

TOWARDS A DEFINITION OF SURVIVABILITY

John C. Knight

Kevin J. Sullivan

*Department of Computer Science
University of Virginia
Charlottesville, VA*

{knight / sullivan}@cs.virginia.edu

(804) 924 2200 (Voice)

(804) 982-2214 (FAX)

*A Position Paper
submitted to:*

The Third Information Survivability Workshop

TOWARDS A DEFINITION OF SURVIVABILITY

John C. Knight and Kevin J. Sullivan

Dependability

Powerful information systems have been introduced into critical infrastructure applications as the cost of computing hardware has dropped and the availability of sophisticated software has increased [3]. In many cases, the provision of service by infrastructure applications is now highly dependent on the correct operation of computerized information systems, and the failure of these information systems will often eliminate infrastructure service quickly and completely. The dependability of these *critical information systems* has therefore become a major concern [5, 6].

Dependability is a property that a system might have and is usually stated as a set of requirements with which the system has to comply. Dependability has many aspects—reliability, availability and safety, and so on [4]—and to permit precise requirements statements about systems, each of these terms has a precise meaning. For example, the reliability of a system, $R(t)$, is defined to be the probability that the system will meet its requirements up until time t when operating in a prescribed environment. Similarly, the availability of a system, $A(t)$, is the probability that the system will be operating correctly at time t . For systems where dependability matters, the system requirements state the minimum acceptable value for the relevant aspect of dependability, such as $R(t)$ or $A(t)$, and it is then the developers' responsibility to show that the necessary dependability will be achieved during system operation.

Different aspects of dependability are suitable for different systems—highly reliable operation is usually needed for an embedded control system, highly available operation is usually needed in a database system, and a high level of safety is needed for a weapons system. It is important to note that a system can achieve one aspect of dependability but not necessarily others. For example, a system that fails fairly frequently but is only unavailable for a very short time whenever it fails has high availability but low reliability. Many systems are built to operate this way intentionally because it is a cost-effective approach to providing service if reliability (in the formal sense) is not required.

In some cases, the events that damage a system, and thereby possibly impact its dependability, do so in such a way that there is no external effect because of appropriate backups. For example, mirrored disks permit completely normal functionality to be continued after the loss of a disk drive. In other cases, damage is so widespread that functionality has to be changed. For example, a wide-area loss of commercial power might force an on-line database service to cease operation or switch to a remote back-up site that has less throughput capacity. It is possible, therefore, to build a system in which some forms of damage have no observable external effect whereas the effects of other forms of damage are observable. This is the case with critical information systems.

Events that damage a system are not necessarily independent nor are they necessarily mutually exclusive. In practice, a sequence of events might occur over time in which each event causes more damage—in effect, a bad situation gets progressively worse. This might happen, for example, if a system is subjected to a coordinated denial-of-service security attack that becomes increasingly widespread as more system elements are attacked.

Survivability

Survivability is a new aspect of dependability that arises from the need to operate certain systems (a) after an event has occurred that damages a system such that the effects of the damage cannot be completely mitigated, and (b) after a sequence of such damaging events has occurred.

As an example of this, note that many critical infrastructure systems are expected to continue providing service even if a significant amount of the hardware fails, or if power fails over a wide area, or a backhoe cuts a major cable, or some sequence of such events occurs [2].

Informally by a *survivable* system we mean one that has the ability to continue to provide service (possibly degraded or different) in a given operating environment when various events cause major damage to the system or its operating environment. Naturally, the more functionality that can be provided the better, and the functionality that is provided had better meet the customers' needs. In addition, it is certain to be the case that customers expect the "usual" functionality "most" of the time, and, depending on the type of damage, different subsets of "critical" functionality if the system has been damaged. Survivability has to capture this notion. Survivability is a system property that is defined by a prioritized list of the functions that the system must provide together with a probability for each function of its being provided. The priorities define how desirable the associated function is. As an example, consider a database system providing some form of information service. Its owners might consider it survivable if, in the given operating environment, it provides full functionality with probability 0.99, a form of limited service just to high priority clients with probability 0.009, and no service with probability 0.001. Note that these probabilities are conditional probabilities since they only apply in the given operating environment. If the environment is not that for which the system was specified, then nothing can be said about service.

Survivability is a system property that can be required in exactly the same way that the other aspects of dependability are can be required. There is no presumption about how survivability will be achieved in the notion of survivability itself—that is a system design and assessment issue. However, the probabilities associated with each level of functionality are important design constraints since they will determine which design choices are acceptable and which are not.

This notion of survivability is not entirely new in that many critical systems in the past have had requirements for reduced or alternate service under some circumstances. The reason for making survivability a new aspect of dependability is that it is a primary form of dependability needed by critical networked infrastructure systems. By defining it precisely, system designers can state exactly what the user of a system can expect over time in terms of the provision of service.

Defining Survivability

In earlier work, Ellison et al. introduced a definition of survivability that captures the intuitive notion [2]. Building on that work, we introduce an initial formal definition since a precise definition of survivability is important if we are to build survivable systems. If we do not state accurately what we mean by a system being survivable, we cannot determine whether we have made a system that is survivable. Building on the informal notion of survivability introduced above, we define a survivable system precisely using the following definitions:

Definition: A system is *survivable* if it complies with its survivability specification.

Definition: A *survivability specification* for a system is:

- (1) A precise statement of the assumed operating environment for the system.
- (2) A set of specifications defining the functions that the system must provide. No functionality might be an acceptable member of this set for some systems.
- (3) An ordering on the set of functions denoting the preferred order of provision of the functions.
- (4) A probability distribution across the set of specifications. The probability associated with a given specification is the probability that the associated specification will be met. Note that any particular specification might include its own dependability requirements relative to the assumed environment.

Intuitively, what this means is that: (a) a survivable system will provide a preferred functionality with a certain probability, (b) if that is not possible for any reason, the system will provide the next preferred level of functionality with a certain probability; (c) and so on in the order of preferred functions.

Faults and Fault Tolerance

The informal notion of an “event” that causes damage which we have used is referred to formally as a *fault* [1]. The process of building a system in such a way that certain faults do not arise is *fault avoidance*. Building systems that are able to react in a requisite way to prescribed faults is *fault tolerance*.

In many cases, systems are built using techniques of replication so that the effects of a fault do not affect the system’s external behavior. Such faults are said to be *masked*. Usually for economic or similar practical reasons, some faults are *non-masked*; that is, their effects are so extensive that normal system service cannot be continued with the resources that remain even if the system includes extensive redundancy. These concepts of masked and non-masked faults are the formal statements of the idea introduced above of events that cause damage whose effects cannot or can be observed in the system’s behavior.

Survivability is *not* fault tolerance—it is a dependability property. Fault tolerance is a mechanism that can be used to achieve certain dependability properties. In terms of dependability, it only makes sense to refer to a system as reliable, available, secure, safe, survivable and so on, or some combination using the appropriate formal definition(s) [4].

Fault tolerance is a mechanism by which some aspect of dependability *might* be achieved but it is not the only mechanism. Fault avoidance can be used also. Thus, for example, by careful component selection it might be possible to reduce the rate of hardware failures in a given system to a negligible level, and by suitably restricting system access it might be possible to eliminate certain types of security attacks.

Fault Tolerance and Survivability

Any aspect of dependability can be required of a system and can be specified with a probability of any value. For example, a computer system might be required to have a reliability of 0.999999 for ten hours of operation ($R(10 \text{ hours}) = 0.999999$). This is a very high value that is very difficult to achieve, and to do so requires many special system design and construction techniques. For lower probabilities, simpler design and construction techniques can be used. Naturally, a reliability of 1 is cannot be achieved in any meaningful case.

A survivability specification could be written with just two functions—full functionality and no functionality—and with a very high probability for full functionality (like 0.999999) and a very small probability of no functionality (like 0.000001). Such a specification would be a survivability specification but it would be better to think of it as a reliability specification. It would be impossibly expensive to achieve such a specification for a critical infrastructure application since it means that no degraded or alternate service is acceptable and so essentially all damage has to be prevented or masked. For a large distributed information system this is just not realistic.

A practical survivability specification will have achievable probabilities and carefully selected functionality specifications. Thus, in such a system, the effects of damage will not be masked necessarily; and, provided the probabilities are met in practice, degraded or alternate service will occur. In effect, this implies that the survivability requirement will be achieved by the fault tolerance mechanism. Note, however, that the N different functions in the survivability specification do not correspond to functions that can be achieved with the resources that remain after N different faults. The N functions in the survivability specification are defined by application engineers to

meet application needs and bear no prescribed relationship to the effects of faults. Many different faults might result in the same degraded or alternate application function.

In order to implement fault tolerance, the faults that have to be tolerated must be defined. A system can never be built that merely tolerates “faults”. It is essential to state what faults have to be tolerated because otherwise the damage that the fault causes cannot be predicted and hence treated. Precisely what faults are anticipated and what the system is required to do when specific faults occur, i.e., how the fault will be treated, must be defined so that the system builders know what states the system might enter and with what probability. This is crucial input to the process that has to lead to a design for a survivable system.

The analysis of faults has to be undertaken by application experts, hardware experts, security experts, disaster experts, and others. This information is essential system information and thus it is not within the purview of the computer system developers. Having to state what faults are likely to occur might seem counter-intuitive in the sense that damage is difficult to predict. But in fact the only way that appropriate responses can be defined is in the context of a known system state and that means a state in which the remaining resources are defined. As a simple example, consider again a database system. Anticipated faults might include processor failure, disk failure, power failure, operator error, and perhaps flooding. In a multi-server environment it is possible to define a reasonable set of (reduced) functions that have to be maintained if a single server fails. If all the servers fail because of a flood, then it is likely that no service could be maintained. These are fundamentally different cases. Faults that were not defined as part of the system analysis, such as a gas explosion that destroys *unspecified* amounts of equipment, cannot possibly be followed with any degree of assurance by any function from the survivability specification. This is because the system will not have been designed to cope with that particular fault.

Conclusion

In this paper we have presented an initial formal definition of survivability and related it to the field of dependability and the technology of fault tolerance.

Acknowledgments

This effort was sponsored in part by the Defense Advanced Research Projects Agency and Rome Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-96-1-0314. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency, Rome Laboratory or the U.S. Government. This effort was sponsored in part by the National Science Foundation under grant number CCR-9804078.

References

- [1] Anderson, T. and P. Lee. *Fault Tolerance: Principles and Practice*. Prentice Hall, Englewood Cliffs, NJ, 1981.
- [2] Ellison, B., D. Fisher, R. Linger, H. Lipson, T. Longstaff, and N. Mead. “Survivable Network Systems: An Emerging Discipline,” Technical Report CMU/SEI-97-TR-013, Software Engineering Institute, Carnegie Mellon University, November 1997.
- [3] Knight, J., M. Elder, J. Flinn, and P. Marx. “Summaries of Three Critical Infrastructure Systems,” Technical Report CS-97-27, Department of Computer Science, University of Virginia, November 1997.
- [4] Laprie, J. “Dependable Computing: Concepts, Limits, Challenges,” Special Issue FTCS-25: 25th International Symposium on Fault-Tolerant Computing, June 1995, pp. 42-54.
- [5] Office of the Undersecretary of Defense for Acquisition and Technology. “Report of the Defense Science Board Task Force on Information Warfare - Defense (IW-D),” November 1996.
- [6] President’s Commission on Critical Infrastructure Protection. “Critical Foundations: Protecting America’s Infrastructures The Report of the President’s Commission on Critical Infrastructure Protection,” United States Government Printing Office (GPO), No. 040-000-00699-1, October 1997.