J Reliable Intell Environ (2015) 1:23–32
DOI 10.1007/s40860-015-0004-4

REVIEW

The application of statistical reliability theory in the context of Intelligent Environments: a tutorial review

Gordon Hunter[1]

Received: 21 January 2015 / Accepted: 31 March 2015 / Published online: 16 June 2015

Abstract Intelligent Environments often require the integration of multi-modal sensing and actuating technologies with high performance real-time computation, including artificial intelligence systems for analysis, learning patterns and reasoning. Such systems may be complex, and involve multiple components. However, in order to make them affordable, Intelligent Environments sometimes require many of their individual components to be low-cost. Nevertheless, in many applications—including safety-critical systems, and systems monitoring the health and well-being of vulnerable individuals, it is essential that these Intelligent Environment systems are reliable, which the issue of affordability must not compromise. If such environments are to find real application and deployment in these types of domain, it is necessary to be able to obtain accurate predictions of how probable any potential failure of the system is in any given timeframe, and of statistical parameters regarding the expected time to the first, or between successive, failures. Such quantities must be kept within what are deemed to be acceptable tolerances if the Intelligent Environment is to be suitable for applications in these critical areas, without requiring excessively high levels of human monitoring and/or intervention. In this paper, an introductory overview of statistical reliability theory is presented. The applicability of this to the context of Intelligent Environments—particularly those involving safety critical or other sensitive issues—is discussed, along with how such reliability modelling can be used to influence the design, implementation and application of an Intelligent Environment.

✉ Gordon Hunter
G.Hunter@kingston.ac.uk

[1] Kingston University, London, UK

Keywords Statistical reliability theory · Probability · System modelling

1 Introduction

The theme of "Intelligent Environments" (IEs) or "Smart Environments" has been much researched over the last 10–20 years [1,33], and the paradigm combines multi-modal sensing and proactive assistive technology with machine learning and AI, often leading to complex multi-component systems. It has found prototype applications in many domains—for example, ambient assisted living [2,3,15], dialogue systems [4], environmental monitoring [5], smart homes [6], offices [7], transport systems [8], and even smart cities [9]. The theme of this new journal is "Reliable Intelligent Environments". However, with the exception of a few studies which have focused-in on the reliability of some individual aspects [8,10] and a number of empirical [11] or simulation [12,13] studies, very little attention appears to have been given to investigating the reliability of such IEs. Such reliability could be essential in safety-critical applications, where the well-being of people [2,3,14,15], animals and/or the natural environment [5] could be at risk in cases of the system failing. Although a distributed system may sometimes display "graceful degradation" when a small number of its components fail, for example through self-reconfiguration or self-healing [1], models of how likely failures of components are, and how serious these are for the performance of the entire system, will still be highly valuable. Some authors have questioned whether a statistically-based risk analysis is appropriate for safety-critical systems: Can any system which has even the smallest risk of failure be acceptable as a "safety-critical" system? However, in reality, there is no such thing as a completely risk free, or absolutely

reliable, system. Indeed, safety and reliability are somewhat related concepts—"reliability" indicates an assessment of probability of a system failing, whereas "safety" relates to assessing the consequences of such a failure, and mitigating against them [34]. A system—even a so-called "safety-critical" one—could be regarded as "safe for purpose" if the expected time before it fails greatly exceeds its expected total duration of use, or if the consequences of such a failure are not at all serious. A much more detailed discussion of reliability theory in the context of safety-critical and similar systems can be found in [35–37]. The primary aim of this paper is to raise awareness of this type of approach.

In this paper, I give a review of the well-established field of statistical reliability theory, and indicate how it could be relevant to the reliability of IEs. This is only intended as a brief introduction to the field, and the reader is referred to existing texts for a more comprehensive coverage of the theory. Although the theory, and its discussion here, will primarily be discussed in the context of hardware reliability, it can also be applied to software and network failures, and indeed part of the following section will discuss the issue of software reliability and defects. Although the applicability of reliability theory to software systems is less well-known than its relevance to physical and hardware systems, software reliability has been modelled statistically since the early 1970s [38]. The reader is referred to [39,40], and the comprehensive review paper by Lyu [38], for examples of this.

The remainder of this paper is organized as follows: the next section provides a review of related literature, discussing some issues of reliability in hardware and software systems, followed by a section illustrating how a systems modelling approach to viewing a complex system such as an Intelligent Environment can lead to a perspective which allows the statistical or probabilistic modelling of its reliability. Section 4 will first introduce some important terminology (Sect. 4.1), followed by an introduction (in Sect. 4.2), to various probabilistic models of reliability. Section 5 discusses how these concepts can be applied to models of Intelligent Environments, with examples, and the paper ends with my conclusions in Sect. 6.

2 Related work

As noted above, relatively little work appears to have been carried out on applying reliability theory to the context of Intelligent Environments. However, there have been some studies which have applied some concepts from this approach to empirical studies of reliability in intelligent systems. Suh et al. [8] address the problem of possible component failure within an intelligent transportation system by adding redundancy to the system through the replication of critical objects, enhancing fault tolerance by avoiding the existence of "simple points of failure" within the system. They compute the "availability" of such a system for fulfilling its purpose as the ratio of its mean time to failure [MTTF, also known as the mean time between failures (MTBF)] to the sum of its MTTF and its mean time to repair (MTTR, i.e. the mean time required to repair the system after a failure). They noted the importance of Brewer's CAP conjecture (or CAP theorem) [16,17], namely that any distributed system can at best simultaneously provide two, but not three, out of consistency (all nodes of the system can access the same data at the same time), availability (failures of individual components do not prevent the operation of remaining components) and partition (fault) tolerance (loss of an arbitrary amount of shared information does not prevent the system from operating). Poledna et al. [10] investigated the predictability of the quality of service in real-time systems with replicated components and flexible scheduling schemes.

Augusto and McCullagh [14] discuss safety considerations, and the implications of not taking these sufficiently into account, in the development of Intelligent Environments, but do not explicitly apply reliability theory to their study. Zhang et al. [11] consider the reliability of location detection in IEs, noting sensor failure, range, location and interference as possible sources of problems. However, again, they do not actually employ any analysis based on reliability theory.

Rather more theoretical work has been carried out on modelling the reliability (or lack of it) of complex software systems. It has been noted that the complexity of the software required to govern IEs can develop complex, unexpected interactions which can lead to instability in the behavior of the system [1,18]. Lipow [19] and Gaffney [20] undertook pioneering work to estimate the number of faults present in a large piece of program code, and this was generalized to predicting and controlling the prevalence and effects of software defects by Compton and Withrow [21]. These studies have been extensively extended by Hatton [22–24] who carried out empirical studies on 123 software packages (55 written in Fortran and 68 in C) covering over 40 different application domains, and also produced a highly general model of the distribution of software defects in terms of the size of the software components, using principles from statistical physics. Although formal methods for "program proving", such as the Z specification method [25–27] can be used to verify the reliability of simple software systems, this is not normally practical for highly complex systems. Similarly, although conventional software engineering practice quite correctly advocates clear specification of requirements, careful design and implementation and rigorous testing [1,28,29], for highly complex systems, possibly involving many thousands of lines of program code, with many conditional or decision statements and highly numerous potential paths through the code, testing to allow for all possible situations which might occur during the entire lifes-

pan of the system being used in application becomes totally impracticable [23]. Hatton [30] has noted that, at best, software systems tend to exhibit approximately one fault per 10,000 lines of code, measured over the working lifetime of the program, but at worst the fault rate may be up to 100 times this—namely one fault per 100 lines of code! Furthermore, even code written in "standardised" forms of programming languages may contain ambiguities—the ISO specification for C lists no fewer than 119 constructions of uncertain definition [23]! Hatton also notes that different types of software defects have widely varying probabilities of resulting in failure, and are ranked by severity according to these probabilities, over the working lifespan of the software system of interest [23].

Bearing in mind the above discussions, and the intrinsic complexity of the hardware and software systems, and their interactions, required to construct and maintain a typical IE, plus the previously highlighted need for safety in and reliability of IEs, emphasizes the necessity for better theoretical understanding and modelling of reliability of IE components and their interdependencies. This motivates the rationale for the remainder of this paper.

3 Systems modelling perspectives of Intelligent Environments

Before moving on to the statistical theory of reliability, it may be instructive and useful to consider some examples of how a complex system such as an Intelligent Environment could be modelled statistically or probabilistically. Crucial to many approaches to the modelling of complex systems are the concepts of the *system*, its constituent *units* or *modules*, their *sub-units*, and ultimately individual *components*. These can be considered as being organized in a hierarchical tree-like structure, with the entire system being made up of the units or modules, and each of those comprising sub-units, with sub-units in turn having sub-sub-units, continuing on recursively until we reach the level of individual simple components. The reliability of a particular type of component can be measured using experimental testing—possibly to destruction or absolute failure, where necessary—of a sample of those components as manufactured. Although individual examples of such a component will vary in their properties—including in their times to or risks of failure—the use of sampling theory (briefly discussed later in this paper) can give confidence limits on the actual population distribution parameters for all such components manufactured in the same way, based on the measurements of those parameters made on a relatively small sample of those components.

Consider the example of a smart home, possibly occupied by an elderly or infirm person. The entire intelligent system making the home a smart home is composed of a wide variety of things—various sensors, actuators, domestic appliances, and computer-based systems. Taking the systems modelling approach, we can identify the primary level units or modules: the home will have an overall supervisory monitoring and control module—based on some kind of computer—plus "peripheral" modules, each fulfilling a particular purpose in terms of data acquisition, data analysis and feature extraction, and taking particular actions. For example, there may be a video surveillance module, an audio surveillance module, a movement sensing and analysis module, a temperature monitoring module, a smell/smoke monitoring module, a module controlling the heating and air conditioning, another controlling the lighting, one to keep the resident informed of the current situation and any causes for concern and another to raise alarm in case of fire or other potentially serious accident. Each of these modules can itself be made up of sub-modules: for example, the video surveillance module would comprise one or more video cameras, plus hardware and software to extract useful features from the video sequences, analyse these (ideally in real time or close to real time), combine information from different camera views (if appropriate), make inferences from what has been captured on camera, and report to the "master" supervisory monitoring and control module. In turn, each sub-module, for example, a video camera, is made up of sub-sub modules. In the case of the video camera, these include the focusing system (comprising lenses, and probably mechanical and/or electronic systems for adjusting the focus), the optical sensor array, a timing system, and so on, with each sub-module being made up of simpler components. The role of the "master" module is to combine evidence from the various modules, make decisions based on these, and give instructions back to the individual modules regarding what actions should be taken. We can either take a "top down" approach by breaking the whole system down into to smaller and smaller sub-units, or a "bottom up" approach considering individual components, then using these to build-up more and more complex units, until we eventually obtain the entire complex system. By modelling the system at appropriate levels, it becomes possible to investigate its reliability, or that of its modules or components, statistically.

4 Statistical reliability theory

The previous section illustrated how a complex system such as an IE can be viewed as being made up of modules and components. Once the modules, sub-modules or components being considered are sufficiently simple, aspects of their reliability can be modelled statistically. Results for components can then be combined to give predictions for the reliability of the simpler modules they comprise, and the reliability of more complex modules predicted on the basis of the sub-

modules of which they are formed. In this way, working from the simplest sub-units upwards, inferences can be made regarding the reliability of the complex system (in our case, the entire IE) as a whole. This section will introduce and review the main aspects of statistical reliability theory. As noted previously, this is a major topic in its own right, and this article can only provide the briefest of introductions. The reader is referred to a standard text such as [31] or [35] for a more detailed discussion of the topic.

4.1 Fundamental concepts and terminology

We can consider reliability at the level of individual (software or hardware) components, modules (collections of closely-related components, such as functions or procedures, "agents" or hardware units), or entire systems. Issues such as the MTTF—or (almost) equivalently, the MTBF following the system being repaired—are clearly of importance (the MTTF and MTBF are identical if it can be assumed that the reliability of the system, module or component does not change after that item has been repaired following any failure). The failure rate of an item is the mean number of failures of that type of item per unit time, which is the reciprocal of the MTBF. The MTTR for an item is the average time required to repair it, restoring it to its normal working state, following a failure. For simplicity, in this paper we shall not distinguish between different types of failure at the individual component level —we effectively assume that any given component always fails in the same way, with identical consequences for that component. However, the nature of failures at the module or system level will still depend on which individual components have failed. The availability, A_i, of an item i, is the fraction of time for which the item is functioning normally, and is defined as:

$$A_i = \mathrm{MTTF}/(\mathrm{MTTF} + \mathrm{MTTR})$$

or equivalently with MTBF replacing MTTF.

4.2 Simple probabilistic models of component and system reliability

In this section, we discuss some relatively simple models of reliability, based on elementary probability theory. These models each make a variety of different sets of assumptions, and each provides good approximations for a range of different situations appropriate to the modelling of IEs. However, no one of these models is appropriate to every IE scenario, and it is necessary to first analyse the system of interest before selecting what type of model is most suitable, making assumptions and approximations relevant to that particular system.

4.2.1 Discrete probabilistic models

Let us initially assume that any single given component has a known probability of failing in a specified unit interval of time. By "component", we could be referring to an item of hardware, or software, or network technology, as appropriate. Such a probability could be estimated from past experience—for example, if we had observed 5 failures out of 100 examples of these components whilst operating continuously over a 4-year period, we could estimate the failure probability as $5/(4\times 100) = 0.0125$ per component per year. Alternatively, the estimates could be based on the results experiments where a sample of components of that type were tested to destruction or complete failure, leading to an estimate of their MTTF and hence to an estimate of their probability of failure in a unit time interval. In cases of more complex systems, failure probabilities could be estimated by Monte Carlo simulation [41]. We do not necessarily assume that this probability of failure is the same across different types of components, but do assume that the failure probability is the same for all identical components, and that these component failure probabilities are constant over time. We further assume that failures of individual components are independent—that is, the failure of one given component does not influence whether any other component fails or not. However, failure (or otherwise) of constituent components clearly does influence continuing function or failure of modules or systems containing those components.

Let us consider two basic types of ways in which components interact. The first, which we shall call "series" interactions—in analogy with electrical circuit theory—is where the input to one component, which we shall call B, directly depends on the output of another, called A (see Fig. 1). In this case, even if B is capable of functioning normally, it cannot do anything useful unless A is also functioning correctly. If either A or B fails (or both fail), then the "series combination" of A and B effectively fails. The combination can be considered as a "super-component" or "mini-module", and combinations of such super-components can be considered in a recursive manner.

In such a series combination, let the probability of A failing in a unit time interval be p_A and the probability of B failing in the same interval be p_B. Note that these are the probabilities of the components becoming incapable of

Fig. 1 "Series" combination of two components A and B, where the ability of B to do anything useful relies on the correct functioning of A. The combination only functions correctly if both A and B are functioning normally. The *arrows* indicate the sequence of any dependency, e.g. of data flow

normal function, which we assume to be independent of each other, in contrast to be able to do anything useful, which clearly does depend on the other component functioning correctly. The corresponding probabilities of normal function are $(1 - p_A)$ for A and $(1 - p_B)$ for B. For the series combination of A and B to function normally, both A and B must be functioning, and the probability of this occurring, under our assumptions, is $(1 - p_A)(1 - p_B)$. Thus, the probability of the series combination failing is $p_{AB} = 1 - (1 - p_A)(1 - p_B) = p_A + p_B - p_A p_B$.

The MTTF for such a combination is then $1/p_{AB}$. Since each of p_A and p_B are in the range 0 to 1, the failure probability of the combination exceeds that of either component, and the mean time to failure is therefore lower than the smaller of the those of the individual components, in agreement with our intuition. A chain cannot be stronger than its weakest link.

Now let us instead consider a situation where components C and D operate independently—that is, each one does not require anything to be done by the other in order to complete its role successfully (Fig. 2). Again in analogy with electrical circuit theory, we refer to this as a "parallel" combination of components. In this situation, we assume that other components (external to the combination) only depend on at least one of C and D functioning normally. This would be the case if C and D replicate the same job, i.e. there is redundancy within the combination. We note that such a combination only fails if both C and D fail. If the probability of C failing is p_C and that of D failing is p_D, then the probability of the combination failing is $p_{C+D} = p_C p_D$, and so the probability of the combination functioning normally is $1 - p_{C+D} = 1 - p_C p_D$. The MTTF of this type of combination is therefore $1/p_{C+D} = (1/p_C)(1/p_D)$, i.e. the product of the individual mean times to failure. Since no probabilities can exceed one, the probability of the combination failing is lower than that of either component failing, and the MTTF of the parallel combination exceeds that of either component. Both of these are consistent with intuition. Two chains tethering the same pair of objects are stronger than either one chain on its own (Fig. 2).

The above concepts of replacing pairs of components, connected in "series" or in "parallel" as appropriate, can be applied recursively to incorporate larger numbers of components, building-up "modules" of several components, and eventually entire systems.

Example Figure 3 below shows a combination of four components A, B, C, D. These could be simple components, such as transistors, more complex physical systems—such as the constituent sub-modules of a video camera, as mentioned previously—or elements of a software system. Suppose, in a given year, individually, A has a probability 0.08 (or 8 %) of failing, B has a probability 0.15 (15 % chance) of failing, C a 0.05 (5 %) failure probability and D 0.07 (7 %) probability of failure. The combination continues to function if either (or both) of B and C function, provided A and D also both function. Otherwise, the combination fails.

In this example, the parallel combination of B and C only fails if both B and C fail. This occurs with probability $p_{B+C} = p_B p_C = 0.15 \times 0.05 = 0.0075$, or less than 1 %. The "module" (the combination, as shown, of all four components) fails if any one (or more than one) of A, D and the parallel combination of B and C fails. It is probably more straightforward to consider the situations under which the module functions correctly, namely when both A and D function, and at least one of B and C work normally. The latter occurs unless both B and C both fail, so the probability of at least one of B and C functioning is $1 - p_B p_C = 1 - 0.0075$, or 0.9925 (>99 %). The probability that both of A and D work is $(1 - p_A)(1 - p_D) = (1 - 0.08) \times (1 - 0.07) = 0.92 \times 0.93 = 0.8556$, or around 86 %. Thus, the probability of the entire combination of A, B, C and D (as shown in Fig. 3) working normally is 0.9925 $\times$ 0.8556 = 0.849183, and so the probability of the "module" failing is $1 - 0.849183 = 0.150817$, or approximately 15 %. Once again, the "series" aspect of the module is less robust than the weakest component in the series. We can therefore see that a "pipeline" in which many devices and/or processes depend on each other in a serial way, such as where the correct functioning of one process entirely depends on the successful completion of all the preceding tasks in the pipeline, the overall reliability can be severely degraded even if each individual stage in the pipeline is relatively reliable!

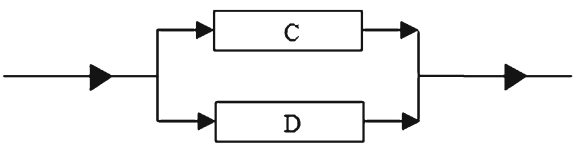

Fig. 2 "Parallel" combination of components C and D. In this case, neither relies on the other to perform its function correctly, and any components or modules external to this combination only relies on at least one of C and D working normally. The *arrows* indicate the sequence of any dependency, e.g. of data flow

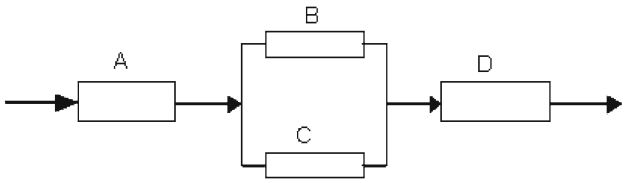

Fig. 3 An example "module" combining components B and C "in parallel", "in series" with components A and D

4.2.2 Combinations of independent identical components

We now consider a situation where a combination or module which contains several identical components, which we assume function or fail independently, i.e. form a parallel combination, but where failure of any individual components may degrade the overall performance of the combination. Examples of this include multiple communication channels, or multiple processors or cores working on the same task. Assume that there are N such components, and the probability of any one of these failing in unit time is p. The probability of any m out of the N components failing in unit time is then given by the binomial probability distribution:

$$P(m = M) = {}^{N}C_{M}\, p^{M}(1-p)^{(N-M)} \quad \text{for } M = 0, 1, 2, \ldots, N$$

where ${}^{N}\mathrm{C}_{M} = N!/(M!(N-M)!)$ [32]. In such cases, the mean number of failures in unit time is given by Np, and the standard deviation of the number of failures per unit time by $\sqrt{(Np(1-p))}$. In this case, individual failures do not make the combination fail completely (unless all N failed), but the more components fail, the more seriously the performance of the combination is impaired.

If N is large, $N!$ is extremely large (greater than can be stored in most integer or floating point representations for $N > 70$) and calculation of ${}^{N}\mathrm{C}_{M}$ becomes impractical except for cases where M is close to zero or close to N, in which situations the majority of the factors involved in the numerator and denominator of ${}^{N}\mathrm{C}_{M}$ cancel-out. In fact, in cases where N is very large, $P(m = M)$ will tend to be very small for all values of M, except for small M if p is relatively small (close to zero), and for large M if p is relatively large (close to 1). If N is large and p is small, we can make an approximation by using the Poisson probability distribution instead of the binomial, which avoids the necessity and complication of calculating ${}^{N}\mathrm{C}_{M}$:

$$P(m = M) = \mathrm{e}^{-\mu}\mu^{M}/M! \quad \text{for } M = 0, 1, 2, \ldots,$$

where $\mu = Np$ is the mean number of component failures per unit time [32].

In this context, the MTTF is not necessarily meaningful, other than noting that the case $m = N$ corresponds to complete failure of the group of components, which may then lead to complete failure of the whole system (depending on the architecture of the system as a whole). The probability of this happening is $p^{N} = (\mu/N)^{N}$ using the binomial distribution, or $e^{-\mu}\mu^{N}/N!$ in the Poisson approximation. This then gives the MTTF of the group of components as $1/p^{N} = (N/\mu)^{N}$, or $\mathrm{e}^{\mu}\ \mathrm{N}!\ /\ \mu^{N}$ under the Poisson approximation. In the case of failures occurring independently, but sequentially, over time, μ can be considered as the average rate of occurrence of failures, and the times at which these failures occur are then said to follow a Poisson process. More complex discrete probability models and distributions, and their scope of applicability, are covered in standard texts on probability theory, such as [32]. In particular circumstances relevant to IEs—for example, where assumptions of neither the binomial nor Poisson models are valid—the use of some such more complicated probability distributions may be required in a statistical reliability analysis.

4.2.3 Continuously variable failure probabilities

Up to now, we have assumed that the probability of a given component failing in any unit time interval was a constant. Although this may be a reasonable approximation for software components, this is not necessarily realistic for hardware components, for which expected failure rates may increase as the components age. Now let us consider the *failure probability density* with respect to time, $f(t)$, which we allow to vary over time t, of a component, module or system. $f(t)$ is defined in such a way that the probability of the object of interest failing in the time interval between t and $(t + \delta t)$ is equal to $f(t) \cdot \delta t$, with $\int_{-\infty}^{\infty} f(t)\mathrm{d}t = 1$. If we define $t = 0$ as the start of the working life of the object of interest, then we can ignore all values of time $t < 0$. The probability of the object failing between times $t = a$ and $t = b$ is then given by $\int_{a}^{b} f(t)\mathrm{d}t$ and, if T is the time at which the object first fails, we can define the cumulative failure probability distribution function $F(t) = P(0 < T < t) = \int_{0}^{t} f(s)ds$, where s is used as a dummy variable. It can then be seen that $f(t) = \frac{\mathrm{d}F}{\mathrm{d}t}$.

The reliability function, $R(t)$, of the object is defined to be the probability that the object does not fail before time t: $R(t) = P(T > t) = 1 - P(T < t) = 1 - F(t)$. We note that no object should fail in zero time, so $R(0) = 1$, and, assuming that no object can last forever without failure, $R(t) \to 0$ as $t \to \infty$.

The MTTF of the object is given by the expected value of the failure time, $\mathrm{MTTF} = \int_{0}^{\infty} t\ f(t)\ \mathrm{d}t$. If $R(t)$ decays towards zero as $t \to \infty$ in such a way that $(tR(t)) \to 0$ as t $\to \infty$, noting that $\frac{\mathrm{d}R}{\mathrm{d}t} = -f(t)$ and integrating by parts, it can be seen that $\mathrm{MTTF} = \int_{0}^{\infty} R(t)\mathrm{d}t$.

The so-called hazard function, $h(t)$, of an object is defined as its relative failure rate at time t compared with its current reliability, i.e. $h(t) = f(t)/R(t)$.

Example Suppose a given component has reliability function given by $R(t) = e^{-0.2t}$, where t is measured in years. The cumulative failure distribution function is then given by $F(t) = 1 - e^{-0.2t}$ and the failure probability density is $f(t) = 0.2\ e^{-0.2t}$. It can be seen that, as would be expected, $F(t)$ is an increasing function with $F(0) = 0$ and $F(t)$ tending towards 1 as $t \to \infty$. Here, $R(t) = e^{-0.2t}$. The MTTF of that component is then $\mathrm{MTTF} = \int_{0}^{\infty} tf(t)\mathrm{d}t = \int_{0}^{\infty} R(t)\mathrm{d}t$.

In this case, the latter expression is the easier to evaluate, giving MTTF = 5. The hazard function for the object is then $h(t) = f(t)/R(t) = 0.2$, i.e. the hazard function, or relative failure rate, is constant.

Noting that the probability of any given object failing in a specified time interval $a \leq t \leq b$ is given by $F(b) - F(a)$, the reliability of "serial" or "parallel" combinations of objects can then be dealt with in an analogous manner to that illustrated in the earlier section where the failure probabilities of each object were fixed.

4.2.4 Normally (Gaussian) distributed component failure times

If the failure times of objects follow certain distributions which are particularly well-understood theoretically, then various quantities of interest for collections of such objects can be calculated easily. One such distribution is the Normal or Gaussian distribution, for which there is an extensive body of theoretical results [32]. The Normal distribution is symmetrical about its mean value, and has identical mean, mode and median.

Consider a set of components for which the "lifetimes" (i.e. working times to failure) are individually normally distributed, such that the lifetime of component i, T_i, is normally distributed with mean μ_i and standard deviation σ_i:

$T_i \sim N(\mu_i, \sigma_i)$. The failure probability density function for the component's lifetime is $f(t) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{\frac{-(t-\mu_i)^2}{2\sigma_i^2}}$. This function cannot be integrated analytically between general limits, but it is straightforward to compute good numerical estimates for the probability that T_i lies in any specified range, using any suitable method of numerical integration:

$P(a \leq T_i \leq b) = \int_a^b f(t)\mathrm{d}t$. In fact, for the "standard" Normal distribution which has mean $\mu = 0$ and standard deviation $\sigma = 1$, the function $\Phi(z) = P(T_i < z)$ is tabulated (see, e.g. [32,42]), and also computed by most statistical software and many programming environments and spreadsheet packages, and the probability that T_i lies in any interval of interest can be calculated from these tabulated (or numerically computed) values: $P(a \leq T_i \leq b) = \Phi(b) - \Phi(a)$. For a component whose lifetime is normally distributed with mean μ_i and standard deviation σ_i, a transformation $Z_i = (T_i - \mu_i)/\sigma_i$, with inverse $T_i = \mu_i + Z_i\sigma_i$, will convert the lifetime T_i into the variable Z_i which follows the standard normal distribution, for which probabilities of Z_i falling into any specified range can be calculated using the tabulated (or computed) function $\Phi(z)$.

Example A component has a "lifetime" (i.e. time to failure) which is normally distributed with mean 10 years and standard deviation 2 years. What is the probability that the component will last (1) less than 8 years, (2) at least 10 years, (3) at least 12 years, (4) between 8 and 12 years?

Firstly, we convert each of the given lifetime values to Z values using the above transformation, with $\mu_i = 10$ years and $\sigma_i = 2$ years. Thus, for (1) $T = 8$ gives $Z = -1$, (2) $T = 10$ corresponds to $Z = 0$, (3) $T = 12$ becomes $Z = +1$. $\Phi(Z)$ is usually only tabulated for positive values of Z, but Φ values for negative Z values can be determined using the symmetry of the Normal distribution function graph. In the case of (1), we note that $P(T < 8) = P(Z < -1) = P(Z > +1)$ by symmetry. Furthermore, $P(Z > +1) = 1 - P(Z < +1)$, and $P(Z < 1) = \Phi(1)$, which is tabulated, taking the value 0.8413 to 4 decimal places. Hence, we find that the required probability, $P(T < 8) = 1 - 0.8413 = 0.1587$, so the probability of the component lasting less than 8 years is 0.1587, or about 16 %. This would mean that, given a large number of such components which were notionally identical, about 16 % of them would last less than 8 years before failing. For (2), we need $P(T \geq 10)$, which is equal to $P(Z \geq 0) = 1 - P(Z < 0) = 1 - \Phi(0) = 1 - 0.5000 = 0.5000$. Thus, the probability of one such component lasting at least 10 years is exactly 50 %. In this instance, we could have spotted this straight away, since the mean lifetime of the component was given as being 10 years, and the symmetry of the Normal distribution about the mean results in values below the mean and values above the mean being equally likely. In the case of (3), we want to find $P(T > 12)$, which corresponds to $P(Z > 1) = 1 - P(Z < 1) = 1 - \Phi(1) = 1 - 0.8413$, from tables of the standard Normal distribution. This gives the probability of the component lasting at least 12 years as 0.1587, again equal to about 16 %. Finally, for (4), the required probability is $P(8 < T < 12) = P(-1 < Z < +1) = P(Z < +1) - P(Z < -1)$. Using the values we have already found, we observe that $P(Z < +1) = 0.8413$ and $P(Z < -1) = 0.1587$, so $P(8 < T < 12) = 0.8413 - 0.1587 = 0.6826$. Thus, the probability of the component lasting between 8 and 12 years is about 68 %. Put another way, out of a large number of such supposedly identical components, we would expect about 68 % of them to last between 8 and 12 years before failing.

For distinct components, of which failures can be assumed to be independent, the "combination rules" derived in the earlier section can be employed. For lifetimes of such independent components, the mean of a sum of lifetimes is equal to the sum of the individual means: if $T = T_1 + T_2 + \cdots + T_N$, then the mean μ is given by $\mu = \mu_1 + \mu_2 + \cdots + \mu_N$. However, it is variances rather than standard deviations of independent quantities which can be added, so the standard deviation of the sum of independent lifetimes is given by the square root of the sum of squares of the individual standard deviations: $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_N^2}$, where σ_i is the standard deviation of the lifetime of the ith component. For scaling of a quantity by a constant factor α, the mean value and standard deviation also get scaled by the same factor α.

Direct consequences of these results are that, given a set of N notionally identical components, with lifetimes T_i independent of each other but identically normally distributed with mean μ and standard deviation σ, then the mean lifetime, $\overline{T} = \frac{1}{N}\sum_{i=1}^{N} T_i$, of the sample of N such components is itself Normally distributed, with mean equal to μ but with standard deviation equal to $\frac{\sigma}{\sqrt{N}}$. For example, a sample of $N = 100$ notionally identical components whose lifetimes are distributed with mean 8 years and standard deviation 2 years will have a sample mean lifetime which will be distributed such that its mean is 8 years, but standard deviation of the sample mean equal to $\frac{\sigma}{\sqrt{N}} = \frac{2}{\sqrt{100}} = \frac{2}{10}$ or just 0.2 years! This result is extensively used in process quality control based on samples of the product being manufactured, particularly if the quality test makes the sampled product unusable or unsaleable after testing. However, the result is of much wider applicability to any type of sample.

The models and results outlined above can be applied to multi-component systems, which in principle can be of arbitrary complexity. However, many of the processes described here require recursive analysis, which will become more complicated for systems with a large number of interacting components, which may not be identical or have identically-distributed properties. However, in the next section, some possible applications of the theory to the field of Intelligent Environments will be proposed.

5 Applications to Intelligent Environments

As noted previously, Intelligent Environments tend to be complex, hybrid systems, involving large numbers of software and hardware components. Adopting a mathematical or statistical approach to modelling their reliability may appear daunting, particularly if the reliability, in the quantitative sense, of individual components may not be precisely known. However, by using estimates of failure rates or probabilities, and taking a "modular" approach to analysis of the system of an entire IE can lead to invaluable predictions about how reliable it is likely to be, in terms of how likely are complete failures of the system, or situations where its performance is severely degraded.

Example 1 Consider once again the combination of components (or modules) illustrated in Fig. 3. Suppose the component A is some sensor (for example, a video camera), and components B and C are some pieces of inference-making software which, other than both being dependent on the output from the sensor A, function independently of each other. Let component D then be another piece of software which takes the outputs from B and C and then uses these to make some decision.

The correct functioning of this module relies on both A and D working correctly, and at least one of B and C functioning normally. The previously-given analysis based on the failure probabilities of the individual components can then be used.

Example 2 Suppose part (one "module") of an intelligent environment incorporates a fairly large number of identical sensors, for example movement sensors. The performance of the environment is robust to a few of these failing, but is severely degraded if 4 or more of them fail, and the system will not function satisfactorily if more than 6 of them fail. We assume here that the correct functioning of the individual components is not monitored continuously, or at least failed components are not replaced or repaired straight away, so all components which fail during the same "unit time interval" (e.g. week or month) do not get replaced until the end of that interval. Suppose that the mean number of these components which fail in the module of interest during any one unit time interval is μ. If the total number N of these sensors in the module is relatively small, then $\mu = Np$, where p is the probability of any one of the sensors failing in that time period, and we can calculate the probability of m of the sensors failing, $P(m)$, in the same time interval using the binomial distribution previously discussed. If N is larger, so that using the binomial distribution becomes impractical, but p is reasonably small (as an approximate guide, $p \leq 0.2$), then we can instead use the Poisson distribution to estimate the value of $P(m)$ for any particular value of m. In either case, the probability of at most S sensors failing in that time period is given by $P(m = 0)+P(m = 1)+\cdots+P(m = S)$, and the probability of more than S of them failing in that time interval is therefore $1-[P(m = 0)+P(m = 1)+\cdots+P(m = S)]$. We are particularly interested in more than 4 sensors failing (which makes the performance of the system severely degraded), namely $1-[P(m = 0)+P(m = 1)+P(m = 2)+P(m = 3)+P(m = 4)]$, and the case of $S = 6$ (more than 6 sensors fail in the same time interval, so the system fails), with probability $1-[P(m = 0)+P(m = 1)+\cdots+P(m = 6)]$. The mean time to such a degradation or failure is given by the reciprocal of the corresponding probability. As a specific example, consider the case where the mean number of sensor failures per month, based on evidence from previous experience, is $\mu = 1.6$, and the number of such sensors in the IE (or the part of it currently of interest) is large. We therefore use the Poisson distribution to calculate the probabilities, $P(m)$, of m of the sensors failing in any given month. Recalling that we are assuming that the performance of the IE is seriously degraded if $m > 4$, and the IE fails to function satisfactorily if $m > 6$, the values of $P(m)$ which we require for our analysis, calculated using the Poisson distribution formula given previously, are given in Table 1 below.

Thus, from Table 1, we can see that, under the specified assumptions, there is a probability of 0.02368, or about 2.4 %,

Table 1 Probabilities of m sensors, no more than m sensors, and more than m sensors failing in 1 month, assuming that failures are independent of each other, but the number of failures per month are Poisson distributed, with a mean of 1.6 failures per month

Number of sensors failing, m	Probability, $P(m)$	Cumulative probability, $P(M \leq m)$	Probability of more than m sensors failing, $P(M > m) = 1 - P(M \leq m)$
0	0.201897 (∼20 %)	0.201897 (∼20 %)	0.79810 (∼80 %)
1	0.323034 (∼32 %)	0.524931 (∼52 %)	0.47507 (∼48 %)
2	0.258428 (∼26 %)	0.783358 (∼78 %)	0.21664 (∼22 %)
3	0.137828 (∼14 %)	0.921187 (∼92 %)	0.07881 (∼7.9 %)
4	0.055131 (∼5.5 %)	0.976318 (∼98 %)	0.02368 (∼2.4 %)
5	0.017642 (∼1.8 %)	0.993960 (∼99.4 %)	0.00604 (∼0.6 %)
6	0.004705 (∼0.5 %)	0.998664 (∼99.9 %)	0.00134 (∼0.1 %)

of at least 4 sensors failing in the same month and the monitoring performance of the IE being seriously degraded, but only a 0.00134, or about 0.13 %, probability of more than 6 sensors failing in the same month and that aspect of the IE failing completely. This gives the mean time before the system's performance is serious degraded as $1/0.02368 \approx 42.2$ months, and the mean time before that part of the IE fails completely as $1/0.00134 \approx 746.3$ months, or a little over 62 years!

Example 3 Consider a situation where a component (or module) of an IE is monitored, and is replaced as soon as it fails. However, only the original and three (notionally identical) replacements of that component are held "in stock". We assume that the time to replace the component "from stock" and repair the IE is negligible, but obtaining the component when not in stock and then repairing the IE would take a long time and result in the IE being unavailable for that period. Let us assume that the lifetimes (i.e. operating time until failure) of each such component is randomly, but identically distributed, following a Normal distribution of mean μ months and standard deviation σ months. Assuming that the lifetimes (once installed) of the individual copies of the components are independent of each other, the total lifetime of the four copies, and thus the time before the IE becomes unavailable due to their failure is also Normally distributed, with mean (4μ) months and standard deviation $\sqrt{(4\sigma^2)} = 2\sigma$ months. For known values of μ and σ, for example $\mu = 20$ months and $\sigma = 4$ months, probabilities of the set of 4 copies of the component lasting less than a specified number, at least a specified number, or more than a specified number of months can be calculated using Normal distribution tables or software, as outlined previously.

The above examples illustrate how elementary statistical reliability theory can be usefully applied to predicting the reliability of parts of Intelligent Environments, or even complete Intelligent Environments, provided that appropriate assumptions can be made. It may be the case that in some circumstances, the precise assumptions which are at least approximately valid may have to be varied, making the reliability analysis more complicated. However, in the majority of situations, provided the Intelligent Environment can be decomposed into "modules" which are individually not excessively complex, it should be possible to undertake at least an approximate theoretical evaluation of the reliability of each module, and thence of the whole system of the Intelligent Environment.

6 Conclusions

This paper has presented some of the issues relating to the reliability of complex systems, such as Intelligent Environments, and highlighted how important it is to gain a better theoretical understanding of, and to be able to model and predict, the reliability of such systems. The basic concepts of statistical reliability theory, along with examples illustrating how these may be applicable to the context of Intelligent Environments, have been presented. Although this article can only hope to give a very brief introduction to this highly important topic, suggestions for further reading to give more in-depth knowledge of the field have been given.

Acknowledgments I would like to thank Dr. Mastaneh Davis, Dr. Vincent Lau and Prof. Philip Scarf for some helpful information, and Prof. Les Hatton for bringing the body of work on software reliability and defects to my attention.

References

1. Augusto JC, Callaghan V, Cook D, Kameas A, Satoh I (2013) Intelligent Environments: a manifesto. Hum-Centric Comput Inf Sci 3:12
2. van den Broek G, Cavallo F, Wehrmann C (2010) AALIANCE ambient assisted living roadmap. In: IOS press ambient intelligence and smart environments series, vol 6. IOS Press, Amsterdam
3. Helal S, Mokhtari M, Abdulrazak B (2009) The engineering handbook on smart technology for aging, disability and independence. In: Computer engineering series. Wiley, USA. ISBN 978-0-471-71155-1
4. McTear M, Raman T (2004) Spoken dialogue technology: towards the conversational user interface. Springer, Berlin
5. Böhlen M, Maharika I, Yin Z, Hakim L (2014) Biosensing in the Kampung. In: Proceedings of the 10th international conference on intelligent environments (IE'14), Shanghai

6. Mozer MC (1998) The neural network house: an environment that adapts to its inhabitants. In: Proceedings of the AAAI spring symposium on intelligent environments, pp 110–114
7. Langensiepen C, Lotfi A, Saifullizam P (2014) Activities recognition and worker profiling in the intelligent office environment using a fuzzy finite state machine. In: Proceedings of the IEEE world congress on computational intelligence (WCCI'14), Beijing
8. Suh W, Chang K, Lee E (2011) Towards reliable intelligent transportation systems for e-government. In: Andersen KN (eds) Proceedings of EGOVIS'11. Lecture notes in computer science, vol 6866, Springer, New York, pp 299–314
9. Komninos N (2015) The age of intelligent cities: smart environments and innovation-for-all strategies (regions and cities). Routledge/Taylor & Francis, Abingdon. ISBN 978-1-138-78219-8
10. Poledna S, Burns A, Wellings A, Barrett P (2000) Replica determinism and flexible scheduling in hard real-time dependable systems. IEEE Trans Comput 49(2):100–111
11. Zhang S, McCullagh PJ, Nugent C, Zheng H, Black N (2011) Reliability of location detection in intelligent environments. In: Novais P, Preuveneers D, Corchado JM (eds) Ambient intelligence—software and applications. Advances in intelligent and soft computing, vol 92. Springer, New York, pp 181–188. ISBN: 978-3-642-19936-3
12. Augusto JC (2009) Increasing reliability in the development of intelligent environments. In: Proceedings of 5th international conference on intelligent environments (IE'09), Barcelona. IOS Press, Amsterdam
13. Augusto JC, Hornos MJ (2013) Software simulation and verification to increase the reliability of intelligent environments. Adv Eng Softw 58:18–34
14. Augusto JC, McCullagh PJ (2011) Safety considerations in the development of intelligent environments. In: Ambient intelligence—software and applications (2nd international symposium on ambient intelligence, ISAMI'11). Advances in intelligent and soft computing, vol 92. Springer, Berlin, pp 197–204. ISBN 978-3-642-19936-3
15. Augusto JC, Huch M, Kameas A, Maitland J, McCullagh P, Roberts J, Sixsmith A, Wichert R (2012) Handbook on ambient assisted living—technology for healthcare, rehabilitation and well-being. In: Ambient intelligence and smart environments book series, vol 11. IOS Press, Amsterdam
16. Brewer E (2000) Towards robust distributed systems. In: Proceedings of the 19th annual ACM symposium on principles of distributed computing (PODC'00). ACM, pp 7–10
17. Gilbert S, Lynch N (2002) Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. ACM SIGACT News 33(2):51–59
18. Zamudio V, Callaghan V (2009) Understanding and avoiding interaction-based instability in pervasive computing environments. Int J Pervasive Comput Commun 5(2):163–186
19. Lipow M (1982) Number of faults per line of code. IEEE Trans Softw Eng 8(4):437–439
20. Gaffney JE (1984) Estimating the number of faults in code. IEEE Trans Softw Eng 10(4):459–464
21. Compton BT, Withrow C (1990) Prediction and control of Ada software defects. J Syst Softw 12:199–207
22. Hatton L (1997) Re-examining the fault density—component size connection. IEEE Softw 14(2):89–97
23. Hatton L (1997) The T-experiments: errors in scientific software. IEEE Comput Sci Eng 4(2):27–38
24. Hatton L (2009) Power-law distributions of component size in general software systems. IEEE Trans Softw Eng 35(4):566–572
25. Lightfoot D (1991) Formal specification using Z. Macmillan Press, USA. ISBN 0-333-54408-0
26. Potter B, Sinclair J, Till D (1991) An introduction to formal specification and *Z*. Prentice-Hall International, USA. ISBN 0-13-478702-1
27. Pfleeger SL, Hatton L (1997) Investigating the influence of formal methods. IEEE Comput 30(2):33–43
28. Satoh I (2003) A testing framework for mobile computing software. IEEE Trans Softw Eng 29(12):1112–1121
29. Sommerville I (2011) Software engineering, 9th edn. Pearson/Addison-Wesley, Boston. ISBN 978-0-13-703515-1
30. Hatton L (2007) The chimera of software quality. IEEE Comput 40(8):102–104
31. Wolstenholme L (1999) Reliability modelling: a statistical approach. CRC Press, USA. ISBN 978 158 488 0141
32. Feller W (1971) An introduction to probability theory and its applications, 3rd edn. Wiley, New York
33. Nakashima H, Aghajan H, Augusto JC (eds) (2009) Handbook of ambient intelligence and smart environments. Springer, Berlin. ISBN 978-0387938073
34. Ng TS, Hung YS (eds) (1995) Safety, reliability and applications of emerging intelligent control technologies. In: International federation of automatics control (IFAC). Pergamon Press–Elsevier, Oxford. ISBN 0-08-042374 4
35. Rausand M, Høyland A (2004) System reliability theory: models, statistical methods and applications, 2nd edn, Chapter 10, Wiley, Hoboken, pp 419–464. ISBN 0-471-47133-X
36. Bozzano M, Villafiorita A (2010) Design and safety assessment of critical systems. CRC Press, Taylor & Francis, Boca Raton. ISBN 978-1-4398-0332-5
37. Rausand M (2014) Reliability of safety-critical systems: theory and applications. Wiley, Hoboken. ISBN 978-1-118-11272-4
38. Lyu MR-T (2002) Software reliability theory. In: Encyclopedia of software engineering. Wiley, New York. doi:10.1002/0471028959.sof329
39. Hamlet D, Mason D, Woitm D (2001) Theory of software reliability based on components. In: Proceedings of the 23rd international conference on software engineering (ICSE'01), pp 361–370
40. Mei D (2007) Software reliability estimation in grey system theory. In: IEEE international conference on grey systems and intelligent services (GSIS'07), Nanjing
41. Korver B (1994) The Monte Carlo method and software reliability theory. Portland State University, Computer Science Department, Portland, OR, USA. Technical Report PSU TR 94–1. http://www-cs-students.stanford.edu/~briank/BrianKorverMonteCarlo.pdf. Accessed 28 Mar 2015
42. Normal Distribution Function. http://www.thestudentroom.co.uk/attachment.php?attachmentid=134337&d=1330530156. Accessed 28 Mar 2015