# Model-Based Evaluation of User-Perceived System Quality

**John F. Meyer**
jfm@umich.edu

WG 10.4 Workshop
Cortina d'Ampezzo, Italy
January 28, 2009

# Outline

- Model-based evaluation in a nut-shell
- Evolution of measure types
  - In the beginning …
  - *X*ability measures: Dependability, performability, survivability
  - Quality of *X* measures: QoS, QoE, QoP
- Accounting for the use environment
  - User-observed failures, MTTF
  - Observed reliability, availability
- User satisfaction
  - Objective measures: Performability, QoS
  - Subjective measures: QoE, QoP
- Model-based evaluation of QoE/QoP

# Model-based evaluation

- Let $S$ denote the (total) system in question, consisting of an *object system* and its *environment*.
- A probabilistic model of $S$ can then be viewed as a stochastic process $X = \{ X_t \mid t \in I \}$ with *state space* $Q$, where
    - $I$ is the *time domain* of the model
    - $Q$ can be taken to be the Cartesian product $Q_C \times Q_E$ , where $Q_C$ and $Q_E$ are the state spaces of the object system and environment, respectively.
- Just how the underlying process $X$ is
    - specified, e.g., by a SAN, a process algebra, etc.
    - and constructed, e.g., by some modeling tool

is important but is not an issue with respect to the discussion that follows.

# Model-based measures

- What $S$ is or does over some designated period $T$ is then represented by one or more measures
  - system-oriented (e.g., resource availability, resource utilization, fault-recovery time, …)
  - user-oriented (e.g., throughput, end-to-end delay, user-perceived quality, …)
- More precisely, such a measure can be generally viewed as a random variable $Y_T$, where
  - $T$ ($T \subseteq I$) is the period during which the system is utilized or observed (could be a single instant $\{t\}$)
  - $Y_T$ takes values in a designated set $A$ of possible outcomes ("accomplishment levels" in performability jargon)
    - $A$ can be binary-valued at one extreme
    - A continuum of values at the other, e.g., the extent to which a user is satisfied with services delivered throughout $T$

# Measure solution

- The underlying stochastic process $X$ needs to be detailed enough to support solution of the probabilistic nature of $Y_T$ ,e.g.,
  - its expected value
  - higher order moments
  - pdf or PDF
- Just how such measures are solved is an important and challenging problem.
- However, the remarks that follow refer mainly to the nature of various measure types in the context of computer and communication system evaluation.

# A little history

- Early model-based evaluations of computer and communication systems were principally concerned with two types of measures.
- Reliability: What a system is
  - Measures of a system’s structural integrity in the presence of faults (independent of how it is used).
  - Includes measures such as availability
- Performance: What a system does
  - Measures of a system’s effectiveness and efficiency in the absence of faults.
- System models supporting evaluations thereof had the following characteristics.

# Model assumptions

- Reliability models (physical faults)
  - Structure is probabilistic ($Q_C$ consists of structure states only)
    - Dynamics are due in part to
      - rates of fault occurrences
      - durations of recovery actions
  - Use environment is fixed ($Q_E$ has a single state representing a heavy workload with constant utilization)
- Performance models
  - Structure is fixed ($Q_C$ consists of internal states only)
  - Use environment is probabilistic
    - Dynamics are due in part to
      - frequencies and durations of user demands (service requests)
      - workload imposed during active use

# Measure implications

- Traditional structure-based measures of system reliability therefore conveyed a binary-valued view of a system's capacity to serve its users:
  - Operational or up, meaning "able to serve"
  - Otherwise the system is non-operational or down
- In most cases, however, this dichotomy does not coincide with what is experienced by a user.
- The difference can be illustrated in terms of something argued long ago by philosophers.

# A dilemma

- Consider the age-old question:
  - If a tree falls in the forest and nobody is there to hear it, does it make a sound?
- Analogously,
  - If a system goes down and nobody is using it, does it fail?
- Just as the answer to the first question depends on the definition of sound, the answer here depends on the meaning of fail.
  - If it refers to the loss of a capacity to serve: Yes
  - If it refers to an improperly delivered service: No

# Xability measures

- Accordingly, more general types of measures began to emerge in the mid-70s, placing greater emphasis on how delivered services are affected by internal and external faults.
- *X*ability measures
    - Dependability: Measures of a system's trustworthiness with respect to delivery of a specified service.
    - Performability: Measures of a system's ability to perform (serve its users) throughout a specified utilization period.
    - Survivability: Measures of a system's ability to fulfill a specified mission in a timely manner.

# Quality of *X* measures

- Quality of *X* measures are likewise concerned with how well services are provided to users.
- These have evolved mainly in the context of telecom, internet, and wireless services.
  - Quality of service (QoS): Objective measures of delivered service.
  - Quality of experience (QoE): QoS as subjectively perceived by end users (end-to-end QoS).
  - Quality of Perception (QoP): Encompasses not only QoE but also end-user ability to analyze and assimilate information provided by the service.
  - Quality of Protection (QoP): Measures of user confidence in system security.
- Why this increased concern with user satisfaction?

# Answer: User-centric applications

- Personal computing
  - Desktops, notebooks, netbooks, …
  - Hand-helds, wrist-tops, …
- Management and control
  - Embedded computers in home appliances, entertainment systems, cars, trains, aircraft, …
  - Home networks, enterprise networks, ATC systems, military C2 systems, …
- Ubiquitous computing
  - Small, inexpensive, robust networked processing devices
  - Distributed at all scales throughout everyday life
- World-wide communication and information sharing

# The use environment

- When evaluating systems from a user’s perspective, two important aspects of the use environment need to be considered:
  - Input: Dynamics of user demands for service
  - Output:  Quality of delivered services
- Both are addressed in the remarks that follow.
- User demands are typically objective in nature, permitting a discussion of their influence in fundamental analytic terms.
- Service quality, on the other hand, can be subjective as well as objective, calling for some higher level considerations.

# Use profile

- To illustrate how user demands influence the basic dependability notions of failure, MTTF, and availability, assume the following.
- A user requests services of a system intermittently, represented by a *use profile* that alternates between time periods of *active use* and *passive use*.
- Generally, these periods have random durations.
- However deterministic profiles, e.g., a predetermined schedule of fixed periods of active use, are included as special cases.

# Use profile (cont'd)

- During active use, the external workload is that associated with a particular service request
    - Active use begins when that service is requested.
    - It ends when delivery of that service is completed.
- During passive use,  a system need not be totally inert (power off) for example, it may occupy itself  with self-imposed tasks aimed at maintaining its readiness.

# Active utilization

- Letting
  - $D_p$ = mean duration of passive use
  - $D_a$ = mean duration of active use

  the mean time $D$ between service requests is the sum

$$D = D_p + D_a \, .$$

- To quantify the influence of user demands, we then define the *active utilization* of a system to be the fraction

$$\rho_a = D_a / D$$

# Special cases

$$\rho_a = D_a / D = D_a / (D_p + D_a)$$

- In case $D_p = 0$ (no passive use) active use is *constant* and $\rho_a = 1$.
- At the other extreme, if $D_a = 0$ then $\rho_a = 0$ and the system is never actively used. In this case, user observed failures cannot occur.

# Workload

- The specific nature of the workload during active use obviously depends on what service is being requested, e.g.,
  - System: Web server
  - Service Demand: A hit followed by a sequence of web transactions (e.g., GET an HTTP document, etc.).
  - Workload: Processing needed to serve the sequence of transactions.
- If service is lost during a period of active use, it is assumed that failure occurs at the time of that loss.
- Similarly, if required capacity to serve is lacking when service is demanded, failure occurs at the time of that demand.

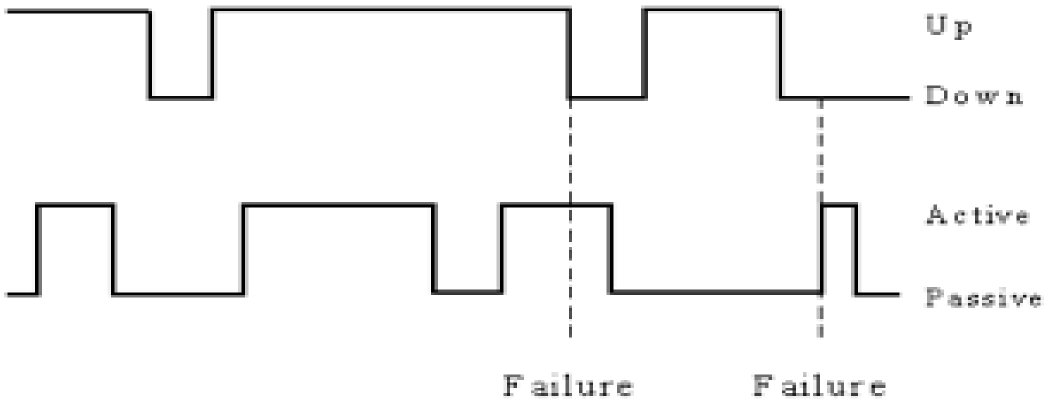
# A simple example – Meaningful relationships

- Consider a system where
  - faults arrive as Poisson process with rate $\lambda$
  - time to recover from a fault is exponentially distributed with mean recovery time $1/\nu$
- When use is active, the workload is sufficiently high to cause a (user-observed) failure when a fault occurs (the system goes down)
- The demand profile consists of alternating exponentially distributed periods of passive and active use, where
  - $D_p = 1/\alpha$ : mean duration of a passive period
  - $D_a = 1/\mu$ : mean duration of an active period
- Accordingly, its active utilization $\rho_a$ can be expressed as

$$\rho_a = \alpha /(\alpha + \mu)$$

# Markov model

- System capacity and use profile:

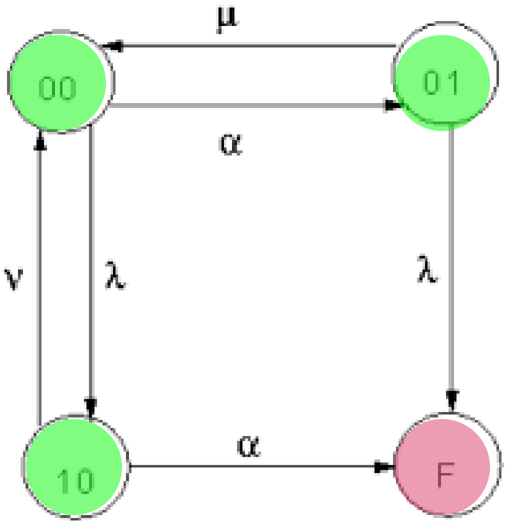


- CTMC for determining time to (observed) failure

State: $(x, y)$ = (system, use)

$x$ = $0$: up; $y$ = $0$: passive
$1$: down $1$: active

Initial state = $00$ with prob. $1$



# Mean time to failure

- Let
  - $p_F(t)$ = probability of occupying state F at time $t$
  - MTTF = mean time to (user-observed) failure
- Then with a little work

$$\boxed{\mathrm{MTTF} = \int_0^\infty t\,dp_F(t) = \frac{1}{\lambda}\left[\frac{(\alpha+\lambda+\mu)(\alpha+\lambda+\nu)-\alpha\lambda}{\alpha(\alpha+\lambda+\mu+\nu)}\right]}$$

- If use is constant ($1/\alpha = 0$) then taking the limit of the above as $\alpha \to \infty$, we have the familiar

$$\boxed{\mathrm{MTTF} = 1/\lambda}$$

- Indeed, constant use (with sufficiently high workload) is a tacit assumption that underlies much of traditional reliability and availability analysis.

# Derived parameters

- To better understand the meaning and implications of this MTTF formula, it is helpful to express it in terms of the following derived parameters.
  1 $\rho_a = \alpha/(\alpha + \mu)$, the *active utilization* (defined earlier)
  2 $\gamma = \lambda/(\lambda + \nu)$, the (constant-use) *unavailability*
  3 $\omega = \lambda/\alpha$, the *fault-demand ratio*
- Note that 1 depends only on parameters of the use profile and 2 depends only on parameters of the system.
- However, 3 depends on both, being the ratio of the rate $\lambda$ at which faults arrive (ignoring recovery periods) and the rate $\alpha$ at which service demands arrive (ignoring active use periods).

# Normalized MTTF

- Normalizing MTTF with respect to its constant-use value $1/\lambda$, i.e., $\mathrm{MTTF}_n = \mathrm{MTTF}/(1/\lambda) = \lambda \cdot \mathrm{MTTF}$

  then

$$\boxed{\mathrm{MTTF_n} = 1 + \omega\left[1 + \frac{1-(\rho_a+\gamma)}{\omega\rho_a+\gamma}\right]}$$

  - Since $0 \le \rho_a \le 1$ and $\omega \ge 0$, the term in brackets is never negative.
  - Hence, for all possible values of the involved parameters $\boxed{\mathrm{MTTF}_n \ge 1}$

- In other words, the average amount of time to a user-observed failure is never worse than when use is constant (the tacit assumption of structure-based analysis).

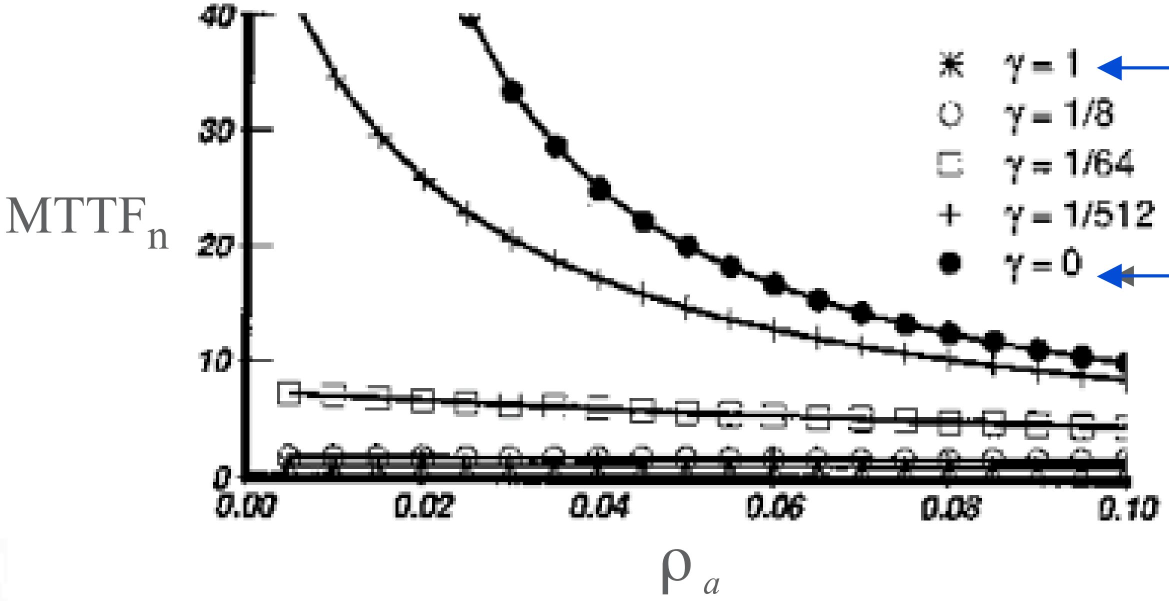
# Some further observations

$$\mathrm{MTTF_n} = 1 + \omega \left[ 1 + \frac{1 - (\rho_a + \gamma)}{\omega \rho_a + \gamma} \right]$$

- $\mathrm{MTTF_n}$ increases (improves) with
  - decreasing active utilization $\rho_a$
  - decreasing (constant-use) unavailability $\gamma$
  - increasing fault-demand ratio $\omega = \lambda/\alpha$
- In the worst case $\omega = 0$, saying that $\mathrm{MTTF_n} = 1$ or, equivalently, $\mathrm{MTTF} = 1/\lambda$.
- Assuming that faults occur ($\lambda > 0$), the mean duration $1/\alpha$ of a passive period must then be $0$, implying that use is constant.
- Hence, constant use is both necessary as well as sufficient (observed earlier) for observed $\mathrm{MTTF}$ to specialize to the usual capacity-based notion.

# Infrequent use

- Even with small (positive) values of the fault-demand ratio $\omega$, $\mathrm{MTTF}_n$ can be appreciable if both active utilization ($\rho_a$) and unavailability ($\gamma$) are small.
- For example, if $\omega = 0.1$ then, for various values of unavailability, the following plots illustrate how $\mathrm{MTTF}_n$ varies as a function of small values of active utilization.



No repair ($\nu = 0$)

Instantaneous repair ($\nu \to \infty$)

# Observed reliability and availability

- Similarly, one can obtain closed-form expressions (in terms of system/use parameters) of
  - observed reliability and availability
  - for various assumptions regarding the use profile.
- A couple of recent (21st century) studies in this regard:
  - K. W. Lee, “Stochastic models for random-request availability,” *IEEE Trans. Reliability*, March 2000, pp. 80-84.
  - D. Wang and K. S. Trivedi, “Modeling user-perceived service availability,” in *Proceedings of the 2nd ISAS*, April 2005, pp. 107-122.

# Application-related studies

- Generally, definitions of and models for observed availability tend to be service-specific.
  - Telecom voice/video
  - Web services
    - Web based travel agency – LAAS paper, DSN 2003
- In particular, model specification requires detailed information regarding the nature of service demands.
- Issues arising in this context have been challenging enough to warrant a forum and a symposium dedicated to this topic:
  - Service Availability Forum (SAF; www.saforum.org)
  - International Service Availability Symposium (ISAS),
- ISASs have been held annually since 2004
  - ISAS 2005, Berlin (Mirek, Niraj)
  - ISAS 2008, Tokyo (Takashi)
  - ISAS 2009 will be in Budapest (András Pataricza)

# The output side

- As noted earlier, two important aspects of the use environment need to be considered when evaluating systems from a user's perspective:
  - Input: Dynamics of user requests for services
  - Output: Quality of delivered services
- We now turn our attention to the output aspect, i.e., the extent to which users are satisfied with services provided by a system.

# User-observed quality by measure type

- Performance: Quality of what is delivered (usually via objective measures such at throughput, delay, jitter, etc.), presuming service is delivered correctly (system is fault-free).
- Dependability: Service quality to the extent that a service is delivered properly in the presence of faults (a failure occurs if it is not).
- Performability: Unification of performance and dependability: Quality of service delivered throughout a specified utilization period (perhaps unbounded).
- QoS (per ITU-T def.): The “collective effect” of service performances (including dependability) which determine the degree of satisfaction of a user of the service.

# Performability and QoS

- Similarities between performability and (telecom) QoS were first discussed in the late 80s and early 90s, e.g.,
  - J. F. Meyer, “Performability evaluation of telecommunication networks," in *Teletraffic Science*, M. Bonatti, Ed., North-Holland, 1989, pp. 1163--1172.
  - A. van Moorsel and B. Haverkort, “A unified performability 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.
- Differences:
  - Performability evaluation has been predominantly measure-driven and model-based since its inception.
  - QoS, on the other hand, has suffered from a lack of measure development, e.g., means of formulating the so-called “collective effect” of lower level “performances.”

# Performability & QoS (cont'd)

- Accordingly, QoS models often take the form of a performability model with service-quality semantics for the performability measure(s) $Y_T$ .
- Until recently, however, QoS has been expressed mainly in objective terms.
- Quoting a Nokia Siemens expert (Kalevi Kilkki, JUCS, Jan. ’08):
  - “In 1999, when an almost passionate effort to incorporate QoS into Internet took place at IETF (Internet Engineering Task Force), I wrote the following definition:

    > QoS is a set of attributes that can be used to define the network’s capability to meet the requirements of users and applications.

    Now I am inclined to remove users from that definition and keep QoS purely as a technical concept that is used to facilitate the interactions between applications and network services.”

# User-perceived quality

- In an effort to differentiate more subjective measures of user-perceived quality from current QoS definitions, additional Quality of *X* concepts have emerged over the past several years.
  - Quality of experience (QoE) per a recent definition by ITU-T Study Group 12 (Geneva, January 2007):
    - The overall acceptability of an application or service, as perceived subjectively by the end-user.
  - Quality of perception (QoP):
    - End-user perception (as in QoE) along with an understanding and assimilation of what is perceived.

# From QoS to QoE/QoP

- What then distinguishes QoE/QoP modeling from more usual performability/QoS modeling?
- Short answer: High level experiments and measurements.
- Generally, obtaining numerical evaluation results for even the simplest models (analytic or simulation) requires results of 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
  - obtained either directly via measurements
  - or indirectly from lower level evaluations (whose models require measurement data for their parameter values).

# Top as well as bottom

- Hence, in addition to requiring experimental data for low level parameters, QoE/QoP evaluation calls for experimentation at the top level involving actual users.
- 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 precise numerical quality values to what is observed.

# Example: SQA of video streams

- There are many ways to conduct SQAs and some have been standardized.
- Subjective video quality has perhaps received the most attention, e.g., as prescribed by
  - ITU-R recommendation BT.500-11
- Problems with SQA:
  - Very expensive in terms of time and human resources
  - Controlled conditions often exclude fault effects that can alter what a user perceives
- For example, SQAs associated with evaluating video codecs account for degradation due to compression algorithms but not for effects caused by accidental or malicious causes.

# Incorporating SQA results in QoE/QoP models

- So what needs to be done?
- Given subjective quality data, such as provided by SQA results, model-based QoE/QoP evaluation can accomplished by extending known performability/QoS modeling techniques as follows.
  - Define meaningful QoE/QoP measures $Y_T$ in terms of subjective quality assessment values.
  - Relative to a given measure $Y_T$ (or a set of such measures), specify and construct a system model that accounts for input parameters of the SQA experiments via
    - parameters of the underlying stochastic process $X$
    - reward assignments that sit on top of $X$, etc.
- Hopefully, $Y_T$ can then be formulated and evaluated (solved) in terms of the system model's stochastic behavior (a challenging problem its own right, but not new to QoE/QoP evaluation).

# Nature of the challenges

- Both are challenging problems.
- 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 $T$
  - $Y_T$ = average subjective quality during $T$
- Moreover, values in the codomain of $Y_T$ need not be quality levels or rates, per se. As with objective QoS, $Y_T$ can specialize to a dependability type measures, e.g,
  - $Y_T$ = the fraction of the utilization period $T$ during which subjective quality is at or above some of acceptable numerical value (interval availability)

# Nature of the challenges (cont'd)

- Challenge 2) calls for
  - a thorough knowledge of the associated quality assessment method
  - Innovative model specification/construction techniques that transcend what's needed for objective quality measures.
- And achieving 2) is obviously the key to successful formulation and solution of the QoE/QoP measures in question.

# Conclusion

- Since the introduction of the concepts of QoE and QoP, considerable attention has been devoted to quality assessment methods such as SQA.
- However, model-based evaluation of QoE/QoP remains in a relatively early stage of development.
- An example of recent progress in this regard is work of the Armor project in France (G. Rubino):
  - Video perception, PSQA, application to P2P systems
  - Ref: G. Rubino, et al., "Coupling QoE with dependability through models with failures," presented at *PMCCS-8*, Edinburgh, Scotland, Sept. 2007.
- The good news: **A lot of exciting work remains to be done!**

# WG 10.4's last true "Winter Meeting"

- #31 W'97 Garmisch-Partenkirchen, DE

- Great to be in Cortina 24 meetings later!!!!