

## Efficient Techniques for Service Oriented Application

Anoop P

Pursuing M.Tech-CSE, Caarmel Engineering College.

---

**Abstract:** A Service-Based application is composed of different services available in a network, which performing the required functions of the architecture. Web services composition is emerging as a technology for the effective automation of business to business co-operation. It allows organizations to form an alliances by connecting their databases, applications and systems to offer "one stops shops" for their clients. In the business world, SLA binds the legal relationships between customer and service provider. This method is independent of specific programming language or operating systems. Preventing the chances for SLA violations is an important thing in business world to keep quality of services as good. Previous research on this topic were only focused for selection of adaptation technique. That is, it designing a software system to providing services either to end-user application or other services distributed in a network via published and discoverable interface to select the best adaptation. It only gives an awareness about selection of adaptation if SLA violation happened. In my thesis work, assume that the goal of service providers is to reducing SLA violations and implements an adaptation mechanism and techniques to reducing the cost of doing so when the violation has occurred. Here, the service provider apply run-time adaptation with the composition process. And, improves the problem of predicting SLO violation by using QoS metrics of web composition. For selecting cost-effective adaptive action, here use algorithms such as branch and bound. And, also discuss that how this proposed work help service providers to reducing the cost of optimization.

**Index Terms:** SLA, SLO, Service Composition, adaptation

---

### I. Introduction

The monitoring and prevention of SLA violations have very much importance in composite services. The service industries need to more focus on quality issues of business services. In the business world, the premises of QoS binds a legal relationship i.e.; Service Level Agreement between clients and service providers. SLA consist of SLOs, which need to ensure the quality of services. SLOs have fixed numerical QoS Objectives. When SLOs are violated, the consequences will make big impact on the efficiency of services.

In a business composition, preventing SLA violations is not for free. So some alternative services are used for business composition to provide faster response time. But those services are more expensive. For improving the end to end runtime of composite services and reducing the probability of SLO violation are required for the success of business composition. Violation of SLA cause strong penalties for service provider. Hence service provider take strong interest in preventing SLA violations. Therefore, there is a compromise between inherent cost required to reducing the SLA violation and preventing SLA violation. But, the researchers not covered this fact efficiently. They only focus on minimize SLA violations not care about the significant to cost need to doing so.

In this paper, give more importance for service provider to minimize total cost of violations and applied adaptation by consider as an optimization problem. And also argue that the real goals of service providers efficiently captured. Additionally, at composition time the possible algorithm to solve optimization problem are applied. Finally evaluate this project with prediction and prevention based on event framework.

This paper is structured as follows: In Section 2, Present an illustrative example and explain motivation behind the work. Following in Section 3, present the background of the paper and in Section 4 explains the related work of the paper and in Section 5 explains the technical details and finally presenting experimental result and concluded in Section 6 & 7.

### II. Motivation

In this paper, consider following example to explain my approach depicted in Fig 1. Which explains the scenario behind Sales Order Service.

The sales process consist of 2 parts: First the customer parts. The customer search for the required product and then registered user send an RFQ to the manufacture of product. The manufacturer of the product respond with an offer consist of delivery time and estimate price. Second, the customer accept this quotation from manufacturer and the take action for payment with help of bank. Since the company's business is based on service-based purely on a service-base notion. The sales service based application applied as a service composition i.e. all activities are implemented by invocation of two or more web services. In this figure give

more importance to customer and manufacturer but the manufactures are also interacts with the banks, SLA service providers and shippers.

For the success of business, manufacture has established some SLA's. Some typical examples of SLO's are listed in Table 1.

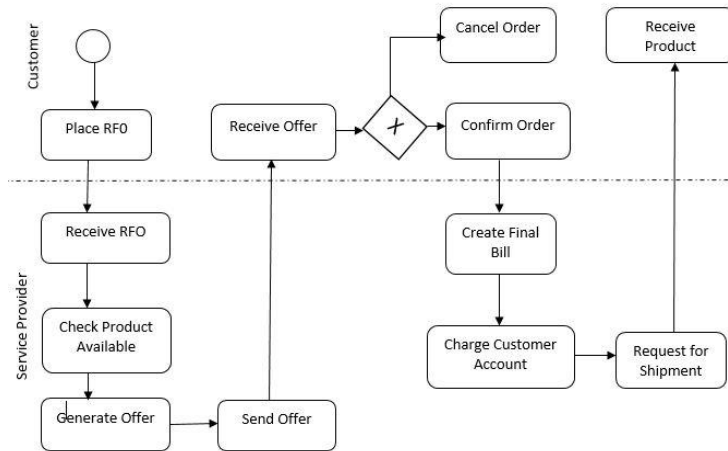


Fig. 1. Motivating Scenario

All the SLO's have penalties for violating associated target values. Therefore, the company has an interest in following these SLO's. The companies may use different runtime adaptations to the process. The SLO's can be of qualitative or quantitative nature. In my paper, survey some adaptation actions which are listed in Table3. The column – and + mention to SLO's in Table1 and suggest that various action has a negative (-) or positive (+) impact on this SLO. Note that, generally these all SLO's are does not need for all the products. Because these may cause some additional costs for the companies and does not get any economic benefit if the product is lower cost. Since the company's business process is implemented as a service composition, these adaptations reduce to adapting composition of services. This can be achieved by invoking different service or by changing the structure of composition itself. Mostly, the previous papers about this topic not discuss how adaptation actions select effectively, they mention only about how adaptations can be technically applied [8, 9].Selecting the most beneficial set of adaptations to prevent predicted violation leads an optimization problem. Discussing this optimization problem is the main part of this paper.

Table 1: Service-Level Objectives

#	SLO Name	Description
1	Time to Offer	Time between receiving the RFQ and responding with an offer
2	Order Fulfillment Time	Time between receiving the order and finishing the process.
3	Process Lead Time	Time between initializing the process and finishing it.
4	Cost Compliance	Cost overrun with regard to the offer in % of the offer.
5	Product as specified	Product is exactly as specified.

Table 2: Target Values and Penalties

#	Target Value	Costs of Violation
1	<=2	Implicit costs-customer will choose a different manufacturer if offer is not received in time
2	<=5	Manufacturer grants 5% discount per 1 day delay, 20% max discount, not additive with SLO=2
3	<=6	Manufacturer grants 5% discount per 1 day delay, 20% max discount, not additive with SLO=3
4	<=5	Manufacture cannot charge more than the offer plus 5%
5	n/a	If wrong product is delivered, manufacture needs to produce and ship specified product within 7 working days and grant a 5% of discount.

### III. Background

To give some background information about this paper, introduce the PREVENT framework, which forms the basis for the research discussed here. Generally, PREVENT is a closed-loop system [9] for self-optimizing service compositions. PREVENT is grounded on the existing SOA runtime environment VRESCO [10]. As outlined in Fig. 2, the PREVENT framework comprises of the steps “monitoring,” “analyzing,” “planning,” and “execute,” as mentioned in the vision of autonomic computing [11]. Initial version of the PREVENT framework early presented in [7].

Usually, the idea behind the PREVENT to generate runtime prediction of SLA violation is use event-based monitoring of composition data. In this paper, focus on implementing Cost-Based Optimizer part in figure 2 which have not discussed in previous papers. For every composition case, this component receives an idea of fixed SLA values from the Violation Predictor component, and determine, which adaption action should be selected. Here use the name ‘optimization time’ as to indicate the time during composition instance execution when optimization happens.

#### 3.1 Prediction of SLOs

Generally, the PREVENT framework is to predict the SLA violation based on the fixed values of SLO by monitoring the data. Here classify 3 different types of info. The data that can be measured at optimization time can be termed as Facts. Opposite of Facts are Unknowns i.e. the data are unknown at optimization time. Data that remains middle ground between Facts and Unknowns are termed as Estimates. The Violation Predictor component uses both estimates and facts from earlier monitored service execution to train a machine learning which can be used to create a numerical estimation of the SLO values at runtime. One model trained per SLO that is to be predicted, and every model is trained from different SLO data. Obviously, historical execution of composite services are needs to assist the training. This prediction cannot be used when no historical data is available. If this is the event, apply an alternative approach, which is not data dependent, e.g., [12].

#### 3.2 Identification of Factors of influence

As stimulus to the machine learning-based SLO Predictor approach needs to be identify the factors which influence the service composition. Often factors of influence are not clearly. Hence business analyst uses a dependency analysis to identify factors of influence.

Dependent analysis is a semi-automated process. In first step, the business analyst needs to specify an initial set of potential factors of influence. These includes both QoS metrics and domain specific metrics which needs to define manually. But QoS metrics are automatically generated. For every potential factors, a monitor is generated or defined, which specifies how this metric can be measured at running time. Second, a data set containing these potential factors to be generated by reproducing the composition in a web service test environment or by monitoring the real execution. Using this data set dependency tree is generated which contains the factors that explain SLO violations in the composition. The third step is use these factors to explain SLO violations in a composition. If the prediction model needs high training in data correlation against the measured data set. If the correlation is not sufficient the business analyst needs to identify the reason for lacking performance.

### **3.3. Adaptation Actions**

The PREVENT adaptation executor can execute a chain of different adaptation for service composition. Here distinguish three types of adaptations: First, Data manipulation action in which the composition fact is not changed instead data flow of composition instance data is changed (e.g. priority is changed to high priority). Second, Service rebinding is used to implement different service (e.g. faster shipping is used). Third, Structural adaptation in which change the logical structure instead of compositional change (e.g. skip some quality control activities). In this adaptation remove some activities in an instance. Here can define any number of adaptation action. Each adaptation action has a non-negative, constant values. Every adaptation action can have any positive or negative impact on any fact or estimate.

## **IV. Related Work**

To the best of my knowledge, no approaches with the accurate point of this cost-based optimization of service composition have been published as yet.

On a key level, this work is based on the belief that both composite and atomic services present approximately measurable quality (QoS). Different techniques suggested in this direction regard monitoring of TCP-level metrics using network analysis technique [15], monitoring based on client feedback [14] or event based monitoring based on event-condition-action rules [15]. Here apply the VRESCO event engine and monitoring in a way alike to approach presented in [15].

Usually, PREVENT is a system to prevent and monitor SLA violations. The run time monitoring of compositional quality is particular interest to us, because it talk about an integrated method toward monitoring based on events [21]. This is approximate similar to PREVENT but these works do not tried to explicate the cause for SLA violation, and neither try to prevent them. The MoDE4SLA approach [3] is a top-down approach for these potential factors of SLA violations. Research in a similar point, but employing data mining techniques alternatively of top-down analysis, has also been introduced in [4]. In this paper, not only try to concentrate on which parts of service composition cause SLA violation, but also actively preclude by implementing targeted adaptive actions. So, my system basically applies the prototype of self-adaptive service composition. This is associated to the area of flexible service composition, as presented in [22]. Likely self-healing process have also been presented as part of the DISC framework [23], which carries out only partial and dynamically defined process. In this paper, present the VieDAME framework, which self-directly monitors the QoS of services applied in the composition, and activate service re-selection if the QoS falls below a given threshold.

This is alike to the PREVENT approach, but our system supports a wide range of adaptation actions. Another self-adapting composition is MASC [24]. However, this paper point more on adaptation for functional reasons, while my goal is the optimization of non-functional aspects. Moreover, the MASC system also does not convey merged cost of adaptation.

The PREVENT approach focus at autonomous optimization of service compositions with respects to cost of adaptation and SLA violation [16]. Later methods attempted to improve on this concept by using more effective heuristic algorithm, e.g., HI\_RELAX\_IP [17], WFlow [18] or the immune algorithm [19]. Different people approach this problem by the combination of local selection and global optimization. This method can also be regarded as a formula, because local selection does not ensure a global optimal solution [20]. In my paper, use local search combining with genetic algorithm to economically clear the QoS optimization problem. The main difference of this paper from other approach is, my goal is to reduce the costs of adaptation arising from SLA violation. Hence the optimization is solve differently.

A long-familiar approach for predicting QoS levels in composite service is QoS aggregation, i.e., the process of computing the quality of composite services based on the QoS of the applied services and aggregation function [13]. This idea of QoS aggregation has been stretched out to SLA aggregation [33] by many authors. As an option to SLA aggregation and QoS aggregation, different machine learning technique to predict the compositional QoS from runtime data [5], [6]. This method is also applied in PREVENT, as described in Section 2. The main advantage of using machine learning algorithm is that it is very comfort to incorporate non-QoS data without need to describe how this data influence the composition performance.

The core part of this paper is the belief that there usually a trade-off to consider between cost of adaptation and preventing SLA violation. Hence, a composite service provider is increasing his own gross by decreasing provider's total costs. Mazzucco et al. [25] describe a framework for gross increasing in web services hosting using dynamic admission methods. Likely, techniques to optimize application servers in a distribute environment have been proposed in [26]. Other studies related to this topics are going with performance-security trade-off [27] or trade-off between costs of monitoring and composition of Quality of Services for service oriented application.

## V. Technical Section

In this section, validate the problem of selecting the most cost-effective adaptation actions to prevent one more predicted SLA violations. Here consider an interaction of the service composition with a given client, who has a given SLA with the composition provider. Let  $I$  be the set of all possible composition instances of this client, and let  $S = \{s_1, s_2, \dots, s_k\}$  be the set SLO set. The penalty function is associated with all the SLO's as  $P = \{p_{s1}, p_{s2}, \dots, p_{sk}\}$ . For prevent violation number of adaptation are applied to an instance  $i$ .  $P(A)$  denotes the power set of all adaption action where  $A = \{a_1, a_2, \dots, a_k\}$ . Selecting the best cost-effective action finding the adaptation action that minimize the total costs for the service provider. The total costs is defined as the sum of the cost of adaptation (AC) and sum of the cost of SLA violation (VC).

### 4.1 Branch-and-Bound

It is a deterministic algorithm for solving optimization problems. The idea behind this approach is that suboptimal solutions can be permuted or identified before they have evaluated or constructed. Here prune a partial solution in two events: 1.) if the partial solution already contain at least one clash, or 2.) if the partial solution already prevents all SLA violation.

### 4.2 Local Search

It is a metaheuristic algorithm. That is final answers are constructed by iteratively improving the first solution. The idea behind this algorithm is search for the neighborhood to get better solution than current one.

### 4.3 Momentic Algorithm

As an alternative to local search algorithm here present a momentic algorithm. It is more powerful formula to generate better solution to cost base optimization. Every following iteration of algorithm it follows 3 step pattern. 1.) Selection of set of solution from population to "survive" into following generation. 2.) Crossover is used to generate new solution based on the selected one of last generation. 3.) Mutation to introduce add new features to population.

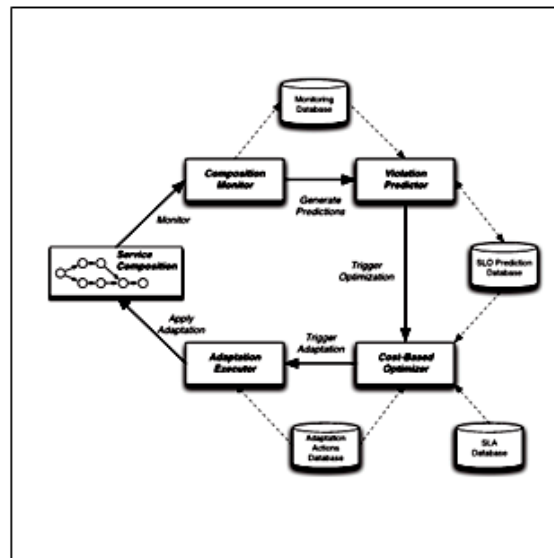


Fig. 2. Overall Framework

```

Algorithm: Bab
Let 'p' be the partial solution
'i' be next alive active index
'a' be the adaptation action
'j' be number of adaptation action
'V' be the target function.
Output: optimal complete solution
1. bab(p,i):
2. if(p==complete solution)
3. return p
4. if(p!=complete solution)
5. for all alive_action apply 'a'
6. set p(j)=0 //complete solution
7. return p
8.
9. While(j>0) //more adaptations
   available
10. Consider a[j]=0
11. solution s1=bab(p,i+1)
12. Consider a[j]=1
13. solution s2= bab(p,i+1)
14.
15. # return better solution from both
16. if(v(s1)<=v(s2))
17. return s1
18. else
19. return s2
    
```

Fig.3.Branch-and-Bound-Algorithm

```

Algorithm: GRASP construction
Let 'n' is the number of total correct solution
'R' is maximum size of Restricted Candidate
Set
'G' is set of start solution.
1. for i=0 to n do
2. PS=empty partial solution
3. While violation cost(PS>0)
4. Then RCS=new RCS(PS,R)
5. If(RCS=null) break;
6. Then a= random(RCS)
7. Then PS(a=1)
8. g.add (PS)
9. end while
10. end for
11. return g
    
```

Fig .4. GRASP Construction Algorithm



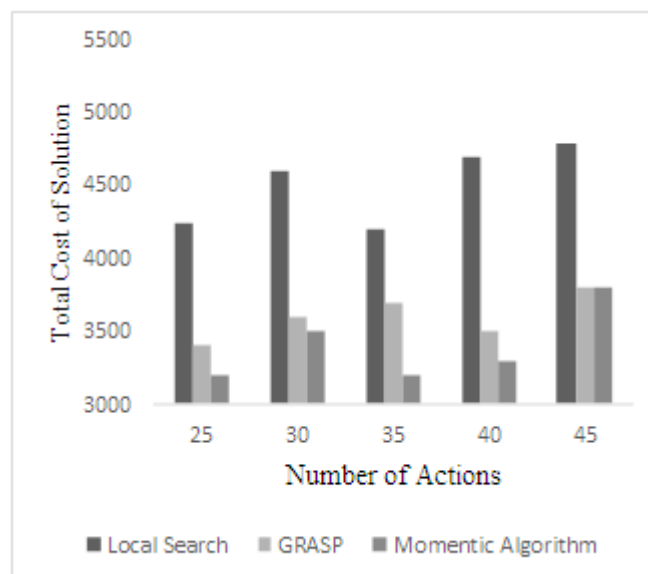


Fig.5. Quality of solution per heuristic

## VI. Experimentation And Result

In the following section, numerically validate the algorithms discussed in Section 4 based on an implementation of the scenario presented in Section 2. For reasons of brevity, only summarize the experiment setup here. More details can be found in the accompanying experimentation webpage. In addition, we do not explicitly evaluate the prediction quality (i.e., the SLO Predictor component) here. The interested reader may find a numerical evaluation of the prediction in [5], as well as in [7]. In experiments, consider the case of an SLA containing up to five SLOs, similar to the previous example.

In Fig. 5, the algorithms are compared with regard to solution quality, measured as predicted total costs for the service provider and observe a quite clear ordering of algorithms in this experiment. GRASP and MA generally perform best.

## VII. Conclusion And Future Work

For providers of composite web services, it is necessary to be able to minimize cases of SLA violations. One possible path to attain this is to predict at runtime, which cases are in risk of violating SLAs, and to utilize various adaptation actions to these instances only. However, it is not easy to recognize which adaptations are the most cost effective manner to prevent any violation, or if it is at all possible to prevent a violation in a cost-effective way. In this paper, modeled this problem as a one-dimensional discrete optimization problem. This paper proposed to develop an efficient method for the success of business using web composition. The service-oriented computing (SOC) paradigm uses services to support the development of rapid, low-cost, interoperable, and massively distributed applications. Services are autonomous, platform-independent entities that can be described, published, discovered, and loosely coupled in novel ways. They perform functions that range from answering simple requests to executing sophisticated business processes requiring peer-to-peer relationships among multiple layers of service consumers and providers. Any piece of code and any application component deployed on a system can be reused and transformed into a network available service.

Aggregate SLOs, which are defined across a number of instances, are out of scope. Similarly, at the instant here do not consider “permanent” adaptations, i.e., adaptations which are done for all future instances. And this work can be extended using dynamic Service Composition. That is, an issues regarding reliability of service compositions are dominantly handled at run time. This is due to the fact that service QoS parameters can change unpredictably and, thus, run time dynamic service selection is an appropriate method to make service compositions more robust to failures.

## References

- [1]. M.P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, “Service-Oriented Computing: State of the Art and Research Challenges,” *Computer*, vol. 40, no. 11, pp. 38-45, Nov. 2007.
- [2]. A. Dan, D. Davis, R. Kearney, A. Keller, R. King, D. Kuebler, H. Ludwig, M. Polan, M. Spreitzer, and A. Youssef, “Web Services on Demand: WSLA-Driven Automated Management,” *IBM Systems J.*, vol. 43, no. 1, pp. 136-158, Jan. 2004.
- [3]. L. Bodenstaff, A. Wombacher, M. Reichert, and M.C. Jaeger, “Analyzing Impact Factors on Composite Services,” *Proc. IEEE Int’l Conf. Services Computing (SCC ’09)*, pp. 218-226, 2009.
- [4]. B. Wetzstein, P. Leitner, F. Rosenberg, S. Dustdar, and F. Leymann, “Identifying Influential Factors of Business Process Performance Using Dependency Analysis,” *Enterprise Information Systems*, vol. 4, no. 3, pp. 1-8, July 2010.

- [5]. P. Leitner, B. Wetzstein, F. Rosenberg, A. Michlmayr, S. Dustdar, and F. Leymann, "Runtime Prediction of Service Level Agreement Violations for Composite Services," Proc. Third Workshop NonFunctional Properties and SLA Management in Service-Oriented Computing (NFPSLAM-SOC '09), pp. 176-186, 2009.
- [6]. L. Zeng, C. Lingenfelder, H. Lei, and H. Chang, "Event-Driven Quality of Service Prediction," Proc. Sixth Int'l Conf. ServiceOriented Computing (ICSOC '08), pp. 147-161, 2008.
- [7]. P. Leitner, A. Michlmayr, F. Rosenberg, and S. Dustdar, "Monitoring, Prediction and Prevention of SLA Violations in Composite Services," Proc. IEEE Int'l Conf. Web Services (ICWS '10), pp. 369-376, 2010.
- [8]. P. Leitner, B. Wetzstein, D. Karastoyanova, W. Hummer, S. Dustdar, and F. Leymann, "Preventing SLA Violations in Service Compositions Using Aspect-Based Fragment Substitution," Proc. Int'l Conf. Service-Oriented Computing (ICSOC '10), 2010.
- [9]. M. Salehie and L. Tahvildari, "Self-Adaptive Software: Landscape and Research Challenges," ACM Trans. Autonomous and Adaptive Systems, vol. 4, no. 2, pp. 1-42, May 2009.
- [10]. A. Michlmayr, F. Rosenberg, P. Leitner, and S. Dustdar, "End-to-End Support for QoS-Aware Service Selection, Binding, and Mediation in VRESCo," IEEE Trans. Services Computing, vol. 3, no. 3, pp. 193-205, July 2010.
- [11]. J.O. Kephart and D.M. Chess, "The Vision of Autonomic Computing," Computer, vol. 36, no. 1, pp. 41-50, Jan. 2003.
- [12]. D. Ivanovic, M. Carro, and M. Hermenegildo, "Towards DataAware QoS-Driven Adaptation for Service Orchestrations," Proc. IEEE Int'l Conf. Web Services (ICWS '10), pp. 107-114, 2010.
- [13]. M.C. Jaeger, G. Rojec-Goldmann, and G. Muhl, "QoS Aggregation for Web Service Composition Using Workflow Patterns," Proc. Eighth Int'l Enterprise Distributed Object Computing Conference (EDOC '04), pp. 149-159, 2004.
- [14]. R. Jurca, B. Faltings, and W. Binder, "Reliable QoS Monitoring Based on Client Feedback," Proc. 16th Int'l Conf. World Wide Web (WWW '07), pp. 1003-1012, 2007.
- [15]. L. Zeng, H. Lei, and H. Chang, "Monitoring the QoS for Web Services," Proc. Fifth Int'l Conf. Service-Oriented Computing (ICSOC '07), pp. 132-144, 2007.
- [16]. L. Zeng, B. Benatallah, A.H.H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-Aware Middleware for Web Services Composition," IEEE Trans. Software Eng., vol. 30, no. 5, pp. 311-327, May 2004.
- [17]. R. Berbner, M. Spahn, N. Repp, O. Heckmann, and R. Steinmetz, "Heuristics for QoS-Aware Web Service Composition," Proc. IEEE Int'l Conf. Web Services (ICWS '06), pp. 72-82, 2006.
- [18]. T. Yu, Y. Zhang, and K.-J. Lin, "Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints," ACM Trans. the Web, vol. 1, article 6, May 2007.
- [19]. J. Xu and S. Reiff-Marganiec, "Towards Heuristic Web Services Composition Using Immune Algorithm," Proc. IEEE Int'l Conf. Web Services (ICWS '08), pp. 238-245, 2008.
- [20]. M. Alrifai and T. Risse, "Combining Global Optimization with Local Selection for Efficient QoS-Aware Service Composition," Proc. 18th Int'l Conf. World Wide Web (WWW '09), pp. 881-890, 2009.
- [21]. L. Baresi, S. Guinea, M. Pistore, and M. Trainotti, "Dynamo+Astro: An Integrated Approach for BPEL Monitoring," Proc. IEEE Int'l Conf. Web Services (ICWS '09), pp. 230-237, 2009.
- [22]. D. Ardagna, M. Comuzzi, E. Mussi, B. Pernici, and P. Plebani, "PAWS: A Framework for Executing Adaptive Web-Service Processes," IEEE Software, vol. 24, no. 6, pp. 39-46, Nov./Dec. 2007.
- [23]. E. Zahoor, O. Perrin, and C. Godart, "DISC: A Declarative Framework for Self-Healing Web Services Composition," Proc. IEEE Int'l Conf. Web Services (ICWS '10), pp. 25-33, 2010.
- [24]. A. Erradi, P. Maheshwari, and V. Tosic, "Policy-Driven Middleware for Self-Adaptation of Web Services Compositions," Proc. ACM/IFIP/USENIX Int'l Conf. Middleware (Middleware '06), pp. 62-80, 2006.
- [25]. M. Mazzucco, I. Mitrani, J. Palmer, M. Fisher, and P. McKee, "Web Service Hosting and Revenue Maximization," Proc. Fifth European Conf. Web Services (ECOWS '07), pp. 45-54, 2007.
- [26]. D. Villela, P. Pradhan, and D. Rubenstein, "Provisioning Servers in the Application Tier for E-Commerce Systems," ACM Trans. Internet Technology, vol. 7, no. 1, article 7, 2007.
- [27]. S.S. Yau, Y. Yin, and H.G. An, "An Adaptive Tradeoff Model for Service Performance and Security in Service-Based Systems," Proc. IEEE Int'l Conf. Web Services (ICWS '09), pp. 287-294, 2009.
- [28]. Y. Zhang, M. Panahi, and K.-J. Lin, "Service Process Composition with QoS and Monitoring Agent Cost Parameters," Proc. IEEE 10<sup>th</sup> Conf. E-Commerce Technology and the Fifth IEEE Conf. Enterprise Computing