

Towards a Rigorous Definition of Information System Survivability

John C. Knight Elisabeth A. Strunk Kevin J. Sullivan
Department of Computer Science
University of Virginia
151 Engineer's Way, Charlottesville, VA 22904-4740
{knight|strunk|sullivan}@cs.virginia.edu

Abstract

The computer systems that provide the information underpinnings for critical infrastructure applications, both military and civilian, are essential to the operation of those applications. Failure of the information systems can cause a major loss of service, and so their dependability is a major concern. Current facets of dependability, such as reliability and availability, do not address the needs of critical information systems adequately because they do not include the notion of degraded service as an explicit requirement. What is needed is a precise notion of what forms of degraded service are acceptable to users, under what circumstances each form is most useful, and the fraction of time such degraded service levels are acceptable. This concept is termed survivability. In this paper, we present the basis for a rigorous definition of survivability and an example of its use.

1. Introduction

The technical community uses the term *survivability* in the context of networked information systems. Systems are sometimes described as being survivable, technologies are claimed to enhance survivability, and so on. Unfortunately, despite the common use of the term, no precise and adequately comprehensive definition of survivability exists, and what constitutes a survivable system thus is not defined well in an engineering sense.

A narrow view that is sometimes taken is that a system is survivable if it can tolerate certain kinds of faults. A server, for example, might be described as survivable if it is able to withstand certain types of security attacks. This doesn't necessarily make the server survivable, however, because it might still fail if it experiences a hardware fault, is damaged by an explosion, and so on. Similarly, a command-and-control system might be claimed to be survivable because it uses intrusion detection, but the system might fail if subjected to battle damage, which is not necessarily what is required. Under these circumstances, a system characterized by its developers as survivable might not meet the needs of

its owners and users because it does not have some essential properties that they thought were implied by the term.

What we need is a precise and adequately comprehensive engineering definition of survivability, analogous to those that have been framed for the dependability characteristics described by Avizienis et al. [3], such as reliability, availability, and security. Reliability, for example, is defined as the probability that a system will meet its (non-reliability) requirements over a given period of time under given operating conditions.

The notion of survivability that we need must have three essential characteristics. First, it must be broad enough to encompass various types of damage to the system. Second, it must include alternate forms of service, each achieving an effective tradeoff between the benefits of a given level of function and the cost of providing it, under the range of operating conditions for which it is defined. Third, it should model the probability that each of these services must be available for use, as a function of the conditions that the system is expected to encounter.

Furthermore, the definition must be precise and unambiguous enough to support an engineering approach to the specification, design, and analysis of critical information systems. If we do not state precisely what we mean by the term *survivable*, we cannot determine whether we have made a system that meets its real requirements. This is not merely an academic point. The owners and users of a system need to be able to determine that, with reasonable assurance, the system will perform adequately in its environment.

In a nutshell, then, the problem that we address in this paper is that we currently lack an engineering definition of survivability for critical information systems: one that allows us to state precisely what properties are required of a system and to do the kind of design analysis that reveal whether they are present before we come to rely on the system. The chief contributions of this work in this dimension are, first, a case for the proposition that we need such an engineering definition of survivability; and second, a proposed basis for a rigorous engineering definition of the term with an example to make the ideas concrete.

The rest of this paper is organized as follows. The next section discusses the need for survivability. Section 3 reviews related work. Section 4 presents a brief summary of two critical infrastructure applications to provide a context for survivability. In section 5 we discuss an intuitive notion of survivability and our proposed basis for a rigorous definition. To illustrate the various aspects of the definition, we present an example in section 6. Finally, in section 7, we present our conclusions.

2. The need for survivability

Powerful information systems have been introduced into both civilian and military critical infrastructure applications as the cost of computing hardware has dropped and the availability of sophisticated software has increased [8]. In many cases, the provision of service by infrastructure applications now depends on the correct operation of information systems, and damage to these systems will lead to a loss of at least part of the service. In some cases, relatively minor damage can lead to a complete cessation of service. We refer to such information systems as *critical information systems*.

The dependability of critical information systems is a major concern [12, 13]. Dependability is a system property that is usually stated as a set of requirements with which the system must comply. It has many facets—reliability, availability, safety, etc. [3]—and to permit exact requirements statements, each such term has a precise meaning.

Different facets 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 might achieve one facet of dependability but not others. Many systems are built to operate this way intentionally because it is a cost-effective approach to providing service if, for example, reliability is not required but availability is.

In specifying dependability for a given system, it is usually the case that full system functionality is required—nothing is stated beyond, perhaps, failure semantics. For critical infrastructure applications, this is insufficient. Some events that damage a system have no external effect because of appropriate redundancy; for example, mirrored disks mask the effects of data loss. In other cases, damage is so widespread that *functionality has to be changed*. A wide-area loss of commercial power, for example, might force a critical on-line database service to switch to a remote backup site that has less throughput capacity or reduced functionality. Such circumstances arise in infrastructure applications with sufficient frequency that comprehensive provision for them must be made. Thus, the different forms of service that a system might be forced to provide during

routine operation must be specified, and users need to be aware of the probability that different services will be provided.

For the developer of a critical information system, knowing what service is required in the event that full service cannot be provided is crucial. This information is essential input to the design process for the critical information system, since achieving even some form of reduced service will almost certainly necessitate specific design choices.

3. Related work

3.1 Current definitions of survivability

The notion of survivability has been used in several engineering disciplines outside of critical information systems. For example, it is a common concept in weapons systems engineering [11, 4]. The survivability of combat aircraft has the following definition [18]:

Survivability: Aircraft combat survivability is the capability of an aircraft to avoid and/or withstand a man-made hostile environment. It can be measured by the probability the aircraft survives an encounter with the environment, P_S .

The Institute for Telecommunications Services, part of the U.S. Department of Commerce, has created a definition of survivability for telecommunications systems [16]:

Survivability: A property of a system, subsystem, equipment, process, or procedure that provides a defined degree of assurance that the named entity will continue to function during and after a natural or man-made disturbance; e.g., nuclear burst. Note: For a given application, survivability must be qualified by specifying the range of conditions over which the entity will survive, the minimum acceptable level or post-disturbance functionality, and the maximum acceptable outage duration.

Both of these definitions seek a framework to define service after some form of damage, and they relate closely to our goal of defining survivability for critical information systems. It is interesting to note that both definitions are probabilistic, and that the second definition includes the concept of degraded or different service and requires that such service be defined.

In the context of software engineering, Deutsch has offered the following definition [6]:

Survivability: The degree to which essential functions are still available even though some part of the system is down.

This definition is a good start in this context, but it is not sufficient. It presents a general intuitive notion of the con-

cept of survivability, but does not allow developers to create a precise specification of what survivability means for a particular system. If applied to a critical information system, the user of the system could be sure of neither which functions had been selected as “essential” nor when (i.e., after what damage) these functions would be provided.

In earlier work specifically on information system survivability, Ellison et al. introduced the following definition [7]:

Survivability: Survivability is the ability of a network computing system to provide essential services in the presence of attacks and failures, and recover full services in a timely manner.

While this definition is more precise, it still does not have the precision needed to determine whether a given system can be deemed survivable. Much is implied by the phrases “essential services”, “attacks and failures”, and “timely manner”. If nothing further is defined, a system developer could not determine whether a specific design is adequate to meet the needs of users, and might even be expected to define the “essential services” himself rather than implementing the services chosen by application experts.

A second problem with a definition of this form is that it provides no decidable criteria for the term being defined. In contrast, the definition of reliability makes a clear distinction between the general *informal* view of a reliable system (it “never” fails) and the *formal* view provided by the definition that a system is reliable if it meets or exceeds a probabilistic goal. By that definition, a system might fail and yet formally still be considered reliable. The same degree of clarity is needed for survivability so that we may consider a system to be survivable and know what that means.

In the field of information survivability more generally, a body of research results has begun to appear. A valuable source of material is the series of Information Survivability Workshops [17]. Many relevant papers also have appeared in various other conferences concerned with dependability.

3.2 Survivability and other related concepts

Avizienis, Laprie, and Randell present a comprehensive discussion of dependability [3], defining it informally as “the ability of a system to avoid failures that are more frequent or more severe, and outage durations that are longer, than is acceptable to the user(s)” [3, p.2]. They define it more precisely as a composite of availability, reliability, safety, confidentiality, integrity and maintainability. Dependability requirements specify, in these dimensions, the “acceptable frequency and severity of the failure modes, and of the corresponding acceptable outage durations (when relevant), for a stated set of faults, in a stated environment” [3, p.5]. Avizienis et al. then argue that survivability and dependability are equivalent—“names for an essential

property” [3, p.12]—with survivability defined more narrowly than dependability: the former in terms of specific threats, and the latter in general. Survivability is seen as a special case of dependability, and survivability requirements as a special case of dependability requirements.

This paper meets a need not adequately addressed by this view, which suggests a form of specification that defines a single service and corresponding single set of dependability requirements. We define survivability in a way that emphasizes the need to specify systems that can provide different forms of service, each with its own complete set of dependability requirements, under different conditions.

The problem with the single-service view is that it might not be cost-effective to provide the core service under all conditions where some level of service is needed. The traditional single-service view can thus lead to systems that are dependable in the sense that they meet their stated dependability requirements, but are not survivable, in the sense of being able to provide degraded service outside the range of threats that the full service is able to deal with. It might not be cost effective to provide assurances of full service across the entire range of threats, so a narrower set of threats is selected to be addressed by the dependability requirements. The decomposition of the system into several forms of service, each having its own dependability requirements, allows different services to be provided—cost-effectively—outside the conditions addressed by the core service.

The informal notion we have used of an event that causes damage is referred to formally as a *fault* [1]. 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 of events that cause damage whose effects cannot or can be observed in the system’s behavior.

Survivability is a measurable system characteristic; it is not synonymous with fault tolerance. Fault tolerance is a mechanism that can be used to achieve certain dependability properties required in a survivability specification by coping with faults that remain in a system once it is deployed. In terms of dependability, it makes sense to refer to a system as reliable, available, safe, and so on, or some combination using the appropriate formal definition(s) [3]. Describing a system as fault tolerant is really a statement about the system’s design, not its dependability.

While fault tolerance is a mechanism by which some facets of dependability might be achieved, it is not the only mechanism. Other techniques, such as fault avoidance, also can be used. 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. In similar ways, fault elimination and fault forecasting can be used as mechanisms to improve a system’s dependability.

Finally, we note that the commonly used but informal term “graceful degradation” is not survivability. Some systems have been described as being capable of degrading gracefully by which is meant that services are removed gradually as resources are lost. In addition to being imprecise, this notion neither specifies which services are lost at which times nor considers individual or overall dependability requirements. For our purposes, it is far from complete.

4. Critical information systems

4.1 Examples of critical information systems

Some background material about critical information systems is helpful in understanding the need for a precise notion of survivability and how it differs from other notions of dependability. Detailed descriptions of four systems are available elsewhere [8].

- *Banking and Financial Services.* The nation’s banking and finance systems provide a very wide range of services implemented by complex, interconnected, networked information systems. The most fundamental service is the financial payment system [15], the mechanism by which value is transferred from one account to another. From the perspective of our use of financial services, the payment system is crucial; essentially nothing works without it. Credit card services, on the other hand, can be suspended for a much longer period of time before financial disaster ensues. The availability of currency lies somewhere in between. If the full set of services could not be maintained, clearly every effort should be made to keep the payment system operational. If that were not possible in its entirety, then payment services for critical government agencies might be an appropriate alternative. The precise functionality made available would depend on the time of day, since private financial systems make use of it differently at different times.

- *Freight Rail Transportation.* The freight-rail transport system moves large amounts of raw materials, manufactured goods, fuels, and food [2]. Operation of the freight-rail system uses computers extensively for a variety of purposes. For example, every freight car in North America is tracked electronically as it moves, and very large databases of car and locomotive locations are maintained. This data permits tracking of specific shipments and scheduling of freight cars for individual trains—a massive task. A particularly important use of this system is *just-in-time* delivery. Train move-

ments are scheduled so that, for example, raw materials arrive at a manufacturing plant just as they are required. In a system such as this, basic equipment tracking is more important than optimization, and certain payloads (such as perishable items) are more important than others. In the event that full service cannot be maintained, equipment tracking is clearly the most important task. If for some reason even that were not possible, then basic functions that would allow the network to be shut down safely would be the preferred service.

4.2 Command-and-control example

To illustrate the concepts described in this paper, we present an example using a hypothetical military command-and-control (C2) system. We assume the very general network topology shown in Figure 1 in which there are a large number of leaf network nodes used by local commanders, a smaller number of intermediate nodes that provide regional or specialized services such as intelligence development or logistics, and a centralized facility that enables commanders to view wide-area information that facilitates strategic decisions.

For this hypothetical system, we assume the following description of normal operation:

- *Full Command, Control, and Analysis.* This is complete and normal functionality. The system provides both central and regional information servers, delivery of crucial information like weather data, transmission of command information, transmission of local status and observational data to regional centers, synthesis of incoming data for the regional and central command centers, and so on.

Given the significance of command and control, it is likely that certain critical services might be desirable if the

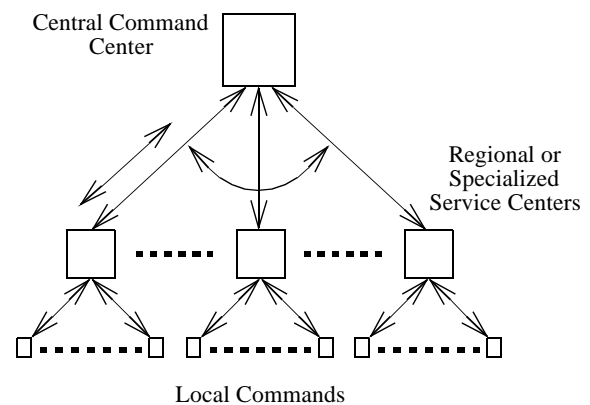


Figure 1. Hypothetical command-and-control system

preferred service is unavailable for some reason. Possible alternate forms of service include:

- *Low Performance.* During peacetime, more time is available for analysis of intelligence and status information since speed of command information is not as critical. This service provides full functionality, but with higher latencies.
- *Regional.* During a regional conflict, access to only that region might be acceptable if full service could not be provided. This service is limited to central command and a single regional center. Synthesis and analysis of wide-area information are unavailable.
- *Maximum Alert.* During wartime, operations might require a very defensive network security stance because of the extreme sensitivity of the material that the system is transmitting. This might be a particularly important issue if a network has been damaged already. This alternate service requires that the system operate with no network traffic because it might be compromised. Local processing is continued, and all possible security measures are in place.
- *Command Only.* During wartime, the priority of command information is far higher than during peacetime. This service limits operation to basic command transmission. Routine data transmission is unavailable, and access to servers is unavailable.

4.3 Critical information system characteristics

Society now faces an unquantified but clearly serious risk that reliance on fragile and insecure information systems will compromise delivery of critical military or civilian infrastructure services. The cost of disruptions grows more rapidly in time now than before computerization, yet increasing reliance on computers increases vulnerability to disruption. The central problem that we face is to devise approaches to critical information system design and evolution that simultaneously enable the efficiency that computers make possible while ensuring that the costs of service stream interruptions remain acceptable in the face of disruptions to underlying information systems.

The scale, sophistication, and makeup of information systems complicate the established notion of dependability considerably. It is not useful to speak of the reliability of a command-and-control system, for example, because some information gathering and analysis equipment is sure to be malfunctioning at any time with little effect. However, there are some forms of damage that would be extremely serious; for example, a loss of information gathering capabilities over a wide area would cripple the entire system because interpolation of missing data would no longer be possible. Such events are inevitable and must be dealt with in some way. The continued provision of some form of service is

more than desirable—in many cases it is essential. In our C2 system, for example, if full service cannot be maintained during a war, then delivery of crucial troop movement commands is required.

To provide a basis for a discussion and to guide a definition, we enumerate the various characteristics of infrastructure applications that affect the concept of survivability. The characteristics are:

- *System Size.* Critical information systems are very large, both geographically and in terms of numbers and complexity of computing and network elements. It is infeasible to engineer such systems so that none of their components fail during normal periods of operation, yet scale precludes comprehensive redundancy.
- *Externally Observable Damage.* In some cases, the effects of damage to a system will be so extensive that they will be visible to the system's users in the form of a change in service or the quality of service.
- *Damage and Repair Sequences.* Events that damage a system might be neither independent nor mutually exclusive; a coordinated security attack for example. In practice, a sequence of events might occur over time in which each event causes more damage despite possible partial repair.
- *Time-Dependent Damage Effects.* The impact or loss associated with damage tends to increase with time. A protracted loss of military communication would be devastating. Officers in the field could take some initiative to compensate for such a loss, but over time the loss would prevent strategic decisions from reaching key personnel.
- *Heterogeneous Criticality.* The requirements for dependability in infrastructure systems vary with function and with time. Our C2 example places different emphasis on communication and analysis depending on whether it is being used during a military conflict and whether an information attack is imminent.
- *Complex Operational Environments.* The operating environments of critical infrastructures carry risks of natural, accidental, and malicious disruptions from a wide variety of sources; sometimes highly variable loads that vary both over time and space; varying levels of criticality of service; and so forth.

The factors in this list combine to present a picture of critical information systems that is quite different from computer systems which exemplify traditional dependability requirements. For example, avionics systems typically require high reliability and telecommunications switches typically require high availability, and they possess few of the characteristics listed above. Dealing with all of the characteristics is essential if a particular critical information system is to be viewed as survivable. With this in mind, we proceed to formulate a definition of survivability.

5. Survivability

5.1 The intuition behind the definition

In an informal sense, by a *survivable system* we mean a system that has facilities to provide one or more alternate services (different, less dependable, or degraded) in a given operating environment. An alternate service would be required to be in effect if an event (such as some form of damage) precludes provision of the system's normal service. This idea is extended to include several alternate services so as to be able to cope with different forms of damage, damage that occurs under different circumstances, and damage that gets worse over time. This notion of survivability of computing systems is not new in that many critical systems have requirements for alternate service under some circumstances; however, such requirements are created in an ad hoc manner, rather than as a rigorous attempt to guarantee certain system properties.

In principle, the notion of survivability could be completely avoided. Apart from incorrect specifications, the only reason that a system fails to provide service is because of the manifestation of faults (degradation, design, and malicious). Many types of faults can be avoided, eliminated, or tolerated, and, for a wide variety of systems, high levels of dependability can be achieved.

The reason that survivability is necessary is primarily one of resources. For example, if a very high level of availability were required for a large distributed system that was vulnerable to coordinated terrorist attacks, the resources required to implement the necessary redundancy to mask the effects of an attack would be prohibitive. Large parts of the system might have to be duplicated completely.

Survivability offers a tradeoff between functionality and resources. This tradeoff is exploited by building elements of the system (such as those implementing the primary functionality) with less provision for coping with faults than normally might be preferred. This reduced provision results in a less expensive implementation of those elements, possibly taking the implementation from a complexity level that was infeasible to one that is feasible. The potential loss of service that ensues is dealt with by providing facilities for alternate service when the primary implementation is unavailable. The user sees a larger value in the system, either in the form of reduced cost or increased options for functionality.

The reduced provision for coping with faults takes the form of *consciously* designing the primary system with no way to cope with certain carefully selected classes of faults. By doing so, the cost is reduced but the potential for failure is increased. Provided the rate of failure is below what is deemed an acceptable threshold and provided alternate ser-

vice is supplied, users are likely to consider operation to be satisfactory.

The determination of what fault classes should not be tolerated by some element of the system translates directly into details of the operating environment. For example, choosing explicitly that the primary system will not deal with a loss of power corresponds to the primary system's being designed to meet its dependability requirements in an operating environment that assumes continuous availability of power. This makes careful definition of the operating environment especially important.

In this section, we present the various technical aspects of the concept of survivability, and we provide the motivation and rationale for their presence. We break down the discussion into five parts—acceptable services, service value, service transitions, service environments, and service probabilities—and discuss each in turn. These technical elements provide the necessary intuitive basis for the more rigorous definition that we present in the next section.

5.1.1. Acceptable services. For any particular system, the required alternate service functionality and associated dependability are application specific. However, there are three general principles that stem from the discussion of information systems in Section 4.3 which will facilitate our discussion of the subject and clarify the framework that we present:

- Users expect the “usual” functionality “most” of the time.
- If normal functionality cannot be provided, users might require different alternate functionalities under different conditions.
- In a very general sense, the amount of functionality provided and the degree to which the users' needs are met affect the utility that the users receive from the systems.

Refining the informal notion of survivability somewhat, we observe that an acceptable (but not necessarily preferred) service combines functions that work harmoniously and provide functionality to the user. To cope with the prospect of different forms of damage and of damage that worsens over time, multiple alternate forms of service might be defined. The *set of acceptable services* in a survivable system is comprised of the different forms of service that the system must be capable of providing so as to ensure that the functionality received by the users meets their needs to the greatest extent possible. Note that a service as used in this context may include several separate functions, such as command transmission and intelligence analysis.

5.1.2. Service value. In an earlier paper [14], we presented a value-aware [5] concept of survivability for critical information systems, viewing a system as providing a service stream that delivers some value to its users. The particular

value it provides at any given point in time depends on user and circumstances; electric service is more valuable in winter than spring, and more for hospitals than residences, for example. Similarly, a military command-and-control system provides attached units with information whose value depends on those units' needs and circumstances. Value delivered to individuals then sums to an aggregate value delivered to a customer base over time. The notion of survivability from the earlier work is that a survivable system adapts in predefined ways in the face of changing, defined circumstances to ensure that the aggregate value delivered to users remains above required thresholds in the face of a variety of possible disruptions.

In this paper, we view value as an important driver of system requirements and operation because it provides a framework for ordering forms of service. For example, in a C2 system faced with system damage, preservation of strategic communications might be more important than the distribution of weather data. This might change if the local tactical needs for weather data outweighed the benefit of strategic communications. To order forms of service, we use the notion of *relative service values*, values that are determined by the user subjectively, and which are functions of operating conditions. The priority ordering of services by value based on conditions is given as a part of the requirements for a survivable system. The mechanism of the system then effects the value-aware but pre-computed switching policy based on monitored conditions. Using this simple notion avoids complex issues not worked out in the earlier paper [14] that must be resolved in order to formulate survivability control systems to explicitly optimize for value.

5.1.3. Service transitions. At any one time, the system will be providing only one member of the set of acceptable services. One element of the set defines a preferred service that the system's users regard as "full" or "normal" service. Application experts would be responsible for specifying the various different forms of acceptable service.

If maintaining a particular service becomes impossible, the system is required to reconfigure so as to comply with a different member of the set of acceptable services. This requirement indicates that a *set of valid transitions* will be needed.

Given that a set of acceptable forms of service is defined, it is necessary to define which member of the set is preferred for any given set of conditions. This selection is effected by choosing the service with the greatest relative value at the time the transition has to be made.

By definition, relative value changes over time as elements of the operating environment change. Thus, one or more services might come to have higher relative value than the service being provided and thereby prompt a change even though no additional damage has occurred.

5.1.4. Operating environment. The relative value of a specific acceptable service is a complex function of a several entities that might include aspects of the operating state, calendar time, etc. For example, in a command-and-control system, the value of the various alternate services might depend on whether conflicts are ongoing, which weapons systems are available, whether strategic changes are being made, even the weather conditions. This notion implies that the *set of details of the operating environment* that affect the relative values of the acceptable services needs to be identified to permit the correct ordering of relative values.

The operating environment is a critical notion in the dependability requirements that will be associated with the various acceptable services since such requirements are stated with respect to an assumed operating environment.

5.1.5. Service probabilities. The existence of a set of acceptable services for a system provides no quantitative indication of the level of service that the system will deliver. For example, if faults arise in a survivable system in such a way that the system is forced to provide an acceptable yet completely minimal service almost all of the time, the system's users are likely to find the service unsatisfactory; they expect the "usual" functionality "most" of the time. Such a system would be survivable in an intuitive sense, but it would not be very useful.

To avoid this problem, there needs to be an assurance that the various different acceptable services will be provided at some adequate level; in particular, that the primary service is. To achieve this, we introduce the notion of *service probabilities*, one for each of the defined acceptable services. The service probability and the dependability requirements of a specific acceptable service work together to solve the problem outlined above.

The service probability associated with any given acceptable service is the probability with which that particular service has to meet its dependability requirements. The dependability requirements for an acceptable service in the sense being used here include a statement of the operating environment under which they must be met. Thus, for example, an availability requirement states a probability of readiness for service under *prescribed* conditions. It is precisely the limitation of the severity of these conditions that is at the heart of the survivability tradeoff—to reduce cost or complexity, some are omitted. So how are the conditions to be omitted determined?

Recall that an acceptable service in a survivable system is built intentionally to cope with fewer faults than are expected to arise. Thus, if the operating conditions remain within the prescribed conditions (none of the faults that were intentionally excluded arise), the acceptable service is expected to meet its dependability requirements, and it will be designed to do that. If the operating conditions fall outside the prescribed conditions (one of the faults that was

intentionally excluded does arise), the acceptable service is *not* expected to meet its dependability requirements.

The service probability for any specific acceptable service is a requirement. To meet it, the probability that the operating conditions are within the prescribed conditions has to be equal to or less than the service probability—in effect it is a statement of how much service loss is acceptable from the survivability tradeoff.

A comprehensive analysis of the anticipated operating environment of a proposed system will include the types of faults expected, their expected frequency of occurrence, their effects, and so on. With this information in mind and the desired service probability for a given acceptable service, it is possible to determine which faults have to be dealt with by the acceptable service and which do not.

If for a specific acceptable service a given fault type is expected to occur with a frequency and with an effect that would preclude meeting that service's dependability requirements, then the acceptable service has to be designed to cope with that fault type. For a given required service probability, this analysis determines the operating conditions that have to be included in the dependability requirements for the service. With that defined, the implementation can be engineered to meet it.

As an example, consider the command-and-control system described above. If the service labeled *Full Command, Control, and Analysis* is required to have an availability of 0.995 with a service probability of 0.999, then in practice its operating conditions must be those for which it was designed with the service probability or better. Suppose that an identified degradation fault will cause loss of primary power (and hence loss of service) to occur with probability 0.05 per hour and with average duration 12 minutes. This would preclude this service from operating roughly one percent of the time if the service did not handle this fault. Thus, it has to be designed to do so; otherwise, it would violate the service probability. If instead the fault occurrence probability were 0.005 (ten times less), then the service would not have to handle the fault. Note that the semantics of this are not the same as simple multiplication of these probabilities; in the event that the system cannot meet its availability requirement, it will switch to an alternate service rather than simply remaining unavailable.

The service labeled *Full Command, Control, and Analysis* might become unavailable as a result of any number of faults, but provided it was designed to deal with them and meets its dependability requirement (specified as an availability), its service will be deemed acceptable by its users (even if it is occasionally unavailable), and no alternate service need be invoked. However, if faults arise for which it was not designed to cope, a transition to an alternate service will be necessary.

5.2 Defining survivability

While a discussion of survivability such as is presented in the previous section provides an outline of the concepts needed to specify a survivable system, it suffers from the drawbacks of any informal discussion. Developers require a rigorous definition that will enable precise system specification in such a way that they can determine with some assurance whether a system that complies with the specification will meet the goals of the user. We present an outline of this definition by specifying a framework within which individual dependability definitions will fall. Above, we have suggested how this approach might work for availability, one aspect of an individual dependability specification. A complete discussion of the details of all the aspects of dependability is left for future work.

The definition that we present in this section depends upon terms that require their own definitions. We have introduced the notions informally above, and we present precise definitions here:

Environmental Factor: Any aspect of the environment in which the survivable system is required to operate that can affect that system's operation. An environmental factor might take any one of several values. An environmental factor that has a value associated with it is an *environmental condition*.

Service Value Factor: Any environmental factor that affects the relative value to the user of an acceptable service. A service value factor that has a value associated with it is a *service value condition*.

Operating Factor: Any environmental factor that affects the system's ability to meet the dependability requirements of an acceptable service. An operating factor that has a value associated with it is an *operating condition*.

Traditional dependability requirements are defined in terms of a specification. For example, if a reliable system is needed, engineers state what they want by specifying that the system meet the functionality requirements with a certain probability for a specific time assuming a certain environment. Our approach to defining survivability, therefore, starts with a specification statement:

Survivable System: A system is survivable if it complies with its survivability specification.

This definition treats survivability as a system property that must be engineered rigorously. Software developers cannot be expected to determine the criteria a survivable system must satisfy. Their job is to engineer the system so that it meets the needs determined by people who know what value the system truly must provide.

This definition might seem simplistic; the apparent simplicity stems from the underlying complexity of the specifi-

cation structure. In order to make this definition a realistic one, the specification framework must guarantee that this definition does indeed cover the intuitive notion of survivability by forcing specifiers to be thorough and precise. Our argument takes the individual concepts in Section 5.1 and places them on a more rigorous footing, using a Z-like notation in places for clarity and simplicity.

Survivability Specification: A survivability specification is a six-tuple, $\{S, E, D, V, T, P\}$ where:

$S \subseteq [\text{Specifications}]$

A set of specifications of acceptable forms of service from the system.

$E: [\text{Service value factors}] \rightarrow \mathbb{P}[\text{Values}]$

A function from the set of service value factors to the set of values that each factor can take. The given set $[\text{Values}]$ holds all possible values that could be associated with any service value factor. Note that these values are distinguished from the relative service values discussed above.

$D \subseteq \{d: [\text{Service value factors}] \rightarrow [\text{Values}] \mid \forall a: [\text{Service value factors}], b: [\text{Values}] \bullet a \mapsto b \in d \Rightarrow b \in E(a)\}$

The set of all combinations of values of the service value factors that the application might encounter.

$V: S \times D \rightarrow \mathbb{N}_1$

A function defining the relative service value each specification provides to the user under each set of environmental conditions.

$T \subseteq S \times S \times D$

The set of valid transitions between acceptable forms of service.

$P: S \rightarrow \{p: \mathfrak{R} \mid 0 < p < 1\}$

A set of probabilistic requirements on the operation of the acceptable forms of service.

The meaning of each of these elements is as follows:

- *S—Specification Set*

S is the set of specifications of acceptable forms of service for the system. Each specification may be formal or informal, but must detail exactly what the service has to accomplish. An implementation of the system will include implementations of each element of S .

As well as functional and non-functional specifications, each member of S will include a specification of all pertinent aspects of dependability (availability, reliability, etc.) for that service and the set of operating conditions under which the dependability specification is required to be met. Specifications with the same functionality but different dependability requirements are separate specifications, and may be included as separate members of S .

In our example, the set S is equal to $\{\text{Full Command, Control, and Analysis; Low Performance; Regional; Maximum Alert; and Command Only}\}$. *Full Command, Control, and Analysis* is the specification that the users expect most of the time.

- *E—Service Value Factors*

E is the set of service value factors to which the system is subject. As the service value conditions change, the relative service values of the acceptable services change. For example, the relative service value associated with our command-and-control system depends on the political situation (peacetime or state of war); in peacetime, *Low Performance* might be the most valuable alternate, whereas in wartime, *Command Only* might be. E also might include certain specific failure modes or threats that will have an impact on which specification element provides the most value. Our example system will react differently to a coordinated security attack than it would to other types of damage.

- *D—Reachable Environmental States*

D is the set of possible combinations of environmental conditions the system might encounter. That is, it is all combinations of values the various elements of E could take on that are practically possible. D could include the entire state space defined by E , but this is not always necessary—a dispersed conflict cannot exist during peacetime, for instance. Each element d_i of D is a set of factor-value pairs allowed by E , which is interpreted as a predicates on the system state.

- *V—Relative Service Values*

V defines an ordering on the user's perceived service value from the specifications in S under each reachable environmental state. The ordering is represented by associating a natural number with each element of S , with higher numbers representing higher values. Our example ordering is shown in Table 1. The orderings must be total in order to ensure that the choice of which transition to take in the event that a different specification should be put into operation is deterministic, as explained below.

- *T—Set of Valid Transitions*

T is a subset of all possible combinations $S \times S \times D$. Each element in T represents one of the transitions between acceptable forms of service that the system may take either because: (1) the relative values of the acceptable services have changed because of changes in environmental factors; (2) the currently operating acceptable service becomes unavailable; or (3) an acceptable service of higher value becomes available. The first element, the *source* specification, is the specification that the system currently satisfies. The second element, the *target* specification, is the specification to which the system may

transition. The third element, the transition guard, is a precondition for that transition to occur. The guard represents the relative value conditions in which the transition may be taken, along with any further restrictions deemed by the user to be necessary. Self loops may be used in the event that the system is anticipated to stay in the same state under certain changes in environmental conditions. The relative service values cause the graph transitions to be deterministic. Upon a change in the environmental conditions of the system (service value conditions or operating conditions), all transitions with the current operating state as their first element are reviewed. Any transitions whose guard (element of D) is not currently valid or whose target specification (second element of T) is unachievable are disallowed. Of the remaining valid transitions, the one whose target specification has the highest relative value under the current service value conditions is chosen, and its target specification is put into operation. The total ordering on relative service values ensures that this choice is deterministic.

Many possible ways of representing this information exist. For our example (Figure 2), we have chosen to depict it graphically because the vertical dimension of the graph maps intuitively to our notion of relative value. A linear textual representation or an explicit transition matrix also would suffice.

- P—*Service Probabilities*

For each member of the set of S, a service probability is specified. For that member of S, the service probability defines the probability that the member of S must meet its dependability requirements, i.e., that analysis has shown

that its operating conditions will be within those specified for normal system behavior.

6. Example Survivability Specification

In this paper, we have used a hypothetical command-and-control system as an example. This section ties together the survivability specification of the example.

The set S contains the following members:

- s_1 *Full Command, Control, and Analysis*. Defines complete and normal functionality, including both central and regional information servers, delivery of crucial information like weather data, transmission of command information, transmission of local status and observational data to regional centers, synthesis of incoming data for the regional and central command centers, and so on.
- s_2 *Low Performance*. During peacetime, more time is available for analysis of intelligence and status information since speed of command information is not as critical. This specification provides full functionality, but with higher latencies.
- s_3 *Regional*. This specification limits service to central command and a single region. Synthesis and analysis of wide-area information are unavailable.
- s_4 *Maximum Alert*. This specification requires that the system operate with no network traffic because it might be compromised. Local processing is continued, and all possible security measures are in place.
- s_5 *Command Only*. This specification limits operation to basic command transmission. Routine data transmission is unavailable and access to servers is unavailable.

Table 1: V for C2 example

	s_1	s_2	s_3	s_4	s_5
d_1	5	4	3	1	2
d_2	5	4	3	1	2
d_3	5	2	4	1	3
d_4	5	2	4	1	3
d_5	5	3	2	1	4
d_6	5	3	2	1	4
d_7	5	1	3	4	2
d_8	5	1	3	4	2
d_9	5	1	3	4	2

Clearly, military value associated with the C2 system varies dramatically over time. Simulations and visualizations of scenarios may be foregone in peacetime, when more time for data analysis is available, but intuitive presentation is crucial during war when decisions must be made quickly. Also, different types of damage may make certain acceptable specifications infeasible; areas of network damage might be dispersed sufficiently that the system cannot continue to provide full service over the entire region, and must go to a position of heightened alert instead. Finally, the geographical spread of a military conflict can affect the services commanders need. In this example, then, E includes the political climate and the security threat status under which the system is operating and the geographical distribution of any present conflict. Which specification provides the most value depends on:

- E: {political climate (c) \mapsto {wartime, peacetime},
 conflict dispersion (d) \mapsto {none, local, dispersed},
 security threat status (s) \mapsto {none, local, distributed}}

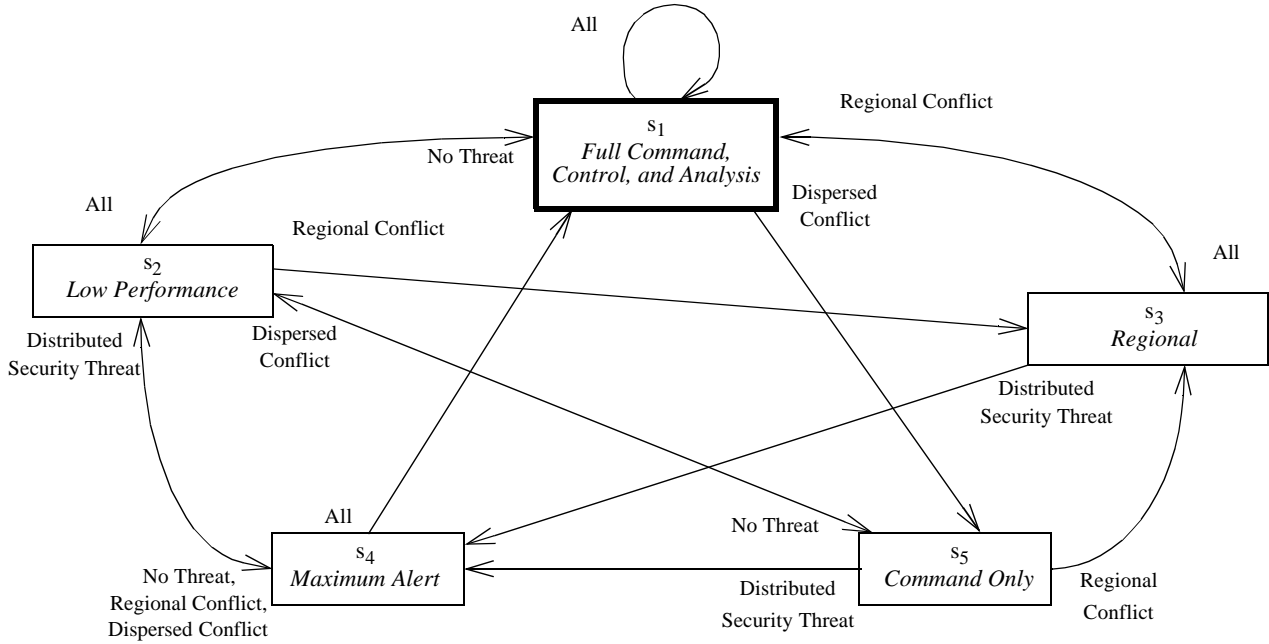


Figure 2. Survivability specification example.

All possible combinations of these are reachable except the ones where peacetime operations have a conflict dispersion other than none, or where wartime operations have a conflict dispersion of none, so that D is:

- $d_1: \{c \mapsto \text{peacetime}, d \mapsto \text{none}, s \mapsto \text{none}\}$
- $d_2: \{c \mapsto \text{peacetime}, d \mapsto \text{none}, s \mapsto \text{local}\}$
- $d_3: \{c \mapsto \text{wartime}, d \mapsto \text{local}, s \mapsto \text{none}\}$
- $d_4: \{c \mapsto \text{wartime}, d \mapsto \text{local}, s \mapsto \text{local}\}$
- $d_5: \{c \mapsto \text{wartime}, d \mapsto \text{dispersed}, s \mapsto \text{none}\}$
- $d_6: \{c \mapsto \text{wartime}, d \mapsto \text{dispersed}, s \mapsto \text{local}\}$
- $d_7: \{c \mapsto \text{peacetime}, d \mapsto \text{none}, s \mapsto \text{distributed}\}$
- $d_8: \{c \mapsto \text{wartime}, d \mapsto \text{local}, s \mapsto \text{distributed}\}$
- $d_9: \{c \mapsto \text{wartime}, d \mapsto \text{dispersed}, s \mapsto \text{distributed}\}$

Table 1 shows V and Figure 2 shows T for this example. In Figure 2, the boxes represent the members of S with which they are labeled, and the arcs represent the transitions in T . Each out-edge is labeled with the member(s) of D under which the transition it represents may be taken. The members are grouped into the following categories for clarity:

- No Threat : all d_i such that political climate = peacetime and security threat status \neq distributed
- Regional Conflict : all d_i such that conflict dispersion = local and security threat status \neq distributed
- Dispersed Conflict : all d_i such that conflict dispersion = dispersed and security threat status \neq distributed

- Distributed Security Threat : all d_i such that security threat status = distributed
- All : all d_i .

Note that this graph is not complete. For example, in some cases, a transition from one form of service to another cannot occur directly because area-wide information was not available in one operating mode but is needed in another. This necessitates transitioning to the *Low Performance* specification briefly to synthesize the information needed to enter the new operating mode.

A reasonable set P for the example is:

$\text{Pr}(\text{Full Command, Control, and Analysis})$	$= 0.9975$
$\text{Pr}(\text{Regional})$	$= 1 - 10^{-4}$
$\text{Pr}(\text{Low Performance})$	$= 1 - 10^{-4}$
$\text{Pr}(\text{Maximum Alert})$	$= 1 - 10^{-6}$
$\text{Pr}(\text{Command Only})$	$= 1 - 10^{-6}$

As an example of how the system might operate, consider first the failure of a major server that occurs in peacetime during normal operations. The server usually performs simulation and visualization activities based on data supplied from all locations served by the system. With the loss of the server these activities will be slowed, but this is acceptable because operations are not as time-critical as they would be during a conflict. The system will transition to *Low Performance* while the system is repaired.

Consider then that an unstable political situation degrades into open warfare. The conflict is localized, but timing requirements are crucial. The system has not yet recovered

from the server failure, so it transitions to the *Regional* service, providing full command and analysis capabilities in that area.

Next, intelligence reports suggest that the hostile regime may have plans to conduct an attack against the system. These reports are credible, and such an attack could provide the enemy with strategic information about friendly forces. The decision is made to transition the system to *Maximum Alert* to prepare it to defend against such an attack.

In this final situation, security of the system is seen as having higher value than complete information: the information provided to commanders will be reduced, but the network is drawn down in such a way that it is hardened against enemy attack. Upon damage recovery or threat removal, any of the above states can transition to any other state that has higher value including the preferred state thereby returning the system to its full functionality.

7. Conclusions and future work

There are many critical information systems upon which both civilian and military critical infrastructure applications rely. Loss of an information system will, in many cases, either reduce the service available from a critical infrastructure application or eliminate it entirely. The existing definitions of survivability, while presenting an intuitive notion of what a survivable system is, do not give specifiers clear guidance on what must be required of a survivable system to ensure that it has all the necessary properties. We claim that the specialized requirements of critical information systems require such a definition. In this paper we have outlined a rigorous definition of survivability and related it to the field of dependability. It now remains to detail precisely how the individual aspects of dependability fit into this framework to complete a definition that can be put into practice and tested against real engineering needs.

Acknowledgments

This work was supported in part by the Defense Advanced Research Projects Agency under grant N66001-00-8945 (SPAWAR), the Air Force Research Laboratory under grant F30602-01-1-0503, the National Science Foundation under grant number CCR-9804078 and ITR-0086003, the National Aeronautics and Space Administration under grant number NAG-1-2290, and Microsoft Corporation. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any of the sponsors. We thank Franklin Webber for his valuable comments on earlier drafts of this paper.

References

- [1] Anderson, T. and P. Lee. *Fault Tolerance: Principles and Practice*. Prentice Hall, Englewood Cliffs, NJ, 1981.
- [2] Armstrong, J.H., *The Railroad: What It Is, What It Does*, Simmons-Boardman, Omaha, NE, 1993.
- [3] Avizienis, A., J. Laprie, and B. Randell, "Fundamental Concepts of Computer System Dependability," IARP/IEEE-RAS Workshop on Robot Dependability: Technological Challenge of Dependable Robots in Human Environments, Seoul, Korea, May 2001.
- [4] Ball, R.E., *The Fundamentals of Aircraft Combat Survivability Analysis and Design*, American Institute of Aeronautics and Astronautics (AIAA), 1985.
- [5] Boehm, B., and K.J. Sullivan, "Software Economics: A Roadmap," in *The Future of Software Engineering*, special volume, A. Finkelstein, Ed., 22nd International Conference on Software Engineering, June, 2000.
- [6] Deutsch, M.S. & Willis, R.R., *Software Quality Engineering: A Total Technical and Management Approach*, Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [7] 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.
- [8] Knight, J., M. Elder, J. Flinn, and P. Marx. "Summaries of Four Critical Infrastructure Systems," Technical Report CS-97-27, Department of Computer Science, University of Virginia, November 1997.
- [9] Myers, J.F., "On Evaluating The Performability Of Degradable Computing Systems", IEEE Trans. Computers, vol. C-29, no. 8, pp. 720-731, August 1980.
- [10] Myers, J.F., W.H. Sanders, "Specification And Construction Of Performability Models" Proceedings: Second International Workshop on Performability Modeling of Computer and Communication Systems, Mont Saint-Michel, France, June 28-30, 1993.
- [11] National Defense Industrial Association Symposium, Proceedings: Aircraft Survivability 2000, Monterey, CA, November 2000.
- [12] 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.
- [13] 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.
- [14] Sullivan, K.J., J.C. Knight, X. Du, and S. Geist, "Information Survivability Control Systems", Proceedings of ICSE 21: Twenty First International Conference on Software Engineering, Los Angeles, CA, May 1999.
- [15] Summers, B.J. (ed.), *The Payment System: Design Management and Supervision*, Int. Monetary Fund, Washington, DC, 1994.
- [16] U.S. Department of Commerce, National Telecommunications and Information Administration, Institute for Telecommunications Services, Federal Standard 1037C.
- [17] <http://www.cert.org/research/isw.html>
- [18] <http://www.aircraft-survivability.com/>