Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 766 – 773

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

An Approach for QoS Based Fault Reconfiguration in Service Oriented Architecture

Saurabh Shrivastava[a*], Ashish Sharma[b], Deepika Shrivastava[c]

[a,b]*GLA University, Mathura, India*

[c]*College of Science & Engineering, Jhansi, India*

Abstract

Web services and service-oriented style of architecture becomes the basis for a new generation of distributed applications and system management tools. With such deployments, Quality of Service (QoS) properties such as response time, availability and service costs are in high demands for SOA-based interoperability. There may be situations where services become faulty, resulting in violation of end to end QoS constraints. The work focuses on identifying the faulty services and their replacement on the basis of QoS constraints. Instead of modifying the whole composition, reconfiguration of the faulty region shall reduce the overheads of computations and service distractions in service delivery.

Keywords: Service Oriented Architecture;QoS;Web Service Business Process Execution Language;Predicate on Probe; WSDL; Directed Acyclic DAG;Web Quality (WebQ)

1. Introduction

Service-oriented architecture refers to a style of building reliable distributed systems that deliver functionality as services, with the additional emphasis on loose coupling between interacting services. Services from

[*] Saurabh Shrivastava. Tel.: 09897377125;.
E-mail address: saurabh.shrivastava@gla.ac.in

doi:10.1016/j.procs.2015.02.145

different vendors and publishers are integrated statically or dynamically using Web Service Business Process Execution Language (WS – BPEL)[1].

To perform a similar task of reconfiguration, more than one service from different service providers exists to serve the needs of the enterprise clients or users. A service composer can thus select a service which suits the need from the competing service providers.It is important that enterprise develops the service composition which not only responds with the accurate results but also maintains the desirable QoS. Generally, clients limit their quality to response time, availability of the service, budget and security issues. As per the requirements of client focuses on the quality of services based on fitness for the purpose and related applications.

In case of any failure it is not suggested to develop a new service composition from scratch[1]. Reconfiguration is a process for optimal service selection which is NP hard in nature, Further it is expected that the system should work in continuity and the recovery do not lead to migration of the system or services.

Jay et.al.[1] proposes approach for the construction of a reconfiguration region of every faulty service. The proposed approach creates a set of services that is smaller in number for each faulty region. Further the approach also includes the services, based on their QoS support. This strategy of constructing reconfiguration regions can help to reduce the overheads of removing the fault, as small numbers of services are involved.

Yu T et al.[2] proposes service process composition prior to its execution to select services along with their service levels of QoS. The selection is based on clients operational and QoS requirements. Along with the deployment it is seen that services do not deliver the required QoS, necessary to meet the end to end requirements. Factors such as host overload, network congestion, and unexpected bulk requests are responsible for the undesired results of service process composition[2].

Widyono et al.[4] proposes service composition approach based on Bellman Ford's strategy in order to obtain an optimal set of services. Further it uses Breadth First Search (BFS) strategy to discover the paths which has highest utility. The updating of highest utility path to every node is an impressing factor. The execution time of this algorithm grows exponentially.

Vander Meer et.al.[5], proposes an approach called FUSION. This approach is for web service composition and automatic execution. It uses abstracting an optimal execution plan from the requirements and specifications of users. The execution is based on the user's plan and verification on the basis of user satisfaction criteria. This proposal provides an appropriate recovery procedures.

Casati et al.[6] proposes service composition based on QoS consideration. This supported specification, performance and management of E – services, modeled as processes and operated by a service process engine. In this each node has rules for service selection, means some QoS parameters are associated with each node. A Quality Broker (QBroker) is initialized to execute a service selection rule and responds back with a list service options.

Zhou et al.[9] proposes a QoS aware service composition approach. They considered services whose service class is specified. This approach expand the limit of options for service selection by allowing services of various granularity to be available for the selection. This approach is based on mixed integer linear programming concepts. This approach optimizes the user defined objectives to meet QoS constraints. They optimized the selection approach by maximizing and minimizing the linear objective function, constrained in nature. This is NP Complete.

Zhang et al.[10] [10] proposes an approach for identifying the faulty service within a service composition. They also proposed the approach of diagnosing and recovering the fault through reconfiguration using Dependency Matrix. This approach is based on the concept of Predicate on Probe (PoP). This dependency matrix is built on the basis of process workflow structures. This matrix is built without any historical knowledge of prior executions. This approach is a polynomial time algorithm.

Whereas the service composition approaches which do not consider QoS parameters in generating a service set or composing them are summarized as :

Yu et.al. [3] proposes this combinatorial approach for service composition. They considered a service composition structure as a simple pipeline structure. They made use of Directed Acyclic Graph (DAG) to find out the execution path of a service composition. For optimal path they executed the algorithm for every reasonable path of execution, as result the complexity and computational overhead is increased. In this approach the QoS parameters were not considered.

Ponnekanti & Fox[3] in their project SWORD proposes a very simple mechanism for web service composition. This scheme made use of a rule based expert system to check the realization of composite service by the existing services and generated execution plans. This project plans the schema at composition time not at run time. Predictability and efficiency increases as no overhead at run time is incurred. But the dynamic nature of web services may fail it.

Patel et al.[7] proposes a framework which is QoS oriented and termed it Web Quality (WebQ). This framework is for adaptive management of web services, and this is based on workflows. WebQ offers adaptive selection process and simultaneously provides binding and execution of web services for the workflows. Their proposal of QoS selection considers service load to make any decision.

Egon et al.[8] surveys the utilization of Abstract State Machine (ASM) methods. This method is used for modeling and validation of web services. This method also models and validates the workflows, interaction patterns and business processes. This method is based on expressive power and accuracy of rule based modeling with visual graph based descriptions.

Yu et.al. [3] also proposes a multidimensional approach for service composition. They considered a service composition structure as a simple pipeline structure with more than one granularity of service support. They also made use of Directed Acyclic Graph (DAG) to find out the execution path of a service composition. For optimal path they executed the algorithm for every reasonable path of execution, as result the complexity and computational overhead is increased. In this approach also the QoS parameters were not considered.

2. Proposed Approach

The problem of service process reconfiguration is resolved using two approaches. The first approach generates an optional path of services from its predecessor services to the completion of service process. This allows switching to a new available optional service path without going through the faulty service. Whereas the second approach provides an alternative service path from the start to the end of a service process. This approach is based on the fact that expansion of the service set is allowed till the faulty region gets an optional path which is QoS effective to get the system in continuity. The expansion of service set for faulty region reconfiguration is a process which directly impacts the cost parameters of the system. Expansion to a large extent may lead to violation of budget constraints. Though finding out the limit to which expansion is allowed is based on the settled budget to which the cost of expansion can be compromised. This is to be kept in tight bound that expansion may not become so expansive that overall reconfiguration costs out of the estimate.

Algorithm1. Service Process Reconfiguration

```
Input: faulty services Sf, Desired QoS = D
Output: Reconfigured Sub Process Ss
1.  Initialize d=0;
2.  for all ri in Rs do
3.  find the QoS[ri]
4.  select another service for ri that meets D
5.  if success then
6.  remove ri from Rs and add to Ss
7.  end if
8.  end for
9.  while Rs = Φ do
10. d = d + 1
11. Expand Region(Rs; d)
12. for all ri in Rs do
13. find the QoS[ri]
14. reconfigure ri to meet D
15. if success then
```

```
16.   remove ri from Rs and add it to Ss
17.   end if
18.   end for
19.   end while
20.   if Rs = Φ ; then
21.   reconfigure the complete process and set it as Ss
22.   end if
23.   return Ss
```

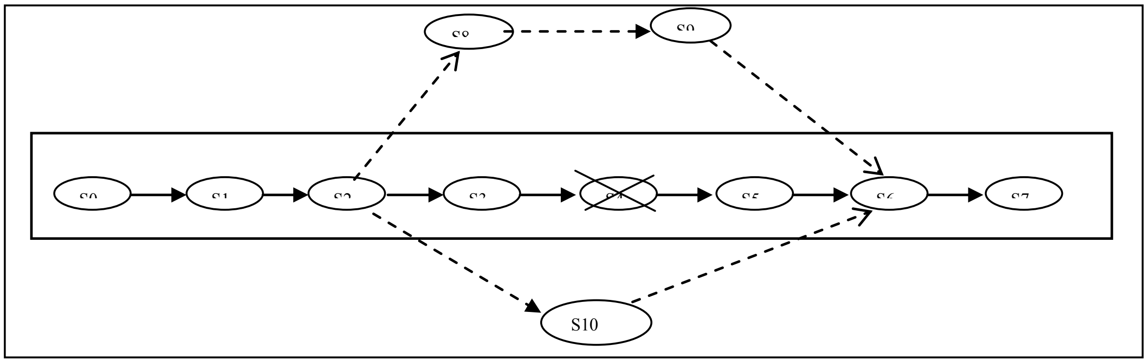

Fig. 1. A DAG of Service Composition

Considering the figure. 2. Services from S0 to S7 within the rectangular box depicts a service composition serving a specific purpose. The dotted paths are the optional paths which can also perform the specific task done by service set S3, S4 and S5. Path 1 is of S8 and S9 whereas Path 2 have only S10 in their service sets. Suppose that service S4 gets a failure and a QoS problem. We will replace this faulty service by another QoS effective service to get the system in continuity. QoS effective here means a service whose selection and integration at faulty region delivers not better but same QoS. In case, we don't have such a service. We can expand the reconfiguration region by involving the replacement of service S5. This strategy may give a more flexibility to reach the desired QoS constraint, but if this region is expanded at large can make us recompose the whole service process from scratch to meet the end to end need of the client.

2.1. Identification of faulty region

Let us assume that each service in a service composition is comprising together to build a system that having a QoS value assigned with them. The approach of finding out the faulty service or the faulty region based on the expansion strategy, starts with finding the faulty service and marking its QoS value to ∞. This service is first analyzed to see whether any other replacement for this faulty service exists or not. If yes, then replace it only if the QoS value of the optional service is within the range of acceptability. Range of acceptability here means the limit to which the minimum or maximum QoS values can be compromised. If there are more than one optional service for the replacement of the faulty service, than apply the Optimal Path Selection (OPS) Strategy to select the optimal service or services for recovering the fault. Suppose there are no optional service than expand the region of the faulty service as discussed above. Set the degree of expansion with the value of D where D varies from 0 to (n-1) and n is the number of services in sequence. After expanding the faulty region at D=1 the region will consist of nearby services included in the faulty service set. Suppose in the above figure S4 being faulty for D=1 we need a service or set of services to replace S4. But if there are no such options, we will increment the value of D. Now at D=2 we have S3 and S5 in the faulty service set. Suppose a replacement for this service set exists we will try to reconfigure the whole faulty region consisting of S3, S4 and S5. Now in this state we will consider service S3 and S5 as source and destination nodes for the BPS algorithm. In case we still do not find any solution we will go on incrementing the value of D till it includes the S0 and S7 as source and destination nodes.

In this case, the whole service composition is to be reconfigured leading to the recomposition of services from scratch. The algorithm shown below works for this specific activity of identifying the faulty service or faulty region. Note that a service violating the range of QoS cannot be considered for the reconfiguration although it proposes a solution to the fault. Considering the above figure, for the service S4 at fault there is no solution but after expanding the region till D=3 and including S2 and S6 we have two more solutions to recover from the fault of S4. We have path 1 consisting (S2, S8, S9, S6) and path 2 comprising of (S2, S10, S6).

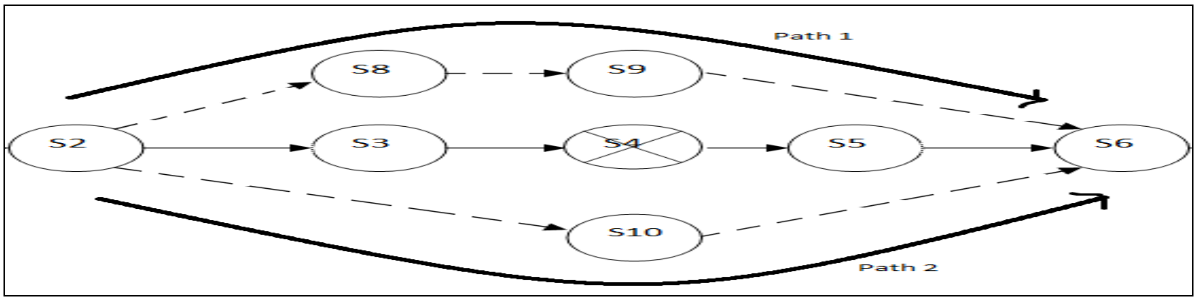

Fig. 2 DAG & optional Paths of faulty service class

Within each path there are some QoS values associated with each service. The figure 2 shows that service provided by the service set S3, S4 and S5 are also provided by the service set of path 1 and path 2. They may or may not give the same QoS, but the path with closer QoS value satisfyng the range will be taken into consideration for fault recovery.

Algorithm 2. Faulty Region Expansion

```
Input : Faulty Region Rs and distance matrix d from Si to Sj
Output : Expanded Region Rs
1. For all ri in Rs do
2. Select all the services at a distance of D[i][j] = 1.
3. For all Si, Sj do
4. If Si, Sj is not start and end node of the system then
5. Add Si, Sj to Rs
6. Else
7. For Sx being a faulty service in ri
8. Find all other nodes at distance D[i][j]<=d
9. Endif
10. Endif
11. Endfor
12. Endfor
13. For all ri in Rs do
14. Merge two regions if they have any common node
15. End for
```

2.2. Optimal Path Selection Strategy

After identifying the faulty service or faulty region we name it as reconfiguration region as it is the region which is to be reconfigured to recover from the fault. The reconfiguration of such a region is focusing on finding out the optimal path having closer QoS constraints to the reconfiguration region. We in this paper have considered the service cost as the QoS parameter to compose a system. Response Time can also be one of the parameter for such a service composition. From the figure 3 we conclude that in earlier system when S4 was in execution the QoS delivered by the path S2, S3, S4, S5 and S6 is 14. Now, after the fault, from the path 1 and path2 we have the QoS as 15 and 19 respectively. Among the replaceable paths i.e. path 1 and 2, the path with closer value to the desired QoS (QoS of the faulty region) will be the optimal path. In case if the path will have QoS value smaller than the desired then the corresponding path will not participate among the replaceable paths as per step 9 and 10 of the algorithm. Finally, path 1 is the optimal path for the reconfiguration of the faulty region. The target is always finding a optimal service set which offers QoS if not better but equal to the desired.

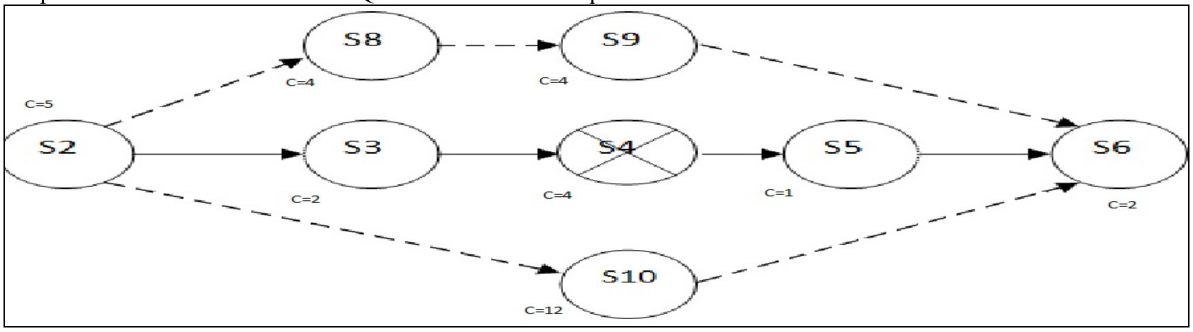

Fig. 3. DAG of faulty service class.

3. Performance Comparision & Complexity Analysis

As far as the study of this proposed algorithm is analyzed, the overall complexity of the algorithm is the complexity of expansion strategy and the complexity of the optimal path selection. For n number of services involved in the composition if expansion done then maximum (n-1) steps will be carried out, resulting in O(n). As source and destination nodes of the service composition are to be left in the expansion. Now if there are m number of replaceable paths, it may go for (m-1) steps to find the optimal path. Whereas, the calculation of QoS values of each replaceable path will need m steps. Finally, the total complexity during the optimal path selection will be $O(m^2)$. The overall complexity of this fault reconfiguration will be $O(n) + O(m^2)$. It is observed seen that a complexity of $O(m^2)$ will be analyzed at last. To best of our understanding and the concern of outputs from the implementation shows that this complexity measurement is better than others. The performance analysis says that the proposed approach considers the QoS parameters, The complexity as discussed above is low. On consideration to response time the time taken for reconfiguration will be reduced in comparison to other approaches. This approach can be implemented during run time as well as plan time. There is no specific constraint of SDL, it can be used for WSDL. The economy of the proposed approach is low and the computational overhead is also low. The parametric evaluation of the studied approaches and the proposed approach as per our understanding is shown in the Table1.

4. Result Analysis

For the result analysis, we considered three models with distinct node numbers and variable parallel structures. Figure 4, Figure 5 and Figure 6 shown are those models on which we applied the service selection strategy as proposed in the paper above. We compared the response time graph with the expanded DFS and BFS strategies. As a result the response time based service selection took lesser time than other existing approaches. Even a line graph is constructed based on the mathematical relationship of values inferred when 5, 10, 20, 30 and 40 node structure as shown in Table 1. is considered and the result derived is favorable.

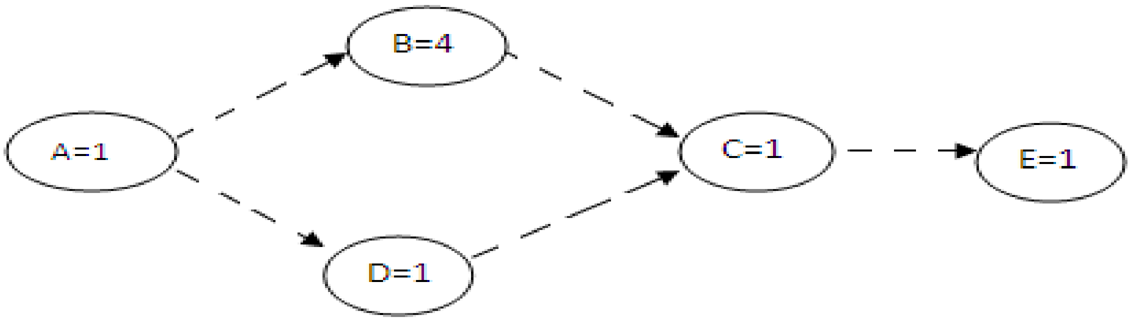

Fig. 4. Model 1 with 5 nodes & 1 parallel structures.

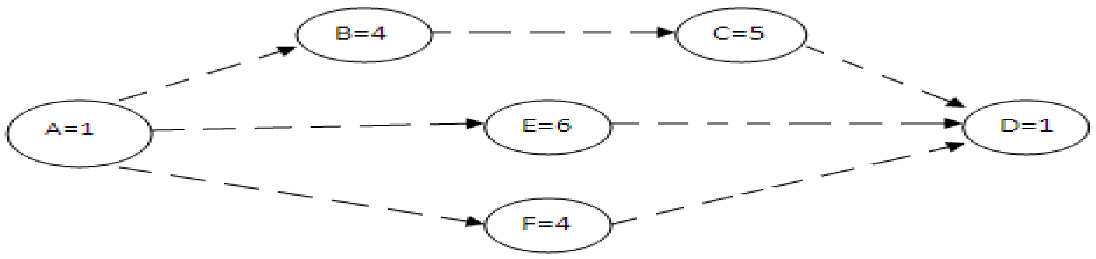

Fig. 5. Model 2 with 6 nodes & 2 parallel structures.

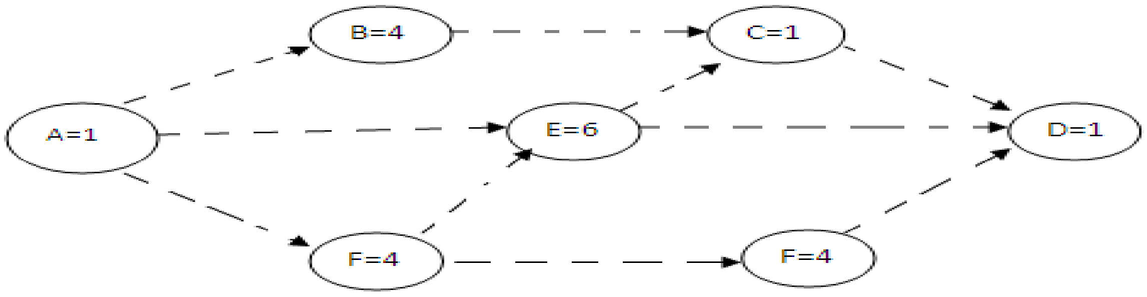

Fig. 6. Model 3 with 7 nodes & multiple parallel structures.

The resultant values when depicted through the line graph drawn based on different service set values is shown in figure 7 in which X axis reflects the number of services (Nodes) and the Y axis reflects the Response Time (ms).

Table 1. Result for models of variable nodes structures

Models	No. of Nodes (Services)	Proposed	Expanded DFS
1	5	0.10	0.10
1	10	0.10	0.10
1	20	0.21	0.21
1	30	0.31	0.31
1	40	0.42	0.42

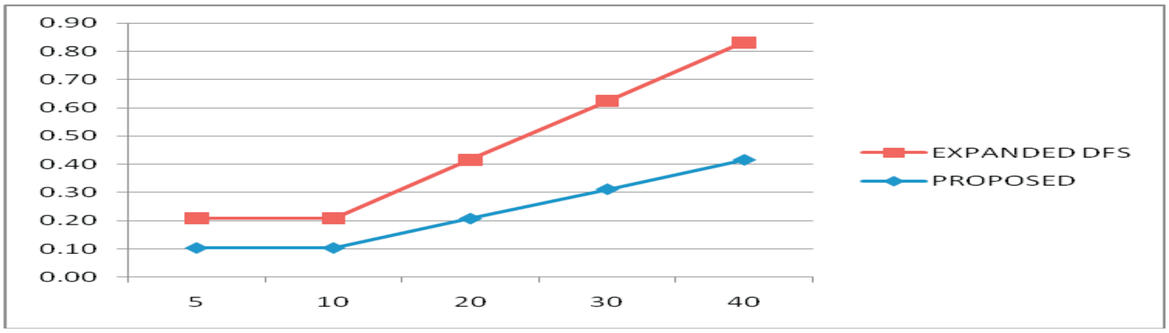

Fig. 7. Line graph for 5, 10, 20, 30 and 40 nodes structure.

5. Conclusion

From the study of various research papers for the approaches of service composition, it is found that the recovery mechanism or failure handling is one of the major issues in SOA environment. QoS consideration plays an important role in service selection and to milestone the users expectations. Therefore, we are paving a way to handle the consistency of a system in SOA environment by proposing the approach for faulty service identification and service reconfiguration. The approach discussed above is effective when a single fault at a time is considered. But if there are multiple faults then some additional constraints are required to be taken into consideration. In future, the same algorithm will be tried to handle multiple faults at an instant**.**

References

1. Kwei-Jay Lin, Jing Zhang, Yanlong Zhai. Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints, *ACM Transactions on the Web*, Vol. 1, No. 1, 2007.
2. Kwei-Jay Lin , Soo Ho Chang. A service accountability framework for QoS service management and engineering . *Springer online*: 2010. pp. 429 - 446.
3. Tao Yu, Kwei-Jay Lin, Service selection algorithms for Web services with end-to-end QoS constraints, *Springer-Verlag* 2005, pp. 103 – 126.
4. Widyono R (1994), The design and evaluation of routing algorithms for real-time channels, *Tech. Rep. TR-94-024*, University of California at Berkeley, International Computer Science Institute.
5. VanderMeer D, Datta A, Dutta K, Thomas H, Ramamritham K, Navathe SB (2003) FUSION: A System Allowing Dynamic Web service Composition and Automatic Execution. *Proc. of IEEE International Conference on E-Commerce*, Newport Beach,CA, USA
6. Casati F, Ilnicki S, Jin L, Krishnamoorthy V, ShanM (2000) Adaptive and dynamic service composition in eflow. *Technical Report*, HPL-2000.
7. Patel C, Supekar K, Lee Y (2003). A QoS Oriented Framework for Adaptive Management of Web service based Workflows Database and Expert Systems. *(DEXA-2003) conferenc*e.Prague, Czech Republic.
8. Egon B¨orger and Bernhard Thalheim, Modeling Workflows, Interaction Patterns, Web Services and Business Processes: The ASM-Based Approach. Proc. ABZ'08 (London 2008). *Springer LNCS*.
9. Bo Zhou, Keting Yin, Honghong Jiang and Shuai Zhang, QoS-based Selection of Multi-Granularity Web Services for the Composition .*Journal of Software*, VOL. 6, NO. 3, MARCH 2011. Pp. 366 – 373.
10. Jing Zhang, Xiaoqi Zhang, Yi-Chin Chang and Kwei-Jay Lin. The Implementation of A Dependency Matrix-based QoS Diagnosis Support in SOA Middleware. *ICST Transactions on eBusiness* July-September 2012 j Volume 12 j Issue 7-9 .