This paper appeared in Proc. 7th Int. Workshop on Performability Modeling of Computer and Communication Systems (PMCCS7), Torino, Italy, Sept. 2005

# An Aggregation Approach to Constructing Hybrid Layered Queueing Models

Pengfei Wu, Murray Woodside
Department of Systems and Computer Engineering
Carleton University
1125 Colonel By Drive,
Ottawa, Ontario, Canada K1S 5B6
{pfwu, cmw}@sce.carleton.ca

## Abstract

*Layered queueing network (LQN) models are effective for large systems but lack the capability to model some kinds of decision logic that may be important for system performance. This paper describes the use of a "complementary model" which focuses on the decision logic, and which provides the results of the logic in a form that the LQN can use. The complementary model is constructed from the same base information as the LQN but system elements away from the focal point are approximated in reduced detail, by aggregating them. The complementary model is thus made consistent with the LQN. The two models together form a "hybrid" multi-formalism model, which is solved by a fixed point iteration. The approach is described through an example which uses Stochastic Petri Nets for the complementary submodel.*

**Index Terms**—Hybrid Performance Models, Use Case Map (UCM), Layered Queueing Network (LQN), Generalized Stochastic Petri Net (GSPN), Fixed Point Iteration, Software Performance Engineering (SPE)

## 1. Introduction

Layered queueing network models have been successfully used to model the performance of quite large systems of a variety of types [Woodside95, Rolia95]. The model structure closely resembles that of layered service systems, such as client server systems, web applications, and web services, however it has also been applied to other architectures, including embedded software.

A limitation of queueing models is seen when logical conditions which are load-dependent govern the system behaviour. The queueing model can represent the mean outcome of the decision as an average combination of behaviours, but it has no way to determine the interaction of the decision with the performance variables. An example is a timeout which triggers alternative behaviour; the timeout depends on delays which are part of the system solution, yet the timeout probability affects those delays. Another modeling formalism, such as Markov Chains or Stochastic Petri Nets, is needed to represent these features. However these formalisms cannot handle the state space sizes that arise with large user populations.

A possible solution is found in combining formalisms, in what we call here "hybrid modeling". There are many approaches to this, some references are: [Ciardo01], [Daly00], [Pooley92], [Trivedi02]. Here we are not considering the general problem of hybrid modeling, but a particular aspect (how to create a complementary model) in a particular case. The case is illustrated by the LQN in Figure 1.

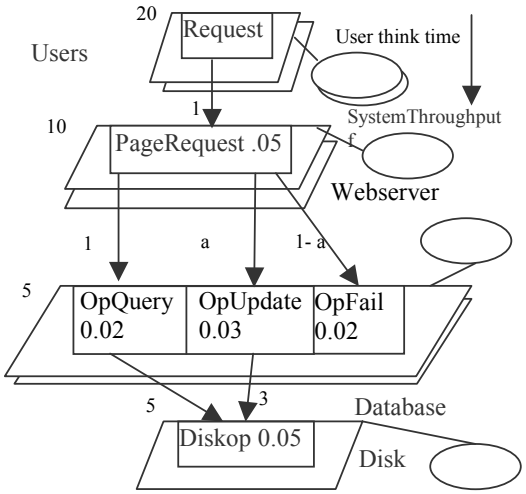


**Figure 1 LQN for a Web Database Application (WDA)**

In Figure 1 the parallelograms represent concurrent tasks, representing the users of a web server, a web server, a database, and its disk. The rectangles in each task represent "entries", which stand for functions of the tasks (Request, PageRequest, etc), and the ovals represent the processors that execute them. The arrows show requests for service from one entry to another, labelled by the mean number of requests.

The clients continuously cycle, sending requests to the web server after a "thinking time" following the previous response. The web server processes the requests, identifies them as query and update operations and forwards them to the database server. Update operations all address one element of shared data, with a write lock. Updates which fail to obtain the lock return the failure to the User. The query operation and update operation include several disk operations respectively.

The arrows labelled by *a* and *1-a* show requests to update the database. OpUpdate is a request that succeeds, OpFail fails. The goal of the complementary model will be to estimate the value of *a*, from the locking mechanism.

## 2. Complementary Model Construction by Aggregation

The process of constructing the complementary model begins where the LQN modeling begins: with a scenario for the use of the system. Figure 2 shows the scenario, represented as a Use Case Map ([Buhr96]).

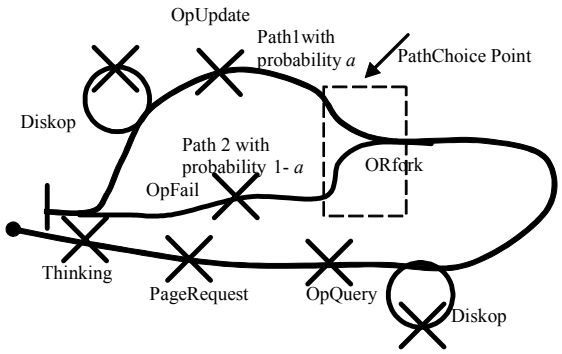


**Figure 2 Scenario for a request in the WDA**

The behaviour described by this scenario begins from the filled circle at bottom left, and follows the path to the bar at the left. Along the way are operations indicated by crosses. We assume that the workload is associated with these operations. The loops for "Diskop" indicate repeated operations, with a repetition count. Performance annotations for Use Case Maps are described in [Petriu03].

The LQN model in Figure 1 was created from this scenario, augmented by component boundaries, by a process such as is described in [Petriu02]. To create the complementary model we begin at the same point, and this is important for matching the two models consistently. Figure 3 illustrates the process of converting a scenario into an LQN and a PN.

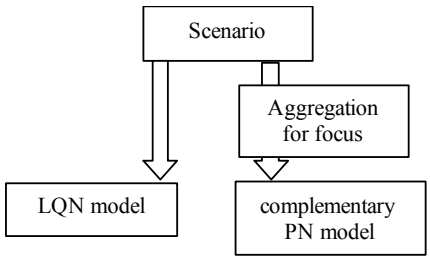


**Figure 3. The derivation of the complementary model from the scenario**

The area of focus is the choice point indicated in the dashed box in Figure 2. Here the PN model will describe in detail the lock availability conditions. The other parts are to be aggregated, and Figure 4 indicates aggregation by boxes around operations. Other aggregations are possible, but different branches of the path should be in separate aggregates.

The four aggregated segments are shown in Figure 4 as Request Aggregation, Query Aggregation, Update Aggregation and Fail Aggregation. This gives the high-level representation for the scenario of WDA in Figure 5.

A GSPN [Ajmone95] is constructed from Figure 5 by putting a place and a timed transition for each "responsibility" (cross). We also introduce some places with tokens for the users and some resources in the WDA system.

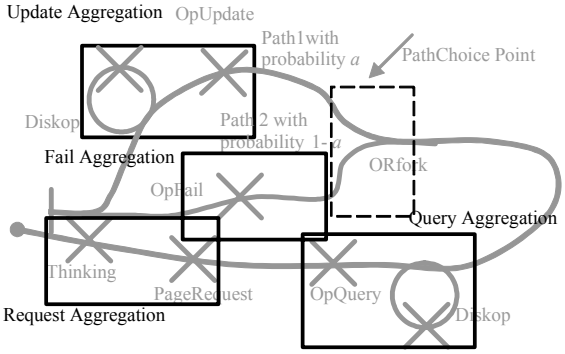


**Figure 4 WDA scenario with four aggregations**

In the GSPN of Figure 6, the place "user" holds tokens representing users in this system. The place "databaseCPU" holds tokens representing database server threads, which may limit concurrency of processing. The place "pool" holds tokens representing write locks for shared data.

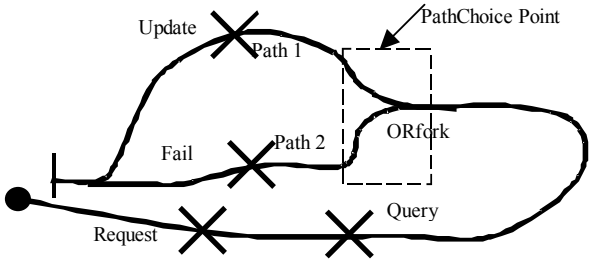


**Figure 5 High-level representation for the scenario in WDA**

The logic of granting a lock is as follows: there is one element of shared data. If the lock is available, a request succeeds; if not, it fails. The transitions "choose_fail" and "choose_succeed" represent this logic.

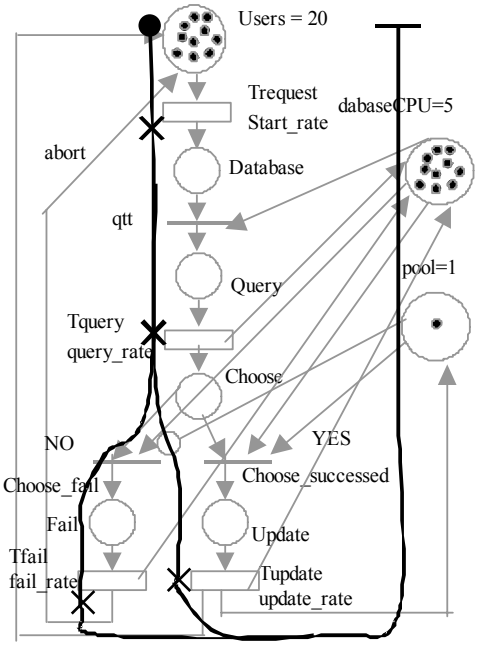


**Figure 6 Definition of GSPN from UCM**

The unknown parameters in the GSPN (the timed transition rates) still need to be defined; many of them will depend on the solution of the LQN.

## 3. Fixed-point Solution

The approach to solving the two models together is to iterate between them, using values found from one to populate missing parameters in the other. The models are solved in turn until values converge (a fixed point iteration [Ciardo91a, Mainkar96]). This is similar to the strategy for iterating for decomposed submodels described by Ciardo and Trivedi in [Ciardo91]. Figure 7 illustrates the solver structure, with separate solvers for the LQN model and the GSPN model, and a controller which links them.

In this hybrid model approach, the most challenging problem is to define the relationship and interactions between sub models. The data required to complete one submodel must be correctly identified in the other, if possible.

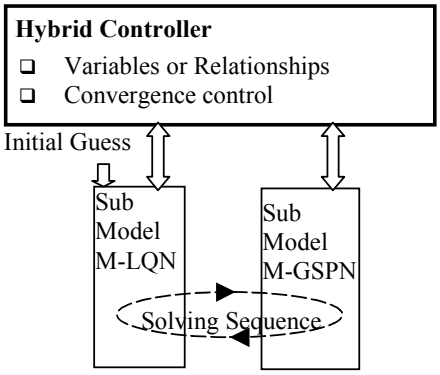


**Figure 7. Hybrid Model Approach Framework**

The present aggregation approach provides several useful properties of the combined submodels:

a. Consistency. Due to the completeness of each sub model for description of the system, the basic behaviour characteristics of system can be correctly represented in the same quantities in different formalisms. To ensure this,
   - The number of users in a closed system and the arrival rate in an open system are kept the same in different formalisms.
   - Probabilities of choosing different paths are defined to be the same in different submodels.

   Because of this consistency, throughput and cycle time results should converge to the same values in all submodels.

b. Focus. Because each formalism has its own strength in description of certain aspects of system behaviour, we can use it to capture details of those aspects. Here the LQN focus is the queues and the GSPN focus is the decision mechanism and probability.

Consistency implies that some quantities are represented in both models. In particular some parameters in one model correspond to solution quantities in the other. The corresponding quantities are listed below.

**Corresponding Quantities in WDA**

| $\mathbf{M_{LQN}}$ | $\mathbf{M_{GSPN}}$ |
|---|---|
| *a* (=success probability) | relative rate of update, compared to requests |
| time between requests | 1/rate of Trequest |
| time for a query | 1/rate of Tquery |
| time for an update | 1/rate of Tupdate |
| time for handling a failure | 1/rate of Tfail |

The correspondence is not an identity, but rather the corresponding quantities are related by equations which must be derived from the roles of the quantities in the submodel.

In WDA the relationships between corresponding quantities are as follows. Defining *f(transition)* as the throughput rate of a transition, we can show that *a* is determined by:

$$a = f(update)/[\, f(update) + f(fail) \,] \quad (1)$$

Then the rates $\mu_r$ , $\mu_q$ ,$\mu_f$ , $\mu_d$, can also be obtained from the consistency in the number of users, the throughput/cycle time, the probabilities on different paths in the WDA system, as follows. Define:

$N$ = number of users in the WDA system,
$f$ = the throughput of WDA system
*Response* is the response time seen by users,
*M(p)* is the average marking of the place *p* in $M_{GSPN}$
*S(e)* is the service time of entry *e* in $M_{LQN}$

The equations that indicate the relationships are:

$$\mu_r = M(Users) \,/\, [N/f - Response] \quad (2)$$
$$\mu_q = M(Query) \,/\, S(OpQuery) \quad (3)$$
$$\mu_d = M(Update) \,/\, S(OpUpdate) \quad (4)$$
$$\mu_f = M(Fail) \,/\, S(OpFail) \quad (5)$$

The iteration was started with an initial guess for *a*, and repeated to convergence. The models always converged.

## 4. Experiment results and analysis

The system was examined with 20 Users, 10 web server threads on web server, and 5 database threads on database server. Each request sent to web server causes one query operation and one update attempt. The query operation includes 5 disk operations and the update operation includes 3 disk operations. The execution demands of page request, query, update, fail, disk operation are 0.05, 0.02, 0.03, 0.02, 0.05 sec respectively.

To vary the intensity of the load, the thinking time was varied from 20 seconds to 2 seconds. The submodel $M_{LQN}$, was solved by LQNS [Franks00], and $M_{GSPN}$, was solved by SPNP V5.01[Ciardo98] to solve it. To evaluate the accuracy of the iterative solution, the system was also solved by simulation (CSIM18 [Schwetman86]).

The results are shown in Table 1. Direct comparisons are tabulated for the write lock holding time H and the web server service time (to process the request). The accuracy is good (errors less than 12%) at the two extremes of light and heavy load. In light load, T is 10 sec. or more, and the utilization of disk, and the bottleneck of WDA system, is less than 70%. For heavy load, T is 4 sec or less and the disk is saturated. In the middle of the range (with T around 6 to 8 seconds), the accuracy is only moderately good, with around 15% error, and the utilization of disk is between 70% and 90%.

**Table 1: Comparison results derived from Hybrid Modeling Approach versus Simulation**

| | Hybrid Modeling Approach (*f*, throughput; *a*, probability of update; *H*, lock holding time; *W*, web service time; *T*, thinking time) | | | | Simulation Results | | Comparison Results | |
|---|---|---|---|---|---|---|---|---|
| *T* (Sec) | *f* (req/sec) | *a* | *H* (Sec) | *W* (Sec) | *H* (Sec) | *W* (Sec) | Error for *H* | Error for *W* |
| 20 | 0.917473 | 0.826999 | 0.237744 | 0.618339 | 0.223404 | 0.634128 | 6.42% | -2.49% |
| 15 | 1.21605 | 0.76415 | 0.263815 | 0.66877 | 0.243168 | 0.697472 | 8.49% | -4.12% |
| 10 | 1.79277 | 0.636085 | 0.329442 | 0.790737 | 0.294635 | 0.858113 | 11.81% | -7.85% |
| 8 | 2.19789 | 0.544062 | 0.392605 | 0.906872 | 0.342109 | 1.005437 | 14.76% | -9.80% |
| 6 | 2.76337 | 0.420283 | 0.520902 | 1.19234 | 0.442254 | 1.336642 | 17.78% | -10.80% |
| 4 | 3.20979 | 0.325029 | 0.678945 | 2.13789 | 0.636157 | 2.089963 | 6.73% | 2.29% |
| 2 | 3.29419 | 0.30612 | 0.716267 | 2.75024 | 0.745952 | 2.868361 | -3.98% | -4.12% |

## 5. Conclusions

This paper demonstrates an approach to hybrid or multi-solver modeling based on analyzing scenarios. Each submodel is a complete model, with a different focus which exploits the strength of its formalism. The approach takes advantage of the complementary strengths of different modeling paradigms and enlarges the value of known and trusted models and solvers. It avoids limitations such as large state spaces in state-based paradigms and the logical control in queueing-based paradigms. Continuing work is formulating this experience into a general approach.


### Acknowledgements

This research was supported by grants from the Natural Sciences and Engineering Research council of Canada.


### References


[Ajmone95] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli and G. Franceschinis "*Modelling with Generalized Stochastic Petri Nets*" John Wiley & Sons, 1995

[Buhr96] R. J. A. Buhr, R. S. Casselman "Use Case Maps for Object-oriented Systems" Prentice Hall, 1996.

[Ciardo91] G. Ciardo and K. S. Trivedi, "*A Decomposition Approach for Stochastic Petri Net Models*" in Proc. Fourth Intl. Workshop. on Petri Nets and Performance Models, Melbourne, Australia, 1991

[Ciardo91a] G. Ciardo and K. Trivedi. Solution of Large GSPN Models. *Numerical Solution of Markov Chains*, W.J. Stewart, Ed., Marcel Dekker Publisher, New York 1991.

[Ciardo98] G. Ciardo, J. K. Muppala, and K. S. Trivedi. "*SPNP Users Manual, Ver. 5.01*" Technical report, Duke University, Durham, NC, 1998

[Ciardo01] G. Ciardo, R. L. Jones, A. S. Miner, and R. Siminiceanu. SMART: Stochastic Model Analyzer for Reliability and Timing In *Tools of Aachen 2001 International Multiconference on Measurement, Modelling and Evaluation of Computer-Communication Systems*, Aachen, Germany, 2001

[Daly00] D. Daly, D. Deavours, J. M. Doyle, P. G. Webster, W. H. Sanders, "*Mobius: An Extensible Tool for Performance and Dependability Modeling*" in Computer Performance Evaluation: Modelling Techniques and Tools: Proceedings of the 11th International Conference, TOOLS 2000, Schaumberg, IL, 2000

[Franks00] G. Franks, "*Performance Analysis of Distributed Server Systems*", Report OCIEE-00-01, Ph. D. Thesis, Carleton University, Ottawa, Canada, Jan 2000

[Mainkar96] V. Mainkar, K. S. Trivedi, "*Sufficient Conditions for Existence of a Fixed Point in Stochastic Reward Net-Based Iterative Models*" IEEE Transactions on Software Engineering, Volume 22 , Issue 9 1996

[Petriu02] D. Petriu, M. Woodside, "*Software Performance Models from System Scenarios in Use Case Maps*", Proc. 12 Int. Conf. on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation (Performance TOOLS 2002), London, UK, April 2002

[Petriu03] D.B. Petriu, D. Amyot, C. M. Woodside, "*Scenario-Based Performance Engineering with UCMNav*", Proc 11th Int. SDL Forum (SDL 2003), Lecture Notes in Computer Science vol LNCS 2708, pp 18 - 35

[Pooley92] R. J. Pooley, "*The Integrated Modeling Support Environment*" in Computer Performance Evaluation -- Modeling Techniques and Tools, Torino, 1992

[Rolia95] J. A. Rolia and K. C. Sevcik, "The *Method of Layers*" IEEE Trans. on Software Engineering, vol. 21, no. 8 pp. 689-700, August 1995

[Schwetman86] H.Schwetman, "*CSIM: A C-Based, Process-Oriented Simulation Language*", Winter Simulation Conference, 1986

[Trivedi02] Kishor S. Trivedi "*SHARPE 2002: Symbolic Hierarchical Automated Reliability and Performance Evaluator*". DSN 2002

[Woodside95] C. M. Woodside, J. E. Neilson, D. C. Petriu, and S. Majumdar, "*The Stochastic Rendezvous Network Model for Performance of Synchronous Client-Server-like Distributed Software*" IEEE Transactions on Computer, vol. 44, no. 1 1995