Evaluating Operator's Cognitive Workload in Six-Dimensional Tracking and Control Task within an Integrated Cognitive Architecture

Yan Fu[1], Chunhui Wang[2], Shiqi Li[1], Wei Chen[1], Yu Tian[2], and Zhiqiang Tian[2]

[1] School of Mechanical Science & Engineering,
Huazhong University of Science & Technology, Wuhan, Hubei Province, 430074, China
[2] National Key Laboratory of Human Factors, China Astronaut Training & Research Center, Beijing, 100094 China
{Laura_fy,sqli}@mail.hust.edu.cn, chunhui89@yahoo.com.cn,
{cctian,tianzhiqi-ang2000}@163.com,
Mileschan@hust.edu.cn

Abstract. Six-dimensional tracking and control task within an Integrated Cognitive Architecture, as a makeup for automated Six-dimensional tracking and control task default. is a common yet highly complex space operation, challenging the human workload. For space exploration system safety, workload is a critical factor in task design and implementation. This research integrates two cognitive architectures: Queuing Network (QN) & Adaptive Control of Thought-Rational (ACT-R) to develop a rigorous computational model for Six-dimensional tracking and control task cognition process. ACT-R represents the human mind as a production rule system. Experiments are set up to build Six-dimensional tracking and control task cognition model and afterwards to validate feasibility of the proposed integrated cognition architecture. Ten subjects of similar training level are chosen to finish manual Six-dimensional tracking and control task with three task difficulty level: one only with displacement margin, one only with posture margin and one with displacement and posture margin. Cognition task analysis is firstly conducted on task performance of subjects. Cognition model of manual Six-dimensional tracking and control task is then built up based on the proposed integration architecture. The proposed integration model developed in the ACTR-QN describes component processes of tracking, decision making and controlling in a 3D environment by ACT-R production rules within QN network. Workload index for each cognition module is calculated based on sector utility throughout the whole task. Human results are compared with the modeled results in the dimension of task time and displacement/posture control trajectory deviation. Workload index is calculated based on the percentage of each module in the time dimension.

Keywords: Mental workload, Simulation, workload, six-dimensional tracking and control task, cognitive modeling.

1 Introduction

Six-dimensional tracking and control task is a very common space exploring task yet highly complex task that involves coordinated control in 3 dimension displacement

D. Harris (Ed.): EPCE 2014, LNAI 8532, pp. 470–479, 2014.

dimension and 3 dimension posture as well as with the execution of multiple critical subtasks. To explore how astronauts perform this complex task, researchers have developed some models to account for and simulate space driving behavior. Some of these models are primarily conceptual models that help one to understand the representational and procedural components of the driving task[1]. Others are computational models that compute, simulate, and predict various aspects of driving behavior [2-4]. These computational models have emerged as powerful tools for both theoretical study of space driving.

Flight control is the most similar to Six-dimensional tracking and control task. The research com-munity has recently witnessed a growing push for integrated performers models – models that unify the many aspects of flight into a single, larger scale computational model of behavior. Past and ongoing efforts toward integrated flight models, which have shown great promise, accounting for aspects of behavior during air traffic and even performance when flight while performing secondary tasks[5]. But the most popular research in performance modeling lies in road driving behavior. Road driving is 2 dimensional driving. In the case of control mechanism, it is comparable to space driving.

The "artifact" for driving is the vehicle itself and the interface between the human and the vehicle. Embodied cognition is the integrated cognitive, perceptual, and motor processes that manipulate the vehicle and execute the desired tasks[3]. Perception-and-action models of control provide a firm theoretical basis for how perception and action interact in basic tasks such as lateral and longitudinal control [6-8]. The approach to integrated driver modeling explored here centers on the development of driver models in the framework of a cognitive architecture. A cognitive architecture is a general framework for specifying computational behavioral models of human cognitive performance[9-12].The architecture embodies both the abilities and constraints of the human system – for instance, abilities such as memory storage and recall, learning, perception, and motor action; and constraints such as memory decay, foveal versus peripheral visual encoding, and limited motor performance. Anderson proposed ACT-R (Adaptive Control of Thought-Rational) cognitive architecture o model road driving. It is a hybrid architecture based on chunks of declarative knowledge and condition-action production rules that operate on these chunks. Aasman developed a driver model developed in Soar architecture. MHP was proposed to model the air navigation in NASA IMPRINT system[13].

However, Building useful models in ACT-R requires a considerable amount of training and practice. Since ACT-R uses a command-line interface to query the model's internal status, it lacks the visualization of information processing and interactions between its modules. A few efforts have been made to improve the usability of ACT-R as an engineering tool. Previous work, though important, has focused primarily on easier construction of the task knowledge and environment. The research work reported in this paper addresses the visualization issue by representing ACT-R as a Queuing Network (QN), one of whose advantages is the visualization of mental information processing. The QN cognitive architecture has been used to model human performance including reaction time, multitask performance, the psychological refractory period, transcript typing, driving with a secondary in-vehicle task, and driver workload measured with the NASA-task load index[14]. Such integration is another step towards unified theories of cognition advocated by Allen Newell[15]. We call the integrated architecture in this paper QN-ACTR.

2 QN-ACTR Integrated Cognition Architecture

At the conceptual level, the module network of ACT-R can be represented as a special case of QN. In a QN, information processing is the process of servers holding and processing entities. In ACT-R, modules process information, and buffers hold information. Therefore, modules and their buffers could be considered as servers in QN. Entities flow between these servers and carry the corresponding ACTR information, including buffer requests, chunks, production rules, and the notice of completion that triggers the next service (e.g., the next conflict resolution cycle). The server structure of QN-ACTR is illustrated in Figure 1.

ACT-R represents the human mind as a production rule system. It assumes two types of knowledge representations: declarative chunks and production rules (rules, for short). A chunk's retrieval time and error rate are determined by its activation level, which is jointly determined by the chunk's learning history and association with other chunks. Rules represent procedural knowledge in the form of condition-action (IF-THEN) pairs, and its action will be fired when its condition matches the current "mental state". A mental state consists of the state of each module, and each module is a cognitive component, such as the vision module and the declarative module. ACT-R "thinks" and "acts" by firing rules until a goal state is reached. Figure 1shows the server structure of QN-ACTR. All the servers are ACT-R modules and buffers, and all the paths between servers are information flows in ACT-R.

ACT-R assumes that human has a serial central processor (the production module in ACT-R) and handles multitask scenarios by fast switching between tasks. Each thread represents the task demands from a task. First, it assumes that the goal buffer can hold more than one goal simultaneously. Second, when multiple threads contend for the procedural resource, the least recently processed thread is al-lowed to proceed. Threaded cognition can be incorporated in QN-ACTR as a special case of QN with a specific type of queuing scheduling mechanism. QN-ACTR is implemented in Micro Saint Sharp (http://www.maad.com/), which is chosen because it is a network-based simulation platform and pro-vides natural support for QN modeling.

QN-ACTR was built in a C#-based discrete event simulation software package, Micro Saint Sharp version 2.2. At the implementation level, modules and buffers in ACT-R were programmed as servers (called task nodes in Micro Saint Sharp) as well as the corresponding data objects that store related parameters. Chunks and production rules were programmed as data objects. ACT-R methods and functions were ported to Micro Saint Sharp functions, which can be called by related servers. Global parameters were set to their default values as in ACT-R.

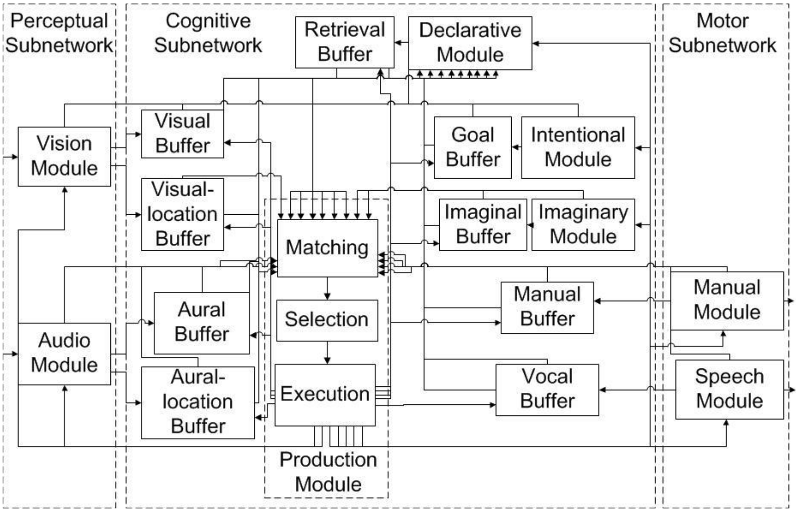

Fig. 1. The server structure of QN-ACTR

The task environment part of a model can be built with task templates supplied with the QN-ACTR. A task template in QN-ACTR is a general description for a type of experiment. A modeler can easily build a task environment by simply setting the template's parameters according to the experiment setup.

After defining the task environment using a template and defining the task-specific knowledge and parameters using the same ACT-R codes, a model is ready to run. In addition to the same text output traces of ACT-R, QN-ACTR can show how information flows in the mind, which is represented and simulated as a QN. For example, Figure 2 is a screenshot that illustrates the implementation of QN-ACTR in Micro Saint Sharp. The server network inside the dashed box represents the same mental structure as the one shown in Figure 1. The server network outside the box represents the task environment (i.e., displays and controls). Servers highlighted by dark borders are busy processing information. In the snapshot of Figure 2, the model is working on three things simultaneously: encoding a visual item, trying to match and select the next production rule, and creating a new chunk in the imaginary module.

QN-ACTR can also visualize the status and details of each module in a separated window. The capability in QN-ACTR can be extended this to audio displays, manual responses, and vocal responses using the "animator" of Micro Saint Sharp. Figure 3 shows a snapshot during the building-sticks task in the ACT-R tutorial.

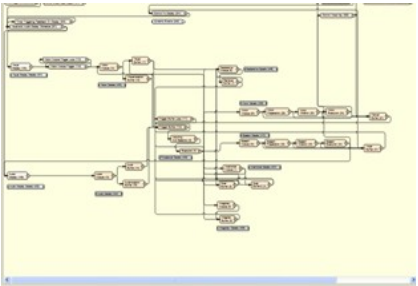

Fig. 2. Visualization of mental information ACTR

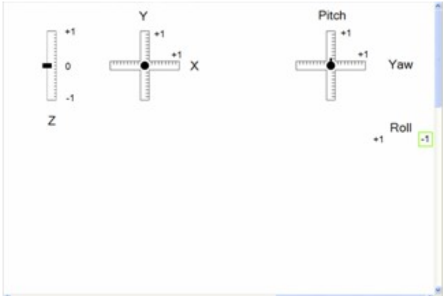

Fig. 3. Visualization of task displays and controls in processing in QN-ACTR

3 Building Six-Dimensional Tracking and Control Task Cognition Model within QN-ACTR Architecture

As mentioned, the ACT-R driver model has three primary components: monitoring, decision-making and controlling. The three components are integrated to run in QN-ACT-R's serial cognitive processor as a tight loop of small cognitive (and related) operations. The entire model is implemented as an ACT-R production system including relevant procedural and declarative knowledge. This section describes each component, the integration of the components into a working implementation, and finally estimation of model parameters and integration with the simulated driving environment.

Control. The control component of the space driver model manages all perception of lower level visual cues and manipulation of spaceship controls for placement control (forward, inward/outward, left/right) and posture control (pitch, yaw, roll).

For simplicity, the model utilizes a longitudinal control law to manipulate forward placement of the ship, very similar to the longitudinal speed acceleration control proposed by Salvucci (2007) for the, namely,

$$\Delta\psi_x = k_{x1}\Delta thw_x + k_{x2} thw_x \Delta t \tag{1}$$

The model encodes the position of the target and derives the time headway thwx to the target. Again, it computes differences from the last instantiation of control, deriving thwx along with the previously mentioned t. These two values then result in an updated value for acceleration.

The acceleration equation attempts to impose two constraints: asteady time headway (Δthwx= 0) and a time headway approximately equal to a desired time headway for following the target. Again, the two constants determine the weights of the two constraints. The acceleration value actually manipulates two controls: A positive value translates to depression of speed acceleration (throttle), and a negative value translates to depression of speed decrease, with values from 0 to 1 representing no depression to full depression, respectively.

For simplicity, the model utilizes a position difference control law to manipulate forward placement of the ship, very similar to the steering control proposed by Salvucci[16] for the, namely,

$$\Delta\varphi_y = k_{y1}\Delta y + k_{y2}\min(y, y_{\max})\Delta t \tag{2}$$

The model encodes the position of the target and derives the displacement difference Δy to the target. Again, it computes differences from the last instantiation of control, deriving the position difference along with the previously mentioned Δt. These two values then result in an updated value for position difference change. Again, the two constants determine the weights of the two constraints. The value actually manipulates two controls: A positive value translates to depression of difference decrease, and a negative value translates to depression of difference increase, with values from 0 to 1 representing no depression to full depression, respectively.

Posture control is different from speed control and focused on the posture change. For simplicity, the model utilizes a posture control law very similar based on the steering model proposed by Salvucci (2007) for car driving, namely,

$$\Delta\varphi_\theta = k_{\theta 1}\Delta\theta + k_{\theta 2}\min(\theta, \theta_{\max})\Delta t \tag{3}$$

For three direction posture control, the main purpose of the control is to decrease the posture difference between the ship and the target. The paper utilizes the same control law to manipulate the difference in the dimension of pitch, yaw and roll. The control law essentially attempts to impose two constraints: a steady posture degree difference ($\Delta\vartheta= 0$) and a posture degree equal to the maximum degree defined by the task. Again, the two constants determine the weights of the two constraints. The acceleration value actually manipulates two controls: A positive value translates to depression of degree difference increase (throttle), and a negative value translates to depression of degree difference decrease, with values from 0 to 1 representing no depression to full depression, respectively.

Perception. The perception component of the driver model handles the continual maintenance of situation awareness. For this model in the 3-D space environment, situation awareness centers critically on the displacement and posture difference of the spaceship to the target. Perception is currently based on a random-sampling model that checks, with some probability *p*monitor, one of six areas – namely, either forward/backward, inward/outward, left/right, pitch, yaw and roll – with the given decision rules. When the model decides to monitor a particular dimension, it moves visual attention to that dimension and determines whether there is any difference. If so, the model notes the vehicle's current critical dimension in ACT-R's declarative memory. Thus, declarative knowledge continually maintains the awareness of these dimensions. The model could, of course, be extended in a straightforward way to note other 5 dimensions.

Decision-making. The decision-making component of the driver model uses the information gathered during control and monitoring to determine whether any tactical decisions must be made. In the 3-D space environment, the most common decision-making opportunity arises in the determination of which dimension to adjust first and how much is required to adjust.

The decision of *which dimension* to change depends on the possibility of ship moving out of the vision field in certain dimension, given that drivers (in the United States) attempt to stay in the center of matrix originated by the target. If the ship is to move out of the vision field from horizontal dimension, the model checks current difference and time of moving outward. If the difference drops beyond a desired time value, the model decides to change the horizontal dimension

4 Model Validation

Just as no single method, measure, or metric will suffice for understanding human driver behavior, no single one will suffice to validate that the model indeed corresponds well to human driving. Nevertheless, one can validate the most critical parts of a driver model by focusing on key scenarios and analyzing the most important observable data involved in these scenarios. To this end, how the ACT-R model fits several aspects of driver data will now be examined in the scenario Six-dimensional tracking and control task. For this specific scenario, the examination focuses on 6 dimension control output: forward displacement, position difference change in other two dimensions as well as 3 degree changes in pitch, yaw and roll dimensions. The data are compared in the form of aggregate results and time-course profiles.

The computational nature of the QN-ACTR driving model, combined with its ability to interact with the same simulation environment that human drivers use, greatly facilitates the collection and comparison of human and model data. Human data from 10 universities students who are trained well in the simulated scenario. Model data were collected by running ten 10-min model simulations in the same conditions and same environment as the original experiment; note that the model, like a human driver, produces variability in behavior, and thus several simulation runs are desirable to achieve more stable results. The following analysis includes a total of 60 times (20 times at each of 3 difficulty level) of driving data for human participants and same number of times for the model simulations. Because the human and model simulation protocols are identical in form, each set is analyzed in the same manner so as to generate directly comparable measures of driver behavior and performance.

Workload data are sampled based on the statistical function of MicroSaintSharp (See Fig 5). ACTR-QN computed and visualized each sub-network utilization values, which are assumed to have linear relationship with corresponding workload components. Figure 4 shows the utilization of perceptual, cognitive, and motor sub-networks. The visualization clearly demonstrates workload increasing with faster presentation rates and provides more detailed estimation about each workload components.

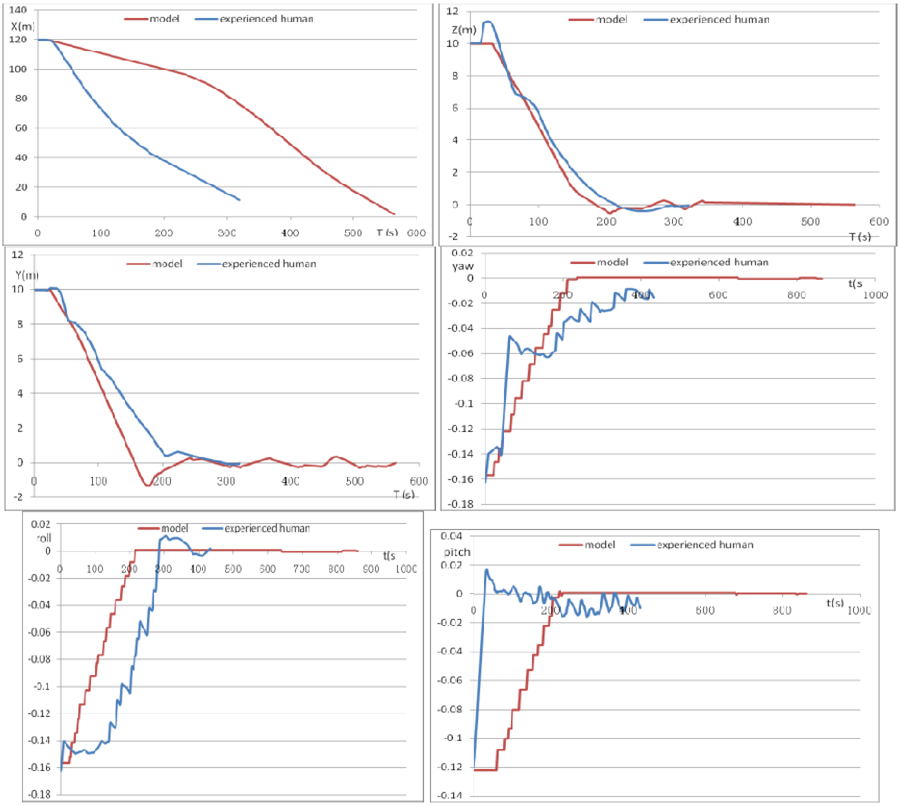

Fig. 4. Tracking Trajectory in 6 dimension (X: forward/backward; Y : left/right; Z: up/down;Pitch, Yaw and roll) for human and model data

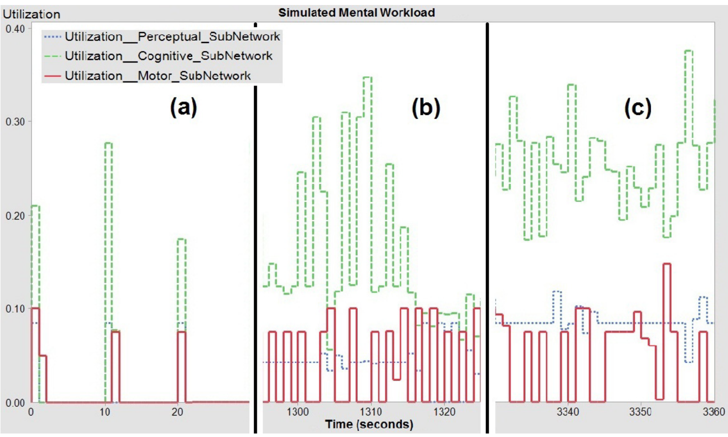

Fig. 5. Visualization of mental workload components in ACTRQNunder different task demands

5 Conclusion

Computational cognitive modeling is quickly maturing to address increasingly complex phenomena at an increasingly high level of rigor. More specifically, cognitive architectures have proven very successful at capturing both lower level performance and higher level decision making in complex dynamic tasks. The QN-ACTR space driving model represents a contribution toward this effort with a novel approach to integrating the lower level (i.e., operational) and higher level (i.e., tactical) aspects of driver behavior in the framework of the QN-ACTR cognitive architecture. Of course, the QN-ACTR Six-dimensional tracking and control task model does not yet provide a complete picture of space driving behavior – further work extending the task, artifact, and/or embodied cognition addressed by the model could take any number of directions. Nevertheless, we are confident that both model and architecture can evolve significantly from the current state of the art to capture a broader and deeper range of the phenomena surrounding driving behavior.

In 3-D space driving, verification results from Six-dimensional tracking and control task model showed that QN-ACTR can produce identical output traces to the human performance (MAPE < 5.0% and R2 > 0.9). The sources of the remaining variances include the difference of built-in random functions between Lisp and C#, which is used in randomly focusing visual attention on the next item, and the difference in rounding digits between Lisp and C#.

QN-ACTR is easy to use. Task-specific knowledge and parameters are defined using the same syntaxes as ACT-R. A task environment is defined by describing the experiment using a task template. The single-discrete-two-stage template is concise and powerful. More templates will be developed to cover other experimental paradigms. Compared with ACT-R, the visualization of the model in QN-ACTR is improved in the aspects of mental information processing and display and control interfaces. Another advantage of QN-ACTR is to define mental workload as network utilization and visualize it. There is currently no theory and measurement for mental workload in the ACT-R 6.0 released version, and the introduction of QN has the potential to improve this.

These mechanisms are what the QN architecture lacks. The QN architecture, on the other hand, represents the mental network with finer granularity. The processing in the QN mental network is more distributed than the processing in ACT-R that centralizes around the procedural module. The procedural module in ACT-R and threaded cognition is assumed to be serial. In contrast, the QN architecture does not have this assumption. Besides, QN does not need executive control to model multitasking performance. We expect that the full integration of ACT-R and QN could combine the advantages from each of them and better model multitasking performance.

In conclusion, QN-ACTR improves the usability of ACTR and the ACT-R implementation of threaded cognition as human factors engineering tools. Future research will examine the benefits of further integration between ACT-R and the QNcognitive architectures.

References

1. Boer, E.R., Hoedemaeker, M.: Modeling driver behavior with different degrees of automation: A hierarchical decision framework of interacting mental models. Paper presented at the 17th European Annual Conference on Human Decision Making and Manual Control held in Valenciennes, France, December 14-16 (1998)
2. Donges, E.: A two-level model of driver steering behavior. Human Factors 20, 691–707 (1978)
3. Groeger, J.A.: Understanding driving: Applying cognitive psychology to a complex everyday task. Psychology Press, Philadelphia (2000)
4. Hildreth, E.C., Beusmans, J.M.H., Boer, E.R., Royden, C.S.: From vision to action: Experiments and models of steering control during driving. Journal of Experimental Psychology: Human Perception and Performance 26, 1106–1132 (2000)
5. Tsimhoni, O., Liu, Y.: Modeling steering using the Queuing Network-Model Human Processor (QN-MHP). In: Proceedings of the Human Factors and Ergonomics Society 47th Annual Meeting, pp. 1875–1879. Human Factors and Ergonomics Society, Santa Monica (2003)
6. Fajen, B.R., Warren, W.H.: Behavioral dynamics of steering, obstacle avoidance, and route selection. Journal of Experimental Psychology: Human Perception and Performance 29, 343–362 (2003)
7. Salvucci, D.D.: Inferring driver intent: A case study in lane change detection. In: Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting, pp. 2228–2231. Human Factors and Ergonomics Society, Santa Monica (2004)
8. Wilkie, R.M., Wann, J.P.: Controlling steering and judging heading: Retinal flow, visual direction and extra-retinal information. Journal of Experimental Psychology: Human Perception and Performance 29, 363–378 (2003)
9. Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., Qin, Y.: An integrated theory of the mind. Psychological Review 111, 1036–1060 (2004)
10. Just, M.A., Carpenter, P.A.: A capacity theory of comprehension: Individual differences in working memory. Psychological Review 99, 122–149 (1992)
11. Levison, W.H., Cramer, N.L.: Description of the integrated driver model (Tech. Rep. No. FHWA-RD-94-092). McLean, VA: Federal Highway Administration (1995)
12. Liu, Y.: Queuing network modeling of elementary mental processes. Psychological Review 103, 116–136 (1996)
13. Aasman, J.: Modeling driver behavior in Soar. KPN Research, Leidschendam (1995)
14. Liu, Y.: Queueing network modeling of humanperformance of concurrent spatial and verbal tasks. IEEETransactions on Systems, Man, Cybernetics 27, 195–207 (1997)
15. Liu, Y., Feyen, R., Tsimhoni, O.: QueueingNetwork-Model Human Processor (QN-MHP): Acomputational architecture for multitask performance inhuman-machine systems. ACM Transactions onComputer-Human Interaction 13(1), 37–70 (2006)
16. Anderson, J.R., Lebiere, C.: The atomic components ofthought. Erlbaum, Mahwah (1998)