Chapter 21

Procedure Specifications in TLA+

21.1 Real Time Theorems

For some of the definitions in the modules specifying the procedures we need to use statements about the mathematical behaviour of functions. We do not intend to go deeply into this. We need some analysis in TLA+, namely the *MeanValueTheorem* (Figure 21.1). This is a straightforward task as illustrated in [Lam93b]. We apply this to the altitude-time function; accordingly, we need the help of module *RealTime* from [AL94].

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 21.1: Module *RealTimeTheorems*

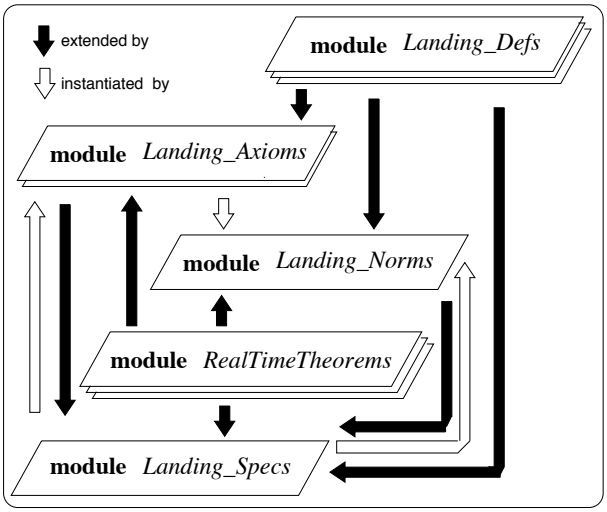

Figure 21.2: Hierarchy of Landing Procedures and Real-Time Specifications

21.2 Procedures

21.2.1 Specification of Landing Procedures

For the specification of landing procedures we use a collection of hierarchical modules, arranged as in Figure 21.2. Abbreviations used in the are explained in Appendix C.

As in Section 16.2, we first define a module containing the basic definitions (Module *Landing_Defs* in Figure 21.3). These definitions concern the parameters *Decision Height* (DH), *Minimum Decision Altitude* (MDA) and *Touch Down Zone Elevation* (TDZE) which are important during the landing phase. Predicates are also defined which explain several states involving altitude.

On a higher level, in Module *Landing_Axioms* we formulate axioms covering several aspects of the landing phase. To avoid cluttering up the module with too much mathematics, we state the assumption *Suitable_Axioms*. This should not affect the quality of the specification, or of the proof. The mathematics we assume is trivial enough to be classified as secure knowledge, and it is possible to handle these axioms formally correctly, as we have noted above regarding the Mean Value Theorem. The module includes definitions from *Landing_Defs* and *RealTimeTheorems*. Axioms are derived from official aviation literature (e.g., [U.Sb, U.Sa]) and expert knowledge (from ourselves and other private and commercial pilots). They define general procedures which are always valid, e.g.:

- *Unique_Approach-Type_Rule*: An approach to landing is either a Instrument Landing System (ILS) approach or a Non-Precision (NP) approach.
- *data_inconsist(x,y)*: Data used during the landing phase is defined to be

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 21.3: Module *Landing_Defs*

inconsistent if (a) institutions or aircrafts are addressed incorrectly; or (b) navigation aids (Navaids) used for the approach are not on the approach plate; or (c) navigation aids appearing on the approach plate cannot be detected during the approach to landing;

- *CompleteApprPlates*: The approach plates are considered to be complete if and only if all Navaids required 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 21.5 and 21.6). Defeasible rules are rules which ought to be adhered to, but – as the proof will show – many of them are not at some point during this incident. Selected rules are:

- *Unique_APT_Rule*: An aircraft can only be cleared to land at an airport whose area it is in;

- *Landing_Criterion*: An aircraft has visual contact with acceptable visibility if either on an ILS-approach it descends below DH or on an NP-approach it reaches or descends below MDA.

- *ILS-APPR_Rule*: An ILS-approach must be broken off if the aircraft is already below DH and the visibility becomes unacceptable[1].

[1] This is an exact translation of [U.Sa], 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 $AC, CRW, APT, This\_Approach$

ASSUMPTIONS

ASSUME

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

AXIOMS

$$APT/ATC\_Id\_Rule \triangleq ((AC)near(APT) \Rightarrow (AC)in\_area(responsible\_ATC[APT]))$$

$$Clearance\_Rule \triangleq ((AC)in\_landing\_phase \Rightarrow landing\_accepted(CRW))$$

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

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

$$data\_inconsist(X, Y) \triangleq (X \neq Y)$$

$$AttendILS{-}Navaids \triangleq ILS\_approach(AC, APT) \Rightarrow navigate(appr\_plate[APT], CRW)$$

$$CompleteApprPlates \triangleq \forall y : \left(\begin{array}{r} y \in Navaids[This\_Approach] \Leftrightarrow \\ \exists\, a : \wedge\ This\_Approach = appr\_plate[a] \\ \qquad \wedge\ y \in appr\_plate[a] \end{array} \right)$$

DEFINITION

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

Figure 21.4: Module *Landing_Axioms*

- *CallAttentionToErrorRule*: As soon as the crew or the air traffic controller notices a use of inconsistent data by the other participant in a communication, he/she must question this data;

- *UniqueApproachPlates*: Under the assumption that only the most current approach plates are used, two plates of the same approach may not contain inconsistent data.

So far we have defined rules describing some of the dependencies between state predicates in TLA+ specifications. To specify a system describing the real world, we need more than just rules. Module *Landing_Specs* (figures 21.8, 21.9 and 21.10) introduces actions, safety and liveness conditions which ensure proper functioning of the system.

Table 21.2.1 presents a list of the state predicates defined so far. 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*, which retain current state information for future states to use: once the value of a history variable is set, it remains set until it is changed by some actions. To ensure this, the changing or unchanging value of such a variable is explicitly defined in each action. Such variables as environmental variables, however, cannot be governed by system agents in this way: consider weather conditions. Weather conditions are what they are and change out of control of participants in standard procedures. For future use of this module it might be useful to add conditions taken from standard procedures concerning, for example, when a situation must be considered dangerous. We have done this in particular actions – see *Accept_landing*(*CRW*) or *CRW_breakoff*(*CRW*).

21.2.2 Standard Operating Procedures

Finally, we define several Standard Operating Procedures (SOPs) concerning the ATCCs tasks (Module *SOP_Specs*, Figures 21.13 and 21.14). Again, to reduce the complexity of the specification (and therefore the proof later on), we make an assumption: we claim that destAPT is the only important information to

module *Landing_Norms*

DECLARATIONS

extends *RealTimeTheorems*
extends *Landing_Defs*
instance *Landing_Axioms*
instance *Landing_Specs*
instance *Naturals*
CONSTANTS $AC, ATC, CRW, APT, APPR, This\_Approach$
VARIABLES $tfc, some\_apt, current\_alt, acceptable\_deviation$

DEFINITIONS

$$ProceduralDef1 \triangleq \Box visibility\_acceptable(CRW) \Rightarrow \Diamond visual\_contact(CRW, APT)$$

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

$$Landing\_Procedures \triangleq \left(\begin{array}{ll} \wedge & (AC)near(APT) \\ \wedge & \Box(AC)in\_landing\_phase \end{array} \right) \Rightarrow \Diamond(AC)lands\_at(APT)$$

$$Unique\_APT\_Rule \triangleq \left(\begin{array}{ll} \wedge & (AC)near(APT) \\ \wedge & cleared\_to\_land(ATC, CRW, some\_apt') \end{array} \right) \Rightarrow (some\_apt = APT)$$

$$Landing\_Criterion \triangleq \Box \left(\left(\begin{array}{ll} \wedge & visibility\_acceptable(CRW) \\ \wedge & \vee \wedge ILS\_approach(AC, APT) \\ & \quad \wedge below\_DH \\ & \vee \wedge NP\_approach(AC, APT) \\ & \quad \wedge \vee at\_MDA \\ & \quad\quad \vee below\_MDA \end{array} \right) \Rightarrow visual\_contact(CRW, APT) \right)$$

$$Landing\_Rule \triangleq \Box \left(landing\_accepted(CRW) \Rightarrow \begin{array}{ll} \vee & \Diamond landing \\ \vee & \Diamond landing\_interrupt(CRW, ATC, APT) \end{array} \right)$$

Figure 21.5: Module *Landing_Norms (Part 1)*

module *Landing_Norms (continued)*

$$ILS\text{--}APPR\_Rule \triangleq \left(\begin{array}{ll} \wedge & ILS\_approach(AC, APT) \\ \wedge & below\_DH \\ \wedge & \Diamond\neg visibility\_acceptable(CRW) \end{array}\right) \Rightarrow \Diamond CRW\_breakoff(CRW)$$

$$ILS\text{--}LandingRule \triangleq \left(\begin{array}{ll} \wedge & ILS\_approach(AC, APT) \\ \wedge & \Box\neg CRW\_breakoff(CRW) \end{array}\right) \Rightarrow \Diamond landing$$

$$ILS\text{--}AltProperty \triangleq \left(\begin{array}{ll} \wedge & ILS\_approach(AC, APT) \\ \wedge & \Box\neg CRW\_breakoff(CRW) \end{array}\right) \Rightarrow \textit{alt is a 'monotone decreasing continuous function of RealTime'}$$

$$Normal\text{--}Progress \triangleq \left(\begin{array}{ll} \wedge & APT \neq destAPT \\ \wedge & \Box\neg distress\_descl(CRW) \\ \wedge & \Box\neg urgency\_decl(CRW) \end{array}\right) \Rightarrow \neg\Diamond(AC)lands\_at(APT)$$

$$CallAttentionToErrorRule \triangleq \Box\left(\begin{array}{l} \forall\, x, y \in \{ATC, CRW\} : \\ \quad \wedge \quad x \neq y \\ \quad \wedge \quad Attend(x, data\_inconsist(Navaids[This\_Approach], \\ \qquad\qquad\qquad\qquad Navaids[appr\_plate[APT]])) \\ \quad \Rightarrow question(x, y, data\_inconsist(Navaids[This\_Approach], \\ \qquad\qquad\qquad\qquad Navaids[appr\_plate[APT]])) \end{array}\right)$$

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

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

Figure 21.6: Module *Landing_Norms* (Part 2)

module *Landing_Norms (continued)*

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

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

DEFINITION

$$\begin{aligned} LNSpec \triangleq\ & \wedge\ ProceduralDef1 \\ & \wedge\ GeneralSafetyRule \\ & \wedge\ Landing\_Procedures \\ & \wedge\ Unique\_APT\_Rule \\ & \wedge\ Landing\_Criterion \\ & \wedge\ Landing\_Rule \\ & \wedge\ ILS{-}APPR\_Rule \\ & \wedge\ ILS{-}LandingRule \\ & \wedge\ ILS{-}AltProperty \\ & \wedge\ Normal{-}Progress \\ & \wedge\ CallAttentionToErrorRule \\ & \wedge\ AttendErrorRule \\ & \wedge\ UniqueApproachRule \\ & \wedge\ AttendNavaids \\ & \wedge\ Deviation\_Breakoff \end{aligned}$$

Figure 21.7: Module *Landing_Norms* (Part 3)

module *Landing_Specs*

DECLARATIONS

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

Definition

$all - variables \triangleq \langle some\_apt, current\_alt, acceptable\_deviation, tfc \rangle$

PREDICATES

$$landing\_interrupt(CRW, ATC, APT) \triangleq \begin{pmatrix} \lor & \Box distress\_decl(CRW) \\ \lor & \Box urgency\_decl(CRW) \\ \lor & \Box\neg flightpath\_acceptable(CRW) \\ \lor & endanger(CRW, trf) \\ \lor & \neg cleared\_to\_land(ATC, CRW, APT) \\ \lor & \neg visibility\_acceptable(CRW) \end{pmatrix}$$

$$\begin{aligned} Init \triangleq\ & \land (AC)near(APT) \\ & \land \neg distress\_decl(CRW) \\ & \land \neg urgency\_decl(CRW) \\ & \land flightpath\_acceptable(CRW) \\ & \land \neg endanger(CRW, tfc) \\ & \land \neg cleared\_to\_land(ATC, CRW, APT) \\ & \land \neg landing\_accepted(CRW) \\ & \land visibility\_acceptable(CRW) \end{aligned}$$

ACTIONS

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

Figure 21.8: Module *Landing_Specs (Part 1)*

module *Landing_Specs (continued)*

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

Figure 21.9: Module *Landing_Specs (Part 2)*

module *Landing_Specs (continued)*

$$
\begin{array}{lcl}
\mathit{Decl\_urgency}(\mathit{CRW}) & \triangleq & \land\ \neg \mathit{urgency\_decl}(\mathit{CRW}) \\
& & \land\ \mathit{urgency\_decl}(\mathit{CRW})' \\
& & \land\ \text{UNCHANGED}\ (\mathit{AC})\mathit{near}(\mathit{APT}) \\
& & \land\ \text{UNCHANGED}\ (\mathit{AC})\mathit{in\_landing\_phase} \\
& & \land\ \text{UNCHANGED}\ \mathit{flightpath\_acceptable}(\mathit{CRW}) \\
& & \land\ \text{UNCHANGED}\ \mathit{cleared\_to\_land}(\mathit{ATC}, \mathit{CRW}, \mathit{APT}) \\
& & \land\ \text{UNCHANGED}\ \mathit{distress\_decl}(\mathit{CRW}) \\
& & \land\ \text{UNCHANGED}\ \mathit{landing\_accepted}(\mathit{CRW}) \\
& & \land\ \text{UNCHANGED}\ \mathit{endanger}(\mathit{CRW}, \mathit{tfc}) \\
& & \land\ \text{UNCHANGED}\ \mathit{visibility\_acceptable}(\mathit{CRW}) \\
\mathit{Consider\_danger}(\mathit{CRW}) & \triangleq & \land\ \neg \mathit{endanger}(\mathit{CRW}, \mathit{tfc}) \\
& & \land\ \mathit{endanger}(\mathit{CRW}, \mathit{tfc})' \\
& & \land\ \text{UNCHANGED}\ (\mathit{AC})\mathit{near}(\mathit{APT}) \\
& & \land\ \text{UNCHANGED}\ (\mathit{AC})\mathit{in\_landing\_phase} \\
& & \land\ \text{UNCHANGED}\ \mathit{flightpath\_acceptable}(\mathit{CRW}) \\
& & \land\ \text{UNCHANGED}\ \mathit{cleared\_to\_land}(\mathit{ATC}, \mathit{CRW}, \mathit{APT}) \\
& & \land\ \text{UNCHANGED}\ \mathit{distress\_decl}(\mathit{CRW}) \\
& & \land\ \text{UNCHANGED}\ \mathit{landing\_accepted}(\mathit{CRW}) \\
& & \land\ \text{UNCHANGED}\ \mathit{urgency\_decl}(\mathit{CRW}) \\
& & \land\ \text{UNCHANGED}\ \mathit{visibility\_acceptable}(\mathit{CRW}) \\
\mathit{CRW\_breakoff}(\mathit{CRW}) & \triangleq & \land\ \mathit{landing\_accepted}(\mathit{CRW}) \\
& & \land\ (\mathit{AC})\mathit{in\_landing\_phase} \\
& & \land\ \neg(\mathit{AC})\mathit{in\_landing\_phase}' \\
& & \land\ \lor\ \mathit{Decl\_distress}(\mathit{CRW}) \\
& & \quad \lor\ \mathit{Decl\_urgency}(\mathit{CRW}) \\
& & \quad \lor\ \neg \mathit{Accept\_flightpath}(\mathit{CRW}) \\
& & \quad \lor\ \mathit{Consider\_danger}(\mathit{CRW}) \\
& & \quad \lor\ \neg \mathit{visibility\_acceptable}(\mathit{CRW}) \\
& & \land\ \text{UNCHANGED}\ (\mathit{AC})\mathit{near}(\mathit{APT}) \\
& & \land\ \text{UNCHANGED}\ \mathit{cleared\_to\_land}(\mathit{ATC}, \mathit{CRW}, \mathit{APT}) \\
& & \land\ \text{UNCHANGED}\ \mathit{landing\_accepted}(\mathit{CRW})
\end{array}
$$

Figure 21.10: Module *Landing_Specs* (Part 3)

module *Landing_Specs (continued)*

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

DEFINITION

$$
\begin{array}{lll}
Rules & \triangleq & \land\ ProceduralDef\,1 \\
& & \land\ APT/ATC\_Id\_Rule \\
& & \land\ Landing\_Procedures \\
& & \land\ Clearance\_Rule \\
& & \land\ Unique\_APT\_Rule \\
& & \land\ Unique\_Approach - Type\_Rule \\
& & \land\ Landing\_Criterion \\
& & \land\ Landing\_Rule \\
& & \land\ Acceptance\_Rule \\
& & \land\ LandingSafetyRule \\
& & \land\ ILS{-}APPR\_Rule \\
& & \land\ ILS{-}LandingRule \\
& & \land\ ILS{-}AltProperty \\
SomeAction & \triangleq & \lor\ Accept\_flightpath(CRW) \\
& & \lor\ Accept\_landing(CRW) \\
& & \lor\ Cleared\_to\_land(ATC, CRW, APT) \\
& & \lor\ Decl\_distress(CRW) \\
& & \lor\ Decl\_urgency(CRW) \\
& & \lor\ Consider\_danger(CRW) \\
& & \lor\ CRW\_breakoff(CRW) \\
& & \lor\ ATC\_breakoff(ATC) \\
Spec & \triangleq & \land\ Init \\
& & \land\ Rules \\
& & \land\ \Box[SomeAction]_{all\_variables} \\
& & \land\ \text{WF}[Cleared\_to\_land(ATC, CRW, APT)]_{all\_variables} \\
& & \land\ \text{WF}[Accept\_landing(CRW)]_{all\_variables}
\end{array}
$$

Figure 21.11: Module *Landing_Specs* (Part 4)

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 21.12: State Predicates from Specifications of Landing Procedures

determine which the En-Route ATCs shall be (usually the flight plan contains information such as *AC call sign*, *AC type*, *transponder code*, *assigned altitude*, *destination airport* and *route of flight* [Nol94, p. 412]).

We do not claim that these specifications are complete. The definitions we have presented were sufficient for our analysis. This may be called *lazy* specification: we specify all we need for the current purposes. Ultimately a *greedy specification* is needed: a specification of all standard procedures and aviation regulations. We think this is feasible along the lines of what we have done here. It would be a lot of work, and it wouldn't serve the current purposes to attempt it here.

module *SOP_Specs*

DECLARATIONS

extends *Naturals*, *Sequences*
extends *ATCcomm*, *ATCcomm_history*
extends *Landing_Specs*
CONSTANTS $AC, ATC, APT, destATC, nextATC, tfc$

ASSUMPTIONS

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

DEFINITIONS

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

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

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

$$(AC)in\_area(ATC) \triangleq position(AC) \in region(ATC)$$

$$false\_FI(ATC, AC) \triangleq (det\_destAPT(AC) \neq destAPT(AC))$$

$$Inter\_ATC\_Handoff(x, y, AC) \triangleq$$

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

Figure 21.13: Module *SOP_Specs* (Part 1)

module *SOP_Specs (continued)*

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

$$ATC\text{–}Responsibility\text{–}Rule \triangleq (AC)in\_area(ATC) \Rightarrow EnRouteProcessing(ATC, AC)$$

DEFINITION

$$\begin{array}{lll} Spec & \triangleq & \wedge\ ATC\text{–}Responsibility\text{–}Rule \\ & & \wedge\ DetermineDestinationProcedure \\ & & \wedge\ EnRouteProcessing(ATC, AC) \\ & & \wedge\ Inter\_ATC\_Handoff(x, y, AC) \end{array}$$

Figure 21.14: Module *SOP_Specs* (Part 2)