Performability Modeling: Back to the Future?

J. F. Meyer

jfm@umich.edu

Department of Electrical Engineering and Computer Science
Computer Science and Engineering Division
University of Michigan, Ann Arbor, Michigan USA

Abstract

After more than 30 years of work concerning its theory, techniques, tools, and applications, performability modeling is well understood by the many people who have been responsible for its development. During this period, other concepts have emerged which likewise aim to express how well a system performs (serves it users) under realistic operating conditions that include occurrences of both internal and external faults. The most prevalent of these are various concepts and measures of quality of service (QoS) and experience (QoE) which are "performability-like" in that they refer to aspects of both system performance (in the strict sense) and dependability. To make a more precise comparison with performability measures, it would be helpful to observe some basic properties of the latter which follow from the original modeling framework. In turn, differences revealed in this comparison could point to certain measure-formulation and model-solution problems that deserve further attention. Accordingly, the title of this talk is asking whether it's time to go back to first principles and, after doing so, suggest what needs to be done to facilitate future work on model-based QoS/QoE evaluation. Presuming a "yes" answer to this question, both avenues are followed in the talk.

I. Introduction

The concept of performability originated in the fall of 1976, when it was first documented in a status report for research sponsored by NASA's Langley Research Center [1]. Generally, an evaluation of performability can be either model-based or conducted experimentally via measurements of an actual system. As indicated in the title of this talk, we are restricting our attention to the former, i.e., the specification, construction, solution, and application of performability models. It is important to note that such models include the measure(s) by which performability is evaluated, either analytically or via simulation. Indeed, a "solution" of a model determines the values of a designated measure or, as is sometimes possible with analytic models, a closed-form expression thereof.

One might argue that the scope of performability modeling is narrower than that of model-based performability evaluation since "modeling" doesn't refer specifically to the evaluation process (model solution). Hence, we have the often used-phrase "modeling and evaluation of ... ," suggesting that evaluation is something in addition to modeling. In our view, however, an unsolved model can be likened unto a musical instrument that's never played. Accordingly, we regard model solution (measure evaluation) as being an integral part of the modeling process.

Alternatively, one might claim that performability modeling is more general than model-based performability evaluation. For example, such modeling can be used to a validate a system with respect to specified performability requirements. But the only reasonable way to accomplish this is to state such requirements in terms of one or more performability measures and then evaluate the measure(s) to determine (by comparison) whether the requirements are met. Bottom line: we regard these two terms as being synonymous.

After more than 30 years of work concerning its theory, techniques, tools, and applications, performability modeling is well understood by the many people who have been responsible for its development. During this period, other concepts have emerged which likewise aim to express how well a system performs (serves it users) under realistic operating conditions that include occurrences of both internal and external faults. The most prevalent of these are various concepts/measures of *quality of service* (QoS) and, more recently, *quality of experience* (QoE) which are "performability-like" in that they refer to aspects of both system performance (in the strict sense) and dependability. They also appear to be similar with respect to unifying lower level performance and dependability considerations. Connections between performability and QoS were first discussed during the late 80s and early 90s; see [2], [3], for example. In a later paper on this subject [4], which concerns *quality of business* (QoBiz) as well as QoE, Aad van Moorsel expressed the hope its contents would "... instigate systematic approaches to QoE and QoBiz analysis for Internet services, comparable to those existing for performability and fault-tolerant systems."

The discussion that follows has a similar aim, where we employ the following "back to the future" approach.

1) Recall the original performability modeling framework.
2) Based on 1), observe certain fundamental properties of performability measures.
3) Using 2), determine deficiencies in the current QoX definitions with regard to their formulation as performability measures.
4) Based on results of 3), identify basic performability modeling problems where further work in a QoX context could remove these deficiencies.

II. Basic concepts

Performability and its associated concepts were first published openly in [5], [6]. Some extensions to this basic framework were added shortly thereafter in papers addressing closed-form solutions [7], [8]. Summarizing the ingredients, let S denote the *total system* in question where, generally, S consists of an *object system* (the system in question that's being evaluated or analyzed) and its *environment* (externally imposed workload, external faults, etc.). Then the *performance* of S over a specified *utilization period* T is a random variable Y taking values in a set A. Elements of A are the *accomplishment levels* that might (or might not) be realized by S. T is the time period during which the system's ability to perform is being assessed. Formally, T is an interval of numbers (time instants) that is either continuous or discrete, is bounded from below (the initial instant of use), and is either bounded from above or, for systems which exhibit meaningful steady-state behavior, unbounded from above. Accordingly, the *performability* of S is the probability measure $Perf$ (denoted p_S in [5], [6]) associated with Y where, for any measurable set B of accomplishment levels ($B \subseteq A$),

$$Perf(B) = \Pr[Y \in B] = \text{ the probability that } S \text{ performs within } B.$$

A *performability measure* (for a total system S) thus consists of a performance variable Y (having codomain A and utilization period T), together with a specification of the extent to which the probabilistic nature of Y is to be described. The latter can range from a complete characterization such as the PDF of Y, to values of $Perf(B)$ for selected choices of B, down to single-number measures such as various moments of Y. Note that our use of the term "measure" refers to both Y (often called a measure or "metric" in its own right) and to the probability measure $Perf$ which expresses the probabilities of (measurable) subsets of accomplishment levels.

Solutions of such measures are based on an underlying *base model* of S i.e., a discrete-state stochastic process $X = \{X_t \mid t \in I\}$, where the index (time) set I must include the utilization period T associated with the variable Y. Thus X may be continuous-time or discrete-time, depending on the nature of the system being modeled. (In the original formulation of this

framework, X was restricted to the utilization period T; however, as several colleagues were kind enough to suggest, T is a user-oriented, rather than system-oriented, consideration and therefore should not so constrain one's perception of total system behavior.) For any $t \in I$, the value of the random variable X_t is the state of the total system S at time t. When generally interpreted, X represents simultaneous variations, as a function of time $t \in I$, in the object system's structure, its internal state, and its environment state. Obviously, X must be also be detailed enough to support solution of the specified performability measure(s).

Finally, a *performability model* consists of one or more performability measures, a base model X and, for each measure, a means of determining values of Y as function state trajectories of X (restricted to T), thus guaranteeing support of measure solution.

III. Properties of a performability measure

When comparing performability and with similar concepts such as quality of service, it is helpful to distinguish *what* property of a system is being measured from *how* values of the measure are formulated. Typically, the name given to a measure (or metric) suggests the meaning of "what," although not necessarily. Generally, "what" is an interpretation of the values of a measure, and is therefore a semantic issue that must dealt with carefully in specific applications. On the other hand, "how" is described mainly by syntactical aspects of both the measure and the system model it's based on. These include

- how values of the measure are formulated in terms of the system model
- the syntactic nature of the values (single numbers, vectors, etc.), and
- if the system model is probabilistic, how probabilities of measure-values are determined.

In the case of a performability measure, "what" is thus the interpretation of elements in the accomplishment set A (and hence values of Y). At one extreme, these can be a continuum of real-number values representing a user's perception of service quality experienced during the period T. At the other, A could be a 2-element accomplishment set $\{0, 1\}$ which distinguishes whether a specified service is performed properly throughout T (in which case $Perf$ reduces to a reliability measure). In this sense, a performability measure is quite general.

As for the "how" of a performability measure, this entails specification/construction of a supporting base model X, formulation of values of Y in terms of trajectories of X, and specification of the extent to which performability (the probability measure $Perf$ associated with Y) is to be determined.

In view of these "what" and "how" aspects, a performability measure has the following distinguishing properties.

1) It is able to account for dynamics of system structure and behavior that affect both performance (in the strict sense) and dependability.
2) It is able to unify performance and dependability aspects by expressing accomplishment in terms of one-dimensional values (typically real numbers).
3) Values of Y can depend on what the system is and does throughout the utilization period.
4) It is a probabilistic measure.

The first is due to the semantics of a base model (see the second to last paragraph of Section II) and the fact that Y is formulated in terms of its state trajectories.

Regarding property 2), it is possible for elements of A (and thus values of Y) to be multi-dimensional, perhaps with separate performance and dependability coordinates. However, property 2) says that this need **not** be the case, i.e., true unification can be achieved by insisting on one-dimensional accomplishment values.

Property 3) is a feature of a performability measure that's often overlooked in applications. When coupled with property 2), it says that a performability measure is able to "sum up" as well as unify. This property is perhaps best exemplified in terms of a general class of performability models

(introduced in the early 1980s) that involve "reward models" [9]. For example, a continuous-time base model X can be augmented by a *reward structure* which associates reward *rates* with state occupancies and reward *impulses* with state transitions. (Generally, such rates and impulses are are expressed by real numbers; when negative, they have the interpretation of a "penalty" or a "cost".) The process X, together with the reward structure, is a *reward model*, where it is *rate-based* if there is no impulse assignment or, equivalently, every transition is assigned an impulse value of 0; *impulse-based* reward models are defined in an analogous manner. In the case of rate-based models, the reward structure is typically described by a real-valued function r defined on the states of X, where $r(q)$ is interpreted as the rate at which reward is accumulated in state q. Relative to a designated utilization period $T = [u, v]$, the *accumulated reward* during T is then "summed up" by the integral

$$Y = \int_u^v r(X_t)dt \ .$$

Since it is representative of what performability measures are capable of expressing, the PDF $F_Y(y) = \Pr[Y \leq y]$ is sometimes referred as a *performability distribution*.

Finally, property 4) holds immediately since, by definition, Y is a random variable with probability measure $Perf$.

IV. Quality of Service

Concerns about QoS originated in the context of telecommunication networks in the mid 1980s. More recently, the prospect of controlling the QoS of Web services has led to similar interests in Internet and enterprise network contexts. In what follows, we restrict our attention to QoS recommendations of the standardization branch of the International Telecommunication Union (ITU-T). We feel that these are representative of past and current thinking in the general setting of large, complex networked systems supporting both communication and computation.

The earliest recommendations in this regard were made by the CCITT (the former ITU-T) in 1985 [10]. In particular, they recognized the important fact that QoS should reflect the combined influence of factors associated with both performance (in the strict sense) and dependability. In the general definition (Recommendation G.106) and at the highest level, this combined influence is expressed as the "the collective effect of *service performances* which determine the degree of satisfaction of a user of the service." In particular, one of these service performances depends on a lower level *availability performance* which is dependability-related. Hence QoS, as so defined, satisfies property 1).

However, there were no recommendations as to how values of this "collective effect" should be formulated in terms of the lower level performances. In other words, how are these effects collected? In many applications, lowest level performances are likely obtained by direct measurements of an actual system. But this does not preclude the need to somehow formulate values of high-level QoS in terms of the directly measured values. Indeed, without a means of doing this, one is hard-pressed to regard this concept as being a measure.

Within the ITU-T, work on developing standards for QoS definitions and QoS control mechanisms has continued over the years, where the responsible Study Groups are mainly SG 2 and SG 12; see ITU-T Recommendations E:800 [11] and G:1000 [12], respectively. In particular, the generally defined "QoS tree" of Recommendation G:106 has been updated in E:800, where the main improvements are some changes in dependability-related terminology. However, neither E:800 nor G:1000 appear to show any progress regarding the issues described above. Therefore, the deficiencies with respect to performability properties 2)-4) remain.

V. Removing the deficiencies

Regarding formulation of a QoS measure, one is tempted to achieve this by first defining measures for the lower level concepts and then literally combining them, i.e., the value of a higher

level measure is simply a vector of the values supplied by measures directly below it. As argued in [2], the pitfall here is that correlations among lower level measures cannot be captured by this simple combinational approach. Performability measures, on the other hand, permit formulation with respect to one-dimensional accomplishment levels (property 2)), thereby avoiding this pitfall.

So what are the alternatives? Generally, what's called for here is an extension of known performability model specification techniques (see [13], for example) so as to accommodate very large networked systems. There are three ingredients of such a specification, namely:

S1) The semantics of the values of Y.
S2) The base model X.
S3) How S2) relates to S1) in a manner that permits the base model (after construction) to support solution of the specified measure.

Suppose now that Y is a QoS measure of the type described in the previous section. Then S1) is "the degree of user satisfaction due to the collective effect of lower level service performances." Although this is somewhat vague, the semantics can be sharpened by successively interpreting lower level nodes in the ITU-T QoS tree.

Regarding S2), X must be refined enough to accommodate the leaves (item performances) of the tree. To play it safe, X should represent the actual system behavior in as much detail as practicable.

Determining S3) becomes the remaining challenge, where accomplishing this yields a QoS (or, more generally, QoX) measure that satisfies properties 2)-4) of a performability measure. Accordingly, this a fundamental problem that deserves a considerable amount of further study.

Finally, one might ask whether model-based QoS evaluation is really necessary. The answer here lies with a variety of applications such as model-based adaptation that call for accurate, fast solutions of service quality. This, in turn, calls for faster performability solution algorithms, particularly in the case of bounded utilization periods.

REFERENCES

[1] J. F. Meyer, "Models and techniques for evaluating the effectiveness of aircraft computing systems," NASA Langley Research Center, Tech. Rep. NASA-CR-149371, Nov. 1976.

[2] ——, "Performability evaluation of telecommunication networks," in *Teletraffic Science*, M. Bonatti, Ed. North-Holland, 1989, pp. 1163–1172.

[3] A. van Moorsel and B. Haverkort, "A unified performabily evaluation framework for computer and communication systems," in *Proc. 2nd Int'l Workshop on Performability Modelling of Computer and Communication Systems*, Le Mont Saint-Michel, France, June 1993.

[4] A. van Moorsel, "Metrics for the internet age: Quality of experience and quality of business," in *Proc. 5th Int'l Workshop on Performability Modelling of Computer and Communication Systems*, Erlangen, Germany, Sep. 2001.

[5] J. F. Meyer, "On evaluating the performability of degradable computing systems," in *Proc. 8th Int'l Symp. on Fault-Tolerant Computing*. Toulouse, France: IEEE Computer Society Press, June 1978, pp. 44–49.

[6] ——, "On evaluating the performability of degradable computing systems," *IEEE Trans. Computers*, vol. C-29, no. 8, pp. 720–731, August 1980.

[7] ——, "Closed-form solutions of performability," in *Proc. 11th Int'l Symp. on Fault-Tolerant Computing*. Portland, ME: IEEE Computer Society Press, June 1981, pp. 66–71.

[8] ——, "Closed-form solutions of performability," *IEEE Trans. Computers*, vol. C-31, no. 7, pp. 648–657, July 1982.

[9] R. A. Howard, *Dynamic Probabilistic Systems, Vol. II: Semi-Markov and Decision Processes*. Wiley, 1971.

[10] *CCITT Red Book, Fasc. III.1: General Characteristics of International Telephone Connections and Circuits*, Int'l Telecommunication Union, Geneva, Switzerland, 1985.

[11] *ITU-T recommendation E:800: Terms and definitions related to quality of service and network performance including dependablity*, Int'l Telecommunication Union, Geneva, Switzerland, Aug. 1994.

[12] *ITU-T recommendation G:1000: Communications quality of service: A framework and definitions*, Int'l Telecommunication Union, Geneva, Switzerland, Nov. 2001.

[13] J. F. Meyer and W. H. Sanders, "Specification and construction of performability models," in *Proc. 2nd Int'l Workshop on Performability Modelling of Computer and Communication Systems*, Le Mont Saint-Michel, France, June 1993.