ISSN 2229-5518

An Innovative Quality of Service (QOS) based Service Selection for Service Orchrestration in SOA

S.Neelavathi and K.Vivekanandan

Abstract— Service Oriented Architecture (SOA) has become a new software development paradigm because it provides a flexible framework that can help reduce development cost and time. SOA promises loosely coupled interoperable and composable services. Service selection in business processes is the usage of techniques in selecting and providing quality of services (QOS) to consumers in a dynamic environment. Single business process model consists of multiple service invocations forming service orchestration. It represents multiple execution paths called modeled flexibility. In certain cases, modeled flexibility can cause conflicts in service selection optimization, making it impossible to simultaneously optimize all execution paths. This paper presents an innovative approach to service selection for service orchestration that addresses this type of conflicts by encompassing status identification based availability estimation with multiple QOS constraints along with an effective quality assessment model. This model captures the expectations from the users on the multiple quality of a service and returns ratings as a feedback on the service usage. This updated rating in the service list can be used by the new user. This proposed method provides optimal services to users consistently and efficiently thereby resulting in more meaningful and reliable selection of services for service orchestration in SOA.

Index Terms— Service Oriented Architecture, Service Selection, Service orchestration, Meta-metrics, Modeled Flexibility, Rating, Multiple QOS level, local selection, global selection.

1 Introduction

Service oriented architecture (SOA) is a new paradigm for software development that promises loosely coupled, interoperable and composable components called services. Service orchestration is the execution of a single transaction that impacts one or more services in an organization. It is called as business process. Business processes are implemented by orchestrating services of different activities involved in it. Multiple QoS-based service selection results in selecting an optimal service for single activity from a set of candidate services, thereby maximizing the QoS of the entire business process. A single business process model can represent multiple execution paths known as modeled flexibility. Modelled flexibility can cause conflicts in service selection if the different optimal services are selected for the common activity in both the execution paths thereby making it impossible to optimize all execution paths. The proposed approach to service selection addresses this type of conflicts through a set of meta-metrics (probability of execution).

- *K.Vivekanandan is Professor in the Department of Computer Science and Engineering at Pondicherry Engineering College, Puducherry.Mobile: 9443777795 Fax: 2655101 Mail id: K.Vivekanandan@pec.edu.*
- *S.Neelavathi is Assistant Professor in the department of Information Technology, PKIET, Karaikal. Mobile: 9443632464. Mail Id: neelabaski@gmail.com*

Status identification based availability estimation for service selection is used along with multiple QoS constraints. It is extended with an effective quality assessment model that is used to match the expectations from the user with that of rating of services held in service list. The service list is divided into four groups with all the services having the triple factor of quality rating for all the multiple QoS constraints. The feedback from the user after service usage is used to update the service list. This proposed approach of service selection with multiple QoS factors results in more meaningful and reliable selection of services used in service orchestration in service oriented architecture. Along with this, it also resolves the conflicts with modeled flexibility in business process by meta-metrics thereby ensuring selection of optimal services for all the service invocations.

2 Service Oriented Architecture

The service orientation and Service Oriented Architecture (SOA) are not new or revolutionary concepts. It is the next stage of evolution in the distributed computing [2]. SOA is not a Technology and is only an architectural approach. SOA is defined as “Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains [2].” SOA includes the previously proven and successful elements from past distributed

ISSN 2229-5518

paradigms. These elements are combined with the design approaches to leverage recent technology in distributed computing [1].

In SOA, the loosely coupled systems do computing in terms of services. SOA separates functions into distinct units, or services, which developers make accessible over a network in order to allow users to combine and reuse them in the production of applications. These services communicate with each other by passing data from one service to another [3], or by coordinating an activity between two or more services. They use the well established standards [4].

This approach is based on the design principles of loose coupling, which is a principle by which the consumer and service are insulated from changes in underlying technology and behavior, Interoperability[2] the principle which provides the ability to support consumers and service providers that are of different programming languages on, different operating systems with different communication capabilities, Encapsulation that allows the potential consumer to be insulated from the internal technology and even the details of behavior of service, Discoverability which is used to realize the benefit of reuse [5]. Seamless integration of various systems allows data access from anywhere anytime, thereby providing services to customers and partners inside and outside the enterprise. It provides a simple scalable paradigm for organizing large networks of systems that require interoperability and develops systems that are scalable, evolvable and manageable and establishes solid foundation for business agility and adaptability.

2.1 Services In SOA

The most fundamental unit of service oriented solution logic is the service. Services in SOA comprises the below 8 distinct design principles.

- Standardized service contract

  Services express their purpose and capabilities via a service contact. It is the most fundamental part of service-orientation.

- Service Loose Coupling

  Coupling refers to the number of dependencies between modules. Loosely coupled modules have a few known dependencies whereas tightly coupled have many unknown dependencies. SOA promotes loose coupling between service consumers and providers.

- Service Abstraction

  This emphasizes the need to hide as much of the underlying details of a service as possible to preserve loosely coupled relationship.

- Service Reusability

  The same services can be reused in multiple applications. The agnostic nature of services enables them to be recombined and reused in different forms.

- Service Autonomy

  Services to carry out their functionalities consistently and reliably, needs to have a significant degree of control over its environment and resources.

- Service Statelessness

  Services are designed to remain stateful only when required. If there are stateful, then the management of excessive state information can compromise the availability of the service.

- Service Discoverability

  Service discovery is the process of discovering a service and interpretation is the process of understanding its purpose and capabilities.

- Service Composability

  It is related to its modular structure. It is composed in three ways. Application is an assembly of services, components and application logic that binds functions together. Service Federation is the collection of services managed together in a large service domain and Service orchestration is execution of single transaction that impacts one or more services in an organization.

2.2 Elements of SOA

The overall architectural model in the Fig 1 shows the elements of SOA. A service provider describes its service using Web Service Description Language (WSDL). The WSDL definition is divided into two parts: the abstract description that defines the service interface, and the concrete description that establishes the transport and location information. This WSDL definition is published to the Universal Discovery Description Interface (UDDI) service registry. SOAP (Simple Object Access Protocol) is the universally accepted standard transport protocol for messages and represents a standardized format for transporting messages. SOAP message contents are presented in message body which consists of XML formatted data. A service requestor issues one or more queries to the UDDI to locate a service and determine how to communicate with that service. Part of the WSDL provided by the service provider is passed to the service requestor. This tells the service consumer what the requests and responses are for the service provider. The service consumer uses the WSDL to send a request to the service provider. The service provider provides the expected response to the service consumer.

ISSN 2229-5518

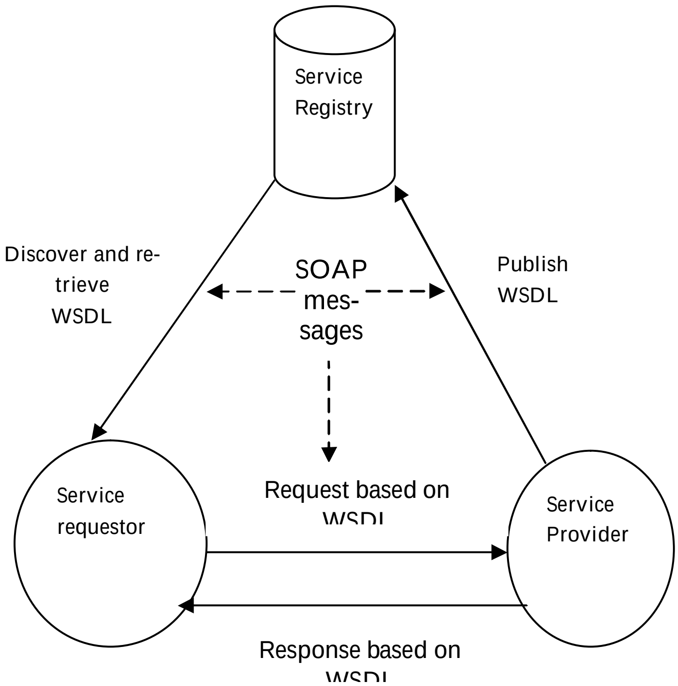

Figure 1 Elements of SOA

3 SERVICE SELECTION IN SOA

It is possible to have multiple service providers in an SOC environment who offer broadly similar services but with differing qualities. It is essential, therefore, that the selection process should select optimal service out of a group of functionality-similar services optimized for a certain property of QoS. The aim of service selection is to provide meaningful and reliable services to consumers consistently and efficiently from the set of candidate services. The figure 2 depicts a typical scenario of service selection. When the selector receives a request from a client, the selector chooses a service to invoke and returns the response message to the client. The selection process can be optimized for a certain property of QoS.

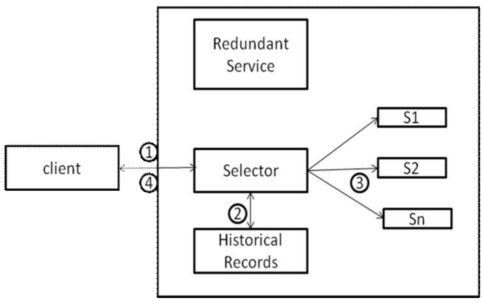

Figure 2 A Typical Scenario of Service Selection

3.1 Quality of service

Quality of service is a combination of several qualities or properties of a service such as Performance, Cost, Reliability, Availability, Reputation, and Fidelity. The performance is the time duration from a request being sent, to when the results are received. Cost refers to the amount of money that the consumer pays for using a service. The reliability is the probability that the requested service is working without a failure within a specified time frame. Availability is the quality aspect of whether the service is present or ready for immediate use. Reputation is the criterion in measuring total trustworthiness of a service and Fidelity is the average marks that are given by different consumers to the same QoS criterion.

Long term relationship between service provider and service consumer generate Service Level Agreement. The service consumer may impose several constraints to be satisfied for the services utilized thereby generating SLA. For Example the cost of service is less than 200$, response time is less than 2ms and service should be available less than 90% of the time etc.

3.1 Classificatiion of service selection methods

Service selection is classified as static service selection and dynamic service selection. The further is selection of optimal services from a set of services with similar functionalities during runtime while the later is selection before the runtime. Another scenario is to select services among several group of services, with each group having similar services called as Local selection and Global selection. Local Selection is to compose a system that can meet certain QoS criteria with selecting one optimal service from each group also referred as service composition whereas Global Selection does not aim at selecting an optimal service in each group, but optimizing the QoS of the composed system. Table I shows the references of static service selection and Table II shows the classification of dynamic service selection methods.

TABLE I
REFERENCES OF STATIC SERVICE SELECTION

static service selection	Pattern recognition based adaptive categorization technique and solution for services selection" IEEE2007.
	A Petri-net based specification model for Web services In Proceedings of ICWS 2004,IEEE
	Automatic Web service composition based on graph network analysis metrics Internet Systems 2005 Springer, 2005.

TABLE II
REFERENCES OF DYNAMIC SERVICE SELECTION

ISSN 2229-5518

TABLE II
REFERENCES OF DYNAMIC SERVICE SELECTION

Based on QoS This selection process selects optimal service out of a group of functionality-similar services optimized for a certain property of QoS.	Under no global constraint	Roland Ukor, Andy Carpenter , *Flexible Service Selection Optimization Using Meta-metrics* Congress on Services-I, IEEE 2009 V.Deora,j.Shao,w.A.Gray, *supporting qos based selection in service oriented architecture* proceedings of the international conference on next generation web services practices , IEEE 2006 Y wang, J Yang,*Relation Based Service Networks for reliable service selection* proceedings of the conference on commerce and enterprise computing,IEEE 2009
	Under single global constraint	Canfora, G., Di Penta, M., Esposito, R., and Villani, M. L.,"*An Approach for QoS-aware Service Composition based on Genetic Algorithms*", Proc. of the 2005 Conf. on Genetic andevolutionary computation, ACM Press, New York, 2005. D.Liu, Z.Shao,C.Yu, *A heuristic Qos-aware service selection approach to web service composition,* International Conference on computer and Information science IEEE 2009 Lingshuang S,Lu Z, et al. *Dynamic Availability Estimation For Service Selection Based On Status Identification,* IEEE International Conference on web Services, 2008 IEEE
	Under multiple global constraint	Bang y, chi-Hung,et al. *Service selection model based on QoS reference vector*, Congress on services , IEEE 2007 *D.A.Menasce et al. On optimal service selection in SOA Performance Evaluation 67* (2010) 659-675 *V.Diamadopoulau et al.Techniques to support Web Service selection* Journal of Network and Computer Applications (2008) D.Liu, Z.Shao,C.Yu, *A heuristic Qos-aware service selection approach to web service composition,* International Conference on computer and Information science IEEE 2009
Based on Semantic web	Achieves the similarity comparison by calculating the semantic distance by QOS and context.	Z Guoping, Z Huijuan, Wang Z *An Approach to QoS-aware service selection in Dynamic Web service composition* IEEE(2007) V.X.Tran et al. *QoS ontology and its QoS-based ranking algorithm for Web services* Simulation Modelling Practice and Theory 17 (2009)
Based on improving protocol or language	Add new actions to standard UDDI to achieve dynamic UDDI process, or design a selecting language like SQL, select Web services by setting restrictive conditions.	Balke W.T, Wagner M, Kim S.M, et (2004) B.Jeong et al. *On the functional quality of service (FQoS) to discover and compose interoperable web services* Expert Systems with Applications (2009)
Based on user preference	Through users' scores on web services, achieves dynamic updating of Web services selection system, thus forming a dynamic selection process with self-evaluation function.	O.Minhyuk, B.Jongmoon,et.al, *An efficient approach for QoS-aware service selection based on a tree-based algorithm* Seventh IEEE international conference on computer and Information science 2008 *TQOS for automatic web service composition* IEEE transactions on services computing (2010)

4 PROPOSED QOS BASED SERVICE SELECTION FOR SERVICE ORCHESTRATION

Service orchestration is the execution of a single transaction that impacts one or more services in an organization, called a business process. In order to maximize the benefits of SOA, service selection is important especially in terms of providing quality of services (QoS) to consumers in a dynamic environment. The proposed work is intended to develop a more effective, meaningful and robust service selection methodology in service orchestration for conflict resolving using Meta metrics .It aims at selecting reliable and optimal services by using more than one relevant QoS category along with an effective quality assessment model.

4.1 Business Process

Business Process consists of multiple activities, each of which is a service invocation. If any of the service invocation fails, the entire transaction should be rolled back to the state that existed before execution of the transaction. Let P be a business process with a set of activities A= {a1, a2,….an}. For each activity $a_i \varepsilon$ A, there exists a set of candidate services S_i = {s1, s2,….sn} where any of the optimal service can be used for the activity. A candidate service is selected based on multiple QoS metrics such as cost, performance, reliability, reputation and fidelity. A solution to the service selection problem is represented as a pair {(ai,si)} which assigns a service $s_i \varepsilon S_i$ for each activity a_i. The Figure 3 shows a business process with four activities. The optimal service is selected for the business dynamically on an instance-by-instance basis.

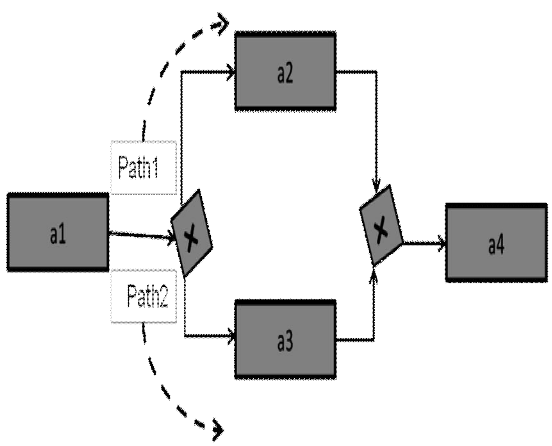

Figure 3 Example Business process

A single business process model can represent multiple execution paths, a phenomenon called modeled flexibility. An optimal solution for path1 may select candidate s_{12} for a_1, while the optimal solution for path2 selects a different candidate s_{11} for a_1. Therefore, it is impossible to obtain a solution that is simultaneously optimal for both paths, resulting in problems with service selection. This raises conflicts with modeled flexibility.

In order, to resolve these conflicts with multiple execution paths in business process, a set of process metrics called meta-metrics is used. They exist solely for the purpose of biasing the evaluation of QoS metrics for candidate services based on priorities. With probability of execution as meta-metric, activities with a higher probability of being executed are given priority to resolve the selection conflicts between the optimal solutions for two or more execution paths.

4.2 STATUS IDENTIFICATION BASED AVAILABILITY ESTIMATION FOR SERVICE SELECTION (SIBE)

The architecture of SIBE is shown in Figure 4. Quality of service assessment model is used to capture the rating from the user for the service invoked. The rating is a triple factor consisting of the expectation from the user for all the five QOS criteria's namely cost, performance, reliability, reputation and fidelity ,along with the provision of actual perceived rating and quality rating offered by the user after using the service. When the user request is made for a service in the form of triple factor, the selector tries to pick up an optimal service from the service list.

The selector maintains four lists of services. Each list represents a separate status stable up, transient down, short term down and long term down. First the selector searches for the optimal service in the stable up list. If the stable up list is empty, the selector tries the transient down list. Finally the selector searches the short term down list and still furthers the long term list. Whenever one service status has been identified, the service is inserted into the appropriate list. The service in the service list is presented along with expectation, actual perceived value and feedback rating for all the QoS attributes from the set of users. Availability which indicates the probability of successful invocation of a service is a key property used in service selection. Multiple QOS criteria such as cost, performance, reliability, reputation and fidelity of the service are taken into consideration.

When the selection process ends, the selector puts the invocation records into historical records and reevaluates the status of the invoked services. The fragmentor algorithm takes invocation records for each service from the historical records and fragments it into equal number of fragments by randomly chosen fragment length. Each fragmented segment consists of results of service invocations in terms of either success or failure. The categorizer uses exponentially weighted average to categorize the services into appropriate list in the service list which is in turn used by the selector to select the optimal service based on user expectation.

The user provides a feedback on the service usage in terms of quality rating for all the QoS criteria's. The service list is updated with the quality rating offered by the recent user. The new service user receives changes from the service list and uses this updated rating for reselecting their optimal services. As the inclusion of rating of many new users will subsequently incur more memory space, provision is made to remove the older entry thereby providing an efficient method of data storage.

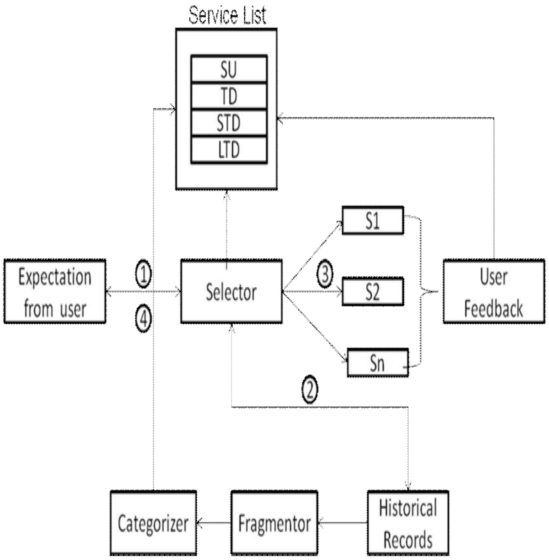

Figure 4 Architecture of SIBE

The figure 5 shows the quality of service assessment model used in SIBE. It captures the rating from the users. The user apart from mentioning the functionalities of the required service is also prompted to specify quality rating for all the QoS criteria's of the intended service. Availability is one of the key QoS attribute. Other QoS attributes considered are cost, performance, reliability, reputation and fidelity. Users specify quality rating in the form of triple factor comprising Expectation from the service, perceived value from the user and the actual rating offered by the user after using the service. The triple factor for cost is denoted as E(c),P(c) and R(c).similarly it is E(per)P(per)R(per) for performance ,E(r)P(r)R(r) for reliability, E(rep)P(rep)R(rep) for reputation and E(F)P(F)R(F) for fidelity.

The service held in four lists consists of triple factor for all the QoS criteria's comprising expectation, perceived value and the actual rating. Among the expectation rating provided by the user for all the QoS attributes, the highest rating of a particular attribute value is taken. Then the particular attribute with highest rating is mapped against the attribute of the service in the service list with the same expectation. If the expectation matches, then the appropriate service from the service list is selected and offered to the user. Meeting the expectation for a single attribute does not mean to satisfy the other attributes to a significant extend. So, the expectation matching can be extended to a maximum number of other attributes, thereby providing a more meaningful and reliable service as the output. The actual rating value i.e feedback provided by the user after the service usage is added into the service list. This updated rating can be used by the user with the same intentions. Further to maintain a fixed set of entries in the service list, the older entries are always removed paving way for new entries of ratings.

Service in service list

User with Expectation	Cost E(c) P(c) R(c)	Performance E(p) P(P)R(p)	Reliability E® P®R®	Reputation E® P® R®	Fidelity E(F) P(F) R(F)
U1	★ 0.3	0.3	♦ 0.3		0.5
U2	0.8	● 0.9		0.8	
U3	★ 0.3		1.0	0.2	0.5
U4		● 0.8	♦ 0.1	0.1	0.3
Average Rating	0.46	0.67	0.47	0.36	0.43

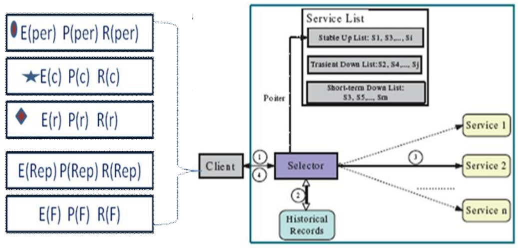

Figure 5 Quality of service assessment model used in SIBE

4.3 PROPOSED ARCHITECTURE FOR SERVICE SELECTION METHOD

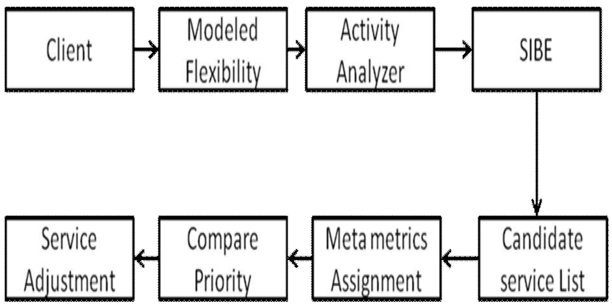

Figure 6 Proposed architecture model

The above Figure 6 shows the proposed architecture model for service selection in service orchestration of a business process model. The client in the above figure is anonymous with the business process. Every business process model represents multiple execution paths, a condition known as modeled flexibility. Activity analyzer analyzes each execution paths for the total number of ac-

ISSN 2229-5518

tivities. For all the activities analyzed, service selection optimization is carried out dynamically on an instance-by-instance basis by SIBE(Status Identification based Availability Estimation for Service Selection)as explained above.

Each execution path represents a set of activities in a service oriented business process. Each activity in the business process involves service invocation resulting in service orchestration for the entire transaction. Service selection in a transaction can be carried out once for all activities of a process, or may be carried out on an activity-by- activity basis. Here service selection on the latter grounds is used. The optimal service selected for each activity in all execution paths is held in the candidate service list. Service orchestration is finally derived by service aggregations of services stored in candidate service list.

Meta metrics assigner check the solution set in the candidate service list comprising of activities along with their optimal services. If path1 and path 2 of business process selects same optimal service for the common activity, then there is no problem. If there is a mismatch between the services in the execution paths for the common activity, then there raises conflict of modeled flexibility. Here priority of execution of activities in execution paths is used as meta-metrics to resolve the conflicts. Comparison of priority among the activities in different execution paths is carried out by analyzing the actual working environment. Then the activity with a higher probability of execution is assigned the optimal service and the same optimal service is also allotted to other common activities in different execution paths by the service adjustor thereby overcoming the conflicts with modeled flexibility. Thus the proposed work represents an innovative approach of service selection for service orchestration by addressing the conflicts with modeled flexibility based on meta-metrics.

5.0 EVALUATION METHOD

Evaluation Environment:

1. Test Bed
2. Actual Service (Google, Yahoo etc.)

Evaluation Metrics:

1. No of services Vs the time needed to selecting optimal service.
2. No of Qos constraints Vs Score value of service.
3. Comparing the meta metrics value Vs no of conflicts arose of selecting different service for same activities in different execution paths.
4. Rating Vs three QoS levels namely low, moderate and high.
5. Selection and invocation of services Vs proxy servers stored.

6.CONCLUSION

This paper presents an innovative approach to multiple Quality of service (QoS) based service selection SIBE(STATUS IDENTIFICATION BASED AVAILABILITY ESTIMATION FOR SERVICE SELECTION) for service orchestration in service oriented architecture. SIBE results in more meaningful and reliable selection of services used in service orchestration. Execution of single transaction that impacts one or more services in an organization is called a business process. Modelled flexibility in business processes can cause conflicts between optimal service selections for activities that are common to multiple execution paths. This approach presented in this paper addresses these conflicts by using a set of meta-metrics along with meaningful service selection.

REFERENCES

[1] Thomas Erl, (2005) *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall, PTR

[2] Judith Hurwitz, Robin Bloor, Baroudi C,Marcia K *Service Oriented Architecture for Dummies* Wiley Publishing, Inc.

[3] *Soa Principles Of Service Design*

[4] Canfora, G., Di Penta, M., Esposito, R., and Villani, M L.,"*An Approach for QoS-aware Service Composition based on Genetic Al gorithms*", Proc. of the 2005 Conf. on Genetic and evolutionary computation, ACM Press, New York, 2005.

[5] D. T. Tsesmetzis, I. G. Roussaki, I. V. Papaioannou, et al."*QoS awareness support in Web-Service semantics,*" in Proceedings of AICT/ICIW 2006, 2006.

[6] Yu and K. Lin, "*Service selection algorithms for Web services with end-to-end QoS constraints*," in Proceedings CEC'04, pp. 129-136, 2004.

[7] R. Ukor and A. Carpenter, "*On modelled flexibility and service selection optimisation*," in 9th Workshop on Business Process Modeling, Development and Support, vol. 335.CEUR-WS, 2008.

[8] D. Ardagna and B. Pernici, "*Global and local qos guarantee in web service selection*," in Business Process Management Work shops, 2005, pp. 32–46.

[9] O.Minhyuk, B.Jongmoon,et.al, "*An efficient approach for QoS-aware service selection based on a Tree-based Algorithm*", Seventh IEEE international conference on computer and Information science

[10] Web Service Business Process Execution Language (WS BPEL), Version 2.0 - OASIS Committee Draft, 17th May 2006.

[11] Lingshuang S,Lu Z, et.al , *Dynamic Availability Estimation For*

ISSN 2229-5518

Service Selection Based On Status Identification, IEEE International Conference on web Services, 2008 IEEE

[12] R. Berbner, M. Spahn, N. Repp, O. Heckmann, R.Steinmetz, *Heuristics for QoS-aware web service composition*, in: Proc. Int'l Conf. on Web Services,Sept. 2006.

[13] G. Canfora, M. Di Penta, R. Esposito, M.L. Villani, *An approach for QoS-aware service composition based on genetic algorithms*, in: Proc. Genetic and Computation Conf., June 2005.

[14] Roland Ukor, Andy Carpenter , *Flexible Service SelectionOptimization Using Meta-metrics* Congress on Services-I, 2009 IEEE

[15] D.A. Menascé, V. Dubey, *Utility-based QoS brokering in service oriented architectures*, in: IEEE 2007 Intl. Conf. Web Services, ICWS 2007, Application Services and Industry Track Salt Lake City, Utah, July 913, 2007, 422430.

[16] D.A. Menascé, H. Ruan, H. Gomma, *QoS management in Service Oriented Architectures*, Performance Evaluation Journal 64 (7-8) (2007) 646663.

[17] T. Yu, K.J. Lin, *Service selection algorithms for composing complex services with multiple QoS constraints*, in: Proc. of 3rd Int'l Conf. on Service Oriented Computing, Dec. 2005, pp. 130143.

[18] M. Jaeger, G. Muhl, S. Golze, *Qos-aware composition of web services: A look at selection algorithm*, in: Proc. 2005 IEEE Intl. Conf. Web Services, ICWS05, 2005.

[19] V.Deora,j.Shao,w.A.Gray, *supporting qos based selection in service oriented architecture* proceedings of the international conference on next generation web services practices , IEEE 2006

[20] M.A. Serhani, R. Dssouli, A Hafid, H. Sahraoui, *A QoS broker based architecture for efficient web service selection*, in: Proc. 2005 IEEE International Conference on Web Services, ICWS05, 2005.

[21] D.Liu, Z.Shao,C.Yu, *A heuristic Qos-aware service selection proach to web service composition,* International Conference on computer and Information science IEEE 2009