



HAL
open science

Multi-Criteria Evaluation of Aircraft Cockpit Systems by Model-based Simulation of Pilot Performance

Andreas Lüdtke, Jan-Patrick Osterloh, Florian Frische

► **To cite this version:**

Andreas Lüdtke, Jan-Patrick Osterloh, Florian Frische. Multi-Criteria Evaluation of Aircraft Cockpit Systems by Model-based Simulation of Pilot Performance. Embedded Real Time Software and Systems (ERTS2012), Feb 2012, Toulouse, France. hal-02189907

HAL Id: hal-02189907

<https://hal.science/hal-02189907v1>

Submitted on 20 Jul 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-Criteria Evaluation of Aircraft Cockpit Systems by Model-based Simulation of Pilot Performance

Andreas Lüdtke, Jan-Patrick Osterloh, Florian Frische

OFFIS Institute for Information Technology, Escherweg 2, 26121 Oldenburg, Germany
{luedtke, osterloh, frische}@offis.de

ABSTRACT

The paper describes a new methodology for simulation and evaluating human-machine interaction in aircraft cockpits with the objective to assess the impact of a new system design on overall flight crew performance. The methodology shall allow to assess multiple performance criteria and to derive needed design improvements more accurately and earlier (in the development process) and with reduced effort compared to existing approaches. The main technical enabler is a cognitive model of pilot behaviour. The focus of the paper is on the application of the new methodology for simulating and evaluating the interaction with a 4D Advanced Flight Management System.

Keywords

Human-Machine Interaction, Aircraft Cockpits, Cognitive Modelling, Human Error

INTRODUCTION

Many systems that incorporate Embedded Systems interact with human operators. This holds for aircraft, cars and ships and even for autonomous aerial vehicles. In systems like these the human-machine interaction plays a major role for guaranteeing transparency of the systems' internal processing. It is a prerequisite for the users' trust in the functionality and services.

In the European project HUMAN¹ we developed a new methodology for evaluating human-machine interaction in aircraft cockpits with the objective to assess the impact of a new system design (the target system) on overall flight crew performance. We considered temporal behaviour (e.g. reaction time), attention allocation and potential human errors as valuable evaluation parameters. The new methodology shall allow to assess these multiple criteria and to derive needed design improvements (incl. safety nets to recognize, prevent and recover from human errors) more accurately and earlier (in the development process) and with reduced effort compared to existing approaches. The main technical enabler is a cognitive model of crew behaviour, called CASCaS (Cognitive Architecture for Safety Critical Task Simulation). Cognitive models are a means to make knowledge about characteristic human capabilities and limitations readily available to system designers in an executable form. They have the potential to automate parts of the analysis of human factors because they offer the

opportunity to simulate the interaction with cockpit systems under various conditions and to predict cognitive processes like the assessment of situations and the resulting choice of actions including erroneous actions. In this way they can be used as a partial "substitute" for human pilots in early development stages when design changes are still feasible and affordable.

Model- and simulation-based approaches are already well-established for many aspects of the study, design and manufacture of a modern airliner (e.g., aerodynamics, aircraft systems, engines). Our work extends the modelling approach to the interaction of flight crews with cockpit systems.

In HUMAN the cognitive pilot model has been developed and validated based on experimental data which has been recorded in flight simulator experiments with 16 human pilots.

For the development of the new methodology we started with a careful analysis of the process defined in the ARP 5056 documents. Then we assigned the tools developed in HUMAN to steps in this process. This paper describes the new methodology focusing on three new software tools: the cognitive architecture CASCaS, a procedure modelling tool called PED and a data analysis suite. We show how the new methodology has been applied to simulate and evaluate human-machine interaction with an Advanced 4D Flight Management System (AFMS) developed by the German Aerospace Center (DLR), Braunschweig.

RELATED WORK

The idea to use cognitive models to predict pilot performance including pilot errors is not new. The recently finished Human Performance Modelling (HPM) element within the System-Wide Accident Prevention project of the NASA Aviation Safety Program performed a comparison of error prediction capabilities of five models [2]: ACT-R, Air-MIDAS, D-OMAR, IMPRINT/ ACT-R and A-SA. It has been demonstrated that each of these models is able to predict some phenomena of pilot performance, but none of them covers a sufficiently complete set of predictions. With CASCaS we strive to consider exactly those cognitive processes that are relevant for design of safety critical systems in the aircraft domain. The innovative aspect of CASCaS is the prediction of human errors resulting from an interaction of (1) learned mental models (Routine Learning/Learned Carelessness), (2) actual limited cognitive performance and (3) safety nets in aircraft cockpit design (e.g. flashing indication, alerts and crew interaction).

¹ Model-based Analysis of Human Errors during Aircraft Cockpit System Design; www.human.aero

TARGET SYSTEM

Within HUMAN we applied the cognitive-model based methodology for the analysis of the interaction between the Pilot Flying and an Advanced 4D Flight Management System (AFMS) developed by DLR, Braunschweig, Germany. It supports onboard management of flight trajectories and the negotiation of trajectory changes with Air Traffic Control (ATC) via Data Link. The AFMS is complemented with an Airborne Human-Machine Interface (AHMI) which represents flight trajectories on a moving map with their status being graphically augmented by different colours and shapes. It provides a horizontal view and a vertical view. The horizontal view shows the lateral trajectory along its waypoints. The vertical view shows the altitude and timing profile of the active trajectory including constraints. The pilots also get information on the violation of altitude and timing constraints in case a modified flight plan uplinked by the ATC cannot be fulfilled by the aircraft.

This system reflects a recent trend of introducing new interaction technologies like graphical manipulation into the aircraft cockpit. In order to leverage the corresponding new air traffic management concept, efficient and robust flight procedures are needed. Safe operation of aircraft is based on normative flight procedures (standard operating procedures) called “Normative Activities”.

METHODOLOGY

Our methodology is similar to the process steps defined in the ARP (Aerospace Recommended Practice) 5056 design process for cockpit systems [10] and includes: 1. Requirements Capturing, 2. Flight Crew Concept of Operations, 3. Solutions Defined, 4. Detailed Design & Implementation, 5. Integrated Test & Evaluation.

The order of phases can vary. Each phase consists of several design sub-steps as shown in Figure 1. HUMAN provides support for human factor analyses in the steps highlighted in dark grey in Figure 1. These tasks can be grouped into three categories: *Task Analysis* (for the sub-steps “Perform preliminary task analysis” & “Update task analysis”), *Simulation* (for the task “Simulate prototypes” and “Evaluate configurations and prepare tests reports”) and *Evaluation* (for the tasks “Evaluate configurations and prepare tests reports” and “Evaluate results and produce reports”).

Within HUMAN, tools have been developed for each of these categories: PED (Procedure Editor) for Task Analysis, CASCaS for Simulation of human behaviour and an Analysis Suite for Evaluation. These tools can be applied to simulate answers to the questions “How long does it take to perform a flight task with the target system?”, “How long does it take to react to target system outputs or information?”, “Are visual events recognized?”, “How does the introduction of the target system change visual attention allocation?” and “Is the target system prone to routine errors?”.

In the following we show the details of the developed tools for Task Analysis, Simulation and Evaluation.

Requirements	Flight Crew Concept	Solutions	Detailed design & implementation	Integrated test & evaluation
Capture the requirements	Define System Configurations	Develop alternative configurations	Model and analyze solution (step 3)	Prepare test plan for cockpit system
Process the requirements	Perform preliminary task analysis	Create and analyze models	Prototype solution (step 3)	Test solution on A/C
	Define Preliminary post-development verification requirements	Create prototype	Define test plan	Evaluate A/C results and produce reports
		Prepare test plans	Implement test activities	Define guidelines for the flight manual
		Simulate prototypes	Evaluate results and produce reports	Define Guidelines for flight crew training
		Evaluate configurations and prepare tests reports	Perform critical design review and produce CDR report	
		Identify solution, and detailed requirement for development	Implement and integrate final solution	
		Perform preliminary design review and produce PDR report	Update final requirement	
		Update intermediate requirements	Update final post-development verification requirements	
		Update post-development verification requirements		
		Update task analysis		

Figure 1: Task View for Design Process for Cockpits Systems

Task Analysis

During Task Analysis the Normative Activities for interaction with the target system are modelled (and analysed subsequently) in form of a rule-based task model using a graphical editor. The rule based language is based on the well-known GOMS notation [12] and allows to formalize a mental representation of flight procedures. All rules consist of a left-hand side (IF) and a right-hand side (THEN) (Figure 2).

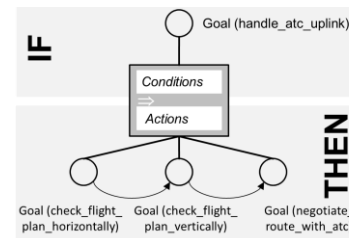


Figure 2: Rules for modelling flight tasks

The left-hand side consists of a goal and Boolean conditions on the current (memorized) state of the environment. The right-hand side consists of motor and percept operators (writing values and reading values in the simulated environment), memory-store operators as well as a set of partial ordered sub-goals.

Rules are connected via goal and sub-goals: the sub-goal of one rule is the top goal of another. In this way rules are used to build up a task tree as shown in Figure 3. It describes a task as a temporally ordered hierarchy of goals (e.g. handle_atc_uplink), sub goals (e.g. check_flight_plan_horizontally, check_flight_plan_vertically) and actions (e.g. perceive a/c position, press VERT button). The task tree shows for each sub goal the involved mental and behavioural steps.

The model foresees a special rule type called reactive rules that are used for immediate behaviour (as opposed to goal-based behaviour) in order to instantly react to events in

the environment – these rules contain no goal on their left-hand side.

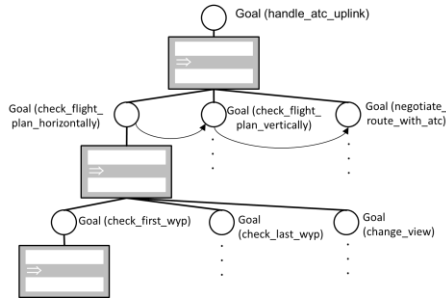


Figure 3: Format of CASCaS task tree

On a sub-symbolic level annotations are added to the task tree to capture priorities for (sub-)goals, duration for actions as well as workload values for actions.

In HUMAN we developed the editor PED for modelling and analysing task trees. PED allows to compute temporal and workload performance criteria. Temporal analysis allows the user to assess the time required to accomplish a task by following whatever path including the possibility to identify the shortest path (best case) and the longest path (worst case). A mean duration to accomplish the task is also calculated (average case). Workload analysis allows the user to visualise the different types of workload associated to a procedure. The PED workload analysis is based on McCracken & Aldrich’s [9] workload category system. It includes visual, auditory, cognitive and psychomotor workload categories. Every element of the task has to be assigned to one of these categories by the modeller. Within the categories elements have to be rated on a scale from 1 (low) to 7 (high). McCracken & Aldrich defined the semantics of these values by giving examples in form of associated action types. For example, 1.0 for visual workload means: something (e.g. the onset of a message) has to be “visually detected”; 5.9 means: a symbol has to be read.

In order to compute duration and workload values PED includes an animation function which allows the user to step through a procedure. At decision points either the user has to make a choice or PED choses non-deterministically. During this step-by-step execution the tool computes the associated duration and workload for the executed path of the task tree.

Additionally, PED offers the functionality to compute every possible execution path of the task tree automatically using model checking technologies.

Task analysis results (in terms of the above parameters) allow the user to assess and compare procedures with regard to duration and induced workload.

Simulation

After the task model has been edited and analysed it is possible to simulate the task in dynamic scenarios. Simulation of the task is performed by means of the cognitive architecture CASCaS. CASCaS can be understood as an “interpreter” that executes task models in a psychological plausible way. Within CASCaS we

implemented capabilities like sophisticated perception (visual focus and peripheral view) [7], attention mechanisms (bottom-up and top-down attention resulting in simulated eye-movements) [8], rule-based knowledge processing (including multi-tasking), learning and crew interaction (e.g. [8], [6]). Figure 4 gives an overview on the modular structure of CASCaS.

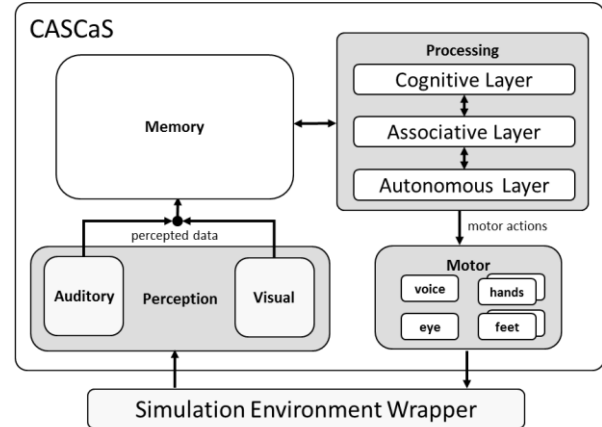


Figure 4: CASCaS Architecture Overview

CASCaS includes three layers of cognitive processing: autonomous layer, associative layer and cognitive layer. This layered architecture is derived from Fitts’ theory of learning stages [13, 14]. Fitts differentiates the degree of conscious control and the type of knowledge that is involved in cognitive processing. On the autonomous layer, humans act highly un-consciously with sensor-motor patterns (e.g. flying an aircraft manually). In familiar situations, humans consciously act based on previously learned rules on the associative layer, and in cases where no rules can be applied, new plans have to be derived on the cognitive layer, which is highly conscious behaviour. These levels are implemented in the Knowledge Processing component of CASCaS. In this paper, we will focus on the associative layer where rule-based task trees are executed.

The percept and motor component interact with a simulated environment by reading and manipulating external variables. Perceived variables are stored in memory where they can be retrieved by the knowledge processing component. The Simulation Environment Wrapper provides data for the percept and motor component by connecting CASCaS to different simulation back-ends. In HUMAN we connected CASCaS to the flight simulator software used by the DLR for experiments with human pilots.

On the associative layer of CASCaS tasks are executed within a cognitive cycle: the next goal to be processed is selected from a goal agenda (the set of goals that have not yet been fulfilled) taking into account a goal priority, then all rules containing the selected goal in their IF-part are collected, from this set one rule is selected, finally the selected rule is fired, which means that the THEN-part is executed: sub-goals are added to the goal agenda, percept actions are send to and performed by the percept component and motor actions are send to and per-formed by the motor

component. This cycle models goal-driven behaviour. It can be interrupted by percept-driven reactive behaviour. The models check at the beginning of every cycle if conditions of reactive rules are true. If this is the case, reactive rules are always preferred to regular rules.

For the percept component we modelled a visual peripheral view of 170 degree horizontal and 110 degree vertical around an optical axis (defined by the gaze direction of the eye) and a visual focus of 2 degrees (Figure 5). This field is projected on a virtual scene in order to compute which objects can be perceived by the model and which objects are out of view. Objects in focus can be read (e.g. the altitude can be read if the altitude bar on the PFD is in focus).

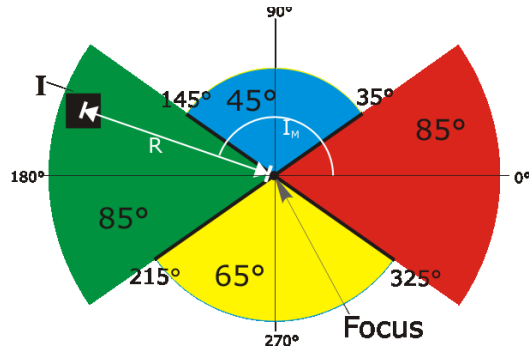


Figure 5: Model of visual focus and peripheral view

For objects in the peripheral view the model can only perceive discontinuity in space or time, e.g. the onset of a warning indication. A discontinuity in space represents a difference in a static property, like colour, brightness, form or orientation, e.g. a green dot in a set of red dots, or a circle in a set of quadrates. In contrast to this, a discontinuity in time (or dynamic discontinuity) denotes a dynamic change, like abrupt onset, flashing or moving of an object. For recognition of these discontinuities we implemented a process based on theories of *Selective Attention*. This process determines a detection probability based on the saliency of a stimulus, i.e. based on how much a stimulus differs from its surroundings. A flashing indication is best recognized on a solid black background. The detection probability decreases if the stimulus occurs in a colourful and dynamic context. More details can be found in [7].

A *routine learning process* is modelled in the memory component based on the psychological theory of Learned Carelessness by Frey and Schulz-Hardt [3]. We assume that pilots have mental models of how to interact with cockpit systems, like the AFMS. The mental model of flight procedures is initially formed based on Normative Activities acquired through handbooks and in simulator training sessions. Then during repetitive performance the mental model is modified by routine learning processes like Learned Carelessness. The theory 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. 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.

In CASCaS, routine learning has been implemented in form of association learning and activation spreading concepts similar to mechanisms in ACT-R [1]. Every memory item has got an activation value which is increased or decreased according to (1) write operations on these items and (2) according to the effort involved in corresponding percept actions. Each time a value is written to the memory by the percept component, an association between this memory chunk and the active goal is created, or if already created, the strength of this association is increased. The effort is computed as the average time that is needed to acquire the value via corresponding percept actions. Like the number of write operations also effort acts as an exhibiting factor: the higher the effort the more the association is increased.

Imagine that a memory item represents the information that the altitude of the last waypoint of a flight plan can either be correct or incorrect. Each time the model performs the goal to check the altitude of the last waypoint and perceives (by checking the vertical view of the AHMI) that it is correct, the corresponding activation is increased (taking into account also the effort of the corresponding percept actions on the vertical view). As said above, the resulting higher activation is dependent on the goal context. If the same value has been perceived several times and a defined activation threshold is exceeded then next time when the model performs the goal to check the altitude of the last waypoint the value will be retrieved from memory instead from the environment: the model exploits a learned regularity within the environment. More details can be found in [6].

The motor component can perform actions, like mouse movements, mouse clicks, hand movements or adjustment of physical instruments, e.g. thrust lever, or autopilot altitude knob. For each of these actions, timing is calculated based on a 3D version of Fitts' Law [4].

CASCaS allows predictions of (1) execution and reaction time, (2) attention allocation and (3) human error. The predictions result from specific interactions of the modelled cognitive processes as described in the following.

Execution Time: This predictive capability allows answering the question "How long does it take to perform a flight task with the target system?" While the model performs a flight task it selects and fires rules from the task tree (cf. Figure 3). By performing the cognitive cycle, one specific path in the tree is executed. The computed execution time is a combination of time to perform the percept actions, memory actions, rules and motor action on the path. In contrast to the execution time computation performed in PED, CASCaS allows to compute execution time in dynamic scenarios. Thus, the time can be computed taking into account specific environmental conditions. This includes task interruptions and thus, task interleaving.

Figure 6 shows sequential as well as parallel processing steps involved in flight task execution.

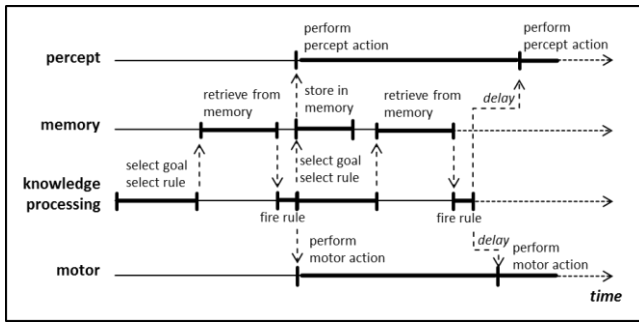


Figure 6: Composition of execution times (dotted arrows denote causal dependencies within the model)

The knowledge processing step “select goal/rule” needs information from the memory in order to check conditions on the left-hand side of rules. This dependency forces sequential processing of the corresponding memory retrieval. After a rule has been selected it is fired and the actions on the right-hand side are executed in parallel: information is stored in memory, percept actions and motor actions are performed. Consequently, the duration of the knowledge processing cycle is computed by: 50 ms for select goal & rule and fire rule + time for memory retrieval (for rule conditions). The next cycle, i.e. the selection of the next goal and rule can be done in parallel to the execution of the right-hand side actions (of the preceding rule). Delays may occur when the percept or motor component receive new actions from the new cycle while they are still busy performing actions from the preceding cycle.

The time for percept action is computed by: 170ms + 2ms per degree of the saccade + fixation time of 200±50. Time for motor action is computed based on Fitts’ Law [4].

Attention Allocation (Effectiveness of Safety Nets): This predictive capability allows answering the questions “Are visual events recognized?” and “How does the introduction of a new system change Visual Attention Allocation?” Processing of Normative Activities requires information from the environment. The rules contain percept actions on their right-hand sides in order to trigger necessary percept actions. The percept component moves the visual focus to the corresponding instruments in order to read the needed data. This results in gaze transitions that are dependent on the data requests from the Normative Activities. This is how we modelled *top-down attention allocation*. The routine learning mechanism has an essential influence on top-down attention allocation, too, because it replaces percept actions with memory retrieval and thus leads to reduced gaze transitions.

Bottom-up attention is modelled via the Selective Attention mechanism. The gaze transitions determine at every point in time which cockpit instruments or which visual events are within the visual field. The probability of event recognition is computed by the Selective Attention mechanism based on saliency attributes. Event recognition triggers a gaze transition to the source of the event. Consequently, attention allocation is influenced by the Normative Activities for target systems (top-down attention) and by the graphical

layout of the user interface (bottom-up attention). Thus, the two questions mentioned above are answered taking into account both types of attention allocation.

Reaction Time: A further question that can be answered is: “How long does it take to react to target system outputs or information?” Reaction time is the time from the onset of a visual event (e.g. the appearance of a message box on the AHMI showing a new ATC message) until the visual focus has been moved onto the visual event. Reaction time has three components: (1) time until the visual event appears in the peripheral view, (2) time until the event is recognized in the peripheral view and (3) time for eye-movement onto the event. Figure 7 shows an example; the small black circle depicts the visual focus, the surrounding ellipse depicts is the peripheral visual field and the letters A, B, C, D represent cockpit instruments.

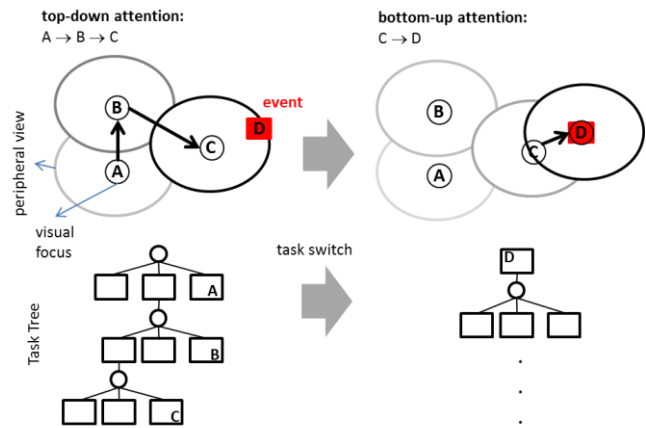


Figure 7: Reaction time to visual events

Step (1) depends on the current eye movements that are directed due to percept actions triggered by rules, i.e. in Figure 7 the attention is shifted from instruments A to B and then to C, as requested in the task tree on the bottom left of Figure 7. Only after the attention has been moved to instrument C, the event on instrument D appears in the peripheral view. Recognition of D in step (2) depends on the Selective Attention mechanism implemented in the percept component. If the model does not recognize the event e.g. due to the characteristics of the graphical context, steps (1) and (2) are repeated until the visual event is recognized or until the visual event has disappeared. If the event has been recognized, the time for step (3) is computed according to the formula 170ms + 2ms per degree of the saccade. The event may trigger a reactive rule and may thus result in a task switch as shown on the bottom right of the figure.

Human Error: CASCaS allows the prediction of human errors that result from an interaction of routine learning, actual limited cognitive performance and safety nets in aircraft cockpit design (e.g. flashing indication, alerts and crew interaction). Routine learning (as described above) can lead to omitting percept actions, thus the mental model about the state of the aircraft might deviates from the actual state.

Every time a goal is activated within the cognitive cycle, activation spreads from the goal to memory items via

learned associations. If the resulting activation values for a memory items are above a predefined threshold then this item can be retrieved from memory, without percept action. This replacement of a percept action with a memory retrieval can lead to the selection of a course of action that is not correct with regard to the Normative Activities (Learned Carelessness).

Safety nets like flashing indications on cockpit displays might be used by designers as a countermeasure against the effects of routine learning. The effectiveness of such a safety net depends on the cognitive performance, which is limited by the restricted cognitive resources. In the current state of implementation of the pilot model the most important limitation is the visual field with its Selective Attention mechanism for visual events in the peripheral view. Due to Selective Attention the model might miss a flashing indication which is intended as a safety net to counteract Learned Carelessness.

Evaluation

During the simulation operator performance data, as well as system and environment data is recorded. With regard to operator performance we record simulated hand and eye movements as well as selected goals, rules and working memory contents. The data format is a timestamp based discrete sequence of parameters allowing analysts to analyse the interaction between the operator and the system at every point in time. The simulation data is used to evaluate multiple criteria of the human-machine interaction. A target criterion could be e.g. reduced execution time for a specific task on a target system as an indicator of system efficiency. Typically, the evaluation is performed by comparing performance between different system setups, e.g. with and without a target system or between different design solutions of the same system. In HUMAN we analysed the interaction *with* target system only.

In order to support target system evaluation based on recorded data, a specific analysis suite has been developed. It supports analysis of the following performance criteria:

- **Gaze behaviour**, such as gaze distribution and duration of fixations (dwell time).
- **Temporal behaviour**, such as task execution time and reaction time.
- **Human error analysis**, such as detection of errors of omission and errors of commission.

The analysis is subdivided in three steps (Figure 8). The first step is the pre-processing of each input simulation data set. Each data set refers to a certain HMI solution. The pre-processing includes consistency checks of the input data in order to guarantee the correctness of the results. In addition, some analyses need application of data filters, such as the gaze analyses. E.g., a very important pre-processing step during gaze analyses is the definition of fixations (minimum length of uninterrupted glances on an AOI). In the second step, the analysis of each data set is performed individually. In the third step, the analysis suite compares the individual results on different levels of aggregation, e.g.

run vs. run or scenario vs. scenario. The results of the analysis are then written and plotted in result data sets and charts.

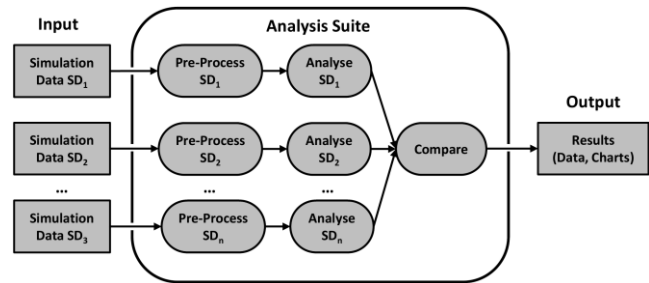


Figure 8: Process for analysis of operator performance supported by Analysis Suite

The output (text and charts) is generated full automatically and can be used by the analysts to evaluate the pros and cons of each HMI solution.

CASE STUDY

In this section, we will present a case study of applying the new methodology to investigate the effect of the AFMS on pilot performance during dedicated scenarios. Based on the methodology issues of the AFMS/AHMI with regard to demanding visual workload, long task execution times and potential human errors have been identified.

For the investigation of the target system we performed experiments with human pilot in a flight simulator owned by the DLR and with the pilot model. For the model-based experiments CASCaS has been connected to the flight simulator software within a Virtual Simulation Platform. In this way it was possible to simulate the same scenarios with both, the human subjects and the model.

The results of the model-based simulation, reported below, have been generated after the model has been improved based on the data from the experiments with human subjects.

Scenario Design

Two different experiment designs (D1, D2) have been developed in order to analyze specific aspects of the interaction between the pilot flying and the AHMI/AFMS. D1 represents a set of 7 full flight scenarios containing three flight phases: cruise, approach and final approach. Each of these scenarios had a duration of 20 to 30 minutes. During flight, different events such as uplinks of new flight plans have been induced. These events prompted the pilot flying to interact with the AHMI. The simulated aircraft was a VFW 614 ATTAS which has been equipped with a generic cockpit model (including topology and functionality) derived from an Airbus A320. In addition to the standard devices (such as primary flight display and navigation display) the cockpit was equipped with the AHMI/AFMS.

D2 represents only one scenario focussing on the task of handling a flight route uplink of the ATC on the AHMI (Handle_ATC_Uplink). The experimental environment consisted of a part-task simulator with the AHMI/AFMS as the main instrument. D2 allowed for many task repetitions and thus evolvement of routine.

Task Modelling and Analysis

We investigated in depth the Normative Activities for handling flight route changes “uplinked” by the ATC (Handle_ATC_Uplink). Uplinks are indicated to the pilots via a message box (Figure 9a).

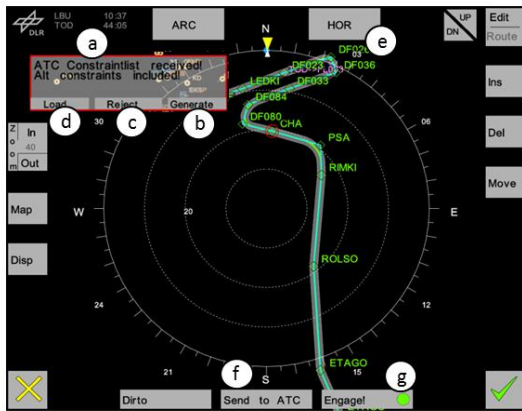


Figure 9: AHMI with main steps of the task Handle Uplink

The pilot flying (PF) first has to generate the trajectory for the modified plan by clicking on the “Generate” button in the message box (Figure 9b). As a result the new trajectory is shown as a dotted line. Afterwards s/he has to check the changes on the horizontal view and then on the vertical view in order to see if they are acceptable. In total the PF has to perform six checks c_1 - c_6 (three on the horizontal view and three on the vertical view). The result of each check is either “ok” or “not ok”. Pilots may only accept the uplink if all checks were “ok”. As soon as a check results in “not ok” the PF is requested to reject the uplink by pressing the “Reject” button on the message box (Figure 9c). In the following the individual checks are described.

Three checks on the horizontal view:

- c_1 : The first waypoint of the flight plan must be located in front of the aircraft. If the waypoint is behind the aircraft, the AFMS will (later) calculate a trajectory containing a circle in order to make the aircraft to turn and fly back to the first waypoint. In order to perform the check the PF has to compare the position of the graphical symbol for the first waypoint and the position of the aircraft symbol.
- c_2 : The PF has to check, if the flight plan ends on the runway, i.e. if the last waypoint has to be located exactly on the runway. In order to perform the check the PF has to scroll to the end of the flight plan, because normally the flight plan is shown only partially due to the screen size. There, at the end of the flight plan, the PF has to compare the position of the graphical symbol for the last waypoint and the position of the runway symbol.

When c_1 and c_2 are “ok”, the PF has to generate the trajectory by pressing the “Load” button (Figure 9d).

- c_3 : The PF has to check that the generated trajectory does not contain circles. This can for example happen, if the performance of c_2 took too long and the aircraft passed the first waypoint while the PF was still busy performing

that check. The AFMS would then (as described above) calculate a trajectory containing a circle. In order to perform the check the PF has to look at the beginning of the trajectory.

The following three checks are performed on the vertical view. In order to access the vertical view the View-button has to be pressed (Figure 9e).

- c_4 : The PF has to check, if the cruise flight level (CFL) is appropriate (in our experiments it should be over 8000 feet and below 32000 feet). In order to perform the check the PF has to look at the CLF label and perceive the displayed value, which is located at the right edge of the interface. Then, the pilot has to cross-check the actual value with the target value provided in the flight map.
- c_5 : The PF has to check that the intercept altitude is appropriate for the airport and runway. In order to perform the check the PF has to look at the intercept altitude label and perceive the displayed value, which is located at the right edge of the interface. Then, the pilot has to cross-check the actual value with the target value provided in the flight map.
- c_6 : The pilot has to check that the altitude of the last waypoint of the trajectory is equals the altitude of the runway. In order to perform the check the PF has to scroll to the end of the vertical flight plan, because normally the vertical flight plan is shown only partially due to the screen size. There, at the end of the vertical flight plan, the PF has to assess if he graphical altitude line ends at zero.

If all checks are “ok” the PF presses the “Send to ATC” button (Figure 9f) to acknowledge the uplink. After feedback from ATC is received s/he has to press “Engage” (Figure 9g) to activate the new trajectory, i.e. the FMS then actively controls the aircraft (via the autopilot) to follow the trajectory.

The procedure has been modelled in form of a hierarchical task tree using the PED tool. The task tree consists of rules.

The task analyses revealed a relatively long duration for the visual checks. In total the longest execution time for the Handle Uplink task has been computed as approx. 60 sec. The shortest execution time has been computed as approx. 2,5 sec (in case the first check results in “not ok” and the flight plan has to be rejected). Within these durations the checks that consume most of the needed time are: c_2 and c_6 .

The workload analysis shows a high induced visual workload (see Figure 10).

Simulation

We used the cognitive architecture CASCaS to execute the task Handle_ATC_Uplink as well as other flight tasks (e.g. the task to monitor essential flight parameters like altitude, speed and heading) in the experimental designs D1 and D2. For experimental design D1 the model executed one of the 7 full flight scenarios 37 times. The simulation of the other scenarios is still on-going. For the experimental design D2 we executed the corresponding scenario 180 times.

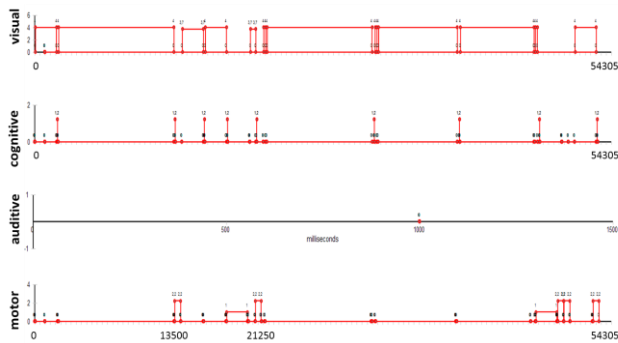


Figure 10: Workload induced by the task Handle_ATC_Uplink

All simulation runs have been performed in a batch mode, i.e. the simulation runs have been performed highly automated without user interaction or supervision. During the simulation runs data has been recorded: simulated hand and eye movements as well as selected goals, rules and working memory contents.

Evaluation

The interaction between the pilot flying (PF) and the AHMI/AFMS has been analysed with regard to gaze behaviour, temporal behaviour and human error. In the following we will present the effect of the AHMI on pilot behaviour as it has been simulated by CASCaS. We will also refer to some validation results of the simulated data.

Gaze Behaviour: We investigated the simulated data with regard to the gaze distribution and dwell time of the model on a set of Areas Of Interest (AOI). As AOIs we considered the HMI of the AFMS (AHMI), Primary Flight Display (PFD), Navigation Display (ND), Engine Display (ENG), Electronic Flight Control Unit (EFCU) and Window (WND). Gaze distribution and dwell time have been analysed separately with regard to the flight phases cruise, approach and final approach. Results of gaze distribution analysis (mean and standard deviation) are depicted in Figure 11.

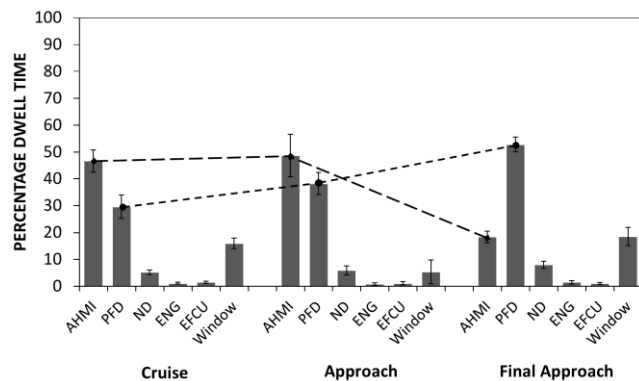


Figure 11: Gaze distribution simulated by the pilot model

During cruise and approach most gazes of the model are directed onto the AHMI, followed by the PFD. This changes during final approach where gazes onto the AHMI decrease and gazes onto the PFD increase. The relevant influence factor for the simulated gaze distribution are the changing task priorities.

In order to test the validity of the model data we compared the model data with the data of the human subjects. We computed the trend fitness and the local fitness. The overall correlation between the model and the human pilots has been measured with $r^2 = .85$. In order to assess the local fitness of the model, we compared the distribution within the human population to the distribution between the model and the human population using the root mean square deviation. The results indicate a high degree of fitness. More details can be found in [11]

We discussed the data with subject matter experts (pilot trainers) in order to interpret the data with regard to the AHMI design. According to the subject matter experts the PFD is the most important instrument during Final Approach, but approx. 80% of the gazes are directed to this instrument in current aircraft (without AHMI/AFMS). Today, the task to handle an ATC change request is distributed between pilot flying (PF) and pilot monitoring (PM). The PM takes over the communication with ATC, while the PF changes the flight parameters. This task sharing is much more difficult with the AHMI/AFMS and has not yet been designed with corresponding procedures. Furthermore, the AHMI uses only visual interaction while in aircraft today, acoustic communication with ATC is paramount. Consequently, the PF has to direct his gazes onto the AHMI relatively often.

The analysis of the simulated *dwell time* (Figure 12) shows that the model at average focuses on the AHMI for 3.2s during cruise, 3.2s during approach and 1.7s during final approach. Dwell time of the model on other displays in average ranges from 0.5s to 2.4s during cruise, from 0.4s to 3.2s and from 0.4s to 2.5s during final approach.

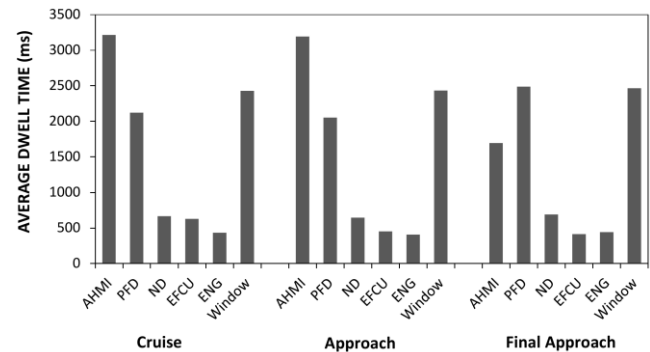


Figure 12: Dwell times simulated by the pilot model

We compared the dwell time of the model with the dwell times of the human subjects [5]. The average dwell times on the AHMI of the subjects are higher than those of the model in all three phases (5.7s during cruise, 4.9s during approach and 1.8s during final approach). We hold the hypothesis that the long dwell times for human subjects on the AHMI during cruise can be explained due to the fact that the AHMI contains redundant information which is normally located on the PFD. We assume the pilots retrieved necessary information from the AHMI instead of the PFD in order to reduce effort. We will extend our model by implementing effort-driven attention allocation to reproduce the data.

Further empirical investigations are necessary to evaluate this mechanism.

Temporal Behaviour: In order to measure the execution time for the task “Handle Uplink” a start event e_{start} and an end event e_{end} have been defined: e_{start} is the pop-up of the message box on the AHMI (cf. Figure 9a), e_{end} is the activation of the “Engage” button (cf. Figure 9g).

The execution time of the model (Figure 13) during approach is slightly longer than in cruise. This can be explained by the fact that during approach the priority of other tasks, like monitoring the flight parameters, is higher than during cruise. Thus, the model interrupts the task Handle_ATC_Uplink more often. The time which the model spends on performing other tasks while interrupting the Handle Uplink task have been included in the measurement of the execution time.

The duration of the Handle_ATC_Uplink task of approx. 1 min. is relatively long compared to other tasks. In order to continue other high priority tasks (e.g. monitoring the essential flight parameters like altitude, speed and heading) the pilot model has to interrupt the task quite often

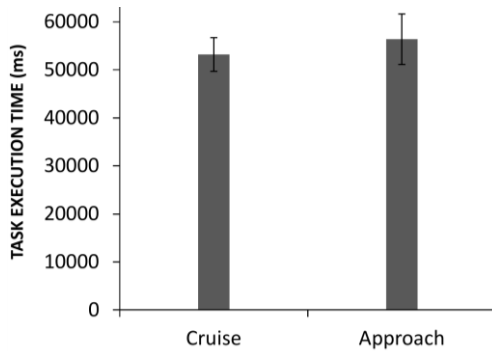


Figure 13: Task execution time for Handle_ATC_Uplink

We compared the execution time of the model with execution time of the human subjects [5]. Different from the model human subjects are faster in the approach phase. A possible explanation is the increased time pressure during this phase. This influence factor is not considered in the model. This finding is consistent with the interpretation of the model behaviour. The model has to interrupt more often during approach in order to continue high priority tasks. Since time pressure has not been modelled, the Handle Uplink is performed with the same accuracy. Possibly the human subjects try to avoid these interruptions and thus execute the ATC uplink faster than during the cruise phase to be able to go on with high priority tasks. This can lead to less accuracy in the performance of the Handle Uplink task. These interpretations are highly hypothetical and need further investigation and data analysis.

Again, we discussed the data with subject matter experts. According to them the execution time induced by the AHMI seems to be too long compared with the situation in current aircraft. Today, ATC communicates a maximum of three flight parameters in one message. These changes can be implemented on the flight instruments in approx. 20 sec. With the AHMI many changes can be communicated in one

message and pilots have to check the whole flight route to find these changes. Naturally this takes time. From these results we derived the guideline for the design of the AHMI and corresponding procedures to avoid long execution times or if this is not possible to foresee explicit break points. Such break points should be between unit tasks in order to minimize the memory load and task switching cost [10]. Memory load results from the need to memorize the context of the interrupted task (e.g. checked values). Task switching cost results from the need retrieve this context when the task is resumed.

Additionally to the execution time we measured the reaction to the message box that pops up on the AHMI to indicate the uplink of a new flight route (cf. Figure 9a). We defined the reaction time (Figure 14) as the difference between the time when the message box pops-up and the time when the pilot fixates the message box. Influence factors for the simulated reaction time are e.g. saliency of events, saliency of the graphical neighbourhood and distance to events.

The reaction times of the model during the cruise and approach phase are similar. The model reacts to the message box on average after approx. 1 sec.

We compared the execution time of the model with execution time of the human subjects [5]. The reaction times of the subjects are similar to those of the model.

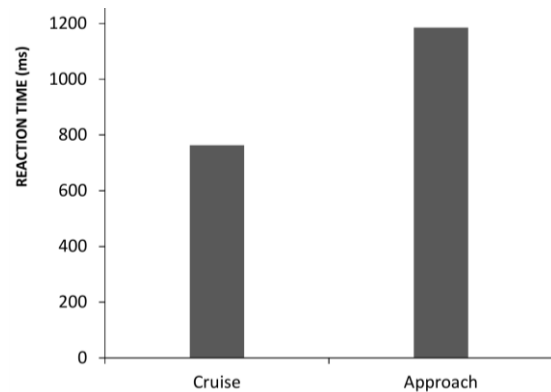


Figure 14: Reaction time for Handle Uplink

According to subject matter experts this reaction time is fully acceptable for the Handle Uplink task

Human Error: The model simulates errors due to Learned Carelessness. The experimental setting D2 foresees a high number of repetitions of the Handle_ATC_Uplink task. D2 includes two experimental conditions. In the first condition, the same scenario is simulated 170 times without variations. During these repetitions all checks constantly result in in “ok”. In the second condition, the same scenario is repeated again 170 times but with small variations. These variations are designed in a way that the check outcomes “ok” and “not ok” alternate.

In the first condition the model predicts learned careless for the two checks c_2 and c_6 . For these checks the activation of the corresponding memory items exceeds the pre-defined threshold after approx. 40 (70) repetitions. Figure 15 shows the evolvment of the association strength for four

exemplary memory items. The association strength of the memory items involved in checks c_5 (crossed line) and c_6 ; triangle line) are above the pre-defined threshold (0.97). For these checks the models does no longer perform percept actions. Instead learned values, needed for the checks, are retrieved from memory. The experiment schedule included (after 170 repetitions) scenarios to test the accuracy of checks. These scenarios contained deviations that could only be identified if the checks have been performed correctly. The pilot model does not recognize these deviations and engages incorrect flight plans. In real flight this behaviour can for example lead to violations of altitude constraints and in consequence to collisions with other aircraft. The influence factors for the resulting Learned Carelessness are the high number of repetitions which always led to the same result (“ok”) and the high effort for performing the checks.

Based on the simulated results we derived the design requirements that the effort involved in performing checks c_2 and c_6 has to be reduced. In order to perform check c_2 and c_6 (see above) pilots have to scroll up to the very end of the horizontal or vertical profile, respectively. The effort can be reduced e.g. by eliminating the need to scroll and by providing explicit hints on profile errors (e.g. via specific symbols) at places of the screen that are easier to access.

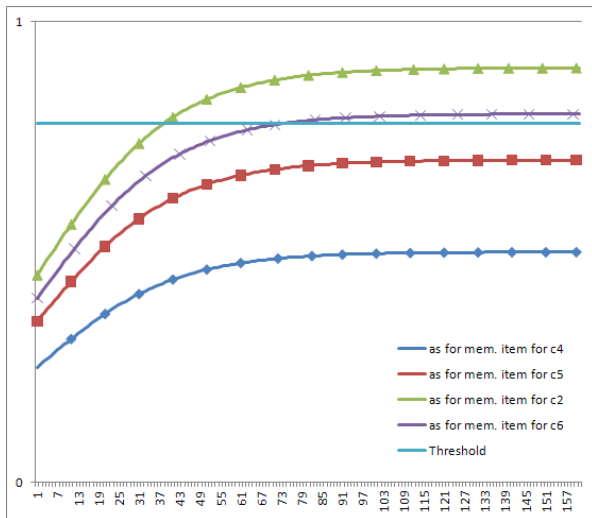


Figure 15: Association strength (*as*) for memory elements

In order to validate the simulated Learned Carelessness we compare the performance of checks of the cognitive model with the performance of checks of the human pilots. This validation is ongoing.

SUMMARY

In this paper we introduced a new methodology for analysing human-machine interaction with target system designs. The methodology is supported by tools for task analysis, model-based simulation and data analysis. A case study has been presented for analysing the interaction with a modern flight management system. In future work we will use the results of the study in order to improve the validity of the simulated data by extending and tuning the mechanisms implemented in the cognitive pilot model.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Commission 7th Framework Programme (FP7/2007-2013) under grant agreement no. 211988.

REFERENCES

- Anderson, J. R., Lebiere, C. (1998). The atomic components of thought. Mahwah, NJ: Erlbaum.
- Foyle, C.F., Hooey, B.L. (2008). Human Performance Modeling in Aviation. Boca Raton, FL: CRC Press, Taylor & Francis Group.
- Frey, D., Schulz-Hardt, S. (1997). Eine Theorie der Gelernten Sorglosigkeit. In H. Mandl (Hrsg.), 40. Kongress der Deutschen Gesellschaft für Psychologie. Göttingen, Seattle: Hogrefe Verlag für Psychologie.
- Grossman, T., Balakrishnan, R. (2004). Pointing at trivariate targets in 3D environments. In Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI). ACM Press.
- Lüdtke, A., Frische, F., Osterloh, J.-P. (2011). Validation of a Digital Human Model for Predicting Flight Crew-Aircraft Cockpit Interaction. In Proceedings of Berliner Werkstatt für Mensch-Maschine Systeme.
- Lüdtke, A., Osterloh, J.-P. (2010). Modeling Memory Effects in the Operation of Advanced Flight Management Systems. In Proceedings of HCI-Aero. New York: ACM Press.
- Lüdtke, A., Osterloh, J.-P. (2009). Simulating Perceptive Processes of Pilots to Support System Design. In T. Gross, et al. (eds), Proceedings Human-Computer Interaction (INTERACT). Springer.
- Lüdtke, A.; Weber, L., Osterloh, J.-P., Wortelen, B. (2009). Modeling Pilot and Driver Behaviour for Human Error Simulation. HCI International. Springer.
- McCracken, J.H. and Aldrich, T.B. (1984). Analysis of selected LHX mission functions: Implications for operator workload and system automation goals (TNA ASI 479-24-84). Fort Rucker, AL: Anacapa Sciences.
- SAE International (2006). ARP5056-Flight Crew Interface Considerations in the Flight Deck design Process for Part 25 Aircraft. standards.sae.org/arp5056/
- Frische, F., Osterloh, J.-P., Lüdtke, A. (2010). Simulating Visual Attention Allocation of Pilots in an Advanced Cockpit Environment. Proceedings of the MODSIM World Conference and Expo.
- Card, Moran and Newell (1983). The Psychology of Human-Computer Interaction. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.
- Fitts, P. M. (1964). Perceptual-motor skill learning. In A. W. Melton (ed.), Categories of human learning. New York: Academic Press.
- Anderson, J. R. (1982). Acquisition of cognitive skill. In Psychological Review, 89.