A Framework for Performance Estimation of Object-Oriented Databases

Dr. Vipin Saxena
Department of Computer Science
B.B. Ambedkar University (A Central University)
Rae Barely Road, Lucknow-25, U.P. (India)

Ajay Pratap
Department of Computer Science
B.B. Ambedkar University (A Central University)
Rae Barely Road, Lucknow-25, U.P. (India)

***Abstract*— The architecture of any database management system is an complex task for the analysis and also for the performance estimation purpose. Various database systems based on varying data model and implementation are present in the market but choosing the best one for performance analysis is an important task. This paper deals with the database estimation methodology which integrates the database analysis task and performance analysis task. Here, we have used analytical modeling technique for the performance estimation, which is a quantitative method of analysis. A framework for the performance estimation of object-oriented database system is also presented in this paper.**

Keywords- OODBMS, Performance, Analytical Modeling, Framework,Bbeanchmarking, Software Performance Engineering (SPE).

I. INTRODUCTION

Database systems have complex architecture but they are the key factors behind the business transformations. Due to the increased popularity of database systems many new database systems based on varying data model and implementation have entered in the market. Choosing the best one in any category is an important task based on performance analysis. There are three major techniques for the performance estimation which are analytical modeling, simulation modeling and benchmarking.

Analytical modeling is a quantitative method to analyze the performance factors. Further, it has two approaches which are queuing model based approach and cost based approach. Simulation model is the second approach for this purpose and it does not estimate the performance of database under consideration in isolation but with the application program running on the top of database. Benchmarking is another method for the performance analysis of multiple databases, which is costly and time taking but provides valid performance results. This method is useful for the comparison of database systems.

II. RELATED WORK

The research work carried out in the field of object-oriented database system and its performance issues has been analyzed. The research papers of various authors in this field are discussed in this section. Beeri [1] has emphasized on efforts to develop a formal framework that contains most features found in current object oriented database systems. The presented framework includes structural object model and deals with higher-order concepts, such as classes and functions as data, methods, and inheritance. Bertino and Martino [2] have presented a clear introduction to the concepts and features of object-oriented database. Several examples of current commercial systems have been used for the illustration of system. A performance analysis algorithm has been proposed by Thakore and Stanley [3], which is applied on parallel object-oriented databases because it always been a critical issue in object-oriented database system. Tiwary et al. [4] have described the experiences with using OO7 both as an application benchmark and as a system benchmark. A framework has been proposed which is designed for both application and system benchmarks and evaluated OO7 using this framework. For the construction of many large applications that manipulate complex data structures has motivated the development of object oriented databases and persistent object stores. Biliris and Panagos [5] have focused on minimizing memory latencies, such as cache-conscious algorithms for sorting and data placement. They have examined four commercial DBMSs and introduced a framework to analyze the query execution time on a DBMS which is running on a server with a modern processor and memory architecture. A memory resident database has been used to focus on processor and memory interactions and exclude the effects from the I/O subsystem.

Chaudhri et al. [6] have discussed the suitability of ODBMS implementations for other application domains and evaluated two pure ODBMSs and one hybrid Object-Relational DBMS to determine the suitability. The results show significant performance differences between the products under test and the hybrid Object-Relational DBMS has consistently shown poor results on most of our tests. The benchmarks have been developed in order to analyze the performance of object-oriented database systems. Amer [7] has presented a layered queuing network based performance model which is transformed from the UML software specification. The transformation is carried out with the help of graph grammar technique. Gorla [8] has developed a

methodology for the design of an efficient storage structure of object-oriented databases that minimized the cost of database operations. In [9] object management group has discussed the fundamentals about the UML profile for schedulability, performance and time Specification. Hillston and Wang [10] have presented a comprehensive review of recent research in the field of model-based performance prediction at software development time. Traditional software development methods focuses on software correctness and then performance issues are introduced in the development process. The model-based performance has been proposed in order to assess the maturity of the field and point out promising research directions.

Marcos et al. [11] have presented a framework for an Object-Relational Database Design Methodology. The proposed framework has defined new UML stereotypes for Object-Relational Database Design and proposed guidelines to translate a UML conceptual schema into an object-relational schema. Petriu and Woodside [12] have defined the performance model of software design which can compute the estimation of delays, throu has closed the field of software engineering and performance analysis. Marzolla [13] has presented a quantitative analysis approach for the performance analysis of system. The performance problems have been identified quantitatively and then phases of the software development cycle are implemented. Alsaadi [14] has described an approach to software performance analysis based on the UML class diagram. This approach has predicted the performance values for use cases operating on persistent data collections and the extent of normalization or demoralization a UML class diagram may need to meet out the performance objectives.

Balsano and Marzolla [15] have also proposed an approach based on queuing networks models for performance prediction of software systems at the software architecture level. UML use case, activity and deployment diagrams helped to derive a performance models based on multi chain and multi class queuing networks. The proposed approach can be fully automated and uses standard UML notations and can be integrated with other performance modeling approaches. They have discussed the integration of QN-based approach with an existing simulation-based performance modeling tool. Alam and Wasan [16] have explained that object oriented database systems are not only suited for certain specific problems but it represents real world very well. They have shown that how the data is more secure in object-oriented database than in relational database and also why do we migrate from RDBMS into OODBMS.

Umoh et al. [17] adopted a state-of-the-art technology to design an Object-Oriented Database Management System (OODBMS) for the management of information in Nigerian Universities. A working prototype of three-tier client server architecture is developed and the client-server and database technologies have been applied to provide an object-oriented web-based data retrieval system. In [18], Saxena and pratap has discussed the problem of impedance mismatch and after analyzing the features of object-oriented database its use is suggested. They demonstrated the development of web application for Indian Postal Services by using Db4o and dot NET platform. UML class, sequence and use case diagrams are designed to develop the web based system.

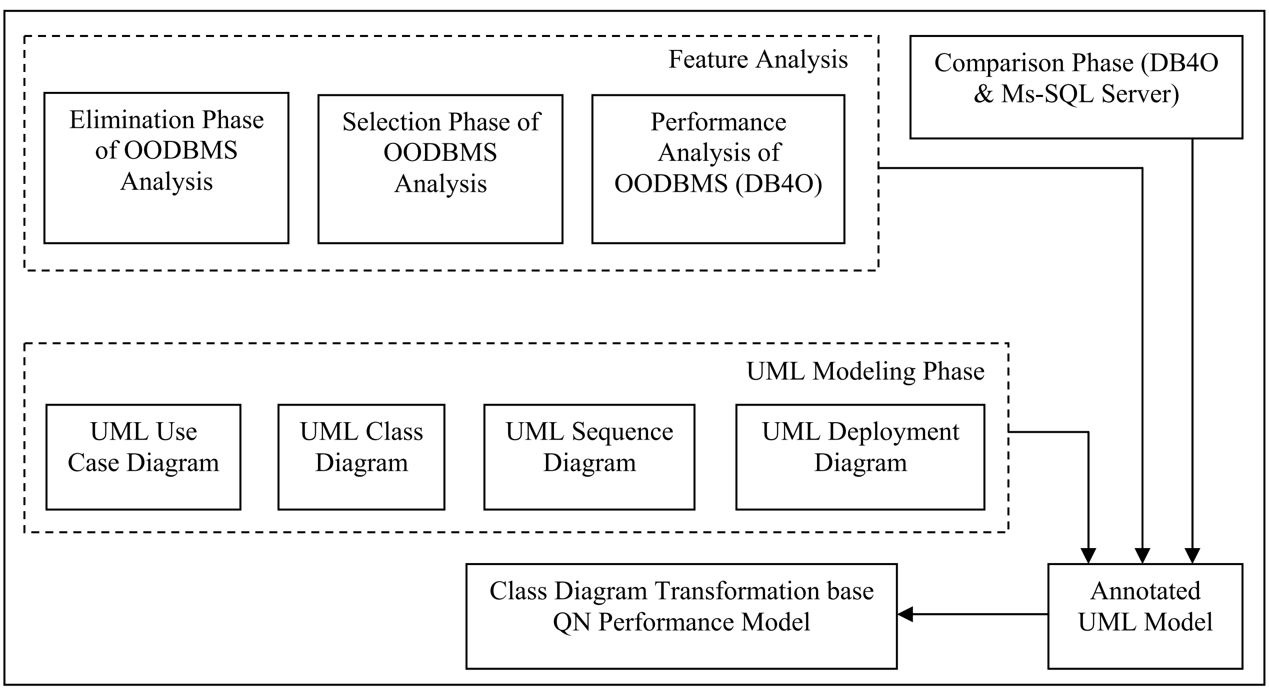

Figure 1: Framework for Performance Estimation of OODBMS

III. FRAMEWORK FOR PERFORMANCE ANALYSIS

A three step approach is adopted for the performance estimation of database system, which includes the following task:

- Analyze the object-oriented database system
- Compare OODBMS and RDBMS
- Proposing UML based performance model

Many commercial and open source OODBMS are available in the market such as O2, GemStone, ONTOS, ObjectStore, Ozone, Objectivity/DB and Db4o. Few of them are selected which may be suitable for our research work. After the selection of appropriate object-oriented database system for the present case, we have to choose the benchmark and establish the experiment environment. Then various performance metrics are calculated. Framework for the performance estimation of object-oriented database is shown in figure 1.

IV. PERFORMANCE ESTIMATION TECHNIQUE

We have three techniques for the performance estimation, which are analytical modeling, simulation modeling and benchmarking. Analytical modeling and simulation can also be used for performance estimation but here we are using benchmarking because it can be applied on complete database system functionality. Benchmarking is important and useful technique for the database estimation. Works of different researchers are available in this and different benchmarks have been proposed by them. Various benchmarks such as Wisconsin, AS^3AP, TPC-C, TPC-D, and TPC-E were proposed for the performance analysis of relational databases and many standard benchmarks such as HyperModel, OO1, ACOB, and OO7 benchmarks are available for the performance analysis of object-oriented database systems. Development of performance metrics for the database system is very difficult task and it is a three step process which has following steps:

- Design of Benchmark
- Execution of Benchmark
- Analysis of Benchmark

Benchmark designing includes the establishment of system environment, which involves designing of system configuration, checking the workload and test data. In the second step, the benchmark is performed and performance data is collected. The performance results are analyzed and compared in the third step. Figure 2 shows the methodology for the database system benchmark.

.

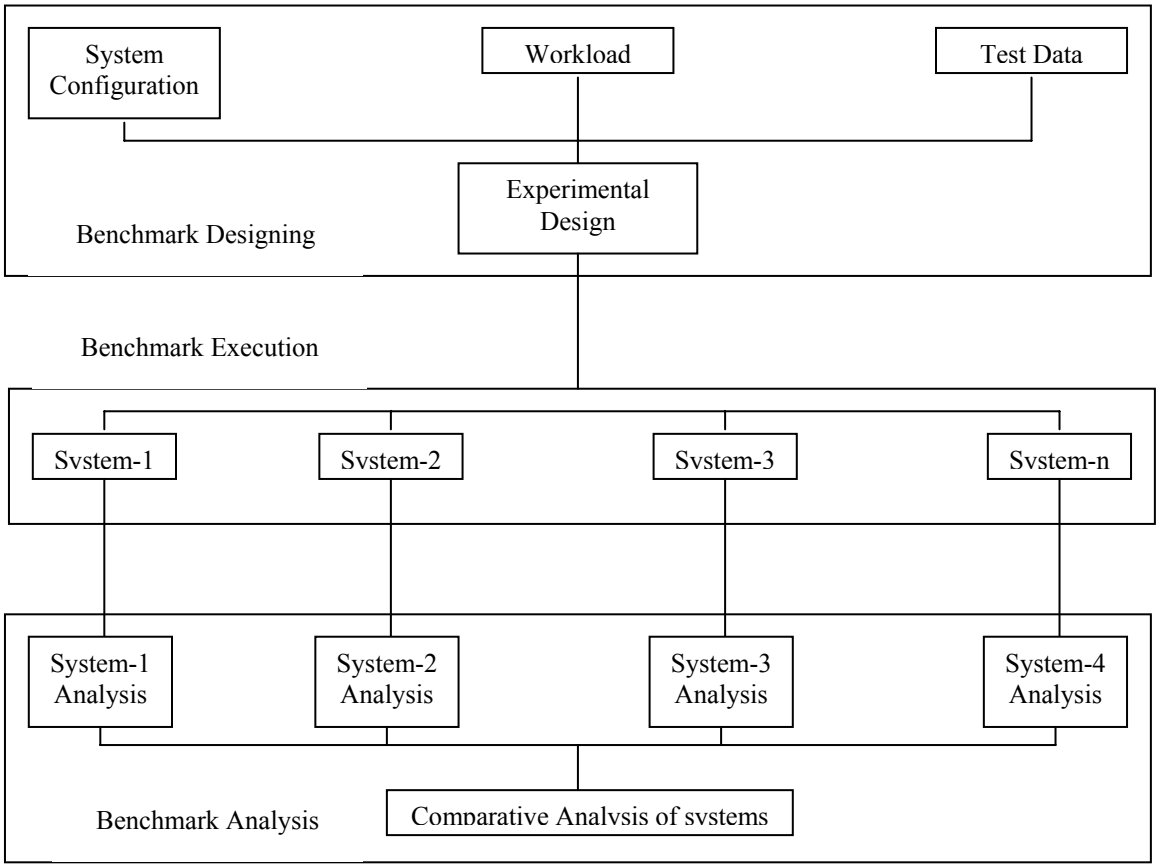

Figure 2: Methodology of Database System Benchmark

HyperModel is an early OODBMS benchmark method which was used to test hypertext and hyperlink application. OO1 is another benchmark defines the common workload characteristic of computer-aided design (CAD) applications and computer-aided software engineering (CASE). This benchmark is based upon part objects and their connections, which is capable to measure the performance of RDBMS, network database systems, hierarchic database systems and OODBMS but the database size for performance estimation can range from 4 MB (Small), 40 MB (Large) and 400 MB (Huge). After OO1, Altair Complex Object Benchmark (ACOB) came in existence and measures the performance of client-server OODBMS architectures. The OO7 Benchmark was published in year 1993 and it is a single-user benchmarking technique, which attempts to provide a true comprehensive test for single-user OODBMS performance. The OO1 benchmark measures the simple navigational and update tasks and covers both OODBMS and RDBMS but it does not support complex objects. The OO7 benchmark is ranked high as it is more comprehensive and complete benchmark method as compare to others benchmarks such as HyperModel, OO1 and ACOB.

Choosing the best benchmark for the performance estimation of object-oriented database system depends on four factors which are portability, simplicity, scalability and relevance. TABLE I represents the comparative report of HyperModel, OO1 and OO7 benchmarking techniques.

TABLE I. BENCHMARKS COMPARISON TABLE

Features	Relevance	Portability	Simplicity	Scalability
HyperModel	High	High	Low	Least
OO1	Least	Highest	Highest	Low
OO7	Highest	High	Least	Low

TABLE II. PARAMETERS OF THE OO7

Parameter	Small	Medium
NumAtomicPerComp	20	200
NumConnPerAtomic	3/6/9	3/6/9
DocumentSize (bytes)	2000	20000
Manual Size (bytes)	100K	1M
NumCompPerModule	500	500
NumAssmPerAssm	3	3
NumAssmLevels	7	7
NumCompPerAssm	3	3
NumPrivateModules	1 Per Client	1 Per Client

TABLE II describes the parameters of the OO7 Benchmark database.

Relevance and portability are two most concern aspects of performance and OO7 benchmark qualifies this test that appeal to the largest number of potential users. There are two sizes of the OO7 Benchmark database: small and medium.

V. PLANNING TO CONDUCT EXPERIMENT

To conduct the experiment, we have to analyze certain factors, which can be selection of benchmark, selection of hardware, selection of software and metrics for performance estimation.

A. Selection of Benchmarking Technique

The estimation technique of database system can be done by analytical modeling or simulation modeling or benchmarking. Here we have chosen benchmarking technique for the performance estimation of object-oriented database system. Benchmarking is also available in two shades which are as following:

- Micro-benchmarks
- Standard benchmarks

Micro-benchmarks are specialized, stand-alone piece of software, which are focused on problem. They have controllable workload and data characteristics and useful for detailed, in-depth analysis. But the major drawback with this benchmark is that they are not standard and neglect contribution of local costs to global/total costs. Generalization of result is also difficult in this benchmark.

Standard benchmarks are publicly available. Metrics used for the analysis are well defined and easily comparable. Sometimes they are very large in size and complicated. These benchmarks have limited dataset variation and limited workload variation. Benchmarks such as TPC-A, TPC-B, TPC-C, OO1 and OO7, used for RDBMS, ORDBMS and OODBMS, are the example of standard benchmark. Let us select standard benchmark OO7 for the performance estimation of OODBMS.

B. Selection for the hardware

CPU: Intel Pentium Dual CPU 2.20 GHz

Main memory: 1 GB RAM

Disk (system): 120 GB Laptop ATA disk @ 5400RPM

C. Selection for the software

Application Software: Db4o version 7.12

Microsoft SQL Server 2008
Microsoft .NET Framework (3.5)

Operating system : Microsoft Windows XP Professional (Version 2002) SP1

Programming language: ASP.NET with C#

D. Metrics: What and how to measure

- *Basic*

  Creation Time of Objects

  Insertion Time of Objects

  Deletion Time of Objects

  Querying Time

- *Comparison*

  Analysis

  Response Time

- *Cold Run and Hot Run*

In cold run, we run a query just after a DBMS has started and neither benchmark relevant data nor DBMS / File system cache is loaded in the main memory of system. This clean state can be achieved by a system reboot. Another way to get this state is to run such application on system which accesses sufficient data to flush file system caches, main memory, and CPU caches. On other hand, in hot run a query is executed such that much relevant data is available close to the CPU when the measured run starts. This state can be achieved by running the query at least once before the actual measures are taken.

VI. Performance Analysis of OODBMS and Its Comparasion With RDBMS

Performance analysis of OODBMS is the third phase of feature analysis in performance framework. Creation, deletion, Insertion of objects and execution of query is performed in this step. The results are taken on hot and cold runs.

Metrics chosen for the comparative estimation are analysis and calculation of Response Time for various tasks. Firstly, the general analysis based comparison is made between relational database management system and object-oriented database system. In second step response time is calculated for the comparative analysis.

Comparison between OODBMS and RDBMS

A. Main Goal

Main goal of object-oriented database system is to provide encapsulation and class independence features for the objects. The classes can be reorganized without affecting its usage in any application. Main objective relational database system is to reorganize the data physically without affecting its usage. That means it ensures the feature of data independence from application programs.

B. Representation used in Data Models

TABLE III. Representations used in OODBMS and RDBMS

OODBMS	**RDBMS**
Object	Entity
Class	Types of entity
Hierarchy of Class	Database Schema
Instance of Class	Tuple
Attribute	Attribute
Relation	Relation
Object ID	Primary Key
Encapsulation	NA
Inheritance	NA

C. Conceptual Model

OODBMS uses consistent conceptual model where the classes of objects represents the concept of application. They use a consistent model for the analysis, designing, programming and accessing the database. On other hand, the RDBMS uses different conceptual model for analysis, designing and programming.

D. Storage

OODBMS can store both data and methods. The storage is n the form of active objects which can execute their methods. It also fulfills the encapsulation property where data the stored data can be used through the methods of their classes.

RDBMS can store only data. The stored data is passive in nature as the operations are brought into use when the data is used. Data partitioning is possible depending on user requirements and applications.

E. Usage of Data

The data structure used for OODBMS may be complex, which involves different types of data types for sound, image, video etc. To increase the performance of system the concept of chained data is used.

The RDBMS stores data using simple data structures as attribute, tuples and relation. Separate tables are used to store different data and join operators are used to fetch data from to combine the data.

F. Redundancy

In OODBMS, The data and methods used are non-redundant which is achieved by using encapsulation and inheritance. In RDBMS, Data non-redundancy is achieved by data normalization which eliminates or reducing data redundancy and the concept s used in the stage of designing the database.

G. Independence:

Independence of classes is also supported as the reorganization of classes is possible without affecting the mode of their usage. It fulfills the feature of data independence, i.e. the data can be reorganized and modified without affecting next higher level of design.

H. Calculation of Response Time

To test the performance a simple class is developed with some attributes and define a primary key. Objects of the class are inserted and then queried.

On other hand, a table is created in RDBMS with same attributes and primary key. Tuples are inserted into the table and an equivalent program is used to query the information.

VII. Proposal of UML based Performance Model

Software Performance Engineering (SPE) is a technique that proposes to use quantitative methods and performance models in order to assess the performance effects of different design and implementation alternatives during the development of a system. In this section, UML class diagram is used to represent the conceptual model of an object-oriented database design. Further, the class diagram is converted into the performance model. The process involves the transformation of class diagram into equivalent class diagram with the help of equivalence rules and by converting it into annotated class diagram and then performance model.

The UML is getting acceptance in the world of software engineering as a standard description provider of system. In this section, we have presented a performance analysis approach based on UML Class diagram. The proposed performance model is obtained from UML class diagram .The development of object-oriented database system is a process which requires maintenance and management of data. UML diagram present the conceptual designs of the object-oriented database system. Various diagrams used to present the system and their purposes are mentioned below:

Use Case diagrams	→ Workloads
Activity diagrams	→ Processing Steps
Deployment diagrams	→ Resources
Sequence Diagram	→ interaction among classes

In this section, an algorithm is proposed to translate an annotated UML diagrams into a Queuing Network (QN) performance model. Workload intensities are based on previous experiments conducted for the analysis of an open object-oriented database system.. The software architecture is described in term of annotated UML Use Case, Activity and Deployment diagrams. The objects of the persistent classes can be stored in object-oriented databases.

Proposed Steps for Generation of QN based Performance Model

Step 1. Design use case diagram and define their workload intensities

Step 2. Design class diagram and defining their statistics

Step 3. Inherit the data integrities constraints from the class diagram.

Step 4. Design sequence diagram with integrity constraints

Step 5. Design deployment diagram

Step 6. Transformation of class diagram to performance based equivalent class diagram

Step 7. Design of execution graph for given sequence diagram

Step 8. Design of Queuing Network based performance model using analytical approach

VIII. Conclusion

The framework designed for the performance estimation of object-oriented database system is very effective. The proposed framework focuses on performance analysis of object-oriented database system, comparative analysis of object-oriented database system and relational database system. From the work done in first sub section, it is concluded that OO7 benchmark is better as compare to hyper model and OO1. Hardware and software requirements are defined for the performance analysis and creation time, insertion time, deletion time and query time is computed. The third section of present work concludes that the proposed steps for the generation of performance model are very efficient. It involves the transformation of class diagram into equivalent annotated class diagrams, which is used for designing of execution graph and then queuing network based performance model is designed which is very effective. Further, to calculate the quantitative values related to performance, QSolver/1 can be used. Value of workload intensities and other inputs are given to a performance tool and their response times can be calculated.

Acknowledgment

The authors are thankful to Prof. R.C. Sobti, Vice-Chancellor, B.B. Ambedkar University (A Central University) Lucknow for providing the excellent facility in the computing lab of B. B. Ambedkar University, Lucknow, India. I am also thankful to Mr. A.B. Chaudhari, C. Berri and Mr. A. Alsaadi

for his support to understand the concepts used in the design and implementation of our paper.

REFERENCES

[1] C. Beeri, "A Formal Approach to Object-Oriented Databases," International Journal of Data & Knowledge Engineering, Vol. 5, No. 4, December 1990, pp.353-382.

[2] E. Bertino and L. Martino, Object-Oriented Database Systems: Concepts and Architectures. Addison-Wesley, 1993.

[3] A. K. Thakore and Y. W. Su Stanley, "Performance Analysis of Parallel Object-Oriented Query Processing Algorithms", Distributed and Parallel Databases, Kluwer Academic Publishers, Boston. 1994, pp. 59-100.

[4] A. Tiwary, V. R. Narasayya and H.M. Levy, "Evaluation of OO7 as a system and an application benchmark," Proc. of OOPSLA Workshop on Object Database Behavior, Benchmarks and Performance, Austin, Texas, 1995.

[5] A. Biliris and E. Panagos, "A high performance configurable storage manager," Proc. of 11th International Conference on Data Engineering, Taipeh, Taiwan, March 1995, pp. 35-43.

[6] A. B. Chaudhri, J. A. McCann and P. Osmon, "A Performance Study of Object Database Management Systems, Theory and Practice of Object Systems," Vol. 5, No. 4, 1999, pp. 263-279.

[7] H. Amer, "Automatic Transformation of UML Software Specifications into LQN Performance Models by using Graph Grammar Techniques," Master Thesis, Carleton University, Ottawa, Canada, 2001.

[8] N. Gorla, "An Object-Oriented Database Design for Improved Performance," Journal of Data & Knowledge Engineering, Vol. 37, 2001, pp. 117-138.

[9] Object Management Group, "UML Profile for Schedulability, Performance, and Time Specification," OMG Adopted Specification ptc/02-03-02, July 1, 2002.

[10] J. Hillston and Y. Wang, "Performance evaluation of UML models via automatically generated simulation models," Proc. of the 19th Annual UK Performance Engineering Workshop, Warwick, UK, 2003, pp. 64-78.

[11] E. Marcos, B. Vela and J.M. Cavero, "Methodological Approach for Object-Relational Database Design using UML," Journal on Software and System Modeling (SoSyM), Springer-Verlag, 2003.

[12] D. C. Petriu and C. M. Woodside, "Performance Analysis with UML," Proc. of workshop on UML for Real, 2003, pp. 221-240.

[13] M. Marzolla, "Simulation-Based Performance Modeling of UML Software Architectures," PhD Thesis TD-2004-1, Dip. Di Informatica, Universit`a Ca' Foscari, Venezia, Italy, February 2004.

[14] A. Alsaadi, "A Performance Analysis Approach Based on the UML Class Diagram," Proc. of the 4th international workshop on Software and performance, ACM New York, USA, 2004, pp. 254-260.

[15] S. Balsano and M. Marzolla, "Performance Evaluation of UML Software Architectures with Multiclass Queuing Network Models," Proc. of the 5th international Workshop on Software and performance (WOSP'05), Palma, Illes Balears, Spain, July 12-14, 2005, pp. 37-42.

[16] M. Alam and S. K. Wasan, "Migration from Relational Database into Object Oriented Database," Journal of Computer Science, Vol. 2, No. 10, 2006, pp. 781-784.

[17] U. A. Umoh, E. O. Nwachukwu, I. J. Eyoh and A. A. Umoh, "Object Oriented Database Management System: A UML Design Approach," The Pacific Journal of Science and Technology, Vol. 10. No. 2. November 2009.

[18] V. Saxena and A. Pratap, "Representation of Object-Oriented Database for the Development of Web Application using Db4o," Journal of Software Engineering and Applications, USA, Vol.3, No.2, June 2012, pp 109-117.

AUTHORS PROFILE

Dr. Vipin Saxena is a Professor & Head, Department of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow, India. He got his M.Phil. Degree in Computer Application in 1991 & Ph.D. Degree work on Scientific Computing from University of Roorkee (renamed as Indian Institute of Technology, Roorkee, India) in 1997. He has more than 16 years teaching experience and 19 years research experience in the field of Scientific Computing & Software Engineering. Currently he is proposing various software designs by the use of Unified Modeling Language for the research problems related to the Software Domains & Advanced Computer Architecture. He has published more than 91 International and National research papers in various refereed Journals and authored four books covering Software Engineering, E-Learning and Operating System. Dr. Saxena is a life time member of Indian Congress.
Mobile: 0091-9452372550
E-mail: vsax1@rediffmail.com

Ajay Pratap is a research student in Department of Computer Science, Babashaheb Bhimrao Ambedkar University, Lucknow, India. Earlier, he got his M.Phil. Degree in Computer Science in 2008 and Master of Computer Application from the said University & presently he is working on performance estimation of object oriented database through UML.
Mobile: 0091-9936-189152
E-mail: pratap_aj@yahoo.co.in
Authors Profile …