DYNAMIC CUSTOMIZATION IN THE BUSINESS PROCESS SERVICE COMPOSITION USING ONTOLOGY

C. Santhosh kumar
PG Scholar, Dept of CSE, Anna university of Technology,
Coimbatore, cjksag@gmail.com.

D.Palanikkumar
Asst.professor , Dept of CSE, Anna university of Technology,
Coimbatore, palanikkumard@gmail.com

ABSTRACT

Making the customers most preferable and building consumer centric structured models lead to the business process profitable. This paper mainly focuses on increasing customization features of service based business processes with the knowledge of existing web services. We present customization by a new Web Ontology Language (OWL) based on the semantic markup language for web based information. The ontology is used for composition of web services. The dynamicity behavior in the web data is proposed for increasing web performance during any changes in web data. The system can get the user needs and properly utilize source interfaces. The results show that the proposed method is more automatic and dynamic than the existing composite web service mechanisms.

Keywords-Web ontology language, Dynamicity behavior, Composition of web service

I. INTRODUCTION

Nowadays web services play a major role in the development of business process. The consumer will get information in a compact and knowledgeable manner. The new human semantic web [1] gives the new way to the current web services. It gives meaning to the web information which leads to the effective information retrieval and extraction process. It will be attained by using Service Oriented Architecture (SOA). In a service-based business process, each activity in the process is treated as a message exchange and it has an advantage of making business process more robust, reusable and dynamic. The individual web service works as a separate business process. In this paper, we focus on customizing the business processes where the processes met a certain degree of flexibility and adaptability in selecting business partners [2][3][16] and adjusting the process parameters and information during a Network problem. Composing the web services of different business process is a method of integrating one or more single services which will increase the business performance and will help the customer in a new direction [5].
Generally composition is done by using Choreography or Orchestrian methods in the BPEL. Here ontology is used for composing web services. The flexibility and adaptability between the web services i.e., Service reuse and the data retrieving capacity during data loss is solved by using automatic functioning and retrieving mechanisms. In this paper, Ontology Versioning Mechanisms pays the way to reach the Dynamicity approach. Such Dynamic behavior will enhance the web performance which leads to the business profit. The Customization in the business process is done according to the following ones. First, conceptualization definition for business process i.e., we have developed a vocabulary for modeling the meanings of concepts and the relationships between those concepts. Second, we represent the conceptualization in an Extensible Markup Language (XML), based on the semantic markup language for Web-based information, i.e., OWL, Resource Description Framework (RDF). Third, after the representation, we choose the required service processes and compose the different web services by using Ontology Mapping process (customization detection). Fourth, the possible causes of discrepancies, inconsistencies between business processes are identified and a suitable remedial action (customization enactment) is taken. Finally for the automatic dynamic functioning ontology versioning mechanisms is followed.

II. RELATED WORK

The new features of the web service will emerge day by day due to many research works. A. Naeve et al. [1] introduces the Human Semantic Web (HSW) as a conceptual interface, providing human understandable semantics in addition to the ordinary (machine) Semantic Web. The service in the business process mainly focuses on consumers. So the user centered consumer based model is to be developed which will make the business process more interactive to the consumers by making the correct selection of domain, interface, and users [2]. H. Juarez-Espinosa et al [3] describes how the visualization component helps in improving data analysis in the user centered Environment by using models. P. C. Xiong [5] presents the formal optimization techniques to meet the high quality of services. [7] Ignazio Palmisano et al. proposes the construction and handling of ontology for the web services by using algorithms, ontology recovery and reconstruction mechanisms.

III. CONCEPTUALIZATION OF SERVICE BASED BUSINESS PROCESS

Business process needs to communicate with one another by calling its web services and exchanging xml messages with it. In order to meet the need, the service ensures smooth collaboration between the participating partner services. To allow the customization of process, the following things required are: 1) Establishment of conceptualization of business process together with the associated markup language (conceptualization definition); 2) construction of ontology for web information; 3) the instantiation of ontology in describing a specific instance of customization. Composition of services is by using ontology. In this paper the customization features such as discrepancies and inconsistencies has to be found (customization detection) and suitable remedial actions has to be conducted by the ontology developer (customization enactment). The automatic functioning mechanism helps in service reuse, retrieving the lost data during the data loss.

IV. CONCEPTUALIZATION DEFINITION

The vocabulary for the subject domain is collected and analyzed. The information in the web service is representing by using some formal representation techniques. The languages here used are XML (Extended Markup Language), OWL-BPC (ontology web language for Business Process Customization), and Resource Description Framework (RDF).
V. ARCHITECTURAL DIAGRAM

![Architectural Diagram]

VI. CUSTOMIZATION DETECTION

Customization Detection is a process of detecting possible data for the business process in order to go with the requirement of other processes. Here the web services are distinguished as a separate business process which was according to the different domains used. The language representation of different web services was discussed during the detection. The change of information, representations, flow activities between the business processes, investigating concurrent execution of services are according to the developer perspective. Here developer uses his knowledge in finding relevant data of the services and methods. In this paper the services of different domain such as Mobile, Printer, and Fax are considered. The following figure shows Mobile service as W1, Printer service as W2, Fax service as W3.

![Web Services of Different Domains]

VII. ONTOLOGY CONSTRUCTION

A. Ontology

Ontology is the formal specification of domain and their relationship. It shares, annotates information in the field and also defines a common vocabulary for researchers who need to share data in a business domain. The ontology is developed to share common understanding of structure of information among people, enable reuse of information, separate domain knowledge from operational knowledge and analyze domain knowledge.

B. Ontology creation

Ontology is created for every web services. The ontology O1 for W1, O2 for W2, and O3 for W3 is to be constructed and is shown in the Figure. 3. The ontology is constructed by using a software framework Protégé 4.1. While creating ontology the data behavior, object behavior, class, individuals are analyzed.
by the ontology developer. In this paper the ontology output is in the form of OWL file which is shown in the Figure. 6.

![Fig. 3 Web services and their corresponding ontology](image)

![Fig. 4 Giving Annotations in the Creation of ontology in Knowledge framework Protégé](image)

![Fig. 5 Indicating Data behavior and Object behavior in the protégé framework](image)

The above Figure 4 & 5 shows the annotations, data and Object behavior given in the Ontology.
VIII. WEB SERVICE COMPOSITION

The Fig.7. Shows that the services WS1, WS2, WS3 will be integrated to form a WS_n i.e., WS_n = WS1+WS2+WS3.

The composition is because of increasing the quality of service and achieving profit in the business process. Generally in the BPEL (Business Process Execution Language) the orchestrian, choreography methods are used for web service composition [9]. In this paper the ontology i.e., OWL is used for composing web services. For composing services the ontology mapping is to be done. Ontology mapping is the method of mapping the individual ontology of different domain services into one.

The above Figure. 8. explains the final domain ontology i.e., O1+O2+O3=O_n. The mapping process does the overlapping i.e., merging the different service ontology into single domain ontology [15][17][18]. This can be carried out by ontology developer. Here the prompt algorithm is used for mapping the ontology i.e., creating a single coherent ontology that includes the information from all sources.

Figure. 9. Shows the definitions for ontology mapping, alignment, merging.
The prompt ontology merging algorithm performs a semi-automatic approach in merging the ontology. It is an extension to protégé-2000. Prompt is such a plug-in for merging two source ontologies. Some features preferred to users are 1) setting the preferred ontology 2) maintaining the user focus 3) providing feedback to the users 4) logging and reapplying the operations. In prompt, we start with the linguistic-similarity matches for the initial comparison, matching classes but concentrate on finding clues based on the structure of ontology and user actions. The prompt algorithm,

In the protégé 4.1 based merging process the two ontologies are merging using a tab. The below figure shows,
IX. ONTOLOGY MAPPING ALGORITHM

First PROMPT creates an initial list of matches based on class names. Then the following cycle happens: Initially, the user triggers an operation by either selecting one of PROMPT’s suggestions from the list or by using an ontology-editing environment to specify the desired operation directly. Secondly, PROMPT automatically executes additional changes by generating a list of suggestions for the user which was based on the structure of ontology, and arguments to the last operation. Finally, it determines conflicts that the last operation introduced in the ontology and finds possible solutions for those conflicts. [21][22] A specific implementation of the PROMPT algorithm will use whatever measure of linguistic similarity among concept names is appropriate. Generally in our Protégé-based implementation, we use Protégé component-based architecture to allow the user to plug in any term-matching algorithm. PROMPT starts with the linguistic-similarity matches for the initial comparison, but concentrate on finding clues based on the structure of the ontology and user’s actions. For each operation in this set, the options provided are: (1) changes that PROMPT perform automatically, (2) new suggestions that PROMPT presents to the user, and (3) conflicts that the operation may introduce and that the user needs to resolve. Moreover the set of ontology-merging operations that we identified includes both the operations that are normally performed during traditional ontology editing and the operations specific to merging and alignment. The event occurred in prompt are,

a. Merge Classes, Slots, binding between Class, and Slot.
b. Performs a deep copy of a class between Ontologies.
c. Perform a shallow copy of a class.

We identified the following conflicts that may appear in the merged ontology as the result of these operations:

- Name conflicts
- Dangling references
- Redundancy in the class hierarchy
- Slot-value restrictions that violate class inheritance

X. CUSTOMIZATION ENACTMENT

After the mapping process, the inconsistent and duplicated data i.e., class, subclass name represents same functions, meaning or irrelevant information can be found and removed by the developer. Without the partial knowledge of the human developer it is time consuming to remove that information. The above proposed things expressed the customization approach in the service based business process.

Generally in the open web environments, the data, systems, services are becomingly increasingly decoupled, decentralized, distributed. The services may appear or reappear at any moment and thus no assumption can be made about contents availability or existence. This leads to the false information retrieval by the web service. The above system lack the dynamic behavior, i.e., during any change in the web environment [7], there will be a data loss due to addition, deletion, and modification of content in the databases. In this paper the change in the web content is focused in the ontology point of view. The dynamic behavior in the system discusses the semi automatic behavior of finding the changes in the ontology (i.e., finding the ontology modification impact by checking the similarity, consistency measures) and according to the comparison of previous unmodified measures with the new measures; the recovering strategy has to be processed.

XI. FINDING THE ONTOLOGY MODIFICATION IMPACT

The term impact refers to a significant change of elements in the ontology. Due to the application of a change operation on one or more elements in the ontology, the addition or deletion of axioms, data property, and object property may occur [24]. The impact can be structural or semantic. Structural impact points out the structural changes and semantic impact points out similar meanings of ontological
elements. Here the information in the database is viewed in the hierarchical taxonomical order. The content can be represented in the form of graph based formalism [11]. Graphs are selected for their known efficiency and similarity to ontology taxonomy. In our paper, the ontology and annotation are represented as graphical fashion in the Fig. 12 and Fig. 13.

Fig. 12. Behavior between the services in the graphical representation

Fig. 13. Graphical representation in Hierarchical view

The impact of change in the ontology is because of the computational resources and decision making process by the developer. Therefore the impact is predicted by capturing, analyzing nature of the operations, and the target ontology elements using different rules and parameters. Here the impact is found and checked by Consistency measures [19] [25], severity and similarity identification methodologies.

Consistency is analyzed based on measures that are evaluated by the developer before the implementation. The following rules stated here are: 1) Cardinality value (number of elements in the set), 2) user-defined constraint, 3) concept hierarchy invariant (no element should have a cyclic graph). Hence the severity has to be found. Severity is the degree of impact of a change operation on ontology. The impact is measured qualitatively using consistency or quantitatively using similarity measures. Using the selected scenarios the severity of change operations shows how difficult change affects the ontologies. [11] To indicate the severity scale, the human expert categorized impacts into:

1. Less impact: change with no effect on consistency (0-5\%).
2. Medium impact: changes with medium impact (5-25\%).
3. High impact: structural inconsistency which needs
involvement of developer (50-75)%. 

4. Crucial impact: significant changes in the dependent ontologies (80-100) %. 

For example the figure shows,

![Fig. 14. Impact in ontology structures before and after change](image)

By comparing ontology graphs, developers suggest that, change causes the medium impact of 24%. The Figure shows that, the Mobile domain with an attributes such as name of the mobile companies i.e., Samsung_phone, g5, sony_ericsson, nokia, and reliance. After the changes, the figure shows an extra item corby, Bpl as new elements in the ontology. The addition, deletion, and replacement of ontologies affect the situational awareness of web services. The annotations and behavior may change, which causes significant modifications in the results while during querying. The similarity [23] can be measured to get logical differences between the changed contents. The human view will only found 40% similarity because of large datasets and also it is time consuming. Some mathematical measures such as logical similarity, concept similarity, relationship similarity shows results in a quantitative way and it is easy to note. Meanwhile the relation between the input concepts (I-I relations), input output concepts (I-O relations), and output concepts (O-O relations) shown below respectively are,

**Logical similarity** 

\[ \text{Logical similarity} = 1/n \sum \text{Concept Similarity} \]

**Concept similarity** 

\[
\text{Concept similarity} = \begin{cases} 
  e^{-\alpha l - \beta h} & \text{if } C_1 \neq C_2 \\
  1 & \text{if } C_1 = C_2 
\end{cases}
\]

- \( n \) : total no of input output parameters 
- \( l \) : distance between concepts 
- \( h \) : height of the common ancestor concept 
- \( \alpha, \beta \) : weight for \( l \) and \( h \) 
- \( C_1, C_2 \) : concepts

**Relationship Similarity** 

\[ \text{Relationship Similarity } RS = \frac{1-I-I \text{ relation similarity} + I-O \text{ relation similarity}}{2} \]

If I-I Relation similarity is only for I-I relation exists. 
If I-O Relation similarity is only for I-O relation exists.

The correlation analysis is used to show the dependency relation between two variables, i.e., x axis as ontology change and y axis shows consistency [26]. The below scatter plot figure generated by the
ontology experts scenario showed that, when the change in ontology increases, consistency of the ontology decreases, i.e., negative correlation.

![Fig. 15. Scatter Plot graph between Ontology Change and the consistency of ontology](image)

After finding the impact, the necessary steps taken here are: First, making the system dynamic. Second, semi automatic, i.e., lost data or change due to adding new concepts can be retrieved and restructured [13]. For that, the Ontology versioning mechanisms have to be applied. It requires us to come up with a solution where such discrepancies while retaining as much knowledge as possible. Versioning ontology does not mean simply giving versions. It gives strong intelligent interferences [12] between the versions. The Figure. 16. of graph based implementation shows below,

![Fig. 16. Ontology Versioning Mechanisms](image)

The solution is that, we will determine some more ontology notations which are discussed in this paper. For denoting triples, we use <subject, predicate, object>. Hence considering ontology in the above figure as an example: first, OV1: A points out the class A in the ontology version 1. Second, B is a subclass of A that can be written as, <B, subclassof, A> and <C, subclassof, B>. This method makes use of separate type graph that is formulated for every ontology. The type graph [6] and versioned ontology are linked by the intelligent interferences X and Y respectively. The type graph contains the copy of each class with no relationships. Each version will have its own graph. Every ontology will have two virtual ontology [12]. The links between each version will be done by using type graphs. Let us consider via some example,

Let OVT1 is the type graph of version K, OVT is any type graph in general. The code for the type graph and corresponding instances is as follows,
For each triple <On:C type rdfs:Class> in On
   Add< OV:C type rdfs:Class> in OV
   Add<OV:C subclassof OV:C> in OVT
For each triple <On:P type rdf:Property> in On
   Add< OV:P type rdf:Property> in OVT
   Add<OV:P type rdf:Property> to OV
   Add<OV:T: P subProperty OV:P> to OV
Copying the axioms into the virtual ontology,
For each triple <On: subclassof On:C2 > in On
   Add<OV:C1subclassof OV:C2> to OV
For each triple <On: subpropertyof On:C2 > in On
   Add<OV:C subpropertyof OV:C2> to OV
The link corresponding instances to type graph are,
For each triple <On:I type On:C>
   Add< OVT:I type OVT:C> in OVT
The link corresponding to the type graph is,
For each triple <On:I type On:C>
   Add< OVT:I typeof OVT:C> in OVT
This gives representation of each ontology with the type graphs and their interferences with the different versions.

Here, we are using backward compatible methodology. The curve line from OVT1 to OT2 points that, all the instances in a type graph versions are mapped onto corresponding classes of all subsequent backward compatible versions of type graph. [13] X, Y interference shows that all classes, properties in the type graph are mapped to their sisters in the virtual graphs of all ontologies, which are backward compatible with the new versions. The thin black line shows that, we can successfully access all relevant ontologies. For large databases, it is ineffective for finding the missed ones and duplicated information. It is time consuming and it costs more.

XII. CONCLUSION
The work [26] looks at representing business process services in the Ontology Web Language (OWL) format. They discussed only about the problems (inconsistencies, discrepancies) and rectification (text categorization) in which ontology faces during the composition process. Their ontology matching rate does not get a correct matching rate of 100%. It needs some human intervention to keep note of changes and also makes more cost. The important one is that system lacks the automatic or semi-automatic behavior of customizing the contents. So the data losses during problem are not identified properly. Our approach evaluates the practical application of system in the real world. We have enhanced the ontology matching rate. From the case study we found that the proposed solution is effective in finding the impacts in a semi-automatic manner.

XIII. REFERENCE


