Defining and Evaluating Resilience: A Performability Perspective

John F. Meyer
jfm@umich.edu

PMCCS-9
Eger, Hungary
September 17,2009

University of Michigan
Electrical Engineering and Computer Science

Outline

- Background
- Contemporary definitions of resilience
  - Safety systems
  - Ubiquitous systems
- A performability perspective
  - Extending the definition
  - Resilience evaluation
- Summary

Resilience

- The notion of system *resilience* is receiving increased attention in domains ranging from
  - safety-critical applications

  to
  - ubiquitous computing.
- When applied to computer and control systems, the term *resilient* has served as a roughly defined synonym for “fault-tolerant” since the mid-1970s.

Robustness aspect

- However, as noted last year by Laprie [1], the preface of a 1985 collection of papers edited by Anderson [2] gave it a more specific meaning by adding robustness as a key attribute, i.e.,
  - the ability of a system to deliver service under conditions that lie beyond its normal domain of operation.
- In effect, this extended usual concerns regarding tolerance of
  - anticipated faults (conditions lying within the normal domain)

  to include
  - unanticipated conditions/changes that a system may face, especially over long periods of utilization.

Application domains

- During the past decade, system resilience has received increased attention in several system domains.
- Some examples:
  - Internet
    - IRIS (Infrastructure for Resilient Internet Systems)
  - Information system technology
    - ReSIST (Resilience for Survivability in IST)
  - Safety systems
    - Resilience engineering [5]
  - Socioeconomic systems
    - Strategies for surviving change [6]; setting is a futuristic (2096 AD) government, where supporters and detractors debate the pros and cons of a proposed ``Resiliency Act."

Resilience definitions

- Contemporary definitions of system resilience differ somewhat according to the assumed nature of a system's application environment.
- A common property, however, is the ability to cope with unanticipated system and environmental conditions that might otherwise cause a loss of acceptable service (failure).

Safety-critical applications

- In a safety system context, David D. Woods has expressed the following view [5, page 21]:

> When one uses the label 'resilience,' the first reaction is to think of resilience as if it were adaptability, i.e., as the ability to absorb or adapt to disturbance, disruption and change. But all systems adapt (though sometimes these processes can be quite slow and difficult to discern) so resilience cannot simply be the adaptive capacity of a system. I want to reserve resilience to refer to the broader capability -- how well can a system handle disruptions and variations that fall outside of the base mechanisms/model for being adaptive as defined in that system.

Similarity with robustness

- Note that Woods' view is similar to the "robustness" aspect of being resilient, per the characterization in the preface of [1].
- On the other hand, it appears to exclude the handling of disruptions that fall "inside" of the adaptive design envelope, i.e., adaptivity, per se.
- Perhaps this was implicit or simply an oversight.

Distributed applications

- With respect to highly-distributed applications such as ubiquitous (pervasive) computing, the ReSIST project cited earlier has devoted considerable work to
  - defining resilience

  and
  - relating it to the notion of dependability.
- Here, the targeted systems are large, networked information infrastructures, referred to as ubiquitous systems.

ReSIST definitions

- Quoting from the Laprie reference cited earlier [1,page G-8]:

With such ubiquitous systems, what is at stake is to maintain dependability, i.e., the ability to deliver service that can justifiably be trusted in spite of continuous changes. Our definition of resilience is then:

The persistence of service delivery that can be justifiably be trusted, when facing changes.

The definition given above builds on the initial definition of dependability, which emphasizes justifiably trusted service.

ReSIST definitions (cont'd)

In a similar spirit, the alternate definition of dependability, which emphasizes the avoidance of unacceptably frequent or severe failures, could be used, leading to an alternate definition of resilience:

The persistence of the avoidance of failures that are unacceptably frequent or severe, when facing changes.

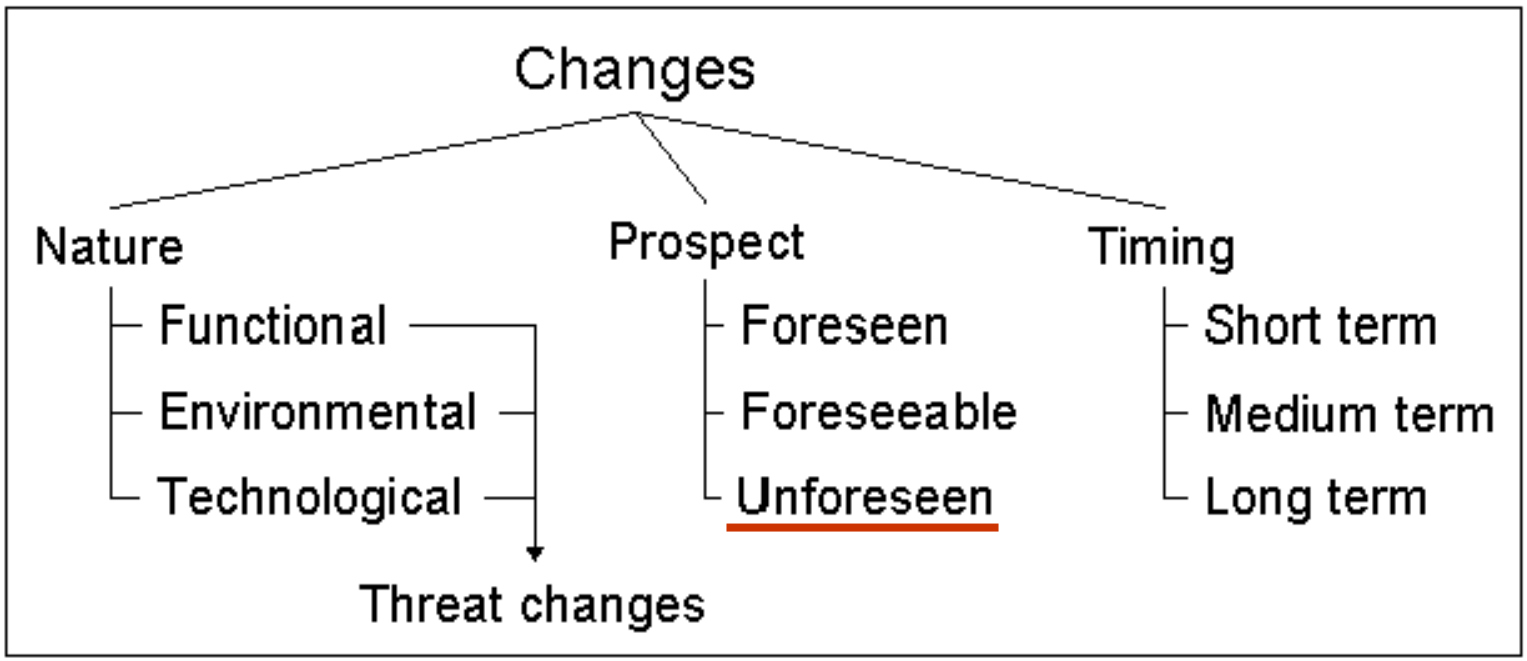
From what precedes, it appears clearly that a shorthand definition of resilience is:

The persistence of dependability when facing changes.

Changes

- Although tolerance of unanticipated changes is not explicit in the ReSIST definitions just quoted, it is nevertheless recognized when “changes” are further elaborated.
- In particular, they introduce a prospect dimension of change that includes an unforeseen category, as indicated in the following ReSIST classification of changes [1, page G-9].

ReSIST classification of changes



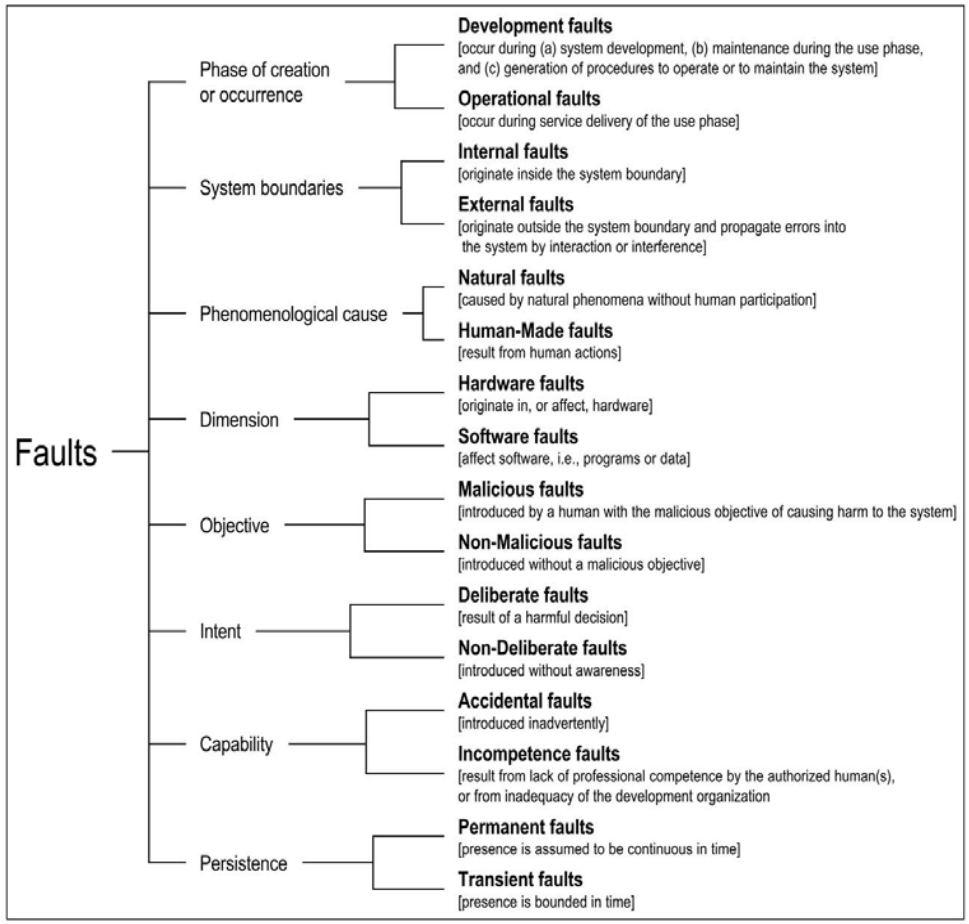
Alternative terminology

- Rather than complicate things by introducing a new term (resilience) into the dependability-related vocabulary,
  - Why not regard unanticipated (unforeseen) changes as simply another class of faults?
- Justification:
  - Concern with unanticipated phenomena in the context of fault-tolerant computing dates back 30+ years ago:
    - *IEEE Workshop on Designing for the Unexpected*, St. Thomas, Virgin Islands, Dec. 1978.
  - The “foreseen” and “foreseeable” aspects of the ReSIST change classification imply that certain changes are fault-like.
  - There is one less term to deal with in a field that’s already overly populated with taxonomies and ontologies.

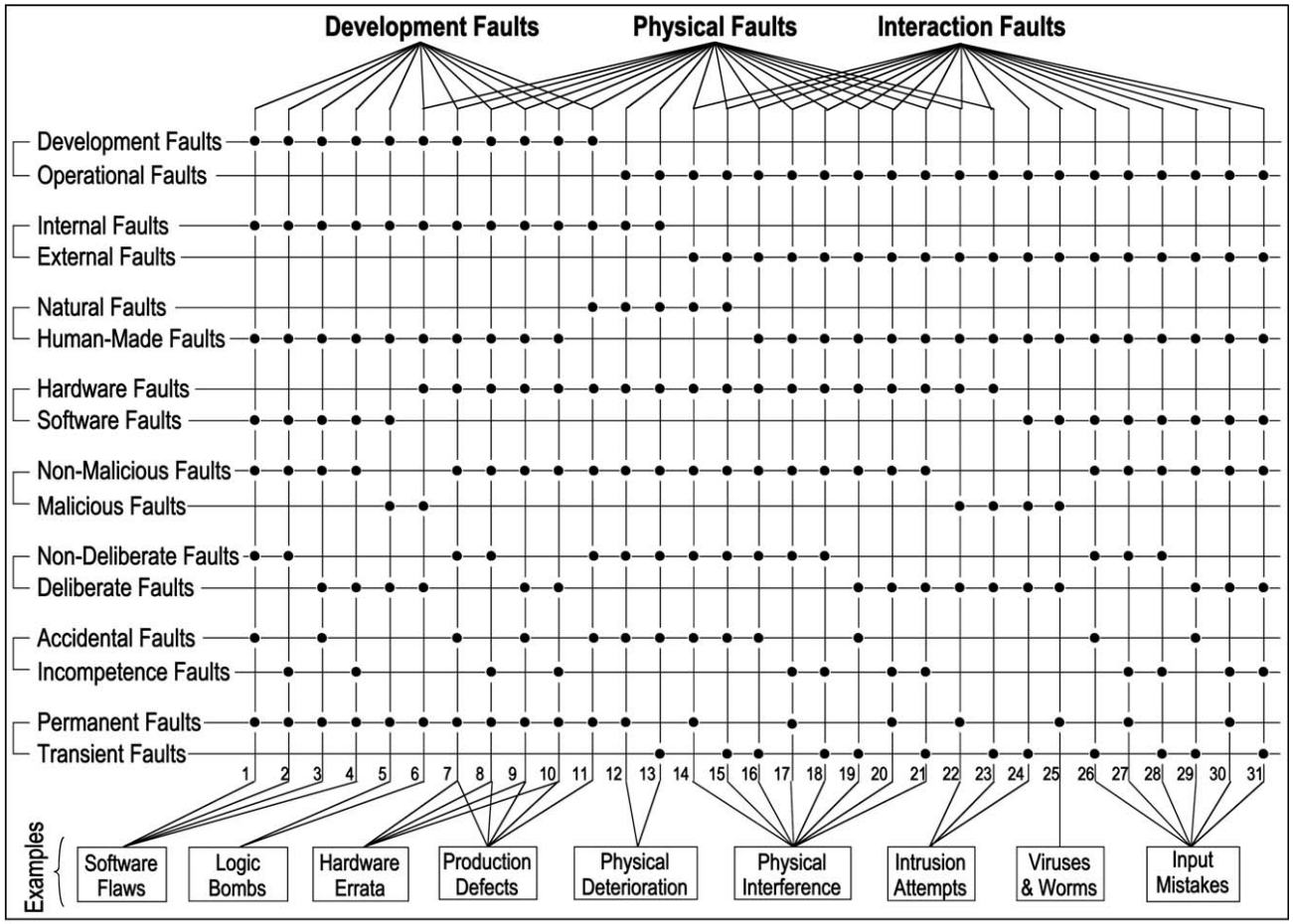
Arguments against this alternative

- On the other hand:
  1) The term “resilience” serves to signal the fact that additional kinds of change are being accounted for.
  2) Current classifications of fault types are sufficiently complicated to discourage further extension.
- Reason 1) is common to all the definitions we’ve reviewed.
- Reason 2) is illustrated by the following 3 slides, courtesy of a 2004 taxonomy of dependable and secure computing [8].

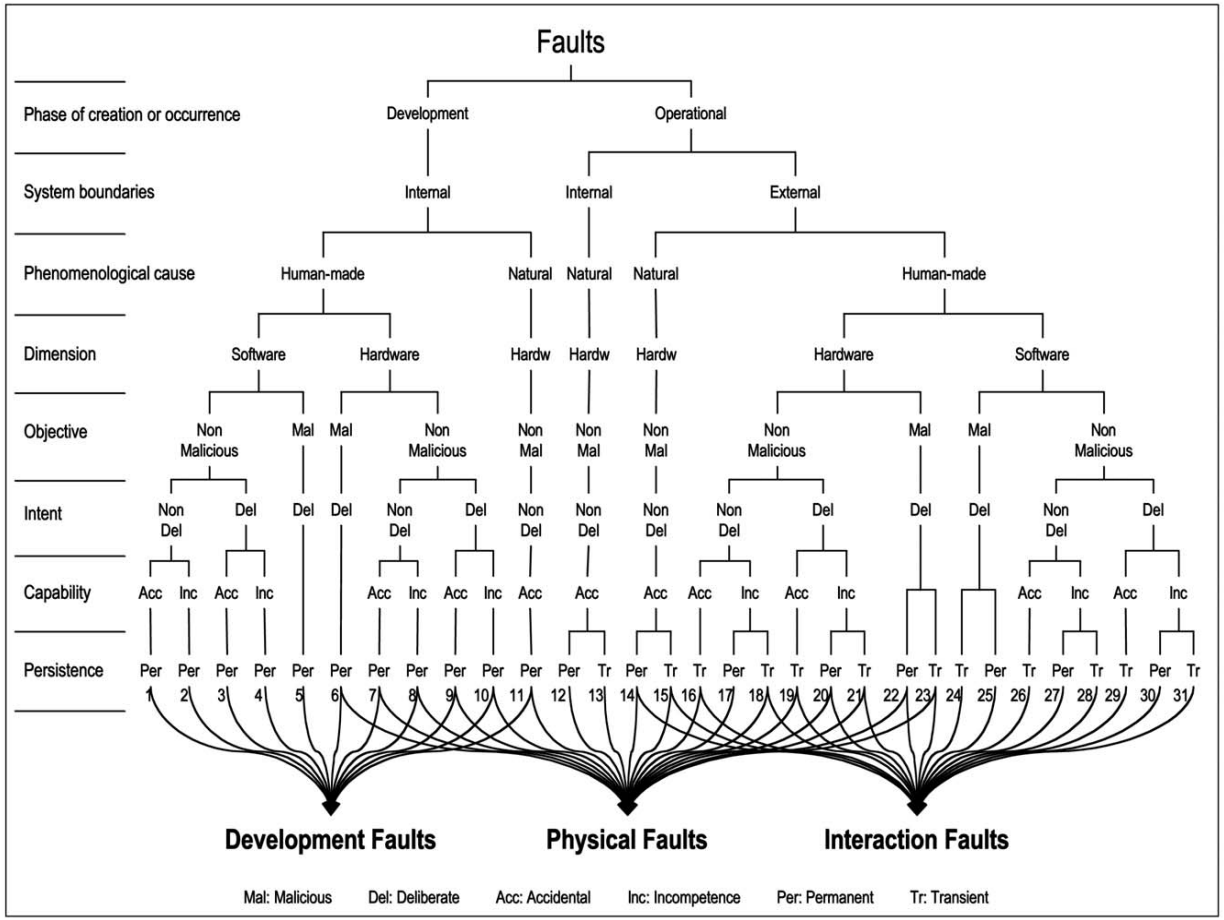
Elementary Fault Classes [8, Fig.4]



Combined fault classes: Matrix [8,Fig.5a]



Combined faults: Tree [8, Fig.5b]

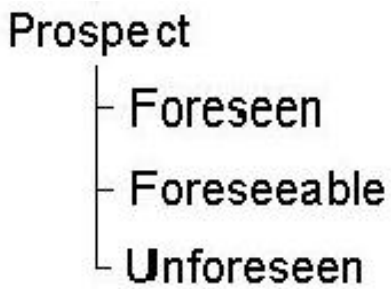


Resilience ontology

- Some minor revisions of these fault classifications are described in a ReSIST final report on a *Resilience Ontology* (deliverable D34, Dec. 2008).
- However, these revisions do not involve the notion of "change," nor does the report elaborate on its meaning.
- Hence, there's some ambiguity as to just how to this term is interpreted in the context of the ReSIST definitions of resilience.

Meaning of "change"

- Interpretation A:
  - "Change" is reserved for phenomena that lie outside of the fault classes defined in a dependable computing context.
  - Concern with faults is then implied by the term "dependability."
- Interpretation B:
  - "Change" includes "fault" as a special case.
  - This is suggested by the "prospect" dimension of the change classification:



Some comments re the two interpretations

- Interpretation A emphasizes concern with conditions whose tolerance is typically associated with the term “robust.”
- Interpretation B has the effect of adding non-fault changes to existing fault classes, where both are generally referred to as changes.
  - Note that this is similar to the alternative considered earlier.
  - In this case, however, the term “fault” is maintained for the special case, thus avoiding a conflict with past usage.
- A recent correspondence from Jean-Claude Laprie indicates that B is the preferred interpretation.
- Accordingly, B is assumed in the remarks that follow.

A common property

- A common property of all the resilience definitions discussed so far is the following.
- They are success-oriented, relying on an underlying complementary concept of *failure*.
  - In a safety context, failures are typically identified with events that incur severe damage to or losses of equipment and human lives.
  - In the terminology of dependability, a “service failure” is identified with a transition from correct to incorrect service delivery [8, Section 2.2].

Why this focus?

- In the case of safety-critical systems, this focus is perhaps justifiable due to the severe nature of failures, thus outweighing other service-related considerations.
- However, in the more general context of ubiquitous systems, it appears to be unnecessarily restrictive.
- Instead, as suggested by the *P* in *PMCCS*, this notion can be extended so as to profit from the advantages of a performability measure.

Properties of a performability measure

- It is able of account for dynamics of system structure and behavior that affect both performance (in the strict sense) and dependability.
- In particular, it can account for degradations in service quality that lie above the threshold of service failure.
- It is able to unify performance and dependability aspects by expressing accomplishment in terms of one-dimensional values (typically real numbers).
- Its values can depend on what a system is and does throughout a specified period of utilization.

A performability extension of resilience

- Just as measures of *performability* [9] generalize measures of dependability (e.g., reliability and availability), the notion of resilience can be extended in an identical manner.
- Specifically, when expressed in the form of the shorthand version of the ReSIST definition, we have:

> *Def.*: *Resilience* is the persistence of performability when facing changes.

Potential advantages of the extension

- Stated informally, a performability measure quantifies a system's "ability to perform in the presence of faults."
- Measures of resilience (as so extended) thus quantify the persistence of such ability in the presence of changes (including faults).
- Hence, this opens doors that are closed to a strict dependability interpretation.
- For example, it permits summarization of an entire history of service quality variations caused by changes that occur over a lengthy, yet bounded period of time.

Resilience evaluation

- For either definition, i.e.,
  - Persistence of x when facing changes, whether x be dependability or performability

  the important added ingredient is the persistence of such with respect to unanticipated changes.
- Just how "persistence" is defined is an issue which we'll address in a moment..
- More important, however, is the consideration of system and environment dynamics that are beyond those typically addressed in the evaluation of x.

Types of unanticipated changes (UCs)

- In particular, they include evolutionary changes in the use environment that occur more slowly over longer periods of system use.
- They also include adaptive changes in system structure and behavior that respond to environment changes and thus permit x to persist.
- Such changes pose a number of challenges, particularly in the case of model-based evaluation.

Challenges

- For example, one must seek means of
  1) accounting for these additional dynamics in the formulation of resilience models and measures, and
  2) accommodating 1) in methods of model-based resilience evaluation (resilience model solution).
- A few suggestions regarding each of these challenges are addressed in the remarks that follow.
- However, they are far from being either
  - inclusive of all that needs to be said or done, or
  - perfected to the point of being immediately applicable.

Characteristics of unanticipated changes

- The following are some physical characteristics of UCs that relate to both 1) and 2).
- Origin
  - Likely to be external.
  - Reasons:
    - Internal changes are confined to the system, per se, wherein changes are typically better understood and therefore more likely to be anticipated.
    - External UCs, on the other hand, can have global and even extraterrestrial origins (e.g., solar radiation, meteor impacts).

Characteristics of UCs (cont'd)

- Temporal nature
  - Discrete UC (DUC):
    - Has a specific time of occurrence (is an event)
    - Likely to occur infrequently.
    - Reason: A change that is observed relatively often becomes anticipated and is thus a fault according to Interpretation B.
  - Continuous (CUC):
    - Change evolves without having a perceptible occurrence time.
    - Likely to evolve slowly.
    - Reason: Rapidly evolving changes are more easily observed and again can be anticipated.

Stochastic implications

- DUCs:
  - Time between occurrences (or to the only occurrence if it's a one-off event) is much longer compared with times between fault occurrences.
  - Hence, occurrence probabilities, even during lengthy utilization periods, are extremely low.
  - Moreover, steady-state solutions are not an option unless all the UCs occur repeatedly and the utilization period is very long or unbounded.
- CUCs:
  - A continuous state space is likely required in order to represent how they evolve.

Resilience measures

- Recalling our extended definition of resilience, i.e.,

> *Resilience* is the persistence of performability when facing changes.

  there is flexibility regarding how measures of resilience are interpreted.
- For example, if “to persist” is “to exist” then a resilience measure is a performability measure that accounts for effects of UCs as well as faults.

Resilience measures (cont'd)

- More restricted interpretations of "persist" correspond to more specialized measures of resilience.
- For example, suppose "persist" has the stronger meaning of "holding on" to some acceptable level of ability to serve, e.g.,
  - stay at or above some lower bound $\boldsymbol{b}$ on

    on the mean service quality (MSQ)
- Resilience in this case is then captured by the performability measure:
  - fraction of time that MSQ $\geq \boldsymbol{b}$.

UC-tolerance mechanisms

- Unclear as to just how these will differ from fault-tolerance mechanisms.
- Many will likely involve adaptation to slowly evolving changes.
- For example, unanticipated growth in demands on a server farm can cause degradations in mean service quality that eventually become unacceptable.
- Tolerance mechanism: Servers are interconnected in a manner that facilitates on-line expansion of the server pool, thereby adapting to this CUC by increasing capacity to serve.

Model-based solutions

- **Q**: Why are solutions of resilience models likely to be more difficult than those of usual performability models?
- **A**: The need to account for the effects of UCs having properties discussed earlier.
- In particular, infrequently occurring DUCs have a time granularity that exceeds that of typical fault occurrences by several orders of magnitude.
- This suggests the following approach to decomposing and solving a resilience model.

Courtois revisited

- A popular performability modeling technique, first applied in [10], is based on Courtois' theory of "near complete decomposability" [11].
- It relies on the underlying assumption that frequently occurring events are likely to approach steady-state behavior between occurrences of changes having much larger inter-arrival times.
- So why not consider a second Courtois-like decomposition in order to accommodate DUCs?

Two-fold time-decomposition

1) Assume that occurrence frequencies are such that
   - DUCs << faults/fault recoveries
   - Fault/fault recoveries << service related events.
2) Postulate a set of system-environment states representing effects of DUCs.
3) Evaluate a "performability rate" for each such state, e.g.,
   - steady-state mean service quality

   via usual means of s-s performability evaluation.
4) Now do a second performability evaluation relative to the DUC dynamics, where reward rates are assigned according to the results obtained in 3).

Summary

- So what have we done?
- Reviewed several notions of resilience.
- Proposed a performability extension thereof.
- Put some syntactic meat on the semantic bones of “unanticipated changes.”
  - They occur very infrequently (DUCs)
  - They evolve very slowly (CUCs)
- Discussed some resilience modeling and evaluation issues.
- Proposed a 2xCourtois decomposition for model-based resilience evaluation.