Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/13S117

Amplification of Availability in Web Services by Earlier Detection of False Requests during Service-oriented Application Development

Priyadharshini Muthukrishnan[1,*] *and Baskaran Ramachandran*[2]

[1] Department of Computer Science and Engineering, KPR Institute of Engineering and Technology, Coimbatore, India
[2] Department of Computer Science and Engineering, College of Engineering, Anna University, Chennai, India

Received: 2 Jun. 2019, Revised: 22 Jul. 2019, Accepted: 26 Jul. 2019
Published online: 1 Aug. 2019

Abstract: The service computing that originated from parallel and distributed computing is the key of today's enterprise computing. Service-oriented application development aims at provisioning loose coupling nature in enterprise applications. Cloud computing is the key technology emerged as realization of service computing which revolves around issues of scalability to a higher degree. Similarly the web services that are found to be yet another form of realization of service computing at most cases lead the service providers to stay unavailable to handle requests causing Denial of Service (DoS). Availability of such web service is an important factor that could solve the issue of scalability. The proposed work Availability Amplification System for Service Oriented Application Development (AASystemforSOAD) is aimed at earlier detection of false requests and thus amplifies the availability of the services at service provider's end.

Keywords: Cloud computing, Denial of service, availability, Service computing, Web service

1 Introduction

Service Oriented Architecture (SOA) is an architectural concept that is incorporated during the early phases of the development life cycle of service-oriented enterprise application development. SOA could be incorporated in form of IT applications where distributed ownership [10] is found to be suitable for the business process and hence it is used for real-time applications. An iterative approach of service-oriented application development with six overlapping phases is identified for the enterprises [8]. Quality of Service (QoS) factors such as security, response time, availability etc., when handled at earlier phases of application development is found to yield better amplification.

The initial phase of service-oriented application development focuses on tasks including collecting business requirements, scoping, identifying expected business goals and proposing required changes to the business operations. In the preceding phase building SOA blueprint SOA architecture for the enterprise is defined [17]. The phase of service enablement does the implementation of SOA, followed by service integration phase implementing the Enterprise Service Bus (ESB) [11]. The next phase is service orchestration where the services created are orchestrated with execution engine and finally enterprise presentation phase is devised and governance process in place is represented in Figure 1. Service enablement phase is responsible for building the services layer that includes service descriptions, contracts, and policies in design perspective but defines the runtime capabilities of services. The runtime capabilities include the security, access control descriptions for services and QoS of services that are defined at service layer and are supported by governance layer.

The resource and service available on request at any time is termed as availability. Denial of Service (DoS) is a common issue related to availability [18]. Service availability is one among the important factors for deciding the quality of service. Frequently unavailable service may have ill effects on the reputation of the service provider, or results in loss of business opportunities.

* Corresponding author e-mail: priyadharshini.m1977@yahoo.com

Fig. 1: Service-oriented Enterprise Application Development

The proposed work AASystemforSOAD is an incorporation done at the analysis, design and implementation phases of service-oriented application development that aims at the availability amplification. The below motivation section sketches the motivation behind the proposed work and is followed by literature section that gives the outline of the various tools and techniques available for providing availability. The section "proposed work"elaborates on the design and implementation of the proposed research work aiming at availability amplification. Finally the results and discussion section give the summary of the result achieved and future scope of improvement.

World Wide Web Consortium (W3C) and Organization for the Advancement of Structured Information Standards (OASIS) are the standard bodies that have devised various WS-* standards addressing the security factors of the web services such as confidentiality, integrity and other factors including reliable messaging, routing, transactions. In particular WS-Security standard for security provides measures to secure web service request and response parameters as well as the information in header which is meant to be used for processing request and response parameters. WS-Security provides confidentiality and integrity, whereas eXtensible Markup Language (XML) processing done to provide confidentiality and integrity leaves DoS attack [7] leading to unavailability of services. Various frameworks are evaluated for security and it is found that Schema hardening and Schema validation could fend for various attacks leading space to DoS attacks which could be solved by streaming Simple Object Access Protocol (SOAP) messages [12]. Hence schema validation that follows an ordering mechanism in the process of reference creation of signatures is proposed to address the above issues [13].

Web service request and response in form of SOAP messages exhibits a stateful behavior only on satisfying a set of constraints termed as "message contracts" for the corresponding service request. There are quite lot of runtime monitoring techniques discussed [2], [3], [4] that poses limitations such as implementing domain specific languages, monitoring at both client and server side, usage of Unified Modeling Language (UML) sequence diagrams for checking behavioral correctness, dynamic change in the runtime properties, state-by-state recording of traces, temporal logics with CTL−FO+ definitions, etc., Message Contract Validator (MCV) is the software component that has an observer that monitors the service invocation and execution as well as checking its conformity with a requirement specification using message contract mapper. Message Contract Language (MCL) has been used for those attempts involving runtime verification [14]

Discretionary Access Control (DAC) and Mandatory Access Control (MAC) were pioneers in access control technology. Role-Based Access Control (RBAC) [6] emerged in the 1990s is found to be a proven technology for managing and enforcing security in computer applications. Previously, access control policies were hard coded directly into the program by the programmer. Later on, due to complexity of the policies separate policy specification languages were developed. eXxtensible Access Control Markup Language (XACML), which is a general-purpose access control policy language standardized by the OASIS, has been broadly adopted to specify access control policies for various applications, especially web services. XACML policies are used to permit the genuine users to access the resources and also deny the sham users. Management of this XACML policy is very important task in order to avoid security seepage. Correcting access control policies manually take more time and also make inconsistencies in the policies. XACML policy Analyzer is a mechanism that facilitates systematic detection and resolution of XACML policy anomalies eliminating the need to construct Binary Decision Diagram(BDD). An efficient anomaly detection and resolution is done at the design level hence reducing the processing time of the Policy Decision Point(PDP) and this could be helpful in cases of larger XACML policies with more number of rules, policies and policy set. This also aims at security in terms of access control and to increase availability of the service [15]. While using XACML policy analyzer, PDP takes care of service invocation and at the end it handles obligations by filtering the anomaly- resolved XACML and this could also be planned well ahead during design phase. This paper is organized as follows: in Section 2, the motivation of the paper is introduced. Section 3 shows the proposed system. Results and discussions are explained in details in Section 4. Section 5, concludes the findings of the paper.

2 Motivation

The earlier service-oriented development methodologies aimed at componentizing legacy applications [[1], [16]] by implementing XML technologies such as SOAP, Web

Service Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI) on top of the existing applications or components that realize the web service is now widely practiced by the software industry [5]. This is widely found as better solution for IT business needs; though it doesn't meet the objectives of integration and interoperability of services. And by no means this is enough to build enterprise applications of the commercial strength.

The designers and developers could not be expected to supervise a complex web service development project without being provided with a healthy design and developing methodology. This makes one to have a sound view on methods and techniques used in web service design and development. Many attempts on introducing new business applications hit roadblocks at deployment, because many software developers think that security is something that needs to be incorporated at later phases of development [20]. The availability which is also an important security as well as quality factor is to be addressed along with that of security. Each business application needs to define the scope of "service security"along with availability in its own application model as well as their activities and deliverables and is hence integrated with Software Engineering (SE) activities.

The increasing use of web services has proven the advantages of service-oriented architectures and continuously applying attacks to them requires utilization of given secure mechanisms that ensure the security at different levels, not only concentrating on implementation [19]. Amazon Web Services (AWS) offers a wide range of security specific tools and features suitable for on-premises environment. In addition, this could be used for cloud environments, which could be thought of in service-oriented enterprise application development. All the above focused revolved around the security factors and the current need of handling the scalability of the Service-oriented enterprise applications. Thus basic motivation for providing the web service security is at the service enablement level of service-oriented enterprise application development where the above security requirements could be incorporated in service layer and operations systems layer, so as to address the basic security factors with availability and with a hectic need to address scalability.

3 The Proposed System

The proposed work concentrates on performing the necessary activities at various software engineering activities involved in service-oriented application development namely analysis, design and implementation. This leads to incorporating the necessary components in the service requester end and provider end that works at run-time yielding availability amplification as sketched in the following Figure 2.

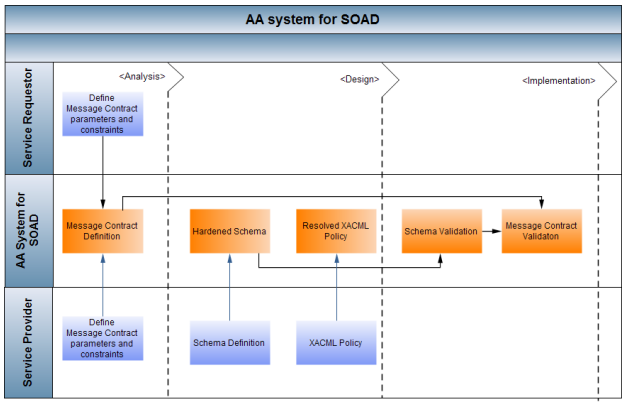

Fig. 2: Functional diagram of proposed system

At analysis level, service provider and service requester define the message contract of the services. The proposed work is initiated at this point where constraints for message contract validation are decided. There are sets of properties that are to be validated as a part of requirements in addition to that of the WSDL contents involving the data present in the SOAP requests for valid service invocations. Also clear analysis of the access control is defined by the service provider by binding with various service requesters.

At design level, the proposed work generates a restricted schema by schema hardening as well as revolving anomalies in XACML policies using XACMLWOAnalyzer which is an extension of the XACMLPolicyAnalyzer discussed in the literature. The pitfalls in the schema generated based on WSDL lead false requests to reach the web server at service provider side. Those loopholes are eliminated from the schema by the schema hardening. During runtime, the hardened schema is used for validating the service request. XACMLWOAnalyzer generates the check points that eliminate the anomalies in the policies as well as eliminates the need for pre-defined rule combining algorithms.

The proposed component XACMLWOAnalyzer as shown in Figure 3 consists of sequence of activities to construct the Boolean expression followed by the algorithms to detect as well as to correct the anomalies in the XACML Policy. The proposed component focused on XACML consists of four main steps which are done by considering presence of obligations in the policies in contrast to the XACMLPolicyAnalyzer which does the same process without considering obligations :

1.Redundancy Detection In Policies (RDIP) algorithm to find the redundancies present in the policies:
Redundant rules have the following two properties:
PROP 1:All rules are pair-wise disjointed obligations and they are taken as a parameter too (Rule id $r_i \neq$ Rule id r_j).

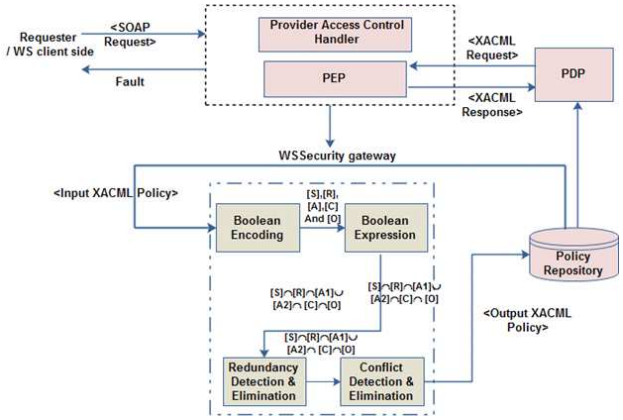

Fig. 3: XACMLWOAnalyzer Architecture

PROP 2:The effects of matched rules contain either "Permit"only or "Deny"only.

2.Redundancy Elimination In Policies (REIP) algorithm where the redundancies in the policies are eliminated.

3.Conflict Detection In Policies (CDIP) algorithm that identifies the conflicts in the policies
Conflicting rules have the following two conditions COND1 and COND2 each with four properties:

COND1:
PROP 1:All rules are pair-wise disjointed (Rule id $r_i \neq$ Rule id r_j).
PROP 2:The effects of matched rules should contain both "Permit"and "Deny"
PROP 3:If condition attribute present in the rules then
"No Condition list length is greater than zero "or
"Condition Match list length is greater than zero "or
"One Condition list length is greater than zero"
PROP 4:If obligations present in both rules r_i and r_j.

COND2:
PROP 1:All rules are pair-wise disjointed (Rule id $r_i \neq$ Rule id r_j).
PROP 2:The effects of matched rules contain either "Permit"only or "Deny"only.
PROP 3:If condition attribute present in the rules then "One Condition list length should be greater than zero"
PROP 4:If obligations present in both rules r_i and r_j.

4.Conflict Correction In Policies (CCIP) algorithm where conflicts in the policies are resolved.

These proposed components at design level concentrates on designing XACML, message contracts as well as schema to amplify the availability by providing restricted access of web services leading to reduction in number of requests reaching the service provider by eliminating in-valid requests.

At implementation level, contract validation and schema validation components are used to ensure that a valid service request is sent to the service provider. The proposed work provides component to provide a logical implication in the service invocation sequence at the web server of the service requestor using a specialized message contract language. The message contract validation is facilitated by the service log. Message contracts are validated against the message contract instance that is generated by the mapper. Message handler in turn gets the inputs from mapper component that is formulated to map the XML file in which the message contracts are designed during the design phase as shown in Figure 4.

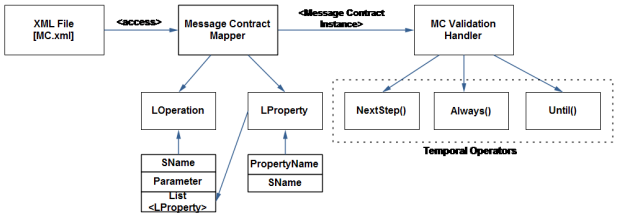

Fig. 4: Message contract mapping and validation model

Schema validation is done using streaming of service request so as to enable the early detection of schema violations in web service requests. Both the validation components are found to eliminate in-valid web services invocations that increase the processing overhead of service requestor and helps availability amplification.

As whole, at each phase of the enterprise application development the proposed work AASystemforSOAD reduces the number of false requests reaching the service provider. This increases the availability of the server at service provider that leads more number of service requests to get processed at a particular point of time.

4 Results and Discussion

The proposed research work is integration of components that does availability amplification at different levels of the service-oriented enterprise application development. The results of the components when implemented individually are presented below.

A sample policy of health care domain with 6 rules, 2 policies and 1 policy set are coined based on the sample reference provided [6]. The average time consumed for conflict detection and redundancy removal for XACMLWOAnalyzer shows an improvement than XACMLPolicyAnalyzer and XAnalyzer. This contribution is evaluated with the performance of the

system measured during runtime. The components XAnalyzer, XACMLPolicyAnalyzer and XACMLWOAnalyzer are run on placing random requests as well as running the random services amongst the services developed and deployed for health-care domain. It is evident that there is an increase in the number of successful invocations, hence availability is found to be increased. For the same set of invocations 25 runs are made and the average is computed and plotted as the graph below as shown in Figure 5. While using XACMLPolicyAnalyzer, PDP takes care of service invocation and at the end handles obligations by filtering the anomaly-resolved XACML. In case of XACMLWOAnalyzer the request is already anomaly-resolved with obligations and thus the time is reduced for handling obligations.

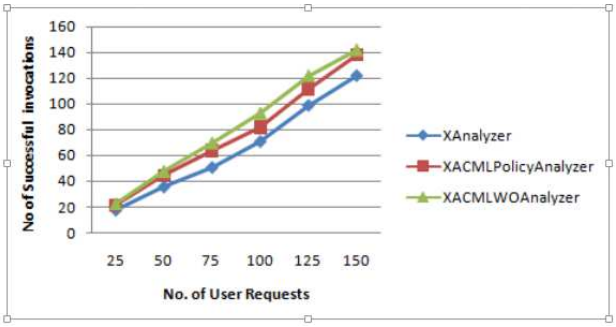

Fig. 5: Availability mapping result of access control

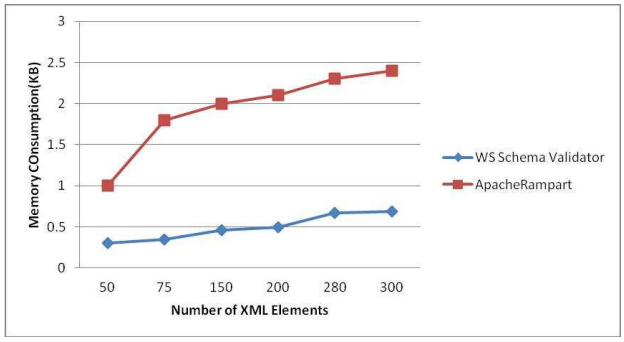

Fig. 6: Schema validation performances in terms of memory consumption

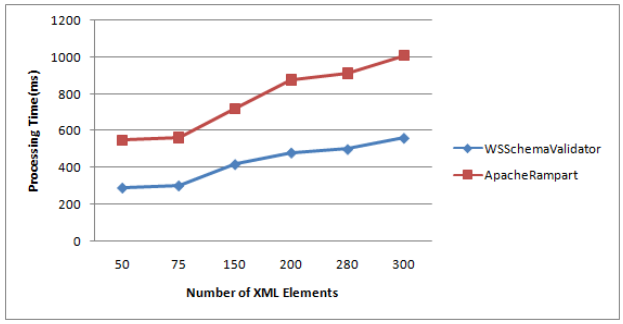

Fig. 7: Schema validation performances in terms of processing time

During the evaluation of proposed schema validation on varying the number of encrypted and signed elements the XML elements generated are 50, 75, 150, 200, 280 and 300. When the run-time or processing time in ms is evaluated for these number of XML elements, it is found to be less than that of apache rampart since process of streaming is introduced and the ordering of the signed elements is also incorporated but when numbers of elements are increased the time for processing the encryption and signature gradually increases as shown in Figure 7. For the same inputs memory consumption is also monitored and that shows a significant decrease than that of Apache Rampart due to the reason of validating on the fly requests instead of parser parsing the entire request that is modeled and verified as shown in Figure 6.

The message contract validation has also been carried out for the same set of service invocations and the results are generated which also show better availability which is final result of two factors processing time with respect to log size and memory consumption which is shown in the literature [13].

The proposed integrated work has the results which show that there is an increase in availability of the services due to the components at each level. The experimentation is done $^{1}/_{4}{}^{th}$ of the total number of possibilities for 100 requests that reach the service provider. Here we have taken four servers for experimentation, thus the experimentation is performed using 25 different invocations of the services and the results are formulated as graph using the average values. Further, the requests have been increased in the order of 25 user requests. Based on the number the servers and scalability of requests the invocations could be coined as well.

The reference system is modeled with Apache Rampart for the schema validation and uses XACMLPolicyAnalyzer that does anomaly detection, resolution and conflict eliminations without considering the obligations of the policies. This reference system is compared against the proposed work with the parameter of number of successful invocations. The proposed work AASystemforSOAD is focused on achieving availability of services by using set of components discussed above namely schema hardening, XACMLWOAnalyzer, schema validation and message contract validation at various levels of service-oriented enterprise application development. The results given above by different components are taken as input and have been compared with the resultant availability amplification achieved by

AASystemforSOAD with the help of Structural Equation Modeling (SEM) [9].

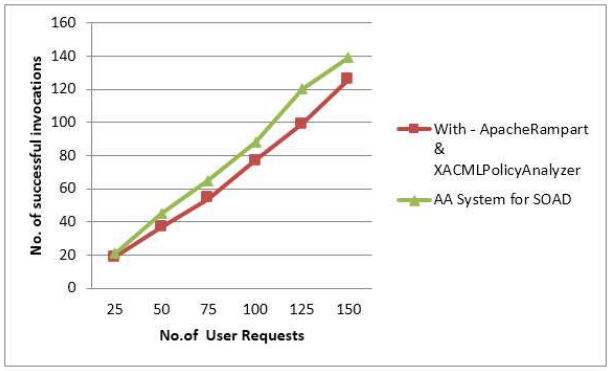

Fig. 8: Amplification of Availability

The structural model in Figure 9 below shows that a strong support for achieving the availability when using AASystemforSOAD than using Apache Rampart and XACMLPolicyAnalyzer is evident from the paths and corresponding amplification factors:

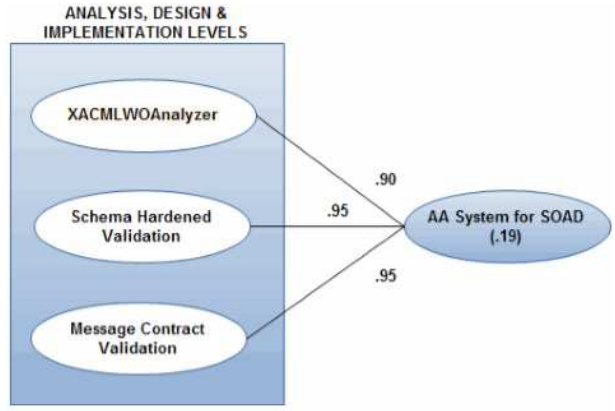

Fig. 9: Structural model for validation of AASystem for SOAD

XACMLWOAnalyzer → AASystem for SOAD, 0.90,
Schema hardened validation → AASystem for SOAD,0.95 and
Message contract validation → AASystem for SOAD (0.95) respectively.
Furthermore, the model accounts for 19% average amplification from various components proposed at analysis, design and implementation levels of service-oriented enterprise application development.

5 Perspective

Thus AASystemforSOAD is found to have increase in availability than the combinations of traditional security standard implementations. The security factors such as confidentiality, integrity, authentication and authorization could be provided using general security approaches that are available along with the security amplification which is provided by the proposed AASystemforSOAD. This system is a part of the entire framework of availability amplification, which makes evident that amplification could be customized as the phased implementation. As a further enhancement each components of the system could be improvised, componentized and integrated as well as new components could be included.

References

[1] A. Arsanjani, Service-oriented modeling and architecture, IBM Developer Works, 1-15 (2004).
[2] H. Barringer, A. Groce, K. Havelund and M. Smith, Formal analysis of log files, Journal of aerospace computing, information, and communication, vol.7, **11**, 365-390 (2010).
[3] H. Barringer, H and K. Havelund, TraceContract: A Scala DSL for Trace Analysis, Seventeenth International Symposium on Formal Methods, 57-72 (2011).
[4] K. Havelund, Monitoring with Data Automata, Sixth International Symposium on Leveraging Applications of Formal Methods, Verification and Validation, 254-273 (2014).
[5] J. Holgersson amd E. Soderstrom, Web Service Security-Vulnerabilities and Threats Within the Context of WS-security, Proceedings of the Fourth International Conference on Standardization and Innovation in Information Technology, 147-156 (2005).
[6] H. Hu, G. Ahn and K. Kulkarni, Discovery and Resolution of Anomalies in Web Access Control Policies, IEEE Transactions on Dependable and Secure Computing, vol.10, **6**, 341-354 (2013).
[7] S. Igni Sabasti Prabhu and V. Jawahar Senthil Kumar, Entropy Based Approach to Prevent the DDoS Attacks for Secured Web Services, International Review on Computers and Software, vol. 8, **4**, 888-891(2013).
[8] S. Kambhampaty, Service oriented Architecture for Enterprise Applications, John Wiley & Sons, India (2008).
[9] K. Karunasena, and H. Deng, Testing and validating a conceptual framework for Evaluating the public value of e-government using Structural Equation Modeling, Proceedings of the 21st Australasian Conference on Information Systems, 13-14 (2010).
[10] J. Namjoshi and A. Gupte, Service Oriented Architecture for Cloud Based Travel Reservation Software as a Service, Proceedings of International Conference on Cloud Computing, 147-150 (2009).
[11] E. Newcomer and G. Lomow, Understanding SOA with Web Services, Delhi, Pearson Education, India (2008).
[12] M. Priyadharshini, R. Baskaran, N. Balaji and M.S.Saleem Basha, Analysis on Countering XML-based Attacks in Web Services, International Review on Computers and Software, vol. 8, **9**,2197-2204 (2013).

[13] M. Priyadharshini, I. Suganya and N. Saravanan, A Security Gateway for Message exchange in Services by Streaming and Validation, IJIRCCE, vol. 1, **3**, 604-612 (2013).
[14] M. Priyadharshini, T. Vimala, R. Baskaran and K. Manju Bharathi, Web Service Message Contract Validation using Server Logs at Implementation level, International Journal of Applied Engineering Research, vol. 10, **13**, 33336-33340 (2015).
[15] M. Priyadharshini, J. Yowan, R. Baskaran, Security Enhancement in Web Services by Detecting and Correcting Anomalies in XACML Policies at Design Level. In: J.L.Mauri, S.M. Thampi, D.B. Rawat, D. Jin (eds) Security in Computing and Communications. SSCC 2014. Communications in Computer and Information Science, vol 467. Springer, Berlin, Heidelberg (2014).
[16] E. Ramollari, D. Dranidis and A.J. Simons, A survey of service oriented development methodologies, Proceedings of 2nd European Young Researchers Workshop on Service Oriented Computing, **75** (2007).
[17] M. Stal, Using architectural patterns and blueprints for service-oriented architecture, In IEEE Software, vol. 23, **2**, 54-61 (2006).
[18] S. Suriadi, D. Stebila, A. Clark and H. Liu, Defending Web Services against Denial of Service Attacks Using Client Puzzles, Proceedings of International Conference on Web Services (ICWS), 25-32 (2011).
[19] M. Ivanova, Security of Web Services: Methods and Contrivance, International Journal of Computers and Technology, vol. 14, **11**, 6229-6239 (2015).
[20] H. Hinton, M. Hondo and B. Hutchison, Security Patterns within a Service Oriented Architecture, IBM whitepaper (2005).

M. Priyadharshini, received her Ph.D. degree from Anna University, Chennai, M.E (Software Engineering) and B.E (Computer Science and Engineering) from Sri Ramakrishna Engineering College, Coimbatore. She is having around 19 years of work experience in training corporate employees, imparting the student's wide knowledge as well as proficient in handling assignments / IT projects and has insight in concept development for enterprises. Her research area includes service oriented architecture, object persistence, computer-aided automation systems, web and enterprise technologies. She holds few publications in her research area which includes CCIS of Springer. She has filed around 8 patents related to additive manufacturing and 3D printing. She has contributed to the University Grants Commission (UGC) Country Wide Class Room (CWCR) program at Educational Multimedia Research Centre (EMMRC), Anna University, Chennai. She has been training employees of corporate clients including HP, CSS, CSC, Aspire Systems, Syntel, L&T etc., Also she is expert in programming, database technologies and advanced frameworks including J2SE, J2EE, MySQL, SQLServer, Oracle, Hibernate, Struts etc., She has been resource person for many workshops, faculty development programmes, seminars.

R. Baskaran received his B.Tech in Electrical and Electronics Engineering, Master Degree in Computer Science and Engineering, Madras University and Doctorate from Anna University in the years 2000, 2001, and 2007 respectively. He is now working as Professor in Department of Computer Science and Engineering, College of Engineering, Guindy, Anna University Chennai. His present research includes Database, Data mining and warehousing and Image Retrieval. He presented more than 100+ Special Lectures in National, International Seminars, Workshops and Development Programs. He is an expert in Data mining. He has published 200 papers in International, National Journals and Conferences. He is a reviewer in IEEE/ ACM JSAC, Elsevier and many international journals. He is a life member of Institution of Electronics and Telecommunication Engineers (IETE), Indian Society for Technical Education (ISTE), and International Association for Engineers (IAEng).