In: F.E. Riter & R.M. Young (eds): Proceedings of the Second European Conference on Cognitive Modelling (ECCM-98). Nottingham: Nottingham University Press. ISBN:1-897676-67-0, pp. 104-111. Nottigham, UK, Apr 1-4, 1998.
http://www.safety-critical.de/doc/eccm98.pdf

A Psychological Model of Air Traffic Control and Its Implementation

Cornelia Niessen, Sandro Leuchter, & Klaus Eyferth
Centre for Man-Machine-Systems Studies
Technical University of Berlin
TIB 4/5-3, Gustav-Meyer-Allee 25
D - 13355 Berlin
{niessen|leuchter|eyferth}@zmms.tu-berlin.de

ABSTRACT
In this paper, we describe a model of en-route air traffic controllers' cognitive activities in a dynamic man-machine system. The implementation of the model MoFl (*Modell der Fluglotsenleistungen*) is based on a production system in the programming language ACT-R (Adaptive Control of Thought - Rational, Anderson, 1993).

KEYWORDS
ACT-R, dynamic mental representation, air traffic control

INTRODUCTION
For various reasons, it can be useful to have a computer model of the operator's cognitive skills (see e.g., Opwis & Spada, 1994). The implementation of complex psychological assumptions

- can provide a more detailed and explicit description of every cognitive process involved than a verbal description,
- can test a theoretical framework by showing if the anticipated effects can be reproduced,
- can serve as a framework for generating hypotheses that support the empirical work, and
- can be used to analyse and predict the effects of future technological changes on the operator's cognitive activities in complex man-machine systems. These insights into the consequences affecting cognitve performance can be helpful for future system design or training concepts.

On the basis of a broad empirical work - interviews, simulation experiments, memory tests, and a card sorting task with experienced and less experienced en-route air traffic controllers and of theoretical considerations, the interdisciplinary research group "En-route Controller's Representation" (EnCoRe) constructed a model MoFl (*Modell der Fluglotsenleistungen*) of the cognitive activities of experienced en-route air traffic controllers. The air traffic control domain serves here as an example to model cognitive processing during control of complex and dynamic situations. The focus has been on issues concerning problems inherent to dynamic situations: mental representation of the changing situations, and the context-dependent flexible coordination of concurrent cognitive tasks. In comparison to other research (Freed & Johnston, 1995, Bass et al., 1995) in our approach we concentrated on modelling the cognitive abilities of air traffic controllers rather than perceptual and motor skills. According to the rate at which traffic situations changes, and the cognitive task of air traffic controllers, perceptual and motor skills were only treated in order to ensure a realistic model - environment interaction.

The implementation of the model is based on a production system in the programming language ACT-R 3.0 (Adaptive Control of Thought - Rational, Anderson, 1993). As programming environment, ACT-R includes a broad and detailed theoretical framework of human cognition. For the most part, ACT-R is suitable for modelling the cognitive performance of en-route air traffic controllers. But, for some aspects of dynamic situations ACT-R does not provide convincing solutions.

The aim of this paper is to present the construction and the implementation of the model. This includes the principles of construction and implementation of our model, and the discussion of two special issues concerning the cognitive architecture of ACT-R: "dynamic representation" and "executive control". This paper is divided into three sections:

- short description of the air traffic control task
- the framework for the implementation: the cognitive architecture ACT-R
- description of the psychological assumptions of the model and its implementation

THE AIR TRAFFIC CONTROL TASK
On the basis of different sources of information (e.g., radarscreen, flight strips, head-phone communication with pilots), air traffic controllers have to control complex, dynamic, and time-constraint traffic situations in order to diagnose risky relationships between aircraft and to solve potential conflicts. Therefore, they have to perceive, comprehend, and anticipate multiple characteristics of many aircraft while new incoming aircraft create new traffic relationships for evaluation. It's a common assumption, that in complex technological systems of a dynamic nature operators develop a mental representation of the task

environment with which they interact. Diagnosis, decisions on future cognitive activities and actions are based on these insights into current and anticipated structures of the changing situation. Air traffic controllers express with the term *picture* (e.g., Whitfield & Jackson, 1982; Falzon, 1982) what is often described as *situation awareness* (e.g., Endsley, 1995; Flach, 1995): a mental representation of the current and future traffic situation.

By modifying the framework of cognitive task analysis (the “decision ladder”, Rasmussen, 1986), extensive interviews with seven experienced controllers provided a first explorative functional analysis of main tasks used to build up and maintain this mental *picture* of the traffic situation.

According to verbal reports of the air traffic controllers, the diagnosis of potential conflicts between aircraft contains stages, which are characterized by an increasing restriction and specification of the problem space. These stages are: *observing* the whole situation, *analysing* the parameters of selected aircraft, and *anticipation*. In the first step (*observation*) the operator monitors the whole situation in order to get a quick overview of the whole traffic situation. The goal of conflict detection demands selection strategies during radar-screening to structure the representation (see e.g. Amaldi & Leroux, 1995). According to the verbal reports, experienced controllers classify the aircraft on the basis of these signals (proximity, vertical movement, etc.) into two groups: those aircraft which have to be further analyzed (*analysing the parameters*) and anticipated (*anticipation*) in order to check for future conflicts, and those which are separated safely during that moment. The initial steps towards intervention and conflict resolution could be described according to Rasmussen´s stages (define task, fomulate procedures, and execute).

In order to model the air traffic controller's *picture* and the processes used to build up and to maintain this mental representation of the changing traffic situation, experiments provided a more detailed analysis of the following topics:

- information selection and recall,
- relational structure of the representation, and
- anticipation and conflict management.

The experimental work with real time simulation was based on a realistic simulation system of the control task called “En-route Controllers Representation - Programmable Airspace Simulation” (EnCoRe-PLuS) (Bierwagen, 1996). This system simulates air traffic control scenarios providing radar screen runs, electronic flight strips, and head-phone communication with a ghost-pilot; it also allows the user to set up experimental procedures and to keep logfiles of all system activities.

The results of this empirical work led to the conceptualization and the implementation of a model that describes the cognitive activities of air traffic controllers. The implementation of the model is connected with a modified version of EnCoRe-PLuS. EnCoRe-PLuS provides a real-time simulation environment. Predefined traffic builds up a simulation scenario that interacts with the model:

- The model can actively access new information about the changing traffic situation and can integrate it to its representation of the current situation.
- The model is informed about events within the task environment (e.g., incoming aircraft)
- The model can intervene with the traffic environment in order to solve conflicts.

MODELLING MENTAL PROCESSES OF EXPERIENCED OPERATORS DURING CONTROL OF A DYNAMIC MAN-MACHINE SYSTEM

For modelling mental processes of experienced air traffic controllers during control we have used the production system ACT-R 3.0. ACT-R provides a suitable framework: 1. as a psychological framework of human cognition, it also describes an environment for implementation, 2. ACT-R is based on explicit and very detailed assumptions about the cognitive architecture, and 3. as an environment for implementation, it is available in the public domain at no costs. In addition ACT-R has been applied to modelling a great number of problem solving tasks and is still in progress (e.g., ACT-R Perceptual - Motor Layer, RPM).

Even within such a framework, the conceptualization and implementation of mental processes in dynamic environments, as in the case of air traffic control, demand additional assumptions about three aspects of the dynamic task environment. 1. The continous changes of the situation. These changes do not allow fixed sequences of cognitive processing, they rather call in a cyclic update of varying relations as a basis of situational awareness. 2. The necessity to predict future states of the situation in order to predict potential conflicts. Such predictions alter the goals of ongoing control activities. 3. The demands to coordinate and to sequence simultanious requirements of the control task.

Widely used concepts for adaptive control of complex task enviroments (e.g., Anderson, 1993; Rasmussen, 1986; Hacker, 1978) concentrate on rather static tasks and on invariant goal structures. For example the cognitive architecture of Anderson's ACT-R does not take into account that in dynamic situations the operator has to continuously update her or his mental representation. In addition, such production systems are directed by a fixed goal hierarchy. But in the case of the changing and complex situation requirements, the controller has to coordinate the cognitive activities. This coordination is context-dependent: it does not follow a pre-defined goal hierarchy.

Recently there are some promising attempts to formulate cognitive architectures that deal with the specific demands of a dynamic task environment. For example, as a conceptual neighbor to ACT-R and SOAR, a new computational framework, the executive - process interactive control (EPIC), is proposed for this kind of human performance (Meyer & Kieras, 1997a,b; Meyer et al., 1995). Perceptual, cognitive, and motor processors have been built up for modelling cognitive processes during the performance of multiple concurrent tasks. The perceptual processor provides a continously update of the task environment. Within the cognitive processor, concurrent tasks can be scheduled by flexible executive processes that control relative task priorities. Also the architecture for human representation in complex system, "Man Machine Interactive Design and Analysis System" (MIDAS), promises a modelling environment that provides an updateable mental representation of the task environment and flexible scheduling of multiple task performance (Corker & Smith, 1993).

The implementation of the model "MoFl" (*Modell der Fluglotsenleistungen*) is based on ACT-R 3.0. The basic assumption is that cognitive skills are composed of production rules. A production rule is a modular piece of knowledge. Combining these rules into a sequence represents complex cognitive processes. ACT-R includes two kinds of knowledge representation: declarative and procedural knowledge. The basic units in declarative memory are so-called working memory elements (WMEs). A WME is an object with identity. It has named slots that can be filled with Lisp objects or references to other WMEs. References to other WMEs can be interpreted as relations, so that a semantic net with WMEs as nodes and references for relations is spread out. ACT-R defines an object-oriented structure for declarative memory. Every node in the net is an object of a certain class. A class is declared by naming all slots an object of this class will have. Subclassing is possible. Every WME has an activation level. It is manipulated by the programming environment. A special structure within the declarative part of the memory is the goal-stack. WMEs can be pushed onto and popped from this structure. The topmost WME is the current goal.

Production rules are the procedural part of memory. They consist of a condition and an action part. Conditions and actions refer to WMEs. The application of a production rule is realized by a simple pattern-matching mechanism. In order to support goal-directed performance, the first condition of every production rule must match the current goal. If all conditions of a production rule are true, then the action part is executed. Possible actions are: manipulation of the goalstack (push and pop), creation and deletion of WMEs, and modification of the slots of already retrieved WMEs. An ACT-R run consists of the continous application of production rules.

The prioritizing of processing is controlled by the activation parameter in ACT-R as well as by the current goal. A production is applied if it fires. A rule can fire if all conditions are fulfilled. Typically the fastest production will fire. The speed of application is mainly computed by the time it takes to retrieve the condition WMEs.

Activation signifies the current relevance of a WME for the processing of information. Sources of activation are the encoding process, execution of a production (addition of new WMEs), and creation of a goal node. The more activated a WME is, the faster it is retrieved. This means that if various WMEs match the pattern of a production rule, the most activated WME is retrieved. If various production rules can be applied, that production rule fires that retrieves the most activated WMEs. A WME can only get retrieved if its activation is above a certain level. But in the case of air traffic control there are three cases in which an inactive WME also has to be retrieved. In the first case, the controller has to update his mental representation continuously. Empirical work showed that controllers reduce the problem space by paying attention to meaningful signals for conflict detection during radar-screening. Because of these signal features, aircraft become focal. That means that they are attention demanding objects, therefore highly activated. Aircraft without these features are *extrafocal* (less activated). For these extrafocal aircraft there is no further demand for processing and they become inactive. But, in contrast to ACT-R, these inactive WMEs have to be retrieved in order to update them. Second, activation is increased not only by the encoding process. It is also guided by the encoding of signal features of aircraft. The third case concerns the context-dependend coordination of a goal. The high activation level of a goal that targets the solution of a detected conflict between aircraft can be decreased, it may be put aside for a while if there is enough time remaining for the solution. But at a certain point, activation has to increase suddenly in order to retrieve this WME and to apply the appropriate production rule in order to solve the conflict. Otherwise the both *inactive* aircraft will collide.

Additional features of ACT-R are learning mechanisms to adjust WME and production parameters, partial matching, and the aggregation of production rules. These features are not used in our model.

THE MODEL

In this section, the psychological assumptions, based on experimental work and theoretical considerations, and the implementation of the main components and functions of the model MoFl are summarized.

MoFl describes three main cycles of information processing, (i.e., *monitoring*, *anticipation*, *problem resolution*) operating on different parts of the situation representation, called the *picture* (see Figure 1). The coordination of these processes is driven by *control procedures*. *Monitoring* and *anticipation* are diagnostic processes (conflict detection), *problem resolution* is the preparatory step for intervention by the controller.

The Monitoring Cycle: Data Selection and Update

The *monitoring cycle* includes data selection procedures and the regular update of aircraft features. In an experiment on data selection, 36 en route controllers had to control familiar and unfamiliar dynamic airspace situations. In order to investigate information selection, data of aircraft on the radar screen and the flight-strip-system were

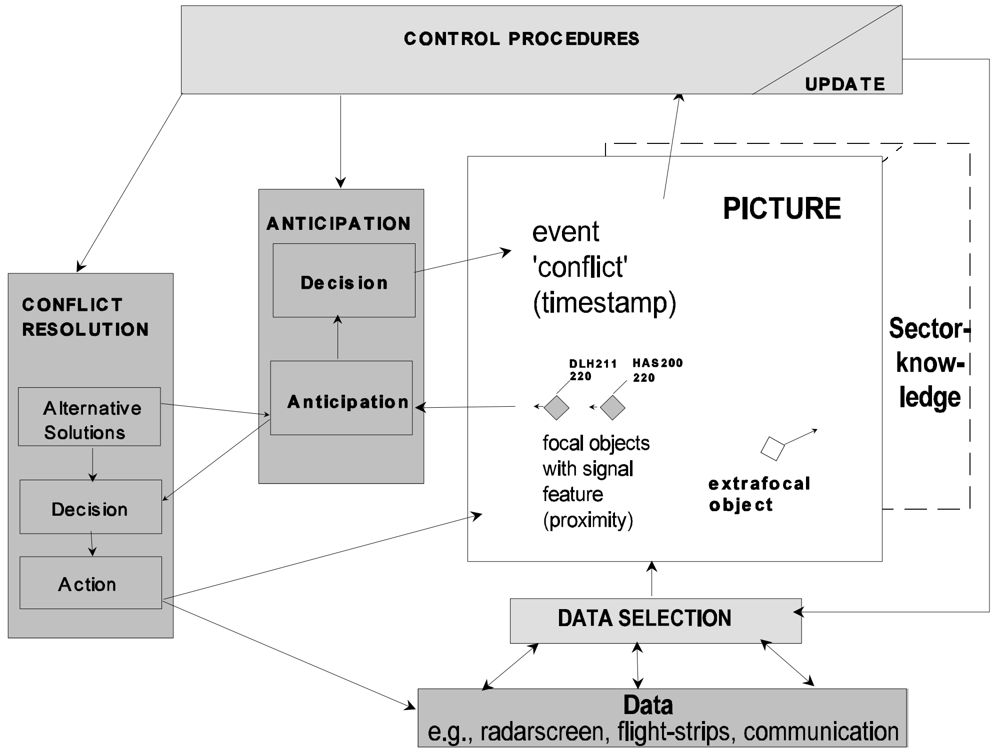

Figure 1: The structure of the air traffic controller's model of cognitive activities

masked, but could be unmasked by moving the pointer of the mouse to the respective location. Frequencies and durations of the unmasking were recorded. The data showed, that the representation of the current traffic situation was build up under considerable reduction of information. The controller selects relevant features of aircraft, especially identification codes, the horizontal and vertical positions of objects, and flight directions. In addition, our interviews and the literature indicate that the controller searches for meaningful signals in order to detect conflicts during radar-screening. These are aircraft features like vertical movements, proximity to other aircraft or to points in airspace where conflicts frequently occur (e.g., Niessen et al., 1997; Amaldi & Leroux 1995).

According to these signal features, aircraft become *focal* (highly activated), that means that they are attention demanding objects. Aircraft without such features are *extrafocal* (less activated). In the dynamic environment of air traffic control, objects have to be updated continuously. There is a relationship between the semantics of objects and the frequency of updating: focal, attention-demanding objects demand a higher monitoring frequency than extrafocal objects. This assumption has been supported by results of a memory test: positions of extrafocal (inrelevant) aircraft were reproduced back in time, whereas positions of attention demanding objects (e.g., conflictions, and climb or decend) were reproduced correctly (for similar results, see Boudes et al., 1995). This bias indicates, that there is an interaction between the semantics of objects and the updating frequency: the more the current position of aircraft demands attention the better they were reproduced.

The communication between the controller and the task environment, and the data selection were implemented as follows: Communication between MoFl and EnCoRe-PLuS is realized by socket communication. Two ways of communication are provided:

C *asynchronous communication*: Special events in the task environment, like pilot-initiated radio communication or signals suddenly appearing on the radar-screen, are announced to MoFl by EnCoRe-PLuS. After every application of a production rule, a Lisp function hooked to the ACT-R specific production-cycle-hook, checks for new messages and triggers appropriate Lisp call-back functions that create new WMEs for further processing.

C *synchronous communication*: MoFl identifies an internal demand for new information about a specific object within the task environment or the internal control-flow suggests to update aircraft information. This demand is fulfilled by an active request to the simulation environment. The response is integrated into the *picture* by call-back functions.

If the data selection procedures are triggered, appropriate goals are put onto the goal-stack to enable the following processing sequence:

1. *choose aircraft*: according to aircraft focality and state of the *picture*, decide which aircraft has to be updated.
2. *make an information request*: according to the state of the object which is going to be updated, choose

which information has to be requested, and trigger the appropriate Lisp function. The response of EnCoRe-PLuS is handled by a call-back function that generates a goal.

3. *take new information into the picture*: This goal is processed by a production that modifies the WME representing this information.
4. *test new data for signal features*: the updated WME is tested for changes of signal features such as changing flightlevel (vertical movement), or proximity to other aircraft.

Anticipation

The next step in diagnosis consists of an *anticipation cycle* which operates on the focal objects. For each attention-demanding (*focal*) aircraft or aircraft relationships, a future state is anticipated seperately. The goal of the anticipation cycle is to create new cognitive processing information about aircraft. Depending on the results of anticipation, aircraft with signal features can then be represented as *events*. An event reflects the type of relation between aircraft or relations between aircraft and airspace features in future time and space. The anticipation allows to decide (*decision*) if the future trajectories of aircraft result either in a conflict, in a safe separation, or the demand for more monitoring. In an experiment on conflict-management, different types of clearcut and potential conflicts were varied in a 70 minutes traffic scenario according to the *Eurocontrol Air Space Model* (EUROCONTROl, 1994). The EUROCONTROL classification has two dimensions: 1. different tracks (same, opposite, crossing), and 2. level- or climb/ decend-flight. 36 controllers had to detect and to solve the conflicts. The data showed that controllers did not differentiate between conflicts (separation minimum: 5 nautical miles) and potential conflicts (10 nautical miles): they intervened in all cases. This indicates that conflict detection is not based on a calculation but on fuzzy estimation. The controllers always chose the safer way by overestimating the risk.

We assume that, if a conflict is detected, the event *conflict* includes an estimation of the time remaining for conflict solution (*timestamp*). Relations which have proved to be safe, are no longer in the focal part of the *picture* and become extrafocal at this time. This indicates that there is almost no demand for cognitive processing, except for updating. If the operator is not sure about the potential conflict, the event *monitoring* becomes *focal*, indicating both a higher frequency of monitoring and also a high demand for further anticipation. This distinction of aircraft relationship has been supported by the results of a card sorting task with 18 air traffic controllers. As expected the controller showed a tendency to classify traffic scenarios on the basis of anticipation.

The anticipation cycle is implemented by sequenced production rules testing four questions:

1. Are aircraft on the same airway, or on crossing airways?
2. Have aircraft the same altitude or is at least one in climb or descend?
3. Simulation of the future movement of aircraft using *velocity leaders*. A velocity leader is an graphical arrow element on the radar screen showing the estimated movement of aircraft for a certain lapse of time. Will there be a violation of the separation criterion (*anticipation*)?
4. How certain was this simulation? Certainty is measured by the time remaining for the violation of the separation criteria. In addition the latest time for conflict solution is calculated (*timestamp*).

According to this sequence focality of aircraft-WMEs is modified, or events are created.

The Picture

The resulting *picture* is characterized as a representation of objects, events, and objects with reference to other objects, and / or airspace structure. Objects with signal features are represented focally, objects without these features extrafocally. In addition, events which indicate the meaning of aircraft relations in future time and space are represented focally. Within the air traffic control domain, the term *picture* describes the idea of a global mental representation of the current and future traffic situation in working memory. From a psychological perspective, we assume the *picture* as an analogous non-symbolic mental representation of the situation. There is some empirical evidence that experienced controllers anticipate future states of aircraft without calculating the trajectories. This indicates that they build up a non-metric, analogous representation of the situation. In assuming such an analogous representation, we follow Craik's (1943) and Johnson-Laird's (1983) basic ideas of a functional internal model that parallels processes of the external world.

The *picture*

- is understood as an active knowledge-based construction of meaningful relations between elements of a situation, and not as an addition of perceptions.
- is incomplete with regard to the content of information and is temporary. The representation is build up by schemata in order to serve current functions, and is not stored in long term memory.
- can be manipulated by drawing inferences, by making predictions, by understanding phenomena, by deciding what further processing or action to take, and by controlling the execution.

The implementation emulates the *picture* as the totality of the cognitively available objects at a given time, their features, and their perceived and infered relations in actual and future time and space in terms of WMEs. Since it is not possible to model an analogous representation of space on digital computers, the implementation's *picture* is a semantic net of airspace objects, anticipated events, and inferenced actions that are represented as WMEs. Some of these objects have spatial positions that make it possible to define them by positions. More sophisticated operations such as retrieval by distance to other airspace objects have to be emulated.

We used the object-oriented features of ACT-R to define the structure of the *picture* (see, Figure 2). Every airspace object has a position on the radar screen. Derived classes are *airways*, *sector boundaries*, and *aircraft* which have additional slots including callsign, speed, and altitude. Aircraft are specialized to *incoming*, *changing altitude*, and *near to another airspace object* (proximity). For every class, instances are generated and modified as WMEs in working memory by data selecting productions during the monitoring cycle. Events represent infered knowledge about aircraft. All events refer to aircraft objects. Instances are generated by production rules in the anticipation and conflict resolution module. They belong to the event-subclasses: *monitoring*, *conflict*, and *resolution*. Conflicts can be *crossing* or *chain*. Conflict events have an additional slot that holds a reference to the conflict partner.

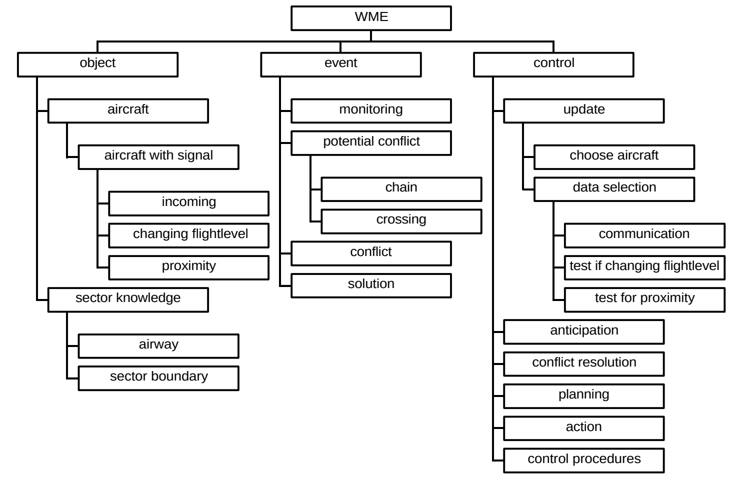

Figure 2: Simplified class hierarchy for the working memory elements

Conflict Resolution

If conflicts are detected, the *problem resolution cycle* initiates several steps to prevent an impending conflict. The controller has to select the most urgent conflict in order to generate or recall solutions (*alternative solutions*). Next, the operator has to check that the solution does not generate new follow up conflicts (*decision*). We assume that the controller checks by running a mental simulation of the solution (as in the *anticipation cycle*). The results of this model are executed (*action*).

The implementation uses a predefined set of standard solutions fitting certain types of conflicts. To use this set the class of the conflict is determined by production-rules. According to this classification some solutions are generated from the standard solution set. The production rules of the simulation in the *anticipation cycle* are triggered by goals indicating the solutions that have to be taken into account. If a solution does not produce follow-up conflicts a solution-WME is generated. A solution consists of a sequence of actions that have to be executed by the model. The time remaining for the first intervention of the sequence is stored in the solution-WME. To execute an intervention sequence Lisp functions interact with the task environment EnCoRe-PLuS.

Control Procedures

The multitude of represented objects, relations, and features within the *picture* demands that the controllers prioritize the processing at any one time. The coordination of the above describes modules (data selection and update, anticipation, conflict resolution, and action) is driven by *control procedures*. We assume that the different processing components cannot be interrupted. The controller has to switch between them: for example, between the solution of a conflict and further monitoring (update including data selection). On the basis of the state of the *picture*, control procedures select the most important and most urgent processing demand.

In ACT-R, Anderson postulates a hierarchical goal structure that directly reflects the task dependency in the environment. To model this hierarchy of goals, several WMEs can be pushed onto the goalstack, a special structure within working memory. Processing is controlled by the current goal, which is the first element of the goalstack. The current goal spreads activation among its neighbors in the semantic net. The system focusses only on this top goal at this time. But, because of the dynamic task environment of air traffic control, there is no fixed hierarchical goal structure. Therefore, the continuously changing situation demands another prioritizing of the processing of simultaneously on-going events at any particular time. In addition, time contraints in this context force a flexible and appropriate selection of the most relevant demand for processing. In order to model this contextualized scheduling of processing, we had to postulate a different concept. Our assumption is that the scheduling of processing is determined by the state of the whole mental representation of the traffic situation.

Several tasks are active at every moment. Every task is done by one of the modules *data selection*, *anticipation*, or *conflict resolution*. The superior control procedures module has to build up an ad hoc process flow depending on the current structure of the *picture*. To achieve this, we assume that the modules cannot be interrupted and are exclusive. The process flow is done by meta productions in the *control procedures* module that trigger a module with an object or event as parameter. In order to trigger a module and make it not interruptible, we introduced a new class of WMEs. These *control*-WMEs are the only ones that get onto the goalstack.

The start of every module is a *top level production*. It is triggered by a *top level goal*. This kind of production will push new subgoals onto the goalstack that will trigger other productions of that module. Every production has to clean the goalstack by popping its trigger-WME. When a module is finished the goalstack should then be clean. The productions of the *control procedures* are triggered by the *controlflow*-goal, which has no parameter. This goal is never popped. Thus when the goalstack is “clean” it is on top of the goalstack and thus the current goal triggers the *control procedures*-module again. Processing radio communication when a plane announces that it is going to enter the sector, is the only reason to interupt a module, make a mark in the working memory, and continue the module. The mark has a high priority so that it will be processed soon.

The meta production rules of the *control-flow*-module for the air traffic controller model use this prioritizing rules:

1. if a solution-WME exists in the *picture* and it is time to solve, then do *action* on this solution, else
2. if a conflict-WME exists and it is time to do, then *conflict resolution*, else
3. if a monitoring-event or an aircraft-WME with a signal (*incoming*, *changing altitude*, or *proximity*) exists in the *picture*, then do *update* and *anticipation* on this WME, else
4. if an aircraft-WME exists, then do *monitoring* on it.

Every solution-WME and every *conflict*-WME has a slot, where it represents when it is supposed to happen. The control productions use a function, that compares this ideal time with the current time. It fires the appropriate action according to a predefined bias.

If the current goal is *controlflow*, only the meta-productions are able to fire. They match patterns against the *picture* according to the prioritization scheme listed above. The chosen action will generate a new *control*-WME (CF) of the appropriate subclass. It refers to the detected aircraft-WME or event-WME. The goalstack consists now of (*controlflow*,CF). This triggers the toplevel production for CF. It will produce new control-WMEs probably refering to the detected WME, pop CF, and put the new control-WMEs onto the goalstack. They trigger new sublevel productions that all pop their trigger. When the module for CF is finished, the goalstack is (*controlflow*), meaning that only the meta-productions are able to fire.

The model deals well with the dynamic environment by using this control scheme. If another task needed interruptible modules, the control procedures would have to be triggered after every production cycle within the module, and the controlflow WMEs would have to be stored in the *picture*, when they are inactive. The meta productions would then trigger the most important controlflow-WME or generate a new one.

CONCLUDING REMARKS: EVALUATION OF THE MODEL

The construction and implementation of the above described model is based on a broad experimental work. Early in 1998 we will evaluate our model with empirical data. Three simulation experiments with experienced air traffic controllers are planned in order to investigate time parameters of conflict detection, the content of the *picture*, and the distribution of activation within the controller's *picture*. These data will be compared to the results of model simulation runs using the same task environment.

ACKNOWLEDGMENTS

This research project has been sponsored by the Deutsche Forschungsgemeinschaft, Ey 4/16-2 , Fr 375/48-2. We gratefully acknowledge the contributions of Thomas Bierwagen (DFS) to the conceptions of our model.

REFERENCES

Amaldi, P. & Leroux, M. (1995). Selecting Relevant Information in a Complex Environment: The Case of Air Traffic Control. In: Norros (Eds.), *5th European Conference on Cognitive Science Approaches to Process Control* (pp. 89-98). Finland: VTT Automation.

Anderson, J.R. (1993). *Rules of the Mind*. Hillsdale: Lawrence Erlbaum.

Bass, E.J., Baxter, G.D. & Ritter, F. (1995). Creating Models to Control Simulations: A Generic Approach. AI & Simulation, pp 18-25.

Bierwagen, T. (1996). Programmsystem EnCoRe-PLuS. Über die Möglichkeiten der programmierbaren Luftraumsimulation. *ZMMS-Forschungsbericht, 96-2*. Technische Universität Berlin.

Boudes, N., Amaldi, P., Cellier, J.M. & Leroux, M. (1995). Forseeing Judgement in an Informationally Rich Environment: The Case of Air Traffic Control. In: Norros (Eds.), *5th European Conference on Cognitive Science Approaches to Process Control* (pp. 76-88). Finland: VTT Automation.

Corker, K.M. & Smith, B.R. (1993). An Architecture and Model for Cognitive Engineering Simulation Analysis: Application to Advanced Aviation Automation. Presented at AIAA Conference on Computing in Aerospace, San Diege, CA. available at http://george.arc.nasa.gov/af/aff/midas/www/AIAA_Final.txt

Craik, K.J.W. (1943). *The Nature of Explanation*. Cambridge: University Press.

Endsley, M.R. (1995). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors, 37(1),* 32-64.

Eurocontrol (1994). The Eurocontrol Airspace Model: An Introduction for Potential Users. Eurocontrol-Workshop "Cognitive Aspects in ATC", May 1996.

Falzon, P. (1982). Display Structures: Compatibility with the Operator's Mental Representation and Reasoning Process. *Proceedings of the 2nd European Annual Conference on Human Decision Making and Manual Control*, pp 297-305.

Flach, J.M. (1995). Situation Awareness: Proceed with Caution. *Human Factors, 37(1),* 149-157.

Freed, M. & Johnston, J.C. (1995). Simulating Human Cognition in the Domain of Air Traffic Control. In M.T. Cox & M. Freed (Eds.), 1995 AAAI Symposium on Representing Mental States and Mechanism.

Menlo Park, CA: AAAI Press.

Hacker, W. (1978). *Allgemeine Arbeits- und Ingenieurspsychologie*. Bern: Hans Huber.

Johnson-Laird, P.N. (1983). *Mental Models*. Cambridge, Massachusetts: Harvard University Press.

Meyer, D.E. & Kieras, D.E. (1997a). A Computational Theory of Executive Processes and Multiple-Task Performance: Part 1. Basic Mechanisms. *Psychological Review, 104(1)*, pp 3-65.

Meyer, D.E. & Kieras, D.E. (1997b). A Computational Theory of Executive Processes and Multiple-Task Performance: Part 2. Accounts of Psychological Refractory-Period Phenomena. *Psychological Review, 104(4)*, pp. 749-791.

Meyer, D.E., Kieras, D.E., Lauber, E., Schumacher, E.H., Glass, J., Zurbriggen, E., Gmeindl, L. & Apfelblatt, D. (1995). Adaptive Executive Control: Flexible Multiple-Task Performance Without Pervasive Immutable Response-Selection Bottlenecks. *Acta Psychologica 90*, pp. 163-190.

Niessen, C., Eyferth, K. & Bierwagen, T. (1997). Modelling Cognitive Processes of Experienced Air Traffic Controller. In: S. Bagnara, E. Hollnagel, M. Mariani & L. Norros (Eds.), *Sixth European Conference on Cognitive Science Approaches to Process Control, Time and Space in Process Control*, 23.9.-26.9.1997 Baveno, Italy.

Opwis, K. & Spada, H. (1994). Modellierung mit Hilfe wissensbasierter Systeme. In: *Enzyklopädie der Psychologie* (pp 199-248). Göttingen: Hogrefe.

Rasmussen, J. (1986). *Information Processing and Human-Machine Interaction.* New York: North-Holland.

Whitfield, D. & Jackson, A. (1982). The Air Traffic Controller's Picture As an Example of Mental Model. In: G. Johannsen & J. E. Rijnsdorp (Eds.), *Proceedings of the IFAC Conference on Analysis, Design and, Evaluation of Man-Machine Systems* (pp 45-52). London: Pergamon Press.