

An approach to cognitive simulation of air traffic controllers based on coloured petri nets

Hardy Smieszek* Christian Karl**

*Technische Universität Berlin, research group prometei, Berlin, 10587

Germany (Tel: +49 30-314-29635; e-mail: hardy.smieszek@zmms.tu-berlin.de).

** Technische Universität Berlin, center of human-machine systems, Berlin, 10581

Germany (e-mail:christian.karl@zmms.tu-berlin.de)

Abstract: Because of intensively growing volumes of air traffic, aerodrome air traffic controllers face increasing workloads, having to deal with more aircraft in the same time. Computerized supporting systems are often meant to compensate for these effects. However, to investigate the actual impact of changed system conditions on the human operator, simulations are needed. Consequently, it is necessary to model the cognitive processes and behaviour of the air traffic controller (ATCO) as well as the airport traffic control system in general. Based on basic guidelines for the development of cognitive simulation, an approach for the development of a timed coloured petri net Model of Airport TRaffIc Control System (MATrICS) is presented. It displays ATCO and airport processes, and supports investigating the performance of both the airport traffic control system in general as well as the ATCO's performance under changed system conditions such as computer support.

Keywords: Cognitive Simulation, Aerodrome air traffic control, Human-Machine Systems

1. INTRODUCTION

Facing steadily growing volumes of air traffic, ground controllers encounter increasing efficiency demands and excessive workloads as they have to deal with more aircraft within the same time. At present, aviation is one of the safest ways of transport. Following Hollnagel, Woods, and Leveson (2006), it is the inherent resilience of the system that makes it safe. In aviation, it is the variability of human performance which enables air traffic controllers (ATCOs) and pilots most of the time to act in a safe way and to take the right action at the right time (Stroeve, Everdij and Bolm, 2011). Above all, ATCOs are responsible for the safe and efficient handling of air traffic where they regularly have to reach a trade-off between efficiency and thoroughness if they are to be successful (Hollnagel, 2009). With increasing air traffic, this trade-off has to be shifted towards efficiency, which can lead to erroneous actions and increasing numbers of incidents and accidents. To help ATCOs conduct their tasks safely and efficiently, changes to the system or the system's conditions will be introduced, e.g. by implementing computerized supporting systems like Advanced Surface Movement Guidance and Control System (A-SMGCS; EATMP, 2005), Integrated Tower Working Position (ITWP; Dubuisson, 2006), or Arrival and Departure Manager (AMAN; Hasevoets and Conroy, 2010, DMAN Dubouchet and Mavoian, 1999). For investigating the impact of changing system conditions on the human operator, not only a model of the system but of the human behaviour and performance is needed. The theoretical approach for the development of such a model and simulation, the MATrICS Model of Airport TRaffIc Control System, as well as the current model version will be presented. The paper is organized as follows: Section

2 gives a brief description of the most important facts about aerodrome air traffic control and principles of Cognitive Simulation. Section 3 describes the individual steps of the development process of Cognitive Simulation according to Cacciabue (1998) in some detail. In section 4, the framework of coloured petri nets is introduced, which is used for modelling and simulation purpose. The description of the several parts of the current model version and the construction process follows in section 5. Section 6 summarises the current work status while section 7 discusses further steps.

2. AERODROME ATC AND COGNITIVE SIMULATION

2.1 Aerodrome air traffic control - a human-machine-system

Air traffic control is often seen as a highly automated sector. However, if the level of automation is classified e.g. after Sheridan (1992), it must be rated as very low (Bierwagen, 2005). Even though new computerised supporting systems are being introduced, full automation of aerodrome air traffic control is neither possible nor envisaged (Hopkin, 2010). Hence, in this complex dynamic system the human operator will always remain responsible for safe and efficient handling of air traffic. At an airport, mainly two positions can be distinguished: The ground controller (GND) and the tower controller (TWR). The latter term is rather confusing as both positions are actually located in the tower. When referring to the positions TWR and GND, we distinguish the controller being responsible for handling landings and take-offs on the runway - the TWR-position, and the controller, responsible for handling all aircraft and vehicle movements on taxiways - the GND position. Although the level of automation is ranged

as low, there are already some tools available which the ATCO can use for his work (which can vary from tower to tower): the ground movement radar, separated radar for the landing traffic, a weather information display, the far view on the airport, a telephone and paper flight (progression) strips (Tavanti and Bourgois, 2006b). The latter is one of the most important tools for the ATCOs. Flight strips “constitute a highly flexible means of planning departure sequences and can be written on, re-ordered and otherwise physically manipulated in this process” (Fields, Amaldi, and Tassi, 2003 after Tavanti and Bourgois, 2006a). The control of aircraft is handed over between controller positions, depending on the aircraft’s status. The possession of a flight strip hence represents controlling the corresponding aircraft; accordingly the process of handing over the control of an aircraft is accompanied by handing over the specific flight strip.

2.2. Principles of Cognitive Simulation

To gather insight into which effects changing system conditions will have on the performance of both the overall system and the human, models and simulations can be used. Thereby a model is “essentially a theoretical account of a process or a system based on a number of hypotheses, conservation principles, and simplifying assumptions...” (Cacciabue, 1998, p. 4), whereas a simulation is an “expression or implementation of a model in a form that is controllable or executable by numerical application or computation.” (Cacciabue, 1998, p. 5). The described *MATrICS* (Model of Airport Traffic Control System) model shall not only describe cognition as it is assumed to take place within the human mind in a microcognitive way, it is also to provide a description of the system as a whole. Therefore a macrocognitive approach is followed, which aims at constructing a model of the human-machine-system aerodrome air traffic control as a whole. Nevertheless, some simplifications have to be made. Macrocognition and microcognition were defined by Cacciabue and Hollnagel (1995) as two basic modelling approaches. Modelling in terms of microcognition deals with investigating how cognition takes place exclusively within the human mind. It emphasises the scientific account of models to gather a deeper understanding of a phenomenon. Macrocognition on the other hand refers to the study of cognition in real tasks, being in interaction with the environment. Macrocognition “rarely looks at phenomena that take place exclusively within the human mind” [...] “but neither advocates the behaviourist ‘black box’ approach” (Cacciabue and Erik Hollnagel, 1995, p. 57). It is also concerned with the mechanisms that are inside the “box”, but only as far as they are able to explain, describe and eventually predict the resulting behaviour. Through this distinction regarding the objective of the modelling purpose, Cacciabue (1998) also introduces the term *Cognitive Simulation* in distinction to *Cognitive Modelling* which is defined as: “... replication, by means of computer programs, of the performance of a person in a selected set of situations. [...] The minimum requirement to the simulation is that it produces the response the person would give. In addition the simulation may also produce a trace of the changing internal mental states of the person.” (Cacciabue and Hollnagel, 1995, p. 58).

3. DEVELOPMENT OF COGNITIVE SIMULATION

3.1 Guidelines

Basic guidelines for the development of cognitive simulation are given by Cacciabue (1998). They are summarized in Fig. 1. As the first step, problem boundaries and the aim of the simulation must be defined. Afterwards, a cognitive task analysis and a field study of working context shall be conducted. The gathered data are then integrated into a theoretical model which enables a descriptive simulation. Since *MATrICS* intends a numerical simulation as well, an appropriate numerical algorithm must be selected in the next step. The implementation in a programming language is the actual model construction, which constitutes the basis for a numerical simulation. As can be seen in the figure below, development of Cognitive Simulation is an iterative process and is not intended to be strictly followed in chronological order. Every step of the development framework allows a step back or forth to alter assumptions. For the *MATrICS* project, these steps are now explained in further detail.

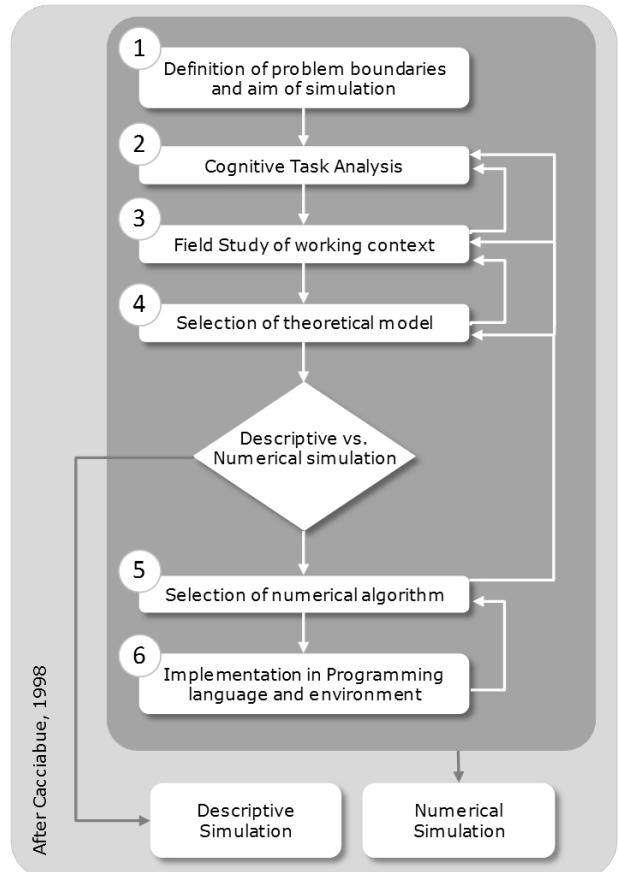


Fig. 1. Basic guidelines for the development of cognitive simulation (according to Cacciabue, 1998).

3.2 Definition of problem boundaries and aim of simulation

To gather a model and simulation in terms of macrocognition, it is necessary to describe not only one aspect of the modelled system (e.g. cognition of the ATCOs) but preferably all of them, so that a holistic description of the system is obtained. Thus, the airport traffic control system in

general, the cognitive processes and behaviour of the ATCOs as well as the interactions between these two have to be described. Cacciabue (1998) suggests dividing models of human-machine-systems into three parts: The Human model, the machine model and the interaction model (Fig. 2). The human model describes the two controllers - their cognitive processes and behaviour (ATCO-Model). The machine model, the way it is within the *MATriCS* project, describes the airport process (AP-Model). Finally, the interaction model (IA-Model) describes interactions between airport and ATCOs as well as the tools being used by both processes to interact with each other and implement changes. For example, the control actions of the ATCOs altering the airport process can be done by instructing the pilot via voice communication.

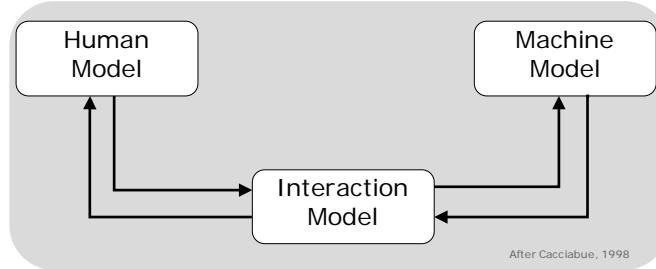


Fig. 2. The Architecture of Human-Machine System Interaction (according to Cacciabue, 1998)

The implementation of *MATriCS* shall provide a visual simulation of specific traffic scenarios in real-time. Besides it should be possible to analyse the performance of the overall system (e.g. extraction of taxi times and delays), as well as the human operator (e.g. work- or taskload estimation). For simplification reasons, only the tasks of the reduced inbound-outbound-process shall be considered as suggested by Werther (2006) and Werther, Moehlenbrink, and Rudolph (2007). That means that solely those tasks of the ATCOs are considered, which directly contribute to the handling of aircraft. Maintenance or emergency tasks will not be included. Furthermore, *MATriCS* focuses on the controllers as the human part of the ATC system. Pilots are considered as part of the aircraft and further as part of the airport process and therefore are not modelled in further detail. All three model parts shall be described using one and the same modelling language. Finally *MATriCS* will be implemented for Berlin-Brandenburg airport (BER), providing a means of analysing realistic systems. Anyhow, it shall as well be easily adjustable to other airports.

3.3 Cognitive task analysis and field study of working context

For the (cognitive) task analysis as the second step, existing task analyses conducted in several sites across Europe have been summarized, including analyses by EUROCONTROL (Buck et al. 1996; Tavanti and Bourgois, 2006a), German Air Navigation Services (DFS) (Human-Factors-Consult (HFC), 2009) and Royal Air Force Institute of Aviation Medicine (Cox, 1994a; Cox, 1994b). Through these analyses, a widespread illustration of the tasks of the ATCOs has been generated, which is further used for constructing the theoretical model (section 3.4). Within the constraints given

through the definition of the problem boundaries (section 3.2) four main-tasks were identified for the GND position according to Cox (1994b):

1. Formulate plan to integrate movements of arrivals and departures
2. Conduct direction of arrivals to stands
3. Initiate push back (see Fig. 3)
4. Conduct direction of departures to runway

For the TWR position three main-tasks were identified:

1. Formulate plan to integrate arrivals and departures
2. Conduct aircraft departures
3. Conduct aircraft arrivals

To gather more information about the working context and the constraints and regulations of the system, a field study of working context has been conducted in the tower of Schönefeld airport in Berlin. Furthermore, several regulations concerning the aerodrome air traffic control system (such as DFS, 2006; ICAO4444, 2007) as well as descriptions about role allocation and supporting systems (Tavanti and Bourgois, 2006b) have been analysed. The (cognitive) task analyses are the basis for constructing the theoretical model (and the human model). The field study of working context will form the basis for the construction of the airport model and the interaction model.

3.4 Selection of theoretical model

The main- and sub-tasks of the ATCOs analysed in the former step were integrated into a theoretical model based on action regulation theory after Hacker (2006), Dörner and Schaub (1995) and Frese and Zapf (1994), to gather a better understanding which sub-task can be assigned to which cognitive process. A theoretical model of the action phases of the GND position based on action regulation theory has already been introduced in earlier research (Smieszek, Huber and Jürgensohn, 2011). The same was done for the TWR position as well. The revised theoretical model describes six action phases of the ATCOs: The first phase is the *definition of the objective*. Within the context of air traffic control, this phase is not performed by the controller himself, but here objectives are predefined which lead to a goal setting that is superior to the other five phases. Objectives of the ATCOs are, within the ICAO4444 procedures for air navigation services, described as follows: "... achieve a safe, orderly and expeditious flow of air traffic on and in the vicinity of an aerodrome with the object of preventing collision(s)..." (ICAO4444, 2007, p. 7-1). The second phase is the *collection of information and orientation* and describes the acquisition of information and the construction of an adequate mental model. To gather information, the ATCO uses several tools as already described in section 2.1. The third phase is aimed at *planning*. It contains the equation of the current traffic situation with the available slots and leads to the establishment of an aircraft separation plan. Based on the mental model gathered in phase two, the actual traffic forecast and planning takes place. Phase four is the actual *decision for an action alternative*, determining which action is taken in the following *phase of action implementation*. Through issuing several clearances (push back, take-off, landing) to various receivers (aircraft, ground traffic) the

ATCO directly intervenes and alters the traffic situation. Within the last *phase of feedback processing*, the altered traffic situation is compared to the planned situation and discrepancies like plan deviations or separation violations can be recognised and included into the new action cycle. Within the described action phases, several sub-tasks of the ATCOs were integrated to gather a more detailed description of the ATCOs work. By doing so, a prototypical action sequence was created as can be seen in Fig. 3, which forms a basis for

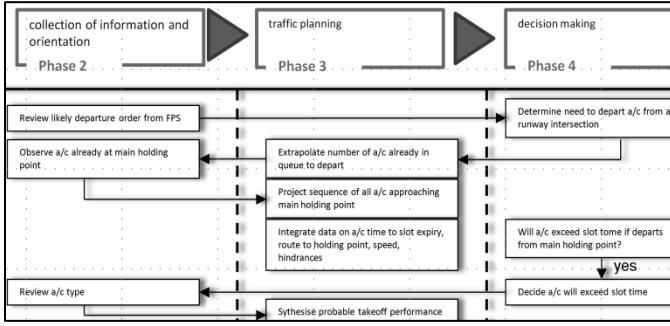


Fig. 3. Example action sequence for main task two (conduct a/c departure) of the TWR-position

the construction of the human model of *MATriCS*. To gather more detailed information about how cognition proceeds within the individual action phases, more specific theoretical models will be used as explained later in this paper.

3.5 Selection of numerical algorithm

Before implementing the actual *MATriCS*-model, an adequate modelling language must be chosen, which is capable of accomplishing the requirements defined in the first step (section 3.2). Therefore several methods for modelling human behaviour and cognitive processes were compared in terms of their capabilities of fulfilling the defined requirements of *MATriCS*. In doing so, cognitive architectures (like ACT-R (Anderson et al., 2004) and CLARION (Sun, 2006)), fuzzy logic (Zadeh, 1965) and petri nets among others were compared. Describing all the advantages and disadvantages of the several methods goes beyond the scope of this paper. Instead it will be described why the framework of coloured petri nets (CPNs, Jensen, 1997) was chosen. First at all, when talking about petri nets, the term “numerical algorithm” is not exactly appropriate. Petri nets have a mathematical foundation and can be described formally. Nonetheless knowledge about the mathematical foundation is not required for understanding petri net-models and not even for applying the framework. The intuitively intelligible graphical representation is a big advantage when it comes to presenting and discussing models and simulations by means of coloured petri nets with persons not familiar with the framework. Furthermore, the mathematical description allows analyses of the petri net-models in terms of mathematical descriptions or state space methods. That way, petri nets make it possible to achieve both outcomes of the Cacciabue framework: A descriptive simulation as well as a numerical simulation. As coloured petri nets offer the possibility of introducing the factor time into the net (timed coloured petri nets), at first it is possible to

represent time and at second a simulation of the modelled system can be performed in real time. This property is especially essential when performance analysis is required. Also, petri nets are stable towards minor changes of the modelled system which means that small modifications of the modelled system do not change the overall structure of the CPN (Jensen, 1997). This is especially interesting when it comes to altering the model. In terms of *MATriCS* this means, that the model can easily be adjusted to represent other airports, which are slightly different, or to represent diverse controllers with individual characteristics. This enables the researcher to describe varying airports and ATCOs using the same general net structure, implementing only a few changes. There are several research projects that use the framework of CPNs (or petri nets in general) for modelling different parts of the aviation system (e.g. (Ruckdeschel and Onken, 1994; Ruckdeschel, 1997; Oberheid and Söffker, 2008; Vidosavljević and Tošić, 2010; Kovács et al., 2005) or for modelling human cognition and behaviour (Werther, 2006b; Werther, 2006a; Werther et al., 2007). Finally, powerful computer tools, such as CPN-tools (CPN-Tools, n.d.), support the construction and analysis of timed coloured petri net models.

3.6 Implementation in programming language and environment

The sixth step of the Cacciabue framework is concerned with the actual implementation of *MATriCS* using the chosen modelling language. It is therefore the essential, most difficult and most time consuming step. All other described work can be seen as preparatory work leading to the construction of *MATriCS*. All knowledge gathered through the preceding steps flows into this one. However, there can be few small changes in the steps taken beforehand, as the development process is an iterative one. This means that several simplifications can be made concerning the task analysis (e.g. reduce tasks for simplification reasons), or that several other theoretical models can be used for describing cognitive processes (e.g. perception) more detailed. The model construction will be done using *CPN Tools*, which is an excellent tool for constructing CPN-models. In the following paragraph the framework of petri nets and coloured petri nets is described in more detail.

4. COLOURED PETRI NETS

4.1 What is a petri net?

One definition of a “core” petri net reads as follows: „A Petri net is a directed graph with two kinds of nodes, interpreted as places and transitions, such that no arc connects two nodes of the same kind.“ (Desel and Juhás, 2001, p.5). Thereby, places (drawn as circles, see Fig. 4) represent conditions or states of a specific system, and transitions (drawn as rectangles) represent crossings or processes of the system. Furthermore arcs indicate an abstract relation between components and the direction of flow. Tokens (drawn as black dots) represent the presence of specific resources or data. When a token is present on a place, the condition modelled by the place is fulfilled and a specific process (the transition) can be

executed. The transition occurs or it *fires*. If the transition fires, information in the neighbouring places change: e.g. it consumes the token from place P1 and produces a new token on place P2 (as shown in Fig. 5). The depicted nets of Figure 4 and 5 are basic nets. There are several extensions and varieties

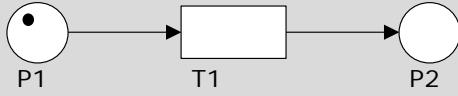


Fig. 6. Example of an ordinary petri net

of petri net classes, such that the term petri net is more a description of the basic concept, underlying a variety of extensions, than the name of a specific modelling language.

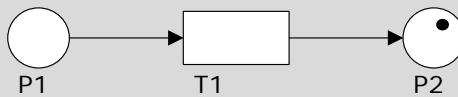


Fig. 4. The example net after occurrence of T1

Within the framework of petri nets, Kurt Jensen developed the class of coloured petri nets in 1979 within his Ph.D. thesis. CPNs are a graphical modelling language for concurrent systems. It states a combination of ordinary petri nets with a programming language called CPN ML. CPNs are a compact and easy to understand method for modelling processes within complex dynamical systems with limited resources in discrete time intervals (Jensen, 1997). As in ordinary petri nets, the nodes of the net are divided: Transitions on the one hand, represented as rectangles, which describe the processes of the system and on the other hand places, represented as cycles or ellipses, representing

the property of token 2 being “Token2”. Therefore, places have an additional property, the so called colour set, which can be compared to variable types in ordinary programming languages. The colour set is shown in Fig. 6 and Fig. 7 as “PLACECOLOR”. This means that at place 1 only those tokens can be present, which are of the colour set “PLACECOLOR”. The nodes are connected via directed arcs, such that “no arc connects two nodes of the same kind” as stated in the definition. The dynamic behaviour of the modelled system is represented through token games (movement of tokens through the net while simulation). The distribution of tokens across the net is called a marking and represents the current state of the system. If all input places of a transition are occupied with a token, the transition is enabled and can fire. This means that all preconditions of a process or an action are fulfilled in the system and so it can proceed or the action can be executed. After the transition has occurred, a new system state is generated.

4.2 Hierarchical coloured petri nets

Since models of complex dynamic systems would quickly become inconvenient and confusing, the concept of hierarchical coloured petri nets is introduced. Complex CPN-models can be organised in sub-modules, providing the human modeller with higher levels of abstractions and therefore only a few details at a time. Using this hierarchical structure, it is possible to construct a large model by combining a number of small ones. These models can be seen as black boxes, where modellers can forget about the details within the modules if intended. This enables the modeller to work on different abstraction levels (Jensen and Kristensen, 2009). To represent sub-modules, so-called substitution transitions are introduced. These transitions allow the user to relate a transition to a more complex CPN, which usually gives a more precise and detailed description of the activity represented by the substitution transition (Jensen, 1997). The depicted substitution transition shown in Fig. 8 is

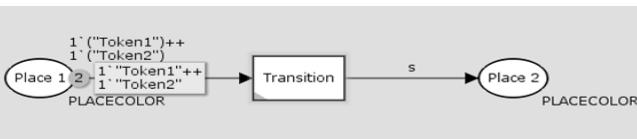


Fig. 5. Example of a coloured petri net

the states of the system. As it can be seen in Fig. 6, in CPNs tokens are not only represented as black dots, but they also represent the number of tokens and a so called token colour, which may be arbitrary complex data values. The current marking of a given place is therefore represented by a small circle, with an integer saying how many tokens there are, and

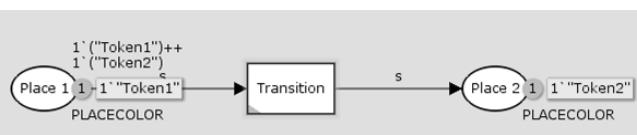


Fig. 7. The example CPN after transition occurrence

a text string next to the cycle with a multi-set, saying what the individual token colours are. In Fig. 6 and Fig. 7 there are two tokens with the property of token 1 being “Token1” and

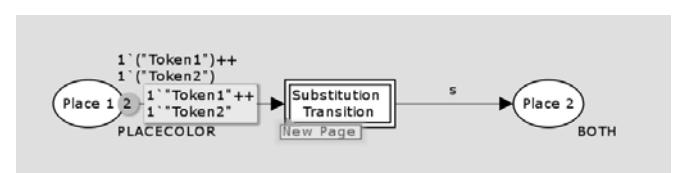


Fig. 8. : Example of a hierarchical CPN

represented as a double framed rectangle. It is not shown what exactly happens when the substitution transition fires, but if intended, the underlying model can be viewed. A more detailed introduction into CPNs and hierarchical CPNs can be found in Jensen (1997) and Jensen and Kristensen (2009).

5. IMPLEMENTATION OF MATriCS

5.1 Twelve steps for the construction of CPN-Models

Jensen (1997) suggests twelve steps which should be followed when constructing CPN-models, which are also applied within the *MATriCS* project. These steps are as follows:

1. Start by identifying some of the most important components of the modelled system
2. Consider the purpose of your model and determine an adequate level of detail
3. Try to find good mnemonic names for objects, processes, states and actions
4. Do not attempt to cover all aspects of the considered system in the first version of your model
5. Choose one of the processes in the modelled system and try to make an isolated net for this process
6. Use the net structure to model control and the net inscriptions to model data manipulations
7. Distinguish between different kinds of token
8. Use different kinds of colour sets
9. Augment the process net by describing how the process communicates/interacts with other processes
10. Investigate whether there are classes of similar processes
11. Combine the subnets of the individual processes to a large model
12. Making a CPN model is very similar to the construction of a program

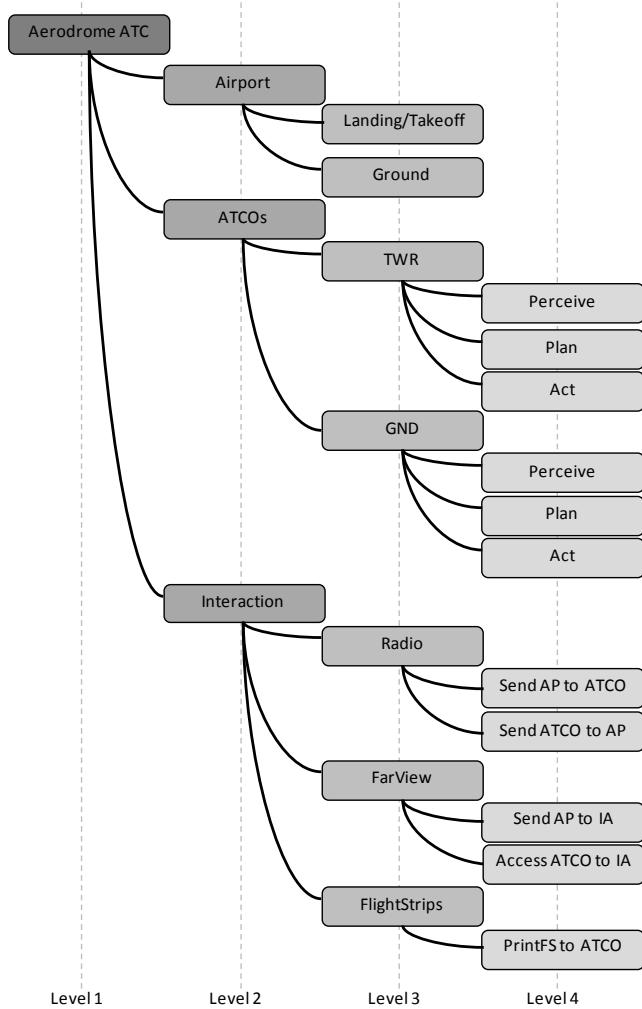


Fig. 9. Page hierarchy of *MATriCS* CP-Net model

Not all twelve points will be discussed in detail, because many of them are self-evident and can easily be done while

constructing models. However, the identification of the modelled system's most important components is an important precondition for the construction of *MATriCS*. Therefore, a workshop has been conducted, aimed at identifying these components and allocating them either to places or transitions for the three parts of *MATriCS*: AP-Model, IA-Model and ATCO-Model. CPN-tools was used for implementing the initial four *MATriCS*-Versions. The current version 0.4 will be described here.

5.2 Identification and allocation of the most important components of the system

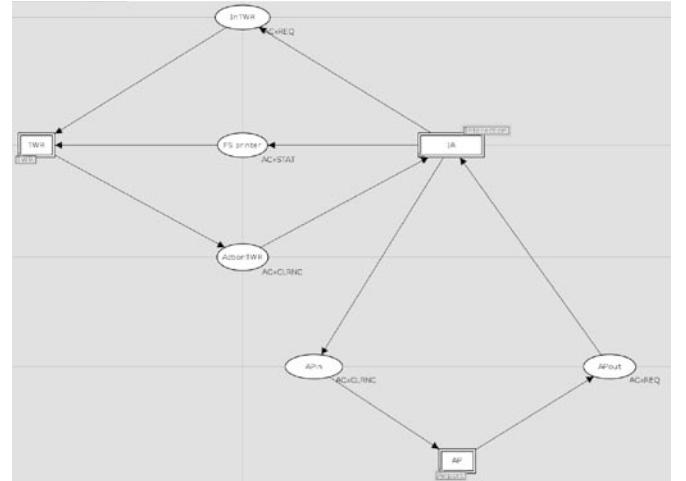


Fig. 10. Super net of version 0.4 of *MATriCS*

As already described in section 3.2, *MATriCS* was divided into three sub-models: The IA-model, the AP-model and the ATCO-model. Considering a real airport, the airport process and the controller are also relatively independent of each other. They can act and proceed individually for a given period of time. Nevertheless, they can correspond by using interaction tools. The airport process can influence the actions of the controller, e.g. an aircraft calling for landing clearance over RTF; and the controller can influence the airport process, e.g. by giving clearances to an aircraft. Also,

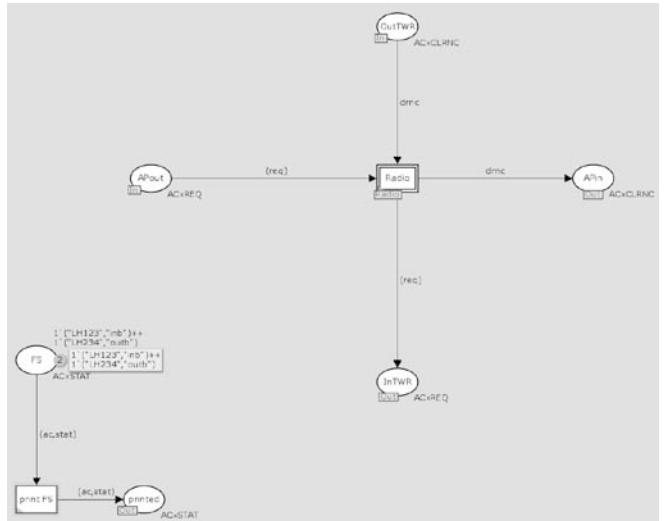


Fig. 11. Interaction model of version 0.4 of *MATriCS*

the ATCO can actively acquire information about the airport process, altering his own behaviour though not influencing the airport process. The ATCO-model for his part is divided into two different models: The TWR-model and the GND-model, as in most of the towers such an allocation of tasks is common practice (as described in section 2.1). Therefore, both ATCO-models are similar but slightly different. Both models were constructed using the task analysis described in section 3.3. In the following sections, the three *MATriCS*-parts will be described in further detail. A description of the more detailed hierarchy of *MATriCS* can be seen in Fig. 9. It must be pointed out, that the current version (0.4) of *MATriCS* is not final yet. The modelling process continues. That is why only the TWR-model is shown in the super net of version 0.4 in Fig. 10. This super net shows the currently three sub-modules TWR, AP and Interaction.

5.3 The Interaction model (IA-Model)

All interactions between airport and controller are indirect, carried out by using interaction devices like radio telephone, radar or the outside view. These interaction devices are modelled within the interaction model (Fig. 11). Thereby, places are concerned with the inputs and outputs of the IA-model, either coming from the ATCO-models (TWR or GND) and going into the AP-model, or coming from the AP-model and going into the ATCO-models. Transitions are concerned with modelling the actual processes like communicating, looking outside the window or printing flight strips. These processes can be extended further to describe characteristics of the used technical systems in more detail in later versions of *MATriCS*, e.g. the quality of the communication devices and their characteristics or the visibility conditions of the outside view. The current version of the interaction model describes the transmission of aircraft requests (either landing or taking off) and the clearance delivery by the TWR-controller both via radio communication. Furthermore, a flight strip printing device is modelled. For evaluating future systems, models of such systems can be implemented within the interaction model, changing the type of work the controller has to do.

5.4 The airport model (AP-Model)

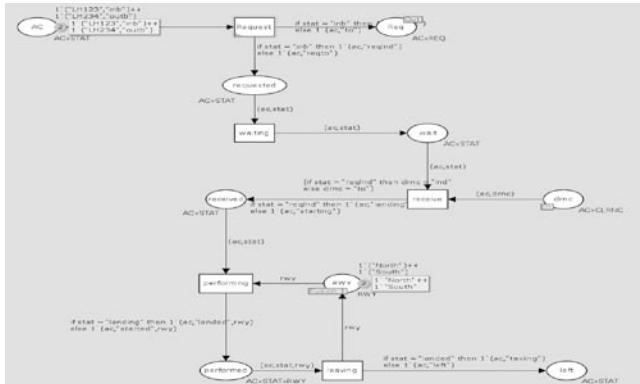


Fig. 12. Airport model of version 0.4 of *MATrICS*

The AP-model describes the airport processes and therefore the traffic evolution (Fig. 12). The states an aircraft can be in (e.g. approaching, called for landing clearance) are modelled as places; the actual processes conducted (e.g. landing, taxiing) are modelled as transitions. Places are a sort of resource reservoir representing runways (RWYs), taxiways (TXYs) and gates such that a free RWY is indicated as a place where a token is present, and an occupied RWY is a place with no present token (Fig. 12; Place "RWY"). The same will apply for TXYs and gates, when the model is extended to ground traffic. Aircraft are modelled as tokens which carry a call sign (e.g. LH123) and a status (e.g. approaching, landing). These tokens are forwarded through the model, such that the status is changing when an action of an aircraft occurs and the traffic evolution is observable. Furthermore, the AP-model needs tokens from the TWR-model in terms of clearances, which are gathered through the IA-model. Equally, the AP-model can send tokens to the TWR-model via the IA-model in terms of requests. Without sending and getting these tokens, no further processes could occur, that means no clearances could be given and no landings and takeoffs could proceed. The current version of the airport model is a rather arbitrary one. It will be adjusted to represent the structure of BER-airport.

5.5 Model of the TWR-Position (TWR-Model)

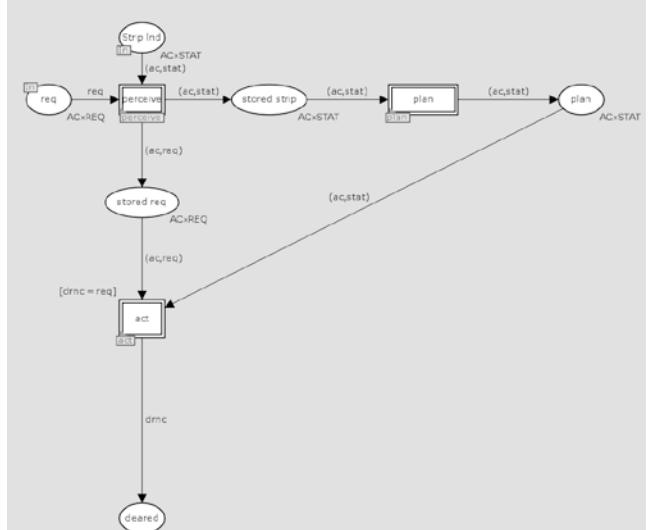


Fig. 13. TWR model of version 0.4 of *MATriCS*

The model of the TWR-position is of course the most complicated one. It is divided into sub-models, describing the individual phases of action regulation as described in section 3.4. Currently, the three phases of *perceive* (phase 2), *plan* (phase 3) and *act* (phase 5) are implemented and consist of sub-nets as can be seen in Fig. 13. The “*perceive*”-subnet (Fig. 14) is concerned with perceiving information coming from the interaction-model. Thus it has currently two inputs, an auditory and a visual one. Through the visual input information from the flight strips can be perceived. Flight strips, as explained in section 2.1, are a means of planning and are received some minutes before the actual aircraft arrives at the concerned sector. The current pieces of information which are contents of the modelled flight strip,

are the aircraft call sign and the intended action (landing or take off). The auditory input aims at perceiving requests coming from the pilots (and therefore coming from the airport model) via the modelled radio communication device. According to Wickens (1984) and Wickens and Hollands (1999), auditory and visual information can be acquired concurrently as it can be assumed, that they access different information processing sources. Both auditory and visual information inputs are then stored in the working memory, which is not yet explicitly modelled. Instead a capacity place is introduced, modelling limited information processing capacity (see Fig. 14).

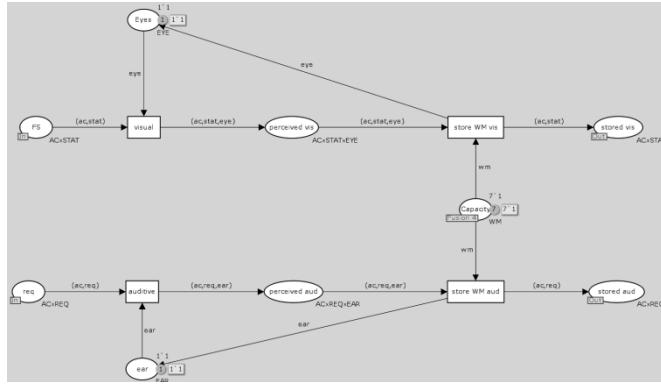


Fig. 14. Perceive subnet of version 0.4 of *MATriCS*

According to Miller (1956), only seven pieces of information can be stored simultaneously, therefore the maximum capacity number of the place is seven. After the information is stored, it is available for the following planning process (Fig. 15). The model describes the process of sorting the flight strips by their arrival and departure times. Therefore, it is defined that incoming flights take priority over outgoing flights. A take-off will only be possible, if there is enough time available for the departing aircraft to finish the take-off before the next landing aircraft will arrive. This is common

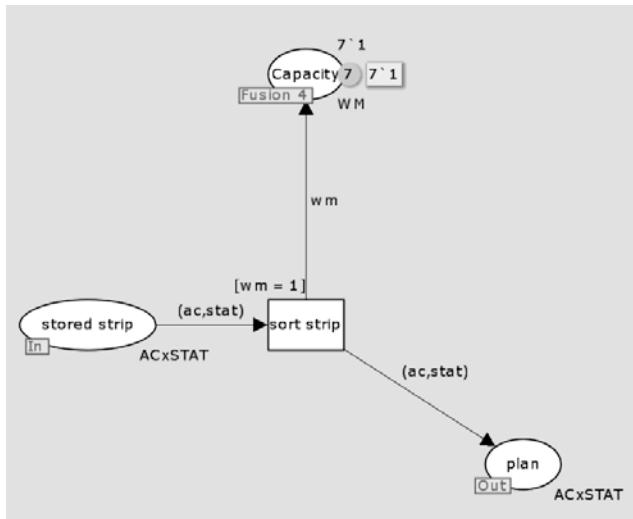


Fig. 15. Plan subnet of version 0.4 of *MATriCS*

practice when controlling airport traffic. The planning algorithm is as follows: At a given model time the time of the aircraft arriving next () is selected and it is calculated, if

the next scheduled departing aircraft is clear off the runway before this time (). Therefore the following condition must be fulfilled:

If the condition is fulfilled, then take-off is possible, else, only landing is possible and the departing aircraft has to wait. Currently there is no such planning algorithm realised, as time is not yet associated with the model. Only a process of forwarding the stored information from the "stored strip" place to the "plan" place and relieving one piece

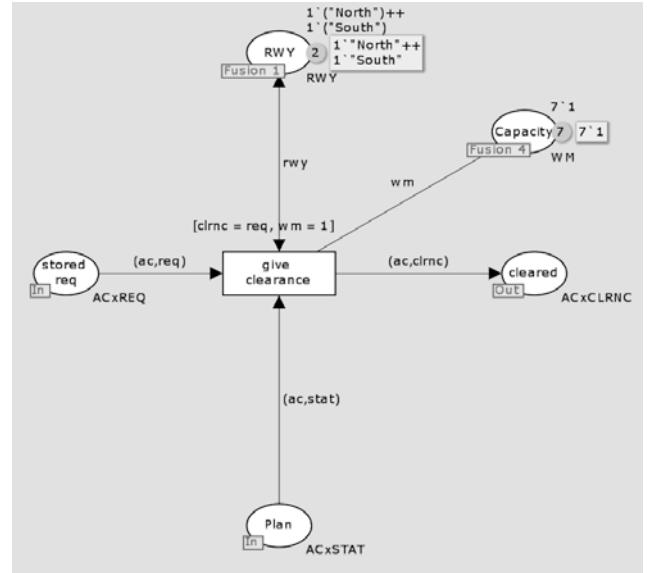


Fig. 16. Act subnet of version 0.4 of *MATriCS*

of information capacity to the modelled working memory is conducted by the "plan"-model. All information gathered through the "perceive" and the "plan"-model are then put together to create an appropriate action (i.e. giving an appropriate clearance) in the "act"-subnet (Fig. 16). Therefore, the stored request is compared to the situation according to the plan. This means that it is checked, which clearance the aircraft has requested and if the next clearance according to the plan is the same as the one which was requested. If so, the clearance is given, if not, no clearance is given. Furthermore, before a clearance can be given, it must be checked, if the runway is available. This is currently modelled by a fusion place "RWY" which always consists of the same marking as the original "RWY"-place in the airport model. In later versions of *MATriCS* there will be a visual check of availability of the runway via the modelled far view within the interaction model. Additionally it is assumed, that if the clearance is given, the modelled controller does not need to know anymore, which aircraft has requested which clearance. So the information space is released again and the capacity of the modelled working memory is increased by one. The clearance, in turn, is then submitted to the interaction model via the output place "cleared" and forwarded through the interaction model into the airport model. Only then the airport process can proceed further and the aircraft can perform (conduct either landing or take-off).

6. CONCLUSION

The modelling-project called *MATriCS* has been presented, which is aimed at constructing a macrocognitive model and simulation of the airport traffic control system. The single steps of the development process have been described in further detail, according to Cacciabue (1998), as well as the problem boundaries and the simulation's aim. Afterwards, the conducted summary of (cognitive) task analyses and the field study of working context have been described. From task analyses, a theoretical model was presented, introducing action phases of the controllers work and giving a prototypical action sequence and a descriptive simulation. The selected algorithm for implementing a computational model is coloured petri nets. The implementation process and the current version of *MATriCS* have been introduced.

7. FURTHER WORK

It is obvious, that the development process of *MATriCS* will still last for some period of time, as there are several issues, the current version does not yet consider. At first to mention is the GND-position, which will be developed. Consequently, the airport model will be extended to represent ground traffic. Also it will be extended to represent the general structure of BER-airport. The new sub-model of GND will be within the ATCO-model. Furthermore, the current version does not model appropriately the information acquisition from far view. A new "FarView"-subpage will be implemented into the interaction model, which provides the information needed by the ATCO-model. With that said, there will be a more appropriate model of the controller acquiring information using the far view. Furthermore within both, the TWR-model as well as the still to be constructed GND-model, there will be a new subnet, modelling working memory more appropriate. Also, the planning algorithm already described in this paper, will actually be implemented. This requires at first the introduction of time into the net. Furthermore, it is intended to also model aircraft arriving early or late and to implement appropriate reactions of the controller to these deviations. If the requirements are fulfilled the model will be analysed in terms of performance analysis for the whole system as well as the individual controller.

ACKNOWLEDGEMENTS

Many thanks go to Henk Blom from National Aerospace Laboratory (NLR) in Amsterdam for his great help in getting the Cox task analysis which we long time thought of it got lost. Furthermore many thanks go to Dr.-Ing. Stefanie Huber for her hints in developing the theoretical model and for her psychological input to the construction of the CPN-Model.

REFERENCES

- Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C. and Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, 111(4), 1036–60.
- Bierwagen, T. (2005). Untersuchungen zur Veränderung der Arbeitsweise durch Automatisierung an einem Fluglotsenarbeitsplatz. *Beiträge zur Mensch-Maschine-Systemtechnik aus Forschung und Praxis*, 111–127.
- Buck, S., Biemans, M.C.H., Hilburn, B.G. and Van Woerkom, P.T.L.M. (1996). *Synthesis of functions (NLR Report TR-970545L)*, Amsterdam: NLR.
- CPN-Tools, The CPN-Tools Webpage. Available at: <http://cpn-tools.org/>.
- Cacciabue, P.C. (1998). *Modelling and Simulation of Human Behaviour in System Control*. Springer, London.
- Cacciabue, P.C. and Hollnagel, E. (1995). Simulation of Cognition: Applications. *Expertise and Technology: Cognition & Human-Computer Cooperation*, 43-54. Lawrence Erlbaum Associates Ltd, Hillsdale, New Jersey.
- Cox, M. (1994a). *Task Analysis of selected operating Positions within UK Air Traffic Control Volume 1: Main Report*. Royal Air Force Institute of Aviation Medicine, Farnborough, Hampshire.
- Cox, M. (1994b). *Task Analysis of selected operating Positions within UK Air Traffic Control Volume 2: Appendices*. Farnborough, Hampshire.
- DFS (2006). *Betriebsanweisung Flugverkehrskontrolle*. Deutsche Flugsicherung, Offenbach am Main.
- Desel, J. and Juhás, G. (2001). What is a Petri Net? - Informal Answers for the Informed Reader. *Lecture Notes in Computer Science 2128*, 1-25. Springer, Berlin.
- Dubouchet, E. and Mavoian, G. (1999). *PHARE Advanced Tools Departure Manager Final Report*, Brussels.
- Dubuisson, S. (2006). *Integrated Tower Working Position - Initial Study Report*, EUROCONTROL.
- Dörner, D. and Schaub, H. (1995). Handeln in Unbestimmtheit und Komplexität. *Organisationsentwicklung*, 14, 34–47.
- EATMP (2005). *Definition of A-SMGCS Implementation Levels*, EUROCONTROL.
- Frese, M. and Zapf, D. (1994). Action as the Core of Work Psychology - A German Approach. *Handbook of Industrial and Organizational Psychology*, 272 – 340. Consulting Psychologists Press, Palo Alto, CA.
- Hacker, W. (2006). *Allgemeine Arbeitspsychologie. Psychische Regulation von Arbeitstätigkeiten* 2nd ed. Huber, Bern.
- Hasevoets, N. and Conroy, P. (2010). *AMAN Status Review 2010*, Brussels.
- Hollnagel, E. (2009). *The ETTO Principle: Efficiency-Thoroughness Trade-Off – Why things that go right sometimes go wrong*. Ashgate, Farnham.
- Hollnagel, E., Woods, D.D. and Leveson, N.G. (2006). *Resilience Engineering - Concepts & Percepts*. Ashgate, Aldershot.
- Hopkin, D.V. (2010). *Air Traffic Control Automation. Handbook of Aviation Human Factors*. CRC Press, Boca Raton.
- Human-Factors-Consult (HFC) (2009). Bericht AP1 – Aufgabenanalyse.
- ICAO4444 (2007). *Air Traffic Management - procedures for air navigation services (fifteenth Edition No. Document 444 ATM/501)*. International Civil Aviation Organisation.
- Jensen, K. (1997). *Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use Volume 1: Basic Concepts*. Springer, Berlin, Heidelberg, 2nd ed.

- Jensen, K. and Kristensen, L.M. (2009). *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Springer, Berlin Heidelberg.
- Kovács, A., Németh, E. and Hangos, K.M. (2005). Modeling and Optimization of Runway Traffic Flow Using Coloured Petri Nets. *International Conference on Control and Automation (ICCA)*, 881–886. Budapest, Hungary.
- Miller, G.A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63, 81–97.
- Oberheid, H. and Söffker, D. (2008). Cooperative Arrival Management in Air Traffic Control - A Coloured Petri Net Model of Sequence Planning. *Petri Nets*, 348–367. Springer, Berlin, Heidelberg.
- Ruckdeschel, W., (1997). *Modellierung regelbasierten Pilotenverhaltens mit Petrinetzen*. VDI Verlag, Düsseldorf.
- Ruckdeschel, W. and Onken, R. (1994). Modelling of Pilot Behaviour Using Petri Nets. *Lecture Notes in Computer Science Vol. 815: Application and Theory of Petri Nets*, 436–453. Springer, Berlin, Heidelberg.
- Sheridan, T.B. (1992). *Telerobotics, Automation and Human Supervisory Control*. MIT Press, Cambridge.
- Smieszek, H., Huber, S. and Jürgensohn, T. (2011). Ein integratives Prozessmodell der Hauptaufgaben von Rollverkehrslotsen als Grundlage zur Modellierung und Simulation. *9. Berliner Werkstatt Mensch-Maschine-Systeme: Reflexionen und Visionen der Mensch-Maschine-Interaktion*. VDI Verlag, Berlin, Düsseldorf.
- Stroeve, S.H., Everdij, M.H.C. and Blom, H.A.P. (2011). Studying hazards for resilience modelling in ATM - Mathematical Approach towards Resilience Engineering in ATM (MAREA). *1st SESAR Innovation Days*, 1–8.
- Sun, R., (2006). The CLARION Cognitive Architecture: Extending Cognitive Modeling to Social Simulation. *Cognition and Multy-Agent Interaction*. Cambridge University Press, New York.
- Tavanti, M. and Bourgois, M. (2006a). *Control Tower Operations: A Literature Review of Task Analysis Studies*. EUROCONTROL, Roma, Italy.
- Tavanti, M. and Bourgois, M. (2006b). *Control Tower Operations: Roles Description*. Brétigny-sur-Orge Cedex, France.
- Vidosavljević, A. and Tošić, V. (2010). Modeling of turnaround process using petri nets. *Proceedings (CD) of the 14th ATRS World Conference*, 1–13. Porto, Portugal.
- Werther, B. (2006a). Colored Petri net based modeling of airport control processes. *International Conference on Computational Intelligence for Modeling Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*.
- Werther, B. (2006b). *Kognitive Modellierung mit Farbigen Petrinetzen zur Analyse menschlichen Verhaltens*, TU Braunschweig: Dissertation.
- Werther, B., Moehlenbrink, C. and Rudolph, M. (2007). Colored Petri Net Based Formal Airport Control Model for Simulation and Analysis of Airport Control Processes. *Digital Human Modeling*, 1027–1036. Springer, Berlin, Heidelberg.
- Wickens, C.D. (1984). Processing Ressources and Attention. *Varieties of Attention*. Academic Press, New York.
- Wickens, C.D. and Hollands, J.G. (1999). *Engineering Psychology and Human Performance* 3rd ed. Prentice-Hall, New Jersey.
- Zadeh, L.A. (1965). Fuzzy Sets. *Information and Control*, 8, 338–353.