International Journal of General Engineering and Technology (IJGET)
ISSN(P):2278-9928; ISSN(E):2278-9936
Vol. 5, Issue 2, Feb – Mar 2016; 15-30

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

ENHANCING USER ADOPTION FOR SERVICE ORIENTED BUSINESS MODEL THROUGH SEMANTICALLY ANNOTATED AND EXTENDED ASSOCIATION RULE MINING IN SLAKY COMPOSER SYSTEM

P. SANDHYA[1] & M. LAKSHMI[2]

[1]Research Scholar, Department of Information Technology, Sathyabama University,
Chennai, Tamil Nadu, India

[2]Professor and Head, Department of Computer Science and Engineering, Sathyabama University,
Chennai, Tamil Nadu, India

ABSTRACT

All types of enterprises including traditional, e-commerce and web service have to cope up with the changes in their business model. A typical traditional grocery store soon transformed into super market where the customer picks his own choice of products. The customer picks the product only after evaluation of price and data of manufacturing. Web service oriented business models like freemium services have flourished in the recent days. We have word to pdf converter provided as an online service that is free of cost for limited sizes and expects a premium for heavy sized files. Users need to have an adoption to these changes. In India we have number of web based business services like Jabong, Flipkart, Myntra, etc. In fact the customer evaluation of the product to be purchased is done at their doorstep. There are several online garment selling services selling customized products to specific customers. OLX is another popular web based service for resale of products. The entire avenue of business has been transformed to buy products in easy and convenient manner. However these smart business modes have certain hurdles in being adopted by customers. Some serious issues include lack of penetration of internet in certain countries, lack of motivation of the customers to accept changes and security threats. Though the changes in technology promote easy utilization of services the mental barriers preventing user adoption must be cleared through user assistance. Web service composition is the process of construction virtual enterprises on the fly by aggregation of appropriate services. Web service composition can be static or dynamic. In case of static composition services for composition are fixed at compile time and cannot be customized to user preferences. Dynamic composition performs runtime composition and considers handling non determinism of services. Automatic composition is automation of composition by a software agent to promote ease of work for composer. To promote automation of service composition web services are annotated to describe themselves using ontology so that the software agent can decide for selection. In this paper we focus on service composition as a user adopted business model using Association Rule Mining. From service workflow patterns we filter the services that are personalized to user's preferences based on collaborative filtering. To implement user adoption OWL-S upper ontology and FOAF are extended. The novelty of the paper is that it promotes selection of services both from adoption and personalized preference perspective. The paper also works on how to frame association rules for new and upgraded services.

KEYWORDS: Association Rule Mining, Collaborative Filtering, Web Service Composition, Semantic Web Service, OWL-S, SLAKY Composer

INTRODUCTION

Web services are self-contained, self-describing, modular applications that are published, located and invoked across the web. With web services we move from human centric web to application centric web. Web services are the building blocks of Service Oriented Architecture. SOA is a flexible set of design principles used during the phases of system development and integration. Upon deployment SOA based architecture provides a loosely integrated suite of services that can be used within multiple business domains. In the context of SOA service refers to a set of related software functionality, together with the policies that should control their usage. The interfaces and bindings of web services are defined and described in XML. SOA architecture generally comprises of three actors. Firstly the service provider who creates and provides the service. The second actor is the service requestor who requests for the web service by finding from the registry. Thirdly the service registry is where the web services are registered by the service provider and discovered by the requestor. Considering the web service protocol stack the first layer is the transport layer with supporting protocols including HTTP, SMTP, FTP, BEEP, etc. The next layer is the XML-Messaging layer that provides bindings in XML based on SOAP or XML-RPC. Above this is the description layer using WSDL for XML based interfaces. The last layer is the discovery layer using UDDI which is an XML based data store. Web service composition refers to the process of collaborating heterogeneous web services. Automatic web service composition is defined as the construction of virtual enterprises as complex services on demand. A composite web service is created by aggregation of services on the fly. To perform an automation of service selection by software agent the service must describe itself apart from functional description using WSDL. These descriptions are provided as semantic descriptions based on OWL-S upper ontology. In the current research scenario choosing business service partners for composition on the fly is usually determined with user-centric metrics and deficient of provider's business specific metrics. Automatic web service composition in today's real world business circumstances is more of a toy model due to the lack of service provider collaboration metrics. SLAKY System is a new model for selection of business service partners by adding service provider metrics to user centric metrics enabling strategic and realistic selection of services for composition. SLAKY System selects services automatically considering the vision, time planning, environmental context, user adoption, usage policies, trust management, risk management, market scenario, native intelligence, and competitive profit management as service provider collaboration metrics in addition to functionality satisfaction metrics for client's requirements. The time planning metric was designed using opus deviser algorithm and profit management using SLAKY BWG algorithm. In this paper we focus on strategic selection of semantic services based on the metric of user adoption. Though there are several applications running as a business service enterprises running as an automatic composite service applications is still at its infancy. The issues include lack of IT infrastructure in a country, lack of runtime collaboration metrics, lack of service standards, lack of inter country business policies, lack of risk management and security threats. Moreover user adoption to new business models is a slower process for majority of the customers. To harness the benefits of business models based on automatic service composition customers need user assistance. In this paper we focus on service composition as a user adopted business model using Association Rule Mining. From service workflow patterns we filter the services that are personalized to user's preferences based on hybrid filtering.

WEB SERVICE COMPOSITION

Web service composition refers to the process of collaboration the heterogeneous web services. B2B composite applications can be built from composition of services offered by multiple business partners based on business processes. Web service composition is aggregation of elementary web services. Existing web services are composed on the basis of rules of composition to meet a demand which cannot be realized by a single service. Web service composition refers to the integration of more than one web service to realize business functionality. Web service composition is classified based on two types. The first classification is static and dynamic composition. The static and dynamic composition techniques are discussed below.

Static composition – The business partners and service components are chosen at design time. For example assume that a travel package service provider offers facilities for tourist that includes train ticket booking service, composed with hotel reservation service and taxi booking service. There are several train reservation services, hotel booking services and taxi booking services. The appropriate services are chosen at design time and composed statically using workflow languages like Business Process Execution Language. These services are fixed. Here we have an example of three services Account Manager WS, Order Manager WS and Stock Manager WS that are composed statically using BPEL.

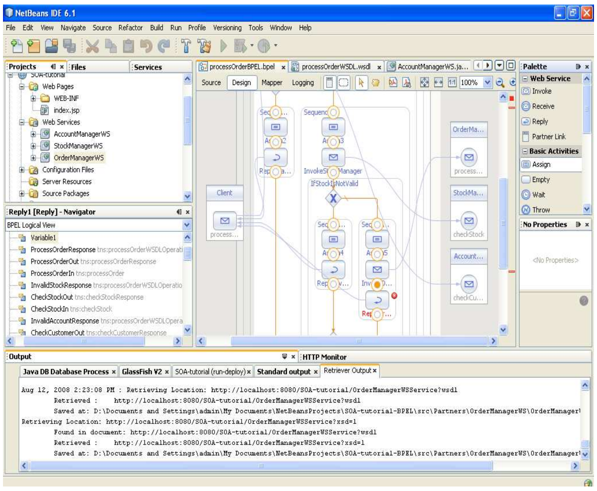


Figure 1: BPEL Static Composition

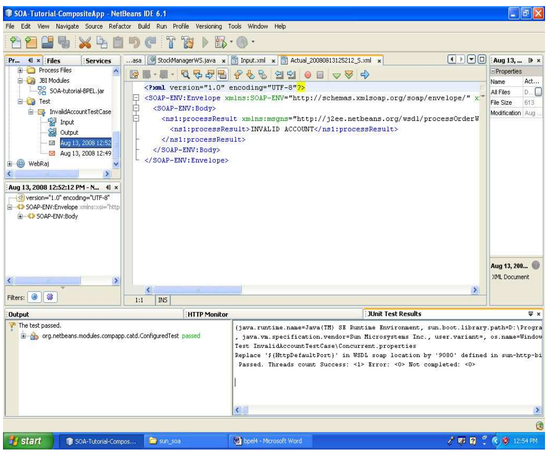


Figure 2: SOAP Response

Dynamic composition – Unlike static composition in dynamic composition the web services have to be discovered at runtime. Services can be chosen based on customer preference, Quality of service, etc. Dynamic composition is highly flexible and also deals with non determinism in the execution environment. In the instance of the travel package at runtime based on certain requirements like user preference or QOS parameters an appropriate web service rendering train ticket reservation, hotel booking and taxi booking can be chosen. Dynamic composition is more beneficial than static composition as the web services are chosen in a customized and qualitative manner.

The second way of classifying web service composition is as manual and automatic. In manual composition flow of processes is hard coded. Automatic composition is guided by dynamic composition and provides ease of composition. To achieve automation of composition several approaches like planning, network analysis, Case Based Reasoning, neural network, genetic algorithm etc. are harnessed. Moreover the services are annotated with service descriptions so that a software agent can do the selection of a service. The services are semantically annotated as OWL-S ontology. A typical example of automatic composition tool is OWL-S Xplan which does automatic composition using the planner Xplan based on Hierarchical Task Network planning.

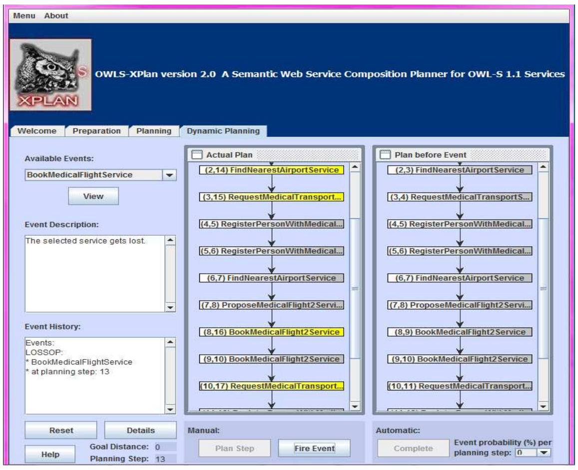


Figure 3: Automatic Composition of Services Using OWL-SX plan

Semantic Web Service Composition using OWL-S upper ontology – Semantic web service annotates semantics to various aspects of web service to automate service discovery, invocation and composition. WSDL describes services at syntactic level requiring human intervention for composition of services. To automate composition meaningfully the semantics of service functionality have to be described using ontology. To obtain service descriptions there are two kinds of ontologies namely Generic web service ontology like OWL-S to specify input, output, precondition and effect and a domain ontology to specify web service domain knowledge such as service parameters and domain ontology. The generic web service ontology has four subontologies namely Profile, Process, Grounding and Service. Profile ontology specifies what a service does, Process ontology specifies how the service works and Grounding ontology specifies how the service is implemented. The Service ontology links ServiceProfile, ServiceProcess and ServiceGrounding which are further specialized as sub-concepts in Profile, Process and Grounding ontology. The profile ontology specifies what the service does as a functionality offered by the service, semantic type of inputs outputs precondition effect, details of service provider, and several service parameters like quality rating and geographic radius. Service discovery is based on this description. For each of the functionality of a specific service there is a profile.

The hasProc relation specifies the association of each Profile instance with the process it describes. Process ontology describes the internal process models of complex services. The process model is used to automatically compose the web services. The Process ontology has a concept called ProcessModel which describes a single Process. The process can be atomic, simple or composite with or without control constructs. The IOPE of Profile and Process are linked by refers To property. Grounding ontology describes the vocabulary to link conceptual description of service specified by Profile and Process to actual implementation details. The implementation details include network protocols, message etc. The WSDLGrounding contains WSDL description that treats Atomic Process as a WSDL operation. Therefore the inputs and outputs of Atomic Process are mapped to corresponding message part in the input message and output message respectively of WSDL operation. The Grounding ontology specializes Service Grounding as a WSDLGrounding. The type of WSDL message part is specified as a OWL-S parameter. Each WSDL Atomic Process Grounding elements of WSDLGrounding grounds an atomic process in Process Model. The semantic description of service is offered by Profile and Process ontology. WSDL offers the syntactic descriptions of the service. The mapping between syntactic and semantic descriptions is given by Grounding ontology. To enrich OWL-S ontology with domain knowledge domain descriptions are specified in domain ontology. The domain ontology is based on OWL and contains a Data Structure and Functionality hierarchy. The functionality hierarchy supports better discovery than a normal key-word based discovery. In this paper we enhance the semantics to include annotations over OWL-S upper ontology to augment user assistance metrics. The augmenting of user assistance metrics can benefit enterprises by enhancing user adoption. User adoption of new business model promotes good return on investment to the enterprises and easy buying for customers.

LITERATURE SURVEY

There are several works done on automatic web service composition. Web service composition was statically done using workflow technique [1]. BPEL4WS is a XML based specification language for specifying processes on web services [2]. Later BPML was used for composition [3]. BPML was originally used in BPM for standard based management of e-business process. BPML provides control flow, data flow and event flow. WSCI is a XML based language that provides a standard for specifying the overall collaboration between the web service providers [4]. WS-CDL is an XML specification that describes the global view of the observable behavior of message exchanges of all the web service participants that are involved in the business collaboration [5]. Dynamic Web service composition was later modeled using AI planning [6]. Mc Ilraith et. al., suggests that the agent knowledge base provides a logical encoding of preconditions and effects of the web service actions in the language of situational calculus [7]. McDermott introduced web service composition based on PDDL [8]. Here he uses a new type of knowledge called value of action. Medjahed [9] used composability rules to check if the services are composable. SHOP2 planner is based on Hierarchical Task Network planning [10].

To map the difference between the concepts that people use and the data that computer interprets web services are annotated with OWL-S or WSMO ontologies. In [11] a framework is proposed which translates BPEL process to a particular type of automata. In [12, 13] Petri nets are used in BPM to capture the various process control flows. [14] Suggests using process algebra to describe, compose and verify web services. Process algebra, automata and Petri nets are used to provide secure composition [15]. [16] Suggests that quality of composition should be maximized as a multi-objective optimization problem with constraints on quality of service and semantic links. The optimization problem is automated in a scalable manner using genetic algorithms. Neural networks were later used to provide user-oriented web service composition [17] and preference aware QOS evaluated web service composition [18]. Finite state machine was proposed in [19] that performs a protocol synthesis problem and automatically generates a target service protocol by reusing the existing one. Lyes Dekar in his work [20] has proposed to achieve dynamic clustering of web services oriented to composition.

The clustering is performed through b-coloring of graphs. An ontology based approach that uses the effects of the web service on its environment entities was proposed in [21]. This is called effect based reasoning. According to [22] association rule mining is used to improve the performance of Semantic Web Service composition. Web service composition is also achieved using case based reasoning [23].

SLAKY SYSTEM ARCHITECTURE

SLAKY is a realistic model for choosing business service partners considering service partner collaboration metrics including vision, time planning [24], environmental context, user adoption, usage policies, trust management, risk management, market scenario, native intelligence and competitive profit management of service partners apart from functionality satisfaction for client's requirements. We have proposed SLAKY architecture because the choice of a service partner should not only satisfy end user's requirements but also meet real world business objectives of the composite application provider. SLAKY composer chooses services such that they give high profit computed by profit management module using SLAKY BWG algorithm [25]. The time planning module was implemented using Opus devisor algorithm. The success of any product is reflected by the adoption of the product by user and industry. In this paper we focus on user adoption during composition. OWL-S upper ontology which describes the semantics for automation web service composition lacks concepts that captures user adoption. Therefore in this paper we extend OWL-S to include certain parameters to define user assistance to achieve user adoption. FOAF ontology is also extended to perform collaborative filtering and content based filtering based on the user's related groups. We also consider new and upgraded services in the due process.

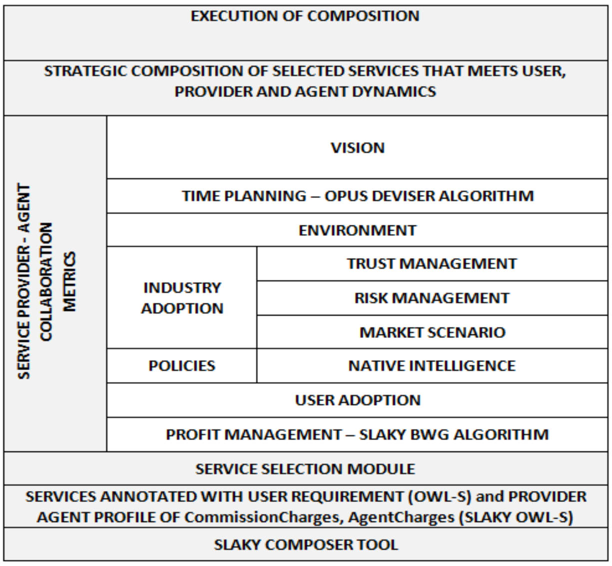


Figure 4: SLAKY Collaboration Stack

SLAKY USER ADOPTION MODULE

The steps involved in SLAKY user adoption module is discussed here:

- The software agent of a composite business model periodically searches the UDDI to find services of specific categories of the composition.
- The set of services includes services that are highly adopted, fairly adopted and poorly adopted. In fact it also contains new services that are yet to be evaluated.
- To select services with best user adoption the agent performs modified Association rule mining on the web server log. To deal with new services the web services are annotated using extended OWL-S specification..
- On performing step 3, if there are no new services then the service compositions with high support and confidence are selected as user adopted services. If there are new services that do not have support, the agent evaluate the new service by executing it once. The test result is compared with that of the service with high, low and medium frequency support of the candidate services of the same category.
- If the new service has good output as the high frequency service then we create a user market by assigning it with a support equal to that of the support of the high frequency service of the same category.

If the new service has average output as the medium frequency service then we create a user market by assigning it with a support equal to that of the support of the medium frequency service of the same category. If the new service has worst output as the low frequency service then we restrict user market by assigning it with a support equal to that of the support of the low frequency service of the same category. If the new service has best output than the high frequency old service then we create a user market by assigning it with a support equal to that of the incremented support of the high frequency service of the same category.

- The agent now has generated service patterns based on association rules with support and confidence that simulates user adoption.
- To provide personalized service the agent then performs a hybrid filtering on the user adopted service workflow patterns.
- Collaborative filtering is done based on the group preferences related to the user including school, college, workplace and locality of residence. The semantics for collaborative filtering is annotated in extended FOAF (Friend of a Friend) ontology.
- The selected services are composed and executed.

SYSTEM ARCHITECTURE AND IMPLEMENTATION

UDDI Browsing

The SLAKY composer (software agent) performs periodic UDDI search to retrieve services that form the composite application. The UDDI search is done periodically to include newly added and upgraded services. However certain services have already been adopted by the users. The agent then performs a periodic association rule generation a by applying Apriori algorithm on the web service log set. The rules reflect the frequctly used patterns of services which inturn justifies the compositions adopted by the users. When certain service is called frequently it suggests that the users are ready to adopt the service.

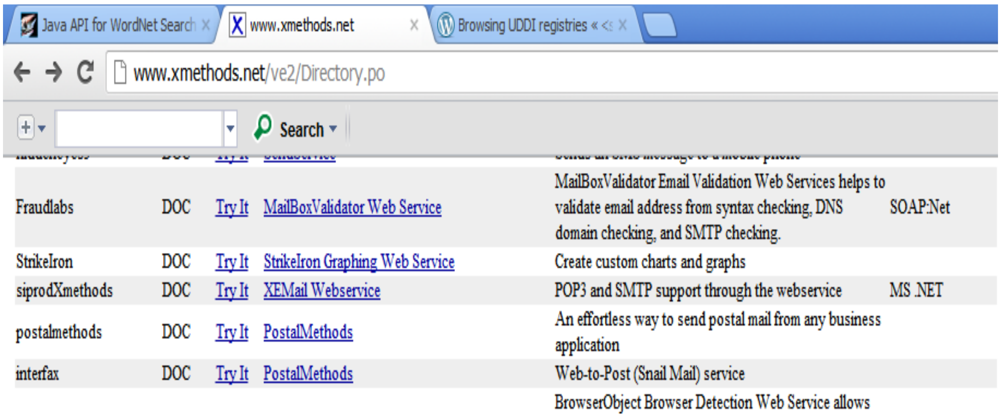


Figure 5: UDDI Browsing

User Adoption Based Association Rule Mining by Modified Apriori Algorithm

In this paper we select services adopted by the user. To analyze user adoption we can program using association rule mining. However ARM does not consider new services for user adoption study. In this paper we have modified ARM Apriori Algorithm to consider new services during mining of user adoptions. To evaluate the equivalence of a new service with existing service we compare Input, Output and Result of both the services. The Association semantics between the outputs can be hyperAssociate, superAssociate, subAssociate or partialAssociate. All of these object properties are functional.

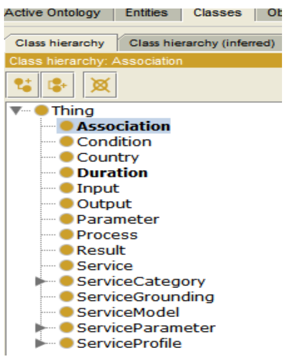


Figure 6: Extended OWL-S profile Ontology

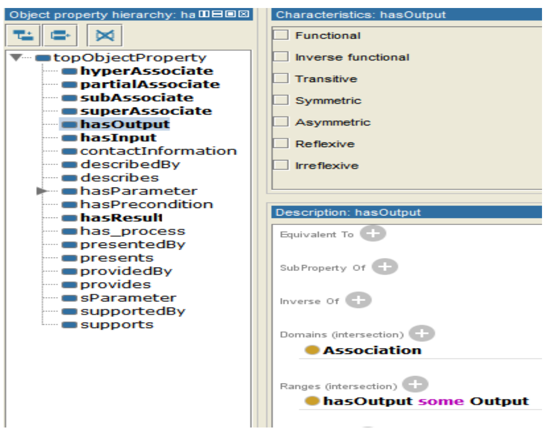


Figure 7: Extended Object Properties – Association and Output

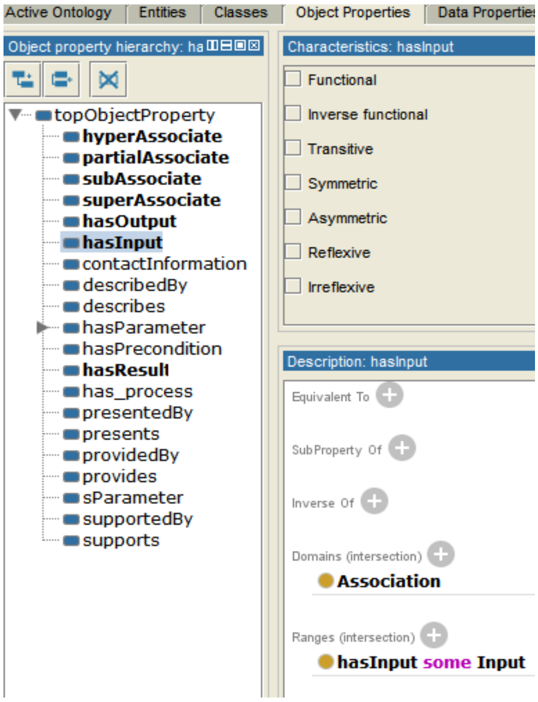


Figure 8: Extended Object Properties – Association and Input

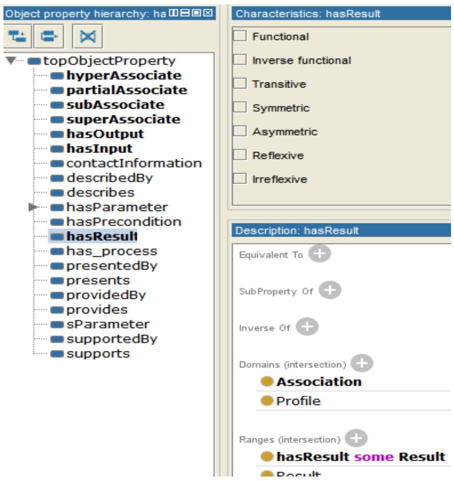


Figure 9: Extended Object Properties – Association and Result

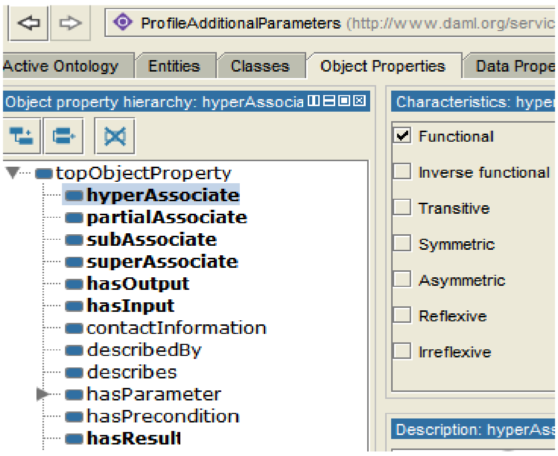


Figure 10: Extended Functional Object Properties

Perform Modified Apriori Algorithm

Input

D, a log of compositions;

min sup, the minimum support count threshold.

Output

L, frequent composition sets in D.

Method

L_1 = find frequent 1-compositionsets(D);

for (k = 2; $L_{k-1} \neq \varphi$; k++) {

C_k = apriori gen(L_{k-1});

for each composition t ∈ D {// scan D for counts

C_t = subset(C_k , t); // get the subsets of t that are candidates

for each candidate c ∈C_t

c.count++;

L_{new} = {c ∈ c_k | c.count=1 && c.status=”new”}

if(c.Output > c_x.Output && c_x = max(support))

replace c.count=c_x.count++;

elseif(c.Output == c_x.Output && c_x = max(support))

replace c.count=c_x.count;

elseif(c.Output == c_x.Output && c_x = avg(support))

replace c.count=c_x.count;

else(c.Output < c_x.Output && c_x = min(support)

replace c.count=c_x.count;

} L_k = {c ∈ C_k | c.count >= min_sup}

}

return L = U_k L_k U_{new} L_{new};

procedure apriori gen(Lk 1 :frequent (k 1)-itemsets)

for each compositionset $l_1 \in L_{k-1}$

for each compositionset $l_2 \in L_{k-1}$

if (l1 [1] = l2 [1]) ^(l1 [2] = l2 [2]) ^..............^(l1 [k 2] = l2 [k 2]) ^(l1 [k 1] < l2 [k 1]) then {

c = l_1 X l_2; // Cartesian product – non transitive to generate candidates

if has infrequent subset(c, L_{k-1}) then

delete c; // prune step: remove unfruitful candidate

else add c to C_k ;

}

return C_k ;

procedure has infrequent subset(c: candidate k-itemset;

L_{k-1} : frequent (k - 1)-compositionsets); // use prior knowledge

for each (k - 1)-subset s of c

if s $\notin$ L_{k-1} then

return TRUE;

return FALSE

Service Personalization by Collaborative Filtering

To achieve service personalization in addition to selection of user adopted service we perform Collaborative filtering based on the group preferences related to the user including school, college, workplace and locality of residence. The semantics for collaborative filtering is annotated in extended FOAF (Friend of a Friend) ontology.

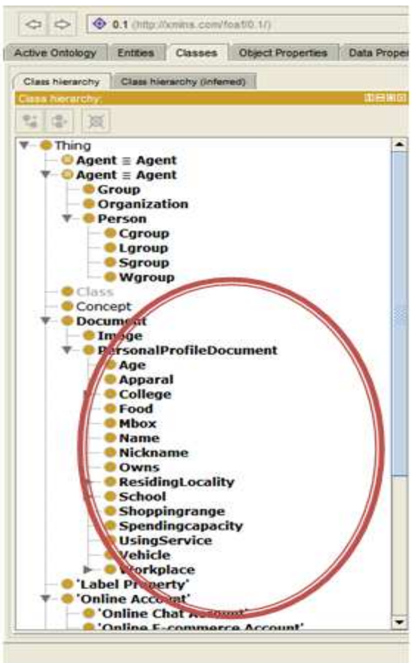


Figure. 11. Extended Concepts in FOAF

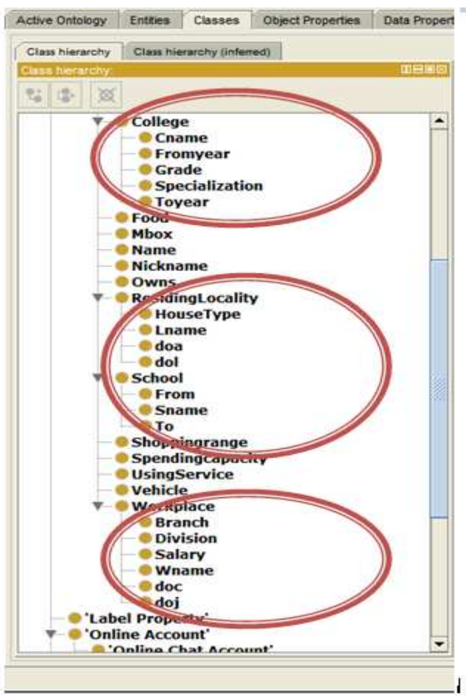


Figure. 12. Extended Concepts in FOAF

Figure 12: Extended Concepts in FOAF

FOAF Based Service Personalization

The FOAF ontology is extended to include user's social network like workplace, locality, school, college, etc. The possible composition set is notified to the client. The client can choose composition of any length. The Inputs are given by the client accordingly. Based on the length of composition user adopted frequent – length composition set is retrieved. From the frequent composition set using FOAF extensions collaborative filtering is done to choose user personalized services. Therefore services are selected in a two step manner including user adopted services and user personalized services. We propose collaborative filtering based on supervised classification. The user adopted composition sets are clustered based on supervised classification. The class labels are considered as social network metrics. They are school, college, workplace, locality etc. Based on these metrics classification of data objects into sub groups is done. Thus services are also chosen from personal preference reflected by social network metric apart from user adopted services.

Implementation

The ontologies were extended from OWL-S upper ontology using protégé_3.4.8 and OWL-S plugin for user adoption. FOAF ontology was extended to implement and capture semantics of user personalization.

TESTING AND CONCLUSIONS

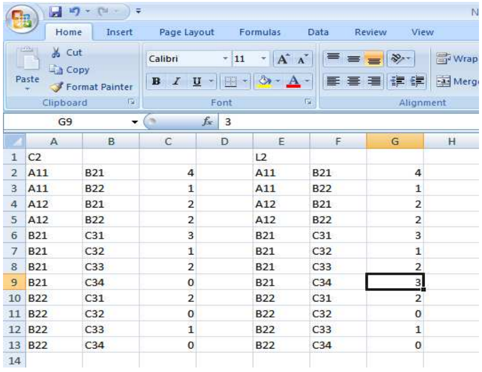
Many web services are not used profusely due to lack of adoption. Our approach finds user adopted services using extended Apriori association Rule Mining. The frequently used composition sets are generated and given to the next step where user preferred services are extracted based on related social network preference. The most novel thing about the paper is that apriori considers associations for new services. The associations are then semantically captured using extended profile ontology of OWL-S. The semantic annotations of user preference are close to his social network and is captured using extended FOAF.

	A	B	C	D
1	Log1	A11	B21	C32
2	Log2	A12	B21	C31
3	Log3	A12	B22	C31
4	Log4	A11	B21	C31
5	Log5	A11	B21	C31
6	Log6	A12	B22	C31
7	Log7	A11	B21	C33
8	Log8	A12	B21	C33
9	Log9	A11	B22	C33
10	Log10			C34
11				

Figure 13: D Composition Set

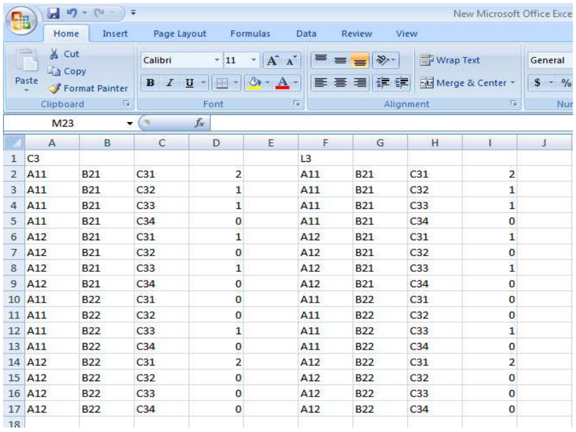
	A	B	C	D	E	F
1	C1			L1		
2	A11	5		A11	5	
3	A12	4		A12	4	
4	B21	6		B21	6	
5	B22	3		B22	3	
6	C31	5		C31	5	
7	C32	1		C32	1	
8	C33	3		C33	3	
9	C34	1		C34	4	
10						

Figure 14: Frequent 1 – Composition Set Including New Items



	A	B	C	D	E	F	G	H
1	C2				L2			
2	A11	B21	4		A11	B21	4	
3	A11	B22	1		A11	B22	1	
4	A12	B21	2		A12	B21	2	
5	A12	B22	2		A12	B22	2	
6	B21	C31	3		B21	C31	3	
7	B21	C32	1		B21	C32	1	
8	B21	C33	2		B21	C33	2	
9	B21	C34	0		B21	C34	3	
10	B22	C31	2		B22	C31	2	
11	B22	C32	0		B22	C32	0	
12	B22	C33	1		B22	C33	1	
13	B22	C34	0		B22	C34	0	
14								

Figure 15: Frequent 2 – Composition Set Including New Items



	A	B	C	D	E	F	G	H	I	J
1	C3					L3				
2	A11	B21	C31	2		A11	B21	C31	2	
3	A11	B21	C32	1		A11	B21	C32	1	
4	A11	B21	C33	1		A11	B21	C33	1	
5	A11	B21	C34	0		A11	B21	C34	0	
6	A12	B21	C31	1		A12	B21	C31	1	
7	A12	B21	C32	0		A12	B21	C32	0	
8	A12	B21	C33	1		A12	B21	C33	1	
9	A12	B21	C34	0		A12	B21	C34	0	
10	A11	B22	C31	0		A11	B22	C31	0	
11	A11	B22	C32	0		A11	B22	C32	0	
12	A11	B22	C33	1		A11	B22	C33	1	
13	A11	B22	C34	0		A11	B22	C34	0	
14	A12	B22	C31	2		A12	B22	C31	2	
15	A12	B22	C32	0		A12	B22	C32	0	
16	A12	B22	C33	0		A12	B22	C33	0	
17	A12	B22	C34	0		A12	B22	C34	0	
18										

Figure 16: Frequent 3 – Composition Set Including New Items

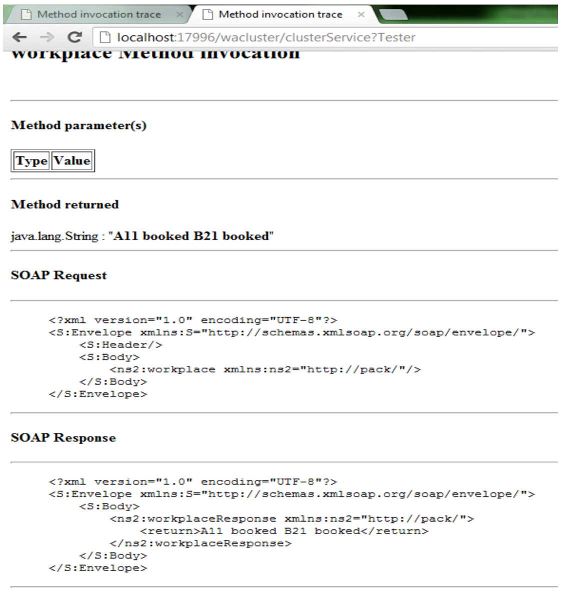


Figure 17: Composition and Executed Based on User Preference

REFERENCES

1. Jinghai Rao and Xiaomeng Su, A Survey of Automated Web Service Composition Methods, Norwegian University of Science and Technology, Department of Computer and Information Science, N-7491, Trondheim, Norway, {jinghai,xiaomeng}@idi.ntnu.no

2. T.Andrews et al. *Business Process Execution Language for Web Services (BPEL4WS)*, Version 1.1. http://ifr.sap.com/bpel4ws/index.html.
3. BPML.org. *Business process Modeling Language (BPML).* http://www. bpmi.org, 2002.
4. A.Arkin et al. *Web Service Choreography Interface 1.0*, 2002. http://www.w3.org/TR/wsci.
5. N.Kavantzas. *Web Services Choreography Description Language Version 1.0*. W3C, April 2004.
6. Joachim Peer, Web Service Composition as AI Planning - a Survey, March 22, 2005
7. S. McIlraith, T., C. Son, and H. Zeng. Semantic Web services. IEEE Intelligent Systems, 16(2):46–53, March/April 2001.
8. D. McDermott. Estimated-regression planning for interactions with Web services. In Proceedings of the 6th International Conference on AI Planning and Scheduling, Toulouse, France, 2002. AAAI Press.
9. B. Medjahed, A. Bouguettaya, and A. K. Elmagarmid. Composing Web services on the Semantic Web. The VLDB Journal, 12(4), November 2003.
10. D. Wu, E. Sirin, J. Hendler, D. Nau, and B. Parsia. Automatic Web services composition using SHOP2. In Workshop on Planning for Web Services, Trento, Italy, June 2003.
11. S.I. Feldman, M. Uretsky, M. Najork, and C.E. Wills, editors, *Proceedings of the 13th International Conference on the World Wide Web (WWW'04)*. ACM Press, New york, NY, 2004.
12. B. Kiepuszewski, A.H.M. ter Hofstede, and W.M.P. van der Aalst. Fundamentals of Control Flow in Workflows. *Acta Informatica*, 39(3):143, 2003.
13. P. Wohed, W.M.P. van-der Aalst, M. Dumas, and A.H.M. ter Hofstede. Analysis of Web Services Composition Languages: The Case of BPEL4WS. In I.-Y. Song, S.W. Liddle, T.W. Ling, and P. Scheuermann, editors, *Lecture Notes in Computer Science*, pages 200{215. Springer-Verlag, Berlin, 2003.
14. G. SalaÄun, L. Bordeaux, and M. Schaerf. Describing and Reasoning on Web Services using Process Algebra, IEEE International Conference on web services, 6-9 July, 2004, pg 43-50.
15. Maurice H. ter Beek, Antonio Bucchiarone, and Stefania Gnesi, A Survey on Service Composition Approaches: From Industrial Standards to Formal Methods, Technical report. 2006
16. Freddy L´ecu´e., Optimizing QoS-Aware SemanticWeb Service Composition, ACM, 2009
17. Yiqiang Li, Zhiliang Zhu, Yu Dai, Bin Zhang, A User-Oriented Web Services Composition Based on Service Mapping Mechanism, International Conference on Advanced Computer Theory and Engineering (ICACTE 2009), 10.1115/1.802977, paper 107, 2009.
18. Xuyun Zhang and Wanchun Dou, Preference-aware QoS evaluation for cloud web service composition based on artificial neural networks, WISM'10 Proceedings of the 2010 international conference on Web information systems and mining, Springer-Verlag Berlin, Heidelberg ©2010, ISBN:3-642-16514-1 978-3-642-16514-6.

19. Ramy Ragab Hassen, Lhouari Nourine and Farouk Toumani, Protocol based web service composition, Service-Oriented Computing – ICSOC 2008, Lecture Notes in Computer Science, 2008, Volume 5364/2008, 38-53, DOI: 10.1007/978-3-540-89652-4_7

20. Lyes DEKAR, A graph b-coloring based scheme for Composition Oriented Web Services Abstraction: COWSA, International Conference on Service Oriented Computing - ICSOC , 2008

21. Puwei Wang and Zhi Jin, Web service composition: An approach using effect based reasoning, Service-Oriented Computing ICSOC 2006 Lecture Notes in Computer Science, 2007, Volume 4652/2007, 62-73, DOI: 10.1007/978 3-540-75492-3_6

22. Shahab Bayati, Ali Farahmand Nejad, Sadegh Kharazmi, Ardeshir Bahreininejad, Using Association Rule Mining to Improve Semantic Web Services Composition Performance, IEEE Press, Karachi, Pakistan. February, 2009.

23. Benchaphon Limthanmaphon and Yanchun Zhang, Web Service Composition with Case-Based Reasoning, ADC'2003. pp.201~208.

24. P. Sandhya, Dr. M. Lakshmi, A Novel Approach for Realizing Business Agility through Temporally Planned Automatic Web Service Composition Using Network Analysis", Worldcomp '11, (SWWS'11), LasVegas, USA on July 18-21, 2011, **CSREA Press, U.S.A, ISBN: 1-60132-202-X, Pg No: 24-31,** http://www.lidi.info.unlp.edu.ar/worldcomp2011-mirror/sww2789.pdf

25. P. Sandhya, Dr. M. Lakshmi, Strategic Composition of semantic web services using SLAKY composer, WEST-2012, Advances in Intelligent Systems and computing, July 13-15, 2012, pg. 411-421, Adnavces in computing and information technology, Springer, ISBN978-3-642-31599-2, ISSN2194-5357, Available online Springerlink.com