Defining and Evaluating Resilience: A Performability Perspective

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*—The notion of system "resilience" is receiving increased attention in domains ranging from safety-critical applications to ubiquitous computing. After reviewing how resilience has been defined in these contexts, we discuss roles that performability can play in both its definition and evaluation.**

I. Introduction

When applied to computer and control systems, the term *resilient* has served as a roughly defined synonym for "fault-tolerant" since the mid-1970s. 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. In effect, this extended usual concerns regarding the tolerance of anticipated faults to include unanticipated conditions/changes that a system may face, especially over long periods of utilization.

During the past decade, system *resilience* has received increased attention due to research efforts in several system domains. Examples include multi-partner projects such as IRIS (Infrastructure for Resilient Internet Systems [3]) in the United States and ReSIST (Resilience for Survivability in IST [4]) in Europe. In the context of safety-critical systems, *resilience engineering* has recently emerged as a means of actively anticipating changes in risk prior to the occurrence of resulting damage (see the many papers in [5], for example). A less technical but nevertheless relevant treatment of this subject is contained in a humorous and thought-provoking book by Foster [6]. The setting in this case is a futuristic virtual government where, in the year 2096, supporters and detractors are debating the pros and cons of a proposed "Resiliency Act."

The section that follows reviews certain definitions of resilience which have been proposed in conjunction the efforts noted above. We then discuss some vital roles that performability concepts and techniques (see [7], for example) can play with regard to both defining resilience and evaluating measures thereof.

II. Resilience

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).

For example, in the context of applications where safety is the principal concern (particularly human safety, where failures can result in the loss of lives), Woods [5, page 21] has expressed the following view:

> 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.

Note that this definition is similar to the "robustness" aspect of being resilient, per the characterization in the preface of [2]. On the other hand, the above appears to exclude the handling of disruptions that fall inside of the adaptive design envelope. Perhaps this was simply an oversight.

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* [8]. Here, the targeted systems are large, networked information infrastructures, referred to simply as *ubiquitous systems*. 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 justifiably be trusted, when facing changes.
>
> The definition given above builds on the initial definition of dependability, which emphasizes justifiably trusted service. 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.

Although tolerance of unanticipated changes (as emphasized in the previous definitions) is not explicit here, it is nevertheless recognized when "changes" are further elaborated in various ReSIST documents. In particular, they introduce a "prospect" dimension of change that includes an "unforeseen" category (see [1, page G-9], for example).

There are other variations on the resilience theme that could be likewise be reviewed. However, the above should serve as adequate background for the purpose noted at the end of Section I.

III. A Performability Perspective

Let us now examine some roles that performability can play in both defining and evaluating resilience.

A. Extending the Definition

Regarding first the definition aspect, the characterizations described in the previous section are "success-oriented" in that they stress the persistence of correct service delivery in the presence of disruptions/changes. ("Service failure" is identified with a transition from correct to incorrect delivery; see [8, Sec. 2.2], for example.)

In the case of safety-critical systems, this focus is perhaps justifiable due to the catastrophic consequences of failure. However, in the more general context of ubiquitous systems, it appears to be unnecessarily restrictive. Instead, this notion can be extended in order to account for degradations in service quality that lie above the threshold of service failure, just as measures of *performability* [9] generalize measures of dependability (e.g., reliability and availability). Accordingly, 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.

Stated informally, a performability measure quantifies a system's "ability to perform in the presence of faults." This extended view of resilience thus 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.

B. Resilience Evaluation

The domain of fault-types considered in both dependability and performability evaluation has expanded considerably over the past 30 years. However, the term "fault" continues to refer mainly to anticipated (foreseen) changes in a system or its environment. Hence, for either definition of resilience – persistence of x when facing changes, whether x be dependability or performability – the added ingredient is persistence with respect to unanticipated changes.

Accordingly, evaluating resilience involves consideration of system and environment dynamics that are beyond those typically addressed in the evaluation of x. 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. For example, one must seek means of

1) accounting for these additional dynamics in the formulation of resilience measures and models, and
2) accommodating 1) in methods of resilience evaluation (resilience model solution).

The remainder of the paper is devoted to discussions of 1) and 2). Of particular note is a technique which we refer to as "Courtois revisited." Although the timing of unanticipated disruptions can be short term as well as long term, we argue that such changes will likely occur much less frequently than anticipated changes such as faults. If for no other reason, events that occur relatively frequently are more likely to be observed repeatedly and, therefore, more likely to be anticipated.

Accordingly, one can employ a popular performability modeling technique (first applied in [10]) based on the "near complete decomposability" theory of Courtois [11]. This permits separation of weakly interacting processes representing anticipated changes from those representing unanticipated changes, thus simplifying both model construction and model solution.

REFERENCES

[1] J.-C. Laprie, "From dependability to resilience," in *Proc. IEEE Int. Conf. on Dependable Systems and Networks*, vol. Supplemental, 2008, pp. G8–G9.

[2] T. Anderson, Ed., *Resilient Computing Systems*. Collins, 1985.

[3] http://www.iris.lcs.mit.edu.

[4] http://www.resist-noe.eu.

[5] E. Hollnagel, W. D., and N. Leveson, Eds., *Resilience Engineering - Concepts and Precepts*. Ashcroft, 2006.

[6] H. D. Foster, *The Ozymandias Principles: Thirty-one Strategies for Surviving Change*. Victoria, B.C., Canada: Southdowne Press, 1997.

[7] B. R. Haverkort, R. Marie, G. Rubino, and K. Trivedi, Eds., *Performability Modelling: Techniques and Tools*. Wiley, 2001.

[8] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004.

[9] 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.

[10] ——, "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.

[11] P. J. Courtois, *Decomposability: Queueing and Computer System Applications*. Academic Press, 1977.