SEEV-VM: ACT-R Visual Module based on SEEV theory

Sebastian Wiese (sebastian.wiese@tu-berlin.de)
Department of Psychology and Ergonomics, Marchstr. 23
Technische Universität Berlin, D-10587 Berlin

Alexander Lotz (rene_alexander.lotz@daimler.com)
Daimler AG, Truck Product Engineering (TP/VES)
D-70546 Stuttgart

Nele Russwinkel (nele.russwinkel@tu-berlin.de)
Department of Psychology and Ergonomics, Marchstr. 23
Technische Universität Berlin, D-10587 Berlin

Abstract

In this publication an adaptation of the ACT-R visual module is presented based on the SEEV theory on attention allocation. By including this theory into the methodology of how the visual module works, a top-down control of attention guidance and bottom-up processing capabilities were implemented. The visual field of the model shifts according to current fixations, mimicking human behavior. Finally, we introduce a possibility of linking this new visual module with environmental sensors of a vehicle to generate data for the model without the need of a modeler generating environmental data. As of now the interpretation of the environment could be visualized differently depending on the understanding of the modeler. Now, the modeler benefits by having a time efficient reproducible source for data generation for driver modeling.

Keywords: ACT-R; Visual Module; SEEV theory; External Sensory Data; Driver Modeling

Introduction

Cognitive architectures, which are based on theoretical constructs with the objective to model real-world thoughts and interactions, offer a possibility to abstract human cognition. While these architectures deploy a method to test applied tasks, these tasks are also required to validate the implemented theory (Russwinkel, et al., 2018). At the same time, with abstraction, there is also a loss of information when forcing data into the required format of these cognitive architectures to create cognitive models. ACT-R (Anderson, et al., 2004), as an established cognitive architecture, offers the abovementioned complexity of separated modular modalities. Especially the visual module, as a main modality delivering information to most cognitive models and interpreting the environment, limits the applied tasks significantly. Currently, visual information is presented in a GUI (Graphical User Interface) and needs configuration. As the modeler dictates the position and characteristics of objects in the GUI, subjective interpretation of these characteristics can make the environment of a model inherently differently. This can affect the outcome of the model.

A task that has been addressed manifold in research is the modelling of the driving task and the ambition to model driver behavior (Salvucci, 2005; Salvucci & Taatgen, 2008). This task mainly consists of knowledge and experience that is applied with motoric outputs based on visual and auditory information. Therefore, multiple ACT-R modules are required to interact during modeling. Especially since (conditional) automated driving, Level 3, is prospected in near future, effects of attention and distraction (Haring, 2012), of driver drowsiness (Gunzelmann, et al., 2011), multi-tasking (Kosanke & Russwinkel, 2016) as well as insights on non-driving related tasks (Salvucci, 2009) have been modeled. Ultimately, combination of a human driver model in ACT-R with the possibility of a direct connection to a vehicle, to possibly adjust vehicle behavior according to the needs of the driver, is a promising vision. This proposal is similar to the ACT-Droid approach (Doerr, et al., 2016), in which a direct connection of a computational model in ACT-R to a system is configured. Also, the interpretation of simulation data through computer vision, ACT-CV (Halbrügge, 2013), has been presented as a means to develop visual information for ACT-R. While ACT - Droid connects to a self-contained system or interface, our new approach utilizes the vehicle as a means to monitor the real-world environment similarly to ACT-CV. Secondly, through the connection with the vehicle it is possible to model driver behavior and direct results to driver assistance systems, to increase system acceptance and possibly assist drivers in difficult situations.

Building rich environments with the default ACT-R device system is difficult. Standard ACT-R provides too few visual object types, making it near impossible to build real world scenes without defining a notation for object connotation like mapping colors to semantic meaning. This makes models hard to understand and extend. The difficulty of designing interfaces within the ACT-R toolchain can be bypassed by using external tools to generate the world around the agent or the interface, such as with abovementioned ACT-Droid. Thus far, the implementation of vehicle environments is tedious. Additionally, the environmental configuration underlies personal interpretation of the modeler and one task can be programmed in multiple ways, yielding the possibility of different calculated results. This drawback is addressed in the present concept by introducing a new adaptation of the

Figure 1: For the module to work, it needs some input data. In this case an annotated camera image. The field of view (red) does not span the entire image. The agent only perceives the color-coded parts, i.e. road surface is invisible to it. The first step is to setup a world simulation and stream data into the ACT-R runtime. The cognitive model manages AoRs (orange). Together they move attention to a car in front (white).

ACT- R visual module. Additionally, a framework for the connection of ACT-R to vehicle data is presented, in which the data of external sensors are interpreted and scaled to allow to feed the visual module with information, evading the necessity to model the environment. A detailed description of this process is presented in this publication.

A new vision module – SEEV-VM

Perceiving real world scenes is a hard task for a cognitive model. It requires the model to comprehend the scene, extract meaning and make assumptions about location and type of information. There is a lot of uncertainty involved, where to precisely find requested information or whether it is present at all. Henderson (2003) identified three different kinds of knowledge that are involved in a gaze guiding mechanism. That are: episodic scene knowledge, remembering where objects were seen lastly or, on a long term, where to expect task-relevant information, but about a specific scene. Scene-schema knowledge provides generalized semantic and context information, e.g. we know how car interfaces look like and can easily orientate oneself in a yet unknown car cockpit. The third is task-related knowledge. This type of knowledge includes learned fixation sequences, e.g. monitoring traffic before and while changing lanes with a car. They have in common that they encode a location with a meaning. This idea, also present in the works of (Oliva, et al., 2003), constitutes the foundation of our proposed visual module: SEEV-VM.

The SEEV approach (Wickens, 2015) can predict a scan path in rich visual environments like airplane cockpits. The visual workspace consists of displays, also called areas of interest that attract attention and contain task relevant information. Every display is defined by four numeric factors: salience, effort, expectancy and value. The SEEV algorithm decides which display will be attended by summing up factors for all displays and comparing the results. This approach combines bottom-up and top-down factors. Wickens (2015) describes **salience** as the physical properties of a display that increase its attraction for the human eye, e.g. high contrasts or bright colors. **Effort** correlates to the distance between the target display and the current point of fixation. **Expectancy** and **value** form the top-down factors: value describes the relevance of information in a display and expectancy the frequency with which information updates. I.e. a high frequency and a high value display will be attended more often, because its information is important and changes frequently, therefore needs to be sampled often.

The proposed visual module (SEEV-VM) is based on the ideas of the SEEV theory and existing vision modules like EMMA (Salvucci, 2000) and PAAV (Nyamsuren & Taatgen, 2013). EMMA extends ACT-R with realistic eye movements by integrating physiological constraints of the human eye. PAAV extends the attention guidance mechanism itself. It integrates bottom-up factors into the existing top-down control of attention.

The SEEV theory provides not only an algorithm for guiding visual attention, but also a representation for top-down control of the attention guidance mechanism. Information is expected to be found in certain places in the environment. In the SEEV-VM these locations are called areas of relevance (AoR) to differentiate from AoI in eye tracking experimental set ups. Both PAAV and SEEV integrate top-down and bottom-up processing into their algorithms, both use numeric values and calculate an attraction value (SEEV) or an activation value (PAAV). It is a reoccurring idea to fuse all factors into a single parameter to base an attention selection decision on. The SEEV-VM module uses a very similar approach and calculates a guidance value for each visual object and AoRs.

The algorithm selects an object to attend based on the guidance values, see Figure 1, then shifts attention towards this object and starts encoding. After encoding, the algorithm immediately repeats the process, searching for an object to fixate. This mechanism runs in an endless loop without directing instructions through the buffer interface. It is assumed that the human eyes always look at something and provide information about the visually perceivable environment. Only when the production system accesses this information, which is using the vision module buffer content, attention is directed at the given object.

Arbitrary visual objects

SEEV-VM supports two different modes of operation: In the **traditional mode of operation**, the SEEV-VM module manages visual objects with function calls. Functions can create, modify (also their semantic meaning, e.g. a traffic

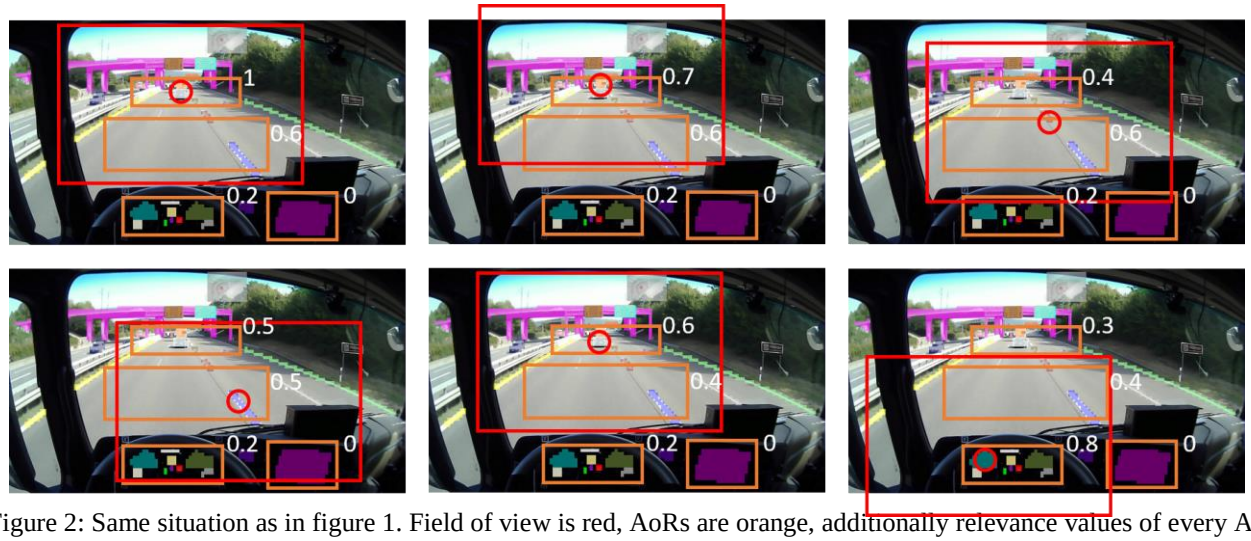

Figure 2: Same situation as in figure 1. Field of view is red, AoRs are orange, additionally relevance values of every AoR are shown next to the AoR. Numbers show the relevance before attention is shifted; red dot shows the result of the attention shift. In the example, attention is directed at the white car in front, reducing relevance value of its AoR to 0.7. After the white car was encoded, the attention guidance mechanism starts again. Inhibition of return prevents the module from looking at the same object again, hence the orange car is attended. On the last part, a production fired that increases relevance of the instruments AoR. Based on salience and effort influences, attention is directed at the speed indicator instead of other instruments.

light can turn green) and remove objects (removing an object makes it invisible to the vision module). The attention guidance algorithm works the same in both modes. The difference is, that in this mode the algorithm iterates over all objects, checking whether an object is inside the current field of view and calculating attraction values. This mode does not support occlusion of objects, if an object is added, it should be visible. For very complex 3D environments, another mode was implemented. In the **pixel-based mode** the module receives a semantically annotated map of the world that matches the current field of view of the agent. The external data source produces images (like in a video game), but instead of pixels with color values, every pixel contains a numeric object identifier. Sensors, such as vehicular external sensors, with internal object identification algorithms can provide this information.

As of the time of implementation, the ACT-R architecture did not provide a standardized and easy way to integrate external data into the simulation runtime. Hence, the SEEV-VM module provides its own communication protocol. The world simulation is linked bidirectional with the vision module, as the SEEV-VM module interacts directly with the world by moving the field of view.

The SEEV-VM module allows a modeler to define visual objects that are not bound to a limited number of categories such as geometric forms, text or buttons. A visual object can be everything, ranging from a smartphone display or other complex objects to its content like icons or lines. The result of the encoding process is a chunk that is placed into the vision modules buffer. This chunk holds characteristics that the modeler can define and physical properties of the object (i.e. location and dimension). The chunk can contain information like distance to the object or other information that is expected to be processed or calculated by the vision system. Figure 1 displays an exemplary scenario of a head-mounted camera with an automatic object recognition that can be passed to the SEEV-VM module. The module can process every color-coded object. The red dot is the center of fixation, the red box the field of view. Orange boxes show areas of relevance. In the example, the vision system manages AoRs to monitor the traffic in front of the car, the instruments and a display for a non-driving related task.

The encoding time is the amount of time the vision system fixates an object until it can place its chunk into the buffer. A modeler can also choose to set this value. This allows the module to adapt to the scope of the simulation. The environment can be made up of several displays that take longer to encode but also provide more information; similar to SEEV approach it takes one long fixation to sample a display. Or, in a more detailed simulation, the content of each display is modelled, these items take less time to encode but only provide their information (their semantic chunks). That means to sample all information of a display, every object of this display needs to be attended.

In order to enable the module to function properly, the modeler must define salience values of objects. Unlike the PAAV module salience is not calculated by the module, because not every object has features like color or shape. It's optional to specify these features. The vision system can be instructed to look for certain features, but there is no guarantee that only objects that match are attended. This works very similar to the PAAV module: feature selection is one factor of many that form the guidance value.

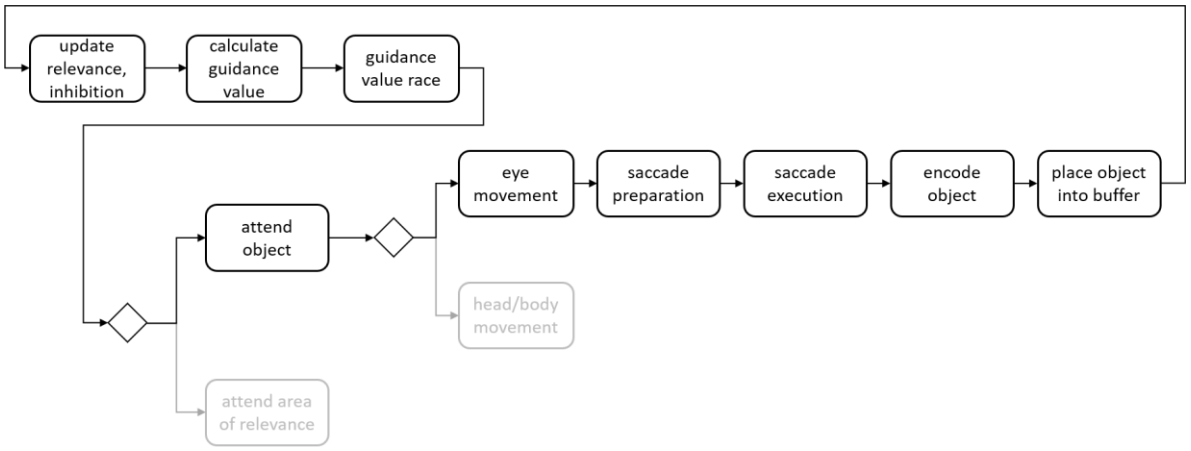

Figure 3: Process diagram of the attention guidance algorithm. The algorithm starts with a recalculation of relevance and inhibition. Inhibition decays over time and relevance increases over time. In the example, an object won the guidance value race, that is a comparison of all guidance values, and an eye movement is necessary to reach its location.

Using SEEV-VM

The SEEV-VM requires an external world simulation, the device interface is no longer used for vision. This world simulation can use many ways to produce semantically annotated maps or lists of objects, that are then communicated with the ACT-R environment and translated to function calls (adding, modifying or removing objects). E.g., the world simulation can use computer vision algorithms to annotate camera images or be a virtual world entirely, similar to ACT-CV (Halbrügge, 2013).

The communication protocol guarantees that both cognitive model and world simulation are synchronized. Hence, besides object extraction, the simulation must allow to stop and advance simulation time.

On the cognitive model side, the visual organization buffer allows to instruct the visual system by providing chunks to create, modify or remove areas of relevance and to set a feature search vector. AoRs have a location and a dimension, they form a rectangular space (orange AoRs in Figure 2), have a relevance value and two additional values that correspond closely to the expectancy value of the SEEV theory. It is possible that an AoR encompasses multiple objects. To sample all information inside this AoR all objects need to be attended. Unlike the original SEEV approach, relevance (in the SEEV approach called value) here changes over time: once an object is attended, relevance of its AoR is reduced (or consumed) for a certain amount of time (based on its refresh rate). The consumption value relates to the number of expected objects, the refresh value to the frequency with which changes are expected. These values are optional. By setting these values to zero the module will not update relevance values of AoRs, but relevance values can be updated via the production system. The SEEV approach is an abstraction of the whole cognitive process, in ACT-R this process is subdivided into smaller, parallel executable processes. Therefore, it is possible to update relevance of an AoR once all information is sampled. This is done by defining productions that count the number of objects in a given AoR. After a certain number of objects attended, another production reduces this AoRs relevance via the visual organization buffer. Later a production fires that increases relevance value again. This approach is more akin to the ACT-R way of modelling cognition. And requires a very detailed modelling of involved processes.

The attention guiding functionality works in three steps: (1) a guidance value is calculated for every visual object and AoR by adding up salience, relevance, feature weights, inhibition of return and effort. (2) A guidance value race determines the object with the highest guidance value. This allows the agent to look at areas that are not currently in the field of view, e.g. to look at the passenger's door mirror (see Figure 2). (3) An attention shift is then initiated, it follows the EMMA model in three stages: (1) preparation of a saccade, (2) execution of the saccade and (3) encoding of the object. Figure 3 shows the workflow of the module.

Attending an AoR forms a special case, which allows the agent to look at an area that is not currently visible. Because there is no object to encode, the module immediately starts to search for objects to attend. The algorithm can initiate a head movement, as the default motor module of the ACT-R system cannot move the agents head, the vision module simulates head movements. A shift of the field of view (red bounding box in Figure 2) simulates this movement. In some cases, a movement of the whole body is needed to look at certain locations; in these cases, the module assumes that it can control the body entirely. This allows the model to visually perceive rich 3D environments regardless of these missing functionalities. SEEV-VM uses parameters to control when to make a head or body movement and how fast these movements are executed.

Vehicular data generation

As described previously, complex 3D environments are difficult to model and the proposed second mode SEEV-VM can receive semantically annotated maps. Modern vehicles

are equipped with multiple internal and external sensors to allow advanced driver assistance systems to function. This information is available within a vehicle on CAN-Bus (ISO 11898-1:2015), Ethernet or FlexRay (ISO 17458-1:2013, 2013) networks for microcontroller communication and holds semantic information about surrounding objects. Depending on the sensor and data definition, multiple value signals are calculated and retrievable within these networks. Sensor types that can observe and classify objects in the proximity of the vehicle are radar, lidars and cameras. These sensors function as the 'eyes' of modern vehicles to provide environmental data for assistance systems (e.g. Adaptive Cruise Control, Lane Keep Assist and Emergency Brake Assistance).

Typically, these sensors are capable of identifying several vehicles, objects or pedestrians during driving, similarly to the way ACT-R models these objects in its environment. Apart from the classification, precise speed, distance and trajectories are calculated as properties of the objects. This information is communicated within the networks and updated rapidly (approximately 0.01-0.1 seconds). In order to model driver behavior, it would be ideal to make this data available to ACT-R. This has three major benefits: (1) modelling of environmental objects would be automated and several different scenarios could be analyzed through the proposed second mode of SEEV-VM operation. (2) Obtained data would only vary depending on sensor setups and are reproducible (attributes are not defined by modelers). (3) A framework could work with offline data after drives or online with a model predicting driver behavior.

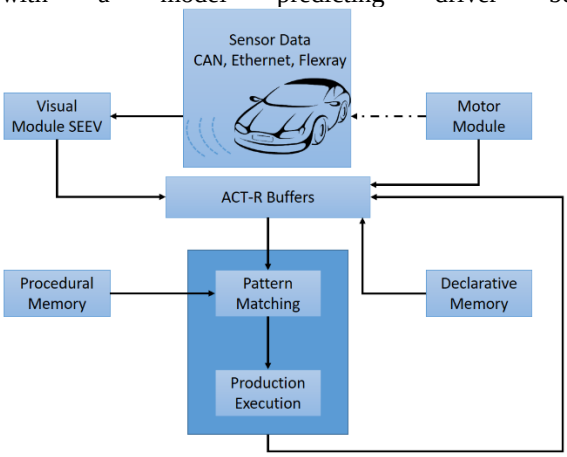

Figure 4: Adapted ACT-R framework with direct link to vehicle and modified visual module.

The adaptations of the proposed data generation enable interpretation of environmental data through sensors of a vehicle, airplane or robot, as presented in Figure 4. In this form, the bus data is interpreted and is available as CSV-files (Comma Separated Values). These CSV-files include the relevant data in lists with timestamps and the values of sensor data (e.g. distance to object, object type, speed of object). Data is interpreted through a parser to translate the data into the three function types included in the SEEV-VM (i.e. add-visual-object, modify-visual-object and remove-visual-object). The adapted ACT-R framework does not connect the motor module to the environment and motoric actions do not alter the environment. This is because the ACT-R motoric module would need an upgrade to allow for extensive motoric control needed during driving. However, if an online interface were implemented the possibility of connecting ACT-R to a prototypic vehicle would be possibly. Nevertheless, the integration of external sensory data does allow simpler modeling of real-world environments with reproducible interpretation of data according the SEEV theory.

Discussion

SEEV-VM is far from finished, there are still some open issues. The module does not support some features found in the default ACT-R vision module. E.g. it has no explicit attended field but introduces inhibition of return to reduce changes to fixate the same object repeatedly. SEEV-VM aims to offer a less precise way to instruct visual attention, giving the model more flexibility to react to variations and dynamics in known situations. As an example, the model does not know precisely where a traffic light is located, but it knows where to expect one. Combined with bottom-up processing capabilities the vision system will find a certain traffic light. The productions can work with a special chunk for the object type of a traffic light, reducing the burden to share productions with different models. It is possible to establish a library of object chunk definitions.

Arbitrary vision chunks not only increase maintainability of a model, but also allow vastly different simulation environments and affordances to connect to an ACT-R agent. This flexibility might also have a downside, as it does not restrict modelers to plausible models: A vision object chunk can contain unrealistically complex information.

In a future version, we plan to standardize the communication protocol to provide an easy to use API to establish a connection between SEEV-VM and simulation environments. It can be envisioned that many different modules (motor, audio) connect to the same simulation server, that delegate commands and information between agent and world. JNI (Hope, Schoelles & Gray, 2014) already provides this functionality and could be modified to support SEEV-VM.

The module has not yet been validated. The SEEV model works well (Wickens, McCarley & Steelman-Allen, 2009), but it's less detailed than its SEEV-VM adaption. In SEEV-VM attention is directed at objects and not at displays that could span entire scenes (e.g. rear window of a car). The module is able to work in the same way, but in regard to ACT-R, cognition is modelled on a finer resolution, requiring chunks of information at a certain point in time. The next step will be to conduct a validation study to evaluate SEEV-VMs approach to modelling visual attention.

Predicting a scan path is essential in determining whether unexpected visual stimuli were recognized or not. In the module, it is very easy to guide attention towards a visual location (not necessary towards an object) by setting

relevance of an AoR. However, finding plausible relevance values is not trivial. Relevance and expectancy (consumption and refresh values) can be seen as results of a learning process, allowing to model experts and novices. In a future work, SEEV-VM has to be validated and we expect to change some parts of the implementation like the default set of parameters. While the SEEV-VM benefits from large flexibility, the subsymbolic parameters need to restrain it in such a way that realistic behavior is generated.

Conclusions

The SEEV-VM module adaptation offers unique development by incorporating the SEEV theory as a foundation for visual attention in ACT-R. While the modeler holds the task of attributing the salience of objects in the environment, SEEV-VM enables ACT-R modes to perceive semantically annotated real-world scenes. By integrating top-down and bottom-up processing it allows the model to react to unexpected events. Setting up AoRs is an easy and abstract way to instruct the visual system, thereby allowing the model to see unexpected things or process objects that are not explicitly represented by productions.

The current substantial effort necessary of modeling visual information in ACT-R needs to be improved to increase the applicability of cognitive modeling to real-world usability testing and to integrate it into applications. Especially tasks and environments that require a lot of visual information are thus far difficult to analyze with ACT-R. This includes the automotive sector in which rich environments can influence drivers in a plethora of facets. The SEEV-VM module adaptation provides the possibility of connecting vehicular BUS-communication to ACT-R and therein deliver semantic data from the surrounding. Multiple and quickly changing scenes are far easier to incorporate into cognitive models, offering the possibility of modeling human-machine-interaction in the vehicular context.

References

Anderson, J. R. et al., 2004. *An Integrated Theory of the Mind*. Psychological Review, 111(4), pp. 1036-1060.

Doerr, L., Russwinkel, N. & Prezenski, S., 2016. *ACT-Droid: ACT-R interacting with Android applications*. In: Proceedings of the 14th International Conference on Cognitive Modeling. University Park, PA: Penn State: s.n.

Gunzelmann, G., Moore, L. R., Salvucci, D. & Gluck, K. A., 2011. *Sleep loss and driver performance: Quantitative predictions with zero free parameters*. Cognitive Systems Research, 12(2), pp. 154-163.

Halbrügge, M., 2013. *ACT-CV: Bridging the Gap between Cognitive Models and the Outer World*. In: E. Brandenburg, et al. eds. Grundlagen und Anwendung der Mensch-Maschine-Interaktion - 10. Berliner Werkstatt Mensch-Maschine-Systeme. Berlin: Universitätsverlag der TU Berlin, pp. 205-210.

Haring, K. S., 2012. *A Cognitive Model of Drivers Attention*. In: Russwinkel, Drewitz & van Rijn, eds. Proceedings of the 11th International Conference on Cognitive Modeling, Berlin. Berlin, Germany: Universitaetsverlag der TU Berlin.

Henderson, J. M., 2003. *Human gaze control during real-world scene perception*. Trends in cognitive sciences, 7(11), pp. 498-504.

Hope, R. M., Schoelles, M. J., & Gray, W. D., 2014. *Simplifying the interaction between cognitive models and task environments with the JSON Network Interface*. Behavior research methods, 46(4), pp. 1007-1012.

ISO 11898-1:2015, 2015. *Road vehicles - Controller area networt (CAN) - Part 1: Data link layer and physical signaling*.

ISO 17458-1:2013, 2013. *Road vehicles - FlexRay communications system*.

Kosanke, H. & Russwinkel, N., 2016. *Doing all at once? Modeling driver workload in an abstract multitasking scenario*. Abstractband der 58. Tagung experimentell arbeitender Psychologen (TeaP).

Nyamsuren, E. & Taatgen, N. A., 2013. *Pre-attentive and attentive vision module*. Cognitive Systems Research, Issue 24, pp. 62-71.

Oliva, A., Torralba, A., Castelhano, M. S. & Henderson, J. M., 2003. *Top-down control of visual attention in object detection.* Proceedings 2003 International Conference on Image Processing, pp. 253-256.

Russwinkel, N., Prezenski, S., Dörr, L. & Tamborello, F., 2018. *ACT-Droid Meets ACT-Touch: Modelling Differences in Swiping Behavior with Real Apps.* Proceedings of the 16th International Conference on Cognitive Modeling (ICCM 2018), 21-24 07, pp. 120-125.

Salvucci, D. D., 2000. *A model of eye movements and visual attention*. Proceedings of the International Conference on Cognitive Modeling, pp. 252-259.

Salvucci, D. D., 2005. *Modeling tools for predicting driver distraction*. Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting.

Salvucci, D. D., 2009. *Rapid prototyping and evaluation of in-vehicle interfaces.* ACM Transactions on Human-Computer Interaction, 16(2).

Salvucci, D. D. & Taatgen, N. A., 2008. *Threaded cognition: An integrated theory of concureent multitasking*. Psychological Review, 115(1), pp. 101-130.

Wickens, C. D., 2015. *Noticing events in the visual workplace: The SEEV and NSEEV models.* In: R. R. Hoffman, et al. eds. Part VI - Perception and Domains of Work and Professional Practice. Cambridge: Cambridge University Press, pp. 749-768.

Wickens, C., McCarley, J., & Steelman-Allen, K., 2009. *NT-SEEV: A model of attention capture and noticing on the flight deck*. In Proceedings of the human factors and ergonomics society annual meeting. Sage CA: Los Angeles, CA: Sage Publications. Vol. 53, No. 12, pp. 769-773.