Integrating Semantic Business Policy into Web Service Composition

Xu Meng and Chen Junliang

State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications,
Beijing, China
xumengmoon@gmail.com, chjl@bupt.edu.cn

Abstract. Web services composition is becoming increasingly important as the 3rd part service providers are now getting ready to provide more complex service-based applications. Accordingly it is critical to integrate the business policy with web service composition dynamically to adapt to changing business environments. Business policy needs to be represented explicitly, to be understood by semantics, and to be applied automatically. To support the business control in the interactive web service composition, this paper proposed a SWRL-based business policy model which does the rule reasoning based on semantics. And a business policy driven services recommend method was present to apply this model to the web service composition, which bridges the gap between business requirements and academic research. As a result, 3rd part service providers can focus on the business goals to be achieved, instead of having to create detailed control and data follows for the work at hand.

1 Introduction

Web services embody the paradigm of Service-Oriented Computing: applications from different providers are offered as web services that can be used, composed, and coordinated in a loosely coupled manner. One of the key challenges for contemporary enterprises is to generate complex services using by available web services on the Internet. The Web Service Composition (WSC) is really a business process which determines how the composition should be structured and scheduled. We believe business processes can be dynamically built by composing web services if they are constructed based on and governed by business rules.

Currently, one of the feasible WSC methods is workflow based schema, just like BPEL, which predefines the workflow of services. However, the predefined method lacks of flexible mechanisms to satisfy the user's personal requirements. The interactive method presented in [1][2] is a reasonable composition solution, which invites users to join the procedure to select the successive service. As semi-automated composition method, it has been argued in several literatures, in which the core algorithm is how to get the candidate services. According to the execution result of the last service (preS), the user could select the successive service from candidate services. When users select the successive services (sucS), they want to find the needed service quickly, rather than search in a great deal and unrelated services.

D. Georgakopoulos et al. (Eds.): ICSOC 2006 Ws, LNCS 4652, pp. 178–189, 2007.

Our work is motivated by the requirements of integration of business process with interactive WSC method. In the exist work of interactive WSC method, they consider the business process rarely, however, business process is critical in the service recommend process. Otherwise, the 3rd SP couldn't control the service logic, so the interactive MSC has less feasibility to be applied in reality. We have developed a semi-automatic WSC platform which utilizes service relation [3][4] to recommend services for users. In this paper, we proposed a solution to integrate business process to interactive WSC, which is achieved though using of business policy and rule inference.

There has been increasing work in designing business policy based WSC system [5]-[8]. However, there are some challenges in developing such a system. Firstly, business policies need be written in a language that both people and machine can easily understand, so business rules should be combined with Semantic Web [9]. Secondly, when conflicts arise in the context of using policies, it needs an efficient and appropriate manner to detect and resolve conflicts. Thirdly, business policy must be integrated to the interactive WSC method to control the business logic dynamically.

We analyze the business policy special for WSC and propose SWRL [10]-based Business Policy Model which includes of knowledge and rules. Based on SBPM, we present a Business Policy driven Services Recommend Method (BPSRM) to provide the candidate services which satisfy the business requirement. Our prototype and case analysis verify that BPSRM could integrate the business policy to WSC dynamically and seamlessly, which allows create personal services semi-automatically and bridges the gap between business requirements and academic research.

The remainder of this paper is organized as follows. Section 2 discusses related works. Section 3 reviews the business rules, introduce the SBPM. In section 4, we present the BPSRM. Section 5 briefly describes the prototype and shows the effect by case analysis. Section 6 provides some concluding remarks and outlines the future work.

2 Related Works

There are many ongoing research efforts in the business policy related technology. Some rule description language was used to express business rules: Defeasible Logic was used to describe the business rule in [6], which has strong expressive power and is executable. While the tool for maintaining rules and reasoning is absent, and it never argued the application to web service composition. RuleML based on SCLP was used to describe business rules in [7], and it has the characteristic that could be extended easily. However, it never refers to web service composition, and it could only reasoning with the condition, rather than control the service logic. Description Logic was applied in [8] to enhance current business integration approaches, and the semantic technology was applied in reasoning of non-function properties when service selecting. The business policy that used in the above works only refers to the service constraints, which is only a part of business rules. In this paper, we use the ontology and rule technology to modeling the business policy, which could express all business policies including service condition constraint and action enabler in focus and could be reasoned base on semantics. We argue the action of the business policy in the service logic control, rather than just in the constraint of selection in the services which have alike function.

In the research of integration business with WSC area, they dealt with business policy in different way. In the workflow method, a hybrid web service composition method [11] explicitly separates business rules from the process specification and adds the business rules to BPEL using the aspect-oriented programming method. PLM-flow [12] could create workflow automatically through the business rule which were defined by template including backward-chain and forward-chain. However in the above method, they never use the reason based on semantics. Based on semantic reason technology, we could use the object orientated method, the rule expression has semantics rather than just denotations. We use SWRL to modeling the business policy supporting semantics, and proposed the business policy expression more general than the above. So in the interactive WSC method, business policy could guide and govern the composition procedure.

3 Business Policy Model

In the following subsections, we present a SWRL – based business policy model (SBPM) which includes of knowledge and business rules. Business rules are defined as SWRL rules that are executed by a rule inferring engine.

3.1 Business Policy

Business rule encompasses a collection of terms (definitions), facts (connection between terms) and rules (computation, constraints and conditional logic) [4], which reflect the business policy. According to [4], the business policy is shown in Table 1.

Table 1. Business policy classification

Type	Rule	definition	Example
Supporting Business Rules	**Inference**	Tests conditions and upon finding them true, establishes the truth of a new face	R1: If customers are younger than 18, they are younger
	Computation	Checks a condition and when result is true, provides an algorithm to calculate the value of a term	R2:Today – birth date = age
Supporting Business Behavior	**Constraint**	Expresses an unconditional circumstance that must be true or false	R3: If customers are under 18, they cannot buy products for adults.
	Action Enabler	Checks conditions and upon finding them true initiates some action	R4: If no fight is found, book a train ticket.

Table 1 shows the two basic types of business rules. Supporting business tasks enable business tasks and processes implemented with web services. Supporting business rules such as computation and inference rules are not directly involved with the web service composition. They provide interpretation tools for operational business rules.

3.2 SBPM

Business policy needs to be represented explicitly, to be understood by semantics, and to be applied automatically. Contemporary literature on the combining of ontology and rule primarily addresses that the rule could be constructed on the ontology to extend the expressive power of ontology [13]. SWRL [9] combines the RuleML and OWL [14] to overcome many limitations in Description Logic and is considerably more powerful than either OWL DL or Horn rules alone.

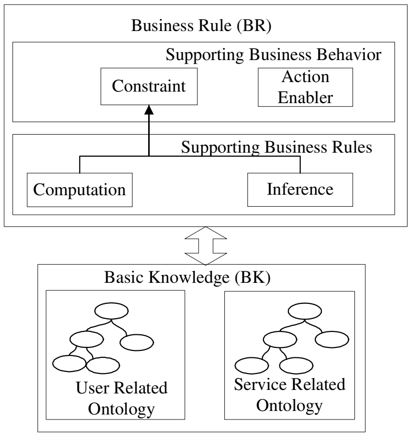

Fig. 1. SWRL-based Business Policy Model

According to the business rule defined by Business Rules Group and the classification in Table 1, a SWRL-based business policy model (SBPM) was shown in Fig.1. SBPM includes two parts: one is Supporting Business Rules - Basic Knowledge (BK) which defines the concept and relationships that should be used in rules; another is Supporting Business Behavior - Business Rules (BR) which is defined by SWRL. BR is supported by BK, and it extends the expressive capability of BR in essence. They construct the business policy in common and SBPM could be a repository of the reasoning system.

3.3 Business Knowledge

We have realized an interactive WSC system which mainly faces to information providing service, and we use the GIS related services and information services as examples in the following text.

Because there are two roles in business activity, which are the user and the available service, we model BK as two parts: one part is user related ontology which saves the information about user; another is service related ontology which saves the service and relationships among them. The user related ontology includes of knowledge that used to describe the user, which is shown in BK_1.

BK$_1$ A snippet of user related ontology

```
DatatypeProperty(gender)
DatatypeProperty(age)
DatatypeProperty(name)
Class(Person)
  intersectionOf(
  restriction (name allValuseFrom(xsd:string))
  restriction (gener allValuseFrom(xsd:string))
  restriction (age allValuseFrom(xsd:integer)))
  restriction (birthDate allValuseFrom(xsd:integer))
Class(Younger)(
  subClassOf(Person))
Class(User)
  intersectionOf(
    subClassOf( Person)
    restriction (ID allValuseFrom(xsd:string))
    restriction(phoneNumber allValuseFrom(xsd:string))
    restriction(email allValuseFrom(xsd:String)))
```

In the service related ontology, we define each type of web service as a class and a web service provided by a provider as an instance in the service related ontology, and organizes services by their topic what is achieved according to the service content. Therefore the service related ontology saves the relationships of service causing by their topics and the occurrence of service description file (in OWL-S). For further description, we have introduced this part in [3]. The example is shown in BK$_2$.

BK$_2$ A snippet of service related ontology

```
ObjectProperty(hasService)
Class(Spot
  intersectionOf(
    restriction (hasService queryAddress)
    restriction (hasService queryPhoneNo)
    restriction (hasService showAroundMap)
    restriction (hasService sendMessage)
    restriction (hasService locate)))
Class(Entertainment
  intersectionOf(
    subClassOf(Spot)
    restriction (hasService queryAverage)))
Class(Cinema
  intersectionOf(
    subClassOf(Entertainment)
    restriction (hasService queryFilmInfo)
    restriction (hasService buyFilmTicket)
    restriction (hasService queryNearCinema)))
Class(sendMessage)
Class(sendMMS
  intersectionOf(
    subClassOf(sendMessage)))
Class(sendSMS
  intersectionOf(
    subClassOf(sendMessage)))
```

3.4 Business Rule

Business rules are usually expressed in the form *if conditions then action* which accords with the syntax of SWRL. The business rule templates use the SWRL expression which is

antecedent => consequent, where *antecedent* and *consequent* = $a_1\wedge\ldots\wedge a_n$ where a_i can be of the form C(x), P(x,y), or swrlb:buildin where C is an OWL description, P is an OWL property, and swrlb:buildin is a SWRL built-ins [11] which support the operation including Comparing, Boolean values, Strings, Date, Time, and et al.

For services invocation and constraints, we defiant five properties to express the relation between user and services:

i) *success*(User, Service). A user has invoked this service successfully, which means that the service has no exception when invoked.
ii) *failed*(User, Service). A user has invoked this service but it returned abnormally, maybe exist some exceptions.
iii) *enable*(User, Service). It defines the constraints that the service could be invoked by the user in the current state.
iv) *disable*(User, Service). It defines the constraints that the service couldn't be invoked by the user in the current state.
v) *do*(User, Service). Service should be executed in one step.

The detail templates of different rule types are shown in Table 2. According to the examples of Table 1, the rules are expressed in the following.

Table 2. Business Rules Templates

	a_i in antecedent	a_i in consequent
Inference	C(x)|P(x, y)|swrlb:buildin	C(x)|P(x, y)|swrlb:buildin
Computation	Empty	C(x)|swrlb:buildin
Constraints	C(x)|P(x, y)|swrlb:buildin	enable(x,y)|disable(x,y)|do(x,y)
Action enabler	success(x, y)|failed(x,y)	enable(x,y)|disable(x,y)|do(x,y)

BR A snippet of BR

```
R1:User(?x)∧Age(?x,?y)∧swrlb:smallerThan(?y,18) ⇒ Younger(?x)
R2:User(?x)∧Today(?y)∧birthDate(?x,?a)∧age(?x,?b) ⇒
swrlb:substract(?b,?y,?a)
R3:User(?x)∧Adult(?x) ⇒ BuyAdultCommodity(?a)∧disable(?x,?a)
R4:User(?x)∧BuyAirTicket(?y)∧failed(?x,?y) ⇒ BuyTrainTicket(?a)
∧do(?x, ?a)
```

4 A Business Policy Driven Web Service Composition

In this section, we give details on how to realize the dynamical WSC and execution governed by SBPM.

4.1 WSC Process

Business policy driven WSC involved five major steps (see Fig.2): business policy repository building, facts creation, rule inference, recommend candidate service and user invocation.

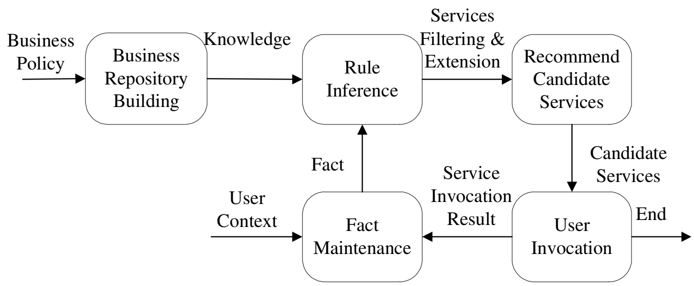

Fig. 2. Business Driven Interactive WSC Process

Firstly, the 3rd part SP input the business policy to build the business policy repository. Secondly, the WSC process begins with a user's request and the user context is the fact of reasoning. Thirdly, the inference engine could do the reasoning with business policy repository and return the results about the business constraints. Fourthly, according to the results, system gets the candidate services and provides them to users. Lastly, the user invokes the service and the execution result is maintained to update the fact. In the following text, we will discuss the inference face and conflict handling in detail.

4.2 Facts and Inference

Context-sensitive technology has been introduced to web service composition [15] in which the user context was defined as user profile and user location. In our paper, we think that the user profile and the services list that invoked by the user in the current session are critical for the successive service. Therefore, we extend the context with the services list that invoked by the user. The user context constructs the reasoning fact, which is defined as follows.

User Context is a 4-tuple <userID, preSList, PreS, F> where userID is the identifier of a user; preSList is a set of services S invoked by the user, S is a 2-tuple <Sname, State> where Sname is the name of service, and State is the result of service Sname, which will be "success" or "fail"; PreS is the service S that was invoked in the last choice; Fact is the assertion that come from preSList and preS.

SWRLJessTab [16] is a Protégé (since v3.2beta) [17] plugin intend to bridge between Protege OWL, RACER and Jess, for reasoning with SWRL. The reasoning results are consisted of *do*, *enable*, *disable* services list.

4.3 Conflict Handling

SWRL has been proposed as a standard in W3C, and it has been extensively studied, has clear semantics, and is supported by automated reasoning techniques. But it falls short as an appropriate basis for our purposes in business policy on inability to deal with rule conflict that most rule languages have. For the rule's conflicts checking, the primary method is to check the conflict manually, and define the priority of rules to deal with the conflict. Whereas SWRL doesn't support priority any, so the priority of rules is infeasible in our model.

Rather than define the priority of rules, we proposed a simple method that define the priority of properties. The priority of properties of service is:

P(*disable*) > P(*do*) >P(*enable*) where P stands for priority.

If the reasoning result has conflict such as *disable*(user, buyTicket) and *enable*(user, buyTicket), we hold the predication *disable*(user, buyTicket) that has the higher priority.

4.4 Service Recommend Algorithm

IO-matching is a relative mature approach, in which the successive service (sucService)'s input is could feed to the previous service (preS) as an input. The core idea of the DSAC is that it uses the IO-matching method as a basic step, then extends and revises it by the result of service relation reasoning. BPSCM add the business policy reasoning to DSAC to make the service logic be governed by business policy. The concrete approach is listed as follows:

i) By using the business inference and conflict handling, we could get three lists – *doList*, *enableList*, and *disableList*.
ii) After DSAC method, we could get the candidate service, which are saved in *doList*.
iii) Remove the services in *disabledList*, add the service in *enableList*, and trigger the services in *doList* automatically.

5 The Prototype and Case Analysis

Integrated Intelligent Service Platform (IISP) [4] realized an interactive web service composition method – DSAC [3]. IISP used a call center agent to assistant the user to complete the service composition procedure. The system recommends candidate services to users by a call center agent, and after the user makes a choice, the agent invokes the selected service and gives back the results. So the user just dial the customer service center, an agent will service you. The basic services of our system are: GIS related services, telecom services and integrated information services.

5.1 Implementation of the Prototype

The working interface of agent is shown in Fig. 3, we could see the candidate services that accord with the business policy are list in the right fuscous frame, and the agent could help the user choose a service according to the user's request or make clues to the user. The user's request could be record by agents, and the inputs and outputs are processed by agents. For space reason, the service execution flow is not detailed here

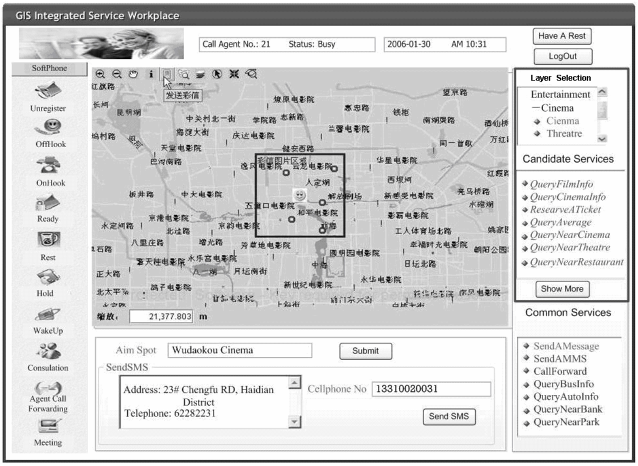

Fig. 3. Interface of IISP to agents

5.2 Case Analysis

In this scenario, we analyze a case to verify that the BPSCM could govern and guide the process of composite service.

Case: suppose that in the business policy repository, the basic knowledge is consisted of BK_1 and BK_2, and the rules include the examples in Table 1. We would like to add the business rules shown as follows to express our business policy.

R5: if the user query the address of some spot, then system send a map around the spot to the user by MMS.
R6: if the user is a younger, he (she) couldn't by the ticket of cinema.
R7: if the user hasn't subscribed the mobile service, he (she) couldn't use the mobile related service.
R8 : If the user hasn't subscribed the mobile service, the system recommend the user to register for the mobile service.
Therefore, we could add the rules to business rule file (BRS) like this:

```
R5:User(?x)∧queryAddress(?y)∧sucess(?x,?y) ⇒
showAround(?a)∧sendAMMS(?b)∧do(?x,?a)∧do(?x,?b)
R6:User(?x)∧Younger(?x)⇒byFilmTicket(?a)
∧disable(?x, ?a)
R7:User(?x)∧haveMsgAuthorization(?x,?y)∧swrlb:equal(?y,
"no")⇒sendMessage(?a)∧disable(?x,?a)
R8:User(?x)∧haveMsgAuthorization ?x,?y)∧swrlb:equal(?y,
"no")⇒applyMsgAuthorization(?a)∧enable(?x,?a)
```

A scenario is the following: When a user named moon calls the call center, the agent accepts the user's request and serves him. The agent tells the user what services he could invoke. Firstly, the user asks that "I want to query the address of XiTian

Cinema", then agent invokes the service *queryAddress,* and input the parameter "XiTian Cinema". Once the service *queryAddress* is invokded, we analysis the process in detail.

I) Fact is shown as follows:

```
Individual(Today)
 Value(year "2006")
 Individual(User )
 Value(ID "moon")
 Value(birthday "1989")
 Value(phoneNo "13800000000")
 haveMessage("no")
 success("moon", "queryAddress")
```

II) After reasoning with the business rule, the new assertions are the following:

```
Individual(User, typeOf Person, User, Younger)//From R1
  value(age "17")                          //From R2
  disable("moon", "buyFilmTicket")     //From R1,R6
  do("moon", "showAround")           //From R5
  do("moon", "sendAMMS")            //From R5
  disable("moon", "sendMessage")   //From R7
  enable("moon", "applayMsgAuthorization") //From R8
```

In the BK2 there are

sendAMMS $\subseteq$ sendMessage and sendASMS $\subseteq$ sendMessage,

therefore we remove the assertion *disable*("moon", "sendMessage") and add two assertions *disable*("moon", "sendAMMS") and *disable*("moon", "sendASMS").

For service *sendAMMS,* there's some conflict because it appears in both *disableList* and *doList*. The rank of *disableList* is higher than *doList*, thus we remove the *do*("moon","sendAMMS").

III) The comparison of DSAC and BPSCM including automate invocation and candidate services are shown in Table 3.

Table 3. Comparison of DSAC and BPSCM

	DSAC	BPSCM
Automate Invocation	null	*showAroundMap*
Candidate Services	*queryNearestRestaurant* *queryNearestCinema* *queryTransferInfo* *buyfilemTicket* *sendASMS*	*queryNearestRestaurant* *queryNearestCinema* *queryTransferInfo* *applyMsgAuthorization*

From the above case, we could draw a conclusion that: Firstly, SBPM has enough expressive power to represent both business rule and knowledge that needed by the business policies. Secondly, the reasoning power of SBPM could be integrated with the web service composition process seamlessly and the service logic could adapt the variety of policy flexibly by BPSRM. Thirdly, SBPM is a general model to get the constraints and restrictions coming from business policy, so it could be applied in other web service compositions method preferably.

6 Conclusion

It is clear that current business policy driven web service composition are not capable of dealing with the complex and entirely business policy and never refer to how to use the business policy to guide the interactive services composition.

In this paper, we have presented a SWRL-based business policy model SBPM which has strong expressive power to represent the business policy, and its reasoning result could be integrated with the web service composition process dynamically. The inferring result which is consisted of constraints and extension of services is combined with the web service composition flexibly. A prototype was implemented to verify our idea, and the case analysis illustrated that BPSRM could get a set of effect candidate services which satisfy business policy for user. As a result, end user can focus their business goals to be achieved, and the interactive WSC could be governed by business goals. Our future work includes researching the performance of SBPM inferring and semi-automatically constructing the ontology and extracting business rules from documents.

Acknowledgments. The National Natural Science Foundation of China under grant No. 60432010 and National Basic Research Priorities Program (973) under grant No. 2007CB307100 support this work. We gratefully acknowledge invaluable feedbacks from related research communities.

References

1. Sirin, E., Parsia, B., Hendler, J.: Filtering and Selecting Semantic Web Services with Interactive Composition Techniques. Intelligent Systems 19(4), 42–49 (2004)
2. Arpinar, I.B., Zhang, R., Boanerges, A., et al.: Ontology-driven Web service Composition Platform. In: Proceedings of the IEEE International Conference on E-Commerce Technology, pp. 146–152 (2004)
3. Xu, M., Meng, X.W., Chen, J.L., Mei, X.: The Research of Load Balancing for Integrated Service Platform. Journal of Beijing University of Posts and Telecommunications, 94–97 (20065A) (in Chinese)
4. Xu, M., Chen, J., Peng, Y., Mei, X.: A Semantic Association-Based service creation method. In: the Proceedings of Web Intelligence Conference, IEEE/WIE/ACM, HongKong, pp. 666–669 (2006)
5. von Halle, B.: Business Rules Applied: Building Better Systems using the Business Rules Approach. Wiley, Chichester (2001)
6. Antoniou, G., Arief, M.: Executable Declarative Business Rules and Their Use in Electronic Commerce. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 6–10. Springer, Heidelberg (2003)

7. Grosof, B.N.: Representing E-Commerce Rules via Situated Courteous Logic Programs in RuleML. In: WITS '01. Proceedings of the 11th Workshop on Information Technologies and Systems (2001)
8. Trastour, D., Preist, C., Coleman, D.: Using Semantic Web Technology to Enhance Current Business-to-Business Integration Approaches. In: Proceedings of the Seventh IEEE International Enterprise Distributed Object Computing Conference, pp. 222–230. IEEE Computer Society Press, Los Alamitos (2003)
9. Spreeuwenbergn, S., Gerrits, R.: Business Rules in the Semantic Web, are there any or are they different? In: Proceedings of 2nd European Semantic Web Conference (2005)
10. Horrocks, I., Patel, P.F.: A Proposal for an OWL Rules Language. In: Proceedings of WWW2004, New York, pp. 723–731 (2004)
11. Anis, C., Mira, M.: Hybrid Web Service Composition: Business Processes Meet Business Rules. In: ICSOC'04. 2ed International Conference on Service Oriented Computing, pp. 30–38. ACM Press, New York (2004)
12. Zeng, L., Flaxer, D., Chang, H., Jeng, J.-J.: PLMflow–Dynamic Business Process Composition and Execution by Rule Inference. In: Buchmann, A.P., Casati, F., Fiege, L., Hsu, M.-C., Shan, M.-C. (eds.) TES 2002. LNCS, vol. 2444, pp. 144–150. Springer, Heidelberg (2002)
13. Antoniou, G., Damasio, C.V., Grosof, B., et al.: Technical report: Combining Rules and Ontologies-A survey, http://rewerse.net/deliverables/m12/i3-d3.pdf
14. Antoniou, G., van Harmelen, F.: Web Ontology Language: OWL. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies in Information Systems, pp. 67–92. Springer, Heidelberg (2003)
15. Maamar, Z., Mostefaoui, S.K., Mahmoud, Q.H.: Context for Personalized Web Services. In: Proceedings of the 38th Hawaii International Conference on System Sciences, pp. 89–98 (2005)
16. Martin, O., Holger, K., Samson, T., et al.: Supporting Rule System Interoperability on the Semantic Web with SWRL. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 974–986. Springer, Heidelberg (2005)
17. Protégé: http://protege.stanford.edu/