Business Processes and Rules
An eGovernment Case-Study

Dimitris Karagiannis

University of Vienna
Department of Knowledge Engineering
Brünnerstraße 72
1210 Vienna, Austria
dk@dke.univie.ac.at

Wilfrid Utz, Robert Woitsch, Andrea Leutgeb

BOC Asset Management GmbH
Bäckerstraße 5
1010 Vienna, Austria
[wilfrid.utz, robert.woitsch, andrea.leutgeb]@boc-eu.com

Abstract

This paper introduces the approach of the EU-funded project FIT (FP6-IST-27090) on adaptive processes in the public administrations, identifies requirements on the design environment to model adaptive processes and discusses the integration of different modeling languages.

The integration of modeling languages is based on the necessity to design business processes and business rules in the context of eGovernment within a comprehensive eGovernment modeling system.

The well-known meta-model approach has been selected to enable the integration of business rule models and business process models to support the adaptive execution of workflows.

Introduction

This position paper presents how business process models and business rule models have been integrated in the IST project FIT in order to support the implementation of an adaptive eGovernment system.

The concepts discussed in the following represent the current development status of the project FIT that has the overall goal to

> develop, test and validate a self-adaptive eGovernment framework based on semantic technologies that will ensure that the quality of public services is proactively and continually fitted to the changing preferences and increasing expectations of e-citizens. [Stojanovic et al 2006]

The technical approach is split into four main areas: framework for self-adaptive eGovernment (service discovery, context awareness, quality of service), personalized front-office (user behavior analysis and adaptation), customized back-office (execution of semantically enriched processes) and a framework to support knowledge sharing (best practices, lessons learned).

The work presented within this paper deals with adaptive process design and execution within the area of back-office processes - meaning customized and fitting service creation and delivery to the e-citizen.

The starting point for modeling the back-office business processes was the ADO*eGov*®[1] method [Palkovits and Wimmer 2003], developed by BOC[2]. ADO*eGov*® is a comprehensive modeling method for public administration based on the BPMS (Business Process Management Systems) – paradigm [Karagiannis 1995].

Within FIT the business process management approach is integrated with business rules. This combination of business processes and business rules makes agile processes possible and is useful for unpredictable, variable, innovative or long-running processes, where task sequences are not predetermined and where knowledge is required at execution time for decision making and problem solving.

The content of this paper presents the solution of conceptual integration as well as the technical integration.

---

[1] ADO*eGov*® has been developed within the national funded project ADO*amt*® (http://www.adoamt.com [10.01.2008]) and is a registered trademark of BOC GmbH

[2] BOC Website. Accessibile: http://www.boc-group.com [24.01.2008]

Business Processes and Business Rules Integration

The approach, which has been developed by the FIT project partners BOC and FHNW[1], addresses the challenge that processes in eGovernment are often quite complex and therefore quite difficult to change. Here the business rules approach can provide a powerful improvement by making business rules that are implicit in the process explicit. One benefit of this separation is that there often are separate reasons for updates to processes or business rules. New regulations or business strategies may affect the business rules without the necessity of changing the business processes.

The ability of business rules to support dynamic changes, allows modifying a business process implementation without changing and redeploying it. On the other hand new applications or procedures might change the business process. Such arguments led to a composite approach to business rules and processes, as also mentioned in [Lienhard and Künzi 2005] or [Schacher and Grässle 2006].

Light weighted and streamlined processes are created applying the business rules approach. These kinds of processes only include the necessary series of steps to accomplish the required work/service from an end-user's perspective.

The FIT project uses the business rules approach for the following application areas: *Variable Process Execution* to determine activities and processes to be executed during process runtime, *Intelligent Resource Allocation* at runtime to select employees based on special skills, to present information depending on user categories or to select a particular Web-Service and *Intelligent Branching and Decision Making* at runtime to control the process flow accordingly.

Modeling Language Integration - Methodology

The business rules approach has grown in importance and popularity in the last few years for agile modeling approaches. Therefore it was implemented within FIT. The developed approach is ontology-based and results in the definition of transparent, flexible and efficient processes in eGovernment.

Business Rules Management is integrated into the ADO*eGov*® method providing the ability to model business rules on different abstraction layers. Three abstraction levels serve three different user groups, the first level is the business/design view, the second level is the interchange level and the third level is the technical/execution level. Each of the three business rules abstraction layers is integrated with the according abstraction layer of the business processes.

[1] FHNW Website. Accessible: http://www.fhnw.ch [10.01.2008]

An organizational framework and methodology supports the designer to formalize the "verbal" rule definition on the highly abstract level to an executable level (see [Hinkelmann et al 2007] for details).

Figure 1 depicts the ADO*eGov*® modeling language and the additional modules, as they were developed within the project.

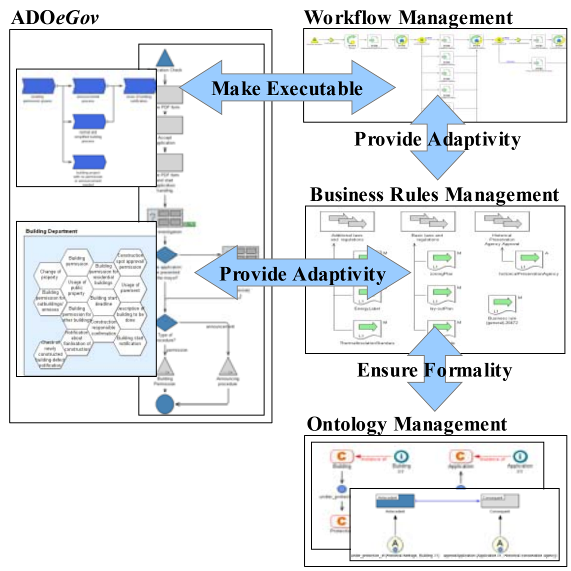

Figure 1: Modeling Language Integration

The figure presents sample models of FIT to show how the connection between different modeling languages is qualified from a conceptual perspective.

The input for the semi-formal representation of business processes and rules are document sources (e.g. laws, regulations etc.), existing process models or interviews/workshops with domain experts, as business rules and processes are often stored in the mind of the domain experts. Other sources could be database analyses or actual workflows in the form of program code. When business rules are defined they are grouped to rulesets and assigned to the corresponding activities, decisions or processes. The added value of the semi-formal representation is that it can be modeled and understood by business people as well as by IT people, as it is clearly structured.

A transformation of the semi-formally defined business processes and rules into an executable and interchangeable representation is necessary. For the execution of workflows in the back-office BPEL (Business Process Execution Language) [OASIS BPEL 2008] has been chosen, for business rules a transformation into an open accessible format using OWL (Web Ontology Language) [W3C OWL 2008] and SWRL (Semantic Web Rule Language) [W3C SWRL 2008] to ensure formality and integration is done.

The implementation approach of the introduced concepts will be presented in the next section.

Flexible Modeling Framework for eGovernment

Based on the meta-modeling framework survey of the University of Vienna, considering the Model Driven Architecture (MDA) [OMG MDA 2008] approach and reflecting the experience resulting from the projects BREIN [BREIN 2008] and FIT the following flexible and holistic modeling framework has been proposed.

Figure 2 introduces the three well-known layers:

- Computational Independent Model (CIM): Models to capturing the real world business, serving as the requirements
- Platform Independent Model (PIM): Workflow models on a platform independent layer that may be derived from the upper layer
- Platform Specific Model (PSM): Executable models bound to a specific platform, which are a refinement of the PIM layer.

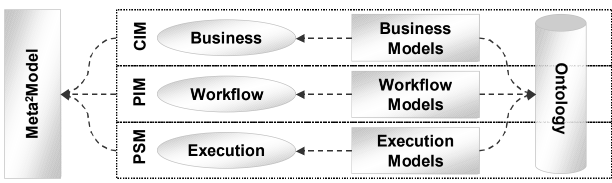

Figure 2: Conceptual Modelling Framework

In order to provide a holistic modeling framework, each layer has to be represented by a set of concrete modeling languages.

To enable the integration of different modeling languages the well-established Meta²Model [Karagiannis and Kühn 2002] is selected that allows the integration approach of different modeling languages. In this way methods on the same layer, as well as methods conceptually on different layers of abstraction can be integrated. Additionally the chosen approach provides the flexibility to make the used methods exchangeable.

Another important element of the framework is the tight integration of ontologies. These allow a homogenous access to the data contained in models of different modeling languages. In such a way ontologies may be used as standardized transport media for content. Similar to Web-Services that standardize the access to functionality, we consider that ontologies standardize the access to content.

The above principles are applied in the context of public administration to provide a holistic and flexible modeling framework for eGovernment. Figure 3 illustrates this modeling framework.

The so-called business modeling level implements the CIM layer by providing modeling languages for life events and business models, business processes as well as for business rules as the above figure depicts. Each of the modeling languages can be realized by different standards. Some languages are mentioned such as IDEF [IDEF 2008], BPMN [OMG BPMN 2008], ADO*eGov*® or similar languages for modeling business processes, as well as high level notations like Barbara von Halle to model business rules [von Halle 2001].

The PIM layer is implemented by high-level workflow modeling languages such as BPEL, SAWSDL [W3C SAWSDL 2008], OWL-S [W3C OWL-S 2008], for the execution of business processes and rule languages such as SWRL or RuleML [RuleML 2008] for the interpretation of business rules. Each of these modeling languages is aligned with an ontology that acts as a centralized model repository.

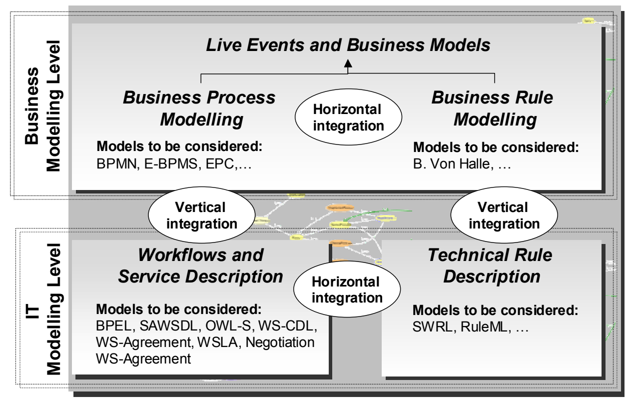

Figure 3: The eGovernment Modeling Framework

The strength of this modeling framework is the possibility to integrate the different models via an integration approach, enabling existing models and existing modeling notations to be used and enhanced with additional modeling aspects.

Based on [Kuehn 2004] there are so-called meta-modeling integration patterns that can be separated in vertical and horizontal integration:

Vertical integration is a typical top-down or bottom-up approach where different levels of abstraction are integrated. For the top-down integration the starting point are the elements of the higher-level method. Method fragments of the lower layer are selected and integrated based on the requirements from the upper method. Another possibility is the bottom-up integration, which is more common in reengineering attempts.

Between the Business Modeling Layer and the IT Modeling Layer vertical integration is necessary. This means that business processes and high-level business rules serve as starting point for a more detailed description on the technical level. The meta-models from the business modeling level are integrated and refined with the meta-models from the IT modeling level.

Horizontal integration is used for the integration of method fragments at the same layer of abstraction, which means we integrate meta-models with the same level of detail. Between business process modeling and business rule modeling, but also between workflows and executable rule description we need modeling languages integration.

Technical Realization of the Modeling Language Integration

The above framework has been implemented following the Meta^{2}Modeling approach. To ensure a centralized model repository that stores business processes as well as business rules BOC's Meta^{2}Model platform has been selected.

The repository is either a rational database or an XML [W3C XML 2008] database that implements the Meta^{2}Model. BOC's platform connects to this model repository and provides a Web-Service in order to make available modeling functionality such as searching, viewing, editing or creating models.

The complex part is to interpret the Meta^{2}Model information. This is implemented in a proprietary configuration language that implements the various modeling languages. In the above show case the four modeling languages ADO*eGov*®, BPEL, Business Rules according to Barbara von Halle, and SWRL and OWL have been implemented in order to be used on top of the Meta^{2}Model platform.

Each of the four modeling languages is implemented in one modeling tool that shows the current modeling language and stores the model in the centralized model repository.

The vertical Meta-Model integration is implemented via a so-called RDF-Tunnel between the modeling tools. Each modeling object can be identified via a unique model id. As the modeling tools are based on the Meta^{2}Model approach, they can refer to the same model repository. This simplifies the unique identification of modeling objects. If external modeling tools that do not implement the Meta^{2}Model approach - hence are not able to connect to the unique model repository - should be integrated additional adaptations are necessary to achieve unique model ids.

A reference from a model within one modeling tool to a model of another modeling tool is stored using the RDF [W3C RDF 2008] notation, where the unique model id from the origin and the unique model id from the target are stored.

In order to enable the access to this inter modeling tool references, the RDF-Tunnel is implemented as a Web-Service that can be filled and queried by the other modeling tools.

As the modeling tools are implemented as Web-Services, it is also possible to follow the inter-modeling tool reference and start the modeling tool that represents the referenced object.

This architecture enables the navigation not only between different model types within one modeling tool, but also the navigation between different modeling languages in several modeling tools.

The concrete realization of the tunnel within the modeling tools depends on the component of the tool. An inter-modeling tool reference within the HTML documentation of the models is much simpler than the realization of inter-modeling tool references within installed Client Applications. Currently the tunnels are on different implementation stages, depending on the component.

Figure 4 depicts the modeling tool structured in a service-oriented way. It identifies the centralized Meta^{2}Model platform that can be accessed via a Web-Service.

The modeling tool distinguishes between basis services that deal with data access, security issues, as well as with logging and reporting, as well as the component services that provide functionality for the users such as acquisition, modeling, analysis of models, model documentation and model exchange via import and export.

A Web-Application user interface implemented in AJAX [Jacobi and Fallows 2006] and Java Applets provides common access to the various services at the server side.

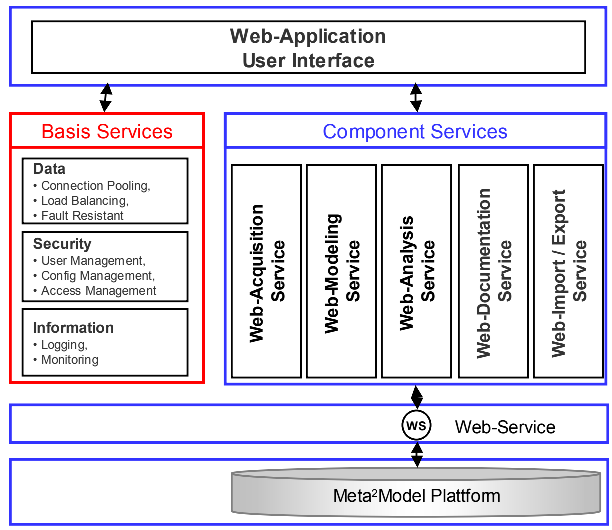

Figure 4: Architecture of the Meta^{2}Model Platform

Outlook

The next steps to be taken in the development of a flexible modeling framework follow a user-centric approach in order to allow flexible and dynamic service binding in accordance with the users/scenarios needs.

On a conceptual level the definition of modeling languages on all levels described above should follow a standardized approach by using ontologies to describe and derive the specific modeling language on one hand, and define the interaction and integration of different languages on the other hand.

From a technical perspective the goal is the provision of the modeling service framework via open-source channels. In addition a modeling service community should be built up, which will provide knowledge on the possibilities and implementation requirements to third-party service providers and allow integration of modeling services.

References

BREIN, EU-Project Brein. Accessible: http://www.eu-brein.com [10.10.2007]

Hinkelmann, K. et al. 2007. FIT Project Deliverable, D8: Semantic modelling of adaptable processes. Accessible: http://www.fit-project.org/publications.htm [5.10.2007]

IDEF Integrated Definition Methods. Accessible: http://www.idef.com/ [24.01.2008]

Jacobi, J., and Fallows, J. 2006. JSF and Ajax – Building Rich Internet Components, Apress L.P.

Karagiannis D. 1995. BPMS: Business Process Management Systems: Concepts, Methods and Technologies, SIGOIS Special Issue, SIGOIS Bulletin 10-13.

Karagiannis, D. and Kühn, H. 2002. Invited Paper, Metamodelling Platforms. In Proceedings of the Third International Conference EC-Web 2002 – Dexa 2002, Aix-en-Provence, France, September 2-6, 2002

Kühn, H. 2004. Methodenintegration im Business Engineering, PhD-Thesis.

Leutgeb, A., Utz, W., Woitsch, R., and Fill, H.-G. 2007. Adaptive Processes in e-Government - A Field Report about Semantic-Based Approaches from the EU-Project FIT. In Proceedings of the International Conference on Enterprise Information Systems (ICEIS 07), Funchal, Madeira – Portugal.

Lienhard, H. and Künzi, U.-M. 2005. Workflow and business rules: a common approach. BPTrends. Accessible: http://www.bptrends.com/publicationfiles/09%2D05%20WP%20Workflow%20and%20Business%20Rules%20%20Lienhard%20%2D%20Kunzi%2Epdf [02.10.2007]

OASIS BPEL, Web Services Business Process Execution Language V2.0. Accessible: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html [24.01.2008]

OMG BPMN, Business Process Modeling Notation (BPMN) Information. http://www.bpmn.org/ [24.01.2008]

OMG MDA, Model Driven Architecture. Accessible: http://www.omg.org/mda/ [24.01.2008]

Palkovits, S. and Wimmer, M. 2003. Processes in e-Government – A Holistic Framework for Modelling Electronic Public Services. In EGOV 2003, Springer Verlag, 213-219.

RuleML, Specification of RuleML. Accessible: www.ruleml.org/spec [24.01.2008]

Schacher, M. and Grässle, P. 2006. Agile Unternehmen durch Business Rules. Springer.

Stojanovic, N. et al. 2006. Fostering self-adaptive egovernment service improvement using semantic technologies. In AAAI Spring Symposium Series "The Semantic Web meets eGovernment", page 1, Stanford University, California, USA.

von Halle, B. 2001. Business Rules Applied – Building Better Systems Using the Business Rules Approach, John Wiley & Sons.

W3C OWL, Web Ontology Language (OWL). Accessible: http://www.w3.org/2004/OWL/ [24.01.2008]

W3C OWL-S, OWL-S Semantic Markup for Web Services, W3C Submission Accessible: http://www.w3.org/Submission/OWL-S/ [24.01.2008]

W3C RDF, Resource Description Framework (RDF). Accessible: http://www.w3.org/RDF/ [24.01.2008]

W3C SAWSDL, Semantic Annotations for WSDL and XML Schema. Accessible: http://www.w3.org/2002/ws/sawsdl/spec/ [24.01.2008]

W3C SWRL, A Semantic Web Rule Language Combining OWL and RuleML, W3C Submission. Accessible: http://www.w3.org/Submission/SWRL/ [24.01.2008]

W3C XML, Extensible Markup Language (XML). Accessible: http://www.w3.org/XML/ [24.01.2008]