ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Automated Synthesis Mediators of Web Services

P Munavijayalakshmi, K.Dinesh Kumar

M.Tech(SE) Student, Sir Vishveshwaraiah Institute of Science & Technology, Madanapalle, Andhra Pradesh, India

Assistant Professor, Sir Vishveshwaraiah Institute of Science & Technology, Madanapalle, Andhra Pradesh, India

ABSTRACT: Interoperability could be a major concern for the code engineering field, given the increasing have to be compelled to compose parts dynamically and seamlessly. This dynamic composition is usually hampered by variations within the interfaces and behaviours of independently-developed parts. to handle these variations while not dynamic the parts, mediators that systematically enforce ability between functionally-compatible parts by mapping their interfaces and coordinating their behaviours are needed. Existing approaches to intermediator synthesis assume that associate degree interface mapping is provided that specifies the correspondence between the operations and knowledge of the parts at hand. During this paper, we tend to gift associate degree approach based on metaphysics reasoning and constraint programming so as to infer mappings between components' interfaces mechanically. These mappings guarantee linguistics compatibility between the operations and knowledge of the interfaces. Then, we tend to analyse the behaviours of parts so as to combine, if potential, a intermediator that coordinates the computed mappings therefore on create the parts move properly. Our approach is formally-grounded to confirm the correctness of the synthesised intermediator. We demonstrate the validity of our approach by implementing the MICS (Mediator synthesis to attach Components) tool and experimenting it with numerous real-world case studies.

1. INTRODUCTION

Interoperability leverages the ability of software products to work with other software products without special effort from end users. The capability to interact and exchange information both internally and with external organizations is a key issue in the economic sector. Successful enterprises understand the risks and exploit the benefits of Information Technology (IT), Business Integration, Business Intelligence (BI) and find ways proving platform for business automation, alignment of IT strategy with the business strategy, cascading IT strategy and goals down into the enterprise, obtain value from IT investments, provide organizational structures that facilitate the implementation of strategy and goals, create constructive relationships and effective communication between the business and IT, and with external partners and measuring its performance. At present situation, organizations can create interoperable systems without inventing new technologies; they can the technologies like web services [1], multi-agent systems [1] and so on. The issues which arrived here is more than interoperability. We assumed that our technologies are matured enough to provide interoperability for software services [2]. Also we assumed that software and applications have their own controls [3]. Now the point is how much control the systems have to comply with the regulations or policies given by the government. Again we assumed that the systems have sufficient controls to comply with regulations but the point is how the systems will align regulations or acts when they want to interact. At least if there is any kind of violation in the policies at execution time, how the systems will be aware of it in terms of software regulations, privacy, security and so on. So the question is what kind of approach we have to follow, what type of architecture we need to come up with these kinds of violations.

First, automatically generating mediators helps developers to manage the bulk of the systems they need to integrate. Indeed, developers increasingly have to incorporate to their systems convenient services such as instant messaging or social interaction. Although, most of these services provide similar functionalities, their interfaces are usually heterogeneous. For example, Google Talk and Facebook chat both have instant messaging capabilities but expose them using different interfaces.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

In this paper we have presented the most challenging issues of interoperability for enterprise software or applications in term of controls. Our ongoing research highlights all of these challenges. The most challenging issues we encountered are Enterprise Content Management, Compliance, and Service Level Agreement (SLA) [4] and Technological barriers. In section two we describe how enterprise content management influences current business systems, in section three, we have described how compliance issues occur in systems operability and how much it is important for the system to align with compliance rules, in section three, we have what are the challenging issues to deal with SLA in terms of interoperability and section five we have shown technological constraints of systems interoperability. In section six we have mentioned about the antitrust laws for enterprise Interoperability.

There must be a contract or combination behind any general interoperability issues there are some policies among the systems or software products. It is, therefore, necessary for at least two systems to agree or to act in concert for there to be a violation of Section 1. The contract or combination need not be formal or written; a "knowing wink" or, under certain circumstances, a consistent, parallel course of conduct between systems can form the basis for an illegal agreement. The contract must be in restraint of interstate or foreign commerce. The term "interstate commerce" is construed liberally to include restraints within a single system or state, if those restraints have a significant impact upon commerce between systems or states. The contract or combination of systems must be an "unreasonable" restraint on competition. Experience has revealed that some restraints, such as price fixing, are always unreasonable; other restraints, such as assigning distributors specific marketing territories, are examined on their facts to determine their reasonableness. Promotional and consumer awareness activities all these activities should be generally analyzed under the rule of reason, unless the activities constitute a pretext for an anti-competitive purpose.

II. EXISTING APPROACHES

The automatic generation of mediators assume the correspondence between the operations of the mediated components to be given in terms of an interface mapping [14] (also called adaptation contract [7]). Identifying such correspondence requires not only knowledge about the components but also knowledge about the domain itself. Research on knowledge representation and artificial intelligence has now made it possible to model and automatically reason about domain information precisely, if not with the same nuanced interpretation that a developer might [15]. In particular, ontologies build upon a sound logical theory to provide a machine interpretable means to reason, automatically, about the semantics of data based on the shared understanding of the domain [16]. They play a valuable role in software engineering by supporting the automated integration of knowledge among teams and project stakeholders [17]. Ontologies have also been widely used for modelling Semantic Web Services and achieving service discovery and composition [18]. OWL-S (Semantic Markup for Web Services) uses ontologies to model both the functionality of a Web Service and the associated behaviour, i. e., the protocol according to which this Web Service interacts [19]. Besides ontology based modelling, WSMO (Web Service Modelling Ontology) relies on ontologies to support runtime mediation based on pre-defined patterns but without ensuring that such mediation does not lead to an erroneous execution (e.g., deadlock) [20]. Hence, although ontologies have long been advocated as a key enabler in the context of service mediation, no principled approach has been proposed for the automated synthesis of mediators by systematically exploiting ontologies. We argue that interoperability should not be achieved by defining yet another ontology nor yet another middleware but rather by exploiting the knowledge encoded in existing domain-specific ontologies together with the behavioural description of components and using them to generate mediators automatically. These mediators bridge the interoperability gap between heterogeneous components by delivering information when it is needed, in the right format, with the original business context intact.

interoperability among pervasive networked systems, in particular accounting for the heterogeneity of protocols from the application down to the middleware layer, which is mandatory for today's and even more for tomorrow's open and highly heterogeneous networks.

As networked systems are becoming increasingly pervasive, they need to compose dynamically with their ever evolving environment according to functionalities they provide and/or request. However, such dynamic composition is greatly chal- lenged by the heterogeneity and autonomy of today's digital systems, which are not designed in concert, but are instead independently developed and deployed within pervasive networking environments. As a result, although

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

networked sys-tems may possibly match from the standpoint of provided and required function- alities, actual behavioral matching is unlikely due to inherent design diversity. Therefore, what is needed for enabling the composition of pervasive networked systems is emergent connectors [28], which embed a mediation process so as to adapt the systems' respective interaction behaviors for the sake of coordination.

III. PROPOSED APPROACH

Focuses on functionally-compatible components, i. e., components that at some high level of abstraction require and provide compatible functionalities, but are unable to interact successfully due to mismatching interfaces and behaviours. This paper concentrates on service-based components (e.g., Web Services) rather that code-based components (e.g., Java libraries). We propose an approach that combines ontology reasoning and constraint programming in order to generate a mapping between the interfaces of these components. Then, we use the generated mappings and examine the behaviours of both components to automatically synthesise a mediator that ensures their safe interaction. The mediator, if it exists, enables the components to progress synchronously and reach their final states with the guarantee that the composed system is free from deadlocks. Specifically, our contributions are:

_ Efficient interface mapping using semantic reasoning and constraint programming. We reason about the semantics of data and operations of each component and use a domain ontology to identify the semantic correspondences between the interfaces of the components, i. e., interface mapping. Interface mapping guarantees that operations from one component can safely be performed using operations from the other component.

_ Automated synthesis of mediators. We explore the behaviours of the two components and generate the mediator that composes the computed mappings so as to force the components to progress synchronously. The mediator, if it exists, guarantees that the mediated system is deadlock-free.

_ Tool-support for automated mediation. We further demonstrate the feasibility of our approach through the MICS (Mediator Synthesis to Connect Components) tool and illustrate its usability using real-world case studies involving heterogeneous components. Our semantic-based approach to the automated generation of mediators leads to a considerable increase in the quality of interoperability assurance between heterogeneous components:
it removes the programming effort and ensures that the components interact successfully while preserving efficient execution time.

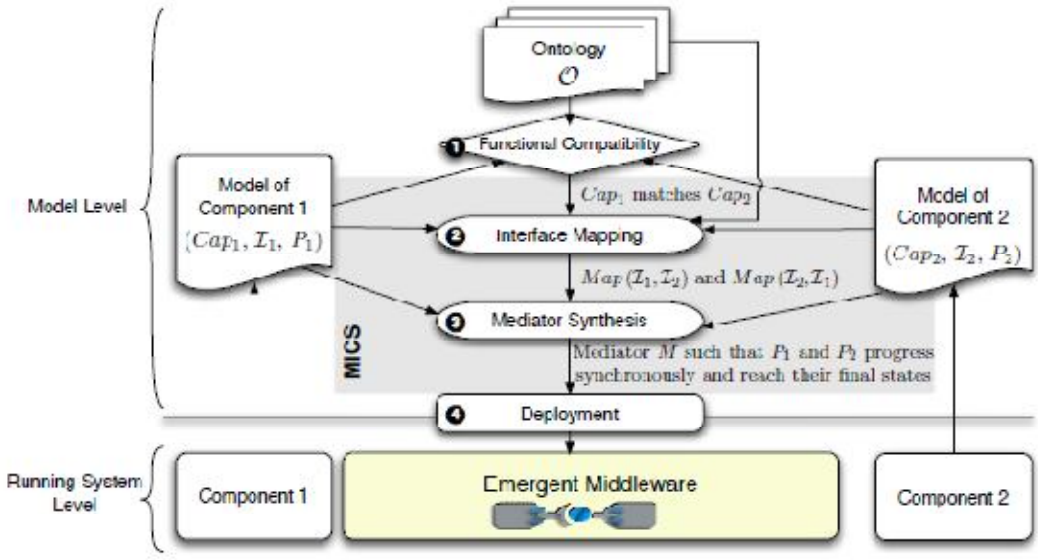

Fig. 1 Overview of our approach to the automated synthesis of mediators

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

3.1 Implementation of Ontology Based Clustering Algorithms

In data mining, clustering is predominantly used due to its wide variety of applications. Many data clustering algorithms have been introduced in recent years to be used in variety of areas which include economics, medicine, computational biology, image processing and mobile communication(Hartigan). Despite its wide use, standardization of the clustering algorithms is it'sthe major setback. An algorithm developed for a given type of data set may produce very poor results in different data sets(Hartigan). Despite huge efforts in trying to standardize clustering algorithms, there has been no major accomplishment. Efforts to standardize clustering algorithms has resulting into the proposal of many algorithms with each having its advantages and disadvantages.

Clustering refers to the process of partitioning given data sets to a homogenous data set group. This is based on features such that objects that tend to be similar are classified into one group and objects that tend to be dissimilar are classified into a different group(Maulik, Bandyopadhyay and Mukhopadhyay).

Ontology refers to formal specifications that are used in conceptualization(Punitha, Thavavel and Punithavalli). This specifications of conceptualization assist the system administrators and programs share information. Ontology comprises set of relationships and events that are utilized in creation of sets of vocabulary for exchanging information between programs and/or system administrators. Ontology is based on the concept that different system administrators have different requirements regarding clustering of algorithms(Punitha, Thavavel and Punithavalli). Empirical and mathematical clustering are utilized in high dimensional space to categorize information into one cluster as required. Cluster analysis is aimed at dividing sets of objects into one homogenous cluster(Gavrilova).

There are various general steps followed during implementation of the ontology – based clustering algorithms(Canadian Society for Computational Studies of Intelligence). The first step involves calculating distance matrix and/or similarity matrix existing between pair of objects by use of specific methods that are ontology based. In this stage, each object consists distinct clusters. The second step involves merging the two closest clusters by use of the distance matrix. This process is also referred to as the clustering process. The third step involves rebuilding and modifying the distance matrix. This is done by the treating each of the merged clusters as one entity. To achieve this, ways of calculating similarities between clusters and objects and ways that estimates the similarity existing between ontology objects and clusters are utilized. This is also known as the evaluation process(Punitha, Thavavel and Punithavalli). Fourthly, in case the required number of clusters have been identified, the process is altered, otherwise, the process is repeated from the second stage. To calculate the similarities between objects, the equation below is utilized;

$$\text{sim}(I_i, I_j) = f_{wp}(TS(I_i, I_j), RS(I_i, I_j), AS(I_i, I_j))$$

In the above equation, TS represents taxonomy similarity, AS represents the attribute similarity and, RS represents relationship similarity. The taxonomy similarity (TS) are the dissimilarity or similarity existing between the classes of the objects (Punitha, Thavavel and Punithavalli). There are various ontology – based clustering algorithms. Some of them are as discussed below.

3.2 Ontology based apriori clustering algorithm

Apriori algorithm is the most influential algorithm used in mining frequent Item sets for Boolean association rules. The generation of association rule in apriori algorithm is achieved through two processes. Initially, there is application of minimum support to facilitate finding of all frequent items in the given database. The second process involves the use of the frequent item sets and minimum confidence constraints to construct rules. Basically, the apriori algorithm is aimed at generating item sets of a given size and then scanning the database for comparison to see if the generated items sets are large. Only the item sets that are large are utilized in generating item sets for the next scanning process. An example of implementation of apriri algorithm is as shown below;

Input: The data set – Clustered Dataset (CD)

$CD\ (r_1, r_2, r_3, r_4, \ldots\ldots, r_{n-1}, r_n)^T$;

The feature set – Clustered Features (CF)

$CF\{f_1, f_2, f_3, f_4, \ldots\ldots, f_{m-1}, f_m\}$;

The ontology feature tree – Ontology-Tree(OT)

Ontology-Tree = {<*Cpt.root,partOf,Cpt.1*>,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

<*Cpt.root,partOf,Cpt.2*>, <*Cpt.1,propertyOf, Ab.1*>, <*Cpt.1,propertyOf, Ab.2,……*}.
Output: feature weight set
Weight={w_1,w_2,w_3,w_4,…,w_{m-1},w_m};
The Clustering results – Clusters (C)
Clusters{c_1,c_2,c_3,c_4,……,c_{k-1},c_k} .
Procedures
BEGIN
Input <– *CF*{f_1,f_2,f_3,f_4,…..,f_{m-1},f_m }
for a=0 to n:
for b=0 to m:
Correlation(*Ab.a*,*Ab.b*)
endfor
endfor
for a=0 to m:
Weight(Ab.a) ←($\sum^{n}_{b=0}$*Correlation*(*Ab.*a, *Ab.*b))/*n.*
endfor
do
begin
forint a to n do:
$L_2 = (\sum^{m}_{k-1} Weight(Ab.k)(field_{ka} - field_{kb})^2)^{1/2}$.
endfor
end
while(Clusters center does not Change) :
Output –>*Clusters*{c_1,c_2,c_3,c_4,……,c_{k-1},c_k} .
END

3.2.1 Ontology-based Clustering Algorithm with Feature Weights

This algorithm entails using the domain knowledge introduced by ontology in calculating the clustering features weight. The building of the domain ontology tree is done using expert knowledge basing on the given data collection (ZHANG and WANG). To get ontology concepts, deeper abstractions are utilized.

3.2.2 Ontology based FP – Growth Based Clustering

This type of ontology based algorithm carries out two scans on the given database. At first, a list of frequent items are computed and sorted by frequency of appearance in a descending order. Secondly, there is compression of the database into an FP – tree. Recursive mining is then performed on the FP – Tree.

3.2.3 Swoogle

Most of web semantic data is highly distributed thus this facilitates easy and faster access of data on the web. The most challenging is how to facilitate for quality data to users and also and relevant domain ontologies and choosing the trusted one. Swoogle therefore are developed to provide the user or the agent with the semantic web search and navigation framework. It uses three kinds of paths to achieve this, interesource path which enhances links between Semantic Web Terms (SWTs), resource document paths which provides usage links between the Semantic Web Documents (SWDs) and SWTs and finally Inter document paths which manifest explicit link between Semantic Web Documents.

3.2.4 PowerAqua

It's a type of algorithm that supports the users in querying and exploring semantic web content. It answers queries by integrating and locating data or information which is normally distributed within the large scale information in the semantic web documents. The information which is heterogeneous in the semantic it's fully exploited using this mechanism to provide knowledge fusion, query disambiguation as well as ranking mechanisms so as to provide the user with the most answers for the queries. This approach evolved from the AquaLog, which is a system for intranets limited to use one ontology at a time.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

3.2.5 MICS TOOL

The mediator-synthesis module relies on these mappings to generate the mediator. In a first step, we generate the parallel composition of the mapping processes and verify that the overall system successfully terminates using the LTSA (Labelled Transition System Analyser) model checker. In a second step, we are improving the algorithm so as to deal with ambiguous mappings, i.e., when an action from one component may semantically be mapped to different actions from the other component. The MICS tool has been implemented using Java.

3.2.6 Travel Agency

Description

This scenario illustrates the role of ontologies in handling heterogeneity both at application and middleware layers. For this purpose, we consider two travel agency systems that have heterogeneous application interfaces and are implemented using heterogeneous middleware protocols (one is implemented using SOAP and the other with HTTP REST). We use application-specific and middleware ontologies to reason about the matching of both application and middleware behaviour.

The first networked system, called EUTravelAgency, is developed as an RPC-SOAP web service. Thus, data is transmitted using SOAP request and response envelopes transported using HTTP Post messages. The service allows users to perform the following operations concurrently:

- Selecting a flight. The client must specify a destination, a departure and a return date. The service returns a list of eligible flights.
- Selecting a hotel. The client indicates the check-in and check-out dates. The service returns a list of rooms.
- Selecting a car to rent. The user indicates the period of rental and their preferred model of car. The service then proposes a list of cars.
- Making a reservation. Once the user has chosen a flight and/or a hotel room and/or a car, they confirm their reservation. The service returns an acknowledgment.

The interface signature for EUTravelAgency (abstracted from WSDL 2.0) is given below, where we provide only the ontology concepts associated with the syntactic terms embedded in the interface:

- SelectFlight({destination,departureDate, returnDate},flightList)
- SelectHotel({checkIndate,checkOutdate,pref}, roomList)
- SelectCar({dateFrom,dateTo,model},carList)
- MakeReservation({flightID,roomID,carID}, Ack)

The second system is called USTravelAgency and allows users to perform the following two operations:

- Finding a trip. The client specifies a destination, departure and return date. The service finds a list of “packages” including a flight and hotel room and car.
- Making a reservation. The user selects a trip package and confirms it. The service acknowledges the reception of the selection.

The interface signature, although giving only embedded ontology concepts, is abstracted as follows:

- FindTrip({destination,departureDate,returnDate,needCar},flightList)
- ConfirmTrip(tripID,Ack)

MICS takes as input the models of two functionally compatible components together with a domain ontology and produces the mediator that enables them to interoperate. MICS is made up of three modules.

The ontology encoding module first classifies the ontology using the Pellet reasoner.6 Pellet is an open-source java library for OWL DL reasoning. Then, the ontology encoding module uses Algorithm 1 to associate bit vectors to the concepts of this ontology.

The interface mapping module computes the mapping between the interfaces of the components given as input, as described in Section 3.2 using the Choco constraint solver.7 Choco is an open-source java library for constraint solving and constraint programming. Choco does not manage ontology relations such as subsumption but thanks to the bit vector representation of concepts and the associated modelling of constraints.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

IV. SIMULATION RESULTS

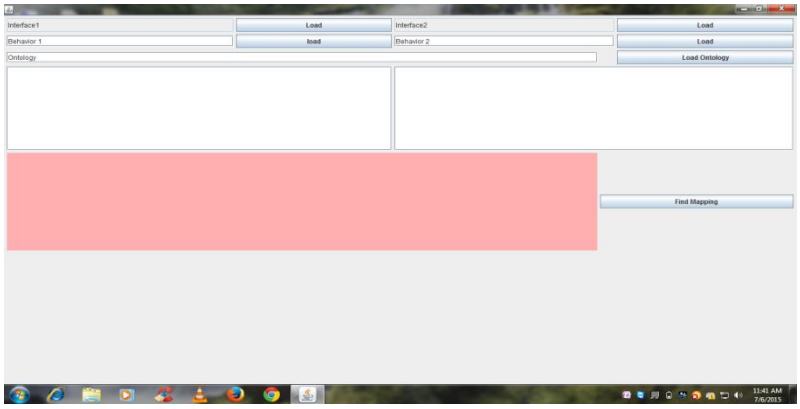

Fig2: MICS TOOL

Fig3: Main Page

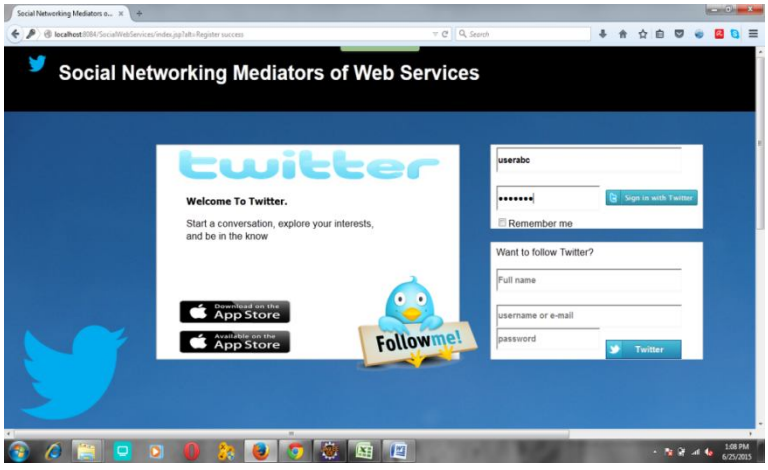

Fig4: Login Details

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

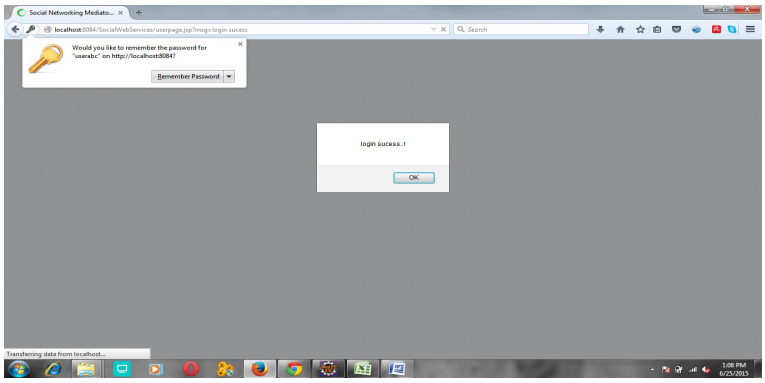

Fig5: Login Successes

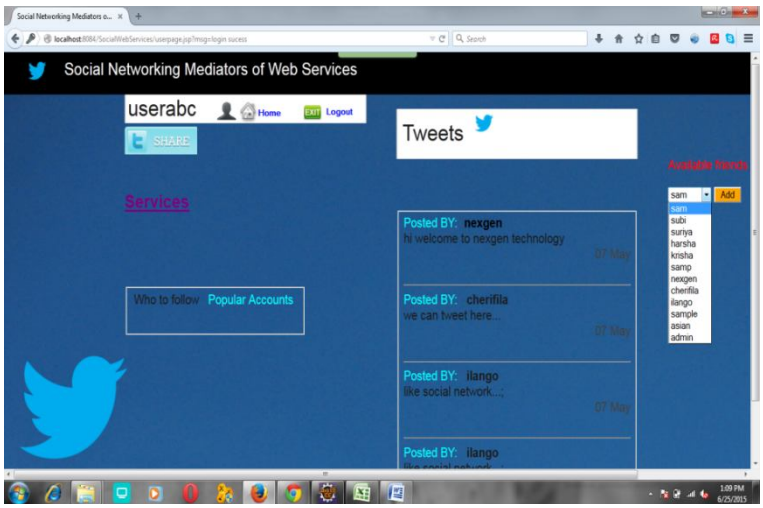

Fig6: Login User Profile Page

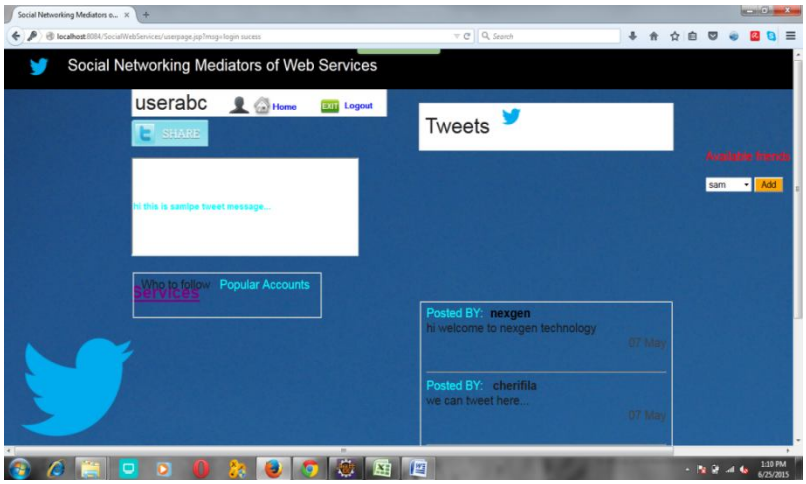

Fig7: To Enter other Services

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

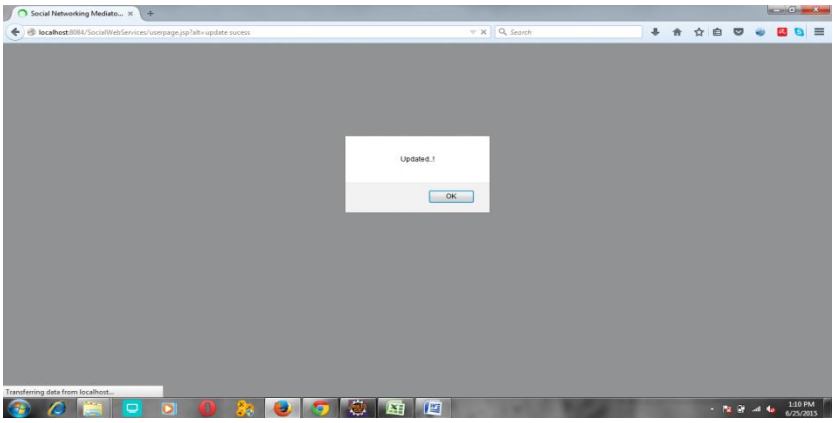

Fig 8: Entering To the Services

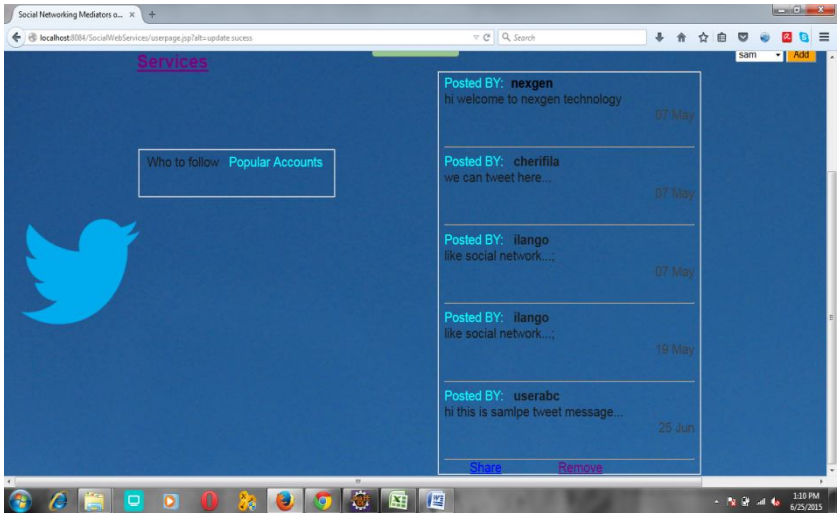

Fig 9: Entering the Services Screen 2

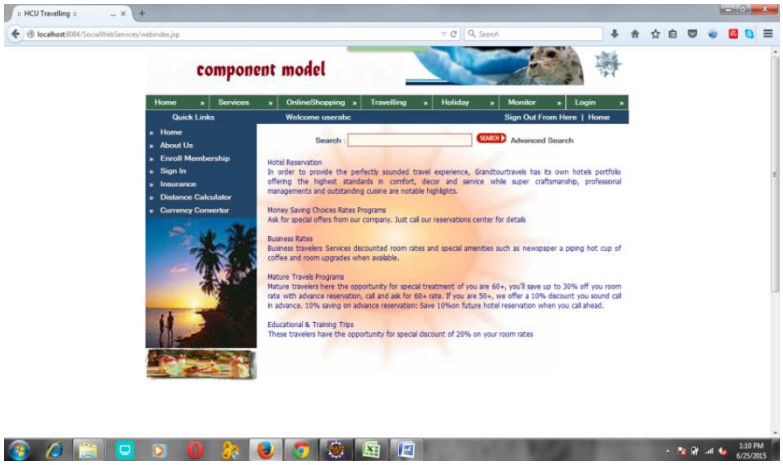

Fig 10: Transform one service to another service

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

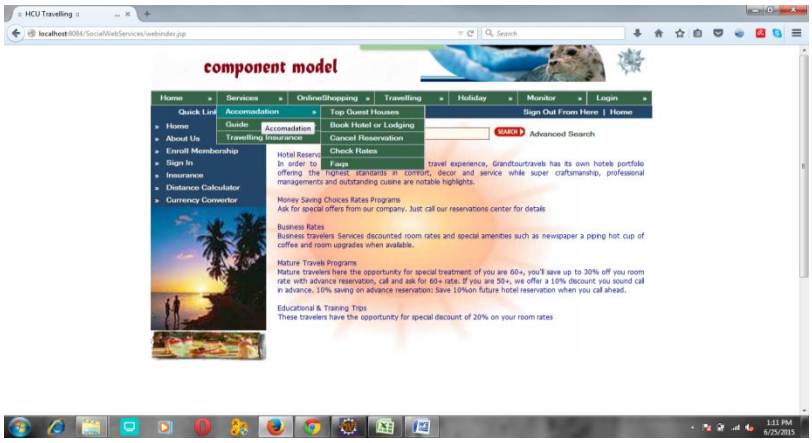

Fig11: We finding the services

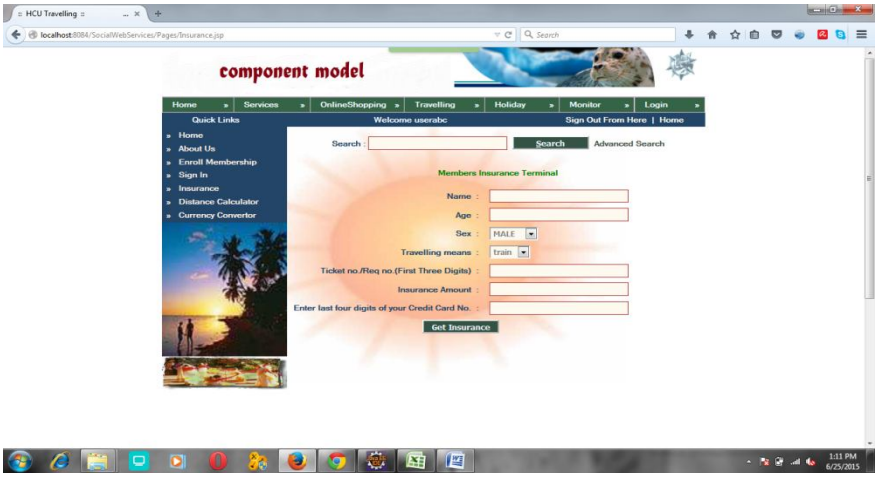

Fig12: Entering our details

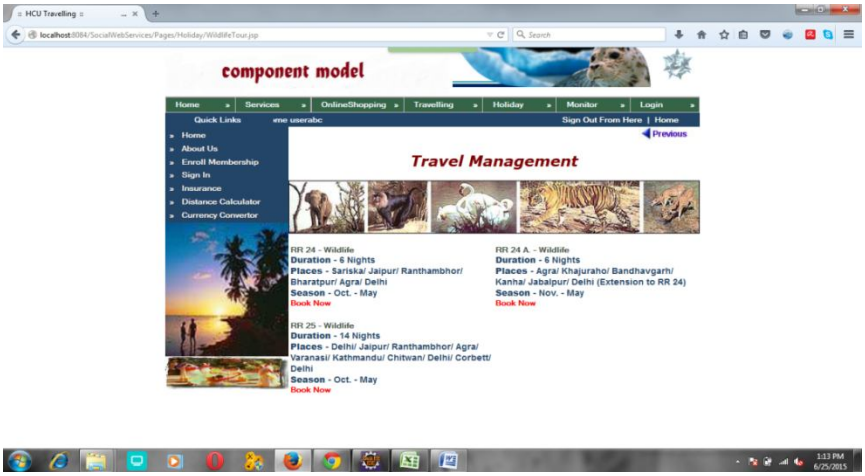

Fig13: Entering another services

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

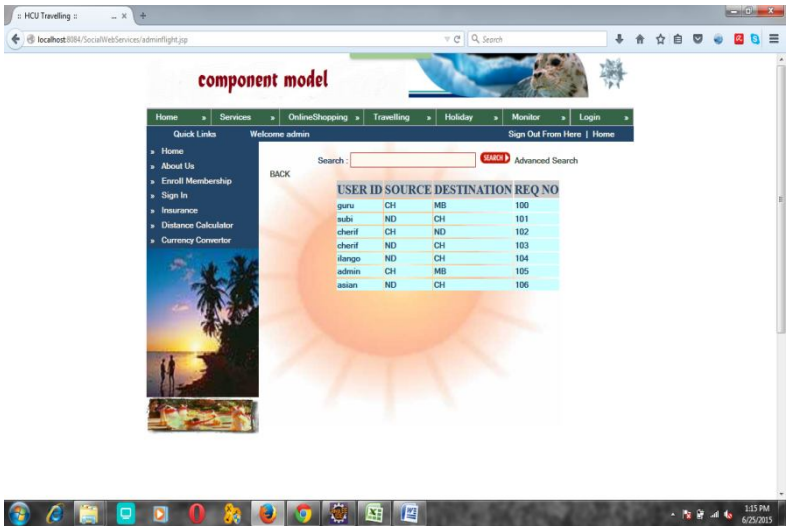

Fig14: Checking the data for user id and requests

V. CONCLUSION

Interoperability may be a key challenge in computer code engineering whether or not expressed in terms of the compatibility of various elements and protocols, in terms of compliance to trade standards or more and more in terms of the flexibility to share and utilize knowledge gathered from totally different systems. The chance of achieving ability between elements while not truly modifying their interfaces or behaviour is fascinating and infrequently necessary in today's open environments. Mediators promote the seamless interconnection of heterogeneous elements by acting the required translations between their messages and coordinative their behaviour. Our core contribution stems from the high-principled automatic synthesis of mediators. During this paper, we tend to conferred associate degree approach to infer mappings between element interfaces by reasoning concerning the linguistics of their knowledge and operations. We tend to then use these mappings to mechanically combine a correctly- construction intermediate. a crucial side of our approach is that the use of ontology's to capture the linguistics information concerning the act elements. This rigorous approach to generating mediators removes the necessity to develop accidental bridging solutions and fosters future-proof ability. we tend to believe that this work holds nice promise for the longer term.

REFERENCES

[1] D. Konstantas, J.-P. Bourrires, M. Lonard, and N. Boudjlida, Interoperability of enterprise software and applications. Springer-Verlag, 2006.
[2] D. Garlan, R. Allen, and J. Ockerbloom, "Architectural mismatch: Why reuse is still so hard," IEEE Software, 2009.
[3] M. Shaw, "Architectural issues in software reuse: It's not just the functionality, it's the packaging," in Proc. SSR, 1995.
[4] V. Issarny, A. Bennaceur, and Y.-D. Bromberg, "Middleware layer connector synthesis: Beyond state of the art in middleware interoperability," in SFM-11, 2011.
[5] M. Autili, P. Inverardi, A. Navarra, and M. Tivoli, "SYNTHESIS: A tool for automatically assembling correct and distributed component-based systems," in Proc. ICSE, 2007.
[6] B. Spitznagel and D. Garlan, "A compositional formalization of connector wrappers," in Proc. ICSE, 2003.
[7] R. Mateescu, P. Poizat, and G. Salaun, "Adaptation of service protocols using process algebra and on-the-fly reduction techniques," IEEE Trans. on Soft. Eng., 2011.
[8] Y.-D. Bromberg, P. Grace, L. R´eveill`ere, and G. S. Blair, "Bridging the interoperability gap: Overcoming combined application and middleware heterogeneity," in Proc. Middleware, 2011.
[9] C. Gierds, A. J. Mooij, and K.Wolf, "Reducing adapter synthesis to controller synthesis," IEEE T. Services Computing, 2012.
[10] G. Wiederhold, "Mediators in the architecture of future information systems," IEEE Computer, vol. 25, no. 3, pp. 38–49, 1992.
[11] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design patterns: elements of reusable object-oriented software. Pearson Education, 1994.
[12] M. W. Maier, "Integrated modeling: A unified approach to system engineering," Journal of Syst. and Softw., 1996.
[13] E. Morris, L. Levine, C. Meyers, P. Place, and D. Plakosh, "System of systems interoperability (sosi): final report," DTIC Document, CMU/SEI, Tech. Rep., 2004.
[14] D. M. Yellin and R. E. Strom, "Protocol specifications and component adaptors," ACM Trans. Program. Lang. Syst., 1997.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

[15] N. Shadbolt, T. Berners-Lee, and W. Hall, "The semantic web revisited," IEEE Intelligent Systems, 2006.
[16] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, The Description Logic Handbook. Cambridge University Press, 2003.
[17] C. Calero, F. Ruiz, and M. Piattini, Ontologies for Software Engineering and Software Technology. Springer-Verlag, 2006.
[18] S. B. Mokhtar, N. Georgantas, and V. Issarny, "COCOA: Conversation-based service composition in pervasive computing environments with qos support," J. of Syst. and Soft.., 2007.
[19] D. L. Martin, M. H. Burstein, D. V. McDermott, S. A. McIlraith, M. Paolucci, K. P. Sycara, D. L. McGuinness, E. Sirin, and N. Srinivasan, "Bringing semantics to web services with OWLS," in Proc. WWW, 2007.