

# Cognitive Modelling of Pilot Errors and Error Recovery in Flight Management Tasks

Andreas Lüdtkke<sup>1</sup>, Jan-Patrick Osterloh<sup>1</sup>, Tina Mioch<sup>2</sup>,  
Frank Rister<sup>3</sup>, Rosemarijn Looijje<sup>2</sup>

<sup>1</sup> OFFIS e.V., Escherweg 2, 26127 Oldenburg, GERMANY

<sup>2</sup> TNO Human Factors, Kampweg 5, 3796 DE Soesterberg, THE NETHERLANDS

<sup>3</sup> Hapag-Lloyd Flug, Flughafenstrasse 10, 30855 Langenhagen, GERMANY

{luedtke, osterloh}@offis.de, {tina.mioch, rosemarijn.looiije}@tno.nl, frank.rister@hamburg.de

**Abstract.** This paper presents a cognitive modelling approach to predict pilot errors and error recovery during the interaction with aircraft cockpit systems. The model allows execution of flight procedures in a virtual simulation environment and production of simulation traces. We present traces for the interaction with a future Flight Management System that show in detail the dependencies of two cognitive error production mechanisms that are integrated in the model: Learned Carelessness and Cognitive Lockup. The traces provide a basis for later comparison with human data in order to validate the model. The ultimate goal of the work is to apply the model within a method for the analysis of human errors to support human centred design of cockpit systems. As an example we analyze the perception of automatic flight mode changes.

**Keywords:** Human Error Prediction, Human-Centred Design, Cognitive Model

## 1 Introduction

Aircraft pilots are faced with a complex traffic environment. Cockpit automation and support systems help to reduce complexity. Currently, a lot of research is done to improve the onboard management of flight trajectories and the negotiation of trajectory changes with Air Traffic Control (ATC). During the flight, many factors may induce changes to the original flight plan, e.g. bad weather, traffic conflicts, or runway changes. In future air traffic management, an aircraft will be equipped with an advanced flight management system that provides information on the current traffic and weather status in an intuitive form. This allows pilots to easily adapt a flight route via a graphical Advanced Human Machine Interface (AHMI). Voice communication between aircraft and ATC will be partly replaced by Data Link communication which provides pilots and controllers with a detailed electronic picture of the time and space (4D) trajectory. This allows efficient negotiation of route changes and improves

predictability of conflicts between aircraft or between planned routes and severe weather conditions.

In order to leverage this new air traffic management concept, intuitive and easy-to-use human machine interfaces as well as efficient and robust flight procedures are needed. Safe operation of aircraft is based on normative flight procedures (standard operating procedures) and rules of good airmanship, which we will refer to as 'normative activities'. We define pilot errors as deviations from normative activities. In the past, several cognitive explanations and theories have been proposed to understand why pilots deviate from normative activities (e.g. [7]). The European project HUMAN, in which the research described in this paper is done, strives to pave a way of making this knowledge readily available to designers of new cockpit systems. We intend to achieve this by means of a valid executable flight crew model which incorporates cognitive error-producing mechanisms leading to deviations from normative activities. The model interacts with models of cockpit systems (like advanced flight management systems) in a virtual simulation environment to predict deviations and its potential consequences on the safety of flight. The ultimate objective of HUMAN is to apply this model to analyze human errors and support error prediction in ways that are usable and practical for human-centred design of systems operating in complex cockpit environments.

This paper focuses on the interaction between two highly relevant cognitive error-producing mechanisms:

- routine learning leading to *Learned Carelessness* (effort-optimizing shortcuts leading to inadequate simplifications that omit safety critical aspects of normative activities) and
- attention allocation (deciding where to allocate the limited cognitive resources) leading to *Cognitive Lockup* (failing to switch attention when currently working on a demanding task).

At the initial stage of HUMAN we performed questionnaire interviews with pilots and human factor experts based on a literature survey of error-producing mechanisms. We identified Learned Carelessness and Cognitive Lockup to be among the most relevant mechanisms for modern and future cockpit human machine interfaces.

This paper describes how we modelled these two processes in one integrated executable cognitive flight crew model and discusses in detail hypotheses derived from the model.

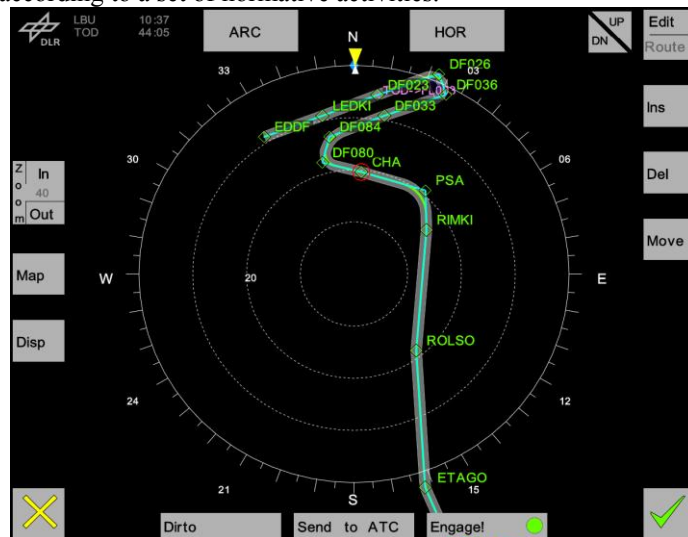
## **2 Re-planning via 4D-Flight Management Systems**

Today, the flight management system, which controls the lateral and vertical movement of an aircraft, is operated by a multi-purpose control display unit (MCDU). The MCDU consists of a small monitor and an alphanumerical keyboard, by which the pilots type in the desired flight plan changes. Flight plans consist of a certain number of waypoints, identified by a three or five letter code, which is entered into the MCDU. The airplane's autoflight system can be coupled to the flight plan, which

then follows the plan automatically. However, clearance requests and reception for the different sections of the flight plan are mandatory, and are today performed via voice communication with the ATC. Problems with this are that communicating route changes via voice is a lengthy and error-prone process [2], and that the interaction with the MCDU is cumbersome and inefficient (e.g. [6]). As described in the introduction, future flight management systems and their user interfaces try to tackle these problems. For our study we use an advanced flight management system and its AHMI, which have been developed by the German Aerospace Institute (DLR, Braunschweig, Germany). Both systems are used as demonstration settings for the current research, without their design playing a role in the validity of the current research.

The AHMI represents flight plans on a map with their status being graphically augmented by different colours and shapes, e.g. if a new trajectory is generated after the flight plan has been changed, it is displayed as a dotted line, while the active trajectory is solid of another colour (cf. Fig. 1). Still, pilots can insert, move or delete waypoints, but also handle a lot of different events, e.g. display weather radar information, allowing graphical re-planning to avoid a thunderstorm. However, the insertions do not necessarily make use of keyboards such as the MCDU - manipulation is done directly on the map by trackball cursor-control. Any trajectory created by the pilot is generated as a data-link, ready to be sent to ATC for negotiation.

The advanced flight management system and its AHMI is used in HUMAN as a target system to demonstrate the predictive capabilities of the cognitive flight crew model by simulating the interaction between system and crew in different re-planning scenarios according to a set of normative activities.



**Fig. 1:** AHMI of the Flight Management target system

Since this is a new system we had to define the normative activities (NA) from scratch. Knowledge acquisition techniques were used to gather first ideas for the

scenarios and NA definition. As a second step, common Standard Operating Procedures (SOP) and Rules of Good Airmanship were the basis of workflow patterns which were applied and refined by test and trainer pilots working in the field of procedure and training-scenario/simulation design.

Next, these procedural workflow patterns were translated into a textual description. This textual description served as the basis for the first plot of NA's in table format. These tables, in turn, were used to model the NAs in the semi-formal task modelling software AMBOSS [20]. The task trees were useful in two ways: first of all, the AMBOSS models were used to reveal the flaws in the NA tables that were undetectable without a simulation. And second, these tables paved the path for a formal model of the normative activities, which are then input for the cognitive architecture. Fig. 2 represents the modelling process.

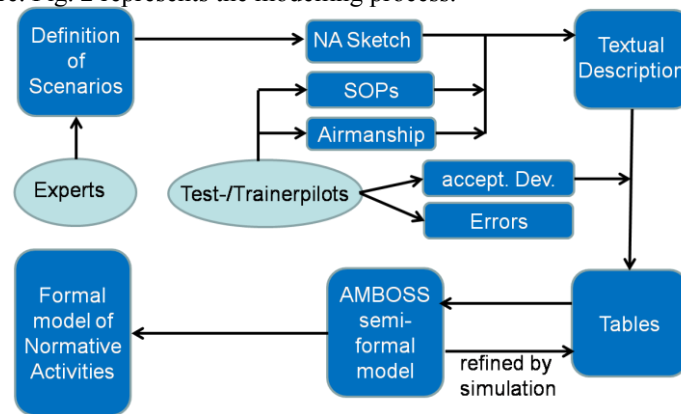


Fig. 2: Modelling Process

The most relevant activities for this paper are those for re-planning. Re-planning means modifying the current flight route via the AHMI by changing the lateral or vertical profile. Changes to the route can be initiated either by the pilots or by the controllers. In the first case the pilots introduce the changes into the route and send it down to ATC (downlink). In the latter case ATC sends a modified route up to the aircraft (uplink). In both cases the last three actions which have to be performed by the pilots are the same: (NA1) Generate the modified route by clicking on the “Dirto” button (Fig. 1, bottom left), as a result the new trajectory is shown as a dotted line; (NA2) click the “Send to ATC” button (Fig. 1, bottom middle) to downlink self-initiated changes or to acknowledge uplinked changes; (NA3) next a feedback from ATC is received in the form of an uplink. Since this uplink may contain further lateral or vertical changes pilots must check the lateral and vertical profile to identify any final modifications. If a change introduced by ATC at this stage is not acceptable for any reason, then the re-planning procedure has to be restarted by the pilots resulting in a new downlink. In case no changes have been received or the changes are acceptable pilots have to press the “Engage” button (Fig. 1, bottom right) to activate the new route.

The trajectory in Fig. 1 represents a typical re-planning scenario which we used in the HUMAN project to generate detailed hypotheses on pilot behavior (see Section 5)

provoking Learned Carelessness and Cognitive Lockup. It starts during cruising at flight level 250 (25,000 feet) on a flight inbound to FRA (Frankfurt, Germany). Passing waypoint ETAGO (approx. 130NM inbound to Frankfurt), a system non-normal message pops up advising the crew of a fuel-pump malfunction. The normative activities require the crew to initiate descent to maximum flight level 100 in order to assure adequate pressure for continuous fuel feed to the respective engine (approx. 60NM earlier than planned). This will be done by a cruise-level alteration in the current flight route via the AHMI followed by a trajectory generation, negotiation and activation (steps NA1, NA2 and NA3) as described above. During descent, the crew receives the latest weather report of Frankfurt which allows preparing for the given approach. The report indicates that there is a thunderstorm approaching the airport which should be monitored from now on by the crew on the weather radar. In the vicinity of waypoint ROLSO, the crew receives a shortcut uplink which clears the flight to proceed directly to waypoint CHA. In this case the pilots are required to check the uplinked changes and either accept them by performing steps NA1, NA2, NA3 or to introduce changes before doing so. The scenario foresees that during NA3 the uplink received by the crew contains the standard flight level for the current arrival segment which is flight level 110, 1000 feet higher than the previous clearance and off the operational envelope regarding the system malfunction. This is to be recognized by the pilots while checking the vertical profile of the uplink, who should correct the altitude and then re-negotiate with ATC starting again with NA1. If the incorrect altitude was engaged by the crew then the aircraft would re-climb to flight level 110.

The main questions which are investigated are:

- Does the pilot model recognize the incorrect altitude?
- Is the pilot model able to recover from the re-climb by initiating a new descent via the AHMI?

In Section 5, we show that the approaching thunderstorm may have a significant effect on the error recovery.

### **3 Cognitive Processes Involved in Re-planning Tasks**

To explain and model why pilots deviate from normative activities, we have focussed on the underlying cognitive processes. In this section, we describe cognitive processes that play a role in re-planning and that are the basis for our crew model.

Cognitive processes can be differentiated by their degree of consciousness. Rasmussen [5] defines three different behaviour levels in which cognitive processing and hence errors can take place: skill-based, rule-based, and knowledge-based behaviour. The level of processing mainly depends on the experience with a task. Anderson [1] distinguishes very similar levels but uses the terminology of autonomous, associative, and cognitive level.

A task that is encountered for the first time is processed on the cognitive level with maximal cognitive effort. This processing is goal driven; alternative plans to reach a goal are evaluated usually through mental simulation, and finally one plan is selected to be executed. With some experience, the associative level is used, where solutions

are stored that proved to be successful; the pilot has for example learned how to handle the cockpit systems in specific flight scenarios. According to Rasmussen [5], processing is controlled by a set of rules that have to be retrieved and then executed in the appropriate context. On the autonomous level routine behaviour emerges that is applied without conscious thought, e.g. manually manoeuvring an aircraft.

When solving a task, people tend to apply a solution on the lower levels first, and only revert to solutions on higher levels when lower-level ones are not available [5] or when the situation requires very careful handling due to unusual and safety relevant conditions.

In our research, we focus on two kinds of error production mechanisms that we associate with the associative and the cognitive level respectively, namely Learned Carelessness and Cognitive Lockup.

*Learned Carelessness:* When re-planning takes place on the associative layer, the procedure may be simplified according to scenarios encountered before. The psychological theory of Learned Carelessness states that humans have a tendency to neglect safety precautions if this has immediate advantages, e.g. it saves time because less physical or cognitive resources are necessary [11]. Careless behaviour emerges if safety precautions have been followed several times but would not have been necessary, because no hazards occurred. Then, people tend to omit the safety precautions and the absence of hazardous consequences acts as a negative reinforcer of careless behaviour.

*Cognitive Lockup:* On the cognitive layer, the cognitive attention may be captured by a task, which causes people to switch between tasks too late or not at all. This usually happens in situations with a high multitask workload, as switching between tasks costs time and effort, and cognitive resources are limited [3].

## **4 Modelling Re-planning in a Layered Cognitive Architecture**

Cognitive architectures were established in the early eighties as research tools to unify psychological models of particular cognitive processes [12]. These early models only dealt with laboratory tasks in non-dynamic environments [13], [14]. Furthermore, they neglected processes such as multitasking, perception and motor control that are essential for predicting human interaction with complex systems in highly dynamic environments like the air traffic environment addressed in HUMAN with the AFMS target system. Models such as ACT-R and SOAR have been extended in this direction [15], [18] but still have their main focus on processes suitable for static, non-interruptive environments. Other cognitive models like MIDAS [16], APEX [17] and COGNET [19] were explicitly motivated by the needs of human-machine interaction and thus focused for example on multitasking right from the beginning. To our knowledge, none of these architectures has a multi-layered knowledge processing, with different levels of consciousness, as proposed in the following.

#### 4.1 The Cognitive Architecture CASCaS

In HUMAN the cognitive architecture CASCaS (Cognitive Architecture for Safety Critical Task Simulation) as depicted in Fig. 3 is used to model the cognitive process described in the previous section. CASCaS is based on research performed by OFFIS in the European project ISAAC (6th Framework Programme) [8], and has been extended in HUMAN to cover two of Anderson's behaviour levels (c.f. section 3).

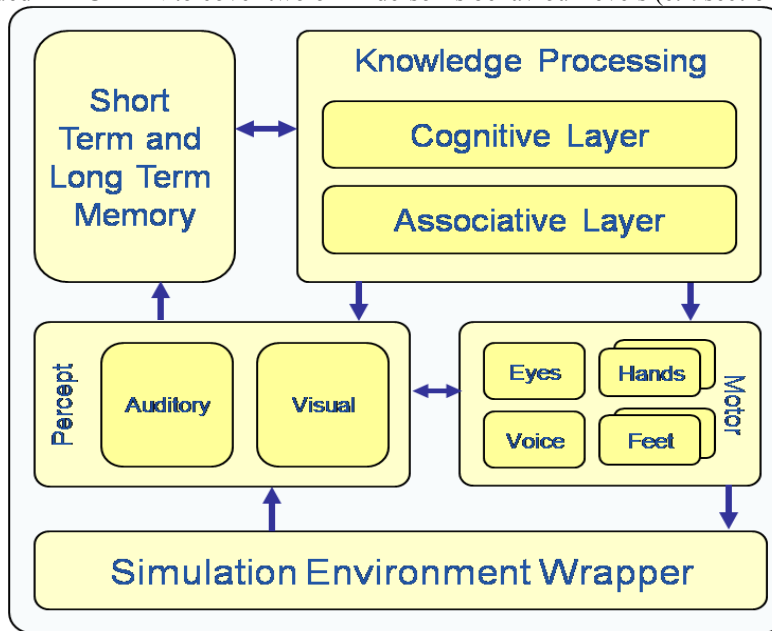


Fig. 3: CASCaS Architecture

The core of CASCaS is formed by the layered *knowledge processing* component that contains the associative and the cognitive layer.

Knowledge for both layers is stored in the *memory component*. The *short-term memory* stores variable-value pairs of data that have been perceived from the environment or derived by applying rules (see below). The *long-term memory* stores flight procedures in form of Goal-State-Means (GSM) rules (Fig. 3). All rules consist of a left-hand side and a right-hand side. The left-hand side consists of a goal in the Goal-Part and a State-part, which specifies Boolean conditions on the current state of the environment, together with associated memory-read items to specify variables that have to be retrieved from memory. The right-hand side consists of a Means-Part containing motor as well as percept actions (e.g. hand movements or attention shifts), memory-store items and a set of partial ordered sub-goals.

Rule 1 in Fig. 4 defines a goal-sub-goal relation between HANDLE\_ATC\_UPLINK and the three sub-goals GENERATE\_ROUTE, NEGOTIATE\_ROUTE and CHECK\_ATC\_UPLINK\_VERT\_PREPARE. The precondition in the goal term imposes a temporal order on the sub-goals, i.e. NEGOTIATE\_ROUTE can only be performed after GENERATE\_ROUTE.

<b>Rule 1:</b>	
	Goal (HANDLE_ATC_UPLINK) (G)goal-Part
IF	Memory-Read(atc_uplink_present) (S)state-Part
	Condition(atc_uplink_present==true)
	→
THEN	Goal (GENERATE_ROUTE) (M)ean-Part
	Goal (NEGOTIATE_ROUTE, precondition=GENERATE_ROUTE)
	Goal (CHECK_ATC_UPLINK_VERT_PREPARE, precondition=NEGOTIATE_ROUTE)
<b>Rule 2:</b>	
IF	Condition(atc_uplink_message==true) (S)state-Part
	→
THEN	Memory-Store (atc_uplink_present, true) (M)ean-Part
	Goal (HANDLE_ATC_UPLINK)

**Fig. 4.** Format of GSM rules

Additionally to the GSM-rules we added a second rule type, called reactive rules. Rule 2 in Fig. 4 is an example for this rule type. The only difference is that reactive rules have no Goal-Part. While GSM-rules represent deliberate behaviour, and are selected by the knowledge processing component during the execution of a flight procedure, reactive rules (State-Means rules) represent immediate or reactive behaviour which is triggered by events in the environment, e.g. in rule 2 of Fig. 4 an ATC uplink message (`atc_uplink_mesage==true`) triggers the goal `HANDLE_ATC_UPLINK`.

The *associative layer* selects and executes rules from long-term memory. It is modelled as a production system. Characteristic for such systems is a serial cognitive cycle for processing rules: A goal is selected from the set of active goals (Phase 1), all rules containing the selected goal in their Goal-Part are collected and a short-term memory retrieval of all state variables in the Boolean conditions of the collected rules is performed (Phase 2). If a variable is absent in memory, a dedicated percept action is fired and sent to the percept component to perceive the value from the environment and to write it into the short-term memory. After all variables have been retrieved, one of the collected rules is selected by evaluating the conditions (Phase 3). Finally the selected rule is fired (Phase 4), which means that the motor and percept actions are sent to the motor and percept component respectively and the sub-goals are added to the set of active goals. This cycle is started when a Boolean condition of a reactive rule is true. In Phase 2 reactive rules may be added to the set of collected rules if new values for the variables contained in the State-Part have been added to the memory component (by the percept component). In Phase 3, reactive rules are always preferred to non-reactive rules. The cognitive cycle is iterated until no more rules are applicable.

The *cognitive layer* reasons about the current situation and makes decisions based on this reasoning. Consequently, we differentiate between a decision-making module, a module for task execution and a module for interpreting perceived knowledge (sign-symbol translator).

The *decision-making module* determines which goal is executed. Goals have priorities, which depend on several factors: first, goals have a static priority value that is set by a domain expert. Second, priorities of goals increase over time if not executed. Implicitly, temporal deadlines are modelled in this way. If, while executing



a goal, another goal has a clearly higher priority than the current one, the execution of the current goal is stopped and the new goal is attended to.

The *task-execution module* executes the goals that have been chosen by the decision-making module. (Sub-)tasks might be passed to the associative layer if rules exist in long-term memory.

The *sign-symbol translator* is based on Rasmussen's differentiation between signs and symbols [5]. This module raises the level of abstraction of the signs perceived by the percept component and stored in short-term memory by identifying and interpreting the situation, and thereby adding extra knowledge to the sign. In addition, background knowledge is applied to judge and evaluate the current situation.

The associative and cognitive layer interact in the following ways: first, the cognitive layer can start (and thus delegate), monitor, temporally halt, resume and stop activities on the associative layer by manipulating the associative layer's goal agenda. Monitoring of the associative layer is realized through determining whether the appropriate goals are placed in the goal agenda.

The associative layer can inform the cognitive layer about the status of rule execution, e.g. current execution is stuck because for the chosen goal no rules are available in long-term memory or execution of a perceived event cannot be started for the very same reason. In these cases the cognitive layer starts to perform the goal or event. Furthermore, the cognitive layer can take over control at any time. Currently this is initiated by setting the parameter "Consciousness". If the value is "associative" then every event will first be processed if possible and the cognitive layer becomes only active if no rules are available. If the value is "cognitive" then the cognitive layer processes each event independent of the availability of rules.

The *percept* component consists of two sub-components, an *auditory* component for receiving sounds or vocal input (in form of variables representing acoustic input), and a *visual* component for perception of visual input (in form of variables representing visual input). While the auditory component is purely reactive to external input, the visual component can be controlled by the knowledge processing component via percept-actions contained in rules. Percept-actions result in eye movements, which are performed by the *eyes* sub-component in the motor component. The eyes component has a detailed model of eye movements, in order to simulate the timing of the eye movements. For a more detailed description of the visual component, see [4]. All information that has been perceived is stored in the short-term memory of the cognitive architecture.

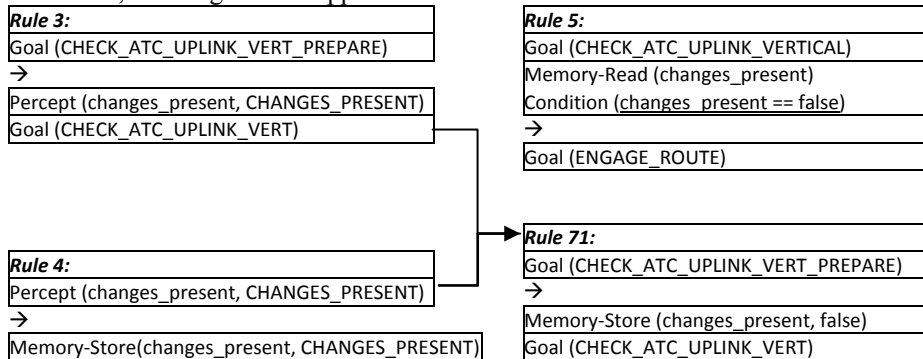
The *motor* component contains, apart from the eye component, modules for *hand* and *feet* movement. These components use the 2D and 3D formulations of Fitt's Law [10] in order to model the timing of the requested movements (via motor actions received from the knowledge processing component). With these components, the cognitive model can for example simulate button presses.

The *Simulation Environment Wrapper* provides data for the percept component and functions for the motor component to manipulate the environment by connecting CASCaS with different simulation backends. In HUMAN we connected CASCaS to the fixed base flight simulator used by the DLR for experiments with human pilots. In this way the model can be executed and data can be recorded in the very same environment in which also human subject pilots interact. This allows validation of the model by comparing model data with human data.

## 4.2 The Error-Producing Mechanisms

Learned Carelessness is modelled on the associative layer by a dedicated learning algorithm. This is modelled by melting two rules into one rule by means of *rule composition* [8]. A precondition for composing rules is that firing of the first rule has evoked the second rule, or more exact, the first rule derives a sub-goal that is contained in the Goal-Part of the second rule. Melting the rules means building a composite rule by combing the left-hand sides of both rules and also combing both right-hand sides. The crucial point is that in this process elements that are contained on the right-hand side of the first and also on the left hand side of the second rule are eliminated. This process cuts off intermediate knowledge processing steps.

Rule 5 in Fig. 5 specifies that it is only allowed to proceed with engaging the route [Goal(ENGAGE\_ROUTE)] if the vertical profile contains no changes ( $\text{changes\_present} == \text{false}$ ). Using rule 3 the current value of the variable is perceived from the AHMI. Rule 4 stores the perceived value into the short-term memory. Mostly when pilots want to engage a route, there are actually no changes to the vertical profile. Thus, most of the time the percept action delivers 'false'. Our pilot model produces a new simplified rule by merging rule 3 and 4 to rule 71, where the existence of changes is no longer perceived from the AHMI but is just retrieved from memory. The percept action has been eliminated and the simplified rule always stores the value 'false' into the memory. Applying rule 71 results in careless behaviour: engaging an uplinked route independent from actual changes in the vertical profile. At the beginning of the simulation, all rules in the long-term memory component are normative, meaning that the application of these rules does not lead to an error.



**Fig. 5:** Composition of rule 3 and 4 to rule 71, which lead in rule 5 to careless behaviour

*Cognitive lockup* is implemented as part of the goal decision mechanism, thus on the cognitive layer. In certain situations, switching between goals does not take place even though the priority of another goal is higher than the currently selected one. The selection mechanism is extended by the parameter Task Switch Costs (TSC), which determines the difference that the priorities need to have to halt the execution of a goal to select a different goal to be executed. Task Switch Costs are described extensively in the literature (e.g. [9]). The TSC depends on the cognitive demands of the current task. The higher the cognitive demands the higher are the costs to switch a task:  $\text{TSC} = \text{StartTSC} + \text{cognitive\_complexity\_current\_task}$ . The parameter StartTSC

denotes the threshold that gives the difference in priority two goals need to have to make an interruption of the one goal and the changing to the other goal possible. This parameter is determined by experimentation. The cognitive complexity of a task is determined by a domain expert and increases the threshold to switch tasks.

## 5 Detailed Hypotheses on Re-planning Behaviour

This section describes hypotheses on pilot behaviour that have been derived by executing the cognitive crew model in the flight scenario described in Section 2. The hypotheses will in the future be used to validate the model behaviour by comparing the simulation traces of the model with traces of real pilot behaviour. Our hypotheses describe predictions generated by the model with regard to a pilot error due to Learned Carelessness, a pilot error due to Cognitive Lockup and the interaction between both mechanisms in the course of error recovery. The predictions are presented in the form of simulation traces.

**Hypothesis 1:** *If checking the vertical profile never shows any irregularities, Learned Carelessness will inhibit this check in the future*

The re-planning procedure prescribes to check the vertical profile after the acknowledge has been received from ATC. It can happen that ATC does not accept the altitude that has been downlinked via the AHMI. In this case altitude changes can be seen in the vertical profile. Since this check costs effort, in terms of time needed for goal selection, percept and motor actions, and since altitude changes by ATC are rather unlikely in that phase of the re-planning procedure the check is prone to be omitted after a certain number of procedure repetitions. Our cognitive model learns a simplified procedure rule (c.f. rule 71 in Section 4.2) in which the check is no longer present. Fig. 6 shows this phenomenon as generated by the pilot model in the scenario of Section 2.

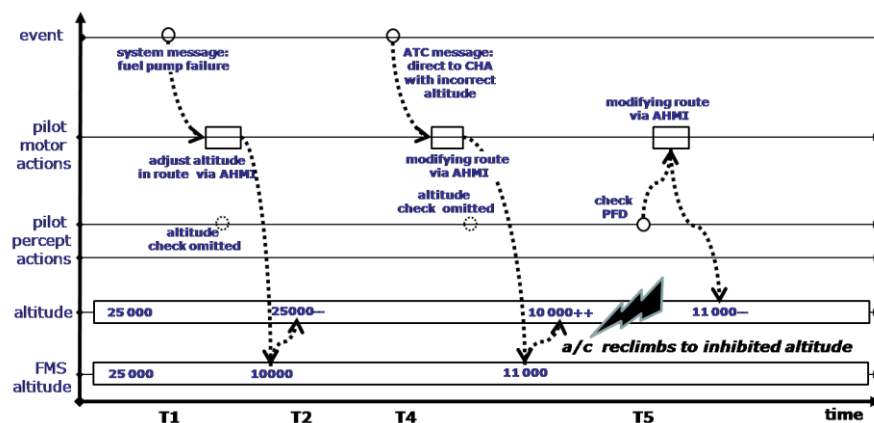


Fig. 6: Pilot error due to Learned Carelessness

At the beginning of the scenario the model has already flown two other experimental scenarios with twelve re-planning events. A simplified rule without the

vertical profile check has been learned in our simulations after the 10th procedure repetition and was first applied during the 11th repetition. At T1 in scenario C the fuel pump fails which requires the pilots to descend to altitude 10000. The pilot model adjusts the altitude of the current route via the AHMI, sends it to ATC and receives an acknowledge which is then engaged. The altitude is not checked but in this case there are no consequences. At T4 ATC sends a shortcut allowing the aircraft to fly directly to waypoint CHA. This uplink contains a vertical profile (altitude 11000) that violates the altitude constraint (altitude 10000) which still holds due to the fuel pump malfunction. The model does not notice this violation because it again omits the altitude check before engaging the changed route. Thus, the aircraft starts to re-climb to altitude 11000. After a certain while, at T5, the model recognizes the climb during regular monitoring of the flight conditions. The model corrects the vertical profile of the route via the AHMI which make the aircraft descend again.

**Hypothesis 2:** *If the cognitive layer keeps control of the proceduralized check on the associative layer, irregularities in the vertical profile will be detected*

We built an alternative version of the pilot model in which the parameter *Consciousness* is set to “cognitive” whenever a system failure is experienced. This value is maintained until the problem is solved. This version of the model has been used to derive an alternative hypothesis for the same scenario (Fig. 7). At T1 the fuel pump failure occurs and *Consciousness* is set to “cognitive”. As a consequence the pilot model performs the modification of the route after T4 on the cognitive layer and thus the original non-careless version of the re-planning procedure is applied. The pilot model recognizes the incorrect altitude, corrects it and sends it to ATC, where the change is accepted and sent back.

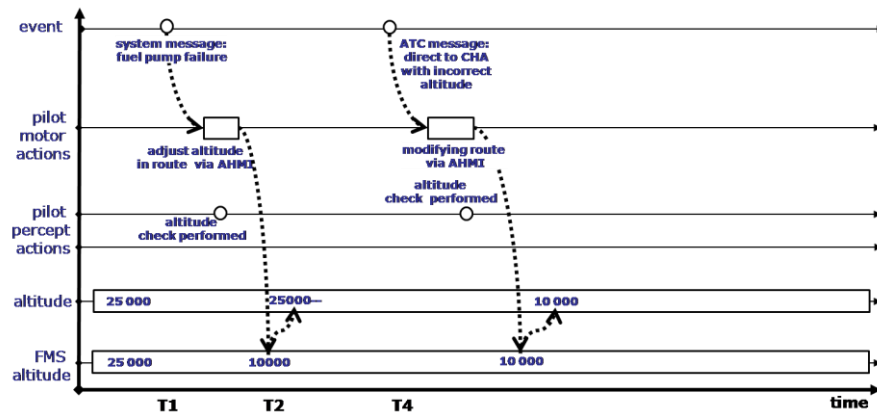


Fig. 7: Conscious procedure execution prevents pilot error

**Hypothesis 3:** *When a task requires high cognitive demand, other tasks might be inadequately neglected – this Cognitive Lockup will delay subsequent recoveries of irregularities in the vertical profile that were not detected by the associative layer due to Learned Carelessness.*

For this hypothesis we assume a variant of the scenario with an additional event which is emitted at T3 (). Pilots receive a weather report update indicating that there is a thunderstorm approaching the airport which should be monitored from now on by the crew on the weather radar. As a result the pilot model is focused on monitoring the thunderstorm that the climb of the aircraft due to the incorrect uplink at T4 is recognized considerably later than in the preceding scenario. The reason is the Cognitive Lockup mechanism. The model does not switch to the regular task of monitoring the flight conditions because monitoring of the thunderstorm is a demanding task.

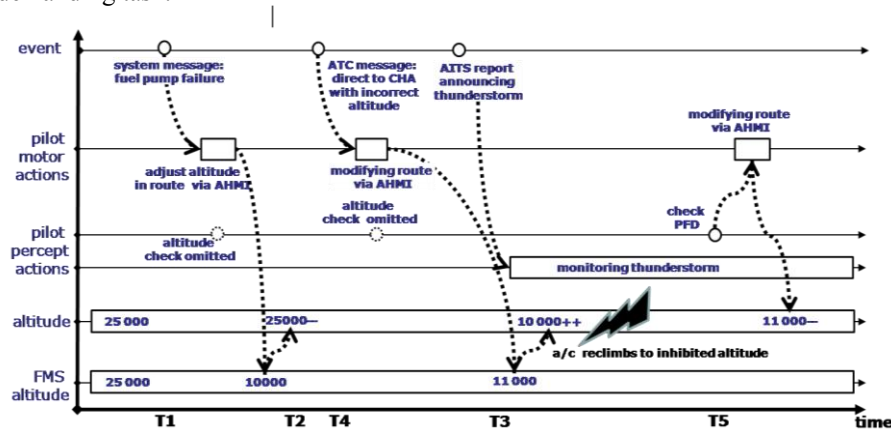


Fig. 8: Error is not recovered due to Cognitive Lockup

## 6 Summary

In this paper we have presented a cognitive model of pilot behaviour that simulates interaction with cockpit systems and predicts pilot errors due to Learned Carelessness and Cognitive Lockup. We described detailed hypotheses on errors and error recovery in the form of simulation traces that have been derived by executing the model in a virtual simulation environment. The next step in the reported research is to compare the model generated traces with traces of human pilots recorded in the same simulation environment.

The work described in this paper is funded by the European Commission in the 7<sup>th</sup> Framework Programme, Transportation under the number FP7 – 211988.

## References

1. Anderson J.R.: Learning and Memory. John Wiley & Sons, Inc. (2000)
2. Edwards, E.: The Emergence of Aviation Ergonomics. In: Wiener & Nagel (eds.): Human Factors in Aviation. San Diego, CA: Academic Press (1988)
3. Kerstholt, J.H.: Dynamic Decision Making. Universiteit of Amsterdam (1996)
4. Osterloh, J.-P., Lüdtkke, A.: Analyzing the Ergonomics of Aircraft Cockpits Using Cognitive Models. In: Proceedings of the 2nd International Conference on Applied Human Factors and Ergonomic (AHFE) 14.-17. July 2008, Hrsg.: Karowski, W.; Salvendy, G., USA Publishing, Las Vegas, Nevada, USA (2008)
5. Rasmussen J.: Skills, Rules, Knowledge: Signals, Signs and Symbols and other Distinctions in Human Performance Models. In: IEEE Transactions: Systems, Man and Cybernetics, SMC-13, pp. 257–267 (1983)
6. Sherry, L., Polson, P., Feary, M. and Palmer, E.: When Does the MCDU Interface Work Well? International Conference on HCI-Aero, Cambridge, MA (2002)
7. Dekker, S.: Failure to Adapt or Adaptions that fail. In: Applied Ergonomics, Vol.34/3 (pp.233-238). London, UK: Reed Elsevier Science Ltd (2003)
8. Lüdtkke, A., Cavallo, A., Christophe, L., Cifaldi, M., Fabbri, M., Javaux, D.: Human Error Analysis based on a Cognitive Architecture. In: Reuzeau, F., Corker, K., Boy, G. (eds.) Proceedings of HCI-Aero, pp. 40-47, Cépaduès-Editions, France (2006)
9. Liefvooghe, B., Barrouillet, P., Vandierendonck, A., Camos, V.: Working Memory Costs of Task Switching. Journal of Experimental Psychology: Learning, Memory, & Cognition, 34, 478-494 (2008)
10. Grossmann, T., Balakrishnan, R.: Pointing at Trivariate Targets in 3D Environments. In CHI'04: Proceedings of SIGCHI. ACM Press, pp. 447-545 (2004)
11. Frey, D., Schulz-Hardt, S.: Eine Theorie der Gelernten Sorglosigkeit. In H. Mandl (Hrsg.), 40. Kongress der Deutschen Gesellschaft für Psychologie. Göttingen, Seattle: Hogrefe Verlag für Psychologie, pp. 604-611 (1997)
12. Newell A. Unified Theories of Cognition, Harvard University Press; Reprint edition. (1994)
13. Anderson J.R. Rules of Mind. Hillsdale, NJ: Lawrence Erlbaum Associates. (1993).
14. Newell A., Rosenbloom P.S. and Laird J.E. Symbolic Architectures for Cognition. In: M.I. Posner (ed), Foundations of Cognitive Science, pp. 93-131. Cambridge, MA: MIT Press. (1989).
15. Anderson J.R., Bothell D., Byrne M.D., Douglass S., Lebiere C. and Qin Y. An integrated theory of the mind. Psychological Review, 111(4), pp. 1036-1060. (2004).
16. Corker K.M. Cognitive models and control: Human and system dynamics in advanced airspace operations. In: N. Sarter and R. Amalberti (eds), Cognitive Engineering in the Aviation Domain, pp. 13–42. Mahwah, NJ: Lawrence Erlbaum Associates. (2000).
17. Freed M. Simulating Human Performance in Complex, Dynamic Environments. PhD thesis, Northwestern University. (1998).
18. Wray R. and Jones R. An introduction to Soar as an agent architecture. In: R. Sun (ed.), Cognition and Multi-agent Interaction: From Cognitive Modeling to Social Simulation, pp. 53–78. Cambridge University Press. (2005).
19. Zachary W., Santarelli T., Ryder J., Stokes J. and Scolaro D. Developing a multi-tasking cognitive agent using the COGNET/iGEN integrative architecture. In: Proceedings of 10th Conference on Computer Generated Forces and Behavioral Representation, pp. 79-90. Norfolk, VA: Simulation Interoperability Standards Organization. (2001).
20. Florian Frische, Tomasz Mistrzyk, Andreas Lüdtkke: Detection of Pilot Errors in Data by Combining Task Modeling and Model Checking. INTERACT 2009: 528-531