Towards Self-Adaptive Disaster Management Systems

Kenneth Johnson
Department of Computer Science
Auckland University of Technology
kenneth.johnson@aut.ac.nz

Javier Cámara
Department of Computer Science
University of York
javier.camaramoreno@york.ac.uk

Roopak Sinha
Department of Computer Science
Auckland University of Technology
roopak.sinha@aut.ac.nz

Samaneh Madanian
Department of Computer Science
Auckland University of Technology
sam.madanian@aut.ac.nz

Dave Parry
Department of Computer Science
Auckland University of Technology
dave.parry@aut.ac.nz

ABSTRACT

Disasters often occur without warning and despite extensive preparation, disaster managers must take action to respond to changes critical resource allocations to support existing health-care facilities and emergency triages. A key challenge is to devise sound and verifiable resourcing plans within an evolving disaster scenario. Our main contribution is the development of a conceptual self-adaptive system featuring a *monitor-analyse-plan-execute* (MAPE) feedback loop to continually adapt resourcing within the disaster-affected region in response to changing usage and requirements. We illustrate the system's use on a case study based on Auckland city (New Zealand). Uncertainty arising from partial knowledge of infrastructure conditions and outcomes of human participant's actions are modelled and automatically analysed using formal verification techniques. The analysis inform plans for routing resources to where they are needed in the region. Our approach is shown to readily support multiple model and verification techniques applicable to a range of disaster scenarios.

Keywords

disaster management, self-adaptive systems, formal verification, probabilistic model checking, constraint solving

INTRODUCTION

Disasters occur suddenly and often without warning. They have the potential to cause loss of human life, impact health, and cause damage to land, buildings and transportation infrastructure. In response, disaster managers must consider plans coordinating personnel and first-responders to distribute scarce resources with the disaster-affected region. Resource allocation plans form the basis for disaster management, disaster medicine and more recently, disaster healthcare (Madanian et al. 2020). During preparation phases of disaster management, sufficient resources are allocated at every location in the region to satisfy resource demands. Finding a suitable allocation essentially corresponds to a constraints solving problem $\alpha \models K$, where resourcing demands are formalised as logical predicates K and an allocation is modelled by the satisfying assignment function α (Johnson, Madanian, et al. 2020). However, due to a disaster's inherently uncertain and unpredictable nature, disaster managers must assume that the scenario evolves: i.e., more resources are required at a location, initial resourcing is inadequate and new locations arise and must be resourced. In response, plans must be continually devised to transport resources to the locations where they are needed.

To address this challenge, we propose the *Self-Adaptive Critical Response (self-cr) System* presented in Figure 1. We illustrate how this adaptive system is used on a real-world case study based on the Auckland Central Business District (CBD), represented by the street-level map in Figure 1.

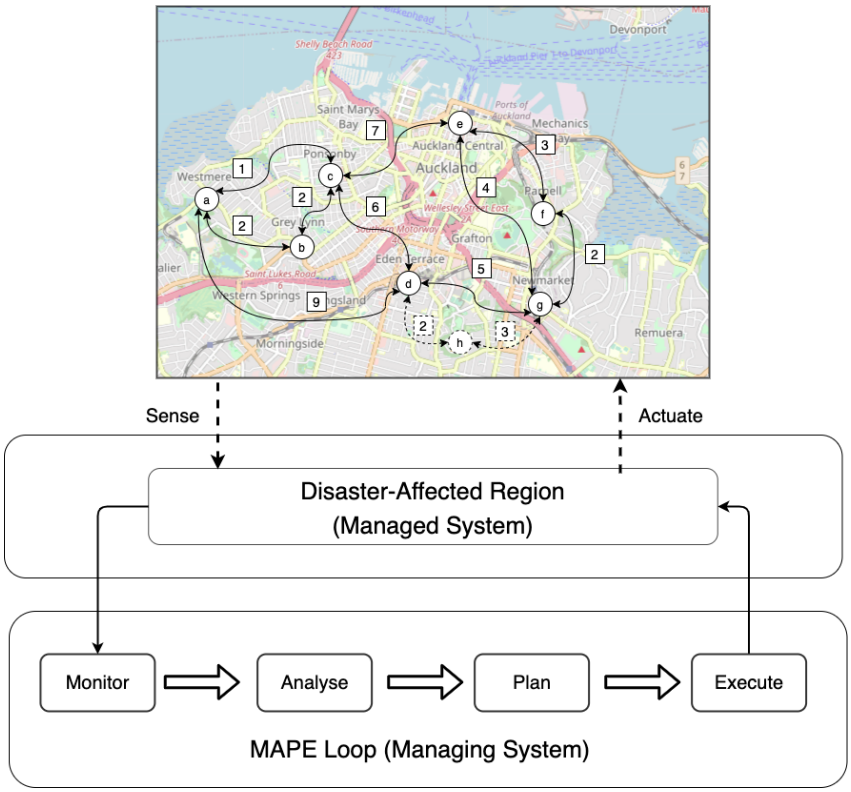

Figure 1. MAPE Loop of the Self-Adaptive Critical Response System (*self-cr*).

The *self-cr* system comprises two main parts:

1. the *managed system* comprising a state representing the disaster-affected region and operations to modify the state, reflecting actions performed in the disaster-affected region by disaster managers, and
2. the *managing system*, comprised of a four phase Monitor-Analyse-Plan-Execute (MAPE) feedback loop (Kephart and Chess 2003), a paradigm commonly used to engineer self-adaptive software-intensive systems.

The *self-cr* system continually senses vital information from the Auckland region:

- changes related to resourcing in Auckland, detected by e.g. reporting and digital audits from health-care professionals at clinics, hospitals or triage stations,
- reports on travel conditions throughout the region, from e.g. online traffic tracking services, social media analysis and eye witness accounts.

The MAPE feedback loop processes sensor input and updates the graph structure in the managed system state: locations are vertices, associated with staff and resources while routes between locations are edges, associated with travel conditions. The updated state acts as input for the **analysis** phase. We use two formal verification techniques to analyse the current state of the managed system. First, constraint solving is used to automatically determine how many resources are needed and at which locations, based on the requirements developed during the disaster preparedness phase. The results of this kind of analysis is useful for deciding on logistical operations for transporting resources from locations with a surplus to those identified as having unsatisfactory amounts.

While it may be obvious to transport resources via the shortest route, infrastructure damage or other events may make this impossible, or at least more prone to failure. The **analysis** phase therefore uses probabilistic model checking (Kwiatkowska et al. 2011; Dehnert et al. 2017) as a means to analyse uncertainties within the region relating to both traffic conditions and emergency worker staffing. Markov models synthesised from the region's

infrastructure network are automatically verified against probabilistic temporal logic formulae encoding the source and destination of transportation through the region. Probabilistic model checking can automatically and *continually* identify optimal routes based on current infrastructure conditions. Additional information on staffing, such as skills, experience and which shifts they work on can be synthesised as Markov models to help choose candidate staff best positioned to successfully perform the transportation and other operations.

Contributions

Our work identifies the need to devise verifiable plans that can react to uncertainty and changes in resource usage and needs during critical response.

Our paper presents work in progress research with the following main contributions:

- Algebraic specification of the *self-cr* system, comprising a MAPE loop as a conceptual framework compatible with a broad range of analysis and verification techniques
- Illustration of the *self-cr* system on a real-world case study based on the Auckland CBD
- A sketch of how the **analysis** phase can be instantiated with two verification techniques for constraint solving and probabilistic model checking

The remainder of the paper is structured as follows. We give an algebraic specification of the managed system comprising a state representing the region and operations mirroring actions performed by disaster managers. We describe the MAPE loop in detail with an example in which plans must be devised to adapt resourcing according to changes in needs and requirements. Focusing on the analysis phase, probabilistic model checking is introduced to verify properties of Markov models representing the region's transport infrastructure. By composing this model with Markov models synthesised from a simple ontology modelling human decision-making, disaster managers are able to plan optimal resource transportation.

RELATED WORK

The MAPE loop in the *self-cr system* is compatible with a range of analysis techniques. There has been a fuzzy logic optimisation approach described in (Sarma et al. 2019) that uses transport costs as the single factor to be optimised. Planning for resource use has started to use Artificial Intelligence techniques such as deep learning to predict evacuation traffic (Aqib et al. 2020).

A formal model of a disaster management system with a focus on modelling wireless sensor and actor networks and their activities using graph theory appears in (Zafar and Afzaal 2017). Modelled as a complex adaptive system using agents, the focus of this system is to detect earthquakes in specific locations and propagating this information throughout the network in a efficient manner. In contrast, the *self-cr system* focuses on assisting with the dynamic and real-time decision-making required in post-disaster recovery efforts, such as resource distribution. Other works focus on formally modelling specific aspects of a disaster management system using carefully selected models, such as emotional agent behaviour (Kefalas et al. 2014) and disaster recovery plans ((Noel) Bryson et al. 2002). Graph theory has also been more widely used in resource distribution and allocation (Johnson, Madanian, et al. 2020; Wagner and Neshat 2010; Kumar and Zaveri 2017), as well as in identifying infrastructure availability during disasters (Hu and Janowicz 2015). In comparison to existing works, the *self-cr system* provides a generalised self-adaptive approach configurable to a wide range of specific use-cases and analysis techniques for post-disaster management.

DISASTER-AFFECTED REGION AS A MANAGED SYSTEM

The self-adaptive approach envisions management of a disaster-affected region as a *managed system*, as shown in Fig. 1. The managed system is used by disaster managers to track first-aid provision and medical resources within the region and respond to evolving demands by reallocating scarce resources. In a mathematical sense, the managed system's state is the triple $\sigma = (G, K, \alpha)$ comprising a graph G, resource demands K and resource allocation α.

Let $State$ be the set of all states of the managed system.

The graph $G = (V, E)$ is a pair such that V is a finite set of vertices representing locations in the region such as hospitals, clinics and triage stations that are critical for resource allocation. Locations are interconnected via edge pairs (u, v) from the finite set E, representing paths (e.g., a sequence of city streets) between two location vertices

$u, v \in V$. Edges may be annotated with additional attributes, such as distance, state of the physical road, etc., via mapping functions.

Resource *demands* are constraints K, formed from logical predicates. These predicates express *how many* of each kind of resource is needed at locations in V. There are a finite number of resource types that we consider, enumerated by the set $R = \{r_1, \ldots, r_n\}$.

To reason about the amount of resources of type r_i are at location $v \in V$, we form its *resource allocation model*; a finite set $Z_v = \{r_{1v}, \ldots, r_{nv}\}$ of variable symbols from the countable set Z.

Resource allocation for $v \in V$ is specified by the function $\alpha_v : Z_v \to \mathbb{Z}$ such that the equation $\alpha_v(zv_i) = n$ means there are $n \in \mathbb{Z}$ of resource type $r_i \in R$ available at location $v \in V$. We collect the set of all resource allocation functions for the region as $\alpha = \alpha_{v \in V} : Z_v \to \mathbb{Z}$. Thus, α is a record of the region's current resource allocation.

This treatment of resource allocation allows for flexibility for expressing general constraints. For example, the high-level requirement *The total amount of allocated resources of type r_i must be between 10 and 50 throughout the region* is formalised by the logical predicate $10 \leq \sum_{v \in V} zv_i \leq 50$ over the resource allocation models of V. Standard ordering relationships on the integers specify resourcing relative to locations. For example, *Location u must have more of resource r_i than v* is specified by the predicate $zu_i > zv_i$.

We suppose the minimal demand for resourcing is specified by the logical predicate $demand_v$ such that $Z_v \subseteq vars(demand_v)$; e.g., the predicate $demand_v$ constraints every variable in Z_v, and possibly contains other variable symbols. Let $demand_v \in K$ for all $v \in V$.

Tests and Operations

The managed system represented a dynamic and evolving disaster-affected region. As such, we will develop, through case studies, algebraic operations of the form $f : State \to State$ such that $\sigma' = f(\sigma)$ is the updated state after a change in the region, modelled by f.

When we collect the operations in Σ together with the states in $State$, we form an algebraic data type $S = (\Sigma, State)$ of the managed system and refer to f as a Σ-operation.

AUCKLAND CBD CASE STUDY

The topmost portion of Figure 1 presents a map of Auckland's Central Business District (CBD) overlaid with a graph of seven city locations labelled a to g that are pertinent to disaster managers. The dashed location h will be added as part of the next scenario. Locations are connected via shortest paths represented by bi-directional edges, labelled with the distance between two locations. The graph G is part of the state tuple $\sigma = (G, K, \alpha)$ that represents the Auckland CBD.

Resource allocation demands in K are typically determined during the disaster preparedness stage. For example, suppose vertex e is the location of a health-clinic, requiring i) built infrastructure ii) staffing medical personnel and iii) medical supplies. In symbols, we set $R = \{built, staff, medical\}$. Based on known population density statistics at e, we might specify the conjunction

$$demand_{\mathtt{e}} \equiv (built_{\mathtt{e}} \geq 20) \wedge (staff_{\mathtt{e}} \geq 5) \wedge (medical_{\mathtt{e}} \geq 50) \tag{1}$$

of logical predicates over model $Z_{\mathtt{e}}$ as resourcing demands. Prior to any disaster, we have satisfactory resource availability throughout the region. This means the resource allocation function α assigns values to resources such that the demand at each location is satisfied. For example, if $\alpha_{\mathtt{e}} = \{built_{\mathtt{e}} := 20, staff_{\mathtt{e}} := 6, medical_{\mathtt{e}} := 54\}$ then (1) is satisfiable. In symbols,

$$\begin{aligned}
[\![demand_{\mathtt{e}}]\!](\alpha_{\mathtt{e}}) &\equiv [\![(built_{\mathtt{e}} \geq 20) \wedge (staff_{\mathtt{e}} \geq 5) \wedge (medical_{\mathtt{e}} \geq 50)]\!](\alpha_{\mathtt{e}}) \\
&\equiv [\![(20 \geq 20) \wedge (6 \geq 5) \wedge (54 \geq 50)]\!] \\
&\equiv true.
\end{aligned}$$

During disaster preparedness, resource demands are carefully considered and a suitable resource allocation supplied. This essentially means computing an α satisfying K. This analysis can be automated using *Satisfiability Modulo Theories (SMT)* software tools e.g., Z3 (Moura and Bjørner 2008) such that $\alpha = smt(K)$ if, and only if, $[\![K]\!](\alpha) \equiv true$. In case, we deem the state of the system to be in *good* health, since the resourcing allocated by α to each location satisfies its demand. If K is not satisfied then resources are lacking at one or more locations, and the state is *bad*.

ADAPTING TO RESOURCE CHANGES

In this scenario, we suppose administrative staff at location e have performed an audit of their supplies and determine: $built_{\mathsf{e}} := 20$, $staff_{\mathsf{e}} := 5$ and $medical_{\mathsf{e}} := 20$. While this location already falls below the necessary 50 units of medical supplies according to (1), staff report larger than expected casualties. Therefore at least 70 units of medical supplies are needed to satisfy demand. Furthermore, two more staff members are required to handle additional casualties. This new information is processed by disaster managers to determine if current resourcing at e is satisfactory. If not, a plan for self-adapting the managed system to bring necessary resources from other locations is devised and carried out.

The following adaptation MAPE feedback phases occur:

Monitor

First, the result of the audit is modelled as an algebraic operation modifying the managed system state, currently recorded as $\sigma \in State$. Let $audit : State \times V \times Z^n \to State$ be the Σ-operation

$$\sigma' = audit(\sigma, \mathsf{e}, built_{\mathsf{e}}, staff_{\mathsf{e}}, medical_{\mathsf{e}}, 20, 5, 20) \tag{2}$$

updating the assignment in state $\sigma' = (G, K, \alpha')$ such that

$$\alpha'_{\mathsf{v}} = \begin{cases} \{built_{\mathsf{v}} := 20, staff_{\mathsf{v}} := 5, medical_{\mathsf{v}} := 20\} & \text{if } \mathsf{v} = \mathsf{e}, \\ \alpha_{\mathsf{v}} & \text{otherwise.} \end{cases} \tag{3}$$

All other elements of the state remain unmodified. The resulting $\sigma' \in State$ reflects the new observations concerning resource amounts, obtained from the audit report.

Next, to accommodate an update to the resource demands at location e, we specify the Σ-operation $update : State \times V \times Demand \to State$ such that the equation

$$update(\sigma, \mathsf{e}, \phi) = K \setminus \{demand_{\mathsf{e}}\} \cup \{\phi\} \tag{4}$$

replaces the minimal demand predicate $demand_{\mathsf{e}}$ with the new predicate ϕ in the set K. If $demand_{\mathsf{e}}$ does not exist in K, then ϕ is added.

In this scenario, we have $\phi \equiv (built_{\mathsf{e}} \geq 20) \wedge (staff_{\mathsf{e}} \geq 5) \wedge (medical_{\mathsf{e}} \geq 70)$.

Analysis

In this phase, we analyse the system state to determine its health. First, we compute the resources necessary for location e based on new demands. Hence, we evaluate $\alpha_s = smt(demand_{\mathsf{e}})$ using an SMT solver to compute the *supply* assignment $\alpha_s = \{built_s := 20, staff_s := 5, medical_s := 70\}$. Lifting standard arithmetic operations to assignments yields the pointwise operation $\alpha_{\mathsf{s}} - \alpha'_{\mathsf{e}}$ e.g.,

$$\begin{aligned} built_s - built'_{\mathsf{e}} &= 20 - 20 = 0 \\ staff_s - staff'_{\mathsf{e}} &= 5 - 5 = 0 \\ medical_s - medical'_{\mathsf{e}} &= 70 - 20 = 50. \end{aligned}$$

From the analysis, we determine that 50 units of medical supplies must be delivered to location e to meet the current demands.

Plan

In this phase, we devise an *adaptation plan* with the goal of adapting the managed system back towards a good state. Formally, an adaptation plan P is a finite sequence f^* of Σ-operations applied in sequence to the current state σ of the system. The empty sequence of Σ-operations means no adaptation is necessary.

These operations are meant to correspond to, and model, actual real-world disaster-relief commands performed by disaster managers. In this scenario, the disaster managers must transport 50 medical aid resources from other locations in region to location e. To model the transport of resources from source to destination, we define the

Σ-operation $transport : State \times V \rightarrow V \times R \times \mathbb{Z} \rightarrow State$ such that amount n of resource r is transported from source location $\mathtt{u}$ to destination $\mathtt{v}$. In symbols,

$$transport(\sigma, \mathtt{u}, \mathtt{v}, r, n) = audit(audit(\sigma, \mathtt{u}, r, \alpha_u(r) - n), \mathtt{v}, r, \alpha_\mathtt{v} + n) \tag{5}$$

if there are enough resources at location $\mathtt{v}$; e.g., $\mid \alpha_\mathtt{v}(r) - n \mid \geq 0$. Otherwise, the state σ remains unmodified.

While the analysis phase revealed the system was in a bad state, it is up to the planning phase to devise a plan to return the system to a good state. To this end, we first query the graph G to identify locations with a surplus of medical supplies. From these locations, we might select two locations: $\mathtt{d}$ and $\mathtt{f}$ with 20 and 30 units respectively who are geographically closer than other candidate locations, based on shortest paths information stored by the graph edges.

Thus, the adaptation plan P to resolve resource demand at location $\mathtt{e}$ comprises the sequence of two $transport$ operations:

1. $transport(\sigma, \mathtt{d}, \mathtt{e}, medical, 20)$ using shortest path $(\mathtt{d}) \rightarrow (\mathtt{g}) \rightarrow (\mathtt{e})$ and
2. $transport(\sigma, \mathtt{f}, \mathtt{e}, medical, 30)$ using shortest path $(\mathtt{f}) \rightarrow (\mathtt{e})$.

Execute

The execute phase is required to actuate the adaptation plan in the region, and is represented by the dashed line from the system to the physical environment in Figure 1. From the perspective of the managed system, simply executing the $transport$ operation updates the current state and resources are at the correct location. However, in the physical environment of the disaster-affected region, resource transport is much more complex. It requires coordination of vehicles, infrastructure availability and personnel to support the operations. In this scenario, we only considered the shortest path between locations, assuming roadways are undamaged and available throughout the region.

ADAPTING RESOURCING FOR A NEW TRIAGING STATION

In this scenario, we suppose it becomes necessary to form a new emergency triaging station. To update the state $\sigma = (G, K, \alpha)$ we add a new location $\mathtt{h}$ t0 G, represented by the dashed node in Figure 1. The new location is connected to existing vertices $\mathtt{d}$ and $\mathtt{f}$, represented by dashed bi-directional edges. In symbols, we update the graph such that: $G' = (V', E')$, where

$$V' = V \cup \{\mathtt{h}\}$$
$$E' = E \cup \{(\mathtt{h}, \mathtt{d}), (\mathtt{d}, \mathtt{h})\} \cup \{(\mathtt{h}, \mathtt{g}), (\mathtt{g}, \mathtt{h})\}$$

where the added edges are labelled with corresponding shortest-path distances. Let $\sigma' = (G', K, \alpha)$ be the updated state of the managed system.

The first-responders report they require built infrastructure for at least 10 beds, two more staff and 40 units of medical aid. We formalise these requirements as the following logical predicate

$$demand_\mathtt{h} \equiv (built_\mathtt{h} \geq 10) \wedge (staff_\mathtt{h} \geq 5) \wedge (medical_\mathtt{h} \geq 40) \tag{6}$$

over resources in R. Updating the (currently nonexistent) resource demands for $\mathtt{h}$ in state σ' , we set $\sigma'' = update(\sigma', \mathtt{h}, demand_\mathtt{h})$.

Currently, there is only makeshift built infrastructure and three first-responders staff with a limited supply of 5 units of medical aid at $\mathtt{h}$. Formally, $\alpha_\mathtt{h} = \{built_\mathtt{h} := 0, staff_\mathtt{h} := 2, medical_\mathtt{h} := 5\}$.

The remaining MAPE loop phases are similar to the previous scenario. We utilise SMT solving $\alpha_s = smt(demand_\mathtt{h})$ to obtain resourcing requirements at location $\mathtt{h}$. By calculating outstanding resource requirements $\alpha_s - \alpha_\mathtt{h}$ determines the need for 10 built infrastructure, 3 staff and 35 units of medical aid to be transported.

ANALYSING PROBABILISTIC MODELS OF INFRASTRUCTURE AVAILABILITY

Transportation route decisions are often made with a high level of uncertainty arising from infrastructure conditions in the affected region. The previous scenarios did not consider this when devising adaptation plans. While $transport(\sigma, u, v, r, n)$ might be performed *optimally* in terms of geographically shortest path between u and v, the route may have a low likelihood of success due to dangerous circumstances or from road damage.

To provide disaster managers the means to evaluate alternative routes based on likelihood of success, we synthesis the Markov model m that represents all paths through the graph G of the disaster-affected region. The analysis phase performs *probabilistic model checking* of a temporal logic formula ϕ on m to compute an optimal path for reaching v from u.

Model Synthesis

We synthesis a Markov Decision Process (MDP) model m from the Auckland CBD graph G to analyse the maximum probability of the $transport(\sigma, \mathtt{d}, \mathtt{e}, medical, 20)$ operation succeeding.

The model is expressed in Prism's high-level modelling language (Kwiatkowska et al. 2011) and a fragment of the model is listed in Figure 2. In Prism, a model is composed of interacting modules. The `transport` module contains state variable `s` that varies between values `0` and `8`. In particular, `0` through to `6` encode locations `a` to `g=6` in G and are specified by constants in the model fragment in Figure 2. The initial state is `d=3`, corresponding to the transport's source location and the other states encode locations from the graph. The two states `succ` and `fail` model the success or failure of the $transport$ operation respectively.

The models' behaviour is specified by transitions labeled with probability distributions that decide the *next* state. Probabilities encode the likelihood of successfully traversing between connected locations. Probabilities are computed via the monitor phase of the MAPE loop and must be continually updated using observations from variety of sensors, on-site personnel reporting, or from social media analysis such as in (Paterson et al. 2019).

To determine the next state, first an available transition from the current state is non-deterministically selected. Then the next state is randomly chosen based on corresponding distribution. For example, the three transitions

```
[da] (s = d) -> pda:(s'=a) + (1-pda):(s'=fail);
[dc] (s = d) -> pdc:(s'=c) + (1-pdc):(s'=fail);
[dg] (s = d) -> pdg:(s'=g) + (1-pdg):(s'=fail);
```

model the choice of available directions from location `d` to `a`, `c` and `g` in the graph. Each have a guard predicate `(s=d)` over the state variable s. If we decide to select the last transition labelled `[dg]` then the updates `s'=g` and `s'=fail` specify new state based on the distribution with probability `pdg` the transport will successfully traverse to location `g`. Otherwise the model will traverse to the $fail$ state with probability `1-pdg`.

When the $transport(\sigma, \mathtt{d}, \mathtt{e}, 20)$ is performed, model m is synthesised and the following adaptation MAPE feedback phases occur.

Model Analysis Using Probabilistic Model Checking

The synthesised Markov model m acts as input to a probabilistic model checker which computes the probability of the model satisfying a probabilistic temporal logic formula ϕ. Let $pmc : M \times P \rightarrow [0, 1]$ be a probabilistic model checker such that $pmc(m, \phi) = p$ if, and only if, the probability of $\phi \models m$ is p. Probabilistic model checking has mature, well established software tools (Kwiatkowska et al. 2011; Dehnert et al. 2017) to automatically analyse a wide range of Markov models.

In this scenario, we are interested in selecting a route with the maximum probability of reaching the *succ* state, from initial state `d`. To analyse the synthesised Markov model, we specify the probabilistic temporal logic formula $\phi \equiv$ `Pmax=?[F(s=succ)]`. which essentially queries the model to answer *what is the maximum probability that we Finally reach a state from the initial state* `s=d` *such that* `s = succ` *is true?* In symbols, we equate

$$\begin{aligned} 0.8633 &= pmc(m, \texttt{Pmax[F (s=succ)]}) \\ &= \text{maximum probability of } transport(\sigma, \mathtt{d}, \mathtt{e}, r, 20) \text{ succeeding in state } \sigma. \end{aligned}$$

Continual model analysis incorporates new observations such as previously unknown road damage to update the path `d` to `e` while resources are still en-route. For example, suppose analysis of pmc is performed after transition $(\mathtt{d}) \rightarrow (\mathtt{g})$ is completed. The transition $(\mathtt{g}) \rightarrow (\mathtt{e})$ has availability reduced to 0.1, as extensive damaged has occurred on the route. Reanalysis of the model m with updated probability transitions yields the alternative path $(g) \rightarrow (f) \rightarrow (e)$ which, although it is not the shortest path, it does have an increased probability of success.

SELF-ADAPTIVE SYSTEMS WITH HUMANS IN THE LOOP

So far, we have focused on the ways model synthesis and analysis can be automated to devise plans. In this section, we focus on the role humans have in the different parts of the *self-cr system* adaptation MAPE loop. For example, humans act as sophisticated sensors that supply information by e.g., interacting with eHealth application to carry out digital auditing or interacting with social media platforms (Paterson et al. 2019). Ultimately, it is the human disaster managers decision-makers that approve planning for emergency workers who become actuation agents, conducting actions that are difficult to automate (e.g., physically process and transport resources between locations).

```
//transport operation model from location d to e
mdp
const int a=0;,...,const int g=6;const int succ = 7;const int fail = 8;
const double pcb = 0.9;
const double pce = 0.8;
const double pdc = 0.78;
const double pdg = 0.89;
const double pge = 0.97;
...
module transport
    s : [0..8] init d;
     [ab] (s = a) -> pab:(s'=b) + (1-pab):(s'=fail);
     [ac] (s = a) -> pac:(s'=c) + (1-pac):(s'=fail);
     [ad] (s = a) -> pad:(s'=d) + (1-pad):(s'=fail);
     ...
     [da] (s = d) -> pda:(s'=a) + (1-pda):(s'=fail);
     [dc] (s = d) -> pdc:(s'=c) + (1-pdc):(s'=fail);
     [dg] (s = d) -> pdg:(s'=g) + (1-pdg):(s'=fail);
     ...
     [gd] (s = g) -> pgd:(s'=d) + (1-pgd):(s'=fail);
     [ge] (s = g) -> pge:(s'=e) + (1-pge):(s'=fail);
     [gf] (s = g) -> pgf:(s'=f) + (1-pgf):(s'=fail);
     [succ] (s = e) -> 1.0:(s'=succ);
     [succ] (s=succ) -> 1.0:(s'=succ);
     [fail] (s=fail) -> 1.0:(s'=fail);
endmodule
```

Figure 2. Synthesised MDP fragment modelling Auckland CBD infrastructure availability.

Humans play a critical role in the success of any action forming a crisis response. However, the MAPE loop planning phase devises plans in terms of geospatial data and infrastructure availability. We describe the *actuation* role of humans within the *self-cr* system. In this sense, there is a growing need to model the impact of human participants on mission goals.

To model human involvement in the *self-cr system* MAPE loop, we use the *Opportunity-Willingness-Capability (OWC) ontology* (Eskins and Sanders 2011). The OWC ontology is a readily applicable and generic framework, used in cyber security applications and for analysing human in the loop of self-adaptive systems (Cámara, Garlan, et al. 2017; Cámara, Moreno, et al. 2015; Li et al. 2020). OWC models effects of human decisions on a *socio-cyber physical system (SCPS)*, composed of interacting humans, physical and digital components. The disaster-affected region is a SCPS composed of

- human participants: disaster managers, casualties, first-responders and emergency workers
- physical locations, roads and resources, and
- eHealth digital systems and smart technologies prevalent in modern cities.

Using our current elementary models, plans devised by the MAPE loop are simply a sequence of operations performed by one or more human participants using the system components. However, in principle, plans often comprise complexities featuring multiple actions taking place in parallel at different locations

Personnel Profiles of Human Participants

To analyse effects of human decisions on the state of the disaster-affected region, we need to include updated information about the system's human participants, such as their skills and competencies, experience and location. We consider in particular, emergency workers who are dispatched to the disaster-affected region as first responders.

Mathematically, we extend states in $State$ to tuples of the form $\sigma = (G, K, \alpha, P)$ where $P = \bigcup_{v \in V} P_v$. The set P_v contains $p \in P_v$ representing the *personnel profile* of a human participant at location v, comprising subsets of metadata tags in the sets SR, DR, EX describing

- $SR \subseteq \{\texttt{VEHICLE-ACCESS}, \texttt{MEDIC}, \texttt{ENGINEER}\}$, identifying skills and responsibilities that correspond to the human participant's disaster relief role
- $DR \subseteq \{\texttt{READY}, \texttt{ACTIVE}, \texttt{RELIEVE}\}$ identifying the human participant's duty roster status: ready to be deployed, active on location, and waiting to be relieved and
- $EX \subseteq \{\texttt{HIGH}, \texttt{MED}, \texttt{LOW}\}$, identifying levels of experience and competencies.

Opportunity-Willingness-Capability

Given the operation $transport(\texttt{d}, \texttt{e}, medical, 20)$, we are interested if it is successful or not. If successful, we expect the resulting system state σ' to represent the achievement of 20 medical supplies transported from location $\texttt{d}$ to location $\texttt{e}$. Otherwise, the operation has not been performed properly.

To perform this Σ-operation successfully, the current state σ of the disaster-affected region must satisfy an intersection of predicates on the states in $State$.

This section defines three functions that categorise states in $State$ by factors relating to a given human participant p's opportunity, willingness and capability to perform the operation on the state $\sigma \in State$ and follows (Eskins and Sanders 2011) closely.

Opportunity

Opportunity formalises prerequisites to be satisfied for $transport$ to be performed. They are essentially formalised as a predicate on σ and codified in (5). To meet the conditions to attempt the operation, the human participant p must i) be physically present at source location $\texttt{d}$ ii) have access to a suitable vehicle, iii) have access to enough medical aid resources to transport. Let $transport^o : State \times P \rightarrow \mathbb{B}$ be defined by the following conjunction of three predicates on the input state σ such that

$$transport^o(\sigma, p) = (p \in P_{\texttt{d}}) \wedge (\texttt{VEHICLE-ACCESS} \in RE) \wedge (\mid \alpha_{\texttt{d}}(medical) - 20 \mid \geq 0). \tag{7}$$

Willingness

Willingness capturing the desire of the human participant p to perform the $transport$ operation.

For example, $transport^w : State \times P \rightarrow [0, 1]$ assigns values to tags in the duty roster DR of participant p such that

$$transport^w(\sigma, p) = \begin{cases} 0.99 & \texttt{READY} \in DR, \\ 0.85 & \texttt{ACTIVE} \in DR, \\ 0.1 & \texttt{RELIEVE} \in DR. \end{cases} \tag{8}$$

Essentially, we have related *willingness* to fatigue, as indicated to p's current duty roster status. Determining such probabilities can be achieved using a range of run-time monitoring techniques that include, among others, brain-computer interaction. For more details about how such techniques have been employed in combination with OWC in the context of self-adaptive systems, please refer to (Lloyd et al. 2017).

Capability

Capability formalises the idea that given the first two predicates of opportunity and willingness that participant p can succeed in completing an operation only some of the time. Therefore capability is expressed in terms of elements of the current state that specify participants p's level of experience, based on previous success of performing similar operations. We define the function $transport^c : State \times P \rightarrow [0, 1]$ such that

$$transport^c(\sigma, p) = \begin{cases} 0.90 & \texttt{HIGH} \in EX, \\ 0.50 & \texttt{MED} \in EX, \\ 0.35 & \texttt{LOW} \in EX. \end{cases} \tag{9}$$

This assignment of values to tags in EX formalise the intuitive idea that higher experience in performing an operation often results in success, whereas inexperience may lead to an increased probability of failure.

To summarise, the tuple $transport^{owc} = (transport^o, transport^w, transport^c)$ provides a formal means to categorise states elements and identify key variables affecting operation performance. These properties are particularly useful for simulating a range of potential first-responder deployment configurations (e.g., which participant should go where) and can inform necessary training and professional development programs for staff in future scenarios.

OWC in localised Staff and Resource Allocation

The OWC ontology can be used to plan localised disasters, such as a fire service dealing with a large building fire. Such an event requires rapid fire response and subsequent adaptation as the situation evolves over the course of the response. It also requires the allocation of available personnel to specific fronts the fire is being fought on. In a recent fire at the SkyCity Convention Centre in Auckland[1], it was found that the dynamic nature of fighting a large fire over several days often makes it difficult to ensure the right allocation of personnel to fronts. For instance, some personnel may need to stay at a front for longer than desired due to non-availability of adequate replacements. In some other cases, personnel waiting in a “holding area" may not be clearly identified and hence cannot be assigned to fronts in a timely manner.

```
//constants extracted from current state and personnel profile of participant [
const int  dMedical = 33;//medical unit count at location d
const bool p_dLocation = true;//is p at location d?
const bool p_vehicle = true;//does p have vehicle?
const bool p_active = true;//is p on active duty
const double pActive = 0.85;//probability associated with Active tag
const double pHigh = 0.90; //probability associated with high experience

module transport_o
  opp : [0..1] init 0;
    [] (opp = 0)&p_dLocation&p_vehicle&p_active&dMedical>=20 -> (opp'=1);
    [] (opp = 1) -> true;
endmodule

module transport_w
  will : [0..2] init 0;
    [] (will = 0) -> pActive:(will'=1) + (1-pActive):(will'=2);
    [] (will = 1) -> true;
    [] (will = 2) -> true;
endmodule

module transport_c
  cap : [0..2] init 0;
    [] (cap = 0)&(opp=1)&(will=1) -> pHigh:(cap'=1) + (1-pHigh):(cap'=2);
    [] (cap = 1) -> true;
    [] (cap = 2) -> true;
endmodule
```

Figure 3. Synthesised Markov Model fragment for OWC analysis of personnel profile.

OWC Markov Model Synthesis and Analysis

Analysing models of humans-in-the-loop during the analysis phase of the MAPE loop can positively impact on the overall success of the adaptive plan chosen by disaster managers. To automate analysis, Markov models are synthesised from the current state σ and the human participant p's personnel profile: $p :=$ ({VEHICLE-ACCESS}, {ACTIVE}, {HIGH}) and current position at location d.

Each function in the $transport^{owc}$ tuple is modelled by a Markov model, presented in Figure 3 in Prism's high-level language. Each module is small with two or three states and all with initial state 0. The modules reference Boolean, integer and double constant parameters that are instantiated from the current system state σ and profile p. In particular,

- `transport_o` transitions to state 1 from the initial state whenever predicates formalised in (7) are satisfied,
- `transport_w` transitions to state 1 according to the corresponding `ACTIVE` probability distribution value instantiated as `pActive = 0.85`, specified by (8). Otherwise the transition to state 2 is triggered to represent that the operation fails to be performed, and

[1] www.stuff.co.nz/national/300125069/skycity-convention-centre-fire-fire-and-emergency-could-not-have-prevented-damage

- `transport_c` transitions to state 1 whenever p has the opportunity and willingness to perform the operation. The operation succeeds according to the probability distribution value instantiated as `pHigh = 0.90`, specified by (9). Otherwise the transition to state 2 is triggered to represent that the operation fails to be performed.

Now, by composing these three modules with the `transport` Prism model listed in Figure 2, we can select a route with the maximum probability of reaching the *succ* state and that the human participant p has the opportunity, willingness and capability to transport resources from `d` to `e`. We formalise this property by the probabilistic temporal logic formula

$$\begin{aligned} 0.66042 &= pmc(m, \texttt{Pmax=?[F((s=succ)\&(opp=1)\&(cap=1)\&(will=1))]}) \\ &= \text{maximum probability } p \text{ performs } transport(\sigma, \texttt{d}, \texttt{e}, r, 20) \text{ successfully in state } \sigma. \end{aligned}$$

CONCLUDING REMARKS

In this paper, we formalised a conceptual framework in which the disaster-affected region is managed by a self-adaptive system feedback MAPE loop. A key technical challenge for analysing disaster models is their changeability. Disasters evolve and their models must also change to reflect what is happening in the real world, as micro-events may make previous analysis results outdated and model reverification necessary. Our approach is not dissimilar to managing digital computing systems and therefore draws on its extensive body of theoretical results from runtime verification (Weyns 2020). The aim is to minimise the amount of costly reverification and reuse previous results whenever possible, often by isolating reverification to parts of the model affected by change (Johnson, Calinescu, et al. 2013), thus reducing the amount of analysis needed.

Another important aspect to consider is uncertainties related to impact of changes in a system (Perez-Palacin and Mirandola 2014). Sensitivity analysis (Goseva-Popstojanova and Kamavaram 2004) is particularly important for disaster management applications as small disruptions could have serious consequences for planning outcomes. For example, unavailability of a single critical route through a city can seriously disrupt resource allocation planning.

Our work in this paper has focused on the **analysis** phase to determine what resources are required and where. We extended logical analysis via probabilistic model checking to determine optimal routes through the region, based on current road conditions and profiles of human participants. Indeed, while the current paper features a simple ontology mapping personnel profiles to task performance, our overall aim is to validate compatibility of first-responder groups' plans during the response to a disaster to support parallel planning operations. In addition, we can also consider analysis techniques such as fuzzy set representation to study several alternatives in resource allocation and transportation planning (Zheng and Ling 2013; Zhang et al. 2019).

Our research programme aims to develop a generic and re-usable self-adaptive system framework, where phases are extendable using standardised interfaces to support

- multiple sensory inputs from digital and human sources, such as wearable sensors for first-responders
- multi-stage analysis and automated planning across several formal verification techniques, extending our current results
- integration of deep-learning and AI techniques to better understand the efficacy of plans and support decentralised planning

The theoretical framework will form the basis of software tools amenable for developing a number of simulation-based applications to test and validate planning, such as analysing critical sections of the graph to help understand disaster plan vulnerabilities. We aim to develop formalised disaster plans that are testable and repeatable across a range of *what-if* disaster scenarios. This may allow planners to identify critical resource placements and information needs by running simulations that change these parameters.

REFERENCES

(Noel) Bryson, K.-M., Millar, H., Joseph, A., and Mobolurin, A. (2002). "Using formal MS/OR modeling to support disaster recovery planning". In: *European Journal of Operational Research* 141.3, pp. 679–688.

Aqib, M., Mehmood, R., Alzahrani, A., and Katib, I. (2020). "A Smart Disaster Management System for Future Cities Using Deep Learning, GPUs, and In-Memory Computing". In: *Smart Infrastructure and Applications: Foundations for Smarter Cities and Societies*. Ed. by R. Mehmood, S. See, I. Katib, and I. Chlamtac. Cham: Springer International Publishing, pp. 159–184.

Cámara, J., Moreno, G. A., and Garlan, D. (2015). "Reasoning About Human Participation in Self-adaptive Systems". In: *Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. SEAMS '15. Florence, Italy: IEEE Press, pp. 146–156.

Cámara, J., Garlan, D., Moreno, G., and Schmerl, B. (2017). "Chapter 7 - Evaluating Trade-Offs of Human Involvement in Self-Adaptive Systems". In: *Managing Trade-Offs in Adaptable Software Architectures*. Ed. by I. Mistrik, N. Ali, R. Kazman, J. Grundy, and B. Schmerl. Boston: Morgan Kaufmann, pp. 155–180.

Dehnert, C., Junges, S., Katoen, J.-P., and Volk, M. (2017). "A Storm is Coming: A Modern Probabilistic Model Checker". In: *Computer Aided Verification*. Ed. by R. Majumdar and V. Kunčak. Cham: Springer International Publishing, pp. 592–600.

Eskins, D. and Sanders, W. H. (Sept. 2011). "The Multiple-Asymmetric-Utility System Model: A Framework for Modeling Cyber-Human Systems". In: *2011 Eighth International Conference on Quantitative Evaluation of SysTems*, pp. 233–242.

Goseva-Popstojanova, K. and Kamavaram, S. (2004). "Software reliability estimation under certainty: generalization of the method of moments". In: *Eighth IEEE International Symposium on High Assurance Systems Engineering, 2004. Proceedings.* Pp. 209–218.

Hu, Y. and Janowicz, K. (2015). "Prioritizing road network connectivity information for disaster response". In: *Proceedings of the 1st ACM SIGSPATIAL International Workshop on the Use of GIS in Emergency Management*, pp. 1–4.

Johnson, K., Madanian, S., and Sinha, R. (2020). "Graph-Theoretic Models of Resource Distribution for Cyber-Physical Systems of Disaster-Affected Regions". In: *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 521–528.

Johnson, K., Calinescu, R., and Kikuchi, S. (2013). "An Incremental Verification Framework for Component-based Software Systems". In: *Proceedings of the 16th International ACM Sigsoft Symposium on Component-based Software Engineering*. CBSE '13. Vancouver, British Columbia, Canada: ACM, pp. 33–42.

Kefalas, P., Sakellariou, I., Basakos, D., and Stamatopoulou, I. (2014). "A Formal Approach to Model Emotional Agents Behaviour in Disaster Management Situations". In: *Artificial Intelligence: Methods and Applications*. Ed. by A. Likas, K. Blekas, and D. Kalles. Cham: Springer International Publishing, pp. 237–250.

Kephart, J. O. and Chess, D. M. (2003). "The vision of autonomic computing". In: *Computer* 36.1, pp. 41–50.

Kumar, J. S. and Zaveri, M. A. (2017). "Graph-based resource allocation for disaster management in IoT environment". In: *Proceedings of the Second International Conference on Advanced Wireless Information, Data, and Communication Technologies*, pp. 1–6.

Kwiatkowska, M., Norman, G., and Parker, D. (2011). "PRISM 4.0: Verification of Probabilistic Real-time Systems". In: *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*. Ed. by G. Gopalakrishnan and S. Qadeer. Vol. 6806. LNCS. Springer, pp. 585–591.

Li, N., Cámara, J., Garlan, D., and Schmerl, B. R. (2020). "Reasoning about When to Provide Explanation for Human-involved Self-Adaptive Systems". In: *IEEE International Conference on Autonomic Computing and Self-Organizing Systems, ACSOS 2020, Washington, DC, USA, August 17-21, 2020*. IEEE, pp. 195–204.

Lloyd, E., Huang, S., and Tognoli, E. (2017). "Improving Human-in-the-Loop Adaptive Systems Using Brain-Computer Interaction". In: *12th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS@ICSE 2017, Buenos Aires, Argentina, May 22-23, 2017*. IEEE Computer Society, pp. 163–174.

Madanian, S., Norris, T., and Parry, D. (Oct. 2020). "Disaster eHealth: Scoping Review". In: *J Med Internet Res* 22.10, e18310.

Moura, L. de and Bjørner, N. (2008). "Z3: An Efficient SMT Solver". In: *Tools and Algorithms for the Construction and Analysis of Systems*. Ed. by C. R. Ramakrishnan and J. Rehof. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 337–340.

Paterson, C., Calinescu, R., Manandhar, S., and Wang, D. (2019). "Using Unstructured Data to Improve the Continuous Planning of Critical Processes Involving Humans". In: *2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pp. 25–31.

Perez-Palacin, D. and Mirandola, R. (2014). "Uncertainties in the Modeling of Self-Adaptive Systems: A Taxonomy and an Example of Availability Evaluation". In: *Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering*. ICPE '14. Dublin, Ireland: Association for Computing Machinery, pp. 3–14.

Sarma, D., Das, A., and Bera, U. K. (2019). "An optimal redistribution plan considering aftermath disruption in disaster management". In: *Soft Computing* 24.1, pp. 65–82.

Wagner, S. M. and Neshat, N. (2010). "Assessing the vulnerability of supply chains using graph theory". In: *International Journal of Production Economics* 126.1, pp. 121–129.

Weyns, D. (2020). *An Introduction to Self-adaptive Systems: A Contemporary Software Engineering Perspective*. Wiley-IEEE Computer Society Press.

Zafar, N. A. and Afzaal, H. (2017). "Formal model of earthquake disaster mitigation and management system". In: *Complex Adaptive Systems Modeling* 5.1, p. 10.

Zhang, J., Liu, H., Yu, G., Ruan, J., and Chan, F. T. (2019). "A three-stage and multi-objective stochastic programming model to improve the sustainable rescue ability by considering secondary disasters in emergency logistics". In: *Computers & Industrial Engineering* 135, pp. 1145–1154.

Zheng, Y.-J. and Ling, H.-F. (2013). "Emergency transportation planning in disaster relief supply chain management: a cooperative fuzzy optimization approach". In: *Soft Computing* 17.7, pp. 1301–1314.