ADAPTIVE PILOT MODELING FOR COCKPIT CREW ASSISTANCE: CONCEPT, REALISATION AND RESULTS

P. Stütz [(1)] and R. Onken [(2)]

[(1)] ESG -GmbH
Einsteinstr. 174
81675 München, Germany
phone: +49-(0)89-9216-2604
fax: +49-(0)89-9216-2632
e-mail: pstuetz@esg-gmbh.de

[(2)] Universität der Bundeswehr
Werner-Heisenberg-Weg 39
85579 Neubiberg, Germany
phone: +49-(0)89-6004-3452
fax: +49-(0)89-6004-2082
e-mail: reiner.onken@unibw-muenchen.de

ABSTRACT

Monitoring of crew behaviour is a vital part of the internal situation assessment process performed within Crew Assistant Systems. A prerequisite for this is the availability of a behaviour models..

The paper describes the concept, realisation and evaluation of an adaptive behaviour model focused on rule based pilot activity. It autonomously acquires and reproduces situation-dependent *individual* pilot behaviour during plan execution.

The model concept provides a hybrid architecture. Normative behaviour, describing deterministic pilot behaviour, is implemented by use of *Petri Nets*. A learning module utilises a *case-based reasoning* mechanism to customise the transition behaviour within the Petri Net structure, thereby achieving online model adaptivity. For this purpose it accesses observed pilot action events stored in a case base.

The second part of the paper goes into detail on the experimental investigation performed. The results proved the basic functionality of the adaptive pilot model as well its capability to predict situation-dependent pilot actions.

INTRODUCTION

A variety of authors [2][4] investigate on the reasons for human error in aviation. The results gained so far can be divided in the following categories:

- Loss of situational awareness

  The sudden appearance of unusual situations force the pilot to change from his observing role to an active one. It is evident that in a lot of accidents the crews where suffering from insufficient awareness concerning aircraft state (*flight path awareness*, *terrain awareness*, *energy awareness*) and aircraft systems (*mode awareness*) leading to erroneous crew actions. Often even the basic necessity to act is not clear (*controlled flight into terrain*).

- Loss of piloting skills

  Today's cockpit avionics favour the loss of sensomotorical and cognitive skills by the crew. Training in simulators only provides limited compensation.

- Loss of self-criticism

  The appropriate time for the pilot to disengage from malfunctioning avionics systems is not recognised. Instead even more resources are wasted finding an explanation for the unintelligible behaviour of the system.

So called *cognitive assistant systems* aim to eliminate these deficiencies by providing hints, warnings and situation-adapted problem solutions. A prerequisite for such intelligent support is a thorough and comprehensive understanding of the overall situation. This includes the aircraft and the environment as well as the crew.

The principal goal is to ensure that the crew is aware of the currently most urgent task and to provide relevant automation to carry out this task in cases when the crew is over-taxed. Mutual understanding of objectives and resources both on the machine and on the human side seems to be a prerequisite for such a symbiotic man-machine relationship, in order to prevent pilot error and to enhance mission success.

The Cockpit Assistant System CAMA (*Crew Assistant Military Aircraft*) is a functional prototype developed according to principals. This knowledge-based system is designed to support military transport crews performing logistic and tactical missions. Financed by the German Ministry of Defence, CAMA was integrated in a flight simulator and a test aircraft [7]. The following investigations on behavioural modelling were carried out during the years 1994 till 1998.

BEHAVIOUR MODELING

As indicated before, the electronic assistant has to go through the same process of situation assessment as the crew in order to provide efficient support. Moreover the system must also be able to derive conclusions on behavioural aspects relating to its human counterpart. In other words, it has to develop an understanding of what the crews has to do and must not do in specific situations.

Monitoring of crew behaviour can be accomplished by comparison between expected and actual pilot actions. Whenever erroneous behaviour is detected, the pilot is warned and the appropriate actions are recommended to the pilot. In addition information on expected behaviour can be used internally to anticipate upcoming phases of high workload and to deduce inherent pilot intents. It is obvious that for the generation of such expected pilot behaviour an ample and reliable knowledge base is needed as reference. In the next paragraphs, the concept and the realisation of a respective *behaviour model* will be shown.

Concept

Other engineering disciplines (e.g. system design or construction) typically require overall models concerning bio-mechanical and cognitive behaviour. These models represent a whole range of behavioural aspects through averaging. In contrary a model suitable for individual pilot assistance also has to take into account individual aspects of behaviour, neglecting this would lead to sub-optimal results.

Fundamental for the concept of the proposed model is the assumption that normative regulations and procedures provide the guidelines for pilot behaviour, and can be described as deterministic pilot behaviour documented in pilot handbooks and air traffic regulations. This normative behaviour then is steadily amended and adapted by the individual pilot within certain tolerances in order to suit his needs and preferences.

In order to imitate this iterative refinement a process within the technical system had to be created allowing a continuous adaptation of a predefined normative core knowledge base by learning from observed pilot activities, thereby establishing the *adaptive pilot model*. This customised knowledge base than allows individually correct statements on upcoming pilot actions regarding the actual situation (Figure 1).

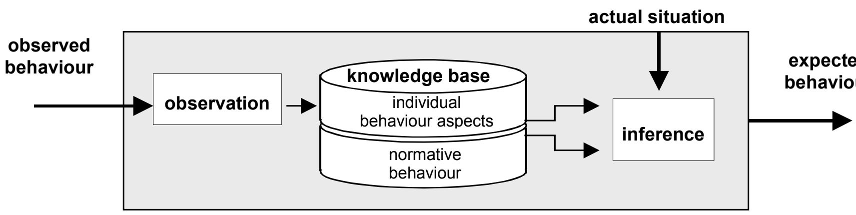


Figure 1: Hybrid concept for a adaptive pilot model combining normative behaviour with individual aspects

Realisation

Realisation of the adaptive model was performed in two major steps. At first the normative model had to be implemented. In a subsequent step an extension providing the adaptive capability was added.

Normative model

Pilot behaviour can be separated into situation assessment and action processing components. Behaviour modelling is performed for all flight segments (taxi, takeoff, departure, IFR/cruise, tactical flight, drop, approach, landing) and concerns the following tasks:

a) situation assessment

recognition of current flight segment and process of plan execution related to flight plan and procedures

b) pilot actions:

primary flight guidance (altitude, course, airspeed, power setting, ...)

operation of flaps, gear, speed brakes, radios

In his analyses [5] showed that *Petri Nets* are the most suitable representation for this mainly rule-based behaviour because of their ability to formulate concurrent, discrete event driven procedures.

Petri Nets:

Petri Nets can be described as a graphical representation of a net graph based on a mathematical theory which enables the analytical verification of system properties. A typical Petri Net consists mainly of the following net primitives:

Places:

Discrete states are represented by places. Examples are flight segments ("final approach"), conditions for subsequent actions ("turn right after passing A") and states of discrete aircraft systems ("flaps 20 degree").

Transitions:

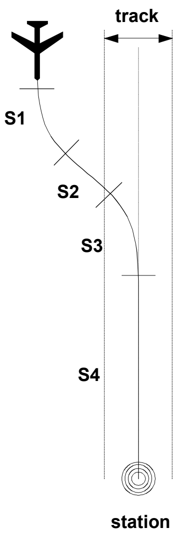
Transitions are used to represent situation state transitions, e.g. between flight segments ("final approach ### landing") and discrete aircraft systems ("landing gear up ### down"). These transitions are typically evoked by fulfilled conditions ("Altitude higher 5000 ft")

Tokens

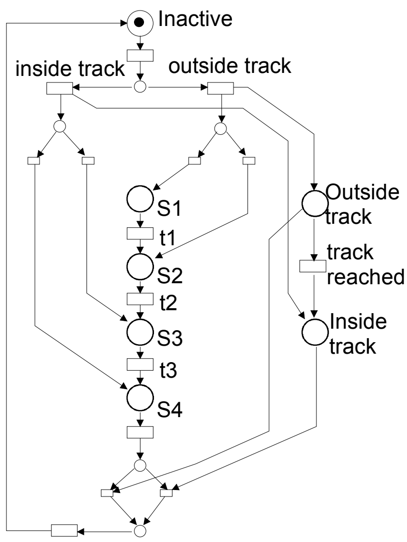
Tokens symbolise the current net state as marks on the relevant places.

Example:

An *interception* is carried out to reach a given (magnetic) course, which leads to a target station (e.g. a radio navaid). This can be required within published departure or approach procedures or can be commanded by air traffic control. In the general case, an interception covers 4 sections (see fig. 2): turning to intercept heading (S1), maintaining on intercept heading (S2), turning to the given course (S3), tracking of the given course (S4). Sections are skipped if the aircraft fulfils the characteristics of a following section, e.g. if the aircraft is already on intercept heading at the time the procedure is started, section S1 is skipped.



a) intercept procedure



b) Petri Net

Figure 2

Adaptive Extension

The second step was to find an appropriate mechanism to automatically adapt the normative statements of the petri net structure mentioned above to individual habits. Several algorithms for *machine learning* and *example based learning* were evaluated. Finally the method of *case based reasoning* was chosen and implemented in a case learning module complementing the normative model core [6][3]. In this module observed pilot action events are stored in a case base and attributed according to their coherence with state transitions in the Petri Net structure. On demand the Petri Net interpreter is able to recall these cases using similarity considerations during runtime, in order to refine its state and transition parameters within given tolerances. This functionality ensures full online adaptivity, but simultaneously considers the primacy of the normative model.

Figure 3 shows modules and functions of the adaptive piloting model. The functions are explained in the following.

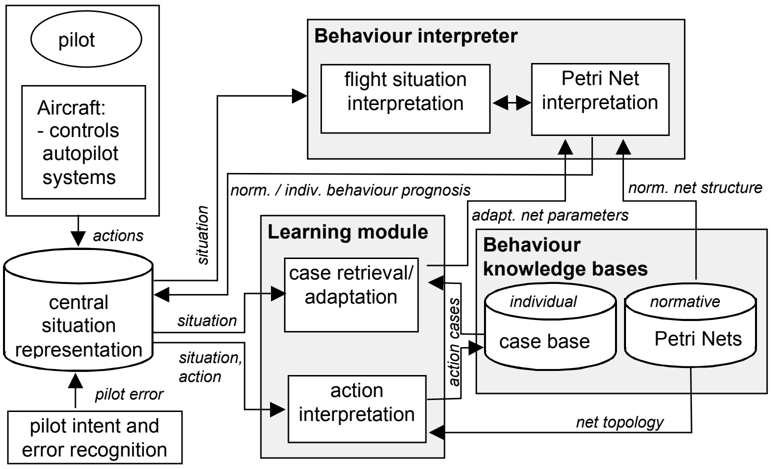


Figure 3: Modules and embedded functions of the adaptive pilot model

Behaviour interpreter

This module represents the real time processor of the Petri Net system. It loads the normative behaviour model stored in a Petri Net description language during system startup. The *Petri Net interpretation* function manages Petri Net states and transitions. State and transition parameters subjected to adaptive modelling are received on request during runtime from the learning module. The transition parameters, either purely normative or adapted are then handed over to the *flight situation interpretation* function to be monitored. When a condition is fulfilled, the Petri Net interpretation is notified and the transition takes place.

Action interpretation

This function supplies the case base with a constant stream of discrete pilot action events. Detected actions are assigned to appropriate state transitions in the Petri Net structure. Therefore this function uses information on net topology. In order to make these action cases useable for a later case retrieval, additional situation attributes under which the action took place are recorded.

Table 1: Case data stored for a course change manoeuvre

Action description		**Situational attributes**			
		required		supplementary	
pre (heading):	337°	*dist_track* (cross track):	0.66 nm	*alt*:	12180 ft
post (heading):	300°	*dist_basis* (distance to wpt.):	10.21 nm	...	
type:	Intercept	α (angle bet. current and next leg):	-56°		
		β (angle bet. planned track and acft. Hdg.)	14°		
		ias (speed):	200 kts		
		flight phase:	Enroute		

Table 2: Case data stored for flap setting action

Action description		**Situational attributes**			
		required		supplementary	
pre (setting):	14°	*ias* (speed):	132 kts	*alt*:	1631 ft
post (setting):	1°	*flight phase*:	Departure	...	

Case base

Table 1 and Table 2 show excerpts of data stored for example action cases. A commercial database tool is used for case storage management. It provides parallel read/write access to the database for case storage and retrieval, thereby enabling the continuous refinement of the knowledge base during runtime. SQL (structured query language) is used for communication.

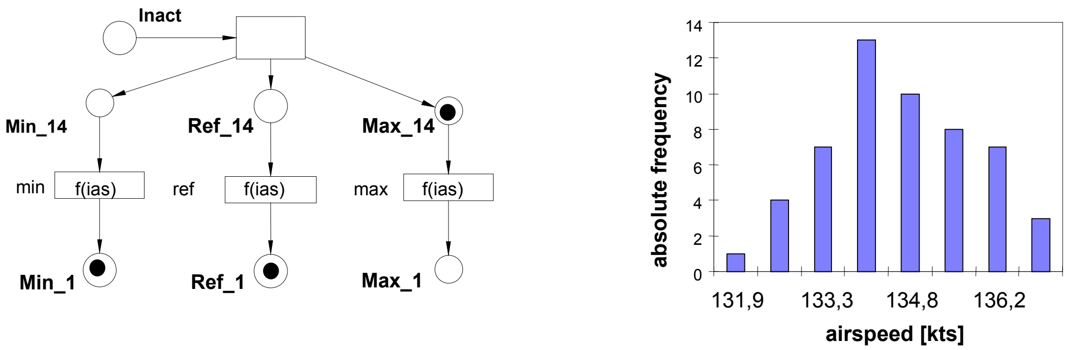
Case Retrieval and Adaptation

This function provides the on-line case base access for the Petri Net system in a way to preserve the overall normative task sequence, but also to take into account individual, admissible deviations. For illustration, it is considered that the Petri Net preconditions of a state-transition (that means a place is occupied by a token) are fulfilled.. The transition condition can now be acquired during runtime from the examples in the case base given for the individual pilot. It is thereby making use of observations which were just recently collected.

Referring to Table 1 and Table 2, the 'transition problem' is described by the pre and post state of the transition. This is passed along with other net status information to the *retrieval stage*. Finding relevant cases can be either *similarity based* or *trivial*.

Trivial case usage:

Trivial usage typically can be found for certain system setting inquiries (e.g. flaps, gear). In this case the simple check for pre and post state identity suffices to retrieve one or more example cases. An example is shown in Figure 4. The net *flaps_departure* models the flap setting behaviour during the departure phase. The net represents a range of possible settings through the run of 3 independent tracks. They symbolise the earliest possible (*Max_x*), the typical (*Ref_x*) and the latest possible transition (*Min_x*), dependent on the actual airspeed. In only normative mode this would correspond to the prescribed flap settings according to stall and maximum flap extension speed. The adaptive model replaces this for the earliest, typical and latest flap setting action considering its experience with an specific pilot. In order to do so, a statistical evaluation is done on all known flap reductions from 14° to 1° for this particular pilot (Figure 4b). An airspeed higher then the 15% percentil (132.5 kts) would allow the *Min*-Transition to fire. Firing of the *Ref*-and the *Max*-transition respectively is triggered by the median (134 kts) and the 85% percentil (135.6 kts) value. Figure 4a shows the Petri Net state at an actual airspeed of 135 kts.



a) fragment from net *flaps_departure* modelling the retraction from 14° to 1°

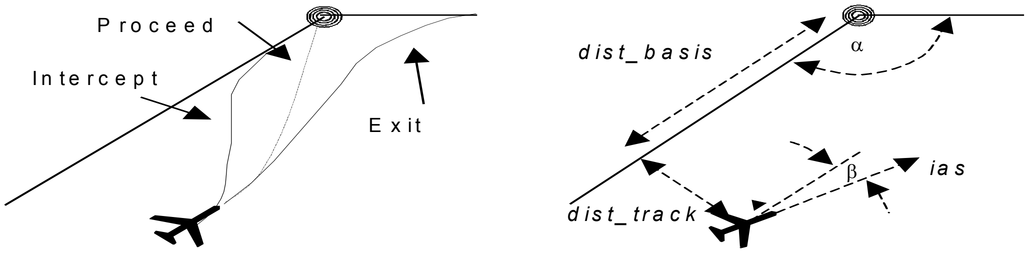
b) histogram of airspeed values when flap reduction form 14° to 1° was observed during departure (54 cases)

Figure 4: Petri Net and frequency histogram for departure flap setting

Similarity based case usage:

The above given example for trivial case usage only took airspeed into consideration as transition parameter. However, a variety of state transitions depend on the occurrence of more complex situations. It is a characteristic for these *multivariate* transitions that the set of parameters which influence the state transition is known; their absolute value and their functional relation, however, has to be estimated. In this case *similarity based* case usage has to be conducted.

Figure 6a shows an example for a more complex decision task. Given a lateral deviation from the current leg of the planned track, the pilot has to decide whether to intercept the current leg (*Intercept*), to directly proceed to the next waypoint (*Proceed*) or to disregard the current leg and to steer towards the following leg (*Exit*). Relevant situational information (Figure 6b) for this decision is considered to the aircraft's relative position (*pos* to the current flight plan leg), defined by the cartesian values *dist_basis* and *dist_track*, the aircraft's speed (*ias*), the aircraft's current heading relative to the actual track (angle β) and the geometrical constellation between the current and the following leg represented by the angle α.



a) alternative 1: *Intercept- Proceed-Exit*

b) relevant situational attributes

Figure 5: decision alternatives after considerable cross track deviation

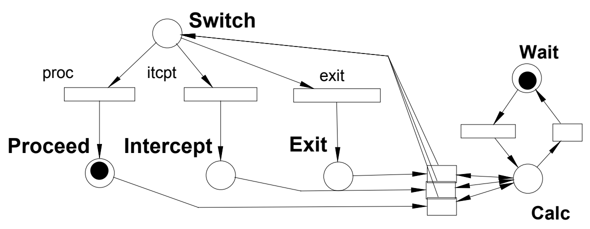


Figure 6: Petri Net fragment from net *Basic_Tracking_Off_Track*

Figure 6 shows the respective Petri Net indicating that the pilot is most likely to proceed directly to the next waypoint. Finding a normative functional relation between the relevant parameters and the final guidance decision is hard to achieve, as objective rules and recommendations do not exist. This freedom typically favours the formation of individual pilot customs and a rigid normative model looses its validity.

The way to tackle this problem in the sense of case-based reasoning is to see in which situation (decribed by the above mentioned parameters) a pilot made these guidance decisions and to conclude that he will come to the same decision again in a similar situation.

The question arises how situational similarity can be determined. In general the situational description of an observed pilot action can contain *metrical* (e.g. position, altitude), *ordinal* (e.g. flight phase) or *nominal* attributes (e.g. waypoint idents). In the given example metrical attributes prevail. For this group of attributes similarity considerations traditionally utilise *distance* measurements, the *euclidian distance* being the most well known measure. Formula 1a shows the relation between distance *d* and similarity *sim* for two attribute values *a* and *b*, whereby d_{max} is the maximum allowable distance. If distance *d* decreases to zero, similarity raises to 1 and identity between *a* and *b* is reached. More information on the determination of similarity especially for attributes other than metrical can be found in [1].

$$sim(a,b) = 1 - \frac{d(a,b)}{d_{\max}}$$

a) relation between distance and similarity

$$SIM() = \frac{w_{pos} sim_{pos}() + w_{\alpha} sim_{\alpha}() + w_{ias} sim_{ias}() + w_{\beta} sim_{\beta}() + w_{fph} sim_{fph}()}{w_{pos} + w_{\alpha} + w_{ias} + w_{\beta} + w_{fph}}$$

b) local similarities *sim* contributing to global similarity *SIM: w: local similarity weight*

Formula 1: similarity measurements

In order to use this approach to compare stored case attributes with the current situation, local similarity values are computed individually for all attributes. After that these values are combined to a global similarity *SIM* through a weighted combination. For the given example the global similarity is made up from the local similarities for position, alpha, airspeed, flight phase and waypoint ident. Note that the last attribute (*fph*) is not metrical (Formula 1b).

Assuming that the aircraft for some reason deviates from the planned track violating a certain threshold. Here the Petri Net interpreter invokes the aforementioned net *Basic_Tracking_Off_Track* and issues a request to the retrieval and adaptation function together with a description of the actual situation in order to conclude for the most likely pilot reaction. This function now tries to retrieve suitable course manoeuvre cases and sorts them according to their situational similarity. The action type of the most similar case (e.g. Intercept) is than passed back to the interpreter and the respective transition is allowed to fire.

EXPERIMENTAL INVESTIAGTION

The adaptive model was implemented in a workstation environment and validated in simulator trials. The subjects were 5 professional air transport pilots with 800 to 2800 hours flight experience. After familiarisation with the fixed base research simulator was assured, a variety of experiments were conducted in order to verify and validate the functions of the adaptive pilot model. The following chapters only focus on the results found on the topic of similarity-based case usage.

Scenario and tasks

In order to assess the prognostic capabilities of the adaptive model, the pilots lateral aircraft guidance behaviour was investigated. A scenario was set up which allowed to repeatedly provided the pilot with off track situations as described in Figure 6. Each pilot had therefore to conduct four IFR flights, two from Frankfurt (EDDF), the other two from Friedrichshafen (EDNY) to Stuttgart (EDDS). Duration of each flight was about 40 minutes. Once the cruise altitude (FL 120) was reached, the pilots were assigned *radar vectors* (*"Lucky07, turn left (right) heading xxx"*), typically given through ATC in order to maintain air traffic separation or to avoid bad weather areas. After these radar vectors forced the pilots to deviate considerably from the pre-planned track, they were requested to disregard the assigned heading and to follow the original flight plan again (*"Lucky07, proceed as filed" or "Lucky07, resume own navigation"*). In this situation, the pilots had to decide whether to intercept the closest flight plan leg, to proceed directly to the respective waypoint or to steer towards the following leg corresponding to Figure 6a and b.

The pilots were able to control the aircraft via sidestick and autopilot. A glass cockpit like primary flight display was used to indicate airspeed, altitude, vertical speed and aircraft altitude. On a navigational display the flight plan, radio navaids and navigational instrument indications were depicted in moving map format. On a third display the flight plan was shown in a alphanumerical flight log format. All pilot actions were recorded with their attributes by the *action interpretation function* as explained above.

Results

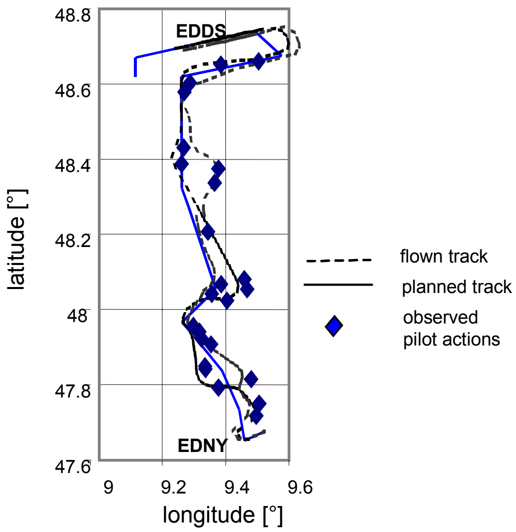


Figure 7: pilot actions (diamonds) plotted according to their geographical position during observation

Figure 7 shows as an example two tracks flown by Pilot 4 from Friedrichshafen (EDNV) to Stuttgart (EDDS), They deviate from the pre-planned track due to the radar vectors given by (simulated) ATC. The diamonds depict the course manoeuvres detected by the *action interpretation* function. These pilot actions were automatically classified according to characteristic features (e.g. start and end heading) as being *Proceed*, *Intercept* or *Exit* actions. An appropriate decision table was used.

The results of such classification can be seen in Figure 8a. The diagram plots all action cases in a common co-ordinate system defined by the attributes *dist_basis* and *dist_track* (see Figure 6d). This representation supports an easy comprehension of the positional relation of the actions recorded. The accumulation of dedicated action types in certain areas suggest that the aircraft's relative position has a significant influence on the pilots decision. A demerger of the remaining overlapping zones can be assumed in other dimensions of situation space (e.g. *ias*, α).

General similarity computation

In order to validate the similarity based case usage mechanism, clinical test requests were sent to the *retrieval and adaptation* function after the case acquisition phase, effectively asking the function "which action case out of the case base is most similar to the test situation and which action type is associated with it".

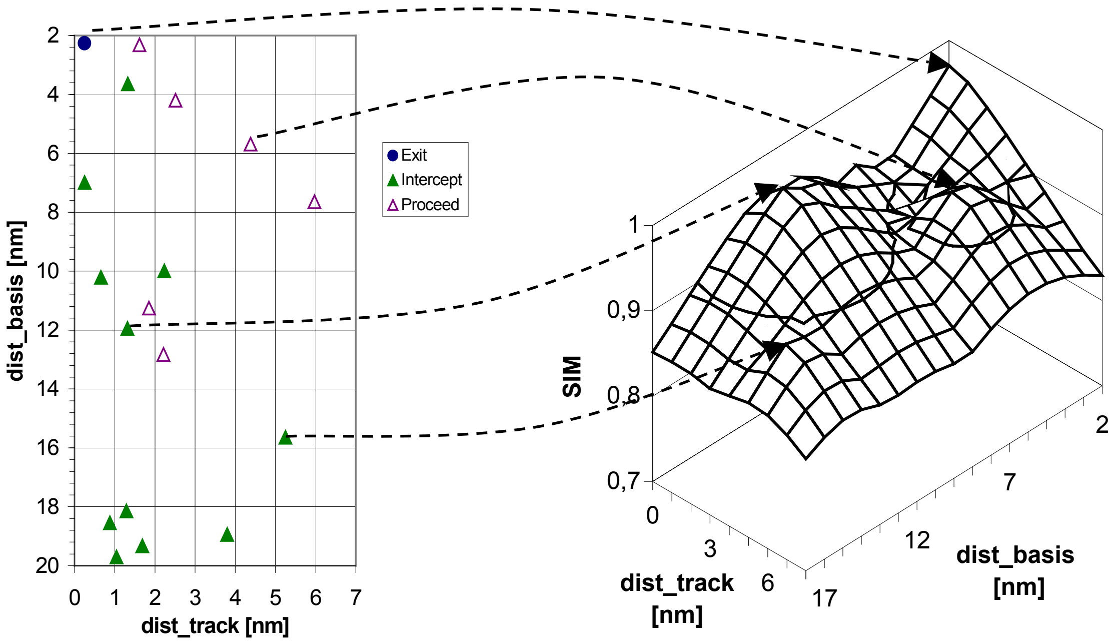
As designed, the *retrieval and adaptation function* calculated and sorted the global similarity of all known *Exit*, *Intercept* and *Proceed* action cases to each provided test situation. Using Formula 1b the global similarity *SIM* measures were computed by a weighted combination of the local similarities *sim* as given in Table 3.

Table 3: weight and variation range of attribute values

attribute	value	weight	d_{max}	*sim*
dist._track *dist_basis*	0 – 8 nm 2 – 24 nm	10	26 nm	$sim_{pos} = 1 - \frac{\sqrt{(dist\_track_C - dist\_track_S)^2 + (dist\_basis_C - dist\_basis_S)^2}}{d_{\max}}$
α	-60° – 30°	5	180°	$sim_{\alpha} = 1 - \frac{\alpha_C - \alpha_S}{d_{\max}}$
β	0°	5	180°	$sim_{\beta} = 1 - \frac{\beta_C - \beta_S}{d_{\max}}$
ias	200 kts	1	100 kts	$sim_{ias} = 1 - \frac{ias_C - ias_S}{d_{\max}}$
fph	-	*0*	-	-

Figure 8b shows the *similarity surface* gained for array of test requests applied on the case base of pilot 5, where the situation was described by varying *dist_track* and *dist_basis* values in steps of 1 nm. However, α was fixed to -45°, β to 0° and *ias* to 200 kts for all requests.

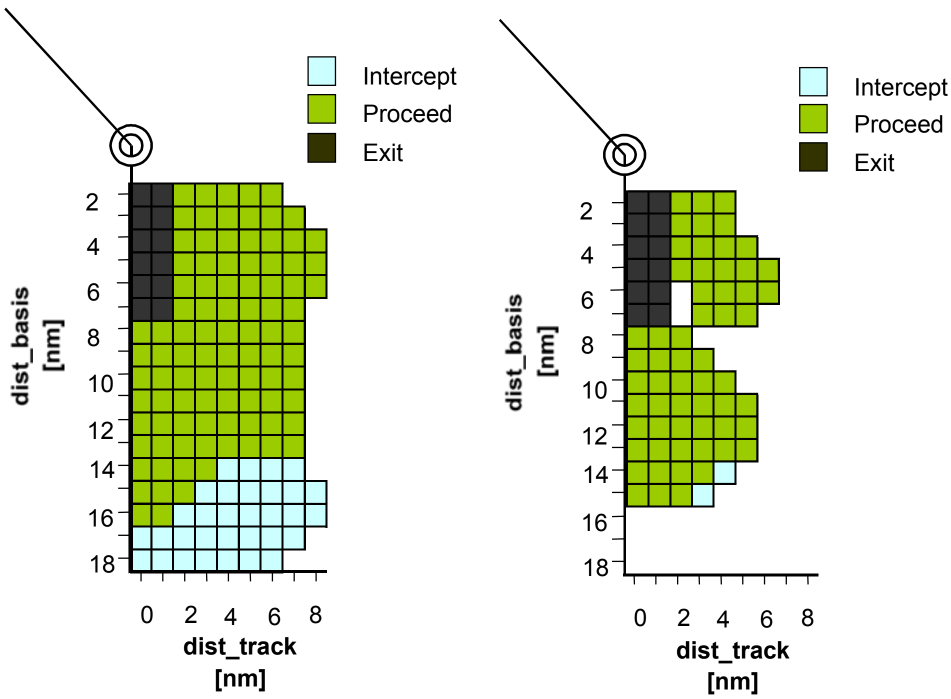
The *similarity surface* represents the global similarity values of the most similar stored action cases compared to the given situation descriptions. Cone-like structures show the strong affinity of certain situational regions to specific stored action cases. These dominant cases (marked by arrows connecting Figure 8a and b) graphically reside directly 'under' the tip of the cones in *dist_track* – *dist_basis*-situation space. These cases yield a strong positional similarity d_{pos} close to 1. The maximum elevation of each cone finally depends on the similarity values gained for the other, basically static situational attributes.



a) classified pilot actions plotted according to their relative position to the planned track (Pilot 5)

b) Global similarity surface; arrows depict the influence of dominant action cases

Figure 8: general similarity measurement for an array of test requests with 0 nm < *dist_track* < 8 nm, 2 nm < *dist_basis* < 18 nm, α = -45°, β = 0° and *ias* = 200 kts



a) SIM_{min} = 0.85

b) SIM_{min} = 0.90

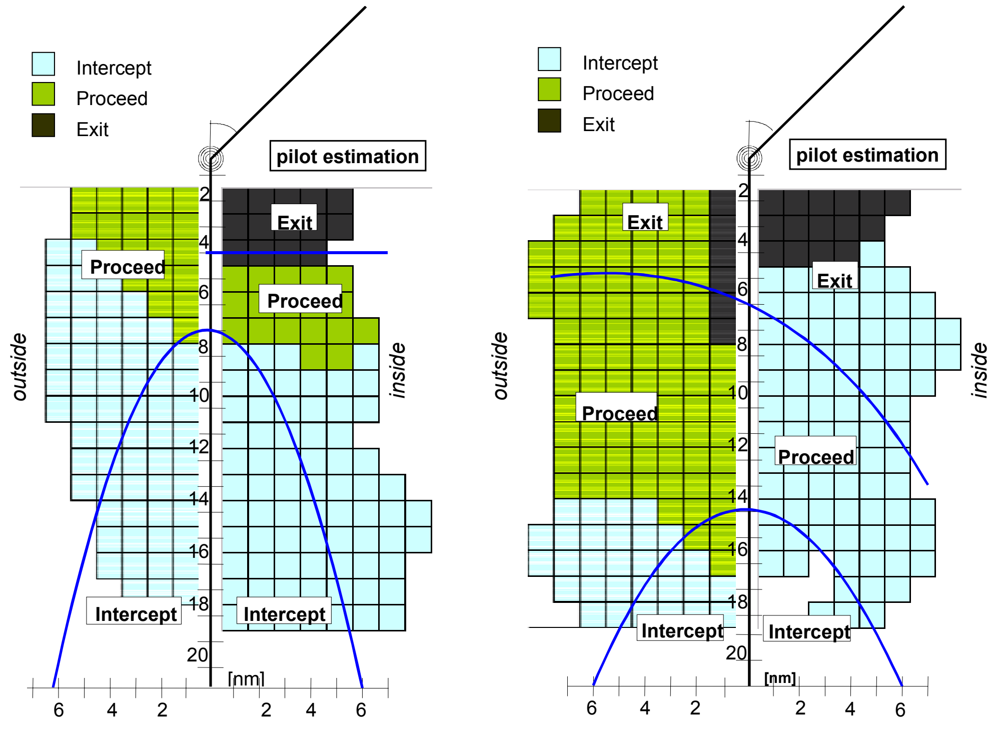
Figure 9: action type array derived for test request array (Pilot 5)

It may be observed that the computed similarity values in Figure 8b decrease significantly in test situations with large *dist_track* and *dist_basis* attributes. This can explained by the absence of action cases with positional relevance in these areas. It is obvious that a minimum similarity value SIM_{min} is required in order to regard a stored case as relevant for a given situation.

Figure 9a and b denote the action types associated with the most similar cases derived for the test requests. In Figure 9a a minimum similarity SIM_{min} of 0.85 was required, in Figure 9a SIM_{min} was at 0.9. White spaces can be understood as areas in situation space where the pilot model refrains from providing a solution due to a lack of relevant case knowledge.

Individual behaviour differences

Figure 10a and b clearly show strong differences among the action type solutions produced by the adaptive model for specific pilots. Obviously the model indicates a much stronger tendency for Pilot 5 to directly proceed to the waypoint compared to Pilot 4 when positions on the *outside* are considered. Even at quite large *dist_basis* values the model does not favour the *intercept*-option. Another peculiarity is that the model does not foresee at all the *Exit* manoeuvre for pilot 4 on the *outside*. To get more insight on the validity of these model statements, the pilots were asked to prepare a rough subjective drawing indicating the areas of their manouveral preferences. Overlaid on the results of the pilot model, the drawings indeed confirm the effects mentioned. Pilot 5 admits himself a quite large area where he would rather choose the *proceed* option than to intercept the planned track, but the tendency of Pilot 4 to prefer the intercept option up to a distance of about 8 miles to the waypoint. This threshold value was stated identically by the pilot model, considering small *dist_basis* values. Likewise Pilot 4 ruled out the *Exit*-option for his course behaviour on the *outside* and the pilot model was able to reproduce the *Exit*-area for Pilot 4 on the *inside* almost identically.



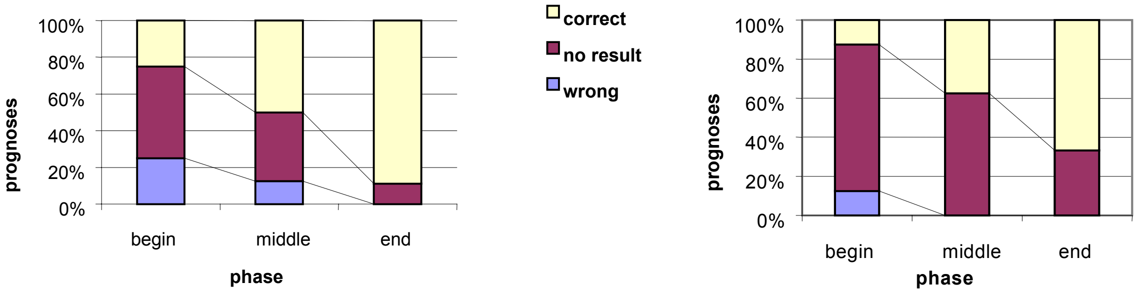
a) Pilot 4 b) Pilot 5

Figure 10: action type array with $|\alpha| = -45°$, $\beta = 0°$, $ias = 200$ kts, $SIM_{min} = 0.85$; areas separated by curved lines depict the pilot's subjective estimate of own manouveral preferences; *inside* and *outside* are used as positional reference considering the turn direction of the subsequent planned leg.

Learning progress and prognostic performance

To further gain objective insight into the prognostic capability of the adaptive pilot model, the retrieval and adaptation function was requested to predict the action types of each action case A_n in the case base by only having access to cases A_1 to A_{n-1}, where n denotes the time order of case recording. Within the three phases *Begin*, *Middle* and *End* the prognosis results were evaluated as *correct*, *wrong* or *no result*. As described before the yield of *no result* depends on the setting of SIM_{min}. Figure 11a and b give the respective results. During the *begin*-phase of case acquisition the probability of finding good matching cases is quite low. Therefore *wrong* or *no results* are in the majority. In this phase a raised level of SIM_{min} biases the system towards *no result* as seen in

Figure 11b. In the following phases correct model prediction become prevalent, finally reaching a score of 90 % for correct predictions in the *end*-phase in Figure 11a.



a) $SIM_{min} = 0.85$ b) $SIM_{min} = 0.90$

Figure 11: Learning progress and prognostic performance during case acquisition phases; Pilot 5

CONCLUSION

The paper started with a description of the demand for pilot behaviour modelling within cockpit crew assistant systems. The need not only for normative models describing the general prescribed behaviour but also for individual models providing information on subjective preferences and customs was emphasised.

In order to realise an appropriate model, a hybrid concept was introduced. The concept uses Petri Nets as a representation of rule-based pilot behaviour in the area of plan execution. Transition behaviour within the Petri Net system then is customised to individual pilot preferences during runtime, using previously observed behaviour examples. Theoretical background for this example-based adaptation process is provided through the paradigm of *case based reasoning*. For closer investigation a prototype was implemented and experiments were conducted with professional pilot.

The pilots had to perform basic decision tasks concerning lateral aircraft guidance within an IFR scenario. The results verified the basic functionality of the adaptive model. Furthermore significant differences in behaviour characteristics were observed and the model's ability to predict this correctly was validated. Further investigation should be carried out expanding the areas of pilot tasks covered by the adaptive models.

REFERENCES

[1] Althoff, K.-D.; Auriol, E.; Barletta, R.; Manago, M.: *A review of industrial case-based reasoning tools*, AI Intelligence, Oxford, 1995

[2] Billings, C. E.: *Aviation Automation - the search for a human-centered approach.* Erlbaum Verlag, Mahwah NJ, 1997

[3] Kolodner, J: *Case-based reasoning*, Morgan Kaufmann, San Mateo, 1993

[4] Onken, R.: *Basic Requirements concerning Interactions in Combat Aircraft*, Workshop on Human Factors/ Future Combat Aircraft, Ottobrunn, 1994

[5] Ruckdeschel, W.; Onken, R.: *Modeling of Pilot Behaviour using Petri Nets*, 16th International Conference on Application and Theory of Petri Nets, Torino, 1995

[6] Stütz, P.; Onken, R.: *Adaptive Pilot Modelling within Cockpit Crew Assistance,* Cognitive Systems Engineering in Process Control (CSEPC), Kyoto, 1996

[7] Stütz, P.; Schulte A.: *Evaluation of the cockpit assistant military aircraft (CAMA) in flight trials*; 3rd International Conference on Engineering Psychology and Cognitive Ergonomics, Edinburgh, 2000