Converting UML Description of Software Architecture to Stochastic Process Algebra and Performance Evaluation

Rahmat Zolfaghari

Islamic Azad University, Hashtgerd Branch, Department of Computer Engineering, Tehran, Iran
gzolfaghari@alum.sharif.edu

Abstract: Important qualitative parameters of the large software systems are determined by indicators of effectiveness of the software's, such as response time, operating power and error rate. Procedure modeling is an approach for evaluating the effectiveness and validation of the systems and, as well as it predicts the requirements of qualitative and quantitative performance and 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 Stochastic Process Algebra **(SPA)** 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 to modeling the parts of system we use state chart and for the interactions between the parts we use the Collaboration chart with the performance profile (using the performance profile is the distinction between the suggested approach and the former ones.). An algorithm is provided for automatic production of the SPA performance model from the XML(Extensible Markup Language) documents and state and collaboration charts with performance profiles(stereotype, label and limitation), using the **ExportXMI** software, ArgoUML is provided, and then they gained SPA performance/operation model in the **PEPAworkbench** tool is loaded for performance analysis, so as the designer can test the fulfill of performance goals of his design according to type of different performance parameters and changing in value and chooses the best option in designing.
[Rahmat Zolfaghari. **Converting UML Description of Software Architecture to Stochastic Process Algebra and Performance Evaluation.** *Rep Opinion* 2013;5(6):44-50] (ISSN: 1553-9865). http://www.sciencepub.net/report. 10

Keywords: Performance evaluation, UML (Unified Modeling Language), SPA (Stochastic Process Algebra) and performance profile

1- 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 [1]. 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 [2]. 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 that delivered 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 architecture software level or given 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 performance profile UML to performance model SPA in 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 SPA 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.

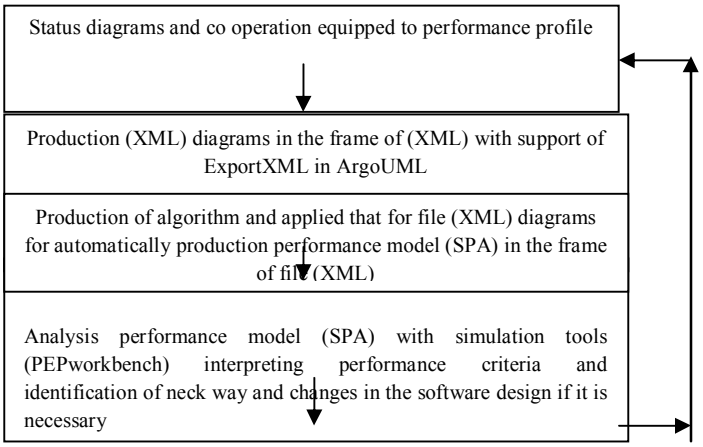

Figure 1: proposed framework for performance evaluation

2-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 SPA for evaluating applied systems' performance. One approach has used UML and SPA for evaluating a multimedia system's service quality. One approach has been used UML and SPA for evaluating web service system performance (customer/servitor).

3- 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.

3-1 UML performance profile

It is used to display performance information in UML charts that include stereotype, label and limitations and provide the following facilities:

- An accurate description of the performance requirements in designing domain.
- Attributing service quality indicators related to performance to a UML model parts.
- Indicating operative parameters that can be used by some tools to calculate predicted performance indicators.
- Presenting performance results obtained from calculation modeling tools or from measurements.

"PAstate" is the basic stereotype used in this paper that have the limitations and labels are presented in the following table:

Table1- Label "PAstate" descriptions

description	Label
The number of action repeat times in the component	PArep
action delay in the component	PAdelay
Maximum capacity of the component	PAcapicity
action rate of the component/operating power	PAthroughput
The name of the operating component	PAdevice
action rate achieved by the component	Parate
Possibility of stable state rate of the component	PAprobe
Name of the action	PAname

4- Stochastic Process Algebra (SPA)

In general, analytical methods of performance are split into two groups: Deterministic method and stochastic method. In Deterministic models, all the quantities are constant. In stochastic models, there is a

degree of probability and there exists stochastic quantities in the models. Line networks, pertness and Stochastic Process Algebra belong to this group.

The SPAs are famous modeling techniques for functional analysis of the systems that have the same trend. Process Algebra is described as the operator of atomic actions by a collection of existences or agents, and is used describe sequential behaviors that have the same trend and the synchronization of their communications. Process Algebra, a formal model from the systems that have the same trend, that is objective (internal behavior of system components can be neglected) and compound (a system can be modeled as the interactions between the subsystems). SPAs are developments of classic Process Algebra (CSP, LOTOS, and CCS) that add quantifying to their models and are appropriate for performance modeling. This development includes relating each a stochastic variable that shows the period of time to each action.

4-1 Stochastic Process Algebra performances

Line networks are stochastically able to analyze limited types of systems and surface petri network models and are not capable of determining a set of descriptions and compound validations. Because of these limitations, Stochastic Process Algebra operators' models such as PEPA, TIPP, MPA, EMPA and GSMPA are developed in recent years. For most of these models, analysis tools are provided. SPAs have appropriate properties for modeling complex, without delay, having same trend, distributed and communicative systems. The most important of these properties are:

- Composition; means that the behavior of a large system can be analyzed according to the behavior of the subsystems. It's shown as:
  $P+Q$
  Where P and Q are system components that are competing in operation, in other words, at the end the system will behave like P or Q.
- Axiomatic laws; these laws will reduce the size of the state processes space. It is shown as $P \overset{def}{\cong} Q$, where P and Q are equivalent and have the same manner in the system and can be a substitution for each other.
- Cooperation/hiding performance; with this performance every part of the system is defined by combining some subparts in parallel and every operation that has no interaction with other subparts as well as is hid before combination and this operation is repeated for every subpart inside part recursively,
  $P \underset{L}{\bowtie} Q|p/L$
  Where P and Q in operations that exist in the "L" set, cooperate in operating them simultaneously and operations that do not exist in the common "L" set will continue independently or be hidden.
- Prefix performance;
  (a,r).P ; where a system component operates an action of type "a" with rate "r" and then "P" is the result.

For modeling SPA systems, combination of exponential distribution and the chain of Markov processes using SPA properties (combination, equivalence and reduction) is a major approach. However this combination, reduction and equivalency of the components, can take place until the original system model is not changed, because in that case extracted performance indicators are not reliable. Therefore we can reduce a SPA model to a Markov chain with time continuity and then calculate the performance indicators.

4-2 Continuous Time Markov Chain(CTMC)

Markov Chain: It is a sequence of stochastic variables containing same identical sampling space and different probability distribution. Meanwhile, each stochastic variable in a Markov chain is dependent on its previous variable. In other words, the probability of transforming from the state of si to s_{i+1} depends on the state of s_i (Discrete).

In Continuous Time Markov Chain, the period in which a system stays in a state itself is an exponential stochastic variable. But in the discrete Markov, each transformation is a stage and the staying period is not discussed.

Our approach also extracts a Markov model out of structural characteristics of the software and this Markov model is the basis for our analysis. Stochastic processes algebra effectiveness analysis software (PEPAworkbench) from continuous time Markov chain is used for analyzing the effectiveness. In other words, SPA model has been resulted from UML diagrams which equipped with effectiveness profile is loaded for analyzing the tool. PEPA State Finder which uses the facilities of Continuous Time Markov Chain, results in the amount of stable state probabilistic for all produced optimum states (optimum systems which have been produced by this software for the system). And then, the effectiveness indicators are calculated by using the mathematical software of MAPLE with the MAPLE file produced by this tool (containing numerical amounts of the stable state probabilistic) and other specific primary data of the system [12].

4-3 stochastic processes algebra Meta model

Stochastic processes algebra composes of a number of constituents/states, transmission and transmission rate in which each component carries out a task of the system with a specific or nonspecific rate. The component stays in a state for a moment, and performs the action with a specific rate and then turns into a new state.

For conducting executive modeling of the systems with SPAs, first, its components are identified and these components are modeled in accordance to SPAs Meta models and then the components which synchronize and cooperate with each other to perform the system tasks are combined.

5- Case Study

In this study, a multimedia system is modeled; concerning that service effectiveness and quality are the main requirements of this system and service effectiveness parameters including; display rate and error rate are calculated and the system is evaluated.

The system includes the constituents 1) source, 2) channel, 3) sink, 4) Timer

Timer undertakes the task of supervising and calculating display delay and variation of the packets. The duty of each components of the system is to perform an action with a specific or nonspecific rate.

Now the modeling of the system (4 components of the system and their cooperation) with SPA is carried out in accordance with SPA Meta model [12].

Channel $\stackrel{def}{=}$ Channel$_0$

Channel$_0$ $\stackrel{def}{=}$ (transmit, $\top$). Channel$_1$

Channel$_i$ $\stackrel{def}{=}$ (transmit, $\top$). Channel$_{i+1}$ + (receive, r_{rec}). Channel$_{i-1}$ + (loss, r_{loss}). Channel$_{i-1}$ $+ 1 \le i \le 4$

Channel$_5$ $\stackrel{def}{=}$ (receive, r_{rec}). Channel$_4$ + (loss, r_{loss}). Channel$_4$

r_{loss}, r_{rec}, $\top$ is the time of performing the action.

The maximum capacity of the channel is 5. ($1 \le i \le 4$).

The channel either takes the packets to the sink or loses them. In modeling the system, the effectiveness parameters are calculated based on the numbers of the lost packets in the channel. These lost packets depend on the relation between the components, the rate of task performance and briefly the design of the architecture of the system which result in the effectiveness parameters and architectural evaluation of the software.

Snik $\stackrel{def}{=}$ Snik$_0$

Snik$_0$ $\stackrel{def}{=}$ (receive, $\top$). Sink$_1$, $1 \le i \le 2$

Sink$_i$ $\stackrel{def}{=}$ (receive, $\top$). Sink$_{i+1}$ (display, r_{disp}). Sink$_{R(i-1)}$

Sink$_{R(i-1)}$ $\stackrel{def}{=}$ (display, r_{disp}). Sink$_{R2}$ $0 \le i \le 2$ 3

Sink$_{Ri}$ $\stackrel{def}{=}$ (reset , r_{reset}). Sink$_i$

receive, display, reset are the name of the action.

$\top$;

is the nonspecific rate.

are the ti $\mathbf{r_{reset}}$, $\mathbf{r_{disp}}$, $\top$;

me of the action. The sink increases the packets and at the end shows a reset signal. The buffer capacity ($0 \le i \le 2$) is 3.

Timer $\stackrel{def}{=}$ Timer$_0$

Timer$_0$ $\stackrel{def}{=}$ (reset , $\top$). Timer$_1$

Timer$_i$ $\stackrel{def}{=}$ (tick , r_{tick}). Timer$_{i+1}$ + Timer$_0$

Timer$_6$ $\stackrel{def}{=}$ (terror , r_{error}). Timer$_1$ + Timer$_0$

reset, tick are the name of the action.

r_{reset}, r_{disp}, $\top$;

is the time of the action. If the packets arrive within the determined time limit, will show tick signal and if not, will show error signal.

5-1 UML diagrams and the algorithm applied on it

Designing the diagram equipped with the systems effectiveness profile is performed in Agro UML software. State and collaboration charts which are appropriate for our modeling are used to design the system. Then an algorithm equipped with effectiveness profile (which is made through ExportXML of ArgoUML software) is provided to produce SPA performance models form XML documents of the plotted charts automatically.

The resulting SPA model in the PEPA Work bench tool (fig.3) for the loaded analysis and the effectiveness parameters are calculated (Table 2). The actual calculated rate with the nominal (primary) rate 30,‹200,‹100,‹2000,60 which are the rate of transmission from the source, the transfer from the channel to the destination, display in the destination, the variation in display and the error are calculated (Fig. 2). The applied algorithm on XML of the state/system component diagram and the system components collaboration diagram are accordance to the algorithm 1 and algorithm 2. After applying these algorithms, the SPA model will be extracted in the form of XML. Of course, the defined algorithms process the XML documents by using DOM/SAX basic classes.

1 : terms ← empty list

2: for each statechat S do

3: for each state s (of S) do
4: beh ← empty list
5: for each outgoing transistion profile(PAdevice) (of s) do
6: w ← profile(PAname) 7: r ← profile(PArate) 8: target ← profile(PAdevice) 9: beh ←

1: cooperations ← empty list
2: for each association a (of the collaboration diagram) do
3: left_inst ← left element of a
4: Right_inst ← right element of a
5: left_sm ← statemchart of the class of left _ins
6: right_sm ← statechart of the class of right _ins
7: left ← leaf(initial state of left_sm,left_ins)
8: right ← leaf(initial state of right_sm ,right _ins
9: sync ← events in left_sm) ∩ (evebts in right_sm
10: cooperations ← cooperations +node(left,sync,right)
11: endfor
12: Root ← first element in cooperations
13: remove first element from cooperations
14: while cooperations is not empty do
15:counter ← 0
16 :for each cooperation c(in cooperations)do
17: root ← insert (root, c)
18: if root has changed then
19: remove c from cooperations
20: counter ← counter+1
21:end if 22:end for
23:if counter is still()then
24:break while
25: endif
26:end while
27:sys_eqn ← convert root to string
28:return sys_eqn

eh+"(w,r).target" 10: end for 11: n ← profile(PAdevice) 12: terms ← terms+"#n=beh_0 [+beh_1 [+…]]" 13: end for 14: return terms

1)The applied algorithm on the state diagram

insert(root,new,times = 0)
1 : if root is a leaf and new s a leaf **then**
2: return Node(root,[],new)
3: **else if** root is a leaf **then**
4: return new
5: elseif new is a leaf then **6: if** root.left contains a new instance **then** new ← left ← insert(root,left,new) **return** Node(new ← left,root,profile(PAname),root.right)
7: elseif root.right contains a new instance **then** new ← right ← insert(root,right,new) **return** Node(root.left,root, profile(PAname), new ← right) **else return** root unchanged
8: endif 8: if root.left contains new.left and root.right contains new.right **then** new_profile(PAname ← root.profile(PAname) ∪ new.profile(PAname) **return** Node(root.left,new_profile(PAname),root.right) **else if** root.left contains new.left **then if** root.right contains a new.right instance **then** new.profile(PAname) ⊆ root.profile(PAname)**then** w_right ← insert(root.right.new.right) **return** Node(root.left.profile(PAname), new_right) **8:endif** new ← left ← insert(root,left,new)
9: return Node(new ← left,root,profile(PAname),root.right) **elseif** root.right contains a new.right **then if** root.left contain anew.left instance **then 10:if** new.profle(PAname) ⊆ root.profile(PAname) **then** new ← left ← insert(root,left,new.left) **return** Node(new ← left,root,profile(PAname),root.right)
11: endif
12: new ← right ← insert(root,right,new) **return** Node(root.left.profile(PAname), new_right)
13: endif 14:if times>0 **return** root unchanged
15: end if 16:swap left and right leaves of new
17: return insert(root,new,*1)*

2) The applied algorithm on the collaboration diagram

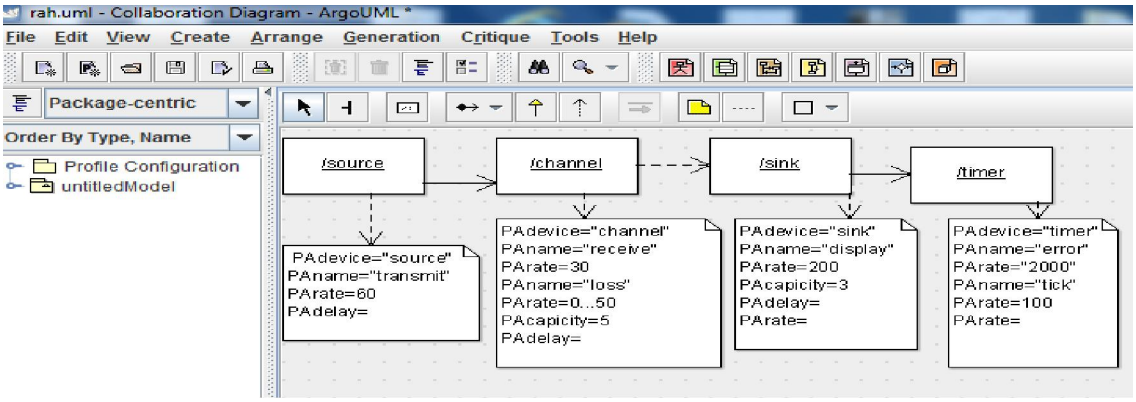

Fig. 2) Colaboration algorithm in ArgoUML Software

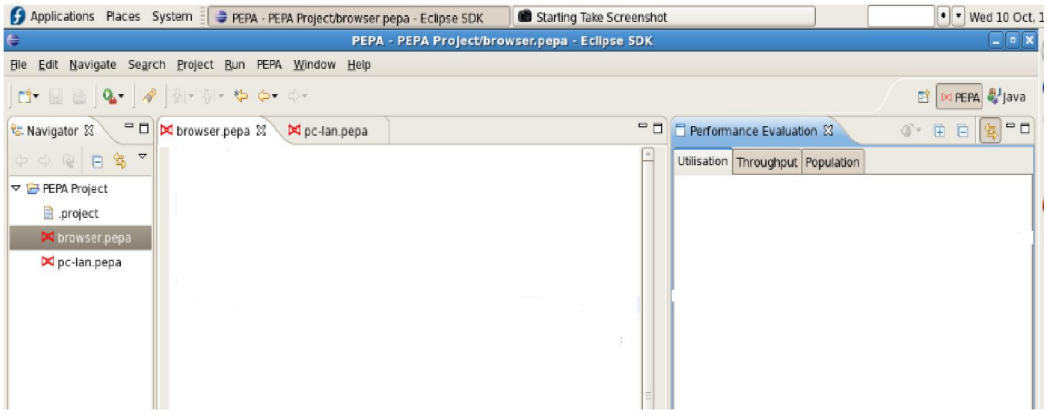

Fig. 3) An image from PEPA Workbench software

Table 2) results of effectiveness evaluations in a multimedia system

Effectiveness parameters	*Lost packets in the relational channel of a multimedia system in time unit*					
	0	10	20	30	40	50
R_display (display/operational power rate)	*29.089*	*28.194*	*26.865*	*24.762*	*22.703*	*20.715*
R_error(error rate)	*10.908*	*11.899*	*11.426*	*11.887*	*12.409*	*12.937*
R_latency(display rate)	*0.182*	*0.128*	*0.096*	*0.076*	*0.063*	*0.054*
Jitter(display variation rate)	*0.0012*	*0.0013*	*0.0014*	*0.0016*	*0.0019*	*0.0023*

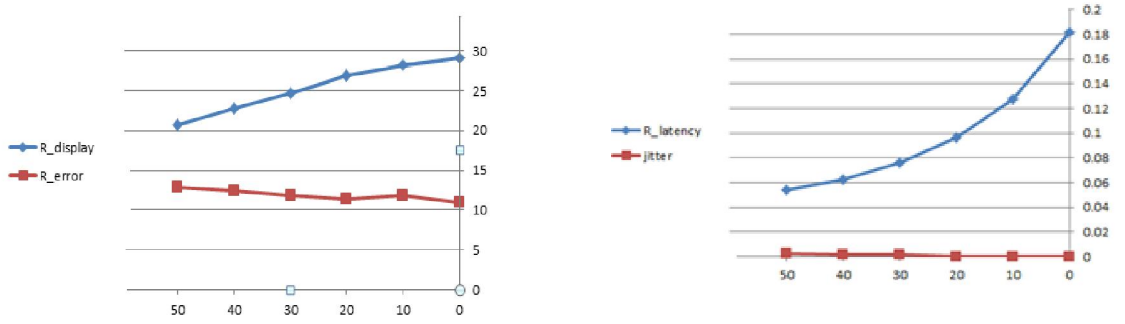

Fig. 4) The diagram based on the number of the lost packets in the channel and the actual calculated effectiveness parameters

6) Conclusions

In this article, an approach is designed for modeling effectiveness engineering based on SPA. The distinctive aspect of the suggested approach in respect with the other approaches, is describing the software with introducing the performance profile of UML into the software model. A practical example from the effectiveness evaluation methodology is presented and the results are studied. An algorithm is provided to produce SPA effectiveness model from the XML documents of the state and collaboration diagrams equipped with effectiveness profiles by using EXPORTXML which is supported by ArgoUML software. Then the resulting SPA effectiveness model is analyzed by PEPA Workbench tool.

However, by using other models of UML models, and adding more details to the modeling process, a more complete body of effectiveness sizes and other qualitative parameters of the software can be examined. This tool can also develop by adding the defined algorithm in this article to PEPA Workbench.

References

[1] L.G williums, G.U. smith "performance Evaluation of Software Architectures" proc. Of wosp'98, santa Fe, New Mexico , USA, PP, 164-177 (1998)

[2] R.Class, Software Runaway, "Lessons learned form Massive Software project Failures, prentice Hall" 1998

[3] smith C.U., Performance Engineering Of Software System, Adison—Wesley(1990)

[4] Cortllessa, V.,Mirandola, R. PRIMA-UML:A Performanc Validation Incremental Methodology on Early UML Diagram 101-129(2002)

[5] Gu, G., Petriu, D.C, XSLT tranformation from UML models to LQN performance models, In [WOSP02], PP. 227-23

[6] Arief, L.B., Speirs, N.A., A UML Tool for an Automatic Generation of Simulation Program, In [WOSP00] PP. 71-76

[7] 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.

[8] 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.

[9] Analysis of a Multimedia Stream Using SPA, H. Bowman, J.W.Brayans and a J. Derrick, University of Kent, 2001

[10] Performance Modelling with UML and Stochastic Process Algebras, Catherine Canvent, Stephen Glimore, Jane Hiliston, Matthew Prowes and Perdita Stevens,2002

[11] http://argouml.tigris.org

[12] http://www.dcs.ed.ac.uk/pepa

[13] http://www.sax.sourceforge.net

[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

6/15/2013