Converting UML Description of Software Architecture to QNM and Performance Evaluation

Rahmat Zolfaghari

Abstract- ***Converting UML Description of Software Architecture to QNM, provides a comparison between all kinds of designs with respect to performance indicators.Present study suggested a method for converting the UML description designing software to Queuing networks model (QN)model, which provides the application of using the UML in designing software with high performance; in other words it putting the performance in designing software and a high quality software is designed.In order tomodeling the parts of system we use deployment diagram for allocating software components to hardware resources and activity diagrams to model software behavior, and use case diagrams to model workloads with the performance profile and An algorithm is provided for automatic production of the QN performance model from the XML(Extensible Markup Language) documents .used diagrams with performance profiles(stereotype, label and limitation), using the ExportXMIsoftware rational rose. performace modelload to software tool for performance analysis, so as the designer can test thefulfill of performance goals of his design according to type of different performance parameters and changing in value and chooses the best option in designing.***

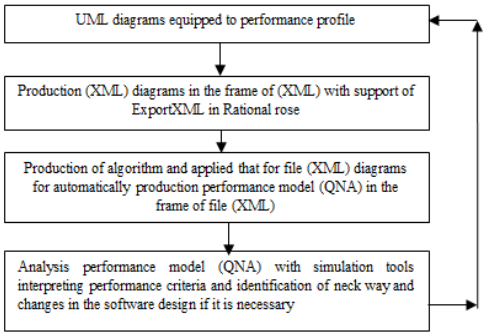
Keywords- ***UML(Unified Modeling Language),Queuing networks model (QNA)and Performance evaluation***

I. INTRODUCTION

The quality of most large and complicated software systems is determined by performance indicators of them, such as response time, error rate and operating power. If the requirements of the performance are not met,the relationship with customers is damaged, the effectiveness of the users is lost, decrease in income, expenses will exceed the allowed margin by reason of redesign and the market will be lost. Also, in a comprehensive analysis,it is indicated that the issues and problems related to performance is the second major and effective reason of software projects successful .Therefore, the performance of software systems is a critical issue in development process and production of software's. Procedure modeling is an important and useful approach of evaluating the effectiveness and validation of the systems, as well as it predicts the requirements of the qualitative and quantitative performance and provides a comparison between all kinds of designs with respect to gained performance indicators; and then we can choose the optimum design that provides the performance aims of software.

Therefore we have to find a way for evaluating the performance, so that we can use analytical and simulation methods; however because of high costs of simulation methods, for evaluating the performance of computer systems, most often we use analytical and formal methods and techniques.System performance engineering means the extent to which responsibilities thatdelivered to system have been completed in spite of considered limitations, such as speed, accuracy, the amount of utilization of resources and memories engaged and ability of system replication. Performance in the architecturesoftware level orgiven the rates of entering, distribution of service requests, display ratio, response times, delays ratio in doing services and error ratio, perception, modeling and analysis.

In this study, we suggest a method to converting the UML description of designing software equipped with performanceprofile UML to performance model QNAin order to evaluate and analyze software systems engineering. UML is a standard language for displaying, describing structure and documenting a software system. We use the UML performance profile in displaying and describing the performance requirements in UML and using a standard symbolizing in describing these characteristics and supporting existing tools. UML models are portrayal, so we cannot perform or analyze them; therefore we convert the model to a formal operative QNA one, and do the quantitative and qualitative analysis of software systems with existing simulation tools. Interpreting the results, we can conclude that the software design is constructive and accept it, or in the case of a neck on the way, we renew the design and optimize it.



Figure(1). proposed framework for performance evaluation

Manuscript Received on July 2013.
Rahmat Zolfaghari, Islamic Azad University, Hashtgerd Branch, Department of Computer Engineering, Tehran, Iran

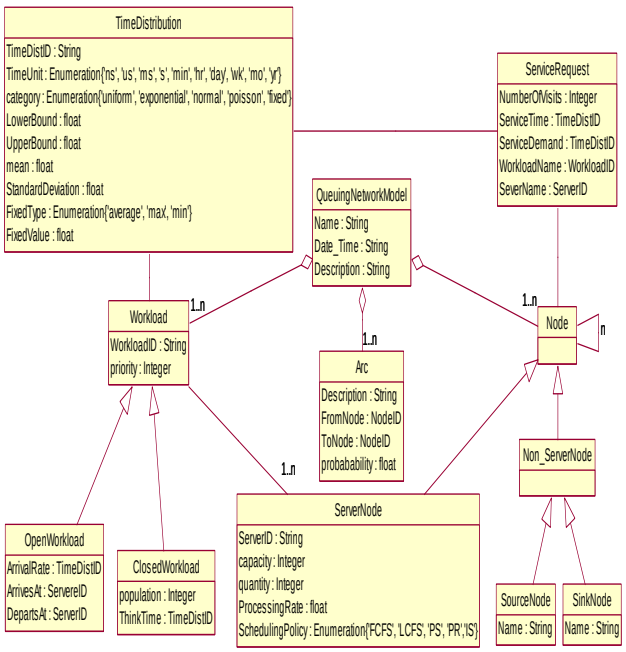
II. RELATED WORKS

Soft ware performance engineering has been established by Smith, is the first comprehensive approach that entered the performance analysis in software production process from the first steps to the end. Numerous approaches for deriving performance models from software architecture UML descriptions are suggested. Some of the approaches are based on architecture patterns; in which the pattern is described by the components forming the structure, their behavior and the way they interact with each other, and in most of these approaches, the performance model is the aim of line networks'. One of the approaches is based on the simulating the UML charts for evaluating performance. The problem of simulation is that very complex models need a lot of time and calculating resources in order to operate. Also to analyze the gained results, complex statistic methods are needed.

One of the approaches has been used UML and QNA for evaluating applied systems' performance. One approach has used UML and QNA for evaluating a video streamquality. One approach has been used UML and QNAfor evaluating web service system performance.

III. UML MODELING LANGUAGE

UML: this methodology has been used to modeling software systems especially driven-object-based .it is an open language and is fully expandable and supporting with a boarder set of tools (RatonalRose,ArgoUML). Various diagrams in UML language exist in order to modeling different aspect of soft ware system that in the case study because of proportion chart diagrams mood and co operation equipped to performance profile have been used to modeling features and interaction between features respectively. The methodology is known for most readers.



Figure(2). QNM metamodel

IV. DEFINITION OF STRUCTURE QNM

As Figure 2 shows, a QNM is composed of one or more Node, one or more of the Arc, one or more Workload. An Arc, a Node to connect to another and determine the service of request transition (with probability) from one to another. A Node shows entity of the operating environment, that provides the service (ServerNode) or determined the topology of model (Non_ServerNode). Non_ServerNode is of two kinds: SourceNode OpenWorkload shows the origin and SinkNode that specifies a point of departure.

A ServerNode provide a service to one or more Workload.A Workload, a series of similar transactions or service requests from customers indicate that they are of two types: OpenWorkload, An open workload shows the potential of unlimited transactions or customers from outside of system in a certain pattern (ArrivalRate) entered his service received and it will crack. Its population is ever changing; ClosedWorkload, shows a steady population workload between Server are in circulation. Thinking outside of the service delay time parameter (ThinkTime) to be determined.Request, the server is working load. ServiceRequest amount of time a service provided by the number of visits for each workload that the server meets the demand for the service (service time multiplied by the number of visits) are identified. Check with a service request, the workload will be transferred to other nodes with a certain probability.Penentrymodelworkloads, workloadsexternal delaypacketservicetime distributionandthe service demandon eachServerNode,is expressedbyaprobability distribution function. TimeDistribution,measured inunitsof time, the probability distribution(exponential, Poisson, uniform, normal and stationary) withdisplayingtheir parameters.

V. METHODOLOGY

5-1UML-basedperformanceevaluations

UML,widelyusedfor modelingsoftware systems, especially applied with object - based way. UMLdiagramsare availableonaparticular aspect ofthesystemare modelingviewpoint. UMLrecent successisprimarilydue tothe following reasons**:**

-static and dynamicaspects ofthe softwarethat allows ustousevariousgraphs, each representingadifferent viewofthesoftware system, in the modelwe have.

- Uniformconceptual frameworkandnotationfor describingthe designandimplementationcan beimplemented.
- UMLisa languagepackfora specificapplicationiscompletelyopen,but languageisextensible.
- It iswidelysupportedbya large setof tools.

Here we examined only a limited subset of graphs that analyze and evaluate the performance of the methodology. Foreign Actors andUse Cases describing the system, or in any other use of the system by each actor, use case diagrams are shown. Brief descriptions of system use cases from the perspective of end users or other systems interacting with external actor they offer.

Each user of the software system displays one or more scenarios and these scenarios in more detail by other diagrams such as sequence diagrams or activity are discussed. Performance evaluation, workload modeling applied to the system, the user profile will be used. Each edge can be attributed to a possible connection between an actor and the corresponding probability of use case or using it expresses. To determine the parameters of the workload, the diagram with information efficiently, the performance profile, UML annotations, we model these parameters are specified in the user request.

Activity diagram manifesting behavior of a use case. In addition to the graph modeling the dynamic behavior of the system can execute loops, conditional branches and structures fork / join model could also be useful in the performance evaluation process. In each case the user will be illustrated with an activity diagram. Operations are required to state the chronological order of connectivity modes of operation are modeled by transitions. Whatever the functionality required decomposed into atomic actions, each uninterrupted by a computerized tool such as CPU, device I / O, disk, etc., are executed. Pair of bars, sync icons decisions and activities was fabricated for modeling and implementing concurrent, conditional and repeating atomic functions, are used. Runtime characteristics (eg, request service, frequencies, priorities, using a unique computer tool, etc.) are determined for each of the notes that are attached to them. Atomic tasks to determine the device name assigned to the editing tools. This set of atomic activities, activity diagrams, computer tools to quantify the behavior of the users during the application process use case are working, as well. According to the above description, a diagram of the topology and configuration information is extracted QNM model.Physical architectureof the systemcompared toitslogical structureorbehavior, thedeploymentdiagramis determined. Computationalnodesofthe graphruntimeconfiguration, networkdesign objectsbetweenthemshowsuse caseanddistribution. Thegraphsof theperformancedatarelatedtophysicalcomputingdevices(eg, schedulingpolicy, a number of existing tools, buffercapacity, etc.) aremarginalnotes.

5-2 UML diagrams annotated with performance information from the QNM

UML diagrams describing the system, use case diagrams, activity and posture, the Rational Rose are designed and modeled. These diagrams automatically using an intermediate XML document Unisys XMI ,are converted. The structure of the document UMLX13-11.dtd is the physical met model UML 1.3 and the following design rules described in XMI-DTD described in XMI 1.1 , is produced. A QNM is produced for each use case. PC Tools Chart of the service centers are involved in QNM. Other service-related parameters of performance information attached to each element of the graph are extracted. Queuing network topology and routing probabilities between each other and how they connect to the service centers, the rejection of the scenarios that are modeled by graphs, are obtained. Each activity in the activity diagram host device specified. Therefore, all activities performed by a device, the workload on the form. Applied to the system workload and the workload off the opened package attached to the data obtained at the start of the activity diagram.

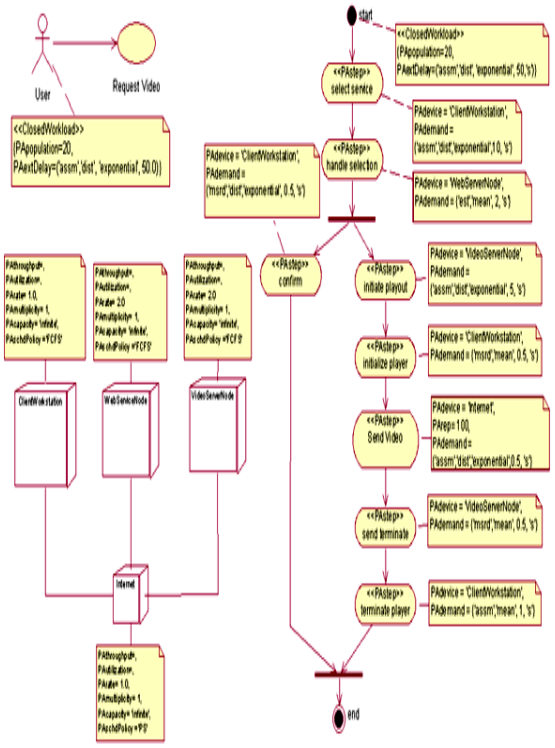
QNMmodelof the resultingXMLdocument formatis displayed. Documentstructure ofXML,based onQNM DTDdefined in a Sectionrealizedthat theis defined. QNMwithrelevant parametersisevaluatedbysimulation modelsoranalyticalandperformance measuressuch asresponse time, throughput andutilizationofresourcescould beachievedwiththis criterion measuretoasperformanceobjectives.

The following section provides a practical example, using process modeling software use case diagrams, UML deployment activities, as well as how to add information to its efficiency and production model QNM, has been studied in more detail.

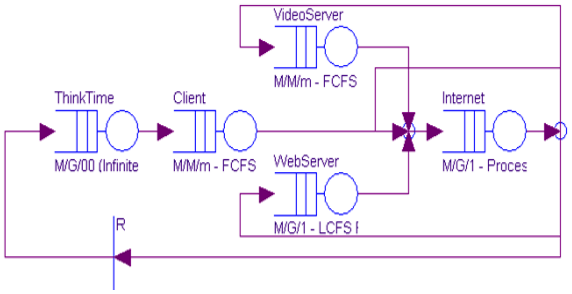
VI. CASE STUDY

in this section, we provide an application example, we describe the methodology outlined in the previous section. Listing provided similar efficacy profile in UML, is shown. In this example, an application includes Web-based video stream. Suppose that a population of N users in the system is fixed. Each user to view the video through a web browser, and the browser will choose to send video over the Internet using a web server provides remote call. View the video on a video server, workstation user, before sending the frame, it will be used. Use case UML diagrams (Use Case), deployment (Deployment) and activity (Activity), in Figure 3 are shown. Model elements of the UML notation with the efficiency of function annotation have been proposed in profiles. Actor (actor) Use case diagram shows a closed workload that users run the same population. Use case, video application scenarios and expresses the activity diagram, this scenario shows more detail. Diagram of the configuration of hardware resources to run these scenarios are shown.

According to the procedureoutlined inSection IV, whichis producedin theform ofQNMmodelis evident. QNM,bysoftware package[PEPSY] WinPEPSY,analysisandperformanceevaluationresults of thenumerical exampleofthe model inTable 1is obtained.Giventheseresultsandotherparametersof$N = 20$Userdataattached toelements of theUMLdiagramsareobtained. The totalthroughputofthe systemis estimated to be0.059728. Softwaredesignerby repeatanalysisQNMmodelwithdifferentinput parametersthesystemcanmeasureindifferent conditions. For example,in Figure(5) the level ofresourceutilizationand response timeto changingcustomernumberis shown. It is observedthatthenumber ofcustomersisgreaterthan10 theresourceutilizationratedoesnot change, but theresponse timeincreases.



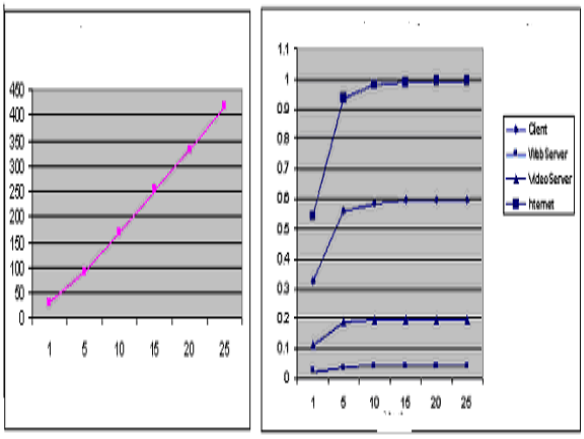
Figure(3). UMLdiagramsdescribe thevideo streamapplicationswithperformance information.



Figure(4). QNMmodelingvideo streamapplications

Table(1). Application performance evaluation of a video stream

	Utilization	throughput	response time	number of requesr	Waiting time	length Queue
Client	0.593	0.05971	23.123	1.354	13.134	0.774
ebServer	0.04	0.01992	2.056	0.045	0.035	0.001
VideoServer	0.192	0.03972	6.168	0.247	1.167	0.045
Internet	0.991	1.99084	9.203	18.375	8.706	17.341



Figure(5). Level ofresourceutilizationand response timewiththenumber ofcustomers

SUMMARY

Inthis paper, an approach forperformance modelingbased onqueuingnetworks, isdesigned and implemented. DescribingsoftwarearchitecturewithUMLdone andtheperformance parameterssuchasannotation, according toperformanceprofile,UML,can beattachedtothe application model.
UMLdiagramsautomaticallyforXMLdocumentsusingXMI Unisys thatthepreparationis supportedinRational Rose, will beconverted. Efficientalgorithm forextracting informationfrom theUMLmodelsand generatesQNM,andprovidesa standardwayto display theQNMstructureis definedforitin the form ofXMLfiles. Performanceresultsare obtainedbyevaluation ofqueuingnetworkmodels. Apractical exampleofperformanceevaluationmethodologyand the resultswerepresented.

REFERENCES

1. *PEPSY-QNS* (**P**erformance **E**valuation and **P**rediction **SY**stem for **Q**ueueing **N**etwork**S**), (http://www4.informatik.uni-erlangen.de/Projects/PEPSY/en/), October 2004.
2. L.G williums, G.U. smith "performance Evaluation of Software Architectures" proc. Of wosp'98, santa Fe, New Mexico , USA, PP, 164-177 (1998)
3. Institute of Electrical and Electronic Engineers. "IEEE Standard Glossary of Software Engineering Terminology." IEEE Standards Collection. New York, NY: Institute of Electrical and Electronics Engineers, 1993.
4. P. Kahkipuro. "UML-based performance modeling framework for component- based distributed systems". In R. R. Dumke, C. Rautenstrauch, A. Schmietendorf, and A. Scholz, editors, *Performance Engineering*, volume 2047 of *LNCS*, pages 167–184. Springer-Verlag, 2001.
5. R.Class, Software Runaway, "Lessons learned form Massive Software project Failures, prentice Hall" 1998
6. smith C.U., Performance Engineering Of Software System, Adison—Wesley(1990)
7. Cortllessa, V.,Mirandola, R. PRIMA-UML:A Performanc Validation Incremental Methodology on Early UML Diagram 101-129(2002)
8. Gu, G., Petriu, D.C, XSLT tranformation from UML models to LQN performance models, In [WOSP02], PP. 227-23
9. Arief, L.B., Speirs, N.A., A UML Tool for an Automatic Generation of Simulation Program, In [WOSP00] PP. 71-76

10. Mirco Tribastone and Stephen Gilmore. Automatic extraction of PEPA performance models from UML activity diagrams annotated with the MARTE profile. In *Proceedings of the 7th International Workshop on Software and Performance (WOSP2008)*, pages 67-78, Princeton NJ, USA, 2008.
11. Mirco Tribastone and Stephen Gilmore. Automatic translation of UML sequence diagrams into PEPA models. In *5th International Conference on the Quantitative Evaluation of SysTems (QEST 2008)*, pages 205-214, St Malo, France, 2008.
12. Analysis of a Multimedia Stream Using SPA, H. Bowman, J.W.Brayans and a J. Derrick, University of Kent, 2001
13. Performance Modelling with UML and Stochastic Process Algebras, Catherine Canvent, Stephen Glimore, Jane Hiliston, Matthew Prowes and Perdita Stevens,2002
14. G. Gu, Dorina C. Petriu, "XSLT transformation from UML models to LQN performance models", In [ACM02], pp. 227– 234.
15. http://argouml.tigris.org
16. http://www.dcs.ed.ac.uk/pepa
17. http://www.sax.sourceforge.net
18. UML 1.3 DTD for XMI 1.1: UMLX13-11.dtd.
19. (http://cvs.berlios.de/cgi-bin/viewcvs.cgi/metadataserver/metadataserver/resources/UMLX13-11.dtd), 2004.

[1] Software Performance Engineering (SPE)
[1]Architectural Patterns
[1] Client/Server
[1] State Chart
[1] Colaboration
[1]Deterministic
[1] Probablistic
[1]Composition
[1] Axiomaticlaws
[1] Cooperation/Hiding Operator
[1]Continuse Time Markov Cain(CTMC)
[1] Packet
[1] Jitter
[1] Extensible Markup Languge(XML)
[1]Plug in

AUTHORS PROFILE

MR Rahmat Zolfaghari, is presently working as faculty in Islamic Azad University Hashtgerd Branch, Department of Computer Engineering, Tehran Iran, He is having 12 years experience both in industry and academia, He received his Software Engineering Bachelor (BS) of Shahid Beheshti University in Iran and Software Engineering Master (MS) of Sharif University of technology in Iran, His research interests are Database, Software Design , Modelling and E_Commerce.