P-CARD: Policy-based Contextual Awareness Realization for Disasters

Amir Talaei-Khoei1, Steven Bleistein2, Pradeep Ray3, Nandan Parameswaran4
1,3,4Asia-Pacific ubiquitous Healthcare Research Centre, University of New South Wales, Sydney, Australia
2Bleistein Consulting Pty Ltd, Sydney, Australia
1amirtk@student.unsw.edu.au, 2steve@Bleistein.com.au, 3p.ray@unsw.edu.au, 4paramesh@cse.unsw.edu.au

Abstract

In an emergency response to a disaster, cooperation, upon which first responders rely to maintain contextual awareness, may not be adequate. This paper addresses how to maintain contextual awareness and extend cooperation among first responders in a disaster. Much research proposes that software agents can be used to aid in maintaining contextual awareness when direct cooperation is not available. However, the process of contextual awareness realization, a critical aspect to use software agents, has largely been ignored. This paper thus proposes a four-step process for contextual awareness realization driven by policies. The process is demonstrated via application to a scenario based on the hazardous material rail incident in Casa Grande, AZ. The paper concludes that refinement of management issues to policies, and then transforming policy to rules to define software agent behaviors is a potentially effective method for contextual awareness realization in software agents. This method enhances cooperation among involved agencies.

1. Introduction

Disasters such as hurricanes and floods like Katrina, pandemics like SARS outbreaks, terrorist attacks like the one on the World Trade Center Towers, and hazardous material incident like what happened in Casa Grande, AZ, in 1983 [1] are a part of contemporary life. Effective emergency response is thus important to governments, public agencies and non-government organization worldwide. Emergency response, a critical phase of disaster management [2], involves various first responder agencies who must cooperate. These agencies, called cooperative roles [3], are typically widely distributed across geographical areas. However, disasters by their nature engender situations in which cooperation may not be adequate.

One of the major questions facing the management of emergency response in disasters is how to utilize emerging Information Technology (IT) tools and techniques to facilitate cooperation between cooperative roles [4,5]. Currently, there are different integrated standards for cooperation in distributed systems such as Internet SNMP [6], OSI [7], RM-ODP [8], TMN [9], TINA [10], and IETF [11] which are based on an assumption that the cooperation between roles is sufficient. Therefore, when cooperation needs to be enhanced, these standards will not be able to effectively manage the emergency response system. The major hindrance to solve this problem is the lack of a mechanism to support cooperation. We borrow the concept of contextual awareness from Computer Supported Cooperative Work (CSCW) [12] for this purpose.

Contextual awareness refers to the behavior of a cooperative member based on her/his knowledge of context, i.e., situational information such as an understanding of what is happening and why [3]. The contexts of other involved cooperative members are critical to contextual awareness. Therefore, when cooperation is inadequate, roles still need to maintain awareness of the contexts of others despite suffering insufficient cooperation, if they are to continue to respond effectively with appropriate behaviors and actions. The process by which a role adopts the right behavior, known as Contextual Awareness Realization (CAR) [13], is a critical aspect of effective contextual awareness in emergency response. In fact, CAR involves the process that each role transforms her/his contextual awareness into some functions as her/his behaviour [3,13,14].

Much research in applications of information technology in disaster management proposes that software agents can be used to aid roles maintaining contextual awareness during an emergency response [15-17]. Indeed, some research suggests that software agents could have improved the emergency response to disasters, potentially stemming the loss of life [17]. While there are bodies of work on contextual awareness [3,4] and on methods to be used in an
emergency response [15,16,18], the process of CAR in emergency response has been largely ignored.

This paper thus addresses CAR in emergency response by proposing a policy driven process to realize contextual awareness among cooperative roles. In this research, roles are represented in an application by intelligent software agents [15,16]. The method enables realization of the right contextual awareness among these software agents as driven by policies. The software agents inform an adaptive behavior to roles for effective emergency response.

The rest of the paper is organized in the following way: Section 2 illustrates the concept of realization by a scenario based on the hazardous material rail incident in Casa Grande, AZ, 1983. Section 3 discusses context-aware software agents and shows how contextual awareness helps cooperative roles in disasters. Section 4 presents a policy driven method to realize awareness in intelligent software agents. Section 5 discusses the implications of the proposed method, the potential of the method to improve intelligent communication systems, how the method fills the challenges discussed in Section 2 and the limitations of the method. Section 6 presents the related work to this study. Section 7 concludes the paper.

2. Illustrating the Problem: Casa Grande Rail Hazardous Material Incident

In 1983, a train arrived in Casa Grande with burning hazardous materials in its cargo. The train engineer could not realize any fire till it got serious. Shortly afterward, the train stopped near the city’s downtown with smoke coming from one of its cars.

Although, the reaction of firefighters was fast, no information was immediately available about the threat of burning hazardous material in the cargo. The train company was not informed until 20 minutes, and even then the cargo information was incomplete.

Firefighters began to put out the fire following the normal procedure for the local firefighting. After 40 minutes, several firefighters became incapacitated. Then, the emergency team realized that the danger and the fire department advised the police to evacuate the area. Since only limited information on toxicity of the smoke was available, emergency team did not know how far the hazardous threat the human health and the discussion of how to evacuate the area took place 30 minutes. As a result, local resident were not asked to evacuate until the situation got dangerous and the area was inundated with smoke. However, at the time, there were six hazardous material handling teams in Arizona, none of them were contacted. The state’s hazardous material coordinator also did not arrive on the scene on time.

Figure 1 Response Operator and insufficient cooperation – Casa Grande Rail Incident.

This example illustrates two challenges in emergency response of disasters: (1) Distribution of agencies across geographical areas, and (2) Sporadic contacts between roles. Therefore, the Casa Grande train incident reveals following concerns [19]:

- Cooperation: knowledge based environment to facilitate cooperation with relevant information.
- Communication: mobile communication ensures information exchange across geographical areas.

The objective of this research is thus to enhance cooperative disaster management systems by exchanging relevant information over mobile networks.

3. Context-aware Software Agents

The dynamic nature of disasters requires adaptive and robust technology to manage the situation. Uliru and Doursat suggest developing emergency systems with intelligent software agents for effective behaviors of roles during an emergency response [15]. Software agents are categorized into passive, reactive, active and adaptive [17,20]: Passive Software Agents do not
participate in a system unless specifically contacted. *Re-active Software Agents* may simply be able to receive a message from another agent and transmit a standard response. *Active Software Agents* have properties that allow them to interact with other agents. *Adaptive Software Agents* are capable to modifying some of their parameters or variable states or, in some instances, their rule set.

According to above definitions for different types of software agents, since the first three categories need coordinated cooperation, we should design adaptive software agents to change rule sets for an effective behaviour to enhance cooperation among roles. This change can be done considering contextual awareness.

A context-aware software agent uses context to consider situational information in order to inform an effective emergency response to cooperative roles. A context-aware software agent is thus a way to represent cooperative roles using adaptive software agents [21].

The scenario illustrated in Figure 2 shows that if the response operator was context-aware, she/he could have contacted the train company and asked for cargo information. Therefore, he would realize that there are hazardous materials, burning in the cargo, which would lead her/him to inform the jurisdictional hazardous material coordinator to send an expert team to the site.

We agree with [16, 19, 22] regarding the usefulness of adaptive software agents. However, as mentioned previously, the realization process for contextual awareness in context-aware software agents has not been addressed. As such, we propose that policies can serve as a driving mechanism to realize the contextual awareness behavior of software agents, which we discus in the next section.

4. Policy Driven Realization of Awareness

In this section, we conceptualize realization of contextual awareness and show in an emergency system, how management issues can be transformed to policies and how policies can serve as a mechanism for realization of awareness in adaptive software agents, and hence in cooperative members.

This section is organized as follows: Section 4.1 conceptualizes cooperative emergency response systems. Section 4.2 defines and presents management issues and explains their conceptualization; Section 4.3 defines and presents policies and explains policy refinement; Section 4.4 builds on concepts of management issues, policy and policy refinement and presents a four step process of realization of contextual awareness; Section 4.5 explains how rules can be used for representation of contextual awareness behaviors in context-aware Software Agents.

4.1. Cooperative Emergency Response Systems

In order to conceptualize cooperative emergency response systems, we leverage systemic modeling method [10] using first order logic in the following way: We attempt to derive understanding parts from behaviours and values of its whole rather than derive behaviours and values of whole from of parts [23]. Therefore, we first look at systems, their behaviours, and values, then go for sub-systems. We define each
entity, or cooperative system as a vector $E$ of (1) system values, and (2) system behavior (see Figure 3):

$$
\sum_{\text{values}}: \text{set of all possible values} \\
\sum_{\text{behavior}}: \text{set of all possible behaviors} \\
\forall E_{\text{values}} \in \sum_{\text{values}} \land \forall E_{\text{behavior}} \in \sum_{\text{behavior}} \mid (1)
$$

$$
E = (E_{\text{values}}, E_{\text{behavior}})
$$

Values show the state of the entity and can be categorized in two types: (a) management values (mng) that are constant during the performance of the system, (b) contextual values (cnt) that shows situational information.

$$
E_{\text{values}}^{\text{mng}} \subseteq E_{\text{values}} \\
E_{\text{values}}^{\text{cnt}} \subseteq E_{\text{values}} \\
E_{\text{values}} = E_{\text{values}}^{\text{mng}} \cup E_{\text{values}}^{\text{cnt}} (2)
$$

Management values regarding contextual values start or stop a behavior, which can be presented by true or false.

$$
b \in E_{\text{behavior}} \\
b(E_{\text{values}}^{\text{mng}}, E_{\text{values}}^{\text{cnt}}) \rightarrow \{\text{true, false}\} (3)
$$

![Figure 3 Conceptualization of Cooperative Emergency Systems.](image)

In order to conceptualize sub-systems in a cooperative emergency response system, we define abstract and refined entities, values and behaviours. The relationship between abstract values and refined values is defined by refinement theory [24] in systematic modeling. Figure 4 shows that a refined value correctly refines an abstract value if, when the refined value makes a transition from $A_{\text{refined}}$ to $B_{\text{refined}}$ with behavior $b_{\text{refined}}$, the abstract value is also making a transition from $A_{\text{abstract}}$ to $B_{\text{abstract}}$ with behavior $b_{\text{abstract}}$, and they should also be connected by the same function (R). In fact, we define cooperation when at least two sub-systems jointly perform behaviour. In this case, each of these sub-systems can be seen as a system with its own values and behaviours, when these values and behaviour are refined from abstract ones in the abstract system (see Figure 3).

![Figure 4 Refinement Theory [24]](image)

### 4.2. Management Issues

Management issues are composed by (a) a regulatory condition that shows “the governing condition by which the actual course of affairs may be judged”; (b) Regulatory method that takes the action when the situation is satisfied [25].

We represent management issues as management values composed by regulatory condition and regulatory method. In this paper, we take hazard material management plan of Arizona Division of Emergency Management under Disaster Mitigation Act of 2000, (DMA 2000) Public Law 106-390, October 30, 2000 [26]. This plan states “Through the powers vested in the Governor, the division shall coordinate the cooperative effort of all agencies including the Federal government, this State and its political subdivisions to alleviate suffering and loss resulting from disaster.” As such, we extract management issues for “reducing the loss of life, property, and human suffering”, illustrated in Figure 5 and revealed by [26], as (1) Regulatory Conditions: disasters, and (2) Regulatory Method: coordinating the cooperative effort of all agencies.

### 4.3. Policies

Policies are technology independent realization mechanisms to address management, which should be defined on performance of system behaviors by applicability conditions and given rights.

Figure 6 represents the policy specification in cooperative systems. A policy is defined by four components: Subject, Target, Behavior, and Operational Information, where the subject performs the behavior on the target when subject and target are sub-systems in the cooperative system [27]. The operational information shows performance information for the behavior. It is composed by (a) Constraint as the applicability conditions of the policy and (b) Modality as the right of subject or target to perform the behavior. In terms of modality, policies
can be categorized into: (1) **Obligation Policies** define the right of Subject to involve in the behavior with Target. (2) **Authorization Policies** define the right of Target to involve in behavior with Subject.

Therefore, a policy has one of the following modality values: positive authorization (permitting), negative authorization (forbidding), positive obligation (requiring), and negative obligation (deterring). To sum up, whenever we have a negative authorization or obligation we will not be able to perform the behavior [27].

4.4. P-CARD: A Process for Realization of Contextual Awareness Behavior

Policies are defined in distributed cooperative environments, when roles are involved in a cooperative behavior. Roles involved in the behavior related to a specific policy have their own responsibility to perform the behavior and their own understanding of situation.

In Sections 4.2 and 4.3, we showed how policies can represent management issues. This section presents a P-CARD as a four-step process based on
refinement theory for realization of contextual awareness behaviors of roles driven by policies, illustrated in Figure 5. The objective of the proposed four-step process is to translate policies to contextual awareness behaviors for each role involved in the cooperative system. This process uses refinement theory and translates policies to awareness behaviors. The refinement results in understanding of context. As such, roles being aware of context will be able to enhance cooperation.

P-CARD is described below and illustrated using the Casa Grade rail incident scenario (see Figure 5):

**Step 1: Identifying appropriate action of roles.**
The objective of this step is to recognize what the cooperative role should do to perform the cooperative behavior. This step represents the cooperative role’s responsibility during the performance of the cooperative behavior. In order to represent this responsibility, we refine cooperative behavior between subject and target into responsibility of subject or target as cooperative roles. In the scenario, the refined cooperative behavior into Operator’s responsibility is “contact”.

**Step 2: Realizing awareness behavior of cooperative roles.**
The objective of this step is to realize awareness behavior of the cooperative role when she/he performs her/his responsibility. The awareness behavior is derived by the policy in the cooperative behavior between roles. We refine Operational information to the awareness behavior. Consequently, the constraint and modality in policy will be refined to the components of awareness behavior. In the scenario, we refine the operational information into Operator’s awareness behavior. Therefore, constraint and modality as parts of operational information will be refined into “fire in train” and “requiring”, respectively. According to what we discussed above, the policy will be applied here, and Operator’s awareness behavior is:

**Operator’s Awareness Behavior:**
*When there is a fire in a train, Operator is required to contact the train company for cargo identification.*

**Step 3: Recognizing accessible situational information.**
The objective of this step is to recognize the situational information that the cooperative role is able to access. When the subject performs a behavior on a target, the context will be changed. Therefore, we represent the context by pre-values which will be changed to post-values by performing the behavior. In order to conceptualize the situation understood by the cooperative role, we need to refine the pre-context into the cooperative role’s pre-situation. This step shows the situational information aware by the cooperative role. In the scenario, pre-context as the abstract value consists of real fact that happens during Casa Grande rail incident. We refine pre-context into the understood situation by Operator as pre-situation “fire in train” that represents understanding of Operation about fire in the train.

**Step 4: Understanding the applicability of awareness behavior.**
The objective of this step is to understand when the cooperative role should use her/his awareness behavior. We compare pre-situation in Step 3 and refined constraint in Step 2. If they address the same condition, then the awareness behavior is applicable. Once the awareness behavior becomes applicable, the cooperative role performs her/his responsibility regarding her/his right defined by refinement of modality of the policy. In the scenario, pre-situation and the refined constraint are the same, “fire in train”. Therefore, the Operator’s responsibility, “contact”, is “required”.

As a result of following the four steps above applied to the Casa Grande rail incident, Operator realizes fire in the train. Operator’s awareness behaviour recognized in Step 2 indicates that in such a situation she/he is required to contact the train company.

### 4.5. Representation of Contextual Awareness Behaviors in Context-aware Software Agents

As discussed in Section 3, we represent cooperative roles with context-aware software agents. We also need a means to represent the awareness behavior into software agents. We do this with rules, as explained below.

We design our system by using adaptive context-aware software agents that dynamically build rule-based reactive software agents guided by “if-then” decision rules. In this approach, the adaptive context-aware software agent finds the possibility of applying the method of the each of management values in the behaviour. If it is applicable for the behaviour, in order to dynamically make a policy, it refines the condition to constraint and method to modality. Then it builds a rule-based reactive software agent which is based on the dynamically-made policy.

In fact, the adaptive context-aware software agent defines the behavioural model of contextual awareness as an automaton with finite number of states when the transition function can be defined as the behavior of the reactive software agent in the situation. Therefore, it dynamically builds a reactive software agent and defines its transition function with rules:
Operator Adaptive Context-aware Software Agent:

if (the Method can be applied in the Behavior) then
Policy = refine policy (Condition, Method);
Build reactive software agent (Policy)

In fact, awareness behavioral model of reactive software agents in a disaster is an automaton with a rule-based transition function consisting of rules such as

if (the situation is applied in the context) then do (or do not do) some actions Q.

Comparing this rule with the awareness behavior structure directs us to the idea of representing the awareness behavior with such rules. As such, we are able to define a rule for each contextual behavior. The condition part of the rules presents when the pre-situation addresses the refined constraint of policy. The action part of the rule presents the right of the cooperative role represented by software agent to do her/his responsibility. While do not address background on agent-based awareness behavior implementation in this paper, readers wishing additional background are recommended to refer to [28]. In the scenario, we represent Operator with a reactive software agent. Thus, the dynamically built software agent will be:

Operator Reactive Software Agent’s Awareness Behavior:

if (situation == fire in train) then requiring(contact to the train company for cargo identification).

Therefore, when in 1983, in Casa Grande, AZ a fire happened in the train and the operator recognized it, this software agent would suggest that the operator contacts the train company to indentify cargo information.

The proposed representation of contextual awareness behaviours can be implemented by context-aware software agents using different agent development environments such as Java Agent Development Environment (JADE). The last two authors of the paper explain in [28] how to implement the contextual awareness behaviours of software agents according to rule-based transition functions.

5. Discussion

In disasters, supportive software agents can contribute to an effective emergency response. The dynamic nature of disasters in which situations can change dramatically makes the realization of the contextual awareness behaviours difficult, particularly when cooperation between roles is inadequate. For example, in the Casa Grande rail incident, the emergency operator did not have sufficient cooperation with agencies such as the train company. As a result, the realization of the right contextual awareness behaviour, i.e. contacting the train company for cargo identification, failed.

Policies are a way to define technology independent behaviours according to different situations without the need for direct orders. However, correct behaviours as driven by policies are dependent upon contextual awareness of the situation. In Section 4, we demonstrated a method that refines policies to contextual awareness behaviors for different situations, and in Section 4.4, we demonstrated that a rule-based transition function [13] as a means of implementing selecting policies in context-aware software agents. We thus propose that policy-driven context-aware software agents could enable cooperative roles in the Casa Grande train incident scenario to select the right behavior according to the situation.

5.1. Potential Solution for Improving Intelligent Communication Systems

The proposed method can be developed in emergency response centers by using intelligent communication systems, such as Avaya [29]. Intelligent communication systems are an application of artificial intelligence in cooperation for producing human-friendly interfaces. One of objectives for intelligent communication systems is Service Logical. Service logical provides communication processing functions consists of (a) Service Logical Program (SLP) to receive users’ requests, and (b) Service Logical Interpreter (SLI) to execute the users’ requests and give the result to the users which can be a suggestion for a situation given to SLP. SLP and SLI as two components of service logical can be implemented by software agents [29].

Figure 7 reviews the Casa Grande rail incident scenario when operators use service logical implemented by context-aware software agents. When the operator contacts to the SLP agent, it changes the context and asks the SLI agent to give the effective behavior to the operator. The contextual awareness behaviors of existing service logical products have been not developed according to a systemic method to translate management issues into contextual awareness behaviors of agents. The presented method in this paper proposes a way to define contextual awareness behaviors of SLI agents according to management issues.
Hence, by installing an intelligent communication system using the proposed method in an emergency response center, operators will be suggested for the right contextual awareness behavior, which leads to enhancing cooperation.

5.2. Bridging the Gap between Management Issues and Software Agents

Research on software agents typically addresses contextual awareness in terms of programming software agents according to context [15-17,21,28]. Such software agents should reason and make suggestions to cooperative roles. The Casa Grande rail incident scenario shows programming software agents without considering management issues has two weaknesses: First, the reasoned suggestions made by software agents are dependent on their understanding of the context, limited by their programming. As such, agents are not able to adapt behavior based on complex management issues. For example, Ulieru and Worthington [17], in their proposal that agents could have been useful during emergency response, do not address how the agents would need to behave different based on the implications of dynamism in disasters, which requires policies as a mechanism to realize management issues. Second, standard approaches to software agents are technology dependent. While involved government agencies in emergency responses often use different technologies [30]; as such, integrated cooperation can be difficult to achieve. Other research addresses specification of policies and management issues [25,27], but fails to address the need to translate these into contextual awareness behaviors of software agents.

The method proposed in Section 4 fills the gap described by providing a translation method between management issues, policy specification and contextual awareness of software agent behaviors.

5.3. Employing Agents as an alternative for Training Humans

One of solutions for increasing awareness in cooperative environments is to take people involved and train them to act based upon awareness of their environment as opposed to train software agents. Literatures [31,32] in human-computer interactions address this issue and describe different problems can be solved by software agents. Some of these problems, related to this research, are pointed out here:

Irrelevant Information: As we mentioned in Section 2, disaster management often needs cooperation in a highly distributed environment. Software agents can support human roles offering integrated automated communication to retrieve information. In order to avoid bombarding an individual with irrelevant or loosely relevant information, using software agents is beneficial [4]. For example, in our scenario there is huge number of situations and different management issues and policies to be applied. It unlikely happens that an individual can manage appropriateness of information in different situations without any mistake, as mistake in disasters can be very costly.

Sporadic contacts of cooperative roles: In cooperative disaster management, when roles need to remotely cooperate, one of challenges is sporadic communication. Software agents provide mediated communication among human roles, which has been confirmed by different cases in [15,19,33,28,32].

5.4. Limitations

There are three limitations related to this study, which we discuss below:

First, the proposed method supports behaviours of agents when they are aware of the context but not aware of the others’ context awareness. If cooperative software agents could be aware of the other roles’ context awareness, then they can define the situation with more available information. It would help them to respond with more effective behaviours. In order to have such awareness, software agents need to cooperate, which may not be allowed or defined in the environment. Therefore the proposed method enhances the cooperation during an emergency response when cooperation is not sufficient and awareness of others’ context is unlikely to achieve.

Second, policies may interact with each other and a new added policy may conflict with existing
previous policies. There are bodies of method for policy conflict recognition such as [34], which can be added to P-CARD as future work.

Third, the method proposed is new and untried either in a simulated or real production environment. An initial proof of concept as an exemplar is made in the Casa Grande rail incident while exemplars are a common way to provide initial validation to a new method in software engineering.

6. Related Work

Currently, there are various approaches and techniques for situation management in emergency response scenarios. Some of researches in this area address development of supportive systems for disaster management. Hinske and Ray [30] provide a five-layered framework for management of Pandemic control Systems, which consists of Organization, Process, Service, Application and ICT. In application layer, Ulieru and Doursat [15] propose a new school of system development for emergency situations based on intelligent software agents. Marsh [18] also proposes a cognitive situation management based on awareness, which can be applied for disasters. In addition to that, Ray and Chattopadhyay [16] take adaptive software agents and awareness approaches in emergency situation management. They address context-aware gents for disaster management and propose a fuzzy model for awareness of software agents in disasters.

6.1. Justification by Previous Work

Much research, as mentioned above, employs software agents to dynamic nature of disasters. However, research in disaster management highlights the significant role of cooperation; it does not address the problem of insufficient cooperation and how to enhance the cooperation. For example, work of Ulieru and Doursat [15] or Ray and Chattopadhyay [16] employs software agents as a means in cooperative environment and not to enhance cooperation. According to our knowledge only Yuan and Detlor [19] reveal the need for enhancing cooperation by using software agents to facilitate knowledge sharing as a proposal to research for this purpose, while they cite the Casa Grande rail incident without contributing to solve the problem.

This paper focuses on the process for realization of awareness in emergency response and proposes use of policies as technology independent mechanisms for realization process.

7. Conclusion

This paper proposes a four-step policy-driven process, called P-CARD to realize contextual awareness among cooperative roles using software agents in emergency response. Software agents are used to inform cooperative roles of an appropriate course of action. Policies, which are refined to context-aware behaviors, are realized in software agents using a rule based transition function.

P-CARD fills a gap in current research and practice by providing a translation method between management issues and policy specification and contextual awareness of software agent behaviors. As policies are technology independent, the proposed method offers possibilities for better intra-governmental agency collaboration during an emergency response.

In addition, the process applied to software agents offers a possible enhancement of capability in intelligent communication systems for supporting decision-making of operators at emergency response centers.

The concept is illustrated by application to a scenario based on the Casa Grande rail incident. This paper shows how a better decision to contact the train company might have been made, if the proposed four-step had been implemented.

References


