Measures of System Quality

John F. Meyer
jfm@umich.edu

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

This is a brief commentary on how measures of computer and communication system quality have evolved over the past 50 years. We take "quality" to be a generic term having various interpretations, the evolution of such being the central theme of the remarks that follow. The systems in question range from hardware and software components to global networks such as the Internet. We restrict our attention to measures that are probabilistic in nature, where their evaluation is based on system models (analytic, simulation, hybrid), actual systems, or some combination thereof. For the most part, however, our comments regarding evaluation are confined to modeling concerns.

More precisely, let (S, E) denote the total system in question, consisting of an *object system* S and its *use environment* E. What S is or does in E can then be quantified via one or more quality measures. Formally, such a measure can be viewed as a random variable Y_T, where T is the period during which the system is utilized or observed (ranging from a single instant $T = \{t\}$ to an unbounded interval). Y_T takes values in a set of outcomes (having some designated interpretation) and is quantified by its probabilistic nature. The latter can range from its mean value $\mathrm{E}[Y_T]$ to a complete description in the form of its probability density function (pdf, if it exits) or its probability distribution function (PDF). Although this abstraction appears to be specific to analytic models, it applies as well to both simulation models and actual systems. In these cases, one obtains estimates of measure values for Y_T, e.g., estimates of $\mathrm{E}[Y_T]$, higher order moments, and probabilities of the form $P[Y_T \leq y]$ or $P[Y_T = y]$ (if Y_T is discrete).

Early evaluations of computer and communication systems were principally concerned with two types of quality measures:

1) Reliability: What a system is, i.e., measures of the structural integrity of S in the presence of faults (independent of E).
2) Performance: What a system does, i.e., measures of the effectiveness or efficiency of S in E, assuming both are fault-free.

In particular, traditional structure-based measures of system reliability conveyed a binary-valued view of a system's ability to serve its users:

- *operational* or *up*, meaning "able to serve";
- *non-operational* or *down* otherwise.

Typically, however, this dichotomy does not coincide with what is experienced by a user in E (either a human or some other system). In particular, if S goes down when no one is using it, does it fail?

Consequently, appropriateness of this up-down, user-independent view of system reliability began to be questioned in the early 1970s. This was due to developments in several areas including degradable computing systems, studies examining the effects of workload on hardware reliability, software reliability, and software fault tolerance. Software issues and, more generally, concerns with design faults were perhaps the most influential. In this case, S is faulty prior to use but, depending on both S and E, it may remain quite useful (until it's too late to fix).

The above precipitated more generally defined quality concepts and measures that emerged between the mid-70s and mid-80s. These placed greater emphasis on how delivered services are affected by internal and external faults. In particular, the following three somewhat related notions of system quality have received considerable attention over the past 25 years (these are defined informally; see the indicated references for more precise definitions).

Dependability: Ability to depend on S to the extent that services are delivered correctly [1].

Performability: Ability of S to perform (e.g., serve users in E) throughout a specified utilization period T (unification of performance and dependability) [2].

Quality of Service (QoS): The collective effect of service performances (including dependability) which determine the degree of satisfaction of a user of the service [3].

A distinguishing feature of the concept of dependability is its treatment of the notion of "failure." Instead of it being a loss of capacity to serve (per traditional measures of reliability and availability), a (service) *failure* is identified with a *transition from correct to incorrect service delivery*. Dissemination of this view during the 1980s and early 90s produced a major change in how various dependability attributes (particularly reliability and availability) were evaluated.

The notion of performability originated as a particular type of probability measure whose properties called for new methods of model construction and solution. It has since taken on the more general meaning given above, resulting in the development and application of a wide variety of techniques and tools for model-based performability evaluation; see [4]-[5], for example.

QoS (per the ITU-T definition given above) appears to refer to quality measures that are more subjective than their dependability and performability counterparts. In reality, however, this has not been the case. Consequently, ITU-T recommendations for more explicitly subjective concepts of quality have emerged during the past decade.

> *Quality of experience* (QoE): The overall acceptability of an application or service, as perceived subjectively by the end-user [6].
>
> *Quality of perception* (QoP): End-user perception (as in QoE) along with an understanding and assimilation of what is perceived.

(QoE and QoP are sometimes referred to as QoSE and QoSP.) Several methods of QoE/P evaluation have been standardized for different media types and applications. In the context of voice and video services, this has come to be known as *subjective quality assessment* (SQA). Generally, SQA is accomplished using a panel of human observers who, following specified rules under controlled experimental conditions, assign numerical quality values to what is observed. Unfortunately, such subjective testing is usually quite time consuming and requires a large number of users to provide reliable results. Moreover, controlled testing conditions often exclude fault effects that can alter what a user perceives. So a question that naturally arises is whether model-based evaluation of QoE/P is feasible and, if so, whether it can produce desired results more efficiently and economically.

Just as quality measures have migrated from low-level concerns with the structure/behavior of *S* to subjective assessments of services delivered to *E*, experimental data that supports model-based measures must likewise move upward and outward. Generally, obtaining evaluation results for even the simplest models (analytic or simulation) requires data from real-world measurements to determine values of underlying model parameters. More complex models typically need additional experimental data, e.g., values for reward model rates and impulses.

This suggests that, in addition to usual data requirements, QoE/P models need to rely on information derived from SQAs. More precisely, what's called for is a means of determining (or at least approximating) how values of objective quality variables map to the perceived-quality values of an SQA. This should be obtainable in some manner from the results of an SQA experiment. Alternatively, a methodology such as Pseudo-SQA (PSQA) can approximate this mapping using an automated learning tool; see [7], for example.

In view of the above, the following general approach to QoE/P modeling appears to be promising (our use of "SQA" here includes special methods such as PSQA).

1) Express a QoE/P measure Y_T in terms of subjective quality assessment values (Y_T is some function of the possible output values of an SQA).
2) Relative to Y_T (or a set of such measures), specify and construct a model of (*S*, *E*) that represents the dynamics of the input variables of the SQA, including how they are affected by faults in *S* and *E*.
3) Formulate Y_T in terms of 2) according to the interpretation given by 1) and the mapping obtained from the SQA.

Y_T can then be evaluated by solving (executing) the model.

Regarding 1), values of Y_T can have a variety interpretations relating to experienced or perceived quality, e.g.,

- Y_T = accumulated subjective quality experienced during *T* (subjective performability),
- Y_T = average subjective quality during *T*.

Moreover, values in the codomain of Y_T need not be quality levels or rates, per se. For example, Y_T can specialize to a dependability measure such as interval availability, i.e.,

- Y_T = the fraction of the utilization period *T* during which subjective quality is at or above some acceptable numerical value.

Conceptually, the combination of 2) and 3) is a translation of low-level dynamics of the (*S*, *E*) model into values of the QoE/P measure Y_T . This requires a thorough understanding of the associated SQA as well as the usual system and environment knowledge that accompanies dependability and performability modeling.

Results in this direction are beginning to surface, e.g., recent PSQA applications such as the one described in [7]. As more elegant forms of media content become commonplace (e.g., 22.2-channel audio, ultra-high definition television, and 3D television), problems associated with subjective measure formulation and evaluation will doubtlessly become even more interesting and challenging.

References

[1] J.-C. Laprie (Ed.), *Dependability: Basic Concepts and Terminology*, in Dependable Computing and Fault Tolerance, vol. 5, Springer-Verlag, 1992.
[2] J. F. Meyer, "On evaluating the performability of degradable computing systems," in *Proc. 8th Int'l Symp. on Fault-Tolerant Computing*, Toulouse, France, pp. 44–49, June 1978.
[3] CCITT Recommendation G.106, Red Book, Fasc. III.1, Connections and Circuits, ITU, Geneva, 1985 (currently ITU-T Recommendation E.800, Sept. 2008).
[4] A. T. Tai, J. F. Meyer, and A. Avizienis, *Software Performability: From Concepts to Applications*, Kluwer Academic Publishers, 1996.
[5] B. R. Haverkort, R. Marie, G. Rubino, and K. Trivedi (Eds.), *Performability Modelling: Techniques and Tools,* Wiley, 2001.
[6] ITU-T Study Group 12, Geneva, Jan. 2007.
[7] A. P. Couto da Silva, P. Rodriguez-Bocca, and G. Rubino, "Coupling QoE with dependability through models with failures," in *Proc. 8th Int'l Workshop on Performability Modeling of Computer and Communication Systems* (PMCCS-8), CTIT, Edinburgh, Scotland, Sept. 2007.