	HUMAN Model-based Analysis of Human Errors during Aircraft Cockpit System Design	

D4.5 – Description of the improvements of the cognitive model and virtual simulation platform

Project Number:	211988
Nature:	Deliverable
Classification:	Public
Version:	V0.3
Parts & Classifications:	Main document (Public)
Work Package(s):	WP3, WP4
Document Timescale:	Project Start Date: March 1, 2008
Start of Document:	T0+30
Final version due to:	T0+31
Time now:	T0+31
Issue Date (dd/mm/yyyy):	14/09/2010
Compiled:	Jan-Patrick Osterloh (OFF)
Authors:	Jan-Patrick Osterloh (OFF)
	Joris Janssen, Tina Mioch (TNO)
Technical Approval:	Jan-Patrick Osterloh (OFF)
Issue Authorisation:	Andreas Lüdtke (OFF)

DISTRIBUTION LIST		
Copy type[1]	Company and Location	Recipient
T	HUMAN consortium	All HUMAN Partners

[1] Copy types:
M = Master copy,
E = Email,
C = Controlled copy (paper),
D = Electronic copy on disk,
T = TeamSite (Sharepoint)

RECORD OF REVISION			
Version	Date (dd/mm/yyyy)	Status Description	Author
V0.1	23/08/2010	Initial Version	JPO
V0.2	02/09/2010	Added Input	JPO
V0.3	08/09/2010	Added Input	JJ

1 Table of Contents

2 Abbreviations and Definitions

Abbreviation	Definition
CASCaS	Cognitive Architecture for Safety Critical Task Simulation
VSP	Virtual Simulation Platform
PF	Pilot Flying
PM	Pilot Monitoring / Pilot Non Flying
AHMI	Advanced Human Machine Interface

3 Introduction

This document describes the improvements of CASCaS after the first cycle. D4.3 describes the analysis of the results of the first cycle. Based on this, new hypotheses have been derived and new requirements have been specified. The architecture of the VSP has not been changed for the 2nd cycle, and is as in D3.4 described. For completion of this document, the architecture description is repeated in the following section. Section 3.2 describes the new requirements, taken from D4.3. Chapter

3.1 Architecture of the VSP

Figure 3.1 shows the architecture of the VSP in HUMAN:

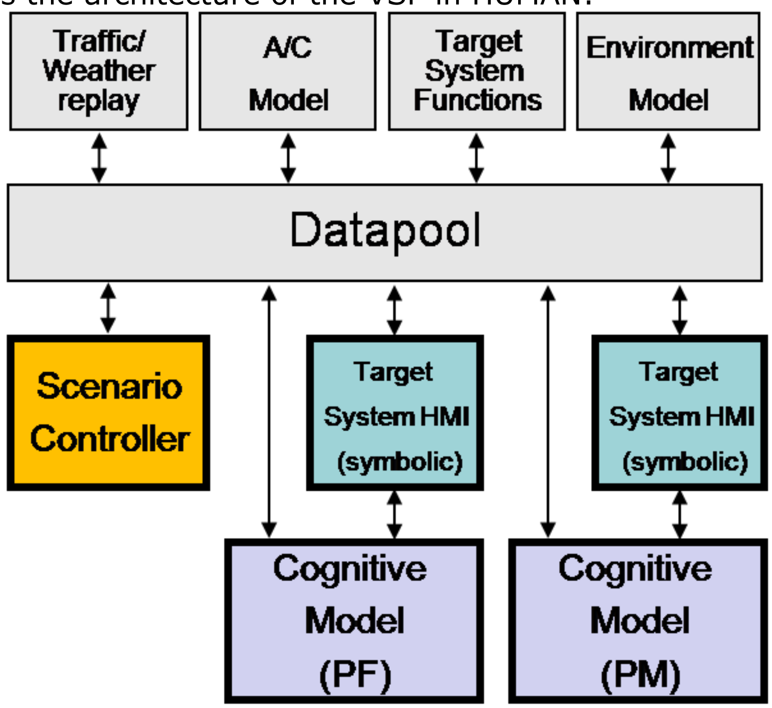


Figure 3.1: HUMAN Simulation Platforms Architecture

In the following, the modules of the VSP as depicted in Figure 3.1 should be described. All grey modules of Figure 3.1 are of general applicability, and a detailed description of them will be in D2.4 and D2.8. The other modules, depicted in colours and a bold box, are HUMAN specific and will be described in this document in more detail.

The *traffic/weather* module will replay pre-recorded traffic and weather (thunderstorms), according to the scenario (see scenario controller). The recoding of the traffic and the weather will be done by DLR, as part of the scenario design. The module will be maintained by DLR.

The scenarios which are flown by the cognitive models (and the human pilots on the PSP) are controlled by the *scenario controller*. This includes mainly

- starting the simulation software (A/C model, environment, datapool, traffic/weather replay, ...),
- starting the cognitive models (one for PF and one for PM),
- determining the end of scenario and start the cogn. models learning phase,
- stopping the simulation software and the cognitive models,
- restarting everything in batch mode.

For further discussion of the Scenario Controller, see D3.4. This module is developed by TNO in cooperation with DLR and OFF.

The *A/C module* simulates the aircraft behaviour (e.g. speed, pitch, roll), according to the physical performance of the aircraft and its actual parameters (e.g. thrust, flaps, slaps) with respect to external conditions, like actual weather conditions (e.g. wind), or the terrain, which are controlled by the *environment model*. Both modules have been developed and are maintained by DLR. The *target system* (main system that will be investigated in HUMAN) is the AHMI (Advanced Human Machine Interface), the User Interface for a 4D Flight Management System, called AFMS (Advanced Flight Management System). The AFMS functionality is represented as "Target System Function" module in Figure 3.1. For the VSP, the AHMI will be replaced by a *symbolic AHMI (SAHMI)*, which allows the Cognitive Model to interact with the AFMS. This is needed, as the Cognitive Model has no real eyes and hands, and thus cannot interact with the real graphical user interface itself. The AHMI and the AFMS have been developed by DLR, and the symbolic AHMI will be developed by BCH.

All modules of the simulations platforms (VSP and PSP) communicate with each other via a dedicated module, the *datapool*. The datapool not only passes messages between the modules, but is also the "flight recorder" (Black Box), which records all information that is necessary to replay the flight.

Last but not least, the VSP consists of two Cognitive Models, one for the pilot flying (PF) and one for the pilot monitoring (PM). In HUMAN, a cognitive model is understood as the combination of the pilot's knowledge (the normative behaviour plus additional knowledge needed to fly an A/C) and the cognitive architecture that interprets this knowledge in order to achieve a certain goal, e.g. flying the A/C in a scenario. In order to derive the normative behaviour, a task analysis is needed.

3.2 Requirements for the VSP – 2nd Cycle

ID	Requirement	Hypotheses/ EPM	TP	Effort	Priority (interest)	Req. Status
DA01	The simulation platform must include the navigation display	H06	OFF, DLR	medium	medium	Accepted
DA02	The cognitive model must include the navigation display and the windows in their tasks/scanning patterns	H06, H10	OFF, TNO	medium	medium	Accepted
DA03	If a value can be found on two different displays, the cognitive model should choose between them, based on the effort	H06	OFF	medium	medium	optional
DA04	The model should have a more realistic scanning behaviour: a) It should use 2-series patterns instead of 3-series patterns	H06/H08 + H01	OFF	high	medium/ necessary	Accepted
DA05	The percentage of time per AOI should take the changes during the flight phases into account	H10/H12	OFF	medium	medium	Accepted
DA06	The uplink procedure has to be implemented more realistic, in order to have similar execution times as the pilots - Visual search at the beginning, - Better checks after generate - Waiting for results of clicks	H14/H19	OFF, TNO	Medium to high	low/ necessary	Accepted
DA07	The cognitive model should degrade monitoring task during uplink task	H27	OFF	low	low	optional
DA08	CASCaS must implement Learned Carelessness on the Associative Layer in situations as described in the HLTC for LC	LC	OFF	high	high	Accepted
DA09	CASCaS must implement Cognitive Lockup on the Cognitive Layer in situations as described in the HLTC for CL	CL	TNO	high	high	Accepted
DA10	CASCaS must implement Selective Attention as described in the description of SA in situations as described in the	SA	OFF	medium	medium	Accepted

	HLTC for SA					
DA11	CASCaS must implement Limitations in Working Memory for situations as described in the HLTC for CL	LWM	OFF	high	medium	Accepted

4 Improvements for the 2nd cycle

The cognitive model consists of two parts: First, the normative behaviour that is needed to fly the scenarios, and second, the cognitive architecture CASCaS, which interprets the normative behaviour during the simulation. The requirements have therefore been divided into the requirements that improve the normative behaviour, and requirements that improve CASCaS.

4.1 Procedural Improvements

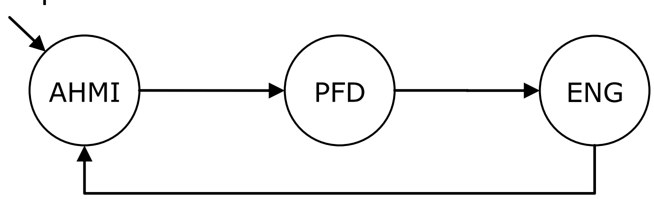
ID	Requirement	Hypotheses/ EPM	TP	Effort	Priority (interest)	Req. Status
DA01	The simulation platform must include the navigation display	H06	OFF, DLR	medium	medium	Accepted
DA02	The cognitive model must include the navigation display and the windows in their tasks/scanning patterns	H06, H10	OFF, TNO	medium	medium	Accepted
DA04	The model should have a more realistic scanning behaviour: b) It should use 2-series patterns instead of 3-series patterns	H06/H08 + H01	OFF	high	medium/ necessary	Accepted
DA05	The percentage of time per AOI should take the changes during the flight phases into account	H10/H12	OFF	medium	medium	Accepted
DA06	The uplink procedure has to be implemented more realistic, in order to have similar execution times as the pilots - Visual search at the beginning, - Better checks after generate - Waiting for results of clicks	H14/H19	OFF, TNO	Medium to high	low/ necessary	Accepted
DA07	The cognitive model should degrade monitoring task during uplink task	H27	OFF	low	low	optional

4.1.1 Navigation Display (DA01/DA02)

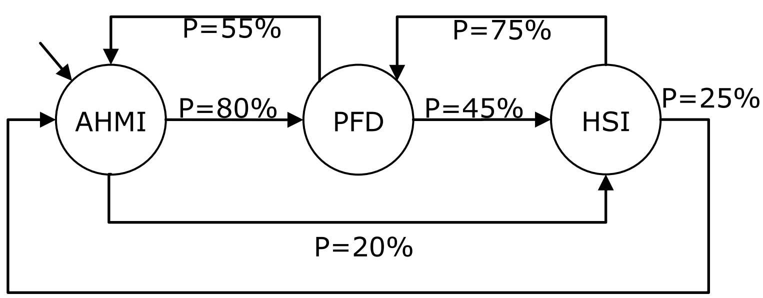
Because of licensing issues, the display cannot be provided as software for the VSP, anyway, the data that is displayed on the HSI (Horizontal Situation Indicator), or NAV-Display is available in the Datapool, and thus a virtual display can be setup in the VSP. The HSI displays data on navigation, mainly ILS and VOR information. This information is redundantly displayed on the PFD and AHMI. Therefore we re-allocated the glideslope and localiser information to the NAV display. In addition, the distance to the Airport has been added to as information on the NAV, instead on AHMI.

4.1.2 Scanning Behaviour (DA04)

The procedure for the scanning behaviour has been improved. In the 1st cycle, the monitoring was implemented as a state automaton that changes in the following sequence:



As the main focus of HUMAN is not on the monitoring task, and because of limited resources for modelling procedures, we decided to model the monitoring task probabilistic. In addition, the Engine Display (ENG) has been replaced with the HSI. Therefore the monitoring automaton has been extended to the following:



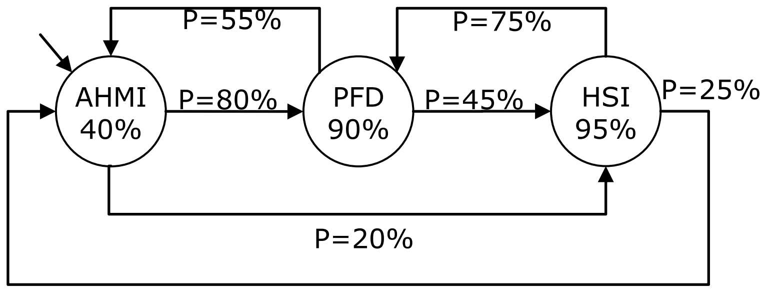
The most important change is that there is a probabilistic choice for each change of AOI in the scanning pattern. The transitions between AHMI and HSI are less probable as between the direct neighbour instruments, as described in D4.3. The probabilistic values have been derived from D4.3, Figure 3 (normalised to 100%). The scanning can be interrupted in each step by a non-monitoring task, e.g. an uplink.

The results of the 2nd cycle will show if this lead to a more realistic monitoring behaviour.

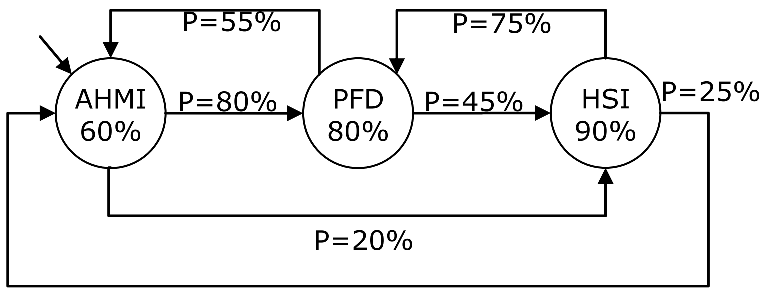
4.1.3 Flight Phases (DA05)

In order to implement the change of attention allocation from AHMI to PFD and NAV during the flight phases, as well as the percentage of time per AOI, we extend the scanning behaviour from above again, by adding the possibility $p_{skip}(AOI)$ to not perform the monitoring of *AOI* in this step (see $p_{skip}(AOI)$ below the AOI name; e.g. in Cruise mode, p_{skip}(AHMI)=40%)

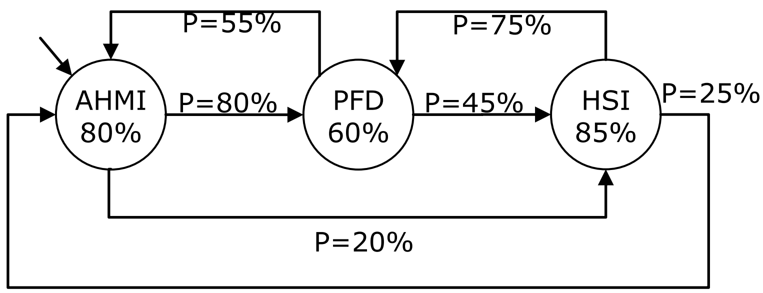
Cruise:



Approach:



Final Approach:



4.1.4 Uplink Procedure (DA06)

For the 2nd cycle, the uplink procedure will be modelled more detailed. We specified multiple checks, where we expect Learned Carelessness (see also D4.6).

In the following, the uplink procedure and its subtasks is described. Current describes the action that is performed for a certain goal (Goal name). In the Condition column, a pre- or post condition is specified. In case of a post-condition (keyword "Post"), the comment column contains an action that has to be performed before the condition is evaluated. If the condition is true, the next goal is the one specified in the "True Action" column, else the next goal in the "False Action" will be used.

Current Number	**Goal Name**	**Condition (Post = Post-Condition after action performed)**	**True Action**	**False Action**	**Comment**
1	X	Uplink is send, i.e. message box with uplink is displayed on AHMI.	2	1	X = current task (e.g. monitoring)
2	Handle uplink	Ensure the following configuration: North up, horizontal view, route mode	7	3|4|5	
3	Change orientation	Post: North up	2	3	Click with mouse on orientation button
4	Change view	Post: Horizontal view	2	4	Click with mouse on view button
5	Change mode	Post: Route mode	2	5	Click with mouse on mode button until desired mode achieved
6	Reject Uplink	-	END	END	Click on Reject button on message Box. Procedure finished
7	Check first waypoint	1st wp of cstr_list before a/c pos	8	6	Includes Zooming
8	Check end of trajectory	Last wp of cstr_list is on runway	9	6	Includes scrolling to last wp and zooming
9	Generate trajectory	Post: trajectory generated	10	9	Click on "Load+Gen"

					Button
10	Change View	Post: Vertical View	11	10	Click with mouse on view button
11	Check CFL acceptable	CFL == DESIRED_CFL	12	6	
12	Check INTerception altitude correct	INT==INT(Airport)	13	6	
13	Check trajectory on runway altitude	Altitude_last_wp==runway_elevation	14	6	Includes zooming/scrolling
14	Change view	Post: Horizontal View	15	14	Click with mouse on view button
15	Negotiate Trajectory	Post: Trajectory accepted	16	15	Click with mouse on "Send to ATC" button
16	Engage Trajectory	Post: Trajectory	END	16	Click with mouse on "Engage" button

Some of the checks require also that the pilot will scroll and zoom to a certain level. In the model, we will not model this directly, but only in terms of time that is needed to perform this. This assumption can be made, because the effort is needed for the Learned Carelessness implementation.

4.1.5 Monitoring during Uplink (DA07)

In the uplink procedure for the 1st cycle, the model was allowed to monitor the other instruments at certain points in the procedure (between generation, negotiation, and engagement). In the data this behaviour has not been observed, therefore we will interrupt the monitoring during the uplink handling. After the uplink has been performed, the monitoring will be resumed.

4.2 CASCaS Improvements

ID	Requirement	Hypotheses/ EPM	TP	Effort	Priority (interest)	Req. Status
DA03	If a value can be found on two different displays, the cognitive model should choose between them, based on the effort	H06	OFF	medium	medium	optional
DA08	CASCaS must implement Learned Carelessness on the Associative Layer in situations as described in the HLTC for LC	LC	OFF	high	high	Accepted
DA09	CASCaS must implement Cognitive Lockup on the Cognitive Layer in situations as described in the HLTC for CL	CL	TNO	high	high	Accepted
DA10	CASCaS must implement Selective Attention as described in the description of SA in situations as described in the HLTC for SA	SA	OFF	medium	medium	Accepted
DA11	CASCaS must implement Limitations in Working Memory for situations as described in the HLTC for CL	LWM	OFF	high	medium	Accepted

4.2.1 Visual Percept on multiple locations (DA03)

This optional requirement has not been implement.

4.2.2 Learned Carelessness (DA08)

As described in D3.4, Learned Carelessness (LC) has been implemented as association learning from goals to short-term knowledge in the memory:

> "To model LC, there is an architectural mechanism that automatically creates associations between items of declarative knowledge and goal items. Every time an item is created in memory a contextual association between this item and the current goal is created. This expresses that this item seems to be relevant in the current context, i.e. the current goal. If the item that shall be written already exists, it will be reactivated. If in this case there is also already a contextual association to the current goal, the strength of this association will be increased by a constant factor l - the association learn rate.

As a result every goal item can have many associations to declarative items. These are the items that have somehow been used during the time this goal was active. The relative strength between two associations originating from the same goal reflects the relative frequency of occurrence of the associated items during the time this goal was active.

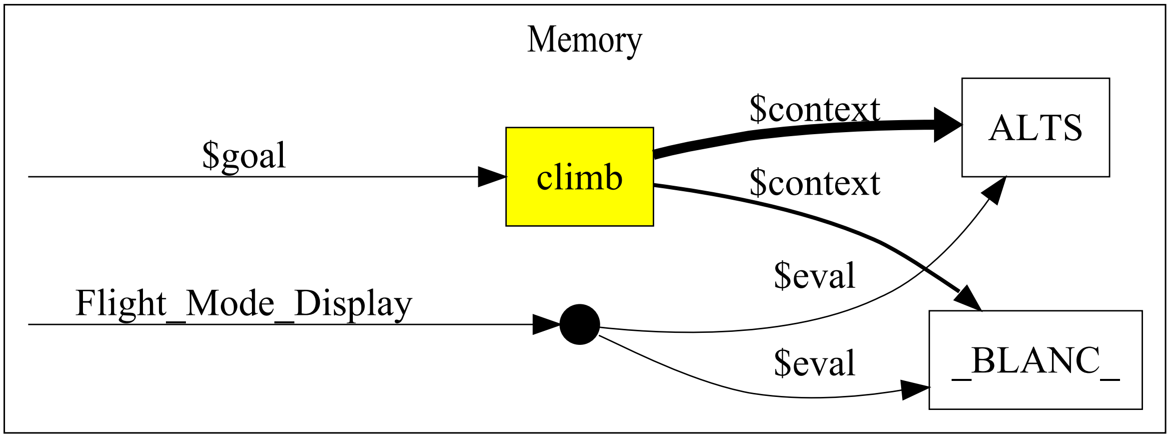


Figure 4.1: Contextual associations to different evaluation items

Take for example the item for the flight mode display (cf. D3.4, section 5.2.1.4.1.3), which has two evaluation items. When the flight mode display is watched during the goal *climb*, the percept component will write the values to memory. In this process contextual associations from the goal to the evaluation items will be created respectively be strengthened. This can be seen in Figure 4.1.
In the current version the process described above is the only process that creates contextual associations. This might be changed in later versions.

Let the association from item i to item k be $r_{i,k}$,
the strength of association r be $str(r)$,
the set of all goal items be G,
the number of associations from any goal to item i be $n_i^{in} = |\{r_{g,i} \mid g \in G\}|$.

With these parameters, the described construct of contextual associations is used to calculate the $S(t,i)$ factor in the activation calculation[2].

$$S(t,i) = \frac{1}{n_i^{in}} * \sum_{k=1}^{|G|} (str(r_{g,i}) * A_{g_i}) \quad (1)$$

[2] Activation A of item i at time t is calculated with:

$$A(t,i) = \beta - decay * ln(t - t_{w_i}) + S(t,i)$$

This just means, that the activation of each goal is distributed along the associations to all associated items. Currently this is only done for contextual associations. As these associations currently are only originating from goal items, the spreading of activation will start at the goal items and immediately stop after one step.
Here the goals serve as sources of activation. At each time only the currently active goal in the goal module has an activation value unequal to zero. Every time a new goal is selected the goal item receives a predefined activation value and the activation of the previously selected goal is set back to zero. In practice this saves us the summation over all goals, while just using the active one."

This allows modelling LC as activation spreading of certain values, which are written very often when a certain goal is active, when this goal becomes activated again. First tests showed that our model produces too much LC. Thus we added an effort factor in the learning mechanism, which actually means that the strength of an association r ($str(r)$) is only increased by the factor l, if the effort for retrieving this information is above a threshold $eff_{percept}$. As described in D4.6, we have also the hypothesis that not only the effort and the frequency have an influence on LC, but also the risk associated with a certain task. The risk factor has currently not been modelled in the learning mechanism. The risk can be modelled in the procedure as explicit knowledge, e.g. by adding further rules with conditions that are only valid in risky situations. Because these are not frequently used, no LC will arise here.

4.2.1 Cognitive Lockup (DA09)

Cognitive lockup is the tendency to deal with disturbances sequentially (Moray and Rotenberg, 1989). This tendency is implemented within the Cognitive Layer of CaSCaS.

As described in D3.4, we hypothesize that the factors '*number of* tasks' and '*stress'* play a role in cognitive lockup. In the following, these two factors are called '*task set switches'* and '*time pressure*' respectively, corresponding to Neerincx (2003) theory of cognitive task load and cognitive lockup.

The first factor that influences cognitive lockup in our model is **time pressure**. Time pressure is the perception that time is scarce. According to Beevis (1999) people experience time pressure when the time required to execute tasks is more than 70% of the total time available for the tasks. Beevis (1999) suggested that people experience high time pressure when 85% of the available time is required to execute the tasks. In this case performance is often impaired in that some tasks are not (well) executed.

To implement the factor *time pressure*, the goals (see D3.4) are extended with extra information. In each goal a deadline formula and a duration is added, to fulfil the extra requirements.

The deadline formula describes the way to calculate the final desired moment for a chosen goal to be finished. This can either be a point in time calculated from the time this goal becomes active (e.g. 5 seconds from now) or a specific moment in the scenario (e.g. before the approach briefing). This formula is represented in a CLIPS expression. The following CLIPS expression shows a representation of a deadline 5 seconds from the moment the goal becomes active.

```
(+ (send [currentTime] get-TimeValue) 5000)
```

Besides the deadline, each goal has a duration. This value is an approximation of the time that is needed to execute this goal. Using this value, the time pressure can be calculated. In the model, time pressure (TP) is implemented as a discrete value (it can be either high (1) or low (0)).

The second component that influences the occurrence of cognitive lockup are **task set switches**. The task set switches (TSS) are implemented as the number of active goals in the goal agenda. In Figure 4.2 for example, at T=4, this number is two (Goal 1 and 2).

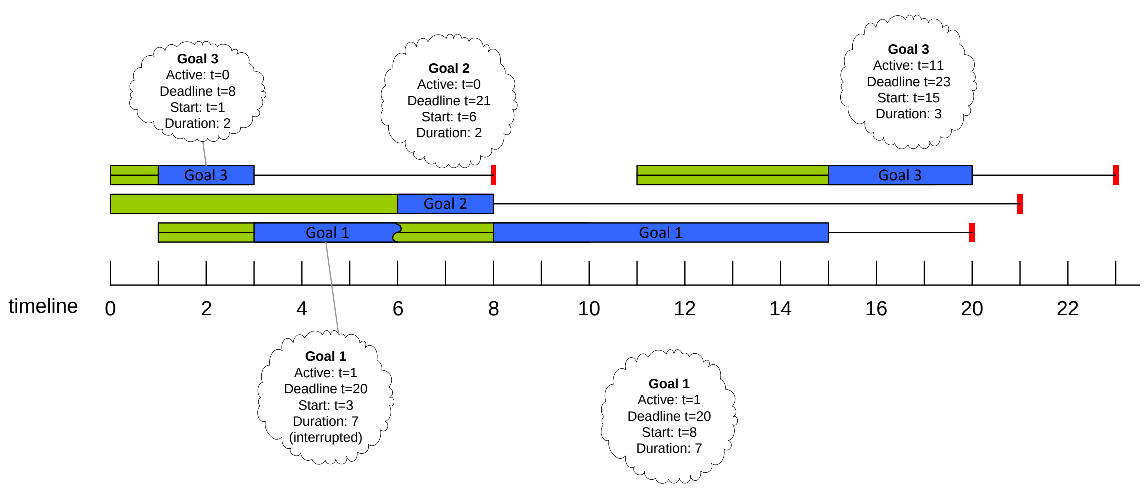


Figure 4.2: Example of goals in the cognitive layer.

In Figure 4.3Figure 4.3: Visual representation of goals on the cognitive layer. the current executed goal is highlighted. On top of the priority of this goal extra switch costs are indicated. These switch costs consist of two components: the number of active goals and the time pressure.

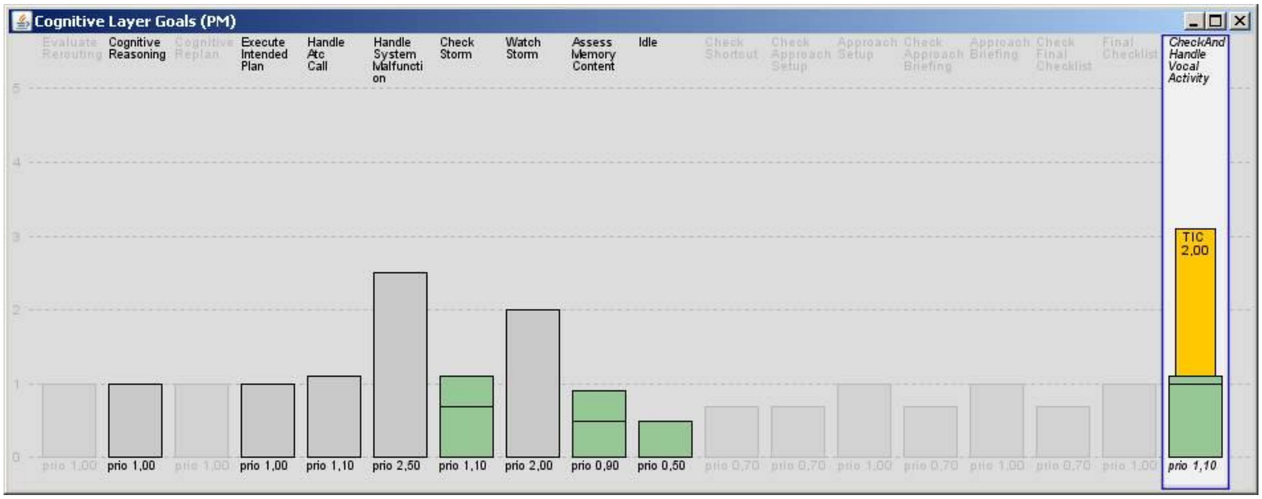


Figure 4.3: Visual representation of goals on the cognitive layer.

TSC = StartTSC + f_{tss} * TSS + f_{tp} * TP

Each of the factors are normalized using a parameter *f*. The final value for these parameters is determined experimentally. By adding the TSC to the priority of the executed goal, the goal selection module has the tendency to postpone the switch of tasks.

4.2.2 Selective Attention (DA10)

The implementation of Selective Attention has not been tested in the 1st cycle. Therefore the implementation of this EPM is as it has been described in section 5.2.1.2.2 in D3.4 (Description of the Virtual Simulation Platform).

4.2.3 Limitations in Working Memory (DA11)

This EPM is implemented as forgetting in memory due to activation decay, as described in section 5.2.1.4 in D3.4 (Description of the Virtual Simulation Platform).

5 Summary

In order to improve the behaviour of the cognitive model after the first cycle, we analysed the data and specified improvements for the model (cf. D4.3). We implemented several improvements in the normative behaviour of the model, in order to make the tasks of the pilot more realistic. In addition, the implementations of the EPMs LC and CL have been improved in order to improve the predictions. For LC the "effort" parameter has been implemented as addition to the "frequency" parameter. For CL the parameters for number of tasks and stress have been further formalised and implemented.

6 References

Beevis, D., Bost, R., Doering, B., Nordø, E., Oberman, F., Papin, J., et al., 1999, *Analysis techniques for man-machine system design, crew system ergonomics*, Ohio

Moray, N., & Rotenberg, I., 1989, Fault management in process control: eye movements and action, *Ergonomics* 32 (11), 1319-1342

Neerincx, M., 2003, Cognitive modeling of pilot errors and error recovery in flight management tasks, *In E. Hollnagel (Ed.), Handbook of cognitive task design (pp. 283–306). CRC.*