Towards "Why...Because"-Analysis of Failures

Diplomarbeit an der Technischen Fakultät der Universität Bielefeld
Februar 1998

vorgelegt von

Karsten Loer

Betreuer:

Prof. Peter Ladkin, PhD
Universität Bielefeld
Technische Fakultät
AG Rechnernetze und verteilte Systeme
Universitätsstraße 25
D-33615 Bielefeld

Prof. Harold Thimbleby
Middlesex University
School of Computing Science
Bounds Green Road
LONDON, N11 2NQ
United Kingdom

Karsten Loer, Am Stückenkamp 10a, 33607 Bielefeld, 20.Februar 1998

Erklärung

Ich versichere, daß ich diese Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und daß die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen hat. Alle Ausführungen, die wörtlich oder sinngemäß übernommem wurden, sind als solche gekennzeichnet.

Bielefeld, den 20. Februar 1998

Karsten Loer

Danksagung

An dieser Stelle möchte ich Professor Peter Ladkin für die das interessante Thema und die hervorragende Betreuung dieser Arbeit danken.

Ebenfalls danke ich Professor Harold Thimbleby für die Betreuung meiner Arbeit.

Ich danke den Teilnehmern des Seminars "Grundlagen der System-Korrektheit und Versagen" für viele wertvolle Diskussionen zu den vorgestellten Methoden.

Besonderer Dank gilt meinen Kommilitonen Heiko Holtkamp, Helga Rietmann, Michael Blume, Sven Wachsmuth und Jessica Weis für ihre Hilfe bei der Korrektur meiner Arbeit.

Für die Bereitstellung einer Vorabversion seines Werkzeuges *wb2dot* danke ich Michael Höhl.

Ich danke meine Eltern dafür, daß sie mir durch ihre unermüdliche Unterstützung dieses Studium ermöglicht haben.

Contents

1 Introduction **1**

2 Analysis of a Sample Case **5**
2.1 The Ontology 5
2.2 Relations Between Nodes and How to Find Them 6
2.3 Non-Events and Deontics 9
2.4 Contrastive Explanation 11
2.5 Proceeding by determining earlier contrast 12
2.6 Possible Worlds 13
2.7 Possible Worlds and Causal Explanation 14
2.8 Investigation of ATC Procedures 15
2.9 A textual Graph 17
2.10 Adapting PADs to describe possible worlds 17
2.11 Summary 20

3 Specification of Subsystems **21**
3.1 Introducing TLA+ 21
3.2 Physical Subsystems 25
3.2.1 Inter-ATC Communication 25
3.3 "Human Subsystems" - The PARDIA Model 30
3.3.1 The Model 30
3.3.2 Specifying the PARDIA Model 31
3.4 Real Time Theorems 34
3.5 Procedures 35
3.5.1 Specification of Landing Procedures 35
3.5.2 Standard Operating Procedures 45
3.6 Summary 47

4 Formal Completeness and Correctness Proof **49**
4.1 The Hierarchical Proof-scheme 49
4.2 EL Proof-Step Templates 51
4.2.1 The High-Level Template 51
4.2.2 Deontic Proof-Step Template 52
4.2.3 CCT as a Derived Meta-Rule 53
4.3 Sufficient Causal Explanation Proof 54
4.3.1 Proof of [1] 54
4.3.2 Proof of [11] 56
4.3.3 Proof of [111] 58
4.3.4 Proof of [1111] 60
4.3.5 Proof of Node 12 66
4.3.6 Proof of (121) 68
4.3.7 Proof of Node 2 73

4.3.8 Proof of Node 22 . . . 77
4.3.9 Proof of Node 3 . . . 78
4.3.10 The Full Explanations . . . 82
4.4 The Final WB-Graph . . . 85

5 Conclusions **87**

A DC-10 Incident Report **93**
A.1 Report from *Flight International* [dWL95] . . . 93
A.2 DC-10 misses Frankfurt - by 300km . . . 93
A.3 Article on this subject in Risks 17.38 [Lad95a] . . . 94
A.4 Article on this subject in Risks 17.40 [Lad95b] . . . 94

B The Rules of EL **97**
B.1 Classical Rules . . . 97
B.1.1 Propositional Rules . . . 97
B.1.2 Quantifiers . . . 98
B.2 Modal Rules . . . 98
B.3 Temporal Rules . . . 99
B.4 Behaviors and the Rules They Engender . . . 100
B.5 Strict Implication . . . 100
B.6 The Deontic Modalities . . . 101
B.7 Counterfactual Conditionals . . . 101
B.8 Lewis Semantics for Counterfactuals . . . 101
B.9 Rules for Counterfactual Conditionals . . . 102
B.10 Defining the Other Modalities from the Counterfactual . . . 103
B.11 The Well-Formed Formulas of EL . . . 104
B.12 Soundness and Completeness of a fragment of EL . . . 104
B.13 Special EL Rules . . . 105
B.14 Axioms and Processes for WBA . . . 107

C Sufficient Causal Explanation **109**
C.1 The Causal Sufficiency Condition . . . 109
C.2 The CCT for [11] and [12] . . . 110
C.3 Inference Rules for the CCT . . . 111
C.4 Causal Sufficiency Through Procedural Necessity . . . 111

D Glossary **115**

List of Figures

2.1 Temporal Succession of Top-Level Nodes . . . 6
2.2 View of the WB-graph after first step . . . 8
2.3 Establishing the first causal chain . . . 9
2.4 Introducing non-events . . . 11
2.5 Possible and realized causalities . . . 13
2.6 Fully causality-based WB-Graph . . . 17
2.7 Textual WB-Graph Syntax representation in EBNF . . . 18
2.8 Textual WB-Graph of the Sample Analysis so far . . . 19
2.9 Predicate Action Diagram for explanation of assumed state 2214 . . . 20

3.1 Sample Module "*ATCtrans*" . . . 22
3.2 Hierarchy of ATC Communications Specifications . . . 25
3.3 Module *ATCproc* . . . 26
3.4 Module *ATCcomm* . . . 27
3.5 Module *ATCcomm_history* . . . 28
3.6 Module *This_ATCcomm_history (Part 1)* . . . 29
3.7 Module *This_ATCcomm_history (Part 2)* . . . 30
3.8 The PARDIA information processing model . . . 30
3.9 Possibly necessary improvements derivable from PARDIA errors . . . 31
3.10 PARDIA Axioms (Part1) . . . 32
3.11 PARDIA Axioms (Part 2) . . . 33
3.12 The PARDIA Norms . . . 34
3.13 Module *RealTimeTheorems* . . . 35
3.14 Hierarchy of Landing Procedures and Real-Time Specifications . . . 36
3.15 Module *Landing_Defs* . . . 36
3.16 Module *Landing_Axioms* . . . 38
3.17 Module *Landing_Norms (Part 1)* . . . 39
3.18 Module *Landing_Norms* (Part2) . . . 40
3.19 Module *Landing_Specs (Part 1)* . . . 41
3.20 Module *Landing_Specs (Part 2)* . . . 42
3.21 Module *Landing_Specs* (Part 3) . . . 43
3.22 State Predicates from Specifications of Landing Procedures . . . 44
3.23 Module *SOP_Specs* . . . 46

4.1 (a) Structure of the intuitive WB-Graph and (b) resulting Proof Structure . . . 49
4.2 (a) Structure of the proved WB-Graph and (b) resulting Proof Structure . . . 50
4.3 Top-Level Module of the Proof . . . 50
4.4 Textual form of the final WB-Graph . . . 85
4.5 Pictorial form of the final WB-Graph . . . 86

5.1 Automatically generated WB-Graph . . . 90

5.2 Architecture of a possible system to assist WB-Graph development. 91
5.3 Need of Resources for the presented Analysis (Worst case estimation from a development case for a WB-Graph with 26 nodes) 91
5.4 Estimated Need of Resources for the Analysis of selected Reports . . 92

B.1 The Rules of EL 106

Chapter 1

Introduction

Commercial aircraft are widely understood to be one of the safest means of transportation available nowadays – every year more people are killed or become injured in traffic accidents in Germany than do people involved in aviation accidents worldwide ([Deu97], [Ran97]). However, most people prefer to use cars or railway as the means of choice instead of aircraft. Apart from the fact that most people think they had more chances to intervene in an dangerous situation when they were in command about the vehicle they use by themselves - which isn't true in most cases [Dör97] - another main cause may be that the effects of the smallest pilot error can be fatal for all passengers he is responsible for, as we can see too often in the media. Since almost every accident or incident in aviation causes injuries and many of them fatalities and since every incident may lead to an accident the next time it occurs, reliable methods must be found to prevent both of them.
This also is valid for other mass transport like rail or subway and safety critical systems like nuclear power stations and chemical factories. For our research we chose the domain of aviation for two reasons. First, aircraft incidents and accidents unfortunately happen with a statistically sufficient frequency, in contrast to, say, incidents in nuclear power stations. Second, they are usually investigated very accurately by professional investigators of regulatory and commercial authorities like the U.S. *National Transportation and Safety Board* (NTSB) and the UK *Aircraft Accident Investigation Branch* (AAIB).

There are in principle two possible ways to avoid these accidents:

(A) One can try to remove all possible errors during the specification stage of a new system. The problem that arises when dealing with aircraft, trains or chemical plants is that they are difficult to handle because of their complexity.
Take an aircraft for example: It interacts with its environment, but usually cannot influence it – we call systems with these properties *open* systems. Furthermore it is a *heterogeneous* system, i.e. it consists of digital, mechanical or other "physical" parts as well as human "subsystems". Finally, most of its systems need to work in *real-time* to allow a reliable operation. There are methods available already to explain or specify (or in some cases prove the correctness of) every part of these open, heterogeneous real-time systems (OHRSs; [LL98b]). Human factors specialists have developed several models to explain and classify the failures of human beings working in complex technical environments (e.g. [Rea97], [Per84]). Computer Scientists and mathematicians developed specification methods to describe the behaviour of complex, concurrent and perhaps coupled systems (e.g. [Lam], [Lam93b], [MWD97],

[Pet62]). Several suggestions were made to introduce the human component into these methods (e.g. [Joh95], [Lev95], [TJ96]). But none of these methods allows one to deal with a combination of all kinds of subsystems at once.
In section 2 we will show how to use the *Explanatory Logic* (EL) from Ladkin ([LL98b] - attached as appendix B) for the specification of OHRSs.

(B) The second possible way is to avoid the reoccurrence of accidents by a meticulous accident analysis. Since we are not professional accident investigators, we need to rely on accident reports from aviation authorities available to the public.
This leads to several problems: Accident reports often are neither complete nor consistent in all findings ([GLL97a], [HL97]). The team of experts investigating the accident consists of flight engineers, computer scientists, human factors experts, metallurgists, pathologists as well as experts from various other disciplines. The terminology they use is inconsistent over all these fields. Different terms are used for the same facts and different facts are assigned the same terms.
Although each research area has developed its own terminology (which is a kind of natural language), the mixture of these languages is full of ambiguities and can cause misunderstandings [TJ96]. This makes it sometimes difficult for the reader to draw the correct conclusions from the findings - and that's the worst effect, since one main task of accident reports is to provide information to the authorities and aircraft manufacturers which allows them to formulate new regulations, design and training requirements, or whatever the report turns out that needs to be improved.

The third big problem with accident reports is the way the information is presented. For example, the report on American Airlines flight 965 accident near Cali, Colombia, on Dec 20, 1995 [oTRoC96], which was analyzed in [GLL97a], is about 75 pages. The report on China Airlines Flight B1816 accident at Nagoya Airport on April 26, 1994 [Com96] is even about 200 pages. All kinds of information (meteorological, technical, medical etc.) are spread through the whole report. A person wishing to understand what exactly lead to the accident and why needs to read the whole report – at least we needed to do this several times. To find particular statements again later on, a generated HTML-document of the report (prepared for the WWW by Hiroshi Sogame and Peter Ladkin, accessible via [RVS]), turned out to be pretty much of help. However, it could not solve the main problems – to easily extract the important information from the mass of text and to get a systematic overall view on the mass of facts and their role in the accident.
We state that one formal language as means of communication common to all expert groups will help integrating each group's findings into one sound model of the incident/accident. Furthermore, we propose that EL is such a language, as we demonstrate in chapter 2. With (textual and pictorial) *Why..Because Graphs* we present a method to clearly arrange the information given in incident and accident reports as well as different other sources. We will show by an example analysis of a newspaper article that *Why..Because-Analysis* (WBA) is not restricted to the analysis of official reports, but it also suffices to take secondary sources on which to base a detailed analysis. Where WBA reaches limits due to a lack of information provided by secondary literature (but sometimes also by the final reports), we explain how to use *Predicate-Action-Diagrams* (PADs) – adopted from [Lam95], and based on Explanatory Logic – to state the right questions to experts, who are able to give the missing information. Since we had only limited access to commercial aviation experts [Ble98] (except for the knowledge we could acquire from a private pilot [Lad98]), we used aviation manuals ([Admb], [Adma]) as information sources.

In chapter 4 we moreover will show that EL not only can be used as a means of description, but also allows a rigorously formal proof of both the correctness and completeness of WBA-results.
Given the truth of the *proof obligations* we use, it follows by logic alone that the explanation of the incident we will give is correct and sufficient. We do not know about any other method that accomplishes this!

Outlook:
During the next four chapters we will:

- Demonstrate, how to generate a WB-Graph from an arbitrary source of information (chapter 2).
- Show, how to specify subsystems, procedures and human behaviour in EL (chapter 3).
- Present the rigorously formal correctness and completeness proof of our example case (chapter 4) and
- Draw conclusions from the experiences we made during the development (chapter 5) and the application of the method and give a perspective for future work.

Chapter 2

Analysis of a Sample Case

The easiest way to get started with the Why..Because Analysis (WBA) is perhaps to analyze a sample case to introduce the major working principles[1].
In this chapter, we will do this analysis for an incident that occurred at Brussels Airport in 1995. As information source we intentionally chose a journal article [dWL95] and two comments on this incident from the RISKS-Digest ([dWL95] and [Lad95a]). We intend to show that these sometimes vague explanations suffice to base an WBA on. In fact, for the development of methods which work particularly in investigations based on information sources of possibly unsatisfying reliability, these articles turned out to be very valuable. The whole article and comments are attached as Appendix A.

On 5 September 1995, a DC10, Northwest Airlines Flight 52, on its way to Frankfurt (FRA) from Detroit, landed in Brussels Airport (BRU), much to everyone's surprise – except for the passengers, cabin crew, who watched the route on the cabin video, and air traffic control (ATC), who thought its destination were BRU. The history included in the articles is simple, and we shall extend it to a rigorous analysis of the incident. This will demonstrate what has to be determined to conclude where the errors occurred. During this analysis the reader will see that there is indeterminacy in the reports we work from.

2.1 The Ontology

Our aim is to distinguish and represent every factor contributing to the final incident and their causal and explanatory interrelations. We distinguish four types of factors [LL98b]:

1. *States* are individuated by the collection of state predicates which are true in that state. So states might be interpreted as types. Furthermore states may have a duration, i.e. the values of the variables of the considered state predicates do not change over a period of time.
   Notation: $\langle \mathbf{A} \rangle$

2. *Events* are particulars, representing specific changes in state. However, events usually belong to event *types*, and the description of an event is usually a description of its type, coupled with an indication of when it occurred.
   Notation: $[\mathbf{A}]$

[1] The formal base of our method –Explanatory Logic– is given in Appendix B.

3. *Processes* are defined as state/event mixes of bounded duration which describe undifferentiated actions [Lew86].
   Notation: $\{\mathbf{A}\}$

4. We call the non-occurrence of awaited events *non-events* (see below).
   Notation: $(\mathbf{A})$

We call all these objects "*nodes*". This ontology is justified in [Lad96a, The Formal Semantics] and [GLL97b].

2.2 Relations Between Nodes and How to Find Them

We start our analysis by extracting step by step the information about what happened and why from some source text[2], starting with the most obvious points and going into detail later on:
On its journey, the Aircraft (AC) crosses Shannon ATC (SATC) region and London ATC (LATC) region, finally reaching Brussels ATC (BATC) and landing at BRU runway (RWY) 25. All these pieces of information are mentioned in [dWL95]. Thus at the top level of our investigation we may define four nodes:

[1] AC lands at BRU RWY 25
⟨2⟩ AC in BATC area
⟨3⟩ AC in LATC area
⟨4⟩ AC in SATC area

The aircraft passes through these nodes in a determinate temporal order:

$$\langle 4\rangle \hookrightarrow \langle 3\rangle \hookrightarrow \langle 2\rangle \hookrightarrow [1] \tag{2.1}$$

Figure 2.1: Temporal Succession of Top-Level Nodes

This leads to a *history*, a statement of temporal succession (denoted by "$\hookrightarrow$") on which we shall base our inquiry. (The source text may not use temporal order - histories make free use of verbal aspect and other devices to establish the temporal order of nodes.) Note that every step of the chain presented in figure (2.1), can be interpreted as a theorem we need to prove when we do a formal analysis of the correctness and completeness of our WB-Graph in chapter 4.

We assume a linear temporal succession, composed of points. Nodes may have (usually will have!) temporal extent or duration. For those which occur "at an instant" (which do not have temporal extent), an assertion "$A \hookrightarrow B$", for two instantaneous states, events, or non-events A and B – processes always have temporal extent – means that A and B both occurred and that B succeeds or is contemporaneous with A. If A and B have temporal extent (thus could be processes also), an assertion $A \hookrightarrow B$ means that there is at least one temporal part of A which precedes all temporal parts of B.

For an axiomatic explanation of the "$\hookrightarrow$" relation, see [Lam86]. We, however, shall only use "$\hookrightarrow$" to state top-level theorems to be proved. At this level of analysis

[2] We use [dWL95] and [Lad95a] as an example here. In more detailed cases we have used entire accident reports from Cali [GLL97a] and Warsaw [HL97].

it is sufficient to attempt a reduction of "$\hookrightarrow$" to causality ("$\twoheadrightarrow^*$", see below). Where this is not possible, one may conclude that the history gave no clues to the causal history, and the analysis must remain incomplete.
Since we wrote about *temporal succession* and *causality* several times now, how are they related anyway? We propose to make the connection between these terms through Mill's observation that:

> "Whether the cause and its effect be necessarily successive or not, the beginning of a phenomenon is what implies a cause, and causation is the law of the succession of phenomena[...]
> Whether the effect coincides in point of time with, or immediately follows, the hindmost of its conditions, is immaterial. At all events it does not precede it; and when we are in doubt, between two co-existent phenomena, which is cause and which is effect, we rightly deem the question solved if we can ascertain which of them preceded the other." [Mil73, Chap. V, Book III, Volume 7]

(It should be evident that we are not considering the viability of backwards causation in this work.)
By introducing new facts and perhaps assumptions not mentioned particularly in the sources, we are now going to improve figure (2.1) successively to achieve finally a causal explanation of the incident.
Before we try to replace "$\hookrightarrow$" by causal chains between its arguments in a first step, we should think about their relationship for a moment.
Using Mill's definition of causality, we find that temporal succession is (at least) a hint towards causality (sometimes taken as more than a hint – see Hume, Mill [Mil73], Moffett *et. al.* [MHCM96] and Johnson [TJ96] as criticized in [Lad96b]). We formulate this insight as follows:

Axiom 1 $\vdash (A \twoheadrightarrow^* B) \Rightarrow (A \hookrightarrow B)$
An n-fold succession of causal-factor relations between chained factors implies a temporal succession between (at least) the first and the last factor of this chain.

We have not yet defined the relation $\twoheadrightarrow^*$. It's the transitive closure of $\twoheadrightarrow$. Although one cannot define the transitive closure of a relation from that relation in first-order logic, one can nevertheless axiomatize it so that it is sound and relatively complete. One simply uses the recursive definition, and since proving is a finitary activity, the things one can prove about the transitive closure are exactly those which follow from a finite number, say n, of proof steps, which are exactly those that follow from an n-fold succession of causal-factor relations:

Axiom 2 $\vdash (A \twoheadrightarrow B) \Rightarrow (A \twoheadrightarrow^* B)$

Axiom 3 $\vdash (A \twoheadrightarrow^* B) \wedge (B \twoheadrightarrow^* C) \Rightarrow (A \twoheadrightarrow^* C)$

Lewis [Lew86] also defines causality as the transitive closure of the causal-factor relation, so the relation $\twoheadrightarrow^*$ is, according to Lewis, causality. But nothing we do hangs on this particular identification.

Using *modus ponens*, Axiom 1 leads to an inference rule we will need for our analysis:

$$\frac{A \quad \twoheadrightarrow^* \quad B}{A \quad \hookrightarrow \quad B} \tag{2.2}$$

This rule means that causalities must be consistent with temporal order.
Our investigation method attempts to reverse this order, like proof-search. We

want to demonstrate figure (2.1) and must use rule (2.2)[3]. This leads us to try to establish a causal chain between two temporally-successive nodes. We not necessarily succeed, because there are temporally successive nodes that are simply not causally related in any sense meaningful to us (traffic jams may happen just after we leave the highway: we would not want necessarily to see a causal connection, unless we knew that we parked our truck across all lanes before we left it). If we fail to establish a causal connection in a particular analysis, then maybe we should focus on different temporally-successive nodes. We will show, how to establish the relation $A \nRightarrow B$ for some A and B in such cases later on. At this point we prefer to first finish the top-level analysis of our case study:

An intuitive way to find intermediate causal factors to replace the temporal "$\hookrightarrow$" by such a causal chain is by stating "Why..Because"-questions:

Why B ? – Because C
Why C ? – Because A

Of course, some arbitrary node can be affected not only by individual but also by multiple factors. We start our analysis by questioning:

"Why did the incident event [1] occur?"

Several potentially relevant facts can be detected in the article (we don't have any formal criterion for "relevance" – this is an intuitive judgment):

[11] Crew (CRW) realizes they are landing at the wrong airport
[12] CRW opts to continue landing

We can test the relations [11] $\nRightarrow$ [1] and [12] $\nRightarrow$ [1] using inference rule (2.8) and find that the relations hold. Thus they give a partial explanation for the immediate causes of [1]. Including these new nodes into the existing graph results in figure 2.2:

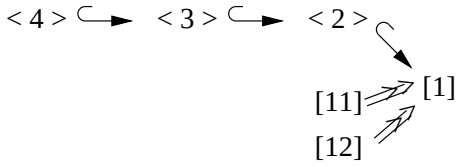

Figure 2.2: View of the WB-graph after first step

Further analysis of [12] – which at this point means scanning the text ([dWL95]) for further information– results in a new node:

⟨121⟩ CRW has safety reasons for continuing landing

We argue that safety reasons are a cause for continuing the landing.

Finding a cause for [11] is not that easy, since we do not find any hints in the text about the reasons why they finally realize the problem. Use of a second source [Lad95a] provides additional information.

[111] CRW gets visual contact to Brussels airport
{112} CRW notices that Brussels' airport layout is different from Frankfurt's

[3] cf. the hierarchical proof method of Lamport [Lam93a].

Notice that we classify {112} as a (cognitive) process, since in our opinion this realization is not a single event, but consists of the combination of several events over a period of time.

Event [111] is caused by

[1111] AC breaks out under clouds,
⟨1112⟩ AC near Brussels Airport

and ⟨2⟩ that we already know from rule (2.1). An expert would say that there were also

⟨1113⟩ CRW procedures

contributing to [111] (we shall see more on including expert knowledge below). Once again, we present the argument that made us establish exactly this causal connections rather than different ones: To get visual contact from a viewpoint to an object ([111]) under given procedures (⟨1113⟩ – e.g. one does not use optical or radar aids) one needs to be close enough to the object (⟨1112⟩,⟨2⟩) and no obstacles must be in between ([1111]). This is based on physical truths.
So actually we have succeeded in filling out a causal chain from ⟨2⟩ to [1] (figure 2.3) by collecting pieces of information directly given in the text and by establishing causal relations between them:

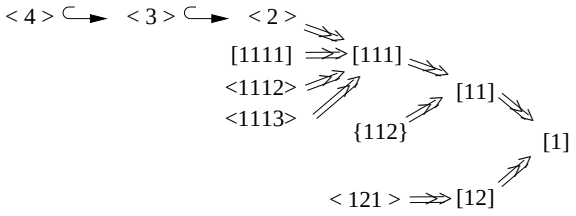

Figure 2.3: Establishing the first causal chain

State ⟨1112⟩ seems to be worthy of further analysis, because according to its flight-plan the aircraft was never supposed to be there under "normal" circumstances. So it raises the question how and why they got there. This harks forward to use of criteria of *contrastive explanation*. See Section 2.4.

Both primary as well as secondary source, as well as the intuitive meaning of "realize", provide information that

(11121) CRW did not realize that they were on wrong course, UNTIL:[11]

This is an interesting point. How can we know, something *did not* happen, since the states and events we used so far and which are likely to appear in a simple history described what *did* happen?

2.3 Non-Events and Deontics

Formally, a *non-event* is a state. Something we await – according to our knowledge about the situation the system is in as well as the obligations following from the procedures which govern the system – does not occur. That kind of argumentation uses *deontic* reasoning: the procedures *ought* to be followed. Therefore the event

ought to have happened. But it *did not* (– and this is explicitly remarked). To capture this formally, we can introduce a new inference rule which says exactly this.

Before giving the inference rule, we need to analyze the situation a little further. We base the rule on the following principle:

> "The existence of a non-event *follows* from procedures in a situation if, given the situation, and that the procedures are followed, the event must necessarily occur, either then or later; but in fact it doesn't" [LL98b].

This involves two modalities, in the technical sense of the term in modal logic: *necessity* and *tense*. Because tense logics and logics of necessity are often considered separately, the same notation is used for both. Clearly we need to distinguish notation. We shall use $\succ$ for the Lewis-Langford [LL32] relation of *strict implication*, and $\Box$, $\Diamond$ for the *always* and *eventually* operators of simple linear-time tense logic.

The deontic axiom says that procedures ought to be followed:

Axiom 4 $\vdash O(Procedures)$

Suppose an event is a necessary consequence of following procedures in the given situation. Since the procedures ought to be followed, the event ought to occur. It should occur. How do we say *necessary consequence*? Since we're talking about procedures or systems with behaviour, we shall use the tense logic TLA (which we believe suffices) and the following axiom:

Axiom 5 $$\frac{(\vdash_{TLA} A \Rightarrow B)}{A \succ B}$$

This can be used as a proof rule in hierarchical proofs in the following way: to prove "$A \succ B$", the first step is to assume A, and the proof proceeds according to the proof rules of TLA as given in [Lad97]. Given this rule for "$\succ$", we may now formulate the *Deontic Rule* which says that, when the occurrence of an event is a necessary consequence of procedures, the event ought to happen:

$$\frac{Hypotheses \wedge \Box Procedures \succ \Diamond Event}{Hypotheses \wedge O(Procedures) \Rightarrow O(\Diamond Event)} \tag{2.3}$$

It follows trivially as a derived rule from Rule (2.3) and Axiom 4 that

$$\frac{\begin{array}{l} Hypotheses \\ (Hypotheses \wedge \Box Procedures) \succ \Diamond Event \end{array}}{O(\Diamond Event)} \tag{2.4}$$

The event may not in fact occur, even though it should have, because it is perfectly possible that the procedures weren't followed and thus allowed the event not to occur. In our analysis, we need to remark and reason with these events that should have occurred but didn't. We call them non-events. What kinds of objects are they? Well, non-events persist: the system state does not change in the relevant way because the event that causes this change does not occur, so non-events describe *states* whose occurrence is inferred from our knowledge of procedures, and of the current situation.

However, it is difficult to formulate this final step, the existence of non-events, as a formal inference rule, because it really tells us explicitly to remark a particular fact. In fact, we have a meta-rule:

Axiom 6 *Explicitly add to the history states* $\langle \neg E \rangle$ *in which* E *is an event,* $O(E)$ *is derivable, and* E *does not occur.*

How do we apply this to the example? We have to identify procedures and the situation, and an event which should have but didn't occur. As far as the Crew is concerned we learn that

> {111211} CRW addresses BATC controller as "Frankfurt" several times,
> ⟨111212⟩ ILS has different frequency for FRA

and that

> [111213] CRW asks for the Bruno VOR's frequency.

All these three occurrences should have alerted the crew to the fact that there was a problem. By use of the deontic reasoning we thus can learn that – contrary to international procedures –

> (111214) Brussels did not question the addressing error although it happened more than once

Adding all this new information leads to figure 2.4

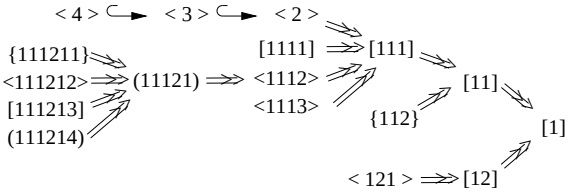

Figure 2.4: Introducing non-events

We have gotten so far, but our explanation is not complete, yet.

How to get from ⟨3⟩ to ⟨2⟩ ?

Our first approach was to try to establish a series of causal factors for the incident itself. We were able to find a chain of causation from ⟨2⟩ to [1]. Thus we try to apply it on the next step, too. Why did the aircraft enter BATC area? Unfortunately, searching for facts effecting ⟨2⟩ produces no satisfying answers because we cannot find any direct statement concerning causes for the transition from LATC area to BATC area. To find the reasons leading to this node we need to expand our method by additional techniques.

2.4 Contrastive Explanation

Contrastive explanation concerns the explanation of facts of the form *why P rather than Q occurred*. Lewis [Lew86][pp229-230] suggests this may be done by giving information about the causal history of P that would not have applied to the history of Q. J.S. Mill's *Method of Difference* [Mil73][III.VIII.2] relies on the principle that a cause must lie among the antecedent differences between a case in which the effect occurs and a case in which it does not. Mill notes that this works best with *diachronic* (before/after) contrasts. Lipton proposes the *Difference Condition*:

> "To explain why P rather than Q, we must cite a causal difference between P and not-Q, consisting of a cause of P and the absence of a corresponding event in the case of not-Q" ([Lip91][p43], cited in [LL98a]).

(Lipton considers that only events may be causes. We are considering causal factors to include nodes of all types, so relevant modifications must be made in this expression of the Difference Condition.) How may we apply the principles of contrastive explanation, in one of these closely-similar forms, to our problem?

We have noted earlier that in any incident there is a contrast to be explained, namely why things happened the way they did rather than the way *they should have done*, or the way *we expected them to have done*. There are two *possible worlds* here: one is the actual world, the way things actually happened, and the other is the world in which the things happened which should have. Let us call this latter the *deontically-correct* world. As stated in the example, in the actual world the aircraft landed at Brussels, and in the deontically-correct world the aircraft landed at Frankfurt. According to any of the three conditions proposed, we may look for a relevant difference in the causal history (let's just say history). We find it in the transition from $\langle 3 \rangle$ to $\langle 2 \rangle$, because in the deontically-correct world, this should have been a transition $\langle 3 \rangle$ to $\langle 2b \rangle$, where

$\langle 2b \rangle$ AC in Maastricht ATC area

Note that $\langle 2b \rangle$ and $\langle 2 \rangle$ are contraries: they cannot both hold in a given history. We are here looking back to the first general difference. Recall from Axiom 1 that causal priority mirrors temporal priority, so that any relevant causally-prior difference will also be temporally-prior: we thus look for the first difference in events and look for causes of that difference.

2.5 Proceeding by determining earlier contrast

The sources we used up to now do not provide much information on the procedures and laws of civil aviation. Having this *expert knowledge* at some point of incident and accident analysis is essential. What we need to know to explain $\langle 2 \rangle$ is that there are

$\langle 21 \rangle$ LATC procedures

to handle flight traffic within the area it controls as well as from and to this area. These procedures are stated explicitly in, for example, the U.S. *Air Traffic Control Handbook*, *Federal Aviation Regulations* (FAR) and *International Civil Aviation Organization* (ICAO) standards, which describe general operating and flight rules as well as standard instrument procedures and communication.

Procedures are similar to a specification of a system. The system is said to fulfill a specification in the same manner that events and states are said to conform to procedures: if the procedures hold (if the specification is fulfilled), the events and states must necessarily follow. Our purpose here is to "debug" procedures and specifications: that is, we do not know whether they were followed and we are attempting to find out.

In this circumstance, we can use deontic reasoning similarly to the manner in which we used this reasoning for considering procedures. Suppose *Spec* is a specification of a system or subsystem that is relied on by the ATC system. If occurrence of a state or event follows from a specification that is:

$$\vdash_{TLA} \quad Spec \rightarrow \Diamond E$$

We do not know if *Spec* is fulfilled, since we are dealing with a failure situation. All we *do* know is that it *ought* to be: We can use the axiom

Axiom 7 $\vdash O(Spec)$

and thus handle system reasoning similar to the manner in which we handled reasoning concerning air traffic procedures: we may conclude by Rule (2.3) that $O(\Diamond E)$. That is: the values of the state predicates are changed by events exactly as specified.

In our examples, a part of the procedures is to get

$\langle 22 \rangle$ ATC data

– especially flight plans – from adjacent ATCs for all aircraft entering an ATCs area and passing this information according to each aircraft's flight-plan to aircraft and other ATCs concerned. This information helps us to explain the circumstances which led the aircraft from LATC to BATC. It is important to mention that we understand and assume that the data was processed as described above, rigorously conformant to ATC procedures. However, intuitively if everything *did* occur as determined in the procedures, something must be wrong with the processed data, since we know they landed in Brussels instead of Frankfurt. Although this information is given directly in our primary source (it is pretty clearly the result of inference rather than direct knowledge at), we can also derive it from the facts we already collected by the use of logic.

2.6 Possible Worlds

Comparing our model (at least the part we analyzed so far) of what *did* happen (figure 2.5 solid lines) to what *should* have happened – namely

$\langle 2b \rangle$ AC in Maastricht ATC area

succeeded by

$\langle 2b1 \rangle$ AC in Frankfurt ATC area
$[2b11]$ AC lands at Frankfurt

(see dashed symbols in figure 2.5) leading to a direction to Frankfurt ... – we can easily determine the difference between these possible "*worlds*".

The situation we're considering here is determinate: we know that one of these worlds occurred and one didn't. We thus use Mill's criterion to identify distinguishing nodes as above.

However, consider the situation in which many events or states could have hap-

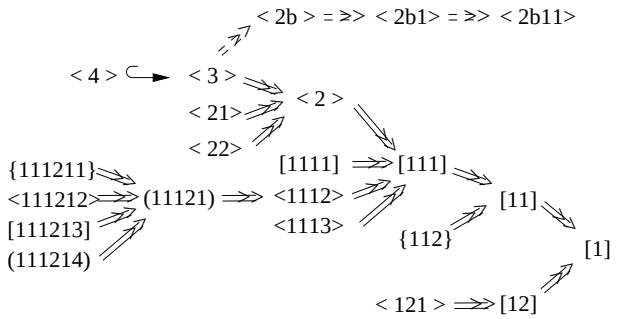

Figure 2.5: Possible and realized causalities

pened, but we do not have enough information (or have not inferred enough information) to be able to tell which was the case and which not. Having more than

one of these possible worlds creates additional work for us because every fact newly introduced in one world needs to be introduced in all other worlds, too. Considering all alternatives leads to exponential growth of the number of worlds to be worked on and thus an exponential growth in the amount of work we have to do. To avoid this ultimately unmanageable situation, we need to find a way to eliminate some of these worlds. Lewis suggests weighting each world by assigning possibilities based on expert knowledge [Lew86]. By defining a threshold, worlds with lower possibilities can be taken out of focus to some extent. To avoid the calculation of the probabilities (in fact, we do not know if this is possible at all under these circumstances), we decided to simply present the possible alternatives.

2.7 Possible Worlds and Causal Explanation

Possible worlds may be interpreted as possible alternatives to the actual situation, in which the actual laws may or may not remain intact. These possible alternatives can be stated – as suggested by David Lewis [Lew86] – as counterfactuals. These again can be used to define the Lewis criterion for causal factors:

> A is a causal factor of B, iff A and B both occurred, and in the nearest possible worlds in which A did not happen, neither did B.

Or, in the words of logic:

$$\frac{\begin{array}{l} A \mathrel{\Box\!\!\rightarrow} B \\ \neg A \mathrel{\Box\!\!\rightarrow} \neg B \end{array}}{A \not\Rightarrow B} \tag{2.5}$$

The semantics of $\Box\!\!\rightarrow$ is a possible-world semantics ([Lew73]). This "*nearest possible world*" relation is Lewis's extension to Kripke semantics. He introduces an additional relation of nearness:

$$\text{World } X \text{ is at least as near as world } Y \text{ to world } W$$

Even though Lewis has given complete sets of inference rules for $\Box\!\!\rightarrow$, during WBA we find ourselves mostly evaluating the truth of assertions involving $\Box\!\!\rightarrow$ by using semantic arguments within the "nearest possible world" semantics [LL98a]. Formally, the semantics is used as follows. Assuming a fix W, the nearness relation yields a binary relation $\preceq_W$, namely

$$X \preceq_W Y \tag{2.6}$$

The relation $\preceq_W$ is axiomatized by Lewis as a *total preorder*. That means that any two worlds can be compared in terms of their similarity to world W; either the one or the other is more similar, or they are both equally similar. The Lewis semantics for $\Box\!\!\rightarrow$ is that

> $A \mathrel{\Box\!\!\rightarrow} B$ in a world W if and only if B is true in all the nearest worlds to W in which A is true.

The case we use is

> $W =$ *the actual world*.

One defines the concept "nearest": a world X is nearest to a world W if and only if, for all worlds Y, world X is at least as near as world Y to world W. Suppose A is true in world W. Then the set of nearest worlds to W in which A is true consists of precisely W itself. Then $A \ \Box\!\!\rightarrow B$ is true in W just in case B is also true. Thus we have the rule:

$$\frac{\begin{array}{c} A \\ B \end{array}}{A \ \Box\!\!\rightarrow B} \tag{2.7}$$

which allows us to reduce the Lewis criterion for counterfactuals in the form in which we use it, to explain the causal-factor relation between facts A and B rather than fictions, to

$$\frac{\begin{array}{c} A \wedge B \\ \neg A \ \Box\!\!\rightarrow \neg B \end{array}}{A \Rightarrow B} \tag{2.8}$$

2.8 Investigation of ATC Procedures

To find the remaining clues to complete the graph, we need to learn about the procedures performed by ATC instances. Since we do not have the knowledge about these procedures yet, acquisition of expert knowledge on the procedures involved is necessary. From [Nol94] we learn that the main tasks to be performed by air traffic control centers (ATCC) concerning the flight information (FI) are:

1. receiving the FI from previous ATCC
2. updating the FI
3. transmitting the FI to the next ATCC

There is also a transmission constraint:

- the FI transmitted from the previous ATCC is reliably (no losses; without alteration) received by the next ATCC (stated in [LL98b]).

According to these constraints, states of one possible world would be:

$\langle 3a \rangle$ ATC procedures are implemented correctly
$\langle 3b \rangle$ FI are correct
$\langle 3c \rangle$ AC position and direction (PD) of flight is consistent with FI

This is a general situation that can naturally be partitioned into three, one for SATC:

$\langle 31 \rangle$ SATC procedures are implemented correctly
$\langle 32 \rangle$ FI are correct at SATC
$\langle 33 \rangle$ AC-PD is consistent with FI at SATC

LATC:

$\langle La \rangle$ LATC procedures are implemented correctly
$\langle Lb \rangle$ FI are correct at LATC
$\langle Lc \rangle$ AC-PD is consistent with FI at LATC

and **B**ATC:

$\langle Ba \rangle$ BATC procedures are implemented correctly
$\langle Bb \rangle$ FI are correct at BATC
$\langle Bc \rangle$ AC-PD is consistent with FI at BATC

We must also make the connection (again, this is *expert knowledge*) amongst the FI (consisting of flight data and flight plan) at the three centers:

$\{SLtrans\}$ FI from SATC are transmitted without change to LATC
$\{LBtrans\}$ FI from LATC are transmitted without change to BATC

Investigating the transmission processes ("*transmitted without change*") results in a further decomposition:

$[SLhuman]$ FI from SATC are correctly entered by SATC controller for transmission to LATC
$[SLchannel]$ FI from SATC are transmitted reliably (no loss, no change) over the channel to LATC
$[LBhuman]$ FI from LATC are correctly entered by LATC controller for transmission to BATC
$[LBchannel]$ FI from LATC are transmitted reliably (no loss, no change) over the channel to BATC

Furthermore we need to make assumptions on the effects the flight information has concerning the aircraft's flightpath:

$[StoL]$ FI from SATC imply next ATC is LATC
$[LtoM]$ FI from LATC imply next ATC is Maastricht ATC (MATC)
$[LtoB]$ FI from LATC imply next ATC is BATC

If all these hypotheses were correct, this should have led to $\langle 2b \rangle$ and finally to $[2b11]$. Since we know, this did not happen, we can conclude that at least one of the states listed above did not occur. From our sources we can take assume that $\{3a\}$ holds. We cannot objectively say, whether $\langle 3b \rangle$ or $\langle 3c \rangle$ or both are false. Let us assume $\langle 3c \rangle$[4]. We then can conclude that $\langle 3b \rangle$ is false, though either the data or the flight plan (or both) transmitted to aircraft and BATC by LATC was erroneous. Consequently, we need to determine, whether

[221] London received false data from SATC
or
[222] data was falsified during processing in LATC.

Although both possibilities are given in [dWL95], we learn from [Lad95a] that [221] was asserted.

We need to analyze the relation "$\langle 4 \rangle \hookrightarrow \langle 3 \rangle$" to replace it by a causal explanation. To illustrate this last step, let us take it here as true that [221]. Since we know about the procedures performed by ATC instances now, we should include four more nodes in the graph:

$\langle 31 \rangle$ SATC procedures are implemented correctly
$\langle 32 \rangle$ FI is correct at SATC
$\langle 33 \rangle$ AC position and direction of flight is consistent with FI at SATC
$[StoL]$ FI from SATC imply next ATC is LATC

[4] We also could assume $\langle 3b \rangle$ at this stage. This is just to define a starting point for further investigations.

This leads to a graph totally based on causality (figure 2.6).

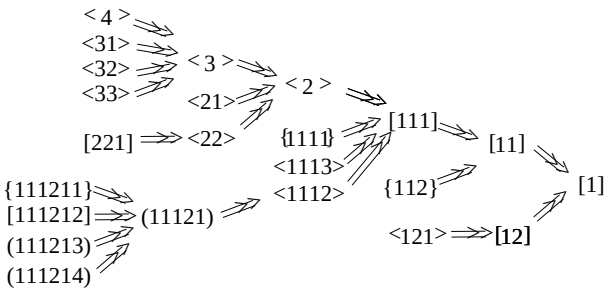

Figure 2.6: Fully causality-based WB-Graph

Given the detailed description of ATC Procedure as presented above, it is a straightforward task to write a specification. We present one written in the TLA+ specification language in chapter 3.

2.9 A textual Graph

Apart from the graphical representation presented above, it is sometimes helpful to also have a textual representation of the graph. In fact, we experienced that an analysis based on this textual WB-graph is sometimes less error-prone, since working on a graphical level results in reasoning errors based on more intuitive instead of purely logical reasoning (see chapter 5).
The Syntax used for the textual WB-Graph is described through an enhanced Backus-Normal-Form (EBNF)[5] – see figure 2.7.

The textual graph of figure 2.6 according to these conventions is presented in figure 2.8

2.10 Adapting PADs to describe possible worlds

Further analysis will be based mainly on assumptions, although it is not absolutely speculation.
As we discovered in our analysis, under the assumption that wrong data were sent by SATC that would be one of the incident's fundamental causes. We would like to use this point to find out more about handling assumptions, since finding why erroneous data was entered to their computers might supply hints on what to change to prevent such incidents in future.
Before we start arguing on assumptions, we introduce a little more notation to differentiate assumed nodes from those based on facts. We denote assumed states by "⟪**A**⟫", assumed events by "⟦**A**⟧", assumed processes by "{{**A**}}" and assumed non-events by "((**A**))".

[5] There are two expressions in this figure which are not part of EBNF:

in the definition of *anychar*:
"*char(0)...char(127)" means "an arbitrary printable ASCII character/letter.*"

in the definition of *descr*:
"*E*XCEPT" means "an arbitrary string except '*' directly followed by '/'."

textgraph = *node* { *node* }.
node = *tag tag* | *tag node_ext node_info*
| *tag* "∧" *node_ext node_info* { "∧" *node_ext node_info* }.
tag = "⟨" *path* "⟩" | "[" *path* "]" | "{" *path* "}" | "(" *path* ")".
node_ext = "⟨−." *number* "⟩" | "[−." *number* "]" | "{−." *number* "}"
| "(−." *number* ")" | "⟪−." *number* "⟫" | "⟦−." *number* "⟧"
| "{{−." *number* "}}" | "((−." *number* "))".
node_info = "/∗" *descr* [*add_info*] "∗/".
add_info = ["//#" *fail_type* "#"] { "//" *add_flags* }.

path = *number* { "." *number* }.
number = *digit* | *firstdigit* { *digit* }.
add_flags = "**@T**" *timestamp add_flags* | "**@A**" *number add_flags* | *descr*.
timestamp = *digit digit* ":" *digit digit* "'" *digit digit* "''".

anychar = *char*(0) . . . *char*(127).
anytext = { *anychar* }.
descr = *anytext EXCEPT* "∗/" .
fail_type = "**PERCEPTION**" | "**ATTENTION**" | "**REASONING**"
| "**DECISION**" | "**INTENTION**" | "**ACTION**".
digit = "**1**" | "**2**" | "**3**" | "**4**" | "**5**" | "**6**" | "**7**" | "**8**" | "**9**" | "**0**".
firstdigit = "**1**" | "**2**" | "**3**" | "**4**" | "**5**" | "**6**" | "**7**" | "**8**" | "**9**".

Figure 2.7: Textual WB-Graph Syntax representation in EBNF

The question, we would like to tackle now is:
Why did SATC send wrong flight data?

We assume that

⟪**2211**⟫ correct information was provided to SATC,
⟦**2212**⟧ information is recognized by air traffic controller
and
⟪**2213**⟫ communication of data between SATC and its adjacent ATC is error-free,

are true – this need not necessarily be so, it is just *one* way to base an argument. To argue e.g. that

⟦**2214**⟧ ATC controller entered erroneous data into computer

we can use different reasoning, depending on what else we assume. Let *Controller* stand for the Shannon controller. Saying that

⟪**2215**⟫ Controller correctly recognized information displayed
and
⟦**2216**⟧ Controller made typing error

would lead to ⟦**2214**⟧ as well as would

⟪**2217**⟫ Controller incorrectly recognized displayed information and
⟦**2218**⟧ Controller typed recognized information correctly.

This reasoning is not deterministic, either of these two circumstances could have caused ⟦**2214**⟧. To make this clear, we adapt *predicate action diagrams* (PADs),

```
[1] /* AC lands at Brussels RWY 25 */
  /\[-.1] /* Crew (CRW) realizes they are landing at the wrong airport */
  /\[-.2] /* CRW opts to continue landing */

  [1.1] /\[-.1] /* CRW gets visual contact to Brussels airport */
        /\{-.2} /* CRW notices that Brussels' airport layout is different
                   from Frankfurt's */

    [1.1.1] /\[-.1] /* AC breaks out under clouds */
            /\<-.2> /* AC near Brussels Airport */
            /\<-.3> /* CRW procedures */
            /\ <2>  /* AC in BATC area */

     <1.1.1.2> (-.1) /* CRW did not realize that they were on wrong course,
                        UNTIL:[111] */

       (1.1.1.2.1) /\{-.1} /* CRW addresses BATC controller as
                              ''Frankfurt'' several times  */
                   /\<-.2> /* ILS has different frequency for Frankfurt */
                   /\[-.3] /* CRW asks for the Bruno VOR's frequency */
                   /\(-.4) /* Brussels did not question the addressing error
                              although it happened more than once */
    <2> /\<-.1> /* LATC procedures */
        /\<-.2> /* LATC uses false flight data for NW052*/
        /\ <3>  /* AC in LATC area */

      <2.2> [-.1] /* London received false data from SATC */

      <3> /\<-.1> /* SATC handoff procedures under this flightplan
                     are to LATC */
          /\<-.2> /* FI is correct at SATC */
          /\<-.3> /* AC-PD is sufficiently consistent with FI at SATC */
          /\ <4>  /* AC in SATC area */

  [1.2] <-.1> /* CRW has safety reasons for continuing landing */
```

Figure 2.8: Textual WB-Graph of the Sample Analysis so far

introduced by Lamport [Lam95] as a graphical notation. We do not base them on the language of TLA, but on EL. Since we speak of procedures and obligations for computer and physical systems as well as organizational procedures, to say that a system defined as a state machine is specified as having state S_2 follow state S_1 when action A is performed is to say that the performance of action A in state S_1, along with the specification, explains why state S_2 happened. In TLA, an implication is provable if and only if (iff) it is a logical truth. So in our formulation, this implication must be interpreted as logical implication, as already axiomatized in axiom 5.

Using this now to interpret the PAD, it means that

$$\mathit{Spec} \wedge S_1 \wedge A \Rrightarrow S_2$$

because

$$\mathit{Spec} \wedge S_1 \wedge A \succ S_2$$

Thus our adaptation of PADs is formally similar to their suggested use by Lamport for specification, but we interpret the transitions as explanatory links also:

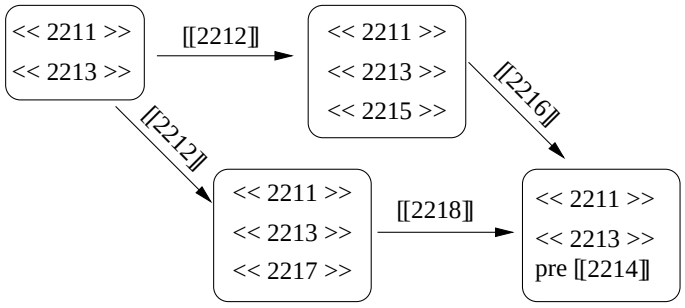

Figure 2.9: Predicate Action Diagram for explanation of assumed state 2214

2.11 Summary

We analyzed a relatively small example of an incident report. By explaining how to handle specific situations by using logic, specifically causality, as well as non-deterministic caused possibility, we informally described the use of multimodal reasoning in incident and accident analysis resulting in both a graphical and textual version of a "Why..Because"-graph.

For our sample case, we detected three major causes:

- [221] London received false data from SATC,
- (11121) CRW did not realize, they were on wrong course, UNTIL:[11]
- (111214) Brussels did not question the addressing error although it happened more than once

To find out, whether the causes found suffice to explain the incident (i.e. whether the *causal completeness test* (CCT) – see appendix C– is fulfilled for all internal nodes of the graph) or whether there are perhaps several causes we missed yet, a formal proof is needed. The logic EL and its inference rules allow us to do so, as we shall demonstrate in chapter 4.

Chapter 3

Specification of Subsystems

The main task of formal specifications[1] is to describe the desired entities in a formally correct way. The descriptions are written in a language that has a well defined semantics, so the specification can be compared against – or executed by a machine or computer program. The language we will use is TLA+, the specification language for the Temporal Logic of Actions [Lam94]. Although the logic EL is an enhancement of TLA, the principles of TLA+ are sufficient for our analysis, too.

3.1 Introducing TLA+

TLA+ has - like every formal language - a well-defined syntax and semantics. We will give an informal description of both in the following paragraphs[2].

A TLA+ specification is organized as a collection of modules with a well-defined appearance. We will take module *ATCtrans* (figure 3.1) as an example, since it uses most of the TLA+ features we introduce.

The module is delimited from other modules by bars on top and bottom. The top bar contains the module's name. Between these bars the module's body is placed. The body is a list of statements, where a statement can be a declaration, a definition, an assumption or a theorem. To improve the readability, horizontal lines (midbars) may be introduced to mark a separation of module parts. These midbars are purely decorative and do not have any formal meaning. Every part of the module may be given a name at will. The five parts of the module *ATCtrans* are named **DECLARATIONS**, **ASSUMPTIONS**, **PREDICATES**, **ACTIONS** and **DEFINITION**. Note that there not necessarily need to be a relation between the name of a module's part and it's contents. In the example module, section **DECLARATIONS** contains mainly declarations and **ASSUMPTIONS** contains assumptions, but formally **PREDICATES**, **ACTIONS** and **DEFINITION** all contain definitions. We separated the definitions into three parts to increase the readability. The **PREDICATES** part contains definitions of predicates, i.e. Boolean expressions containing constants and variables. Actions – Boolean expressions containing constants, variables and primed variables – are placed in the **ACTIONS** part. In the specifications presented in this thesis, we will place a definition of the relations between actions (usually named *SomeAction* or *Spec*) defined in a module, as well as safety and sometimes liveness-conditions, in the **DEFINITION** part at the module's bottom.

[1] We are not concerned about informal specifications in this work. These are documents written in an (often natural, and therefore unprecise) language, readable by humans, but less useful for technical implementations we intend to achieve.

[2] Further information can be found in [Lam96], [Lam97] and [Blu97].

module *ATCtrans*

DECLARATIONS

extends *Naturals*, *Sequences*

CONSTANTS *ChannelSize*, *Messages*, $\perp_{Messages}$, *msg*

VARIABLE *channel*

ASSUMPTIONS

ASSUME $ChannelSize > 0$

ASSUME $\perp_{Messages} \notin Messages$

ASSUME $msg \in Messages \Rightarrow \exists\, fid, fdata : msg = \langle fid, fdata \rangle$

PREDICATES

$Init \triangleq channel = \langle\rangle$

ACTIONS

$$\begin{aligned} Send(msg) \triangleq\ & \land\ msg \in Messages \\ & \land\ Len(channel) < ChannelSize \\ & \land\ channel' = channel \circ \langle msg \rangle \\ Receive \triangleq\ & \land\ channel \neq \langle\rangle \\ & \land\ channel' = Tail(channel) \end{aligned}$$

DEFINITION

$$\begin{aligned} SomeAction \triangleq\ & \lor \exists\, msg : Send(msg) \\ & \lor Receive \\ Safety \triangleq\ & \land Init \\ & \land \Box[SomeAction]_{channel} \end{aligned}$$

Figure 3.1: Sample Module "*ATCtrans*"

Declaration -statements are marked by keywords CONSTANT, CONSTANTS, VARIABLE or VARIABLES. Module *ATCtrans* defines the constant symbols *ChannelSize*, *Messages*, $\perp_{Messages}$ and *msg* as well as the variable symbol *channel*.
Note that there are no types available in TLA+, since it is based on the untyped formalism of Zermelo-Fraenkel set theory with the axiom of Choice [Lam97].
Apart from the constant and variable symbols described above which may be used free once they are declared, in formulas there's also the possibility to use variables bound to an existential or universal quantifier. All symbols bound in formulas as well as symbols used as free variables in the module must be named uniquely.

Definition -statements are always stated in a form

$$symbol \triangleq formula,$$

where *formula* can be any valid formula of TLA. The symbols defined can also have parameters (e.g. *Send(msg)*), which are mapped to symbols used in the formula it defines. We will argue more on definitions below.

Assumption -statements are marked by the keyword ASSUME followed by an expression `expr`. The symbol `expr` might contain can be declared or defined anywhere in the module. The only restriction made to `expr` is that it has to be a valid TLA formula. Since we use the logic EL , we have to modify this restriction. We allow `expr` to be any valid EL formula. Module *ATCtrans* gives samples of assumptions representing set theory and algebraic formulas.

Theorem -statements are marked by the keyword THEOREM followed by an expression `expr`. The constraints concerning `expr` for theorems are the same as those for assumptions. We will use theorems in the correctness proof only.

For system specifications particularly *actions* are of interest. In TLA+ *actions* are formulated as definitions, describing the change of variable's value(s) under defined circumstances. So formally every system specified in TLA+ is a state machine and the specified actions describe the transitions between states. To distinguish between the value before and after the action was executed, the *prime* operator " ′ " is introduced. Take action *Receive* for example:

$$\begin{aligned} Receive \triangleq\ & \wedge\ channel \neq \langle\rangle \\ & \wedge\ channel' = Tail(channel) \end{aligned}$$

The meaning of this definition is:

> Given that *channel* is not an empty sequence, after the "execution" of *Receive* it's value is defined to be its current value reduced by the first element of the sequence.

Note that in TLA+ definitions of actions must not only describe the behaviour of distinct variables that change values, but also that the remaining free variables are left unchanged. We use only one free variable in this example; if we had more than one, the definition had to be supplemented by UNCHANGED-statements for the remaining variables. (See module *Landing_Specs* for example.)
There are also *action operators* provided in TLA+, which allow to state that an action effects a variable or not. For example we can state that *Receive* results in a change of variable *channel* by:

$\langle Receive \rangle_{channel}$ which equals to $Receive \wedge (channel' \neq channel)$

Alternatively we can state that either some action occurs or the variables remain unchanged, by:

$\Box[SomeAction]_{channel}$ which equals to $\Box(SomeAction \vee (channel' = channel))$

We will use this notation later in this chapter to formulate definitions concerning *safety*.

For the explanation of *Receive* we used several definitions concerning sequences. Sequences are not TLA+ primitives, but they can be defined using TLA+. Since the specification will grow with the complexity of the specified system, at some point it is foreseeable that it will be intricate. TLA+ allows to write modularized specifications, which solve this problem. So specifications concerning *Sequences* and *Naturals* can be specified in separate modules[3] and these can be included into new modules. TLA+ provides two ways to achieve this. The statement "**extends** `modulelist`" adds the declarations and definitions as well as assumptions and theorems from the module(s) of `modulelist` to the current module. We use this in *ATCtrans*. The second way is to use one of three possibilities the **instance**-statement gives:

[3]We do not present these modules here. They contain all basic operations on sequences and naturals and are presented in [Lam94].

* "**instance** `modulelist`"
  adds the declarations and definitions, **not** assumptions and theorems, from the module(s) of `modulelist` to the current module. (See module *Landing_Norms* for example.)

* "**instance** `modulename` $x_1 \leftarrow expr_1$, ... , $x_k \leftarrow expr_k$"
  adds the declarations and definitions from module `modulename` to the current module and assigns to each symbol x_i (where $1 \leq x_1 \leq k$) a definition of symbol $expr_i$. Therefore $expr_i$ must be a defined symbol of module `modulename`.

* "`symbol` $\triangleq$ **instance** `modulename` $x_1 \leftarrow expr_1$, ... , $x_k \leftarrow expr_k$"
  is the same as the statement above, except that it defines symbols `symbol`.x_i instead of x_i. This is useful if a module is included more than once to obtain separate copies of the same specification (e.g. to guarantee the same type of behaviour for several components – see module *This_ATCcomm_history* for example).

To allow an argument about the behaviour of a system during a period of time, the predicate logic subset of TLA rules we use so far do not suffice. We need to be able to explain that (a) a statement X always holds or (b) that it does not holt currently, but at some time in the future and then perhaps remains valid or (c) possibly becomes invalid again. TLA+ uses the modal operators $\Diamond$ (at some time in the future) and $\Box$ (always). So we have the means to express the situations listed above:

$$\text{(a) } \Box X \quad \text{(b) } \Diamond\Box X \quad \text{(c) } \Diamond X$$

Now we are able to specify initial conditions, actions and *safety conditions*, i.e. conditions that ensure that either some action occurs, or the variables remain unchanged (cf. definition of symbol *Safety* in module *ATCtrans*). But there is still something missing.
Imagine the systems, we are going to specify are traffic lights at a busy crossroad. It is perhaps interesting to know that the traffic lights *can* give way under particular circumstances. For the function of the system (and the people waiting in the queues) it is essential that these circumstances really *do* occur, so the lights will give way. Furthermore we need to rule out that this just happens once. We must ensure it occurs infinitely often. To state these *liveness* conditions TLA+ provides the concept of *fairness*.
Remember the interpretation of a system specified in TLA as a state machine. An action under this interpretation is a description of a transition between two states (the state before and the state after the execution of the action). An action A is said to be *enabled* ("ENABLED A") in a state s_n, iff there exists a state s_{n+1} such that the pair of these two states satisfies the definition of A. An action therefore can only be executed if it is enabled.
The fairness conditions state:

Weak Fairness Notation: $WF_v[A]$
If A is enabled at a current state and is continually enabled, it must happen – sometime. Referring to the traffic light example this means: If a car is standing at the crossroad and waiting for the traffic light to give way, once the light switches to and remains "green" – enabling the car to continue its way – the car must at some time go ahead to ensure a proper flow of traffic. This proposition concerns only the car; it has no influence on the behaviour of the traffic lights.
The logical formula of Weak Fairness is [Lad97]:

$$WF_v[A] \triangleq \Box\Diamond\neg\text{ENABLED}\,\langle A\rangle_v \vee \Box\Diamond\langle A\rangle_v$$

(We will use Weak Fairness in module *Landing_Specs*.)

Strong Fairness Notation $SF_v[A]$
As we saw in the traffic light example, we also need to insure that the light gives way infinitely often, enabling the waiting cars to go ahead. That is if a formula $\langle A \rangle_v$ is enabled infinitely often, then infinitely many $\langle A \rangle_v$ steps must occur. The logical formula of Strong Fairness is:

$$SF_v[A] \triangleq \Diamond\Box\neg \text{ENABLED } \langle A \rangle_v \vee \Box\Diamond\langle A \rangle_v$$

Notice that if an action ever becomes enabled forever, then it is enabled infinitely often. Therefore:

$SF_v[A] \Rightarrow WF_v[A]$ Strong Fairness implies Weak Fairness!

Provided with all these definitions – it is still not all of TLA and TLA+, but it is all we will need for this work – we are now going to start with the specification.

3.2 Physical Subsystems

Since the semi-formal analysis presented in chapter 2 turned out that a possible error during the transmission of flightdata between the air traffic control centers (ATCCs) – called *Handoff* in aviation terminology – may be a key cause of the incident, we will focus on the function of communication systems at first.

3.2.1 Inter-ATC Communication

The specifications of the ATC handoff procedures are written as TLA+ modules using FIFO channels to handover between ATCCs and using sets to represent the data (the "flight slips") of flights being handled within a specific ATCC.
We already introduced module *ATCtrans* (figure 3.1) in the previous section. It defines the basic actions on the lowest system level. An ATCC is connected to the world by a *channel*. It can send data to this channel or receive data, when it is available from the channel. These definitions of actions of course are insufficient at this level to characterize the whole communication system. However they can be used by modules specifying a more sophisticated level of the system. So we define a hierarchy of modules enhancing the system (and getting closer to reality) step by step. This method is illustrated in figure 3.2.

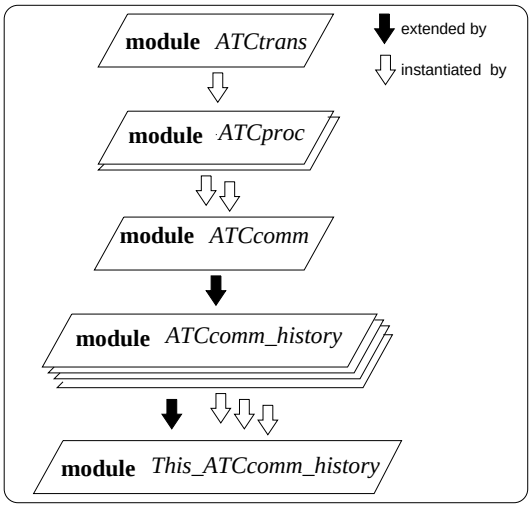

Figure 3.2: Hierarchy of ATC Communications Specifications

Module *ATCproc* (figure 3.3) introduces a storage, where an ATCC stores messages it gets from a channel or which should be sent to a channel. The former is defined by action *Download*, the latter by action *Upload(fid)*. The definitions of these actions use *Send(msg)* and *Receive* from the instantiated module *ATCtrans*. In *ATCproc* assumptions are made that the channel can contain at least one message and that all messages are valid. "To be valid" in this context means that all messages in scope have the form "$\langle fid, fdata \rangle$". An upload of messages with particular flight IDs (fid) to the channel is possible only if this message is currently present in the storage. It is removed from the storage as soon as the message is transmitted. A message can be downloaded only if the maximal storage size is not exceeded.

module *ATCproc*

DECLARATIONS

extends *Naturals*, *Sequences*

CONSTANTS $ChannelSize, StorageSize, Messages, \perp_{Messages}, msg$

VARIABLES $storage, channel$

$ATCtx \triangleq$ **instance** *ATCtrans*

ASSUMPTIONS

ASSUME $ChannelSize > 0$

ASSUME $\perp_{Messages} \notin Messages$

ASSUME $msg \in Messages \Rightarrow \exists\, fid, fdata : msg = \langle fid, fdata \rangle$

PREDICATES

$$Init \triangleq \begin{array}{l} \wedge\ channel = \langle\rangle \\ \wedge\ storage = \langle\rangle \end{array}$$

ACTIONS

$$Upload(fid) \triangleq \exists\, !\, msg \in storage : \begin{array}{l} \wedge\ msg[1] = fid \\ \wedge\ storage' = storage \setminus \{msg\} \\ \wedge\ ATCtx.Send(msg) \end{array}$$

$$Download \triangleq \begin{array}{l} \wedge\ |storage| < StorageSize \\ \wedge\ storage' = storage \cup \{Head(channel)\} \\ \wedge\ ATCtx.Receive \end{array}$$

DEFINITION

$$SomeAction \triangleq \begin{array}{l} \vee\ \exists\, msg : Upload(msg) \\ \vee\ Download \end{array}$$

$$Safety \triangleq \begin{array}{l} \wedge\ Init \\ \wedge\ \Box[SomeAction]_{storage,\, channel} \end{array}$$

Figure 3.3: Module *ATCproc*

At this point, we have specified the upload and download actions one ATCC can. To establish a communication, at least two ATCCs need to be included in the specification. We therefore instantiate *ATCproc* twice (once for each ATCC) in the

next module and define a Handoff action (figure 3.4).

module *ATCcomm*

DECLARATIONS

extends *Naturals*, *Sequences*

CONSTANTS *ChannelSize*, *StorageSize*, *Messages*, $\perp_{Messages}$, *msg*

VARIABLES $x, y, x\_ATCdata, y\_ATCdata, storage, channel$

$X\_ATC \triangleq$ **instance** *ATCproc* **with** $storage \leftarrow x\_ATCdata$

$Y\_ATC \triangleq$ **instance** *ATCproc* **with** $storage \leftarrow y\_ATCdata$

$Vars = \langle x, y, x\_ATCdata, y\_ATCdata, channel \rangle$

ASSUMPTIONS

ASSUME $ChannelSize > 0$

ASSUME $\perp_{Messages} \notin Messages$

ASSUME $msg \in Messages \Rightarrow \exists\, fid, fdata : msg = \langle fid, fdata \rangle$

ASSUME $x, y \in \{SATC, BATC, MATC, LATC, ...\}$

ASSUME $\exists\, !\, msg \in x\_ATCdata : msg[1] = fid$

PREDICATES

$Init \triangleq \land\ fid \in valid\_fids$
$\land\ channel = \langle\,\rangle$
$\land\ x\_ATCdata = \langle\,\rangle$
$\land\ y\_ATCdata = \langle\,\rangle$

ACTIONS

$Handoff(x, y) \triangleq \land\ X\_ATC.Upload(fid)$
$\land\ Y\_ATC.Download$

DEFINITION

$Safety \triangleq \land\ Init$
$\land\ \Box[Handoff(x, y)]_{Vars}$

Figure 3.4: Module *ATCcomm*

In principle, we now have everything we need to explain the communication between ATCCs. However, to use the specification for the proof of possible errors concerning the actions defined, we need to specify under which circumstances we interpret an action to be executed incorrectly. To differentiate, whether the Handoff was done correctly or not, we introduce a *history variable*. Once the value of a history variable is set, it remains set until some action changes it. In figure 3.5 we present the module *ATCcomm_history* using *persistent_data* to state, that the data during handoff was changed or not. We assume *persistent_data* is a record containing the fid and flight data of the aircraft (AC) whose data is transferred currently. In logic we define the handoff to be executed correctly if the history variable remains unchanged. Intuitively one would argue vice versa: The fact that the data has not changed let us assume that it was transferred correctly. Similarly we define an action to be incorrect if the value of the history variable changes during execution. To be able to easily differentiate between these two cases, we split the *Handoff* action into $Handoff_{correct}(x, y)$ and $Handoff_{incorrect}(x, y)$.

module *ATCcomm_history*

DECLARATIONS

extends *Naturals*, *Sequences*, *ATCcomm*

VARIABLE *persistent_data*

CONSTANTS *dest*, *xdests*, *ydests*

ASSUMPTIONS

ASSUME $persistent\_data[destination] = dest$

ASSUME $xdests \cap ydests \neq \emptyset$

PREDICATES

$Init \triangleq persistent\_data = x\_ATCdata$

ACTIONS

$$Handoff_{correct}(x, y) \triangleq \begin{array}{l} \wedge\ persistent\_data[destination] \in xdests \\ \wedge\ Handoff(x, y) \\ \wedge\ \text{UNCHANGED}\ persistent\_data \end{array}$$

$$Handoff_{incorrect}(x, y) \triangleq \begin{array}{l} \wedge\ persistent\_data[destination] \in ydests \\ \wedge\ Handoff(x, y) \\ \wedge\ persistent\_data' \neq persistent\_data \end{array}$$

DEFINITION

$$SomeAction \triangleq \begin{array}{l} \vee\ Handoff_{correct}(x, y) \\ \vee\ Handoff_{incorrect}(x, y) \end{array}$$

$$Safety \triangleq \begin{array}{l} \wedge\ Init \\ \wedge\ \Box[SomeAction]_{persistent\_data} \end{array}$$

Figure 3.5: Module *ATCcomm_history*

Finally, we define the possible communications for this incident (module *This_ATCcomm_history* - figures 3.6 and 3.7). Since three ATCCs are concerned, we need to define three instances of the module *ATCcomm_history* with a complete initialized set of variables for each ATCC. Including a fourth copy of *ATCcomm_history* by an extension statement avoids redefining the history variable and assumptions.

module *This_ATCcomm_history (Part1)*

DECLARATIONS

extends *Naturals*, *Sequences*
extends *ATCcomm_history*
CONSTANTS *ChannelSize*, *StorageSize*, *Messages*, $\perp_{Messages}$, *msg*
VARIABLES x, y, *storage*, *channel*, *SATCdata*, *LATCdata*, *MATCdata*
SL_ATCcommhist $\triangleq$ **instance** *ATCcomm_history* **with**
$x\_ATCdata \leftarrow SATCdata$, $y\_ATCdata \leftarrow LATCdata$,
dest $\leftarrow$ "FRA", *persistent_data* $\leftarrow NW052\_data$
xdests $\leftarrow$ {"FRA", "BRU", ...}, *ydests* $\leftarrow$ {"FRA", "BRU", "AMS", ...}
LB_ATCcommhist $\triangleq$ **instance** *ATCcomm_history* **with**
$x\_ATCdata \leftarrow LATCdata$, $y\_ATCdata \leftarrow BATCdata$,
dest $\leftarrow$ "FRA", *persistent_data* $\leftarrow NW052\_data$
xdests $\leftarrow$ {"BRU", "AMS", ...}, *ydests* $\leftarrow$ {"BRU", "AMS", ...}
LM_ATCcommhist $\triangleq$ **instance** *ATCcomm_history* **with**
$x\_ATCdata \leftarrow LATCdata$, $y\_ATCdata \leftarrow MATCdata$,
dest $\leftarrow$ "FRA", *persistent_data* $\leftarrow NW052\_data$
xdests $\leftarrow$ {"FRA", ...}, *ydests* $\leftarrow$ {"FRA", "TRN", ...}

ASSUMPTIONS

ASSUME $ChannelSize > 0$
ASSUME $\perp_{Messages} \notin Messages$
ASSUME $msg \in Messages \Rightarrow \exists\, fid, fdata : msg = \langle fid, fdata \rangle$
ASSUME $x, y \in \{SATC, BATC, MATC, LATC, ...\}$
ASSUME $\exists\, !\, msg \in x\_ATCdata : msg[1] = fid$
ASSUME "FRA" $\notin LB\_ATCcommhist.ydests$
ASSUME "BRU" $\notin LM\_ATCcommhist.ydests$

PREDICATES

$Init \triangleq \wedge\ persistent\_data = SATCdata$
$\wedge\ SATCdata = \langle\,\rangle$
$\wedge\ LATCdata = \langle\,\rangle$
$\wedge\ MATCdata = \langle\,\rangle$

Figure 3.6: Module *This_ATCcomm_history (Part 1)*

module *This_ATCcomm_history (continued)*

ACTIONS

$StoL_{correct} \triangleq SL\_ATCcommhist.Handoff_{correct}(SATC, LATC)$
$StoL_{incorrect} \triangleq SL\_ATCcommhist.Handoff_{incorrect}(SATC, LATC)$
$LtoB_{correct} \triangleq LB\_ATCcommhist.Handoff_{correct}(LATC, BATC)$
$LtoB_{incorrect} \triangleq LB\_ATCcommhist.Handoff_{incorrect}(LATC, BATC)$
$LtoM_{correct} \triangleq LM\_ATCcommhist.Handoff_{correct}(LATC, MATC)$
$LtoM_{incorrect} \triangleq LM\_ATCcommhist.Handoff_{incorrect}(LATC, MATC)$

DEFINITION

$$\begin{aligned} SomeAction \triangleq\ & \vee\ StoL_{correct} \\ & \vee\ StoL_{incorrect} \\ & \vee\ LtoB_{correct} \\ & \vee\ LtoB_{incorrect} \\ & \vee\ LtoM_{correct} \\ & \vee\ LtoM_{incorrect} \end{aligned}$$

$$\begin{aligned} Safety \triangleq\ & \wedge\ Init \\ & \wedge\ \Box[SomeAction]_{persistent\_data} \end{aligned}$$

Figure 3.7: Module *This_ATCcomm_history (Part 2)*

3.3 "Human Subsystems" - The PARDIA Model

3.3.1 The Model

The *PARDIA*-Model is an information processing model of human functions. We use it to *classify* errors, rather than to *explain* them. However, we believe that this classification suffices to provide information about *what* has to be improved. We leave the decision of *how* these improvements are to be done to experts on this field. The levels of our model are:

P*erception* **A***ttention* **R***easoning* **D***ecision* **I***ntention* **A***ction*

(...thus the name PARDIA.)
Probably the easiest way to explain the different levels is to characterize the typical behaviour of an operator in a typical working situation – e.g. the behaviour of a pilot in an operational aviation situation. An operator's interaction with the system is considered to form a sequence as demonstrated in figure 3.8 [GLL97a].

Perception:	An annunciation of the system state is presented to the pilot;
Attention:	the pilot notices the annunciation;
Reasoning:	figures out what are the possible actions to take;
Decision:	decides on an action;
Intention:	forms the intention to carry it through;
Action:	and finally carries it out.

Figure 3.8: The PARDIA information processing model

A similar model, but with only four categories model is used by the american National Aeronautics and Space Administration (NASA) [NAS]:

> "The human operator senses the environment to acquire data, filters that data through mechanisms of perception and attention to turn it into information, processes that information to select decisions and actions, and then communicates intent and effects behavioral interaction with the environment, technologies, or other people".

We use the PARDIA model with six categories, because there are examples of failures lying in each of these separate categories. When the source of the error is classified, we can derive particular improvements and possible points of departure for further investigations, as shown in figure 3.9.

Error class	Classification of possible improvements
Perception:	instrument/cockpit layout
Attention:	user interface design human cognition
Reasoning:	human cognition operator training revision of handbooks
Decision:	operator training revision of handbooks Procedures
Intention:	psychological investigations physiological investigations
Action:	operator training modification of control elements user interface design

Figure 3.9: Possibly necessary improvements derivable from PARDIA errors

3.3.2 Specifying the PARDIA Model

As shown in figure 3.8, we interpret the human information processing as a sequence of actions. Technically we represent this part of human behaviour as a state machine[4]. We introduce two modules, which we believe are sufficient to explain the human information processing.

PARDIA-Axioms

Module *PARDIA-Axioms* (figures 3.10 and 3.11) defines a set of axioms[5] stating the relationships between the attitudes of the model. This module constitutes a part of the meaning of the PARDIA attitudes. Each attitude is defined as a state predicate taking two arguments:

1. An agent X (represented by a variable), as well as
2. an object A (represented by an action or state predicate) she acts on.

To express, e.g. that the Crew attends an open door, we would define:

$$Attend(CRW, open(door))$$

[4]Of course we know that our cognitive-linguistic and psychology colleagues would not accept this simple model of the human behaviour. Neither do we. Again, we do not intend to *explain* human error. Our aim is the *classification* of human error and, as we will show in this thesis and as already shown in [GLL97a], this simple model is sufficient for achieving this goal.

[5]Axioms are formulas we require to be true at any time.

module *PARDIA-Axioms (Part 1)*

DECLARATIONS

CONSTANTS X, A, B

DEFINITIONS

$IsAction(A) \triangleq (\text{ENABLED } A \not\equiv A)$

$IsStatePredicate(A) \triangleq (\text{ENABLED } A \equiv A)$

AXIOMS

$A1 \triangleq Perceive(X, A) \wedge (A \equiv B) \Rightarrow Perceive(X, B)$

$A2 \triangleq Attend(X, A) \wedge (A \equiv B) \Rightarrow Attend(X, B)$

$A3 \triangleq Reason(X, A) \wedge (A \equiv B) \Rightarrow Reason(X, B)$

$A4 \triangleq Decide(X, A) \wedge (A \equiv B) \Rightarrow Decide(X, B)$

$A5 \triangleq Intend(X, A) \wedge (A \equiv B) \Rightarrow Intend(X, B)$

$A6 \triangleq Act(X, A) \wedge (A \equiv B) \Rightarrow Act(X, B)$

$A7 \triangleq O(A) \Rightarrow Reason(X, A)$

$A8 \triangleq Attend(X, A) \Rightarrow Reason(X, A)$

$$A9 \triangleq \begin{pmatrix} \wedge & Act(X, A) \\ \wedge & IsAction(A) \end{pmatrix} \Rightarrow A$$

$$A10 \triangleq \begin{pmatrix} \wedge & Act(X, A) \\ \wedge & IsStatePredicate(A) \end{pmatrix} \Rightarrow A'$$

Figure 3.10: PARDIA Axioms (Part1)

Note that all attitudes are defined as state predicates. They describe the "state" the human mind is in. The transitions between these states are defined as axioms. Some of these axioms require to distinguish whether the object of an attitude is an action or a state predicate. In the first part of the module we therefore introduce two state predicates to provide a logical solution to fulfill this requirement.
A formula A is logically equivalent to a state predicate, if ENABLED A is equivalent to A [LL98b]. We define this point as predicate "*IsStatePredicate(A)*". Similarly A is logically equivalent to an action if above equivalence for a state predicate is not fulfilled.

PARDIA-Norms

Apart from the axioms presented above, we formulate certain principles which are intended to hold. We claim that these principles *ought* to hold in a normal behaviour: At some time the Crew should pay attention to something they perceived; reasoning about the attended should lead to a decision. The decision itself should cause the Crew to create some intention. Finally, an intention should lead to an action. However, the properties of *PARDIA-Norms* are defeasible. Any of these steps may fail, and this failure may be part of an incident history [LL98b].

module *PARDIA-Axioms (continued)*

$$A11 \triangleq \left(\begin{array}{ll} \wedge & Perceive(X, A) \\ \wedge & IsStatePredicate(A) \end{array} \right) \Rightarrow A$$

$$A12 \triangleq \left(\begin{array}{ll} \wedge & Attend(X, A) \\ \wedge & IsStatePredicate(A) \end{array} \right) \Rightarrow A$$

$$A13 \triangleq \left(\begin{array}{ll} \wedge & Decide(X, A) \\ \wedge & \neg Decide(X, \neg A) \end{array} \right) \Rightarrow O(Intend(X, A))$$

$$A14 \triangleq \left(\begin{array}{ll} \wedge & Reason(X, A) \\ \wedge & \neg Reason(X, \neg A) \end{array} \right) \Rightarrow O(Decide(X, A))$$

$$A15 \triangleq \left(\begin{array}{ll} \wedge & Perceive(X, A) \\ \wedge & A \Rightarrow B \end{array} \right) \Rightarrow Perceive(X, B))$$

$$A16 \triangleq \left(\begin{array}{ll} \wedge & Reason(X, A) \\ \wedge & A \Rightarrow B \end{array} \right) \Rightarrow Reason(X, B))$$

$$A17 \triangleq \left(\begin{array}{ll} \wedge & Decide(X, A) \\ \wedge & A \Rightarrow B \end{array} \right) \Rightarrow Decide(X, B))$$

$$A18 \triangleq \left(\begin{array}{ll} \wedge & Act(X, A) \\ \wedge & A \Rightarrow B \end{array} \right) \Rightarrow Act(X, B))$$

$$A19 \triangleq \left(\begin{array}{ll} \wedge & Reason(X, A) \\ \wedge & Reason(X, \neg A) \end{array} \right) \Rightarrow O(Decide(X, A) \vee Decide(X, \neg A))$$

DEFINITION

$$Spec \triangleq \wedge\ A1 \wedge A2 \wedge A3 \wedge A4 \wedge A5 \wedge A6 \wedge A7 \wedge A8 \\ \wedge\ A9 \wedge A10 \wedge A11 \wedge A12 \wedge A13 \wedge A14$$

Figure 3.11: PARDIA Axioms (Part 2)

module *PARDIA-Norms*

DECLARATIONS
CONSTANTS CRW, A

NORMS

$N1 \triangleq Perceive(CRW, A) \leadsto Attend(CRW, A)$

$N2 \triangleq Reason(CRW, A) \leadsto Decide(CRW, A)$

$N3 \triangleq Decide(CRW, A) \leadsto Intend(CRW, A)$

$N4 \triangleq Intend(CRW, A) \leadsto Act(CRW, A)$

DEFINITION

$Spec \triangleq N1 \wedge N2 \wedge N3 \wedge N4$

Figure 3.12: The PARDIA Norms

3.4 Real Time Theorems

For some definitions of the next section it is necessary to make statements about the mathematical behaviour of functions. It is not the aim of this work to go too deep into this topic. We demonstrate how to include analysis into TLA+ by the introduction of the *MeanValueTheorem* (figure 3.13). With the help of module *RealTime* from [Lam93b] this is a straight forward task.

module *RealTimeTheorems*

DECLARATIONS
extends *Naturals*
extends *RealTime*
CONSTANTS $lower\_bound, upper\_bound$
VARIABLES f, t

ASSUMPTIONS
$$MeanValueTheorem \triangleq \left(\begin{array}{ll} \wedge & \frac{\delta f}{\delta t} < 0 \; in \; [lower\_bound, upper\_bound] \\ \wedge & f = upper\_bound \\ \wedge & \Diamond(f = lower\_bound) \end{array} \right) \Rightarrow \forall x \in [lower\_bound, upper\_bound] : \Diamond(f = x)$$

DEFINITIONS
$$RTSpec \triangleq MeanValueTheorem$$

Figure 3.13: Module *RealTimeTheorems*

3.5 Procedures

3.5.1 Specification of Landing Procedures

For the specification of landing procedures we use a collection of hierarchical modules (see figure 3.14).

As in section 3.2, we first define a module containing the basic definitions (module *Landing_Defs*, figure 3.15). These definitions are about the parameters *Decision Height* (DH), *Minimum Decision Altitude* (MDA) and *Touch Down Zone Elevation* (TDZE) which are important during the landing phase[6]. Furthermore, predicates are given, which explain several states concerning the altitude.

On a higher level, in module *Landing_Axioms* (figure 3.16) we formulate axioms covering several aspects of the landing phase[7]. The module includes definitions from *Landing_Defs* and *RealTimeTheorems*. All axioms we introduce are derived from official aviation literature ([Admb],[Adma]) or expert knowledge we gained from personal communications with private and commercial pilots ([Lad98],[Ble98]). They define general procedures, which are always valid, e.g.:

- *Unique_Approach-Type_Rule*: The landing approach is either a Instrument Landing System (ILS) or a Non-Precision (NP) approach.
- *data_inconsist(x,y)*: The data used during the landing phase is defined to be inconsistent if (a) navigational aids (Navaids) are used for the approach,

[6] There are too many abbreviations to explain them all. In case more information should be necessary, an extensive glossary is attached as Appendix D.

[7] To avoid too much mathematics, we introduce a "quick and dirty" solution by stating the assumption *Suitable_Axioms*. We claim that this inaccuracy does not affect the quality of neither the specification, nor the proof. To avoid complicating the proof unnecessarily, we cut out a closer inspection of axioms, which we believe are mathematically trivial enough to be classified as secure knowledge. It is possible to handle these axioms formally correct, as we hope the *RealTimeTheorem*-example demonstrates, but is does not affect the method we wish to present as the main part of this work.

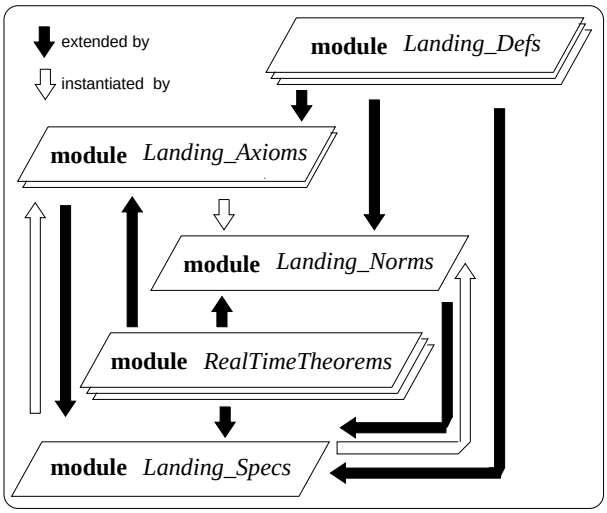

Figure 3.14: Hierarchy of Landing Procedures and Real-Time Specifications

module *Landing_Defs*

DECLARATIONS

CONSTANTS $DH, MDA, TDZE$

VARIABLES alt

DEFINITIONS

$below\_DH \triangleq (alt < DH)$

$below\_MDA \triangleq (alt < MDA)$

$at\_MDA \triangleq (alt = MDA)$

$landing \Rightarrow (alt = TDZE)$

Figure 3.15: Module *Landing_Defs*

but are not on the approach plate, or (b) navigational aids appearing on the approach plate cannot be detected during the landing.

- *CompleteApprPlates*: The approach plates are considered to be complete, iff all Navaids provided for an approach appear on the approach plate for this landing.

There are also several defeasible rules in the specification of the landing procedures. These rules are defined in module *Landing_Norms* (figures 3.17 and 3.18). They ought to be regarded by the participants, but – as the proof will show – most of them at some time are not in this incident.
Let us again pick out selected rules:

- *Unique_APT_Rule*: An aircraft can only be cleared to land at an airport in who's area it is.
- *Landing_Criterion*: An aircraft has visual contact under an acceptable visibility if either on an ILS-approach it goes down below DH or on an NP-approach it reaches or goes below MDA.
- *ILS-APPR_Rule*: An ILS-approach has to be broken off, if the aircraft is already below DH and the visibility becomes unacceptable[8].
- *CallAttentionToErrorRule*: As soon as the Crew or the air traffic controller attends a use of inconsistent data by the respective other communication participant, he/she has to question this problem.
- *UniqueApproachPlates*: Under the assumption that only the most current approach plates are used, two plates of the same approach may not contain inconsistent data.

Up to this point, in the specification of landing procedures we defined rules describing some of the dependencies between state predicates. To specify a system describing the real world, we need more than just rules. Module *Landing_Specs* (figures 3.19, 3.20 and 3.21) introduces actions, safety and liveness conditions, ensuring a proper function of the system.

We would like to focus on the types of state predicates we used so far. Table 3.5.1 presents a list of these predicates. They are sufficient for the current analysis, although we believe additions will be necessary for other analyses.
We distinguish between three kinds of state predicates/boolean variables:

1. *performatives* – speech acts whose utterance performs an action; like "promise", "assert", "request"... (See [Aus75],[Sea69] for further information on this topic.)
2. *environmental* variables describe influences of / interactions with environmental factors to the system.
3. *system* variables.

Performatives and environmental variables often are *history variables*, i.e. once their value is set, it remains set until it is changed by some actions. Environmental variables cannot be changed by system agents, e.g. weather conditions. For future use of this module it might be useful to add conditions taken from SOPs concerning, e.g. when a situation has to be considered as dangerous. We did this in distinct actions used in our particular case (see *Accept_landing*(*CRW*) or *CRW_breakoff*(*CRW*)).

[8] This is an exact translation of [Adma], title 14, Code of Federal Register, Chapter 1, 91.175 (c) 3 (e) 1 ii.

module *Landing_Axioms*

DECLARATIONS

extends *RealTimeTheorems*
extends *Landing_Defs*
instance *Landing_Specs*
CONSTANTS $ac, atc, crw, apt, this\_approach$
VARIABLES x, y

ASSUMPTIONS

ASSUME

$$Suitable\_Axioms \triangleq \left(\begin{array}{c} Suitable\ mathematical\ axioms\ describing\ the\ altitude\ function \\ desired\_alt[APPR] : position \rightarrow altitude \\ for\ the\ APPRs\ described\ in\ the\ approach\ procedures \end{array} \right)$$

AXIOMS

$$APT/ATC\_Id\_Rule \triangleq ((ac)near(apt) \Rightarrow (ac)in\_area(responsible\_atc[apt]))$$

$$Clearance\_Rule \triangleq ((ac)in\_landing\_phase \Rightarrow landing\_accepted(crw))$$

$$Unique\_Approach - Type\_Rule \triangleq (ILS\_approach(ac, apt) \Leftrightarrow \neg NP\_approach(ac, apt))$$

$$Acceptance\_Rule \triangleq \Box(landing\_accepted(crw) \Rightarrow \Box landing\_accepted(crw))$$

$$data\_inconsist(x, y) \triangleq (x \neq y)$$

$$AttendILS{-}Navaids \triangleq ILS\_approach(ac, apt) \Rightarrow navigate(appr\_plate[apt], crw)$$

$$CompleteApprPlates \triangleq \left(\begin{array}{l} y \in Navaids[this\_approach] \triangleq \\ \qquad \exists\, a : \wedge\ this\_approach = appr\_plate[a] \\ \qquad\qquad \wedge\ y \in appr\_plate[a] \end{array} \right)$$

DEFINITION

$$\begin{array}{lll} LASpec & \triangleq & \wedge\ APT/ATC\_Id\_Rule \\ & & \wedge\ Clearance\_Rule \\ & & \wedge\ Unique\_Approach - Type\_Rule \\ & & \wedge\ Acceptance\_Rule \\ & & \wedge\ data\_inconsist(x, y) \\ & & \wedge\ AttendILS{-}Navaids \\ & & \wedge\ CompleteApprPlates \\ & & \wedge\ Suitable\_Axioms \end{array}$$

Figure 3.16: Module *Landing_Axioms*

module *Landing_Norms*

DECLARATIONS

extends *RealTimeTheorems*
extends *Landing_Defs*
instance *Landing_Axioms*
instance *Landing_Specs*
instance *Naturals*
CONSTANTS $ac, atc, crw, apt, tfc, appr, this\_approach$
VARIABLES $some\_apt, current\_alt, acceptable\_deviation$

DEFINITIONS

$$ProceduralDef1 \triangleq \Box visibility\_acceptable(crw) \Rightarrow \Diamond visual\_contact(crw, apt)$$

$$GeneralSafetyRule \triangleq \Box\neg endanger(crw, tfc)$$

$$Landing\_Procedures \triangleq \begin{pmatrix} \wedge & (ac)near(apt) \\ \wedge & \Box(ac)in\_landing\_phase \end{pmatrix} \Rightarrow \Diamond(ac)lands\_at(apt)$$

$$Unique\_APT\_Rule \triangleq \begin{pmatrix} \wedge & (ac)near(apt) \\ \wedge & cleared\_to\_land(atc, crw, some\_apt') \end{pmatrix} \Rightarrow (some\_apt = apt)$$

$$Landing\_Criterion \triangleq \Box \left(\left(\begin{pmatrix} \wedge & visibility\_acceptable(crw) \\ \wedge & \vee \wedge ILS\_approach(ac, apt) \\ & \quad \wedge below\_DH \\ & \vee \wedge NP\_approach(ac, apt) \\ & \quad \wedge \vee at\_MDA \\ & \quad\quad \vee below\_MDA \end{pmatrix} \Rightarrow visual\_contact(crw, apt) \right) \right)$$

$$Landing\_Rule \triangleq \Box \left(landing\_accepted(crw) \Rightarrow \begin{matrix} \vee & \Diamond landing \\ \vee & \Diamond landing\_interrupt(crw, atc, apt) \end{matrix} \right)$$

$$ILS{-}APPR\_Rule \triangleq \begin{pmatrix} \wedge & ILS\_approach(ac, apt) \\ \wedge & below\_DH \\ \wedge & \Diamond\neg visibility\_acceptable(crw) \end{pmatrix} \Rightarrow \Diamond CRW\_breakoff(crw)$$

$$ILS{-}LandingRule \triangleq \begin{pmatrix} \wedge & ILS\_approach(ac, apt) \\ \wedge & \Box\neg CRW\_breakoff(crw) \end{pmatrix} \Rightarrow \Diamond landing$$

$$ILS{-}AltProperty \triangleq \begin{pmatrix} \wedge & ILS\_approach(ac, apt) \\ \wedge & \Box\neg CRW\_breakoff(crw) \end{pmatrix} \Rightarrow \begin{matrix} alt\ is\ a\ 'monotone\ decreasing\ continuous \\ function\ of\ RealTime' \end{matrix}$$

$$Normal{-}Progress \triangleq \begin{pmatrix} \wedge & apt \neq destAPT \\ \wedge & \Box\neg distress\_descl(crw) \\ \wedge & \Box\neg urgency\_decl(crw) \end{pmatrix} \Rightarrow \neg\Diamond(ac)lands\_at(apt)$$

Figure 3.17: Module *Landing_Norms (Part 1)*

module *Landing_Norms (continued)*

$$CallAttentionToErrorRule \triangleq \Box \left(\begin{array}{l} \forall x, y \in \{atc, crw\} : \\ \quad \wedge \quad x \neq y \\ \quad \wedge \quad Attend(x, data\_inconsist(Navaids[this\_approach], \\ \qquad\qquad Navaids[appr\_plate[apt]])) \\ \quad \Rightarrow question(x, y, data\_inconsist(Navaids[this\_approach], \\ \qquad\qquad Navaids[appr\_plate[apt]])) \end{array} \right)$$

$$AttendErrorRule \triangleq \left(\begin{array}{ll} \wedge & A \\ \wedge & question(x, y, A) \end{array} \right) \Rightarrow Attend(y, A)$$

$$UniqueApproachPlates \triangleq \left(\begin{array}{ll} \wedge & current(appr\_plate[a]) \\ \wedge & current(appr\_plate[b]) \\ \wedge & data\_inconsist(appr\_plate[a], appr\_plate[b]) \end{array} \right) \Rightarrow (a \neq b)$$

$$AttendNavaids \triangleq navigate(appr\_plate[apt], crw) \Rightarrow \forall x \in Navaids[appr\_plate[apt]] : Attend(crw, x)$$

$$Deviation\_Breakoff \triangleq (|(current\_alt) - (desired\_alt[appr])| > acceptable\_deviation) \Rightarrow CRW\_breakoff\,(crw)$$

DEFINITION

$$\begin{array}{lll} LNSpec & \triangleq & \wedge\ ProceduralDef\,1 \\ & & \wedge\ GeneralSafetyRule \\ & & \wedge\ Landing\_Procedures \\ & & \wedge\ Unique\_APT\_Rule \\ & & \wedge\ Landing\_Criterion \\ & & \wedge\ Landing\_Rule \\ & & \wedge\ ILS{-}APPR\_Rule \\ & & \wedge\ ILS{-}LandingRule \\ & & \wedge\ ILS{-}AltProperty \\ & & \wedge\ Normal{-}Progress \\ & & \wedge\ CallAttentionToErrorRule \\ & & \wedge\ AttendErrorRule \\ & & \wedge\ UniqueApproachRule \\ & & \wedge\ AttendNavaids \\ & & \wedge\ Deviation\_Breakoff \end{array}$$

Figure 3.18: Module *Landing_Norms* (Part2)

module *Landing_Specs*

DECLARATIONS

extends *RealTimeTheorems*
extends *Landing_Defs*
extends *Landing_Axioms*
extends *Landing_Norms*
CONSTANTS $ATC, CRW, APT, TFC, APPR$

PREDICATES

$$landing\_interrupt(CRW, ATC, APT) \triangleq \left(\begin{array}{ll} \vee & \Box distress\_decl(CRW) \\ \vee & \Box urgency\_decl(CRW) \\ \vee & \Box\neg flightpath\_acceptable(CRW) \\ \vee & endanger(CRW, CRW) \\ \vee & \neg cleared\_to\_land(ATC, CRW, APT) \\ \vee & \neg visibility\_acceptable(CRW) \end{array} \right)$$

$$\begin{array}{lll} Init & \triangleq & \wedge\ (ac)near(apt) \\ & & \wedge\ \neg distress\_decl(crw) \\ & & \wedge\ \neg urgency\_decl(crw) \\ & & \wedge\ flightpath\_acceptable(crw) \\ & & \wedge\ \neg endanger(crw, crw) \\ & & \wedge\ \neg cleared\_to\_land(atc, crw, apt) \\ & & \wedge\ \neg landing\_accepted(crw) \\ & & \wedge\ visibility\_acceptable(crw) \end{array}$$

ACTIONS

$$\begin{array}{lll} Accept\_flightpath(CRW) & \triangleq & \wedge\ \neg flightpath\_acceptable(CRW) \\ & & \wedge\ flightpath\_acceptable(CRW)' \\ & & \wedge\ \text{UNCHANGED}\ (AC)near(APT) \\ & & \wedge\ \text{UNCHANGED}\ (AC)in\_landing\_phase \\ & & \wedge\ \text{UNCHANGED}\ cleared\_to\_land(ATC, CRW, APT) \\ & & \wedge\ \text{UNCHANGED}\ distress\_decl(CRW) \\ & & \wedge\ \text{UNCHANGED}\ landing\_accepted(CRW) \\ & & \wedge\ \text{UNCHANGED}\ urgency\_decl(CRW) \\ & & \wedge\ \text{UNCHANGED}\ endanger(CRW, CRW) \\ & & \wedge\ \text{UNCHANGED}\ visibility\_acceptable(CRW) \end{array}$$

$$\begin{array}{lll} Accept\_landing(ATC, CRW, APT) & \triangleq & \wedge\ visibility\_acceptable(CRW) \\ & & \wedge\ \neg distress\_decl(CRW) \\ & & \wedge\ \neg urgency\_decl(CRW) \\ & & \wedge\ flightpath\_acceptable(CRW) \\ & & \wedge\ \neg endanger(CRW, CRW) \\ & & \wedge\ cleared\_to\_land(ATC, CRW, APT) \\ & & \wedge\ \neg landing\_accepted(CRW) \\ & & \wedge\ landing\_accepted(CRW)' \\ & & \wedge\ \neg(AC)in\_landing\_phase \\ & & \wedge\ (AC)in\_landing\_phase' \\ & & \wedge\ \text{UNCHANGED}\ (AC)near(APT) \\ & & \wedge\ \text{UNCHANGED}\ cleared\_to\_land(ATC, CRW, APT) \\ & & \wedge\ \text{UNCHANGED}\ flightpath\_acceptable(CRW) \\ & & \wedge\ \text{UNCHANGED}\ distress\_decl(CRW) \\ & & \wedge\ \text{UNCHANGED}\ urgency\_decl(CRW) \\ & & \wedge\ \text{UNCHANGED}\ endanger(CRW, CRW) \\ & & \wedge\ \text{UNCHANGED}\ visibility\_acceptable(CRW) \end{array}$$

Figure 3.19: Module *Landing_Specs (Part 1)*

module *Landing_Specs (continued)*

$$
\begin{aligned}
\mathit{Cleared\_to\_land}(ATC, CRW, APT) \triangleq\ & \wedge \neg \mathit{cleared\_to\_land}(ATC, CRW, APT) \\
& \wedge \mathit{cleared\_to\_land}(ATC, CRW, APT)' \\
& \wedge \text{UNCHANGED } (AC)\mathit{near}(APT) \\
& \wedge \text{UNCHANGED } (AC)\mathit{in\_landing\_phase} \\
& \wedge \text{UNCHANGED } \mathit{flightpath\_acceptable}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{distress\_decl}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{landing\_accepted}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{urgency\_decl}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{endanger}(CRW, CRW) \\
& \wedge \text{UNCHANGED } \mathit{visibility\_acceptable}(CRW)
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Decl\_distress}(CRW) \triangleq\ & \wedge \neg \mathit{distress\_decl}(CRW) \\
& \wedge \mathit{distress\_decl}(CRW)' \\
& \wedge \text{UNCHANGED } (AC)\mathit{near}(APT) \\
& \wedge \text{UNCHANGED } (AC)\mathit{in\_landing\_phase} \\
& \wedge \text{UNCHANGED } \mathit{flightpath\_acceptable}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{cleared\_to\_land}(ATC, CRW, APT) \\
& \wedge \text{UNCHANGED } \mathit{landing\_accepted}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{urgency\_decl}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{endanger}(CRW, CRW) \\
& \wedge \text{UNCHANGED } \mathit{visibility\_acceptable}(CRW)
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Decl\_urgency}(CRW) \triangleq\ & \wedge \neg \mathit{urgency\_decl}(CRW) \\
& \wedge \mathit{urgency\_decl}(CRW)' \\
& \wedge \text{UNCHANGED } (AC)\mathit{near}(APT) \\
& \wedge \text{UNCHANGED } (AC)\mathit{in\_landing\_phase} \\
& \wedge \text{UNCHANGED } \mathit{flightpath\_acceptable}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{cleared\_to\_land}(ATC, CRW, APT) \\
& \wedge \text{UNCHANGED } \mathit{distress\_decl}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{landing\_accepted}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{endanger}(CRW, CRW) \\
& \wedge \text{UNCHANGED } \mathit{visibility\_acceptable}(CRW)
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Consider\_danger}(CRW) \triangleq\ & \wedge \neg \mathit{endanger}(CRW, CRW) \\
& \wedge \mathit{endanger}(CRW, CRW)' \\
& \wedge \text{UNCHANGED } (AC)\mathit{near}(APT) \\
& \wedge \text{UNCHANGED } (AC)\mathit{in\_landing\_phase} \\
& \wedge \text{UNCHANGED } \mathit{flightpath\_acceptable}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{cleared\_to\_land}(ATC, CRW, APT) \\
& \wedge \text{UNCHANGED } \mathit{distress\_decl}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{landing\_accepted}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{urgency\_decl}(CRW) \\
& \wedge \text{UNCHANGED } \mathit{visibility\_acceptable}(CRW)
\end{aligned}
$$

$$
\begin{aligned}
\mathit{CRW\_breakoff}(CRW) \triangleq\ & \wedge \mathit{landing\_accepted}(CRW) \\
& \wedge (AC)\mathit{in\_landing\_phase} \\
& \wedge \neg (AC)\mathit{in\_landing\_phase}' \\
& \wedge \vee \mathit{Decl\_distress}(CRW) \\
& \quad \vee \mathit{Decl\_urgency}(CRW) \\
& \quad \vee \neg \mathit{Accept\_flightpath}(CRW) \\
& \quad \vee \mathit{Consider\_danger}(CRW) \\
& \quad \vee \neg \mathit{visibility\_acceptable}(CRW) \\
& \wedge \text{UNCHANGED } (AC)\mathit{near}(APT) \\
& \wedge \text{UNCHANGED } \mathit{cleared\_to\_land}(ATC, CRW, APT) \\
& \wedge \text{UNCHANGED } \mathit{landing\_accepted}(CRW)
\end{aligned}
$$

Figure 3.20: Module *Landing_Specs (Part 2)*

module *Landing_Specs (continued)*

$$
\begin{array}{lll}
ATC\_breakoff(ATC, CRW, APT) & \triangleq & \wedge\ cleared\_to\_land(ATC, CRW, APT) \\
& & \wedge\ \neg cleared\_to\_land(ATC, CRW, APT)' \\
& & \wedge\ \text{UNCHANGED}\ (AC)near(APT) \\
& & \wedge\ \text{UNCHANGED}\ (AC)in\_landing\_phase \\
& & \wedge\ \text{UNCHANGED}\ flightpath\_acceptable(CRW) \\
& & \wedge\ \text{UNCHANGED}\ distress\_decl(CRW) \\
& & \wedge\ \text{UNCHANGED}\ landing\_accepted(CRW) \\
& & \wedge\ \text{UNCHANGED}\ urgency\_decl(CRW) \\
& & \wedge\ \text{UNCHANGED}\ endanger(CRW, CRW) \\
& & \wedge\ \text{UNCHANGED}\ visibility\_acceptable(CRW)
\end{array}
$$

DEFINITION

$$
\begin{array}{lll}
Rules & \triangleq & \wedge\ ProceduralDef1 \\
& & \wedge\ APT/ATC\_Id\_Rule \\
& & \wedge\ Landing\_Procedures \\
& & \wedge\ Clearance\_Rule \\
& & \wedge\ Unique\_APT\_Rule \\
& & \wedge\ Unique\_Approach - Type\_Rule \\
& & \wedge\ Landing\_Criterion \\
& & \wedge\ Landing\_Rule \\
& & \wedge\ Acceptance\_Rule \\
& & \wedge\ LandingSafetyRule \\
& & \wedge\ ILS-APPR\_Rule \\
& & \wedge\ ILS-LandingRule \\
& & \wedge\ ILS-AltProperty \\
SomeAction & \triangleq & \vee\ Accept\_flightpath(crw) \\
& & \vee\ Accept\_landing(crw) \\
& & \vee\ Cleared\_to\_land(atc, crw, apt) \\
& & \vee\ Decl\_distress(crw) \\
& & \vee\ Decl\_urgency(crw) \\
& & \vee\ Consider\_danger(crw) \\
& & \vee\ CRW\_breakoff(crw) \\
& & \vee\ ATC\_breakoff(atc) \\
Spec & \triangleq & \wedge\ Init \\
& & \wedge\ Rules \\
& & \wedge\ \Box[SomeAction] \\
& & \wedge\ \text{WF}[cleared\_to\_land(atc, crw, apt)] \\
& & \wedge\ \text{WF}[Accept\_landing(crw)]
\end{array}
$$

Figure 3.21: Module *Landing_Specs* (Part 3)

state predicate	classification
$(ac)in\_landing\_phase$	system
$(ac)in\_area(atc)$	system
$(ac)lands\_at(apt)$	system
$(ac)near(apt)$	system
$appr\_plate[apt]$	system
at_MDA	system
$below\_DH$	system
$below\_MDA$	system
$cleared\_to\_land(crw, atc, apt)$	performative
$current(a)$	system
$det\_destAPT(ac)$	system
$distress\_decl(crw)$	performative
$endanger(crw, tfc)$	environment/performative
$fdata(ac)[field]$	system
$flightpath\_acceptable(crw)$	environment/performative
$intermediateATC(ac, apt1, apt2)$	system
$ILS\_approach(ac, apt)$	system
$landing$	system
$landing\_accepted(crw)$	performative
$landing\_interrupt$	(disjunct of predicates)
$LOC(a)$	system
$navigate(a, x)$	system
$NP\_approach(ac, apt)$	system
$position(ac)$	system
$question(a, y, a)$	performative
$responsible\_atc(apt)$	system
$urgency\_decl(crw)$	performative
$visibility\_acceptable(crw)$	environment/performative
$visual\_contact(crw, apt)$	system

Figure 3.22: State Predicates from Specifications of Landing Procedures

3.5.2 Standard Operating Procedures

Finally we define several Standard Operating Procedures (SOPs) concerning the ATCCs tasks (module *SOP_Specs*, figure 3.23). Again, we make an assumption to reduce the complexity of the specification (and therefore the proof later on). We claim that destAPT is the only important information to determine the En Route ATCs[9]. However, we are convinced that the specified system still provides the basis for a sufficient proof.

[9] Usually the flight information contains: AC call sign, AC type, transponder code, proposed departure time, requested altitude, destination airport and route of flight [[Nol94], p.412].

module *SOP_Specs*

DECLARATIONS

extends *Naturals*, *Sequences*
extends *ATCcomm*, *ATCcomm_history*
extends *Landing_Specs*
CONSTANTS $atc, crw, apt, destATC, nextATC, tfc$
VARIABLES x, y

ASSUMPTIONS

ASSUME $\forall\, ac, (ac)in\_area(atc) \;:\; \exists\,!\, msg \in storage(atc) \;:\; msg[1] = fid(ac)$

DEFINITIONS

$$RespDestATC\_Determination\_Rule \triangleq (det\_destATC(ac) = responsible\_atc(det\_destAPT(ac)))$$

$$DetermineDestinationProcedure \triangleq (det\_destAPT(ac) = fdata(ac)[destAPT])$$

$$nextATC(ac) \triangleq \lor\, det\_destATC(ac) \lor\, intermediateATC(ac, det\_destAPT(ac), destAPT(AC))$$

$$(ac)in\_area(atc) \triangleq position(ac) \in region(atc)$$

$$false\_FI(atc, ac) \triangleq (det\_destAPT(ac) \neq destAPT(ac))$$

$$Inter\_ATC\_Handoff(x, y, ac) \triangleq \left(\begin{array}{ll} \land & (ac)in\_area(x) \\ \land & (ac)in\_area(y) \\ \land & nextATC(ac) = y \\ \land & \Diamond \left(\begin{array}{ll} \land & \neg(ac)in\_area(x) \\ \land & (ac)in\_area(y) \end{array} \right) \\ \land & \left(\begin{array}{ll} \lor & ATCcomm\_history.Handoff_{\,correct}(x, y, fid(ac)) \\ \lor & ATCcomm\_history.Handoff_{\,incorrect}(x, y, fid(ac)) \end{array} \right) \end{array} \right)$$

$$EnRouteProcessing(atc, ac) \triangleq \left(\begin{array}{ll} \land & (ac)in\_area(atc) \\ \land & det\_destAPT(ac) = fdata(ac)[destAPT] \\ \land & atc \neq det\_destATC(ac) \end{array} \right) \Rightarrow \Diamond Inter\_ATC\_Handoff(atc, nextATC(ac), ac)$$

$$ATC{-}Responsibility{-}Rule \triangleq (ac)in\_area(atc) \Rightarrow EnRouteProcessing(atc, ac)$$

DEFINITION

$$Spec \triangleq \land\, ATC{-}Responsibility{-}Rule \land\, DetermineDestinationProcedure \land\, EnRouteProcessing(atc, ac) \land\, Inter\_ATC\_Handoff(x, y, ac)$$

Figure 3.23: Module *SOP_Specs*

3.6 Summary

We presented a short introduction to the syntax and semantics of selected features of TLA+ we need for the specifications and the proof.
Furthermore we described the PARDIA model for the classification of human error and we gave specifications for

1. Physical Subsystems,
2. "Human Subsystems",
3. Real Time Theorems,
4. Landing Procedures and
5. Standard Operating Procedures.

We do not claim that these specifications are complete. The definitions we presented were sufficient for our analysis so far. The method we use may be called *lazy* specification, i.e. we specify all we need for the current purposes. A greedy specification, i.e. specifying everything, would be helpful for further analyses, but this is much more than we are able to provide with this work.

Chapter 4

Formal Completeness and Correctness Proof

Proving a complex system like the WB-Graph developed in chapter 2 based on the specifications of chapter 3 requires the use of a strict proof scheme. Without a systematic approach it is difficult to maintain the overview over the proof. We decided to use a hierarchical proof scheme, allowing the systematic decomposition into substeps, which can be proved separately. This allows the development of proof templates (section 4.2) we will use for the proof of the graph.

4.1 The Hierarchical Proof-scheme

The principle we use to decompose the proof into substeps is *inverse natural deduction*. We try to find proof-steps, which we can assume to be correct for the current step. So we can complete the current step under this assumption and prove the correctness of these proof-steps later on.
To find these steps is straight forward in our case. Since we wish to prove the correctness of a WB-graph, the structure of the proof is directly given by the structure of the graph. All we need to do is to prove the completeness and correctness for the internal nodes of the graph (see figure 4.1).

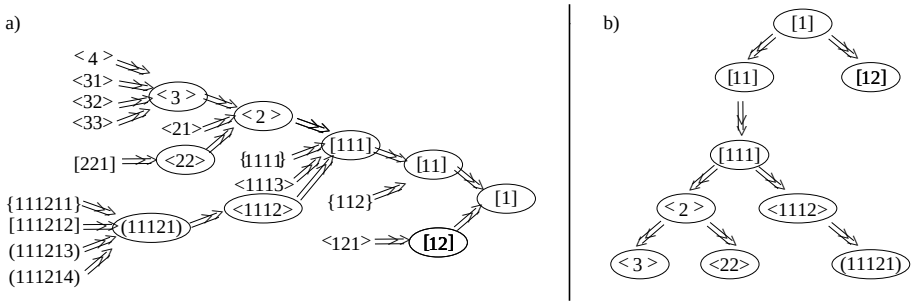

Figure 4.1: (a) Structure of the intuitive WB-Graph and (b) resulting Proof Structure

During the proof, we determined inaccuracies in the intuitive WB-Graph presented in figure 2.6. It was necessary to implement several changes (see chapter 5) and eventually we got the graph shown in figure4.5.

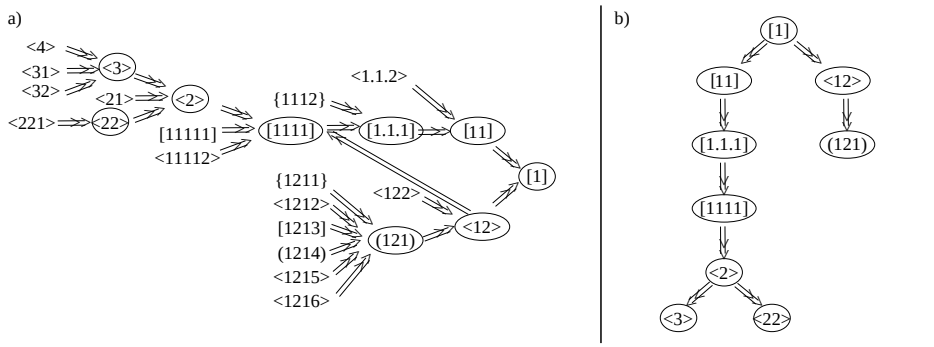

Figure 4.2: (a) Structure of the proved WB-Graph and (b) resulting Proof Structure

Similar to the hierarchical specifications we presented in chapter 3, we split the proof into several modules. Each module contains one of the proof-steps illustrated in figure 4.2 (b). The proof-steps are formulated as theorems. We formulate the main theorem –"*WB-Graph is causally sufficient*"– in a top-level module and include the subproofs as shown in figure 4.3.

module *WB-Graph_is_causally_sufficient*

DECLARATION

$[1]\_suff \triangleq$ **instance** $[1]\_is\_explained\_causally\_sufficient$
$[11]\_suff \triangleq$ **instance** $[11]\_is\_explained\_causally\_sufficient$
$[111]\_suff \triangleq$ **instance** $[111]\_is\_explained\_causally\_sufficient$
$[1111]\_suff \triangleq$ **instance** $[111]\_is\_explained\_causally\_sufficient$
$\langle 12\rangle\_suff \triangleq$ **instance** $\langle 1112\rangle\_is\_explained\_causally\_sufficient$
$(121)\_suff \triangleq$ **instance** $(11121)\_is\_explained\_causally\_sufficient$
$\langle 2\rangle\_suff \triangleq$ **instance** $\langle 2\rangle\_is\_explained\_causally\_sufficient$
$\langle 22\rangle\_suff \triangleq$ **instance** $\langle 22\rangle\_is\_explained\_causally\_sufficient$
$\langle 3\rangle\_suff \triangleq$ **instance** $\langle 3\rangle\_is\_explained\_causally\_sufficient$

DEFINITION

THEOREM $\wedge$ $[1]\_suff$.THEOREM
$\wedge$ $[11]\_suff$.THEOREM
$\wedge$ $[111]\_suff$.THEOREM
$\wedge$ $[1111]\_suff$.THEOREM
$\wedge$ $\langle 12\rangle\_suff$.THEOREM
$\wedge$ $(121)\_suff$.THEOREM
$\wedge$ $\langle 2\rangle\_suff$.THEOREM
$\wedge$ $\langle 22\rangle\_suff$.THEOREM
$\wedge$ $\langle 3\rangle\_suff$.THEOREM

Figure 4.3: Top-Level Module of the Proof

4.2 EL Proof-Step Templates

Proving *causality* is a recursive procedure. Most steps have two substeps which are also assertions of $\Rrightarrow^*$. Eventually, however, the recursion must terminate with *base step* rather than a *recursion step* [LL98b]. we give templates for

1. the high-level proof,
2. deontic proof steps and
3. causal completeness test (CCT) steps[1].

The template for item 3 formulates the CCT-criterion as a Meta-Rule derived from rule C.4.

4.2.1 The High-Level Template

$\langle 1\rangle 1.\ A \hookrightarrow B$
PROOF:
  $\langle 2\rangle 1.\ A \Rrightarrow^* B$
  PROOF:
    $\langle 3\rangle 1.\ A \Rrightarrow^* C$
    PROOF:
      ...
      $\langle 4\rangle 1.\ A \Rrightarrow^* D_0$
      PROOF:
        $\langle 5\rangle 1.\ A \Rrightarrow D_0$
        PROOF:
          $\langle 6\rangle 1.\ A \wedge D_0$
          PROOF:
            $\langle 7\rangle 1.$ A
            PROOF: *********
            $\langle 7\rangle 2.$ D_0
            PROOF: *********
            $\langle 7\rangle 3.$ Q.E.D.
              ($\wedge$-introduction) from $\langle 7\rangle 1$ and $\langle 7\rangle 2$. □
          $\langle 6\rangle 2.\ \neg A \mathrel{\Box\!\!\rightarrow} \neg D_0$
          PROOF:
            ******Lewis semantics and possible worlds reasoning on C and D_0******.
          $\langle 6\rangle 3.$ Q.E.D.
            Directly follows by Inference Rule 2.8 from $\langle 6\rangle 1$ and $\langle 6\rangle 2$. □
        $\langle 5\rangle 2.$ Q.E.D.
          Directly follows by Inference Rule 2 from $\langle 5\rangle 1$. □
      ...
      $\langle 4\rangle 2.\ D_0 \Rrightarrow^* D_1$
      PROOF:
        $\langle 5\rangle 1.\ D_0 \Rrightarrow D_1$
        PROOF: ******Similar to this step in $\langle 4\rangle 1$******
        $\langle 5\rangle 2.$ Q.E.D.
        PROOF:
          Directly follows by Inference Rule 2 from $\langle 5\rangle 1$. □
      ...
      $\langle 4\rangle 3.\ D_n \Rrightarrow^* C$

[1] For more information about causal completeness and sufficient causal explanation, see appendix C.

PROOF: ******Similar to ⟨4⟩1******
...
⟨4⟩4. Q.E.D.
Directly follows by Inference Rule 2 from .....
...
⟨3⟩2. $C \Rightarrow\!\!\!\!\succ^* B$
PROOF: ******Similar to ⟨3⟩1**********
⟨3⟩3. Q.E.D.
Directly follows by Inference Rule 3 from ⟨3⟩1 and ⟨3⟩2. □
⟨2⟩2. Q.E.D.
Directly follows by Inference Rule 2.2 from ⟨2⟩1. □

4.2.2 Deontic Proof-Step Template

⟨1⟩1. $O(\Diamond Event)$
PROOF:
⟨2⟩1. *Hypotheses*
PROOF: **********
⟨2⟩2. $(Hypotheses \wedge \Box Procedures) \succ \Diamond Event$
PROOF:
⟨3⟩1. $\vdash_{TLA} (Hypotheses \wedge \Box Procedures) \Rightarrow \Diamond Event$
PROOF:
FOLLOW: $\vdash_{TLA}$ rules only.
TLA*************TLA
⟨3⟩2. Q.E.D.
Directly follows by Inference Rule 5 from ⟨3⟩1. □
⟨2⟩3. Q.E.D.
Directly follows by Inference Rule 2.4 from ⟨2⟩2. □

4.2.3 CCT as a Derived Meta-Rule

⟨1⟩1. $C \boxdot\!\!\Rightarrow B$
PROOF:
⟨2⟩1. C
PROOF: ******
⟨2⟩2. $\neg C \boxdot\!\!\rightarrow \neg B$
PROOF:
⟨3⟩1. $\neg(A_1 \wedge A_2 \wedge A_3 \wedge ... \wedge A_n) \boxdot\!\!\rightarrow \neg B$
PROOF:
⟨4⟩1. $\neg A_1 \vee \neg(A_2 \wedge A_3 \wedge ... \wedge A_n) \boxdot\!\!\rightarrow \neg B$
PROOF:
⟨5⟩1. $\neg A_1 \boxdot\!\!\rightarrow \neg B$
PROOF:
*****
⟨5⟩2. $\neg(A_2 \wedge A_3 \wedge ... \wedge A_n) \boxdot\!\!\rightarrow B$
PROOF:
⟨6⟩1. $\neg A_2 \vee \neg(A_3 \wedge ... \wedge A_n) \boxdot\!\!\rightarrow \neg B$
PROOF:
⟨7⟩1. $\neg A_2 \boxdot\!\!\rightarrow \neg B$
PROOF: ******
⟨7⟩2. $\neg(A_3 \wedge ... \wedge A_n) \boxdot\!\!\rightarrow \neg B$
...
PROOF:
⟨8⟩1. $\neg(A_{n-1} \wedge A_n) \boxdot\!\!\rightarrow \neg B$
PROOF:
⟨9⟩1. $\neg A_{n-1} \vee \neg A_n \boxdot\!\!\rightarrow \neg B$
PROOF:
⟨10⟩1. $\neg A_{n-1} \boxdot\!\!\rightarrow \neg B$
PROOF: ******
⟨10⟩2. $\neg A_n \boxdot\!\!\rightarrow \neg B$
PROOF: ******
⟨10⟩3. Q.E.D.
Follows by Inference Rule C.5 from ⟨10⟩1 and ⟨10⟩2. □
⟨9⟩2. Q.E.D.
Follows by De Morgan's law applied to the antecedent of ⟨9⟩1. □
⟨8⟩2. Q.E.D.
*****
⟨7⟩3. Q.E.D.
Follows by Inference Rule C.5 from ⟨7⟩1 and ⟨7⟩2. □
⟨6⟩2. Q.E.D.
Follows by De Morgan's law applied to the antecedent of ⟨6⟩1. □
⟨5⟩3. Q.E.D.
Follows by Inference Rule C.5 from ⟨5⟩1 and ⟨5⟩2. □
⟨4⟩2. Q.E.D.
Follows by De Morgan's law applied to the antecedent of ⟨4⟩1. □
⟨3⟩2. Q.E.D.
Follows immediately by the definition of C.
⟨2⟩3. $\neg B \boxdot\!\!\rightarrow \neg C$
PROOF: ******
⟨2⟩4. Q.E.D.
Directly follows by Inference Rule C.4 from ⟨2⟩1, ⟨2⟩2 and ⟨2⟩3. □

4.3 Sufficient Causal Explanation Proof

In the following sections we will present the modules – and their proofs – used to state the top-level theorem in figure 4.3. Furthermore, we will formulate the assumptions we made during the proof in module *Proof-Obligations* (section 4.3.10). In the final section of this chapter we will present the textual and pictorial form of the causally sufficient explained WB-Graph.

4.3.1 Proof of [1]

For a better readability we write down the used WB-graph nodes before we present the proof steps. There is no formal need to do so!

[1] /* AC lands at Brussels RWY 25 */
[11] /* CRW opts to continue landing */
⟨12⟩ /* AC near Brussels Airport */

module *[1]_is_explained_causally_sufficient*

VARIABLES
extends *Landing_Specs*
extends *PARDIA−Axioms*
extends *PARDIA−Norms*
CONSTANTS $CRW, AC, BRU, APPR$

$Hypotheses \triangleq \land [11] \land \langle 12 \rangle$
$Procedures \triangleq \land Landing\_Specs.Spec \land PARDIA{-}Axioms.Spec \land PARDIA{-}Norms.Spec$

THEOREM $(Hypotheses \land Procedures) \Box\!\!\Rightarrow \Diamond[1]$

PROOF:

⟨1⟩1. $[11] \Leftrightarrow Decide(CRW, \Box(AC)in\_landing\_phase)$
PROOF:
We interpret 'opts to continue landing' as 'remain in landing phase', which we formulate as a state pedicate.
⟨1⟩2. $\langle 12 \rangle \Leftrightarrow (AC)near(BRU)$
PROOF:
We interpret ⟨12⟩ as a state predicate.
⟨1⟩3. $(Hypotheses \land Procedures) \Box\!\!\Rightarrow \Diamond[1]$
PROOF:
⟨2⟩1. *Hypotheses*
PROOF:
⟨3⟩1. [11]
PROOF:
[11] under the interpretation of ⟨1⟩1 is true from the sources.
⟨3⟩2. ⟨12⟩
PROOF:
⟨12⟩ under the interpretation of ⟨1⟩2 is true from the sources.
⟨3⟩3. Q.E.D.
Follows by $(\land - intro)$ from ⟨3⟩1, and ⟨3⟩2.
⟨2⟩2. *Procedures*

PROOF:
We assume that *Landing_Specs.Spec*, *PARDIA-Axioms.Spec* and *PARDIA-Norms.Spec* hold.

⟨2⟩3. $(Hypotheses \wedge \Box Procedures) \succ \Diamond[1]$

PROOF:

⟨3⟩1. $\vdash_{TLA} (Hypotheses \wedge \Box Procedures) \Rightarrow \Diamond[1]$

PROOF:

⟨4⟩1. $\left(\begin{array}{ll} \wedge & Decide(CRW, \Box(AC) in\_landing\_phase) \\ \wedge & (AC) near(BRU) \\ \wedge & \Box Landing\_Specs.Spec \\ \wedge & \Box PARDIA-Norms.Spec \\ \wedge & \Box PARDIA-Axioms.Spec \end{array}\right) \Rightarrow (AC)\_lands\_at(BRU)$

PROOF:

⟨5⟩1. $\left(\begin{array}{ll} \wedge & Decide(CRW, \Box(AC)\_in\_landing\_phase) \\ \wedge & \Box PARDIA-Norms.Spec \\ \wedge & \Box PARDIA-Axioms.Spec \end{array}\right) \leadsto \Box(AC)\_in\_landing\_phase$

PROOF:

⟨6⟩1. $\left(\begin{array}{ll} \wedge & Decide(CRW, \Box(AC)\_in\_landing\_phase) \\ \wedge & \Box PARDIA-Norms.Spec \end{array}\right) \leadsto Intend(CRW, \Box(AC)\_in\_landing\_phase)$

PROOF:
This is *PARDIA-Norms.N3*.

⟨6⟩2. $\left(\begin{array}{ll} \wedge & Indend(CRW, \Box(AC)\_in\_landing\_phase) \\ \wedge & \Box PARDIA-Norms.Spec \end{array}\right) \leadsto Act(CRW, \Box(AC)\_in\_landing\_phase)$

PROOF:
This is *PARDIA-Norms.N4*.

⟨6⟩3. $\left(\begin{array}{ll} \wedge & Act(CRW, \Box(AC)\_in\_landing\_phase) \\ \wedge & \Box PARDIA-Axioms.Spec \end{array}\right) \Rightarrow \Box(AC)\_in\_landing\_phase$

PROOF:
Since $\Box(AC)\_in\_landing\_phase$ is a state predicate, this follows from Axiom *PARDIA-Axioms.A10*.

⟨6⟩4. Q.E.D.

PROOF:
Follows by temporal logic from ⟨6⟩1, ⟨6⟩2 and ⟨6⟩3.

⟨5⟩2. $\left(\begin{array}{ll} \wedge & \Box(AC)\_in\_landing\_phase \\ \wedge & (AC) near(BRU) \end{array}\right) \Rightarrow (AC)\_lands\_at(BRU)$

PROOF:
This is an Axiom (*Landing_Specs.LandingRule*)!

⟨5⟩3. Q.E.D.
Follows by propositional logic from ⟨5⟩1 and ⟨5⟩2.

⟨4⟩2. Q.E.D.
Follows by definition of strict implication (inference rule 5). □

⟨2⟩4. $\Diamond[1]$

PROOF:
This is a fact explicitly given in the sources.

⟨2⟩5. Q.E.D.
Directly follows by Inference Rule C.6 from ⟨2⟩1, ⟨2⟩2, ⟨2⟩3 and ⟨2⟩4. □

4.3.2 Proof of [11]

The Proof of [11], although correct according to the formal criteria explained therein, is not intuitively satisfactory because one major hypothesis is not used; therefore the treatment of the incident is not at the level of granularity which we find intuitively most appropriate. Ongoing research will handle [11] more satisfactorily.

Nodes used in the following module:

[11] /* CRW opts to continue landing */
[111] /* CRW realizes they are landing at the wrong airport */
⟨112⟩ /* CRW has safety reasons for continuing landing */
⟨113⟩ /* Standard Operating Procedures */

module *[11]_is_explained_causally_sufficient*

DECLARATIONS
extends *PARDIA−Axioms*
extends *PARDIA−Norms*
extends *Landing_Specs*
CONSTANTS *CRW*, *AC*, *TFC*

$$\begin{aligned} \mathit{Hypotheses} &\triangleq \ \wedge\ [111] \\ &\quad\ \wedge\ \langle 112\rangle \\ \mathit{Procedures} &\triangleq \ \langle 113\rangle \end{aligned}$$

THEOREM $(\mathit{Hypotheses} \wedge \mathit{Procedures}) \ \Box\!\!\Rightarrow \Diamond[11]$

PROOF:

⟨1⟩1. $\langle 112\rangle \Leftrightarrow \mathit{Reason}(\mathit{CRW}, (\Diamond\neg(\mathit{AC})\mathit{in\_landing\_phase} \Rightarrow \mathit{endanger}(\mathit{CRW}, \mathit{TFC})))$
PROOF:
We interpret ⟨112⟩ as a state predicate.

⟨1⟩2. $\langle 113\rangle \Leftrightarrow \wedge\ \mathit{PARDIA\_Norms.Spec} \wedge\ \mathit{PARDIA\_Axioms.Spec} \wedge\ \mathit{Landing\_Specs.Spec}$
PROOF:
We consider the SOPs for this case to be described sufficiently by our specifications *PARDIA_Norms.Spec*, *PARDIA_Axioms.Spec* and *landing_specs.Spec*.

⟨1⟩3. $[11] \Leftrightarrow \begin{pmatrix} \vee & \mathit{Decide}(\mathit{CRW}, \Box(\mathit{AC})\mathit{in\_landing\_phase}) \\ \vee & \mathit{Intend}(\mathit{CRW}, \Box(\mathit{AC})\mathit{in\_landing\_phase}) \\ \vee & \mathit{Act}(\mathit{CRW}, \Box(\mathit{AC})\mathit{in\_landing\_phase}) \end{pmatrix}$
PROOF:
Interpretation: Without further information, "opts to continue" can mean – according to the PARDIA-model – they decided, intended or already acted to stay in landing phase.

⟨1⟩4. $(\mathit{Hypotheses} \wedge \mathit{Procedures}) \ \Box\!\!\Rightarrow \Diamond[11]$
⟨2⟩1. *Hypotheses*
PROOF:
⟨3⟩1. ⟨112⟩
PROOF:
Under the interpretation of ⟨1⟩1 we consider the truth of ⟨112⟩ to stated in the sources.
⟨2⟩2. *Procedures*
PROOF:

From the sources we may assume $\langle 113 \rangle$ under the interpretation of $\langle 1 \rangle 2$ were followed.

$\langle 2 \rangle 3$. $(Hypotheses \wedge \Box Procedures) \succ \Diamond[11]$

PROOF:

We intend to use Inference Rule 5. Beforehand we need to use some formal "human factors reasoning". We use the axioms and definitions of modules *PARDIA-Axioms* and *PARDIA-Norms* to obtain formulae usable in the TLA-steps.

$\langle 3 \rangle 1$. $\vdash_{TLA} (Hypotheses \wedge \Box Procedures) \Rightarrow \Diamond[11]$

PROOF:

$$\langle 4 \rangle 1. \begin{pmatrix} \wedge & Reason(CRW, (\Diamond\neg(AC)in\_landing\_phase \Rightarrow endanger(CRW, TFC))) \\ \wedge & \Box PARDIA\_Norms.Spec \\ \wedge & \Box PARDIA\_Axioms.Spec \\ \wedge & \Box Landing\_Specs.Spec \end{pmatrix} \Rightarrow \Diamond \begin{pmatrix} \vee & Decide(CRW, \Box(AC)in\_landing\_phase) \\ \vee & Intend(CRW, \Box(AC)in\_landing\_phase) \\ \vee & Act(CRW, \Box(AC)in\_landing\_phase) \end{pmatrix}$$

PROOF:

$$\langle 5 \rangle 1. \begin{pmatrix} \wedge & Reason(CRW, (\Diamond\neg(AC)in\_landing\_phase \Rightarrow endanger(CRW, TFC))) \\ \wedge & \Box PARDIA\_Norms.Spec \\ \wedge & \Box PARDIA\_Axioms.Spec \\ \wedge & \Box Landing\_Specs.Spec \end{pmatrix} \Rightarrow \Diamond Decide(CRW, \Box(AC)in\_landing\_phase)$$

PROOF:

$$\langle 6 \rangle 1. \begin{pmatrix} \wedge & Reason(CRW, (\Diamond\neg(AC)in\_landing\_phase \Rightarrow endanger(CRW, TFC))) \\ \wedge & \Box PARDIA\_Axioms.Spec \\ \wedge & \Box Landing\_Specs.Spec \end{pmatrix} \Rightarrow Reason(CRW, \Box(AC)in\_landing\_phase)$$

PROOF:

$$\langle 7 \rangle 1. \begin{pmatrix} \wedge & (\Diamond\neg(AC)in\_landing\_phase \Rightarrow endanger(CRW, TFC)) \\ \wedge & Landing\_Specs.GeneralSafetyRule \end{pmatrix} \Rightarrow \Box(AC)in\_landing\_phase$$

PROOF:

Follows by modus tollens and STL2.

$\langle 7 \rangle 2$. Q.E.D.

Follows by $PARDIA\_Axioms.A16$ with $\langle 7 \rangle 1$. □

$$\langle 6 \rangle 2. \begin{pmatrix} \wedge & Reason(CRW, \Box(AC)in\_landing\_phase) \\ \wedge & PARDIA\_Norms.Spec \end{pmatrix} \Rightarrow \Diamond Decide(CRW, \Box(AC)in\_landing\_phase)$$

PROOF:

$PARDIA\_Norms.N2$:

$Reason(CRW, \Box(AC)in\_landing\_phase) \rightsquigarrow Decide(CRW, \Box(AC)in\_landing\_phase)$

$\langle 6 \rangle 3$. Q.E.D.

Follows by ($\wedge$-intro) from $\langle 6 \rangle 1$ and $\langle 6 \rangle 2$. □

$\langle 5 \rangle 2$. Q.E.D.

Follows immediately by Propositional Logic. □

$\langle 4 \rangle 2$. Q.E.D.

Replacement of Hypotheses and Procedures with interpretations given in $\langle 1 \rangle 1$, $\langle 1 \rangle 2$ und $\langle 1 \rangle 3$. □

$\langle 3 \rangle 2$. Q.E.D.

Follows by Inference Rule 5. □

$\langle 2 \rangle 4$. $\Diamond 11$

This is stated in the sources.

$\langle 2 \rangle 5$. Q.E.D.

Follows immediately from $\langle 2 \rangle 1$, $\langle 2 \rangle 2$, $\langle 2 \rangle 3$ and $\langle 2 \rangle 4$ by EL-Rule C.6. □

4.3.3 Proof of [111]

Nodes used in the following module:

[111] /* Crew (CRW) realizes they are landing at the wrong airport */
[1111] /* CRW gets visual contact to Brussels airport */
{1112} /* CRW notices that Brussels' airport layout is different from Frankfurt's */

module *[111]_is_explained_causally_sufficient*

THEOREM $([1111] \wedge \{1112\}) \Box\!\!\Rightarrow [111]$

PROOF:

⟨1⟩1. $([1111] \wedge \{1112\}) \Box\!\!\Rightarrow [111]$
⟨2⟩1. $[1111] \Rightarrow\!\!\!\!\!- [111]$
PROOF:
⟨3⟩1. $[1111] \wedge [111]$
PROOF:
⟨4⟩1. [1111]
PROOF: This is a fact explicitly given in the sources.
⟨4⟩2. [111]
PROOF: This is a fact explicitly given in the sources.
⟨4⟩3. Q.E.D.
($\wedge$-introduction) from ⟨4⟩1 and ⟨4⟩2. □
⟨3⟩2. $\neg[1111] \Box\!\!\rightarrow \neg[111]$
PROOF:
The CRW did not realize their wrong position earlier although sufficient hints were available. *Given that* instruments providing the information necessary to determine the position without visual contact were ignored, the only other way to obtain this information is visual contact. The theorem follows by *modus tollens*.
⟨3⟩3. Q.E.D.
Directly follows by Inference Rule 2.8 from ⟨3⟩1 and ⟨3⟩2. □
⟨2⟩2. $\{1112\} \Rightarrow\!\!\!\!\!- [111]$
PROOF:
⟨3⟩1. $\{1112\} \wedge [111]$
PROOF:
⟨4⟩1. {1112}
PROOF: This is a fact explicitly given in the sources.
⟨4⟩2. [111]
PROOF: This is a fact explicitly given in the sources.
⟨4⟩3. Q.E.D.
($\wedge$-introduction) from ⟨4⟩1 and ⟨4⟩2. □
⟨3⟩2. $\neg\{1112\} \Box\!\!\rightarrow \neg[111]$
PROOF:
Apart from technical means which were earlier ignored or at least misinterpreted by the CRW the only way to differentiate between airports is to compare their layouts. *Given that* the evidence of the instruments was ignored, if they hadn't recognized the layout they saw was different from the expected (flight documents contain 2D maps of the destination airport) they wouldn't have recognized their mistake at that time.
⟨3⟩3. Q.E.D.
Directly follows by Inference Rule 2.8 from ⟨3⟩1 and ⟨3⟩2. □
⟨2⟩3. $\neg[111] \Box\!\!\rightarrow \neg([1111] \wedge \{1112\})$

Given that no use of evidence from instrumentation was made, if neither [1111] nor {1112} then no evident was available that could call their attention to the wrong airport.

⟨2⟩4. Q.E.D.

Follows by definition of " $\Box\!\!\Rightarrow$ " and from steps ⟨2⟩1,⟨2⟩2 and ⟨2⟩3. □

4.3.4 Proof of [1111]

Nodes used in the following module:

[1111]	/* CRW gets visual contact to Brussels airport */
[11111]	/* AC breaks out under clouds */
⟨11112⟩	/* CRW procedures */
⟨12⟩	/* AC near Brussels Airport */
⟨2⟩	/* AC in BATC area */

module *[1111]_is_explained_causally_sufficient*

VARIABLES

extends *Landing_Specs*
extends *Naturals*
extends *RealTime*
$RTT \triangleq$ **instance** *RealTimeTheorems*
with *lower_bound* $\leftarrow$ *TDZE*,
upper_bound $\leftarrow$ *current_alt*,
$f \leftarrow alt$
CONSTANTS $AC, ATC, CRW, APT, TFC, APPR$
VARIABLES alt

$Hypotheses \triangleq \land [11111] \land \langle 12 \rangle \land \langle 2 \rangle$
$Procedures \triangleq \langle 11112 \rangle$

ASSUMPTIONS

$ASSUMPTION1 \triangleq DH \in (TDZE, alt]$

THEOREM $(Hypotheses \land Procedures) \Box\!\!\Rightarrow \Diamond[1111]$

PROOF:

⟨1⟩1. $[11111] \Leftrightarrow below\_clouds(AC)$
PROOF:
We interpret [11111] as a state predicate.
⟨1⟩2. $\langle 12 \rangle \Leftrightarrow (AC) near (APT)$
PROOF:
We interpret ⟨12⟩ as a state predicate.
⟨1⟩3. $\langle 2 \rangle \Leftrightarrow (AC) in\_area(atc)$
PROOF:
We interpret ⟨2⟩ as a state predicate.
⟨1⟩4. $\langle 11112 \rangle \Leftrightarrow \land ILS\_approach(AC, APT) \land \Box\neg CRW\_breakoff(CRW)$
PROOF:
As described below, a part of the procedures (i.e. $ILS\_approach(AC, APT)$ and $\Box\neg CRW\_breakoff(CRW)$) are sufficient for our needs.
⟨1⟩5. $(Hypotheses \land Procedures) \Box\!\!\Rightarrow \Diamond[1111]$
⟨2⟩1. *Hypotheses*
PROOF:
⟨3⟩1. [11111]
PROOF:

[11111] under the interpretation of ⟨1⟩1 is true from the sources.

⟨3⟩2. ⟨12⟩

PROOF:

⟨12⟩ under the interpretation of ⟨1⟩2 is true from the sources.

⟨3⟩3. ⟨2⟩

PROOF:

⟨2⟩ under the interpretation of ⟨1⟩3 is true from the sources.

⟨3⟩4. Q.E.D.

∧-intro from ⟨3⟩1, ⟨3⟩2 and ⟨3⟩3.

⟨2⟩2. *Procedures*

PROOF:

⟨3⟩1. ⟨11112⟩

PROOF:

⟨11112⟩ under the interpretation of ⟨1⟩4 is true from the sources.

⟨2⟩3. $(Hypotheses \wedge \Box Procedures) \succ \Diamond[1111]$

PROOF:

⟨3⟩1. $\vdash_{TLA} (Hypotheses \wedge \Box Procedures) \Rightarrow \Diamond[1111]$

PROOF:

We intend to reformulate this expression in logic. Formulation of the goal

⟨4⟩1. $[1111] \Leftrightarrow \Diamond visual\_contact(CRW, APT)$

PROOF:

Interpretation: [1111] corresponds to a state predicate.

as well as use of the logical equivalents defined above results in:

⟨4⟩2. $\left(\begin{array}{ll} \wedge & below\_clouds(AC) \\ \wedge & (AC) near (BRU\_APT) \\ \wedge & (AC) in\_area(atc) \\ \wedge & \Box \left(\begin{array}{ll} \wedge & ILS\_approach(AC, APT) \\ \wedge & \Box\neg\, CRW\_breakoff(CRW) \end{array} \right) \end{array} \right) \Rightarrow \Diamond visual\_contact(CRW, APT)$

PROOF:

⟨5⟩1. $\Diamond\Box visual\_contact(CRW, APT) \Rightarrow \Diamond visual\_contact(CRW, APT)$

PROOF:

Derivable from TLA-Rules. □

⟨5⟩2. $\left(\begin{array}{ll} \wedge & \Box ILS\_approach(AC, APT) \\ \wedge & \Box\neg\, CRW\_breakoff(CRW) \\ \wedge & \Diamond\Box \left(\begin{array}{ll} \wedge & ILS\_approach(AC, APT) \\ \wedge & below\_DH \end{array} \right) \end{array} \right) \Rightarrow \Diamond\Box visual\_contact(CRW, APT)$

PROOF:

⟨6⟩1. $\left(\begin{array}{ll} \wedge & \Box ILS\_approach(AC, APT) \\ \wedge & \Box\neg\, CRW\_breakoff(CRW) \end{array} \right) \Rightarrow \Diamond\Box visibility\_acceptable(CRW)$

PROOF:

⟨7⟩1. $\left(\begin{array}{ll} \wedge & \Box ILS\_approach(AC, APT) \\ \wedge & \Box\neg\, CRW\_breakoff(CRW) \\ \wedge & \Diamond below\_DH \end{array} \right) \Rightarrow \Diamond\Box visibility\_acceptable(CRW)$

PROOF:

⟨8⟩1. $\left(\begin{array}{ll} \wedge & \Box ILS\_approach(AC, APT) \\ \wedge & \Box\neg\, CRW\_breakoff(CRW) \\ \wedge & below\_DH \end{array} \right) \Rightarrow \Box visibility\_acceptable(CRW)$

PROOF:

Follows by Propositional Logic from $ILS - APPR\_Rule$.

$\langle 8 \rangle 2.$ $\left(\begin{pmatrix} \wedge & \Box A \\ \wedge & B \end{pmatrix} \Rightarrow C \right) \Rightarrow \left(\begin{pmatrix} \wedge & \Box A \\ \wedge & \Diamond B \end{pmatrix} \Rightarrow \Diamond C \right)$

PROOF:

$\langle 9 \rangle 1.$ $\left(\begin{pmatrix} \wedge & \Box A \\ \wedge & \Diamond B \end{pmatrix} \right) \Rightarrow \Diamond \begin{pmatrix} \wedge & \Box A \\ \wedge & B \end{pmatrix}$

PROOF:

$\langle 10 \rangle 1.$ $\begin{pmatrix} \wedge & \Box A \\ \wedge & \neg \Diamond \begin{pmatrix} \wedge & \Box A \\ \wedge & B \end{pmatrix} \end{pmatrix} \Rightarrow \neg \Diamond B$

PROOF:

$\langle 11 \rangle 1.$ $\begin{pmatrix} \wedge & \Box A \\ \wedge & \Box \begin{pmatrix} \vee & \neg \Box A \\ \vee & \neg B \end{pmatrix} \end{pmatrix} \Rightarrow \Box \neg B$

PROOF:

$\langle 12 \rangle 1.$ $\begin{pmatrix} \wedge & \Box \Box A \\ \wedge & \Box \begin{pmatrix} \vee & \neg \Box A \\ \vee & \neg B \end{pmatrix} \end{pmatrix} \Rightarrow \Box \neg B$

PROOF:

$\langle 13 \rangle 1.$ $\Box \begin{pmatrix} \wedge & \Box A \\ \wedge & \begin{pmatrix} \vee & \neg \Box A \\ \vee & \neg B \end{pmatrix} \end{pmatrix} \Rightarrow \Box \neg B$

PROOF:

$\langle 14 \rangle 1.$ $\begin{pmatrix} \wedge & \Box A \\ \wedge & \begin{pmatrix} \vee & \neg \Box A \\ \vee & \neg B \end{pmatrix} \end{pmatrix} \Rightarrow \neg B$

PROOF:

This is always a logical Truth !

$\langle 14 \rangle 2.$ Q.E.D.

PROOF:

Follows by STL4 from $\langle 14 \rangle 1$. □

$\langle 13 \rangle 2.$ Q.E.D.

Follows by STL5 from $\langle 13 \rangle 1$. □

$\langle 12 \rangle 2.$ Q.E.D.

Follows by STL3 from $\langle 12 \rangle 1$. □

$\langle 11 \rangle 2.$ Q.E.D.

Follows by Propositional Logic from $\langle 11 \rangle 1$. □

$\langle 10 \rangle 2.$ Q.E.D.

PROOF:

Follows by Propositional Logic from $\langle 10 \rangle 1$. □

$\langle 9 \rangle 2.$ $\Diamond \begin{pmatrix} \wedge & \Box A \\ \wedge & B \end{pmatrix} \Rightarrow \Diamond C$

PROOF:

$\langle 10 \rangle 1.$ $\begin{pmatrix} \wedge & \Box A \\ \wedge & B \end{pmatrix} \Rightarrow C$

PROOF:

With:

$A \triangleq \wedge\ ILS\_approach(AC, APT)$
$\wedge\ \neg\, CRW\_breakoff(CRW)$

$B \triangleq below\_DH$

$C \triangleq \Box visibility\_acceptable(CRW)$ and Propositional Logic this is $ILS - APPR\_rule$.

$\langle 10 \rangle 2.$ Q.E.D.

PROOF:

Follows from $\langle 10 \rangle 1$ by STL4 ($\Diamond$-Form). □

⟨9⟩3. Q.E.D.
PROOF:
Follows by Propositional Logic from ⟨9⟩1 and ⟨9⟩2. □
⟨8⟩3. Q.E.D.
Derivable from ⟨8⟩1 by ⟨8⟩2 with:
$A \triangleq \wedge\ ILS\_approach(AC, APT)$
$\wedge\ \neg CRW\_breakoff(CRW)$
$B \triangleq below\_DH$
$C \triangleq \Box visibility\_acceptable(CRW)$. □

⟨7⟩2. $\left(\begin{array}{ll} \wedge & \Box ILS\_approach(AC, APT) \\ \wedge & \Box\neg CRW\_breakoff(CRW) \end{array}\right) \Rightarrow \Diamond below\_DH$
PROOF:
⟨8⟩1. $\left(\begin{array}{ll} \wedge & \Box ILS\_approach(AC, APT) \\ \wedge & \Box\neg CRW\_breakoff(CRW) \end{array}\right)$
$\Rightarrow \forall\, x \in [TDZE, current\_alt] : \Diamond alt = x$
PROOF:
⟨9⟩1. $\left(\begin{array}{ll} \wedge & \Box ILS\_approach(AC, APT) \\ \wedge & \Box\neg CRW\_breakoff(CRW) \end{array}\right) \Rightarrow \Diamond landing$
PROOF:
This is an Axiom $(ILS - LandingRule)$!
⟨9⟩2. $\left(\begin{array}{ll} \wedge & \Box ILS\_approach(AC, APT) \\ \wedge & \Box\neg CRW\_breakoff(CRW) \end{array}\right) \Rightarrow \left(\begin{array}{l} alt\ is\ a\ 'monotone\ decreasing \\ continuous\ function\ of\ RealTime' \end{array}\right)$
PROOF:
This is an Axiom $(ILS - AltProperty)$!
⟨9⟩3. $\left(\begin{array}{l} alt\ is\ a\ 'monotone\ decreasing \\ continuous\ function\ of\ RealTime' \end{array}\right) \Rightarrow \frac{\delta\ alt}{\delta t} < 0\ in\ [TDZE, current\_alt]$
PROOF:
This is the definition of a 'decreasing continuous function'.
⟨9⟩4. $\left(\begin{array}{ll} \wedge & \frac{\delta\ alt}{\delta t} < 0\ in\ [TDZE, current\_alt] \\ \wedge & alt = current\_alt \\ \wedge & \Diamond(alt = TZDE) \end{array}\right)$
$\Rightarrow \forall\, x \in [TDZE, current\_alt] : \Diamond alt = x$
PROOF:
This is RTT.MeanValueTheorem.
⟨9⟩5. Q.E.D.
PROOF:
Follows by Propositional Logic from ⟨9⟩1, ⟨9⟩2, ⟨9⟩3
and ⟨9⟩4. □
⟨8⟩2. $\left(\begin{array}{ll} \wedge & \forall\, x \in [TDZE, current\_alt] : \Diamond alt = x \\ \wedge & ASSUMPTION1 \end{array}\right) \Rightarrow \Diamond below\_DH$
PROOF:
Follows immediatly from Calculus and definition of $below\_DH$.
⟨8⟩3. $\Diamond\Box below\_DH \Rightarrow \Diamond below\_DH$
PROOF:
Derivable by TLA-Rules.
⟨8⟩4. Q.E.D.
PROOF:
Follows from ⟨8⟩1, ⟨8⟩2 and ⟨8⟩3 by Propositional Logic. □
⟨7⟩3. Q.E.D.
PROOF:
Follows from ⟨7⟩1 and ⟨7⟩2 by Propositional Logic. □

$\langle 6\rangle 2.\ \left(\begin{array}{ll} \wedge & \Diamond\Box \mathit{visibility\_acceptable}(CRW) \\ \wedge & \Diamond\Box \left(\begin{array}{ll} \wedge & \mathit{ILS\_approach}(AC, APT) \\ \wedge & \mathit{below\_DH} \end{array}\right) \end{array}\right) \Rightarrow \Diamond\Box \mathit{visual\_contact}(CRW, APT)$

PROOF:

$\langle 7\rangle 1.\ \Diamond\Box \left(\begin{array}{ll} \wedge & \mathit{visibility\_acceptable}(CRW) \\ \wedge & \left(\begin{array}{ll} \wedge & \mathit{ILS\_approach}(AC, APT) \\ \wedge & \mathit{below\_DH} \end{array}\right) \end{array}\right) \Rightarrow \Diamond\Box \mathit{visual\_contact}(CRW, APT)$

PROOF:

$\langle 8\rangle 1.\ \Box \left(\begin{array}{ll} \wedge & \mathit{visibility\_acceptable}(CRW) \\ \wedge & \left(\begin{array}{ll} \wedge & \mathit{ILS\_approach}(AC, APT) \\ \wedge & \mathit{below\_DH} \end{array}\right) \end{array}\right) \Rightarrow \Box \mathit{visual\_contact}(CRW, APT)$

PROOF:

$\langle 9\rangle 1.\ \left(\begin{array}{ll} \wedge & \mathit{visibility\_acceptable}(CRW) \\ \wedge & \left(\begin{array}{ll} \wedge & \mathit{ILS\_approach}(AC, APT) \\ \wedge & \mathit{below\_DH} \end{array}\right) \end{array}\right) \Rightarrow \mathit{visual\_contact}(CRW, APT)$

PROOF:

$\langle 10\rangle 1.\ \left(\begin{array}{ll} \wedge & \mathit{visibility\_acceptable}(CRW) \\ \wedge & \vee \wedge \mathit{ILS\_approach}(AC, APT) \\ & \quad \wedge \mathit{below\_DH} \\ & \vee \wedge \mathit{NPA\_approach}(AC, APT) \\ & \quad \wedge \mathit{at\_MDA} \end{array}\right) \Rightarrow \mathit{visual\_contact}(CRW, APT)$

PROOF:

This is an Axiom (*Landing_Criterion*).

$\langle 10\rangle 2.\ \mathit{ILS\_approach}(AC, APT) \Leftrightarrow \neg \mathit{NPA\_approach}(AC, APT))$

PROOF:

This is an Axiom ($\mathit{Unique\_Approach} - \mathit{Type\_Rule}$).

$\langle 10\rangle 3.$ Q.E.D.

Follows from $\langle 10\rangle 1$ and $\langle 10\rangle 2$ by Propositional Logic. □

$\langle 9\rangle 2.$ Q.E.D.

Follows immediately from STL4. □

$\langle 8\rangle 2.$ Q.E.D.

Follows immediately from STL4 ($\Diamond$-Form). □

$\langle 7\rangle 2.$ Q.E.D.

Follows immediately from STL6. □

$\langle 6\rangle 3.$ Q.E.D.

Follows by Propositional Logic from $\langle 6\rangle 1$ and $\langle 6\rangle 2$. □

$\langle 5\rangle 3.$ Q.E.D.

Follows by Propositional Logic (transitivity of implication) from $\langle 5\rangle 1$ and $\langle 5\rangle 2$. □

$\langle 4\rangle 3.$ Q.E.D.

Follows from steps $\langle 4\rangle 1$ and $\langle 4\rangle 2$. □

$\langle 3\rangle 2.$ Q.E.D.

Follows by definition of strict implication (inference rule 5). □

$\langle 2\rangle 4.\ \Diamond[1111]$

PROOF:

This is a fact explicitly given in the sources.

$\langle 2\rangle 5.\ \neg\Diamond[1111] \Box\!\!\rightarrow \neg(\mathit{Hypotheses} \wedge \Box \mathit{Procedures})$

PROOF:

$\langle 3\rangle 1.\ \Box\neg visual\_contact(CRW, APT)\Box\rightarrow \left(\begin{array}{ll} \vee & \neg below\_clouds(AC) \\ \vee & \neg(AC)near(APT) \\ \vee & \neg(AC)in\_area(atc) \\ \vee & \neg\Box\left(\begin{array}{ll} \wedge & ILS\_approach(AC, APT) \\ \wedge & \Box\neg CRW\_breakoff(CRW) \end{array}\right) \end{array}\right)$

PROOF:

We argue that in the nearest possible world,

$$\neg\Box\left(\begin{array}{ll} \wedge & ILS\_approach(AC, APT) \\ \wedge & \Box\neg CRW\_breakoff(CRW) \end{array}\right)$$

is the least possible alternative. Under this assumption, $\neg below\_clouds(AC)$, $\neg(AC)near(APT)$ or $\neg(AC)in\_area(atc)$ would all lead to $\neg visual\_contact(CRW, APT)$.

$\langle 3\rangle 2.$ Q.E.D.

Follows by Propositional Logic, STL2 and STL3 as well as use of definitions from $\langle 2\rangle 1$ and $\langle 2\rangle 2$.

$\langle 2\rangle 6.$ Q.E.D.

Directly follows by Inference Rule C.6 from $\langle 2\rangle 1$, $\langle 2\rangle 2$, $\langle 2\rangle 3$, $\langle 2\rangle 4$ and $\langle 2\rangle 5$. □

4.3.5 Proof of Node 12

Nodes used in the following module:

⟨12⟩ /* AC near Brussels Airport */
(121) /* CRW did not realize that they were on wrong course, UNTIL:[111] */
⟨122⟩ /* AC cleared to BATC according to ATC procedures */

module ⟨*12*⟩*_is_explained_causally_sufficient*

THEOREM $((121) \wedge \langle 122 \rangle) \mathrel{\Box\!\!\Rightarrow} \langle 12 \rangle$

PROOF:

⟨1⟩1. $((121) \wedge \langle 122 \rangle) \mathrel{\Box\!\!\Rightarrow} \langle 12 \rangle$
⟨2⟩1. $(121) \mathrel{\gg} \langle 12 \rangle$
PROOF:
⟨3⟩1. $(121) \wedge \langle 12 \rangle$
PROOF:
⟨4⟩1. (121)
PROOF: This is assumed in the sources and can be derived from the CRW's behavior.
⟨4⟩2. ⟨12⟩
PROOF: This is a fact explicitly given in the sources.
⟨4⟩3. Q.E.D.
(∧-introduction) from ⟨4⟩1 and ⟨4⟩2. □
⟨3⟩2. $\neg(121) \mathrel{\Box\!\!\rightarrow} \neg\langle 12 \rangle$
PROOF:
The flight destination was Frankfurt. If they had realized they were on the wrong course, they would likely have taken actions to return to the course they were supposed to be on unless other more pressing considerations intervened. This is exactly what the counterfactual says.
⟨3⟩3. Q.E.D.
Directly follows by Inference Rule 2.8 from ⟨3⟩1 and ⟨3⟩2. □
⟨2⟩2. $\langle 122 \rangle \mathrel{\gg} \langle 12 \rangle$
PROOF:
⟨3⟩1. $\langle 122 \rangle \wedge \langle 12 \rangle$
PROOF:
⟨4⟩1. ⟨122⟩
PROOF: We assume this from facts given in the sources.
⟨4⟩2. ⟨12⟩
PROOF: This is a fact explicitly given in the sources.
⟨4⟩3. Q.E.D.
(∧-introduction) from ⟨4⟩1 and ⟨4⟩2. □
⟨3⟩2. $\neg\langle 122 \rangle \mathrel{\Box\!\!\rightarrow} \neg\langle 12 \rangle$
PROOF:
We consider, that in the nearest possible world the SOPs are followed. In this particular case we assume that an AC only enters a specific area when it has obtained an ATC clearance before. Therefore without a clearance to BATC area the AC will not be able to get into Brussels Airport area.
⟨3⟩3. Q.E.D.
Directly follows by Inference Rule 2.8 from ⟨3⟩1 and ⟨3⟩2. □
⟨2⟩3. $\neg\langle 12 \rangle \mathrel{\Box\!\!\rightarrow} \neg((121) \wedge \langle 122 \rangle)$
In the nearest possible world they had not been near Brussels airport, since their destination was Frankfurt. Because their flightpath in that case had led

from LATC area to MATC area they weren't supposed to be cleared to BATC. Since they are assumed to follow the SOPs in the nearest possible world, they would have realized hints about a possible wrong course.

⟨2⟩4. Q.E.D.

Follows by definition of " $\Box\!\!\Rightarrow$ " and from steps ⟨2⟩1 and ⟨2⟩3. □

4.3.6 Proof of (121)

Nodes used in the following module:

(121) /* CRW did not realize that they were on wrong course, UNTIL:[111] */
{1211} /* CRW addresses BATC controller as "Frankfurt" several times */
⟨1212⟩ /* ILS has different frequency for Frankfurt. */
[1213] /* CRW asks for the Bruno VOR's frequency. */
(1214) /* Brussels did not question the addressing error although it happened more than once */
⟨1215⟩ /* Situation remains safe during landing */
⟨1216⟩ /* Current approach plates are used */

module *(121)_is_explained_causally_sufficient*

VARIABLES
extends *PARDIA−Axioms*
extends *PARDIA−Norms*
extends *Landing_Specs*
CONSTANTS $CRW, AC, ATC, BATC, BRU, FRA, this\_approach, freq$

$$
\begin{array}{lcl}
Hypotheses & \triangleq & \wedge \{1211\} \\
 & & \wedge \langle 1212 \rangle \\
 & & \wedge [1213] \\
 & & \wedge (1214) \\
 & & \wedge \langle 1215 \rangle \\
 & & \wedge \langle 1216 \rangle \\
Procedures & \triangleq & \wedge PARDIA-Axioms.Spec \\
 & & \wedge Landing\_Specs.Spec
\end{array}
$$

THEOREM $(Hypotheses \wedge Procedures) \ \Box\!\!\Rightarrow \Diamond(121)$

PROOF:

⟨1⟩1. $\{1211\} \Leftrightarrow Perceive(BATC, addressing\_error(CRW, BATC))$
PROOF:
We claim, that the addressing error has not only been made but must also have been perceived, since this information wouldn't be in the sources otherwise.

⟨1⟩2. $\langle 1212 \rangle \Leftrightarrow$
$Attend(CRW, data\_inconsist(appr\_plate[this\_approach][LOC[freq]], appr\_plate[FRA][LOC[freq]]))$
PROOF:
Since mentioned in the sources, the CRW must at least have paid attention to the fact of inconsistent data.

⟨1⟩3. $[1213] \Leftrightarrow Attend(CRW, BrunoVor[freq] \notin Navaids[appr\_plate[this\_approach]])$
PROOF:
From the fact, they asked for the Bruno VOR frequency, we can derive, that they at least attended that this information was not presented in the approach plate they used.

⟨1⟩4. $(1214) \Leftrightarrow \neg question(BATC, CRW, addressing\_error(CRW, BATC))$
PROOF:
We interpret (1214) as a state predicate.

⟨1⟩5. $\langle 1215 \rangle \Leftrightarrow \wedge \Diamond(AC) lands\_at(BRU)$
$\wedge \Box\neg distress\_decl(CRW)$
$\wedge \Box\neg urgency\_decl(CRW)$
$\wedge \Box\neg endanger\_decl(CRW, TFC)$

PROOF:
Since we do not find contrary information in the sources, we may adopt ⟨1215⟩ as hypothesis.

⟨1⟩6. $\langle 1216\rangle \Leftrightarrow \wedge\ current(appr\_plate[this\_approach])$
$\wedge\ current(appr\_plate[FRA])$

PROOF:
Since we do not find contrary information in the sources, we may adopt ⟨1216⟩ as hypothesis.

⟨1⟩7. $(121) \Leftrightarrow \wedge\ O(Attend(CRW, APT \neq destAPT))$
$\wedge\ \neg Attend(CRW, APT \neq destAPT)$

PROOF:
(121) is a non-event and therefore we cannot simply express it as a state. According to the principle of contrastive explanation we need to state that some action awaited to be done was not executed.

⟨1⟩8. $(Hypotheses \wedge Procedures)\ \Box\!\!\Rightarrow \Diamond(121)$

⟨2⟩1. *Hypotheses*

PROOF:

⟨3⟩1. {1211}

PROOF:
{1211} under the interpretation of ⟨1⟩1 is true from the sources.

⟨3⟩2. ⟨1212⟩

PROOF:
⟨1212⟩ under the interpretation of ⟨1⟩2 is true from the sources.

⟨3⟩3. [1213]

PROOF:
[1213] under the interpretation of ⟨1⟩3 is true from the sources.

⟨3⟩4. (1214)

PROOF:
(1214) under the interpretation of ⟨1⟩4 is true from the sources.

⟨3⟩5. ⟨1215⟩

PROOF:
⟨1215⟩ under the interpretation of ⟨1⟩5 is true from the sources.

⟨3⟩6. ⟨1216⟩

PROOF:
⟨1216⟩ under the interpretation of ⟨1⟩6 is true from the sources.

⟨3⟩7. Q.E.D.
Follows by $(\wedge - intro)$ from ⟨3⟩1, ⟨3⟩2, ⟨3⟩3, ⟨3⟩4, ⟨3⟩5 and ⟨3⟩6. □

⟨2⟩2. *Procedures*

PROOF:

⟨3⟩1. $\wedge\ PARDIA{-}Axioms.Spec$
$\wedge\ Landing\_Specs.Spec$

PROOF:
We assume that the procedures are implemented as specified in *PARDIA-Axioms.Spec* and *Landing_Specs.Spec*.

⟨2⟩3. $(Hypotheses \wedge \Box Procedures) \succ \Diamond(121)$

PROOF:

⟨3⟩1. $\vdash_{TLA} (Hypotheses \wedge \Box Procedures) \Rightarrow \Diamond(121)$

PROOF:

⟨4⟩1. $(121) \Rightarrow \Diamond(121)$

PROOF:
Follows by ◇-intro.

$$\langle 4\rangle 2. \left(\begin{array}{ll} \wedge & Perceive(BATC, addressing\_error(CRW, BATC)) \\ \wedge & Attend(CRW, data\_inconsist(appr\_plate[this\_approach][LOC[freq]]), \\ & \qquad\qquad appr\_plate[FRA][LOC[freq]])) \\ \wedge & Attend(CRW, BrunoVor[freq] \notin Navaids[appr\_plate[this\_approach]]) \\ \wedge & \neg question(BATC, CRW, addressing\_error(CRW, BATC)) \\ \wedge & current(appr\_plate[this\_approach]) \\ \wedge & current(appr\_plate[FRA]) \\ \wedge & \Diamond(AC) lands\_at(BRU) \\ \wedge & \Box\neg distress\_decl(CRW) \\ \wedge & \Box\neg urgency\_decl(CRW) \\ \wedge & \Box\neg endanger\_decl(CRW, TFC) \\ \wedge & \Box PARDIA{-}Axioms.Spec \\ \wedge & \Box Landing\_Specs.Spec \end{array}\right)$$

$$\Rightarrow \left(\begin{array}{ll} \wedge & O(Attend(CRW, APT \neq destAPT)) \\ \wedge & \neg Attend(CRW, APT \neq destAPT) \end{array}\right)$$

PROOF:

$$\langle 5\rangle 1. \left(\begin{array}{ll} \wedge & \Diamond(AC) lands\_at(BRU) \\ \wedge & current(appr\_plate[this\_approach]) \\ \wedge & current(appr\_plate[FRA]) \\ \wedge & \Box\neg distress\_decl(CRW) \\ \wedge & \Box\neg urgency\_decl(CRW) \\ \wedge & \Box\neg endanger\_decl(CRW, TFC) \\ \wedge & \Box PARDIA{-}Axioms.Spec \\ \wedge & \Box Landing\_Specs.Spec \end{array}\right)$$

$$\Rightarrow \neg Attend(CRW, APT \neq destAPT)$$

PROOF:

$$\langle 6\rangle 1. \left(\begin{array}{ll} \wedge & \Box Attend(CRW, APT \neq destAPT) \\ \wedge & \Box\neg distress\_decl(CRW) \\ \wedge & \Box\neg urgency\_decl(CRW) \\ \wedge & \Box\neg endanger\_decl(CRW, TFC) \\ \wedge & \Box PARDIA{-}Axioms.Spec \\ \wedge & \Box Landing\_Specs.Spec \end{array}\right) \Box\!\!\rightarrow \neg\Diamond(AC) lands\_at(APT)$$

PROOF:

$$\langle 7\rangle 1. \left(\begin{array}{ll} \wedge & \Box Attend(CRW, APT \neq destAPT) \\ \wedge & \Box PARDIA{-}Axioms.Spec \end{array}\right)$$

$$\Rightarrow APT \neq destAPT$$

PROOF:

This is an axiom (*PARDIA-Axioms.A12*)!

$$\langle 7\rangle 2. \left(\begin{array}{ll} \wedge & APT \neq destAPT \\ \wedge & \Box\neg distress\_decl(CRW) \\ \wedge & \Box\neg urgency\_decl(CRW) \\ \wedge & \Box\neg endanger\_decl(CRW, TFC) \\ \wedge & \Box Landing\_Specs.Spec \end{array}\right) \Rightarrow \neg\Diamond(AC) lands\_at(APT)$$

PROOF:

This is a rule (*Landing_Norms.Normal-Progress*).

⟨7⟩3. Q.E.D.

PARDIA-Axioms and *NormalProgress* hold in the nearest possible world, in which A holds. ⟨6⟩1 therefore follows by (∧-intro) from ⟨7⟩1 and ⟨7⟩2. □

⟨6⟩2. Q.E.D.

With: $A \triangleq Attend(CRW, APT \neq destAPT)$

$B \triangleq \Diamond(AC) lands\_at(APT)$

we can also derive $\neg B$ by a derived inference rule: $\dfrac{A \quad A \,\Box\!\!\rightarrow \neg B}{\neg B}$

But we know that B occurred.

Therefore by modus tollens we can derive $\neg A$. □

$$\langle 5\rangle 2.\ \left(\begin{array}{ll} \wedge & Attend(CRW, data\_inconsist(appr\_plate[this\_approach][LOC[freq]]), \\ & \qquad appr\_plate[FRA][LOC[freq]])) \\ \wedge & current(appr\_plate[this\_approach]) \\ \wedge & current(appr\_plate[FRA]) \\ \wedge & \Box PARDIA{-}Axioms.Spec \\ \wedge & \Box Landing\_Specs.Spec \end{array}\right) \Rightarrow O(Attend(CRW, apt \neq destAPT))$$

PROOF:

⟨6⟩1. $O(Procs)$

PROOF:

This is an EL-Axiom (cf. Axiom 7).

$$\langle 6\rangle 2.\ \left(\begin{array}{ll} \wedge & Attend(CRW, data\_inconsist(appr\_plate[this\_approach][LOC[freq]]), \\ & \qquad appr\_plate[FRA][LOC[freq]])) \\ \wedge & current(appr\_plate[this\_approach]) \\ \wedge & current(appr\_plate[FRA]) \\ \wedge & \Box PARDIA{-}Axioms.Spec \\ \wedge & \Box Landing\_Specs.Spec \end{array}\right) \Rightarrow Attend(CRW, APT \neq destAPT)$$

PROOF:

$$\langle 7\rangle 1.\ \left(\begin{array}{ll} \wedge & Attend(CRW, data\_inconsist(appr\_plate[this\_approach][LOC[freq]], \\ & \qquad appr\_plate[FRA][LOC[freq]])) \\ \wedge & \Box PARDIA{-}Axioms.Spec \end{array}\right) \Rightarrow data\_inconsist(appr\_plate[this\_approach][LOC[freq]], appr\_plate[FRA][LOC[freq]]))$$

PROOF:

This is an Axiom (*PARDIA-Axioms.A12*).

$$\langle 7\rangle 2.\ \left(\begin{array}{ll} \wedge & data\_inconsist(appr\_plate[this\_approach][LOC[freq]], \\ & \quad appr\_plate[FRA][LOC[freq]]) \\ \wedge & current(appr\_plate[this\_approach]) \\ \wedge & current(appr\_plate[FRA]) \\ \wedge & \Box Landing\_Specs.Spec \end{array}\right) \Rightarrow (APT \neq destAPT)$$

PROOF:

This is a rule (*Landing_Norms.UniqueApproachPlates*).

⟨7⟩3. Q.E.D.

Follows by (∧-intro) from ⟨7⟩1 and ⟨7⟩2. □

⟨6⟩3. Q.E.D.

Follows from EL (SDL Rule K_O): $O(A \Rightarrow B) \Rightarrow (O(A) \Rightarrow O(B))$ □

⟨5⟩3. Q.E.D.

Follows by (∧-intro) from ⟨5⟩1 and ⟨5⟩2. □

⟨4⟩3. Q.E.D.

Insertion of interpretations from ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩6, ⟨2⟩2 and ⟨1⟩7 into ⟨3⟩1. □

⟨3⟩2. Q.E.D.

Follows by definition of strict implication (Inference Rule 5). □

⟨2⟩4. ◇(121)
PROOF:
This is a fact explicitly given in the sources.
⟨2⟩5. Q.E.D.
Directly follows by Inference Rule C.6 from ⟨2⟩1, ⟨2⟩2, ⟨2⟩3 and ⟨2⟩4. □

4.3.7 Proof of Node 2

Nodes used in the following module:

⟨2⟩ /* AC in BATC area */
⟨3⟩ /* AC in LATC area */
⟨21⟩ /* LATC procedures */
⟨22⟩ /* LATC uses false flight data for NW052 */

module *⟨2⟩_is_explained_causally_sufficient*

VARIABLES
extends *SOP_Specs*
CONSTANTS $AC, BATC, LATC$

$$\begin{aligned} \mathit{Hypotheses} &\triangleq \land \langle 22\rangle \\ &\quad\ \land \langle 3\rangle \\ \mathit{Procedures} &\triangleq \langle 21\rangle \end{aligned}$$

THEOREM $(\mathit{Hypotheses} \land \mathit{Procedures}) \Box\Rightarrow \Diamond\langle 2\rangle$

PROOF:

$$\langle 1\rangle 1.\ \langle 22\rangle \Leftrightarrow \left(\begin{array}{ll} \land & \mathit{fdata}(ac)[\mathit{destAPT}] = \text{`}BRU\text{`} \\ \land & \mathit{nextATC}(AC) = BATC \\ \land & \mathit{responsibleATC}(\text{`}BRU\text{`}) = BATC \\ \land & \mathit{destAPT}(AC) = \text{`}FRA\text{`} \end{array} \right)$$

PROOF:
We interpret "false flightdata" as deviation of the used data - determined (det) from the ATCs FDC - from the original data. All we use from the flighdata is the information on the AC's destination airport. This piece of information usually suffices to calculate a flight route containing the start and landing ATCCs as well as all intermediate ATCCs. We do not specify this process at this point and define this information as a part of the conjunct describing ⟨22⟩ instead.

$\langle 1\rangle 2.\ \langle 3\rangle \Leftrightarrow (AC)\mathit{in\_area}(LATC)$
PROOF:
We interpret ⟨3⟩ as a state predicate.

$\langle 1\rangle 3.\ \langle 21\rangle \Rightarrow \mathit{SOP\_Specs.ATC{-}Responsibility{-}Rule}$
PROOF:
In the sources the *Procedures* are assumed to be implemented correctly. We claim $\mathit{SOP\_Specs.ATC{-}Responsibility{-}Rule}$ is sufficient to explain the ATC procedures significant for this case.

$\langle 1\rangle 4.\ (\mathit{Hypotheses} \land \mathit{Procedures}) \Box\Rightarrow \Diamond\langle 2\rangle$
⟨2⟩1. *Hypotheses*
PROOF:
⟨3⟩1. ⟨22⟩
PROOF:
⟨22⟩ under the interpretation of ⟨1⟩1 is true from the sources.
⟨3⟩2. ⟨3⟩
PROOF:
⟨3⟩ under the interpretation of ⟨1⟩2 is true from the sources.

⟨3⟩3. Q.E.D.
∧-intro from ⟨3⟩1 and ⟨3⟩2. □

⟨2⟩2. *Procedures*
PROOF:
⟨3⟩1. ⟨21⟩
PROOF:
⟨21⟩ under the interpretation of ⟨1⟩3 is true from the sources.
⟨3⟩2. Q.E.D.
Follows immediately from ⟨3⟩1. □

⟨2⟩3. $(Hypotheses \wedge \Box Procedures) \succ \Diamond\langle 2\rangle$
PROOF:
⟨3⟩1. $\vdash_{TLA}(Hypotheses \wedge \Box Procedures) \Rightarrow \Diamond\langle 2\rangle$
PROOF:
We argue in the forward sense of natural deduction.
We reformulate this expression in logic. Formulation of the goal as well as use of the logical equivalents defined above results in:

⟨4⟩1. $\left(\begin{array}{ll} \wedge & fdata(ac)[destAPT] = \text{`}BRU\text{`} \\ \wedge & nextATC(AC) = BATC \\ \wedge & responsibleATC(\text{`}BRU\text{`}) = BATC \\ \wedge & destAPT(AC) = \text{`}FRA\text{`} \\ \wedge & (AC)in\_area(LATC) \\ \wedge & \Box ATC{-}Responsibility{-}Rule \end{array}\right) \Rightarrow \Diamond(AC)in\_area(BATC)$

PROOF:

⟨5⟩1. $\left(\begin{array}{ll} \wedge & fdata(ac)[destAPT] = \text{`}BRU\text{`} \\ \wedge & nextATC(AC) = BATC \\ \wedge & responsibleATC(\text{`}BRU\text{`}) = BATC \\ \wedge & destAPT(AC) = \text{`}FRA\text{`} \\ \wedge & (AC)in\_area(LATC) \\ \wedge & \Box SOP\_Specs.ATC{-}Responsibility{-}Rule \end{array}\right) \Rightarrow \Diamond Handoff(LATC, BATC, AC)$

PROOF:

⟨6⟩1. $\left(\begin{array}{ll} \wedge & (AC)in\_area(LATC) \\ \wedge & \Box ATC{-}Responsibility{-}Rule \end{array}\right) \Rightarrow EnRouteProcessing(LATC, AC)$

PROOF:
Follows by STL2 and Propositional Logic from
SOP_Specs.ATC-Responsibility-Rule.

⟨6⟩2. $\left(\begin{array}{ll} \wedge & EnRouteProcessing(LATC, AC) \\ \wedge & (AC)in\_area(LATC) \\ \wedge & fdata(ac)[destAPT] = \text{`}BRU\text{`} \\ \wedge & nextATC(AC) = BATC \\ \wedge & responsibleATC(\text{`}BRU\text{`}) = BATC \\ \wedge & destAPT(AC) = \text{`}FRA\text{`} \end{array}\right) \Rightarrow \Diamond Handoff(LATC, BATC, AC)$

PROOF:

⟨7⟩1. $\left(\begin{array}{ll} \wedge & det\_destAPT(AC) = \text{`}BRU\text{`} \\ \wedge & fdata(ac)[destAPT] = \text{`}BRU\text{`} \end{array}\right) \Rightarrow det\_destAPT(AC) = fdata(AC)[destAPT]$

PROOF:
Follows by SOP_Specs.DetermineDestinationProcedure.

⟨7⟩2. $\left(\begin{array}{ll} \wedge & det\_destAPT(AC) = \text{`}BRU\text{`} \\ \wedge & fdata(ac)[destAPT] = \text{`}BRU\text{`} \end{array}\right) \Rightarrow LATC \neq det\_destATC(AC)$

PROOF:

$\langle 8\rangle 1.$ $\left(\begin{array}{ll} \wedge & det\_destAPT(AC) = \text{`}BRU\text{`} \\ \wedge & responsibleATC(\text{`}BRU\text{`}) = BATC \end{array}\right) \Rightarrow (det\_destATC(AC) = BATC)$

PROOF:

Follows immediately from $SOP\_SPECS.RespDestATC\_Determination\_Rule$.

$\langle 8\rangle 2.$ $BATC \neq LATC$

PROOF:

We now from the sources that this is true.

$\langle 8\rangle 3.$ Q.E.D.

Follows by Propositional Logic from $\langle 8\rangle 1$ and $\langle 8\rangle 2$.

$\langle 7\rangle 3.$ $\left(\begin{array}{ll} \wedge & EnRouteProcessing(LATC, AC) \\ \wedge & (AC)in\_area(LATC) \\ \wedge & det\_destAPT(AC) = fdata(AC)[destAPT] \\ \wedge & LATC \neq det\_destATC(AC) \end{array}\right)$
$\Rightarrow \Diamond Handoff(LATC, nextATC, AC)$

PROOF:

Follows by Propositional Logic. □

$\langle 7\rangle 4.$ $\left(\begin{array}{ll} \wedge & \Diamond Handoff(LATC, nextATC(AC), AC)) \\ \wedge & nextATC(AC) = BATC \end{array}\right) \Rightarrow \Diamond Handoff(LATC, BATC, AC)$

PROOF:

Follows by Propositional Logic. □

$\langle 7\rangle 5.$ Q.E.D.

Follows by Propositional Logic (transitivity of $\Rightarrow$) from $\langle 7\rangle 1$, $\langle 7\rangle 2$, $\langle 7\rangle 3$ and $\langle 7\rangle 4$. □

$\langle 6\rangle 3.$ Q.E.D.

Follows by Propositional logic from $\langle 6\rangle 1$ and $\langle 6\rangle 2$. □

$\langle 5\rangle 2.$ $\Diamond Handoff(LATC, BATC, AC) \Rightarrow \Diamond(AC)in\_area(BATC)$

PROOF:

$\langle 6\rangle 1.$ $\Diamond \left(\begin{array}{ll} \wedge & (AC)in\_area(LATC) \\ \wedge & (AC)in\_area(BATC) \\ \wedge & nextATC(AC) = BATC \\ \wedge & \Diamond \left(\begin{array}{ll} \wedge & \neg(AC)in\_area(LATC) \\ \wedge & (AC)in\_area(BATC) \end{array}\right) \\ \wedge & \left(\begin{array}{ll} \vee & ATCcomm\_history.Handoff_{correct}(LATC, BATC, fid(AC)) \\ \vee & ATCcomm\_history.Handoff_{incorrect}(LATC, BATC, fid(AC)) \end{array}\right) \end{array}\right)$
$\Rightarrow \Diamond$(AC)in_area(BATC)

PROOF:

Follows immediately by Propositional Logic.

$\langle 6\rangle 2.$ Q.E.D.

PROOF:

Insertion of definition of $SOP\_Specs.Handoff(LATC, BATC, AC)$ into $\langle 5\rangle 2$.

$\langle 5\rangle 3.$ Q.E.D.

Follows by Propositional logic from $\langle 5\rangle 1$ and $\langle 5\rangle 2$. □

$\langle 4\rangle 2.$ Q.E.D.

Insertion of interpretations from $\langle 1\rangle 1$, $\langle 1\rangle 2$, and $\langle 1\rangle 3$ into $\langle 2\rangle 3$. □

$\langle 3\rangle 2.$ Q.E.D.

Follows by definition of strict implication (inference rule 5). □

$\langle 2\rangle 4.$ $\Diamond\langle 2\rangle$

PROOF:

This is a fact explicitly given in the sources.

$\langle 2\rangle 5.$ $\Box\neg\langle 2\rangle \Box\!\!\rightarrow \neg(Hypotheses \wedge Procedures)$

PROOF:

$\langle 3\rangle 1.\ \Box\neg(AC)in\_area(BATC)\ \Box\rightarrow \left(\begin{array}{ll} \vee & \neg(AC)in\_area(LATC) \\ \vee & \neg SOP\_Specs.ATC{-}Responsibility{-}Rule \\ \vee & \neg fdata(ac)[destAPT] = `BRU` \\ \vee & \neg nextATC(AC) = BATC \\ \vee & \neg responsibleATC(`BRU`) = BATC \\ \vee & \neg destAPT(AC) = `FRA` \end{array}\right)$

PROOF:

We argue, that in the nearest possible world the flight data is wrong at BATC. Under this assumption, all possible consequences ($\neg fdata(ac)[destAPT] = `BRU`$), ($\neg nextATC(AC) = BATC$), ($\neg responsibleATC(`BRU`) = BATC$) and ($\neg destAPT(AC) = `FRA`$) will lead to $\neg(AC)in\_area(BATC)$.

$\langle 3\rangle 2.$ Q.E.D.

Follows by Propositional Logic, STL2 and STL3 as well as use of definitions from $\langle 2\rangle 1$ and $\langle 2\rangle 2$.

$\langle 2\rangle 6.$ Q.E.D.

Directly follows by Inference Rule C.6 from $\langle 2\rangle 1$, $\langle 2\rangle 2$, $\langle 2\rangle 3$, $\langle 2\rangle 4$. □and $\langle 2\rangle 5$. □

Remarks:

Possible to specify FDP-Procedures as a table based on *Letters of agreement* between ATCCs. (Example Nolan p.413)

4.3.8 Proof of Node 22

Nodes used in the following module:

⟨22⟩ /* LATC uses false flight data for NW052 */
[221] /* London received false data from SATC */

module *⟨22⟩_is_explained_causally_sufficient*

VARIABLES
extends *This_ATCcomm_history*

THEOREM $[221] \Box\!\!\Rightarrow \langle 22 \rangle$

PROOF:

⟨1⟩1. $[221] \Box\!\!\Rightarrow \langle 22 \rangle$
⟨2⟩1. $[221] \not\Rightarrow \langle 22 \rangle$
PROOF:
⟨3⟩1. $[221] \wedge \langle 22 \rangle$
PROOF:
⟨4⟩1. $[221]$
PROOF: This is a fact assumed in the sources.
⟨4⟩2. $\langle 22 \rangle$
PROOF: This is a fact assumed in the sources.
⟨4⟩3. Q.E.D.
($\wedge$-introduction) from ⟨4⟩1 and ⟨4⟩2. □
⟨3⟩2. $\neg[221] \Box\!\!\rightarrow \neg\langle 22 \rangle$
PROOF:
⟨4⟩1. PROOF:
In the nearest possible world the transfer of flight data would have occurred as specified in TLA-module *This_ATCcomm_history*. Assuming a correct transmission ($\neg[221]$), we can derive:

$$\begin{array}{lll} \neg[221] & \Rightarrow & StoL_{correct} \\ & \Rightarrow & \text{UNCHANGED } persistent\_data \\ & \Rightarrow & \neg\langle 22 \rangle \end{array}$$

⟨3⟩3. Q.E.D.
Directly follows by Inference Rule 2.8 from ⟨3⟩1 and ⟨3⟩2. □
⟨2⟩2. $\neg\langle 22 \rangle \Box\!\!\rightarrow \neg[221]$
In the nearest possible world transmitting correct data is greater more probable than making several incorrect handoffs resulting in finally the same correct flightdata. According to the specification of *This_ATCcomm_history*, we can infer:

$\neg\langle 22 \rangle \Rightarrow persistent\_data[destination]' = \text{“FRA”}$

Therefore:

$$\begin{array}{lll} \left(\begin{array}{ll} \wedge & persistent\_data[destination]' = \text{“FRA”} \\ \wedge & persistent\_data[destination] = \text{“FRA”} \end{array} \right) & \Rightarrow & \text{UNCHANGED } persistent\_data \\ & \Rightarrow & StoL_{correct} \\ & \Rightarrow & \neg[221] \end{array}$$

⟨2⟩3. Q.E.D.
Follows by definition of “ $\Box\!\!\Rightarrow$ ” and from steps ⟨2⟩1 and ⟨2⟩2. □

4.3.9 Proof of Node 3

Nodes used in the following module:

⟨3⟩	/* AC in LATC area */
⟨31⟩	/* SATC handoff procedures under this flightplan are to LATC */
⟨32⟩	/* FI is correct at SATC */
⟨4⟩	/* AC in SATC area */

module *⟨3⟩_is_explained_causally_sufficient*

VARIABLES
extends *SOP_Specs*
CONSTANTS $AC, destAPT, APT, SATC, LATC, BATC$

$$\begin{array}{lcl} Hypotheses & \triangleq & \wedge\ \langle 32\rangle \\ & & \wedge\ \langle 4\rangle \\ Procedures & \triangleq & \langle 31\rangle \end{array}$$

THEOREM $(Hypotheses \wedge Procedures) \Box\!\!\Rightarrow \Diamond\langle 3\rangle$

PROOF:

⟨1⟩1. $\langle 32\rangle \Leftrightarrow \left(\begin{array}{ll} \wedge & fdata(ac)[destAPT] = det\_destATP(AC) \\ & = destAP(AC) = \text{`}FRA\text{`} \\ \wedge & nextATC(AC) = LATC \\ \wedge & responsibleATC(\text{`}BRU\text{`}) = BATC \end{array}\right)$

PROOF:
All we use from the flighdata is the information on the AC's destination airport. This piece of information usually suffices to calculate a flight route containing the start and landing ATCCs as well as all intermediate ATCCs. We do not specify this process at this point and define this information as a part of the conjunct describing ⟨32⟩ instead.

⟨1⟩2. $\langle 4\rangle \Leftrightarrow (AC)in\_area(SATC)$
PROOF:
We interpret ⟨4⟩ as a state predicate.

⟨1⟩3. $\langle 31\rangle \Leftrightarrow \left(\begin{array}{ll} \wedge & nextATC(AC) = LATC \\ \wedge & SOP\_Specs.Spec \end{array}\right)$
PROOF:
Our interpretation of ⟨31⟩ is that generally the procedures are followed as specified in *SOP_Specs* that therefore the next ATCC the flightdata will be sent to, is LATC.

⟨1⟩4. $(Hypotheses \wedge Procedures) \Box\!\!\Rightarrow \Diamond\langle 3\rangle$
⟨2⟩1. *Hypotheses*
PROOF:
⟨3⟩1. ⟨32⟩
PROOF:
⟨32⟩ under the interpretation of ⟨1⟩1 is true from the sources.
⟨3⟩2. ⟨3⟩
PROOF:
⟨4⟩ under the interpretation of ⟨1⟩2 is true from the sources.
⟨3⟩3. Q.E.D.

$\wedge$-intro from $\langle 3\rangle 1$ and $\langle 3\rangle 2$. □

$\langle 2\rangle 2$. *Procedures*
PROOF:
$\langle 3\rangle 1$. $\langle 31\rangle$
PROOF:
$\langle 31\rangle$ under the interpretation of $\langle 1\rangle 3$ is true from the sources.
$\langle 3\rangle 2$. Q.E.D.
Follows immediately from $\langle 3\rangle 1$. □

$\langle 2\rangle 3$. $(Hypotheses \wedge \Box Procedures) \succ \Diamond\langle 3\rangle$
PROOF:
$\langle 3\rangle 1$. $\vdash_{TLA}(Hypotheses \wedge \Box Procedures) \Rightarrow \Diamond\langle 3\rangle$
PROOF:
We argue in the forward sense of natural deduction.
We reformulate this expression in logic. Formulation of the goal as well as use of the logical equivalents defined above results in:

$$\langle 4\rangle 1. \left(\begin{array}{ll} \wedge & fdata(ac)[destAPT] = det\_destATP(AC) \\ & = destAP(AC) = \text{`}FRA\text{`} \\ \wedge & nextATC(AC) = LATC \\ \wedge & responsible\_atc(\text{`}BRU\text{`}) = BATC \\ \wedge & (AC)in\_area(SATC) \\ \wedge & \Box nextATC(AC) = LATC \\ \wedge & \Box SOP\_Specs.Spec \end{array}\right) \Rightarrow \Diamond(AC)in\_area(LATC)$$

PROOF:

$$\langle 5\rangle 1. \left(\begin{array}{ll} \wedge & fdata(ac)[destAPT] = det\_destATP(AC) \\ & = destAP(AC) = \text{`}FRA\text{`} \\ \wedge & nextATC(AC) = LATC \\ \wedge & responsible\_atc(\text{`}BRU\text{`}) = BATC \\ \wedge & (AC)in\_area(SATC) \\ \wedge & \Box nextATC(AC) = LATC \\ \wedge & \Box SOP\_Specs.Spec \end{array}\right) \Rightarrow \Diamond Inter\_ATC\_Handoff(SATC, LATC, AC)$$

PROOF:
$$\langle 6\rangle 1. \left(\begin{array}{ll} \wedge & (AC)in\_area(SATC) \\ \wedge & \Box SOP\_Specs.Spec \end{array}\right) \Rightarrow EnRouteProcessing(SATC, AC)$$
PROOF:
Follows by STL2 and Propositional Logic from *SOP_Specs.ATC-Responsibility-Rule*.

$$\langle 6\rangle 2. \left(\begin{array}{ll} \wedge & EnRouteProcessing(SATC, AC) \\ \wedge & fdata(ac)[destAPT] = det\_destATP(AC) \\ & = destAP(AC) = \text{`}FRA\text{`} \\ \wedge & nextATC(AC) = LATC \\ \wedge & responsible\_atc(\text{`}BRU\text{`}) = BATC \\ \wedge & (AC)in\_area(SATC) \\ \wedge & \Box nextATC(AC) = LATC \\ \wedge & \Box SOP\_Specs.Spec \end{array}\right) \Rightarrow \Diamond Inter\_ATC\_Handoff(SATC, LATC, AC)$$

PROOF:
$\langle 7\rangle 1$. $(fdata(ac)[destAPT] = det\_destATP(AC) = destAP(AC) = \text{`}FRA\text{`}) \Rightarrow SATC \neq det\_destATC(AC)$
PROOF:

$\langle 8\rangle 1.$ $\left(\begin{array}{ll} \wedge & fdata(ac)[destAPT] = det\_destATP(AC) \\ & = destAP(AC) = \text{`}FRA\text{`} \\ \wedge & responsible\_atc(\text{`}BRU\text{`}) = BATC \end{array}\right)$
$\Rightarrow (det\_destATC(AC) = BATC)$

PROOF:

Follows immediately from $SOP\_Specs.RespDestATC\_Determination\_Rule$.

$\langle 8\rangle 2.$ $BATC \neq SATC$

PROOF:

We now from the sources that this is true.

$\langle 8\rangle 3.$ Q.E.D.

PROOF:Follows by Propositional Logic from $\langle 8\rangle 1$ and $\langle 8\rangle 2$.

$\langle 7\rangle 2.$ $\left(\begin{array}{ll} \wedge & EnRouteProcessing(SATC, AC) \\ \wedge & (AC)in\_area(SATC) \\ \wedge & fdata(ac)[destAPT] = det\_destATP(AC) \\ & = destAP(AC) = \text{`}FRA\text{`} \\ \wedge & SATC \neq det\_destATC(AC) \end{array}\right)$
$\Rightarrow \Diamond Inter\_ATC\_Handoff(SATC, nextATC, AC)$

PROOF:

Follows by Propositional Logic. □

$\langle 7\rangle 3.$ $\left(\begin{array}{ll} \wedge & \Diamond Inter\_ATC\_Handoff(SATC, nextATC(AC), AC)) \\ \wedge & \Box nextATC(AC) = LATC \end{array}\right) \Rightarrow$
$\Diamond Inter\_ATC\_Handoff(SATC, LATC, AC)$

PROOF:

Follows by STL2 and Propositional Logic. □

$\langle 7\rangle 4.$ Q.E.D.

Follows by Propositional Logic (transitivity of $\Rightarrow$) from $\langle 7\rangle 1$, $\langle 7\rangle 2$ and $\langle 7\rangle 3$. □

$\langle 6\rangle 3.$ Q.E.D.

Follows by Propositional logic from $\langle 6\rangle 1$ and $\langle 6\rangle 2$. □

$\langle 5\rangle 2.$ $\Diamond Inter\_ATC\_Handoff(SATC, LATC, AC) \Rightarrow \Diamond (AC)in\_area(LATC)$

PROOF:

$\langle 6\rangle 1.$ $\Diamond \left(\begin{array}{ll} \wedge & (AC)in\_area(SATC) \\ \wedge & (AC)in\_area(LATC) \\ \wedge & \Box nextATC(AC) = LATC \\ \wedge & \Diamond \left(\begin{array}{ll} \wedge & \neg(AC)in\_area(SATC) \\ \wedge & (AC)in\_area(LATC) \end{array}\right) \\ \wedge & \left(\begin{array}{ll} \vee & ATCcomm\_history.Handoff_{\,correct}(SATC, LATC, fid(AC)) \\ \vee & ATCcomm\_history.Handoff_{\,incorrect}(SATC, LATC, fid(AC)) \end{array}\right) \end{array}\right)$
$\Rightarrow \Diamond$(AC)in_area(LATC)

PROOF:

Follows immediately by Propositional Logic.

$\langle 6\rangle 2.$ Q.E.D.

PROOF:

Insertion of definition of $SOP\_Specs.Inter\_ATC\_Handoff(SATC, LATC, AC)$ into $\langle 5\rangle 2$.

$\langle 5\rangle 3.$ Q.E.D.

Follows by Propositional logic from $\langle 5\rangle 1$ and $\langle 5\rangle 2$. □

$\langle 4\rangle 2.$ Q.E.D.

Insertion of interpretations from $\langle 1\rangle 1$, $\langle 1\rangle 2$, and $\langle 1\rangle 3$ into $\langle 2\rangle 3$. □

$\langle 3\rangle 2.$ Q.E.D.

Follows by definition of strict implication (inference rule 5). □

$\langle 2\rangle 4.$ $\Diamond\langle 3\rangle$

PROOF:

This is a fact explicitly given in the sources.

⟨2⟩5. Q.E.D.

Directly follows by Inference Rule C.6 from ⟨2⟩1, ⟨2⟩2, ⟨2⟩3 and ⟨2⟩4. □

4.3.10 The Full Explanations

Finally we can summarize all *proof obligations* (assumptions we made for the proof). Given the truth of these assumptions, *it follows by logic alone* that the explanation of the incident we gave is correct and sufficient.

module *Proof-Obligations*

THEOREM
History $\langle 4 \rangle \hookrightarrow \langle 3 \rangle \hookrightarrow \langle 2 \rangle \hookrightarrow [1]$ *is completely explained (according to our definitions),* **iff**

1. *our specifications are sufficient to explain the part of the world we needed to focus on. We defined*
   - *A Hierarchy of Communication Procedures:*
     - *ATCtrans:*
       Specification of the communication channel between ATCCs.
     - *ATCproc:*
       Specification of low level communication functions of one ATCC.
     - *ATCcomm:*
       Specification of medium level communication functions (Handoff) from one ATCC to another.
     - *ATCcomm_history:*
       Specification of higher level communication functions between ATCCs.
     - *This_ATCcomm_history:*
       Specification of higher level communication functions between ATCCs for this special history.
   - *Cognitive Procedures:*
     - *PARDIA_Axioms:*
       Specification of axioms stating the relationships between the attitudes of the PARDIA model.
     - *PARDIA_Norms:*
       Specification of certain PARDIA principles which are intended to hold.
   - *Standard Operating Procedures (SOPs):*
     - *AC_ATC_comm.spec:*
       Specification of Communication Procedures between Aircrafts and ATCCs.
     - *Landing_Axioms:*
       General specification of domain rules, which are valid at all times.
     - *Landing_Norms:*
       Specification of SOPs concerning the landing.
     - *Landing_Specs:*
       Specification of rules to be kept in the landing phase (mainly Landing_Axioms and Landing_Norms).
     - *SOP_Specs:*
       Specifiation of the part of SOPs that covers an ATCCs responsibility for evaluation and transmission of flight data.
   - *TLA-Specifications of mathematical functions which describe the behaviour of physical flight parameters over a peroid of time, like altitude for example:*

– *RealTimeTheorems*

2. *these specifications are fullfilled in every case in which no contrary information is explicitly given.*

3. *our TLA interpretions of the text correspond sufficiently to the reality. We assumed:*

   - $\Box\neg ATC\_breakoff$
   - $[11] \Leftrightarrow Decide(CRW, \Box(AC)in\_landing\_phase)$
   - $\langle 12\rangle \Rightarrow (AC)near(BRU)$
   - $[111] \Rightarrow (APT \neq destAPT)$
   - $[1111] \Leftrightarrow \Diamond visual\_contact(crw, apt)$
   - $[11111] \Leftrightarrow below\_clouds(ac)$
   - $\langle 11112\rangle \Leftrightarrow \wedge\ ILS\_approach(ac, apt)$
     $\wedge\ \Box\neg CRW\_breakoff\,(crw, appr)$
   - $\langle 2\rangle \Leftrightarrow (ac)in\_area(atc)$
   - $\langle 21\rangle \Rightarrow SOP\_Specs.ATC{-}Responsibility{-}Rule$
     In the sources the Procedures are assumed to be implemented correctly. We claim $SOP\_Specs.ATC{-}Responsibility{-}Rule$ *is sufficient to explain the ATC procedures significant for this case.*
   - $\langle 22\rangle \Leftrightarrow \left(\begin{array}{ll} \wedge & fdata(ac)[destAPT] = \text{`}BRU\text{`} \\ \wedge & nextATC(AC) = BATC \\ \wedge & responsibleATC(\text{`}BRU\text{`}) = BATC \\ \wedge & destAPT(AC) = \text{`}FRA\text{`} \end{array}\right)$
     We interpret "false flightdata" as deviation of the used data - determined from the ATCs FDC - from the original data. All we use from the flightdata is the information on the AC's destination airport. This piece of information usually suffices to calculate a flight route containing the start and landing ATCCs as well as all intermediate ATCCs. We do not specify this process at this point and define this information as a part of the conjuncts describing $\langle 22\rangle$ *and* $\langle 32\rangle$ *instead.*
   - $\langle 3\rangle \Leftrightarrow (AC)in\_area(LATC)$
   - $\langle 31\rangle \Leftrightarrow \left(\begin{array}{ll} \wedge & nextATC(AC) = LATC \\ \wedge & SOP\_Specs.Spec \end{array}\right)$
     Our interpretation of $\langle 31\rangle$ *is that generally the procedures are followed as specified in* SOP_Specs *that therefore the next ATCC the flightdata will be sent to, is* $LATC$*.*
   - $\langle 4\rangle \Leftrightarrow (AC)in\_area(SATC)$

4. *several assumptions used in the modal part of proofs of* $[111]$, $\langle 12\rangle$ *and* $\langle 22\rangle$ *are correct:*

   - *Instruments providing the information necessary to determine the position without visual contact were ignored. Therefore the only other way to obtain this information is visual contact. - (used in*$[111]$*)*
   - *The evidence of the instruments was ignored. Thus, if they hadn't recognized the layout they saw was different from the expected (flight documents contain 2D maps of the fact of the destination airport) they wouldn't have recognized their mistake at that time.- (used in*$[111]$*)*

- *No use of evidence from instrumentation was made. If neither* [1111] *nor* {1112} *then no evident was available that could call their attention to the wrong airport.- (used in*[111]*)*
- *The flight destination was Frankfurt. If the Crew had realized they were on the wrong course, they would likely have taken actions to return to the course they were supposed to be on unless other more pressing considerations intervened.- (used in*⟨12⟩*)*
- *In this particular case we assume that an AC only enters a specific area when it has obtained an ATC clearance before. Therefore without a clearance to BATC area the AC will not be able to get into Brussels Airport area.- (used in*⟨12⟩*)*
- *Since they are assumed to follow the SOPs in the nearest possible world, they would have realized hints about a possible wrong course.- (used in*⟨12⟩*)*

4.4 The Final WB-Graph

Given the proof we just finished, we're now able to provide the textual and pictorial form of the causally sufficient WB-Graph:

```
[1] /* AC lands at Brussels RWY 25 */
  /\[-.1] /* CRW opts to continue landing */
  /\<-.2> /* AC near Brussels Airport */

  [1.1] /\[-.1] /* CRW realizes they are landing at the wrong airport */
        /\<-.2> /* CRW has safety reasons for continuing landing */
        /\[-.3] /* Standard Operating Procedures */

    [1.1.1] /\[-.1] /* CRW gets visual contact to Brussels airport */
            /\{-.2} /* CRW notices that Brussels' airport layout is different
                       from Frankfurt's */
      [1.1.1.1]  /\[-.1] /* AC breaks out under clouds */
                 /\<-.2> /* CRW procedures */
                 /\<1.2>
                 /\<2>   /* AC in BATC area */

        <2> /\<-.1> /* LATC procedures */
            /\<-.2> /* LATC uses false flight data for NW052 */
            /\<3>   /* AC in LATC area */

          <2.2> [-.1] /* London received false data from SATC */

          <3> /\<-.1> /* SATC handoff procedures under this flightplan
                         are to LATC */
              /\<-.2> /* FI is correct at SATC */
              /\<4>   /* AC in SATC area */

  <1.2> /\(-.1) /* CRW did not realize that they were on wrong course,
                   UNTIL:[111] */
        /\<-.2> /* AC cleared to BATC according to ATC procedures */

    (1.2.1) /\{-.1} /* CRW addresses BATC controller as ``Frankfurt''
                       several times */
            /\<-.2> /* ILS has different frequency for Frankfurt. */
            /\[-.3] /* CRW asks for the Bruno VOR's frequency. */
            /\(-.4) /* Brussels did not question the addressing error
                       although it happened more than once */
            /\<-.5> /* Situation remains safe during landing */
            /\<-.6> /* Current approach plates are used */
```

Figure 4.4: Textual form of the final WB-Graph

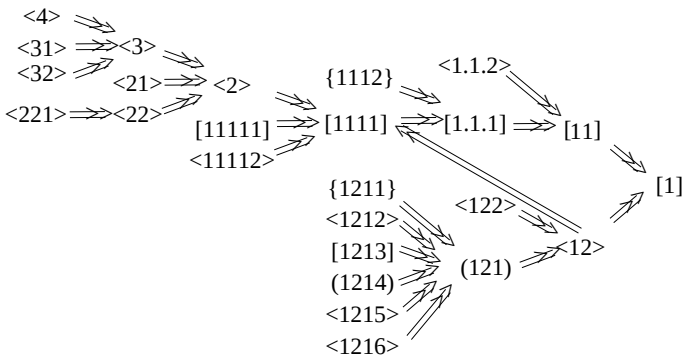

Figure 4.5: Pictorial form of the final WB-Graph

Chapter 5

Conclusions

We explained the "Why..Because"-Analysis (WBA) method for failure analysis. By analyzing a sample case based on secondary information from journal articles we demonstrated the applicability of this method to a complete incident analysis. We collected facts from the articles and presented how to use modal logic, possible world semantics and predicate action diagrams (PADs) to gain and introduce additional information from other sources into the analysis. As result of the semi-formal analysis of chapter 2, we presented a textual and pictorial form of a WB-Graph, displaying the causality relations between the facts we found. By a rigorously formal proof, we determined several necessary changes to be made to the intuitive graph. Our final result is a WB-Graph we claim is explained causally sufficient under the proof obligations of section 4.3.10. Given the truth of these assumptions, it follows by explanatory logic (EL) that the explanation we gave about the incident is correct and sufficient.

Some comments on the experiences we made during analysis:

Formal Methods

The use of formal methods as a means of description assures that we can judge objectively. Of course, we cannot avoid that the reader introduces his subjective interpretation into our analysis. This is a problem of accident and incident reports, we cannot solve. However, the use of formal methods forces the analyst to argue in an objective way.
Formal methods are not always easy to use. As soon as more than one type of system is concerned, the required methods become increasingly complex. Several suggestions were made to introduce modal logic and other formalisms (e.g. Petri Nets [Joh95], TLA [Joh97] or requirements engineering [MHCM96]) into the analysis of systems which contain human factors. These methods work well for sample systems, as the publications demonstrate. But they do not cover problems like the scoping of variables and the rapidly increasing size of specifications due to interactions between particular components. These problems arise during our analysis of *complete* accident reports. The module structure of TLA+ helps to cope with the complexity of specifications. Moreover, TLA+ defines a scoping of variables.
The main difficulty we had to cope with during the specification phase was to keep the specifications consistent. Every new formula has to be checked for consistency with the existing formulas. Often modifications of existing formulas are required; possibly effecting other definitions.
Sometimes an unprecise translation from the natural language formulations of the intuitive WB-Graph to the logic EL is difficult. Due to the adaptation of differ-

ent modal logics to EL, it is possible to create contradict formulas. Therefore, the translation from these statements to logic must be very precise. Formulating the supposed causes during the phase of intuitive WB-Graph construction precisely in natural language reduces the number of these translation problems significantly. As soon as an appropriate specification is found for the statements in the graph, the proof becomes much easier.

Perhaps the difficulties listed above contributed to the fact, that formal methods are not very popular. Nevertheless, we claim that they are unavoidable in engineering. Probably the best evidence for the correctness of this point of view is the success of compilers during the last years: The methods used in the development of compilers are purely formal. However, most people dealing with computers in engineering accept compilers as their daily work's usual tool.
WBA, of course, is not as mature yet as compilers are today. Compilers had several decades for their development; WBA just started to grow up. We are sure, that EL is not necessarily complete yet. In fact since it includes ZF set theory and arithmetics, it cannot be complete in general at all. I can at most be complete for a restricted field. We showed, that EL in its current form is sufficient for the specifications and verifications we needed, yet.

Since we use rigorously formal methods for the analysis, we believe (at least) some part's automation of our work should be possible. One could think of using generic theorem-provers like *Isabelle* [PN98] for parts of correctness and sufficiency proof. At several points we detected difficulties that prevented us from using straight forward proof rules. We believe, at these points an automatic system would have these difficulties, too. However, further research in developing additional proofrules and e.g. an Isabelle-theory could reduce these difficulties. In fact, *we* did solve them; so this problem's automatic solution should be possible.

Proofs

While giving the correctness and completeness proof given in chapter 4 we needed to incorporate several changes into the intuitive WB-Graph. We introduced four new nodes:

```
[1.1.3] /* Standard Operating Procedures */
<1.2.2> /* AC cleared to BATC according to ATC procedures */
<1.2.1.5> /* Situation remains safe during landing */
<1.2.1.6> /* Current approach plates are used */
```

One node could be shown to be redundant:

```
<3.3> /* AC-PD is sufficiently consistent with FI at SATC */
```

Furthermore, we needed to make some major changes to the structure of the graph[1] (compare figures 2.6 and 4.5).

These changes demonstrate, that the intuitive argument is not necessarily logically correct. The use of formal methods helps to find these inadequacies.

[1] Notice that by changing the structure of the WB-Graph also the enumeration of nodes was changed. This enumeration was necessary to obtain a new systematic enumeration. Nevertheless, we kept the original nodes `<2>`, `<3>` and `<4>` introduced in figure 2.1 through the whole thesis to maintain at least the basic nodes.

WB-Graphs

Two forms of graph representation were presented:

1. The textual WB-Graph presents the information in a notation determined syntactically by an EBNF (fig. 2.7). This representation is useful during the specification and verification phase.

2. The pictorial WB-Graph depicts the information in an intuitive graphic. We claim, this representation helps to gain a quick structural overview.

Both representations are compact enough to gain a quick overview on the important facts leading to the incident. We believe they are a suitable means to present the wide-spread information of incident and accident reports in a compact form.

Currently, tools are developed to generate pictorial graph from the textual form automatically ([Höh98], using [EKN98]). A result displaying the graph, and a statistics (included in the Legend) is shown in figure 5.1.

This is just a first step of the automation of our work. A suggestion for a system assisting the development of WB-Graph is given in figure 5.2. The system combines tools for assisting the input of data, a database for the storage of this data, a consistency check using the language DATR (see [EGK97] for further information) as presented in [Ger97] as well as tools for the creation of WB-Graphs in textual, pictorial and DATR-format. The system may installed on a World Wide Web or LAN Server to provide access to a group of users. The implementation of the system should be straight forward, since the tools suggested to use already exist. An interface to an automated theorem prover assisting correctness proofs is conceivable.

The PARDIA Model

We explained the PARDIA-Model of human error. In this thesis, as well as in other analyses ([GLL97a]), the sufficiency of this model to classify human errors was demonstrated. The model cannot explain human error, but it can classify the errors and suggest possible improvements (as demonstrated in figure 3.9). Further effort and support from psychology colleagues might help to improve this classification and perhaps allow to make more precise suggestions concerning possible improvements.

Resource-Calculations

For this development-case we needed the resources shown in figure 5.3. These numbers may be interpreted as "worst case" estimation. We believe with a growing expertise, we will be able to reduce the resource requirements significantly.

The amount time required for the development of the intuitive Graph heavily depends on the availability of expert knowledge of the aviation domain. So the amount of time does, which was used for the formulation of specifications. Under the premise, that sufficient access to expert knowledge is available, we claim the resources required for position 1 are linear to the size of the problem[2].
Although expert knowledge would reduce one big problem during the development of sufficient specifications – the lack of knowledge on the domain – we claim, that the necessary amount of resources for position 2 primarily depends on the size of the problem. Every new formula introduced needs to be checked for consistency against all existing formulas.
Given proper specifications, the resources for the correctness and completeness proof

[2]Let us measure this size in the number of nodes in the WB-Graph.

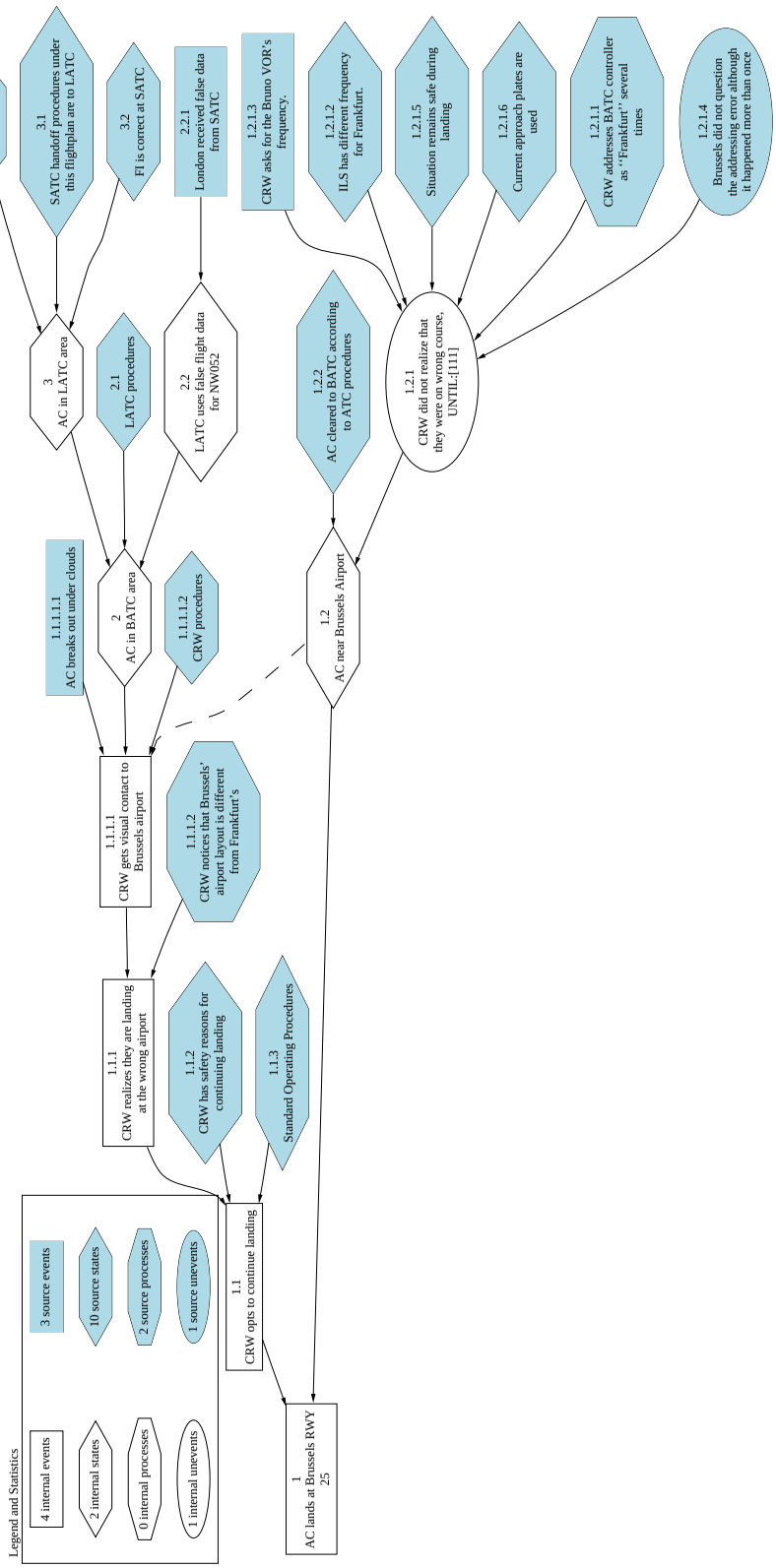

Figure 5.1: Automatically generated WB-Graph

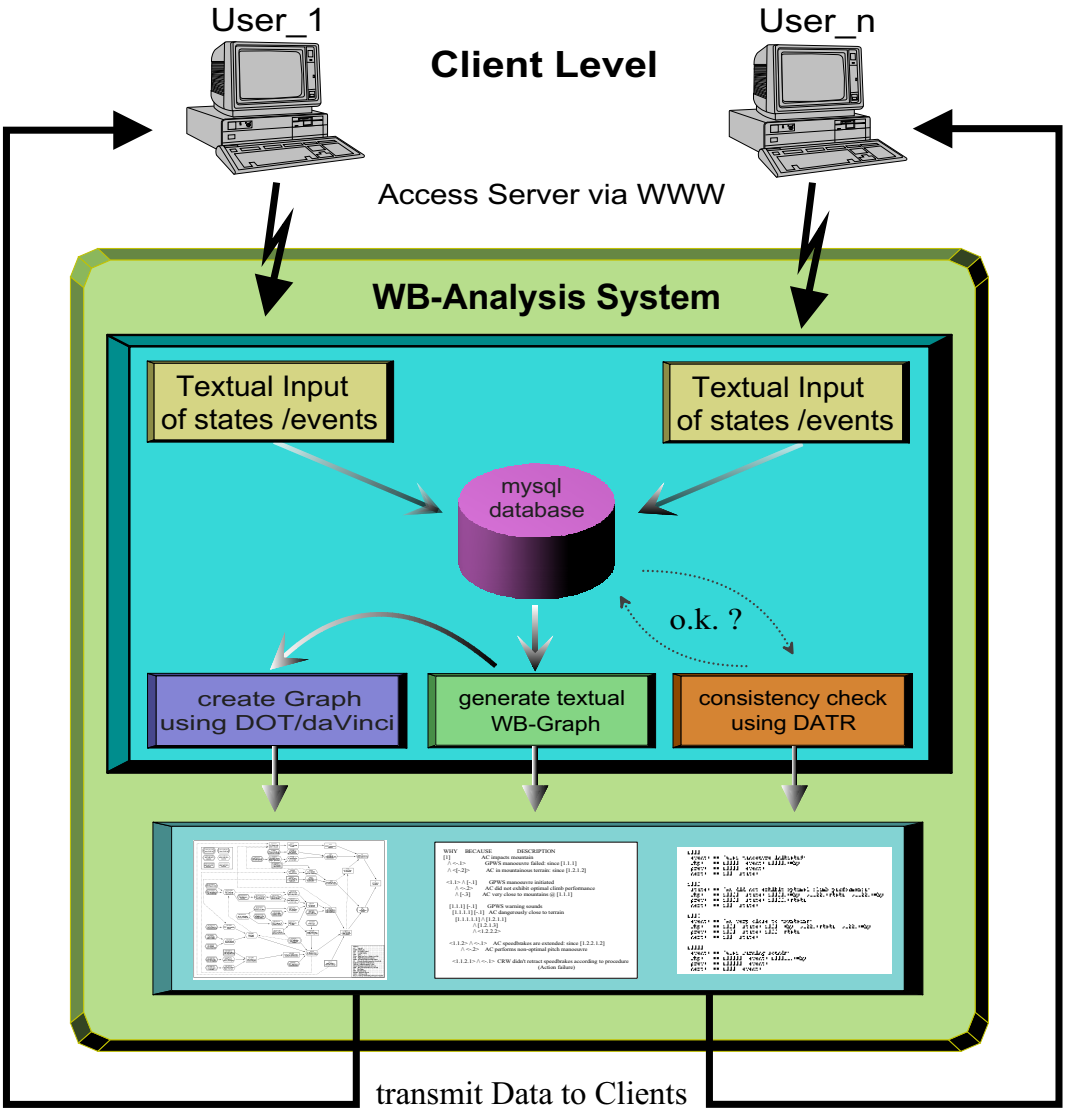

Figure 5.2: Architecture of a possible system to assist WB-Graph development.

Pos.	Activity	approx. Man-hrs	Dependencies
1	Development of intuitive WB-Graph	80	Expert Knowledge/ Problem size
2	Formulation of Specifications	400	Expert Knowledge/ Problem size
3	Completeness and Correctness Proof	320	Problem size

Figure 5.3: Need of Resources for the presented Analysis (Worst case estimation from a development case for a WB-Graph with 26 nodes)

	estimated Man-Hours		
	NW052	Cali	Nagoya
Dev. of intuitive WB-Graph	80	250	300
Specifications	400	$-+$	$-+$
Proof	320	900	1200
Total:	800	> 1150	> 1500
Size [Number of nodes]	26	≈ 70	≈ 100

Figure 5.4: Estimated Need of Resources for the Analysis of selected Reports

depend primarily on the size of the problem. We usually can derive all the knowledge from the specifications.

These observations allow to estimate the required resources for other analyses: The biggest example we know yet, is the report concerning the China Airlines Flight B1816 accident at Nagoya Airport on April 26, 1994 [Com96]. The intuitive graph consists of 88 nodes [LL9X]. We believe it will grow to approximately 100 nodes during the proof. As intermediate example let us consider American Airlines flight 965 accident near Cali, Colombia, on Dec 20, 1995 [oTRoC96]. From the intuitive analysis ([GLL97a]) we know about 59 causally relevant factors – this number may grow to around 70 during a proof. Using the methods given right now, we estimate the (worst-case) need for resources as shown in figure 5.4. We cannot estimate the numbers of required man-hours for the specifications exactly, since all three cases deal with incidents and accidents which occurred during landing phase. Therefore the specifications likely will be restricted to Landing Procedures. This would allow the re-use and expansion of the existing specifications. So, starting from nothing will not be required.

The most effective action to reduce these numbers would be to develop a basic set of specifications describing domain-specific systems and procedures. One could e.g. translate [Adma] into EL and create "library modules" for different fields. This would reduce the specification work mainly to search the logical counterpart to a procedure/system description from the given libraries. The number of formulas to be introduced newly into the specification would be small.
A second approach would be to develop methods for assisting the proof, as described above. Although the potential of resource reduction is perhaps smaller than for the optimization suggested above, this approach seems to be more prospering, since the complexity is lower.
However, we know what can be done and we also know, how to do it. We can estimate the need of resources and therefore provide reliable time-schedules for a possible project. Implementing the suggested improvements can only decrease the need for resources for future projects.

Appendix A

DC-10 misses Frankfurt runway – by 300km

A.1 Report from *Flight International* [dWL95]

A.2 DC-10 misses Frankfurt - by 300km

A Northwest Airlines McDonnell Douglas DC-10-40 carrying 241 passengers from Detroit to Frankfurt missed its intended destination by 300km (160nm), landing at Brussels Airport by mistake on 5 September.
The pilots of Flight 52 only realised their error when they lined the aircraft up for the approach at Brussels. In spite of that, they decided to land anyway. [...]
Early reports from Brussels air traffic control (ATC) attributed the original error to Shannon ATC, alleging that an incorrect code had been entered into the aircraft's ARC flight-plan data, redesignating the aircraft's destination to Brussels.
The Irish Aviation Authority denies this, saying that the crew had acknowledged its destination as Frankfurt, and that the correct data was passed to the London Air Traffic Control Center, the last such center before Brussels.
By the time the aircraft entered the Brussels control region, however, its destination has been redesignated, Brussels ATC maintains. [...]
The aircraft's planned track for Frankfurt would normally have taken it over Belgium at its cruising altitude of 37,000ft (11,300m), according to ATC conditions. The upper-airspace (above 24,500ft) over Belgium, however, is handeled by the Maastricht ATCC in the Netherlands.
A senior Brussels ATC official confirms that the aircraft was cleared by the LATCC as it left the London control region to descend to 24,000ft and contact Brussels. The crew started the descent and called Brussels on the assigned frequency, addressing the controller as "Frankfurt" and announcing its intention to land.
Brussels did not question the addressing error which, Northwest says, occurred more than once in subsequent transmissions.
Brussels approach instructed the crew to descend in-bound via Bruno, a VOR navigation beacon on one of the standard approaches to Brussels Airport. The crew had to ask ATC for the VOR's frequency. The aircraft was subsequently cleared for an instrument-landing system (ILS) approach to Brussels's runway 25L, which is the same runway orientation as at Frankfurt, but with different ILS frequencies.
At some point the crew finally realised they were landing at the wrong airport and opted to continue the landing for safety reasons, says Northwest. The airline has said that, whatever errors ATC may have made, if any, the crew must "share responsibility" for what happened.

A.3 Article on this subject in Risks 17.38 [Lad95a]

Fly NorthWest Airlines to unknown destinations

<ladkin@techfak.uni-bielefeld.de> Sun, 8 Oct 1995 17:14:49 +0100

The International Herald Tribune for Monday Oct 2, p1, has a report on a DC10, NorthWest Flight 52, on its way to Frankfurt from Detroit. They landed in Brussels, much to everyone's surprise except for the passengers, cabin crew and air traffic control.

A controller in Shannon changed the destination in the en-route computers for some reason no-one has fathomed. So everyone after that sent NW52 merrily on the way to Brussels. The cabin crew and passengers noticed, because the cabin flight-path display was showing them going to Brussels rather than to Frankfurt (the cities are 200miles=325km away from each other). The flight crew first noticed when they broke out under the clouds on approach to Brussels, and noticed that the layout of the airport was not similar to Frankfurt. Sensibly, they decided to continue the landing. And will remain landed until the investigation figures everything out. A spokesman for NorthWest pointed out that the crew *should* have known where they were.....

That reminds me of the time I was flying Chicago to SFO and following the ground on my WAC (World Aeronautical Chart). The routing went south of the Colorado/Wyoming boundary, past Aspen, and then over the Green river canyon, which is some 250km past Aspen. Just then, the captain announced "We're just passing Aspen, Colorado, out of the left window." But we got there OK. Even United pilots can recognise the Mina and Coaldale transitions to the Modesto arrival when it hits them ;-)

Peter Ladkin.

A.4 Article on this subject in Risks 17.40 [Lad95b]

<ladkin@techfak.uni-bielefeld.de> Wed, 18 Oct 1995 08:20:00 -0700

An article in Flight International, 11-17 October 1995, p8, entitled 'DC-10 misses Frankfurt runway–by 300km', considers the aftermath.

Brussels ATC attributed the original error to the Shannon ATC controller entering an incorrect code to the ATC flight-plan data. The Irish Aviation Authority denies this, saying the correct code was passed to London ATCC, the last such ATCC before Brussels. Brussels maintains that when the aircraft got to them, the destination data had been changed. 'A senior Brussels ATC official' confirms that NW52 was cleared by London ATCC as it left the London control region to descend to 24,000 ft (I think they mean Flight Level 240 but I'm not sure – I'll use FL's anyway). The aircraft's planned track for Frankfurt would have taken it over Belgium at FL370 under control of Maastricht ATCC in the Netherlands, which handles traffic over FL245 across Belgium.

NW52 also addressed Brussels as 'Frankfurt' on contact, and numerous times thereafter. Brussels ATC didn't question the 'addressing error', apparently. They were also cleared to a VOR, Bruno, that they didn't recognise, and asked for the

frequency. They were cleared for an ILS RWY 25L approach, which is the same runway orientation as at Frankfurt, but with a different ILS frequency. NW says that the crew must share responsibility, no matter what happened with ATC (this is in any case what aviation law requires).

It looks like there is a lot for them to discuss.

Peter Ladkin

Appendix B

The Rules of EL

(This is a chapter of [LL98b].)
EL uses the TLA rules, which were defined in [Lam94] and given translation into natural-deduction form in [Lad97]. The sections explaining the classical, temporal and invariance rules modified versions of those in [Lad97].

B.1 Classical Rules

One may simply assume some set of rules or procedures for classical (propositional and predicate) logic, as does Lamport in the TLA definition [Lam94].

For completeness, we give a Prawitz-type natural deduction system [Pra65] with the classical rules, presented in the traditional manner: proofs 'move' downwards towards their desired conclusion.

B.1.1 Propositional Rules

$$(\&\text{-intro})\quad \frac{A \quad B}{A \wedge B} \qquad (\&\text{-elim})\quad \frac{A \wedge B}{A} \quad \frac{A \wedge B}{B}$$

$$(\vee\text{-intro})\quad \frac{A}{A \vee B} \quad \frac{B}{A \vee B} \qquad (\vee\text{-elim})\quad \frac{A \vee B \qquad [A] \; C \qquad [B] \; C}{C}$$

Note that these rules strictly speaking build proof *trees*, not simple sequences of formulas, since proof branches converge in certain rules. Branches converge as we move 'downwards' in the proof towards the desired conclusion. The root of the tree is the desired theorem, and the leaves are the initial assumptions.

Define $(\neg A)$ to be the formula $(A \Rightarrow \bot)$

$$(\Rightarrow\text{-intro})\quad \frac{[A] \; B}{A \Rightarrow B} \qquad (\Rightarrow\text{-elim})\quad \frac{A \quad A \Rightarrow B}{B}$$

$$(\bot_{\mathsf{I}})\quad \frac{\bot}{A} \qquad (\bot_{\mathsf{C}})\quad \frac{[\neg A] \; \bot}{A}$$

A restriction on both $\bot$ rules is that A is different from $\bot$. A restriction on $\bot_{\mathsf{C}}$ is that A is not of the form $B \Rightarrow \bot$ (but this is only for convenience: Exercise:-

Show that if A has the form $\neg B$ in $\perp_C$ that there's another way of deriving the conclusion). Note that $\perp_C$ subsumes $\perp_I$.

The use of $\perp$ is logically important: if $\perp_C$ is used, the theorems are those of classical propositional logic; if $\perp_I$ is used, the theorems are those of intuitionistic propositional logic. The *formal* difference between the two logics is thus reduced just to that of how negation is treated.

B.1.2 Quantifiers

we may include rules for quantifiers as follows:

$$(\forall\text{−intro}) \quad \frac{A}{\forall x \,:\, A[a/x]} \qquad (\forall\text{−elim}) \quad \frac{\forall x \,:\, A}{A[x/t]}$$

in which a does not occur in the assumption set of premise A, nor within the scope of a quantifier binding x in A; resp. no occurence of x in A occurs within the scope of a quantifier binding a variable of t (i.e. the standard logic stuff usually embodied in the definition of *<term> is free for <variable> in <formula>*).

The rules for the existential quantifier are

$$(\exists\text{−intro}) \quad \frac{A[x/t]}{\exists x \,:\, A} \qquad (\exists\text{−elim}) \quad \frac{\exists x \,:\, A \qquad \overset{[A[a/x]\,]}{B}}{B}$$

with appropriate restrictions here too.

B.2 Modal Rules

Following the approach of Prawitz [Pra65], we first define the notion of *essentially tense-modal* formula:

- $\Box A$ is essentially tense-modal;
- $\perp$ is essentially tense-modal;
- the essentially tense-modal formulas are closed under $\wedge$, $\vee$, $\exists$

The rules are

$$(\Box\text{−intro}) \quad \frac{A}{\Box A} \qquad (\Box\text{−elim}) \quad \frac{\Box A}{A}$$

in which the assumption set of the premise of ($\Box$−intro) must include only essentially-modal formulas.

This set of rules defines a natural deduction system whose theorems are those of S4 (as shown in [Pra65]).

Axioms are proof rules with conclusions but without hypotheses, and rather than write an axiom in proof rule notation as $\frac{}{A}$ we write $\vdash A$. The modal notions are more commonly formulated as Hilbert-style systems, with restricted proof rules (usually just *modus ponens* and *necessitation*) and many axioms.

So, alternatively, one may add the rule of *necessitation*:

$$\frac{F}{\Box F}$$

with the constraint that $assumptions(F) = \emptyset$,
and various of the axioms

(K) $\vdash \Box(A \wedge B) \equiv (\Box A \wedge \Box B)$
(T) $\vdash \Box B \Rightarrow B$
$(S4)$ $\vdash \Box B \Rightarrow \Box\Box B$
$(S5)$ $\vdash \Diamond_A B \Rightarrow \Box_A \Diamond_A B$

as one wishes, to obtain the system one needs [HC96]. The basic propositional modal logic contains the axiom K along with the rule of necessitation, and is itself known as K. A model for a modal logic extending K is a collection of classical (propositional or predicate-logical) models, along with a binary relation $\mathcal{W}$ on these models, known as *accessibility*. If $X \mathcal{W} Y$, we say that world Y is accessible from world X. We speak of formulas being satisfied by worlds in a model M, thus $X \models A$ in M means that formula A is true at world X in model M. The inductive rules for truth at a world in a model for the classical connectives and quantifiers are identical to those of classical logic; for example

$$X \models (A \wedge B) \quad \triangleq \quad X \models A \ \wedge \ X \models B$$

$$X \models \neg A \quad \triangleq \quad \neg(X \models A)$$

suffice for propositional logic, since $\{\neg, \wedge\}$ form a sufficient set of connectives for classical propositional logic. The interpretation of the operators $\Box, \Diamond$ uses the accessibility relation essentially:

$$X \models \Box A \quad \triangleq \quad \forall Y (X \mathcal{W} Y \ \Rightarrow \ Y \models A)$$

The rule for $\Diamond$ may be derived from the rule for $\Box$, since $\Diamond \ \triangleq \ \neg\Box\neg$, and is

$$X \models \Diamond A \quad \triangleq \quad \exists Y (X \mathcal{W} Y \ \wedge \ Y \models A)$$

In the Kripke (possible worlds) semantics for modal logic, the accessibility relation in any model for (T) is reflexive and, conversely, any Kripke model with a reflexive accessibility relation is a model for (T). The logic S4 is axiomatised by the axioms $(K) + (T) + (S4)$. The accessibility relation in any model for S4 is reflexive and transitive and, conversely, any Kripke model with a reflexive, transitive accessibility relation is a model for S4. The logic S5 is axiomatised by the axioms $(K) + (T) + (S4) + (S5)$. The accessibility relation in any model for S5 is reflexive, transitive and symmetric, that is, an equivalence relation; conversely, any Kripke model with a reflexive, transitive and symmetric accessibility relation is a model for S5.

B.3 Temporal Rules

We need tense-logical rules sufficient for '*simple linear-time temporal logic*' (STL), as some computer scientists call it. It is equivalent to S4.3. There are also some special TLA rules for handling the '*prime*' operator. While we could extend the Prawitz rules with axioms for S4.3, the existing TLA system is formulated axiomatically by Lamport [Lam94]. We include the Lamport axioms and rules here. We refer readers interested in a natural deduction formulation to that presented in [Lad97].

STL1: $\dfrac{F}{\Box F}$
in which $assumptions(F) = \emptyset$
STL2: $\vdash \Box F \Rightarrow F$
STL3: $\vdash \Box\Box F \equiv \Box F$
STL4: $\dfrac{F \Rightarrow G}{\Box F \Rightarrow \Box G}$
STL5: $\vdash \Box(F \wedge G) \equiv (\Box F) \wedge (\Box G)$
STL6: $\vdash (\Diamond\Box F) \wedge (\Diamond\Box G) \equiv \Diamond\Box(F \wedge G)$
STL7: $\vdash \Box\Diamond\Box F \equiv \Diamond\Box F$

To these STL axioms we add the **Basic TLA Rules** which specifically axiomatise '*prime*':

TLA1. $\dfrac{P \wedge (f' = f) \Rightarrow P'}{\Box P \equiv P \wedge \Box[P \Rightarrow P']_f}$ TLA2. $\dfrac{P \wedge [\mathcal{A}]_f \Rightarrow Q \wedge [\mathcal{B}]_g}{\Box P \wedge \Box[\mathcal{A}]_f \Rightarrow \Box Q \wedge \Box[\mathcal{B}]_g}$

where the notation

$$[\mathcal{A}]_f \triangleq \mathcal{A} \vee (f' = f)$$

B.4 Behaviors and the Rules They Engender

In order to allow proof by induction over temporal time-steps, we need to add an induction rule:

INV1: $\dfrac{I \wedge [\mathcal{N}]_f \Rightarrow I'}{I \wedge \Box[\mathcal{N}]_f \Rightarrow \Box I}$

in which $\mathcal{N}$ is an action formula, thus defining a relation between a given state and the next. We suppose also that f is the state function which is a sequence whose elements are the variables of $\mathcal{N}$ (TLA includes ZF, so sequences can be formed). The rule says intuitively: if it is provable that I is preserved under state-relation $\mathcal{N}$, then it follows if $\mathcal{N}$ always holds and I holds, I will always hold. We also add the technical rule:

INV2: $\vdash \Box I \Rightarrow (\Box[\mathcal{N}]_f \equiv \Box[\mathcal{N} \wedge I \wedge I']_f)$

These two rules are known as the 'Invariance Rules'. We hope that it is clear that these rules are sound. They suffice for many purposes, for example to prove *safety properties* of systems.

B.5 Strict Implication

We observe that the only place that strict implication is used in EL is to restrict the subproof to TLA logic. We do not anticipate use of combinatorial properties of $\succ$ in reasoning about failures in EL. We nevertheless define them, in case they should be needed.

Strict implication $\succ$ is supposed to represent alethic modal implication, that is, (logical or philosophical) *necessity*. Rather than being axiomatised directly as a binary logical operator, $B \succ C$ is usually defined as

$$B \succ C \triangleq \Box_A(B \Rightarrow C) \tag{B.1}$$

and $\Box_A$ is axiomatised. We take this course and take $\Box_A$ to be axiomatised by the S5 logic, as is commonplace for the alethic modality intended to capture logical necessity.

Alternatively, we could use a Prawitz-like approach, defining the notion of *essentially alethically-modal* formula, and then giving rules for $\Box_A$ isomorphic to those above for $\Box$, which yields the logic S4, and adding the $(S5)$ axiom.

B.6 The Deontic Modalities

For the deontic part, we may use *Standard Deontic Logic* (SDL) [MWD97]. Along with *modus ponens* and classical logic, SDL consists of the axioms/rules:

(K_O) $\quad \vdash O(A \Rightarrow B) \Rightarrow (O(A) \Rightarrow O(B))$

(D_O) $\quad \vdash \neg O(\bot)$

(N_O) $\quad \dfrac{A}{O(A)}$

(P) $\quad \vdash P(A) \equiv \neg O(\neg A)$

(F) $\quad \vdash F(A) \equiv O(\neg A)$

However, a clean way of implementing SDL inside an S5 modal logic was discovered by Anderson [And58] (see also [MWD97]). Given an alethic S5-operator $\Box_A$, define a propositional constant V (standing intuitively for '*Violation*'), and further define

$$O(B) \equiv \Box_A(\neg B \Rightarrow V) \tag{B.2}$$

Since we have an alethic S5 operator for the definition of strict implication, we may use the Anderson definition in order to define the deontic modality.

B.7 Counterfactual Conditionals

We handle the counterfactual conditional $A \,\Box\!\!\rightarrow B$ mainly through semantics, although one logical property is needed, to prove the Theorem 1, namely Rule C.4, which is found amongst some rules for counterfactuals given in Section B.9 below.

B.8 Lewis Semantics for Counterfactuals

The Lewis possible-world semantics for counterfactuals [Lew73] is based on the Kripke semantics for modal logic, with an additional relation of nearness:

> World X is at least as near as world Y to world W

The Lewis semantics for $\Box\!\!\rightarrow$ is that

> $A \;\Box\!\!\rightarrow B$ in a world W if and only if B is true in all the nearest worlds to W in which A is true.

We use semantic arguments about nearest possible worlds to evaluate formulas of the form $A \,\Box\!\!\rightarrow B$. We repeat here the mathematical structure concerning nearness amongst possible worlds which is necessary to make such arguments. Fix W. The relation of nearness allows us to define a binary relation $\preceq_W$, namely

$$X \preceq_W Y \tag{B.3}$$

The relation $\preceq_W$ is, according to Lewis, axiomatised as a *total preorder*. A preorder is a binary relation $\preceq$ that satisfies the following properties:

$$x \preceq x \quad (\textit{reflexivity}) \tag{B.4}$$

$$(x \preceq y) \wedge (y \preceq z) \Rightarrow (x \preceq z) \quad (\textit{transitivity}) \tag{B.5}$$

A preorder is *total* if and only if any two objects in the preorder are comparable:

$$(x \preceq y) \vee (y \preceq x) \quad (totality) \tag{B.6}$$

Given a preorder $\preceq$, one can define a relation $\simeq$ as:

$$x \simeq y \quad \triangleq \quad (x \preceq y) \wedge (y \preceq x) \tag{B.7}$$

and the relation $\simeq$ is then an *equivalence relation*, namely a relation that satisfies:

$$x \simeq x \quad (reflexivity) \tag{B.8}$$

$$(x \simeq y) \Rightarrow (y \simeq x) \quad (symmetry) \tag{B.9}$$

$$(x \simeq y) \wedge (y \simeq z) \Rightarrow (x \simeq z) \quad (transitivity) \tag{B.10}$$

An equivalence relation partitions everything into a collection of mutually disjoint sets called equivalence classes, such that every object in the domain of the equivalence relation belongs to exactly one of the equivalence classes: x and y belong to the same equivalence class just in case $x \simeq y$. The equivalence relation $\simeq$ defines a *linear order* or *total order* $\preceq_{equiv}$ on the equivalence classes: let $[x]$ be the equivalence class of x. Then

$$[x] \preceq_{equiv} [x] \quad (reflexivity) \tag{B.11}$$

$$([x] \preceq_{equiv} [y]) \wedge ([y] \preceq_{equiv} [x]) \Rightarrow ([x] = [y]) \quad (antisymmetry) \tag{B.12}$$

$$([x] \preceq_{equiv} [y]) \wedge ([y] \preceq_{equiv} [z]) \Rightarrow ([x] \preceq_{equiv} [z]) \quad (transitivity) \tag{B.13}$$

$$([x] \preceq_{equiv} [y]) \vee ([y] \preceq_{equiv} [x]) \quad (totality) \tag{B.14}$$

That means that any two worlds can be compared in terms of their similarity to world W; either the one or the other is more similar, or they are both equally similar.

Finally, it should be noted that an additional requirement on $\preceq_W$, besides that it is a total preorder, is that W is more similar to itself than any other world which is similar to it:

$$X \preceq_W X \Rightarrow W \preceq_W X \tag{B.15}$$

A mathematical way of phrasing the semantics of $\Box\!\!\rightarrow$ [Lew73, p16] is:

- $A \Box\!\!\rightarrow B$ is true at world W if and only if either
  1. A is true at no world accessible from W at any distance; or
  2. A is true at some world accessible from W, and $A \Rightarrow B$ is true at all worlds in $\{X | X \preceq_W A\}$.

B.9 Rules for Counterfactual Conditionals

It follows from the semantics for $\Box\!\!\rightarrow$ that all the inference rules for logic still hold if all the formulas are prefaced by '$A \Box\!\!\rightarrow$': that is, for example

$$\text{(\&-intro)}\quad \frac{S \;\Box\!\!\rightarrow\; A \qquad S \;\Box\!\!\rightarrow\; B}{S \;\Box\!\!\rightarrow\; A \wedge B}$$

$$\text{(\&-elim)}\quad \frac{S \;\Box\!\!\rightarrow\; A \wedge B}{S \;\Box\!\!\rightarrow\; A} \qquad \frac{S \;\Box\!\!\rightarrow\; A \wedge B}{S \;\Box\!\!\rightarrow\; B}$$

$$\text{(}\vee\text{-intro)}\quad \frac{S \;\Box\!\!\rightarrow\; A}{S \;\Box\!\!\rightarrow\; A \vee B} \qquad \frac{S \;\Box\!\!\rightarrow\; B}{S \;\Box\!\!\rightarrow\; A \vee B}$$

$$\text{(}\vee\text{-elim)}\quad \frac{S \;\Box\!\!\rightarrow\; A \wedge B \qquad [S \;\Box\!\!\rightarrow\; A] \qquad [S \;\Box\!\!\rightarrow\; B]}{S \;\Box\!\!\rightarrow\; C \qquad S \;\Box\!\!\rightarrow\; C}{S \;\Box\!\!\rightarrow\; C}$$

$$\text{(}\Rightarrow\text{-intro)}\quad \frac{[S \;\Box\!\!\rightarrow\; A]}{S \;\Box\!\!\rightarrow\; B}{S \;\Box\!\!\rightarrow\; (A \Rightarrow B)}$$

$$\text{(}\Rightarrow\text{-elim)}\quad \frac{S \;\Box\!\!\rightarrow\; A \qquad S \;\Box\!\!\rightarrow\; A \Rightarrow B}{S \;\Box\!\!\rightarrow\; B}$$

These should be derived rules from a reasonable axiomatisation of the counterfactual conditional. Lewis axiomatises his preferred logic of counterfactuals, VC, as follows [Lew73, p132]. There are two rules:

$$\text{Modus Ponens} \tag{B.16}$$

$$\frac{(A_1 \wedge \ldots \wedge A_n) \Rightarrow B}{((C \;\Box\!\!\rightarrow\; A_1) \wedge \ldots \wedge (C \;\Box\!\!\rightarrow\; A_n)) \;\Rightarrow\; (C \;\Box\!\!\rightarrow\; B)} \tag{B.17}$$

in which the second Rule B.17 is actually a rule schema, one rule for each value of $n > 1$. The axioms are:

Axiom 8 *Truth-functional tautologies*

Axiom 9 *Definitions of the non-primitive related operators (which we don't use here)*

Axiom 10 $A \;\Box\!\!\rightarrow\; A$

Axiom 11 $(\neg A \;\Box\!\!\rightarrow\; A) \;\Rightarrow\; (B \;\Box\!\!\rightarrow\; A)$

Axiom 12 $(A \;\Box\!\!\rightarrow\; \neg B) \;\vee\; (((A \vee B) \;\Box\!\!\rightarrow\; C) \equiv (A \;\Box\!\!\rightarrow\; (B \Rightarrow C)))$

Axiom 13 $(A \;\Box\!\!\rightarrow\; B) \;\Rightarrow\; (A \Rightarrow B)$

Axiom 14 $(A \wedge B) \;\Rightarrow\; (A \;\Box\!\!\rightarrow\; B)$

B.10 Defining the Other Modalities from the Counterfactual

From a counterfactual satisfying VC (or the Lewis semantics in general), one may define a modal operator

$$\Box_C A \;\triangleq\; (\neg A) \;\Box\!\!\rightarrow\; A \tag{B.18}$$

called by Lewis the *outer modality*. If the outer modality satisfies in addition the rule ($S5$), then the outer modality forms the S5 modal logic. The ($S5$) axiom for the outer modality corresponds to the following axiom for $\Box\!\!\rightarrow$:

Axiom 15 $\neg(A \;\Box\!\!\rightarrow\; \neg A) \;\Rightarrow\; ((A \;\Box\!\!\rightarrow\; \neg A) \;\Box\!\!\rightarrow\; \neg(A \;\Box\!\!\rightarrow\; \neg A))$

We call the logic VC with the additional Axiom 15 VCU (after [Lew73, p121]).

This means that if we take a *possible world* to be a model of TLA (a *behavior*, a sequence of classical models ordered like the non-negative integers), in which each pertinent state variable is a assigned a definite set-valued value), and the set of possible worlds to satisfy the Kripke semantics (that is, that there is a binary relation of *accessibility* between possible worlds), and furthermore the worlds accessible from any given world W are totally-preordered by *nearness* to W, then we may axiomatise $\Box\!\!\rightarrow$ as above (including Axiom 15), define the S5 alethic modality $\Box_C$ from $\Box\!\!\rightarrow$ using Definition B.18, (that is, $\Box_A \triangleq \Box_C$), define $\succ$ from $\Box_C$ using Definition B.1, and define the deontic modality O from $\Box_C$ using Definition B.2.

We may define the operator $\Rightarrow\!\!\!\!\succ$ from $\Box\!\!\rightarrow$:

Axiom 16 $A \Rightarrow\!\!\!\!\succ B \triangleq (A \wedge B \wedge (\neg A \Box\!\!\rightarrow \neg B))$

The transitive closure $\Rightarrow\!\!\!\!\succ^*$ is not definable in a first-order way from the relation of which it is the transitive closure, $\Rightarrow\!\!\!\!\succ$, as is well-known. Thus the axioms giving the recursive definition of $\Rightarrow\!\!\!\!\succ^*$ remain in EL. (This is not to claim that $\Rightarrow\!\!\!\!\succ^*$ is not definable in VCU-EL, but rather ignorance as to whether it is.)

B.11 The Well-Formed Formulas of EL

We define the *well-formed formulas* of EL to be those formulas of VCU, with the extra defined primitives as in Section sec:definition-modalities, plus $\Rightarrow\!\!\!\!\succ^*$, and with formulas of TLA substituted for the propositional primitives. That is, the primitive formulas are those of TLA, and the well-formed formulas of EL are the formulas constructed from these by the syntactic rules of VCU with the addition of the defined operator symbols and the symbol $\Rightarrow\!\!\!\!\succ^*$. This ensures that TLA-tense-operators may only occur *strictly within the scope of* the other modalities.

This syntax definition makes semantic sense. *Possible worlds* in EL are those ways the world might have been. We have construed 'the world' as a temporal succession of states and events, namely those states and events that happened during the incident, in addition to a lot of others. So 'the world' is a model for TLA, a TLA *behavior*. Alternative worlds are alternative behaviors. Thus a *possible world* in the Kripke sense for the counterfactual logic VC (and thus for VCU) consists of a TLA behavior, which is a model for TLA formulas. The VCU logic makes assertions concerning propositions true or false in these possible worlds. Thus the VCU logic is syntactically built from 'propositional primitives' which describe propositions true or false in these possible worlds, i.e. TLA formulas.

Hence the definition of well-formed formulas for EL follows directly from the requirement that a *possible world* for the $\Box\!\!\rightarrow$ logic consists of a TLA behavior.

B.12 Soundness and Completeness of a fragment of EL

Let us denote by VCU-EL the fragment of EL axiomatised by VCU, with the defined operators (and therefore without $\Rightarrow\!\!\!\!\succ^*$), on top of the TLA primitives, plus the TLA axioms and rules.

The binary relation of *accessibility*, needed for the Kripke semantics on possible worlds, is definable from the ternary relation *nearness* as follows. Let accessibility by $accessible(W, X)$. Then

$$accessible(W, X) \triangleq X \preceq_W X \tag{B.19}$$

so all one needs for the semantics is a ternary relation

$$as{-}near(W, X, Y) \triangleq X \preceq_W Y$$

whose projections on the first argument W, namely $\{\langle X, Y\rangle | X \preceq_W Y\}$, are total preorders with least element W (W must be least, according to Axiom B.15), that satisfy the *Uniformity Condition*

Axiom 17 *For any W and Z, $\{Y | accessible(W, Y)\} = \{Y | accesible(Z, Y)\}$*

The Uniformity Condition is the semantic counterpart of Axiom 15: VC-models that satisfy the Uniformity Condition are VCU-models, and VCU-models satisfy the Uniformity Condition [Lew73, pp120-121]. Axiom 15 is said to be the *characteristic axiom* for the Uniformity Condition.

If one defines all the modal operators from primitive $\Box\!\!\rightarrow$, then the resulting logic must be a conservative extension of VCU, and thus its soundness follows from the soundness of VCU, as shown in [Lew73, pp122,124,133]. Its completeness follows from the fact that Axiom 15 is the characteristic axiom for the Uniformity Condition.

However, since the propositional primitives of VCU-EL are TLA formulas, this argues for the soundness and completeness of VCU-EL only relative to TLA.

TLA is complete for proof of formulas that occur during hierachical verifications in TLA (namely, of the form $TLA.Spec_1 \Rightarrow TLA.Spec_2$), and it is sound, assuming that ZF set theory is sound [Lam94].

Thus follows the relative soundness and completeness of VCU-EL from that of VCU and TLA. EL itself is not complete, because the intended interpretation of the relation $\twoheadrightarrow^*$ is as the transitive closure of $\twoheadrightarrow$, whereas many other relations, such as the universal relation on nodes and their Boolean combinations, also satisfy Axioms 2 and 3. That this relation, which exists in every model of VCU-EL, satisfies Axioms 2 and 3 shows that EL is sound.

However, EL itself contains some extra rules which are not necessarily derived rules of VCU-EL, although many are. So the soundness and completeness of EL does not necessarily follow from that of $VCU - EL$.

B.13 Special EL Rules

Although we could define the relation $\hookrightarrow$ using the tense-logical rules, we don't need to and haven't. The EL rules as expressed here clearly do not form a non-redundant set, since some of them follow from others, as noted in the text. Furthermore, many of them are derived rules if non-temporal modalities of EL are defined from $\Box\!\!\rightarrow$, as in Section B.9. They are listed in Figure B.1

Of these,

- Axiom 1, Axiom 2, Axiom 3, Rule 2.2, Rule 2, and Rule 3 are not derived rules of VCU-EL, because the operator $\twoheadrightarrow^*$ does not appear in VCU-EL; neither could it be reasonably defined in EL as far as we can see;

- Rule 2.5, Axiom 4, Rule 2.7, Rule 2.8, Rule 5 and Rule C.4 are derived rules of VCU-EL;

- Rule 2.3 and Rule 2.4 have syntactically restricted components ('*Procedures*', '*Event*') devised by the intent and judgement of the analyst, and therefore cannot be derived rules of VCU-EL. Rule 2.4 is, however, obviously derivable from Rule 2.3. Rule 2.3 itself is not derived, when *Procedures* and *Event* are

Axiom 1 $\vdash (A \Rrightarrow^* B) \Rightarrow (A \hookrightarrow B)$

Axiom 2 $\vdash (A \Rrightarrow B) \Rightarrow (A \Rrightarrow^* B)$

Axiom 3 $\vdash (A \Rrightarrow^* B) \wedge (B \Rrightarrow^* C) \Rightarrow (A \Rrightarrow^* C)$

Rule 2.2 $$\frac{A \Rrightarrow^* B}{A \hookrightarrow B}$$

Rule 2 $$\frac{A \Rrightarrow B}{A \Rrightarrow^* B}$$

Rule 3 $$\frac{\begin{array}{l} A \Rrightarrow^* B \\ B \Rrightarrow^* C \end{array}}{A \Rrightarrow^* C}$$

Rule 2.5 $$\frac{\begin{array}{l} A \mathrel{\Box\!\!\rightarrow} B \\ \neg A \mathrel{\Box\!\!\rightarrow} \neg B \end{array}}{A \Rrightarrow B}$$

Axiom 4 $\vdash ((A \mathrel{\Box\!\!\rightarrow} B) \wedge A) \Rightarrow B$

Rule 2.7 $$\frac{A}{(A \mathrel{\Box\!\!\rightarrow} B) \equiv (A \wedge B)}$$

Rule 2.8 $$\frac{\begin{array}{l} A \wedge B \\ \neg A \mathrel{\Box\!\!\rightarrow} \neg B \end{array}}{A \Rrightarrow B}$$

Rule 5 $$\frac{(\vdash_{TLA} A \Rightarrow B)}{A \succ B}$$

Rule 2.3 $$\frac{(Hypotheses \wedge \Box Procedures) \succ \Diamond Event}{(Hypotheses \wedge O(Procedures)) \Rightarrow O(\Diamond Event)}$$

Rule 2.4 $$\frac{\begin{array}{l} Hypotheses \\ (Hypotheses \wedge \Box Procedures) \succ \Diamond Event \end{array}}{O(\Diamond Event)}$$

Rule C.4 $$\frac{\begin{array}{l} C \\ \neg C \ \mathrel{\Box\!\!\rightarrow}\ \neg B \\ \neg B \ \mathrel{\Box\!\!\rightarrow}\ \neg C \end{array}}{C \ \Box\!\!\Rightarrow \ B}$$

Rule C.6 $$\frac{\begin{array}{l} Hypotheses \\ Procedures \\ (Hypotheses \wedge \Box Procedures) \ \succ \ \Diamond Event \\ \Diamond Event \end{array}}{(Hypotheses \wedge Procedures) \ \Box\!\!\Rightarrow \ \Diamond Event}$$

Figure B.1: The Rules of EL

propositional variables, and $O(P)$ is defined as $\Box(\neg P \Rightarrow V)$ - here is a simple argument. The consequent of the rule is equivalent to

$$\Box((\mathit{Hypotheses} \wedge \Box(P \vee V)) \Rightarrow (\Diamond E \vee V))$$

Suppose we consider an instance in which $\mathit{Hypotheses} \equiv \top$, i.e.,

$$\Box(\Box(P \vee V) \Rightarrow (\Diamond E \vee V))$$

then this is equivalent to

$$\Box\Box(P \vee V) \Rightarrow \Box(\Diamond E \vee V))$$

which in turn is equivalent in S5 to

$$\Box(P \vee V) \Rightarrow \Box(\Diamond E \vee V))$$

The antecedent of the rule is equivalent to

$$\Box P \Rightarrow \Box\Diamond E$$

Suppose now we take $P \equiv \neg V$: then the antecedent is

$$\Box\neg V \Rightarrow \Box\Diamond E$$

and the consequent is

$$\Box(\Diamond E \vee V))$$

A set of possible worlds in which there is a violation in some world, but in which $\Diamond E$ doesn't hold in this world, satisfies the antecedent but not the consequent. Therefore when P and E are not suitably restricted, this rule is unsound. So in particular it is not derivable from VCU-EL.

- For similar considerations as for Rule 2.4, Rule C.6 is not a derived rule of VCU-EL.

This completes the definition of EL.

B.14 Axioms and Processes for WBA

Certain special axioms must be included for each WBA analysis, namely that (human) procedures and (machine) specifications are deontic axioms. These are properly regarded not as axioms of EL itself, since various procedures or machine specifications could be contradictory if care is not taken with scoping the variables used. In any particular application of EL to perform a WBA, the following axioms must be added, for each set of procedures or system specifications:

$$\vdash O(\mathit{Procedures})$$

$$\vdash O(\mathit{Spec})$$

Similarly, there are two 'meta-axioms' that are more properly processes to be followed by the analyst in constructing a WBA, and are not themselves part of the EL logic. These are

Explicitly add to the history those states $\langle\neg E\rangle$ in which E is an event, $O(E)$ is derivable, and E does not occur.

For any B which has a causal factor (that is, for which there is an A such that $A \Rrightarrow B$ is established), the set $\{A|A \Rrightarrow B\}$ must be a satisfactory set of causal factors for B.

Appendix C

Sufficient Causal Explanation

(This is a chapter of [LL98b].)

We may have found some causal factors for [1], namely [11] and [12], but how do we know that [11] and [12] suffice to provide a causal explanation of [1]? How do we know there isn't some essential [13] that must also be present? A causal explanation doesn't just search after *some* causal factors, it also searches after a *sufficient* explanation.

Suppose we have determined that $A_1 \Rightarrow B, A_2 \Rightarrow B, \ldots, A_n \Rightarrow B$. We shall say that $A_1 \ldots A_n$ form a *sufficient set* of causal factors for B just in case it would be impossible for B not to happen (not to be the case) if $A_1 \ldots A_n$ had happened. In our situation, this is a counterfactual conditional, because we are performing a failure analysis of an actual incident, and so $A_1, \ldots, A_n, B$ did in fact all happen (were all the case).

C.1 The Causal Sufficiency Condition

These considerations lead to the *[Causal Sufficiency Criterion]* (CSC): $A_1, \ldots, A_n$ form a sufficient set of causal factors for B (equivalently, $A_1 \wedge A_2 \wedge \ldots \wedge A_n$ is a sufficient cause for B) if and only if

$$\neg B \ \Box\!\!\rightarrow \neg(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \tag{C.1}$$

which leads to the

Causal Completeness Test (CCT): $A_1, \ldots, A_n$ form a *satisfactory set* of causal factors for B if and only if

$$\begin{array}{l} \wedge\ A_1 \Rightarrow B \\ \wedge\ A_2 \Rightarrow B \\ \wedge\ \ldots \\ \wedge\ A_n \Rightarrow B \\ \wedge\ \neg B \ \Box\!\!\rightarrow \neg(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \end{array} \tag{C.2}$$

This in turn yields another meta-axiom for causal histories:

Axiom 18 *MetaAxiom: For any B which has a causal factor (that is, for which there is an A such that $A \Rightarrow B$ is established), the set $\{A | A \Rightarrow B\}$ must be a satisfactory set of causal factors for B.*

C.2 The CCT for [11] and [12]

So is it the case that [11] and [12] form a sufficient set for [1]? We argue informally, because we have as yet not introduced a method for analysing the formal semantics of sentences containing intensional contexts, such as '*realizes that*' or '*believes that*' or *judges that*'. It follows from [11] and the properties of the intensional context 'realize' that if one realizes that P, then P must be true (else one is, say, merely 'imagining that P'). So we imagine we should be able to prove

$$\vdash [11] \rightarrow \textit{AC is in the course of landing on a particular runway}$$

and this runway happens to be BRU RWY 25. Suppose that in the nearest worlds in which the AC had not landed on this runway (this is the $\neg B$ hypothesis, with $B = [1]$), then it could not have been the case both that the AC was landing and the CRW opted to continue: either the AC was not in landing configuration and process (e.g., someone had inadvertently triggered the TOGA, *take-off/go-around* autopilot switch) or the CRW had not opted to continue (i.e., they opted to go around, this being according to procedure the only available other option - let us presume that opting to do something does not necessarily mean a conscious explicit decision was taken: one can opt to continue simply by continuing). Since it could not have been the case that (AC was landing and the CRW opted to continue), it could not have been the case that $[11] \wedge [12]$ (by contraposition, since this implies the former). Thus the counterfactual is established and the CSC is fulfilled. Since we already know $[11] \nRightarrow [1]$ and $[12] \nRightarrow [1]$, the CCT is fulfilled.

In order to make this argument, we have not only relied on particular properties of the intentional context '*realize that*', we have also implicitly relied on particular properties of referential expressions. For although the AC was landing on BRU RWY 25, the CRW may not have realised *this* fact in its entirety, namely that this runway was BRU RWY 25. It suffices that they realized that they were landing on a runway at the 'wrong' airport, and that it follows from the semantics of 'realize' that the AC was indeed landing on whichever runway it was landing on. It also is the case that the runway the aircraft was landing on is in fact identical with BRU RWY 25. But this identity has to be asserted outside the intentional context – intentional truths are not necessarily preserved under syntactical substitution of equireferential expressions (in fact, the definition of '*intentional context*' is roughly speaking that there are two equireferential expressions that yield different truth values when placed in this context). But these are subtleties of semantics that are well handled for our purposes by philosophical logic, and it strays from our purpose to do more than remark them here. The moral is, however, that arguments for counterfactual conditionals, whether for $\nRightarrow$ or for the CCT, may depend on logical analyses of the node descriptions.

In general, it is not the case that satisfactory sets for a given B are unique. They depend on how one wants to analyse the situation; how much one wants to say, what depth and detail of analysis one wants to pursue, the limitations of the language chosen to express the nodes. All of this is very much the choice of the investigator. For example, in our incident, we regard it as sufficient that [11] and [12], whereas someone with great interest in the psychology and consequences of flight-deck decision making may wish to analyse these circumstances further into a causally-explanatory chain using, say a PARDIA (perception-attention-reasoning-decision-intention-action) model such as we have used.

A similar situation exists in pure axiomatic mathematics. One is provided with proof rules sufficient to get any job done, but what proofs are constructed and how are up to the individual wishes and skill of the user. Proofs may be more detailed or less detailed, easy-to-follow or cleverly slick, pro forma or creative. Yet

the criteria for being a valid proof remain constant throughout the enterprise. So with WBA. We have no wish to regulate whether an analysis is most subtle, or how it indicates what future steps to take to prevent recurrences, or whether it must use the latest theory of human-machine interaction. We wish to lay out criteria and reasoning rules for providing a formally-complete causal explanation, according to assumptions that an analyst makes in a particular case. We thereby make the assumptions clear, explicit and precise, exhibit their role in the explanation, and make the reasoning clear. This is a worthy goal, just as a formal verification of an algorithm or a safety-critical design is worthy in so far as it gives us certainty that the system design fulfils its purpose. The issue of whether the algorithm or design is somehow the best or the most efficient is not addressed by the issue of verification. Likewise, we don't address such individual properties of causal explanation.

C.3 Inference Rules for the CCT

One can provide inference rules that ensure that the CCT is fulfilled. We introduce the relation $\Box\!\!\Rightarrow$ to stand for 'necessary and sufficient cause'. Thus the goal of the CCT is to show

$$A_1 \ \wedge \ A_2 \wedge \ \ldots \ A_n \ \Box\!\!\Rightarrow \ B \tag{C.3}$$

We therefore introduce the rule

$$\frac{\begin{array}{l} C \\ \neg C \ \Box\!\!\rightarrow \ \neg B \\ \neg B \ \Box\!\!\rightarrow \ \neg C \end{array}}{C \ \Box\!\!\Rightarrow \ B} \tag{C.4}$$

and the rule

$$\frac{\begin{array}{r} A \ \Box\!\!\rightarrow \ C \\ B \ \Box\!\!\rightarrow \ C \end{array}}{(A \vee B) \ \Box\!\!\rightarrow \ C} \tag{C.5}$$

and claim

Theorem 1 *Rules C.4 and C.5 suffice to establish the CCT when it is provable.*

The proof of the theorem may be found in Appendix 4.2.3.

C.4 Causal Sufficiency Through Procedural Necessity

There is also another way in which causal sufficiency can be explained. Suppose we are able to derive that

$$\Box \mathit{Procedures} \ \succ \ \Diamond \mathit{Event}$$

that is, that the occurrence of *Event* follows necessarily upon the implementation of the *Procedures*. Then it is a sufficient explanation why the *Event* occurred simply that the *Procedures* were followed.

For example, if we have implemented *Procedures* as a state machine, and this state machine is in state S, and through action A this state machine attains state S_1, then it is a causally sufficient explanation why the machine is in state S_1 that

- the specification of the state machine was followed; and
- the machine was shortly before in state S; and

- the action A then occurred.

This is to say the following, formally. Suppose we can demonstrate that from the *Procedures* and *Hypotheses* (former state, plus 'environmental' conditions being in order, plus action A being made) that the *Event* must occur:

$$\vdash_{TLA} Hypotheses \wedge \Box Procedures \;\Rightarrow\; \Diamond Event$$

then we can conclude that

$$(Hypotheses \wedge \Box Procedures) \;\succ\; \Diamond Event$$

by Rule 5. And we further suppose that the *Hypotheses* and *Procedures* were followed: then *Event* must occur, and *Hypotheses* and *Procedures* are a sufficient explanation for it:

$$\frac{\begin{array}{l} Hypotheses \\ Procedures \\ (Hypotheses \wedge \Box Procedures) \;\succ\; \Diamond Event \\ \Diamond Event \end{array}}{(Hypotheses \wedge Procedures) \;\Box\!\Rightarrow\; \Diamond Event} \tag{C.6}$$

The following consideration speaks against acceptance of this rule. Suppose we're grilling sausages. If we leave the sausages under the grill, and the grill functions correctly, they will eventually cook. We put sausages under the grill, and turn the grill on. Unbeknownst to us, someone took the sausages from the grill and put them in the microwave for two minutes, and then back under the grill. Lo and behold, the sausages are cooked. It is a sufficient explanation for the fact that the sausages were cooked that they were under the grill and the grill was on and functioning correctly, but it is not the explanation *in this case* for why the sausages cooked.

This is a standard problem in the philosophy of science that we cannot hope to solve in so short a space. However, we will endeavor to indicate why this is no counter to our treatment using Rule C.6. First, it *is* a sufficient explanation for the cooking of the sausages that they were under the grill and the grill functioned. If it is only necessary that the sausages were cooked in order for this to be a causal factor explaining other events (for example, that we were all sufficiently nourished and the party was a success), then it must be that the actual turn of things – that someone used the microwave to speed the process – is irrelevant to the explanation of the final event (that the party was a success). However, if the party was a success only because the sausages were cooked *in time* then this timely cooking is part of what is to be explained – and if they had been left under the grill alone, they would not have been cooked in time. Therefore we would not be able to use Rule C.6 in order to conclude that it suffices as an explanation to note that the sausages were under the grill and the grill functioned correctly – because one of the hypotheses is false, namely that

> if the sausages are under the grill and the grill functions correctly, then the sausages are cooked *in this short time*

This shows the importance of a criterion called by Mill the *Method of Difference*, by Lipton the *Difference Condition*, [and] endorsed by Lewis [...], for determining if one has sufficient richness in the vocabulary to make a proper causal explanation. This richness must be present to determine the existence of deciding events. We're claiming in the present case that the time(liness) of cooking is a significant explanatory factor, and therefore that it must be somehow included in the vocabulary in

which we describe the events. But how can we determine such a need for detail? By using the Difference Condition as follows.

Consider the coarse description that *the sausages were cooked*. There are two outcomes consistent with this coarser description : the party was a success and the party was not a success. What distinguishes these two outcomes? The Difference Condition says there is a causal factor that distinguishes them. We infer (by possible worlds reasoning as before) that the timeliness of the cooking is such a distinguishing factor. We must therefore include it. Once included, the timeliness of the cooking, not just the cooking itself, must be explained, and this feature serves as above to disallow the correctness of the hypothesis that the correct functioning of the grill (alone) with the sausages could have brought about the timely cooking. Therefore the hypothesis, that from the operation of the grill (alone) the sausages were cooked in a timely fashion, is false and therefore the simple application of Rule C.6 does not suffice to establish grounds for the sausages being cooked in a timely fashion.

Although expressed in words, this reasoning has direct bearing on the description of procedures in a formal language such as TLA+, which we shall demonstrate later. A description in TLA+ of procedures can be 'higher-level' or 'lower-level', abstract or detailed. The level of detail of the procedures to be described will depend upon the description of the event whose occurrence these procedures explain. TLA+ has the ability to talk about timeliness of procedures to the level of detail required to do physics (so, a real-time component).

We propose that the Difference Condition forces a suitably-detailed description of events that the misapplication of Rule C.6 through a too-coarse description of causal factors does not happen. It would, however, weaken the case we are making for Rule C.6 if the only reason for application of the Difference Condition were to justify this rule. We therefore seek (and find!) wider need for the Difference Condition, which we now describe.

Appendix D

Glossary

GLOSSARY:
=========

AD : Airworthiness Directive
ADC : Air Data Computer
AFS : Automatic Flight System
ALT : Altitude
ALT SEL : Altitude Selector
AOA : Angle of Attack
AP : Auto-Pilot
APU : Auxiliary Power Unit
A/THR : Automatic Thrust
AT : Auto Throttle
ATC : Air Traffic Control
ATCC : Air Traffic Control Center
ATS : Auto-Throttle System
ATT : Attitude
BATC : Brussels Air Traffic Control
BEA : Bureau Enqu^etes Accidents
BKN : Broken
CAP : Captain
CAS : Computed Airspeed
CGCC : Center of Gravity Control Computer
CAT : Category
CMD : Command
CN : Consigne de Navigabilite
CRW : Crew
CVR : Cockpit Voice Recorder
CWS : Control Wheel Steering
DFDR : Digital Flight Data Recorder
DGAC : Direction G^en^erale de l' Aviation Civile
DH : Decision Height
ECAM : Electronic Centralized Aircraft Monitoring
BFCU : Electronic Flight Control Unit
EFIS : Electronic Flight Instrument System
ENG : Engine
EPR : Engine Pressure Ratio
FAA : Federal Aviation Administration
FAC : Flight Augmentation Computer
FADEC : Full Authority Digital Electronic Control
FCC : Flight Control Computer
FCOM : Flight Crew Operating Manual
FCU : Flight Control Unit
FD : Flight Director
FI : Flight Information (Flight data and Flight Plan)
FIDC : Fault Isolation and Detection Computer
FIDS : Fault Isolation and Detection System
FL : Flight Level
FMA : Flight Mode Annunciator
FMC : Flight Management Computer
FMS : Flight Management System
F/O : First Officer
FMC : Flight Warning Computer
GA : GO AROUND
GCU : Generator Control Unit
GPWC : Ground Proximity Warning Computer
GPWS : Ground Proximity Warning System

```
GS       : Glide Slope
HDG      : Heading
HDG/SEL  : Heading Selector
HPC      : High Pressure Compressor
HPT      : High Pressure Turbine
ICAO     : International Civil Aviation Organization
IGS      : Instrument Guidance System
IGV      : Inlet Guide Vane
IND      : Indicator
ILS      : Instrument Landing System
IRS      : Inertial Reference System
IRU      : Inertial Reference Unit
LAND     : Landing
LATC     : London Air Traffic Control
L/D      : Landing
LIG      : Landing Gear
LOC      : Localizer
LPC      : Low Pressure Compressor
LPT      : Low Pressure Turbine
LVL/CH   : Level Change
MAC      : Mean Aerodynamic Chord
MAN THR  : Manual Thrust
MDA      : Minimum Decision Altitude
MIC      : Microphone
MTP      : Maintenance and Test Panel
NAV      : Navigation
NPA      : Non Precision Approach
NTSB     : National Transportation Safety Board
OOT      : Out Of Trim
OVC      : Overcast
PCM      : Pulse Code Modulation
PF       : Pilot Flying
PFD      : Primary Flight Display
PIC      : Pilot in Command
PNF      : Pilot Not Flying
QNH      : Pressure Setting to Indicate Elevation above Mean Sea Level
R ALT    : Radio Altitude
RET      : Retract
RMI      : Radio Magnetic Indicator
RWY      : Runway
SATC     : Shannon Air Traffic Control
SB       : Service Bulletin
SCT      : Scattered
SGU      : Symbol Generator Unit
SOP      : Standard Operating Procedure
SPD      : Speed
SPD/MAC  : Speed/Mach
SRS      : Speed Reference System
SW       : Switch
TCC      : Thrust Control Computer
TCD      : Ministry of Transport Civil Aviation Bureau Directive
TFC      : Traffic
THR      : Thrust
THR L    : Thrust Latch
THS      : Trimmable Horizontal Stabilizer
TIPS     : Technical Instruction Processing Sheet
TRP      : Thrust Rating Panel
VAPP     : Approach Target Speed
VOR      : VHF Omnidirectional Radio Range
V/S      : Vertical Speed
Vs       : Stall Speed
VTG      : Target Speed
W.STA    : Wing Station
```

Bibliography

[Adma] U.S. Federal Aviation Administration. Airman's information manual. Published annually in full, updated bimonthly.

[Admb] U.S. Federal Aviation Administration. Airman's information manual, pilot/controller glossary. Published annually in full, updated bimonthly.

[And58] A. R. Anderson. A reduction of deontic logic to alethic modal logic. *Mind*, 67:100–103, 1958.

[Aus75] J.L. Austin. *How to do things with words, 2nd ed.* Oxford University Press, 1975.

[Ble98] George A. Bleyle. Personal Conversations, 1997,1998.

[Blu97] Michael Blume. Kleine Einführung in die Spezifikationssprache TLA+. Tutorial, 1997.

[Com96] Aircraft Accident Investigation Comission. *Aircraft Accident Investigation Report 96-5, China Airlines, Airbus Industrie A300B4-622R, B1816, Nagoya Airport, April 26, 1994.* Ministry of Transport, Japan, 1996.

[Deu97] Statistisches Bundesamt Deutschland. Verkehrsstatistik 1993-1996. `http://www.statistik-bund.de/basis/d/bd19_t03.htm`, 1997.

[Dör97] Friedrich Dörner. *Die Logik des Mißlingens.* Rowohlt Taschenbuch Verlag GmbH, Reinbek bei Hamburg, 1997.

[dWL95] H. de Wulf and D. Learmount. DC-10 misses Frankfurt runway – by 300km. *FLIGHT INTERNATIONAL*, 148(4493):8, October 1995.

[EGK97] R. Evans, G. Gazdar, and B. Keller. The DATR Lexical Knowledge Representation Language. Homepage:`http://www.cogs.susx.ac.uk/lab/nlp/datr/datr.html`, 1997.

[EKN98] J. Ellson, E. Koutsofios, and S. North. graphviz – tools for viewing and interacting with graph diagrams. accessible from `http://www.research.att.com/sw/tools/graphviz/`, 1998.

[Ger97] T. Gerdsmeier. A tool for building and analysing WB-Graphs. Technical Report RVS-RR-97-02, RVS Group, Faculty of Technology, University of Bielefeld, Feb 1997. Available at [RVS].

[GLL97a] T. Gerdsmeier, P. Ladkin, and K. Loer. Analysing the Cali Accident With a WB-Graph. In *Workshop on Human Error and Systems Development, 19th-22nd March 1997, Glasgow University, Scotland, GAAG TR-97-2*, pages 2–15, 1997.

[GLL97b] T. Gerdsmeier, P. Ladkin, and K. Loer. Formalising Failure Analysis. Technical Report RVS-Occ-97-06, Networks and distributed Systems Group, Faculty of Technology, Bielefeld University, Bielefeld, Germany, 1997.

[HC96] G.E. Hughes and M. Cresswell. *A New Introduction to Modal Logic*. Routledge, 11 New Fetter Lane, London EC4P 4EE, 1996.

[HL97] M. Höhl and P. Ladkin. Analysing the 1993 Warsaw accident with a WB-Graph. Technical Report RVS-Occ-97-09, Networks and distributed Systems Group, Faculty of Technology, Bielefeld University, Bielefeld, Germany, 1997.

[Höh98] Michael Höhl. wb2dot: A Tool for the Conversion of Textgraphs to dot-source. `http://www.rvs.uni-bielefeld.de/~mhoehl`, 1998.

[Joh95] C.W. Johnson. The application of Petri nets to represent and reason about human factors problems during accident analyses. pages 93–112, 1995.

[Joh97] C.W. Johnson. The epistemics of accidents. *International Journal of Human-Computer Studies*, (47):659–688, 1997.

[Lad95a] P. Ladkin. Fly NorthWest Airlines to unknown destinations. *The Risk Digest*, 17(38), October 1995. http://catless.ncl.ac.uk/Risks/17.38.html#subj1.

[Lad95b] P. Ladkin. Re: Fly NorthWest Airlines to unknown destinations. *The Risk Digest*, 17(40), October 1995. http://catless.ncl.ac.uk/Risks/17.40.html#subj3.

[Lad96a] Peter Ladkin. Explaining Failure with Tense Logic. Technical Report RVS-RR-96-13, Networks and distributed Systems Group, Faculty of Technology, Bielefeld University, Bielefeld, Germany, 1996.

[Lad96b] Peter Ladkin. Some Dubious Theses in the Tense Logic of Accidents. Technical Report RVS-RR-96-14, Networks and distributed Systems Group, Faculty of Technology, Bielefeld University, Bielefeld, Germany, 1996.

[Lad97] Peter Ladkin. Using the Temporal Logic of Actions: A Tutorial on TLA Verification. Technical Report RVS-RR-97-08, Networks and distributed Systems Group, Faculty of Technology, Bielefeld University, Bielefeld, Germany, 1997. Invited Tutorial on TLA, Second International Conference on Temporal Logic, Manchester, England, 14-18 July, 1997.

[Lad98] Peter Ladkin. Personal Conversations, 1997,1998.

[Lam] Leslie Lamport. The Temporal Logic of Actions (TLA) Page. `http://www.research.digital.com/~tla/`.

[Lam86] Leslie Lamport. The mutual exclusion problem: Part i – A theory of interprocess communication. *Journal of the ACM*, 33:313–326, April 1986.

[Lam93a] Leslie Lamport. How to Write a Proof. February 1993. http://www.research.digital.com/SRC/proofs/proofs.html.

[Lam93b] Leslie Lamport. Hybrid Systems in TLA+. April 1993. in Hans Rischel and Ander P Ravn editors, *Hybrid Systems*, Lecture Notes in Computer Science, Berlin 1993, Springer Verlag.

[Lam94] Leslie Lamport. The temporal logic of actions. *ACM Transactions on Programming Languages and Systems*, 16(3):872–923, May 1994. Also available through [Lam].

[Lam95] L. Lamport. TLA in pictures. In *IEEE Transactions in Software Engineering 21(9)*, pages 768–775, 1995.

[Lam96] Leslie Lamport. The Module Structure of TLA+. September 1996. SRC Technical Note 1997-002, available from `ftp://ftp.digital.com/pub/DEC/SRC/technical-notes/SRC-1996-002.html`.

[Lam97] Leslie Lamport. The Operators of TLA+. April 1997. SRC Technical Note 1997-006a, available from `http://gatekeeper.dec.com/pub/DEC/SRC/technical-notes/abstracts/src-tn-1997-006.html`.

[Lev95] Nancy Leveson. *Safeware: System Safety and Computers.* Addison-Wesley Publishing Company Inc., 1995.

[Lew73] David Lewis. *Counterfactuals.* Oxford University Press, Inc., Blackwell, 1973.

[Lew86] David Lewis. *Philosophical papers, Vol.II.* Oxford University Press, Inc., 200 Maddison Avenue, New York, New York 10016, 1986.

[Lip91] Peter Lipton. *Inference to the Best Explanation.* Routledge, 1991.

[LL32] C.I. Lewis and C.H. Langford. *Symbolic Logic.* Dover Publications, New York, 1932.

[LL98a] Peter Ladkin and Karsten Loer. Analysing Aviation Accidents using WB-Analysis – An Application for Multimodal Reasoning. accepted for *AAAI Spring 1998 Symposium on Multimodal Reasoning*, Stanford, USA, March 1998.

[LL98b] Peter Ladkin and Karsten Loer. Why-Because Analysis: The Formal Logic of Failure. RVS-Bk-98-01, in preparation. Preliminary version available from `http://www.rvs.uni-bielefeld.de` soon., Jan 1998.

[LL9X] Peter Ladkin and Karsten Loer. Why-Because Analysis of the Nagoya Accident Report. in preparation, 199X.

[MHCM96] J. Moffett, J. Hall, A. Coombes, and J. McDermid. A Model for a Causal Logic for Requirements Engineering. *Journal of Requirements Engineering*, 1(1):27–46,, March 1996.

[Mil73] John Stuart Mill. *Collected Works Vol. VII - A System of Logic Books I-III.* University of Toronto Press, London: Routledge & Kegan Paul, 1973.

[MWD97] J.J.-Ch. Meyer, R.J. Wieringa, and F.P.M. Dignum. Chap. 2: The Role of Deontic Logic in the Specification of Information Systems. to appear *Logics for Databases and Information Systems*, Jan Chomicki, Gunter Saake (Eds.), Kluwer Academic Publishers, 1997.

[NAS] NASA. Human Factors -Mission Statement. Office of Aeronautics homepage, `http://olias.arc.nasa.gov/mission.html`.

[Nol94] Michael S. Nolan. *Fundamentals of Air Traffic Control.* Wadsworth Publishing Company, Belmont, California 94002, 1994.

[oTRoC96] Aeronautica Civil of The Republic of Colombia. *Aircraft Accident Report: Controlled Flight Into Terrain, American Airlines Flight 965, Boeing 757-223, N651AA, Near Cali, Colombia, December 20, 1995.* Santafe de Bogota, D.C.-Colombia, Sep 1996. Also available at [RVS].

[Per84] C. Perrow. *Normal Accidents - Living with High-Risk Technologies.* Basic Books, Inc., 1984.

[Pet62] C.A. Petri. *Kommunikation mit Automaten.* Bonn: Institut für Instrumentelle Mathematik, Bonn, 1962. Schriften des IIM Nr. 2, 1962.

[PN98] L. C. Paulson and T. Nipkow. Isabelle Homepage. `http://www.cl.cam.ac.uk/Research/HVG/isabelle.html`, 1998.

[Pra65] Dag Prawitz. *Natural Deduction: A Proof-Theoretical Study.* Almqvist and Wiksell, Uppsala, 1965.

[Ran97] Harro Ranter. Aviation safety web pages - statistics. `http://web.inter.nl.net/users/H.Ranter/year-fig.htm`, 1997.

[Rea97] J. Reason. *Human Error.* Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, 1997.

[RVS] RVS. AG Rechnernetze und verteilte Systeme – WWW-Server. `http://www.rvs.uni-bielefeld.de`.

[Sea69] John A. Searle. *Speech Acts.* Cambridge University Press, 1969.

[TJ96] A.J. Telford and C.W. Johnson. Extending The Application Of Formal Methods To Analyse Human Error And System Failure During Accident Investigations. Technical Report TR-1996-6, Department of Computer Science, University of Glasgow, Glasgow, United Kingdom, 1996.