A Formal Method to Analyze Human Reasoning and Interpretation in Incident Management[†]

Tibor Bosse[1], Mark Hoogendoorn[1], Catholijn M. Jonker[2], and Jan Treur[1]

[1]Vrije Universiteit Amsterdam, Department of Artificial Intelligence, De Boelelaan 1081a,
1081 HV Amsterdam, The Netherlands, {tbosse, mhoogen, treur}@cs.vu.nl

[2]Delft University of Technology, Department of Mediametics, Mekelweg 4,
2628 CD Delft, The Netherlands, catholijn@mmi.tudelft.nl

Abstract

The study of human reasoning often concentrates on reasoning *from* an already assumed interpretation of the world, thereby neglecting reasoning *towards* an interpretation. In recent literature within Cognitive Science, means taken from the area of nonmonotonic logic are proposed to analyze the latter aspect of human reasoning. In this paper this claim is further worked out and tested against empirical material of human reasoning during critical situations (incident management). Empirical and simulated reasoning traces have been analyzed by comparing them and by automatically checking properties on them.

1. Introduction

In recent years, from the area of Cognitive Science, there is an increasing interest in tools originating from the area of nonmonotonic reasoning. In (Stenning and van Lambalgen, 2006) it is shown how the empirical study of human reasoning processes has been too much dominated by an emphasis on classical, deductive logic. This applies equally well to the socalled rule-based or syntactic stream (e.g., Braine and O'Brien, 1998; Rips, 1994), as to the model-based or semantic stream (e.g., Johnson-Laird, 1983; Johnson-Laird and Byrne, 1991). In their analysis of human reasoning they claim that much more important than the question whether reasoning should be considered from a syntactical or semantical perspective, is the distinction between: a) reasoning *towards* an interpretation, and b) reasoning *from* an interpretation. The latter type of reasoning is reasoning within an already unambiguously determined formalized frame, and can be analyzed by means of classical logic. The first type of reasoning, however, still has to find such a frame and has to deal with ambiguities and multiple interpretation possibilities, and does not have a unique outcome. It is at this point that they propose nonmonotonic logic as a more adequate analysis tool for human reasoning processes. Within nonmonotonic logic it is possible to formalize reasoning processes that deal with multiple possible outcomes, which can be used

---

[†] This paper is a significantly extended and improved version of Bosse, T., Hoogendoorn, M., Jonker, C.M., and Treur, J., A Formal Empirical Analysis Method for Human Reasoning and Interpretation. In: *Proceedings of the 8th International Conference on Cognitive Modeling, ICCM'07*. Taylor and Francis, 2007, to appear.

to model different possibilities of interpretation; see (Engelfriet and Treur, 2003) for a similar perspective. Thus, from an empirical angle, within the area of human reasoning within Cognitive Science, a new, more empirical perspective was introduced to study nonmonotonic reasoning processes.

The current paper reports research to further work out and test this empirical perspective in the context of incident management. Detailed reports are available that describe what went wrong in the management of well-known disasters, see, e.g., (Ministry of the Interior, 1996). These reports provide empirical data showing how humans reason under the pressure of a critical situation. Cases taken from them form the basis of the research reported in this paper to further detail and illustrate the use of the Stenning-van Lambalgen perspective on reasoning and interpreting and use this perspective to detect and understand errors within incident management. The leading example is an airplane crash.

The outline of the paper is as follows. The aircrash example is presented in Section 2. Section 3 presents an abstract formalization of a reasoning process leading to multiple interpretations, and Section 4 shows how Default Logic can be used to specify such processes. To obtain simulation of such reasoning, variants of Default Logic are considered in which control decisions can be represented. To this end, in Section 5 a temporalized form of Default Logic is chosen to simulate the possible reasoning traces for the case study. In Section 6 a number of properties of such reasoning traces are formalized and checked. Section 7 presents the conclusions.

2. The Incident Management Domain

The domain of incident management is characterized by people working under severe pressure, in which split second decisions have to be made which can have a huge impact on the successfulness of the whole operation. In addition, such decisions often have to be made without having complete information on the current state of affairs. As a result of these factors, errors are frequently observed within incident management organizations.

One well-known example of erroneous functioning of an incident management organization in the Netherlands is that of the Hercules airplane crash at the military airport of Eindhoven [Ministry of the Interior, 1996]. The plane flew into a flock of birds just before landing, causing one of the engines to fail, making the plane tilt to one side. As a result, the plane crashed on the runway and caught fire, carrying a military brass band in the cargo room and a crew of four people. The Air Traffic Controller (ATC) immediately hit the alarm button, also having knowledge that a military brass band is on board of the plane. Afterwards he claimed to have informed the alarm center operator of this fact, who in turn stated never to have received the information. As a result, the operator did inform fire fighters, but declared the wrong scenario (for merely the crew on board). After the fire fighting forces had arrived at the scene, one of them contacted the air traffic controller, asking how many people are on board of the plane. Since the air traffic controller reasoned under the assumption that the message of a military brass band being on board had been passed through to the fire fighters, he answered that this is unknown, interpreting the question as a request for the exact amount of people on board. The fire fighter therefore assumed that only the crew was on board of the plane, which is not an assumption that should have been made according to training material [NIBRA, 2001], especially not because over 50% of these types of planes carry passengers in the cargo room. Due to the incorrect assumption, the brass band in the cargo room was not rescued until 30 minutes after the crash, which could have been well within 10 minutes, possibly saving precious lives.

3. Multiple Interpretations

In a broad sense, reasoning toward an interpretation can be viewed as an activity where an agent, given some initial information (or set of beliefs) *X*, performs some manipulation to this information and arrives at a new state with different information. So a (partial) view on a situation (in the domain the agent is reasoning about) is transformed to another partial view. In general the mechanism may be non-deterministic in the sense that multiple possible views on the world can result from such a reasoning process. In (Engelfriet and Treur, 2000) two levels of abstraction for the specification of such reasoning were described:

1. Specification of a set of multiple belief sets for any initial set X
Specification of the possible belief states for the agent abstracting from the specific reasoning patterns that lead to them. This describes the input-output behavior of the agent's reasoning process.

2. Specification of a set of reasoning traces for any initial set X
Specification of the different reasoning traces that lead to the possible belief states.

Of course, a connection exists between the two levels in the sense that from a specification of the lower level of abstraction (the reasoning traces) in an unambiguous manner a specification of the higher level can be determined. One could say the specification at the lower level gives in some sense a refinement or specialization of the specification at the higher level (as in the case of conventional software specifications at different levels of abstraction). Given specifications of two different levels, relative verification is possible: to establish whether the lower level one indeed refines the higher level one. At a lower level different specifications can refine the same higher level specification.

To obtain a reasoning trace, a number of subsequent reasoning steps have to be made. Each reasoning step may introduce an additional assumption, that provides a constraint on the reasoning steps that still can be made. For example, if there are two possibilities, one to generate an assumption *a*, and another one that generates an assumption *b* and it is known that *a* implies *not b*, then introducing *a* makes it impossible to introduce *b* later on, and vice versa. So the choice to apply one of these two reasoning steps indicates a branching point for the reasoning process. This is an element in common for practically all approaches to nonmonotonic logic. Moreover, many translations between different approaches have been made. For more details and approaches in nonmonotonic logic and their relationships, see (Marek and Truszczynski, 1993).

Ignoring the detailed reasoning steps, nonmonotonic reasoning can be formalized at the more abstract level as follows. A particular interpretation for a given set of formulae considered as input information for the reasoning, is formalized as another set of formulae, that in one way or the other is derivable from the input information (output of the reasoning towards an interpretation). In general there are multiple possible outcomes. The collection of all possible interpretations derivable from a given set of formulae as input information (i.e., the output of the reasoning towards an interpretation) is formalized as a collection of different sets of formulae. Note that these formalisms also apply to reasoning from an interpretation. A formalization describing the relation between such input and output information is described at an abstract level by a multi-interpretation operator. The input information is described by propositional formulae in a propositional language $\mathbf{L_1}$. An interpretation is a set of propositional formulae, based on a propositional language $\mathbf{L_2}$.

Definition 1 (Multi-Interpretation Operator)
a) A *multi-interpretation operator* **MI** with input language $\mathbf{L_1}$ and output language $\mathbf{L_2}$ is a function $\mathbf{MI} : \mathsf{P}(\mathbf{L_1}) \rightarrow \mathsf{P}(\mathsf{P}(\mathbf{L_2}))$ that assigns to each set of input facts in $\mathbf{L_1}$ a set of sets of formulae in $\mathbf{L_2}$.
b) A multi-interpretation operator **MI** is *non-inclusive* if for all $\mathbf{X} \subseteq \mathbf{L_1}$ and $\mathbf{S}, \mathbf{T} \in \mathbf{MI(X)}$, if $\mathbf{S} \subseteq \mathbf{T}$ then $\mathbf{S} = \mathbf{T}$.
c) If $\mathbf{L_1} \subseteq \mathbf{L_2}$, then a multi-interpretation operator **MI** is *conservative* if for all $\mathbf{X} \subseteq \mathbf{L_1}$, $\mathbf{T} \in \mathbf{MI(X)}$ it holds $\mathbf{X} \subseteq \mathbf{T}$.

The condition of non-inclusiveness guarantees a relative maximality of the possible interpretations. Note that when **MI(X)** has exactly one element, this means that the set $\mathbf{X} \subseteq \mathbf{L_1}$ has a unique interpretation under **MI**. The notion of multi-interpretation operator is a generalization of the notion of a nonmonotonic belief set operator, as introduced in (Engelfriet, Herre, and Treur, 1998). The generalization was introduced and applied to approximate classification in (Engelfriet and Treur, 2003). A reasoner may explore a number of possible interpretations, but often, at some point in time a reasoner will focus on one (or possibly a small subset) of the interpretations. This selection process is formalized as follows (see Engelfriet and Treur, 2003).

Definition 2 (Selection Operator)
a) A *selection operator* **s** is a function $\mathbf{s} : \mathsf{P}(\mathsf{P}(\mathbf{L})) \rightarrow \mathsf{P}(\mathsf{P}(\mathbf{L}))$ that assigns to each nonempty set of interpretations a nonempty subset: for all **A** with $\phi \neq \mathbf{A} \subseteq \mathsf{P}(\mathbf{L})$ it holds $\phi \neq \mathbf{s(A)} \subseteq \mathbf{A}$. A selection operator **s** is *single-valued* if for all non-empty **A** the set **s(A)** contains exactly one element.
b) A *selective interpretation operator for* the multi-interpretation operator **MI** is a function $\mathbf{C} : \mathsf{P}(\mathbf{L_1}) \rightarrow \mathsf{P}(\mathbf{L_2})$ that assigns one interpretation to each set of initial facts: for all $\mathbf{X} \subseteq \mathbf{L_1}$ it holds $\mathbf{C(X)} \in \mathbf{MI(X)}$.

It is straightforward to check that if $\mathbf{s} : \mathsf{P}(\mathsf{P}(\mathbf{L_1})) \rightarrow \mathsf{P}(\mathsf{P}(\mathbf{L_2}))$ is a single-valued selection operator, then a selective interpretation operator **C** for multi-interpretation operator **MI** can be defined by the composition of **MI** and **s**, i.e., by setting $\mathbf{C(X)} = \mathbf{s(MI(X))}$ for all $\mathbf{X} \subseteq \mathbf{L_1}$.

In this section some interpretations that play a role in the analysis of the plane crash accident are taken as the leading example. This information was derived based on training material, see (NIBRA, 2001).

3.1 Initial ATC interpretation

This Section first addresses the informal representation using textual description of the possible observations, interpretations, and actions. Thereafter, the formal description is addressed.

Informal Description. The first part concerns the ATC receiving initial observations from the external world, as shown in Table 1. Hereby W denotes the world state, O observations, I interpretations, and π the actions. Note that in all tables, the correct observations, interpretations, and actions are denoted in ***bold italics***. Two possibilities are denoted here, namely one observation including the fact that the ATC knows that a military brass band is on board, and the other one where he the ATC does not observe the presence of a brass band.

World State	Description	Party	Obs	Description
W_0	Initial world state, just after the crash of the Hercules plane. No communication has taken place, knowledge is present about a military brass band being on board.	ATC	***O_0***	Observation that a Hercules plane has crashed on the runway. Furthermore, the observation includes the fact that a military brass band is on board of the plane.
		ATC	O_1	Observation that a Hercules plane has crashed on the runway.

Table 1. Initial observations to ATC

After having received the observations, the ATC needs to interpret the situation, as shown in Table 2. The correct interpretation is the fact that a Hercules plane has crashed, and more than 25 people are on board.

Party	Obs	Interpretation	Description
ATC	***O_0***	***I_0***	Hercules plane crashed, minimum of 25 people on board of the plane.
		I_1	Hercules plane crashed, certainly more people on board than merely the flying crew.
		I_2	Hercules plane crashed, unknown amount of people on board of the plane.
ATC	O_1	I_2	
		I_3	Hercules plane crashed, possibly more people on board of the plane than merely the flying crew.

Table 2. ATC observations leading to an interpretation

Such interpretations can lead to the actions specified in Table 3, which involve communications to the operator.

Party	Interpretation	Action	Description
ATC	***I_0***	***π_0***	Call operator with the message that a Hercules plane has crashed, furthermore, mention that at least 25 people are on board of the plane, and request to call 06-11 for backup.
		π_1	Call operator with the message that a Hercules plane has crashed, request to call 06-11 for backup.
		π_2	Call operator with the message that a Hercules plane has crashed, furthermore, mention that at least 25 people are on board of the plane, and request to call for backup, do not use 06-11 but call the different parties directly to avoid delays.
		π_3	Call operator with the message that a Hercules plane has crashed, request to call for backup, do not use 06-11 but call the different parties directly to avoid delays.
	I_1	π_1	
		π_3	
		π_4	Call operator with the message that a Hercules plane has crashed, furthermore, mention that certainly more people are on board besides the flying crew, and request to call 06-11 for backup.
		π_5	Call operator with the message that a Hercules plane has crashed, furthermore, mention that certainly more people are on board besides the flying crew, and request to call for backup, do not use

			06-11 but call the different parties directly to avoid delays.
	I_2	π_1	
		π_3	
		π_6	Call operator with the message that a Hercules plane has crashed, furthermore, tell him to expect the worst, possibly lots of passengers on board, and request to call 06-11 for backup.
		π_7	Call operator with the message that a Hercules plane has crashed, furthermore, tell him to expect the worst, possibly lots of passengers on board, and request to call for backup, do not use 06-11 but call the different parties directly to avoid delays.
		π_8	Call operator with the message that a Hercules plane has crashed, furthermore, tell him information regarding passengers is being retrieved, and request to call for backup, do not use 06-11 but call the different parties directly to avoid delays. Furthermore, call a party that knows the amount of people on board of the plan.
	I_3	π_1	
		π_3	
		π_6	
		π_7	
		π_8	

Table 3. ATC interpretations leading to an action

Figure 1 gives an overview of a subset of the possibilities addressed in the Tables above, namely the ones that are mentioned in the disaster report. According to the report, there is a difference in opinion as to whether or not the ATC communicated to the operator that there are more than 25 people on board. The Figure shows the world state at time 0, W_0, and as a consequence of the communication to the operator, W_1 and W_2, which correspond with the two interpretations above. A difference is made between the observation (O_0), the internal representation made from that (I_0), and the interpretation of the situation in terms of actions to take (pi_0 and pi_1). There are two moments of interpretation: from observations to internal representation, and from internal representation to actions.

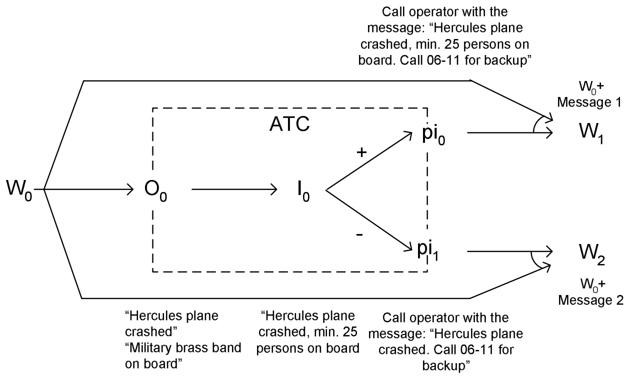

Figure 1. Reasoning Traces based on Interpretations

Formalisation. Given the subset depicted in Figure 1, the following formalization of the situation can be made. First, the ATC receives certain initial observations:

observation(plane_crash, pos), observation(cargo_plane, pos),
observation(passengers_on_board, pos).

Note that the sign 'pos' indicates that the element has been observed as being true, whereas a 'neg' indicates it is observed to be false. Focusing on the ATC, the analysis results in two interpretations that differ only in the communication to the operator, formalized as follows:

Common part of the interpretations

```
observation(passengers_on_board,pos)
observation(cargo_plane,pos)
observation(plane_crash,pos)
belief(plane_crash_occurred,pos)
belief(passenger_count(more_than_25),pos)
not belief(passenger_count(maximum_4),pos)
not belief(passenger_count(unknown),pos)
action(communicate_to(plane_crash,operator),pos)
action(communicate_to(call_backup_via_06_11,operator),pos)
```

Interpretation 1: common part +

```
action(communicate_to(passenger_count(more_than_25),operator),pos)
not action(communicate_to(passenger_count(maximum_4),operator),pos)
not action(communicate_to(passenger_count(unknown),operator),pos)
```

Interpretation 2: common part +

```
not action(communicate_to(passenger_count(more_than_25),operator),pos)
not action(communicate_to(passenger_count(maximum_4),operator),pos)
not action(communicate_to(passenger_count(unknown),operator),pos)
```

3.2 Operator Interpretation

For the operator interpretation the formalisation has been omitted for the sake of brevity. The informal is specified in the tables below. After the ATC has communicated the situation description to the operator, several possible worlds exist. The worlds are defined as follows: $W_{x+1} = W_0 + \pi_x$. The observations resulting from those worlds are shown in Table 4.

World State	Description	Party	Obs	Description
W_1	**$W_0 + \pi_0$**	Operator	**O_2**	Hercules plane has crashed, at least 25 people on board, should call 06-11.
		Operator	**O_3**	Hercules plane crashed, should call 06-11.
		Operator	O_4	Hercules plane crashed.
W_2	**$W_0 + \pi_1$**	Operator	**O_3**	
		Operator	O_4	
W_3	$W_0 + \pi_2$	Operator	O_4	
		Operator	O_5	Hercules plane crashed, at least 25 people on board, call emergency services directly.
		Operator	O_6	Hercules plane crashed, call emergency services directly.
W_4	$W_0 + \pi_3$	Operator	O_4	
		Operator	O_6	
W_5	$W_0 + \pi_4$	Operator	O_3	
		Operator	O_4	
		Operator	O_7	Hercules plane crashed, certainly more people than merely the crew on board, should call 06-11
W_6	$W_0 + \pi_5$	Operator	O_4	
		Operator	O_6	
		Operator	O_8	Hercules plane crashed, certainly more people than merely the crew on board, call emergency services directly.
W_7	$W_0 + \pi_6$	Operator	O_3	
			O_4	
			O_9	Hercules plane crashed, assume more people than merely

				the crew on board, call 06-11.
W_8	$W_0 + \pi_7$	Operator	O_4	
		Operator	O_6	
		Operator	O_{10}	Hercules plane crashed, assume more people than merely the crew on board, call emergency services directly.
W_9	$W_0 + \pi_8$	Operator	O_4	
		Operator	O_6	
		Operator	O_{11}	Hercules plane crashed, amount of people on board being requested, call emergency services directly.

Table 4. Worlds leading to an operator observation

Thereafter an interpretation is made by the operator; the interpretation only has a limited number of options as shown in Table 5.

Party	Obs	Interpretation	Description
Operator	$\boldsymbol{O_2}$, $\boldsymbol{O_3}$, O_7, O_9	I_4	Scenario 3 (more than 10 people involved), request to call 06-11 for backup.
		$\boldsymbol{I_5}$	Scenario 2 (between 3 and 10 people involved), request to call 06-11 for backup.
		I_6	Scenario 1 (less than 3 people involved), request to call 06-11 for backup.
		I_7	Scenario 3.
		I_8	Scenario 2.
		I_9	Scenario 1.
	O_4	I_7	
		I_8	
		I_9	
	O_5, O_6, O_{10}, O_{11}	I_7	
		I_8	
		I_9	
		I_{10}	Scenario 3, request to call emergency services directly.
		I_{11}	Scenario 2, request to call emergency services directly.
		I_{12}	Scenario 1, request to call emergency services directly.

Table 5. Operator observation leading to an interpretation

Finally, based on these interpretations, actions are derived that ought to be performed. These actions are expressed in Table 6.

Party	Interpretation	Action	Description
Operator	I_4, $\boldsymbol{I_5}$, I_6	$\boldsymbol{\pi_9}$	Declare the interpreted scenario (note that actually different π's are present for each scenario), and call 06-11 for backup.
	I_7, I_8, I_9	π_{10}	Declare the interpreted scenario (note that actually different π's are present for each scenario).
	I_{10}, I_{11}, I_{12}	π_{11}	Declare the interpreted scenario (note that actually different π's are present for each scenario), and call the emergency services directly for backup.

Table 6. Operator interpretations leading to an observation

Note that in Figure 2 the relevant part of the observations, interpretations, and actions are shown. Hereby the initial worlds are related to the actions performed by the ATC as discussed in Section 3.1. The numbering of the states in the figure does not match the numbering used in the tables. To maintain clarity in the figure, the states have been numbered in sequence. The matching state can easily be found in the table by looking at the appropriate section.

3.3 On Scene Commander Question

After the operator has declared the scenario as addressed in the previous Section, the second person comes into play, which is the On Scene Commander (OSC). For now, the assumption is that two possible worlds exist for this scenario, namely a world in which scenario 2 is declared but not received by the OSC and either a message with a communication of the amount of people (W_{10}), or without (W_{11}) is communicated. These options are specified in Table 7.

World State	Description	Party	Obs	Description
W_{10}	Hercules airplane crashed, operator declared scenario 2, which was not received by anyone. Furthermore, ATC has communicated that at least 25 people are on board of the plane.	OSC	***O_{12}***	Hercules plane crashed.
			O_{13}	Hercules plane crashed scenario 2 applicable.
			O_{14}	Hercules plane crashed scenario 2 applicable, at least 25 people on board of the plane.
W_{11}	Hercules airplane crashed, operator declared scenario 2, which was not received by anyone.		O_{12}	
			O_{13}	

Table 7. Worlds leading to an OSC observation

After having received these observations, an interpretation can be made, see Table 8.

Party	Obs	Interpretation	Description
OSC	***O_{12}***	***I_{12}***	Hercules plane crashed, unknown amount of people on board. Must obtain the amount of people on board in order to properly determine strategy.
		I_{13}	Hercules plane crashed, unknown amount of people on board. Investigate plane to see whether people are observed to be present in the cargo room.
OSC	O_{13}	I_{12}, I_{13}	
		I_{14}	Hercules plane crashed, scenario 2 applicable, therefore between 3-10 people on board. Request the exact amount of people on board of the plane.
OSC		I_{15}	Hercules plane crashed, scenario 2 applicable, therefore between 3-10 people on board. Investigate plane to see whether where these people are located.
OSC	O_{14}	$I_{12}, I_{13}, I_{14}, I_{15}$	

Table 8. Observations leading to an OSC interpretation

Finally, the actions are performed, two of them involving asking information from the appropriate parties as expressed in Table 9.

Party	Interpretation	Action	Description
OSC	***I_{12}***	π_{12}	Ask the operator how many people are on board of the plane.
		π_{13}	Ask the Air Traffic Controller how many people are on board of the plane.
	I_{13}	π_{14}	Walk around the plane, look through openings and windows to see whether people are present within that part of the plane.
	I_{14}	π_{12}, π_{13}	
	I_{13}	π_{14}	

Table 9. Observations leading to an OSC interpretation

Again, the relevant parts of the Tables are depicted in Figure 2.

3.4 ATC Response

Now a selection is again made of the worlds that are possible (given the worlds that are the result of the actions performed by the OSC), namely precisely those worlds that are mentioned in the disaster plan. In this case these worlds only concern the communication of questions to the ATC. That are precisely two worlds: (1) W_{10} with action π_{13} from the OSC, and (2) W_{11} with π_{13} action of the OSC, these are named W_{12} and W_{13} respectively, as shown in Table 10.

World State	Description	Party	Obs	Description
W_{12}	Hercules airplane crashed, military brass band on board of the plane. Operator declared scenario 2, which was not received by anyone. Furthermore, ATC has communicated that at least 25 people are on board of the plane. OSC has requested the amount of people on board of the plane.	ATC	***O_{14}***	Hercules airplane crashed, message of approximately 25 people has been communicated, OSC has requested the amount of people on board of the plane.
			O_{15}	Hercules airplane crashed, military brass band on board of plane, OSC has requested the amount of people on board of the plane.
W_{13}	Hercules airplane crashed, operator declared scenario 2, which was not received by anyone. OSC has requested the amount of people on board of the plane.		***O_{14}***	
			O_{15}	

Table 10. Worlds leading to the second set of ATC observations

An interpretation is created based on these observations, see Table 11.

Party	Obs	Interpretation	Description
ATC	***O_{14}***	***I_{16}***	The approximate amount of people has already been communicated whereas the OSC asks for the amount of people on board. Therefore, he means to ask what the exact amount of people is. The exact amount of people is unknown.
		I_{17}	The approximate amount of people has already been communicated however there is not guarantee that OSC heard this. OSC asks for the amount of people on board. Therefore, he could mean to ask what the exact amount of people or the approximate. The exact amount of people

			is unknown whereas the approximate amount is not. Need additional information to distinguish between these two.
ATC	O_{15}	I_{18}	The approximate amount of people has not been communicated to OSC yet, OSC asks the amount of people on board of the plane, he wants to know an approximation.

Table 11. Observations leading to ATC interpretations

Finally, such an interpretation leads to the actions as specified in Table 12 of which selections are shown in Figure 2.

Party	Interpretation	Action	Description
OSC	$\boldsymbol{I_{16}}$	$\boldsymbol{\pi_{15}}$	Communicate to the OSC that an unknown amount of people is on board of the plane.
		π_{16}	Communicate to the OSC that the *exact* amount of people on board of the plane is unknown.
	I_{17}	π_{17}	Ask the OSC whether he wants to know the exact amount of people on board of the plane, or whether he wants to have an approximate number.
	I_{18}	π_{18}	Communicate to the OSC that the amount of people on board of the plane is 25.
		π_{19}	Communicate to the OSC that the *approximate* amount of people on board of the plane is 25.

Table 12. Second interpretation of the ATC leads to actions

3.5 OSC Response

Finally, the OSC takes the answer given by the ATC into account. Two world states are distinguished (the most likely resulting from the actions of the ATC), namely W_{14} and W_{15} which are based on W_{12} and W_{13} respectively, with in addition the answer "unknown" from the ATC. First, the resulting observations are shown in Table 13.

World State	Description	Party	Obs	Description
$\boldsymbol{W_{14}}$	Hercules airplane crashed, military brass band on board of the plane. Operator declared scenario 2, which was not received by anyone. Furthermore, ATC has communicated that at least 25 people are on board of the plane which did not reach OSC. OSC has requested the amount of people on board of the plane which was said to be unknown.	OSC	$\boldsymbol{O_{16}}$	Hercules plane crashed with an unknown amount of people on board.
			O_{17}	Hercules plane crashed where the *exact* amount of people on board of the plane is unknown.
$\boldsymbol{W_{15}}$	Hercules airplane crashed, operator declared scenario 2, which was not received by anyone. OSC has requested the amount of people on board of the plane which was said to be unknown.		$\boldsymbol{O_{16}}$	

Table 13. Second set of observations for the OSC

An interpretation is created based on these observations as expressed in Table 14.

Party	Obs	Interpretation	Description
OSC	***O_{16}***	***I_{19}***	The amount of people on board of the plane is unknown, since typically these planes fly with people on board, assume that passengers are on board.
		I_{20}	The amount of people on board of the plane is unknown, since this plane is a cargo plane, assume the plane only carries a crew. A severe fire is present in the cockpit, assume the pilots cannot have survived.
		I_{21}	The amount of people on board of the plane is unknown, since this plane is a cargo plane, assume the plane only carries a crew.
OSC	O_{17}	I_{22}	Apparently the exact amount of people on board of the plane is not known, therefore assume worst case: a lot of people on board.
		I_{19}, I_{20}, I_{21}	

Table 14. Observations lead to interpretations.

Finally, such an interpretation leads to action (Table 15).

Party	Interpretation	Action	Description
OSC	***I_{19}***	***π_{20}***	After the 90% knock down, open the cargo room doors or create an entrance a.s.a.p.
		π_{21}	After the 90% knock down, open the cockpit first, then proceed to the cargo room.
	I_{20}	***π_{22}***	Extinguish the plane, do not attempt to rescue.
		π_{21}	
	I_{21}	π_{21}	
	I_{22}	π_{20}	
		π_{21}	

Table 15. Second interpretation of the OSC leads to actions

Figure 2 presents an overview of the various interpretations specified in the Sections above.

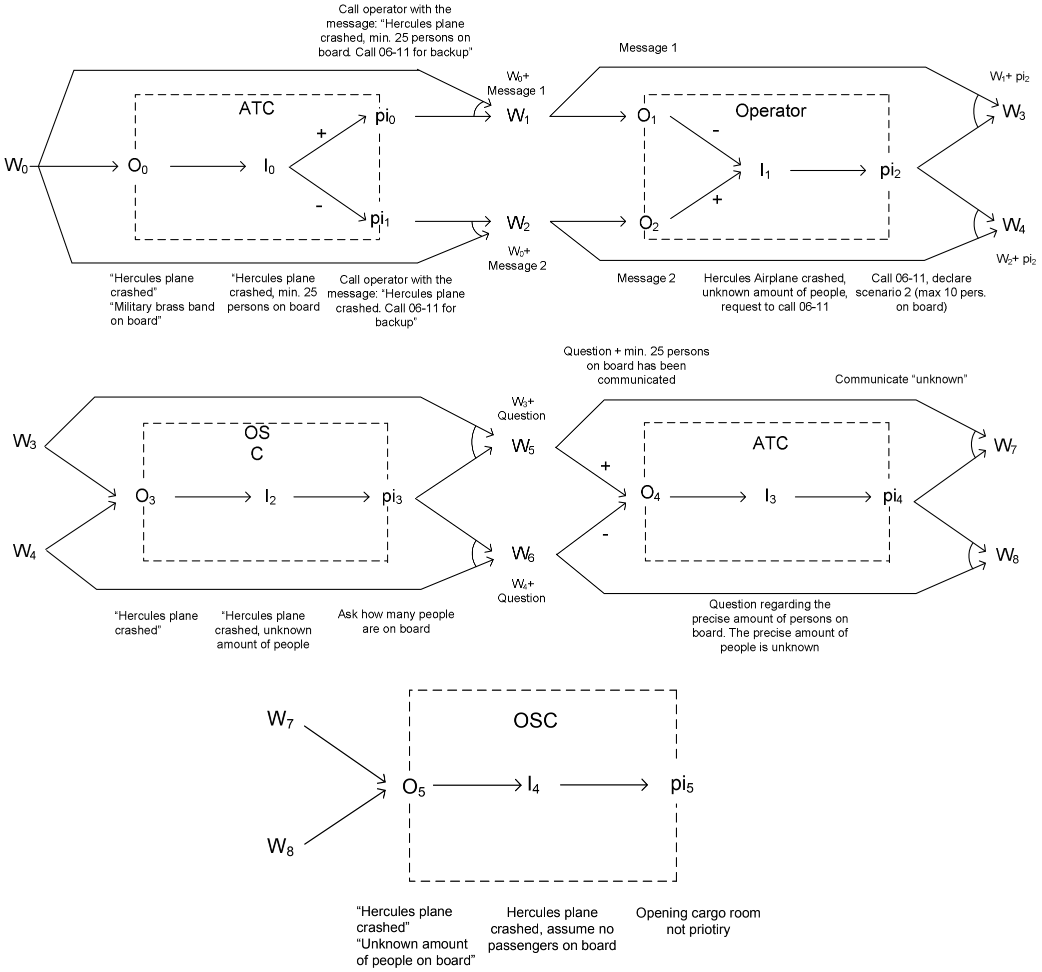

Fig. 2. Interpretations within the Hercules disaster

4. Representing Interpretation in Default Logic

The *representation problem* for a nonmonotonic logic is the question whether a given set of possible outcomes of a reasoning process can be represented by a theory in this logic. More specifically, representation theory indicates what are criteria for a set of possible outcomes, for example, given by a collection of deductively closed sets of formulae, so that this collection can occur as the set of outcomes for a theory in this nonmonotonic logic. In (Marek, Treur and Truszczynski, 1997) the representation problem is solved for default logic, for the finite case. Given this context, in the current paper Default Logic is chosen to represent interpretation processes. For the empirical material analyzed, default theories have been specified such that their extensions are the possible interpretations.

A *default theory* is a pair **⟨D, W⟩**. Here **W** is a finite set of logical formulae (called the background theory) that formalize the facts that are known for sure, and **D** is a set of default rules. A default rule has the form: $\boldsymbol{\alpha}$**:** $\boldsymbol{\beta_1, .., \beta_n / \gamma}$. Here $\boldsymbol{\alpha}$ is the precondition, it has to be satisfied before considering to believe the conclusion $\boldsymbol{\gamma}$, where the $\boldsymbol{\beta}$s, called the justifications, have to be consistent with the derived information and **W**. As a result $\boldsymbol{\gamma}$ might be believed and more default rules can be applied. However, the end result (when no more default rules can be applied) still has to be consistent with the justifications of all applied default rules. For convenience we only consider **n = 1**. Moreover, in the examples, *normal default theories* will be used: based on defaults of the form $\boldsymbol{\alpha}$**:** $\boldsymbol{\beta / \beta}$. For more details on Default Logic, such as the notion of extension, see, e.g., (Reiter, 1980; Marek and Truszczynski, 1993). For the possible interpretations presented in the previous section (for which the formalization has been shown), the following Default Theory has been specified.

Set of defaults D

{observation(plane_crash, pos) **:** belief(plane_crash_occurred, pos) /
belief(plane_crash, pos) **}**
{observation(plane_crash, pos) ∧ observation(cargo_plane, pos) ∧
observation(passengers_on_board, pos) **:**
belief(passenger_count(more_than_25), pos) /
belief(passenger_count(more_than_25), pos) **}**
{observation(plane_crash, pos) ∧ observation(cargo_plane, pos) ∧
¬observation(passengers_on_board, pos) **:**
belief(passenger_count (maximum_4), pos) /
belief(passenger_count (maximum_4), pos) **}**
{observation(plane_crash, pos) ∧ observation(cargo_plane, pos) ∧ \
¬observation(passengers_on_board, pos) **:**
belief(passenger_count (unknown), pos) /
belief(passenger_count (unknown), pos) **}**
{belief(plane_crash_occurred, pos) **:**
action(communicate_to(plane_crash, operator), pos) /
action(communicate_to(plane_crash, operator), pos) **}**
{belief(plane_crash_occurred, pos) ∧
belief(passenger_count(PN:PASSENGER_NUMBER), pos) **:**
action(communicate_to(passenger_count(PN:PASSENGER_NUMBER), operator), pos) /
action(communicate_to(passenger_count(PN:PASSENGER_NUMBER), operator), pos) **}**
{belief(plane_crash_occurred, pos) **:**
¬action(communicate_to(passenger_count(PN:PASSENGER_NUMBER), operator), pos) /
¬action(communicate_to(passenger_count(PN:PASSENGER_NUMBER), operator), pos)**}**
{belief(plane_crash_occurred, pos) ∧ belief(passenger_count(more_than_25), pos) **:**
action(communicate_to(call_backup_via_06-11, operator), pos) /
action(communicate_to(call_backup_via_06-11, operator), pos) **}**

Background theory W

observation(plane_crash, pos).
observation(cargo_plane, pos).
observation(passengers_on_board, pos).
belief(passenger_count (unknown), pos) →
¬belief(passenger_count (maximum_4), pos) ∧
¬belief(passenger_count(more_than_25), pos)
belief(passenger_count (maximum_4), pos) →
¬belief(passenger_count (unknown), pos) ∧
¬belief(passenger_count(more_than_25), pos)
belief(passenger_count (more_than_25), pos) →
¬belief(passenger_count (unknown), pos) ∧
¬belief(passenger_count(maximum_4), pos)
action(communicate_to(passenger_count (unknown), operator), pos) →
¬action(communicate_to(passenger_count (maximum_4), operator), pos) ∧
¬action(communicate_to(passenger_count(more_than_25), operator), pos)
action(communicate_to(passenger_count (maximum_4), operator), pos) →
¬action(communicate_to(passenger_count (unknown) , operator), pos) ∧
¬action(communicate_to(passenger_count(more_than_25), operator), pos)
action(communicate_to(passenger_count (more_than_25), operator), pos) →
¬action(communicate_to(passenger_count (unknown), operator), pos) ∧
¬action(communicate_to(passenger_count(maximum_4), operator), pos)

It has been checked automatically that the default theory above is appropriate, using SModels, a system for answer set programming in which a specification can be written in (an extended form of) logic programming notation. The notation includes simple statements such as *a.* which states that the atom a is true. Furthermore, rules are specified as *a :- b.* which states that if b holds, a will hold as well. Reasoning by means of a closed world assumption is supported as well by means of the not, e.g. *a :- not b* means that in case b is not derived, *a* can be derived. Finally, explicit negations are noted by the ’-‘. Result of running an SModels program is a set of stable models. The translation of the default rules specified above to an SModels specification is straightforward. A default of the form {$a_1 \wedge ... \wedge a_n$**:**b/b} can be represented in SModels in the following way: *b :- a_1 , ..., a_n, not -b*. The last element of the rule represents the fact that the opposite has not been derived. Strict constraints of the form a → b are included by simply adding *b :- a* to the specification. The following two stable models are found by SModels, which indeed correspond to the two intended interpretations. Note that in this case, the computation time needed to output these models is limited, namely within 1 millisecond after having read the input file.

```
smodels version 2.26. Reading...done
Answer: 1
Stable Model:
   observation(passengers_on_board,pos)
   observation(cargo_plane,pos)
   observation(plane_crash,pos)
   belief(plane_crash_occurred,pos)
   belief(passenger_count(more_than_25),pos)
   -belief(passenger_count(maximum_4),pos)
   -belief(passenger_count(unknown),pos)
   action(communicate_to(plane_crash,operator),pos)
   action(communicate_to(call_backup_via_06_11,operator),pos)
   action(communicate_to(passenger_count(more_than_25),operator),pos)
   -action(communicate_to(passenger_count(maximum_4),operator),pos)
   -action(communicate_to(passenger_count(unknown),operator),pos)
Answer: 2
Stable Model:
   observation(passengers_on_board,pos)
   observation(cargo_plane,pos)
   observation(plane_crash,pos)
   belief(plane_crash_occurred,pos)
   belief(passenger_count(more_than_25),pos)
   -belief(passenger_count(maximum_4),pos)
   -belief(passenger_count(unknown),pos)
   action(communicate_to(plane_crash,operator),pos)
   action(communicate_to(call_backup_via_06_11,operator),pos)
   -action(communicate_to(passenger_count(more_than_25),operator),pos)
   -action(communicate_to(passenger_count(maximum_4),operator),pos)
   -action(communicate_to(passenger_count(unknown),operator),pos)
```

5. Simulation by Temporalized Default Rules

In this section, a generic simulation model for default reasoning is specified (based on the executable temporal language LEADSTO; cf. Bosse et al., 2007), and applied to the case study. As discussed in the Section 3, to formalize one reasoning trace in a multiple interpretation situation, a certain selection has to be made, based on control knowledge which serves as a parameter for the interpretation to be achieved. Variants of Default Logic in which this can be expressed are Constructive Default Logic (Tan and Treur, 1992) and Prioritized Default Logic (Brewka, 1994; Brewka and Eiter, 1999). A *Prioritized Default Theory* is a triple **⟨D,W, <⟩**, where **⟨D,W⟩** is a Default Theory and **<** is a strict partial order on **D**. *Constructive Default Logic*, see (Tan and Treur, 1992), is a Default Logic in which selection functions are used to control the reasoning process. Selection functions take the set of consequents of possibly applicable defaults and select one or a subset of them. A selection function can represent one of the different ways to reason from the same set of defaults, and thus serves as a parameter for different reasoning traces (achieving different interpretations). This knowledge determines a selection operator (see Section 3).

The generic simulation model for default reasoning described below is an executable temporal logical formalization of Constructive Default Logic, based on the temporal perspective on default and nonmonotonic reasoning as developed in (Engelfriet and Treur, 1998). The input of the model is (1) a set of normal default rules, (2) initial information, and (3) knowledge about the selection of conclusions of possibly applicable rules. The output is a trace which describes the dynamics of the reasoning process over time. Globally, the model can be described by a generate-select mechanism: first all possible (default) assumptions (i.e., candidate conclusions) are generated, then one conclusion is selected, based on selection knowledge. Such selection knowledge could, e.g., also reflect the probability of particular occurrences. After selection, the reasoning process is repeated. In the language LEADSTO, the generic default reasoning model can be described by the following local dynamic properties (LPs):

LP1 Candidate Generation

If I have derived (x,s1), and I have a default rule that allows me to assume (y,s2), and I do not have any information about the truth of y yet, then (y,s2) will be considered a possible assumption.

∀x,y:info_element ∀s1,s2:sign
derived(x, s1) ∧ default_rule(x, s1, y, s2, y, s2) ∧ not derived(y, pos) ∧
not derived(y, neg) → possible_assumption(y, s2)

Note that the sort sign consists of the elements pos and neg.

LP2 Candidate Comparison

Each possible assumption is a better (or equally good) candidate than itself.

∀x:info_element ∀s:sign
possible_assumption(x, s) → better_candidate_than(x, s, x, s)

If (x,s1) is a possible assumption, and (y,s2) is no possible assumption, then (x,s1) is a better candidate than (y,s2).

∀x,y:info_element ∀s1,s2:sign
possible_assumption(x, s1) ∧ not possible_assumption(y, s2) →
better_candidate_than(x, s1, y, s2)

If (x,s1) is a possible assumption, and (y,s2) is a possible assumption, and it is known that deriving (x,s1) has priority over deriving (y,s2), then (x,s1) is a better candidate than (y,s2).

∀x,y:info_element ∀s1,s2:sign
possible_assumption(x, s1) ∧ possible_assumption(y, s2) ∧
priority_over(x, s1, y, s2) → better_candidate_than(x, s1, y, s2)

LP3 Candidate Selection

If (x,s1) is a possible assumption, and it is the best candidate among all possible assumptions, then it will be derived.

∀x:info_element ∀s1:sign
possible_assumption(x, s1) ∧ [∀y:info_element ∀s2:sign
better_candidate_than(x, s1, y, s2)] → derived(x, s1)

LP4 Persistence

If (x,s) is derived, then this will remain derived.

∀x:info_element ∀s:sign
derived(x, s) → derived(x, s)

The generic default reasoning model described has been used to simulate the reasoning process as performed by the Air Traffic Controller in the Hercules disaster (see Section 2). An example simulation trace is shown in Figure 3. In this figure, time is on the horizontal axis, and different states are on the vertical axis. A dark box on top of a line indicates that a state property is true; a light bow below a line indicates that it is false. As shown in Figure 3, there are initially three important aspects of the world: the fact that there is a plane crash, that it involves a cargo plane, and that there are passengers on board. At time point 1, the ATC correctly observes these three information elements. Next, he starts the interpretation process: according to his default rules, he generates two possible assumptions: there is a plane crash, and the passenger count is over

25. Based on his selection knowledge, first the former assumption is derived (time point 4: derived(belief(plane_crash, pos), pos)). As the latter possible assumption does not conflict with the former, the possible assumption that the passenger count is over 25 is derived as well (see time point 11). Next, the ATC generates four possible assumptions on actions: (1) communicating that there is a plane crash, (2) communicating that the emergency number 06-11 should be called, (3) communicating that the passenger count is over 25, and (4) *not* communicating that the passenger count is over 25. The first two possible actions are translated to actions; after that, the ATC selects the conclusion *not* communicating the passenger count over the conclusion for communicating the passenger count; thus, this information does not reach the operator.

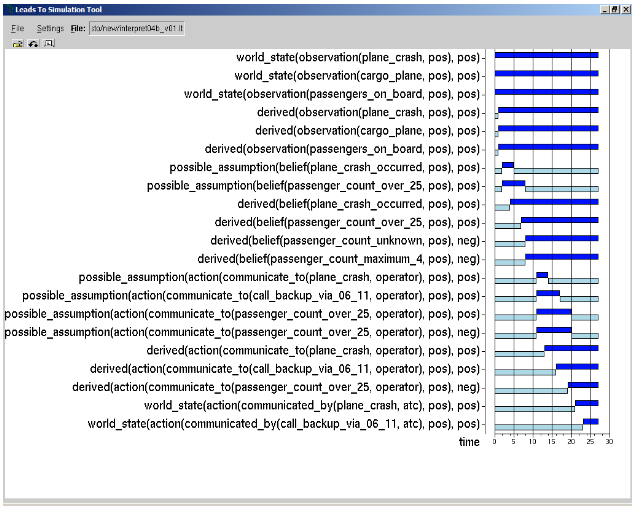

Figure 3. Simulation trace of the reasoning of the ATC

It is important to note that the trace shown in Figure 3 corresponds to one possible course of affairs. This means that it corresponds to one path through Figure 1, which is in this case the path W_0 - O_0 - I_0 - pi_1 - W_2. In default reasoning terms, the trace eventually results in one extension for the set of default rules shown in Section 3. By changing the selection knowledge, different extensions are generated. Although in this paper only one partial example is shown (due to space limitations), the complete reasoning processes of four different parties involved in the Hercules disaster have been modeled. Moreover, for all of these reasoning processes, all different settings of selection knowledge have systematically been selected. This way, a large number of traces have been generated, which together cover all possible reasoning traces based on multiple interpretations for this domain, including the (non-optimal) ones reported in the empirical material.

6. Verification of Properties for Traces

This section addresses the automated verification of properties against two types of traces. First of all, traces that include full information are addressed. In these traces, the interpretation of the particular agent under analysis is available as well as the observations and actions performed by the agent. The second type of trace addressed is a trace merely consisting of the external information (i.e. observations and actions). Note that all of these properties are specified independent of the specific case study, and can therefore easily be reused.

6.1 Analysis of Complete Traces

Verification of a simulated or empirical default reasoning trace including complete information can address a number of aspects. First it can address whether all conclusions in the trace are grounded by justified application of default rules. Next it can be verified whether the process has been exhaustive, i.e., whether for all applicable default rules the conclusion occurs. These properties have been given a temporal form (in the spirit of Engelfriet and Treur, 1998), and specified in the temporal predicate logical language TTL cf. (Bosse et al., 2006). All of these properties have been checked automatically and shown to be satisfied for traces as the one presented in Figure 3, using the TTL Checker environment.

groundedness(γ:TRACE):
∀t:TIME, i:info_element, s:sign
[state(γ, t) |= derived(i, s) ⇒ grounded (γ,i,s,t)]

grounded(γ:TRACE, i:info_element, s:sign, t:TIME):
[follows_from_default(γ,i,s,t) ∨ follows_from_strict_constraint(γ,i,s,t) ∨
world_fact(γ,i,s,t)]

world_fact(γ:TRACE, i:info_element, s:sign, t:TIME):
∃t2:TIME < t state(γ, t2) |= world_state(i, s)

follows_from_strict_constraint(γ:TRACE, i:info_element, s:sign, t:TIME):
∃C:CONJUNCTION, t2:TIME < t [state(γ, t2) |= strict_constraint(C, i, s) &
∀i2:info_element,s2:sign [element_of(i2, s2, C) ⇒
state(γ, t2) |= derived(i2, s2)]]

Note that elements of the sort CONJUNCTION refer to conjunctions of <info_element, sign> pairs.

follows_from_default(γ:TRACE, i:info_element, s:sign, t:TIME):
∃t2:TIME < t, C:CONJUNCTION
[state(γ, t2) |= default_rule(C, i, s, i, s) & ∀i1:info_element,s1:sign
[element_of(i1, s1, C) ⇒ state(γ, t2) |= derived(i1, s1)]
& ∀t3≥t ∀s'≠ s not state(γ, t3) |= derived(i, s')

consistency(γ:TRACE):
∀i:info_element, s:sign, t:TIME
[state(γ,t) |= derived(i, s) ⇒
¬∃t2:TIME, s2:sign [s ≠ s2 & state(γ,t2) |= derived(i, s2)]]

exhaustiveness(γ:TRACE):
∀t:TIME, i:info_element, s:sign, C:CONJUNCTION
[state(γ, t) |= default_rule(C, i, s, i, s) &
∀i2:info_element,s2:sign [element_of(i2, s2, C) ⇒
state(γ, t) |= derived(i2, s2)] &
¬∃t2:TIME, s3:sign [s ≠ s3 & state(γ, t2) |= derived(i, s3)]
⇒ ∃t3:TIME [state(γ, t3) |= derived(i, s)]

derived_persistency(γ:TRACE):
∀t1, t2 [state(γ, t1) |= derived(i, s) & t1<t2 ⇒ state(γ, t2) |= derived(i, s)]

These verification properties assume that all information is fully available, including the interpretation that has been derived. In empirical traces however, such information might not be present. Such information could be obtained by interviews and added to the traces, but this does not always give an adequate representation of reality, since people tend to avoid admitting mistakes in incident management. The following section shows how properties can be verified for empirical traces, without having knowledge on the interpretation. In addition, it specifies properties on correctness of interpretation based upon selection of the most specific default rule.

6.2 Analysis of Externally Observable Traces

In this section verification properties are specified assuming traces that merely consist of the observations received by the agent, and the actions that have been performed by the agent. Note that conflicting observations at the same time point are not allowed. Several different properties are identified. First of all, a derivable interpretation is defined, which is simply an interpretation that can be derived based upon the observations received, and a default rule:

derivable_int(γ:TRACE, t:TIME, C:CONJUNCTION, i:info_element, s:sign):
state(γ, t) |= default_rule(C, i, s, i, s) & ∀i2:info_element, s2:sign
[element_of(i2, s2, C) ⇒ ∃t':TIME ≤ t
[state(γ, t') |= observation(i2, s2) & ¬[∃s3:SIGN, t'':TIME ≤ t & t'' ≥ t'
[state(γ, t'') |= observation(i2, s3) & s2 ≠ s3]]]]

An interpretation is considered to be correct if it follows from the most specific default rule that can be applied:

most_specific_int(γ:TRACE, t:TIME, i:info_element, s:sign):
∃C:CONJUNCTION [derivable_int(γ, t, C, i, s) &
∀C2:CONJUNCTION ≠ C, s2:SIGN
[derivable_int(γ, t, C2, i, s2) & s ≠ s2 ⇒ size(C2) < size(C)]]

Based upon such most specific interpretations, actions to be performed can be derived:

derivable_ac(γ:TRACE, t:TIME, C:CONJUNCTION, i:info_element, s:sign):
state(γ, t) |= default_rule(C, i, s, i, s) & ∀i2:info_element, s2:sign
[element_of(i2, s2, C) ⇒ most_specific_int(γ, t, i2, s2)]

An action is considered to be correct in case it follows from the most specific default rule that is applicable:

most_spec_ac(γ:TRACE, t:TIME, i:info_element, s:sign):
∃C:CONJUNCTION
[derivable_ac(γ, t, C, i, s) & ∀C2:CONJUNCTION ≠ C, s2:SIGN
[derivable_ac(γ, t, C2, i, s2) & s ≠ s2 ⇒ size(C2) < size(C)]]

Given the fact that it can now be derived what the correct actions are, properties can be verified against empirical traces to investigate the performance shown in that empirical trace. A first property which can be verified is whether the correct actions have been performed in the empirical trace without taking too much time to start the performance of this action (i.e. within duration d):

correct_action(γ:TRACE, t:TIME, i:info_element, s:sign, d):
[most_spec_ac(γ, t, i, s) &
[¬∃t':TIME < t most_spec_ac(γ, t', i, s)] &
[¬∃t'':TIME > t & t'' < t + d ¬most_spec_ac(γ, t'', i, s)]]
⇒ ∃t''':TIME ≥ t & t'''≤ t + d [state(γ, t''') |= world_state(i, s)]

Of course, things do not necessarily run so smoothly, therefore, detection of errors is of crucial importance. An error first of all occurs when an action is not performed that should have been performed according to the correct interpretation:

missing_action(γ:TRACE, t:TIME, i:info_element, s:sign, d):
most_spec_act(γ, t, i, s) &
[¬∃t':TIME < t most_spec_ac(γ, t', i, s)] &
[¬∃t'':TIME > t & t'' < t + d ¬most_spec_ac(γ, t'', i, s)] &
[¬∃t''':TIME ≥ t & t'''≤ t + d [state(γ, t''') |= world_state(i, s)]

Furthermore, an error occurs when an action can be performed that is not derivable from the correct interpretation:

incorrect_action(γ:TRACE, t:TIME, i:info_element, s:sign, d):
state(γ, t) |= world_state(i, s) &
¬∃t':TIME ≤ t & t' ≥ t – d [most_spec_ac(γ, t', i, s)]

The properties specified above have been automatically verified against the empirical trace of the Hercules disaster. The analysis shows that the correct_action property is not satisfied for the Hercules disaster trace, due to the fact that the trace does not show that the ATC has passed the information on the number of people on board of the plane. As a result, the missing_action property holds. Finally, the incorrect_action property is not satisfied, as only missing actions occur in the trace. These results comply to the human analysis of the Hercules disaster.

7. Conclusion

This paper shows how a number of known techniques and tools developed within the area of nonmonotonic logic and AI can be applied to analyze empirical material on human reasoning and interpretation within Cognitive Science; cf. (Stenning and van Lambalgen, 2006). The formal techniques exploited in the empirical analysis approach put forward are:

(1) multi-interpretation operators as an abstract formalization of reasoning towards an interpretation,
(2) default logic to specify a multi-interpretation operator,
(3) a temporalized default logic to specify possible reasoning traces involved in a multi-interpretation process,
(4) an executable temporal logical language to specify a generic executable default reasoning model to simulate such reasoning traces, and
(5) an expressive temporal logical language to specify and verify properties for reasoning traces

As such, this work synergizes the *protocol analysis* tradition of (Ericsson and Simon, 1993), which addresses elicititation of verbal reports from research participants, with the *model checking* tradition introduced by e.g. (Huth and Ryan, 2004) , which addresses verification of behavioural properties against formal specifications. It has been shown how indeed the introduced techniques and tools obtain an adequate formalization and analysis of empirical material on human reasoning in critical situations in incident management. Two types of empirical material have been used. First of all, training material which describes the procedures to be followed, being the basis for the default theory regarding the human reasoning. As a result of this default theory, simulated traces have been generated, and have been compared to the given empirical traces (based upon disaster reports, the second type of empirical material). It has been shown that these traces can accurately model human reasoning (i.e. the traces match with the human reasoning reported in the disaster reports), including errors that might occur in the process. Note that the generation of simulation traces using the formalized training material can bring to light flaws in procedures as well. It might for example be the case that a wrong procedure can be chosen because the conditions for selecting such a procedure are not detailed enough. As a result, errors in the incident management process might show up, which can be seen in the generated simulation traces. Regarding the detections of errors in such reasoning processes, relevant properties of both simulation as well as empirical traces have been verified and results were shown of this verification process, thereby identifying reasoning errors. The properties and default rules presented in this paper have all been specified in a generic fashion, such that they can easily be reused for studying other cases. Therefore the modeling effort of this first case study (which involves a significant amount of effort) is expected to reduce as the knowledge part of this case study can greatly be reused.

The presented approach can be used to enable automated detection of interpretation errors in incident management. Such detection could potentially avoid unwanted chains of events which might result in catastrophic consequences. Such a goal is quite ambitious, and makes rather strong assumptions about the ability to, for example, analyze human communication real time. A more feasible goal on the short term is to analyze historical cases and to formalize the current procedures using default logic and generate simulation results for particular accidents, thereby analyzing the correctness of these procedures. As a first case study to investigate the suitability of the presented approach for this purpose, the Hercules disaster has been used, showing promising results. The disaster is representative for many of the disasters that occur. It is however future work to perform a more thorough evaluation, using a variety of cases.

An important issue related to the approach presented in this paper is its scalability. Of course, in case of a huge incident management organization, calculating all possible interpretations of the entire combination will be difficult. The idea is however that only the interpretations are generated that are useful in the particular situation. Hereby certain selection knowledge can be used to for example choose the most appropriate default rules. Using such a selection greatly reduces the number of options, and hence, makes the approach more scalable.

When performing a more thorough evaluation as mentioned above, in addition to the use of formal analysis techniques for the purpose of verification, more emphasis will be placed on formal methods for the purpose of protocol analysis. Whereas the current paper assumes that the ontology to formally express the verbal reports of a case study is given, future work will also address the question how to construct such an ontology, and how to map parts of the ontology to fragments of the verbal reports. To this end, different formal protocol analysis techniques will be investigated and compared. In the last decades, more interest is being paid to the application of formal methods to protocol analysis, see, e.g., (Meadows, 2003). For the current purposes, it will be useful to explore to what extent existing formal methods to protocol analysis can be reused. For example, van Langevelde and Treur (1991) propose a formal framework that can be used to analyze complex reasoning tasks, by decomposing the task into a number of primitive subtasks, which can be specified using standard logics. Another promising approach is put forward by Bosse, Jonker, and Treur (2006), who describe an approach to formalize and analyze the dynamics of assumption-based reasoning processes.

Note that the executable temporal logical language LEADSTO, which was used for simulation in Section 5, is not the only language that can be used for this purpose. Also other languages and tools are suitable, such as SModels, a system for answer set programming in which a specification can be written in (an extended form of) logic programming notation, see (Niemelä et al., 2000).

An approach to interpretation processes different from the one based on nomonotonic logic as adopted here, is by abductive inference, see e.g. (Josephson and Josephson 1996). For future research it will be interesting to explore the possibilities of abductive inference to model interpretation processes in comparison to nonmonotonic logic approaches.

References

Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A., and Treur, J. (2006). Specification and Verification of Dynamics in Cognitive Agent Models. In: Proc. of the 6th Int. Conf. on Intelligent Agent Technology, IAT'06. IEEE Computer Society Press, 2006, pp. 247-254.

Bosse, T., Jonker, C.M., Meij, L. van der, and Treur, J., (2007). A Language and Environment for Analysis of Dynamics by Simulation. *International Journal of Artificial Intelligence Tools*, vol. 16, 2007, pp. 435-464.

Bosse, T., Jonker, C.M., and Treur, J. (2006). Formalization and Analysis of Reasoning by Assumption. *Cognitive Science Journal*, volume 30, issue 1, 2006, pp. 147-180.

Braine, M.D.S., and O'Brien, D.P. (eds.) (1998). *Mental Logic*. Lawrence Erlbaum, London.

Brewka, G. (1994), Adding priorities and specificity to default logic. In: MacNish, C., Pereira, L., and Pearce, D., (eds.), Proc. of JELIA'94, LNAI, vol. 838. Springer Verlag, pp. 247-260.

Brewka, G., and Eiter, T. (1999), Prioritizing Default Logic: Abridged Report. In *Festschrift on the occasion of Prof.Dr. W. Bibel's 60th birthday*. Kluwer.

Engelfriet, J., Herre, H., and Treur, J. (1998), Nonmonotonic Reasoning with Multiple Belief Sets, *Annals of Mathematics and Artificial Intelligence,* vol. 24, pp. 225-248.

Engelfriet, J., and Treur, J. (1998), An Interpretation of Default Logic in Minimal Temporal Epistemic Logic. *Journal of Logic, Language and Information*, vol. 7, pp. 369-388.

Engelfriet J., and Treur J. (2000). Specification of Nonmonotonic Reasoning. *Journal of Applied Non-Classical Logics*, vol. 10, 2000, pp. 7-27.

Engelfriet, J., and Treur, J. (2003), Multi-Interpretation Operators and Approximate Classification. *Int. Journal of Approximate Reasoning*, vol. 32, pp. 43-61.

Ericsson K.A., and Simon, H.A. (1993), Protocol analysis; Verbal reports as data (revised edition). Cambridge, MA: Bradford books/MIT Press.

Huth, M., and Ryan, M. (2004), Logic in computer science:modeling and reasoning about systems. Cambridge, UK, Cambridge University Press.

Johnson-Laird, P.N. (1983). *Mental Models*. Cambridge: Cambridge University Press.

Johnson-Laird, P.N., and Byrne, R.M.J. (1991). *Deduction*. Hillsdale, NJ:Erlbaum.

Josephson, J.R. and Josephson, S.G. (eds.) (1996), Abductive Inference: Computation, Philosophy, Technology. New York: Cambridge University Press

Langevelde, I.A. van, and Treur J. (1991). Logical Methods in Protocol Analysis. In: M. Linster, B. Gaines (eds.), Proceedings of the European Knowledge Acquisition Workshop, EKAW '91, GMD-Studien 211, pp. 162-183.

Marek, V.W., and Truszczynski, M. (1993), *Nonmonotonic Logics*. Springer Verlag.

Marek, V.W., Treur, J., and Truszczynski, M. (1997), Representation Theory for Default Logic. *Annals of Mathematics and AI*, vol. 21, pp. 343-358.

Meadows, C. (2003). Formal methods for cryptographic protocol analysis: Emerging issues and trends. IEEE Journal on Selected Areas in Communication, vol. 21, issue 1, 2003, pp. 44-54.

Ministry of the Interior (1996), *Airplane Crash Airbase Eindhoven 15th July 1996* (in Dutch), SDU, The Hague.

NIBRA (2001), *Fire Watch First Class: Airplane Fire Fighting (in Dutch).*

Niemelä, I., Simons, P., and Syrjänen, T. (2000). Smodels: a system for answer set programming. In: Proceedings of the 8th International Workshop on Non-Monotonic Reasoning, Breckenridge, Colorado, USA, April 2000.

Rips, L.J. (1994). *The Psychology of Proof: Deductive reasoning in human thinking*. MIT Press, Cambridge, Mass.

Stenning, K., and van Lambalgen, M. (2006), A working memory model of relations between interpretation and reasoning. *Cognitive Science Journal*, Elsevier Science Inc., Oxford, UK. In press.

Reiter, R. (1980) A logic for default reasoning. Artificial Intelligence, 13:81-132.

Tan, Y.H., and Treur, J. (1992) Constructive Default Logic and the Control of Defeasible Reasoning. In: B. Neumann (ed.), *Proc. ECAI'92*. Wiley and Sons, 1992, pp. 299-303.