Agent-Based Analysis and Support for Incident Management*

Mark Hoogendoorn[1,†], Catholijn M. Jonker[2], Jan Treur[1], and Marian Verhaegh[3]

[1]Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands,
tel. +31 20 598 7772, fax. +31 20 598 7653
{mhoogen, treur}@cs.vu.nl
[2]Radboud University Nijmegen, Nijmegen Institute of Cognition and Information
Montessorilaan 3, 6525 HR Nijmegen, The Netherlands
C.Jonker@nici.ru.nl
[3]Quartet Consult, Jaap Edenlaan 16, 2807 BR Gouda, The Netherlands
info@quartetconsult.nl

Abstract. This paper presents an agent-based approach for error detection in incident management organizations. The approach consists of several parts. First, a formal approach for the specification and hierarchical verification of both traces and properties. Incomplete traces are enriched by enrichment rules. Furthermore, a classification mechanism is presented for the different properties in incident management that is based on psychological literature. Classification of errors provides insight in the functioning of the agents involved with respect to their roles. This insight enables the provision of dedicated training sessions and allows software support to give appropriate warning messages during incident management.

Keywords: Error detection, incident management, formal analysis, handling incomplete information, agent-based support.

1 Introduction

The domain of incident management is characterized by sudden events which demand immediate, effective and efficient response. Due to the nature of incident management, those involved in such processes need to be able to cope with stress situations and high work pressure. In addition to that, cooperation between these people is crucial and is not trivial due to the involvement of multiple organizations with different characteristics (e.g. police, health care, fire department). As a result of these difficulties, often errors occur in an incident management process. If such errors

---

* This paper is partly based on Hoogendoorn, M., Jonker, C.M., Treur, J., and Verhaegh, M., Agent-Based Analysis and support for Incident Management, In: Klusch, M., Rovatsos, M., and Payne, T. (eds.), *Proceedings of the Tenth International Workshop on Cooperative Information Agents (CIA 2006)*, LNCS 4149, 2006, pp. 109-123.

† Corresponding author

are not handled properly, this may have great impact on the successfulness of incident management.

Research within the domain of computer science and artificial intelligence is being performed to see whether automated systems can improve the current state of affairs in incident management (see e.g. Oomes and Neef, 2005; Storms, 2004). One of the problems is that the information available is incomplete and possibly contradictory and unreliable. As a result, more advanced techniques are needed to enable automated systems to contribute an improvement of the incident management process.

This paper presents an agent-based approach to monitor, analyze and support incident management processes by detecting occurring errors and providing support to avoid such errors or to limit their consequences. The approach is tailored towards the characteristics of incident management. First of all, the approach includes a method which deals with incomplete information. In addition, a diagnostic method based on refinement within the approach can signal whether certain required properties of the incident management organization are not satisfied, and pinpoint the cause within the organization of this dissatisfaction. The approach is based on the organizational paradigm nowadays in use in agent systems (Boissier et al. 2005; Giorgini et al. 2004) which allows the abstraction from individual agents to the level of roles. Such an abstraction is useful as typically specification of the requirements in this domain is done on the level of roles (e.g. the police chief should communicate a strategy for crowd control). In case errors are observed in role behavior, they are classified to have more insight in what kind of errors are often made by a particular agent participating in the organization, in order to propose a tailored training program for this agent. In the future the approach as a whole can be incorporated in cooperating software agents for monitoring and providing feedback in training sessions, and software agents which can even monitor incident management organizations on the fly, giving a signal as soon as errors are detected, and providing support to avoid their occurrence or to limit their consequences.

Section 2 introduces the domain of incident management and, more specifically, the situation in the Netherlands. Thereafter, Section 3 introduces the formal language used to specify traces and behavior. Section 4 presents an approach for handling incomplete information by means of enrichment rules whereas Section 5 presents a simple example of a specification of properties in the form of a hierarchy. Section 6 presents such properties for incident management organizations. Furthermore, Section 7 presents the classification scheme for errors, including specific incident management decision rules. Results of a case study are presented in Section 8 and finally, Section 9 is a discussion.

2 The Domain of Incident Management

In this Section, a brief introduction to the domain of incident management in the Netherlands is given. In the Netherlands four core organizations are present within incident management: (1) the fire department; (2) the police department; (3) health care, and (4) the municipalities involved. The first three parties mentioned each have their own alarm center in which operators are present to handle tasks associated with the specific organization.

A trigger for starting up an incident management organization is typically a call to the national emergency number, which is redirected to the nearest regional alarm center in which all three parties have their own alarm center. The call will be redirected to the most appropriate alarm center of the three parties. In case the operator of that alarm center considers the incident to be severe enough to start up the full incident management organization, he informs the alarm centers of the other organizations as well. Initially, the three alarm centers will send the manpower they think is appropriate for the incident reported. There are four levels of incident management organization, depending on the seriousness of an incident. The first level of incident management organization concerns the operational level. The police units, fire fighting units and health care units are involved. After the manpower has arrived on the scene, each part of the organization in principle acts on its own, each having a different coordinator of actions. In the case of the fire department this is the commander of the first truck to arrive, for health care it is the paramedic of the first ambulance and for the police there is no such coordinator as they have a supporting role. Each of the coordinators are in charge to manage the operations in the disaster area until the dedicated operational leaders of the organization arrive at the scene. The responsibilities of the organizations are briefly described as follows: the fire department takes care of the so called "cause and effect prevention", the health care organization is in charge of providing medical care, and the police takes care of routing of the various vehicles and crowd control. After the initial phase without structural coordination, an organization is formed in order to coordinate all actions of the individual organizations in case this is still necessary. *********** The fire department is usually in charge of the operational side of this organization and the mayor of the municipality is in charge of the policy part. The mayor is responsible for the formation of the disaster staff for coordinating policy decisions, and is therefore informed of the situation. The operational coordination structures are formed after deliberation between the various parties on the scene has resulted in a mutual demand for such a coordination structure. In case it is decided to scale up, the operators of the alarm centers start warning the relevant people. The second level scales up to tactical and strategic coordination. A regional operational team will be formed, which is responsible for adjusting the activities in the coordinating centers to the activities in the disaster area . The tactical coordination will be executed by a regional operational team, managed by the operational leader. The mayor is responsible for starting up the strategic coordination. The strategic coordination is done by the municipal policy team. Within the operational coordination a commander disaster area is added. The commander of the fire department usual is acting as the commander disaster area. In case the disaster area involves a region, the third level of incident management has to be started up. The provincial and national coordinating centers will be alarmed to assist executing the tactical coordination. At a strategic level the provincial governor and the minister of interior will be alarmed.

In case the full coordination structure is in place, the organization resembles the structure shown in Figure 1. For more details on the full coordination structure, see (Netherlands Institute for Fire Service and Disaster Management 2003).

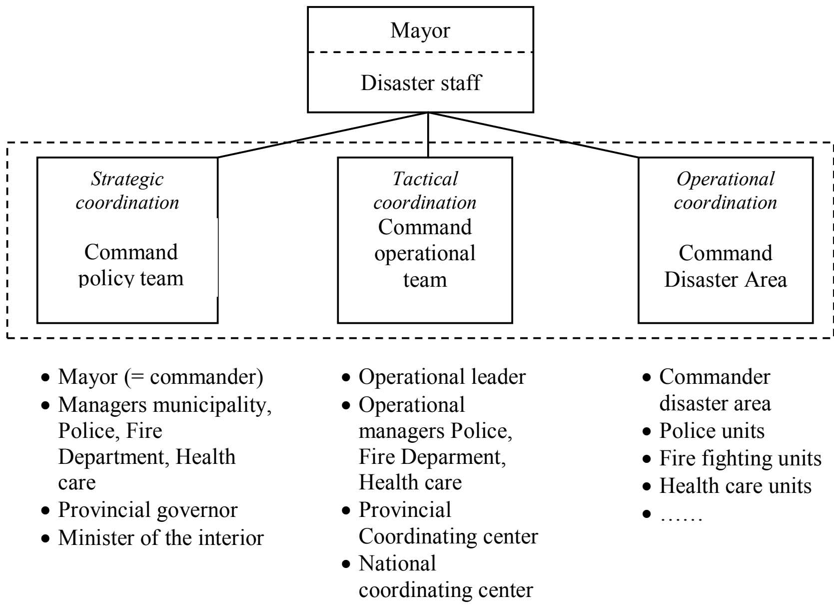

Fig.1. Full coordination structure for incident management

3 Modeling Method Used

This section describes the language TTL (for Temporal Trace Language) (Jonker and Treur, 2002) used for expressing dynamic properties as well as the expression of traces. Furthermore, the language meta-TTL is introduced for second-order dynamic properties.

3.1 The Language TTL for Dynamic Properties

To formally specify dynamic properties that are essential in an incident management organization, an expressive language is needed. To this end the Temporal Trace Language is used as a tool; cf. (Jonker and Treur 2002). For the properties occurring in the paper informal, semi-formal or formal representations are given. The formal representations are based on the Temporal Trace Language (TTL), which is briefly described as follows; for more formal details, see Appendix A.

A state ontology Ont is a specification (in order-sorted logic) of a vocabulary. A state for ontology Ont is defined as an indication of which state properties expressed in ontology Ont hold in the state and which do not hold. The set of all states is modeled by the sort STATE. A fixed *time frame* T is assumed which is linearly ordered. A *trace* or *trajectory* γ over a state ontology Ont and time frame T is an indication of which state occurs at which time point, for example if a discrete time frame based on natural numbers is taken, a trace is a sequence of states γ_t ($t \in T$). The set of all traces over state ontology Ont is modeled by the sort TRACE. Depending on the application, the time frame T may be dense (e.g., the real numbers), or discrete (e.g., the set of integers or natural numbers or a finite initial segment of the natural numbers), or any other form, as long as it has a linear ordering. A *dynamic property* over state ontology Ont is a temporal statement that can be formulated with respect to traces based on the state ontology. Such temporal statements can express, for example, a temporal relationship between the fact that in a given trace a certain state property holds at a certain time point and another state property holds at some other time point. For more formal details, see Appendix A.

3.2 The Language Meta-TTL for Second-Order Dynamic Properties

The formalizations of the properties sometimes take the form of second-order dynamic properties, i.e., properties that refer to dynamic properties expressed within TTL. Such second-order dynamic properties are expressed in meta-TTL: the meta-language of TTL. Again, for more formal details, see Appendix A.

4 Handling Incompleteness of Information by Enrichment Rules

The trace of occurrences as logged during or reported from an incident management process usually is incomplete and therefore difficult to analyze. To overcome this incompleteness problem, additional assumptions have to be made on events that have occurred but are not explicitly mentioned in the logged trace. Such assumptions are addressed in this section. These extra assumptions enrich the trace with elements that are derived from the information in the trace itself, for example at later time points in case an analysis is performed afterwards. An example is the assumption that if at some time point an estimation of the situation is communicated, then at previous time points the necessary information to make that assessment was received or observed by the communicating role.

Addition of such elements to enrich a trace are based on rules which express that given certain trace elements, an additional element can be assumed. These rules in principle can be of two forms: Strict rules which can always be applied and provide conclusions that are certain, and defeasible rules which are used in case strict rules are insufficient to obtain a trace with a reasonable amount of information. However, it is not always possible to claim that a rule is a strict rule. Therefore, such rules are considered premises for the whole analysis.

Examples of such rules are presented below, note that the formal form of these rules can be found in Appendix B. Rule EP1 states that everybody present on the scene is assumed to have an internal judgment about the seriousness of the disaster:

***EP1:* Internal judgment at scene**
if   at time t role R is present at the scene
and   situation S is the case
and   S is classified as being a disaster
then   there exists a later point in time t2 < t+d at which R has an internal judgment that this situation is a disaster

Furthermore, in case a role receives a communication that the situation is a disaster and this role does not communicate that he does not believe it being a disaster, then it is assumed that he has the internal judgment that it concerns a disaster:

***EP2:* Internal judgment based on communication**
if   at time t R1 communicates to R2 that the current situation S is a disaster
and   there exists no time point at which R2 communicates to R1 he thinks the situation is not a disaster
then   at every time point t2 > t R2 interprets the current state of affairs as being a disaster

5 Property Hierarchies: A Simple Example

This section shows how, for a simple example, properties to be satisfied within an organization can be represented in the form of a property hierarchy. Specifying properties in such a hierarchy has as an advantage that diagnosis of properties can be done in a top down fashion. Such a diagnostic process starts by checking highest level property, and in case such a property is not satisfied pinpoints the error by gradually going down the tree to the unsatisfied properties, while leaving the satisfied properties and their refinements aside. Property hierarchies obtained from the field of incident management are specified in the section hereafter.

The simple example concerns the evacuation of a building in case of an alarm. The overall goal of such a process is to evacuate all people from the building within a certain duration *d* after the alarm has sound:

***SP1(d):* Evacuate building**
if   at time t the evacuation alarm sounds in a building
then   there exists a time point t1 later than t and before t + d such that at t1 there are no more people inside the building.

Again, the formal forms for these properties can be found in Appendix B. Of course such a goal is not simply accomplished by itself: more refined properties can be formulated that together enable reaching this goal. These properties are shown in the tree in Figure 2. Three main, more refined properties constitute the achievement of the goal. First of all, a certain percentage of people will simply leave the building upon their own initiative after hearing the alarm (SP2). People leaving on their own initiative leave the building before αd, where α is between 0 and 1. The percentage of people that do not leave the building by themselves, are told to do so by an appointed person that checks all the rooms to be sure the building is empty (SP3). This results in these people leaving the building as well. Note that the duration *d* to be set, depends on the allowed percentage of people not directly responding. In case of a tight *d* this percentage should be low, otherwise the appointed person can never be done in time. A parameter β with a value between α and 1 is used to specify when these people should have been asked to leave the building (which should be before β*d*). Finally, the appointed persons themselves leave the building (SP4), due to the parameter β at which the people should have been informed, the appointed persons have a time frame between β*d* and d to leave the building.

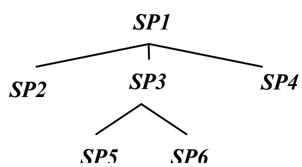

Fig. 2. Property hierarchy for the evacuation of a building upon an alarm

SP2(α, d, p): **Leave immediately**

if	at time t the evacuation alarm sounds in a building
then	there exists a time point t1 later than t and before t + αd (with α between 0 and 1) such that at least *p* percent of all person initially in the building are outside at time point t.

SP3(α, β, d, p): **Leave after correction**

if	at time t the evacuation alarm sounds in a building
and	at time t + αd at least *p* percent of the people are outside of the building already
and	at time t + αd person P is still in the building
and	person P is not responsible for emptying the building
then	there exists a time point t1 later than t + αd and before t + βd (with β between α and 1) and a person AP such that person AP is appointed for emptying the building
and	person P is told at time point t1 by AP to leave the building
and	before t + d this person P is outside of the building

This property again can be refined into two lower level properties. First of all, given the same condition, the communication by the appointed person takes place (SP5), and secondly, once a person receives this communication he leaves the building (SP6).

SP5(α, β, d, p): **Communicate correction**

if	at time t the evacuation alarm sounds in a building
and	at time t + αd person P is still in the building
and	at time t + αd at least *p* percent of the people are outside of the building already
and	person P is not responsible for emptying the building
then	there exists a time point t1 later than t + αd and before t + βd and a person AP such that person AP is appointed for emptying the building
and	person P is told at time point t1 by AP to leave the building

SP6(β): **Leave after receiving communication of correction**

if	before time t + βd person P is told by AP to leave the building

then there exists a later point in time t1 before t + d at which person P is no longer in the building.

The final property indeed specifies that the appointed person leave the building as well:

SP4(α, d, p): **Appointed persons leave before deadline**
if at time t the evacuation alarm sounds in a building
  and person P is an appointed person
  and at time t + αd at least *p* percent of the people are outside of the building already
then before t + d this person P is outside of the building

6 Property Hierarchies for Incident Management Organizations

This section presents generic properties for incident management organizations in the Netherlands. Only the informal and semi-formal forms are presented here. For the formal form of these properties, see Appendix B.

6.1 Warning of Relevant Parties

The warning of relevant parties by the operator is a high level property stating that: "the operator should alarm all necessary parties in case it is informed of an incident":

P1(d): **Warn relevant parties**
if at time t the operator is informed about an incident type I by a role R1,
 and for incident type I role R2 should be informed according to the disaster plan
then there exists a time t2 later than t and before t + d at which R2 is informed about the incident type I

This property can be refined into a number of similar properties restricted to specific categories of roles that should be informed. For diagnosis, at the highest level property P1(d) can be checked, for example with the result that P1(d) is not satisfied which means that not all relevant parties were informed (but without information on which specific categories were not informed). At one level lower, the diagnosis can be refined by checking the refined properties, resulting in an indication of which of the categories of relevant roles were not informed.

6.2 First Arriving Ambulance

Second, the behavior of the first arriving ambulance is addressed. First, a formal definition of the first arriving ambulance is given:

first_arriving_ambulance(γ:TRACE, t:TIME, A:AMBULANCE)
An ambulance is the first arriving ambulance if:
the ambulance arrives at the scene of an incident at time t
and there does not exist a time t' < t at which another ambulance arrived at the scene of the incident

On the highest level, the first arriving ambulance behavior is described by three important aspects: (1) signaling the green alarm light; (2) communicating a situation report, and (3) presence of at least one person belonging to the ambulance until the officer on duty arrives at the scene:

***P2:* First arriving ambulance global behavior**
if at a time t ambulance A is the first to arrive at the scene
and at time t3 > t the officer on duty arrives at the scene
then for all t2 ≥ t and t2 < t3 at least one person belonging to the ambulance should be present at the ambulance
and for all t4 ≥ t the ambulance is signaling the green alarm light
and there exists a time t5 later than t at which the driver of that ambulance communicates a correct interpretation of the situation to the operator.

This property can be related to lower level properties as shown in Figure 3. When trying to diagnose why the highest level property is not satisfied, the properties on the lower level can be checked. In case such a property is not satisfied, and it concerns a leaf property, at least one cause for the non-fulfillment of the high-level property has been found. Otherwise, go further down the tree to find the cause. In the tree a number of properties are present to enable satisfaction of P2. First of all, the signaling of the green light, as expressed below.

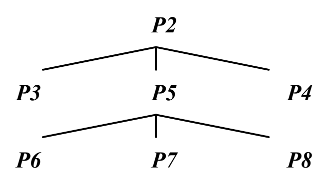

Fig. 3. Property hierarchy for the first arriving ambulance.

***P3:* First ambulance green light behavior**
if at a time t ambulance A is the first to arrive at the scene
then for all later points in time t2 the ambulance is signaling the green light.

Second, the presence of a person belonging to the ambulance for the time until the officer on duty is present:

***P4:* First arriving ambulance personnel presence**
if at a time t ambulance A is the first to arrive at the scene
and at time t3 > t the officer on duty arrives at the scene
then for all t2 ≥ t and t2 < t3 at least one person belonging to the ambulance should be present at the ambulance

Finally, a property expressing the communication of the correct situation to the operator:

***P5(d):* First arriving ambulance interpretation**
if at a time t ambulance A is the first to arrive at the scene
then at a later point in time t2 < t + d the driver of that ambulance communicates a correct interpretation of the situation

Note that parameter d includes the time to interpret the situation plus the time to start communicating that particular interpretation. Testing whether the interpretation was correct can be performed afterwards (e.g., the amount of casualties). The property P5 can be refined again into three lower level properties. First of all, when arriving at the scene, the paramedic should investigate the current state of affairs:

P6(d): **Paramedic investigation**
if at a time t ambulance A is the first to arrive at the scene
and at time t a paramedic is in the ambulance
then at a later point in time t2 < t + d the paramedic of that ambulance will start an investigation and not be at the ambulance any more

Second, the paramedic will return, communicating the current situation:

P7(d): **Paramedic communication**
if at a time t ambulance A is the first to arrive at the scene
and at time t the paramedic is in the ambulance
and at time t2 the physical position of the paramedic is not inside the ambulance
then at a later point in time t3 < t2 + d the paramedic of that ambulance will communicate a correct interpretation of the situation to the driver

Finally, once the driver has received the communication, he will communicate this to the operator:

P8(d): **Driver communication**
if at a time t the driver of the first ambulance at the scene receives a situation description from the paramedic
then at a later point in time t2 < t + d the driver of that ambulance communicates a correct interpretation of the situation to the operator

6.3 Disaster Staff Activation

Furthermore, properties have been specified for the formation of the disaster staff and activities following from the disaster staff. On the highest level the correctness of these processes in the disaster staff can be described as follows: In case the operator has the internal judgment that the current situation is a disaster, the operational leader will eventually output actions belonging to a strategy communicated by the disaster staff.

P9: **Successful disaster staff**
if at time t the operator judges the current situation as a disaster
then there exists a later point in time t2 at which the disaster staff communicated a strategy
and there exists an even later time at which the operational leader communicates an action appropriate for the strategy according to the disaster plan.

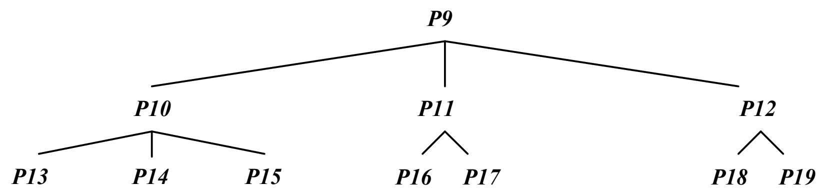

Fig.4. Property hierarchy for the disaster staff activation and functioning.

Such properties can be related to lower-level properties as shown in Figure 4. On the intermediate level, three properties are present. First, the correct initiation of a disaster staff is expressed:

P10: **Correctly activated disaster staff**
if at time t the operator interprets the current situation being a disaster
then at a later point in time t2 the disaster staff will be informed (and assumed to be present as a result)

Thereafter, in case the disaster staff is formed, it should be active, which is characterized by an output in the form of a strategy:

P11: **Active disaster staff**
if at time t the organizational unit called disaster staff is informed
then at a later point in time $t2 > t$ the organizational unit outputs a strategy S

Finally, such a strategy should lead to actions be taken by the operational leader:

P12: **Active operational leader**
if at time t the operational leader is informed of a strategy S to be applied
then at a later point in time $t2 > t$ the operational leader will command the appropriate actions according to the disaster plan to the roles.

Each of these intermediate properties can again be split up to properties for individual roles within the organization. In order to obtain property P10 a number of properties need to hold. First of all, the mayor should be warned by the operator:

P13(d): **Warn mayor**
if at time t the operator interprets the current situation being a disaster
then at a later point in time $t2 > t$ and $t2 < t + d$ the operator communicates the occurrence of a disaster to the mayor.

Thereafter, the mayor should decide to form the disaster staff:

P14: **Form disaster staff**
if at time point t the mayor interprets the current state of affairs as being a disaster
then at a later point in time $t2 > t$ the mayor forms the organizational unit called disaster staff

Finally, in case the mayor communicates the decision to form the disaster staff, the operator should warn the appropriate parties:

P15(d): **Warn rest disaster staff**
if at time t the operator receives the request of the mayor to form the disaster staff
and role R is part of the disaster staff
then at a later point in time $t2 > t$ and $t2 < t + d$ the operator communicates to role R that the disaster staff is being formed.

Regarding the intermediate property P11 the following properties need to hold for satisfaction of the intermediate property. First, after the mayor has decided to form the disaster staff he will eventually request advice from his disaster staff.

***P16:* Start deliberation**

if	at time t the mayor decides to form the disaster staff
then	at a later point in time t2 > t the mayor starts a deliberation within the disaster staff by requesting advice

After such advice is received, he should choose the appropriate strategy:

***P17:* Choose strategy**

if	at time t starts a deliberation within the disaster staff by requesting advice
then	at a later point in time t2 the mayor communicates a strategy to the operational leader

Finally, the intermediate property P12 is refined to two other properties. First, the operational leader should discuss the strategy with his operational team:

***P18:* Choose action**

if	at time t the mayor communicates a strategy S to the operational leader
then	at a later point in time t2 > t the operational leader requests his operational team for advice how to implement S

Finally, the operational leader communicates actions to be performed, based on the advices obtained in the discussion.

***P19:* Communicate action**

if	at time t the operational leader request his operational team for advice how to implement S
then	at a later point in time t2 the operational leader will communicate actions appropriate for strategy S according to the disaster plan

6.4 Ambulance Routing

Finally, properties are specified regarding ambulance routing. The police should act as follows:

***P20:* Route plan includes all wounded nests**

if	at time t there are n wounded nests
and	at a later time point t2 > t the police communicates details concerning the route to be taken by the ambulances to cpa (the central ambulance post)
then	this communication should contain such a route description that ambulances will be sent to all wounded nests.

An alternative property not following standard procedure expresses that the routing is done based explicitly on victim locations:

***P21:* Send ambulance to all wounded on the scene**

if	at time t there is a wounded person at a position P
then	at a later time point t2 an ambulance will be sent to position P
and	at an even later time point t3 that ambulance will be at position P

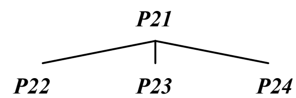

Fig.5. Property hierarchy for ambulance routing

The property hierarchy for this high-level property is shown in Figure 5 and can be decomposed into several other properties. First of all, a wounded person will result in a communication to the operator of the physical position of this wounded person:

P22: **Communicate wounded location**
if at time t there is a wounded person at a position P
then at a later time point t2 this position will be communicated to the operator

For every communication received by the operator, he eventually communicates the location to an ambulance:

P23: **Send ambulance to wounded**
if at time t a wounded person is communicated to be at a position P
then at a later time point t2 an ambulance will be sent to position P

Finally, once the ambulance gets this communication it will arrive at the location at a later point in time:

P24: **Ambulance arrives at wounded**
if at time point t an ambulance is sent to position P
then at a later time point t2 that ambulance will be at position P

7 Human Error Types

This Section presents a classification scheme for the properties in incident management. Such a classification can help to determine the dedicated training needed. The human error classification presented by James Reason (1990) is therefore adopted, who introduces a General Error Modeling approach which identifies three basic error types: (1) skill based slips; (2) rule based mistakes, and (3) knowledge based mistakes. This classification scheme is also used in (Duin, 1992) in which incident management is investigated. Rule based, and knowledge based errors come into play after the individual has become conscious of a problem, which is not the case for skill based slips. In that sense, skill based errors generally precede detection of the problem whereas rule based and skill based mistakes arise during subsequent attempts to find a solution to the problem. Skill based and rule based level error occur when humans use stored knowledge structures whereas knowledge based errors occur when such knowledge structures have been exhausted. Errors are much more likely to occur at the knowledge based level. Table 1 shows how the distinction between the different error types based on several dimensions.

Dimension	**Skill-based errors**	**Rule-based errors**	**Knowledge-based errors**
Type of activity	Routine actions	Problem-solving activities	
Focus of attention	On something other than in the task in hand	Directed at problem-related issues	
Control mode	Mainly by automatic processes (schemata)	(stored rules)	Limited, conscious processes
Predictability of error types	Largely predictable (actions)	(rules)	Variable
Ratio of error to opportunity for error	Though absolute numbers may be high, these constitute a small portion of the total number of opportunities for error		Absolute numbers small, but opportunity ratio high
Influence of situational factors	Low to moderate; intrinsic factors (frequency of prior use) likely to exert the dominant influence		Extrinsic factors likely to dominate
Ease of Detection	Detection usually fairly rapid and effective	Difficult, and often only through external intervention.	
Relationship to Change	Knowledge of change not accessed at proper time	When and how anticipated change will occur unknown	Changes not prepared for or anticipated.

Table 1. Distinctions between different error types (from (Reason, 1990))

For the properties specified for incident management the following classification scheme is used. Skill based properties are those properties that are part of the very basic training of incident management workers. For example, how to start the water pump on a fire truck. A property is classified as a rule based property in case an incident management plan literally includes the property. Finally, a property is called a knowledge based property in case an incident management plan states that a decision needs to be taken, but does not specify how to come to this solution. Using this classification scheme, none of the properties from Section 5 are routine based, whereas properties P1, P3, P5, P6, P8, P13, P14, P15, P16, P19, P22, and P24 are rule based properties. Finally, properties P7, P17, P18, P20, and P23 are knowledge base properties. Note that only the leaf properties are categorized as these are the properties that define the individual role behavior within the organization.

In order to identify which types of error the different participants in the incident management organization are making, the following formula is used (formal form in Appendix B):

if an agent A is allocated to a particular role R in a particular period between t_1 and t_2 in trace γ,
and a situation S occurs in that same period in which property P is relevant for role R whereby the type of property P for role R is of type X (where X is either skill based, rule based or knowledge based)
and the property has a specification which does not hold in the fragment of this trace
then an error of type X is made concerning property P by role R played by agent A.

8 Case Study

As a means to validate the approach presented above, a disaster which has been thoroughly investigated in the Netherlands is taken as a case study. The disaster concerns a bar fire which occurred in Volendam, the Netherlands, at New Years Night of the year 2001. The logs of the disaster have been thoroughly described in (Ministry of the Interior, 2001) and have been formalized using the approach presented in Section 3. Thereafter, the trace enrichment rules from Section 4 have been applied. A part of the resulting trace is shown in Figure 6, which uses the same ontology as used for the formalization of the properties in Section 5. On the left side of the Figure, the atoms are shown that occur during the incident management whereas the right side shows a timeline where a dark box indicates an atom being true at that time point and a gray box indicates the atom being false. The trace is used to verify whether the properties as specified in Section 5 indeed hold for the Volendam disaster. The following properties were shown not to hold: P2, P4, P5, P7, P8, P9, P10, P14, and P20. In other words, in the Volendam case study the first ambulance did not comply to the global desired behavior because the information was not communicated properly, and because there exist time points at which nobody was present at the ambulance. Furthermore, the disaster staff was not activated properly because the mayor did not communicate that the disaster staff should be formed, and finally the ambulance routing of the police was incorrect, but luckily the direct routing of the health care services was satisfied. These results exactly comply to the conclusions in the disaster report (Ministry of the Interior, 2001) which resulted from a thorough investigation of a committee specialized in incident management.

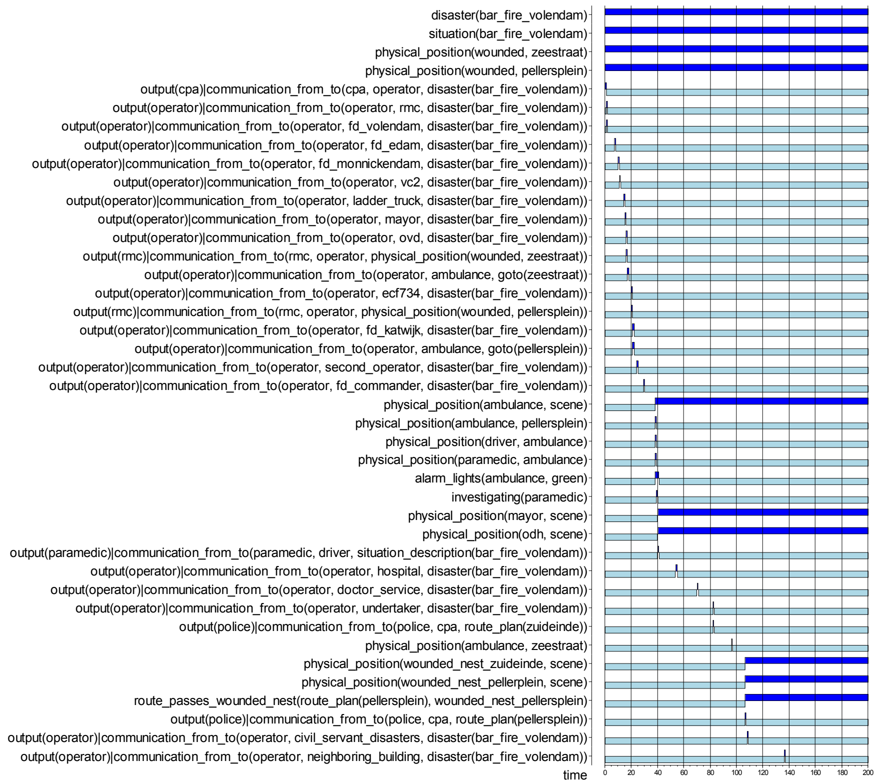

Fig.6. Partial trace of the Volendam case study

9 Discussion

This paper presents an agent-based approach which can be used for error detection in incident management organizations. The approach consists of several parts. First, a formal approach for the specification of both traces and properties that can be verified against these traces is presented. In domains like incident management, traces might be incomplete. Therefore, enrichment rules for these traces are identified to cope with this incompleteness. Furthermore, the properties that ought to be verified against these traces can be specified in a hierarchical fashion: in case the highest level property is not satisfied, the cause of this dissatisfaction can be determined by looking at the properties one level deeper in the tree, which continues until a leaf property is found which is not satisfied. Finally, a classification mechanism is presented for the different properties based on psychological literature. In case an error is observed such a classification immediately gives insight in the functioning of a particular agent playing a role, which enables performing dedicated training sessions or giving appropriate warning messages.

In the future, the approach presented can be incorporated in personal agents of people involved in incident management. Such agents automatically log all incoming and outgoing information in the form of traces and have knowledge on the property the particular role the agent is playing is required to fulfill. In case properties are observed not to be satisfied, a reminder or warning can for instance be given to the person. Such agents can be useful for training sessions, as it can be observed what kind of mistakes a person typically makes, but could possibly even be used during actual incident management.

Many information systems have been developed or proposed that support processes involved in incident management. Already in the 1980s (Wallace and Balogh, 1985) a decision support system for disaster management has been proposed. In (Oomes and Neef, 2005), a system is proposed for the support of scaling up an incident management organization. (Lee and Vught, 2004) presents the IMI system which can be used as an information source and a communication system, enabling crucial information to be sent to the appropriate people immediately, and information sources such as disaster plans to be widely available and easily usable. The reasoning behind these systems is to minimize the errors that occur in incident management. Despite these efforts, errors will continue to occur in incident management due to the stress, pressure, and incomplete information. Minimizing the consequences of such an error is therefore a necessity. This is exactly what can be established using the system presented in this paper.

In the field of information agents, support systems have also been developed for incident management (see e.g. Storms, 2004). In such systems however, the agents again do not check whether errors are made, but simply provide people with information to make sure they are aware of their tasks. This does however not offer a mechanism to detect errors and avoid a chain of unwanted events. Approaches for e.g. detection of protocols (see e.g. Rossie and Busetta, 2004), also called overhearing, have been introduced. These approaches are however more focused on recognizing patterns, not on detection of errors.

Error detection itself is another related research field. In (Fromentin at al., 1995) behavioral properties for a parallel computing system can be specified, and can be checked on the fly. The properties are however specified as simple sequences of states, whereas the TTL language as used in this paper has the ability to express timing parameters between these states, often a necessity in incident management. In (Jard et al., 1994) properties for error detection are specified by means of a finite state machine which again does not allow for time parameter specification.

Acknowledgements

The authors wish to thank the Dutch Ministry of Economic Affairs for funding this research. Furthermore, the authors would like to thank the Netherlands Institute for Fire Service and Disaster Management for sharing their expertise in the domain of incident management.

References

Boissier, O., Dignum, V., Matson, E., Sichman, J. (eds.), Proc. of the 1st Workshop From Organizations to Organization Oriented Programming in MAS (OOOP), 2005.

Duin, M.J. van, Learning from Disasters (in Dutch), PhD Thesis, Leiden, 1992.

Fromentin, E., Raynal, M., Garg, V.K., and Tomlinson, A., On the Fly Testing of Regular Patterns in Distributed Computations, Information Processing Letters 54:267-274, 1995.

Giorgini, P., Müller, J., Odell, J. (eds.), Agent-Oriented Software Engineering IV, LNCS, vol. 2935, Springer-Verlag, Berlin, 2004.

Jard, C., Jeron, T., Jourdan, G.V., and Rampon J.X. "A general approach to trace-checking in distributed computing systems", In Proc. IEEE International Conference on Distributed Ccomputing Systems, pp. 386-404, Poznan, Poland, June, 1994.

Jonker, C.M., Treur, J. Compositional verification of multi-agent systems: a formal analysis of pro-activeness and reactiveness. International. Journal of Cooperative Information Systems, vol. 11, 2002, pp. 51-92.

Lee, M.D.E. van der, Vugt, M. van. IMI – an information system for effective multidisciplinary incident management. In: Carlé, B., Walle, B. van der (eds.), Proceedings of the International Workshop on Information Systems for Crisis Response and Management '04, Brussels, Belgium. 2004.

Ministry of the Interior, Investigation Bar Fire New Years Night 2001 (in Dutch), SDU Publishers, The Hague, 2001.

Municipality of Amsterdam, Disaster Plan (in Dutch), 2003.

Netherlands Institute for Fire Service and Disaster Management, training-course disaster management (in Dutch), 2003

Oomes, A.H.J., Neef, R.M., Scaling-up Support for Emergency Response Organizations, In: Walle, B. van, and Carle, B. (eds.), Proceedings of ISCRAM 2005, pp. 29-41, 2005.

Reason, J., Human Error, Cambridge University Press, 1990.

Rossi, S., Busetta, P, Towards Monitoring of Group Interactions and Social Roles via Overhearing, In: Klusch, M., Ossowski, S., Kashyap, V., and Unland, R. (eds), Cooperative Information Agents VIII, LNAI 3191, Spinger-Verlag, pp. 47-61, 2004.

Storms, P.A.A., Combined Systems: A System of Systems Architecture, In: Proceedings of ISCRAM 2004, pp. 139-144, May 2004, Brussels.

Wallace, W.A., Balogh, F. de, Decision Support Systems for Disaster Management, Public Administration Review, Vol. 45, Special Issue: Emergency Management: A Challenge for Public Administration, pp. 134-146, 1985.

Appendix A: Temporal Trace Language (TTL)

This appendix presents a formal description of the language TTL and meta-TTL.

A.1 The Language TTL for Dynamic Properties

In TTL (Jonker and Treur, 2002), ontologies for states are formalized as sets of symbols in sorted predicate logic. For any ontology Ont, the ground atoms form the set of *basic state properties* BSTATPROP(Ont). Basic state properties can be defined by nullary predicates (or proposition symbols) such as hungry, or by using n-ary predicates (with n>0) like has_temperature(environment, 7). The *state properties* based on a certain ontology Ont are formalized by the propositions (using conjunction, negation, disjunction, implication) made from the basic state properties and constitute the set STATPROP(Ont).

In order to express dynamics in TTL, important concepts are *states*, *time points*, and *traces*. A *state* S is an indication of which basic state properties are true and which are false, i.e., a mapping S: BSTATPROP(Ont) $\rightarrow$ {true, false}. The set of all possible states for ontology Ont is denoted by STATES(Ont). Moreover, a fixed *time frame* T is assumed which is linearly ordered. Then, a *trace* γ over a state ontology Ont and time frame T is a mapping $\gamma : T \rightarrow$ STATES(Ont), i.e., a sequence of states γ_t $(t \in T)$ in STATES(Ont). The set of all traces over ontology Ont is denoted by TRACES(Ont).

The set of *dynamic properties* DYNPROP(Ont) is the set of temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner. Given a trace γ over state ontology Ont, a certain state at time point t is denoted by state(γ, t). These states can be related to state properties via the formally defined satisfaction relation, indicated by the infix predicate |=, comparable to the Holds-predicate in the Situation Calculus. Thus, state(γ, t) |= p denotes that state property p holds in trace γ at time t. Likewise, state(γ, t) |$\neq$ p denotes that state property p does not hold in trace γ at time t. Based on these statements, dynamic properties can be formulated in a formal manner in a sorted predicate logic, using the usual logical connectives such as $\neg$, $\wedge$, $\vee$, $\Rightarrow$, and the quantifiers $\forall$, $\exists$ (e.g., over traces, time and state properties). For example, consider the following dynamic property for a pattern concerning belief creation based on observation:

if	at any point in time t1 the agent observes that the situation is a disaster,
then	there exists a time point t2 after t1 such that
	at t2 in the trace the agent believes that the situation is a disaster

This property can be expressed as a dynamic property in TTL form with free variable γ as follows:

$\forall$t:T [state(γ, t) |= observes(itsadisaster) $\Rightarrow$ $\exists$t' $\geq$ t state(γ, t') |= belief(itsadisaster)]

The set DYNPROP(Ont, γ) is the subset of DYNPROP(Ont) consisting of formulae with γ occurring in which is either a constant or a variable without being bound by a quantifier. For a more elaborate explanation of TTL, see (Jonker and Treur, 2002).

A.2 The Language Meta-TTL for Second-Order Dynamic Properties

The language meta-TTL includes sorts for DYNPROP(Ont) and its subsets as indicated above, which contain TTL-statements (for dynamic properties) as term expressions. Moreover, a predicate holds on these sorts can be used to express that such a TTL formula is true. When no confusion is expected, this predicate can be left out. To express second-order dynamic properties, in a meta-TTL statement, quantifiers over TTL statements can be used.

Appendix B: Properties Formally Specified in TTL

This Appendix presents the properties as presented in an informal or semi-formal form in the paper in a formal form using TTL.

B.1 Internal Judgement Properties (Section 4)

EP1: **Internal judgment at scene**
∀ R:ROLE, t:TIME, S:SITUATION
[state(γ, t) |= physical_position(R, scene) &
state(γ, t) |= current_situation(S) &
state(γ, t) |= disaster(S)]
⇒ ∃t2>t & t2 < t+d [state(γ, t2) |= internal_judgment(R, disaster(S))]

EP2: **Internal judgment based on communication**
∀R1,R2:ROLE, P:POSITION, t:TIME, S:SITUATION
[state(γ, t) |= communication_from_to(R1, R2, disaster(S)) &
¬∃t'>t [state(γ, t') |= communication_from_to(R2, R1, not(disaster(S)))]]
⇒ ∀t2 > t [state(γ, t2) |= internal_judgment(R2, disaster(S))]

B.2 Building Evacuation Property Hierarchy (Section 5)

SP1(d): **Evacuate building**
∀t:TIME
[state(γ, t) |= alarm_bell_sounds
⇒ ∃t1:TIME > t [t1 < t + d & ¬∃P:PERSON [state(γ, t1) |= in_building(P)]]]

SP2(α, d, p): **Leaving immediately**
∀t:TIME
[state(γ, t) |= alarm_bell_sounds
⇒ ∃t1:TIME > t [t1 < t + αd & ∃I:INTEGER [percentage_out_between(γ, t, t1, I) & I ≥ *p*]]]

The percentage of people getting already outside of the building can be expressed in TTL as

percentage_out_between(γ, t, t1, I) ⇔
∃ I2, I3:INTEGER
[amount_out_between(γ, t, t1, I2) & amount_people_in_building_at(γ, t, I3) &
I2/I3 * 100 ≤ I1 < I2/I3 * 100 +1]

where:

amount_out_between(γ, t, t1, I) ⇔
[$\sum_{\forall P:PERSONS}$ case([state(γ, t1) |= ¬in_building(P) & state(γ, t) |= in_building(P)], 1, 0)] = I

and

amount_people_in_building_at(γ, t, I) ⇔
[$\sum_{\forall P:PERSONS}$ case([state(γ, t) |= in_building(P)], 1, 0)] = I

Here for any formula f, the expression case(f, v1, v2) indicates the value v1 if f is true, and v2 otherwise.

SP3(α, β, d, p): **Leaving after correction**
∀t:TIME, P:PERSON, I :INTEGER
[[state(γ, t) |= alarm_bell_sounds &
percentage_out_between(γ, t, t + αd, I) & I ≥ *p* &
state(γ, t + αd) |= in_building(P) &
state(γ, t + αd) |= ¬person_for_emptying_building(P)]
⇒ ∃t1:TIME > t + αd, AP:PERSON
[t1 < t + βd &
state(γ, t1) |= person_for_emptying_building(AP) &
state(γ, t1) |= communication_from_to(AP, P, leave_building) &
∃t2:TIME > t1 [t2 < t + d & state(γ, t2) |= ¬in_building(P)]]

SP5(α, β, d, p): **Communicate correction**
∀t:TIME, P:PERSON, I:INTEGER
[[state(γ, t) |= alarm_bell_sounds &
percentage_out_between(γ, t, t + αd, I) & I ≥ *p*
state(γ, t + αd) |= in_building(P) &
state(γ, t + αd) |= ¬person_for_emptying_building(P)]
⇒ ∃t1:TIME > t + αd, AP:PERSON
[t1 < t + βd &
state(γ, t1) |= person_for_emptying_building(AP) &
state(γ, t1) |= communication_from_to(AP, P, leave_building)]]

SP6(β, d): **Leaving after receiving communication of correction**
∀t:TIME, P:PERSON, AP:PERSON
[state(γ, t) |= communication_from_to(AP, P, leave_building)
⇒ ∃t1:TIME > t [t1 < t + (1-β)d & state(γ, t1) |= ¬in_building(P)]]

SP4(α, d, p): **Appointed persons leave before deadline**
∀t:TIME, P:PERSON, I:INTEGER
[[state(γ, t) |= alarm_bell_sounds &
percentage_out_between(γ, t, t + αd, I) & I ≥ *p* &
state(γ, t) |= person_for_emptying_building(P)]
⇒ ∃t1:TIME > t [t1 < t + d & state(γ, t1) |= ¬in_building(P)]]

B.3 Incident Management Organization Property Hierarchy (Section 6)

P1(d): **Warn relevant parties**
∀I:INCIDENT_TYPE, t:TIME, R1, R2:ROLE
[state(γ, t) |= communication_from_to(R1, operator, I) &
state(γ, t) |= according_to_plan_should_be_involved_in(R2, I)]
⇒ ∃t2 > t & t2 < t + d [state(γ, t2) |= communication_from_to(operator, R2, I)]]

first_arriving_ambulance(γ:TRACE, t:TIME, A:AMBULANCE)
[state(γ, t) |=physical_position(A, scene) & ¬∃t'< t, [∃B:AMBULANCE [state(γ, t') |=physical_position(B, scene)]]

P2: **First arriving ambulance global behavior**
∀A:AMBULANCE, t, t2:TIME
[first_arriving_ambulance(γ, t, A) &
state(γ, t2) |= physical_position(officer_on_duty, scene) &
¬∃t'''< t2 [state(γ, t''') |= physical_position(officer_on_duty, scene)]]
⇒

∀t3 < t2
[t3 ≥ t ⇒ [∃R:ROLE [state(γ, t3) |= physical_position(R, A)]]]
& ∀t4 > t [state(γ, t4) |= alarm_lights(A, green)]
& ∃t5 > t, X:SITUATION[state(γ, t5) |= communication_from_to(driver, operator, situation_description(X)) &
situtation(X)]]

P3: **First ambulance green light behavior**
∀A:AMBULANCE, t:TIME
[first_arriving_ambulance(γ, t, A) ⇒ ∀t2:TIME > t [state(γ, t2) |= alarm_lights(A, green)]]

P4: **First arriving ambulance personnel presence**
∀A:AMBULANCE, t, t2:TIME
[first_arriving_ambulance(γ, t, A) &
state(γ, t2) |= physical_position(officer_on_duty, scene) &
¬∃t''< t2 [state(γ, t'') |= physical_position(officer_on_duty, scene)]]
⇒ ∀t3 < t2 [t3 ≥ t ⇒ [∃R:ROLE [state(γ, t3) |= physical_position(R, A)]]]

P5(d): **First arriving ambulance interpretation**
∀A:AMBULANCE, t:TIME
first_arriving_ambulance(γ, t, A)
⇒ ∃X:SITUATION, t2:TIME < t + d & t2>t
state(γ, t2) |= physical_position(driver, A) &
state(γ, t2) |= communication_from_to(driver, operator, situation_description(X)) &
state(γ, t2) |= situtation(X)]

P6(d): **Paramedic investigation**
∀A:AMBULANCE, t:TIME
[first_arriving_ambulance(γ, t, A) &
state(γ, t) |= physical_position(paramedic, A)]]
⇒ ∃t2:TIME < t + d & t2 > t
[state(γ, t2) |= not physical_position(paramedic, A) & state(γ, t2) |= investigating(paramedic)]

P7(d): **Paramedic communication**
∀A:AMBULANCE, t,t2:TIME
[first_arriving_ambulance(γ, t, A) &
state(γ, t) |= physical_position(paramedic, A) & t2 > t &
state(γ, t2) |= not physical_position(paramedic, A) &
state(γ, t2) |= investigating(paramedic)]
⇒ ∃t3:TIME < t2 + d & t3 > t2, X:SITUATION
[state(γ, t3) |= physical_position(paramedic, A) &
state(γ, t3) |= communication_from_to(paramedic, driver, situation_description(X)) &
state(γ, t3) |= situtation(X)]

P8(d): **Driver communication**
∀A:AMBULANCE, t,t2:TIME, X :SITUATION
[first_arriving_ambulance(γ, t, A) &
state(γ, t2) |= communication_from_to(paramedic, driver, situation_description(X))
⇒ ∃t3:TIME < t2 + d & t2 > t [state(γ, t3) |= communication_from_to(driver, operator, situation_description(X))]

P9: **Successful disaster staff**
∀t:TIME
[state(γ, t) |= internal_judgement(operator, disaster)
⇒ ∃t2:TIME > t, S:STRATEGY
[state(γ, t2) |= communication_from_to(disaster_staff, operational_leader, S) &
∃t3:TIME > t2, A:ACTION, R:ROLE
[state(γ, t3) |= appropriate_action_according_to_plan(S, A) &

state(γ, t3) |= accompanying_role(A, R)] &
state(γ, t3) |= communication_from_to(operational_leader, R, perform(A))]

***P10:* Correctly activated disaster staff**

∀t:TIME, R:ROLE
[state(γ, t) |= internal_judgement(operator, disaster) &
state(γ, t) |= part_of(R, disaster_staff)
⇒ ∃t2:TIME > t + d [state(γ, t2) |= communication_from_to(operator, R, form_disaster_staff)]]

***P11:* Active disaster staff**

∀t:TIME
[state(γ, t2) |= part_of(R, disaster_staff) &
state(γ, t2) |= communication_from_to(operator, R, form_disaster_staff)
⇒ ∃S:STRATEGY, t2 > t [state(γ, t2) |= communication_from_to(disaster_staff, operational_leader, S)]

***P12:* Active operational leader**

∀t:TIME, S:STRATEGY, A:ACTION, R:ROLE
[state(γ, t2) |= communication_from_to(disaster_staff, operational_leader, S) &
state(γ, t) |= appropriate_action_according_to_plan (S, A) &
state(γ, t) |= accompanying_role(A, R)
⇒ ∃t2:TIME > t state(γ, t2) |= communication_from_to(operational_leader, R, perform(A))]

***P13(d):* Warn mayor**

∀t:TIME
[state(γ, t) |= internal_judgement(operator, disaster) &
⇒ ∃t2:TIME > t & t2 < t + d [state(γ, t2) |= communication_from_to(operator, mayor, disaster)]

***P14:* Form disaster staff**

∀t:TIME
[state(γ, t) |= internal_judgement(mayor, disaster) &
¬∃t'< t [state(γ, t') |= internal_judgement(mayor, disaster)] &
⇒ ∃t2 > t [state(γ, t2) |= communication_from_to(mayor, operator, form_disaster_staff)]

***P15(d):* Warn rest disaster staff**

∀t:TIME, R:ROLE
state(γ, t) |= communication_from_to(mayor, operator, form_disaster_staff) &
state(γ, t) |= part_of(R, disaster_staff)
⇒ ∃t2:TIME > t + d [state(γ, t2) |= communication_from_to(operator, R, form_disaster_staff)]]

***P16:* Start deliberation**

∀t:TIME
[state(γ, t) |= communication_from_to(mayor, operator, form_disaster)
⇒ ∃t2:TIME > t [state(γ, t2) |= communication_from_to(mayor, disaster_staff, request_advice)]]

***P17:* Choose strategy**

∀t:TIME
[state(γ, t) |= communication_from_to(mayor, disaster_staff, request_advice)
⇒ ∃S:STRATEGY, t2:TIME > t state(γ, t2) |= communication_from_to(mayor, operational_leader, S)]

***P18:* Choose action**

∀t:TIME, S:STRATEGY
[state(γ, t) |= communication_from_to(mayor, operational_leader, S)
⇒
∃ t2:TIME > t state(γ, t2) |= communication_from_to(operational_leader, operational_team, request_advice(S))]

***P19:* Communicate action**

∀t:TIME, S:STRATEGY, A:ACTION, R:ROLE
[state(γ, t) |= communication_from_to(operational_leader, operational_team, request_advice(S)) &
state(γ, t) |= appropriate_action_according_to_plan (S, A) &
state(γ, t) |= accompanying_role(A, R)
⇒ ∃ t2:TIME > t state(γ, t2) |= communication_from_to(operational_leader, R, perform(A))]

***P20:* Route plan includes all wounded nests**

∀W:WOUNDED_NEST, R:ROUTE_PLAN, t:TIME
[state(γ, t) |= physical_position(W, scene) &
state(γ, t) |= communication_from_to(police, cpa, R)]
⇒ state(γ, t) |= route_passes_wounded_nest(R, W)

***P21:* Send ambulance to all wounded on the scene**

∀W:WOUNDED, P:POSITION, A:AMBULANCE, t:TIME
[state(γ, t) |= physical_position(W, P) &
⇒ ∃t2 > t [state(γ, t2) |= communication_from_to(operator, A, goto(P))] &
∃t3 > t2 [state(γ, t3) |= physical_position(A, P)

***P22:* Communicate wounded location**

∀W:WOUNDED, P:POSITION, t:TIME
[state(γ, t) |= physical_position(W, P) &
⇒ ∃R:ROLE, t2 > t [state(γ, t2) |= communication_from_to(R, operator, physical_position(W, P))]]

***P23:* Send ambulance to wounded**

∀W:WOUNDED, P:POSITION, R:ROLE, t:TIME
[state(γ, t) |= communication_from_to(R, operator, physical_position(W, P))
⇒ ∃t2 > t, A:AMBULANCE [state(γ, t2) |= communication_from_to(operator, A, goto(P))]]

***P24:* Ambulance arrives at wounded**

∀P:POSITION, A:AMBULANCE, t:TIME
[state(γ, t2) |= communication_from_to(operator, A, goto(P))
⇒ ∃t2 > t [state(γ, t3) |= physical_position(A, P)

B.4 Type Error Definition (Section 7)

Type Error ≡

∀γ:TRACE, t_1, t_2:TIME, A:AGENT, R:ROLE, P:DYNPROP, Q:DYNPROPEXPR, S:SITUATION, X:PROPERTY_TYPE
[holds_in_period(has_role(A, R), γ, t_1, t_2) &
holds_in_period(S, γ, t_1, t_2) &
holds_in_period(relevant_for(P, R, S), γ, t_1, t_2) &
holds_in_period(type_for(P, R, X), γ, t_1, t_2) &
holds_in_period(has_specification(P, Q(R, γ, c_1, c_2)), γ, t_1, t_2) &
¬holds(Q(R, γ, t_1, t_2))]
⇒ makes_error_of_type(A, R, P, X, γ, t_1, t_2)