Non Functional Requirement Classification for Service-Oriented Data Warehousing

***Anjana Gosain, **Sangeeta Sabharwal, ***Rolly Gupta**

*University School of Information Technology, GGS Indraprastha University, Delhi, India
**Netaji Subhas Institute of Technology, Delhi University, Delhi, India
***Netaji Subhas Institute of Technology, Delhi University, Delhi, India
*anjana_gosain@hotmail.com, **ssab@nsit.ac.in, ***guptarolly02@yahoo.com

Abstract **-** Recently, Data Warehouse has proven to be a powerful technology for the integration of heterogeneous data into a multidimensional repository for decision-support analysis. The complex ETL (extraction, transformation and loading) process and the aggregation-intensive queries are affected by a sequence of domain specific NFRs (Non Functional Requirement). This advocates the use of service oriented NFR approach for building a data warehouse specification.

However, the specification and classification of service oriented systems and service oriented data warehousing (DW) systems have not been addressed to the appropriate level as attempted for non service-oriented systems. In this paper, we propose a new framework for the classification of NFR (Non-functional requirements) with respect to engineering service oriented and service oriented data warehousing systems. In addition, this proposed classification is supposed to be of significance in terms of NFRs classification for service engineering and service-oriented DW engineering.

Index Terms— Non Function Requirement specification, SOC, QoS, NFR, Data Warehousing

I. INTRODUCTION

In recent years Data Warehouse (DW) has proven to be a powerful technology for integrating distributed operational data into a comprehensive repository for predicting future decisions. Therefore, the design of these systems is rather different from the design of the conventional operational systems that is involved with the supply of data to the warehouse systems. The former involves information requirements of decision makers, structure and allocated requirements of the latter systems also. Software engineers need to deal with the complex process of ETL (extracting, transforming, and loading) data while managing the deployment of solution; that can precisely and timely integrate a number of heterogeneous source systems. On the basis of deployment, they present analytical results in an accurate and reliable form; thus offering flexibility at the front-end with the support of a complete, non-redundant dimensional model. Thus, both operational and strategically visionary approach have to be wrapped up in a multidimensional package in order to meet corporative analytical requirements pervading pure decision-support functionality as well as quality constraints like integrity, accessibility, performance, etc [23]. This advocates the use of DW Requirements Engineering techniques for building a data warehouse specification precisely.

To pursue the goal, we propose a methodological approach for the requirements analysis of data warehouse systems. The approach provides an service-oriented method for the guidance of requirements engineers during the data warehouse specification process. The non-functional requirement acts as a constraint on software product, or on software development process and externally related constraints. However, much knowledge is required regarding the requirements specifications for service oriented systems i.e. from functional view to non-functional view. The proposed approach is general with regard to non-functional requirements and paves the paths in order to understand the positive and negative aspects of a certain requirement in regard to the process of data warehouse.

The NFR Framework [6] helps to produce a solution that is able to embrace the quality characteristics of a particular domain. So, we aim to devise a new classification of non-functional requirements for service oriented systems (service engineering) and service-oriented DW systems (service-oriented engineering). We have designed a hierarchical tree of non-functional requirements in respect to the data warehouse engineering. Such a classification helps in identifying and guiding the NFR specifications for service-oriented systems. In the paper, a review of related work concerning service-oriented engineering and NFRs classification is presented in section 2. In Section 3, we introduce the new framework of NFR classification. Lastly, we conclude our discussion regarding the proposed classification and future work directions.

II. RELATED WORK

A. Service Oriented Computing

Service-Oriented Computing (SOC) is a new outcome for the development of services that can be reusable and loosely coupled for creating flexible applications. Erl [10] defined SOC as a platform for new generation computing that encompasses service-oriented paradigm and service-oriented architecture to create and assemble one or more services. Both language and platform independent services can be addressed. Papazoglou et al [18] considered SOC as a logical separation of functionality in three main architectural layers namely, service foundation, service composition and service management and monitoring (semantics, QoS and properties

of NFR services). They also addressed the Service-Oriented Architecture (SOA) as a logical approach to realize SOC. Sommerville [19] considers SOA as a new technology for the development of distributed applications as shown in Figure 1.

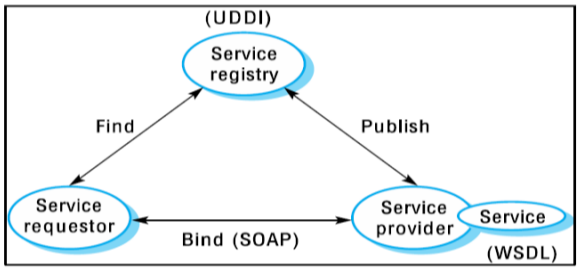

Figure 1: Service-Oriented Architecture [19].

Web services serve as a promising technology for implementation of SOA. It uses Internet as the medium for distributed computing and offers interoperability. Web services are implemented by means of open standards such as WSDL (Web Service Description Language), UDDI (Universal, Description, Discovery and Integration). It allows service requestors to browse and find the required service, and SOAP (Simple Object Access Protocol SOAP) as shown in Figure 1[19]. Sommerville [19], describes the design service interfaces considering service operations. However, it lacked the description of NFRs and service semantics.

Furthermore [22], entails business process driven software development. The process relies on business process understanding and analysis by taking into account functional requirements and NFRs. Sometimes, composition of services may also be required for complex business processes.

B. Non-Functional Requirement Classification

In recent days, customers not only concentrate about the embedded functionality in DW services but they also pay attention to the desired quality of services, such as reliability, security, and efficiency. Such requirements are categorized under NFRs.

In the attempt to develop a framework for software quality characteristics, Boehm [4] proposes some form of NFRs classification as depicted in Figure 2. Boehm also proposed quality evaluation metrics in order to assess adherence to NFRs. For example, low-level quality attributes are logically implied as the guarding conditions for such type of adherence. Using this tree, three questions are deduced: namely,(1) How it is possible that the developed software can change its environment?,(2) how adhering this software is with respect to quality requirements such as reliability, efficiency and usability and (3) Extent to which such software can be maintained? In spite of such worthy classification provided by Boehm, but the framework is still not complete and lacks more reflective quantification techniques. Such type of technique can be a service oriented paradigm. However, the classification by Boehm can be considered as a landmark for further refinements. Davis [7] introduced NFRs as non-behavioural requirements and identified seven qualities as a further refinement in the classification of Boehm.

Chung et al [6] proposed a process and goal based NFR-framework, in which goals are decomposed and refined to sub goals for arriving at low-level operationalisations soft goals by the use of the SIG (Soft Goal Interdependency Graph). The low level operationalisations denote some additional functional requirements for satisfying the high level NFRs through the "Label Propagation Algorithm". NFR can have a varied range such as "satisfied", "weakly denied" etc. This illustrates tradeoffs between the soft goals. Thus, this framework may be considered as a comprehensive approach for the determination of the satisfaction and tradeoffs of between low-level NFRs. Burgess et al [5] proposed the concept for SIG to develop Soft goal Interdependency Rule set Graph (SIRG). SIRG will act as an automated technique for determination of optimal set of the low-level operationalisations to attain better NFRs' satisfaction. However, the SIRG requires some comprehensive case studies.

The IEEE Standard [14] is a notable example to classify and specify NFRs. Gilb [12] classified the requirements to functions, qualities, costs and constraints [12]. Except the first, the last three can be NFRs. Along this context, qualities can denote "function performance" and constraint relate to "restrictions on the freedom of requirements or can be design". Gilb's classification emerged out of presence of unwanted or undesirable requirements as specified in SRS (Software Requirement Specification) documentation. Such type of requirements may be false, unclear, and/or not possible to assess satisfaction level.

Sommerville and Kotonya [15] introduced NFRs as a medium of restrictions and constraints among system services. However, this classification may be considered most comprehensive in terms of NFRs types. They classified NFR under three key categories i.e. product, process, and externally related NFRs. The product category relates to the possible or desired attributes that may be possessed by a system. The second category relates to constraints and restrictions laid on the development process over the system. Finally, the third category relates to externally related NFRs
related to organisational regulations, national or international standards, or can be interoperability requirements.

Glinz [13] further classified NFRs as performance and quality related requirements. They described NFR using four facets: *kind*,
representation, *satisfaction*, and *role*. Lamsweerde [21] later classified NFRs according to QoS (Quality of Service), compliance, architectural constraints and development constraints as in Figure 3.

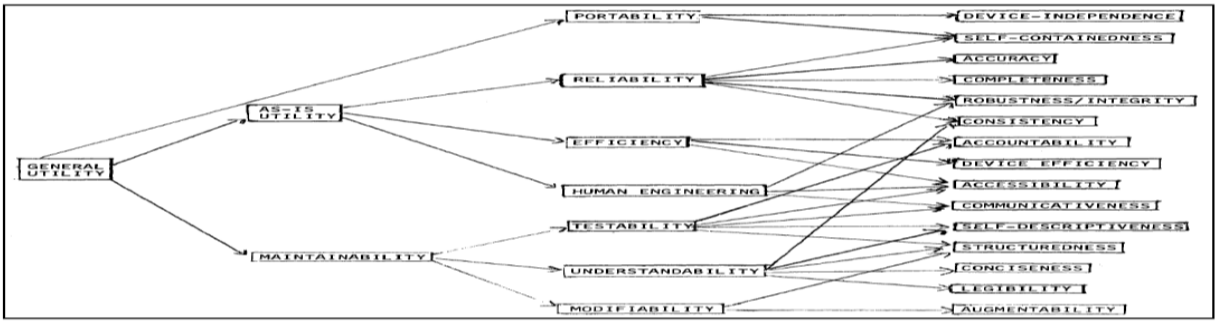

Figure 2:Boehm's Characteristics Tree [4].

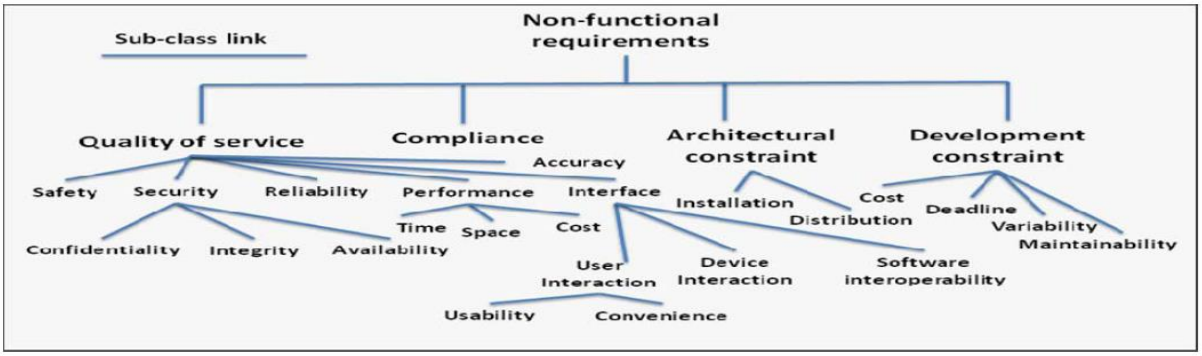

Figure 3: Lamsweerde Non-Functional Requirements on Software Services [21].

QoS relates to quality requirements such as security, performance, etc. *Compliance* relates to confirmation to standards, organisational regulations and external standards and laws. Architectural constraints implies structural constraints in relation to the developed software along with its operational constraints. D*evelopment constraints considers the* governance in developing the anticipated software, such as maintainability, schedules, etc.

C. Service- Oriented NFR

In order to relate NFRs to service oriented development, it is concerned with identification of QoS attributes, tradeoff between them, and how they affect the development of service-oriented applications. However, no classifications attempts to have been reported in the literature to specialize or arrive at a new classification of NFRs for service oriented DW engineering. Galster and Bucherer [11] proposed NFRs classification regarding services and service oriented systems.

O'Brien et al [16] introduced different quality attributes affecting SOA and then discussed ten attributes related issues along with recommended solutions in order to satisfy quality characteristics. They also discussed tradeoffs between quality attributes, for example: developers may be required to be aware of the negative effect of security on performance and interoperability, and the harm that can be caused due to the absence of some quality attributes such as availability. They observed the limitation of current standards [16] in addressing quality attributes to attain the appropriate NFR specification.

As a number of standards have been developed for QoS (WS-Reliable). However, a Service Level Agreement (SLA) is needed to be established between the parties in order to attain an agreement on the desired QoS. But no classification provides an explicit classification of NFRs in relation to service oriented DW engineering.

Later, Ameller and Franch [2] proposed a Service Level Agreement Monitor (SALMon). It acts as a monitoring technique on SLA. This technique is used to monitor services in order to carry out the appropriate decisions when unexpected actions occurs while adopting SOA in order to meet SLA requirements. An ISO/IEC 9126 based classification of the desired characteristics in relation to web services as presented in Figure 4. It emphasis on the technical and non-technical characteristics of NFR services. This quality model was chosen because of its generic features and ability to develop characteristic hierarchies. In this quality model, characteristics chosen are refined and decomposed to sub levels.

Galster and Bucherer [11], presented a taxonomy for NFRs along with service-centric systems. They classified NFR into three categories, namely: process requirements, service requirements and external requirements. This proposal denoted the ability to relate the taxonomy to the service level

as well as the system level. Dobson and Sommerville [9] demonstrated an attempt of developing QoS ontology, namely the QoSOnt for service-centric systems, but disregards domain knowledge.

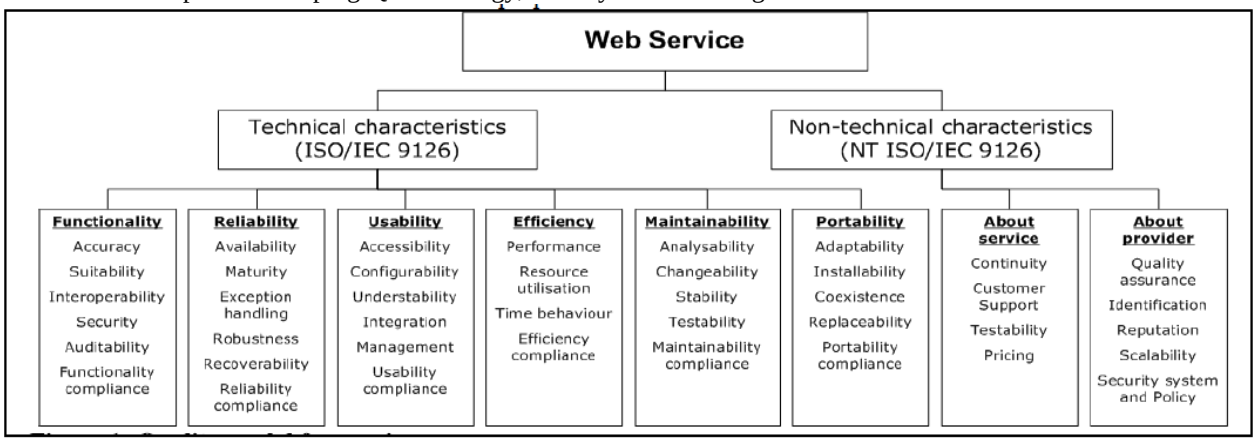

Figure 4: Quality Model [2].

III. PROPOSED NFR FOR SERVICE ENGINEERING AND SERVICE ORIENTED DATA WAREHOUSE ENGINEERING

This proposed classification shown in Figure 5 is designed on the basis of NFR classification given by Kotonya and Sommerville and Van Lamsweerde. In the proposed classification, we addressed service engineering and DW-oriented service engineering as high-level categories leading to further sub classifications.

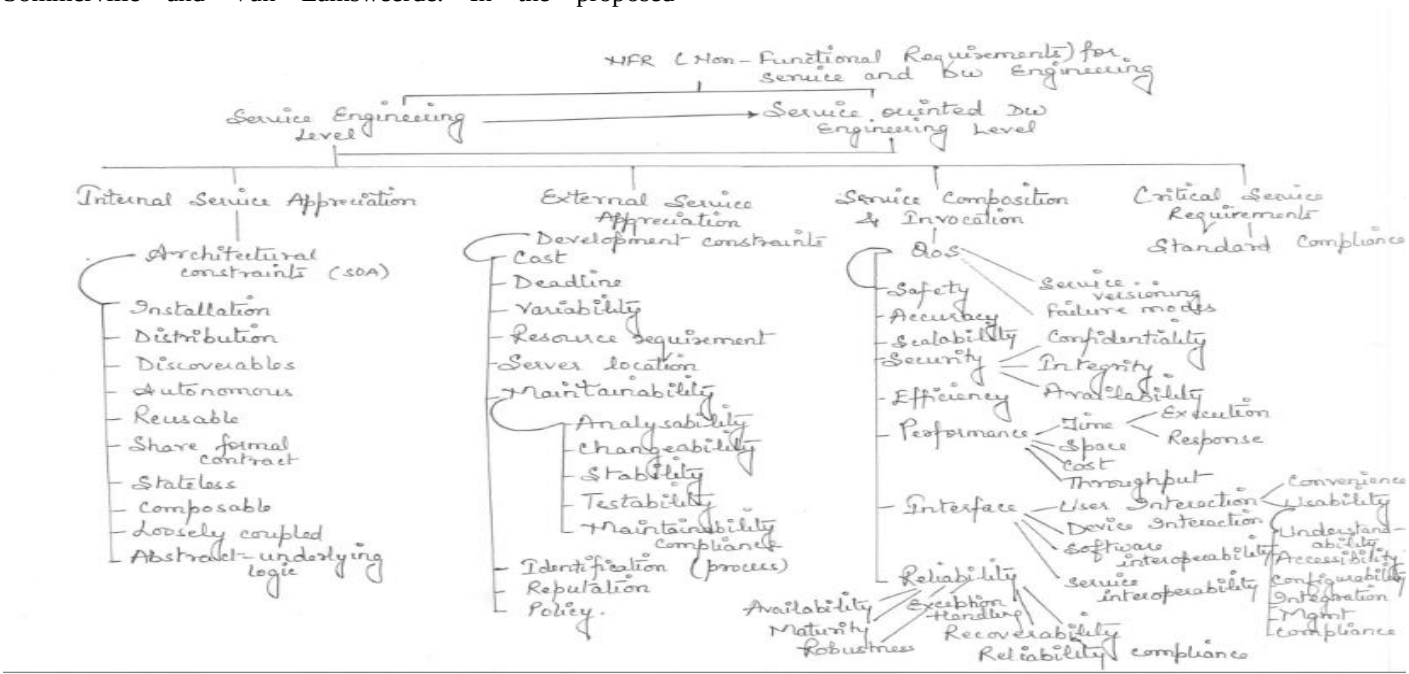

Figure 5: A Non-Functional Requirements Multilevel Taxonomy on Service-Oriented DW Engineering.

A. NFR for Service Engineering

The constraints are based on the desired QoS, compliance to requirements, development standards, and architectural constraints. However, this leads to a service which is reusable, loosely coupled with independent operational requirements. But, it adhers to related organizational, national/international standards and regulations. The Service composition and invocation applicable to ***QoS*** requirements concerns for desired characteristics of a service that is running independently. It also integrates other services at run-time. The External service appreciation of ***Development requirements*** are related to service engineering process stages ; starting with the service identification stage. It starts from

designing, servicing, implementation and deployment stages. ***Compliance requirements*** belonging to Critical service requirement implies for the conformation related to standards, laws or rules, organizational regulations, etc . Internal service appreciation related to ***Architectural constraints***, are mainly concerned with structural constraints or to service design. They adhere to Erl's [10] SOA principles namely, abstract underlying logic, loosely coupled, reusable, autonomous, services share standardised contracts, stateless, discoverable and composable .

B. NFR for Service Oriented DW Engineering

The constraints are based on the QoS attributes, standards's compliance, the development process, and architectural constraints. The benefits are provided by the Data warehouse through the use of *reusable services. S*ervice engineering NFRs are applicable to single service, while service-oriented DW engineering NFRs considers the overall behaviour of the application developed from reusable services.

The QoS attributes, and standards and regulation compliance regulate the overall behavior of the DW-oriented application. DW engineering process influences the NFR of development and architectural constraints. If the construction of service oriented application is done by the composition of service process, then the development process NFR concerns for requirements of workflow development, service discovering, service selection, modelled workflow refinement , refined workflow enactment and testing of services and the overall application.

IV. CONCLUSION

The proposed classification in this paper has addressed the non-functional requirements for both service oriented engineering and service oriented DW-oriented engineering. The latter is dependent on the former in terms of classification of NFRs. This newly proposed classification is mainly based on Sommerville and Lamsweerde and service oriented development. The proposed classification is the basis for a shared understanding of NFRs in both of the areas i.e. service engineering and DW-oriented engineering. The NFR classification facilitates the development of the semantics behind the identified set of NFRs in every category and subcategories of the hierarchy.

Future work can be done on completing the specification of the classification. So, that it could serve as the general ontology for engineering of service oriented DW applications. Therefore, this proposed classification paves the path for advancing the state of the art in the process of quantification, for particular application domains.

REFERENCES

[1] Aburub F., Odeh M. and Beeson I., "Modeling Non-Functional Requirements for Business Process", Information and Software Technology Journal, vol. 49, no. 11-12, pp. 1162-1171, 2007.

[2] Ameller D. and Franch X., "Service Level Agreement Monitor (SALMon)", in Proceedings the 7th IEEE International Conference on Composition- Based Software Systems (ICCBBSS 2008), Madrid, Spain, pp. 224-227, 2008.

[3] Baader F., Horrocks I. and Sattler U., "Description Logics for the Semantic Web", Kunstliche intellihenz, vol. 16, no. 4, pp. 57-59, 2002.

[4] Boehm B., Brown J. and Lipow M., "Quantitative Evaluation of Software Quality", in Proceedings of 2nd IEEE International Conference on Software Engineering, California, USA, pp. 592-605, 1976.

[5] Burgess C., Krishna A. and Jiang L., "Towards Optimising Non-Functional Requirements", in Proceedings of the 9th International Conference on Quality Software, Jeju, South Korea, pp. 269-277, 2009.

[6] Chung L., Nixon B. and Mylopoulos J., Non- Functional Requirements in Software Engineering, Kluwer Academic Publishers, London, 2000.

[7] Davis A., Software Requirements: Objects, Functions and States, Prentice Hall, Upper Saddle River, 1992.

[8] Dobson G., Hall S., and Kotonya G., "A Domain Independent Ontology for Non-Functional Requirments", in Proceedings of the. IEEE International Conference on e-Busniess Engineering (ICEBE 07), Hong Kong, China, pp. 563-566, 2007.

[9] Dobson G., Lock R. and Sommerville I., "QoSOnt : A QoS Ontology for Service-Centric Systems", in Proceedings of the 31st EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 05), Porto, Portugal, pp. 80-87, 2005.

[10] Erl T., SOA: Principles of Service Design. Prentice Hall, Upper Saddle River, 2007.

[11] Galster M. and Bucherer E., "A Taxonomy for Identifying and Specifying Non-functional Requirements in Service-Oriented Development", in Proceedings of IEEE Congress on Services Part I, Honolulu, USA, pp. 345-352, 2008.

[12] Gilb T., "Towards the Engineering of Requirements", Requirements Engineering Journal, vol. 2, no. 3, pp. 165-169, 1997.

[13] Glinz M., "Rethinking the Notion of Non- Functional Requirements", in Proceedings of the 3rd World Congress for Software Quality, Munich, Germany, pp. 55-64, 2005.

[14] IEEE, IEEE Recommended Practice for Software Requirements Specification. IEEE standard 830-1993, 1993.

[15] Kotonya G. and Sommerville I., Requirements Engineering: Processes and Techniques, Jhon Wiley, Chichester, 1998.

[16] O'brien L., Merson P. and Bass L., "Quality Attribute for Service-Oriented Architectures", International Workshop on Systems Development in SOA Environment (SDSOA), Minneapolis, USA, 2007.

[17] OMG (2011) Business Process Model and Notation (BPMN) [online]. Available from: http://www.omg.org/spec/BPMN/2.0/PDF/. [Accessed 28 July 2011 at 3:36 am].

[18] Papazoglou M., Traverso P., Dustdar S., and Leymann F., "Service-Oriented Computing: State of the Art and Research Chanllenges", IEEE Computer, vol. 40, no. 11, pp. 38-45, 2007.

[19] Sommerville I., Software Engineering, Eight Edition, Addison-Wesley, Harlow, 2007.

[20] Valipour M., Amirzafari B., Maleki K. and Daneshpour, N., "A Brief Survey of Software Architecture Concepts and Service Oriented Architecture", in Proceeding the 2nd IEEE International Conference on Computer Science and Information Technology 2009 (ICCSIT 2009), Chengdu, China, pp.34-38, 2009.

[21] Van Lamsweerde A., Requirements Engineering: From System Goals to UML Models to software specifications, Jhon Wiley, Chichester, 2009.

[22] Yousef R., Odeh M., Coward D., and Sharieh A., "BPAOntoSOA: A Generic Framework to Derive Software Service-Oriented Models from Business Process Architectures", in Proceeding of the 2nd International Conference of Applications of Digital Information and Web Technologies (ICADIWT '09), London, England, pp. 50-55, 2009.

[23] Abelló, A., Samos, J., Saltor, F. "Benefits of an Object Oriented Multidimensional Data Model". Lecture Notes in Computer Science, a. 1944, pg. 141 ff, Proc. of Objects and Database 2000 (ECOOP Workshop), France, 2000.

AUTHORS

First Author – Anjana Gosain, Ph.D, University School of Information Technology, GGS Indraprastha University, Delhi, India. Email: anjana_gosain@hotmail.com

Second Author – Sangeeta Sabharwal, Ph.D., Netaji Subhas Institute of Technology, Delhi University, Delhi, India. Email: ssab@nsit.ac.in

Third Author – Rolly Gupta, M.Tech , Netaji Subhas Institute of Technology, Delhi University, Delhi, India. Email: guptarolly02@yahoo.com

Correspondence Author – Rolly Gupta, Email: guptarolly02@yahoo.com, rollygupta02@gmail.com, Contact: 09899148597.