

AUTOMATED SUPPORT FOR ADAPTIVE INCIDENT MANAGEMENT

Hans Abbink¹, Roel van Dijk¹, Tamas Dobos¹, Mark Hoogendoorn², Catholijn Jonker², Savas Konur², Peter-Paul van Maanen², Viara Popova², Alexei Sharpanskykh², Peet van Tooren¹, Jan Treur², Jeroen Valk¹, Lai Xu², Pinar Yolum²

¹ *Almende, Westerstraat 50, 3016 DJ Rotterdam, The Netherlands*

Email: {hans, roel, tamasd, peet, jeroen}@almende.com URL: http://www.almende.com

² *Department of Artificial Intelligence, Vrije Universiteit Amsterdam*

De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands.

Email: {mhoogen, jonker, skonur, pp, popova, sharp, treur, xu, pyolum}@few.vu.nl URL: http://www.few.vu.nl

Keywords: Cybernetic, Incident management, Adaptive, Multi-agent

Abstract: The project CIM, started in 2003, addresses the problem of automated support for incident management. In this paper some intermediate results are shown, especially on automated support of analysis of errors in traces of incident management. For such traces it can be checked automatically which dynamic properties hold or fail. The potential of the approach is shown in the formal analysis of a given empirical trace. The approach can also be applied in conjunction with simulation experiments.

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 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. Real life examples in the Netherlands of incidents that took on disastrous proportions because of inadequate human intervention are the plane crash in Amsterdam and the Hercules disaster in Eindhoven.

To manage an incident usually many parties have to cooperate. This means that incident management has a distributed or multi-agent character. A specific type of errors likely to occur in such situations has to do with interaction and coordination between these parties. Organising this multi-agent cooperation in a dynamic and adaptive manner, while minimising the number of errors is one of the main challenges.

In this paper the aims and some intermediate results of the project CIM (Cybernetic Incident Management) are presented. First, in Section 2 an overview of the aims of CIM is given. One specific part of the project deals with development of methods to provide automated support for the

analysis of what may have gone wrong in specific (simulated or empirical) traces of incident management. Some first results on this theme are presented in subsequent sections. In Section 3 an informal analysis of traces of two real life case studies is presented, and some first categorisation of the types of errors is made. In Section 4 a formalisation of the trace of one of these case studies is shown. Section 5 discusses a number of dynamic properties that have been formalised and automatically checked for the formalised trace from Section 4. Section 6 is a final discussion.

2 AIMS OF THE PROJECT CIM

In current practice, procedures for dealing with incidents are mostly on paper and have low accessibility. The execution of procedures is completely dependent on people and one wrong assessment or forgotten protocol can worsen the situation. Experience in dealing with incidents is limited because of the low occurrence leading to the repetition of mistakes. Analysis and reconstruction in retrospect is difficult because of the chaos during the incident and the lack of real time tracking of the actions and decisions of the people involved.

The aim of the 4-year project CIM (2003-2007) is to gather knowledge in order to create a constantly adapting system that encompasses both people and supporting software and that has the ability to

process and assess information in an adaptive, interactive and intelligent fashion to support human decisions. As a result, the execution of procedures and the assessment of information can be achieved more effectively. Due to the self-learning abilities of the system, experience is united in the system, not only in people but also in software agents and protocols. Because the system automatically records and even facilitates communication between involved parties, evaluation in retrospect can take place based on accurate data.

Knowledge in the system is contained in the communication structures and the supporting software in the form of distributed agents that are able to obtain and weigh information dynamically. The maintenance and evolution of the system is achieved by performing simulations and training sessions, both virtually as well as real-life. Measuring the effectiveness of a response and using this as feedback can improve the quality of the protocols and the system itself.

From a research point of view, the parties involved are the technical University Delft, the Free University Amsterdam, the Centre for Mathematics and Computer Science and Almende. From a commercial perspective CMotions and Falck are involved. Falck is the second largest security services provider globally with operations in more than 50 countries throughout the world. CIM applies best on the safety services market, ambulance services, emergency centres and fire departments, which Falck considers a potential growth market for future years. Almende is the spokesperson for the project. Almende is a Dutch ICT research company that specializes in complex agent based network solutions, principles of self-organisation, and system dynamics.

3 ANALYSIS OF TWO CASES

In this section an informal analysis of two real life case studies is presented: the Hercules disaster (Section 3.1), from [IBR, 1996], and the Dakota incident (Section 3.2), from [IBR, 1997]. In each of these cases various errors have been identified. In Section 3.3 a categorisation of them is given.

3.1 The Hercules disaster

On October 16th, 1996 at 6:03 p.m. a Hercules military airplane crashed at the airport of Eindhoven, in the Netherlands. This incident involves many examples of miscommunications and lack of communication and is therefore a well known example of a non optimal working disaster prevention organisation. An informal description of the events that took place during the rescue phase is presented below. The events during the alarm phase

are presented, after that the emergency assistance during that period is described.

Alarm phase. The Air-Traffic Control Leader on duty anticipated an accident and activated the so-called crash bell at 6:03 p.m. Through the intercom installation he announced that a Hercules plane had landed with an accident and pointed out the location of the plane. The Assistant Air-Traffic Control Leader at the same time contacted the Emergency Centre of the Fire department at the Airbase and reported the situation. The Fire department immediately took action. The Airbase Fire department must, when reporting to external authority, report which scenario is applicable. There are three different types of scenarios: Scenario 1: A maximum of 2 people involved, Scenario 2: More than 3 and less than or equal to 10 people. Scenario 3: More than 10 people. This all can be found on a checklist and also has consequences for the activities that should take place and the amount of authorities that need to be informed.

The Air-Traffic Control Leader on duty knew that at least 25 people were on board of the plane, this was due to a private source. He called the Emergency Centre of the Fire department at the Airbase around 6:04 p.m. with the order to call 06-11 (the national emergency number at that time). The Chief of the Airbase Fire department ('On Scene Commander', OSC) asked Air-Traffic Control for the number of people on board of the plane at 6:04 p.m. According to this person, the answer was 'nothing known about that'. Following from this the OSC reported Scenario 2 through the walkie-talkie. The Emergency Centre operator states not to have heard this but does not want to state that this has not been said. At 6:06 p.m. the Emergency Centre operator calls 06-11 and is connected to the Central Post for Ambulances (CPA). From that point on, the Emergency Centre operator got help from a fire fighter. Together they tried to inform several governmental officials.

At 6:12 p.m. the Regional Emergency Centre of the Fire department (RAC) Eindhoven phoned air-traffic control with the question whether backup was needed, the response was 'negative'. At 6:12 p.m. the Emergency Centre employee and the aforementioned fire fighter decided to follow Scenario 2 of the disaster plan (there were at least 4 people on board of the Hercules because that's the usual crew for this type of plane). At 6:15 p.m. the first civil fire trucks pulled out.

Emergency Assistance. Immediately after the announcement of the Air-Traffic Control Leader the Airbase Fire department went to the scene with a Land Rover Command vehicle (LARO) with the OSC and two Major Airport Crash Tenders (MAC's) each manned with a crew of 3 people. The OSC

thought that only the crew was on board and till the moment passengers had been found he handled accordingly.

At 6:19 p.m. there was complete control over the fire at the right wing and engine. Thereafter, at 6:25 p.m. the first civil fire trucks arrived on the scene. After their arrival the OSC contacted the chief of the first fire truck who was told that probably four people were on board of the plane. After pumping water to the MAC's at 6:38 p.m. they started extinguishing the left engine.

6:33 p.m. was the exact time point when the decision was made to go inside the plane and use a high-pressure hose to extinguish some small fires inside the plane. After that, at 6:37 p.m. the fire fighters were in the plane for the first time and shortly thereafter the first casualty was discovered. Almost at the same time 20 to 30 other casualties were discovered.

3.2 The Dakota incident

In the Dakota incident, other factors are involved in the emergency rescue process. For instance, some officers are not familiar with emergency procedures/protocols for the incident. The wrong procedures/protocols are picked up. An inefficient rescue procedures/protocols consequently is followed. Another example is that an overload of some of the partners can potentially cause some mistakes during the rescue process. However, miscommunications and inappropriate decisions are also involved in the rescue process. On September 25, 1996 a Dakota PH DDA of the Dutch Dakota Association (DDA) left Texel International Airport Holland. The plane had 6 crewmembers and 26 passengers on board. Shortly after take off the crew reported engine trouble to Texel International Airport Holland (TIA). Around 4:36 p.m. the crew contacted the Navy airbase The Kooy (MVKK) and stated that it wanted to make an emergency landing on The Kooy. After a short while, The MVKK observed that the Dakota disappears from the radar screen. The MVKK immediately sent a helicopter, initiated a team of rescue helicopters and alarmed the coast guard centre (KWC). At 16:46 the KWC passed the correct information of the incident to Regionale Alarmcentrale Kop van Noord-Holland (RAC) and asked the RAC to alarm the relevant partners. Unfortunately, the RAC only organised the rescue boats and vessels and did not alarm other parties, that should be warned in the incident. At 16:55, the KWC reported the incident to Noord Hollands Dagblad (a Dutch newspaper) and RTL TV station. Consequently, the KWC got many requests for information from the ANP (Dutch press office). The KWC is thus under a lot of pressure.

Through the ANP, the National Centre for Coordination (LCC) got the message that the Dakota has crashed. At 17:03 the LCC contacted the KWC, the KWC asked the LCC to help by providing a trauma team.

Coincidentally, a big drill for ambulances was ready to start. The Drill leader asked the president of the Dutch health service (GGD) whether the drill should still go on. At 17:05 the president of the GGD called RAC to inquire if the accident is for real. The RAC responded that neither the KWC nor the harbour office (HK) knew what was going on. The GGD even agreed to start the drill.

At almost the same time, the KWC asked the MVKK to take care of the wounded and told the LCC that the trauma team should be sent to MVKK. At 17:07 the LCC made an appointment with the Ministry of Public Health, Wellbeing, and Sports (VWS), the VWS finally arranged the trauma team. At 17:17 the first helicopter with casualties landed at Gemini Hospital (Gemini), the Gemini called the RAC to ask what the purpose of this is. The RAC replied that they only knew a plane had crashed and did not know anything more.

At 17:20 the RAC asked the KWC to get a trauma team from Gemini to MVKK. Meanwhile the centre for ambulances (CPA) of Amsterdam, the mayors of Den Helder and Wieringen, and the commander of the regional fire department are notified. After a while the arrangements of a crisis centre finally set up at the Navy. At 18:44 all bodies are found and transported. There is only one survivor of the incident.

3.3 Categorisation of Error Types

Based on the above informal traces of the Hercules and Dakota incidents, in this section a first attempt is made to categorize the probable causes of the mistakes made during the incident managing phase after the crashes.

3.3.1 Incomplete Information

First of all a property of urgent situations would be that a lot of decisions are made based on incomplete information. There may not be enough time or resources to gather all relevant information to support a decision, and therefore a wrong decision might be made. For example, in the Hercules case the operator of the Airbase Fire department has no knowledge of the amount of people on board of the plane, while he has to decide on who to call and what kind of backup to request, without this information. An approach that can be used for planning (which is a typical task in incident management) using incomplete information is for example presented in [Etzioni et al., 1992].

3.3.2 Contradictory Information

A second property is that a lot of decisions are made based on contradictory information. One might think of urgent situations in which a decision is made, in spite of a lacking sound support for it, which causes mistakes. For example, in the Dakota incident two numbers are mentioned to be the main information telephone number for relatives of the casualties.

3.3.3 Incorrect Information

Similar to the above properties, information can also be incorrect. Incorrect information obviously misleads incident management, and may cause errors. This incorrectness might be caused by, for instance, ill communication, accident, or misinterpretation. For example, in the Hercules disaster the air-traffic control leader knew how many people were on board of the plane, however he replied a request of the on scene commander for the amount of passengers with a denial of the fact that he knew the amount of people on board.

Decision making with incorrect information can be incorporated in reflective agents, an example of the modelling of reflective agents can be found in [Brazier and Treur, 1999]. In [Fargier et al., 1995] a constraint satisfaction framework for decisions under uncertainty is presented.

3.3.4 Use of Different Protocols

Another property is that in larger scale incidents a lot of parties are involved, and therefore it becomes more probable that different rules or protocols are used in situations where the same should have been used. In these situations parties might expect others to have different behaviours than they have in reality. For example, in the Hercules case the operator of the Airbase Fire department had in mind that the protocol involved calling 06-11. This was however another protocol in case of a less severe accident. This caused unnecessary delays.

3.3.5 Exception Handling

If disaster plans do not deal with different scenarios, or parties are not familiar enough with such plans, it is possible that exceptions are not handled well. For instance, in the Dakota incident the commander of the regional fire department is surprised when he hears from the Regional Alarm Centre about the incident and that the region is not involved in its management. Because it was not asked, the commander didn't take any further action. If he had, it probably would have been very helpful. In these cases a back-up plan should be available.

3.3.6 Work Overload

Finally, if tasks are not delegated properly to parties, or a party is not aware of the possibility of delegation, work overload most probably occurs, and might be another cause for errors. For example, in the case of the Dakota incident, during the first period after the crash, the coast guard had a lot to do, and therefore did not pay enough attention to initiating or delegating activities ashore. In these kinds of situations the coast guard should be relieved of the tasks that can be done by others.

Software agents can be a very useful help in supporting people to cope with all the incoming information and making the right decision, see for example [Brown et al., 1998].

4 FORMALISATION OF AN EMPIRICAL TRACE

Informal traces of events, such as the trace presented in Section 3.1 of the Hercules disaster, can be formalised using the formal language TTL [Jonker and Treur, 2002]. Formalising such a trace has several benefits. First of all, specific properties which should hold for a trace can be verified. An example of such a property in the case of an airplane crash is that a fire truck should be at the disaster area within 3 minutes according to the International Civil Aviation Organisation (ICAO). Some properties (like the example just mentioned) can often easily be checked by hand, but in more complex cases, a mistake may have been caused by a wrong chain of events. These types of causes are usually difficult to determine, and the formalisation can help for this purpose.

Abbreviation	Abbreviates
AFD	Airbase Fire Department
ATC	Air Traffic Control
CPA	Central Post Ambulances
MAC	Major Airport Crash tender
OSC	On Scene Commander
OSO	On Scene Operations
MHS	Medical Health Servies
OvD	Officer On Duty
CvD	Commander on Duty
0611	The national emergency number

Table 1: A list of all abbreviations

Another benefit of the formalisation is in the case where the protocol for the disaster prevention organisation was incorrect. After the protocol has been rewritten it can be formalised by means of executable properties and the scenario in which the previous protocol failed can be used as an input. Resulting from this, a simulation can be performed which in turn will result in a trace of the functioning

of the disaster prevention organisation when using the new protocol. By means of this trace the properties that failed with the previous protocol can again be verified to see whether the new protocol has indeed improved the functioning. In case the properties are again not satisfied the cause if this can be determined and the protocol can be revised until the desired properties are all satisfied.

An example of a formalisation of a trace is shown in Figure 1. It models the events that occurred during the Hercules disaster. Only a part of the trace is shown for the sake of brevity. On the left side of the picture the atoms are shown that are present in the trace. All atoms have the format

```
output(role)|communicated_from_to('src', 'dst', 'type', 'information')
```

The 'role' indicates the role that outputs this information, whereas the 'src' and 'dst' model the source and destination role (notice that 'role' = 'src' always holds). A list of all the abbreviations used for the roles is shown in Table 1.

The types of communication are based on speech act [Austin, 1976]. In the full trace also atoms containing input are present. Behind the atom there is a time line, indicating when the atom is true in the trace. For example, the first atom

```
output(ew)|communication_from_to(ew, 'ATC', observe, crash)
```

which states that the external world outputs a crash of a Hercules to air-traffic control, is true during the first minute after the crash, as he observes the crashed plane during that period.

```
output(ew)|communication_from_to(ew, 'ATC', observe, crash)
output(ew)|communication_from_to(ew, 'ATC', observe, hercules)
output('ATC')|communication_from_to('ATC', 'AFD', inform, crash)
output('ATC')|communication_from_to('ATC', 'MHS', inform, crash)
output('AFD')|communication_from_to('AFD', 'ATC', request, use_runway)
output('ATC')|communication_from_to('ATC', 'AFD', permit, use_runway)
output('ATC')|communication_from_to('ATC', 'AFD', request, call_0611)
output('AFD')|communication_from_to('AFD', 'ATC', request, n_of_people_in_plane)
output('ATC')|communication_from_to('ATC', 'AFD', inform, nothing_known)
output('AFD')|communication_from_to('AFD', 'O611', inform, crash)
output('AFD')|communication_from_to('AFD', 'CPA', inform, crash)
output('O611')|communication_from_to('O611', 'Marechaussee', inform, crash)
output('ATC')|communication_from_to('ATC', 'AFD', request, need_backup)
output('AFD')|communication_from_to('AFD', 'ATC', inform, negative)
output('ATC')|communication_from_to('ATC', 'RFD', inform, negative)
output('AFD')|communication_from_to('AFD', 'AFD', declare, scenario2)
output('AFD')|communication_from_to('AFD', 'Marechaussee', inform, crash)
output('AFD')|communication_from_to('AFD', 'RFD', inform, crash)
output('OSC')|communication_from_to('OSC', commander_ts1, request, deliver_water_mac3)
output('OSC')|communication_from_to('OSC', commander_ts1, request, ext_left_engine_powder)
output('OSC')|communication_from_to('OSC', commander_ts1, request, tools_opening_plane)
output('OSC')|communication_from_to('OSC', commander_ts1, request, ext_fires_with_high_pres)
output('OSC')|communication_from_to('OSC', 'OvD', inform, amount(people, 4))
output('ATC')|communication_from_to('ATC', 'AFD', inform, amount(people, 40))
output('AFD')|communication_from_to('AFD', all_units, inform, amount(people, 40))
output('AFD')|communication_from_to('AFD', ff_specialist, inform, four_zero)
output('OSC')|communication_from_to('OSC', ff_specialist, request, deliver_water_mac1)
output('OSC')|communication_from_to('OSC', ff_specialist, request, deliver_water_mac3)
output('CvD')|communication_from_to('CvD', 'RFD', inform, fully_on_scene)
output('CvD')|communication_from_to('CvD', 'RFD', inform, condition_of_plane)
output('CvD')|communication_from_to('CvD', 'RFD', inform, will_start_more_detailed_investigation)
output('AFD')|communication_from_to('AFD', off_duty_firemen, request, relieve_colleagues)
output('CvD')|communication_from_to('CvD', 'RFD', inform, amount(wounded_severely_burnt, 5))
output('CvD')|communication_from_to('CvD', 'RFD', request, use_truck_ts4_for_cooling)
output('OSC')|communication_from_to('OSC', head_afd, inform, fax_with_passengers)
output('CvD')|communication_from_to('CvD', 'RFD', inform, amount(wounded_trans_alive, 10))
output('CvD')|communication_from_to('CvD', 'RFD', inform, amount(casualties, at_least(26)))
```

A verification of properties that should hold for the disaster prevention organisation is presented in the next section.

5 VALIDATION OF A TRACE

After having obtained a formalised trace, either by formalisation of an empirical trace or by a simulated trace, it is useful to verify some essential properties of the trace. By means of this verification one can determine what precisely went wrong in the trace. Below a number of properties are expressed that in particular are relevant for the Hercules case. Properties are represented in structured semi-formal format. All of them have been formalised using TTL (Temporal Trace Language) [Jonker and Treur, 2002]. For one of the properties it is shown what the formalisation looks like.

All given properties have been verified using a special software environment *TTL Checker* that has been developed for the purpose of verifying TTL properties over traces. The results of verification are given below.

P1: Information Correctness

At any point in time t1, if AFD generates a request for ATC about the number of people on the plane, then at a later point in time t2 ATC will communicate the correct answer to AFD.

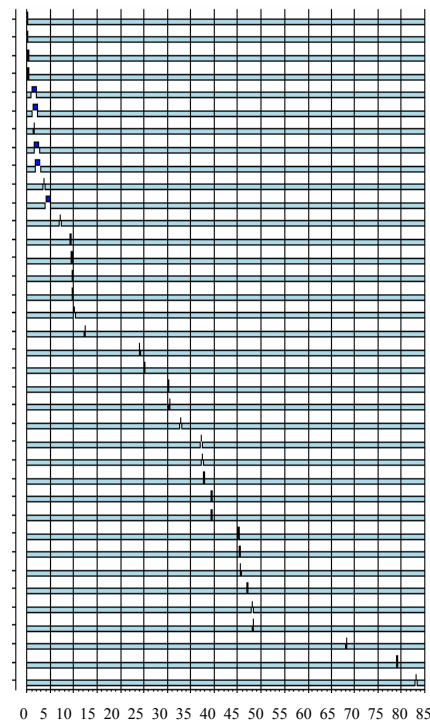


Figure1: Formalised empirical trace of the Hercules disaster

Automated verification showed that this property is not satisfied in the given trace.

P2: Choice of Protocols

At any points in time t1 and t2, t2>=t1, if ATC generates information to AFD about the plane crash at t1, and that the number of passengers is more than 10 at t2, then at a later point in time t3 AFD declares Scenario 3.

Formalization of the property P2:

```

∀t1, t2, x [ t2>=t1 ⇒ [ state(γ, t1, input('AFD'))]=
  communication_from_to('ATC','AFD', inform, crash)
& x>10 & state(γ, t2, input('AFD'))=
  communication_from_to('ATC','AFD', inform, amount(people, x)) ]
⇒ ∃t3>t2 & state(γ, t3, output('AFD'))=
  communication_from_to('AFD','AFD', declare, scenario3) ]
    
```

This property is not satisfied for the given trace.

P3: Timely Information Delivery

At any point in time t1, if ATC generates information for AFD about the plane crash, then at a later point in time t2, t2 ≤ t1+2 AFD will communicate this information to RFD.

This property is not satisfied for the given trace.

P4: MAC Timely Coming to the Disaster Area

At any point in time t1, if AFD receives information from ATC about the plane crash, then at a later point in time t2 MAC will join AFD, and at a still later point in time t3 will come to the disaster area in less than 3 minutes upon the plane crash information reception.

This property is satisfied for the given trace.

P5: Sufficient Number of Ambulances, Called Immediately

At some time point t1, if CPA communicates ambulances with the request to come to the disaster area, then at no later point in time t2 CPA will ask for additional ambulances.

This property is not satisfied for the given trace.

As one can see from the results of properties verification, given above, 4 from 5 properties aren't satisfied over the trace. By analyzing the obtained results one can get insight in which types of errors occurred in the scenario and which points of the disaster plan weren't fulfilled.

6 DISCUSSION

The project CIM started in 2003. In this paper some intermediate results were shown, especially on automated support of analysis of errors in traces of incident management. The potential of the approach was shown in the formal analysis of a given empirical trace. However, the approach can also be applied in conjunction with simulation experiments. If a large number of simulated traces are generated, for example by varying parameters or initial information, for these simulated traces it can be checked automatically which dynamic properties hold or fail for which of the traces

Usually one can specify dynamic properties at different aggregation levels, from global properties,

to more local properties, and establish hierarchical interlevel relations between the properties. If one of the global properties doesn't hold, then verification of properties at intermediate levels can follow to identify were the cause of the problem can be found. The verification process can be continued up to the lowest level, consisting of the simplest local properties. See [Jonker et al., 2002] for more details of this diagnostic approach.

Further potential uses of the checking of dynamic properties is by investigating whether or not certain mistakes would still exist after a modification of the relating protocols. The possibility to check such properties formally, can provide a clue to get to solutions or recommendations in terms of improved protocols.

REFERENCES

[Austin, 1976] J.L. Austin. *How to do things with words.* Oxford University Press, 2nd edition, 1976.

[Brazier and Treur, 1999] Brazier, F.M.T., Treur, J., Compositional modelling of reflective agents. *International Journal of Human-Computer Studies*, vol. 50, 1999, pp. 407-431.

[Brown et al., 1998] Brown, S. M.; Santos Jr., E.; Banks, S. B.; and Stytz, M. R. Intelligent interface agents for intelligent environments. In *Proceedings of the 1998 AAAI Spring Symposium on Intelligent Environments.*

[Etzioni et al., 1992] O. Etzioni, S. Hanks, D. Weld, D. Draper, N. Lesh, and M. Williamson. An Approach to Planning with Incomplete Information. In *Proc. 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning*, 1992.

[Fargier et al, 1995] Fargier H., Lang J., Martin-Clouaire R. et Schiex T., A constraint satisfaction framework for decision under uncertainty. In : *Proc. of the 11th Int. Conf. on Uncertainty in Artificial Intelligence*, 1995.

[IBR, 1996] Inspectie Brandweezorg en Rampenbestrijding, *Vliegtuigongeval Vliegbasis Eindhoven 15 juli 1996*, SDU Grafische Bedrijf, The Hague, 1996.

[IBR, 1997] Inspectie Brandweezorg en Rampenbestrijding, *Dakota-incident Waddenzee 1996*, SDU Grafische Bedrijf, The Hague, 1997.

[Jonker et al., 2002] Jonker, C.M., Letia, I.A., and Treur, J., Diagnosis of the Dynamics within an Organisation by Trace Checking of Behavioural Requirements. In: Wooldridge, M., Weiss, G., and Ciancarini, P. (eds.), *Agent-Oriented Software Engineering, Proc. of Second Int Workshop AOSE'01.* Lecture Notes in Computer Science, vol. 2222. Springer Verlag, 2002, pp. 17-32.

[Jonker and Treur, 2002] 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, 2002, pp. 51-92.