An Automated Environment for Design Based Performance Prediction of Component Based Software Products

Jasmine K.S, Dr. R. Vasantha

Sr. Lecturer, Dept.Of MCA,

R.V.college of Engineering,

Mysore road, Bangalore. -560059,Karnataka, India

***Abstract*—** Component-Based software engineering provides an opportunity for better quality and increased productivity in software development by using reusable software components [10]. One of the most critical aspects of the quality of a software system is its performance. The systematic application of software performance engineering techniques throughout the development process can help to identify design alternatives that preserve desirable qualities such as extensibility and reusability while meeting performance objectives [1]. In the present scenario, software engineering methodologies strongly focus on the functionality of the system, while applying a "fix- it-later" approach to software performance aspects [3]. As a result, lengthy fine-tunings, expensive extra hard ware, or even redesigns are necessary for the system to meet the performance requirements. In this paper, we propose an automated environment for the design based performance prediction of component based software products to reduce the overhead associated in the later phases while developing a performance guaranteed software product with the help of Unified Modeling Language (UML).

***Keywords*—** Software Reuse, Component-based development, Unified Modeling Language, Software performance, Software components, Performance engineering, Software engineering.

Jasmine K.S is with RVCE, Visveswaraya Technological University, Sr.Lecturer, Dept of MCA, Bangalore, India (phone: +919342969571; e-mail: jasminesadeep@yahoo.co.in).

Dr. R. Vasantha is with RVCE, visveswaraya Technological University, Prof, Dept of ISE, Bangalore, India (e-mail: vasanthaprak@yahoo.com).

.

Manuscript received August 5, 2008.
Manuscript revised August 20, 2008.

I. INTRODUCTION

THE design and construction of future software systems will require the integration of software analysis and design methods with Software Performance Engineering (SPE) in reuse based software development. This integration allows software designers to explore design alternatives and select a design that provides the best overall combination of understandability, reusability, modifiability and performance so that software systems can meet performance goals [1]. Central to improve the practice of performance implementation is the understanding that good design and management of resources will avoid the component communication bottleneck. Effective planning enables the organization to identify what type of practices is required for their products and plan ahead of time [2].

A. Component-Based development and Reuse

The reuse approach to software development has been used for many years. However, the recent emergence of new technologies has significantly increased the possibilities of building systems and applications from reusable components. Large scale component reuse leads to savings in development resources, enabling these resources to be applied to areas such as quality improvement. Experience has shown that component-based development requires a systematic approach to and focus on the component aspects of software development [19]. The building of systems from components and the building of components for different systems require established methodologies and processes not only in relation to the development/maintenance aspects, but also to the entire component and system lifecycle. There are a number of software engineering disciplines and processes, which require specific methodologies for application in component-based development.

Current thinking is that the progress of software development in the near future will depend very much on the successful establishment of CBSE and this is recognized by both industry and academia.

B. Software Performance Engineering (SPE)

SPE is a method for constructing systems to meet performance objectives [12]. The process begins early in development and uses quantitative techniques to identify satisfactory designs and to eliminate those that are likely to have unacceptable performance before developers invest significant time in their implementation. SPE continues through the detailed-design, implementation and testing phases to predict and manage the performance of the evolving software and to monitor and report actual performance against specifications and predictions.

In particular, performance properties are essential in the context of component based software production for two basic reasons [13]:

1. Among multiple component implementations providing the same functional behavior, the clients will choose those components that best fit their performance requirements.

2. If components have performance specifications, then the performance of the system can be compositionally derived based on its components, while the component implementations need not be re-analyzed in each new context where they are used.

Our research work aims at developing a design based, implementation independent performance guaranteed software product by combining the most recent advances in the fields of: (i) Component based software engineering (CBSE) (ii) Software Performance Engineering (SPE) and (iii) UML modeling of CB systems design. Our basic idea is to adapt the SPE approach to CB development in the design phase to achieve success in both the components and CB applications that guarantee specific performance requirements.

C. Present state in software reuse world & SPE

In the research community, there are notable approaches to software performance engineering. Recent interest in software architectures has underscored the importance of architecture in achieving software quality objectives, including performance. While decisions made at every phase of the development process are important, architectural decisions have the greatest impact on quality attributes such as modifiability, reusability, reliability, and performance [11].

The methodology for performance engineering demands extra effort and capabilities. Much recent researches are aimed at automating the performance modeling process [3][7][8]. But there is a need to specify performance parameters in these models. It requires skilled people. Our research aimed at facilitating this modeling process in the design level with the help of most widely used software-modeling language, namely unified modeling language (UML). Consequently UML diagrams, especially sequence diagrams, collaboration diagrams, activity diagrams and deployment diagrams play an important role in this process.

II. Implementation

A. Performance Prediction Methodology

The SPE process begins with the system's use cases [6]. Use cases describe the major functionalities of the system. Here we focus on the scenarios that describe the use cases. The scenario shows the objects that participate and the messages that flow between them. Performance scenarios are the subset of the use case scenarios that are executed frequently, or those that are critical to the perceived performance of the system. We use Unified Modeling Language (UML) sequence diagrams (SD) to represent performance scenarios. The SD objects represent the components involved, and the SD messages represent the requests of execution of a component service or correspond to information/data exchanged between the components.

We can show synchronous and asynchronous messages in the UML using different types of arrowheads. In Figure 1 the communication between CompB and CompC is a synchronous communication and between CompC and CompD is an asynchronous communication. Also CompD has a self-delegation loop. All these examples use standard UML notations. Additional extensions to the sequence diagram notation are in [14].

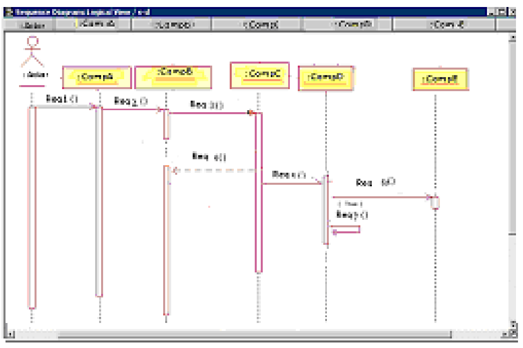


Fig1: Sequence Diagram

A UML activity diagram shows the operational workflow of a system i.e., it will tell us which activities are executing sequentially and concurrently.

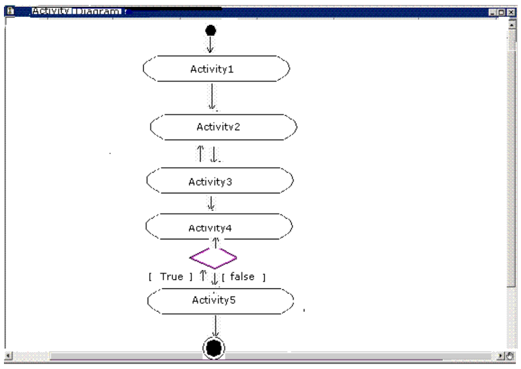


Fig2: Simple Activity Diagram

In fig 2, activity 1 to activity 4 is sequential in nature. Then a condition check is taking place, if the condition is true (corresponding to the self-delegation loop in sequence diagram, control will go back to action 4 itself. If the condition is false, the control will go to activity5.

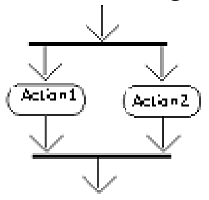


Fig3: Activity Diagram depicts concurrent activities

In fig 3 action 1 and action 2 are concurrent activities.

A UML deployment diagram (DD) shows the allocation of the software components of the system to the processing nodes and the interconnection between the processing nodes (processes, workstations, I/O devices). The same diagrams can be re-used for similar applications, by only updating the associated parameters. The SD and DD diagrams have to be annotated with the proper performance values and parameters (PAs). For example, system and component execution times, response times, resource utilization (CPU utilization, disk, memory, network) I/O rates and average service time, network utilization, message size etc. A sample DD is shown in fig 4.

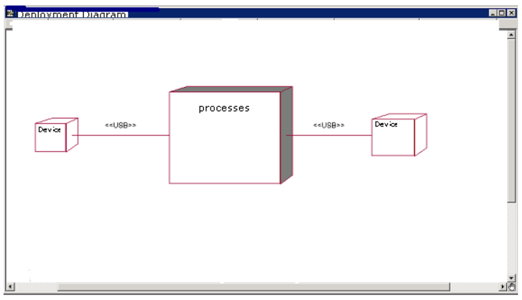


Fig4: Deployment Diagram

UML collaboration diagram describes how the software components interact. An illustration is given in fig 5.The transformation from a sequence diagram into a collaboration diagram is a bi-directional function. The difference between sequence diagrams and collaboration diagrams is that collaboration diagrams emphasize more on the structure than the sequence of interactions. Within sequence diagrams the order of interactions is established by vertical positioning whereas in collaboration diagrams the sequence is given by numbering the interactions. By observing the number of arrows leading to a particular component, the utilization of that component can be predicted. So the requests sent to that component by other components have to wait, therefore response time will be more for them, resulting in performance degradation.

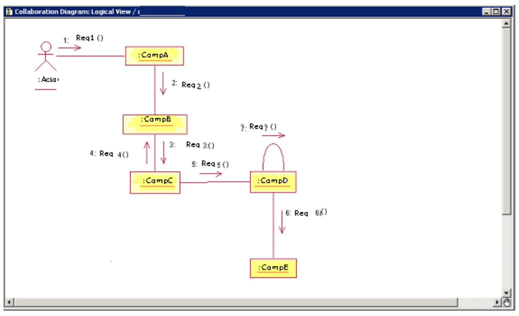


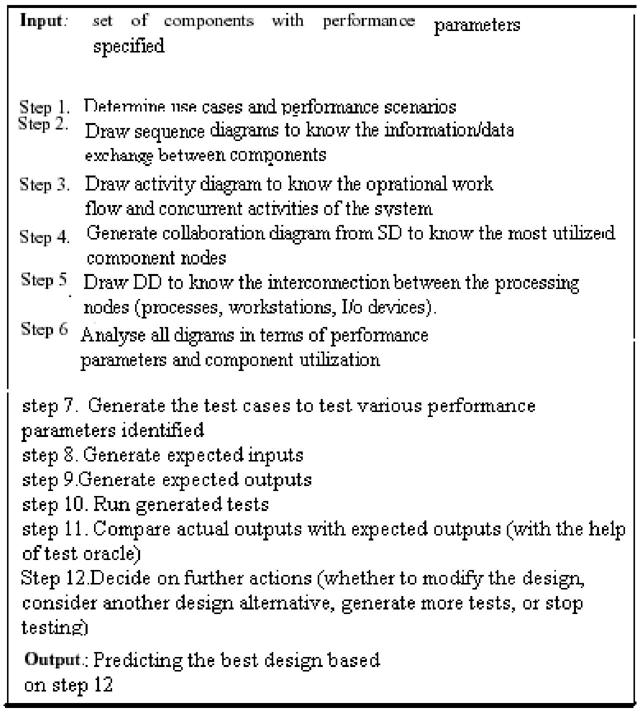
Fig5: Collaboration diagram

Considering the DD nodes, the PA attributes concern the resource scheduling policy (i.e. the strategy by which the resource handles the different jobs), the resource utilization and the resource throughput that represents the amount of work provided per unit of time by a resource belonging to a certain node.

In fig5, there is a two-way communication taking place between CompB and CompC.Also the CompD has to respond to CompC and it also has a self-loop. So from the diagram, CompC and CompD are the most utilized component nodes compared to other component nodes. So the performance attributes of these components have to be monitored seriously.

B. *Stages in Performance Prediction Process*

In the design phase of the SPE based Software development, the following steps can be adopted.



Input: set of components with performance parameters specified

Step 1. Determine use cases and performance scenarios
Step 2. Draw sequence diagrams to know the information/data exchange between components
Step 3. Draw activity diagram to know the oprational work flow and concurrent activities of the system
Step 4. Generate collaboration diagram from SD to know the most utilized component nodes
Step 5. Draw DD to know the interconnection between the processing nodes (processes, workstations, I/o devices).
Step 6 Analyse all digrams in terms of performance parameters and component utilization

step 7. Generate the test cases to test various performance parameters identified
step 8. Generate expected inputs
step 9.Generate expected outputs
step 10. Run generated tests
step 11. Compare actual outputs with expected outputs (with the help of test oracle)
Step 12.Decide on further actions (whether to modify the design, consider another design alternative, generate more tests, or stop testing)

Output: Predicting the best design based on step 12

Fig6: Steps involved

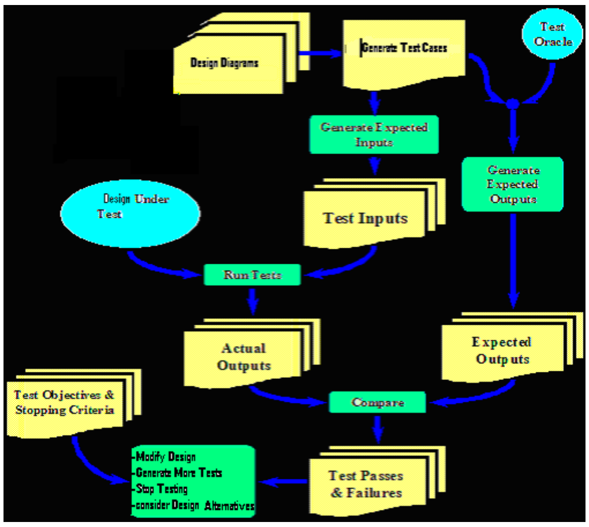


Fig7: Proposed Automated Environment

III. CONCLUSION

This paper presents automated environment for implementation of the engineering approach to encompass performance prediction in component-based systems on the basis of design specification. We have defined an original approach that relies on, the most recent advances in the fields of: (i) Component based software engineering (CBSE) (ii) Software Performance Engineering (SPE) and (iii) UML modeling of CB systems. A stepwise approach to adapt the SPE approach to CB development in the design phase to achieve success in both the components and CB applications that guarantee specific performance requirements is given. Future work can be focused on its application to case studies coming from the industrial world.

ACKNOWLEDGMENT

This study has been possible thanks to the support and collaboration of several industry people they are directly or indirectly contributed towards the successful completion of this research.

REFERENCES

[1] Mangano, “An Approach to Performance Evaluation of Software Architectures”, Workshop on Software and Performance, Santa Fe, NM, ACM, 1998, pp. 178-190.

[2] B. W. Boehm, "Verifying and Validating Software Requirements and Design Specifications," IEEE Software, vol. 1, no. 1, pp. 75-88, 1984.

[3] Tom Verdict, Bart Dhoedt, Frank Gielen, and Piet Demeester, Senior member, IEEE, “ automatic Inclusion Of Middleware Performance Attributes into Architectural UML Software Models”, IEEE Trans. On Soft.Engg. VOL 31,no.8, August 2005.

[4] D.E. Harms and B.W. Weide, “Copying and Swapping: Influences on the Design of Reusable Software Components”, IEEE Trans. Soft. Eng. 17, 5 (1991), 424-43s.

[5] Yan liu, Member, IEEE, Alan Fekete, Member IEEE Computer Society, and Ian Gorton, Member, IEEE, “Design-Level Performance prediction of Component-Based Applications”, IEEE Trans. On Soft.Engg. VOL 31,no.11, November 2005.

[6] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language, User Guide*, Reading, MA, Addison Wesley, 1999.

[7] V.Cortelessa, A, D’Ambrogio, and G.Iazolla, “Automatic Derivation of software performance models from case Documents”, performance evaluation, vol.45, no.s 2-3,pp.81-105, July 2001.

[8] P.G. Gu and D.C. Petriu, “XSLT transformation from UML Model to LQN performance models”, proc. Third Int’l workshop Software performance (WOSP ‘2002), pp.227-234.July 2002.

[9] R. Kazman, et al., “Scenario-Based Analysis of Software Architecture”, IEEE Software, vol. 13, no. 6, 1996, pp.47-55.

[10] X.Wu and M. Woodside, “Performance Modeling from Software components”, proceedings of the Fourth International Workshop on Software and Performance, pp.290-301, 2004.

[11] Balsamo and Mangano 1998] S. Balsamo, P. Inverardi, and C. Mangano, “An Approach to Performance Evaluation of Software Architectures”, Workshop on Software and Performance, Santa Fe, NM, ACM, 1998, pp. 178-190.

[12] C. U. Smith, *Performance Engineering of Software Systems*, Reading, MA, Addison-Wesley, 1990.

[13] Sitaraman M., et al., “Performance specification of software components”, Proc. of SSR '01, p. 310. ACM/SIGSOFT, May 2001.

[14] Connie U. Smith and Lloyd G. Williams, *Responsive, Scalable Systems: Practical Performance Engineering for Object-Oriented Software*, 2001.

[15] Cortellessa, V., Mirandola, R. PRIMA-UML: a Performance Validation Incremental Methodology on Early UML Diagrams, Science of Computer Programming, 44 (2002), 101-129, July 2002, Elsevier Science

[16] Lavenberg S.S. *Computer Performance Modeling Handbook*, Academic Press, New York, 1983.

[17] UML Profile for Schedulability, Performance, and Time Specification: ttp://cgi.omg.org/docs/ptc/02-03-02.pdf.

[18] UML Documentation version 1.4 Web site. Online http://www.rational.com/uml/resources/documentation/.

[19] Crnkovic, I. and Larsson, M. “A Case Study: Demands on Component-based Development”, Proceedings 22nd International Conference on Software Engineering, ACM Press, 2000.

Jasmine K.S born in the Ernakulam District Of Kerala state on October 14th in the year 1971. She received BSc degree in Mathematics from Mahatma Gandhi University, Kerala in 1991, MSc degree in computer science from Kerala University, Kerala in 1994 and M.Phil degree in computer science from Bharathidasan University, Tamilnadu in 2005. She is currently doing her PhD in computer science in Mother Teresa University, Kodaikanal, Tamilnadu.

She is a Senior lecturer in the Department of MCA, R.V.College of Engineering, Bangalore. Since from 1995, she is working as a lecturer in the field of computer science. During 98-99,She held a visiting faculty position at Visveswarapura College of science, Bangalore. She has authored 11 research papers in the national and international level conferences. Her research interests include Software reuse, Software performance, Software testing, data mining and experimental software engineering.

Ms. Jasmine is the member of Indian society for technical education (ISTE), Computer society of India (CSI) and International Society for computer applications (ISCA).

Dr.R.Vasantha received BSc degree, majored in Physiscs, Chemistry and Mathematics from University of Mysore, India in 1976, MSc degree in Mathematics from Manasa Gangotri, University of Mysore, India in 1978 and PhD from Indian Institute of Science, Bangalore, India in 1985.

She is a professor in the Department of Information science and Engineering, R.V.College of Engineering, Bangalore. She got more than 20 years of research experience. During Oct.1991-Oct.1994, she worked as a Scientist in National Aeronautical Laboratory, Bangalore, India, on Turbulence modeling of aerofoil.Sept1988-Sept.1991: Worked as Research Associate in the Dept. of Mechanical Engineering, University of New South Wales & University of Sydney, Sydney, Australia on turbulence modeling. Oct.1987-Sept.1988: Worked as a Senior Research Associate in the School of Mathematics, University of East Anglia, Norwich, England, on the initiation of detonation waves, Oct.1986-Oct.1987: Worked as a Senior Visiting Fellow in the School of Mathematics, University of East Anglia, Norwich, England. Aug.1985-Aug.1986: Worked as Research Associate in Dept. of Aerospace Engineering, IISc, Bangalore, India. She also has many years of teaching experience. During 1978-1980, she worked as a Lecturer, 1994-2002: Worked as Associate Professor of Mathematics, 2002-2006: Worked as Professor & HOD of Mathematics in an Engineering College in India. Her research interests are in the field of computational fluid mechanics. She is the author and coauthor of several publications appearing in international journals, books, and conference proceedings in the fields of Applied Mathematics and Computational fluid mechanics.

Dr. Vasantha is a Gold medal contender for PhD dissertation. And also she Won Scholarship from B.Sc. till the end of Ph.D.