# A formal method to analyse human reasoning and interpretation in incident management

Tibor Bosse and Mark Hoogendoorn*

Department of Artificial Intelligence
Vrije Universiteit Amsterdam
De Boelelaan 1081a
1081 HV Amsterdam, The Netherlands
E-mail: tbosse@cs.vu.nl
E-mail: mhoogen@cs.vu.nl
*Corresponding author

Catholijn M. Jonker

Department of Mediametics
Delft University of Technology
Mekelweg 4
2628 CD Delft, The Netherlands
E-mail: catholijn@mmi.tudelft.nl

Jan Treur

Department of Artificial Intelligence
Vrije Universiteit Amsterdam
De Boelelaan 1081a
1081 HV Amsterdam, The Netherlands
E-mail: treur@cs.vu.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 the recent literature within cognitive science, the means taken from the area of nonmonotonic logic are proposed to analyse the latter aspect of human reasoning. In this paper, this claim is further worked out and tested against the empirical material of human reasoning during critical situations (incident management). Empirical and simulated reasoning traces have been analysed by comparing them and by automatically checking their properties.

**Keywords:** formal methods; human reasoning; interpretation.

**Reference** to this paper should be made as follows: Bosse, T., Hoogendoorn, M., Jonker, C.M. and Treur, J. (2008) 'A formal method to analyse human reasoning and interpretation in incident management', *Int. J. Emergency Management*, Vol. 5, Nos. 1/2, pp.164–192.

**Biographical notes:** Tibor Bosse is an Assistant Professor of Artificial Intelligence at the Vrije Universiteit Amsterdam. He obtained his PhD in Artificial Intelligence in 2005, graduating on the topic 'Analysis of the Dynamics of Cognitive Processes'. From 2002, his research has mainly focused on the integration of artificial intelligence and cognitive science. With his colleagues, he has developed a methodology to analyse and model the dynamics of cognitive processes. This methodology makes use of a hybrid language called TTL, which is based on a combination of symbolic and mathematical constructs and used to develop integrative models comprising both knowledge and agent aspects and dynamical systems. In addition to artificial intelligence and cognitive science, he also applied this methodology within various other disciplines, including biology, human factors and philosophy of mind. This has led to over 80 publications in various international conferences and journals.

Mark Hoogendoorn is an Assistant Professor at the Vrije Universiteit Amsterdam (Department of Artificial Intelligence). From August 2007 until September 2007, he was a Visiting Researcher at the Department of Computer Science and Engineering of the University of Minnesota. He obtained his PhD degree from the Vrije Universiteit Amsterdam in 2007. In his PhD research, he focused on organisational change within multi-agent systems, applying his research in projects in various domains, including incident management, logistics and the naval domain. His current research interests include multi-agent systems, cognitive modelling and ambient intelligence.

Catholijn Jonker (1967) is a Full Professor of Man-Machine Interaction at the Faculty of Electrical Engineering, Mathematics and Computer Science of the Delft University of Technology, the Netherlands. She studied Computer Science and did her PhD studies at Utrecht University. After a postdoctorate position in Bern, Switzerland, she became an assistant (later Associate) Professor at the Department of Artificial Intelligence of the Vrije Universiteit Amsterdam. From September 2004 until September 2006, she was a Full Professor of Artificial Intelligence/Cognitive Science at the Nijmegen Institute of Cognition and Information of the Radboud University Nijmegen. She chaired the De Jonge Akademie (Young Academy) of the Royal Netherlands Society of Arts and Sciences (KNAW) in 2005 and 2006 and she is a member of the same organisation from 2005 through 2010. Her recent publications address cognitive processes and concepts such as trust, negotiation and the dynamics of individual agents and organisations.

Professor Dr. Jan Treur has been a Full Professorship in Artificial Intelligence at the Vrije Universiteit Amsterdam since 1990. He is heading the Department of Artificial Intelligence, which consists of about 45 researchers. He is an internationally well-recognised expert in agent technology, cognitive modelling and knowledge engineering. He is or has been a member of the programme committee of many of the main conferences and workshops and many journals in these areas. His extensive list of publications (*e.g.*, see www.cs.vu.nl/~treur) covers major scientific publication media in artificial intelligence and cognitive science, including top-level conferences and journals. Some of his recent involvements are organising the First and Second International Workshop on Human Aspects in Ambient Intelligence at the European Conference on Ambient Intelligence (AmI'07) in November 2007 and the International Conference on Intelligent Agent Technology (IAT'08) in December 2008. Moreover, he recently designed a strongly multidisciplinary Bachelor study programme in Human Ambience at the Vrije Universiteit Amsterdam, combining subjects from ambient intelligence, artificial intelligence, computer science, psychology and biomedical sciences.

This paper is a significantly extended and improved version of Bosse, T., Hoogendoorn, M., Jonker, C.M. and Treur, J. (2007) 'A formal empirical method for human reasoning and interpretation', in R.L. Lewis, T.A. Polk and J.E. Laird (Eds.) *Proceedings of the 8th International Conference on Cognitive Modeling, ICCM'07*, Taylor and Francis, pp.241–246.

## 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 (in press), it was shown how the empirical study of human reasoning processes has been too dominated by an emphasis on classical, deductive logic. This applies equally well to the so-called rule-based or syntactic stream (*e.g.*, Braine and O'Brien, 1998; Rips, 1994), as well as the model-based or semantic stream (*e.g.*, Johnson-Laird, 1983; Johnson-Laird and Byrne, 1991). In their analysis of human reasoning, they claimed that much more important than the question of whether reasoning should be considered from a syntactical or semantical perspective is the distinction between:

- reasoning *towards* an interpretation
- reasoning *from* an interpretation.

The latter type of reasoning is reasoning within an already unambiguously determined formalised frame and can be analysed 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 proposed nonmonotonic logic as a more adequate analysis tool for the human reasoning processes. Within nonmonotonic logic, it is possible to formalise the reasoning processes that deal with multiple possible outcomes, which can be used 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 and 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. The 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 using 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 formalisation of a reasoning process leading to multiple interpretations and Section 4 shows how default logic can be used to specify such processes. To obtain a simulation of such reasoning, variants of default logic are considered, in which control decisions can be represented. To this end, in Section 5, a temporalised 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 formalised and checked. Section 7 presents the conclusions.

## 2 The incident management domain

The domain of incident management is characterised by people working under severe pressure in which split-second decisions have to be made, which can have a huge impact on the success 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 organisations.

One well-known example of the erroneous functioning of an incident management organisation 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 and making the plane tilt to one side. As a result, the plane, carrying a military brass band in the cargo room and a crew of four people, crashed on the runway and caught fire. The Air Traffic Controller (ATC) immediately hit the alarm button, also having knowledge that a military brass band is on the plane. Afterwards, he claimed to have informed the alarm centre operator of this fact who, in turn, stated never to have received the information. As a result, the operator did inform the firefighters, but declared the wrong scenario (merely for the crew on board). After the firefighting forces had arrived at the scene, one of them contacted the ATC, asking how many people are on the plane. Since the ATC reasoned under the assumption that the message of a military brass band being on board had been passed on to the firefighters, he answered that this is unknown, interpreting the question as a request for the exact amount of people onboard. The firefighter, therefore, assumed that only the crew was on the plane, which is not an assumption that should have been made according to their 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 min after the crash, which could have been well within 10 min and possibly saving precious lives.

## 3 Multiple interpretations

In a broad sense, reasoning towards an interpretation can be viewed as an activity where an agent, given some initial information (or set of beliefs) $X$, performs some manipulation of this information and arrives at a new state with different information. A (partial) view of a situation (in the domain the agent is reasoning about) is transformed to another partial view. In general, the mechanism may be nondeterministic, 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 *the 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 behaviour of the agent's reasoning process

2 *the 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 that the specification at the lower level gives in some sense a refinement or specialisation of the specification at the higher level (as in the case of conventional software specifications at different levels of abstraction). Given the specifications of the 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 specifications.

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 can still 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. The choice to apply one of these two reasoning steps indicates a branching point for the reasoning process. This is an element common for practically all the 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 formalised at the more abstract level as follows. A particular interpretation for a given set of formulae considered as input information for reasoning is formalised as another set of formulae that, in one way or the other, is derivable from the input information (the 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 considered as input information (*i.e.*, the output of the reasoning towards an interpretation) is formalised as a collection of different sets of formulae. Note that these formalisms also apply to reasoning from an interpretation. A formalisation 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 $L_1$. An interpretation is a set of propositional formulae based on a propositional language $L_2$.

*Definition 1 (multi-interpretation operator):*

- A multi-interpretation operator *MI* with input language $L_1$ and output language $L_2$ is a *function* $MI{:}\mathcal{P}(L_1) \rightarrow \mathcal{P}(\mathcal{P}(L_2))$ that assigns to each set of input facts in $L_1$ a set of sets of formulae in $L_2$.
- A multi-interpretation operator *MI* is *noninclusive* if for all $X \subseteq L_1$ and $S, T \in MI(X)$, if $S \subseteq T$, then $S = T$.

- If $L_1 \subseteq L_2$, then a multi-interpretation operator *MI* is *conservative* if for all $X \subseteq L_1$, $T \in MI(X)$, it holds $X \subseteq T$.

The condition of noninclusiveness guarantees a relative maximality of the possible interpretations. Note that when $MI(X)$ has exactly one element, this means that the set $X \subseteq L_1$ has a unique interpretation under *MI*. The notion of the multi-interpretation operator is a generalisation of the notion of a nonmonotonic belief set operator, as introduced in Engelfriet *et al.* (1998). The generalisation was introduced and applied to an 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 formalised as follows (see Engelfriet and Treur, 2003).

*Definition 2 (selection operator):*

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

It is straightforward to check that if $s{:}\mathcal{P}(\mathcal{P}(L_1)) \rightarrow \mathcal{P}(\mathcal{P}(L_2))$ 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 $C(X) = s(MI(X))$ for all $X \subseteq L_1$.

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

## 3.1 Initial air traffic controller interpretation

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

### 3.1.1 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, $O$ the observations, $I$ the interpretations and $\pi$ the actions. Note that in all the tables, the correct observations, interpretations and actions are denoted in *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 the ATC does not observe the presence of a brass band.

**Table 1** The initial observations of the air traffic controller

| *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 the plane. |
| | | ATC | $O_1$ | Observation that a Hercules plane has crashed on the runway. |

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.

**Table 2** The air traffic controller's observations leading to an interpretation

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

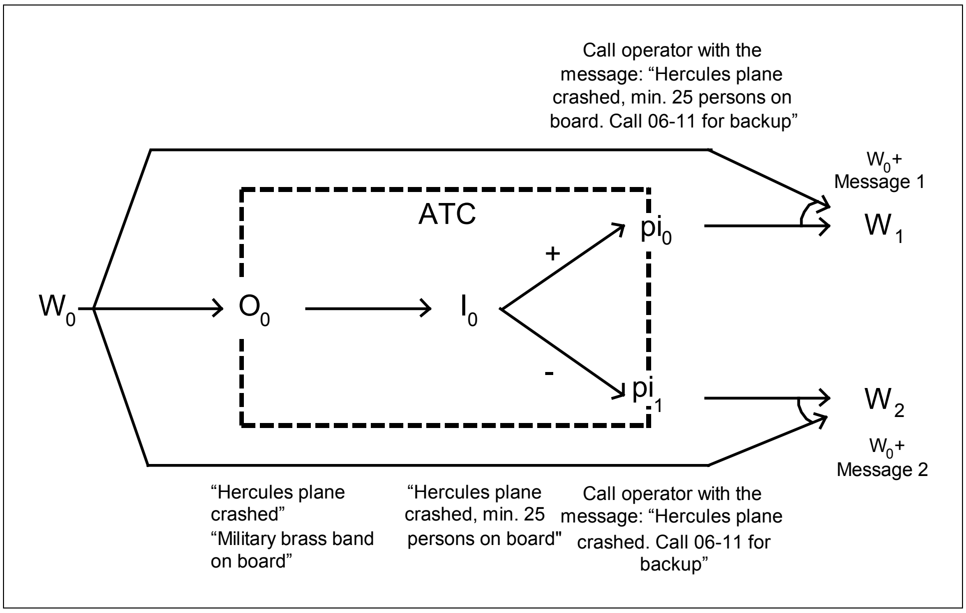
Such interpretations can lead to the actions specified in Table 3, which involve the communications with the operator.

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 onboard. 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 the actions to take ($pi_0$ and $pi_1$). There are two moments of interpretation: from observations to internal representations and from internal representations to actions.

**Table 3** The air traffic controller's interpretations leading to an action

| *Party* | *Interpretation* | *Action* | *Description* |
|---|---|---|---|
| ATC | $I_0$ | $\pi_0$ | Call operator with the message that a Hercules plane has crashed, furthermore, mention that at least 25 people are on board the plane and request to call 06–11 for backup. |
| | | $\pi_1$ | Call operator with the message that a Hercules plane has crashed, request to call 06–11 for backup. |
| | | $\pi_2$ | Call operator with the message that a Hercules plane has crashed, furthermore, mention that at least 25 people are on board the plane and request to call for backup, do not use 06–11 but call the different parties directly to avoid delays. |
| | | $\pi_3$ | 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. |
| | $I_1$ | $\pi_1$ | |
| | | $\pi_3$ | |
| | | $\pi_4$ | 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. |
| | | $\pi_5$ | 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. |
| | $I_2$ | $\pi_1$ | |
| | | $\pi_3$ | |
| | | $\pi_6$ | 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. |
| | | $\pi_7$ | 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. |
| | | $\pi_8$ | Call operator with the message that a Hercules plane has crashed, furthermore, tell him that 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 the plane. |
| | $I_3$ | $\pi_1$ | |
| | | $\pi_3$ | |
| | | $\pi_6$ | |
| | | $\pi_7$ | |
| | | $\pi_8$ | |

**Figure 1** The reasoning traces based on the interpretations




### *3.1.2 Formalisation*

Given the subset depicted in Figure 1, the following formalisation 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 true, whereas a 'neg' indicates that 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, formalised 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 The operator's interpretation*

For the operator's interpretation, the formalisation has been omitted for the sake of brevity. The informal interpretation 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.

**Table 4** The worlds leading to the operator's observations

| *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. |

**Table 4** The worlds leading to the operator's observations (continued)

| *World state* | *Description* | *Party* | *Obs* | *Description* |
|---|---|---|---|---|
| $W_7$ | $W_0 + \pi_6$ | 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. |

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

**Table 5** The operator's observations leading to an interpretation

| *Party* | *Obs* | *Interpretation* | *Description* |
|---|---|---|---|
| Operator | $O_2$, $O_3$, $O_7$, $O_9$ | $I_4$ | Scenario 3 (more than ten people involved), request to call 06-11 for backup. |
| | | $I_5$ | Scenario 2 (between three and ten people involved), request to call 06-11 for backup. |
| | | $I_6$ | Scenario 1 (less than three 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. |

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

**Table 6** The operator's interpretations leading to an action

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

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.

**Table 7** The worlds leading to the on-scene commander's observations

| *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 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 the plane. |
| $W_{11}$ | Hercules airplane crashed, operator declared Scenario 2, which was not received by anyone. | | $O_{12}$ | |
| | | | $O_{13}$ | |

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

**Table 8** The observations leading to the on-scene commander's interpretation

| *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 three to ten people on board. Request the exact amount of people on board the plane. |
| OSC | | $I_{15}$ | Hercules plane crashed (Scenario 2 applicable), therefore, between three to ten people on board. Investigate plane to see whether where these people are located. |
| OSC | $O_{14}$ | $I_{12}, I_{13}, I_{14}, I_{15}$ | |

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

**Table 9** The interpretation leading to the on-scene commander's actions

| *Party* | *Interpretation* | *Action* | *Description* |
|---|---|---|---|
| OSC | $I_{12}$ | $\pi_{12}$ | Ask the operator how many people are on the plane. |
| | | $\pi_{13}$ | Ask the ATC how many people are on the plane. |
| | $I_{13}$ | $\pi_{14}$ | Walk around the plane, look through openings and windows to see whether people are present within that part of the plane. |
| | $I_{14}$ | $\pi_{12}$, $\pi_{13}$ | |
| | $I_{13}$ | $\pi_{14}$ | |

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

## 3.4 The air traffic controller's response

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, those worlds that are mentioned in the disaster plan. In this case, these worlds only concern the communication of questions to the ATC. These are precisely two worlds:

1 $W_{10}$ with action $\pi_{13}$ from the OSC

2 $W_{11}$ with action $\pi_{13}$ of the OSC.

These are named $W_{12}$ and $W_{13}$, respectively, as shown in Table 10.

**Table 10** The worlds leading to the second set of the air traffic controller's observations

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

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

**Table 11** The observations leading to the air traffic controller's interpretations

| *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 no guarantee that the OSC heard this. The OSC asks for the amount of people on board. Therefore, he could mean to ask what the exact amount of people are 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 the OSC, yet the OSC asks for the amount of people on the plane, he wants to know an approximation. |

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

**Table 12** The second interpretation of the air traffic controller leading to actions

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

## *3.5 The on-scene commander's response*

Finally, the OSC takes the answer given by the ATC into account. Two world states are distinguished (the most likely state 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 of 'unknown' from the ATC. First, the resulting observations are shown in Table 13.

**Table 13** The second set of observations for the on-scene commander

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

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

**Table 14** The observations of the OSC leading to interpretations

| *Party* | *Obs* | *Interpretation* | *Description* |
|---|---|---|---|
| OSC | $O_{16}$ | $I_{19}$ | The amount of people on 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 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 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 the plane is not known, therefore assume worst case: a lot of people on board. |
| | | $I_{19}$, $I_{20}$, $I_{21}$ | |

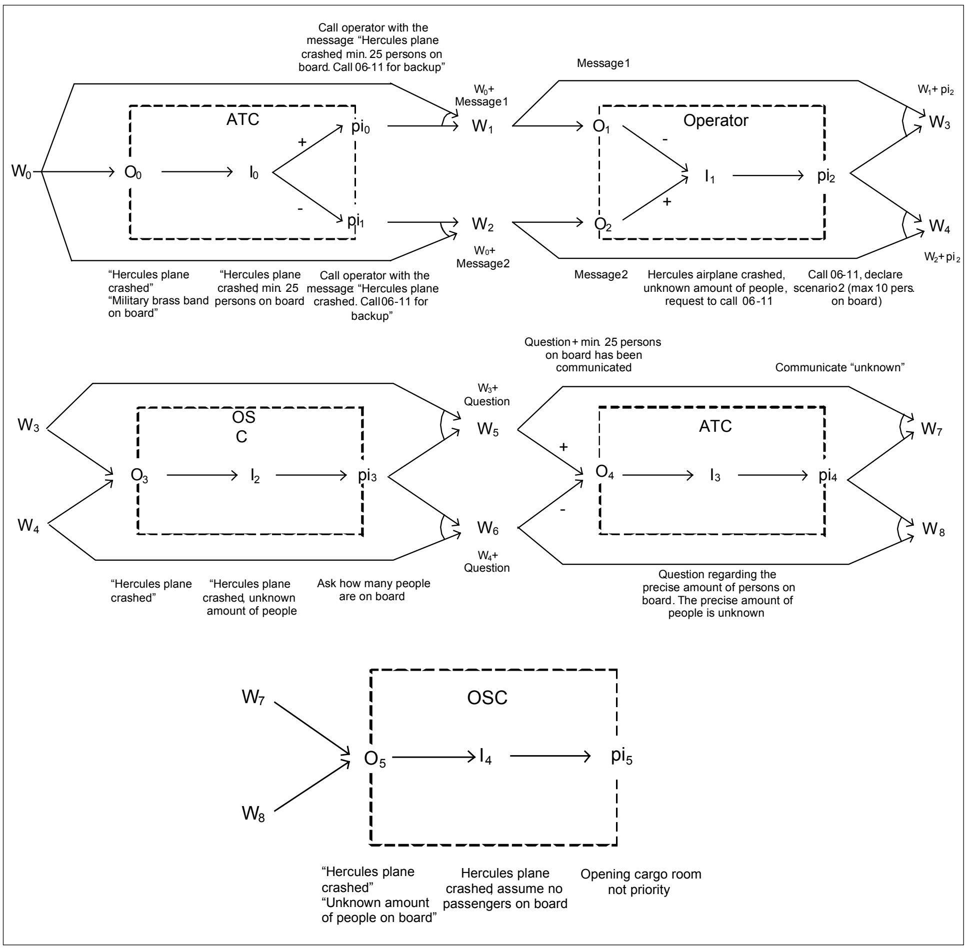
Finally, such an interpretation leads to action (Table 15).

**Table 15** The second interpretation of the on-scene commander leading to actions

| *Party* | *Interpretation* | *Action* | *Description* |
|---|---|---|---|
| OSC | $I_{19}$ | $\pi_{20}$ | After 90% knockdown, open the cargo room doors or create an entrance ASAP. |
| | | $\pi_{21}$ | After 90% knockdown, open the cockpit first, then proceed to the cargo room. |
| | $I_{20}$ | $\pi_{22}$ | Extinguish the plane, do not attempt to rescue. |
| | | $\pi_{21}$ | |
| | $I_{21}$ | $\pi_{21}$ | |
| | $I_{22}$ | $\pi_{20}$ | |
| | | $\pi_{21}$ | |

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

**Figure 2** The interpretations within the Hercules disaster




## 4 Representing interpretation in default logic

The *representation problem* for a nonmonotonic logic is the question of whether a given set of the possible outcomes of a reasoning process can be represented by a theory in this logic. More specifically, the representation theory indicates what the criteria for a set of possible outcomes are, 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 *et al.* (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 the interpretation processes. For the empirical material analysed, default theories have been specified, such that their extensions are the possible interpretations.

A *default theory* is a pair $\langle D, W\rangle$. Here, $W$ is a finite set of logical formulae (called the background theory) that formalise the facts that are known for sure and $D$ is a set of default rules. A default rule has the form $\alpha$: $\beta_1, \ldots, \beta_n/\gamma$. Here, $\alpha$ is the precondition; it has to be satisfied before considering belief in the conclusion $\gamma$, where the $\beta$s, called the justifications, have to be consistent with the derived information and *W*. As a result, $\gamma$ 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 the applied default rules. For convenience, we only consider $n = 1$. Moreover, in the examples, *normal default theories* will be used, based on the defaults of the form $\alpha$: $\beta/\beta$. For more details on default logic, such as the notion of extension, see *e.g.*, Reiter (1980) and Marek and Truszczynski (1993). For the possible interpretations presented in the previous section (for which the formalisation 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) $\rightarrow$
 ¬belief(passenger_count (maximum_4), pos) $\wedge$
 ¬belief(passenger_count(more_than_25), pos)
belief(passenger_count (maximum_4), pos) $\rightarrow$
 ¬belief(passenger_count (unknown), pos) $\wedge$
 ¬belief(passenger_count(more_than_25), pos)
belief(passenger_count (more_than_25), pos) $\rightarrow$
 ¬belief(passenger_count (unknown), pos) $\wedge$
 ¬belief(passenger_count(maximum_4), pos)
action(communicate_to(passenger_count (unknown), operator), pos) $\rightarrow$
 ¬action(communicate_to(passenger_count (maximum_4), operator), pos) $\wedge$
 ¬action(communicate_to(passenger_count(more_than_25), operator), pos)
action(communicate_to(passenger_count (maximum_4), operator), pos) $\rightarrow$
 ¬action(communicate_to(passenger_count (unknown) , operator), pos) $\wedge$
 ¬action(communicate_to(passenger_count(more_than_25), operator), pos)
action(communicate_to(passenger_count (more_than_25), operator), pos) $\rightarrow$
 ¬action(communicate_to(passenger_count (unknown), operator), pos) $\wedge$
 ¬action(communicate_to(passenger_count(maximum_4), operator), pos)

It has been automatically checked if 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 $\{a_1 \wedge ... \wedge a_n\text{:b/b}\}$ 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 $\rightarrow$ 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 ms 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 temporalised 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 Section 3, to formalise 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. The variants of default logic in which this can be expressed are constructive default logic (Tan and Treur, 1992) and prioritised default logic (Brewka, 1994; Brewka and Eiter, 1999). A *prioritised default theory* is a triple $\langle D,W,<\rangle$, where $\langle D,W\rangle$ 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 the 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 the default reasoning described below is an executable temporal logical formalisation of constructive default logic based on the temporal perspective on default and nonmonotonic reasoning, as developed in Engelfriet and Treur (1998). The inputs of the model are:

- a set of normal default rules
- initial information
- knowledge about the selection of conclusions of the 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 the 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 LEADSTO, the generic default reasoning model can be described by the following Local Dynamic Properties (LP):

- *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 described generic default reasoning model has been used to simulate the reasoning process as performed by the ATC 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 the different states are on the vertical axis. A dark box on top of a line indicates that a state property is true; a light box 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 the actions:

1 communicating that there is a plane crash

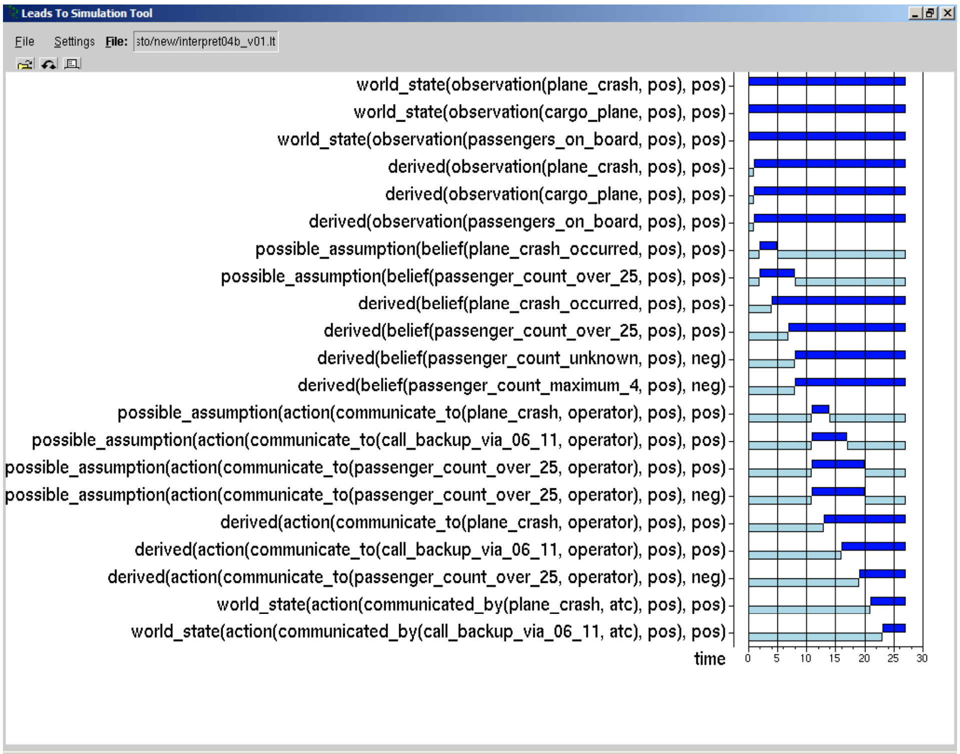
2 communicating that the emergency number 06-11 should be called

3 communicating that the passenger count is over 25

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 of *not* communicating the passenger count over the conclusion of communicating the passenger count. Thus, this information does not reach the operator.

**Figure 3** The simulation trace of the reasoning of the air traffic controller (see online version for colours)




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 only one partial example is shown in this paper (due to space limitations), the complete reasoning processes of the four different parties involved in the Hercules disaster have been modelled. Moreover, for all of these reasoning processes, all the different settings of selection knowledge have systematically been selected. This way, a large number of traces have been generated, which together cover all the possible reasoning traces based on the multiple interpretations for this domain, including the (nonoptimal) ones reported in the empirical material.

## 6 The verification of properties for traces

This section addresses the automated verification of properties against two types of traces. First of all, the 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 traces 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 The analysis of complete traces*

The verification of a simulated or empirical default reasoning trace including complete information can address a number of aspects. First, it can address whether all the conclusions in the trace are grounded by a justified application of the 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 Temporal Trace Language (TTL) (cf. Bosse *et al.*, 2006b). All of these properties have been automatically checked and shown to be satisfied for traces, as 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 the properties can be verified for empirical traces without having knowledge on the interpretation. In addition, it specifies the properties on the correctness of interpretation based upon the selection of the most specific default rule.

## *6.2 The 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 on 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 on such specific interpretations, the actions to be performed can be derived:

**derivable_ac(γ:TRACE, t:TIME, C:CONJUNCTION, i:info_element, s:sign):**

$\text{state}(\gamma, t) \models \text{default\_rule}(C, i, s, i, s)\ \&\ \forall i2{:}\text{info\_element}, s2{:}\text{sign}$

$[\ \text{element\_of}(i2, s2, C) \Rightarrow \text{most\_specific\_int}(\gamma, t, i2, s2)\ ]$

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):**

$\exists C{:}\text{CONJUNCTION}$

$[\ \text{derivable\_ac}(\gamma, t, C, i, s)\ \&\ \forall C2{:}\text{CONJUNCTION} \neq C, s2{:}\text{SIGN}$

$[\ \text{derivable\_ac}(\gamma, t, C2, i, s2)\ \&\ s \neq s2 \Rightarrow \text{size}(C2) < \text{size}(C)\ ]\ ]$

Given the fact that it can now be derived what the correct actions are the 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):**

$[\ \text{most\_spec\_ac}(\gamma, t, i, s)\ \&$

$[\ \neg\exists t'{:}\text{TIME} < t\ \text{most\_spec\_ac}(\gamma, t', i, s)\ ]\ \&$

$[\ \neg\exists t''{:}\text{TIME} > t\ \&\ t'' < t + d\ \neg\text{most\_spec\_ac}(\gamma, t'', i, s)\ ]\ ]$

$\Rightarrow \exists t'''{:}\text{TIME} \geq t\ \&\ t''' \leq t + d\ [\ \text{state}(\gamma, t''') \models \text{world\_state}(i, s)\ ]$

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

**missing_action(γ:TRACE, t:TIME, i:info_element, s:sign, d):**

$\text{most\_spec\_act}(\gamma, t, i, s)\ \&$

$[\ \neg\exists t'{:}\text{TIME} < t\ \text{most\_spec\_ac}(\gamma, t', i, s)\ ]\ \&$

$[\ \neg\exists t''{:}\text{TIME} > t\ \&\ t'' < t + d\ \neg\text{most\_spec\_ac}(\gamma, t'', i, s)\ ]\ \&$

$[\ \neg\exists t'''{:}\text{TIME} \geq t\ \&\ t''' \leq t + d\ [\ \text{state}(\gamma, t''') \models \text{world\_state}(i, s)\ ]$

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):**

$\text{state}(\gamma, t) \models \text{world\_state}(i, s)\ \&$

$\neg\exists t'{:}\text{TIME} \leq t\ \&\ t' \geq t - d\ [\ \text{most\_spec\_ac}(\gamma, t', i, s)\ ]$

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 on the information on the number of people on the plane. As a result, the missing_action property holds. Finally, the incorrect_action property is not satisfied, as only the 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 Artificial Intelligence (AI) can be applied to analyse the empirical material on human reasoning and interpretation within cognitive science (cf. Stenning and van Lambalgen, in press). The formal techniques exploited in the empirical analysis approach put forward are:

- multi-interpretation operators as an abstract formalisation of reasoning towards an interpretation
- default logic to specify a multi-interpretation operator
- a temporalised default logic to specify the possible reasoning traces involved in a multi-interpretation process
- an executable temporal logical language to specify a generic executable default reasoning model to simulate such reasoning traces
- an expressive temporal logical language to specify and verify the properties for reasoning traces.

As such, this work synergises the *protocol analysis* tradition of Ericsson and Simon (1993), which addresses the elicititation of verbal reports from research participants, with the *model checking* tradition introduced by, *e.g.*, Huth and Ryan (2004), which addresses the verification of behavioural properties against formal specifications. It has been shown how the introduced techniques and tools indeed obtain an adequate formalisation and analysis of the empirical material on human reasoning in critical situations in incident management. Two types of empirical material have been used, first, the training material which describes the procedures to be followed, being the basis for the default theory regarding human reasoning. As a result of this default theory, simulated traces have been generated and 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 the errors that might occur in the process. Note that the generation of simulation traces using the formalised training material can bring to light flaws in the 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 detection of errors in such reasoning processes, the relevant properties of both simulation as well as empirical traces have been verified and the results were shown for 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 to study other cases. Therefore, the modelling effort of this first case study (which involves a significant amount of effort) is expected to be reduced as the knowledge part of this case study can greatly be reused.

The presented approach can be used to enable the 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, analyse human communication in real time. A more feasible goal in the short term is to analyse historical cases and to formalise the current procedures using default logic and generate simulation results for particular accidents, thereby analysing 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. This disaster is representative for many of the disasters that occur. It is, however, for 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 the case of a huge incident management organisation, calculating all the possible interpretations of the entire combination will be difficult. The idea is, however, that only the interpretations that are useful in a particular situation are generated. 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 of 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 the existing formal methods to protocol analysis can be reused. For example, Langevelde and Treur (1991) proposed a formal framework that can be used to analyse 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 *et al.* (2006a), who described an approach to formalise and analyse 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. Other languages and tools are also 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 that is 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. and Treur, J. (2006a) 'Formalization and analysis of reasoning by assumption', *Cognitive Science Journal*, Vol. 30, No. 1, pp.147–180.

Bosse, T., Jonker, C.M., van der Meij, L., Sharpanskykh, A. and Treur, J. (2006b) 'Specification and verification of dynamics in cognitive agent models', *Proc. of the 6th Int. Conf. on Intelligent Agent Technology, IAT'06*, IEEE Computer Society Press, pp.247–254.

Bosse, T., Jonker, C.M., van der Meij, L. and Treur, J. (2007) 'A language and environment for analysis of dynamics by simulation', *International Journal of Artificial Intelligence Tools*, Vol. 16, pp.435–464.

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

Brewka, G. (1994) 'Adding priorities and specificity to default logic', in C. MacNish, L. Pereira and D. Pearce (Eds.) *Proc. of JELIA'94, LNAI*, Springer Verlag, Vol. 838, pp.247–260.

Brewka, G. and Eiter, T. (1999) 'Prioritizing default logic: abridged report', *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, 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 and B. Gaines (Eds.) *Proceedings of the European Knowledge Acquisition Workshop, EKAW '91*, GMD-Studien 211, pp.162–183.

Marek, V.W., Treur, J. and Truszczynski, M. (1997) 'Representation theory for default logic', *Annals of Mathematics and AI*, Vol. 21, pp.343–358.

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

Meadows, C. (2003) 'Formal methods for cryptographic protocol analysis: emerging issues and trends', *IEEE Journal on Selected Areas in Communication*, Vol. 21, No. 1, pp.44–54.

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

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', *Proceedings of the 8th International Workshop on Non-monotonic Reasoning*, Breckenridge, Colorado, USA, April.

Reiter, R. (1980) 'A logic for default reasoning', *Artificial Intelligence*, Vol. 13, pp.81–132.

Rips, L.J. (1994) *The Psychology of Proof: Deductive Reasoning in Human Thinking*, Cambridge, MA: MIT Press.

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

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, pp.299–303.