Extension]
eGov Form: ExtensionApplicationFor Move PA-2-deregister
eGov Form: ExtensionApplicationFor Move PA-2-register
Check Application Extension
<<pre-condition>> {additional infomation is needed}
[ContentNOK]
Notification
[ContentOK;]
Commit
Application Data / Process-ID
<<note>> {citizens' notification}
Provide Information Of Application
Provide Information Of Application
<<note>> {PAs responsible for (de)registration}
Key:
eGov-WS
Sub Process

Fig. 1 R1PM File an application for move: De-registration in a Municipality

The process starts with the completion of an electronic form (*Fill in application for move*). Next, the information is checked in that a differentiation is made between checking the standard information (*Check Application*) and additional information (*Check Application Extension*). If the information is correct, the application is activated (*Commit*). The activities to pass the information on to the relevant public authorities and applicant (*Provide Information of Application*) are sub-processes that are in turn created as reference processes in a separate model.

4.2 Hierarchy of reference models

Reference models are samples for specific processes in a public administration (e.g. "Application for de-registration for natural persons", "Application for issue of registered parking permit", "Application for approval of dog breeding"). They describe – irrespective of the Municipality– the processing sequence, the activities necessary for this and the statutory regulations affecting it. This type of reference model is designated below as Reference1Processmodels (R1PM) or as basic reference models.

Because of Switzerland's Federal structure, the Cantons can agree specific regulations that lead to adaptations of the processes. Canton-specific reference models largely concur and only vary very slightly. Reference1Processmodels which represent the commonalities of the models in the sense of generalisation, can be further specialised on various levels, e.g. with regard to Cantonal or Municipality-specific requirements. The derived reference process models are called Reference2Processmodels (R2PM).

On the other hand, reference processes can be abstracted to generic processes, such as "Make application", "Check status" or "Issue information". These are known as Reference0Processmodels (R0PM).

The OntoGov framework explicitly administers how the processes of process hierarchy differ, so that changes to the abstract models can be transferred to the particular specialisations.

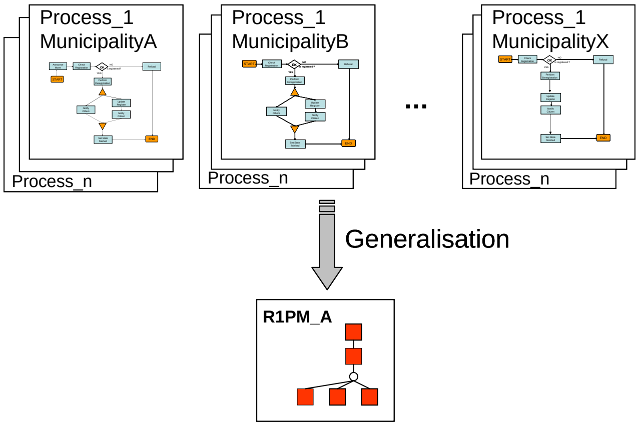


Fig. 2 Generalisation of reference process models

4.3 Procedure

For the creation of basic reference models, specific processes (e.g. "Application for de-registration in a Municipality") from various Municipalities are analysed and then generalised. In this way, common process features can be compiled in a comprehensive basic reference model, in that activities are either modelled as alternative ramifications or generalised by using more abstract process steps [RuGT99, 228]. The aim is to create a process model that can be used irrespective of the Municipality.

Different levels of reference models are derived from these basic reference models (e.g. R1PM_A) by means of content abstraction or specialisation:

- Abstraction: formation of generalised reference process models (e.g. "*Processing of an application*") with the aim of providing generally applicable process models for the simplified creation of basic reference models.
- Specialisation: formation of Municipality-dependent reference process models (e.g. " *Application for de-registration of natural persons in the Municipality of Olten*") with the aim of creating executable process models.

Fig. 3 illustrates the procedure. The interrelationships and dependencies between the individual reference process models are described in Chapter 5.

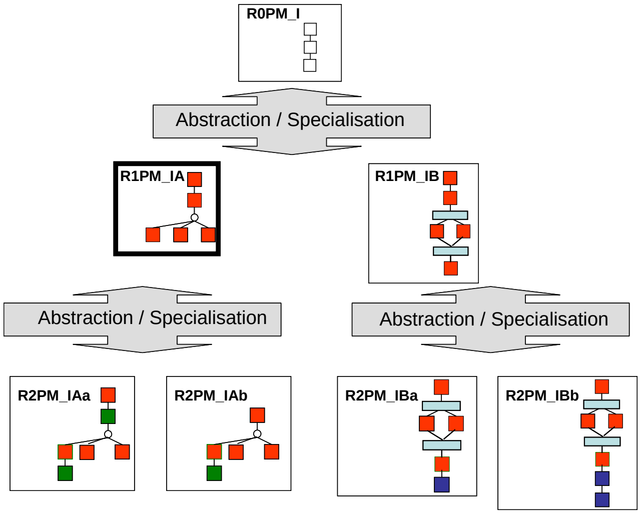

R0PM_I
Abstraction / Specialisation
R1PM_IA
R1PM_IB
Abstraction / Specialisation
Abstraction / Specialisation
R2PM_IAa
R2PM_IAb
R2PM_IBa
R2PM_IBb

Fig. 3 Derivations from reference process models

Reference process models of one level only ever have *one* reference process model at the next abstraction level. Therefore, reference process models *R2PM_IAa* and *R2PM_IAb* can only be abstracted to reference process model *R1PM_IA*, reference process models *R1PM_IA* and *R1PM_IB* only to *R0PM_I*. This limitation was introduced to reduce the complexity – above all in relation to the partially automated updating of specialised reference process models (see Chapter 5).

4.4 Abstraction

Fig. 4 shows the generic (sub-)process for the submission of an application (*Reference0Processmodel*: *R0PM_FAA*). Data is entered (*Fill in application*), checked for completeness and accuracy (*Check Application*) and saved when the application is activated (*Commit*).

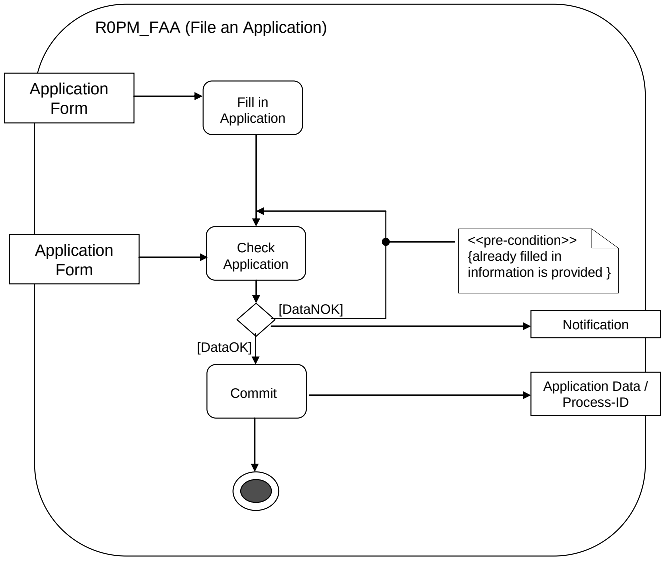

R0PM_FAA (File an Application)
Application Form
Fill in Application
Application Form
Check Application
<<pre-condition>> {already filled in information is provided }
[DataNOK]
Notification
[DataOK]
Commit
Application Data / Process-ID

Fig. 4 R0PM File an application

Apart from the actual process model, *Reference0Processmodels* can also contain further generally applicable outline conditions, such as statutory regulations and organisational information, e.g. about necessary resources. The Civil Code therefore forms the foundation for handling applications, no matter what type they are.

4.5 Specialisation

Each reference process model can be further specialised. Thus several reference process models (e.g. *R1PM Application for the issue of a registered parking permit*" or "*R1PM de-registration in a Municipality*") can be derived from one generic reference process model (e.g. from "*R0PM Make application*"). The latter can be further specialised to become "*R2PM de-registration in the Municipality of Olten*".

The specialisation of reference process models is possible as follows:

- Reduction: activities or sub-processes are removed
- Extension: additional activities or sub-processes are inserted
- Replacement: one activity is replaced by another, but the process sequence is not altered.

Fig. 5 shows the specialisation of the example already introduced "De-registration in a Municipality" (R1PM_FAFM File an Application for Move) to "De-registration in the Municipality of Olten" (R1PM_FAFM File an application for move in Olten).

The first change for R2PM concerns the service FillInApplicationForMove. It is replaced (1) by the Municipality-specific activity FillInApplicationForMoveInOlten, a Municipality-specific implementation with a link to the administration software. The second change concerns the activity CheckApplicationExtension, it is deleted (2). The third change inserts the additional activity UploadDocs (3).

Each specialisation of the reference model is explicitly recorded for reasons of traceability (see Chapter 5). However, the greater the degree of change, the more difficult it is to evaluate these changes automatically (see Chapter 6).

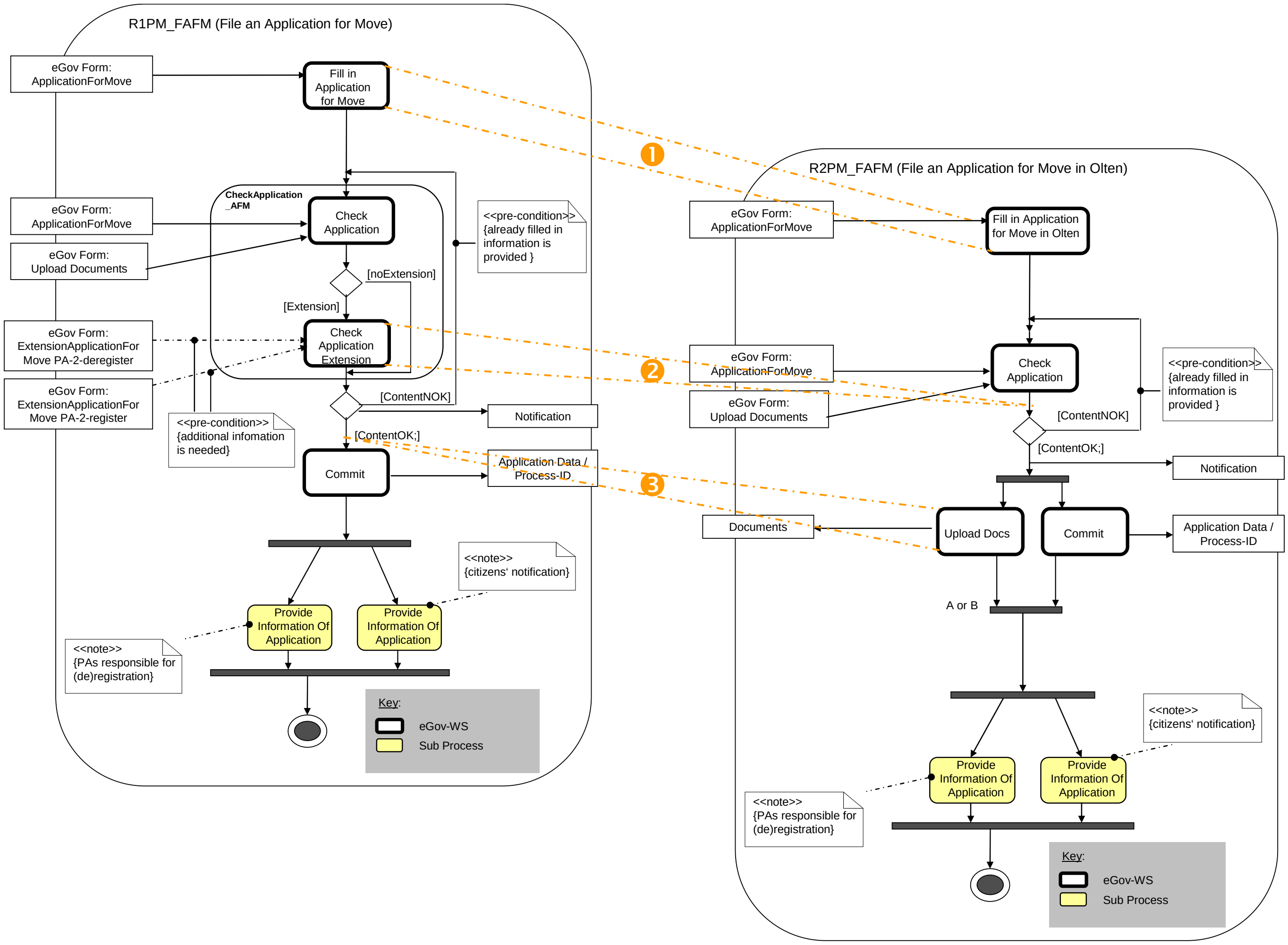

R1PM_FAFM (File an Application for Move)
eGov Form: ApplicationForMove
Fill in Application for Move
eGov Form: ApplicationForMove
eGov Form: Upload Documents
CheckApplication_AFM
Check Application
<<pre-condition>> {already filled in information is provided }
[noExtension]
[Extension]
eGov Form: ExtensionApplicationFor Move PA-2-deregister
eGov Form: ExtensionApplicationFor Move PA-2-register
Check Application Extension
[ContentNOK]
<<pre-condition>> {additional infomation is needed}
Notification
[ContentOK;]
Commit
Application Data / Process-ID
<<note>> {citizens' notification}
Provide Information Of Application
Provide Information Of Application
<<note>> {PAs responsible for (de)registration}
Key:
eGov-WS
Sub Process
1
2
3
R2PM_FAFM (File an Application for Move in Olten)
eGov Form: ApplicationForMove
Fill in Application for Move in Olten
eGov Form: ApplicationForMove
eGov Form: Upload Documents
Check Application
<<pre-condition>> {already filled in information is provided }
[ContentNOK]
[ContentOK;]
Notification
Documents
Upload Docs
Commit
Application Data / Process-ID
A or B
<<note>> {citizens' notification}
Provide Information Of Application
Provide Information Of Application
<<note>> {PAs responsible for (de)registration}
Key:
eGov-WS
Sub Process

Fig. 5 Specialisation of process elements from R0PM to R1PM (UML notation)

5 Lifecycle Management

Reference modeling must cope with the problem of change. We distinguish two types of changes:

- General changes, e.g. because of law changes or overall improvements as of new IT innovations can lead to new versions of reference process models
- Local changes on the level of the municipality, e.g. organizational changes or implemention of new information systems, local optimizations etc. lead to new versions of the concrete processes that are derived from the reference model.

To ensure consistency of services it is important that all these changes are documented. For example, if a reference model has been modified because a law has changed, all its specializations and concrete implementations must retrace these modifications.

The dependencies between reference process models and modifications during the lifecycle of reference processes and concrete processes are recorded in the form of design decisions and reasons for these decisions. Three types of design decision can be distinguished for the modelling of changes to process models:

- **Delete design decision** to delete an activity (reduction)
- **Add design decision** to add a new activity (extension)
- **Replace design decision** to replace an activity (replacement)

Fig. 6 is a schematic representation of the dependencies between reference process models and the decisions forming the basis of changes.

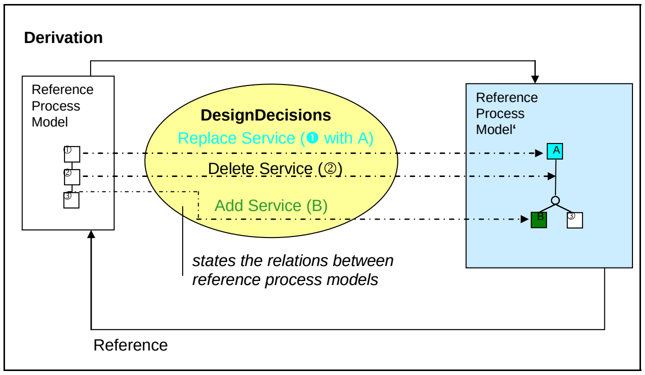


Fig. 6 Schematic representation of the dependencies between reference process models

Current modelling methods do not consider processes in an isolated way, but permit various views of a process, which are each represented in a separate model. A well-known method is ARIS (Architecture integrated Information Systems) [Sche95] which integrates the data, organisation and functions of a company in one process model. BPMS methodology [Kara95] distinguishes processes, organisational structure, resources and data/documents.

This approach is also followed here. As all administrative processes require a legal basis, a separate model is provided to represent laws and regulations, so that the following model types can be used for integrated process modelling:

- Process
- Organisation
- Data
- Technology (here this term relates to the IT system landscape of an organisation which influences the implementation of processes)
- Laws/regulations

All of the decisions that lead to the specialisation or versioning of reference process models must be well-founded. Therefore, at least one reason must be applied to each design decision (in practical applications, decisions can be grouped together; the entire group then only has to be founded on a single reason). According to the different viewpoints, the reasons for design decisions are classified in four categories which are decisive for the specialisation of reference process models:

- **Legal-based reason:** Virtually all the processes of public administrative organisations are based on legal foundations or decrees that are derived from them. Therefore, such regulations are also determining for the design of a process. For example, the separation of powers requires that selected processes are dealt with by different regulatory and political authorities.
- **Organisational-based reason:** The way in which an administrative organisation is structured, into what organisational units it is divided and what duties are carried out by which sections influences the structuring of a process to a great extent. For instance, if a task is carried out by two different organisational units, then this duty will be modelled by two activities.
- **Data-based reason:** The data to be processed and the flow of information can also affect the process model. If, for instance, data is required in an activity that is determined during the process, then a specific order of activities can result from this.
- **Technical-based reason:** It is not only organisational and legal specifications that define the design of a process, but also technology. Therefore, data validation – e.g. for security reasons – can take place in two separate process steps (e.g. on a web-server, the checking for completeness of data that it is mandatory to enter and checking personal data in internal administration software).

A reason includes a description in natural language and a formal reference to the elements forming the basis of the decision from the appropriate models. For example, in the case of a legal-based reason, a reference to the relevant article of law is recorded.

The linking of design decisions and reasons takes place in OntoGov by using a lifecycle model, in that the reasons for the modelling decision are explicitly modelled. The model is based on the IBIS approach (Issue Based Information System, [KuRi70]). IBIS starts from a topic, into which different positions can be accepted. These positions are based on several arguments. Analogous to IBIS, in OntoGov a design decision is regarded as a topic which is based on other design decisions or the above-mentioned aspects, irrespective of whether it was encountered in a process model or in another model. As decisions can also be based on other design decisions, a line of argument results. Fig. 7 shows the structure of a line of argument according to the graphic notation gIBIS [CoBe98, 140–152].

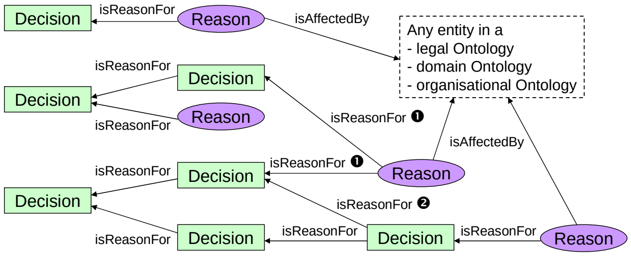


Fig. 7 Linking decisions and reasons

Apart from the natural language documentation, it is vital for knowledge-based support to also formally describe the design decisions [RuGT99, 231]. In this approach the process knowledge is recorded, in that the process model is formally linked with the nominated models (process, organisation, data, technology, laws/regulations). In this way, design decisions and reasons that define reference models become transparent and traceable.

Given that the legal basis "*BGB, Article 0815, para. 1*" regulates the various responsibilities of two public administrative organisations, this legal text can then be regarded as the reason for the design decision to model two activities separately from one another. In this way the design decision is directly linked with the activities affected and the reason for this decision provides a reference to the appropriate article in the laws/regulations. The concrete implementation is explained in detail in Section 7.

6 Identification and evaluation of changes

In the event of a change, e.g. to a law, firstly all the reference process models and their activities affected by the change are identified. This takes place by means of the lifecycle model being queried for all reference process models and activities whose design decisions are based on the legal provision "*BGB, article 0815, para. 1*". The query is effected in OntoGov using search masks with relevant pull-down lists (e.g. for all the reasons). The result is a list of the reference process models and activities affected. When selecting a reference process model, the activities affected and the design decision(s) forming the basis are displayed.

If the change is a mere formality, e.g. it is no longer para. 1 but para. 2 of article 0815 that provides the legal foundation, then updating can be automated. In this case, the reason is modified accordingly and then automatically propagated to all reference process models and activities concerned. If the change is only to apply to one reference process model or one activity, then the old reason must firstly be removed and then the new reason created and assigned.

If a reason is completely removed, e.g. because article 0815 has been deleted, then it is possible to determine those reference process models which include activities whose design decisions are based on this reason alone. It is proposed that these activities be deleted.

On the other hand, if there are still other reasons for a design decision, only a potential change is indicated.

If processes have to be adapted, then the following procedure can be adopted:

- The reference process models affected are determined as mentioned above,
- The process that is highest in the hierarchy is selected (e.g. *R0PM File an application*),
- A new version of the reference process model is created,
- Process changes, the design decisions behind them and the new reason are documented.

Once all of the adaptations have been carried out, the modifications can then be propagated to all derived, specialised reference process models – e.g. by setting an appropriate status. This is made possible by evaluation of the information in the lifecycle model and process knowledge from the process models. Fig. 8 shows the procedure.

Automatically created new versions of derived reference process models (here: *R1PM_A4MoveV4*) are given the status "Draft". They must then be checked manually and if necessary further adapted.

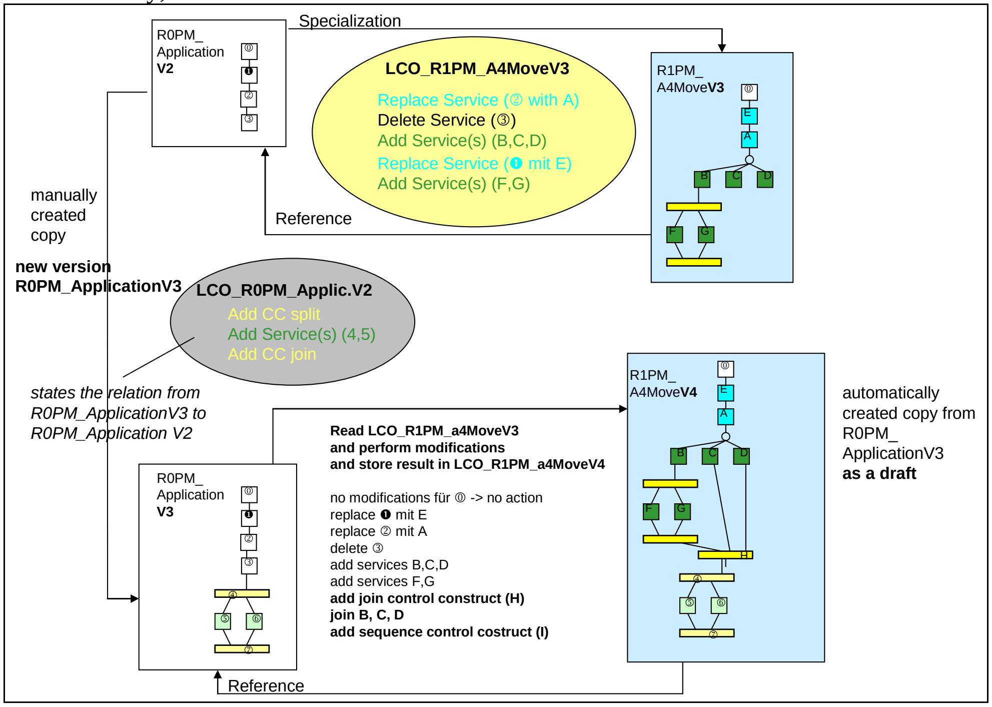


Fig. 8 Change propagation in reference process models

7 Ontological Aspects of the Implementation

Tools for modelling reference models vary greatly in terms of their efficiency. The bottom limit is defined by pure graphic editing of models, whereas more powerful modelling tools have a central repository for model management [FeLo04, 336f.]. OntoGov goes even further here, in that in addition to reference models, their dependencies and reasons for design decisions should also be managed.

Reference process models, as well as all the knowledge necessary for their execution or documentation is represented in the form of ontologies. An ontology is a technical model of a part of the world, via whose terms and interrelationships a group of experts/users achieved unity (according to [Grub93]).

Ontologies are used in the OntoGov project in order to express complex circumstances explicitly and technically. In the modelling phase these characteristics are used to define the dependencies between various models and to record complex transformations from a reference model to a derived model. In addition, the syntactic correctness of a model can be safeguarded with the aid of rules and in the event of changes to the reasons, the processes affected can be automatically identified.

Following [ABHK98] we distinguish different kind of knowledge represented in enterprise, domain and information ontologies, respectively. In addition, we have a lifecycle ontology for representing design decisions and reasons.

7.1 E-government Service Ontology

The process reference models are represented in the service ontology, corresponding to the enterprise ontology of [ABHK98]. The language for process reference models is based on a version of OWL-S (http://www.daml.org/services/owl-s/1.1) extended by the concepts of the BPML standard (http://www.bpmi.org/BPML.htm), a process ontology that is described in the Web Ontology Language (OWL) standardised by the World Wide Web Consortium (http://www.w3.org/2004/OWL). In addition, the models have been extended with the constructs mentioned in Section 5 for the management of design decisions. In the following sections, the modelling and execution environment is illustrated by means of our example.

7.2 User Interface

A key element of the modelling environment is a graphics editor which is used for the specification of the sequence of events. Further important information is added to the process using this editor, including design decisions, organisational resources used, applications, etc.

Graphically created process models are automatically translated into the process ontology.

The use of a graphics editor makes modelling easier, so that (reference) process models can be independently created and maintained by specialist personnel from the administrative organisations.

Fig. 9 shows the user interface with the example of a modelled process *R1PM_FAFM* (*File an application for move*) and the associated ontology that is created in the background.

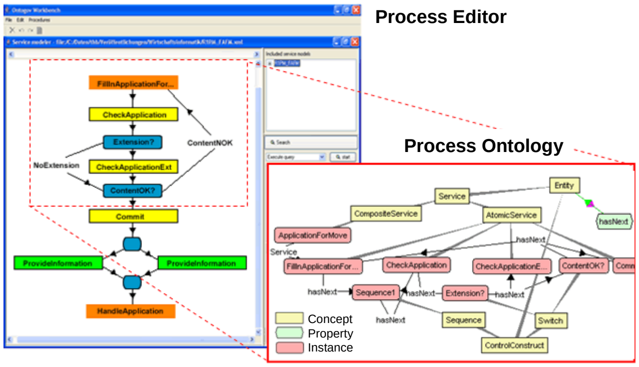
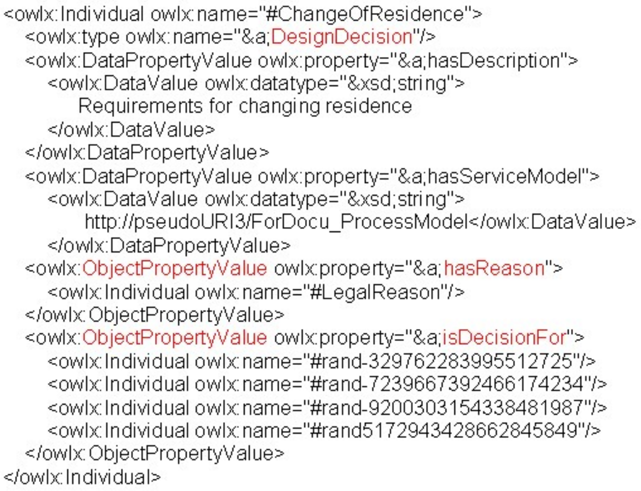


Fig. 9 R1PM File an application for move in the process editor and as an ontology

7.3 Lifecycle Ontology

Design decisions and reasons are represented in the lifecycle ontology that mainly consists of the elements shown in fig. 7. The lifecycle ontology is realised as a shared information source between all provided services. The advantage of this approach is the ability to share reasons between different service ontologies but also to allow referring design decisions from other models as reasons.

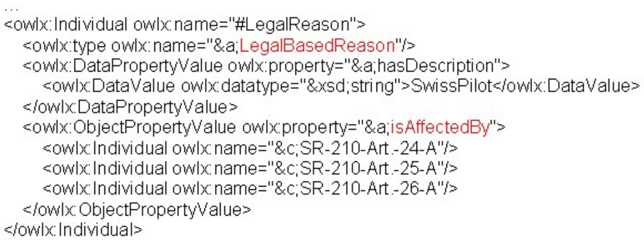
All references to the service model as well as to the domain oriented ontologies (domain, legal and organisational ontology) are stored in the lifecycle ontology as attributes. The following extractions of an exemplary lifecycle ontology show how the references between a *DesignDecision* and the affected services (or control-constructs) are realised by the *ObjectPropertyValue* attribute “isDecisionFor”. In the same way the reasons a design decision is based on, are referenced by the *ObjectPropertyValue* “isReasonFor”



```
<owlx:Individual owlx:name="#ChangeOfResidence">
   <owlx:type owlx:name="&a;DesignDecision"/>
   <owlx:DataPropertyValue owlx:property="&a;hasDescription">
      <owlx:DataValue owlx:datatype="&xsd;string">
          Requirements for changing residence
      </owlx:DataValue>
   </owlx:DataPropertyValue>
   <owlx:DataPropertyValue owlx:property="&a;hasServiceModel">
      <owlx:DataValue owlx:datatype="&xsd;string">
           http://pseudoURI3/ForDocu_ProcessModel</owlx:DataValue>
      </owlx:DataPropertyValue>
   <owlx:ObjectPropertyValue owlx:property="&a;hasReason">
      <owlx:Individual owlx:name="#LegalReason"/>
   </owlx:ObjectPropertyValue>
   <owlx:ObjectPropertyValue owlx:property="&a;isDecisionFor">
      <owlx:Individual owlx:name="#rand-329762283995512725"/>
      <owlx:Individual owlx:name="#rand-7239667392466174234"/>
      <owlx:Individual owlx:name="#rand-9200303154338481987"/>
      <owlx:Individual owlx:name="#rand5172943428662845849"/>
   </owlx:ObjectPropertyValue>
</owlx:Individual>
```

Fig. 10 Representation of a DesignDecision and the references to related services and Reasons.

Fig. 11 shows an example of a *LegalBasedReason*. The individuals, a Reason is based on, are referenced by the *DataPropertyValue* attribute “*isAffectedBy*”.



```
…
<owlx:Individual owlx:name="#LegalReason">
   <owlx:type owlx:name="&a;LegalBasedReason"/>
   <owlx:DataPropertyValue owlx:property="&a;hasDescription">
      <owlx:DataValue owlx:datatype="&xsd;string">SwissPilot</owlx:DataValue>
   </owlx:DataPropertyValue>
   <owlx:ObjectPropertyValue owlx:property="&a;isAffectedBy">
      <owlx:Individual owlx:name="&c;SR-210-Art.-24-A"/>
      <owlx:Individual owlx:name="&c;SR-210-Art.-25-A"/>
      <owlx:Individual owlx:name="&c;SR-210-Art.-26-A"/>
   </owlx:ObjectPropertyValue>
</owlx:Individual>
```

Fig. 11 LegalBasedReason and its references to the legal ontology

If a new model is to be derived from an existing reference process model, firstly a copy of this reference model is created. This copy is then used as a basis for further processing. Subsequent changes to the copy are recorded in the lifecycle ontology in the form of a design decision with a reference to the reference model and an associated reason. Fig. 12 explains the procedure.

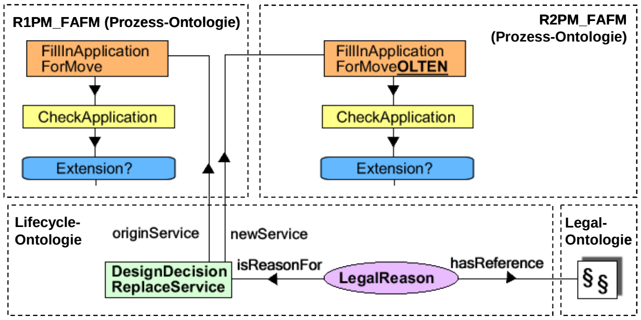


Fig. 12 Process specialisation – linking lifecycle ontology with other ontologies

A copy (*R2PM_FAFM*) is created from the reference model (*R1PM_FAFM*) and then changed. The change is recorded in the lifecycle ontology in the form of an instance of design decision. This includes formal links to previous activity (*originService*) and to the new activity (*newService*) and refers to a reason (*LegalReason*), which again references an element in the legal ontology.

Design decisions can be linked with a single activity or several activities, as well as with the entire process model. Here, each element can have several design decisions as its basis.

7.4 Lifecycle Synchronisation

A main objective of the discussed approach is to keep all models up-to-date and consistent. For lifecycle aspects it will be the case that instances reasons are based on are changing (e.g. a law is modified). In this regard the change procedures are applied to detect changes, affecting reasons of the lifecycle ontology.

Lifecycle synchronisation is performed for a particular service and is actuated by the domain expert who is responsible for the service. Lifecycle synchronisation analyses the evolution log and selects changes that are related to reasons and design decisions that are link to the service. After the synchronisation, affected reasons (as well as the depending design decisions) are tagged by changing the state from “valid” to “toBeChecked”. The elements in the lifecycle ontology remain in this status until the user has checked the indicated changes and explicitly took actions for changing the states into valid again. That might be done by removing the related ‘deleted’ instances from the reason or deleting the reason itself.

Fig. 13 gives an example of the possible changes.

① Two instances of the legal ontology have been modified (e.g. L1 is changed, L2 is deleted).

② When ‘Synchronize Lifeccycle’ for Service1 is started, changes are propagated to the lifecycle ontology where the affected Reasons (R1 and R2) and depending Design Decisions (DD1 and DD2) are indicated.

③ DD1, respectively R1 get status attribute ‘toBeChecked’ and the additional (status) attribute ‘isAffectedBy_Modified’ is set to L1A, indicating which of the Reasons for DD1 has been modified.

④ DD2, respectively R2 get status attribute ‘toBeChecked’ and the additional (status) attribute ‘isAffectedBy_Deleted’ gets value L2A, indicating which of DD2’s Reasons are deleted.

⑤ When starting synchronisation for Service 2 again changes are propagated to the lifecycle ontology. The affected Design Decision (DD4) gets status ‘toBeChecked’; the status of the underlying Reason (R2) is already changed. DD3 will remain untouched.

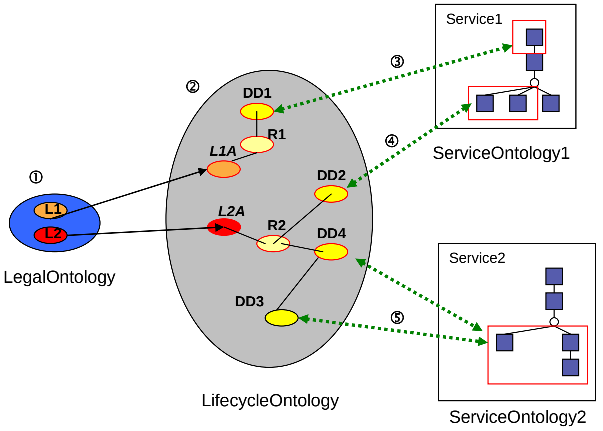


Fig. 13 Change Propagation for LifecycleOntology (Part 1)

To put back the status of Design Decisions and Reasons to ‘valid’ they have to be edited. Fig. 14 depicts the respective procedure.

① Dealing with Service1, DD1 and R1 are checked by the user and the modification of L1A is accepted.

② DD1 and R1 get status attribute ‘valid’ and the additional (status) attribute (for L1A) is removed.

③ R2 is deleted by the user from DD2 so R2 gets status attribute ‘deleted’. DD2 gets a new Reason (R3) with ‘isAffectedBy’ attribute L3A.

④ To make the manually made changes visible, ‘Synchronize Lifecycle’ has to be performed again for Service2 as synchronization comprises not only the matches between domain oriented ontologies and the Lifecycle ontology but also within the Lifecycle ontology itself .

⑤ When DD4 is deleted manually by the user, DD3 gets status attribute ‘toBeChecked. To make the Design Decision valid again, either a new reason has to be added or DD3 has to be removed as well.

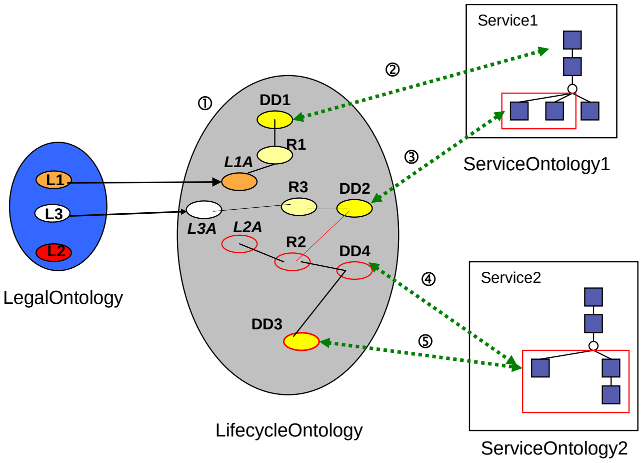


Fig. 14 Change Propagation for LifeCycleOntology (Part 2)

7.5 Service Configuration

Once process modelling is completed, web services are allocated to individual activities. This takes place in the configuration environment. In this way, web services relevant for the execution of an activity can be specified directly. Alternatively, selection criteria can be specified, via which appropriate services can be identified for the duration.

The easiest way for Municipalities to participate in the OntoGov e-Government services is to specify an e-mail address. For the duration, the relevant web service will be determined for the particular Municipality and the data or the link to a secure data server (data hub) sent to the Municipality by e-mail.

If the Municipality is interested in closer integration, then a web service can be replaced at any time and the data imported, for example, directly into the legacy system of the Municipality – whilst taking into consideration the necessary security regulations.

8 Conclusion and outlook

A large proportion of a Municipality's duties are regulated by the law. Therefore, the sequence of processes is essentially the same. This means they are exceptionally well suited to reference modelling, in that the common core of a process can be modelled as a reference process and reused many times. The reference models developed in this way have a normative character in the sense of the characterisation of [FeLo04, 333], whereby standardisation should rather be understood as a recommendation, as the public authorities are independent in the implementation of their processes – as long as the legal regulations are complied with.

A procedure has been presented in this article, with which reference models can be modelled in the form of ontologies. The reference models are organised in a specialisation hierarchy which makes it easier to find the appropriate reference models. Also, reference models are supplemented by the explicit modelling of process knowledge using legal foundations and reasons for design decisions. Thus, for instance, in the event of amendments to the law, the processes concerned can be simply identified and adapted.

Within the context of the OntoGov project, the procedure is demonstrated using the example of a pilot process for the relocation of private persons. Under the auspices of the Swiss Federal Chancellery, the reference process is modelled and adapted for Municipalities in different Cantons.

A future task will be the integration of traditional processing and the internet-based activation of processes. This can be achieved using a common reference model, from which the two process versions can be derived – in the same way as the specialisation described above. This would have the advantage that the consistency of the two process versions can be safeguarded and redundancies in modelling would be avoided.

Remark

This article was written within the context of the "OntoGov" EU project. The Swiss partners in the project – and therefore in particular the creation of this article – are sponsored by the Federal Office for Education and Science (BBW).

References

[AAHP04]Abecker, A.; Apostolou, D.; Hinkelmann, K.; Probst, F.; Stojanovic, L.; Tambouris, T.: Ontology-enabled eGovernment Service Configuration – the OntoGov Approach. In: Wimmer, M. A. (Hrsg.): e|Gov Days: State-of-the-Art 2004. books@ocg.at, Wien 2004, Band 177, S. 324-331.

[ABHK98] Abecker, A.; Bernardi, A.; Hinkelmann, K.; Kühn, O.; Sintek, M. (1998): Towards a Well-Founded Technology for Organizational Memories. IEEE Intelligent Systems and their Applications, Vol. 13, Nummer 3, S. 40 - 48, May/June 1998. Reprint in: James W. Cortada, John A. Woods (Eds.), The Knowledge Management Yearbook 1999-2000, Butterworth-Heinemann, pp. 185-199

[BFSt00] Bundesamt für Statistik, Gruppe für Wissenschaft und Forschung; Bundesamt für Kommunikation (Hrsg.): Informationsgesellschaft Schweiz. Standortbestimmung und Perspektiven. Neuchâtel 2000, S. 43-45.

[BrBu04] vom Brocke, J.; Buddendick, C.: Organisationsformen der Referenzmodellierung. In: Wirtschaftsinformatik 46 (2004) 5, S. 341-352.

[Broc03] vom Brocke, J.: Referenzmodellierung. Gestaltung und Verteilung von Konstruktionsprozessen, Logos Verlag, Berlin 2003, S. 299-308.

[CoBe98] Conklin, J.; Begeman, M.: gIBIS – A Hypertext Tool for Exploratory Policy Discussion. In: Proceedings of CSCW98 , Seattle, Washington 1998, S. 140-152.

[FeLo04] Fettke, P.; Loos, P.: Referenzmodellierungsforschung. In: Wirtschaftsinformatik 46 (2004) 5, S. 331-340.

[Grub93] Gruber, T.: A Translation Approach to Portable Ontology Specifications. In: Knowledge Acquisition, Volume 5. Academic Press Ltd., London 1993, S. 199-220.

[HiKT02] Hinkelmann, K.; Karagiannis, D.; Telesko, R.: PROMOTE – Methodologie und Werkzeug für geschäftsprozessorientiertes Wissensmanagement. In: Abecker, A.; Hinkelmann, K.; Maus, H.; Müller H.J. (Hrsg.): Geschäftsprozessorientiertes Wissensmanagement. Springer-Verlag, Berlin, Heidelberg 2002, S. 65-90.

[Kara95] Karagiannis, D.: BPMS: Business Process Management Systems. ACM SIBOIS Bulletin, Vol. 16, ACM Press 1995, S. 10-13.

[KuRi70] Kunz, W.; Rittel, H.W.J.: Issues as Elements of Information Systems. WP-131, University of California, URL: http://www-iurd.ced.berkeley.edu/pub/WP-131.pdf. visited on 5 December 2005.

[NäSc02] Nägele, R.; Schreiner, P.: Potenziale und Grenzen von Business Process Management Tools für geschäftsprozessorientiertes Wissensmanagement. In: Abecker, A.; Hinkelmann, K.; Maus, H.; Müller H.J. (Hrsg.): Geschäftsprozessorientiertes Wissensmanagement. Springer-Verlag, Berlin, Heidelberg 2002, pp. 25-46.

[OfHo04] Off, T.; Horn, E.: Referenzmodelle für eGovernment. In: Reichard, C.; Scheske, M.; Schuppan T.: Potenziale des eGovernment; 2004. http://www.thomasoff.de/egov-referenzmodelle/index_0.htm, visited on 23 January 2006.

[RuGT99] Rupprecht, Ch.; Peter, G.; Rose, Th.: Ein modellgestützter Ansatz zur kontextspezifischen Individualisierung von Prozessmodellen. In: Wirtschaftsinformatik 41 (1999) 3, S. 226-237.

[Sche95] Scheer, A.-W.: Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse. Springer-Verlag, Berlin, Heidelberg 1995.

[SMRV04] Seel, S; Martin, G.; Rünger, G.; Voigt, M.; Petersohn, H.: Referenzarchitektur für eGovernment. URL: http://vdbsrv.de/viomatrix/924/imgs//downloads/rafeg_210604_final.pdf, visited on 23 January 2006.