Chapter 2

Reliability and Availability

The two concepts reliability and availability are talked about, written about, equated with each other, and given star status but, in the main, remain somewhat one-dimensional concepts. In this chapter, and throughout this book, I hope to show that these concepts, particularly availability, have other dimensions and interpretations as well.

Introduction to Reliability, Availability, and Serviceability

Reliability represents the probability of components, parts, and systems to perform their required functions for a desired period of time, without failure, in specified environments with desired confidence. Reliability, in itself, does not account for any repair actions that may take place. Reliability accounts for the time it will take the component, part, or system to fail while it is operating. Reliability does not indicate how long it will take to get the unit under repair back into working condition and is often qualitative—*this car is much more reliable than my last one*. This assessment is usually based on how long it spent in the repair and servicing states.

Availability, to continue with the car analogy, is how long the car is in working order and how long you can travel before it fails, measured in operational and usage times, respectively (see Figure 2.2). These you will mentally note as the time between failures and how long it takes you to get the car back on the road.

> "Reliability, availability, and serviceability (RAS) is a computer hardware engineering term. The phrase was originally used by IBM as a term to describe the robustness of their mainframe computers. The concept is often known by the acronym RAS." See Wikipedia.

IBM did not *invent* RAS, but they put heavy emphasis on it in System/370 in 1970. They put robust features into succeeding mainframes to increase their ability to stay operational and then coined the acronym *RAS* for the outcome. RAS (reliability, availability, and serviceability) is evolving and the concept has been adopted by all hardware vendors and has now spread to the software domain.

Computers designed with higher levels of RAS have a multitude of features that protect data integrity and help them stay available for long periods of time without failure—this data integrity and uptime is a particular selling point for mainframes and fault-tolerant systems, despite their being more expensive.

In fact, reliability and availability characteristics could be developed for liveware.

Note: Reliability and availability do not have a one-to-one or a mathematical relationship that is universal (like Ohm's law, $V = IR$, or Einstein's $E = mc^2$). It is perfectly possible for component 1 to be less reliable than component 2 but deliver better availability. This scenario might occur when failures of 1 can be diagnosed and repaired much more quickly than those of 2 so that overall, the total outage time of 1 can be less than that of 2. Hence, the availability of 1, A%, is greater than that of 2, B%, even though reliability of 1, R_1, is lower than that of 2, R_2. Another way that less reliable components can "win" the availability contest is by using redundancy (duplicate components), like the original redundant arrays of independent disks (RAID) designs, but redundancy comes at a dollar cost.

See Availability Architectures in Chapter 4 for a discussion of RAS features.

Before we get into the body of this book, I'll outline some major areas that we will deal with in Chapters 6 and 7—a true view from 30,000 feet, part of the process of *learning via different viewpoints and angles.*

RAS Moves Beyond Hardware

An Intel document dated 2005 lists the following as factors that help RAS as well as pure hardware:

- *Highly reliable platforms:* Hardware, firmware, and software
- *Extensive hardware and software testing:* Vendor and user
- *Rigorous change management:* Supports my (and others') ideas that *volatility* can mean *outage*
- *Redundant architectures:* Used only where needed
- *Highly trained staff:* Liveware issues loom large in outages, as we will see
- *Well-established emergency procedures:* Runbooks, day-to-day and disaster recovery (DR)

We now need to add at least some mention of *security*, which can affect availability significantly these days, as do the other elements above; see http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/reliability-availability-and-serviceability-for-the-always-on-enterprise-paper.pdf. This excellent paper contains a RAS table and a detailed glossary of RAS terms. Our RAS discussion in Chapter 3 of this book is generic and does not imply that all vendors have all the features mentioned in that chapter in their RAS environment.

Aside: I was reading an article about these things and its author, who is obviously knowledgeable, issued the *so what* comment about RAS features. In my mind, the *so what* is that the customer would like to know they are there, rather like knowing that there are safety features and procedures on aircraft without understanding them in detail. Basic RAS is the best platform to build on.

Availability: An Overview

Reliability is denoted mathematically as $R(t)$, a function of time described by a graph, which we will see shortly. At some time in the life of an item, its reliability may be assumed to be constant, to allow calculations to be made as we wallow at the bottom of the ubiquitous *bathtub* curve (see Chapter 5 under Availability Concepts).

Some Definitions

These definitions may seem superfluous but they often come in useful when we discuss with people about system availability—often boring but a good *anchor point* here to start a discussion of an important topic. Getting availability and reliability terms straight up front is akin to saying "according to Hoyle" when starting a game of cards. It prevents arguments. In the definitions that follow, an *item* may be a circuit board, a component thereof, a LAN, or even a whole server configuration and its associated network(s).

IEEE*[*]*: *Reliability*. The ability of a system or component to perform its required functions under stated conditions for a specified period of time.

IEEE: *Availability*. The degree to which a system or component is operational and accessible when required for use.

ITU-T*[†]*: *Reliability*. The ability of an item to perform a required function under given conditions for a given time period.

ITU-T: *Availability*. The ability of an item to be in a state to perform a required function at a given instant of time or at any instant of time within a given time interval, assuming that external resources, if required, are provided.

NIST FIPS: *Availability*. This refers to "ensuring timely and reliable access to and use of information…" (44 U.S.C., Sec. 3542). A loss of availability is the disruption of access to or use of information or an information system.

> ***Note:*** The definitions here point to *service* as the key entity in the definition of *availability*, the main thesis of this book.

Shorter Oxford English Dictionary, 1988: *Reliability*. The quality of being reliable; reliableness.

Reliable. That may be relied upon, in which reliance or confidence may be put, trustworthy, safe, and sure.

These are all qualitative definitions. None of them, although comprehensible, implies any *quantitative properties* related to these two concepts.

[*] IEEE (Institute of Electrical and Electronics Engineers) 1990. *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*. New York, NY. http://en.wikipedia.org/wiki/Special:BookSources/1559370793.

[†] International Telecommunications Union, recommendations E800. https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-E.800-198811-S!!PDF-E&type=items.

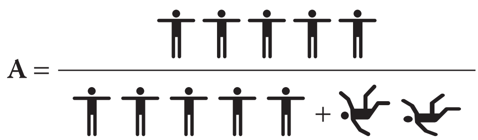
Quantitative Availability

Reliability of an *item* can be quantified by an equation (see Appendix 4) and is a function of MTBF,* whereas availability is not a fixed function of the *item* but varies depending on the reliability of the *item(s)* and the time taken to diagnose a problem and correct it. It is a function of MTBF and other parameters.

For the moment, let us take the number **A** representing availability as

$$\frac{\text{Time working}}{\text{Time working} + \text{Time not working}}$$

The plethora of definitions in the topic of reliability and availability can be confusing (as we will see) but the following diagram may ease the pain:



(2.1)

A Primitive Availability Equation

> ***Aside:*** This visual representation reminds me of people of the Zulu and Navajo nations, where there is no written language—there may be others I don't know about. The Navajo language was used by the U.S. military intelligence in World War II for encoding messages in their Pacific operations, knowing the enemy couldn't possibly have had a hardcopy Navajo dictionary.

As we shall see, availability represents the probability that the system is capable of conducting its required function when it is called upon given that it has not failed or is undergoing a repair or an update action. Therefore, not only is availability a function of reliability, but it is also a function of the reparability, maintainability, or serviceability.

Figure 2.1 illustrates the factors that affect the definition of a system and its availability, that is, functioning according to its specification for a period of time. It should be noted here that there is a difference between the hours a system is actually working (switched on, lights flashing) and the hours it should be available to users, that is, fully supported, all relevant hardware and software initialized and all lights blazing.

Availability: 7 R's (SNIA)†

Mark Fleming (employed at IBM in 2008, the date of his presentation) lists 7 R's that impact availability:

Redundancy: To eliminate single points of failure (SPoFs).
Reputation: What is the track record of the key suppliers in your solution?

---

* Mean time between failures, covered in detail later along with its confusing companion, MTTF, mean time to failure. Don't worry, we'll get there by the end of Appendix 2.

† You can find the presentation on the SNIA website (www.snia.org).

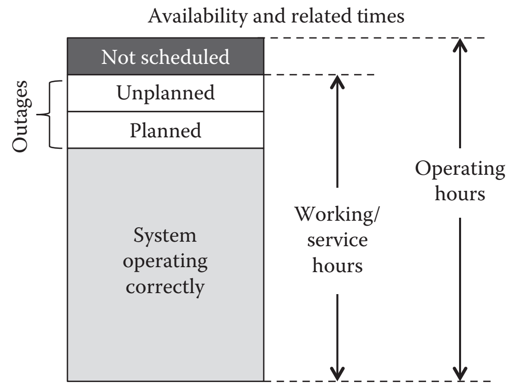


Figure 2.1 Visual availability: Definition of times.

Reliability: How dependable are the components and coding of the products?
Reparability: How quickly and easily can suppliers fix or replace failing parts?
Recoverability: Can your solution overcome a momentary failure without impacting the end users?
Responsiveness: A sense of urgency is essential in all aspects of high availability (HA).
Robustness: Can the solution survive a variety of forces working against it?

This same list can be found in Rick Schiesser's book *IT Systems Management* (2002)* so I am unsure of the true origin of the 7 R's discussion. It is the message that matters.

I put this in here because it is a useful memory jogger and even if you can't name all the components, at least you know you have seven to find!

With due respect to Mark Fleming, storage networking industry association (SNIA), and Rick Schiesser, I would add to and expand upon a few of their R's:

Resilience: Add as a measure of the ability and speed of recovery (*a la* the resilient boxer, who is able to get up and resume the fight where he left off after a knockdown).
Reputation: To Mark Fleming's words above in this category, I would add your own organization's reputation in avoiding commercial failures that impact existing customers and website failures that deter would-be customers.
Reparability: How soon can you complete problem determination (what and where) to decide which supplier or third party to ring if needed? In addition, for customer-replaceable field replaceable units (FRUs), what stock do you keep onsite? A few hours' driving to fetch an FRU will blow availability targets into the middle of the following week.
Return on investment (ROI): You will probably be asked to justify any extra expenditure or, put another way, say what will it cost the organization if we *don't* put this scheme in place and we get caught out?

On the last point, about ROI, 25 years ago, a senior information technology (IT) manager at a customer I looked after had a grand plan for a *new data center*. He was warned by an experienced external data center person that he should prepare a *cost case* when he presented his grand plan to the board as they would probably ask for such a thing. He didn't and they did!

* *IT Systems Management*, Prentice Hall PTR, ISBN 0-13-087678-X.

The situation then rebounded, however: he rang me and asked me to help prepare a cost case! Panic! I called around IBM data center personnel and, with their help, managed to put together a cost case involving productivity (IT staff/unit of power), software, and other savings associated with centralization and modernization.

I used the costs of the support staff per unit of power (the IBM RPP—relative processor power, then the 370/158)—IBM had 0.8 people/RPP, the customer at that time 1.8 per RPP. IBM was aiming for 0.2 people per RPP and I think we used the 0.8 figure as the benchmark for the customer's potential savings!

Someone in your organization will ask the very same question about *your* grand plans—HA, DR, or any other. You will need some sort of financial case involving total cost of ownership (TCO) (see Appendix 3) and benefits.