Ontology based Dynamic Customization for Composite Web Services

Thirumaran.M
Assistant Professor
Dept. Of CSE
Pondicherry Engg College

Dhavachelvan.P
Professor &HOD
Dept. Of CSE
Pondicherry University

Shanmugapriya.R
PG Scholar
Dept. Of CSE
Pondicherry Engg College

Aishwarya.D
PG Scholar
Dept. Of CSE
Pondicherry Engg College

ABSTRACT
The emerging semantic web era makes the entire web, user friendly to the humans by shifting the producer centric paradigm to consumer centric paradigm. Enhancing the web components still more user friendly increases its business value. For this purpose customization has to be done at runtime to provide sophisticated service to the business people and the customers. Business process Customization done for business goal analysis with BPEL, OWL, has shifted to OWL-BPC [web ontology language for business process customization] based on semantic markup language for web based information. We represent the conceptualization in an Extensible Markup Language (XML), based on the semantic markup language for Web-based information, i.e., OWL. The novelty of the work done in this paper is Customization done during or after the runtime time. Enabling the customers to customize the service and process during Requirement process, design process and testing the correctness of process logic while deploying process, modifying dynamically, substituting services and handling Runtime Exceptions according to the rules as services requested by customers. The framework is designed to handle the runtime customization. Dynamic customization is emphasized in the paper and it is the advantage of the proposed approach. We present an architectural description of the problem as a validation of the proposed approach.

Keywords
Semantic web, Consumer-centric, Goal analysis, Dynamic customization.

1. INTRODUCTION
Runtime customization is the idea that many portals have introduced to the internet world. We can develop new service, add new services or change any Existing service. Semantic web should support a shift of social interaction patterns from a producer-centric paradigm to a consumer–centric one. Runtime customization is the idea that it provides a sophisticated service to the customers. We can develop new service, add new services or change any Existing service according to the consumer's wish or consumer-Centric according to the somatic web. We present a representation of this conceptualization in a new Extensible Markup Language (XML) , based on the semantic markup language for Web-based information, i.e., OWL[9]. We name the conceptualization OWL-BPC for OWL on Business Process Customization. OWL-BPC[1] has been explained to conceptualize the problem of business process customization done at runtime. The requirement, design and testing 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 to be done at runtime to modify, create a new process. 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 standard language such as the Business Process Execution Language (BPEL) [7] or Web Ontology Language for Web Services (OWL-S) [8]. A service-based business process allows more agility and flexibility in the process due to loose coupling. Apart from that service can be reused, and dynamic binding of the service has to be done. 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. We would like to point out that, in this paper, we focus on the business scenarios where the business processes can be supported by dynamic and automatic service composition. In such scenarios, the instantiation of business processes allows a certain degree of flexibility in selecting business partners and adjusting the process parameters for the partners. In other words, here, we will only discuss service based business processes where the idea of runtime customization, runtime exception handling is done. We refer to the customization of business process as a machine-enabled capability of adapting a business process of a company according to the process of the customer or business partner that it is collaborating with. A generic solution to this issue has not been proposed so far due to reasons, such as lacking a proper definition of a body of knowledge for the customization of business processes and lacking its standardized representation and rationales of inference. The Customization in the business process is done according to the following ones. First, conceptualization definition for business process modeling explaining the meanings of concepts. The relationships between those concepts can also be done. 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 inconsistencies between business processes are identified and a suitable remedial action (customization enactment) is taken. This can be as the web services needed to be modified at runtime can also be done. Thus the dynamic customization for the goal analysis is done, what need to be done at runtime in order to add, modify or delete a service to form the required goal ontology or the target ontology. Our Research effort in this paper is that Customization done during or after the instantiation time. The Secondary Business process developed should collaborate with the Primary Business Process. The Secondary Business process is the goal ontology or the target ontology, which is mainly created or modified according to the new goal ontology. Secondary business process should be customizable during the runtime. The Rules should be added in order to develop the process during the runtime. Multiple ontologies need to be accessed from several applications. Mapping could provide a common layer from which several ontologies could be accessed and hence could exchange information in semantically sound manners. Developing such mappings has been the focus of a variety of works originating from diverse communities over a number of years. Thus there is a need for dynamic customization where runtime changes are needed for better customer support which is explained in the example. Let us consider a Travel Plan as the goal ontology. The travel plan contains transportation, hotel, and tourism. The transportation again contains air ticket reservation, Train ticket reservation. Hotel available in that particular city will be under the hotel domain. The tourism domain has the tourism places available in that particular city. New services need to be added, existing service has to be modified, if needed any service need to be deleted. Thus taking the travel plan with the goal ontology (existing plan) it may contain air ticket reservation, train reservation, if needed we should be able to add a new service (i.e.) bus service to the transportation domain in travel plan (Goal ontology). If needed information in the particular service has to be updated or changed. Bus_fare need to be changed. New Bus_no has to be added to the bus service, new bus route if any has to be added. The particular Bus_no and its route if changed have to be updated. Thus this requires runtime customization. In tourism domain if we need to add new tourism_places to the city, it must be possible at runtime. Let us take the example that if a new tourist place (park, zoo, and temple) has developed it has to be added as one of the tourist place in that particular city. Thus the input will be the services, output will be the goal ontology obtained after customization of the services. Thus the goal ontology will change accordingly, as and when new services are added to goal according to the customization request given by the customers. Thus in order to consume a service the Service can be available as a process. Thus all the services are in the form of WSDL format. The needed service can be consumed, according to the Requirement list of the consumers.

2. LITERATURE SURVEY:

The goal of semantic web is to shift the social interaction pattern from a producer-centric paradigm to consumer centric one. The paper discusses about the Static Customization in OWL-BPC [1]. It discusses about the Semantic web from Producer – centred to consumer centric paradigm. They focus on User Requirements, Design and Testing done at End user. OWL-BPC supports both static and dynamic customization. Static customization is explained in OWL-BPC. First, a conceptualization definition for business process customization leverages about the existing knowledge of business processes and Web services. For such a definition, we have developed a vocabulary of business process customization for modelling the meanings of concepts and the relationships between these concepts. Second, a representation of this conceptualization in a new Extensible Mark-up Language (XML) mark-up language, based on the fact of semantic mark-up language for Web-based information, i.e., OWL. We name the conceptualization OWL-BPC for OWL on Business Process Customization. Third, a framework for customizing service-based business processes based on OWLBPC by identifying the possible causes of discrepancies / inconsistencies between collaborating business processes (customization detection) and then taking suitable remedial actions (customization enactment) is done. The solution and framework has designed to do the following: 1) semantic inconsistencies like semantic mismatching of process parameters have been done; 2) behavioural mismatches between services which may or may not be compatible has to be done; and 3) address misaligned rendezvous requirements. These capacities are applicable to business processes with heterogeneous domain ontology. The Semantic Web is the second generation of the Web, which helps sharing and reusing data across application, enterprise, and community boundaries is explained in [2]. Ontology defines a set of representational primitives with which a domain of knowledge is modelled. The main purpose of the Semantic Web and ontology is to integrate heterogeneous data and enable interoperability among disparate systems. This paper classifies the ontologies developed for software engineering; it reviews the current efforts on applying the Semantic Web techniques on different software engineering aspects, and presents the benefits of their applications. We also foresee the possible future research directions. This paper introduces the Human Semantic Web (HSW) [3] as a conceptual interface, providing human-understandable semantics on top of the ordinary (machine) Semantic Web, which provides machine-readable semantics based on RDF. The HSW is structured in the form of a Knowledge Manifold and makes use of Unified Language Modeling (based on the Unified Modeling Language). The Semantic Web is discussed in terms of three levels of semantic interoperability: isolation, coexistence and collaboration. The HSW-browser Conzilla combines the semantics of RDF with the human-understandable semantics of UML in order to enable more powerful forms of human-computer interaction such as querying the Semantic Web through Edutella and supporting the concept-in context methodology. The interaction of business models [4] is used in consumer centric manner instead of using a producer centric approach for customizing the business process in cloud environment. The knowledge based human semantic web is used for customizing the business process. Thus to the business process to be customized as the primary business process and those that it collaborates with as secondary business process or SBP. Automatic customization enactment is an automated process of taking actions to perform the customization on the PBP according to the detected customization spots and the automatic reasoning on the customization conceptualization knowledge framework. Business process customization using process merging techniques [5] is explained in this paper. Service composition techniques lies in the field of business process management. Essentially a 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. These include the design and creation of the process itself or the modification of an existing one when business requirements change. One way of creating a new business process is by the combination of two existing ones which naturally should retain the behavioral features of both

original processes. This paper, discuses about the formal language to express behavioral properties of processes together with its semantics, and we show how it supports process merging.WS-BPEL is explained in [6]. Executable processes are business processes which can be automated through an IT infrastructure. These paper discuses about novel profile that extends the existing Abstract Process Profile for Observable Behavior by defining a behavioral relationship. It also shows that our novel profile allows for more flexibility when deciding whether an executable and an abstract process are compatible.The goal of semantic web is to shift the social interaction pattern from a producer-centric paradigm to consumer centric one. The paper discusses about the Static Customization in OWL-BPC [1]. It discusses about the Semantic web shifting Producer – centred to consumer centric paradigm. They focus on user Requirements, Design and Testing done at End user. OWL-BPC supports both static and dynamic customization. Static customization is explained in OWL-BPC. 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) [9] service profiles, process models and grounding. Research efforts reported in this paper seek to establish a generic solution to the problem of customization of service based processes from the following three aspects. First, we present a conceptualization definition for business process customization that leverages existing knowledge of business processes and Web services. For such a definition, we have developed a vocabulary of business process customization for modelling the meanings of concepts and the relationships between these concepts. Second, we present a representation of this conceptualization in a new Extensible Mark-up Language (XML) mark-up language, based on the de facto semantic mark-up language for Web-based information, i.e., OWL .We name the conceptualization OWL-BPC for OWL on Business Process Customization. Third, we present a framework for customizing service-based business processes based on OWLBPC by first identifying the possible causes of discrepancies / inconsistencies between collaborating business processes (customization detection) and then taking suitable remedial actions (customization enactment). Our solution and framework can do the following: 1) deal with semantic inconsistencies like semantic mismatching of process parameters; 2) resolve behavioural mismatches between services which may or may not be compatible; and 3) address misaligned rendezvous requirements. Such capacities are applicable to business processes with heterogeneous domain ontology.

3. PROPOSED WORK:

We need a framework so that it should be flexible for us to modify the system in order to change according to the customer's need. Thus there is a need for Dynamic customization of the services. Customization done during or after the instant ion time is called Dynamic Customization. This can be done by adding rules to the jena engine during the runtime. Service can be added according to the requirement of the goal ontology. Service if needed should be modified at runtime in order to avoid the time, cost of developing a model. The proposed architecture has been designed as shown in Fig 1. The input is Customization request given by the user for customizing the services. It is analyzed by the Domain Analyzer (e.g. Travel Domain).

3.1 Service registry:

Service registry consists of the various services provided to the user. A service registry is used for achieving reuse of the service. Service providers or developers store the services in the service registry. All the web services are available here. If possible available service can be used. If needed new services can be added to the service registry based on the requirement of the customer.

3.2 Rule repository:

Rule repository is the business rule store. Business rules are that rules added to the repository according to a particular business.

3.3 OWL-BPC:

It consists of classes and if needed for customization, new classes can be added to OWL-BPC (web ontology language for business process customization).

3.4 Customization pattern:

Customization pattern consists of various patterns found in the requests of the consumers. Several patterns are available as per the requirements of the customers.

3.5 Customization Manager

Customization Detector

In the Customization Detector, the Scoper and Instrumentor identify all the customizable contents of the PBP and identify the ones that do need a customization because of their discrepancies with the SBP. The Customization Detector relies on the Jess Rule Engine to inference on the OWL-BPC ontology for the knowledge of business process customization.

Customization Enactor

In the customization Enactor it acts based on the Event-Condition- Action. If any particular event and condition together satisfy a required action, it is said to act correctly. The required action will be performed.

Event Logger

The result is recorded by the Record Writer in Event Logger. Various events are stored in the event log. So the events take place accordingly.

Exception Handler

Exception handler is used for handling run time exceptions. Many runtime exceptions have to be handled.

3.6 Business Process Property Evaluator

Dependability detector

It detects the dependability of the operations between the two collaborating processes. There will be certain dependability between any two processes. Only if certain events occur in a process, the forthcoming process will relay on the previous process result. So it will be detected by the dependability detector to continue processing.

Execution planner

After detecting dependability and resolving the concurrency between two collaborating processes, execution is planned by Execution planner. Execution planner has the entire process execution list. Which process will occur first, and which process will occur next.

Concurrency resolver

Concurrency (Parallel) concerns between two collaborating processes will be resolved by concurrency resolver. Which event in a process will occur first? When two processes collaborate, which events in the process will take place.

Runtime Manager

The runtime operations after planning by Execution planner are executed and managed by Runtime Manager. Runtime manager has to plan events. Runtime exceptions have to be handled.

3.7 Ontology Manager:

Jena Engine

It can be done in .net and Java applications. Classes can be added to it. Modification to the services can be done in this Jena Engine. The available services can be replaced by altering the services in order to match the changes. New rules can be added to Jena Engine.

Ontology Builder

Protégé tool is used as an ontology builder. The OWL file is developed by the protégé tool. The protégé tool contains several formats available in OWL, RDF, and XML format. OWL format have to be selected, it is taken and can be used by any application with an interface in order

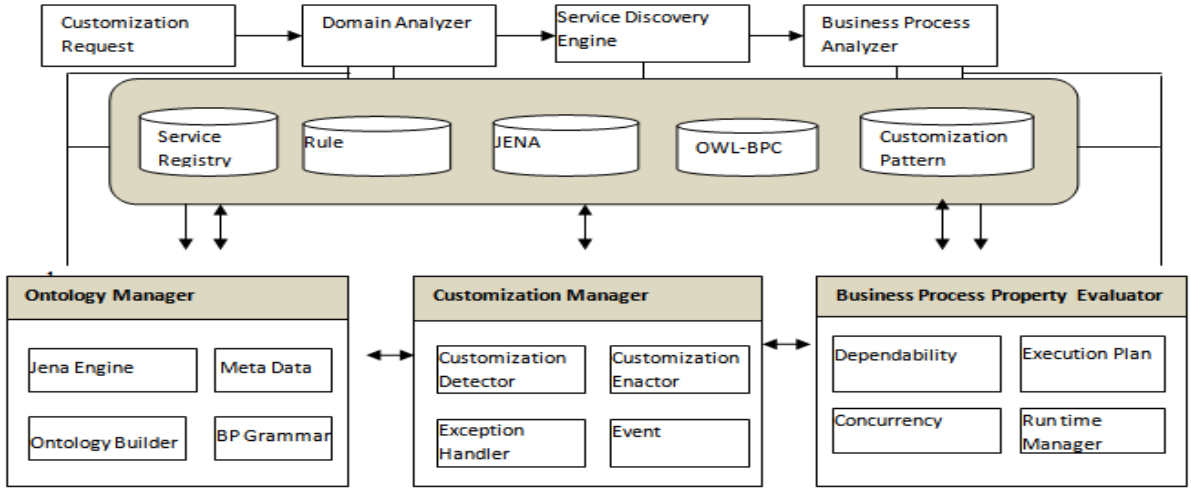

Fig1. Architecture Diagram for Dynamic Customization

BP Grammar

It contains Business rules in order to develop according to the customers need. Business process contains several rules to be followed called the BP Grammar. It can be the syntactic rules, to make the application to interface with the developed owl file.

Meta Data

Meta data contains the information in the many form as of OWL, XML, RDF. Metadata – Resource Description Framework (RDF), the presentation format of the metadata in support of describing and interchanging knowledge of customizing service-based processes is used. The Metadata service is one where the service modified is done here it can be stored in XML format done, during the runtime customization. The contents can be added, modified or deleted as needed. Thus this service is provided by metadata Service. The Architecture framework is designed so that the given customization request is taken to the domain analyzer. The customization request is given by the customers. The domain analyzer has all the information about the domain. Service Discovery Engine searches the service available from the service registry to find the required services, if needed new services can be added to the service registry. Thus the customers will customize the required services according to the goal ontology. Customers to customize service and process during defining process, verifying and checking the correctness of process logic while deploying process, modifying dynamically, substituting services and handling abnormal situations according to the rules provided. Thus rules or needed service can be added. Adding new service, modifying the available service can be done according to the goal ontology or the target ontology. This can be done during or after the instant ion time. Thus the implementation is shown in the screenshots in fig.2 the visa reservation, thus the passport_id is used as a comman reference id and the common fields are mapped to the other web services to airline booking shown in fig.3. The available hotels in the city are also displayed. Thus all the process is made as a single process. All the available service is mapped making it a comman single process. The customer can book a hotel reservation if needed. Apart from this if the customers have booked for all reservation, cancelling one should not cancel any other reservation. Adding any service in the middle of the travel plan should be possible.

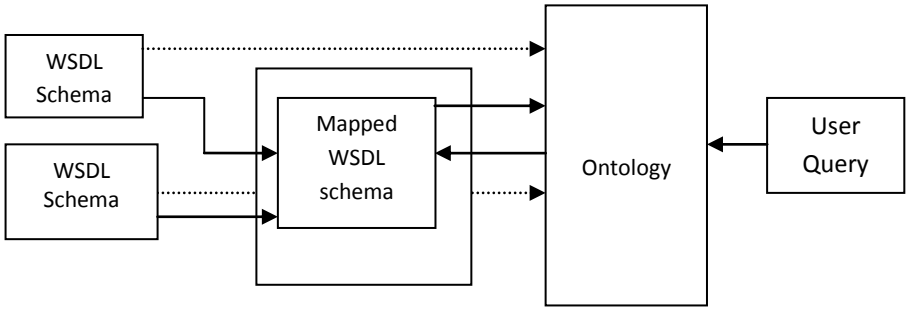

Fig2: Mapping Done to XML Schema Based On User Query

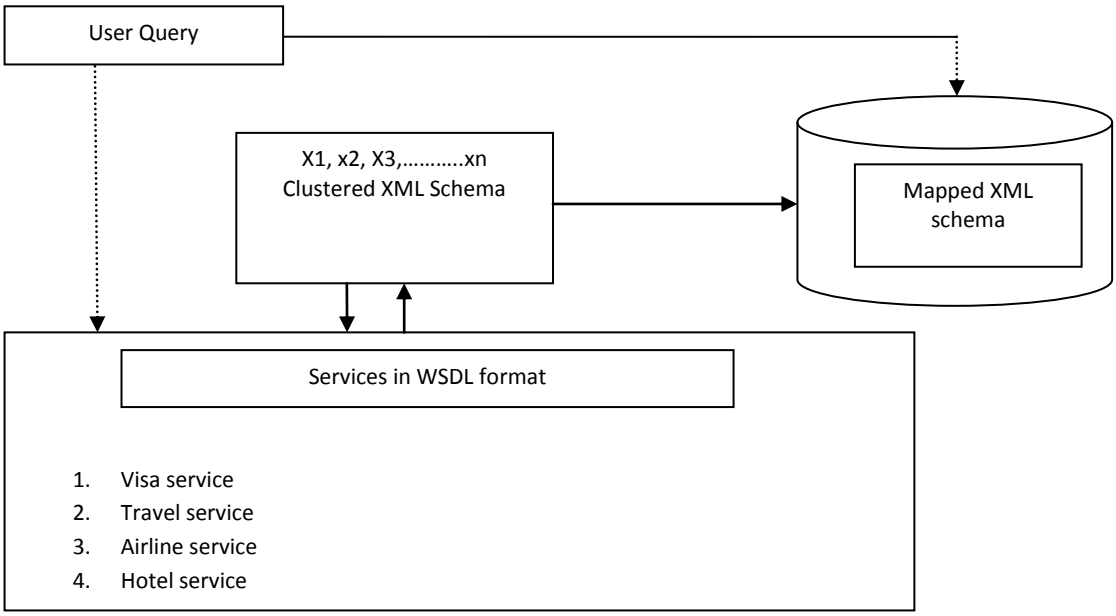

Fig 3 User Query is checked in both repository ad inWSDL pool

Ontology. As we have explained in the above example, if a new service bus service is added to the goal ontology. we will get the required goal ontology. Thus with the similar service, new service has to be added. Unwanted service can be replaced with the existing one. Thus this change has to be done. The new service has to be added to the new goal ontology, in order to meet the customers need. This kind of customization done at runtime provides a sophisticated service to the consumers. Thus the drawbacks of the static customization are that no attributes can be added here. Thus dynamic retrieval of data and the processing time to get the best F-Measure is not possible here. Dynamic mapping is not possible here, so that the mapped process will be stored in the repository. Thus during the Service Discovery, the retrieval of the service for the first time will have a maximum time. Thus during the second time, the processing time, will be less for obtaining the same services. Thus this kind of customization done during the runtime is called Dynamic customization. Apart from this changes done to the domain ontology also have to be done, adding a new service or modifying the available service during the runtime should also be possible.

4. METHODOLOGY:

Dynamic customization is that customization done during or after the instant ion time. Similarity in service can be measured. Similarity between {(Existing goal ontology) Services} with {(new goal ontology) services} can be compared. Thus the comman service with both can be obtained. Difference in service can also be measured.

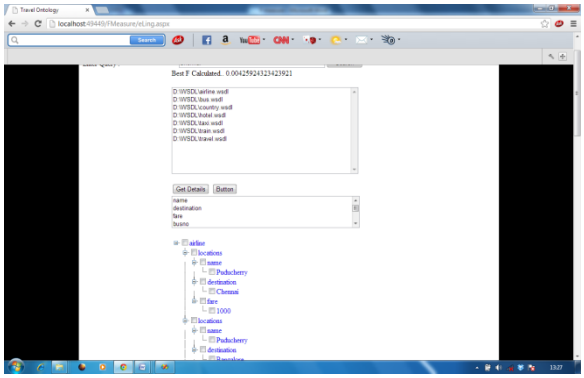

Fig4. Retrieval of service through Mapping from WSDL Pool

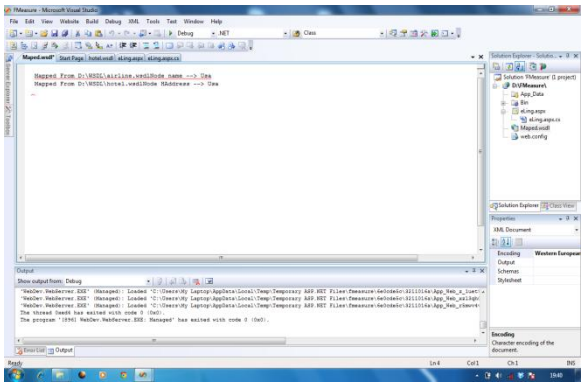

Fig5. Mapping done at runtime for WSDL for User Query Usa.

Table1:Customization done for retrieval of a service

Name of the Service	Customization for the service (Processing time for 1st time)	Customization for the service (Processing time for 2st time)	Customization for the service (Processing time for 3rd time)	Customization for the service (Processing time for 4th time ,,,,, etc)
Puducherry	0.1892	0.02350	0.00616	
Chennai	0.1406	.0932	.009571	
Nasik	0.07456	.01876	.007823	
Usa	.0081	.0019	.0023	
Bangalore	.0762	.0125	. 00603	

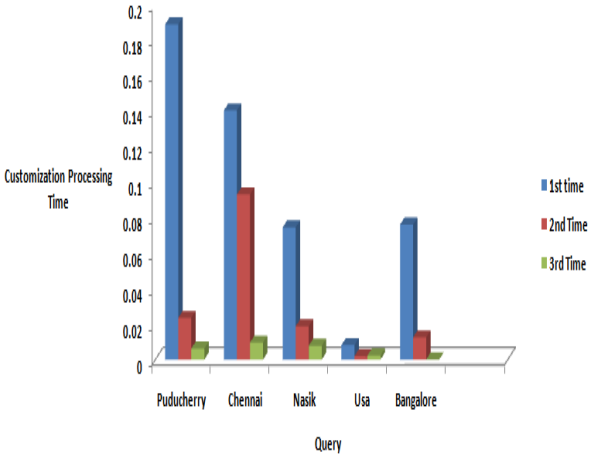

Fig5.Barchart for Customizationtime for a given query

5. CONCLUSION

In this paper we have described about the Dynamic customization. Thus the framework is designed to handle runtime customization by mapping. The business partners can sustain or strive in this business competition only if they can perform well with better Customer satisfaction. They should have to give the customers what they need. They have to adopt in a better way, in order to reuse, modify the existing service to achieve the needed changes in order to avoid the wastage of time and cost. This is achieved with the runtime customization of services. Mass customization has been the fashion. Better runtime support has to be done. As we get new requirement from the customers. The Goal ontology also changes. According to a new goal ontology new services has to be added. Similarity of service is also measured. After examining the similarity of the services. Thus a virtual model has to be developed for the new goal ontology. The exsiting goal ontology has to be compared. The new services needed has to taken and it has to be added to the target goal ontology. Service matching, has been done. Apart from this runtime Exceptions handling also have to be supported. . The drawbacks of static customization such as Addition of attributes at runtime should be possible, Mapping at runtime should be possible. Customization time descreses as and when the services are used again and again. Thus a Dynamic Customization should be designed to achieve the following. Thus Addition of Attributes at runtime should be possible, Runtime mapping should be possible. Customization time decreses as and when the services are used again and again. Thus the benefit of Dynamic customization done here supports, runtime customization which is the most expected one for the customers. Runtime changes to the domain, creation and addition of new process if needed is done. Further clustering can be done so that the new services added has to be assigned to a particular cluster so matched forming a new required goal ontology.

6. PERFORMANCE METRICS

F-Measure:
Thus F-measure is an evaluation Techniques used here.
F-measure is formally defined as:

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

N- Denotes no of Web services.
Xi- denotes n (n-1) number of times the services is checked
Thus the customization time decreases if the same query is retrived again and again. This kind of customization done at runtime is called dynamic customization.

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

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.

7. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template.

8. 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] Yajing Zhap, TuPeng, "Ontology Classification for Semantic-Web-Based Software Engineering", Journal, Vol.2, Pg. 303 – 307, oct-dec.2009.

[3] 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.

[4] 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.

[5] 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.

[6] D. König, N. Lohmann, S. Moser, C. Stahl, and K. Wolf, "Extending the compatibility notion for abstract WS-BPEL processes," in Proc. 17th Int.Conf. World Wide Web, 2008, pp. 785–794.

[7] 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.

[8] BPEL. [Online]. Available: http://www.oasisopen.org/committees/tc_home.php?wg_abbrev=wsbpel.

[9] OWL-S. [Online]. Available: http ://www.ai.sri.com/dam1/services/owl-s/1.2 .

[10] OWL. [Online]. Available: http://www.w3.org/TR/owl-features/ .

[11] 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.

[12] D. He and X. Wu, "Ontology-based feature weighting for biomedical literature classification," in Proc. IEEE Int. Conf. Inf. Reuse Integr., 2006,pp. 280–285.

[13] R. Katz-Haas, "Ten guidelines for user-centered Web design," Usability Interface, vol. 5, no. 1, Jul. 1998. [Online]. Available: http://www.stcsig.org/usability/newsletter/9807-webguide.html