Ontology based Dynamic Customization for Service Clustering

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
Customization has been the fashion in todays Business Scenario. As to respond to the customers regarding the consumers requirement customization is an important need today. Dynamic customization is an important part in internet portals. Today in order to satisfy the need of the consumers. The novelty of the work done in this paper 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 webservice added has to be matched with the most similar domain so forming a cluster. If the service so added do not match to the Available domain, a new Domain ontology has to be formed or created. This kind of customization done at runtime is called Dynamic customization.

Keywords – dynamic customization, goal ontology, owl bpc.

1. INTRODUCTION
Customization is done according to the customers need. Dynamic customization has to be done in order to develop a particular project. Clustering of service has to be done in order to add a new service to the goal ontology, as needed by the customers. So the required new ontology can be formed by altering the existing goal ontology. If the new service so formed does not match to the required domain, then a new service has to be added to a new domain. This new domain also have to be added to the goal ontology.

Let us explain this with an example. Let us consider the travel plan as the goal ontology. The travel plan in the existing goal ontology contains the various domain and its webservices. The new services added has to find the similarity and then the webservice has to be added to the respective domain or a new domain has to be formed. The goal ontology changes according to the need of the customers. Dynamic custamization provides runtime support so that new service can be added at runtime in order to form a new goal ontology by modifying the service. Our existing goal ontology contains various services such as transportation service, room reservation service and tourism service. Thus this service in turn contains many subservice like airline booking, visa service etc. room reservation contains the hotel booking service. Tourism service contains many tours in that city. Thus if the new goal ontology required by the consumer is different. New service has to be added to the service registry, to form a new goal ontology is at the domain level. Thus only if the service is available we can add the needed service to it .Thus dynamic customization also has to be done at process level. This can be explained with the help of a casestudy. Let us take an Case1: Let us say a person wants to customize a particular service alone, he must be able to customize that particular service alone. Let us say if the person wants to customize a particular service, to book a airticket he must be possible to customize the Airticket Service alone. Case2: If a person wants to cancel a particular service, it must be possible. Thus if the person A customize a airticket , a visa service, reserves a hotel and a tourism in a particular city say dubai. He must be possible to cancel the tourismservice in that city. Case3: if a person B customizes the Airticket to Dubai to thiruvanthapuram. If he wants it must be possible to add other services for him. So other service like tourism in that city say Kovalam should be added. Thus this kind of customization done during the runtime provides a sophisticated service to the customers. So he need not reserve again the same service. Thus needed service alone can be added to it. Again dynamic customization has to be done at application also, if needed the colour of the form can to be changed. Thus this can be done by altering the available service than to create the entire process again. This kind of customization done at runtime is called dynamic customization. Thus the available process will be considered as the PBP and the new Goal Ontology is called the SBP. Thus the Secondary Business Process has to collaborate with the Primary Business Process. Before the deployment process is done we have to form a Virtual model.

2. RELATED WORKS
The main 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. 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-basedbusiness (customization detection) and then taking suitable possible causes of discrepancies / inconsistencies between collaborating business processes based on OWL-BPC by identifying the remedial actions (customization enactment) is done. The solution and framework was designed to do the following: 1) Semantic inconsistencies like semantic mismatching of process parameters has been done; 2) behavioural mismatches between services which may or may not be compatible has to be done; and 3) address misaligned rendezvous requirements. Such capacities are applicable to business processes with heterogeneous domain ontology is also explained. 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 between 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 for semantic web is used for customizing the business process. 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. semantic web is used for customizing the business process. 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. 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. This 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 for customizing the business process. 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. Third,a framework for customizing service-based business process with heterogenous domain ontology is also explained.

3. EXISTING WORK

Here customization is explained using Web ontology language for business process customization is explained(OWL-BPC). Here the automatic customization Enactment, customization Detection is done. Here the static customization of webservice is done. 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 OWLBPC. 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

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 Heterogenous goal ontology.

4. PROPOSED WORK

Thus the Dynamic customization of service has to be done for the runtime support of the consumers. The virtual model is done here before the deployment phase. After the virtual model is designed. The designed virtual model is checked if it satisfy the requirement, testing, Change in customization = difference (New Goal Ontology) to (Existing Goal Ontology). Finally deployment of the model as needed by the customers has to be done. Thus before any model is build. Thus the virtual model has to be designed. Thus the Primary Business Process has to collaborate with the Secondary Business Process. The Secondary Business process is one which is developed according to the customers need. Thus it uses the PBP. The PBP contains the existing services, it is possible for us to reuse the service. Thus the new service is added to the SBP, and the change in customization can be defined as shown.

Clustering done at Domain level: Thus according to the requirement of the new goal ontology, a new service can be added as shown in fig2. clustering have to be done to form a new goal ontology. Thus the required service is grouped under a common domain and in turn it is grouped under a new goal ontology. Thus according to the requirement list, the new service has to be created, and added if it is not found in the service registry. Thus the existing goal ontology have to be taken and it has to be analysed and changed as needed. Thus in the below fig1. It shows a virtual model, a design before the services are used. Thus the PBP or the Primary Business Proccess is that it contains the already available Existing Goal Ontology. The SBP or the Secondary Business Process is one where the new service that need to be collaborated is added here. Thus the new goal ontology will be formed after collaboration. This is done before they are deployed in order to find any drawbacks. Thus newservice can be added according to the requirement. This can be done by a search in web. Thus the available service can be obtained from the web. Thus the related service and the available service is obtained form this search. Thus the correct service can be added as needed by the requirement list .

ALGORITHM

```
// Clus ---- >cluster
// Dom ---- >Domain
// Prs ----- > process
// Go ---- > goal ontology
// Eo ------ > Existing ontology
// Ws  ----> service
// nGo-----> new goal ontology
// rws--- > required webservice
// Let  Eo contain the dom
Let m= 1 to n which denotes the no of webservices
Let i = 1 to n which denotes the no of domain
Domain
{
for(i=1;i<=n;i++)
for(m=1;m<=n;m++)
{
    if (Ws(m) == Dom(i)) //check if the webservice belongs to the available domin.
  {
   Dom(i) = Dom(i) + Ws(m);
  }
   Else
   {
     Dom(i) = ndom + dom(i); //create new domain and add it to the domain.
    }
}
}
//Process
{
If (Ws(n)== nGo)          //check if the new process contains all the webservice in the existing ontology.
Then
{
for(i=1;i<=n;i++)
{
rws = rws + ws(i) ;       //check if webservice is available and add it to required web service
}
Np = Np +  rws;           //required webservice is added to new process.
}
}
//Add Process
for(m=0;m<=n;++)
{
 If (Ws(m))=!nGo)
 Then
{
   nGo = nGo + Ws(m);
```

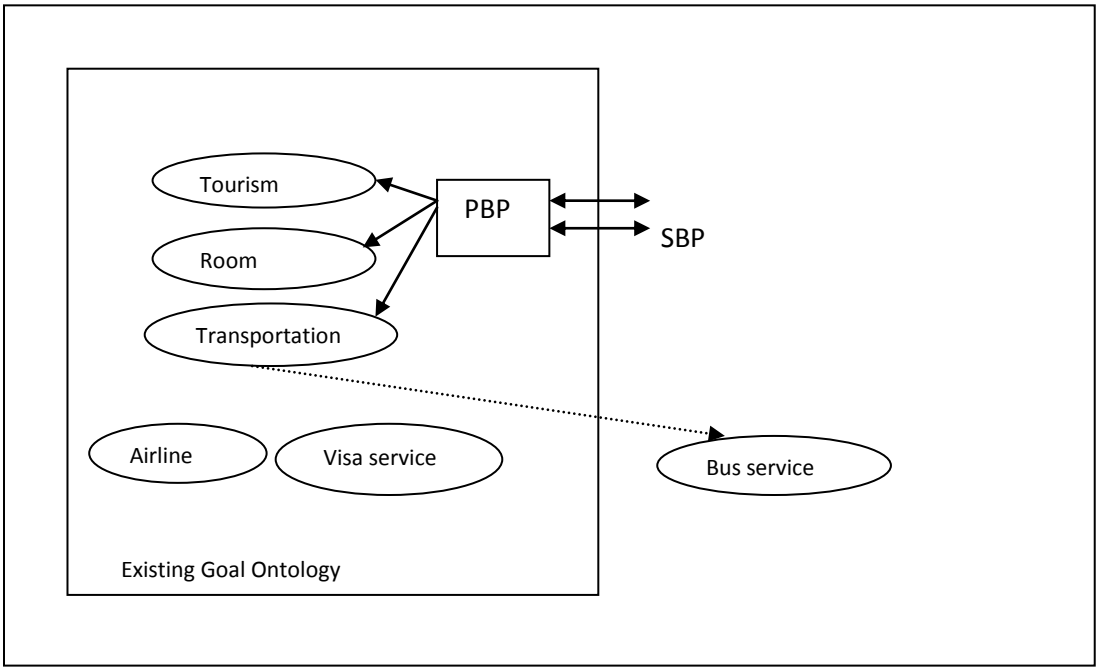

Fig 1: Virtual Model

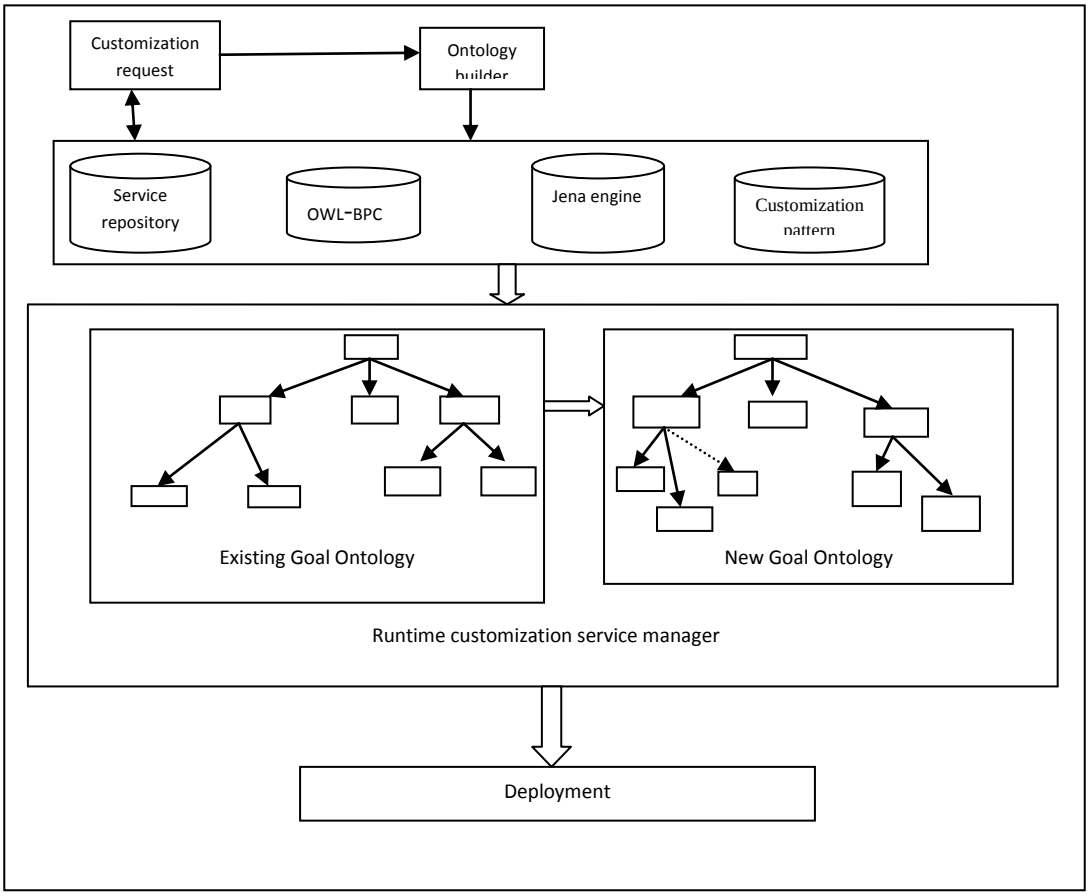

Fig2. Architecture diagram for Runtime customization of the webservice

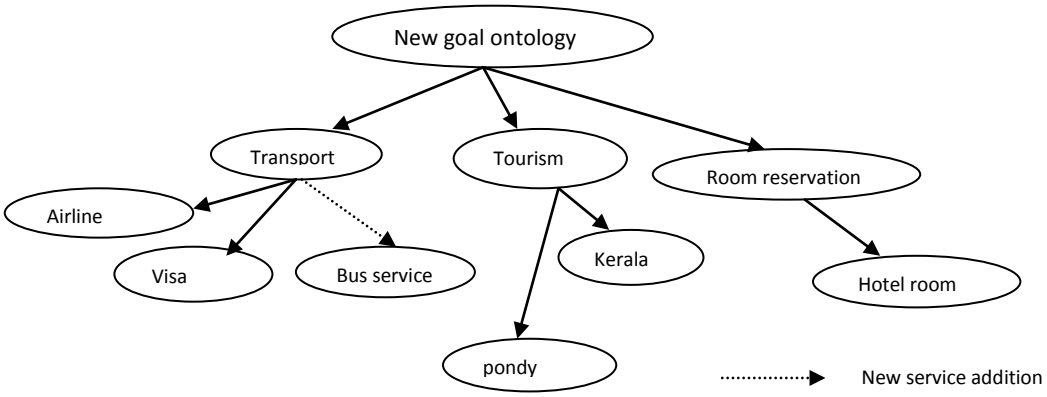

Fig3. New Goal ontology with a new service

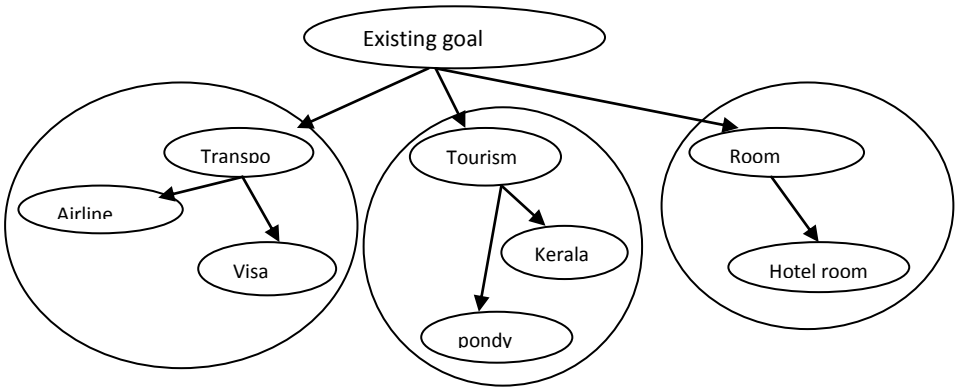

Fig4. Existing Goal ontology

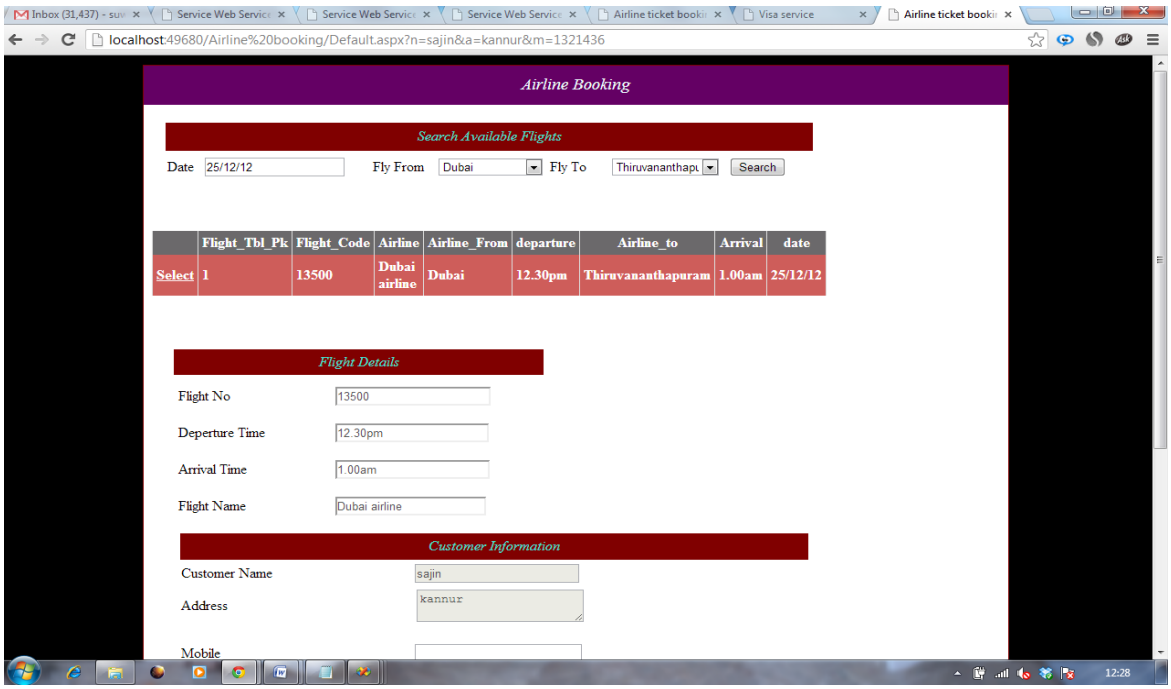

Fig5: Here it does not Contains the tourism service

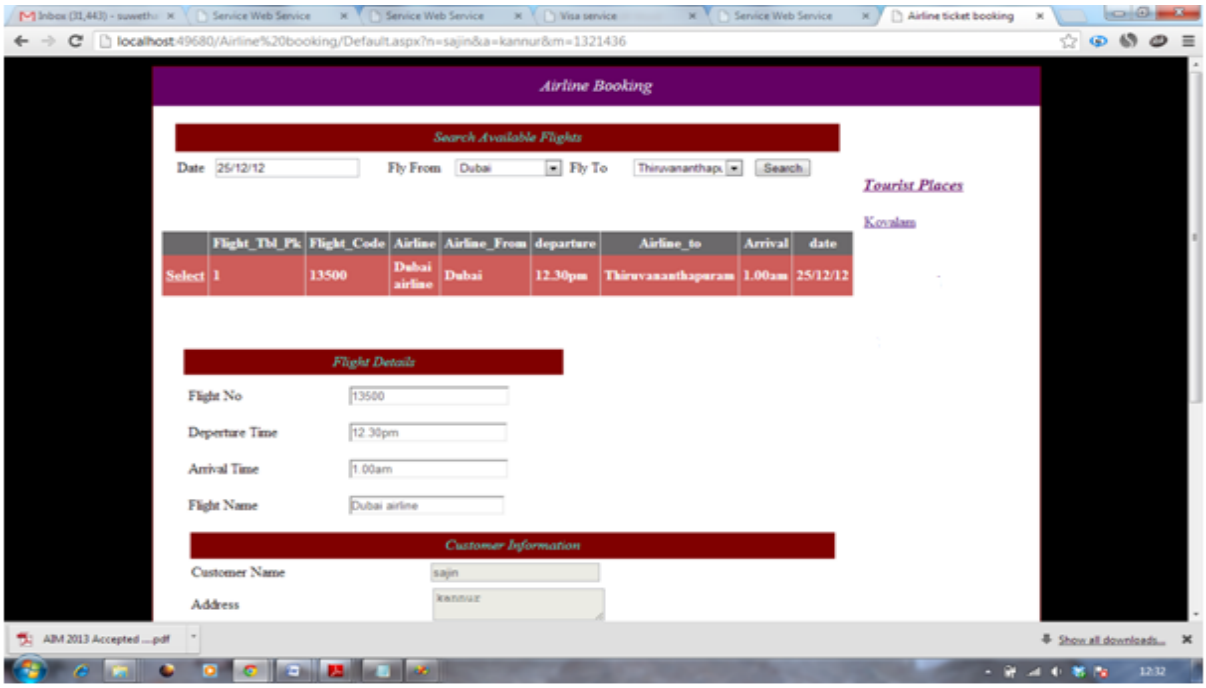

Fig6: Tourism service is added with kovalam as one place

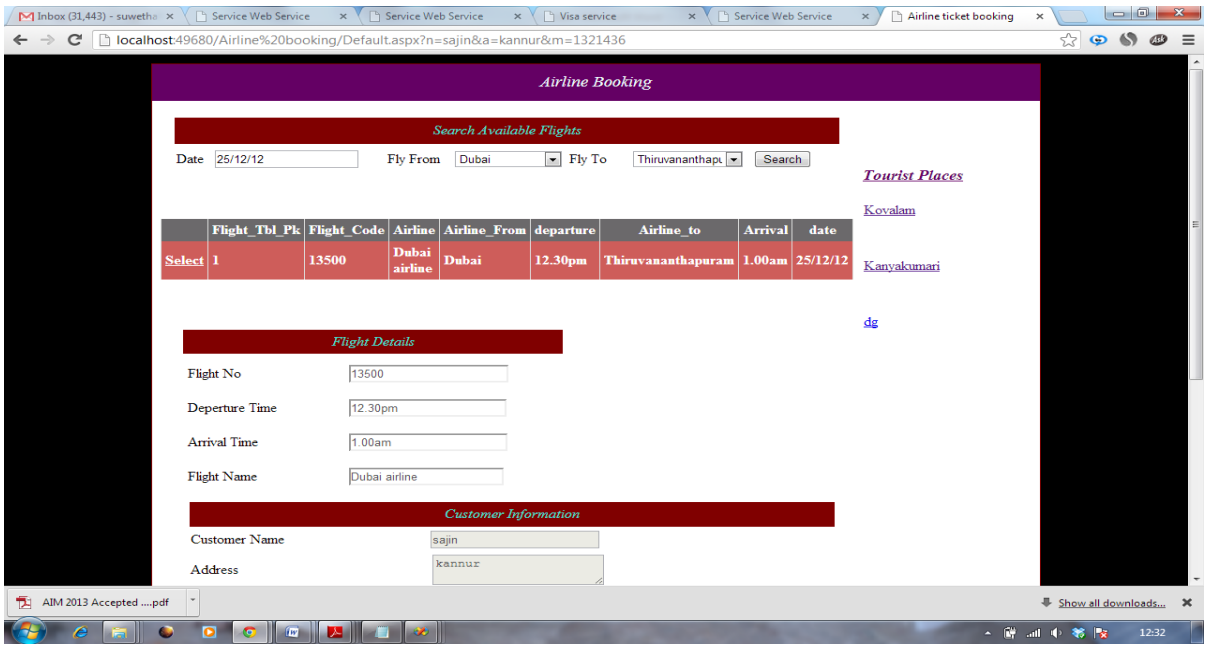

Fig7: Tourism service is added with many places

}//if we need to add a new webservice to the process , if the service is not available in the process then add the needed service to the process

}

//Cancel process

for(m=0;m<=n;++)

{

If(Ws(m) =! Rli)

then

{

nGo = nGo – Ws(m);

}

Thus the algorithm above explains about the domain which explains about the service level customization. Thus new service 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. Thus if the needed service matches the available cluster,then the service is added to that particular domain. If the newly added service do not have a match in the available cluster a new cluster or a new domain have to be added forming a new Goal ontology. Thus the service is added according to the requirement list. Thus the input is the services from the requirement list. Output is the goal ontology formed from the existing goal ontology with needed service added. Here a new domain or a service need to be added.Thus a new Service has to be added to the travel plan. Thus as per the requirement list the tourism service need to be added to the New Goal Ontology. Thus initially in the travel plan it contains only Air service, visa service. But now the customers have should be in a position to customize his service at runtime. Thus the tourism service has to be added. Let us consider the Tourism places in fig.6, first it only contains one tourism place that is Kovalam. Thus if we need to add many service (i.e.) kanyakumari a new tourist place. This should be added as a new service to the tavel plan to the tourismdomain. This should be possible at runtime. This kind of customization is only possible in Dyanmic Customization and it is not possible in static customization. Thus in the fig2, screenshot2 shows tourism places such as Kovalam, kanyakumari, dg. Thus If needed a new service hotel say a domain can be added. This kind of customization is not possible in static customization. Since in static customization the service designed is a fixed one. Thus if needed new service can be added only in Dynamic customization. This kind of

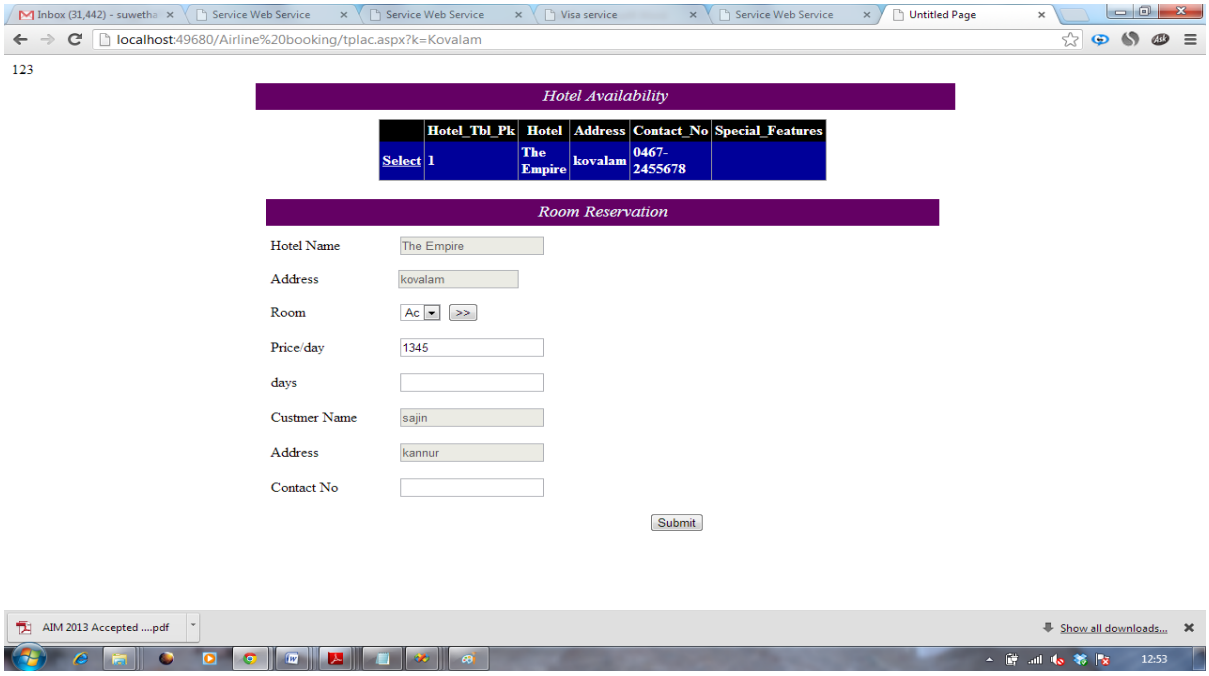

Fig8: Screenshot3 shows a new service hotel

customization is only possible in Dynamic customization.

5. PERFORMANCE METRICS

Similarity is measured. Thus the available service can be reused, the needed service can be added. Thus the similar service between the both is known. Thus the service in the existing goal ontology can be reused by the new goal ontology. Thus F-measure is a evaluation Techniques used here.

F-Measure:

The F-measure is often used in conjunction with Precision and Recall, as a weighted average of the two. If the weight is set to 0.5 (which is usually the case), Precision and Recall are deemed equally important. F-measure is formally defined as:

F-measure = (2* {Precision * Recall}) / ({Precision + Recall})

Precision:

Precision is the fraction of retrieved instances that are relevant. Thus to add a new service according to the requirement list, a search has to be done. Precision measures the number of correctly identified items as a percentage of the number of items identified. In other words, it measures how many of the items that the system identified were actually correct, regardless of whether it also failed to retrieve correct items. The higher the Precision, the better the system is at ensuring that what has been identified is correct.Thus Precision can be represented as

Precision = ({Relevant concepts} ∩ {Related Concepts}) / ({Relevant Concepts})

Recall:

Recall is the fraction of relevant instances that are retrieved. Recall measures the number of correctly identified items as a percentage of the total number of correct items. In other words, it measures how many of the items that should have been identified actually were identified, regardless of how many spurious identifications were made. The higher the Recall rate, the better the system is at not missing correct items.

Recall = ({Relavant concepts} ∩ {Retrieved concepts}) / ({Retrieved concepts})

6. CONCLUSION

Thus ontology based Service clustering is done. Clustering is done at domain level. Thus the required service as needed is clustered with its respective domain. Apart from that process level clustering also is done. Thus a new Goal ontology developed is based on clustering done at domain level and process level.

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