Dealing with knowledge intensive services in e-Government A case study

Daniela Feldkamp, Knut Hinkelmann, Holger Wache
University of Applied Science Northwestern Switzerland,
Riggenbachstrasse 16, CH-4600 Olten, Switzerland
daniela.feldkamp@fhnw.de, knut.hinkelmann@fhnw.ch, holger.wache@fhnw.de

Abstract: Governmental processes are complex and knowledge-intensive. Most process management systems fail to support them in an adequate way. On the other hand semantic technologies like Semantic Web Services allow to incorporate knowledge in process. But often this techniques are overdimensioned and can not be executed properly. Here we proposed a more practical motivated approach where knowledge-intensive parts of a process are controlled, enacted and supported by business rules on top of ontologies. With the help of a case study we demonstrate how to model knowledge-intensive processes.

1 Introduction

Public services are based on legal rules and regulations binding for all municipalities. These rules and regulations are exposed to rapid changes. Furthermore these processes are dealing with people's concerns which means they are dealing with different circumstances every time. In this sense e-government services are often knowledge intensive processes where in general the actual process execution and the involved participants and administrations depend on various factors including

- exceptional situations
- unforeseeable events
- unpredictable situations, high variability
- highly complex tasks

Consequently knowledge intensive services are weakly structured and can only be automated to a very limited extent. Modeling all possible variants of a process can lead to complex process model, which is hard to maintain. Sometime, this modeling is even impossible if the tasks are mutually depending on each other.

Roger T. Burlton propose: "Separate the know from the flow" (cf. (Ros03)). Instead of trying to model complex knowledge in the control flow Burlton encourage to model this knowledge explicitly in a more appropriate format and use it to guide the control flow.

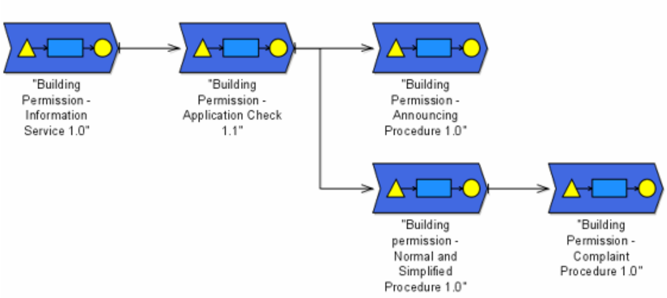

Figure 1: Processmap

Semantic Web Services are one possibility to represent control flow and knowledge at the same time. But unfortunately formalism like OWL-S have some conceptual deficits (Sab05) and by now no stable enacting environment exists.

In our practical motivated approach we use business rules to represent this knowledge. The knowledge-intensive tasks of a process are separated from the normal control flow and encapsulated in so-called variable processes. Business rules control the enactment of these separated tasks. In the following we explain with the help of a case study how business rules can be used to model flexible process behavior and how they help to model complex and knowledge-intensive process in an elegant way. But first we introduce in our case study.

2 Analysis of existing e-government services

To find an appropriate use case we evaluated several e-government services from different municipal authorities. For the case study we selected the ”Building Permission” process of the Stadt Vöcklabruck[1] as the most interesting service. It comprises five sub-processes (”Building Affairs Information Service”, ”Application Check”, ”Announcing Procedure”, ”Normal and Simplified Procedure” and ”Complaint Procedure”). The service is rather complex involving different actors and consisting of several knowledge intensive tasks. In fact it is hard to model this process with the classical, static process model in an adequate way because of its complexity.

Figure 1 depicts the process landscape with its five sub-processes:

- *Building Affairs Information Service* (in German: ”Baubewilligung Information Service”): In this sub process the citizen navigates on the web site, selects the appropriate service and submits the application form.
- *Application Check* (in German: ”Formale Prüfung”): Based on the data the application form the appropriate approval process is selected.
- *Announcing Procedure* (in German: ”Bauanzeige”): The applicant provides data

[1] Web site of Stadt Vöcklabruck: http://www.voecklabruck.at/.

about the planned construction. The administraion checks whether the construction is according to legal and building regulations.

- *Normal and Simplified Procedure* (in German: "Baubewilligung"): The process flow is like the "Announcing Procedure". In contrast to the "Announcing Procedure" neighbours and other citizen must be involved in the process.
- *Compliant Procedure* (in German: "Berufung"): A formal objection is possible against any official notification.

In particular the process "Application Check" based on the citizens' application data, the appropriate approval process is selected for our case study because of its complexity. Figure 2 (I) tries to illustrate the process in the traditional way. For example to approve a building application it may be necessary to conduct several checks including inspection on location (task C), approval of application by historical preservation agency (task B), and approval of environmental compatibility (task A). All tasks depends on each other. The control flow in Figure 2 (I) tries to express that the outcome of one task (e.g. A) may require the involvement of the other task (e.g. B and/or C). However the outcome of one task also may be that the application will fail; then no further test are required. Incorporating this behavior in the control flow would increase the complexity enormously. In fact not all aspects of the control can be modeled in an appropriate way.

3 Supporting knowledge-intensive processes

To deal with knowledge intensive services, we separate the know from the flow. In order to capture the knowledge-intensive part of processes and to model them declaratively and adequately our approach base on Semantic Web techniques. Ontologies are used to capture the domain knowledge by introducing and relating the terms of the domain. Ontology instances refer to the current process. Based on the ontology instances business rules (vH02; Bol01) express the process knowledge. In particular business rules extend process execution on three ways:

- *Variable process execution*: Determine activities and processes to be executed thereby accounting for dependencies between activities.
- *Intelligent resource allocation at run time*: Selection of employees based on special skills and selection of particular web services adequate for the actual circumstances.
- *Intelligent branching and decision making*: deriving workflow-relevant data using inferences and computing values.
- *Consistency checking*: Avoid violation of integrity constraints and guidelines.

Variable processes, which could hardly be modeled by the classical strict process model, are replaced by a new object type, labeled with "KIT" (Knowledge intensive task). This

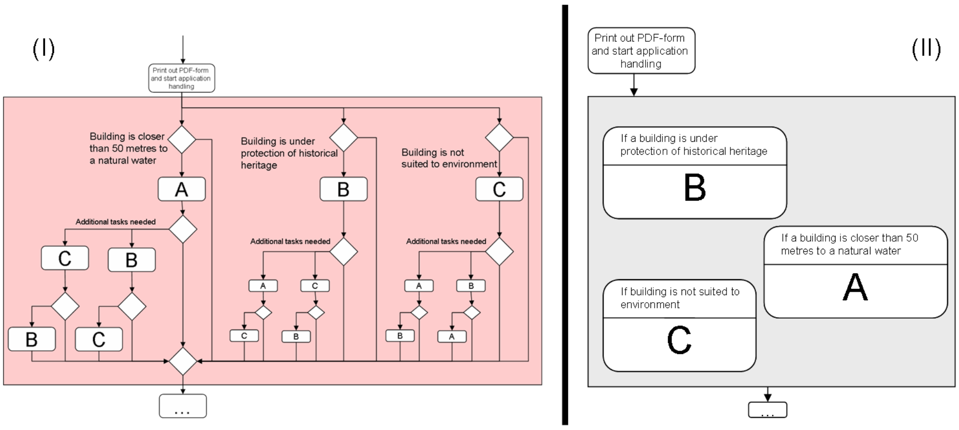

Figure 2: Comparison of the Process Models

object is related to a pool of activities, which are linked to action-enabling rules. At run time the associated action-enabling rules[2] select the activities that have to be executed depending on the actual context of the process instance. All activities and processes can be linked to inference rules, which allow for resource allocation and support the user in decision making, while integrity constraints and guidelines (in combination with inference rules) ensure consistency checking and compliance.

For Stadt Vöcklabruck we replace knowledge-intensive tasks by ”KIT”. All possible activities, which should be executed in a specific context, are linked to business rules.

Figure 2 (II) illustrates the service model created using our approach. Each of the three tasks is related to an action-enabling rule. These rules are invoked first to dynamically determine and instantiate the appropriate tasks.

The outcome of a process may require the involvement of another task; but here we do not need to express this explicitly. As an effect of the invocation the ontology instances may be modified. As a “side effect” further rules can be fired probably resulting in the instantiation of further tasks. For example, when the historical preservation agency has changed the building project and requests an inspection on location (task C), this results in a change of the application data in the ontology. The action-enabling rule for task C can fire and task C will be instantiated.

Storing rules independent from the flow has the advantage of modifying rules independently from the business logic. For example, the modification of the rule which invoke task A ”*If the building is closer than **50** meters to natural water*” to ”*If the building is closer than **100** meters to natural water*” will result in one modification in our approach. In the traditional model the expression has to be changed in several switches/gateways.

At the moment we investigate the concrete architecture for invoking the rules in the process environment. More precisely a BPEL(AW03) engine is used to execute the standard part of processes but knowledge-intensive and flexible activities are controlled, enacted and

[2]Action enabling rules are rules, which trigger another rule or a process (step) (vH02)

supported by an integrated rule engine over the ontologies.[3]

4 Conclusion

Many processes can not be modeled and executed with current workflow engines; even government processes are knowledge-intensive and require flexible execution with the ability to handle exceptions. On the other hand Semantic Web technology offers the ability to incorporate the knowledge into process execution. But current approaches like OWL-S lack a mature implementation and may provide unneeded and immature functionality (Sab05). We therefore proposed a more light-weighted process execution which incorporates business rules for knowledge-controlled and flexible enactment and ontologies as the common knowledge base.

References

[AW03] F.; Dholakia H.; Goland Y.; Klein J.; Leymann F.; Liu K.; Roller D.; Thatte Trickovic I. Andrews, R.; Curbera and S. Weerawarana. *Business Process Execution Language for Web Services*, May 2003.

[Bol01] Harold Boley. The Rule Markup Language: RDF-XML Data Model, XML Schema Hierarchy, and XSL Transofrmations*. INAP2001, October 2001. Invited Talk.

[MHS07] Boris Motik, Ian Horrocks, and Ulrike Sattler. Bridging the Gap Between OWL and Relational Databases. In *Proc. of the 16th International World Wide Web Conference (WWW 2007)*, Banff, Alberta, Canada, May 8–12 2007. To appear.

[Ros03] Ronald G. Ross. *Principles of the Business Rules Approach*. Addison-Wesley, 2003.

[Sab05] Marta Sabou. *Building Web Service Ontologies*. Phd thesis, Vrije Universiteit Amsterdam, 2005.

[vH02] Barbara von Halle. *Business Rules Applied, Building Better Systems Using the Business Rules Approach*. Wiley and Sons, New York, 2002.

[3]The authors are aware about the limitations to combine ontological (i.e. description logic) reasoning with rule reasoning. First we want to use a very restricted representation formalism for rules and ontology. Second the rules are also restricted to access only instances. In consequences our approach may be related to the database-oriented view (cf. (MHS07)).