NNGT Int. J. on Software Engineering, Vol. 1, July 2014

Dynamic Web Services Composition Based on QoS

Taycir Bouasker
IT Department
Polytechnic School of Tunisia
Tunis, Tunisia
LIP2 laboratory
bouasker.taycir@gmail.com

Mahjour Langar
IT Department
National Engineering School of Tunis
Tunis, Tunisia
LIP2 laboratory
mahjoub.langar@gmail.com

Abstract—**Service Oriented Architecture has received much of interest from architects and decision makers trying to profit from its different great advantages. Within SOA (Service Oriented Architecture), applications are constructed from reusable specialized components available on the network named web services. Web services composition is one of the most useful assets of this paradigm, thus it is a hot topic receiving more and more attention in order to propose better approaches dealing with WSC (Web Service Composition) issues and requirements. Efforts advanced in this topic may be classified according to the degree of automation in its creation or the binding time (design time or runtime). However, web services must, generally, operate in large scale business environment that are seldom static. Despite the great number of studies exploring the dynamic composition, few are those taking into account and dealing with quality of service rate which tend to be volatile. Thus, non-functional properties have to be continuously monitored to select the best web service at invocation time.**

In this paper, we introduce a QoS (Quality Of Service) based web service composition approach and we focus especially in a web service discovery and selection strategy whose goal is to have as much information as possible about available web services and specifically up to date knowledge of current QoS parameters values so that presently the best web service is selected.

Keywords-SOA; Web service; WS Composition; WS Selection; QoS monitoring.

I. INTRODUCTION

Service Oriented Architecture is one of the most significant paradigm for distributed application development and emerges as a considerable approach for IS (Information Systems) governance. Designed with the purpose to support various applications connection and data changing, this approach enables the reusability, i.e., within it applications are constructed from reusable components called services. SOA is more popularized with the evolution of standards such as web services.

Web services are defined as self-describing, self-contained, repeatable business activities or applications made available on a network and having one or more methods that are invoked through standard protocols, which can operate with other web services in order to fulfill a more complex task or process in widely available and easily detectable form. This technology presents the implementation of service oriented computing paradigm as a communication way between applications and data servers, it allows the invocation of various remote functions implemented on distributed and heterogeneous systems using HTTP and XML.

The philosophy of web services goes beyond the technical solution to make some procedures available via the web to present a new paradigm that affects grid computing and the development of software systems. These loosely coupled and interoperable software, standardized by the Organization for the Advancement of Structured Information Standards (abbr. OASIS) and the World Wide Web Consortium (abbr. W3C), provide better development opportunities and ready to use functionalities.

However, usually no single web service may completely satisfy the end user goal, i.e., companies objectives are generally not simple enough to be achieved using a simple one web service so, to meet such complex needs a mechanism for combining multiple services in a new personalized one, richer and more interesting is adopted and known as web service composition. This concept is the focus of many research communities, who propose different approaches to satisfy and deal with composition issues and requirements. The classification of advanced efforts in this topic is related to the degree of automation in the composition creation or to the binding time (at design time or at runtime).

Whether it is static or dynamic, a composition mechanism must satisfy various issues like connectivity, scalability, correctness, and so on. Among the most important composition requirements, we focus on the quality of participating web services as functional properties are no longer sufficient to characterize a single web service or to determine the most appropriate one to achieve desired goals. De facto, the adoption and orientation towards the web service technology incite available web services proliferation and a great number of web services is accessible in the internet, obviously there might have some functionally similar web services. Hence, a distinction strategy is necessary to choose a single web service to integrate into the composition.

DOI : 01.IJSE.2014.1.5

Some works propose to model the desired web service and determine the appropriate business process in a first step. Then, once the complex goal is decomposed in elementary functionalities, a web service is selected according to functional goals and binding for each one and the composite web service is created. This introduce the static composition where the participating components are selected priory, compiled and deployed at design phase, so even if it considers non-functional attributes values, those are not up to date as the selection is done at design time.

Since the service environment is a highly flexible and dynamic one (Schahram Dustdar and Wolfgang Schreiner, 2005), the static composition is difficult to maintain and the use of more adaptive and scalable solutions, i.e., an approach that takes into account the available services, their functionality and goal either before or during the execution of web services is required. Furthermore, in such dynamic environment quality of service (QoS) parameters tend to be volatile, so a web service selected at design time may have a degraded quality or even be no more available during its invocation. So we are more interested in dynamic composition proposals which consist in web services coordination at runtime based on up to date information.

The contribution of this paper is to analyze web service composition process issues and propose a novel dynamic web service composition architecture that ensures continuously most appropriate web service selection based on QoS monitoring system and a WS classification strategy. Thanks to the proposed architecture, the composition is updated, in other words participating services are usually replaced by the best ones feeting functional and non functional requirements.

This paper is organized as follows. In addition to the introductory section, section two includes some dynamic composition approaches where researchers consider the QoS issue. The third section is devoted to the proposed web service composition architecture. Finally, we present works in process, and discuss some perspectives.

II. RELATED WORK AND MOTIVATION

Web service composition is one of the most important advantages of service-oriented architecture but its optimization is still a challenging research area. A great number of researches are advanced in this context. Static web service composition may be too restrictive since new services become available on a daily basis and the number of service providers is constantly growing [1, 2], so dynamic composition is the only means [3] to adapt to those changes. In this section, we overview related works on dynamic web service composition and highlight their limitations that we target to improve in this paper.

Numerous approaches for dynamic web service composition were proposed. Usually authors refers to other areas assets such as network analysis [4], dependency graphs [5], discrete calculations [6] and probabilistic [7], the genetic algorithms [8] and multi-agent system [9, 10], where they draw similarities between the dynamic composition and optimization algorithms and approaches. The best approach is obviously the one that best satisfy the composition requirements including the QoS optimization.

Web service quality is defined as a combination of several parameters that have been raised in several works [14, 15, 16, and 19] and standardization work made by OASIS and W3C consortium where authors give a complete list of well-defined attributes.

Various works are trying to incorporate quality of service properties in dynamic composition [11, 12, 14, and 17] so that the selection phase of web service is based on values of quality of service parameters. Thus the selection module is the key point for a suitable dynamic composition. El Masri propose a web service broker framework to discover and rank available web services based on functional behavior and some non-functional properties, thus creating a dataset of ranked services [14]. In [18], the author using the multi-agents systems transforms the selection process in a complexes tasks allocation problem and proposes a decentralized selection process. In [15], the writer suggests to add a Multicriteria Evaluation Component (MEC) whose role is the evaluation of a set of services in order to generate a set of recommended composite web service. Some works focuses just in one or two parameters that are considered as the most basic and influencing attributes, i.e., in [20] availability and reliability are studied, and authors introduce a way to estimate the probability of a service being up and its failure trend in order to select the appropriate one.

In [21], the focus is mainly on response time, reliability and composability. The team proposes a QoS model for selection and coordination problem which supports dynamic composition, then develop a WS-TSC algorithm to select the services where they used the principle of linear programming transformed QoS parameters formula in web services composition flow into a single objective function to solve the optimization of the selection.

These different efforts result in great number of frameworks and architectures trying to pay attention to the non-functional characteristics of web services. However, generally, the QoS attributes evaluation is done in first steps in order to create a set of web services with their functional and non-functional descriptions with a little difference in QoS parameters chosen in the one approach or the other, and then this created register is used for selecting and composing new services. Nonetheless, the volatile aspect of web services environment imposes those values updating, the selection has to be based on up to date knowledge about the service quality.

Moreover, despite released works and standardization trials of quality of service parameters listing and definitions, some of them have to be more fix, i.e., the response time must

DOI : 01.IJSE.2014.1.5

consider the service provider location and its of the invocation.

In the following section we will present our novel architecture and explain its different components as well as their roles.

III. The Proposed web service composition architecture

Dynamic composition is very advocated by the majority of research on service-oriented architecture and web services, the most explicit and consistent argument is the volatility and the instability of the environment in which they are used. Researchers in this field are convinced by the advantages of dynamic composition in terms of flexibility, maintainability and failure risk reduction of composite services, that a large number of architecture were developed in order to better implement this composition in the sense that it meets most requirements. In this section, we present the proposed architecture.

As shown in Figure1, our architecture includes three major parties: a goal decomposer, a WSSE (web service selection engine) and a WSCE (web service composition engine).

The former component allows the end user goal decomposition into a set of basic objectives that can be achieved through a single web service since this goal is generally a complex task that no single available web service can achieve completely.

Then, the WSSE has as input those resulting elementary functionalities and it has a dual role. In fact, for the first execution, this engine has to loop through these basic goals in order to discover and collect, for each one, web services that meet the functional requirements, save its information in the dataset and launch the QoS attributes measurement so that the best web service able to fulfill the desired task is selected. This optimal selection is guaranteed through a quality management engine QoSME that provide recent QoS information. After that, the WSSE's role is to maintain this ideal selection by continuously monitoring considered non-functional properties of the various web services in order to always have updated information within the database.

Once the selection step is achieved, the picked service is sent to the WSCE. This later part in our architecture allows, as its name suggests, grouping and coordinating selected services in order to gradually achieve and create the composite service.

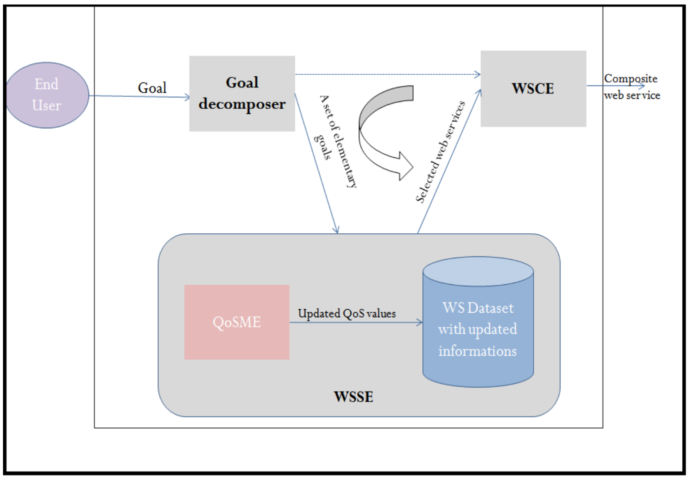

Figure 1: QoS aware dynamic web service composition architecture

A. *Web Service Selection Engine (WSSE)*

The selection module is a fundamental part in web service composition architecture especially with the proliferation of services offering similar functionalities around the web which make the single service selection a more complicated task. Functional properties are no longer sufficient, which justifies and encourages the use of non-functional characteristics or quality of service (QoS) as selection criteria.The suggested selector shown in Figure1 of section 3, includes a dataset where web services information are saved and a quality management engine to evaluate and monitor quality of service parameters.

Web Services Dataset:

The database of web services presents an important component in our architecture since it contains the ready to use services and the basic information about those services. In addition to the functional properties, the description file and the endpoint of each service, this dataset includes quality of service attributes values.

In order to choose the most influencing factors of web service selection, some works were advanced in the literature to analyze each factor. In [26], for example, authors have conducted a study where they refers to grey relational, the grey system theory and the grey relational analysis that, applied to quality of service factors, it shows that best practices, compliance, availability, successability, reliability, latency and response time are the most influencing parameters. Best practices and compliance are related to the web service description and its conformance to web services standards. So measured and monitored attributes in our case are availability, reliability, successability, latency and response time.

The availability has been usually defined as the ratio of upTime to measured time which ignores and neglects the measurement time (when is the availability test made?). The same for latency and response time which refers to duration from the time of sending a request to the time of receiving a response without considering the point of measurement (Where is it measured?).

Moreover, those services are classified to different QoS levels according to latest QoS information.Three classes are distinguished: recommended, acceptable and critical ones.The first one contains services providing best quality value, whereas in the second category we have services with medium and acceptable quality rate, and the last class lists low-quality services.

This classification is very important and useful in two ways: First, it facilitates and accelerate the selection procedure. Then, based on the service class, the monitoring frequency may be moderated, ie, critical-quality services which are more risky ones are supposed to be monitored more than others in order to detect their quality degradation and verify if they are broken or no longer available, while recommended services with high quality require less control. Doing so, we have a faster optimal selection with reduced monitoring cost.

Quality of Service Management Engine (QoSME):
The QoSME is a key element in the proposed architecture shown in Figure1. It presents simultaneously the dataset updater and the web services selector. Seen as the former or the later, this component is in continuous communication with the services dataset.

Having as input a basic functional requirement, this component seeks to find and select the appropriate web service. This service is considered as the best, if it ensures a perfect result and a complete response to the received request. In addition to functional properties, non-functional attributes, that we have already listed above are taken into account during the selection. The evaluation of these parameters of quality is performed continuously with different frequencies (depending on the class of service) in the background. So this engine is rather a daemon process automatically launched. For each QoS parameter a corresponding script is written and for each services class (recommended, acceptable, critical), a monitoring plan and specialized settings are made.

Initially the services found and their non-functional properties are recorded and classified in the web services dataset, for each elementary functionality the best service is selected for the first composition. Then monitoring and evaluation continue to ensure up to date knowledge about web services and maintain the optimality of the selection and so that of the composition and the composite service.

The dataset updating includes the removal of services no longer available, the updating of quality parameters values and therefore the ranking and classification services.

B. *Motivating example: Online trip planning service*

To highlight and show the role of each component in our architecture, we consider as an example a user whose goal is to plan a trip online and make the necessary reservations. No available single service can meet this need, then it is a matter of composing a set of web services that coordinate to achieve the ultimate goal.

That said, a necessary first step to achieve this composition is to identify the basic functional requirements to make the trip planning. This task is ensured by the goal decomposer component in our case. A brief and initial analysis of the ultimate goal, we can already detect three services: a plane ticket booking service, an hotel reservation service and a credit card verification service.

Once basic functionalities are determined, we have to find available services that fulfill those jobs and choose the best web service for each request in order to optimize the composite service quality and best meet the end user goal. This phase is carried out, in our case, with the WSSE and the WSCE. The former undertakes the discovery, quality evaluation and the selection of appropriate services and the later conducts the services coordination.

To explain further the selection phase, we will take as an example the credit card validation service. Searching for available softwares able to verify the credit card number results in a list of web services, which makes the non deterministic choice, why will we choose the one or the other? This non determinism reveals again the necessity of non-functional criteria consideration in the selection module. So QoSME evaluates the list of available services, select the best one to detriment of others and update the dataset with web services information.

As to other services, the same selection procedure is applied. Even the composition is completely achieved, the QoSME stills running to monitor services in order to always offer the best available service, eliminate no longer useful services and keep the latest information and updates about services.

C. *Practical Work*

In our work, we are more interested in the selection phase of the best service currently available. So, after having designed our overall composition architecture (Figure1), we focused on the web service selection engine which contains two components: the WS DataSet and the QoSME respectively to store discovered services and monitor their non functional properties values. We present in the rest of this section the work done so far writing this paper.

WS DataSet:
As explained above, this component's role consists in discovered or collected web services storage.
First, services are functionally structured on two levels, this aiming to facilitate and reduce the appropriate service selection time. Then, in addition to its functional properties and its web address (URL), the considered quality attributes values are recorded for each web service.

DOI : 01.IJSE.2014.1.5

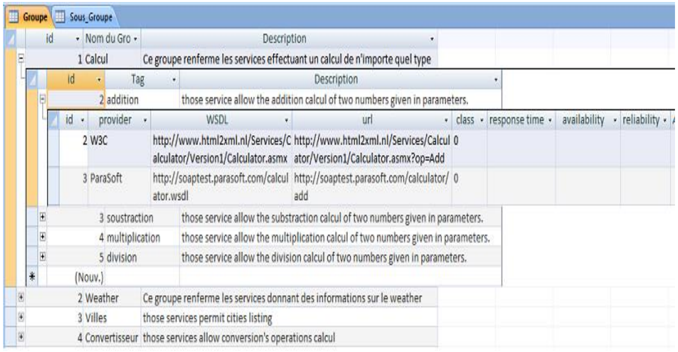

Figure 2: WS DataSet

Initially, these quality values are calculated at discovery phase, then updated continuously thanks to the QoSME whose role is to monitor the stored web services quality level. Depending on its overall quality value, each web service is classified in one of the classes previously defined in the first sub-section of Section III.

Figure 2 shows a screenshot of the data sheet from our database: each service is included in a specific category according to its functionality and defined by its provider, its web address and the response time, availability and reliability values.

The WS DataSet is, continuously, connected and in communication with the QoSME to update its various records and thus ensure and provide the latest quality information.

QoSME:
This second part of the web service selection engine is a key element in our approach. Indeed, it is responsible for the freshness of quality information services at the time of their use, thus it eliminates the risk of invoking a no longer available service or one with degraded quality while another more powerful coexists. So it is essential for optimizing the selection process.

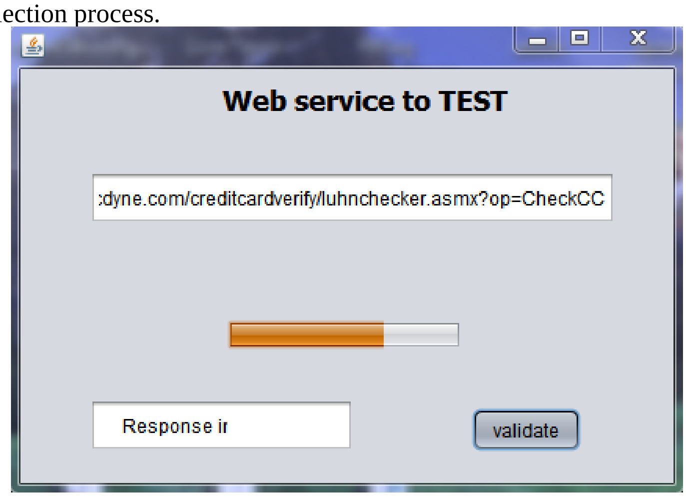

Figure 3: Response Time Test Execution

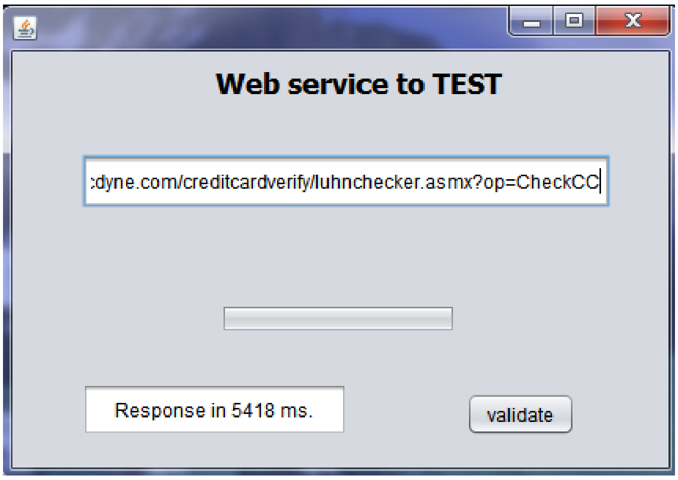

Figure 4: Response Time and Availability Test Result

To implement this element, we have to prepare a script for each quality parameter; those scripts will be executed at different frequencies depending on the web service to evaluate class.

We expose for example, the response time calculation interface shown in Figures 3and 4 which display the time required to get the response from a service whose URL is passed as parameters.

IV. Conclusion

In this article, we introduced our research efforts; we are interested in dynamic web service composition approaches. We conducted extensive research in order to obtain an overview of works done and architectures proposed in this context. Then, we proposed dynamic web service composition architecture composed of three major parts: a goal decomposer, a web service selection engine and a web service composition engine.

We focused in the optimal just in time selection of web services; the selection is done just at invocation times which guarantee the novelty of provided information and reduces the composition failure risk. For this purpose we used QoS attributes measurement and monitoring using a QoS management engine.

Currently, we are implementing the different parts of the proposed architecture, starting with the selection module. We are collecting available web services informations in order to prepare a sufficient number of services in our database to perform our tests since the discovery phase is out of our interest. On the other hand, we are defining the considered QoS attributes with new formulas to reflect the time and the point of invocation factors. We aim to create test and evaluation scripts of different attributes according to the new definitions and achieve the selection module creation.

DOI : 01.IJSE.2014.1.5

Our ultimate goal but not the last is to implement different parts of the dynamic composition architecture proposed above and expose its strengths and advantages via a test phase.

REFERENCES

[1] Jinghai Rao and Xiaomeng Su (2005), A Survey of Automated Web Service Composition Methods, Jorge Cardoso and Amit Sheth (Eds.): In Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition SWSWPC 2004, San Diego, CA, USA, July 6, 2004, Revised Selected Papers, Springer Berlin Heidelberg, pp 43-54.

[2] Schahram Dustdar and Wolfgang Schreiner (2005), A survey on web services composition, Int. J. Web and Grid Services, Vol. 1, No. 1, pp.1–30.

[3] Atif Alamri, Mohamad Eid and Abdulmotaleb El Saddik (2006), Classification of the state-of-the-art dynamic web services composition techniques. Int. J. Web and Grid Services, Vol. 2, No. 2, pp.148–166.

[4] M. Suresh Kumar and P. Varalakshmi (2013). A Novel Approach for Dynamic Web Service Composition through Network Analysis with Backtracking. Natarajan Meghanathan, Dhinaharan Nagamalai and Nabendu Chaki (Eds.): In Proceedings of the Second International Conference on Advances in Computing and Information Technology (ACITY) July 13-15, 2012, Chennai, India - Vol. 3, pp 357-365.

[5] Zhang Hua, Fu Yan, Gao Hui (2012), A Web Service Composition Algorithm Based on Dependency Graph, Yuhang Yang and Maode Ma (Eds.): In Proceedings of the International Conference on Green Communications and Networks (GCN 2011), Springer Netherlands, pp 1511-1518.

[6] Kai Nie, Houxiang Wang, Jiao He (2013), Semantic Web Service Automatic Composition Based on Discrete Event Calculus. In Proceedings of the 2012 International Conference on Information Technology and Software Engineering, Lecture Notes in Electrical Engineering Vol .210, pp 477-485.

[7] Juan Guo, Xianwen Fang, Dawei Zheng, Xiuyun Zhang (2013), Research on Credibility of Web Service Composition Based on Stochastic Petri Net. Zhixiang Yin, Linqiang Pan and Xianwen Fang (Eds.): In Proceedings of The Eighth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), Springer Berlin Heidelberg, Advances in Intelligent Systems and Computing Vol. 212, pp 965-973.

[8] Mingxiang Lu, Haisheng Li, Qiang Cai (2012), A Method of Semantic Web Service Automatic Composition Based on Genetic Algorithm. Wei Zhang (Ed.): In Selected papers from 2012 International Conference on Software Engineering, Knowledge Engineering and Information Engineering (SEKEIE 2012), Advances in Intelligent and Soft Computing Vol. 162, pp 727-734.

[9] Jiajia Li., Bin Li., and Xiaowei zhang. (2012), A web services composition design Framework based on Agent Organization. Physics Procedia Vol. 24, Part C, pp 2004-2009.

[10] Yasmine Charif, Nicolas Sabouret (2013), Dynamic service composition enabled by introspective agent coordination. In Autonomous Agents and Multi-Agent Systems Vol. 26, Issue 1, pp 54-85.

[11] Jigyasu Nema, Rajdeep Niyogi and Alfredo Milani (2012), A Framework for QoS Based Dynamic Web Services Composition. Beniamino Murgante, Osvaldo Gervasi, Sanjay Misra , Nadia Nedjah, Ana Maria A. C. Rocha,David Taniar and Bernady O. Apduhan (Eds.): In 12th International Conference, Salvador de Bahia, Brazil, June 18-21, Part III, pp 528-538.

[12] Demian Antony D'Mello, Manasa Shetty, Akshatha Prabhu, Akshaya and Adithi Shenoy (2012). Optimal Selection of Web Services for Efficient QoS Driven Web Service Compositions: In International Conference, ADCONS 2011, Surathkal, India, December 16-18, 2011, Revised Selected Papers on Advanced Computing, Networking and Security, Lecture Notes in Computer Science Vol. 7135, pp 57-66.

[13] Mihai Suciu, Denis Pallez, Marcel Cremene and Dumitru Dumitrescu (2013), Adaptive MOEA/D for QoS-Based Web Service Composition.Martin Middendorf and Christian Blum (Eds.): In 13th European Conference, EvoCOP 2013, Vienna, Austria, April 3-5, 2013 on Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science Vol. 7832, pp 73-84.

[14] Eyheb Al-Masri (2007), QoS-based Discovery and Ranking of Web Services. In the 16th International Conference on Computer Communications and Networks, pp 529-534 , (13-16 Aug. 2007).

[15] Samir Youssef (2010). Extension de l'Architecture Conventionnelle des Services web : prise en compte de la qualité de service dans la sélection des SW. In Proceedings of the XXVIII Congress INFORSID, Marseille, May 2010, pp 195-210.

[16] Yannis Makripoulias, Christos Makris, Yiannis Panagis, Evangelos Sakkopoulos, Poulia Adamopoulou and Athanasios Tsakalidis (2006), Web Service discovery based on Quality of Service. In IEEE International Conference on Computer Systems and Applications, aiccsa, pp.196-199.

[17] M. Angelaccio, and A. D'Ambrogio (2006), A model-driven framework for managing the QoS of collaborative P2P service-based applications. In Proceedings of the 15th International Workshops on Enabling Technologies : Infrastructures for Collaborative Enterprises, June 2006, pp. 95-102.

[18] Zaki Brahmi and Mohamed Mohsen Gammoudi (2011), Optimal Service Selection for Dynamic and Decentralized Web services Orchestration framework based on Complexes Tasks Allocation Approach. In 4th International Conference on Information Systems and Economic Intelligence SIIE 2011.

[19] Kyriakos Kritikos and Dimitris Plexousakis (2009), Requirements for QoS-based Web Service Description and Discovery. In Services Computing, IEEE Transactions on Vol. 2 Issue 4, pp 320-337.

[20] Rajeev Pratap Singh, K.K.Pattanaik (2013), An Approach to Composite QoS Parameter based Web Service Selection. In the 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013), Procedia Computer Science 19 pp 470-477.

[21] Dong Rang-sheng et al (2010), Dynamic Web Services Composition Based on QoS Model. In Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference, oct 2010,Vol.11, pp 823-826.

[22] Faisal Mustafa, T. L. McCluskey (2008), Dynamic Web Services Composition: Current Issues. In Proceedings of Computing and Engineering Annual Researchers' Conference 2008: CEARC'08, University of Huddersfield, Huddersfield, pp 48-54.

[23] Bassam Al-Shargabi, Alia Sabri, and Asim El Sheikh (2010), Web Service Composition Survey: State of the Art Review. In Recent Patents on Computer Science 2010, 3, pp. 91-107.

[24] Demian Antony D'Mello, V.S. Ananthanarayana, and Supriya Salian (2011), A Review of Dynamic Web Service Composition Techniques. In Advanced Computing Communications in Computer and Information Science Vol. 133, pp 85-97.

[25] V. Portchelvi, V. Prasanna Venkatesan and G. Shanmugas undaram (2012), Achieving Web Services Composition – a Survey. In Software Engineering 2012, 2(5), pp 195-202.

[26] XIAO Xiao-Cong,WANG Xiang-Qun, FU Kai-Yao, ZHAO Yi-Jiang (2012), Grey Relational Analysis on Factors of the Quality of Web Service. In International Conference on Medical Physics and Biomedical Engineering (ICMPBE2012), Physics Procedia Vol. 33, pp 1992-1998.

DOI : 01.IJSE.2014.1.5