CRET: A Crisis Response Evaluation Tool to improve Crisis Preparedness

Tridib Mukherjee and Sandeep K. S. Gupta
Impact Lab, Department of Computer Science and Engineering
School of Computing & Informatics
Arizona State University, Tempe, AZ 85287
http://impact.asu.edu
sandeep.gupta@asu.edu

***Abstract*—In recent years, crisis response has become cyber-physical in nature because of the increased use of computing technologies by the responders. As such, crisis preparedness requires objective evaluation of crisis response in addition to the traditional drills. This paper develops a generic *Crisis Response Evaluation Tool (CRET)* for off-line objective crisis response evaluation to improve preparedness. The evaluation is performed through model-based engineering, which allows specification and automated analysis of crisis response behavior. An established state-based stochastic model is used to describe the behavior of crisis response processes. The effectiveness of a planned action is measured in terms of the action's *qualifiedness* (also called the *Q-value*)—which depends on the probability of any additional crises and the conformance to a temporal window-of-opportunity before which any action has to be performed. CRET uses Abstract Architecture Description Language (AADL) to specify the stochastic crisis response behavior model. Using this specification, CRET objectively analyzes the planned actions' Q-values under different circumstances; thus enabling an objective evaluation for crisis preparedness.**

***Index Terms*—Crisis Response, Crisis Preparedness, Criticality, Stochastic Model, Human-centered computing.**

I. Introduction

Crisis preparedness is getting increasing attention in the Department of Homeland Security (DHS). In 2005, $7.4 billion fund was budgeted for emergency preparedness and response, which was around 20% of the total DHS budget. Since 2003, the DHS has awarded approximately $8 billion to the state governments to improve crisis preparedness. Given the importance of crisis preparedness, it is imperative to develop proper technologies to aid crisis preparedness. With the advent of critical computing tools and facilities in recent years [1]–[3], crisis management is becoming more and more cyber-physical in nature consisting of three major components—i) human inhabitants, ii) physical components, and iii) computing systems.

Most of the available preparedness measures focus on how to prepare the human inhabitants to behave in response to the crises [4]–[10]. This paper seeks to compliment this approach by *planning and preparing* in a synergistic manner such that the crisis response systems become aware of the inhabitants' behavior; thus enabling high confidence cyber-physical crisis response system design and development.

A report from the National Academy of Sciences [11] 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." Verification of crisis response using real-life scenarios is undesirable because of the extreme nature of the crises putting lives at risk. Consequently, automated and objective evaluation, verification, analysis and planning of crisis response is imperative. As such, this paper uses the concept of model based verification and analysis—a widely used approach for automated design and verification of critical computing systems—from the software engineering domain. Such an approach can analyze the functional and non-functional requirements of systems depending on the systems' behavioral models. Crisis response behavior model, developed in our previous work [12], is used for crisis response verification and analysis in this paper.

In this regard, the concept of *criticality* is used to describe crises. The changes in the system's environment which eventually lead the system into a crisis (loss of life/property) are called critical events. The immediate effects of the critical events are defined as criticalities. Each criticality is associated with a window-of-opportunity within which any response should be planned, facilitated, and actuated to avoid loss of lives and properties. A Criticality Response Modeling (CRM) framework has also been established [12] which uses a state-based stochastic model to describe the behavior of criticality responses. The CRM framework identified an objective evaluation metric, called the *Q-value* (or *qualifiedness*), to measure the effectiveness of the response actions in managing crises.

This paper uses the CRM framework and Q-value metric to develop a preparedness tool, which involves verification and analysis of the planned response actions using the state-based stochastic behavior model. The major contribution (summarized in Figure 1) of the paper is:

- **Criticality Response Evaluation Tool (CRET)** using Abstract Architecture Description Language (AADL)—most widely used standard from SAE International for model-based engineering of real-time embedded and safety-critical systems [13]—for CRM and response planning policy specification. Analyses were performed to process

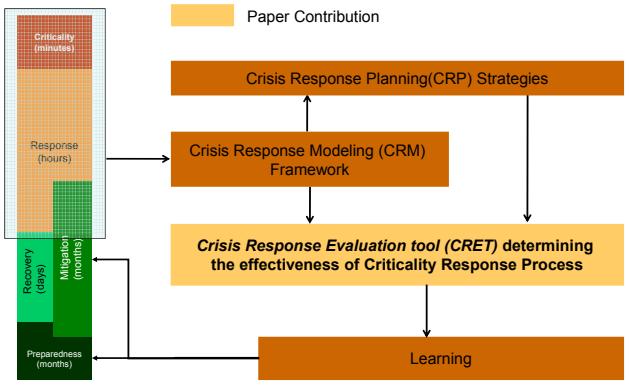

Fig. 1. Paper Contributions.

the AADL specification in terms of the achieved Q-values of the planned response actions.

The rest of the paper is organized as follows: Section II presents the related work on the crisis preparedness tools, model based engineering and probabilistic planning. Section III provides the preliminaries on criticality, criticality response behavior model, CRM framework, response planning strategies, and model based engineering using AADL. Section IV presents the AADL based CRET tool followed by a discussion on handling dynamic crisis scenarios. Finally, Section VI concludes the paper.

II. Related Work

This sections provides a brief overview on the related work on crisis preparedness, model-based engineering and probabilistic planning. The most important crisis preparedness tools [4] include—i) the automated and on-demand research services such as ewatch [5], CustomScoop [6], Factiva [7], and The Wall Street Journal Online [8]; ii) videoconferencing/webcasting for meetings and training; and iii) resources such as the Public Information Emergency Response (PIER) system [9] that enable creation and launching of instant websites for a crisis without a web-master. All these tools and many others lead to cumbersome documents, instructions, personnel development and so on [10], and none provide an objective crisis response evaluation. This paper develops an evaluation tool to fill this gap. In this regard, model based engineering from the software engineering domain is used.

The conventional computer system specification and verification models—e.g. discrete event models, finite state machines, finite automata, data-flow model, and synchronous event models—have normally leveraged the discrete time behavior of the traditional computer systems [14]. Cyber-Physical Systems (CPS)—crisis response is a special type of CPS (Section I)—however control the continuous time physical processes. There has been a surge in formal methods for automated verification of such systems since the 1990s [15], [16]. Although these models incorporate the continuous time behavior of the physical processes, they do not capture the uncertainties of the process steps as in crisis response due to the human interactions. The same problem persists for models developed for the mission-critical and safety-critical systems (e.g. avionics) [17], [18]. Although stochastic automata captures the probabilistic nature of events' occurrences [19], the actions triggered by the events are assumed to be deterministic. A state-based stochastic behavior model for crisis-response was proposed in [20]. This paper uses the stochastic behavior model to analyze and verify the planned actions for crisis response.

III. Preliminaries

This sections provides a brief overview on the theory of criticality including the stochastic crisis response behavior model from our previous work [20]. The planning and evaluation requirements are then identified followed by a overview on model-based engineering using the industry standard AADL.

A. Criticality: Concepts and Characteristics

Criticality is defined as follows:

Definition: Criticality is the effect on the CPS 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.

A CPS under one or more criticalities are said to be in a **critical state**, otherwise it is in the **normal 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**. Response actions to mitigate the criticalities (thus taking the CPS from a critical state to the normal state) are stochastic in nature due to possible human involvement in performing the actions.

B. State-based Stochastic Model

Criticality response behavior can be modeled as state-based stochastic systems, where the occurrences of the criticalities take the system to the critical states from the normal state. Mitigative actions are performed to bring the system back to the normal state. Critical state further consists of several critical sub-states. When a criticality occurs in the system (in normal state), it moves into the critical state. All subsequent criticalities keep the system in the critical state. The critical state encompasses many system states which are reached in response to the occurrence of different criticalities. State transitions occur as a response to either critical events (the associated link is called the *critical link* (CL)) or response or mitigative actions (the associated link is called the *mitigative link* (ML)). Each state transition is associated with a probability value, capturing the probability of occurrence of the criticalities and the successful completion of the response actions for the CLs and MLs, respectively.

C. Criticality Response Modeling (CRM) Framework

Criticality Response Modeling (CRM) framework identifies the different components of the stochastic model to apply for real crisis situations [12]. As depicted in Figure 2, the CRM framework consists of five main components:

1) Identification of criticalities and responses.
2) Determination of windows-of-opportunity.
3) Determination of critical states.
4) Determination of critical state transition probabilities.
5) Application of the stochastic model to plan and evaluate the response action at any critical state.

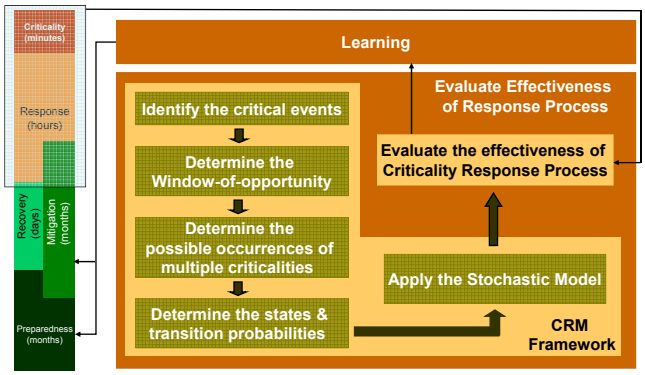

Fig. 2. Modeling Framework for Criticality Management [12].

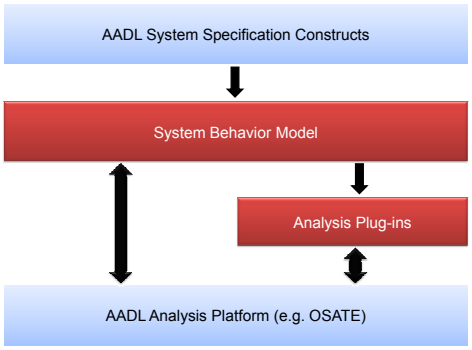

Fig. 4. System behavior analysis in AADL.

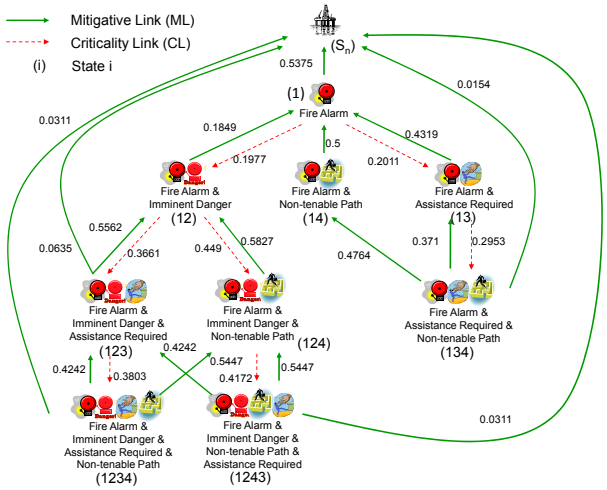

Fig. 3. State transition diagram for the criticalities and response actions in the scenario of fire and explosion in OGPP [12].

Figure 3 shows the application of the stochastic model for fire emergencies in offshore Oil & Gas Production Platforms (OGPP). The four criticalities identified are—i) fire alarm ($c1$), ii) imminent health dangers to the occupants ($c2$), iii) trapped occupants (assistance required) ($c3$), and iv) evacuation path not tenable ($c4$). The critical states are identified based on the evacuation process employed for the fire emergencies in OGPP [21]. The transition probabilities are based on a previous study on the human error probabilities in the OGPP [21]. The windows-of-opportunity are taken as the average time for survival of humans under asphyxiation [12]. The following subsection discusses how the stochastic model can aid in crisis response planning and evaluation.

D. Evaluation of Criticality Response

The response planning at any state requires the determination of the ML while the evaluation of the response involve the determination of the *qualifiedness (Q-value)* of the planned ML at any state. The Q-value is calculated based on:

1) the probabilities of the selected actions (i.e. the probabilities associated with the corresponding MLs);
2) the probabilities of additional criticalities after taking the planned actions (i.e. the probabilities associated with the CLs originating from the intermediate states); and
3) the conformance to the windows-of-opportunity.

Previous work has verified the applicability of Q-value by applying the CRM framework for fire emergencies in OGPP [12]. This paper develops CRET, the response evaluation tool (Section IV) using model-based engineering approach. Industry standard AADL is used for this purpose.

1) Verification using Abstract Architecture Description Language (AADL): Abstract Architecture Description Language (AADL) is a widely used standard from SAE International to verify functional and non-functional requirements of critical real-time embedded systems [13]. It enables automated analysis of such systems by providing component based framework to model hardware and software components and their interactions. The power of the language stems from its ability to describe the systems' behavior at different level of abstractions, ranging from very high level system design to low level software and hardware design. Figure 4 shows how AADL constructs are used to specify and analyze (verify) a system behavior. Crisis preparedness can benefit if such abstraction capabilities can be used to describe the crisis response behavior model and analyze the planned actions for criticality response. Before getting into the details on the AADL based crisis response analysis, the following subsection gives an overview on the previously developed Crisis Response Planning (CRP) policies. The CRM framework and the CRP policies are then represented and analyzed using AADL in Section IV.

E. Criticality Response Planning (CRP)

An optimal action at any state (in the pursuit to reach the normal state) is the one which has the maximum Q-value at that state. However, the maximum Q-value calculation is a highly complex operation, which can be exponential to the number of critical states [12]. In this paper we use, different heuristic planning algorithms from our previous work [22] based on: i) Markov Decision Planning (MDP) technique; and ii) the actions' properties at each state.

1) Markov Decision Planning (MDP) for Criticality Response: Markov Decision Planning (MDP) is a special type of probabilistic planning technique which ensures reaching a goal state while optimizing the expected utility values [23]. This approach associates a *utility* function, for each action at each state, with a recursive definition based on: i) the immediate

reward if the action is performed at the state; ii) the probability of success in performing the action (this is analogous to the transition probability associated to the corresponding ML); and iii) the expected value of the optimal utility at the next state after the action is performed. The criticality response actions are associated with a utility function in the following MDP based response planning strategies [22]:

1) Locally maximum criticality Mitigation Per unit Time (LMPT); and
2) Subsequent Criticality Aware locally maximum criticality Mitigation Per unit Time (SCAMPT).

LMPT assigns the immediate reward to the actions based on the actions' time and the number of criticalities mitigated; without the knowledge of subsequent criticalities. SCAMPT has a similar reward for the actions however SCAMPT considers the subsequent criticalities. SCAMPT gives weights to both the number of criticalities mitigated and the time to perform the action based on the probabilities of the subsequent criticalities due to the action. Different techniques, such as value iteration and policy iteration [23], can be used to generate the response plan yielding optimal utility.

2) Greedy Planning Strategies: There can be several greedy strategies where at each state, the *immediate reward or cost is maximized or minimized, respectively*. Following are the three basic greedy strategies: i) *Maximum Probability (MP)*, which selects actions with maximum success probabilities; ii) *Minimum Time (MT)*, which selects actions with minimum associated time to perform them; and iii) *Maximum Mitigated Criticalities (MMC)*, which selects the actions such that the number of immediate criticalities mitigated is maximized. It should be noted here that there can be different greedy strategies depending on the composition of multiple state-local parameters. This is also true for the MDP strategies where the immediate reward can be designed based on the combination of multiple state-local parameters.

IV. Criticality Response Evaluation Tool (CRET)

Given the background on the criticality response modeling and planning, this section uses model based verification and analysis of crisis response to develop a crisis preparedness tool. In this regard, AADL is used to describe the crisis response behavior model. This section first describes the abstract architecture for crisis response systems followed by the specification of different components of the CRM framework using the AADL language constructs.

A. AADL based architecture description

The criticality response system architecture has three types of system components—i) the criticality detection components, which monitors for the possible criticalities in the system; ii) the criticality response decision making component, which uses the state based stochastic model, CRM framework, and the CRP strategies to plan for the criticality response actions; and iii) the response actuation components, which is responsible to perform the actions planned by the decision making component. Figure 8 shows how these components interact with each other. When a criticality detection component detects any criticality in the system, it notifies the decision making component. The decision making component considers such notification as input event

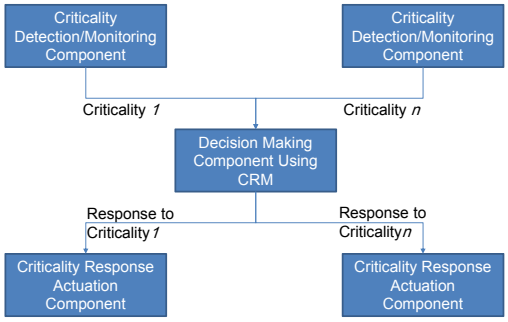

Fig. 5. Abstract architecture description for criticality response.

to the component, which takes the system to a critical state. Response actions are then planned using the CRP strategies and notified to the actuation component.

CRET Spec 1 OGPP System Specification in AADL

```
  ⋮
system implementation OGPP.imp1
 subcomponents
   FM: system FireMonitoringComponent.imp1;
   EM: system EvacuationMonitoringComponent.imp1;
   DM: system DecisionMakingComponent.imp1;
     ⋮
system implementation DecisionMakingComponent.imp1
 subcomponents
   P1: process CRM.imp1;
   ⋮
end DecisionMakingComponet.imp1
 ⋮
```

In the example of fire emergencies in OGPP (Section III-C), there are two detection components—i) the fire monitoring component, and the ii) evacuation monitoring component. The former detects the occurrence of fire and explosion in OGPP, while the later is responsible for detecting any trapped people, non-tenable evacuation path, and imminent health hazards. Based on the detected criticalities, the decision making component determines the current system state and uses a specified CRP strategy to plan the response actions. Specification 1 shows a sample AADL specification of the different components of OGPP. For verification of the crisis response, it is imperative to specify the response behavior model in the decision making component and analyze the Q-values of the planned response actions. As such, this paper focuses on the decision making component to specify the different aspects of the CRM framework. The next subsection describes the various constructs in the AADL for this purpose.

B. AADL Constructs Used

The decision making component has four basic aspects based on the CRM framework: i) Criticality Specification, ii) State Specification, iii) Transition Specification, and iv) Response Policy Specification. To this effect, the following AADL constructs have been identified:

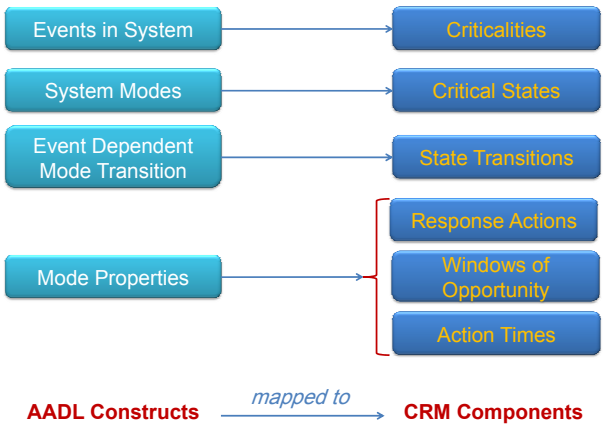

Fig. 6. Mapping of *identified* AADL constructs to the CRM components.

- **Modes:** *Modes* in AADL correspond to different operational states of a system. For criticality response, modes can be associated with the normal and critical states in the stochastic model.
- **Mode Transitions & Events:** *Mode transitions* refer to the *events* that are responsible for changing the operational state of a system. For criticality response, mode transitions correspond to the critical state transitions and the events correspond to criticalities.
- **Mode Properties:** Attributes of a system in any mode can be described by *properties* in AADL. Properties are grouped together in a Property set. In AADL we can associate properties with modes, as a result we can capture mode specific properties of the system. For crisis response, the windows-of-opportunity, the action times and probabilities of success can be provided as properties of the critical states. Such provisioning captures the situation-dependency of the window-of-opportunity for any criticality by associating it to the critical states.

Figure 6 depicts the mapping of the identified AADL constructs to specify the different components of the CRM framework. The following subsections describe the mapping along with the example specification of the stochastic model for the fire emergencies in OGPP (Figure 3).

C. Criticality Specification

The process *CRM* inside the decision making component (as shown in Specification 1) implements a thread *Criticalities&Responses* to specify the criticalities and response actions. The criticalities are specified as the incoming events from the monitoring subcomponents (see Specification 2). Similarly, the response actions are specified as the outgoing events. The windows-of-opportunity of the criticalities are specified as part of the property associated to each state (Specification 4). Such state-dependent property allows the window-of-opportunity to be specified as a situation dependent parameter. Users can input different windows-of-opportunity for different system states.

CRET Spec 2 OGPP Criticality Specification

```
  ⋮
process implementation CRM.impl
  subcomponents
    thread: thread Criticalities&Responses.impl;
    ⋮
end DecisionMakingComponent.impl
  ⋮
thread Criticalities&Responses
features
  c1: in event port; --fire alarm
  c2: in event port; --imminent health hazard
  c3: in event port; --assistance required to others
  c4: in event port; --non-tenable path
  a1: out event data port; --responses
    ⋮
end Criticalities&Responses;
  ⋮
```

D. State and Transition Specification

Each state is specified by a mode as shown in Specification 3. The state numbers ($s1$ to $s1234$) follow the same numbering as in Figure 3. Specification 3 also shows the CLs for each state in Figure 3 . The MLs can also be similarly specified.

CRET Spec 3 OGPP State and State Transition Specification

```
thread implementation Criticalities&Responses.impl
  modes
    Normal: initial mode;              --normal state
    s1: mode;                          --state (1)
      ⋮
    s1243: mode;                       --state (1243)
      ⋮
    Normal - [c1] -> s1;  --CL at the normal state
    s1 - [c2] -> s12;     --CLs at state (1)
    s1 - [c3] -> s13;
    s12 - [c3] -> s123;   --CLs at state (12)
    s12 - [c4] -> s124;
    s13 - [c4] -> s134;   --CL at state (13)
    s123 - [c4] -> s1234; --CL at state (123)
    s124 - [c3] -> s1243; --CL at state (124)
    ⋮
end Criticalities&Responses;
  ⋮
```

E. State Property Specification

The state properties include—i) the transition probabilities associated with the CLs and MLs, ii) the time to perform actions associated with the MLs, iii) the windows-of-opportunity of the criticalities in the state, iv) response policy, and v) name (description) of the state. These properties are first declared as the property-set for all states. Specification 4 shows the sample state property specification for the fire emergencies in OGPP. Property values for only one state, $s1$, is provided. Specifications for the other states follow the same format.

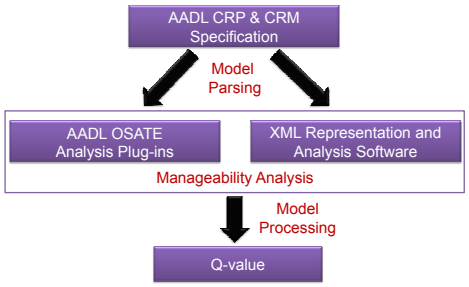

Fig. 7. Manageability Analysis: a combination of model parsing and model processing. Two different analysis mechanism was developed—i) developing analysis plug-ins for the AADL/OSATE tool; and ii) intermediate model representation using generic XML syntax and developing analysis software.

CRET Spec 4 OGPP State Property Specification

```
  ⋮
property set state_properties is
  actions:     list of aadlstring applies to (all);
  mlProbs:     list of aadlreal applies to (all);
  clProbs:     list of aadlreal applies to (all);
  critStates: list of aadlstring applies to (all);
  actionTime: list of aadlreal applies to (all);
  window-Opp: list of aadlreal applies to (all);
  policySpec: Enumeration (maxq, MP, MT, MMC,
                  SCAMPT, LMPT) applies to(all);
  modeName:   aadlString applies to (all);
end state_properties;
      ⋮
thread implementation Criticalities&Responses.impl
   ⋮
  properties
    state_properties::policySpec => maxq;
    state_properties::mlProbs => (0.5376)
                                  in modes (s1);
    state_properties::clProbs => (0.2613,0.2011)
                                    in modes(s1);
    state_properties::critStates => ("s12","s13")
                                    in modes(s1);
    state_properties::window-Opp => (10.0)
                                    in modes(s1);
    state_properties::modeName => "Fire in OGPP"
                                    in modes(s1);
      ⋮
end Criticalities&Responses.impl;
  ⋮
```

F. Response Policy Specification

The response action selection criteria at any state is specified by the policy specification. The policy specification should allow for any CRP policy specification in future and should also facilitate customization of any action selection policy. To allow for such generic extensibility, we further convert the AADL model in XML syntax, which allows for any policy customization (Figure 8). To this effect, there are three options provided to specify the response actions—i) *per-state*, ii) *state-independent*, and iii) *combination of the per-state and state-independent* response action specification. The MDP based strategies can be specified as a state-independent specification, since the action selection criteria are same for all the states. Similarly, the optimal strategy (select response actions with maximum Q-values) requires state-independent policy specification. A per-state policy specification allows the flexibility to specify different action selection criteria in different states. A combination of the per-state and state-independent specification is more general and allows all the states to have a same selection criteria except the ones with different specification.

G. Evaluation of the specified criticality response model

The evaluation is performed in two different was—i) through developing manageability analysis plug-ins for the AADL model (the OSATE tool is used for this purpose); and ii) through intermediate representation of the AADL model in XML syntax and developing generic tool for manageability analysis. The AADL/OSATE analysis plug-in uses the available APIs to parse the AADL model and uses the parsed information to calculate the Q-value of the planned actions as described in Section III-D. These information were further used to develop an intermediate representation of the AADL model in XML syntax. This representation allows for more generic CRP specification, transcending beyond the proposed CRP strategies (Section IV-F). The Q-values of the generated response actions are then calculated for evaluation.

V. Discussion

This section discusses various open issues in CRET for crisis preparedness in dynamic and composite crisis situations.

A. Dynamics in Crisis Response

In many scenarios, crisis response is dynamic and situation dependent in nature. For example, a Unmanned Air Vehicle (UAV), used to accomplish a mission in the enemy territory, can get tracked by the enemy. The windows-of-opportunity and the response action success probabilities for such criticalities is dynamic and depends on the enemy capabilities, which may be unknown beforehand. Such dependency could be a function of the number of enemy threats, types of the threats, enemy mobility and distance to the UAV. The critical state could depend on the phase in which the UAV is in the enemy *killchain* (F2T2EA—fix, find, track, target, engage and assert). Depending on the number of enemy threats, the state space would dynamically change.

Figure 9 shows a simple scenario. There is one UAV to find-and-destroy a set of enemy targets inside the high risk enemy territory. The scenario has only two enemy Surface-to-Air Missile (SAM) launching stations. The phases of the killchain between one SAM and one UAV are mapped to the possible critical states. There are two branches in the state transition to represent threats due to two SAMs. With increase in SAMs and/or UAVs the state transition branches can be dynamically added. The window-of-opportunity to respond to the first criticality (i.e. when in Find state of any branch) depends on the average time to get destroyed in the kill-chain after getting fixed by the enemy. When in Engage state, the window-of-opportunity depends on the maximum speed of the UAV and the missile launched by an enemy SAM.

To handle such dynamic scenarios, CRET needs to allow for specifying the dependencies of the CRM parameters (e.g.

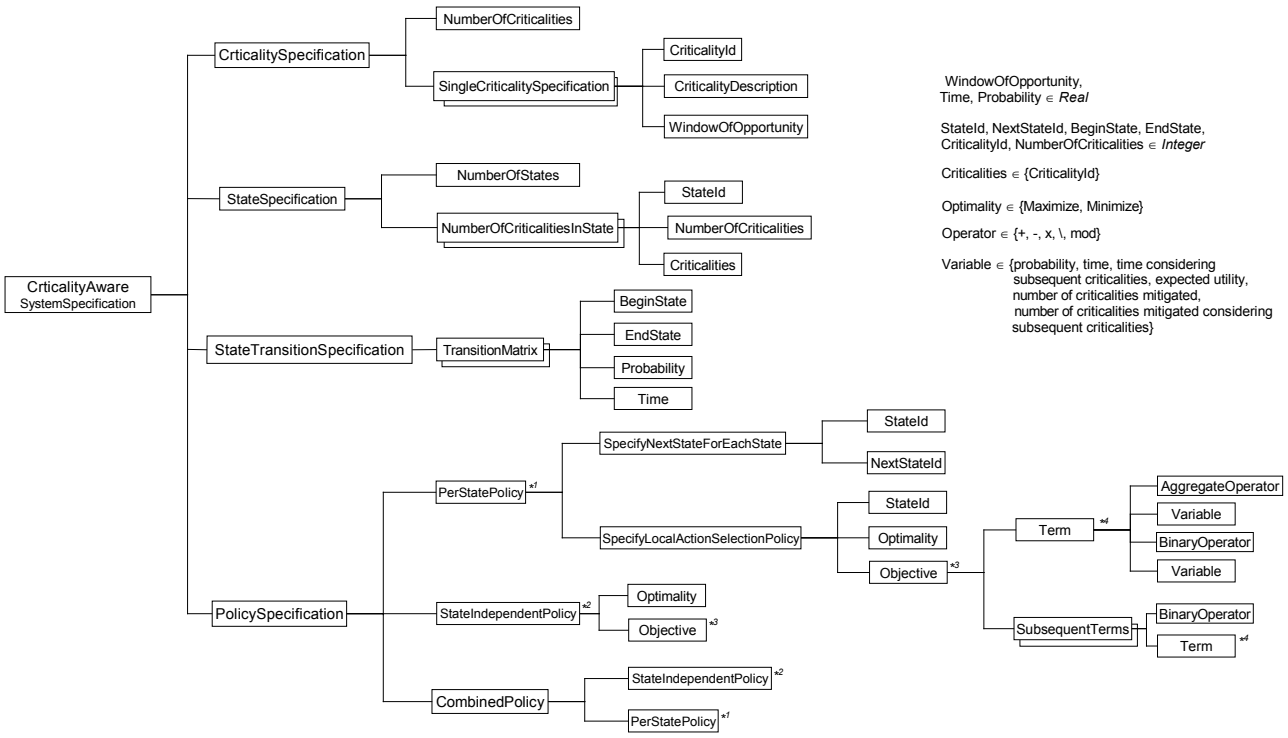

Fig. 8. Hierarchical view of the XML schema for criticality aware system specification in CRET.

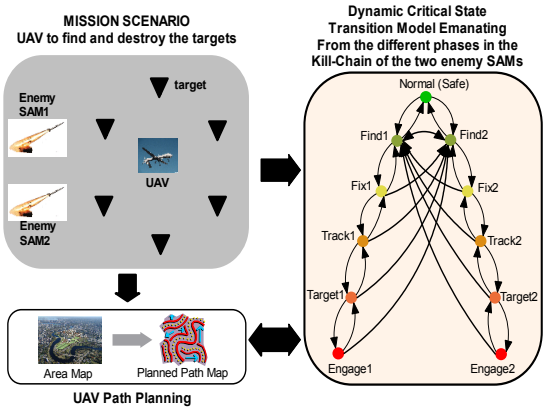

Fig. 9. Dynamic model generation example for a UAV engaged in killchain by two enemy surface to air missiles [24].

windows-of-opportunity, probability for CLs and MLs, etc.). While analyzing the crisis response, such dependencies need to be taken into account to determine the Q-values.

B. Distributed Decision Making and Model Composition

To use CRET for preparedness of an emergency situation, the stochastic model needs to be first mapped to the emergency situation. There can be different models for different emergency situations and the composition of the models may be important if there is a combination of multiple emergencies.

For example, if there is a fire-breakout in a hospital, then there are two types of emergencies—i) medical emergencies; and ii) fire emergency (which in effect might lead to additional medical emergencies). Independently, these two emergencies have different state-spaces for the stochastic model and different model parameter values. For the fire emergency, the state space would be similar to that of the OGPP (Figure 3). For the medical emergencies, the state space would involve the medical state of the patients and the state-transitions would depend on the medical equipment used. However, if these two emergencies occur together, the state space would be a composition of the individual state spaces. The windows-of-opportunity and the actions' success probabilities could become different for a composed stochastic model. Composition of the models for such different concurrent emergency situations therefore would involve—i) identification of the composite state space (this would be a subset of the cross-product of the independent state spaces); and ii) determination of the model parameters for the new composite state space (this would depend on the dependencies among the emergencies).

An alternative could also be to distribute the decision making component (Figure 8) to different sub systems. For example, in case of the above scenario, the hospital system could be designed to compose of medical subsystems and fire-management subsystems. Each subsystem could have its own decision making component (Figure 8) for crisis response. The model composition would incorporate the interactions among the different subsystems. It is possible that the criticality in one subsystem gets automatically handled by the other subsystem.

For example, if there is a fire in the hospital, then the resulting medical emergency may be handled by the medical subsystems, not affected by the fire.

VI. Conclusions

Crisis Response Evaluation Tool (CRET) has been developed to improve the crisis preparedness—a major concern in homeland security. The concept of criticality and the Criticality Response Modeling (CRM) Framework was used for this purpose. The CRET tool use the stochastic state-based response behavior model supported by the CRM framework. CRET used AADL to specify the stochastic crisis response behavior described by the CRM framework along with the Criticality Response Planning (CRP) policy specification. In future, CRET can integrate crisis preparedness and planning with the critical embedded systems in various domains—e.g. Unmanned Air Vehicles (UAVs) and smart-cars. Such systems already use model based engineering using AADL. Preparedness capabilities in such systems would improve the crisis response design through synergistic evaluation of both the computing entities and the physical environment.

Acknowledgments

The authors are thankful to Sailesh Kandula and Krishna Kumar Venkatasubramaniam for their technical contributions.

References

[1] S. Mehrotra *et al.*, "Project RESCUE: challenges in responding to the unexpected," www.itr-rescue.org/.

[2] D. Mosse *et al.*, "Secure-citi critical information-technology infrastructure," www.cs.pitt.edu/s-citi/.

[3] U. Gupta and N. Ranganathan., "Firm: A game theory based multi-crisis management system for urban environments," in *Intl. Conf. on Sharing Solutions for Emergencies and Hazardous Environments*, 2006.

[4] J. Berstein, "The future of crisis-preparedness, april 16, 2003," http://www.bernsteincrisismanagement.com/docs/the_future_of_crisis-preparedness_an_interview_with_jonathan_bernstein.html.

[5] "Ewatch," www.ewatch.com/.

[6] "Customscoop," www.customscoop.com/.

[7] "Factiva," www.factiva.com/.

[8] "The wall street journal online," www.wsj.com/.

[9] "Public information emergency response (pier) system," www.piersystem.com/.

[10] K. Felix, "Leadership for it preparedness.(useful tools)(crisis preparedness: Leadership for it disaster recovery)(brief article), september 1, 2006," http://www.highbeam.com/doc/1G1-152583949.html.

[11] 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.

[12] T. Mukherjee and S. K. S. Gupta, "A modeling framework for evaluating effectiveness of smart-infrastructure crises management systems," in *IEEE International Conference on Technologies for Homeland Security (HST)*. Waltham, MA, USA: IEEE Computer Society Press, 2008.

[13] "Abstract architecture description language (aadl)," www.aadl.info/.

[14] S. Edwards, L. Lavagno, E. Lee, and A. Sangiovanni-Vincentelli, "Design of embedded systems: formal models, validation, and synthesis," *Proceedings of the IEEE*, vol. 85, no. 3, pp. 366–390, Mar 1997.

[15] N. Lynch, R. Segala, and F. Vaandrager, "Hybrid i/o automata," *Inf. Comput.*, vol. 185, no. 1, pp. 105–157, 2003.

[16] D. K. Kaynar, N. Lynch, R. Segala, and F. Vaandrager, "Timed i/o automata: A mathematical framework for modeling and analyzing real-time systems," in *RTSS '03: Proceedings of the 24th IEEE International Real-Time Systems Symposium*. Washington, DC, USA: IEEE Computer Society, 2003, p. 166.

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

[18] M. Hinchey, P. C. Michael Jackson, B. Cook, J. P. Bowen, and T. Margaria, "Software engineering and formal methods," *Communications of the ACM (CACM)*, vol. 51, no. 9, pp. 54–59, Sep 2008.

[19] J. Bryans, H. Bowman, and J. Derrick, "Model checking stochastic automata," *ACM Trans. Comput. Logic*, vol. 4, no. 4, pp. 452–492, 2003.

[20] T. Mukherjee, K. Venkatasubramanian, and S. K. S. Gupta, "Performance modeling of critical event management for ubiquitous computing applications," in *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*. Terromolinos, Spain: ACM Press, 2006.

[21] D. G. DiMattia, F. I. Khan, and P. R. Amyotte, "Determination of human error probabilities for offshore platform musters," *Journal of Loss Prevention in the Process Industries*, vol. 18, pp. 488–501, 2005.

[22] T. Mukherjee and S. K. S. Gupta, "CRM: A Formal Method to Model & Evaluate Crises Response of Distributed Cyber-Physical Systems." Submitted to the *IEEE Transaction on Parallel and Distributed Systems (TPDS)*.

[23] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: a survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[24] T. Mukherjee and S. K. S. Gupta, "MCMA+CRET: A Mixed Criticality Management Architecture for Maximizing Mission Efficacy and Tool for Expediting Certification of UAVs." in *IEEE International Workshop on Mixed Criticality: Roadmap to Evolving UAV Certification*, San Francisco, CA, USA, 2009.