Performance Modeling of Critical Event Management for Ubiquitous Computing Applications *

Tridib Mukherjee, Krishna Venkatasubramanian, Sandeep K. S. Gupta
Department of Computer Science and Engineering
Arizona State University
Temp, AZ, 85287
{tridib,kkv,sandeep.gupta}@asu.edu

ABSTRACT

A generic theoretical framework for managing critical events in ubiquitous computing systems is presented. The main idea is to automatically respond to occurrences of critical events in the system and mitigate them in a timely manner. This is different from traditional fault-tolerance schemes, where fault management is performed only after system failures. To model the critical event management, the concept of *criticality*, which characterizes the effects of critical events in the system, is defined. Each criticality is associated with a timing requirement, called its *window-of-opportunity*, that needs to be fulfilled in taking mitigative actions to prevent system failures. This is in addition to any application-level timing requirements.

The criticality management framework analyzes the concept of criticality in detail and provides conditions which need to be satisfied for a successful multiple criticality management in a system. We have further simulated a criticality aware system and its results conform to the expectations of the framework.

Categories and Subject Descriptors: C.3 [Special-Purpose Application based Systems]: Real-time and embedded systems; C.4 [Performance of Systems]: Fault tolerance, Modeling techniques; D.4.8 [Performance] Stochastic analysis; I.6.5 [Model Development]: Modeling methodologies

General Terms: Performance, Reliability, Security, Theory.

Keywords: Autonomic Computing, Event Management, Proactive Computing, Safety-Critical Systems, Ubiquitous Computing.

1. INTRODUCTION

Ubiquitous Computing (UC) systems (also known as Pervasive Computing systems) [12] consist of a possibly large number of heterogeneous, massively distributed computing entities (e.g. embedded wireless sensors, actuators, and various miniaturized computing and novel I/O devices), whose goal is to seamlessly provide users with an information rich environment for conducting their day-to-day activities [1]. In order to provide effective services, in such an environment, the UC system components need to be able to interact seamlessly with one another. Any system, including UC systems, broadly put, can be considered to be in one of the two types of states - *normal* or *abnormal*. In a normal state, a system provides services for routine events. For example - a smart hospital, responding to the arrival of critical patients may automatically allocate appropriate resources such as available beds and contact emergency personnel. These routine services may, however, be inadequate when the system is in an abnormal state. For example, allocating resources when a natural disaster brings an influx of a large number of patients to the hospital is a far more demanding task. The changes in the system's environment (external and/or internal) which lead the system into an abnormal state are called *critical events* (e.g. large influx of patient). The resulting effects, of the critical events, on the system are called *criticality* (e.g. unable to allocate resources for patients) [5].

Critical events and the consequent criticalities require unconventional response from the system to be contained. For example, in a disaster scenario, extra medical personnel may be brought in from neighboring hospitals to manage the influx. We refer to such containment actions (e.g. response to a criticality) as *mitigative actions* (e.g. bringing in extra medical staff) and the ability of handling critical events as *manageability*. Further, critical events usually have a timing element associated with them requiring the system to contain its effects within a certain amount of time. This is fundamentally different from traditional systems, where any timing requirement is only provided by the application such as in real-time systems. Examples of timing requirement associated with criticalities include - the *golden*[1] or *critical hour*[2] for many medical emergencies such as heart attacks and diabetic coma and the mean-time before shutting down a server/rack in a datacenter with a failed cooling system is approximately few minutes (this depends upon the specifics of a datacenter). We refer to this time period after the occurrence of a critical event as its *window-of-opportunity*.

The criticality management process, to be effective and in order to facilitate timely mitigative actions, has to detect critical events as soon as they occur. This requires a level of proactivity, unlike fault management in traditional systems, where fault tolerance is exclusively employed in the event of faults. Even though, criticality management is initiated in response to critical events, handling them within their associated timing requirement, prevents the occurrence of system faults. For example - in the case of a datacenter, the server racks (in the event of a failed cooling system) need to be shut down within the datacenter dependent window-of-opportunity

*Supported in part by MediServe Information Systems, Consortium for Embedded Systems and Intel Corporation.

MSWiM'06, October 2–6, 2006, Terromolinos, Spain.

[1] http://en.wikipedia.org/wiki/Golden_hour_(medicine)
[2] http://en.wikipedia.org/wiki/The_Critical_Hour

to prevent its failures. Criticality management therefore ensures higher availability of the system by endeavoring to prevent failures.

Inclusion of timing constraints on mitigative actions makes criticality aware systems look similar to real time systems. However, there are some fundamental differences between the two. Real-time systems intend to schedule a set of tasks such that they are completed within their respective timing deadline. However, the time to execute the tasks are fixed for the CPU where it would be executed [9]. This is unlike the uncertainty involved, due to possible human involvement, in the time taken to perform mitigative actions in response to critical events. Further, in real time systems, completing tasks within their timing deadline guarantee successful execution of jobs, whereas in case of criticality management, successful completion of mitigative actions may not be deterministic as it may also depend on the human behavior in response to critical events and the level of expertise in performing the mitigative actions.

The *contributions* of this paper include a generic theoretical framework for criticality management and the derivation of conditions required for their effective manageability. This framework models the manageability as a stochastic process by considering the various states the system may possibly reach (as a result of criticalities) and the probabilities of transitioning between them. The state transitions occur because of either new criticalities or mitigative actions which take the system toward normality. To understand this framework better, we simulated a criticality aware system by considering three types of criticalities.

2. RELATED WORK

Ubiquitous systems are special type of embedded systems which have been made possible by miniaturization of computing and communication devices. Many embedded systems, e.g. heart pacemaker and computer networks in modern cars, are safety-critical systems [7]. Methodologies for designing and developing have been well documented [4, 8]. This paper deals with ubiquitous systems, which we refer to as criticality-aware systems, that fall at the intersection of safety-critical systems, autonomic computing systems (sharing the characteristics of self-manageability)[3], and proactive computing systems (sharing the characteristics of "dealing with uncertainty")[4].

In recent years, development of relevant information technology for disaster or crisis management is getting increasing attention from teams of interdisciplinary researchers. A report from the National Academy of Sciences [3] defines *crises* as "Crises,whether natural disasters such as hurricanes or earthquakes, or human-made disasters, such as terrorist attacks, are events with dramatic, sometimes catastrophic impact," and "Crises are extreme events that cause significant disruption and put lives and property at risk - situations distinct from "business as usual."" In this paper, we refer to these application-dependent crises as critical events and their effect on a ubiquitous computing system as *criticality*.

The RESCUE project[5] is an interdisciplinary effort, which involves computer scientists, engineers, social scientists and disaster science experts, with the goal of developing information technology for delivering "the right information to the right people at the right time during crisis response." This effort is focused on fast and effective multimodal data gathering, analysis, dissemination and presentation in a disaster situation [10]. The Secure-CITI project[6] is geared towards providing Emergency Managers for resource allocation and decision making [11]. In contrast, this paper concentrates on modeling the entire criticality management system which consists of physical, human, and virtual components. To the best of our knowledge this is the first work towards the goal of identifying crucial system parameters and properties, and determining the necessary and sufficient conditions in terms of these parameters for the system to satisfy these system properties.

The Caltech Infospheres Project[7] is focusing on "developing reliable distributed applications by composing software components in structured way." Applications of such compositional system - "systems built from interacting components" - include any application that requires "a sense-and-respond approach to data analysis and problem solving, for instance, crisis management." Ubiquitous computing applications can be viewed as sense-and-respond (S&R) systems. In [2], the author hints at evaluating S&R systems in terms of *timeliness* and *appropriateness* of response (which we refer to as mitigative actions).

In [5] the notion of criticality was presented informally and it was applied to access control problem in smart spaces. Further, it had a limited scope to manageability as it only addressed situations with single criticality. Here, we take a more comprehensive and formal approach and rigorously analyze the manageability of the system when multiple simultaneous critical events occur. Examples of multiple criticalities in a system could include situations, such as, massive patient influx in disasters (criticality 1) and lack of required medical equipments (criticality 2) for treatment.

In [6], a game-theoretic based resource management system for multi-crisis handling is presented. This work is mainly concerned with strategies for expedited, fair (socially optimal) resource allocation in multi-crisis situations in urban setting. Example of multi-crisis scenarios includes an urban area with concurrently occurring crisis events such as airplane crash at an airport, fight at a football stadium, gas leak in a neighborhood, and multi-car accident on a highway. The types of resources include fire trucks, police units, and medical ambulances. As opposed to this work, this paper focuses on generic performance modeling framework for such systems.

3. SYSTEM MODEL

We define criticality in a criticality-aware ubiquitous system system as follows:

Definition 1. Criticality is the effect on the system and its inhabitants, as a result of events in the physical environment, which, without timely mitigative actions involving possible human activities, would lead to loss of lives or properties.

Systems under one or more criticalities are said to be in a **critical state**. Criticality is associated with a time-constraint, called the **window-of-opportunity**, that determines the delay between the occurrence of the event causing the criticality and the resulting disasters (such as loss of lives or properties). Events, that cause criticality, are called the **critical events**. Any system in a critical state, therefore, 1) has to detect and evaluate the severity of the critical events, 2) plan and schedule appropriate mitigative actions according to the available resources, 3) manage these actions to minimize any uncertainty due to human involvement, and 4) impose these actions to proactively avoid any disaster in the system. Figure 1 depicts the overall system with both physical and virtual components. Criticality awareness in the virtual entities should be able to effectively handle the criticalities and provide facilities for bringing the system back to the normal state. However, improper evaluation

[3] www-03.ibm.com/autonomic/pdf/autonomic_computing.pdf
[4] www.intel.com/research/documents/proactivepdf.pdf
[5] www.itr-rescue.org/.
[6] www.cs.pitt.edu/s-citi.
[7] www.infospheres.caltech.edu/

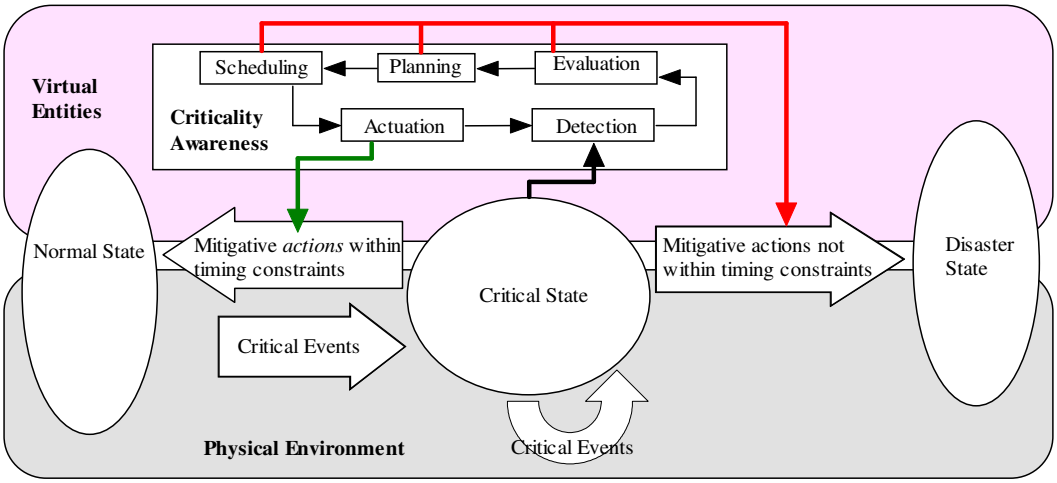

Figure 1: System components of an ubiquitous system and their interactions

of criticalities, mis-planning the mitigative actions, or missing the timing requirements while scheduling these actions may fail to prevent the disaster.

Let E be the set of events in the physical environment and $CE \subset E$ be the set of critical events. The effects of event $i \in CE$ on the system and its human inhabitants is denoted as criticality c_i. Let C be the set of all active (uncontrolled) criticalities in the system when i occurs. A criticality c_x is active at any instance of time:

1. if the critical event x has already occurred,
2. the mitigative action is not successfully performed till that instance, and
3. the window-of-opportunity for the criticality is not over.

Each criticality $c_i \in C$ is characterized by the 3-tuple (o_i, w_i, A_i), where

- o_i is the time when the critical event i occurred,
- w_i is the window-of-opportunity for c_i, and
- A_i is the set of possible mitigative (or corrective) actions for c_i in the system.

It is mandatory that any mitigative action in A_i is performed within w_i in order to avoid disasters caused by the critical event i.

The value of w_i is not independent of other criticalities and is also dependent on the human behavior in the physical environment. Suppose $H(t)$ models the probability of panic in the human inhabitants in the physical place at time t ($t > o_i$), in response to the occurrence of event i. Then, the w_i value for criticality c_i at time t can be given as $w_i(t) = f_w(i, C, H(t))$ where f_w, for all $c_i \in C$, is a function over $\Re_+$ that defines the dependency of w_i with other active criticalities in the system and the resulting human behavior as predicted. The average value of w_i can therefore be obtained as follows:

$$w_i = \frac{\int_{o_i}^{o_i+W_i} f_w(i, C, H(t))dt}{W_i}, \quad (1)$$

where W_i is an upper bound for the window-of-opportunity. For noncritical events, W_i tends to ∞, thereby making w_i undefined. For critical events, however, this value is the best case value obtained when there is no other active criticalities and the humans are all trained to behave according to the situation. It should be noted here that $W_i \geq f_w(i, C, H(t))$, where $o_i \leq t \leq o_i + W_i$.

Each mitigative action $m_j \in A_i, \forall c_i \in C$, is further characterized by the 4-tuple $(t_{ji}, R_{ji}, e_{ji}, p_{ji})$, where:

- t_{ji} is the average time taken to perform the action m_j,
- R_{ji} is the set of resources that are required to perform the action m_j ,
- e_{ji} is the average cost in performing m_j, and
- p_{ji} is the average probability of successful completion of m_j within w_i for mitigating criticality c_i.

The components that constitute the cost of mitigative action (e_{ji}) are application dependent parameters such as gathering certain resources to perform the actions and error probability of the resources under criticality c_i. If $M_h(m_j)$ models the probability of error in the mitigative action m_j, due to human involvement, t_{ji} can be represented as an application defined function over $\Re_+$ as follows:

$$t_{ji} = f_t(i, m_j, \Omega, M_h(m_j)), \quad (2)$$

where Ω is the set of available resources (after the occurrence of event i) and can be calculated as follows: if $\Omega(t)$ is the set of resources at time t, then $\Omega = \int_{o_i}^{o_i+w_i} (\Omega(t)dt)/w_i$. Further, due to the involvement of human activities in the mitigative actions, the successful completion of the actions is uncertain. Apart from this, the unavailability of resources while performing the actions can also lead to uncertain outcome of the mitigative actions (especially in completing the action within the stipulated time). If m_j is the mitigative action enforced for mitigating criticality $c_i \in C$, then, we can characterize the *average probability of successful completion* of m_j, p_{ji}, by the following function over $[0, 1]$:

$$p_{ji} = f_p(m_j, \Omega, R_{ji}, w_i, M_h(m_j)) \quad (3)$$

The **probability of disaster** i.e. the probability of loss of lives or properties as a result of critical event i can be given as $1 - p_{ji}$ when mitigative action m_j is employed.

4. FUNDAMENTAL PROBLEMS

Considering this model, the fundamental research questions that are addressed in this paper are: 1) what are the basic principles for the detection, evaluation and planning for criticalities?; and 2) how to find the sequence of actions that either maximize the probability of success and/or minimize the cost requirement in mitigating the criticalities in the system. The window-of-opportunities for the criticalities and the availability of resources in the system determine the constraints within which the objective has to be achieved. Thus the problem can be formalized as follows:

$$minimize \sum_{\forall c_i \in C} e_{ji}, \quad (4)$$

where $m_j \in A_i$, The above problem intends to find the most cost-effective mitigative actions for the critical events. The following problem, however, intends to minimize the probability of disaster due to critical events:

$$minimize(1 - \prod_{\forall c_i \in C} p_{ji}). \quad (5)$$

5. PROPERTIES OF CRITICALITY

This section identifies two generic properties for modeling criticality aware systems: Responsiveness, and Correctness. **Responsiveness** measures the speed with which the system is able to initiate the detection of a critical event. Therefore, the higher the responsiveness the more time there is to take corrective actions. **Correctness** ensures that mitigative actions are executed only in response to a critical event. Before we proceed to analyze these properties, we make some simplifying assumptions about our system model:

1. All criticalities are detected correctly, that is, their types and properties are accurately known at the time of detection; and
2. We do not address malicious attacks on the detection process.

The modeling of accuracy of any event detection system involves determination of probabilities of false positives and false negatives. These are well-known and well-studied factors. Incorporation of these probabilities, although important for overall modeling of the system, would unnecessarily complicate the modeling in this paper. Similarly, the uncertainty and problems introduced due to presence of malicious entities may be important for many applications. However, we ignore this issue here and leave it for future work.

5.1 Responsiveness to Criticality

This section, presents an analysis of responsiveness to criticalities in a system. It begin with analysis for the simple case of single criticality in a system and then expands it to multiple criticalities.

5.1.1 Single Criticality

Responsiveness captures the fraction of time in w_i after which, the corrective actions for a critical event are initiated. Upon occurrence of any critical event in the physical environment, the criticality aware system should be able to detect the resulting criticality. If Δ is the time to process (evaluate the parameters of the critical event and enforce mitigative actions) the detected event, then the criticality can be controlled iff $\exists\, m_j$ such that,

$$\Delta + t_{ji} \leq w_i \quad (6)$$

Further, Δ has two main components:

1. the time to initiate the detection (Δ_{ini}), and
2. the time to process (identify the nature of the event) any detected event (Δ_{proc}).

We characterize responsiveness with a term called **Responsiveness Factor** (RF) which is defined as:

Definition 2. $RF(i) = \frac{w_i}{\Delta_{ini}}$ for all criticality c_i.

Figure 2 shows the various states the system goes, over a timeline, through while mitigating single criticality. In the first case, criticality is not mitigated before the expiration of the w_i. In the latter case, criticality is mitigated successfully before w_i expires. Higher the RF for a criticality, lesser is the time (in terms of fraction of window-of-opportunity) taken to respond to it. For successfully handling of any criticality, the value of its RF must be greater than 1, otherwise, the time to respond to a critical event will be greater than the total time available for it (limited by w_i).

We next define **Utilization Factor** as follows:

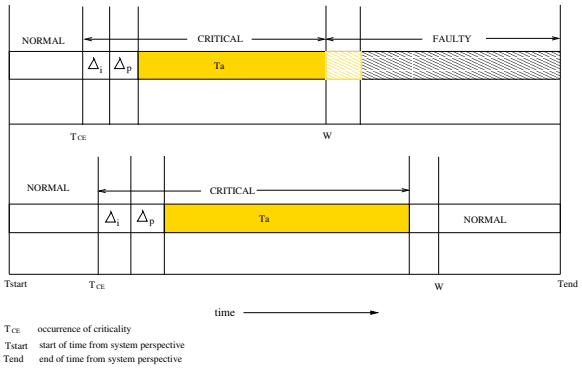

Figure 2: Single Criticality Timing Diagram. Note that Δ_{ini} and Δ_{proc} are depicted as Δ_i and Δ_p, respectively; and W and T_a denote the duration of window-of-opportunity and the time for performing the mitigative action for the single criticality, respectively.

Definition 3. $UF(i, j)$ is the Utilization Factor for controlling the criticality and is defined as the fraction of time taken (by action $m_j \in A_i$) for critical event processing and taking necessary controlling actions in w_i for criticality c_i.

Responsiveness Condition

Replacing Δ in Equation 6 with its constituents, we get $\Delta_{ini} \leq w_i - (\Delta_{proc} + t_{ji})$. Therefore,

$$RF(i) \geq \frac{w_i}{w_i - (t_{ji} + \Delta_{proc})}$$

$$\Rightarrow RF(i) \geq \frac{1}{1 - UF(i, j)} \quad (7)$$

Equation 7 is referred to as **Responsiveness Condition for Single criticality (RCS)**.

Equation 7 signifies that as the amount of time required to process and control a criticality increases, the time available to initiate its detection decreases, thereby imposing higher responsiveness requirements. Therefore, in summary, *the mitigation process has to meet the RCS condition to prevent system faults.*

5.1.2 Multiple Criticalities

In this section we generalize the above analysis for multiple criticalities. In a system where multiple critical events have been experienced, mitigative actions cannot be taken in an arbitrary manner. For example, in an hospital emergency department (ED), a patient suddenly experiences a life-threatening ventricular fibrillation, may require defibrillator treatment. If the defibrillator equipment suddenly malfunctions, then we have scenario with multiple criticalities where we have to fix the defibrillator (through replacement or repair) before treating the patient. This characteristic imposes a *priority* over the handling process, where certain criticalities need to be handled before certain others. Figure 3 illustrates a system where multiple critical events have occurred. The criticalities are mitigated according to their *priority*, thereby allowing the third criticality, which occurred last, to be mitigated first and so on.

Controllability Condition

Let c_i be a criticality which has occurred, and $C_H \subset C$ be the set of (uncontrolled) critical events which have higher priority than c_i. Then, c_i can be controlled iff (in the worst case) for any $m_j \in A_i$,

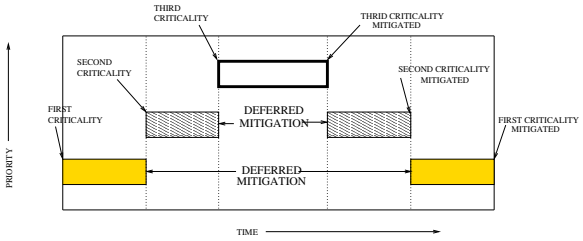

Figure 3: Multiple Criticalities

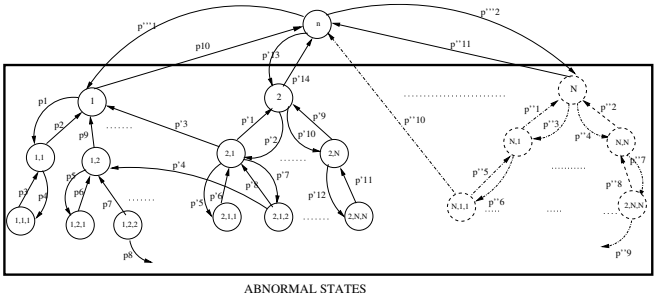

Figure 4: Critical State Transition Diagram

$$\Delta + \sum_{\forall c_x \in C_H, k \in A_x} (\Delta_{proc} + t_{kx}) + t_{ji} \leq w_i \quad (8)$$

Here,

$$\frac{\sum_{\forall c_x \in C_H, k \in A_x} (\Delta_{proc} + t_{kx})}{w_i}$$

denotes the fraction of time, in w_i, required for handling criticalities with priorities higher than c_i, thereby deferring the mitigation of c_i. We refer to this as the **Deference Factor (DF)** for criticality c_i by the higher priority criticalities in C_H ($DF(i, C_H)$). Therefore, Eq. 8 can be re-written as:

$$1 \geq \frac{1}{RF(i)} + UF(i, j) + DF(i, C_H) \quad (9)$$

Equation 9 signifies the necessary and sufficient condition for successfully handling any criticality c_i in the system. Note that, the condition is sufficient only for the case when the probability of success for all mitigative actions is 1. In the case of single criticality, the DF becomes 0, thereby degenerating equation 9 to 7. We call the inequality in Equation 9 as **Controllability Condition ($\mathcal{CC}$)** for any criticality c_i. This condition acts as the constraint for both the objective functions (Equations 4 and 5). *In this paper, we will only consider the objective function of Equation 5 (the same methods can be applied to solve for the objective of Equation 4).*

5.1.3 Periodicity of Detection

From the analysis of both single and multiple criticalities, it is imperative that - to guarantee responsiveness, timely detection of criticality is necessary. To achieve this, we need to periodically monitor for critical events. This suggests the need for an automatic (periodic) monitoring process which detects critical events with the minimum possible delay. Detecting critical events, in a non-automatic manner cannot guarantee a required responsiveness for the criticality, as the initiation of the detection process can have any arbitrary amount of delay. To model the **periodicity of the criticality detection** process, we designate $\mathbf{t}_p$, where $\alpha \leq t_p \leq \min_{c_i \in CE}(\Delta_{ini}(c_i))$, as the period after which the detection process is repeated. Here α is a system dependent constant which provides the lower bound for the detection initiation period, and CE is the set of all possible criticalities. For a system all possible criticalities are assumed to be known *a priori*, much like exceptions in traditional systems.

5.2 Correctness of Criticality

Correctness ensures that any controlling steps are executed by a system only in case of a critical event. This qualitative property of criticality cannot be analytically modeled and depends on the design and implementation of the system. For example, in a bank environment, a criticality could be an unauthorized entry into the vault. To detect this criticality and take mitigative actions (locking the exits, calling the law enforcement), the detection system (hardware and software) has to work accurately, which is totally dependent upon the hardware and software technologies used. The probability distribution of accurately detecting a critical event determines the probability of occurrence of the criticality in our system. If, for example, the detection process is not very accurate and reports a criticality even if there is no causing critical event, the probability of occurrence of that critical event, in our system analysis, becomes high.

6. MANAGEABILITY OF CRITICALITIES

In this section, we analyze the manageability of the critical events as a stochastic process. The analysis presented here pertains to the generalized case of multiple criticalities (single criticality manageability being a trivial specialization). Further, in all the subsequent analysis, we assume the maintenance of the correctness property. Further, for the simplicity of notational representation, we have assumed that for any c_i, only one action can be taken. The determination of which action to take from the set A_i is trivial (the action m_j which has the highest p_{ji} that meets the $\mathcal{CC}$ condition).

6.1 Stochastic Model

When a criticality occurs in the system (in normal state), it moves into the critical state (Figure 1). All subsequent criticalities keep the system in the critical state. In order to model the handling of multiple criticalities, we have to first express the critical state in more detail. The critical state, as shown in Figure 1 encompasses many system states which are reached in response to the occurrence of different criticalities. For example in Figure 4, state $(1, 2)$ is reached when the criticality c_2 occurs before successful handling of criticality c_1. The arcs between states represent the state transitions, each of which is associated with a probability value. State transitions occur as a response to either critical events or mitigative actions. The state transition diagram is organized in a hierarchical format. Each occurrence of a criticality moves the system down this hierarchy (the associated arc is called *critical link* (CL)) and each mitigative action moves it upward (the associated link is called the *mitigative link* (ML)). The set of all CLs and MLs of any node i is referred as $CL(i)$ and $ML(i)$, respectively. The probabilities associated with a criticality link c (an outgoing downward arc) originating from a particular state s represents the probability of occurrence of the critical event associated with c in state s. On the other hand, the probability associated with a mitigative link m (an outgoing upward arc) originating from a particular state s represent the probability of successfully handling a criticality using the mitigative action corresponding to link m.

Let C be the set of all criticalities which can occur in the system. In Figure 4, initially the system is in the normal state. Now suppose a criticality $c_2 \in C$ occurs, it will immediately move the system into the state represented as (2). Before the system has a chance to successfully mitigate the effects of c_2, another criticality $c_1 \in C$ occurs. This event will further move the system down the hierarchy to the state represented by $(2, 1)$. Now in order for the system to move up to the normal state it has to address (mitigate) both criticalities before their respective window-of-opportunities. The process of mitigation can be done in two ways in this case, by mitigating criticality c_1 before c_2 or vice versa, the order of which may be application and criticality dependent. Therefore, there are potentially two paths in the hierarchy that the system can take to reach the normal state. If both paths can be taken (the order of criticality mitigation is immaterial), then the choice of the paths depends upon two factors:

1. the probability of reaching the neighboring state (up the hierarchy) from the current state, and
2. the average probabilities of success for reaching the normal state from each of the neighbor state.

These average probabilities of success of reaching the normal state from any neighbor state depends not only on the probabilities of the MLs along the path but also on the possible criticalities (i.e probabilities of the CLs) at the intermediate sates taking the system down the hierarchy through CL. It should be noted that the sum of probabilities for all outgoing state transition arcs from any state is at most equal to 1, and the sum of probabilities of all outgoing CLs from any state (denoted as $PC(i)$ for state i) is less than 1.

As stated earlier we concentrate on objective function in Equation 5 which is concerned with finding a sequence of mitigative actions (or next mitigative action) which minimizes the probability of disaster (failure of the system to take appropriate corrective actions.).

6.2 Minimizing Probability of Disaster

Consider the system which, as a result of multiple criticalities, is in the state x (Figure 5). The goal is to find the path which has the highest probability of success to reach the normal state (which signifies the probability of mitigation for all the uncontrolled criticalities). In order to do that, the system has to find a neighbor state (up in the hierarchy) which has the highest probability of successfully reaching the normal state[8]. Let state i be any such neighboring state. The probability of successfully reaching the normal state(represented as state n) from state i is given by a recursive expression as follows,

$$P_{i,n} = \begin{cases} 1 & i = n \\ (1 - PC(i))APM(i) + APC(i) & i \neq n \ \& \ \mathcal{CC} \\ 0 & \neg\mathcal{CC} \end{cases}$$

where $p_{i,j}$ is the probability associated with the arc that goes from state i to state j, $PC(i) = \sum_{(i,j)\in CL(i)} p_{i,j}$ is the average probability of criticality at state i, $APM(i) = \sum_{(i,k)\in ML(i)} p_{i,k}P_{k,n}$ is the average probability of reaching i if no additional criticality occurs at state i, and $APC(i) = \sum_{(i,j)\in CL(i)} p_{i,j}P_{j,n}$ is the average probability of reaching n if an additional criticality occurs at state i. If the $\mathcal{CC}$ is not met for any state transition path $u \rightarrow v$ in the hierarchy then $P_{u,v}$ will be zero. Now, the probability of success to reach n from x through neighbor state i is given by:

$$Q_{x,i,n} = \begin{cases} p_{x,i}P_{i,n} & \mathcal{CC} \\ 0 & \neg\mathcal{CC} \end{cases}$$

[8]We assume that once a criticality has been mitigated it does not occur again before reaching the normal state.

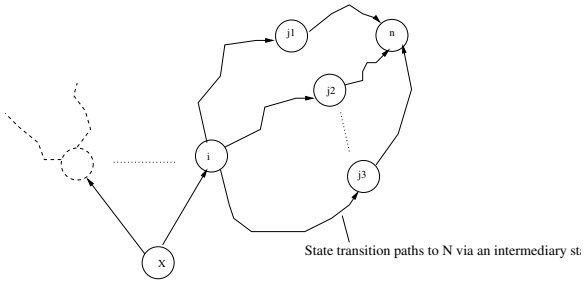

Figure 5: Paths from State X to Normal State

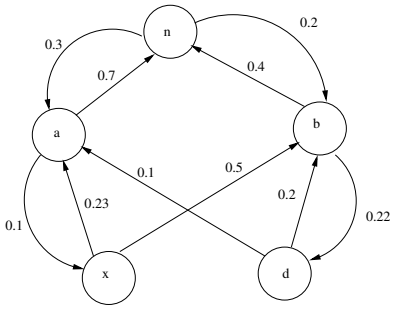

Figure 6: Example of State Transition Path

Therefore the **best state** to move to from state x (in the pursuit of reaching n) is the state u (such that $(x, u) \in ML(x)$) which meets the following requirement: $Q_{x,u,n} = \max_{(x,i)\in ML(x)} Q_{x,i,n}$. The value $Q_{x,u,n}$ is referred to as the **Q-value** for $ML(x)$.

6.3 An Example

Consider a system with a state transition hierarchy as shown in Figure 6. It represents a system which can have only two types of criticalities (c_1 and c_2). Further, assume that the system is currently in the state x and w_1 and w_2 for c_1 and c_2 are 10 and 20 units, respectively. In this hierarchy, each ML (i, j) has an associated $t_{i,j} = 1$ unit ($t_{i,j}$ denotes the time required to take the mitigative action of criticality c_p that transits the system from i to j). To find the state transition path to n from x, we first individually compute the average probability of success (of reaching n) from each of the neighbors of state x. In this example, the neighbors of state x are states a and b. The average probability of reaching n from a is given (by applying Equation 10) as $P_{a,n} = p_{a,n}$. This is because, there is only one path from a to n and the system will not move back from state a to state x. Further, we see that the condition $\mathcal{CC}$ is met here because to move from a to n ($t_{a,n}$) is 1 unit, making UF $\frac{1}{10}$ (when the system is at a). Also, as there are no other criticalities, DF is 0, and $\frac{1}{RF}$ is 0 (assuming instantaneous initiation of detection). Therefore, the $\mathcal{CC}$, which yields ($0 + 0.1 + 0 \leq 10$), holds true. Thus we have $P_{a,n} = 0.7$.

Similarly, the average probability of reaching state n from state b is given by $P_{b,n} = (1 - p_{b,d})p_{b,n} + (p_{b,d}P_{d,n})$. It can further be easily verified that $\mathcal{CC}$ is met if any path through b ($x \rightarrow b \rightarrow d \rightarrow a \rightarrow n$, $x \rightarrow b \rightarrow d \rightarrow b \rightarrow n$ or $x \rightarrow b \rightarrow n$) is taken. The above equation reduces to $P_{b,n} = (1 - p_{b,d})p_{b,n} + p_{b,d}[p_{d,a}(1 - p_{a,x})p_{a,n} + p_{d,b}p_{b,n}] = 0.34346$. Now, the probability of success in reaching n from x through a and b are $Q_{x,a,n} = p_{x,a}P_{a,n} = 0.161$ and $Q_{x,b,n} = p_{x,b}P_{b,n} = 0.17173$. Therefore, the best way to reach n from x is through state b. However, if we set w_2 to be 2 units (instead of 20 units), then we have $P_{b,n} = (1 - p_{b,d})p_{b,n} +$

$0 = 0.312$ as the path $x \rightarrow b \rightarrow d \rightarrow b \rightarrow n$ does not meet the $\mathcal{CC}$ making $Q_{x,b,n}$ to be 0.156. However as $Q_{x,a,n}$ is still 0.161, the path $x \rightarrow a \rightarrow n$ is now the better choice.

We next state an important theoretical result.

6.4 Manageability Theorem

The following theorem show the manageability of criticalities and the satisfaction of the liveness property.

Theorem 1. All the criticalities can be mitigated iff the maximum Q-value for any given state is greater than 0.

The following is a corollary to the above theorem:

Corollary 1. If the maximum Q-value for any given state is greater than 0 then all the criticalities can be mitigated within their respective window-of-opportunities.

Formal proof for the theorem and corollary are provided in the appendix. An informal explanation follows.

It can be claimed that all the criticality can be mitigated if at least one path can be successfully taken to the normal state. If the maximum Q-value is greater than 0, there is at least one path through which it is possible to reach the normal state without violating the $\mathcal{CC}$ condition. This is because the average probability of transiting from any neighbor to the normal state would become zero if the $\mathcal{CC}$ condition can never be satisfied. This would in turn make the Q-value zero and vice-versa. Same argument is applicable if the average probability of reaching the normal state from any of the neighbors is non-zero but it is not possible to maintain $\mathcal{CC}$ condition if any of the neighbors are selected (making Q-value 0). On the other hand, if the maximum Q- value is not greater than 0, it means there is no neighbor through which the Q- value is greater than 0. Therefore, there is no path to the normal state meeting the $\mathcal{CC}$ condition. Thus, it is impossible to mitigate all the criticalities.

7. SIMULATION STUDY

This section presents a simulation based study to better understand the behavior of the manageability of criticalities. For sake of simplicity, we present simulation results for a system which can expect only three types of criticalities, thereby generating an abnormal state hierarchy with three levels. Further, we assumed that two criticalities of the same type will not occur in the abnormal state, for example if c_1, c_2 and c_3 are the three criticalities which can occur in the system, the system will never experience criticalities such as - (c_1, c_2, c_1) or (c_1, c_2, c_2) or (c_3, c_3) and so on. It should be noted that this simplified system can be easily extended to accommodate multiple occurrences of same criticalities.

The simulator was developed on the .NET platform using $C\#$ programming language. The criticalities in our simulation model were implemented as timer events. Further, each criticality was assigned a separate window of opportunities, which were 60 (c_1), 120 (c_2) and 180 (c_3) seconds. Therefore, from the time of occurrence, there is exactly 1, 2, and 3 minutes to mitigate the criticalities. We implemented the state transition hierarchy as an adjacency matrix with the values representing the probabilities of state transition. The probabilities associated with CLs therefore determine the timer triggers that result in the lower level criticality. We assumed that the weight associated each ML is 10 units. The adjacency matrix used for this simulation is presented in Figure 7.

We first study the variation of the maximum Q-value for each abnormal state with respect to the periodicity of criticality detection (t_p) which determines the responsiveness. Figure 8 shows this variation. As expected we find that as the value of t_p increases, the maximum Q-value associated with a state either remains the same or decreases (drastically in some cases). This is because as

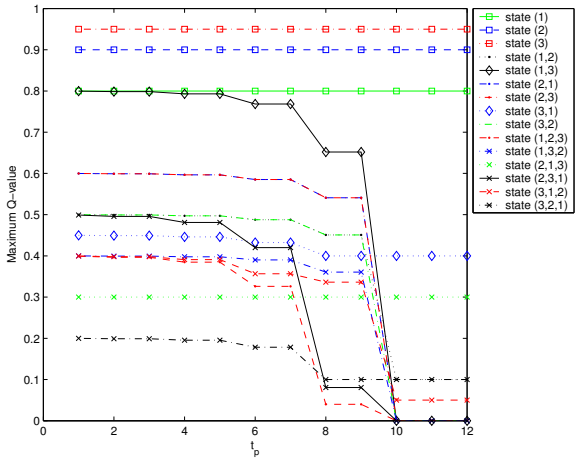

Figure 8: Variation of maximum Q-value w.r.t t_p

we increase the interval of criticality detection, we are essentially delaying the mitigative action after a criticality has occurred. This delay leads to, in some cases, the violation of the $\mathcal{CC}$, resulting in the un-manageability of that set of criticalities. Similar study on the variation of the average maximum Q-value with respect to each level in the hierarchy (Figure 9) shows that the number of criticalities, varies inversely to the average Q-value. This is because, increase in the number of criticality decreases the chances of satisfaction of the $\mathcal{CC}$.

From a given abnormal state, the system takes a path toward the normal state based on the maximum Q-values computed at each intermediate state. In this final experiment, we computed the average probabilities associated with the final path thus taken to reach the normal state from all abnormal state. We study the variation of this property with respect to t_p (Figure 10). We find that, in most cases, as the t_p increases the probability of successfully reaching the normal state from any abnormal state remains the same or decreases. In some cases, however, the probability of success counter-intuitively increases. This is because, as t_p increases, the path being taken to reach the normal state (one with the highest Q-value) does not meet the $\mathcal{CC}$. Therefore, we have to take another path with the next best Q-value and which meets the $\mathcal{CC}$, to reach the normal state. It is interesting to note that the maximum Q-value does not ensure the highest probability of success in the path thus taken, because the path with the maximum probability of success might have a CL with a high associated probability in the intermediate states, thus decreasing the Q-value (preventing it from being the best choice).

8. CONCLUSIONS

In this paper we presented and analyzed in detail the concept of criticality. We further built a criticality management framework and developed conditions which need to be met for handling single and multiple criticalities. To illustrate our framework's applicability, we simulated a criticality aware system. Future work includes employing the concept of criticality to real-life systems and study its manageability capabilities.

9. REFERENCES

[1] F. Adelstein, S. K. S. Gupta, G. Richard, and L. Schwiebert. *Fundamentals of Mobile and Pervasive Computing*. McGraw Hill, 2005.

[2] K. M. Chandy. Sense and respond systems. *31st Int. Computer Management Group Conference (CMG)*, Dec. 2005.

[3] Computer Science and Telecommunication Board, National Research Council. Summary of workshop on information technology research for crisis management. The National Academy Press, Washington D.C., 1999.

STATES	0	(1)	(2)	(3)	(1,2)	(1,3)	(2,1)	(2,3)	(3,1)	(3,2)	(1,2,3)	(1,3,2)	(2,1,3)	(2,3,1)	(3,1,2)	(3,2,1)
0	-	0.3	0.4	0.2	0	0	0	0	0	0	0	0	0	0	0	0
(1)	0.8	0	0	0	0.15	0.05	0	0	0	0	0	0	0	0	0	0
(2)	0.9	0	-	0	0	0	0.05	0.05	0	0	0	0	0	0	0	0
(3)	0.95	0	0	-	0	0	0	0	0.025	0.025	0	0	0	0	0	0
(1,2)	0.1	0.3	0.5	0	-	0	0	0	0	0	0.1	0	0	0	0	0
(1,3)	0	0.8	0	0.15	0	-	0	0	0	0	0	0.05	0	0	0	0
(2,1)	0	0.3	0.6	0	0	0	-	0	0	0	0	0	0.1	0	0	0
(2,3)	0	0	0.6	0.1	0	0	0	-	0	0	0	0	0	0.3	0	0
(3,1)	0.4	0.45	0	0.1	0	0	0	0	-	0	0	0	0	0	0.5	0
(3,2)	0	0	0.5	0.4	0	0	0	0	0	-	0	0	0	0	0	0.1
(1,2,3)	0	0	0	0	0.4	0.4	0	0.2	0	0	-	0	0	0	0	0
(1,3,2)	0	0	0.4	0	0.2	0.2	0	0	0	0.2	0	-	0	0	0	0
(2,1,3)	0.3	0.3	0	0	0	0	0.1	0.2	0	0	0	0	-	0	0	0
(2,3,1)	0	0	0	0	0	0	0.5	0.3	0.2	0	0	0	0	-	0	0
(3,1,2)	0.5	0	0	0.35	0.2	0.2	0	0	0.4	0	0	0	0	0	-	0
(3,2,1)	0.1	0.1	0.1	0.1	0	0	0.2	0	0.2	0.2	0	0	0	0	0	-

Figure 7: Abnormal State Transition Matrix

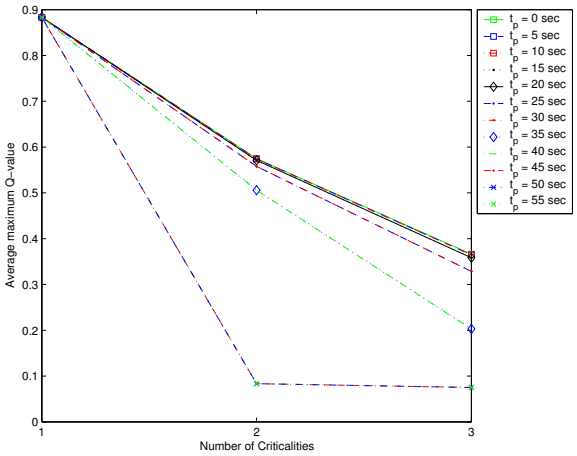

Figure 9: Variation of Q w.r.t Number of Criticalities

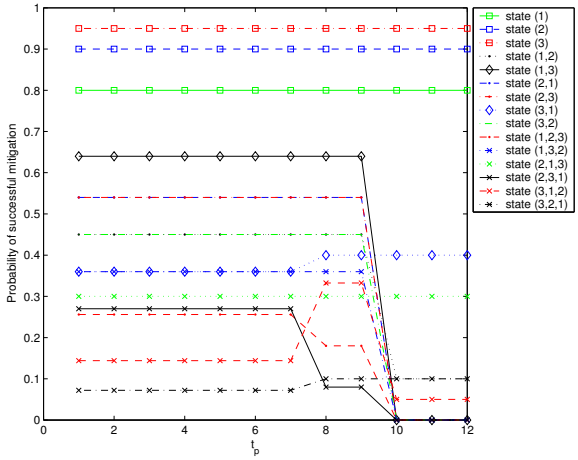

Figure 10: Variation of Probability of Success w.r.t t_p

[4] B. P. Douglass. http://www.nohau.se/articles/pdf/safcritdes.pdf.

[5] S. K. S. Gupta, T. Mukherjee, and K. K. Venkatasubramanian. Criticality aware access control model for pervasive applications. *PerCom*, pages 251–257. IEEE Computer Society, 2006.

[6] U. Gupta and N. Ranganathan. FIRM: A Game Theory Based Multi-Crisis Management System for Urban Environments. *Intl. Conf. on Sharing Solutions for Emergencies and Hazardous Environments*, 2006.

[7] J. C. Knight. Safety-critical systems: Challenges and directions. In *Proc of ICSE'02*, may 1992.

[8] N. Leveson. *Safeware: System Safety and Computers*. Addison-Wesley, 1995.

[9] J. W. S. Liu. *Real Time Systems*. Prentice Hall, 2000.

[10] S. Mehrotra et al. Project RESCUE: challenges in responding to the unexpected. In *Proc of SPIE*, volume 5304, pages 179–192, Jan. 2004.

[11] D. Mosse et al. Secure-citi critical information-technology infrastructure. In *7th Annual Int'l Conf. Digital Government Research (dg.o 06)*, may 2006.

[12] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):66–75, January 1991.

APPENDIX
A. PROOFS
A.1 Proof of Theorem 1

We first prove that at any state j, $\max_{\forall(j,i)\in ML(j)}(Q_{j,i,n}) > 0$ iff $P_{j,n} > 0$. If $P_{j,n} > 0$, then $APM(j) > 0$, as $PC(i) < 1$ and it is not possible that $APC(j) > 0$ and $APM(j) = 0$ because in that case it means that the occurrence of additional criticalities meet the $\mathcal{CC}$ condition but it is not met if there is no additional criticalities. Therefore, there is at-least one outgoing ML from j, say (j, u), such that $p_{j,u}P_{u,n} > 0$ or $Q_{j,u,n} > 0$, which implies that $\max_{\forall(j,i)\in ML(j)}(Q_{j,i,n}) > 0$.

Now, if $\max_{\forall(j,i)\in ML(j)}(Q_{j,i,n}) > 0$, there is at least one state u such that $p_{j,u}P_{u,n} > 0$. Therefore, both $p_{j,u}$ and $P_{u,n}$ are greater than 0. It implies that there is at least one path from j to n (through state u), for which the probability of success to reach n is greater than 0, making $P_{j,n} = (1 - \sum_{(j,k)\in CL(j)} p_{j,k})p_{j,u}P_{u,n} + \sum_{(j,k)\in CL(j)} p_{j,k}P_{k,n} > 0$ because $\sum_{(j,k)\in CL(j)} p_{j,k} < 1$.

Now, we prove that the maximum Q-value from a state is greater than 0 iff there is at least one path from that state to n. We prove the *if* part by induction on the level of any state x (L_x) in the state hierarchy. If $L_x = 1$, then $\max_{\forall(x,i)\in ML(x)} Q_{x,i,n} = p_{x,n} > 0$. It is therefore obvious that there is a direct link from x to n. We assume that if L_x is r, then there is at least one path from x to n. Now, for $L_x = r + 1$, $\max_{\forall(x,i)\in ML(x)} Q_{x,i,n} > 0$ implies, that there is at least one outgoing ML from x, say (x, j), for which $Q_{x,j,n} > 0$ i.e. both $p_{x,j} > 0$ and $P_{j,n} > 0$. From above, it follows that $\max_{\forall i\in ML(j)}(Q_{j,i,n}) > 0$. As $L_j = r$ (because (x, j) is an outgoing ML from x), it follows from the induction hypothesis that there is at least one path from j to n. Therefore, there is a path from state x to n through j.

We prove the *only if* part also by induction on L_x. If $L_x = 1$, then $\max_{\forall(x,i)\in ML(x)} Q_{x,i,n} = p_{x,n}$. As there is a path from x to n, it follows that $p_{x,n} > 0$. We assume that if $L_x = r$ and there is a path from x to n, then $\max_{\forall(x,i)\in ML(x)} Q_{x,i,n} > 0$. Now, for $L_x = r + 1$, if there is a path from x to n, it follows that there is at-least one outgoing ML from x, say (x, j), such that $p_{x,j} > 0$ and $\max_{\forall(j,i)\in ML(j)}(Q_{j,i,n}) > 0$ (from the induction hypothesis). Therefore, we have $p_{x,j} > 0$ and $P_{j,n} > 0$ making $\max_{\forall(x,i)\in ML(x)} Q_{x,i,n} > 0$.

A.2 Proof of Corollary 1

The maximum time to mitigate the criticalities from given state i is $t_{i,j}$ if (i, j) has maximum Q-value at i. Therefore the total amount of time to reach n from i is given by $t_{i,n} = t_{i,j} + t_{j,n}$, where, $t_{i,n}$ is the time required for reaching n from i (The Q-value calculation ensures the maintenance of the $\mathcal{CC}$ condition and reduces $t_{i,n}$ to 0 if a path does not exist).