ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

A Trend Analysis on Web Service Composition

M. Gnana Sagaya Sharmila[1], Dr. S. Justin samuel[2]
Student, Dept. of Information Technology, Sathyabama University, Chennai-600 119, Tamil Nadu, India[1]
Professor, Dept. of Information Technology, Sathyabama University, Chennai-600 119, Tamil Nadu, India[2]

Abstract— The Web is working from the group of pages toward a group of services that making through the internet. A Web Service is a transmission between two devices over the Web. Composition of web services is a good approach for the integration of worldwide enterprise applications functionalities. When a request is given, there may be several web services required and implored to gather into a solution. Hence, the composition graph produced for a given user request may have several candidates with different qualities at each task level. Then, several composition paths may exist to deliver the same solution with different gathered quality. The optimal path should be based on the quality performance. In this paper we propose different methods to find out the optimal path across the all possible paths.

Index Terms— Web Service Composition, different quality.

I. INTRODUCTION

A Web Service is a transmission between two electronic devices over the World Web. Web Services are changing, and the composition cannot meet the real requires of the application. There are several web services which can encounter the same requirements but with different quality parameters. There are several techniques for finding the best web services for a given user request. Nowadays, much research is focus on Web service composition. But composition of web services with same requirements and different quality parameters is a research area. There are many algorithms for finding the best path among the composition path. Multiple compositions may provide same response to the user requirements. The main objective is to find the best composition of services based on the quality parameters like cost, response time, reliability and availability etc.

The best motivating example of the web service composition is tour planner. Assume that there are three tasks in the planning namely, ticket booking, hotel booking, rental Car booking. For each task have several services from several providers may be available. The planner has to find out the best plan according to the user request. The whole problem can be formulated as a complete composition graph.

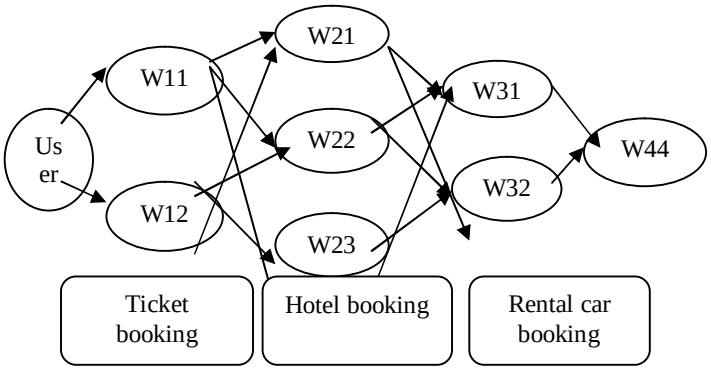

Fig 1.Web service composition problem

II.LITERATURE SURVEY

AUTOMATIC WEB SERVICE COMPOSITION:

In this paper [1] a web service is a program attainable over the web that may change in the world. A main future in the web's development is making services everywhere at once available. As automation increases, these web services will be approached directly by the applications. In this paper, as organizational structure that permits persons to discover, deploy, synthesize and compose services automatically. There are two approaches in the automatic web service composition.

1. Semantic based approach.
2. Syntactic based approach.

1. SEMANTIC BASED APPROACH:

To create services everywhere available we require a semantic based approach. A service is identified by its input variable, the output it produces, and the side-effect(s) it may cause. The input variable needs some pre-conditions, and output variable needs some post-conditions. In the semantic approach, the inputs and outputs, additionally semantic description of the side effect is examined in the matching procedure.

Semantic web service composition approaches [6] can be classified into two categories: Semantic web composition

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

approaches with QOS support and Semantic web composition approaches without QOS support.

A. QOS SUPPORT APPROACHES:
In this section we have presented different QOS support approaches.
1) LECUE ET AL.APPROACH:
There are five different types of links between the input and output [7]. Exact Plug-in, Subsume, Intersection, Disjoint, Causal Link Matrix (CLM) which considers only the functional properties. The author extends CLM to CLM+ to support the non functional properties. After receiving a request, the most suitable service is discovered from the service repository, and then the semantic connections between services are stored on CLM+ which can be used to compute the composition that represents the possible service composition that matches the service request.

2) ARDAGNA ET AL.APPROACH:
Ardagna [8] processes with adaptive web services (PAWS) locate the Business Process Execution Language (BPEL) process with global and local restrictions that introduce to the QOS features. Service Level Agreement (SLA) is used to express the restrictions. Advance service retrieval module discovers the best service that has the required interface and does not disobey the restrictions for each task in the created process. If no service matches the requirement, then the mediator solves the service interface descriptions. . Multiple candidate services are selected for each task and for each process; only one candidate service is executed in the BPEL engine. PAWS also allow the capability services to be replaced with other candidate services and the recovery actions to undo the results of the capability services.

B. NON QOS SUPPORT APPROACHES:
In this section we have presented dissimilar non-QOS based approaches.
1) LEGER ET AL.APPROACH
For effective retrieval of composed service author [9] has integrated DL (Description Logic) reasoning and Situation Calculus along with the extended version of Go log (logic programming language) which is sclGolog. For integration they have used DL reasoning between the input, output parameters (DL based description) of service in order to infer causal links (semantic matchmaking between parameters) along with the situation calculus. The extended Go log (offline interpreter suitable for reasoning about conditional parameters) interpreter can compute conditional web service compositions and is able to elaborate a strategy for automated branching by means of causal links and laws.
2) RAO ET AL.APPROACH:
Rao [10] presents a mixed initiative framework for semantic web service discovery and composition allowing user involvement over many key decisions by suggesting and identifying the inconsistencies. Users can decide that how much to delegate to supporting functionality. The composition engine combines Web Ontology Language (OWL) ontology's with Jess with a planning functionality based on the Graph Plan algorithm. Graph Plan provides reach-ability analysis to determine whether a given state can be reached from another state and disjunctive refinement. Planning is used to propose composition schemas to the user. According to the authors, this is the most realistic approach.

PROBLEMS ON SEMANTIC WEB SERVICE:
1. Discovery problem.
2. Composition problem.

Discovery problem is defined as a query seeking a service, automatically discovering a service from the database that matches the query parameters. Composition problem is defined as matching service is not found; the composition problem requires automatically discovering a directed acyclic graph of services that can be produced to get desired service.
When the number of nodes in the graph is equal to one, the composition problem reduces to the discovery problem. When all nodes in the graph have not more than one incoming edge and not more than one outgoing edge, the problem reduces to a sequential composition problem.
Discovery and composition can be viewed as a single problem. Discovery is a simple case of composition where the number of services complicated in composition is exactly equal to one. The characteristic of the engines is correctness, Query execution time, incremental updates, and cost function.
The semantic based discovery and composition engine reported in the following Part.
Many approaches are provided to solve the discovery and composition problem. Our approach is based on the multi-step narrowing solution based on the composition algorithm.

SERVICE COMPOSITION ALGORITHM: (SCA)
The first step is finding the set of compassable services. Part substitution techniques can be used to find the different parts of a whole task and the selected services can be composed into one by applying the correct

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

sequence of their implementation. The correct sequence of implementation can be determined by applying the pre-conditions and post-conditions of the individual service. The aim is to obtain a single Solution, which is a directed acyclic graph of services that can be composed together to produce the requested service in the query.

EFFICIENCY AND SCALABILITY ISSUES:

Correctness: The narrowing algorithm will always find a correct solution in the minimum possible steps.

Pre-processing: The system initially pre-processes the repository and converts all service descriptions into prolog terms. The semantic relations are also processed and loaded as prolog terms in memory. Once the pre-processing is done, then discovery or composition queries are run against all these prolog terms and hence we obtain results quickly and efficiently.

Execution efficiency: After pre-processing the repository, our system is quite efficient in processing the query.

Programming efficiency: The use of Constraint Logic Programming helped us in coming up with a simple and elegant code.

Scalability: The system allows for incremental updates on the repository.

Use of external database: The repository grows extremely large in size, and then saving off results from the pre-processing phase into some external database might be useful.

2. SYNTACTIC BASED APPROACH:

In syntactic approach, the services being sought in response to a query simply have their inputs syntactically match those of the query. Several attempts are examined to build a service discovery, and composition etc. These approaches are (USDL, OWL-S, WSML, WSDL-S) based on the semantic web. WSDL only label the syntactical aspects of the web service.

In this paper [2] service is defined as applying ability for the advantage of another. When a single service cannot satisfy business requirements, we need to create a group of services as a business process in order to attain them. In this paper on the problem of web service composition, the business requirements are described as a query with the known parameters as input and the expected parameters as output. If all input parameters of a Web service are within the output parameters of another Web service, they can be connected. If the chained Web services receive the known parameters as input and make the desired output parameters, it is a solution to the Web service composition problem. This process is called the syntactic matching problem in Web service composition. Well organized Composition algorithms with better running time performance are favored. Planning techniques from AI are currently used for automatic web service composition. Most of the automatic composition algorithms based on AI planning techniques do not scale well when the no of web services increases.

PROBLEM LAYOUT:

Another AI planning technique issues a unique search space where connections between actions and prepositions are expressed. These planning graphs provide a solution to the syntactic matching problem, with possible redundant web services. Instead of discovering a solution by a backward search in the planning graph, we put our efforts into removing redundant web services carried in the planning graph. We have suggested several strategies to elaborate the planning graph. The planning graph can be established in polynomial time. This polynomial time gives a solution to the web service composition problem. We do not attempt to give all possible solutions, but just one that can be discovered in the shortest time. In this paper, business requirements are expressed as a composition request.

PLANNING GRAPH BASED SERVICE COMPOSITION ALGORITHM:

The classical planning graph algorithm uses a backward search for creating final solutions, which is the most time-consuming part of the planning graph techniques. Researchers have been working on improving it. Instead of improving the backward search, we put efforts into removing the redundant web services during the establishment of the planning graph. We still guarantee at least one solution will be kept after we remove these redundant web services.

QOS MODEL IN WEBSERVICE:

To obtain a composition plan, we should first create a QOS model to describe the QOS aspects of web services. To create an appropriate model, service requester and service provider should agree on same definitions to the extent possible. After creating a QOS model, the second step is QOS based on web service discovery and selection. Unfortunately, WSDL only addresses functional aspects of a web service and does not contain any useful description for non-functional requirements.

QOS-BASED WEB SERVICE COMPOSITION:

In this paper [3] web services can enable computer-computer communication in a diverse environment. Many enterprises and corporations supply dissimilar web

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

services to be more responsive and cost-effective. A number of standards and protocols have been planned to use and publish web services over the internet. Some of the commonly used standards are UDDI, SOAP, and WSDL. UDDI is an XML based record that supplies a standard set of specifications for service description and discovery. Web service provider registers their web services into UDDI registries. SOAP is an XML based protocol specification for exchanging information between peers in the decentralized, distributed environment. WSDL is used to describe the interfaces of all web services regardless of the underlying technology.

Web service composition generates new functionalities by aggregating different services based on a specific workflow. When there are more than one candidate web services for a process, there will be various integrations of web services having the same functionality with different quality parameters. For instance, if there are *m* tasks and *n* candidate web services, the number of all possible plans is nm. In general, discovering a composition plan that satisfies a client's QOS requirement is a time-consuming optimization problem. Integrating web services of high QOS values in a logical computation time has been accepted as an important problem of web service composition. We need to discover a composition plan satisfying client's constraints without checking all combinations. This will be impractical even if there are a few services and tasks in the workflow.

FUNCTIONAL CONSTRAINTS:

There are two functional requirements available in the web service composition [11]. These are Quality Constraints and Environmental constraints. The quality constraints consist of 11 quality attributes. Quality attribute is considered for the final composite web service and not for the services contributing to the composition operation. Each of these constraints has a specific and full explanation which is presented in reference [12].Typical QOS factors associated with a web service are executive cost and time, availability, successful execution rate, reputation, and usage frequency. Also, there are other properties other than the above-mentioned factors, such as reliability, security and so on.

A. QUALITY CONSTRAINTS:

From the quality constraints, the sections of cost and security are elaborated here. The details of other section can be found in [12].

1) Cost: The cost of web service composition can be classified into 6 categories.

- **Service cost**
- **Cost of Composition Application**
- **Network cost**
- **Cost of Transaction lost**
- **Web services Replacement cost**
- **Execution price**

2) Security:

Availability: Information/services are available.
Confidentiality: Access to information/service is only offered to operators with permission.
Integrity: Sets of information are not prone to unauthorized changes or corruption.

B. ENVIRONMENTAL CONSTRAINTS:

Besides user preferences, there are other environmental constraints which constitute environmental category include 5
Branches

Time: The time of the service can also be very important. The existence of a time constraint for an applicant requires the service to be offered and delivered at an acceptable time.
Service Creation Time: The services have been produced after a specific date.
Service Update Time: The services have been updated after a specific date and are compatible with newer technology.
Service status: A few of pre composition constraints can be the ability of the service to be free and active.
Geographical Location: The location or geographical situation where the service being offered can be very important to the composer.

PROBLEM LAYOUT:

A problem of web service composition is usually an NP-hard. Several solutions have been suggested so far to solve this problem that one of them such as is based on Leaner programming and some are based on AI. Genetic algorithm is effective approach to solve some kind of hard problem. Our approach uses the genetic algorithm to solve this problem. To escape from local optimums, we present some modifications in crossover, mutation and selection approach.

THE GENETIC ALGORITHM:

In this section, we present our approach to find an optimal web service composition plan. Since the number of all composition plans of this problem is very large (nm), some ideas to improve GA are presented so that it quickly converges the appropriate composition plan. We introduce some idea for initialization, crossover and

ISSN (Online) : 2319 – 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)
On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

mutation of chromosomes. Also the method to escape from local optimum is represented. If the algorithm cannot find the optimal plan in a specific time, without losing best plans of previous step, the algorithm will escape from local optimum.

COMPARATIVE STUDY:

In Planning Graph based Service Composition Algorithm (PGSCA), all requests in the data set can be solved by using full matching composition. Consequently, some simple algorithms may work. However, it should be noted that PGSCA can address both full- and partial matching Web service composition problems. Running time is used as the metric to compare the performance between PGSCA and the Service Composition Algorithm (SCA) .The main idea of SCA is to build a new chain to record a possible solution path for a Web service which can be served as the first element of the solution path. The output parameters of the Web service will be added to a set of available Parameters for the chain. Then the algorithm scans the whole Web service repository to find whether there is a new Web service in which input parameters are in the set of available parameters for a chain. If so, the Web service will be added to the chain and its output parameters will be added to the set of available parameters for the chain. The algorithm continues to scan the repository and may add a new Web service to a chain again. If it cannot find such a Web service in one scan, the algorithm will terminate immediately.

The idea of SCA is clear, but it is not an efficient algorithm because it scans the Web service repository again and again until it cannot find a new Web service that can be added to a Chain. In other words, it does not tell us under what conditions it should be terminated. Moreover, SCA may leave out some solutions since it does not consider all possible composition paths. However, PGSCA will terminate when it reaches the fixed point level of a simplified planning graph. In addition, it can find the composition path if the Web service composition problem has a solution, though it may not be a concise one. Since the number of all composition plans of this problem is very large (nm), some ideas to improve GA are presented so that it quickly converges the appropriate composition plan

III .CONCLUSION

In this survey paper, we have prepared the way of finding the best composition plan among the Multi-Path web service composition. QOS factors are also available in the Multi-Path web service composition. Several approaches are used to find out the best composition plan. This composition plan satisfies the user requirements. All algorithms are based on the sequential pattern. In future we are going to upgrade the algorithm to solve parallel patterns also.

REFERENCES:

[1] Kona, Ajay, Gupta," Automatic Composition of Semantic Web Services" IEEE International Conference on Web Services,2007
[2] Zhen, Yan, "An Efficient Syntactic Web Service Composition Algorithm based on the Planning Graph Model", Proceeding of the IEEE International Conference on Web services,2008
[3] Mohammad, Alrifai, "Combining Global Optimization with local Selection for Efficient QOS-aware Service Composition", Proceeding of the 18th International Conference on World Wide Web ACM, New York, NY, USA-2009
[4] M. Aflame Amir, V. Derhami, M.Ghasemzadeh, "QOS based Web Service Composition Based on Genetic Algorithm" Proceeding of the journal of AI and Data mining, Vol 1, No 2,2013
[5] Osama Kayed Qtaish, Zulikha Bt Jamaludin, Massudi ahmuddin, "Multi-Path QOS aware Service Composition" Proceeding of International Journal of Engineering Research and Applications, Vol .2, Issue 2, Mar-Apr 2012
[6] Furkh Zeshan and Radziah Mohamad," Semantic Web Service Composition Approaches: Overview and Limitations" Proceeding International Journal on New Computer Architectures and their Applications,2011
[7] Lecue, F.Silva , E.pires, and L.F," A FrameWork for Dynamic Web Service Composition. In: Gschwind, T.Pautasso, C.(eds.) Emerging Web Services Technology,Birkhauser Basel, Vol .II,2008
[8] Ardagna, D., Comuzzi, M., Mussi, E., Pernici, B., Plebani, P.: PAWS: A Framework for Executing Adaptive Web-Service Processes. In: Software, vol. 24, pp. 39–46. IEEE, Los Alamitos (2007)
[9] F.Lecue, A.Leger, Delteil, "DL Reasoning and AI Planning for Web Service Composition. In: Web Intelligence,pp.445-453. IEEE, Los Alamitos,2008
[10] Rao, J.Dimitrov, D.Hofmann, P., and Sadeh,N.M.:"A mixed Initiative Approach to Semantic Web Service Discovery and Composition" Sap's guided procedures framework. In ICWS, pp.401-410. IEEE Computer Society Press, Los Alamitos,2006
[11] Amine Achaean saran, Hassan Haghighi," An Approach to Classify Existing Constraints as Inputs for Web Service Composition" Proceeding of the International Journal of Computer Application" Vol 169, No 12, May 2013
[12] Akhavan, sarraf, Amine, "An Approach to Classify Existing Constraints as Inputs for Developing Web Services Composition, MSC Thesis, Shahid Beheshti University, Iran ,2012