



# Automated verification of disaster plans in incident management

Mark Hoogendoorn

*Department of Artificial Intelligence, Vrije Universiteit Amsterdam,  
Amsterdam, The Netherlands*

Catholijn M. Jonker

*Nijmegen Institute of Cognition and Information,  
Radboud University Nijmegen, Nijmegen, The Netherlands, and*

Viara Popova and Alexei Sharpanskykh

*Department of Artificial Intelligence, Vrije Universiteit Amsterdam,  
Amsterdam, The Netherlands*

## Abstract

**Purpose** – The purpose of this paper is to create a formal specification language for disaster plans in order to remove possible inconsistencies between disaster plans, and to enable the automated verification of properties from such plans against logs of actual incidents.

**Design/methodology/approach** – Different types of properties in disaster plans have been identified and formalized using order-sorted predicate logic, enabling automated comparison of plans and verification of such properties against logs by means of software tools. Actual disaster plans and logs have been used as a case study to show the working of the approach.

**Findings** – The automated approach can be used quite easily and result in important findings. For the case study disaster plans crucial differences were found that could have catastrophic consequences. Furthermore, it is shown that in the logs of a well-known incident the disaster plan was not followed.

**Practical implications** – If the approach is introduced in practice, disaster plans would be stored in a formal format, enabling the automated comparison of disaster plans, and immediate detection of derivation from a disaster plan in case of an incident.

**Originality/value** – Other literature about the formal modelling of disaster plans that includes both structural and dynamical aspects and allows representation of organizational structure at multiple aggregation levels has not been found. Nor has comparing the disaster plans using such a formal model, and using the model of the disaster plan to check empirical traces for compliance with this plan, been addressed in prior literature.

**Keywords** Disasters, Specification languages, Automation, Plans, Safety measures

**Paper type** Research paper



## 1. Introduction

Disasters are unforeseen events that cause great damage, destruction and human suffering. The question that keeps rising is: “Could we have done anything to prevent this?” The key element is the distinction between incidents and disasters. Incidents are

---

This paper is a significantly extended version of Hoogendoorn *et al.* (2005).

---

disturbances in a system that can lead to an uncontrollable chain of events, a disaster, when not acted on properly.

Incidents will keep occurring. People can make mistakes and nature can be unpredictable. Typically this causes chaotic situations and the resulting problems are very complex and have to be solved within limited time. Examples of incidents that took on disastrous proportions because of inadequate human intervention are the crash of a Boeing 747 in an urban area in Amsterdam and the Hercules disaster in Eindhoven in the Netherlands.

In order to cope with such incidents, every municipality in The Netherlands has its own disaster plan. A disaster plan contains the blueprint of how to handle incidents with the aim of preventing incidents to grow into disasters. The plan describes the relations with all organizations that might possibly be involved, like the mayor, the fire department, police, ambulances, hospitals, other municipalities, Provincial Government, National Government. When comparing municipalities both commonalities and differences stand out. The commonalities encompass such basic elements as a local government, the availability of some kind of police force, fire department, and ambulance services. Small municipalities might not have their own forces of the kind mentioned, but have to share them with other municipalities. Big cities have subdivided their forces in smaller units that predominantly serve specific parts of the city. More fundamental differences involve the infrastructure of the municipality (e.g. forms of public transportation, the road plan, water ways, bridges), but also the enterprises and organizations available within the boundaries of the municipality like airports, factories, restaurants, stadiums and theatres.

Given that each municipality has its own organizations, enterprises, infrastructure, and general layout, it seems self-evident that the disaster plans also differ. On the other hand, the disaster plans form only a blueprint of handling incidents. For every entity in the municipality that carries a predictable risk a more detailed plan has to exist, a so-called disaster prevention plan. The advantage of separating disaster plans from disaster prevention plans is that the disaster plan is applicable in all situations and is a relatively compact document. This line of reasoning entails again that the disaster plans of different municipalities should have, and in fact, do have a lot in common. On the basis of the above, one might expect that disaster plans were developed from a common template. In general, they are not. Some municipalities use a common starting point; others develop their own disaster plan from scratch. It raises the question how comparable these disaster plans actually are.

Another question is to what extent disaster plans are followed when incidents occur. The identification of differences between the occurrences during an incident and the specification of the disaster plan is of particular importance for the detailed analysis of incidents and, as a result, improvement of incident management (e.g. by performing dedicated training sessions, and possibly by making necessary corrections in disaster plans). The data about the actual events and actions occurring during the incident management process are often available in the form of informal logs. Since the manual analysis of such logs is a time-consuming and error-prone process, tools for the automated analysis would be of use.

This paper presents an approach to support incident management based on disaster plans. The contribution of the approach is three-fold: First of all, the paper presents a method to formally describe disaster plans. Using this formal description, disaster

---

plans of different municipalities can be checked for consistency, to avoid problem that could arise once these municipalities have to combine their forces to manage an incident. Furthermore, the formal description of the disaster plans allows for the automated verification of such disaster plans against the empirical data that describe incident management processes occurred in reality.

To illustrate the proposed approach two disaster plans of municipalities of Eindhoven (Gemeente Eindhoven, 1993) and Uithoorn (see Gemeente Uithoorn, 2003) have been used as a case study for this paper. Eindhoven is a relatively large city in the Netherlands with approximately 200,000 residents. A large-scale aviation accident occurred at the airport in 1996 of which logs have been obtained (Inspectie Brandweerzorg en Rampenbestrijding, 1996). Uithoorn is a much smaller town than Eindhoven. However, Uithoorn belongs to a group of municipalities including Amsterdam and 6 surrounding municipalities that base their disaster plans on a common template.

The paper is organized as follows. In section 2 the formal specification method for disaster plans is presented, whereas section 3 shows how such formal description of different disaster plans can be compared. Thereafter, section 4 addresses the verification of formal properties obtained from disaster plans against logs. Finally, section 5 is a discussion.

## 2. Formal specification of disaster plans

This section provides some general guidelines for extracting a formal model of the disaster plan from a textual disaster/incident plan and thus bridging the gap between informal and formal representation. In principle, any modelling approach for organizations and any formal language for modelling organizations can be used as a point of departure. For example, a formal language based on description logic for specifying disaster management is introduced in (Grathwohl *et al.*, 1999). In this paper, the modelling approach of (van den Broek *et al.*, 2005) and (Hoogendoorn *et al.*, 2004) based on an order-sorted predicate logic (Manzano, 1996) is used for formal modelling of the structure of an incident management organization. Based on the formal structural description from a disaster plan, different scenarios of organizational behaviour can be specified and analyzed, using for example the Temporal Trace Language (Jonker and Treur, 2002).

The formal description of the incident management organization (identified by a name of sort ORGANIZATION) is associated with the disaster plan in which it is specified by the following predicate:

is\_based\_on: ORGANIZATION x DISASTER\_PLAN.

Based on experience in modelling disaster plans the following stages are advocated: phase identification, structure analysis and modelling, task and responsibility analysis, organizational change modelling. Each of these stages is explained in more detail. The comparison of disaster plans is discussed after the modelling steps.

### 2.1 Phase identification

In each disaster plan a number of phases of incident management are identified. Typically, they are grouped in three general phases depending on the severity of the situation:

- (1) *Small incident* – no co-ordination between police, fire department and medical forces is needed, the highest level of decision-making and co-ordination only involves functionaries of these three institutions.
- (2) *Serious incident* – involvement of the mayor is needed at the highest level of decision-making. Typically a disaster management team is formed at the city hall.
- (3) *Severe incident involving more than one municipality* – co-ordination between the municipalities is needed. Typically, the National Coordination Centre is also involved.

The first step in this modelling approach is to identify which particular phases are covered by the disaster plan. The Eindhoven disaster plan identifies five phases:

- (1) local incident;
- (2) local calamity or disaster;
- (3) local incident, calamity, or disaster with use of regional coordination;
- (4) inter-local incident; and
- (5) inter-local calamity or disaster.

With each phase an organization structure (denoted by its name of sort ORGANIZATION) is associated. For this the following predicate is used:

`is_organization_in_phase`: ORGANIZATION x PHASE is introduced.

### 2.2 Structure analysis and modelling

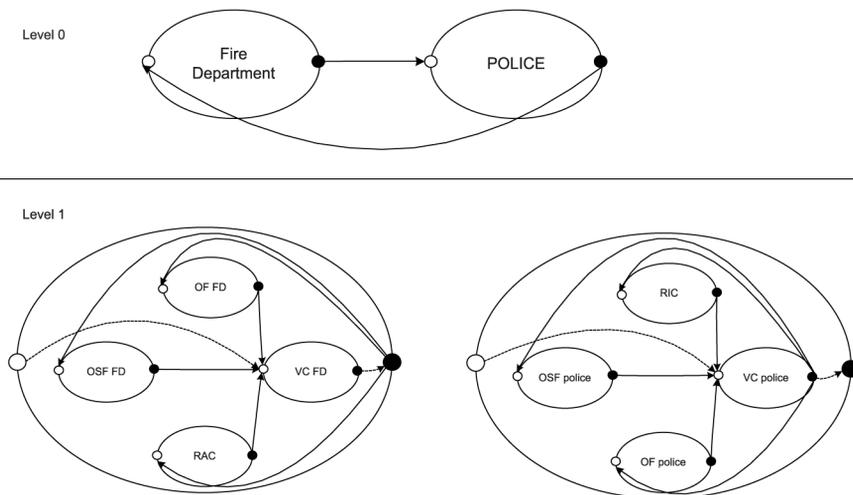
Each phase of incident management has its own organizational structure. Therefore, the structure of the organization has to be analysed and modelled for each phase. Structure analysis aims at identifying all parties involved and their relevant organizational roles and relationships:

- Disaster plans typically contain lists of all parties involved. Institutions like the fire department, ambulance services, police, municipal service and other associated institutions are almost always involved. These institutions exist irrespective of whether an incident occurs or not. However, disaster plans also refer to parties like the operation team, regional coordination centre, and management team, depending on the phase and scope of a disaster/incident and only exist during these phases. The structure can consist of roles that contain other roles and so forth.
- After identifying the roles in the organization at a certain phase, the communications between roles or composite roles need to be identified. For example, a policy team always maintains communication with fire department action centre. With respect to communication and interaction the disaster plans studied by the authors are typically incomplete, making it difficult and in some cases impossible to identify the exact links in the structural model.

The structure of an incident management organization can be described at different aggregation levels, which allows managing the level of complexity and refinement of an organization representation. The aggregation levels refer to a level of the

organization consisting of roles and the interaction between those roles. A model of an organization with several aggregation levels also contains a specification of the inter-level relations of those aggregation levels. Therefore, a model of an organizational structure consists of roles, interaction links, interlevel links, and structural properties regarding those elements:

- (1) A role represents a subset of functionalities, performed by an organization, abstracted from instances of real agents. At the highest aggregation level, the whole organization can be represented as one role. Further, each role can be decomposed into several other roles, until the necessary level of aggregation is achieved. Graphically, a role is represented as an ellipse with white (input interfaces) and black (output interfaces) dots (see Figure 1). A role that is composed of (interacting) sub-roles, is called a composite role. Each role has an input and an output interface, which facilitate in the communication with other roles. Although, in this paper, the emphasis is on the organization structure of incident management, an organization is realized by the agents (or sets of agents) fulfilling the roles.
- (2) An interaction link represents an information channel between two roles. Graphically, it is depicted as a solid arrow, which denotes the direction of possible information transfer. For example, interaction links between roles Fire Department and Police in Figure 1 represent the possibility of communication between them.
- (3) An inter-level link connects a composite role with one of its sub-roles. It relates two adjacent aggregation levels. Graphically, it is depicted as a dashed arrow, which shows the direction of inter-level transition (see Figure 1).
- (4) Structural properties specify the number of instances of a specified role and the various role- sub-role relations. Although the structure of an organization can be specified partly using graphs (see Figure 1), a formal textual language is needed to specify the structural properties. Sorts are introduced for the basic



**Figure 1.**  
Example of an  
organisation structure,  
described at two adjacent  
aggregation levels

---

elements of an organization and relations between them (i.e. ROLE, AGENT, ORGANIZATION, INTERLEVEL\_LINK, and INTERLEVEL\_LINK). Furthermore, a set of relations is defined to specify the structural aspects of the organization. A complete overview is given in (van den Broek *et al.*, 2005), here only a few examples are given:

- `is_role_in`: ROLE x ORGANIZATION identifies a role in an organization; and
  - `has_sub-role_in`: ROLE x ROLE x ORGANIZATION defines a sub-role of a composite role in an organization.
- (5) Examples of structural properties are: `is_role_in(FD,ORG1)`, `has_sub-role_in(FD,VC_FD,ORG1)`.

Often, structural properties are valid during the whole period of organization existence and can be considered as static. But in rapidly developing and adapting organizations (e.g. incident management organizations) structural change processes gain special importance. Structural properties for such organizations will be described later.

For each of the phases identified in the previous step, the structure of the organization has been identified; only the second phase is presented in this paper, see Figure 2.

The abbreviations used in the Figure are the following: OSF stands for On Scene Forces, Off Scene Forces are abbreviated to OF and GGD is an abbreviation for the Medical Services. Finally, CoRT stands for Command Disaster Area. Inter-level connections between composite roles and their sub-roles are often omitted because the disaster plan does not specify any of these relationships. A partial specification of this Figure in the formal language as presented is shown in Figure 3.

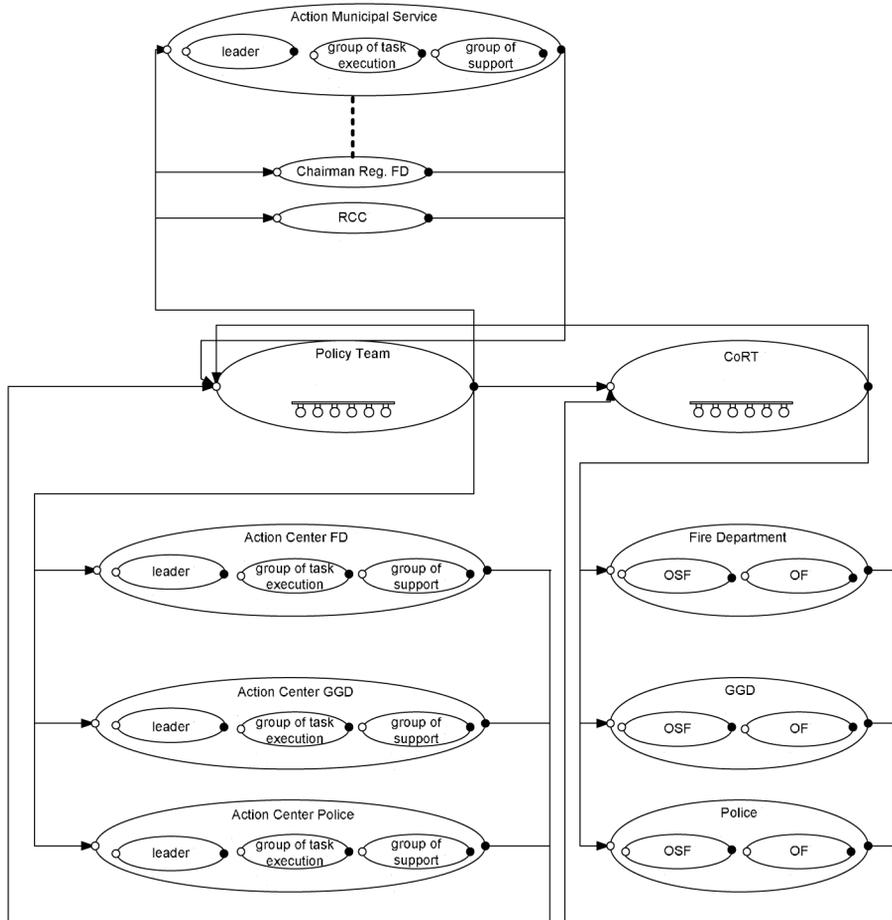
### 2.3 Tasks and responsibilities analysis

Having identified the organizational structure in the different phases of incident management, the tasks and responsibilities of the roles have to be determined. Problems at this stage might be vague and unclear formulations of the tasks, no detailed information for the responsibilities per task and per role.

The dynamics of an organization are formed by the execution of tasks by the organization and the change of an organization. To analyze and model the first of these, the tasks and responsibilities of the different structural elements of the organizational model have to be identified. An ontology based on the order-sorted predicate language is introduced that provides a way to express statements describing the hierarchy of tasks, responsibilities of roles for certain tasks in a particular situation and leadership within a composite role. The introduced ontology is useful for any organization that encounters change on a regular basis.

The main sorts are TASK, PHASE, ROLE, and ORGANIZATION. Using these sorts, the language can be extended with a set of relations to specify tasks, responsibilities and the phases of an organization:

- (1) *Primary co-ordination of task* – which role co-ordinates the execution of the task.
- (2) *Secondary co-ordination of a task* – in some situations the primary coordinating role can be replaced by the secondary coordinating role. That might happen for example when the particular type of disaster has specifics that can more appropriately be handled by the secondary coordinating role.



**Figure 2.**  
Structure of the  
Eindhoven disaster  
prevention organisation in  
the local incident phase

- (3) *Primary execution of a task* – the role(s) that execute the task.
- (4) *Secondary execution of a task* – for particular disasters where the emphasis is shifted towards an institution (role) not involved in the primary execution of the task, this institution can also become involved in it.
- (5) *Operational leadership within a complex role* – the role that takes the leadership of the complex role (group, institution, etc.)

To specify such information the following relations re-introduced:

- *is\_subtask\_of\_in*: TASK x TASK x ORGANIZATION, to describe the task-subtask ordering in the organization;
- *executes\_task\_primary\_in*: ROLE x TASK x ORGANIZATION, describes which role is the principle performer of a task in the organization;



- `executes_task_secondary_in`: ROLE x TASK x ORGANIZATION, describes which role is the secondary performer of a task in the organization;
- `coordinates_task_primary_in`: ROLE x TASK x ORGANIZATION, describes which role is the principle coordinator of a task in the organization;
- `coordinates_task_secondary_in`: ROLE x TASK x ORGANIZATION, describes which role is the secondary coordinator of a task in the organization; and
- `operational_leadership_in`: ROLE x ROLE x ORGANIZATION, describes which role is the leader in a part of the organization.

From the analysis of the disaster plans considered so far a certain level of similarity in the task and process hierarchy has been discovered. This indicates that it is possible and beneficial to build a general ontology of tasks in disaster situations. A partial one was built on the information available from these two disaster plans and it is considered to analyze more in order to adjust and refine the ontology.

Some examples of structural relations from the Eindhoven disaster plan are: the fire department is in charge of the task of fighting the fire, the police is responsible for evacuating the people, and the medical services are responsible for collecting contaminated goods. These examples are formally represented in Figure 3.

#### *2.4 Organizational change modelling*

Knowing the organizational structures during the different phases of incident management is not enough to model a disaster plan. The last but vital part of the modelling is the specification of organizational change. This entails the identification of all conditions of organizational change. They normally depend on the different incidents/disasters. Typical problems that occur during this phase are lack of information concerning the triggers that cause change. Often the decision to change the organization is left to a deliberation group without stating specific definitions of the triggers.

The modelling process delivers a lot of information concerning how thoroughly a disaster plan is specified. In case some unclear parts are identified, the disaster plan can be improved in a number of ways, e.g. using experts and/or training. Another option is to organize a training dedicated to an unclear part.

The disaster plan of Eindhoven is vague about organizational change: it is left to the mayor and its advisors to decide on the appropriate phase. However, the triggers can be derived by comparing the definitions of each of the phases. For example, going from phase 1 (a local incident) to phase 2 (a local disaster) means that the public is actually seriously threatened. The change of organization involves the following elements: An operational team is added to the organization that is responsible for the action centres of the regional emergency services. Furthermore, some of the communication lines are changed.

To formally specify changes to be performed within an organization the language shown is used. The language takes as a basis the structural language as introduced before and the responsibilities and tasks language as defined in the previous section. Sorts used to represent these elements are `STRUCT_ELEMENT` and `RESPONS_TASK_ELEMENT`. The sorts are combined into the sort `ORG_ELEMENT`. Functions are defined for adding, deleting and modifying an organization element (which can also be seen as a combination of add and delete):

- *add*:  $\text{ORG\_ELEMENT} \rightarrow \text{ORG\_CHANGE\_ELEMENT}$ , describes an organizational element being added.
- *delete*:  $\text{ORG\_ELEMENT} \rightarrow \text{ORG\_CHANGE\_ELEMENT}$ , describes an organizational element being deleted.
- *modify*:  $\text{ORG\_ELEMENT} \times \text{ORG\_ELEMENT} \rightarrow \text{ORG\_CHANGE\_ELEMENT}$ , describes that the first organization element is modified to the second argument.

Besides the need to specify what needs to be changed also the triggers that cause the change need to be formally specified. For this the following predicate is introduced:

- *is\_trigger\_for\_from\_to*:  $\text{TRIGGER} \times \text{ORG\_CHANGE\_ELEMENT} \times \text{PHASE} \times \text{PHASE}$ , describes that when a trigger occurs the phase is changed (if necessary) from the present phase to some other phase, and the organization is changed according to the specification defined in  $\text{ORG\_CHANGE\_ELEMENT}$ .

Examples of the use of this ontology are shown in Section 2.5.

### 2.5 Example formal description

Figure 3 shows a part of the formal specification of the disaster plan of Eindhoven, covering each of the aspects as addressed in this section.

## 3. Comparing of disaster plans

A comparison of disaster plans consists of the following elements: comparison of phases, comparison of organizational structures in comparable phases, comparison of the task structure in comparable phases, and comparison of the responsibilities scheme in comparable phases. The comparison of phases is a rather straightforward matter. Comparison of the organizational structures entails the identification of comparable and incomparable structures within the organization at each of the phases of incident management, and a comparison of the ontologies used. The comparison of task structures concentrates on the tasks identified in each disaster plan, and discusses comparable and incomparable tasks. Given the comparable tasks, the comparison of responsibilities entails the allocation of responsibilities to roles. This section presents the results of the comparison of the two disaster plans as introduced before.

For the purpose of comparison of the disaster plans described above a number of relevant properties have been identified. These properties constitute two groups:

- (1) local municipality properties; and
- (2) regional coordination properties.

The first group describes properties that do not influence the incident management organization of other (neighbouring) municipalities and can therefore differ between these municipalities. Properties in the second group do influence the incident management organization of other municipalities. In case of an inter-local incident these kinds of properties have to be the same to enable a proper functioning of the disaster management organization.

Consider an example of local municipality properties:

- Property 1.  
Informal form

DPM  
17,1

The command centre of surroundings of the incident area (ComRT) is a part of the incident management organization of municipality X in phase 4.

Formal form

$[is\_role\_in(ComRT,ORG4) \wedge$   
 $is\_based\_on(ORG4,X) \wedge$   
 $is\_organization\_in\_phase(ORG4,PHASE4)]$

**26**

This property holds for X = Uithoorn and does not hold for X = Eindhoven.

Consider two examples of regional coordination properties:

Property 2.

Informal form

The mayor of the biggest municipality coordinates the work of the Managing Platform Centre (MPC) in the incident management organization of municipality X in phase 4.

Formal form

$[coordinates\_task\_primary\_in(biggest\_municipality\_mayor, regional\_collaboration\_in\_MPC,$   
 $ORG4) \wedge is\_based\_on(ORG4, X) \wedge is\_organization\_in\_phase(ORG4,PHASE4)].$

This property holds for X = Uithoorn and does not hold for X = Eindhoven:

Property 3.

Informal form

The mayor of the municipality that was the first involved in an incident, coordinates the work of the Managing Platform Centre (MPC) in the incident management organization of municipality X in phase 4.

Formal form

$[coordinates\_task\_primary\_in(mayor\_involved\_first, regional\_collaboration\_in\_MPC,$   
 $ORG4) \wedge$   
 $is\_based\_on(ORG4,X) \wedge is\_organization\_in\_phase(ORG4,PHASE4)]$

This property holds for X = Eindhoven and does not hold for X = Uithoorn.

The formal approach in the comparison of disaster plans allows us to go further and analyze these differences and investigate whether they indeed lead to serious consequences. An example of such analysis is given in the following paragraphs. It is already known (see property 1) that the role ComRT is present in the disaster plan of Uithoorn but not in that of Eindhoven. This role represents the team responsible for activities in the surroundings of the disaster area including traffic regulation, isolation of the area, etc. In both plans the team CoRT is present which co-ordinates the onscene operations. Is this difference fundamental? Maybe the tasks of ComRT for the case of Uithoorn are actually assigned to CoRT in the case of Eindhoven. This hypothesis is expressed in property 4, and decomposed into properties 5 through 8 to ease the formal proof process, as depicted in Figure 4. The formal relations are:

- Property 5  $\wedge$  Property 6 | = Property 4
- Property 7  $\wedge$  Property 8 | = Property 6

Property 4.

Informal form

The set of tasks assigned to CoRT in the disaster plan of Eindhoven is the same as the set of tasks assigned to CoRT or ComRT in the disaster plan of Uithoorn.

Property 5.

Informal form

All tasks of CoRT in the disaster plan of Eindhoven are either tasks of CoRT or of ComRT in the disaster plan of Uithoorn:

Formal form

“ T:TASK “O:ORGANIZATION:

coordinates\_task\_primary\_in(CoRT,T, O)^

is\_based\_on(O,'Eindhoven')

$\Rightarrow \exists' :ORGANIZATION$  [coordinates\_task\_primary\_in(CoRT,T, O) $\vee$

coordinates\_task\_primary\_in(ComRT,T, O)] ^ is\_based\_on(O,'Uithoorn')

Property 6.

Informal form

All tasks of CoRT or ComRT in the disaster plan of Uithoorn are also tasks of CoRT in the disaster plan of Eindhoven.

Property 7.

All tasks of CoRT in the disaster plan of Uithoorn are also tasks of CoRT in the disaster plan of Eindhoven.

$\forall T:TASK$  “O:ORGANIZATION:

coordinates\_task\_primary\_in(CoRT,T,O)^

is\_based\_on(O,'Uithoorn')

$\Rightarrow \exists O' :ORGANIZATION$  coordinates\_task\_primary\_in(CoRT,T,O') ^ is\_based\_on(O', 'Eindhoven')

Property 8.

All tasks of ComRT in the disaster plan of Uithoorn are also tasks of CoRT in the disaster plan of Eindhoven.

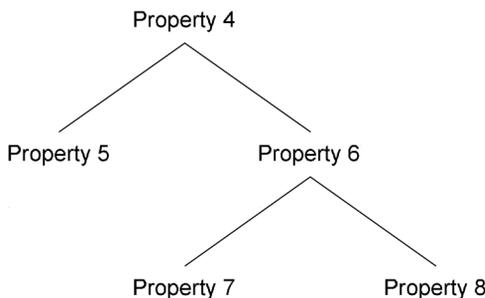
$\forall T:TASK \forall O:ORGANIZATION:$

coordinates\_task\_primary\_in(ComRT,T,O) ^ is\_based\_on(O,'Uithoorn')

$\Rightarrow \exists O' :ORGANIZATION$  coordinates\_task\_primary\_in(CoRT,T,O') ^ is\_based\_on(O', 'Eindhoven')

By checking properties 5, 7 and 8, it is discovered that the functions of CoRT in the case of Eindhoven and CoRT and ComRT in the case of Uithoorn indeed overlap. Therefore, while the absence of ComRT is certainly a difference between the two disaster plans, in reality the difference is smaller than expected at first sight.

The comparison between the disaster plans of Uithoorn and Eindhoven revealed two differences in the regional coordination. The first concerns leadership: which mayor is in charge of the disaster management organization in case of an inter-local incident.



**Figure 4.**  
The decomposition of  
property 4 represented in  
an and-tree

---

The Uithoorn plan states that the mayor of the biggest municipality is the leader. The Eindhoven plan states that the mayor of the municipality where the incident started is in charge. Imagine that these are neighbouring municipalities and that an incident that affects both municipalities is first discovered in Uithoorn, which is the smallest municipality of the two. According to the Eindhoven disaster plan Uithoorn remains in charge, and therefore does not take any initiative in forming an inter-local incident management organization. Uithoorn however, thinks Eindhoven will take the initiative as it is the biggest municipality involved in the incident. To prevent this kind of errors, such differences should be avoided. The second regional coordination difference concerns the incident phases described in the disaster plans. There does not exist a one-to-one mapping between these phases, therefore the municipality that has the lead in the incident management organization might declare a certain phase that cannot be interpreted by the other municipalities involved. For example, in the Uithoorn disaster plan, a phase is present where there is multidisciplinary coordination without the mayor being involved. In the Eindhoven disaster plan there doesn't exist any phase including multi-disciplinary coordination in which the mayor is not involved in the disaster prevention organization.

Differences in local municipality properties were also observed in the comparison of the disaster plans. These differences include elements such as splitting up the command of the disaster area in the disaster plan of Uithoorn, while this remains one group in the Eindhoven disaster plan. These differences can, however, be formally mapped to each other, and are therefore not as crucial.

#### **4. Verification of disaster plan properties against logs**

In order to determine to which extent disaster plans are followed in reality, when incidents occur, an automated verification method is proposed. By means of this method, the formal specification of a disaster plan is checked automatically on formalized empirical data concerning an incident. This empirical data are usually represented in the form of informal logs (also called traces) that contain events. Such informal logs can be formalized using the formal language TTL (Jonker and Treur, 2002). The translation from a log of events to a formal trace is currently done by hand. However, for the future there are plans to develop a methodology that supports non-expert users in making this translation. After such a formalization of a log has been created, the formal properties extracted from the disaster plan can be automatically verified against the formalized trace. This section first of all shows what such a formalized trace looks like, and thereafter presents results of checking the properties obtained from the Eindhoven disaster plan to the logs of the Hercules airplane crash in 1996.

##### *4.1 Formalizing an empirical trace*

An example of a formalization of a trace is shown in Figure 5. It shows the most relevant parts of the occurrences during the Hercules incident. The ontology used in the trace is identical to the one introduced in section 2 on formally describing a disaster plan. In the left side of the figure, the relevant so-called atoms in the trace are shown whereas the right part represents a time line. In the time line a black box indicates that the atom is true whereas a grey box indicates that it is false.

As can be seen in the trace, from time point 0 to 10 the phase declared is phase 2 whereas between 10 and 30 phase 3 holds. Furthermore, at time point 9 a trigger is observed for changing the organization, namely that the current situation has been

declared a disaster. The partial structure of the organization at different time points is shown in the figure as well. During the entire incident, the OSC (for On Scene Commander) role is part of the organization and of the On Scene Forces group. Furthermore, the OSC is never part of the Command Disaster Area (abbreviated to CoRT in the trace). Finally, the operational team role is added to the organization from time point 10 and on.

#### 4.2 Verification of properties against a formalized trace

After having obtained a formalized trace, properties extracted from the disaster plans can be verified against such a trace. By means of this verification one can determine what part in the example incident management process described by the trace did not follow the disaster plan.

For such verification, based on the formal representation of the disaster plan, a set of facts is defined in the form `follows_from_disaster_plan(X)`, where `X` is a relation from the formalized disaster plan. Then, based on the identified facts dynamic properties are specified that can be verified on the formalized empirical trace by means of the dedicated software environment TTL Checker. To enable automated verification, dynamic properties should be expressed by formulae in the Temporal Trace Language. The software environment takes a TTL formula and one or more traces as input, and checks whether the formula holds for the trace(s).

Below, a number of dynamic properties in the form of TTL formulae are considered, based on the disaster plan for the Eindhoven municipality. These properties have been checked automatically on the formalized empirical trace, a part of which is depicted in Figure 5.

First of all, it is checked whether the organizational structure in the different phases indeed corresponds to the disaster plan. Note that this property only concerns the sub-role relationship, similar properties can be specified for the other structural relationships:

Property 9.

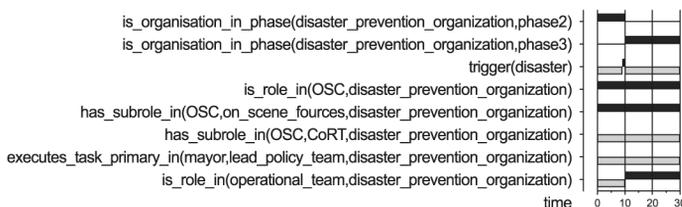
Informal form

For all time points  $t$  in trace  $\gamma$ , if the phase at time point  $t$  is  $P$ , and the disaster plan specifies that a particular role  $R2$  should have a sub-role  $R1$  in organization  $O$  in phase  $P$ , then role  $R1$  is indeed a sub-role of role  $R2$  in organization  $O$  at time  $t$ .

Formal form

$\forall t:\text{TIME}, \forall R1,R2:\text{ROLE}, \forall P:\text{PHASE}, \forall O, O':\text{ORGANIZATION}:$

$[[\text{state}(\gamma, t) \mid = \text{is\_organization\_in\_phase}(O, P) \ \&$   
 $\text{follows\_from\_disaster\_plan}(\text{has\_sub-role\_in}(R1, R2, O')) \ \&$   
 $\text{follows\_from\_disaster\_plan}(\text{is\_organization\_in\_phase}(O', P))]]$   
 $\Rightarrow \text{state}(\gamma, t) \mid = \text{has\_sub-role\_in}(R1, R2, O)]$



**Figure 5.**  
Partial empirical trace of  
the Eindhoven plan crash

The relation  $\text{state}(\gamma, t) \models p$  denotes that within the state  $\text{state}(\gamma, t)$  at time point  $t$  in trace  $g$  the state property  $p$  holds. This property is not satisfied in the given trace, because the OSC role should be part of the CoRT role in both phase 1 and 2 according to the disaster plan, whereas it is not in the trace.

A second property concerns the checking whether the tasks and responsibilities mentioned in the disaster plan are indeed performed. Again, this property just shows an example of how to check one relationship for the tasks, the rest of the relationships can be checked in a similar fashion:

Property 10.

Informal form

For all time points  $t$  in trace  $\gamma$ , if the phase at time point  $t$  is  $P$ , and the disaster plan specifies that a particular role  $R$  should be the primary executor of a task  $T$  in phase  $P$ , then role  $R$  is indeed the primary executor of this task  $T$  at time  $t$ .

Formal form

$\forall t:\text{TIME}, R:\text{ROLE}, \forall P:\text{PHASE}, \forall T:\text{TASK} \forall O, O':\text{ORGANIZATION}:$   
 $[[\text{state}(\gamma, t) \models \text{is\_organization\_in\_phase}(O, P) \ \&$   
 $\text{follows\_from\_disaster\_plan}(\text{executes\_task\_primary\_in}(R, T, O')) \ \&$   
 $\text{follows\_from\_disaster\_plan}(\text{is\_organization\_in\_phase}(O', P))]]$   
 $\Rightarrow \text{state}(\gamma, t) \models \text{executes\_task\_primary\_in}(R, T, O)]$

This property is again not satisfied, as the mayor role should be the primary executor of the task to lead the policy team, whereas he does not perform that task.

A final property which has been checked against the trace is to investigate whether the organizational change processes in the organization have been successful, as shown in property 11:

Property 11.

Informal form

For all time points  $t$  in trace  $\gamma$ , if the phase at time point  $t$  is  $P$  and a trigger  $T$  holds, and furthermore the disaster plan specifies that in phase  $P$  given trigger  $T$  a new phase  $P2$  should hold, and roles should be added, then at a later point in time  $t2$  phase  $P2$  will be the case, and the organizational element will have been added.

Formal form

$\forall t:\text{TIME}, \forall OL:\text{ORG\_ELEMENT}, \forall P1, P2:\text{PHASE}, \forall T:\text{TRIGGER} \text{ "O, O':ORGANIZATION}:$   
 $[[\text{state}(\gamma, t) \models \text{is\_organization\_in\_phase}(O, P1) \ \&$   
 $\text{state}(\gamma, t) \models \text{trigger}(T) \ \&$   
 $\text{follows\_from\_disaster\_plan}(\text{is\_organization\_in\_phase}(O', P1)) \ \&$   
 $\text{follows\_from\_disaster\_plan}(\text{is\_trigger\_for\_from\_to}(T, \text{add}(OL:\text{ORG\_ELEMENT}), P1, P2))]]$   
 $\Rightarrow \exists t' t' > t [\text{state}(\gamma, t') \models \text{ORG\_ELEMENT} \ \& \ \text{state}(\gamma, t') \models \text{is\_organization\_in\_phase}(O, P2)]]$

This property is satisfied in the trace. The phase transitions do go according to the disaster plan. The initial organization however is, as has already been stated, not correct. Since the change is only concerned with transitions between phases, this property does hold.

## 5. Discussion

In this paper a formal framework for modelling and comparing disaster plans and checking disaster plans on empirical traces is presented and applied to a number of case studies. The framework extends earlier work of (Hoogendoorn *et al.* (2004) and

---

(van den Broek *et al.* (2005)), with specific constructs and reusable patterns for the domain of incident management, in specific for disaster plans. The approach uses formal graphical, and textual languages, in casu sorted first-order predicate logic and TTL (see (Jonker and Treur, 2002)). More specifically, sorted first-order predicate logic is used for formalizing structural properties in disaster plans and TTL is used for expressing causal temporal properties for the automated verification on formalized empirical traces by means of the dedicated software.

When compared with the work of (Grathwohl *et al.*, 1999) the framework presented in this paper is more generic from several perspectives. The first advantage is that the framework allows modelling on different levels of abstraction, and is, therefore, capable of modelling the Dutch disaster plans, which are on a highly abstract level of abstraction when compared to the plans that Grathwohl *et al.* (1999) modelled. The second advantage is that simulation of the models in different situations is possible. The third advantage is the software support for checking the model against simulation and transcribed real traces.

Narzisi *et al.* (2006) introduce an approach for the verification of properties against simulation traces of an agent-based system which models human behaviour in incidents. They do however not address using empirical logs from the incident management field within their work. Furthermore, the paper work does not concern the formal specification of disaster plans and automated verification of the properties described in such plans, which is one of the main contributions of this paper.

With respect to incident management this work contributed by proposing a formal approach for the modelling and comparison of disaster plans. The approach is explained in detail and tested in two case studies. The main results are the classification of differences into local differences and inter-local differences. The local differences effect only incident management in the municipality itself. The local differences can be fundamental or not when comparing the actual incident management. For example, two disaster plans differed in having only one or two zones around the epicentre of the incident. This difference has clear effects on the organizational structure prescribed in the disaster plans. However, the tasks associated with the zones are comparable. The same holds for the associated responsibilities. In other words, the organizational structure differs, but the dynamics are comparable. The inter-local differences are counterproductive when municipalities have to cooperate in case inter-local incidents. Comparing two disaster plans in this manner revealed a possible conflict regarding leadership. The consequence is clear: all neighbouring municipalities should use the same rules for determination of leadership. Therefore, all municipalities in The Netherlands should share those rules.

In the future, systems such as the IMI system (Lee and Vught, 2004) will contain many disaster plans. Making sure that these disaster plans are consistent with each other is of crucial importance for inter-local incident management. The plans in the system can be formalized, and verifying whether a new plan is consistent with the plans currently in the database would simply entail formalizing that plan and performing verification. In case the plan is indeed consistent the plan can be added to the database, including the formal description. On the long run an entirely different approach can be followed. Instead of taking an informal disaster plan as a point of departure, in future disaster plans should be first and foremost formal plans, from which an informal plan that is readable for human beings is automatically generated.

---

## References

- Gemeente Eindhoven (1993), *Rampenplan*, Gemeente Eindhoven, Eindhoven, May.
- Gemeente Uithoorn (2003), *Rampenplan*, Gemeente Uithoorn, Uithoorn, October.
- Grathwohl, M., de Bertrand de Beuvron, F. and Rousselot, F. (1999), "A new application for description logics: disaster management", *Proceedings of the International Workshop on Description Logics '99, Linköping, Sweden, 1999*.
- Hoogendoorn, M., Jonker, C.M., Schut, M. and Treur, J. (2004), "Modelling the organisation of organisational change", *Proceedings of the 6th International Workshop on Agent-Oriented Information Systems, AOIS'04, Riga*.
- Hoogendoorn, M., Jonker, C.M., Popova, V., Sharpaskykh, A. and Xu, L. (2005), "Formal modelling and comparing of disaster plans", in Carlé, B. and van de Walle, B. (Eds), *Proceedings of the 2nd International Conference on Information Systems for Crisis Response and Management ISCRAM '05*, pp. 97-107.
- Inspectie Brandweerzorg en Rampenbestrijding (1996), *Vliegtuigongeval Vliegbasis Eindhoven, 15 juli 1996*, SDU Grafische Bedrijf, The Hague, 1996.
- Jonker, C.M. and Treur, J. (2002), "Compositional verification of multi-agent systems: a formal analysis of pro-activeness and reactiveness", *International Journal of Cooperative Information Systems*, Vol. 11, pp. 51-92.
- Manzano, M. (1996), *Extensions of First Order Logic*, Cambridge University Press, Cambridge.
- Narzisi, G., Mysore, V., Nelson, L., Rekow, D., Triola, M., Halcomb, L., Portelli, I. and Mishra, B. (2006), "Complexities, catastrophes and cities: unraveling emergency dynamics", *Proceedings of the International Conference on Complex Systems (ICCS 2006), Boston, MA, USA, June 25-30, 2006*.
- van den Broek, E.L., Jonker, C.M., Sharpanskykh, A., Treur, J. and Yolum, P. (2005), "Modeling and analyzing multi-agent organizations", paper presented at the 4th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS'05, Utrecht University, Utrecht, 25-29 July.
- van der Lee, M.D.E. and van Vugt, M. (2004), in Carlé, B. and van der Walle, B. (Eds), "IMI – an information system for effective multidisciplinary incident management", *Proceedings of the International Workshop on Information Systems for Crisis Response and Management '04, Brussels, Belgium, 2004*.

## Further reading

- Abbink, H., van Dijk, R., Dobos, T., Hoogendoorn, M., Jonker, C.M., Konur, S., van Maanen, P.P., Popova, V., Sharpanskykh, A., van Tooren, P., Treur, J., Valk, J., Xu, L. and Yolum, P. (2004), "Automated support for adaptive incident management". In *Proceedings of the First International Workshop on Information Systems for Crisis Response and Management, ISCRAM'04. Brussels, 2004*.
- Breuer, K. and Satish, U. (2003), "Emergency management simulations: an approach to the assessment of decision making processes in complex dynamic environments", in Gonzalez, J.J. (Ed.), *From Modeling to Managing Security: A System Dynamics Approach*, oyskoleForlaget, Kristiansand, pp. 145-56.

## Corresponding author

Mark Hoogendoorn can be contacted at: [mhoogen@cs.vu.nl](mailto:mhoogen@cs.vu.nl)