Research Paper Computer Science

Enhancing Service Provider's Profitability and Determining the Frequency of Selection of Web Services for Composition Using Dynamic Programming Approach

P. Sandhya | Research Scholar, Sathyabama University

Dr. M. Lakshmi | Head, Department of Computer Science Engineering, Sathyabama University

ABSTRACT

The aggregation of web services to create virtual enterprises is called web service composition. Web service selection is a process of choosing the most appropriate service to execute the task. Most of the web service selection algorithms select the service that is client centric. Services that are meeting client's needs are selected. However the client centric business model considering the requirement of the client alone is a failed model. We propose to select services such that it meets both client and service provider metrics. Service providers being major stakeholders should maximize their profit and minimize its loss. We have developed a stack called SLAKY that suggests the service provider's metrics. We use a dynamic programming approach to find appropriate services for selection. Dynamic programming is a multi stage decision process. The original problem is broken into sub problems which are handled efficiently from the computational viewpoint.

KEYWORDS | Web service composition, Dynamic programming, SLAKY stack

INTRODUCTION

Web service composition is the process of aggregation of services to create a virtual enterprise on the fly. In today's business scenario there is a shift from legacy business model to services based business model. This shift has paved way for composing independent services into related and composed value added services. Web service composition can be classified as static or dynamic composition. Composition of services done at design time is called static composition. The service partners are preplanned and knit using workflow languages like BPEL – Business Process Execution Language. The drawback of static composition is that there is no option of changing the business partners. For example in a travel package the residency hotel service composed can be expensive but the client is forced to choose the expensive hotel service which is not affordable to him. The composition process is not customized. A snapshot of static web service composition knitting AccountManager, StockManager and OrderManager web services using BPEL is shown below.

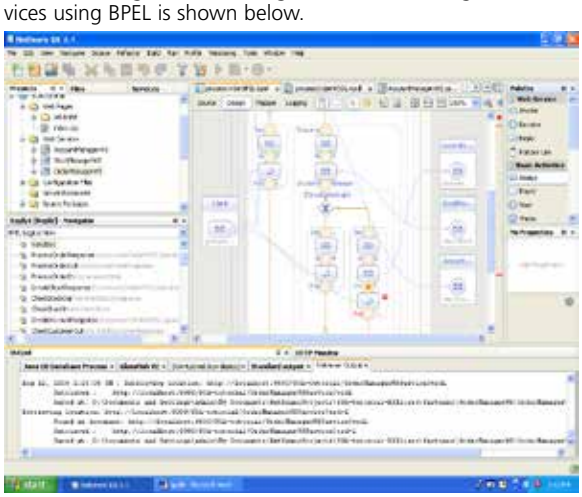


Fig. 1. Static composition using BPEL

Dynamic composition composes services at runtime. The advantage of dynamic composition is the service partners can be decided at runtime. It also deals with the non determinism of services. Consider the following travel package.



Fig. 2. Flight booking service – Yatra.com

The package has an independent flight booking service. Here the client is having a myriad of flight services from which he can choose any service at runtime

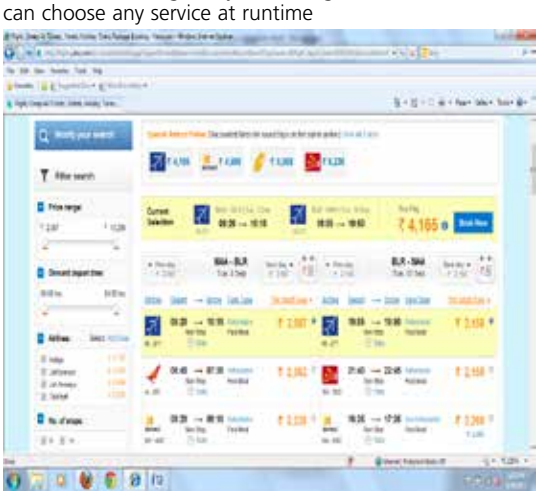


Fig. 3. Flight booking results

Yatra.com also offers composite services that include both flight and hotel booking.



Fig. 4. Service composition in Yatra.com



Fig. 5. Service composition results in Yatra.com

Clients can make a dynamic selection of the results of the composition of flight and hotel service in a customized manner. Web service composition can also be automated for the ease of the programmer. The selection of services is done by an agent. To choose the service agent uses several algorithms. For example OWL-S Xplan is an automatic web service composer based on HTN-planning algorithm.

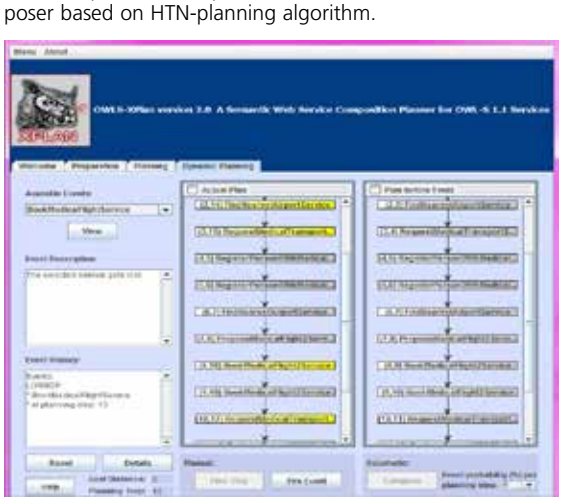


Fig.6. Automatic web service composition using OWL-SXplan

The agent decides the selection of service based on the functionality of the service. The functionality of a service is publicly published using WSDL interface. However, the semantics of the service must also be published for the agent to select a service that best suits the client's requirements. The services therefore should use ontology to advertise its semantics for the agent to select. There are two upper ontologies namely OWL-S (Ontology Web Language for Services) and WSMO (Web Service Modeling Ontology) to describe the services. Services must describe their profile, process and grounding using OWL-S for the agent to determine a service for selection.

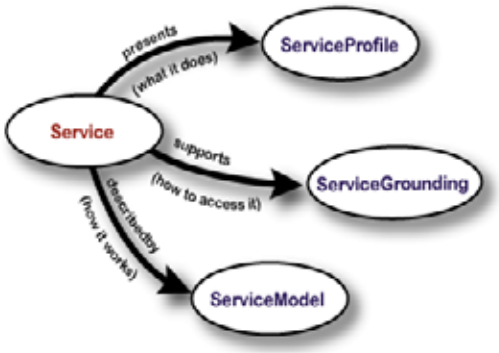


Fig. 7. OWL-S upper ontology [2]

Services that use the ontology for service selection and composition are called as semantic web services. Semantic web service annotates semantics to various aspects of web service to automate service discovery, invocation and composition. OWL-S previously DAML-S is an ontology for services that allows automation of discovery, invocation, composition, etc. OWL-S is an OWL-based Web service ontology, which supplies Web service providers with a core set of markup language constructs for describing the properties and capabilities of their Web services in unambiguous, computer-intepretable form. OWL-S markup of Web services will facilitate the automation of Web service tasks, including automated Web service discovery, execution, composition and interoperation.

Though there are several works done on web service composition most of the work concentrates on providing a good service to the client. However choosing service partners to collaborate on the fly is a Herculean task. Therefore business models today consider business using web service composition as a toy model. Most of the research work on service composition is done considering the client's metrics. Lack of research in choosing service partners considering business collaboration metrics leads to poor adoption of business enabled using web service composition. In this paper we propose a service collaboration stack called SLAKY. The stack provides service provider metrics including vision, time planning, environmental context, user adoption, usage policies, trust management, risk management, market scenario, native intelligence and competitive profit management. In this paper we focus on selecting services for composition as a multi stage decision process. We use dynamic programming to select appropriate services and what is its frequency of execution.

Literature SURVEY

There are several works done on the automatic web service composition. Web service composition was statically done using workflow technique [3]. BPEL4WS is an XML based specification language for specifying processes on web services [4]. Later BPML was used for the composition [5]. BPML was originally used in BPM for standard based management of e - business processes. BPML provides control flow, data flow and event flow. WSCI is an XML based language that provides a standard for specifying the overall collaboration between the web service providers [6]. 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 [7]. Dynamic Web service composition was later modeled using AI planning [8]. 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 [9]. McDermott introduced web service composition based on PDDL [10]. Here he uses a new type of knowledge called value of

the action. Medjahed [11] used composability rules to check if the services are composable. SHOP2 planner is based on Hierarchical Task Network planning [12]. To map the difference between the concepts that people use and the data that the computer interprets web services are annotated with OWL-S or WSMO ontologies. In [13] a framework is proposed which translates BPEL process to a particular type of automata. In [13, 14] Petri nets are used in BPM to capture the various process control flows. [15] Suggests using process algebra to describe, compose and verify web services. Process algebra, automata and Petri nets are used to provide secure composition [16]. [17] 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 [18] and preference aware QOS evaluated web service composition [19]. The finite state machine was proposed in [20] that performs a protocol synthesis problem and automatically generates a target service protocol by reusing the existing one.

Lyes Dekar in his work [21] 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 [22]. This is called effect based reasoning. According to [23] association rule mining is used to improve the performance of Semantic Web Service composition. Web service composition is also achieved using case based reasoning [24]. Paper [25] suggests that as Web Services proliferate, it becomes more difficult to find a service that can perform a given task, and to coordinate with several services. The algorithm follows different strategies to prune the space of possibilities while minimizing the evaluation cost. The algorithm includes DP-BF that combines a best first strategy with a dynamic-programming technique and produces good Web Service compositions by exploring a small portion of the search space. Paper [26] suggests that there are several candidate services to carry out one task in a composite service, so a choice needs to be made to help users select the most suitable one. Hence Weighted Multistage Graph for composite service was proposed to attain good quality of services by transforming the problem of service selection into the one of how to get a longer path. Paper [27] states that research in business models based on web service composition concentrates more on client metrics than service provider's metrics. Hence business models based on web service composition is considered as a toy model. In this paper we therefore define a collaboration stack consisting of service providers metrics so that service partners can collaborate at runtime in a realistic manner. This stack is called as SLAKY collaboration stack. In this paper we focus on determining optimal selection web services and its frequency of selection.

SLAKY COLLABORATION STACK

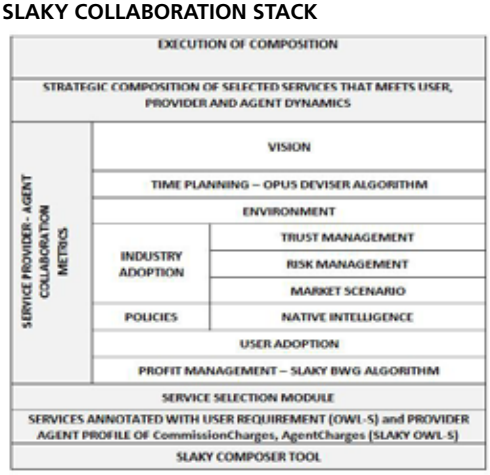


Fig.8. SLAKY Collaboration Stack

SLAKY is a realistic model for choosing business service partners on the fly by considering service partner collaboration metrics including vision, time planning, environmental context, user adoption, usage policies, trust management, risk management, market scenario, native intelligence and competitive profit management apart from the functionality satisfaction of client's requirements. In SLAKY composer competition management module [28]; time planning module [29]; native intelligence module [30] and user adoption module [31] have already been implemented. In this paper we determine the frequency of the service selection to enhance profitability for the service provider using dynamic programming.

DYNAMIC PROGRAMMING

The values of decision variables do not change for situations including allocation, transportation, assignment, scheduling and planning. Thus these problems are static in nature. In certain situations the decision variables vary with time. These situations are dynamic in nature. The techniques dealing with dynamic problems are called dynamic programming. Dynamic programming is a mathematical technique dealing with the optimization of multistage decision problems. In this approach decision regarding a certain problem will be optimized in stages rather than simultaneous approach. The original problem is broken down into sub problems which can be handled more efficiently. There can be a situation when a series of decisions with the outcome of each depends upon the results of the previous decision in the series. The technique of dynamic programming aims at optimizing the situation as a whole and the decisions for the stage may be sub-optimal. The decision making process consists of selecting a combination of plan from a large number of alternative combinations. So before making a decision it is required that:

All the decisions of a combination are specified before a decision is evaluated.

The optimal policy can be selected only after all the combinations are evaluated.

The enumeration method is time and effort consuming. All combinations may not satisfy the limitations; the number of combinations may be too large to allow exhaustive listing. The advantages of dynamic programming are that it allows multistage decision making. In dynamic programming the outcome of the decision depends upon a small number of variables. A stage decision does not alter the number of variables on which the outcome depends but only the changes the numerical value of these variables. An optimal policy has the property that whatever the initial stage and decision are the remaining decisions must constitute optimal policies with regard to the state resulting from first decision.

In dynamic programming the problem is broken down into sub problems referred to as a stage. At each stage there are a number of alternatives and the decision-making process involves the selection of one alternative which is called as stage decision. Stage decision may not be optimal but the overall decision for the entire problem is optimal. The variables which specify the condition of decision process and summarize the current status of the system are called state variables. The number of state variables should be less to reduce complexity. During the decision making at each stage the decision is made to change the state of the problem with the aim of maximizing the return. At the next stage decision are made using the values of state variables those results from the previous stage decisions.

WEB SERVICE SELECTION AS A DYNAMIC PROGRAMMING PROBLEM

Automatic web service composition is also a multistage decision problem. The values of decision variables do not change for situations including allocation, transportation, assignment, scheduling and planning. Thus these problems are static in nature. In certain situations the decision variables vary with time. These situations are dynamic in nature. In this paper we use dynamic programming to perform the automatic web service composition. Using dynamic programming approach we find the optimal pos-

sible combinations of services and the frequency of executing the service composition. Web service composition is an aggregation of related services. Let us assume that there are three domains of services. The services are flight ticket reservation, hotel booking and taxi booking. As there are three domains the application is a three stage problem. Assume that there are four services for each domain of service. We construct a table that contains the payoff of web service selection statistics of each service of a domain pertaining to the frequency per week. The payoff for flight booking, hotel booking and taxi booking should be explicitly stated. The client specifies the budget. Our goal is to find the optimal combination of the three domains of services in the travel package. In this paper we also determine the frequency of selecting the combinations of services [32].

ALGORITHM FOR WEB SERVICE SELECTION USING DYNAMIC PROGRAMMING PROBLEM

Algorithm: To select combination of services and frequency to maximize the return of service selection with expenditure not exceeding the specific limit of budget specified by client

Inputs:

Table of payoffs for each service X weekly frequency, Cost allocated for each domain, Client request for a service domain, budget specified by client

Output:

Optimal service composition, Frequency of execution of service compositions

Step 1: Decompose the service selection problem into i number of stages where i is the stage in consideration.

where i=1,2,3…n

where n is total number of service domains

Step 2: Search the UDDI and find the alternative services in each domain i. Let the number of alternatives in each domain be j.

where j=1,2,3…m

Let us assume that there are equal number of alternatives m in each domain i

Step 3: Let each service S_{ij} in a domain i have an associated pay-off.

pay-off (P_{ij}) =

Service cost X Number of service selections + Profit

Step 4: Construct a table such that it is populated by pay-off of services X weekly frequency

Step 5: The capital for each service domain is C_i where i=1,2,3..n, C_i is the state variable

Step 6: Let the budget specified by the client be I

Step 7: Start a loop for each stage i

If C_i is the capital allocated to stage i and $R_{ij}(P_{ij})$ is the return corresponding to the cost P_{ij}, then the optimal return is

$fi(X_i) = Max [R_{ij}(P_{ij})] + f_{i-1}(X_{i-1})$

This equation is used at various levels of expenditure. The various alternatives are evaluated and the one that gives the largest expected sales is selected.

Step 8: Repeat the loop until i=n

Step 9: End the loop

Thus we have extended dynamic programming for automatic web service composition with optimal payoff and frequency. The algorithm can be applied in business models based on web service composition

SYSTEM ARCHITECTURE AND IMPLEMENTATION

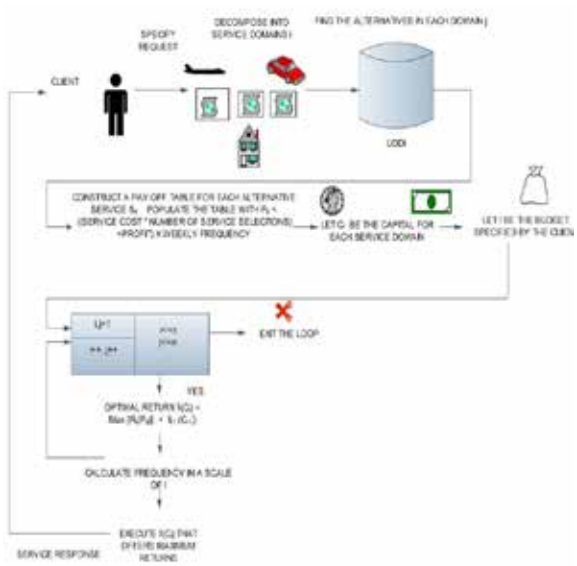


Fig.9. System Architecture

As specified earlier let us assume that there are three domains of services. The services are flight ticket reservation, hotel booking and taxi booking. As there are three domains the application is a three stage problem. Assume that there are four services for each domain of service. We construct a table that contains the payoff of web service selection statistics of each service of a domain pertaining to the frequency per week.

	A	B	C	D
1		pay-off		
2	Frequency/Week	Flight	Hotel	Taxi
3	1	220	150	100
4	2	275	250	175
5	3	325	300	225
6	4	350	320	250

Table.1. Pay-off table

The capital for flight booking, hotel booking and taxi booking are Rs. 2000, Rs.1000 and Rs. 500. The client specifies Rs. 4500 as the budget for a week. Our goal is to find the optimal combination of the three domains of services in the travel package.

In the first stage various alternatives for flight reservation are evaluated and the alternative with largest expected payoff is selected.

$fi(X_i) = Max [R_{ij}(P_{ij})] + f_{i-1}(X_{i-1})$

	A	B	C
1	Stage 1, State -1	Capital=2000 (Return in thousands)	Frequency
2	0	0	0
3	500	0	0
4	1000	0	0
5	1500	0	0
6	2000	220	1
7	2500	220	1
8	3000	220	1
9	3500	220	1
10	4000	275	2
11	4500	275	2

Table.2. Stage 1

The capital for flight service is Rs. 2000/-

When X_1 =0, 500, 1000 and 1500 then it is not possible to select flight.

When X_1 =2000, 2500, 3000 and 3500 then the flight service is selected once. The return is Rs. 220000/-

When X_1 =4000, 4500 then the flight service can be selected twice. The return is Rs. 275000

	A	B	C	D	E	F	G	H
1	Stage 2, State -2				Capital=1000			
2		0	1	2	3	4	Return	Frequency
3	500	0	0	0	0	0	0	0
4	1000	0	150	0	0	0	150	1
5	1500	0	150	0	0	0	150	1
6	2000	220	150	250	0	0	250	2
7	2500	220	150	250	0	0	250	2
8	3000	220	150+220	250	300	0	370	1
9	3500	220	150+220	250	300	0	370	1
10	4000	275	150+220	250+220	300	320	470	2
11	4500	275	150+220	250+220	300	320	470	2

Table.3. Stage 2

The capital for hotel service is Rs. 1000/-

When X_2 = 3000, 3500 then four alternatives are possible, one flight service, one flight and one hotel service, two hotel services and three hotel service.

The return of one flight is Rs. 200000, one flight and one hotel is Rs.370000, two hotels is Rs. 250000 and three hotels is Rs. 300000. Hence one hotel and one flight combination are optimal when X_2 = 3000.

When X_2 =4000, 4500 then there are five alternatives namely the flight service is selected twice, one hotel and one flight service is booked, two hotels and one flight, three hotel selections and four hotel selections.

The return is Rs. 275000 for first alternative, Rs. 370000 for second alternative, Rs. 470000 for third alternative, Rs. 300000 for fourth alternative and Rs. 320000 for the fifth alternative. The third alternative is the best alternative. The return of third alternative is (150+220) X 1000=370000/- Hence the third alternative is the optimal option.

Let stage 3 be for booking of taxi service. The capital allocated for taxi service is Rs. 500/- Using taxi is optional to the user similar to hotel service. The payoff in combination with other services can be plotted. Stage 3 table is given below.

	A	B	C	D	E	F	G	H
1	Stage 3 - State 3				Capital=Rs.500/-			
2		0	1	2	3	4	Total sales	Frequency
3	500	0	100	0	0	0	100	1
4	1000	150	100	175	0	0	175	2
5	1500	150	100+150	175	225	0	250	1
6	2000	250	100+150	175+150	225	250	325	2
7	2500	250	100+250	175+150	225+150	250	375	3
8	3000	370	100+250	175+250	225+150	250+150	425	2
9	3500	370	100+370	175+250	225+250	250+150	475	3
10	4000	470	100+370	175+370	225+250	250+250	545	2
11	4500	470	100+470	175+370	225+370	250+250	595	3

Table.4. Stage 3

The capital for taxi service is Rs. 500/- When X_3 =4500 then the maximum expected sales is Rs. 595000/-. The optimal decision is to select three taxi reservations, one hotel reservation and one flight service. In case X_3 =4000 then the optimal policy is to select two taxi booking, one hotel and one flight service that yields Rs.545000 as returns.

DISCUSSION

```
Stage 1:

 Returns: Rs. 275000
Stage 2:

 Returns: Rs. 370000
Stage 3:

 Returns: Rs. 545000
```

Fig.10. Result of returns

The comparison of selecting services with and without dynamic programming is shown below.

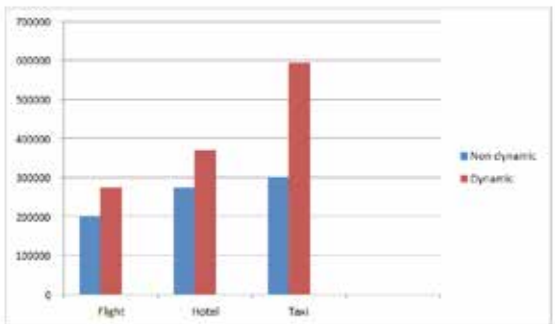


Fig.11. Comparison of service selection with and without dynamic approach

The web service composer will select X_1 =4000, 4500 for flight reservation as it has high returns. In case of hotel booking the third alternative is the optimal selection as the returns is Rs. 470000. Taxi reservation provides optimal selection when X_3 =4500 that yields Rs. 595000 as returns.

Conclusion

In this paper we have used dynamic programming to select optimal services and also the frequency of executing the composition of services. We have created a pay-off table such that the software agent should select optimal services using dynamic programming. The service selection gives expected returns and also meets the client's functionality demands. Web service selection is treated as a multi stage decision-making process that spans time intervals. It is similar to calculus. It also permits one to determine optimal decisions for future time periods regardless of any earlier decisions. Computation technique is hard. As a future enhancement the computation technique must be fine-tuned so that it is simpler.

REFERENCES

[1]http://www.yatra.com/ | [2]http://www.w3.org/Submission/OWL-S/ | [3]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 | [4]T.Andrews et al. Business Process Execution Language for Web Services (BPEL4WS), Version 1.1. http://ifr.sap.com/bpel4ws/index.html. | [5]Owen, BPML.org. Business process Modeling Language (BPML). http://www. bpmi.org, 2002. | [6]A.Arkin et al. Web Service Choreography Interface 1.0, 2002. http://www.w3.org/TR/wsci. | [7]N.Kavantzas. Web Services Choreography Description Language Version 1.0. W3C, April 2004. | [8]Joachim Peer, Web Service Composition as AI Planning - a Survey, March 22, 2005 | [9] S. McIlraith, T., C. Son, and H. Zeng. Semantic Web services. IEEE Intelligent Systems, 16(2):46–53, March/April 2001. | [10]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. | [11]B. Medjahed, A. Bouguettaya, and A. K. Elmagarmid. Composing Web services on the Semantic Web. The VLDB Journal, 12(4), November 2003. | [12] 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. | [13]S.I. Feldman, M. Uretsky, M. Najork, and C.E. Wills, A Survey on Service Composition Approaches From Industrial Standards to Formal Methodseditors, Proceedings of the 13th International Conference on the World Wide Web (WWW'04), New York, NY, pages 621 - 630. ACM Press, New york, NY, 2004. | [14]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. | [15]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, Proceedings of the 22nd International Conference on Conceptual Modeling (ER'03), Chicago, IL, volume 2813 of Lecture Notes in Computer Science, pages 200{215. Springer-Verlag, Berlin, 2003. | [16]G. SalaÃun, L. Bordeaux, and M. Schaerf. Describing and Reasoning on Web Services using Process Algebra, 2004. | [17]Freddy L´ecu´e , Optimizing QoS-Aware SemanticWeb Service Composition, ACM, 2009. | [18] Yiqiang Li, Zhiliang Zhu, Yu Dai, Bin Zhang, I, 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. | [19]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. | [20]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 | [21]Lyes DEKAR, A graph b-coloring based scheme for Composition Oriented Web Services Abstraction: COWSA, International Conference on Service Oriented Computing - ICSOC , 2008 | [22]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 | [23]Shahab Bayati, Ali Farahmand Nejad, Sadegh Kharazmi, Ardeshir Bahreininejad, Using Association Rule Mining to Improve Semantic Web Services Composition Performance, The 2nd IEEE International Conference on Computer, Control & Communication (IEEE-IC4 2009), IEEE Press, Karachi, Pakistan. February, 2009. | [24]Benchaphon Limthanmaphon and Yanchun Zhang, Web Service Composition with Case-Based Reasoning, ADC'2003. pp.201~208. | [25]Eduardo Blanco, Yudith Cardinale, Mar´ia-Esther Vidal, Techniques to Produce Optimal-Web Service Compositions, 2008 IEEE Congress on Services 2008 - Part I | [26]Gao, Jun Na, Bin Zhang, Lei Yang, Qiang Gong, Optimal Web Services Selection Using Dynamic Programming, Yan Northeastern University, P. R. China, Proceedings - International Symposium on Computers and Communications 07/2006; DOI:10.1109/ISCC.2006.116 In proceeding of: Computers and Communications, 2006. ISCC '06. Proceedings. 11th IEEE Symposium on Source: IEEE Xplore | [27]Nikolay Mehandjiev, Freddy Lecu´e, Usman Wajid, Provider-Composer Negotiations for Semantic Robustness in Service Compositions, Proceedings of the 7th International Joint Conference on Service-Oriented Computing, ICSOC-ServiceWave '09, ISBN: 978-3-642-10382-7, pg. 205-220, Springer-Verlag Berlin, Heidelberg ©2009. | [28]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 (2002) | | | | | | | | [29]P. Sandhya, Dr. M. Lakshmi, A Novel Approach for Realizing Business Agility through Temporally Planned Automatic Web Service Composition Using Network AnalysisWorldcomp '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 | [30]P. Sandhya, Dr. M. Lakshmi, Modeling Native Intelligence Semantics For Indigenous Selection Of Web Services Using Slaky Composer, International Journal on Cloud Computing: Services and Architecture (IJCCSA),Vol.2, No.5, October 2012 | [31]P. Sandhya, Dr. M. Lakshmi, Enhancing User Adoption For Service Oriented Business Model Through Semantically Annotated And Extended Association Rule Mining In Slaky Composer System, International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR) ISSN 2249-6831 Vol.2, Issue 4, Dec 2012 5-18 © TJPRC Pvt. Ltd. | [32]S. Kalavathy, Operations Research with C Programs, Third Edition, 2010 |