



**HAL**  
open science

# Adaptive and Compliant Policy Implementation: Creating Administrative Processes Using Semantic Web Services and Business Rules

Yiwei Gong, Marijn Janssen

► **To cite this version:**

Yiwei Gong, Marijn Janssen. Adaptive and Compliant Policy Implementation: Creating Administrative Processes Using Semantic Web Services and Business Rules. 12th Conference on e-Business, e-Services, and e-Society (I3E), Apr 2013, Athens, Greece. pp.298-310, 10.1007/978-3-642-37437-1\_25 . hal-01470541

**HAL Id: hal-01470541**

**<https://inria.hal.science/hal-01470541>**

Submitted on 17 Feb 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Adaptive and Compliant Policy Implementation: Creating Administrative Processes using Semantic Web Services and Business Rules

Yiwei Gong and Marijn Janssen

Faculty of Technology, Policy and Management, Delft University of Technology,  
Jaffalaan 5, 2628 BX Delft, The Netherlands  
{Y.Gong, M.F.W.H.A.Janssen}@tudelft.nl

**Abstract.** Adapting to and complying with frequently changing policy against low costs requires flexible business processes. Yet, existing systems are unrelated, consist of operating silos, involve many human translation tasks and policies are hardcoded in business processes. From a technology perspective, both Semantic Web Service (SWS) and Business Rule (BR) can be used to create flexible business processes that can adapt to changes in policy. Despite their potential these technologies are seldom integrated or used in administrative organizations. SWS can be adopted to present functionality needed in processes, whereas the elicitation of BR is an important ingredient for process creation. In this paper, we derive and test guidelines for creating administrative processes that are dynamical created using SWSs and BRs that are derived from policies. A case study is used to demonstrate and evaluate the guidelines. The guidelines can serve as a foundation for developing agile and compliant architectures.

**Keywords:** Business Process Management, Business Rule, Semantic Web Service, Compliance, Administrative organization, Flexibility, Adaptability

## 1 Introduction

Policy changes frequently and needs to be implemented by administrative organizations against low costs and within a short time frame. This has already resulted in a broad acceptance of Service-Oriented Architecture (SOA), which are merely used for creating static processes [1], but which help to reuse components during design time. As such, process design has been limited to pre-defining processes. Yet, frequent changes in policy require that defining processes should not be labor-intensive and time-consuming in advance but that they are created on the fly during run time. In the effort of achieving higher process flexibility, both Semantic Web Service and Business Rule technologies are used. But they are hardly used in integration, whereas the integration might result in higher levels of adaptability.

*Semantic Web Service* (SWS) provides a declarative, ontological framework for describing services in a machine-readable format that can also facilitate logical reasoning [2]. Using SWS, service descriptions can be interpreted based on their mean-

ings and can be composed into complex processes corresponding to the needs of a service request [3]. While transforming the organization into a service-oriented manner, the impact and requirement of a SOA on the organization and the way how resources are properly allocated in such a service-oriented organization is still not clear. However, the policy implementation strongly influences the operational processes in public organizations and complicates their use of SWS technology.

A *Business Rule* (BR) is a statement that defines or constrains some aspects of the business. It is intended to assert business structure or to control or influence the behavior of the business [4]. Although there are many knowledge representation techniques that can be used to express BRs, the rule technique is most frequently used in expressing BRs. In a classical rule-based approach, the logic of a process is represented by a set of rules, which are associated with business activities and specify the properties of them [5], e.g. the preconditions of its execution. It is a challenge for the administrative organizations how they can consistently and conveniently use BRs among those different formats.

BR is a common approach in knowledge intensive organizations [6, 7]. Aligning SWS and BR technology allows dynamic creation of processes in which pre-definition of processes is no longer necessary. To enable this, administrative organizations must integrate SWS and BR components. Integrating SWS and BR requires the organization to have a proper technical architecture solution, which has been reported in [8]. In this paper, we abstract from the architecture by deriving guidelines to create adaptive processes using both SWSs and BRs, taking into account the flexibility and compliance of the processes. The guidelines are generic, and can be implemented using different technologies. The rest of this article is organized as follows. In the next section we discuss the typical policy implementation procedure in administrative organizations. Then, we discuss the guidelines for administrative organizations to implement policy into operational processes using SWSs and BRs. Thereafter, we use an illustrative case study to demonstrate the modeling process. The paper ends with a conclusion and prospects for future work.

## 2 Policy Implementation

Roughly speaking, policy implementation takes the new or updated legislation of policy as a starting point, translates this into executable policy and finally implements this in the operational administrative processes [9]. Policy regulates human behavior in the real world [10] and is often written in natural language. Administrative organizations take the policy as input and implement it in their operational processes. For this purpose they need to first translate the policy into a computer-readable format which requires policy-interpretation and the collection of additional information. Once the policy is specified, it can be executed using business processes. Table 1 summarizes the expression of policy at different levels, and their possible formats and users. Policy-making is a political activity dominated by politicians and lawyers which provides often the starting point for implementing policy, whereas policy representation is the knowledge representation of policy that can be understood by com-

puter applications. Operational processes concerns the execution of policy by running business processes.

**Table 1.** Characterization of policy at different levels

<b>Level</b>	<b>Typical Formats</b>	<b>Typical Users</b>
1. Policy-making	Text	Humans
2. Policy Representation	Ontology models, RIF rules	Inference engine, process engine
3. Administrative processes	ECA rules, Production rules, Programming languages	Software agent, Web Service, process engine

## 2.1 Policy-making

Policy is formulated with the objectives of realizing desired behavior. Policy can be ambiguous and different sources of policy might be conflicting. Therefore, policies are not directly executable and need to be interpreted using extra information. Legal experts need to first determine what the requirements are in the decision-making related to legislation and then find out what materials (documents or other forms of information) should be provided to prove whether the requirements have been satisfied. After that, the providers of those materials and the requirements of those materials all need to be clarified. In the example of the Highly Skilled Migrant policy (see Section 4.1), legal experts need to first identify the income requirement, education level requirement, etc. Then they need to interpret what documents are required, including, for example, an employment contract, salary sheet and diploma. This step might add extra requirements to those of legislation, e.g., “the applicant must provide salary statements of the past 3 months”. All these requirements are part of the knowledge necessary to make a decision about the application. After that, they also need to indicate the sources of the required materials or information. For instance, the salary sheet is from the migrant, and the employment contract is from the employer. Different options in collecting certain information are often feasible. Finally, a sequence of document checking and decision making is formed. The policy interpretation can be considered as a high level process design. Although such an interpretation is human based and not understandable by computers, the business logic and requirements described in the interpretation make it critical for following process design.

## 2.2 Policy representation

For deploying the business logic and requirements into the ICT systems of the organization, the interpretation is further modeled during knowledge engineering [11]. According to Boer et al. [12], knowledge models such as Web Ontology Language (OWL) models can be built based on sources of text, and traceability can be maintained between the models and the texts. Knowledge acquisition from text is a complicated and well-established knowledge engineering technique, and it is out of the scope of this paper. We focus on the challenge to connect ontology models, e.g., those represented by OWL, to BRs and SWSs in operational processes. Besides, the recent

development of BRs in the Semantic Web domain focuses on a standard representation of BRs, e.g., the Rule Interchange Format (RIF) from W3C.

### 2.3 Administrative processes

Automatic operational processes involve the execution of BRs that directly control process behaviors. Operational rules describe how an operation should be done. Operational BRs are subdivided into event-condition-action (ECA) rules and production rules. ECA rules are rules of the form “ON Event IF Condition DO Action”. Production rules are rules of the form “IF Condition DO Action”. Despite various rule technology solutions, rules are usually only applied in decision making in an automated process. Weigand et al. [13] have pointed out that many Business Rule approaches are restricted as they “concentrate on the application for decision-making and do not address the business process constraints that underlie service composition”. In a SOA environment, this means the BR is hard coded in process, like BPEL. Although BPEL is the de facto standard for business process descriptions, BPEL is not flexible [13]. Service composition in BPEL relies on hardcoded BRs. This results in higher complexity of maintaining and updating processes and BRs, which is often a labor-intensive job. Furthermore, once the policy has been manually interpreted and translated to BRs and if it has been mixed with other sources of information, process compliance becomes ambiguous.

## 3 Guidelines for Creating Administrative Processes

Administrative organizations desire flexible business processes and must ensure that the business processes comply with policy. Concerning that many administrative organizations have BR and SOA infrastructure that support the second and third levels of the previous section we formulated the following guidelines for those organizations to use BR and SWS for dynamic creation of business processes. Those guidelines allow a clear coordination of the three levels, which is conceptualized in Fig.1. This integration will result in dynamic creation of business processes which can flexibly respond to the changes of policy and ensure the compliance all the time.

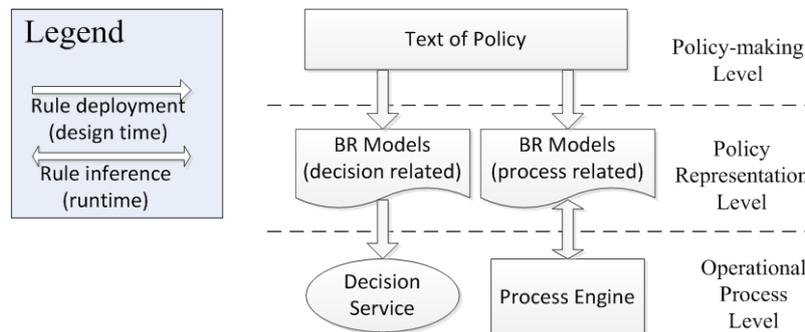


Fig. 1. Coordinating the components at the three levels

**Guideline 1: Develop a domain ontology to create interoperability between SWSs and BRs.**

Studies about the integration of Web Service and Business Rule technologies mainly focus on using BRs in the service composition. Orriëns et al. [14] have indicated that BRs can be used to determine how the composition should be structured and scheduled. BRs and Web Services were not compatible because they use different ontologies. Although they both can be semantic, they are not having a collective semantics foundation, as the same term/object can have different properties and a different relationship with other terms in different ontologies. Without a unified ontology, BRs for service composition had to know the exact interface of related services instead of their semantic description. This renders the semantic level of Web Services meaningless. D'Mello et al. [15] argued that rule driven composition requires BRs and SWSs to be documented in a unified ontology for successful composition. Besides, many BRs are originated from sources of law. The consistency of those BRs has to be concerned for compliance with policy.

**Guideline 2: Distinguish between decision and supportive services.**

In a service-oriented system, services can be invoked from anywhere and managed independently. This allows the encapsulating of all forms of resources (provided by either humans or computers) into services and Semantic Web Service technology is used to describe them. For administrative organizations, we suggest defining two kinds of Business Services. The term “decision service” is used to describe a self-contained callable component with a view of all conditions and actions that need to be considered in making an operational business decision [16]. Decision services refer to policy, which implies that the use of these decision services is mandatory and they are the main services that can satisfy the service request with final decision making. Containing the policy related decision making logic, it is an essential step of the process to comply with legislation enforcement, but might not be able to satisfy the request on its own. In the execution of the decision services, other supportive services might need to be composed with it. In contrast, supportive services are not derived from policy but depend on other concerns of the organization. Supportive services can be used as infrastructural services with high reusability. Services like information retrieval services have a purpose to provide information/data for decision making. They allow a single information source to be reusable for multiple decision services. Other supportive services may facilitate decision services by providing user interfaces or report generation. They should be designed to be reusable as much as possible.

**Guideline 3: Separate decision related and process related business rules.**

BRs for decision making are widely used in administrative organizations. Yet, the contribution of BRs is not limited to decision making. In a SOA environment business rules is also able to control the composition of services into processes. Creating process related business rules avoid the logic of process flows being hardcoded in the applications (normally within a process unit for connecting the sequenced tasks). Distinguishing decision and process related BR allows a clear traceability from the policy making level to operational process level (see Fig. 1).

The separation of decision and process related BRs allows the decision services to encapsulate decision related BRs. A clear allocation of decision related BRs enable a

fast response of changes in policy. Identifying process related BRs can avoid hard-coded BRs in process engine. This allows easier adaption of process flows.

The above three guidelines aim to facilitate the design of Business Process Management systems for administrative organizations. Those guidelines are technology independent and allow the organization to improve their current BR and SOA infrastructure. In the next section we will use an illustrative case study to demonstrate how to use those guidelines to create business processes for implementing policy.

## **4 An Illustrative Case Study**

The Immigration and Naturalization Service (Immigratie en Naturalisatie Dienst, or IND) handles the admission of foreigners in the Netherlands. It is responsible for the execution of a complex set of regulations coming from different policies, including international law, national law, case decisions and so on. IND makes a large number of decisions, i.e. some 300.000 a year in the areas of asylum, standard objections to decisions, and naturalization. Under pressure from frequently changing policies, IND is one of the administrative organizations that have a desire for flexible and compliant business processes. In this chapter, we use the guidelines described in the previous chapter to show how they can help IND to realize this goal.

### **4.1 Adaptive Policy Scenario**

A use case dealing with a Highly Skilled Migrant (HSM) policy that applies for a residence permit was selected to demonstrate the use of our guidelines. The policy is introduced to enable qualified foreigners to work in the Netherlands. In 2007, the annual income limitation on an applicant is at least 46.541 EUR or 34.130 EUR if the applicant is under 30. In 2008 there was a change in policy to encourage foreign intelligent graduates to work in the Netherlands. The foreign graduate who obtained a Bachelor or Master Degree at an accredited Dutch educational institution within one year before becoming employed, the income limitation was changed to 25.000 EUR annually.

In 2009 there was another change in legislation. Employees must have a gross annual income of at least 49.087 EUR, or 35.997 EUR if they are under the age of 30. There are two different situations to which the reduced wage criteria apply (25.800 EUR gross a year). The first situation is aimed at graduates that obtained a Bachelor or Master Degree at an accredited Dutch educational institution, similar to the 2008 situation. The second situation concerns Master and Ph.D. students who graduated in the Netherlands or at a university listed in the top 150 of two internationally recognized rankings.

### **4.2 Building a Domain Ontology**

In our modeling approach, the first step is to define a domain ontology to provide consistent understanding of concepts. In Fig. 2 most concepts used in service descrip-

tion are involved and connected. For space reason, we didn't include every concept as this figure is enough to demonstrate the structure of a domain ontology.

As a typical case system, the domain ontology contains the identification of different clients under the *Person* class. We also defined an *Action* class, and its two subclasses with different semantic meanings. *Check* is for the provision functionality providing data by a supportive service. *Decide* indicates a decision service for providing a decision-making. *Resident Permit* and its subclasses sound like services. But in this example, they are final deliverables of processes instead of services. Therefore, they are defined as entities in the ontology model. Moreover, the data property of an entity (e.g. "person" has "ID" and "date of birth"; "applicant" has "income"; "graduate" has "institute" and "degree", "graduation date"; "decision" has "result") might eventually need to be included into the domain ontology for operational reasons. To improve the visual appeal, data properties are not presented in the diagram.

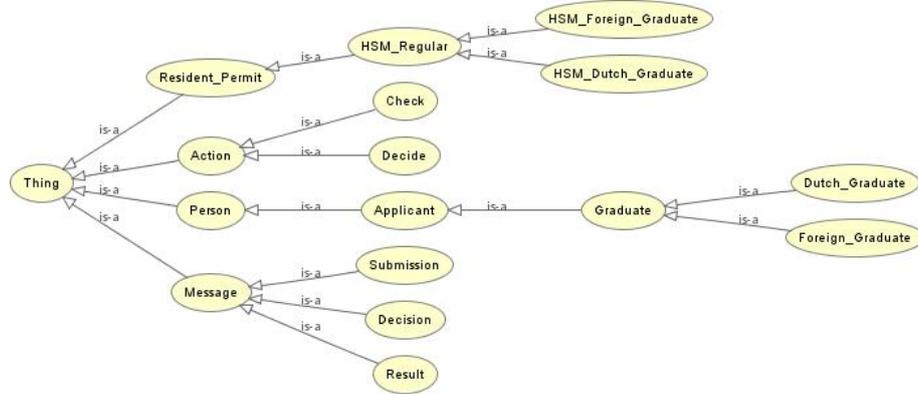


Fig. 2. Illustrative domain ontology in the HSM case (created using Protégé)

### 4.3 Defining Business Services

Using those concepts provided by the domain ontology model, we listed the designed services that used in HSM processes and their semantic representation in the following table. BSs are redefined using the criteria of Parnas [17], who suggested that each module should hide some design decision from the rest of the system. To simply express these services, we adopt a SWS description containing its functional semantics and the I/O information, so that a Web service can be specified as {service functionality, input set, output set} [18]. These services are summarized in the following table. Due to space reason, please refer to [8] for further detail of SWSs and domain ontology specification.

We assume that the system needs three decision services to decide whether an application can have a HSM resident permit. We need three different DS for this, because the policy distinguishes regular applicants, Dutch graduates and foreign graduate with different criteria for decision-making. More DS can be added if further exceptions are issued. Manual decision-making task can also be added as a new DS into

the system to allow the control of special cases in the system. In this example, we only involve automatic DSs.

The system also needs several supportive services. To start the process, the system needs at least one supportive service (SS01) for submission intake. Alternative intake services can be issued to allow multiple submission channels (e.g. one for electronic channel, and one for front offices). We also need a supportive service (SS02) to check the income of applicants, a supportive service (SS03) to check the Dutch educational institution if the applicant is a Dutch graduate, and a supportive service (SS04) to check the top 150 ranking if the application is the applicant is a foreign graduate. Furthermore, we need one supportive service (SS05) for informing the application about the final result of decision making, namely whether he or she will receive a resident permit.

**Table 2.** The example of services in the HSM case

<b>SWS</b>	<b>Specification</b>	<b>Description</b>
Decision Services		
DS01	{{Decide, HSM}, { Income, Date_of_Birth}, {Decision}}	Decide whether the application satisfies regular HSM requirement
DS02	{{Decide, HSM_Dutch_Graduate}, { Income, Date of Birth, Degree, Result (Institute), Graduation_date}, {Decision}}	Decide whether the Dutch graduate satisfies HSM requirement
DS03	{{Decide, HSM_Foreign_Graduate}, { Income, Date_of_Birth, Degree, Result (Ranking), Graduation_date}, {Decision}}	Decide whether the foreign graduate satisfies HSM requirement
Supportive Services		
SS01	{{Check, Income}, {ID}, {Income}}	Check the income information
SS02	{{Check, Institute_Accreditation}, {Institute}, {Result (Institute)}}	Check whether an institute is accredited Dutch educational institution
SS03	{{Check, Ranking}, {Institute}, {Result (Ranking)}}	Check whether an institute is listed in the top 150 of a certain ranking

All the vocabulary used in SWS description and also the BRs descriptions in the next section should be involved in a domain ontology model, because the use of those vocabulary and terms must be consistent to allow understanding between components.

#### 4.4 Defining Business Rules

To ensure compliance with policy, BRs that are used in the decision service are separated from the BRs that describe the process logic. Rules used in decision service are closely related to the legislation. In the HSM case, the legislation directly indicates the hard requirements on the age and income of applications. This (piece) of

legislation should be connect to the rule to enable traceability. Being traceable to the piece of legal text allows easy maintenance of the rules, if any changes in legislation can impact them. Many expert systems can fulfill such a requirement. The BRs used by Web services are mainly productive rules. Their forms can be various depending on the implementation technology of the Web services. In our prototype, BRs are represented in RIF-Core [19]. An example of rules used in decision service DS01 is given as follows. This rule strictly responds on the HSM regular application according to the criteria of 2009. Note that most of the IRIs used in the examples of this paper are fictitious and do not represent real entities. To improve the visual appeal, the example takes advantage of the syntactic shortcuts (allowed by the RIF Datatypes and Built-ins specification). For instance, 100 is used instead of "100"^^"xs:integer". For the same reason, we directly use the mathematical and binary relations operations. For example, in practice, ?a>10 should be defined as External(pred:numeric-greater-than(?a 10), where pred is the prefix defined as Prefix(pred <http://example.com/rif-builtin-predicate#>).

```
Prefix(func <http:// example.com/rif-builtin-function#>)
Prefix(act <http://example.com/action#>)

Forall ?decision ?income ?date_Of_Birth(
  If And(?decision # act:decision
        Or(?income >= 49087
           And(?income >= 35997
              External(func:age(?date_Of_Birth)<30)
            )
        )
  )
  Then Do(Modify(?decision[result->'granted']))
)
```

To create dynamic processes, we also need process related BRs, namely the BRs that are executed by the process engine. The crux is indicating which decision service should be invoke by process engine to deal the applications they are processing. As the functions of those decision services have been semantically described, the BRs then should indicate which decision service is for which kind of applications. We use the following RIF-core rules for giving an example. In this example, the Base directive provides yet another shortcut. The Base directive expands <hsm\_Regular> IRIs to "http://example.com/service#hsm\_regular"^^rif:iri.

```
Prefix(act <http://example.com/action#>)
Prefix(per <http://example.com/person#>)
Base(<http://example.com/service#>)

Forall ?Applicant(
  act:decide(?Applicant <hsm_Regular>):- ?Appli-
  cant#per:applicant
```

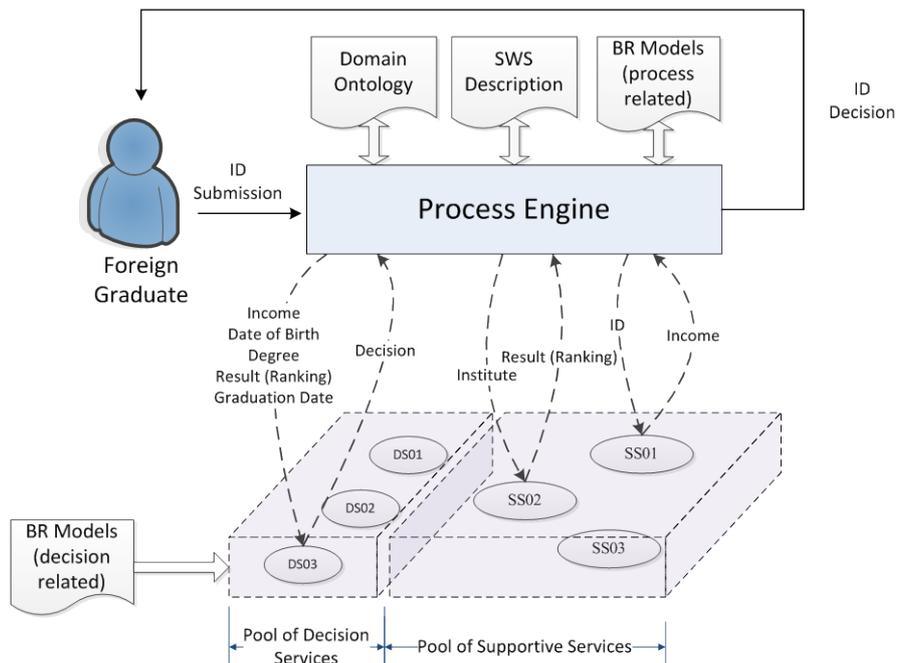
```

act:decide (?Applicant <hsm_Dutch_Graduate>):- ?Applicant#per:dutch_graduate
act:decide (?Applicant <hsm_Foreign_Graduate>) :- ?Applicant#per:foreign_graduate)

```

#### 4.5 Testing Process Creation

In process creation, the following figure demonstrates the interactions among different system components. Process Engine first interacts with clients and receives HSM application submissions. To process the submission, the Process Engine invokes related Web services to make use of resources or knowledge for decision making. Finally, the decision result will be returned to the clients. The invocation of decision services is steered by the process related BRs. The selection of supportive services is based on the matching according to the semantic description of their functionalities and IOPEs.



**Fig. 3.** An example of process creation for HSM foreign graduate applicant

To evaluate our guidelines, we invited IT architects, legal experts and process managers from IND to review the guidelines and the case study. Based on their experience, the IND experts evaluated our guidelines and case study by comparing them with their current way of business process management. The evaluation shows the added value of the guidelines including:

1. Faster adaptation of business processes to adopt changes in policy. As decision related BRs are allocated into decision services. It is easier to track the impact of

changes in policy. The creation of processes is conducted by process related BRs. Adapting and testing those BRs is easier than reconfiguring the process engine.

2. Easier check for compliance with policy. As hardcoded and pre-defined processes can be avoided, checking the consistency between BRs and policy interpretation would be easier than checking the consistency between hardcoded processes and policy interpretation. If any mistake happens, conflicting logic is easier to find out in BRs models than in hardcoded processes.
3. Clear integration of SWS technology. Being aware of how SWSs is used in process modeling facilitates the adoption of them.

## 5 Conclusion and Future Work

To satisfy requirements from frequently changed policy, administrative organizations desire adaptive and compliant policy implementation. This requires coordination of policy-making, policy representation and operational process level, and dynamic creation of operational processes. For having adaptability and compliance, the organization needs to coordinate functionality in process creation, and ensure communication between the elements. The proposed guidelines describe how to use BR and SWS technology to create administrative processes. A domain ontology enables communication between SWS and BR to achieve interoperability between those technologies. This allows them to share the same interpretation of the terms used in different components. Web services are used as the way to access and manage resources of the organization. SWS technology is used to describe the services such that BRs can understand them. Dynamic processes are created by process engine according to the BRs and using Web services as an ingredient. As BRs are not hardcoded in processes, the compliance to policy will be easier to check. Administrative organizations that implement frequently changed policy and would like to integrate multiple technologies to achieve higher flexibility in their business processes can benefit from those guidelines.

For future research, our guidelines provide a foundation for developing flexible and adaptive architectures and processes. The use in cross-organizational processes and supply chain management can be explored.

## References

1. Weske, M.: Introduction. In: Kuroпка, D., Staab, S., Tröger, P., Weske, M. (eds.): *Semantic Service Provisioning*. Springer, Berlin Heidelberg (2008)
2. Wang, H.H., Gibbins, N., Payne, T.R., Redavid, D.: A formal model of the Semantic Web Service Ontology (WSMO). *Information Systems* **37** (2012) 33-60
3. Sabol, T., Furdík, K., Mach, M.: Employing Semantic Technologies for the Orchestration of Government Services. In: Vitvar, T., Peristeras, V., Tarabanis, K. (eds.): *Semantic Technologies for E-Government*. Springer, Heidelberg (2010) 47-74
4. Morgan, T.: *Business rules and information systems: aligning IT with business goals*. Addison-Wesley, Indianapolis, IN, USA (2002)

5. Lu, R., Sadiq, S.: A Survey of Comparative Business Process Modeling Approaches. In: Abramowicz, W. (ed.): *Business Information Systems, Lecture Notes in Computer Science*, Vol. 4439. Springer, Heidelberg (2007) 82-94
6. Van der Aalst, W.M.P., Weske, M., Grünbauer, D.: Case handling: a new paradigm for business process support *Data & Knowledge Engineering* **53** (2005) 129-162
7. Lienhard, H., Künzi, U.-M.: Workflow and Business Rules: a Common Approach. In: Fischer, L. (ed.): *Workflow handbook 2005*. Future Strategies Inc., Lighthouse Point (2005) 129-140
8. Gong, Y., Janssen, M.: Creating Dynamic Business Processes Using Semantic Web Services and Business Rules. In: Janowski, T., Davies, D. (eds.): *5th International Conference on Theory and Practice of Electronic Governance (ICEGOV 2011)*. ACM Press, Tallinn, Estonia (2011)
9. Gong, Y., Janssen, M.: From policy implementation to business process management: Principles for creating flexibility and agility. *Government Information Quarterly* **29** (2011) 61-71
10. Hu, Y.-J., Yeh, C.-L., Laun, W.: Challenges for Rule Systems on the Web In: Governatori, G., Hall, J., Paschke, A. (eds.): *Rule Interchange and Applications, Lecture Notes in Computer Science*, Vol. 5858. Springer, Heidelberg (2009) 4-16
11. Kordelaar, P., Van Teeseling, F., Hoogland, E.: Acquiring and Modelling Legal Knowledge Using Patterns: An Application for the Dutch Immigration and Naturalisation Service. In: Cimiano, P., Pinto, H.S. (eds.): *Knowledge Engineering and Management by the Masses, Lecture Notes in Computer Science*, Vol. 6317. Springer, Heidelberg (2010) 341-349
12. Boer, A., Van Engers, T., Winkels, R.: Traceability and Change in Legal Requirements Engineering. In: Casanovas, P., Pagallo, U., Ajani, G., Sartor, G. (eds.): *AI Approaches to the Complexity of Legal Systems. Complex Systems, the Semantic Web, Ontologies, Argumentation, and Dialogue, Lecture Notes in Computer Science*, Vol. 6237. Springer, Heidelberg (2010) 74-92
13. Weigand, H., Van den Heuvel, W.-J., Hiel, M.: Business policy compliance in service-oriented systems. *Information Systems* **36** (2011) 791-807
14. Orriëns, B., Yang, J., Papazoglou, M.P.: A Framework for Business Rule Driven Service Composition. In: Benatallah, B., Shan, M.-C. (eds.): *Technologies for E-Services, Lecture Notes in Computer Science*, Vol. 2819. Springer, Heidelberg (2003) 14-27
15. D'Mello, D.A., Ananthanarayana, V.S., Salian, S.: A Review of Dynamic Web Service Composition Techniques. In: Meghanathan, N., Kaushik, B.K., Nagamalai, D. (eds.): *Advanced Computing, Communications in Computer and Information Science*, Vol. 133. Springer, Heidelberg (2011) 85-97
16. Taylor, J., Raden, N.: *Smart (Enough) Systems: How to Deliver Competitive Advantage by Automating Hidden Decisions*. Prentice Hall Press (2007)
17. Parnas, D.L.: On the criteria to be used in decomposing systems into modules. *Communications of the ACM* **15** (1972) 1053-1058
18. Shin, D.-H., Lee, K.-H., Suda, T.: Automated generation of composite web services based on functional semantics. *Web Semantics: Science, Services and Agents on the World Wide Web* **7** (2009) 332-343
19. W3C: RIF Core Dialect (W3C Proposed Recommendation 22 June 2010). World Wide Web Consortium (2010) <http://www.w3.org/TR/2010/REC-rif-core-20100622/>