A Formal Empirical Analysis Method for Human Reasoning and Interpretation

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.

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 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. The leading example is an airplane crash.

In the next section, the aircrash example is presented. Thereafter, an abstract formalization of a reasoning process leading to multiple interpretations is specified, followed by a section showing 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, a temporalized form of Default Logic is chosen to simulate the possible reasoning traces for the case study. Moreover, a number of properties of such reasoning traces are formalized and checked. Finally, the last section presents the conclusions.

The Incident Management Domain

Within incident management people are working under severe pressure; having incomplete information, decisions have to be made quickly, which can have a huge impact on the success of the whole operation. This paper focuses on one example: that of the Hercules airplane crash at the military airport of Eindhoven in the Netherlands (Ministry of the Interior, 1996). This example is taken because it is representative for the occurrences in incident management. The plane, carrying a military brass band in the cargo room and a crew of four people, flew into a flock of birds just before landing, causing one of the engines to fail, which made the plane tilt to one side. As a result, the plane crashed on the runway and caught fire. The Air Traffic Controller (ATC) had information that a military brass band was on board of 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 fire fighters, but declared the wrong scenario (i.e., 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 were on board of the plane. Since the air traffic controller assumed that the message of a military brass band being on board had been passed through to the fire fighters, he answered that this was 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, thus the brass band was not rescued.

Multiple Interpretations

Reasoning towards an interpretation can be formalized at an abstract generic 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. 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 $\mathbf{MI} : \mathbf{P(L_1)} \rightarrow \mathbf{P(P(L_2))}$ 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 $\mathbf{S, T} \in \mathbf{MI(X)}$, if $\mathbf{S} \subseteq \mathbf{T}$ then $\mathbf{S = T}$.

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 $\mathbf{s} : \mathbf{P(P(L))} \rightarrow \mathbf{P(P(L))}$ that assigns to each nonempty set of interpretations a nonempty subset: for all **A** with $\phi \neq \mathbf{A} \subseteq \mathbf{P(L)}$ 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 $\mathbf{C} : \mathbf{P(L_1)} \rightarrow \mathbf{P(L_2)}$ 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 $\mathbf{s} : \mathbf{P(P(L_1))} \rightarrow \mathbf{P(P(L_2))}$ 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 $\mathbf{C(X) = s(MI(X))}$ 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. The part chosen focuses on the ATC and its interaction to the operator. This information was derived based on training material, see (NIBRA, 2001). An issue is the difference in opinion as to whether or not the ATC communicated to the operator that there are more than 25 people on board. Initial observations of the ATC are:

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)

Fig. 1 provides an overview of ATC's first decision making. It 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.

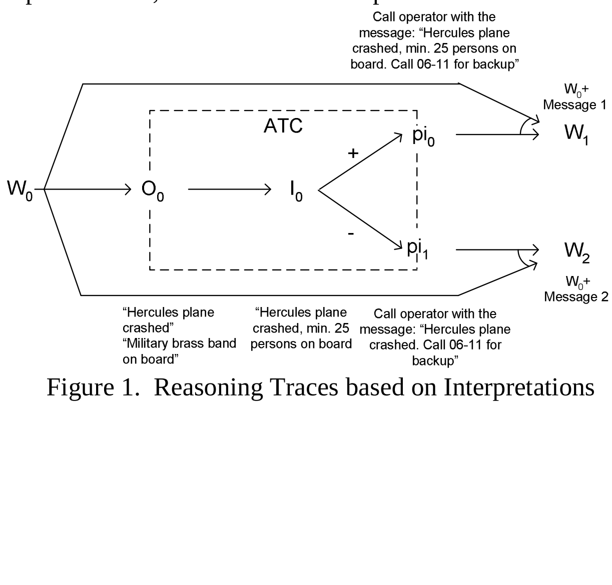

Figure 1. Reasoning Traces based on Interpretations

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: $\alpha: \beta_1, .., \beta_n / \gamma$. Here α is the precondition, it has to be satisfied before considering to believe the conclusion γ, where the βs, called the justifications, have to be consistent with the derived information and **W**. As a result γ 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 $\alpha: \beta / \beta$. 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, 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)

Simulation by Temporalized Default Rules

In this section, a generic simulation model for default reasoning is specified (based on the executable temporal LEADSTO language; cf. Bosse et al., 2005), and applied to the case study. As discussed in the section regarding multiple interpretations, 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. 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 the section on multiple interpretations).

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 LEADSTO language, 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 the section explaining the domain). An example simulation trace is shown in Figure 2. 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 2, 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

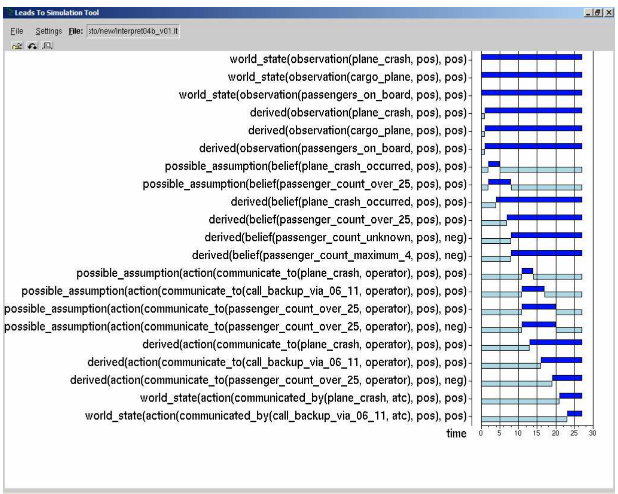

Figure 2. Simulation trace of the reasoning of the ATC

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.

It is important to note that the trace shown in Figure 2 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 the section regarding multiple interpretations. 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.

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.

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 2, 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):

$$[\text{follows\_from\_default}(\gamma,i,s,t) \vee \text{follows\_from\_strict\_constraint}(\gamma,i,s,t) \vee \text{world\_fact}(\gamma,i,s,t)]$$

world_fact(γ:TRACE, i:info_element, s:sign, t:TIME):

$$\exists t2{:}\text{TIME} < t \;\; \text{state}(\gamma, t2) \models \text{world\_state}(i, s)$$

follows_from_strict_constraint(γ:TRACE, i:info_element, s:sign, t:TIME):

$$\begin{array}{l} \exists C{:}\text{CONJUNCTION},\ t2{:}\text{TIME} < t \;\; [\ \text{state}(\gamma, t2) \models \text{strict\_constraint}(C, i, s)\ \& \\ \forall i2{:}\text{info\_element}, s2{:}\text{sign}\ [\ \text{element\_of}(i2, s2, C) \Rightarrow \\ \qquad \text{state}(\gamma, t2) \models \text{derived}(i2, s2)\ ]\ ] \end{array}$$

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):

$$\begin{array}{l} \exists t2{:}\text{TIME} < t,\ C{:}\text{CONJUNCTION} \\ [\text{state}(\gamma, t2) \models \text{default\_rule}(C, i, s, i, s)\ \&\ \forall i1{:}\text{info\_element}, s1{:}\text{sign} \\ \quad [\ \text{element\_of}(i1, s1, C) \Rightarrow\ \text{state}(\gamma, t2) \models \text{derived}(i1, s1)\ ] \\ \quad \&\ \forall t3 \geq t\ \forall s' \neq s\ \ \text{not state}(\gamma, t3) \models \text{derived}(i, s') \end{array}$$

consistency(γ:TRACE):

$$\begin{array}{l} \forall i{:}\text{info\_element}, s{:}\text{sign}, t{:}\text{TIME} \\ [\ \text{state}(\gamma,t) \models \text{derived}(i, s) \Rightarrow \\ \neg\exists t2{:}\text{TIME}, s2{:}\text{sign}\ [s \neq s2\ \&\ \text{state}(\gamma,t2) \models \text{derived}(i, s2)]\ ] \end{array}$$

exhaustiveness(γ:TRACE):

$$\begin{array}{l} \forall t{:}\text{TIME}, i{:}\text{info\_element}, s{:}\text{sign}, C{:}\text{CONJUNCTION} \\ [\text{state}(\gamma, t) \models \text{default\_rule}(C, i, s, i, s)\ \& \\ \forall i2{:}\text{info\_element}, s2{:}\text{sign}\ [\ \text{element\_of}(i2, s2, C) \Rightarrow \\ \qquad \text{state}(\gamma, t) \models \text{derived}(i2, s2)\ ]\ \& \\ \neg\exists t2{:}\text{TIME}, s3{:}\text{sign}\ [s \neq s3\ \&\ \text{state}(\gamma, t2) \models \text{derived}(i, s3)] \\ \Rightarrow \exists t3{:}\text{TIME}\ [\text{state}(\gamma, t3) \models \text{derived}(i, s)] \end{array}$$

derived_persistency(γ:TRACE):

$$\forall t1, t2\ [\ \text{state}(\gamma, t1) \models \text{derived}(i, s)\ \&\ t1{<}t2 \Rightarrow \text{state}(\gamma, t2) \models \text{derived}(i, s)\ ]$$

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.

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):

$$\begin{array}{l} \text{state}(\gamma, t) \models \text{default\_rule}(C, i, s, i, s)\ \&\ \forall i2{:}\text{info\_element}, s2{:}\text{sign} \\ \quad [\ \text{element\_of}(i2, s2, C) \Rightarrow\ \exists t'{:}\text{TIME} \leq t \\ \qquad [\ \text{state}(\gamma, t') \models \text{observation}(i2, s2)\ \&\ \neg[\exists s3{:}\text{SIGN}, t''{:}\text{TIME} \leq t\ \&\ t'' \geq t' \\ \qquad\quad [\ \text{state}(\gamma, t'') \models \text{observation}(i2, s3)\ \&\ s2 \neq s3\ ]\ ]\ ]\ ]\ ] \end{array}$$

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):

$$\begin{array}{l} \exists C{:}\text{CONJUNCTION}\ \ [\ \text{derivable\_int}(\gamma, t, C, i, s)\ \& \\ \forall C2{:}\text{CONJUNCTION} \neq C, s2{:}\text{SIGN} \\ \quad [\ \text{derivable\_int}(\gamma, t, C2, i, s2)\ \&\ s \neq s2 \Rightarrow\ \text{size}(C2) < \text{size}(C)\ ]\ ] \end{array}$$

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):

$$\begin{array}{l} \text{state}(\gamma, t) \models \text{default\_rule}(C, i, s, i, s)\ \&\ \forall i2{:}\text{info\_element}, s2{:}\text{sign} \\ \quad [\ \text{element\_of}(i2, s2, C) \Rightarrow \text{most\_specific\_int}(\gamma, t, i2, s2)\ ] \end{array}$$

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):

$$\begin{array}{l} \exists C{:}\text{CONJUNCTION} \\ \quad [\ \text{derivable\_ac}(\gamma, t, C, i, s)\ \&\ \forall C2{:}\text{CONJUNCTION} \neq C, s2{:}\text{SIGN} \\ \qquad [\ \text{derivable\_ac}(\gamma, t, C2, i, s2)\ \&\ s \neq s2 \Rightarrow \text{size}(C2) < \text{size}(C)\ ]\ ] \end{array}$$

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):

$$\begin{array}{l} [\ \text{most\_spec\_ac}(\gamma, t, i, s)\ \& \\ \ [\ \neg\exists t'{:}\text{TIME} < t\ \ \text{most\_spec\_ac}(\gamma, t', i, s)\ ]\ \& \\ \ [\ \neg\exists t''{:}\text{TIME} > t\ \&\ t'' < t + d\ \ \neg\text{most\_spec\_ac}(\gamma, t'', i, s)\ ]\ ] \\ \ \Rightarrow \exists t'''{:}\text{TIME} \geq t\ \&\ t''' \leq t + d\ [\ \text{state}(\gamma, t''') \models \text{world\_state}(i, s)\ ] \end{array}$$

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):

$$\begin{array}{l} \text{most\_spec\_act}(\gamma, t, i, s)\ \& \\ [\ \neg\exists t'{:}\text{TIME} < t\ \ \text{most\_spec\_ac}(\gamma, t', i, s)\ ]\ \& \\ [\ \neg\exists t''{:}\text{TIME} > t\ \&\ t'' < t + d\ \ \neg\text{most\_spec\_ac}(\gamma, t'', i, s)\ ]\ \& \\ [\ \neg\exists t'''{:}\text{TIME} \geq t\ \ \&\ t''' \leq t + d\ [\ \text{state}(\gamma, t''') \models \text{world\_state}(i, s)\ ] \end{array}$$

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):

$$\begin{array}{l} \text{state}(\gamma, t) \models \text{world\_state}(i, s)\ \& \\ \neg\exists t'{:}\text{TIME} \leq t\ \ \&\ t' \geq t - d\ [\ \text{most\_spec\_ac}(\gamma, t', i, s)\ ] \end{array}$$

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.

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 tradition of (Ericsson and Simon, 1993) with the model checking tradition introduced by e.g. (Huth and Ryan, 2004). It has been shown how indeed these techniques and tools obtain an adequate formalization and analysis of empirical material on human reasoning in critical situations in incident management. Simulated traces have been generated, compared to the given empirical traces and found adequate. Relevant properties of both simulation as well as empirical traces have been verified and results were shown of this verification process. 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.

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. 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.

Note that the executable temporal logical language LEADSTO, which was used for simulation in the simulation section, 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., Van der Meij, L, and Treur, J. (2005) LEADSTO: a Language and Environment for Analysis of Dynamics by SimulaTiOn. In: Eymann, T., et al. (eds.), *Proc. of MATES'05*. Lecture Notes in AI, vol. 3550. Springer Verlag, pp. 165-178. Extended version in: *Int. J. of AI Tools*, 2007.

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. (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

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.

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.