

# Web ontology language - Business Process Customization Based Dynamic Customization for Composite Web Service

*Malathi.T<sup>1</sup>, Kalpana.K<sup>2</sup>*

<sup>1</sup>Gnanamani College of Technology, Department of Computer Science and Engineering,  
Namakkal 637018, India  
*malathits@gmail.com*

<sup>2</sup>Gnanamani College of Technology, Department of Computer Science and Engineering,  
Namakkal 637018, India  
*kalpanagctse@gmail.com*

**Abstract.** We propose to develop a trust system based on processing the payment reports to maintain a trust value for each node. The nodes that relay message more successfully will have higher trust values, such as the low-mobility and the large-hardware-resources nodes. Based on these trust value, we will propose a trust-based routing protocol to route message through the highly trusted node (which performance packet relay more successfully in the past) to minimize the probability of dropping the messages, and thus improve the network performance in terms of throughput and packet delivery ratio. The nodes submit lightweight payment report to the AC to update their credit accounts, and temporarily store undeniable security token called Evidences. The reports contain the alleged charges and rewards of different sessions with security proofs. The AC verifies the payment by investigating the consistency of the reports, and clears the payment of the fair reports with almost number of cryptographic operation. For cheating reports, the evidences are requested to identify and evict the cheating nodes, that submit correct reports. Evidence aggregation technique is used to increase the storage area of the Evidences. Evidences are submitted and the AC applies cryptographic operations to verify them only in case of cheating, but the nodes always submit security tokens. However, the trust system should be secure against singular and collusive attacks, and the routing protocol should make smart decisions regarding node selection with low overhead.

**Keywords:** Web Ontology Language, Human Semantic Web, Secondary Business Process, Resource Description Framework Business Process Execution Language, Business Process Customization

## 1. INTRODUCTION

Semantic web should support a shift of social interaction patterns from a producer-centric paradigm to a consumer-centric one. We present a representation of this conceptualization in a new Extensible Markup Language (XML) markup language, based on the semantic markup language for Web-based information, i.e., OWL. We name the conceptualization OWL-BPC for OWL on Business Process Customization. OWL-BPC has been explained to conceptualize the problem of business process customization. The design process must be customer-centered. A branch of research efforts

on semantic Web seeks to integrate a machine-understandable knowledge framework with the user-centric human factors, so called "Human Semantic Web". We focus on the Business scenarios where the business processes can be supported dynamically.

In consumer-centric business modeling, an important task is to develop semantic-based frameworks that make a business process easier for consumers to do business with. This will demand a measure of business process customization. Automating this task has been

made easier by service-oriented architecture. In a service-based business process, each activity in the process is treated as a message exchange with an operation supported by some Web service. The process itself can then be described as a composition of Web services using a standardized language such as the Business Process Execution Language (BPEL) or Web Ontology Language for Web Services (OWL-S). A service-based business process by nature allows more agility in the process due to loose coupling, service reuse, and dynamic binding.

In a service-based business process, customization may be enabled by automatically adapting the process to match the business partner's practice indicated by their business processes. Such practice includes service interface specifications, Web Ontology Language (OWL) service profiles, process models and grounding.

An OWL-BPC based static customization does not support runtime customization, therefore dynamic customization framework need to be designed to provide sophisticated service to the consumers to handle runtime exceptions.

## 2. RELATED WORK

The interaction between businesses models used in consumer centric manner instead of using a producer centric approach for customizing the business process in cloud environment. Collaboration of SBP with PBP is explained.

The business process can be considered as a composition of services, which is usually prepared by domain experts, and many tasks still have to be performed manually. The design and creation of the process itself or the modification of an existing one when business requirements change. It explains about process merging.

Making software systems service-oriented is becoming the practice, and an increasingly large number of service systems play important roles in today's

business and industry. Currently, not enough attention has been paid to the issue of optimization of service systems. In this paper, we argue that the key elements to be considered in optimizing service systems are robustness, system orientation, and being dynamic and transparent. We present our solution to optimizing service systems based on application-level QoS management

Web Ontology Language. The Ontology Web Language (OWL) is a set of mark-up languages which are designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL ontologies describe the hierarchical organization of ideas in a domain, in a way that can be parsed and understood by software. OWL has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web.

The advent of Semantic Web and its relevant technologies, tools and applications provide a new context for exploitation. The "expression of meaning" relates directly to numerous open issues in e-learning. In this special issue the focus is two-fold: On the one hand to stress the importance of applying Semantic Web techniques towards constructing systems that provide value to learners, and - on the other hand - to reveal research opportunities that can initiate interesting projects over the forthcoming years. In the W3C Semantic Web activity a list of priorities has set the challenging landscape for the realization of the next generation web: The creation of a Policy Aware Infrastructure, the Ontological Evolution, the promotion of a Web of Trust, and the facilitation of Information Flow and Collaborative Life.

The Semantic Web is the second generation of the Web, which helps sharing and reusing data across application, enterprise, and community boundaries. Ontology defines a set of representational primitives with which a domain of knowledge is modeled. The main

purpose of the Semantic Web and ontology is to integrate heterogeneous data and enable interoperability among disparate systems. Ontology has been used to model software engineering knowledge by denoting the artifacts that are designed or produced during the engineering process. The Semantic Web allows publishing reusable software engineering knowledge resources and providing services for searching and querying.

### 3. OWL-BPC ONTOLOGY

#### 3.1 AUTOMATIC CUSTOMIZATION ENACTMENT

*Events:* Events in ECA are of our interest if they have happened or will happen. In the context of the problem of business process customization, events are always related to the customizable contents of the business process. For example, the collection of records of the customizable content output can be directly converted to the events that occur. Events will be passed to the event server for processing. If the event matches the event part of a rule, the rule will be fired. Events may be designed to carry additional information, referred to as parameters, which the rule may use in its execution. This is especially important in our case. For example, customizing content is required to be taken at the same time that content is customized. Function Time defined in the Customization Function ontology needs to be passed to the rule so that the timing of customization is guaranteed.

*Condition:* Each rule has a left-hand side (LHS) and a right-hand side (RHS). The LHS of the rule contains patterns of events, and the RHS represents the actions that will be taken once the event that occurred matches any pattern in the LHS of the same rule. A rule is assigned a name and is designed to take in parameter values. An ECA also provides an option to conditionally specify an action. When the rule is invoked upon the occurrence of some event, the condition will be evaluated. True evaluations lead to the execution of the RHS. Otherwise, no action will be taken.

*Action:* Action will be performed only if the event and condition performed will be equal to the action. Only then the action will take place. The above table shows the Rules and their Action.

#### 3.2 AUTOMATIC CUSTOMIZATION DETECTION

Automatic Customization detection is an automated process of detecting possible elements of a business process that need to be especially treated in order to suit the requirement of other process. Primary Business process or PBP is the business processes that need to be customized. The PBP is designed so that while executing it the web services are invoked for the activities of the SBP Secondary Business Process. Based on the aforementioned assumption, we rely on a static analysis on the goals of interacting partners in an interactive collaboration in order to suggest possibly appearing events pertinent to the business process. The approach taken is to compare goals of the partners semantically by traversing the ontology graph. The goal analysis for scoping customizable contents can be briefly described as an algorithm listed in Algorithm.

1: ScopingCustomizableContents

$((G_0, O_0, bp_0), (G_1, O_1, bp_1), \dots, (G_m, O_m, bpm), \dots$

$(GM, OM, bpM))$

$bp_0$  The PBP, which is to be customized.

$G_0$  The goal taxonomy of  $bp_0$ .

$O_0$  The domain ontology of  $bp_0$ .

$bpm(m! = 0)$  The  $m$ th SBP that  $bp_0$  collaborates with.

$Gm(m! = 0)$  The goal taxonomy of  $bpm$ .

$Om(m! = 0)$  The domain taxonomy of  $bpm$ .

$M + 1$  is the total number of processes in consideration.

2:  $p = 0$ ;

3: **L0:** For  $bpp$  and each business process  $bp$  in the set of

business processes  $\{bpm\}$ , which  $bpp$  collaborates with,

{

4:  $i = j = k = 0$ ;

5: Identify the goal of the business process in the goal

taxonomy  $G_m$ ;

6: Assign the goal to  $g$ ;

7: **L1**: Reduce  $g$  to a set of subgoals  $\{sg_i\}$  by reductions defined in  $G_m$ ;

8: for each subgoal  $sg$  in  $\{sg_i\}$ {

9: if  $sg$  is not implementable {

10:  $g = sg$ ;

11: Go to L1;}

12: else

13: for each objective  $o$  returned by  $obj(sg)$ , which implements {

14: /\* $obj(sg)$ , defined in  $G_m$ , returns a list of operational objectives that implement  $sg$ \*/

15: Append  $o$  to  $\{oj\}$ ;} }

16: for each objective  $o$  in  $\{oj\}$ {

17: for each action  $a$  returned by  $esb(o)$ , which establishes

18: /\* $esb(o)$ , defined in  $G_m$ , returns a list of actions that establish  $o$ \*/

19: Append  $a$  to  $\{ak\}$ ;} }

20: Run OnExCat on the actions in  $\{ak\}$ ;

21: Infer on  $\{ak\}$  to identify actions dependent on each other;

22: Record a set of tuples made of actions  $\Gamma_p = \{t | t = (a_0k_0, a_1k_0, \dots, a_qk_i, \dots)\}$ , where  $a_qk_i$  is an action in the collaborating processes that is relevant to action  $a_{ki}$  of  $bp_p$  ( $q$  enumerates through the actions that are relevant to a particular action  $a_{ki}$ ).

23: if  $p < M$  {

24:  $p = p + 1$

25: Go to L0;}

26: Return  $\{\Gamma_m\}$

#### 4. PERFORMANCE METRICS

*Ontology Matching rate*: It is the performance metric used in the project. Automatic matching is expected to give a good matching rate. However, in the case that automatic matching results are erroneous, we may involve an extra step of human screening. Such involvement provides a guarantee that incorrect

matching results will not be propagated to customization detection and enactment.

*Confidence Level*: One way to reduce such extra human screening effort is to only check on the cases with a confidence level lower than some threshold. The performance can be measured using the following

*F-Measure*: Thus F-measure is an evaluation Techniques used here.

F-measure is formally defined as:

$$F\text{-Measure} = \frac{N}{\sum (1/x_i)}$$

N- Denotes no of Web services.

$X_i$ - denotes  $n(n-1)$  number of times the services is checked

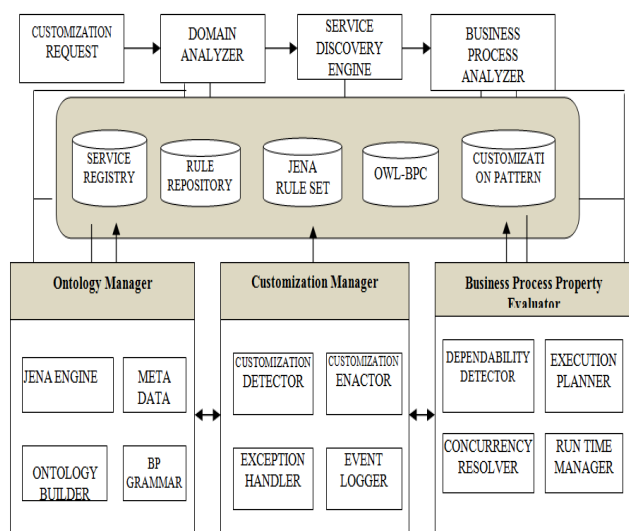
Thus the customization time decreases if the same query is retrieved again and again. This kind of customization done at runtime is called dynamic customization.

$$f(x, y) = (x^2 - 10\cos 2\pi x) + (y^2 - 10\cos 2\pi y) + C$$

X, Y denotes the position of each WSDL file for a given Query. It represents a position a lattice. C denotes a constant which can be 10 or 20 according to the number of loops.

#### 5. SYSTEM MODELS

The proposed architecture has been designed as shown in Fig 5.1. The input is Customization request given by the user for customizing the services. It is analyzed by the Domain Analyzer



**Figure 5.1 Ontology based Business Process Customization Model**

Dynamic customization has to be done in order to give a sophisticated service to the customers. Any process done during or after the instantiation time is called dynamic customization. To do this we have designed an Architecture framework for the dynamic customization where it can handle all the runtime exceptions.

## CONCLUSION

Thus the System was able to perform dynamic customization. Thus customization done during or after the runtime should be supported. Thus the system designed will perform the customization at domain level and process level. Runtime mapping of the services is possible. As the query is given runtime mapping changes according to the given query. This kind of customization done at runtime is called dynamic customization. Runtime mapping of a service is done. Thus for a given query clustering of service is done at runtime. Further enhancement about the project is that dynamic customization is done at domain level and process level. Thus here for a given query the related services will be clustered. Let us give query transportation, thus the needed services is clustered. Thus as need arises services has to be added, thus the services are added to the corresponding cluster. Thus runtime addition of service should be possible. The

novelty of the work done is that clustering has to be done to cluster the service to a particular domain. The similarity of service or the service that matches to the cluster has to be grouped. The most similar service has to be clustered to the so matched Domain. New web service added has to be matched with the most similar domain so forming a cluster. Thus the algorithm above explains about the domain which explains about the service level customization. Thus new services need to be added. Prs denotes the process. Go represents the Goal Ontology. Thus in this algorithm new service added will have a check to the particular cluster say domain.

## References

- [1] Qianhui Liang, Xindong Wu, E.K. Park, TaghiM.Khoshgoftaar and Chi-Hung Chi, "Ontology-Based Business Process Customization for Composite Web Services," IEEE Transaction., vol.41, no.4, pp.717-728, July 2011.
- [2] KameshKumar, P., " Harmonizing the business process customization using ontology," Advances in Engineering, Science and Management (ICAESM), 2012 International Conference., pg 788 – 791, 30- 31 March 2012.
- [3] Bulanov, P,"Business process customization using process merging techniques", Service-Oriented Computing and Applications (SOCA), 2011 IEEE International Conference ., pp.1 – 4. 12-14 Dec. 2011.
- [4] D. König, N. Lohmann, S. Moser, C. Stahl, and K. Wolf, "Extending the compatibility notion for abstract WS-BPEL processes," in Proc. 17th International Conference of World Wide Web, 2008, pp. 785–794.
- [5] Q. Liang, X. Wu, and H. Lau, "Optimizing service systems based on application-level QoS," IEEE Trans. Serv. Comput., vol. 2, no. 2, pp. 108–121, Apr.–Jun. 2009.
- [6] A. Naeve, "The human semantic Web: Shifting from knowledge push to knowledge pull," J. Semantic Web Inf. Syst. (IJSWIS), vol. 1, no. 3,pp. 1–30, 2005.
- [7] A. Naeve, M. Lytras, W. Nejdl, J. Harding, and N. Balacheff, "Advances of semantic Web for e-learning: Expanding learning frontiers," Brit. J.Educ. Technol., vol. 37, no. 3, pp. 321–330, 2006.
- [8] D. He and X. Wu, "Ontology-based feature weighting for biomedical literature classification," In the Proceedings of IEEE International Conference. Inf. Reuse Integr., 2006,pp. 280–285.

[9] Q. A. Liang and H. Lam, "Web service matching by ontology instance categorization," in Proc. IEEE Int. Conf. Serv. Comput., 2008, pp. 202–209.

#### **AUTHOR PROFILE**

**Malathi.T** received the M.Sc. degree in Software science from Vivekanandha college of Arts and science, 2005, M.Phil from Pride 2007. She is pursuing towards the M.E degree in Computer Science and Engineering from Gnanamani College of Technology, Affiliated to Anna University, Chennai since September 2012. Her research area is Computer Networks & Web services.

**Kalpana.K** received the M.E degree in Computer Science and Engineering from St Peter's university in 2012, Now working as Assistant Professor in Gnanamani College of Technology. Her research area includes Computer Networks and Database Management Systems.