International Journal of Soft Computing 5 (4): 164-170, 2010
ISSN: 1816-9503

Web Service Categorization Using Structural Metrics

M. Sathya, P. Dhavachelvan and G. Sureshkumar
Department of Computer Science, SET, Pondicherry University, India

Abstract: Service-oriented systems have become popular and presented many advantages in developing enterprise services. The service categorization is the inevitable process in web service composition for choosing most suitable services from the available services that suits to the user's requirements. Many researchers propose several service categorization techniques like those that functional metric based service categorization and non-functional metric based service categorization. Nevertheless, those methods did not address the structural properties of the target services. We are proposing a structural metric based categorization technique for effective Web Services Composition (WSC). Coupling is the most important structural attribute of services when they integrated into a system. Structural metrics are useful to evaluate service's quality according to its ability of coupling. We are aiming at using the coupling metrics to measure the maintainability, reliability, testability and reusability of services. We have identified two Service Colony Metrics (SCM) to formulate the Weighted Dependency Metric (WDM) which is used as the fitness function of Genetic Algorithm based Service Categorization (GASC) technique for optimizing the composite service categorization. This dependency metric based GASS technique have been tested in e-learning web services system and the experimental results shows improved performance when compared with non-functional metrics based categorization techniques.

Key words: Service catogorization, structural metrics, coupling metrics, web service composition, e-learning web services, GASC

INTRODUCTION

In the present study, the web services and service oriented computing are the most promising research and development areas. The research contributions will help to build cost effective and reusable service oriented applications for enterprises. Achieving the desired quality is a complicated task for all service oriented development. It has been widely accepted that high quality software should exhibit low coupling, complexity and high cohesion. Structural software does not directly describe the quality of a software product rather they are used as predictors for measuring the external quality attributes.

Coupling is the most important structural attribute of service oriented software system and is presented by relationship between services. This relationship among services shows the dependency between them. If a service has more relationship with other services, it will depend on others much more. Therefore, if a service has more dependency relationship, the coupling between this service and others will become tighter. It can be seen from relationship in service oriented software as the more relationship exists inside software and the tighter coupling attribute is. In this study, the concept of coupling is used for measuring the strength of association established by a connection from one service to another and taking into account service properties such as stateness, dependency and interaction. This study proposes two service colony metrics which is used as the fitness function of Genetic Algorithm based Service Categorization (GASC) for optimizing the composite service selection.

Web services are the powerful mechanism for integrating existing software applications over the web, independently of programming language, execution platform or transport protocol. Web services keep on emerging at an every increasing pace. As web services increases, many businesses are providing similar services with overlapping functionalities. Therefore, it is very difficult for users to select an appropriate service among the sea of services. A number of mechanisms for web service categorization have been carried in the literature. Simplest among them is key word based categorization. This categorization mechanism searches for exact match in the UDDI registry. This research same as searching for a material in any search engine.

The existing syntactic-based service categorization technologies are insufficient for building a full-fledged composite service. The limitations of using this mechanism in web service is retrieving irrelevant services

Corresponding Author: M. Sathya, Department of Computer Science, SET, Pondicherry University, India

to consumers. A number of techniques for overcoming this issue are proposed. The most common approach among this is matchmaking technique. It is used with semantic descriptions for the service's functional attributes are needed to be matched. Functional based service categorization makes use of domain ontology to achieve feasible solution to the service request. To provide appropriate solution to service categorization request, the QoS of services need to be addressed. Due to the dynamic and unpredictable nature of the web providing the suitable QoS is really a challenging task which needs to address the requirements of both service consumer and service provider. The next level of service categorization mechanism, this study focus is to select the services based on their external quality by using colony metrics. The proposed colony metrics are used as fitness function of continuous Genetic Algorithm (GA) for optimizing the composite service categorization. The GA, a naturally evolutional heuristic methodology that can efficiently solve the problems of function optimization and control system was first put forward by Hollandin. Most of GA researches are binary oriented.

Many variant of genetic algorithms are discussed in Haupt and Haupt (2004). One difficulty with conventional genetic algorithms is that they suffer from premature convergence caused by an early homogenization of the genes. The process of controlling the population diversity becomes more vital when Genetic Algorithms (GA) are used for the optimization in non-stationary environments like web services where periodical changes occur. Thus GA with real-coded representation is capable of continuously adapting to changes in the environment when searching for optimal solutions (Kubalik, 2005). Therefore, this study applies a continuous parameter GA that is in which parameters are real numbers.

Moreover, the coupling/colony metrics used as fitness, subsequently is defined as that it meets the solution falls between <0 and ≤1.

LITERATURE REVIEW

Coupling is defined here as the degree to which each service relies on each one of the other services. Since it is the degree of interaction between services, the basic idea of coupling metric is to count how many interactions there are in between services. Nevertheless there is substantial difference depending on what counts as an interaction, how the counting is done and normalized.

Various coupling metrics have been presented in the literature. Choi and Lee (2007) proposed a dynamic coupling metric in order to measure the coupling accurately between classes allowing the dynamic property in the object level. In addition, they proved the theoretical soundness of the proposed metric by the axioms of Briand and suggested the accuracy of the proposed metric through a comparison with conventional metrics. Shen *et al.* (2008) proposed a fine-grained coupling metrics suite for Aspect-Oriented (AO) systems to measure software changes during system evolution.

They also presented a correlation model in terms of intermediate processes for better evaluating the relation between coupling metrics and system maintainability. To investigate the practicability of their proposed model, they have implemented a coupling metrics analysis tool called AJMetrics and performed an empirical study on eight AspectJ benchmarks.

Perepletchikov *et al.* (2007) proposed a set of metrics for quantifying the structural coupling of design artifacts in service-oriented systems. The metrics which are validated against previously established properties of coupling are intended to predict the quality characteristic of maintainability of service-oriented software.

Qian *et al.* (2006) provided a practical guide for evaluating decoupling between service-oriented components in the service composition such as Business Process Execution Language (BEPL). They suggested that s lower decoupled distributed software application would be much easy to understand, update and expand in the future. Quynh and Thang (2009) proposed a suite of metrics to evaluate service's quality according to its ability of coupling. They used the coupling metrics to measure the maintainability, reliability, testability and reusability of services.

Their proposed metrics are operated in run-time to bring more exact results. Li and Henry (1993) defined the Message Passing Coupling (MPC) as the count of the number of send statements that is found in methods of one class to other classes. Chidamber and Kemerer (1994) introduced the Response For Class (RFC) as a measure of the number of methods that can potentially be executed in response to a message received by an object of that class. Chidamber and Kemerer (1994) defined Coupling between Object Classes (CBO) as the count of the number of classes to which it is coupled and further elaborated in the definition as two classes are coupled when methods of one class use methods or instance variables defined by one class use methods or instance variables defined by the other class. These measures are not a dynamic measure of coupling because it does not count the number of invocations during execution but these count the number of methods and variables invoked.

Prasad and Nagar (2009) defined a new set of operational measures for the conceptual coupling of classes which are theoretically valid and empirically studied. In this study, they showed that these metrics capture new dimensions in coupling measurement, compared to existing structural metrics.

EASE SERVICE CATEGORIZATION MECHANISMS

A number of mechanisms exist in the literature for service categorization process starting from key word categorization, matchmaking with functional attributes which makes uses of domain ontology and service categorization based on the non-functional properties such as performance, time, cost, reputation etc. Several service matchmaking techniques have been developed to meet the needs of both consumers and providers.

Chua and Mustapha (2006), Zeng and Benatallah (2004) addressed this issue of selecting web services by maximizing user satisfaction expressed as utility functions over QoS attributes (Kaufmann *et al*., 1991; Sirin *et al*., 2004) developed a goal-oriented and interactive composition approach that uses matchmaking algorithms to help users filter and select services while building their composition service. In functional semantic is taken into consideration thereby avoiding unsatisfied result which are not of customer interest (Shin *et al*., 2009). They proposed a composition method that explicitly specifies and uses the functional semantics of web services based on domain ontology. Here the researchers have defined the functional semantics of a service as describing what a service actually does.

The service functionality of a service is represented by a pair of its action and the object of the action. In (Klusch and Kapahnke, 2008) a hybrid semantic web service categorization of semantic services in SAWSDL based on logic based matching as well as text retrieval strategies are proposed. With the predominant increase of Web services as a business solution to enterprise application integration, the QoS offered by Web services are essential for service providers and their service consumers. Due to the dynamic and unpredictable nature of the web providing the suitable QoS is really a challenging task. Zhang *et al*. (2006) proposed a QoS model and used hierarchy policy approach to capture goals of users, applications, environment and resources to form rational service composition and adaptation action. The researchers have proposed a Service Adaptation Evaluation (SAE) algorithm to handle service adaptation problem and service composition decision problem in pervasive computing environment.

In order to enable quality-driven web service categorization, Liu *et al*. (2004) proposed an open, fair, dynamic and secure framework to evaluate the QoS of a vast number of web services. The three key aspects that are developed in this technique include extensible QoS model, preference-oriented service ranking, fair and open QoS computation. The QoS model in this technique is designed to evaluate the QoS of web services without changing the computational model. Genetic algorithms approaches for service composition (Canfora *et al*., 2005) are proposed by Gerardo Canfora. His research focused on how genetic algorithms best suits for service composition when compared to integer programming. Continuous GA has been applied to various fields of science from physics, mathematics to medical sciences.

Kuo *et al*. (2008) proposed continuous genetic algorithm based fuzzy neural network for learning fuzzy IF-THEN rules. This approach uses CGA to enhance its performance. Young *et al*. (2007) proposed continuous GA to assimilate sensor data for a toxic contaminant release. Here, the speed is tuned by optimizing the parameters of the CGA.

STRUCTURAL METRIC BASED CATEGORIZATION TECHNIQUE

Coupling between services comes in various forms. The composition of various service component results in a low coupling and high decoupling if the component of each service satisfies its properties such as independent, stateless and self-contained.

Service component by theoretical is a stand-alone uni but in reality service component depend on other service component they require or depend on component itself has its state (Yacoub *et al*., 2000). The interaction among services exists in two stages namely interaction state indirectly and keep state directly. This study proposes two separate metrics namely Mean Service Relentless Colony Metric (MSRCM) and Mean Service Transitive Colony Metric (MSTCM) for this. These metrics are used in formulating fitness function for CGA to produce optimized service categorization technique for effective web service composition.

Continuous genetic algorithm: The initial step in web service categorization is to locate the services. This is called as service discovery. Service discovery deals with the process of locating or discovering related service descriptions that describes a particular web service using the Web Service Description Language (WSDL). Whereas service categorization deals with choosing a service implementation among the located services to satisfy the customer need. After being the services are located based on service discovery for the user request Q, the composite services for Q can be defined as abstract services of WS_1, WS_2 ...WS_n and is shown in Eq. 1:

$$Q = \{WS_1, WS_2, WS_3 \ldots WS_n\} \quad (1)$$

Each web services in turn may have concrete services with WS_{11}, WS_{12}, WS_{13} ….. WS_{1n}. Therefore, services WS1 to WSn can be represented each as a single chromosome.

Representation of chromosomes and population generation: A chromosome is represented by a string of variable length. To each element in the string, i.e., a gene, the summation of coupling metric value is assigned according to a floating point number coding scheme (Xu *et al.*, 2004).

Therefore, genetic operators such as selection, reproduction and mutation suitable for real numbers are employed. Figure 1 shows the set of chromosomes represented as collection of abstract services and Fig. 2 shows the floating point representation:

$$\text{CoupM (Q)} = \Sigma^{n}_{j=1} \text{CoupM (q)} \quad (2)$$

Coupling metrics values CoupM (Q) in gene (a single service) of the initial population are generated by summation of CoupM (q_1)-CoupM (q_n) according to the Eq. 2.

Where; CoupM (Q_1) … CoupM (Q_n) denote the coupling metrics or parameters of each abstract service WS_1, WS_2, WS_3 … Ws_n. The population generated is considered to select a sub-set of individuals for further evolution. This process is called selection and the most common selection process would be to always select the top scoring chromosomes and execute it. This study uses roulette-wheel selection. With this selection process even the chromosome with a lower fitness has a chance to be selected. In this selection, parents are selected according to their fitness. The better the chromosomes are the more chances to be selected they have. For example, four different abstract services with fitness equaled to 2.5, 2.1, 4.3 and 1.1 are placed on the wheel for selection. The selection is then based on randomly generated number. For web service selection, the steps for applying roulette-wheel can be formulated as:

- Calculate sum of all chromosome finesses in population-sum S (is performed only once for each population)
- Generate random number from interval (0, S)-r
- Go through the population and sum fatnesses from 0-sum s. When the sum s is >r, stop and return the chromosome

The selection of individuals will be crossed by a crossing operator. With services, this means that two particular chromosomes will be combined together to form a new ones. Thus results in new offspring. Here, the crossover operator used is standard two-point crossover. With 2 point crossover, 2 points are selected randomly on the parent chromosomes. Then, every gene between the 2 points is swapped between the parent chromosomes. With 2 service example, the 2 point crossover is applied to randomly select 2 abstract services and thus yielding 2 new abstract services. The commonly applied mutation operator randomly selects a service (i.e., a position in the genome) and randomly replaces the corresponding with another service one among those available. But with continuous genetic algorithm where the values are real numbers, mutation is applied by adding or subtracting a small number to randomly selected gene. The fitness function is a function that evaluates the closeness of every chromosome to the optimal solution by calculating value obtained. The idea behind this is to minimize the coupling among service components for effective service composition. Therefore, in general the fitness function is:

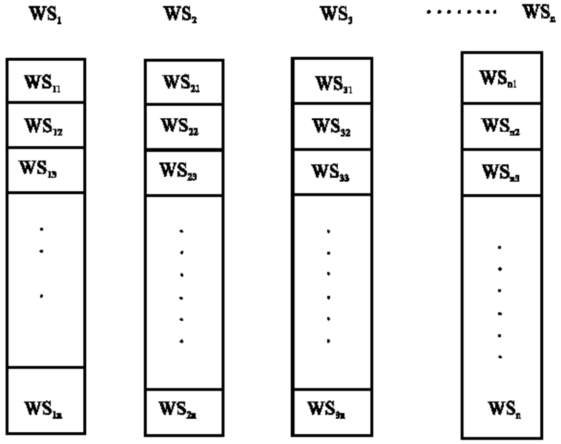

Fig. 1: Service representation as chromosomes

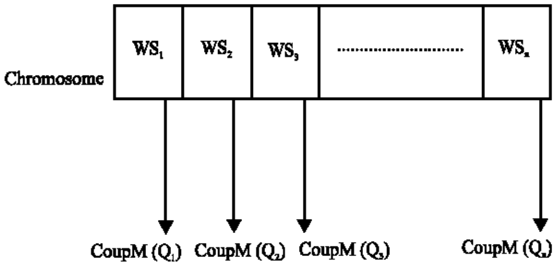

Fig. 2: Real value representation

$$\text{Min ColoMet} = f\text{ (MSRC, MSTC)} \quad (3)$$

Where:
ColoMet α = Mean service relentless colony
ColoMet α = Mean service transitive colony

ColoMet is directly proportional to mean service relentless colony metric and mean service transitive colony metric. The lower the service colony metrics as fitness function in genetic algorithm, the lesser the coupling among service components and achieve better maintainability, reliability, testability and reusability of services.

SERVICE COLONY METRICS

The techniques and measuring of existing coupling/colony metrics are classified by procedural programming and object oriented programming. This study proposes a two decoupling metrics namely Mean Service Relentless Colony Metric (MSRCM) and Mean Service Transitive Colony Metric (MSTCM) based on para meters of service stateness and interaction.

Mean service relentless colony metric: The relentless colony metric checks whether there exists a weighted link between service components. If there exist a link, the binding time is calculated. The stateness of a service can be classified as stateful or stateless. Stateful service defines that the service is aware of who made the request and remember what happened in the past. The stateless service component can research in request/response mode without knowing the knowledge of where the request has come from. For stateful service to be aware of other, it needs to have a common registry or belongs to same domain to track the status of interaction among service components. In general stateful services have a bind (dependency) with its own service component state or share with other service component. The dependency is defined as:

$$\mathrm{MD} = \frac{1}{i}\sum_{j=1}^{i} C \tag{4}$$

Where n is the number of service components in the domain and Cj is the service component which has its state. If it is stateful or have link then Cj = 1 otherwise Cj = 0. It means that if Cj = 1 then there exist dependency among service components and the domain is same.

Therefore, the service components are dependent and tightly coupled. In general, the lower the dependency is the looser the coupling between service components. There also exist indirect state binding in which a number of service components uses a same state. This is known as relentless dependency and is defined as:

$$\mathrm{MSRC} = \frac{1}{i}\sum_{k=1}^{i}\sum_{l=1}^{j} R_{kl} \tag{5}$$

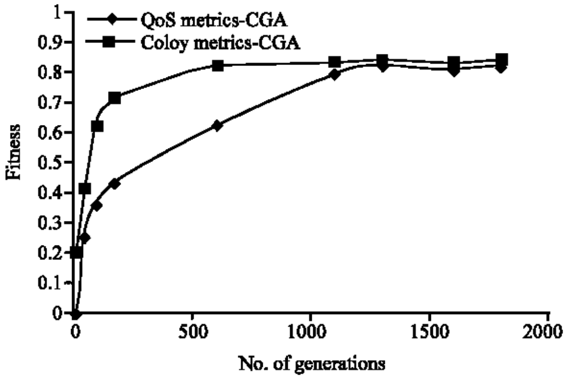

Fig. 3: Comparison of service categorization using colony metrics and QoS metrics

Where:

R_{kl} = The average participation time for a service component n to tie with other service component or relentless data j indirectly. The lower the mean relentless dependency is the lesser the coupling among service components
k = Service component
l = Relentless data
RD = Relentless dependency
n = Number of service components
j = Number of relentless repository

Mean service transitive colony metric: Any service oriented software system comprises of set of services defined as:

$$S = \{S_1, S_2 \ldots S_n\} \tag{6}$$

Where S is abstract service component which may be either aggregation of concrete service components or service composition components that lead to containment composition which is shown in Fig. 3. Let Cj = {Cj_1, Cj_2 .....Cj_m} be the concrete service components of the composition component Sj. The strength of coupling between pair of services can be formulated by finding the number of dependency paths between the service components.

Let CCji represents the set of dependency paths between service component Si that have interaction with service component Sj forj≠i (Cjj is defined to be null). The interaction from Sj-Si exists if and only if CCjj is not null. Thus, it reflects the direct coupling of one service to another. The interaction in directed service Cji is not necessarily equal to CCij. CCj the set of all dependency paths from service component Sj to all other service components can be defined as:

$$CC_j = U_{1\le i\le n}\, CC_{ji} \tag{7}$$

Now we have to associate a weight with each dependency pairs that reflects direct coupling from one service to another. This weight should also reflect the fact that a service interacts many concrete services that has a greater chance of interacting services from any composition component. The direct colony metric of service component S_i-S_j is defined as:

$$DC\ (i, j) = CC_{ij}/CC_i + C_i \qquad (8)$$

Where CCi+Ci indicate the total number of dependency paths exists from service component Si. The value of dependency colony metric ranges from 0-1. That is:

$$0< = DC\ (i, j) < = 1 \qquad (9)$$

To measure the transitive dependency pairs among concrete services of composite service components, indirect coupling between services are included. Suppose that DC (i, j) and DC (j, k) have dependency but that DC (i, k) is zero. Thus although, there is no direct coupling between service Si and service Sk, there is a dependency because there exist a path from service Si to service Sj which in turn have a path to service Sk. Since the strength of this dependency depends on two direct coupling of which it is composed, the measure is provided by DC (i, j) *DC (j, k). That is the strength of the coupling is the product of all DC paths. Thus, TC (i, j, π), the transitive coupling between service Si and service Sj is defined as:

$$TC\ (i, j, \pi) = \prod es, t\ c\ \pi\ CCs, t/CCs+Cs \qquad (10)$$

Where es, t denotes the path between s and t. In general there may be more than one path having a non-zero TC dependency between any two services. This raises a issue how paths may be combined to produce an overall measure of the coupling between services. The solution is to select the path with largest transitive dependency and define mean transitive dependency (i, j), the strength of coupling between two service component Si and service component Sj is defined as:

$$MSTC\ (i, j) = 1/n2\text{-}n\ TC\ (i, j, \pi_{max}) \qquad (11)$$

Where π is the set of all paths from service Si to service Sj. With the discussed service colony/coupling metrics, the final shape of the fitness function can be structured as:

$$\begin{aligned}MinColoMet = f(MSRC = \frac{1}{n}\sum\nolimits_{l=1}^{j} R\frac{1}{n}\\ \sum\nolimits_{k=1}^{n}\sum\nolimits_{l=1}^{j} R_{kl} * \frac{1}{n_2} - nTC(i, j, \prod\nolimits_{max}))\end{aligned} \qquad (12)$$

EXPERIMENTAL RESULTS

To test the performance of the service colony metrics as a fitness function in continuous genetic algorithm based service categorization, a prototype model was built for e-Learning web services management. Each course materials or documents of e-learning application was developed as web services and registered in UDDI registry.

The platform used for implementation process was microsoft .NET 2005. MySQL 5 was used as database for UDDI registry. This prototype starts working by getting the user query for selecting e-document. After the query has been given by the user, the system locates the related services. For example, the user query computer networks, the system generates the related services by semantically matching it with the domain ontology created. The e-Learning domain ontology was created with the use of Protege 3.1.

The identified service colony metrics measures the external quality of located services and initial population for genetic algorithm is computed. The fitness of each service is calculated based on the Eq. 12.

The performance of the proposed technique was analyzed by observing the fitness factors over the population generation and compared with QoS (non-functional quality of service) aware service categorization using continuous genetic algorithm which we developed. It is proven that service colony metrics based service categorization performs well than QoS aware service categorization technique. After a total of 2000 generations, no improvement in fitness was found and the fitness was observed. The Fig. 3 shows the service catgorization performance.

CONCLUSION

In this study, two service coupling/colony metrics are proposed for web service categorization. Service properties such as direct dependency and indirect dependency are discussed and are used as fitness function in continuous genetic algorithm.

The performance of service categorization for effective service composition out performs the earlier QoS based service categorization technique and produces a near optimal solution.

RECOMMENDATIONS

The future enhancement aims to identify more number of service colony metrics for measuring the quality of a service in all aspects of service properties. Work-in-progress is also devoted to apply memetic algorithm instead of CGA.

REFERENCES

Canfora, G., M.D. Penta, R. Esposito and M.L. Villani, 2005. An approach for QoS-aware service composition based on genetic algorithms. Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, June 25-29, ACM, New York, pp: 1069-1075.

Chidamber, S.R. and C.F. Kemerer, 1994. A metric suite for object-oriented design. IEEE Trans. Software Eng., 20: 476-493.

Choi, M. and J. Lee, 2007. A dynamic coupling for reusable and efficient software system. Proceedings of the IEEE 5th International Conference on Software Engineering Research, Management and Applications, Aug. 20-22, Busan, South Korea, pp: 720-726.

Chua, H.N. and S.M.F.D.S. Mustapha, 2006. Web services selection based on multiple-aspect similarity function. Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Dec. 18-22, Hong Kong, China, pp: 605-609.

Haupt, R.L. and S.E. Haupt, 2004. Practical Genetic Algorithms. 2nd Edn., John Wiley and Sons, New Jersey, ISBN: 0471455652.

Kaufmann, A., M.M. Gupta and B. Esposito, 1991. Introduction to Fuzzy Arithmetic. Van Nostrand Reinhold Company, New York, pp: 384.

Klusch, M. and P. Kapahnke, 2008. Semantic web service selection with SAWSDL-MX. German Research Center for Artificial Intelligence. http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-416/paper1.pdf.

Kubalik, J., 2005. Using genetic algorithms with real-coded binary representation for solving non-stationary problems. Adaptive Nat. Comput. Algorithms, Part II: 222-225.

Kuo, R.J., S.M. Hong, Y. Lin and Y.C. Huang, 2008. Continuous genetic algorithm-based fuzzy neural network for learning fuzzy IF-THEN rules. J. Neurocomput., 71: 2893-2907.

Li, W. and S. Henry, 1993. Object oriented metrics that predict maintainability. J. Syst. Software, 23: 111-122.

Liu, Y., A.H. Ngu and L.Z. Zeng, 2004. Qos computation and policing in dynamic web service selection. Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers and Posters, May 19-21, New York, USA., pp: 66-73.

Perepletchikov, M., C. Ryan, K. Frampton and Z. Tarim, 2007. Coupling metrics for predicting maintainability in service-oriented designs. Proceedings of the 18th Australian Software Engineering Conference, April 10-13, Melbourne, VIC Australia, pp: 329-340.

Prasad, L. and A. Nagar, 2009. Experimental analysis of different metrics (Object-Oriented and Structural) of software. Proceedings of the 1st International Conference on Computational Intelligence, Communication Systems and Networks, July 23-25, Indore, pp: 235-240.

Qian, K., J. Liu and F. Tsui, 2006. Decoupling metrics for services composition. Proceedings of the 1st IEEE/ACIS International Conference on Computer and Information Science, July 10-12, Honolulu, HI USA., pp: 44-47.

Quynh, P.T. and H.Q. Thang, 2009. Dynamic coupling metrics for service oriented software. Int. J. Comput. Sci. Eng., 3: 282-287.

Shen, H., S. Zhang and J. Zhao, 2008. An empirical study of maintainability in aspect-oriented system evolution using coupling metrics. Proceedings of the 2nd IFIP/IEEE International Symposium on Theoretical Aspects of Software Engineering, June 17-19, Nanjing, Jiangsu China, pp: 233-236.

Shin, D.H., K.H. Lee and T. Suda, 2009. Automated generation of composite web services based on functional semantics. Web Semantics: Sci. Services Agents World Wide Web, 7: 332-343.

Sirin, E., B. Parsia and J. Hendler, 2004. Filtering and selecting semantic web services with interactive composition techniques. IEEE Trans. Intell. Syst., 19: 42-49.

Xu, Z., Z.Y. Dong and K.P. Wong, 2004. Genetic algorithm-based spinning reserve dispatching in a competitive electricity market. Int. J. Syst. Sci., 35: 847-853.

Yacoub, S.M., H.H. Ammar and T. Robinson, 2000. A methodology for architectural-level risk assessment using dynamic metrics. Proceedings of the 11th International Symposium on Software Reliability Engineering, (ISSRE'00), IEEE Xplore, pp: 210-221.

Young, G.S., C.T. Allen and S.E. Haupt, 2007. A genetic algorithm method to assimilate sensor data for a toxic contaminant release. J. Comput., 2: 85-93.

Zeng, L. and B. Benatallah, 2004. Qos-aware middleware for web service composition. IEEE Trans. Software Eng., 30: 311-327.

Zhang, B., Y. Shi and X. Xiao, 2006. A Policy-based adaptation method for service composition. Proceedings of the 1st International Symposium on Pervasive Computing and Applications (SPCA'06), Urumchi, China, pp: 619-624.