Software Reliability and System Reliability

Steven J Zeil
Old Dominion Univ.
Spring 2012

Outline

1 Introduction

2 Dependability

3 Failure Behavior of an X-ware System
   - Atomic
     - Reliability
     - Failure Rates and Hazard Functions
     - Reliability and the Hazard Rate
     - Discrete, Independent Failures
   - Systems made up of components
     - The Single Interpreter
     - Multiple Interpreters

Theme

“by using deliberately simple mathematics, the classical reliability theory can be extended in order to be interpreted from both hardware and software viewpoints”
Introduction Dependability Failure Behavior of an X-ware System

Outline

1. Introduction
2. Dependability
3. Failure Behavior of an X-ware System

Basic Definitions

*Dependability* is defined as the trustworthiness of a computer system such that reliance can justifiably be placed on the service it delivers.

Attributes:
- availability
- reliability
- safety
- confidentiality
- integrity
- maintainability

Impairments and Means

Impairments:
- faults
- failures
- errors

Means
- fault preventions
- fault removal
- fault tolerance
- fault forecasting

Failure Classification

- Domain
  - Value
  - Timing
- Perception
  - Consistent
  - Inconsistent
- Consequences
  - benign . . . catastrophic
Introduction

1 Introduction

Dependability

2 Dependability

Failure Behavior of an X-ware System

3 Failure Behavior of an X-ware System
   - Atomic
   - Reliability
   - Failure Rates and Hazard Functions
   - Reliability and the Hazard Rate
   - Discrete, Independent Failures
   - Systems made up of components
     - The Single Interpreter
     - Multiple Interpreters

Outline

Introduction Dependability Failure Behavior of an X-ware System

Time to Failure

The key random variable is the time to failure, $T$. Denote the probability that the time to failure $T$ is in some interval $(t, t + \Delta t)$ as

$$P(t \leq T \leq t + \Delta t)$$

Given the cdf $F(T)$ and pdf $f(T)$,

$$P(t \leq T \leq t + \Delta t) = F(t + \Delta t) - F(t) \approx f(t)\Delta t$$

Reliability Function

$$F(t) = P(0 \leq T \leq t) = \int_0^t f(x)dx$$

The reliability function is the probability of success at time $t$ (i.e., the prob. that the time to failure exceeds $t$)

$$R(t) = P(T > t) = 1 - F(t) = \int_t^\infty f(x)dx$$

Failure Rate

The failure rate is the probability that a failure per unit time occurs in the interval $[t, t + \Delta t]$, given that a failure has not occurred before $t$.

$$\text{Failure rate} \equiv \frac{P(t \leq T < t + \Delta t | T > t)}{\Delta t}$$

$$= \frac{P(t \leq T < t + \Delta t)}{(\Delta t)P(T > t)}$$

$$= \frac{F(t + \Delta t) - F(t)}{(\Delta t)R(t)}$$

Failure rate
   - measurable
   - easier to understand than the prob. density function
**Hazard Rate**

The *hazard rate* is defined as the limit of the failure rate as the interval $\Delta t$ approaches zero.

$$z(t) = \lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{(\Delta t)R(t)} = \frac{f(t)}{R(t)}$$

The hazard rate is an instantaneous rate of failure at time $t$, given that the system survives up to $t$. $z(t)dt$ represents the probability that a system of age $t$ will fail in the small interval $t$ to $t + dt$.

**Converting**

$$z(t) = \frac{f(t)}{R(t)} = \frac{dF(t)}{dt} \frac{1}{R(t)}$$

$$\frac{dF(t)}{dt} = -\frac{R(t)}{R(t)}$$

Combining gives

$$\frac{dR(t)}{R(t)} = -z(t)dt$$

Integrate both sides w.r.t. $t$:

$$\ln R(t) = -\int_0^t z(x)dx + c$$

Because $R(0) = 1$, $c = 0$

Exponentiate both sides:

$$R(t) = \exp \left[ -\int_0^t z(x)dx \right]$$

or, differentiating

$$f(t) = z(t) \exp \left[ -\int_0^t z(x)dx \right]$$
Single failure

Suppose we measure time in terms of # of discrete inputs. Let $p$ be the prob of failure on a given test input given that no prior failure has occurred on prior inputs. If all failure domain inputs are independent

$$R(k) = (1 - p)^k$$

Let $t_e$ be time required to execute one test case.

$$t = kt_e$$

Execution Duration

Now, assume that there is a finite limit for $p/t_e$ as $t_e$ becomes vanishingly small

$$\lambda = \lim_{t_e \to 0} \frac{p}{t_e}$$

$$R(t) = \lim_{t_e \to 0} (1 - p(t_e))^{t/t_e} = \exp(-\lambda t)$$

which is the exponential distribution

Markov Chain Model

Better known approach is dismissed in one paragraph

- pipelines?
- Markov approach
  - We should look at one of these later

Hierarchical Structures

Systems can be decomposed into subsystems forming a hierarchy of

- function calls
  - Might not be a tree
  - might not form clean layers
- levels of abstraction
  - Here called “interpreters”
Consider an application built on $C$ components (e.g., ADTs)

- Each component $C_i$ has a failure rate $\lambda_i$
- The entire collection of components can be in any of $S$ valid states.
  - Presumably each component has some number of discrete states, so $S$ is the power set of all component states.
- Add an $S+1$st state to represent a failure state.
  - This state is an absorber/terminal state

**Component States**

Can components be well modeled by discrete states?
- Can failures be modeled as a state change?
  - e.g., Consider a numeric calculation that is supposed to be within $\pm 0.01$ of an ideal solution but that is $\pm 0.1$ for selected input values. Is that a state of simply a function of the inputs?
  - In the example above, what are the implications of the failure state being terminal
    - if the interpreter fails because of the error?
    - if the interpreter recovers from the error?

**State Transitions**

The collection of components has its own set of transition properties

- $\gamma_j$ is prob that a component in state $j$ stays in state $j$
  - $1/\gamma_j$ is mean sojourn time in state $j$
- $q_{jk} \equiv$ prob that system in state $j$ will make a transition to state $k$

$$\sum_{k=1}^{S} q_{jk} = 1$$

**System Failure Rate**

“A system failure is caused by the failure of any of its components. The system failure rate $\xi_j$ in state $j$ is thus the sum of the failure rates of the components under execution in this state, denoted by

$$\xi_j = \sum_{i=1}^{C} \delta_{ij} \lambda_i$$

where

$$\delta_{ij} = \begin{cases} 1 & \text{if } C_i \text{ is currently in state } j \\ 0 & \text{ow} \end{cases}$$
**Can We Just Add Up Failure Rates?**

- A very common practice
- Suppose two components fail independently with rate $\lambda_1$ and $\lambda_2$.
  - Then the rate of coincident failure would be $\lambda_1\lambda_2$.
  - If the $\lambda_i \ll 1$, then $\lambda_1\lambda_2 \ll \lambda_i$ and why would we even bother with systems where the failure rate was not very small?
- On the other hand, if we are dealing with long time periods and/or require extreme reliability, these can add multiplies of many orders of magnitude that bring $\lambda_1\lambda_2$ back into significance
- And there is substantial evidence that failures are not independent
- It is also known that faults can hide or magnify one another

**Small Failure Rates**

“A natural assumption is that the failure rates are small with respect to the rates governing the transitions from the execution process or, equivalently, that a large number of transitions resulting from the execution process will take place before the occurrence of a failure — a system that would not satisfy this assumption would be of little interest in practice. This assumption is expressed as follows: $\gamma_j \gg \xi_j$”

- I’m not sure I believe this either.
  - Counter-evidence: Hoppa, Mitchell

**System Failure Rate**

$$\lambda(t) \equiv \lim_{dt \to 0} \frac{1}{dt} P\{\text{first failure occurs between } t \text{ and } t + dt\}$$

Let $P_j(t)$ be prob that the system is in state $j$

$$\lambda(t) = \frac{\sum_{j=1}^{S} \xi_j P_j(t)}{\sum_{j=1}^{S} P_j(t)}$$
Equilibrium

If $\gamma_j \gg \xi_j$, then we execute a long time before failure. So, ignoring failures, we can solve for the equilibrium probabilities $\vec{\alpha}$

$$\vec{\alpha} \cdot \mathcal{A}' = 0$$

So $P_j(t)$ converges to $\vec{\alpha}_j$, and

$$\lambda(t) = \frac{\sum_{j=1}^S \xi_j P_j(t)}{\sum_{j=1}^S P_j(t)} = \sum_{j=1}^S \vec{\alpha}_j \xi_j$$

So the failure rate of the entire system becomes the weighted average of the failure rate of its components, weighted by the relative time spent executing each component.

Component Execution Time

$$\lambda(t) = \sum_{j=1}^S \vec{\alpha}_j \xi_j = \sum_{j=1}^S \vec{\alpha}_j \sum_{i=1}^C \delta_{ij} \lambda_i = \sum_{i=1}^C \lambda_i \sum_{j=1}^S \delta_{ij} \vec{\alpha}_j$$

Let $\pi_i = \sum_{j=1}^S \delta_{ij} \vec{\alpha}_j$ (average portion of time when component $i$ is executing)

$$0 \leq \sum_{i=1}^C \pi_i \leq C$$

Multiple Interpreters

Although the text goes on to derive this case separately, I fail to see anything in the above discussion that actually depends on the number of layers of abstraction. The “interpreter” was, I presume, just one of the $C$ components.

- If not, the derivation completely neglected the possibility of a failure of the top-level application even when the lower-components were OK.