Handling Failures in Semantic Web Service Composition Through Replacement Policy in Healthcare Domain

Lakshmana Kumar Ramasamy[1], Seifedine Kadry[2], M. Amala Jayanthi[3], Wei Wei[4], Seungmin Rho[5]

[1] Department of Computer Applications, Hindustan College of Engineering and Technology, India
[2] Department of Mathematics and Computer Science, Beirut Arab University, Lebanon
[3] Department of Computer Applications, Kumaraguru College of Technology, India
[4] College of Computer Science and Engineering, Xi'an University of Technology, China
[5] Department of Software Engineering, Sejong University, Korea
research.laksha@gmail.com, s.kadry@bau.edu.lb, amalajayanthi.m.mca@kct.ac.in,
weiwei@xaut.edu.cn, smrho@sejong.edu

Abstract

Consistency of web service composition is a challenge for developing business applications. As web services are naturally changeable, the way to deliver consistent web services composition over unreliable web services could pose a significant problem. We propose a framework for semantic web services in healthcare domain that automatically performs web service discovery, composition and quality of service assurance, and, performs error handling through the replacement policy and fault-tolerant composition of web services that mixes both exception managing and transaction approaches. The framework enables the development of personalized healthcare systems.

Keywords: Semantic web, Web service, Service composition, Quality of service, Healthcare

1 Introduction

Service-Oriented Computing (SOC) promoted by Everything-as-a-service model has become a paradigm for designing, and delivering web applications resulting in the explosive growth of web services on the Internet [1]. Web Services provide a practical approach for the machine-to-machine interaction in a network. Web services provide a paradigm and help to develop distributed applications through its engineering design and also it provides many add-on services for accessing business applications remotely. These paradigms provide a new insight for the development of Service Oriented Architecture (SOA) [2]. In SOA, web services are communicated through Simple Object Access Protocol (SOAP) protocol and the web services are hosted in a universal registry known Universal Description, Discovery, and Integration (UDDI) where all the service providers host their web service endpoints.

In the era of SOC, Semantic Web plays a vital role and it is the extension of the conventional internet with the distinction that focuses importance on abstract knowledge [3] Semantic Web provides common formats for the interchange of information. It additionally provides a standard language for recording however knowledge relates to world objects. Semantic Web is that the new generation internet that tries to represent data such it will be employed by machines for automation, integration, and reprocess across applications. Moreover, Semantic Web is expressly declaring the information embedded in web based applications, providing semantics based access to the net and extracting huge amounts of data available from wireless sensor networks and Internet-of-Things (IoT) [4].

Semantic web is largely based on and build upon ontologies, shared representations of all domain concepts and their relationships, which are very useful for information retrieval tasks [5], automated development of web application components and web services [6], and developing vocabularies for business models [7]. Ontologies are commonly developed using Web Ontology Language (OWL), which provides a standard that supports well in semantic search engines. Web service discovery and composition requires significant effort for solving composite tasks [8] especially when a vast amount of services are widely available. In Semantic Web Service Discovery based on Ontology Mapping (SWSDOM), web services are discovered by aggregating linguistic similarity, structural similarity and instance similarity [9]. The major challenge faced in the service-oriented approach is how to automate Web service composition. The challenge needs to address different methods; tools and it should also be low-cost, time efficient and easy-to-use [10].

*Corresponding Author: Seungmin Rho; E-mail: smrho@sejong.edu
DOI: 10.3966/160792642020052103011

Related approaches for automated web service composition include the use of Formal Concept Analysis (FCA) to classify web services, and generate a class hierarchy of similar web services, which are further clustered to enable semantic matching, and generation of the composition plans [11]; representation of web services as conditional directed acyclic graph and using centrality measure of social networks to rank web services for composition based on their trust rating [12]; Petri internet model for interactions between net services and building ontologies that represent domain knowledge that facilitates the mixture of the Petri internet models with semantics-based mediation [13], and fuzzy logic based search [14]; and the genetic algorithm based approach that can be used for both functional and the class testing [15].

The problem of finding web services most fitting for composition can be addressed using machine learning and heuristical optimization techniques such as Ant Colony Algorithm [16], Harmony Search [17], multi-step optimization [18], and natural language requests [19-20]. Cloud patterns [21] can be employed to match interfaces and automate the composition process, while cloudification helps to improve performance of web services [22]. Providing on-demand services in the cloud raises the need to map distributed resources with personalized service requests [23]. Using Markov Decision Process [24] can help to solve different optimization problems through dynamic programming and reinforcement learning. The selection of the continuously changing service can be supported by ontology based on Quality of Service (QoS) and through this approach the different Services can work together towards improving the overall quality of the service [25].

Selection of appropriate rules for web service composition requires to consider the QoS characteristics of web services [26]. Transactional properties guarantee consistent outcome and proper execution of the composite service. A versatile web service transaction model [27] satisfies the request for the point and permits flexible web service composition and recovery from by replacement of web services containing an unsuccessful web service. To enhance concurrency, the results of the submitted sub-transactions can be replaced for alternative synchronic executions of sub-transactions [28]. In [29], the authors addressed the problem of addressing service observation through QoS and conjointly the authors has planned a service observation system that is policy primarily based on who monitors the whole standing of the service. Pinto [30] evaluated the web service composition performance and the authors have also examined how the different tools can be integrated and the entire performance was examined.

Several testing techniques can be considered for web service testing such as penetration testing [31] and unit testing [32], and integration testing [33]. Kim & Hong [34] suggested a technique to improve the reliability for the web service composition through Business Process Execution Language (BPEL). This method uses aspect-oriented programming when the service fault occurs. If a service failure occurs, a set with different recovery polices was specified and these policies helps to handle and recover web service from the fault during the web service composition. These policies even help to monitor and manage web services [35]. Another approach employed UML for Services (UML-S) for a Model Driven Engineering (MDE) of Web services and their interactions Business Process Model and Notation (BPMN) provides notations for describing messaging and control flows in web service compositions. To ensure fault tolerance of web service, simulation-based reliability analysis can be used [37]. Through this approach, service oriented applications can change dynamically to attain their reliability and improve performance measures [38].

The use of web services in the healthcare domain is already high and increasing. There are various reference architectures involving the service creation and composition in healthcare domain. One example is Parlay X that provides a set of standard Application Programming Interfaces (APIs) for web services for the network [39]. The integration of semantic web technology and healthcare systems is the subject of intensive research. Basically, in the healthcare IT infrastructure, the messages are usually shared between a service provider and a healthcare information broker, and also they may pass the information enclosed in those communications to other third healthcare parties. The information is received by the broker about the ease of use of medication associated records whenever a server creates or updates medication-related information. Then the broker keeps track of the database that has a list of parties where patient details can be obtained. When there is a need for a healthcare service provider to retrieve the medication history for a particular patient, then the system process a query to the broker by giving patient ID. The healthcare broker checks with all the third parties who have data related with medication for the searched patient, and it collects the data and reverts to the requested provider. So the broker acts as an intelligent messaging mediator, which can dynamically gather, process and analyse data. Healthcare providers are forced to increase their productivity of the current support systems due to explosive expansion driven by internal growth and acquisition. The number of services, new services built over existing and new infrastructure, are considered to be the main factors behind the complexity growth.

Healthcare applications provide complex services to the users and to the businesses in which the outcome can not be achieved by a single web service. A composite web service framework is needed to solve a complex functionality, through composing multiple web services from various service providers. However, a rate of failure in composite service is much higher than a rate of failure in an atomic service. During execution of the composite web service, if one service fails or is not available, then the entire service composition fails. To address this problem, we propose an error handling mechanism for web service composition through the replacement policy for the healthcare applications domain.

This paper is structured as follows: Section 2 illustrates the importance of bringing web services into healthcare domain and analyzes different approaches proposed by various researchers. Section 4 details about the proposed methodology, algorithms and architecture for web service composition. Section 5 narrates the system implementation for execution of composed web services in different scenarios. Section 6 presents conclusion and future works.

2 Web Service Composition Based on Telecom Approach

2.1 Business Model

Telecommunication Information Networking Architecture Consortium (TINA-C) provides a model that suits for every business enterprise, and the model was adaptable for various vendors who are in providing services and maintaining their relationship. The model provided the TINA business model for standardization of web services. Figure 1 displays about the TINA business model that provides a set of specifications, to maintain the standardization between the healthcare service provider and software industries [40]. This model helps to build a uniform platform for providing various healthcare services. The community users are able to build their own services or applications over a network virtualization framework without any complexities dealt with underlying network technologies by maintaining the functionality of network providers based on SOA. The building blocks of SOA are web services and workflows of web services.

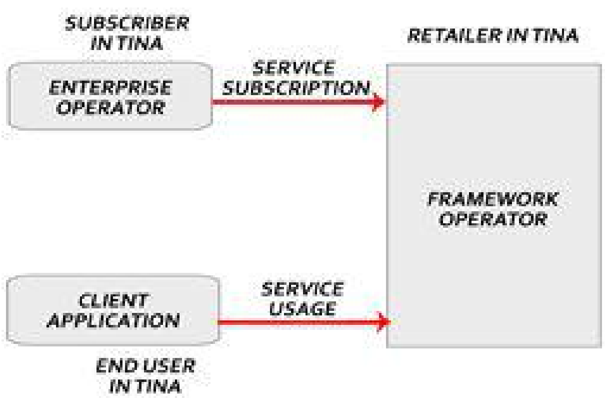

Figure 1. Telecom Information Networking Architecture (TINA)

Figure 2 provides a business model based on Parlay standards. Parlay X provides a group of API's for the software industries to access the functionalities of telecommunications. This model provides features for the users, operators and service provider to simply the Telecom application development. The services are accessed by the APIs by the different service vendors. Several hindrances are there on the way of accepting the web service technology, even though it promises the universal integration. Research is been in progress to address these hindrances that enables the adoption of this technology for the future. Click-to-dial architecture utilizes Parlay X web services for telephonic communications over Internet Protocol (IP) television (IPTV). With the implemented system, IPTV users can connect calls over hand set or telephones just with a button on the IPTV remote control [41-48].

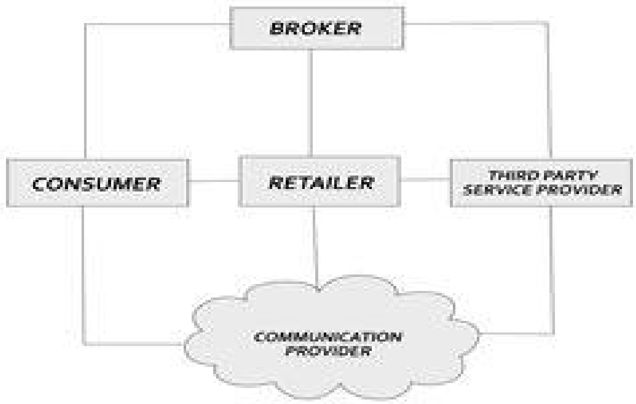

Figure 2. Parlay business model

2.2 Architecture

Figure 3 represents the proposed architecture. The broker was developed who acts as an agent for simplifying the communication between web service suppliers and requester to satisfy the user's requirements. The role of an agent is to find the right web services from the UDDI repository based on the requirements from the request. This process aids to solve the business process by composing the newly find web services and then prepares them for the execution. The monitoring is done during web service execution and if any service failure insists then the failed set of web services are determined and in parallel the equivalent service subsets are identified. Then the identified similar subsets are ranked as per the policy and the best equivalent subset is selected. The failed subsets are replaced by the new ones and thus provide the proper web service composition. As a result, the proposed architecture can overcome the faults during the execution of a complex web service.

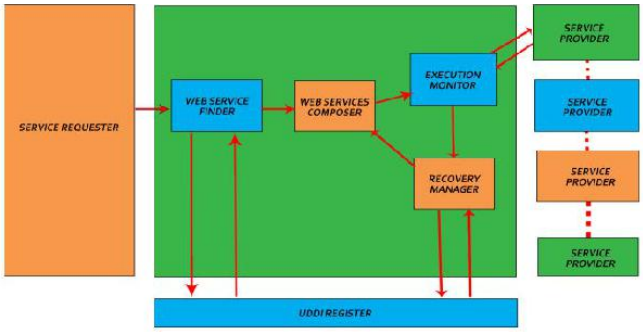

Figure 3. System architecture for web service composition

2.3 Algorithms

We propose three algorithms for identifying the failed web service subset, to find the equivalent subset, and to find the replacement service subset. The working principle of each algorithm is presented below.

Algorithm 1. Failed subset identification

1. Calculate the distinction set from the composite web service set and the failed web services.
2. Extract the subsets of all the remaining web services.
3. Calculate the Union of all the sets in the extracted remaining services set with the failed services set.

Algorithm 2. Equivalent subset identification

1. Formulates a rule to match the failed subset and the newly formed subset.
2. Identify QoS constraints for matching the failed service subset with the new set.
3. Match each QoS constraint with the new service set. If they are matched, the new set is considered to be an equivalent service subset.
4. Add an equivalent service set to the UDDI repository

Algorithm 3. Replacement subset identification

1. Search in UDDI repository to find the equivalent subset for the replacement.
2. Checked each web service was for the replacement and create manageable subset.
3. If the manageable subset is not formed, then scan for the QoS ranking of each web service.
4. Calculate the set distinction for the failed service subset and the available composite web services set.
5. Find the union of the equally identified set and the already calculated distinction set.
6. Create the new composite service set.

5 Experimental Validation

5.1 Simulated Environment

For the experiment, we have developed web services with the similar functionalities and deployed them on a several computers. After that we have developed a composite web service and deployed it on different computer.

The computers used for this experiment have the same configuration: Intel Core-i3 2.40 GHz (4 CPUs), 2 GB RAM, Microsoft Windows 7 professional 64 bit, and JDK 1.7. All the computers are connected through Local Area Network (LAN).

We have developed various versions of a fault-tolerant composite service, in which each version uses replacement policy and replica to handle the errors. To validate our technique, we inject faults within the system and simulate with various faults such as network fault, association loss and server crash fault. We use code insertion, code modification, timeout victimization, and timer fault injection techniques. If any fault happens within the execution of a composite net service then rather than composing entire service once more solely the failing service set is replaced.

To collect the results, the composite service is invoked 1,000 times and the response time is recorded and average time computed.

5.2 Results

Our model was analyzed with atomic service and composite service. Reliability factor and the composition time were measured. We have also analyzed the working process of our model during the failure. Table 1 gives the semantic similarity comparison between replica and replacement policy. The experiment was conducted for the reliability and composition time with including and excluding web service semantics. The results show that the replacement policy has proved 82% reliability through the inclusion of service semantics. The existing replica method can provide reliability of only 54% without using service semantics and 61% by enabling service semantics.

Table 1. Replica and replacement policy comparison through semantic similarity

Methods	Reliability (Composition Time, ms)	
	Without Semantics	With Semantics
Replica	54 % (270 ms)	61% (640 ms)
Replacement Policy	73% (344 ms)	82% (624 ms)

In Table 2, QoS constraints was added for replica service and the replacement policy and the results were compared. Without including the semantic and QoS, the replica can provide 54% reliability, and the replacement policy can achieve 79% reliability, but in case of semantic inclusion, the replica can achieve 61% reliability and the replacement policy can reach 86% reliability. From these results we can see that the replacement policy is far better than replica even without the inclusion of semantic. For both Table 1 and Table 2, we have done 10 iterations with various services and the results were averaged.

Table 2. Replica and Replacement policy comparison through QoS constraints

Methods	Reliability (Composition Time, ms)	
	Without Semantics & QOS Ranking	With Semantics & QOS Ranling
Replica	54 % (270 ms)	61% (640 ms)
Replacement Policy	79% (519 ms)	86% (783 ms)

In order to compare the efficiency of our proposed semantic model, we have done an experiment in a simulated environment with 150 web services taken from the Parlay X standards and the results are shown in Table 3.

Table 3. Efficiency comparison of the proposed and existing system

S. No	Existing w/o Semantics	Proposed work with Semantics
1	0.1	0.54
2	0.12	0.65
3	0.32	0.75
4	0.41	0.94
5	0.5	0.8

The experiment was conducted in 10 iterations, and each time 15 web services were executed and the time was measured. From the greater result in efficiency, we have found to include semantics in our proposed method.

In Figure 4, we have validated the accuracy of our proposed method we have conducted an experiment in a simulated environment for the service composition by including QoS metrics. We have conducted 10 iterations with 250 web services.

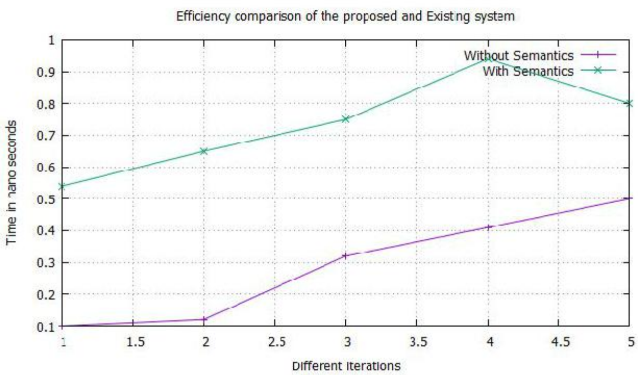

Figure 4. Efficiency comparison of the proposed and existing system

We have considered 1500 web services from the UDDI repository which are given by third party service providers for the healthcare domain. These services are then enhanced with semantic annotations then the composition was done. The standard of the web services was under the policy of ParlayX. The graph was plotted for different iterations and we have visualized the results for 5 iterations. The results are presented in Figure 5., which shows the composition time for both replica and replacement policy. Once the failure occurs, the time taken by the replica to composite the service is much higher than replacement policy. So in this scenario the replacement policy was considered to be the best option to handle the composition even after the failure occurs.

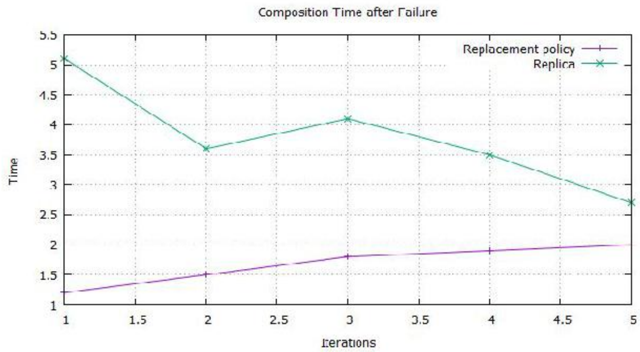

Figure 5. Composition time after service failure

Reliability of both replica and replacement policy methods was analyzed and found that the reliability of replacement policy was better and we have used 1500 web services to find the better reliability and the results are presented in Figure 6.

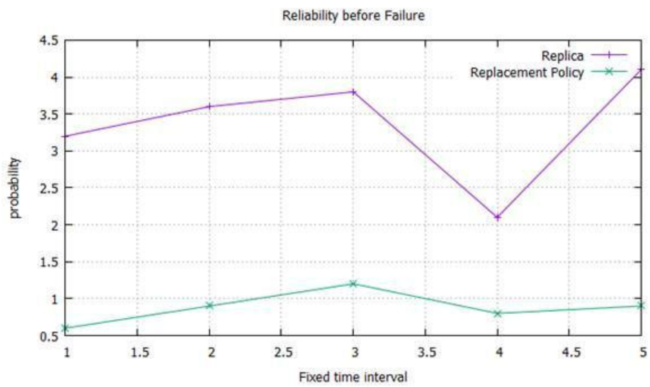

Figure 6. Reliability before service failure

In Figure 7, the reliability of the analyzed methods after the failure occurs is shown. We have identified that replacement policy was better than replica in all the iterations. For experiments, we have used the same set of web services as in case of the composition process.

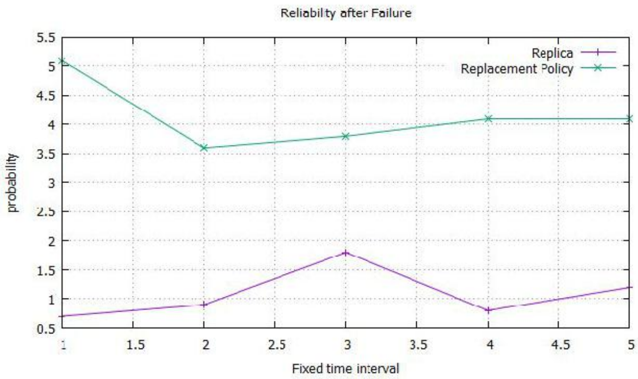

Figure 7. Reliability after service failure

6 Conclusion

Web service composition plays a vital role in the domain of web services which merely in need of automation especially in the discovery and selection of services. The proposed framework concentrates on the selection of the set of services that discovers web services to satisfy the user-specified requirements.

In this paper, we have proposed a novel approach for semantic web service composition model for the healthcare domain using the web service replacement policy. If a web service set fails at the execution, then the replacement service subsets are discovered dynamically.

The experimental results show that the proposed technique works well when compared to previous methods. The execution time is much faster than other random search approaches. The proposed technique reduces the amount of web service subsets by half. Summarizing, the planned technique considerably improves the success rate and execution time just in case of failures throughout execution of a composite service.

Therefore, applying replacement policy in the healthcare service domain provides a great effect on improving computation time when it comes for handling thousands of users and millions of service requests. This mechanism also helps the medical semantic web service to lead the system to determine, choose, and compose web services automatically.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019R1F1A1060668).

References

[1] F. Xie, J. Wang, R. Xiong, N. Zhang, Y. Ma, K. He, An Integrated Service Recommendation Approach for Service-based System Development, *Expert Systems with Applications*, Vol. 123, pp. 178-194, June, 2019.

[2] J. De Bruijn, M. Kerrigan, U. Keller, H. Lausen, J. Scicluna, The Web Service Modeling Ontology, *Modeling Semantic Web Services: The Web Service Modeling Language*, Springer-Verlag Berlin Heidelberg, 2008, pp. 23-28.

[3] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, *Scientific American*, Vol. 284, No. 5, pp. 34-43, May, 2001.

[4] M. Ahmadinia, A. Movaghar, A. M. Rahmani, Semantic Data Gathering of Physical Entities in Semantic Sensor Networks Using Software Agents, *Information Technology and Control*, Vol. 47, No. 2, pp. 167-183, June, 2018.

[5] O. Y. Sowunmi, S. Misra, N. Omoregbe, R. Damasevicius, R. Maskeliūnas, A Semantic Web-based Framework for Information Retrieval in E-learning Systems, *International Conference on Recent Developments in Science, Engineering and Technology (REDSET)*, Gurgaon, India, 2017, pp. 96-106.

[6] R. Damaševičius, V. Štuikys, J. Toldinas, Domain Ontology-based Generative Component Design Using Feature Diagrams and Meta-programming Techniques, *European Conference on Software Architecture (ECSA 2008)*, Paphos, Cyprus, 2008, pp. 338-341.

[7] V. Žitkus, R. Butkiene, R. Butleris, R. Maskeliunas, R. Damaševičius, M. Woźniak, Minimalistic Approach to Coreference Resolution in Lithuanian Medical Records, *Computational and Mathematical Methods in Medicine*, Vol. 2019, Article ID 9079840, March, 2019.

[8] P. Rodriguez-Mier, C. Pedrinaci, M. Lama, M. Mucientes, An Integrated Semantic Web Service Discovery and Composition Framework, *IEEE Transactions on Services Computing*, Vol. 9, No. 4, pp. 537-550, July-August, 2016.

[9] Y. Shi, G. Li, J. Li, Y. Li, Framework of Semantic Web Service Discovery Based on Ontology Mapping, *International Conference on Research Challenges in Computer Science (ICRCCS'09)*, Shanghai, China, 2009, pp. 77-80.

[10] B. Medjahed, A. Bouguettaya, Enabling Interactions on the Web: A Taxonomic Perspective, *Service Composition for the Semantic Web*, Springer New York, 2011, pp. 13-72.

[11] A. Abid, M. Rouached, N. Messai, M. Abid, T. Devogele, A Semantic Matching Engine for Web Service Composition, *International Journal of Business Information Systems*, Vol. 30, No. 1, pp. 92-108, January, 2019.

[12] S. Bansal, A. Bansal, G. Gupta, M. B. Blake, Generalized Semantic Web Service Composition, *Service Oriented Computing and Applications*, Vol. 10, No. 2, pp. 111-133, June, 2016.

[13] A. Marakhimov, J. Yim, J. Joo, Petri Net Based Semantic Web Service Composition, *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 9, No. 2, pp. 281-290, February, 2014.

[14] M. Madkour, D. el Ghanami, A. Maach, A. Hasbi Context-Aware Service Adaptation: An Approach Based on Fuzzy Sets and Service Composition, *Journal of Information Science and Engineering*, Vol. 29, No. 1, pp. 1-16, January, 2013

[15] E. Maythaisong, W. Songpan, Hybrid Testing Approach for Web Services Composition Using Genetic Algorithm, *2016*

IEEE Conference on e-Learning, e-Management and e-Services (IC3e), Langkawi, Malaysia, 2016, pp. 188-193.

[16] N. H. Rostami, E. Kheirkhah, M. Jalali, An Optimized Semantic Web Service Composition Method Based on Clustering and Ant Colony Algorithm, *International Journal of Web & Semantic Technology (IJWesT)*, Vol. 5, No. 1, pp. 1-8, January, 2014.

[17] A. Bekkouche, S. M. Benslimane, M. Huchard, C. Tibermacine, F. Hadjila, M. Merzoug, QoS-aware Optimal and Automated Semantic Web Service Composition with User's Constraints, *Service Oriented Computing and Applications*, Vol. 11, No. 2, pp. 183-201, June, 2017.

[18] P. Rodríguez-Mier, M. Mucientes, M. Lama, Hybrid Optimization Algorithm for Large-scale QoS-aware Service Composition, *IEEE Transactions on Services Computing*, Vol. 10, No. 4, pp. 547-559, July-August, 2017.

[19] J. Sangers, F. Frasincar, F. Hogenboom, V. Chepegin, Semantic Web Service Discovery Using Natural Language Processing Techniques, *Expert Systems with Applications*, Vol. 40, No. 11, pp. 4660-4671, September, 2013.

[20] S. S. Kamath, V. S. Ananthanarayana, Discovering Composable Web Services Using Functional Semantics and Service Dependencies Based on Natural Language Requests, *Information Systems Frontiers*, Vol. 21, No. 1, pp. 175-189, February, 2019.

[21] B. Di Martino, G. Cretella, A. Esposito, Cloud Services Composition through Cloud Patterns: A Semantic-based Approach, *Soft Computing*, Vol. 21, No. 16, pp. 4557-4570, August, 2017.

[22] R. Buzys, R. Maskeliunas, R. Damaševičius, T. Sidekerskiene, M. Woźniak, W. Wei, Cloudification of Virtual Reality Gliding Simulation Game, *Information*, Vol. 9, No. 12, Article No. 293, December, 2018

[23] Y. Lu, X. Xu, A Semantic Web-based Framework for Service Composition in a Cloud Manufacturing Environment, *Journal of Manufacturing Systems*, Vol. 42, pp. 69-81, January, 2017.

[24] S. S. Pillai, N. C. Narendra, Optimal Replacement Policy of Services Based on Markov Decision Process, *IEEE International Conference on Services Computing*, Bangalore, India, 2009, pp. 176-183.

[25] E. M. Maximilien, M. P. Singh, A Framework and Ontology for Dynamic Web Services Selection, *IEEE Internet Computing*, Vol. 8, No. 5, pp. 84-93, September-October, 2004.

[26] J. El Haddad, M. Manouvrier, M. Rukoz, TQoS: Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition, *IEEE Transactions on Services Computing*, Vol. 3, No. 1, pp. 73-85, January- March, 2010.

[27] H. Saboohi, S. A. Kareem, Increasing the Failure Recovery Probability of Atomic Replacement Approaches, arXiv preprint arXiv:1211.6370, 2012.

[28] D.-P. Dang, X. Jiang, N. Wang, Y.-T. Yao, W.-B. Yao, Concurrency Control Of Real-Time Web Service Transactions, *Journal of Information Science And Engineering*, Vol. 34, No. 1, pp. 261-287, January, 2018.

[29] L. D. Chou, C. C. Chen, P. C. Tseng, Design and Implementation of a Policy-Based Monitoring System for Web Services, *Journal of Information Science and Engineering*, Vol. 25, No. 5, pp. 1357-1372, September, 2009.

[30] A. A. Pinto, O. A. S. Carpinteiro, B. G. Batista, D. M. L. Filho, M. L. Peixoto, B. T. Kuehne, A Performance Evaluation of an Automatic Web Services Composition System, In *2017 International Conference on High Performance Computing & Simulation (HPCS)*, Genoa, Italy, 2017, pp. 839-845.

[31] J. K. Alhassan, S. Misra, A. Umar, R. Maskeliūnas, R. Damaševičius, A. Adewumi, A Fuzzy Classifier-based Penetration Testing for Web Applications, *International Conference on Information Technology & Systems (ICITS 2018)*, Libertad, Ecuador, 2018, pp. 95-104.

[32] S. Misra, A. Adewumi, R. Maskeliūnas, R. Damaševičius, F. Cafer, Unit Testing in Global Software Development Environment, *International Conference on Recent Developments in Science, Engineering and Technology (REDSET 2017)*, Gurgaon, India, 2017, pp. 309-317.

[33] D. Barisas, E. Bareiša, Š Packevičius, Automated Method for Software Integration Testing Based on UML Behavioral Models, *International Conference on Information and Software Technologies (ICIST 2013)*, Kaunas, Lithuania, 2013, pp. 272-284.

[34] J. P. Kim, J. E. Hong, Dynamic Service Replacement to Improve Composite Service Reliability, *2011 Fifth International Conference on Secure Software Integration and Reliability Improvement (SSIRI)*, Jeju Island, South Korea, 2011, pp. 182-188.

[35] A. Erradi, P. Maheshwari, V. Tosic, Recovery Policies for Enhancing Web Services Reliability, *International Conference on Web Services (ICWS'06)*, Chicago, IL, USA, pp. 189-196.

[36] C. Dumez, A. Nait-sidi-moh, J. Gaber, M. Wack, Modeling and Specification of Web Services Composition Using UML-S, *2008 4th International Conference on Next Generation Web Services Practices*, Seoul, South Korea, 2008, pp. 15-20.

[37] Y. Shu, Z. Wu, H. Liu, Y. Gao, A Simulation-Based Reliability Analysis Approach of the Fault-Tolerant Web Services, *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, Bangkok, Thailand, 2016, pp. 125-129.

[38] Z. Zheng, M. R. Lyu, A QoS-aware Middleware for Fault Tolerant Web Services, *2008 19th International Symposium on Software Reliability Engineering (ISSRE)*, Seattle, WA, USA, 2008, pp. 97-106.

[39] H. Lofthouse, M. J. Yates, R. Stretch, Parlay X Web Services, *BT Technology Journal*, Vol. 22, No. 1, pp. 81-86, January, 2004.

[40] P. Prozeller, TINA and the Software Infrastructure of the Telecom Network of the Future, *Journal of Network and Systems Management*, Vol. 5, No. 4, pp. 393-410, December, 1997.

[41] U. Sedlar, L. Zebec, J. Bester, A. Kos, Bringing Click-to-dial Functionality to IPTV Users, *IEEE Communications Magazine*, Vol. 46, No. 3, pp. 118-125, March, 2008.

[42] W. Wei, Q. Xu, L. Wang, X. H. Hei, P. Shen, W. Shi, L. Shan,

GI/Geom/1 Queue Based on Communication Model for Mesh Networks, *International Journal of Communication Systems*, Vol. 27, No. 11, pp. 3013-3029, November, 2014.

[43] P. Zheng, Y. Qi, Y. Zhou, P. Chen, J. Zhan, M. R.-T. Lyu, An Automatic Framework for Detecting and Characterizing Performance Degradation of Software Systems, *IEEE Transactions on Reliability*, Vol. 63, No. 4, pp. 927-943, December, 2014.

[44] H. Dou, Y. Qi, W. Wei, H. Song, A Two-time-scale Load Balancing Framework for Minimizing Electricity Bills of Internet Data Centers, *Personal and Ubiquitous Computing*, Vol. 20, No. 5, pp. 681-693, October, 2016.

[45] P. Wang, Y. Qi, X. Liu, Power-Aware Optimization for Heterogeneous Multi-tier Clusters, *Journal of Parallel and Distributed Computing*, Vol. 74, No. 1, pp. 2005-2015, January, 2014.

[46] Y.-N. Qiao, Q. Yong, H. Di, Tensor Field Model for Higher-order Information Retrieval, *Journal of Systems and Software*, Vol. 84, No. 12, pp. 2303-2313, December, 2011.

[47] J. Yan, Y. Qi, Q. Rao, Detecting Malware with an Ensemble Method Based on Deep Neural Network, *Security and Communication Networks*, Vol. 2018, Article ID 7247095, March, 2018.

[48] X. Wang, Y. Qi, Z. Wang, Y. Chen, Y. Zhou, Design and Implementation of SecPod: A Framework for Virtualization-based Security Systems, *IEEE Transactions on Dependable and Secure Computing*, Vol. 16, No. 1, pp. 44-57, January-February, 2019.

Biographies

Lakshmana Kumar Ramasamy is working as an Assistant professor & technical lead in Hindusthan College of Engineering and Technology, Coimbatore. Tamil Nadu. He is a global chapter Lead for MLCS (Machine Learning for Cyber Security) for the Coimbatore chapter. He has a passion for software development and holds the International certification on SCJP (Sun Certificated Java Programmer) and SCJWCD (Sun Certificate Java Web Component Developer. He also holds the Data Science certification from John Hopkins University and also the Amazon Cloud Architect certification from Amazon Web Services. His keen interest is to write special issues for publishers like Elsevier, Inderscience, Springer and IGI-global and he holds publications in good IF journals.

Seifedine Kadry is working as an Associate Professor at Beirut Arab University, Lebanon. He received the bachelor's degree in applied mathematics from Lebanese University, in 1999, the M.S. degree in computation from Reims University, France, and EPFL, Lausanne, in 2002, the Ph.D. degree from Blaise Pascal University (France), in 2007, and the HDR degree in engineering science from Rouen University in 2017. His current research interests include education using technology, smart cities, system prognostics, stochastic systems, and probability and reliability analysis. He is a Fellow of IET, ACSIT, and ABET program evaluator.

M. Amala Jayanthi is working as assistant professor in the Department of Computer Applications, Kumaraguru College of Technology, Coimbatore. She has 7 years of experience as academician. She received her Master of philosophy in Department of Computer Science, St. Joseph's College (Autonomous), Bharathiyar University, Tiruchirappalli, Tamil Nadu, India. She received her MCA degree from Kalasalingam University. Her current area of research is Educational Data mining and Data Analytics. She has published 11 research papers in various reputed journals.

Wei Wei received the M.S. and Ph.D. degrees from Xi'an Jiaotong University, in 2005 and 2011, respectively. He is currently an Associate Professor with the Xi'an University of Technology. His research interests include wireless networks and wireless sensor networks application, mobile computing, distributed computing, and pervasive computing.

Seungmin Rho received the M.S. and Ph.D. degrees in computer science from Ajou University, South Korea, in 2003 and 2008, respectively. He visited Multimedia Systems and Networking Laboratory, University of Texas at Dallas, from 2003 to 2004. Before, he joined the Computer Sciences Department, Ajou University. From 2008 to 2009, he was a Postdoctoral Research Fellow with the Computer Music Laboratory, School of Computer Science, Carnegie Mellon University. He was a Research Professor with the School of Electrical Engineering, Korea University, from 2009 to 2011. In 2012, he was an Assistant Professor with the Division of Information and Communication, Baekseok University. He is now a Faculty Member with the Department of Software, Sejong University, South Korea. His current research interests include database, big data analysis, music retrieval, multimedia systems, machine learning, knowledge management, and computational intelligence. He has published 100 papers in refereed journals and conference proceedings in these areas. He has been involved in more than 20 conferences and workshops as various chairs and more than 30 conferences/workshops as a Program Committee Member. He has edited a number of international journal special issues as a Guest Editor, such as Multimedia Systems, Information Fusion, Engineering Applications of Artificial Intelligence, New Review of Hypermedia and Multimedia, Multimedia Tools and Applications, Personal and Ubiquitous Computing, Telecommunication Systems, and Ad Hoc and Sensor Wireless Networks. He has received a few awards, including Who's Who in America, Who's Who in Science and Engineering, and Who's Who in the World, in 2007 and 2008, respectively