The use of ACT-R to develop an attention model for simple driving tasks

Kerstin Sophie Haring (ksharing@fennel.rcast.u-tokyo.ac.jp)
Katsumi Watanabe (kw@fennel.rcast.u-tokyo.ac.jp)
Research Center for Advanced Science and Technology, The University of Tokyo
4-6-1, Komaba, Meguro-ku, Tokyo, 153-8904, Japan

Marco Ragni (ragni@cognition.uni-freiburg.de)
Lars Konieczny (lars@cognition.uni-freiburg.de)
Center for Cognitive Science, University of Freiburg
Friedrichstr. 50, 79098 Freiburg, Germany

Abstract

Driving a car is obviously a complex task and the construction of an ACT-R model of human attention while performing this task is similarly complex along multiple dimensions and presents a challenge to architecture and modeler. This work is a first attempt to develop an integrated driver model of attention in the cognitive architecture ACT-R. The model is able to keep a traffic lane, identifies traffic signs and crossroads in a sparse, simulated environment.

Keywords: Driver behavior model; cognitive architecture; ACT-R; Attention

Introduction

For most of us, driving a car is one of our everyday tasks. But even for experienced drivers, just the task itself it is a cognitive challenging task involving a big range of human senses like sight, hearing, touch and acceleration. And this is not yet considering secondary tasks like talking on the phone or visual distractions like city illuminations. Luckily, most driving task are not as challenging as the Traffic Light Tree in Fig. 1, an artificial scenario by the French sculptor Pierre Vivant.

Fig. 1: The (thankfully not on a crossroad) installed traffic light sculpture by Pierre Vivant.

Current attempts to model human attention while driving a car are realized in a quite more simple environment, yet they are quite an important first step towards the modeling of these highly complex tasks. Vice versa, it also can provide an indication for the future development of a cognitive architecture by showing what cannot be realized yet.

Fig. 2: Screenshot of the environment interaction with ACT-R. The red circle indicates the current visual focus of attention of the model.

The simulation environment for this model was restricted to the components the cognitive architecture can recognize. Nevertheless, basic driving scenarios simulating human visual attention and driver behavior could be implemented.

The screenshot form the driving environment, which was separately implemented in Lisp for this work, shows from top-down another (blue) vehicle, the focus of attention (red circle) and the navigation point (N) to keep the vehicle in the center of the road. This model focuses on basic reference points like the horizon, a leading car, the border and the center line of the road, crossroads and traffic signs. For example, the model of a driver in the screenshot in Fig. 2 sets the focus of visual attention on the outer border of the road which enables it to reevaluate the center for the N point. In the next step, it will shift the focus of attention to the front and (hopefully) detect the car in front. If so, possible next steps could be the comparison of the distance to a (here fixed) safety distance or an overtaking procedure.

The here presented cognitive model should simulates through ACT-R human attention while driving in a simplified environment and produces the behavior for scenarios with other cars, crossroads and traffic signs.

The cognitive architecture

The ACT-R (Anderson, 1993; Anderson 2007) cognitive architecture proposes artificial, computational processes that aim to act like a human cognitive system. Most of its basic

assumptions are inspired by the progress of cognitive neuroscience. The tasks that humans can perform should, in theory, consist of a series of discrete operations. ACT-R is primarily used to model experimental psychological data.

ACT-R compromises theories about the operation mode of human information processing and describes a comprehensive computer model of human cognition. The architecture is not only a proposed unified theory of cognition, it is also a programming environment, a production system with a development environment where it is for example possible to set parameters or run simulations. ACT-R is a framework in which the researcher can create models (programs) for different tasks. Running this model produces a simulation of human behavior.

As many cognitive architectures, ACT-R contains a number of modules which can be accessed through their limited-size buffers. For each module, a dedicated buffer serves as an interface with that module. The state of ACT-R at a given time is the content of the buffers at that time. Buffers are connected to the modules and are changed by production rules. Every buffer and (nearly) every module can be allocated to a cortex region. This enables an interesting mapping between buffers and neural processes (Anderson 2007).
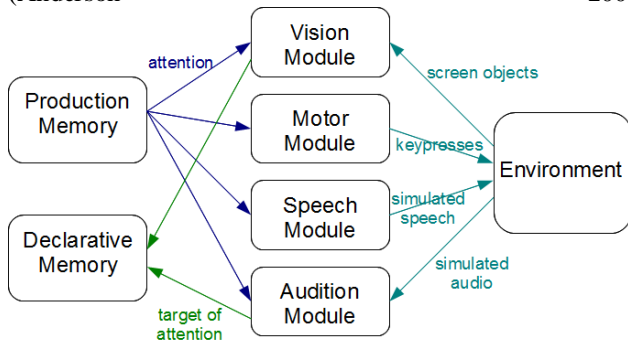

Fig. 3: ACT-R system diagram (Byrne, 2001). The Cognitive Layer and each of the Perceptual-Motor modules run in parallel, but each component is itself serial.

The main assumption of ACT-R is the representation of knowledge as either declarative or procedural knowledge. Declarative knowledge, consisting of facts, is represented in form of *chunks,* or small logical units which encode simple facts (e.g. the fact: "Sapporo is in Japan"). Procedural knowledge, representing knowledge about how we do things, is represented in form of *production rules,* condition-action rules that generate a specific action (e.g. manipulate declarative knowledge) if the conditions of this rule are fulfilled. In other words, ACT-R's knowledge representation is split in two kind of memory modules, perceptual-motor modules and memory modules.

The diagram in Fig. 3 shows the ACT-R in action. For the visual attention, the environment provides screen objects to the vision module. The target of attention is put into the declarative memory in form of a chunk.

We decided to choose ACT-R for this task because it has a visual search, is a well-accepted cognitive architecture, and was already used in the past to evaluate the attention during a driving task. A crucial advantage of the ACT-R architecture is that the three main components used in this model (control, monitoring and decision-making) can be implemented directly. This takes into account human constraints and results in a cognitive adequate model of human attention.

Previous work

Most developed approaches can be distinguished into two categories: task specific and generic approaches. *Task specific approaches* such as Cosmodrive (Bellet et al., 2007) and Pelops (Benmimoun, 2004) reproduce the cognitive functions of a car driver. In contrast to task specific approaches, *generic approaches* can model various aspects of human behavior. Therefore, it is necessary for these architectures to include a theory of human information processing. Examples for such architectures in which driver models have been implemented are ACT-R (Anderson, 1993; Salvucci, 2006), SOAR (Aasman,1995) and QN-MHP (Liu et al., 2006).

Driver models were described by Aasman (1995) in the cognitive architecture SOAR and by Liu (1996) in Queuing Network-Model Human Processor (QN-MHP). Although these models already exist in other cognitive architectures and the central ideas remain the same in any architecture, the ACT-R model of a driver shows a broader spectrum of application (Salvucci 2001; 2006).

Salvucci (2006) developed a first integrated cognitive model of human driving behavior in ACT-R. He showed in his work the generality and the applicability using the cognitive architecture ACT-R for the specific task of driving. His model is designed to keep a standard vehicle on a multi-lane highway with moderate traffic. The model is also able to recognize the distance to a vehicle ahead and to make the decision for overtaking. As driving is a highly complex task and not readily implementable, this model has some limitations. The model solely was meant to interact with a highway environment without recognition of traffic signs, crossings or slip roads. An implementation limitation was the use of the previous version ACT-R 5.0 and its incompatibility to newer versions. It was also not possible to make the ACT-R model interact directly with a driving simulator.

Regardless the challenges, Salvucci proposed to develop a driver model in the context of embodied cognition, task and artifact (ETA) framework (Byrne, 2001), an idea which was adopted in this work.

Cognitive model

A driver model can be a powerful instrument with several possible fields of application, such as the development of intelligent driver assistant systems. Our model is implemented the newest version ACT-R 6 (Anderson, 2007) and using the standard ACT-R development environment

running on an open source LISP, which not only guarantees support and accountability, but also enables the research community to use the developed model for further research.

Driver Modeling

We introduce a computational model of human attention in a car driving task implemented in the ACT-R architecture. As described previously, this model aims to account for the embodied cognition, task and artifact (ETA) framework.

The complex task of driving a car can be divided into basic subtasks. These must be integrated and interleaved to achieve the continuously changing parent task. Michon (1985) identified three levels of skills and control for the driving task: operational (control), tactical (maneuvering), and strategical (planning). He claims that a comprehensive model should take into account the various levels.
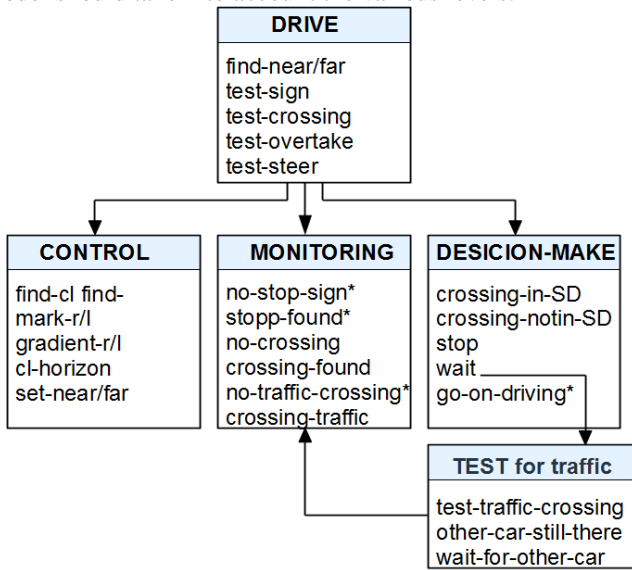

Fig. 4: Schematic representation of the production rules of the driver model in a simple crossroad scenario. The title of a box indicates the current goal and the corresponding production rules. The arrows show the flow of control and the asteriks the return to the parent-goal.

The independent subtasks of the (simplified) parent task *drive* (see Fig. 4) were implemented as *control*, the operational process controlling the input, *monitoring*, the tactical process interacting with the environment, and *decision making*, also analogous to the tactical level of Michon (1985), managing maneuvers like overtaking. These subtasks are processed serially. Every production of the top level goal *drive* has sub-goals, which incorporate the three components.

Development Environment The theory of ACT-R is embedded in the ACT-R software in form of Common Lisp functions. This model is implemented in the open source Clozure CommonLisp 1.3 and the current version of ACT-R 6.0 under the operating system Ubuntu 9.04. In order to

make the simulation environment interact with the ACT-R system, it was directly implemented in LISP with simple graphics and the extension with the LTK Lisp Toolkit. As it was not possible to make ACT-R directly interact with a driving simulator, we decided to use a Lisp-implementation of a driving environment.

Model Specification

As mentioned earlier, human attention during a driving task is quite complex along multiple dimensions. It is not yet possible to model every aspect of human attention within a cognitive architecture for such a complex task. To limit the scope of the project, model is hold quite simple. The model focusses on simple visual perception and attention shifts how they might occur in a sparse, artificial environment.

The first issue to address was to implement the three components control, monitoring and decision making as a loop of cognitive operations in the serial processor of the ACT-R architecture. The UML diagram in Fig. 5 shows the behavior of the cognitive model. This diagram identifies one primary loop, which corresponds largely to the control component in Fig. 4. The primary loop implements the identification of the near and the far point, in other words, the points responsible for the stable navigation in the middle of the road. From the initial state, the model finds the road marks and sets the near point for stable navigation on the road. The model then fires a production rule screening for a traffic sign, changes the state according to the result and sets the far point. In our model, the near and far point are used as control components and explained in detail in the next paragraph. If the model reaches the state *find far* it will continuously repeat the primary control loop.
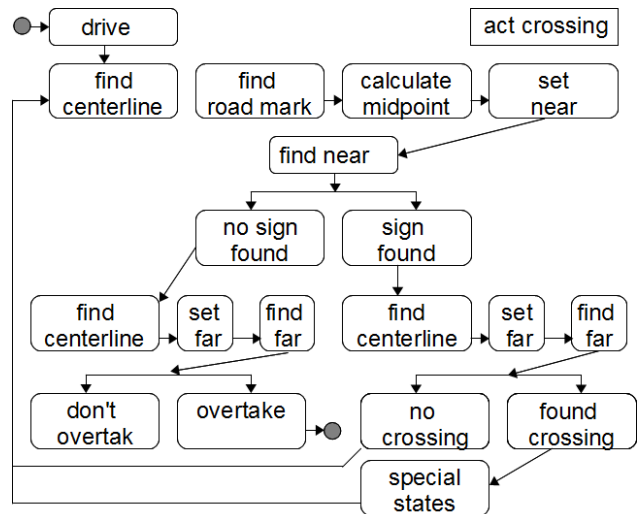

Fig. 5: UML-Diagram of the driver model. To execute the task drive, the model runs through several states.

This primary loop can be extended in case the monitoring component finds a special state like an intersection, that is, the condition part of a production rule investigating the right road marks on the right detects a crossing and the action part of this rule changes the state of the model, testing for other

given constraints. According to the result, the model might change the state or repeat the primary control loop updating the near and the far point for stable navigation.

Control

The control component of attention while performing a driving task manages the perception of lower level visual cues and the control over the vehicle (e.g., stopping). The model uses the simple concept of two salient visual attributes, based on earlier findings on locomotion (Llewellyn, 1971). Steering is described (Land & Horwood, 1995) as divided in two levels, guidance and stabilization, by using a „far" and a „near" region. Models of steering developed under this assumption have been proven to be consistent with empirical evidence.

This task specific information was required to construct the model. An issue to be addressed was what kind of strategies might be used by a human in a driving environment. Salvucci & Gray (2004) base the perception of a cognitive model on a near and a far point for guidance and stabilization. This model extends the idea of two levels to the extend, that with the far point, also other salient attributes are encoded. The visual attention of the model does not only switch in between the near point in the middle of the road and the horizon (or any other straight point ahead), but also encodes crossings, traffic signs or other cars coming from the right hand side on a crossing. For the here created artificial road environment, these two points account to capture the relevant aspects of the environment. The idea behind this wider use of the far point is the possible extension in further implementations. The far point could be used to determine other attributes relevant or irrelevant for the driving task and give an account, how errors while driving (e.g. overlooking of a traffic light before a noisy background during monitoring).
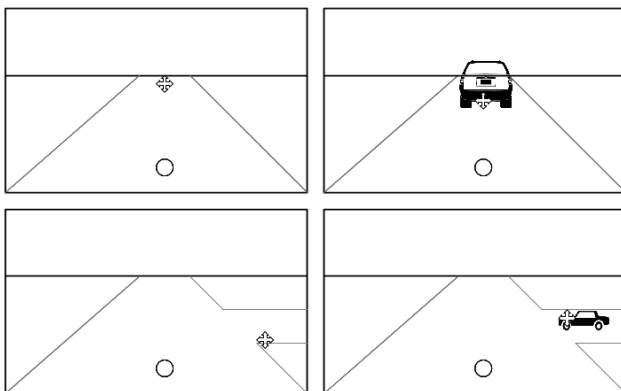

Fig. 6: Near and far points for a straight road with a vanishing point and a road segment with a lead car.

Fig. 6 illustrates the near and far points and how they are set in some possible situations during the driving task. The near point determines the position in the middle of the road. The far point is used to identify the direction of driving and other non-control points like vehicles, crossroads and traffic signs. If there is a lead vehicle, the distance between the two

points is determined, and in case it falls below a certain safety distance, the model can react according to that. At a crossroad the model will set the far point on the border of the crossing or on the vehicle approaching from the right. If the model decides to continue driving, it will not look again for another car at the crossroad, which is surely an issue for future implementations. Stopping is implemented by setting the far point onto the near point. The model will continue a loop until the other vehicle is not on the crossroad anymore and out of the safety distance.

Monitoring

The monitoring component captures the environment continuously and updates the declarative memory. In the here implemented driving environment, the situation awareness mainly focuses on other vehicles around or traffic signs. The model shifts the focus of visual attention towards a certain object which is then encoded as visual attribute in the declarative memory. This shift is achieved through three different methods of shifting attention: First by specific locations or directions, second by specific characteristics, and third by objects, that have not been in in the focus of attention yet. The combination of these methods of attention shift enables the model to create complex search strategies through the production rules. As ACT-R has a build-in memory decay mechanism, it might be possible to predict driver errors because the chunks encoding the current environment decay and can be forgotten if not updated continuously. Another source of possible driver errors could be the potential failure in encoding relevant information (e.g. to overlook a traffic sign or a vehicle).

Decision Making

The information provided by the control and monitoring component is used to determine if and what decisions must be made on the tactical level concerning the maneuvering (e.g. stopping or overtaking). Our focus on decision making assumes that the attributes in the environment are encoded correctly. The decision how to proceed (in what state of the model) is based on a pattern matching with the knowledge about the environment. If there is no crossing encoded, the corresponding production rule will not fire and the primary loop will continue. The decision whether to stop or to continue driving depends on the encoded traffic sign or on other vehicles. In our environment, the model always recognized these situations correctly, but it would be interesting in future implementations to observe the behavior and decision of the model in case, an error during encoding of attributes occurs.

In order for the model to produce a decision making process similar to humans, encoding a visual attribute and shifting visual attention cannot occur at the same time. For this model, the focus of attention is for example either on the near or far point or encoding a traffic sign.

Results and Discussion

Obviously, the model presented here does not account for all aspects of human attention during driving, especially not in a naturalistic environment. There are still quite some practical limitations in both, the architecture and the modeling effort itself. This study is an attempt to capture some of the difficult behaviors involved in driving. It also shows some of the limitations of the ACT-R architecture.

This study presents a simple simulation environment and a cognitive model of driver attention during car driving that is able to interact during run-time.

To obtain an integrated driver model of human driving behavior, it is essential to develop models in an architecture which is not task specific and can also model human behavior also in a different context, like ACT-R. This model is a first attempt to recognize, still simplified, traffic signs and crossroads and might make a first step towards the vision of accident-free driving. A majority (over 80%) of the automobile accidents are caused by the driver themselves (Statistisches Bundesamt, 2011). Nearly 16% of the accidents happen while turning and during exit, followed by disregarding the right of way (15%) and not-adapted speed (15%). Theoretically, the cognitive driver model could give a deeper insight for around 30% of the human errors while driving. However, it has to be taken into account that the model is still interacting with a simplified environment and not yet taking into account driver´s prior experience, which could be implemented by an increased attention in potentially high accident risk situations. The model and the environment do not present a complete picture of driver behavior yet, but they form a base to extend the ETA framework in any direction.

The ACT-R architecture limits the employment of the three components control, monitoring and decision making by using a serial cognitive processor. The serial processing of the subtasks is typical for the human bottleneck of information processing (Anderson et al, 2004). The resulting model is not an optimal model in a mathematical sense, but approximates human behavior and makes it possible, to mimic human cognitive capacities, simulate the dynamic nature of human driving behavior, and therefore to produce a cognitive adequate model of human driving behavior. If the model is, for example, occupied with an attention shift, it cannot simultaneously update the near point. Also, the model can only fire on production rule at a time and only one visual operation can be executed at a time. These processes take a certain time which are written in an output file. This file contains the time, the active buffer and the according event. This enables the researcher to compare the produced data with human data.

The knowledge representation comprehends declarative knowledge in chunks and procedural knowledge in production rules. For example, the scenario at a crossroad was implemented in 73 explicit production rules, which are highly detailed and is therefore open to future extensions of the model.

This study did not validate the model data so far. Future research could compare the output file data with human data, specially compare the attention shift of the model to human drivers over eye-tracking and the reaction times. But one must remember, that only most critical parts of key scenarios can be validated as no single method is sufficient enough to understand the complex task of human driving behavior yet.

ACT-R for complex tasks

Modeling such complex tasks in the cognitive architecture ACT-R presents quite some technical challenges to the modeler. For a complex driving task and the validation, the ACT-R model and the participants should interact in the same environment. However, for this validation, it must be possible from the technical side to connect the ACT-R model directly to the simulation environment, which can be technical challenging. Also thinkable is to develop a Lisp version of a driving simulator which can easily interact with ACT-R. If the simulator allows to extract the same information ACT-R does, the output files could be compared, even though the multiple implementation might be a potential source of errors. Also, the current version of ACT-R has some difficulties to directly recognize other components than the already implemented. The attempt to model such complex tasks in generic cognitive architectures show the applicability as well as the still remaining technical limitations.

However, such a complex task might raise the question not only about the limitations of the architecture itself, but also the modeling of human behavior. It might be a good approach to study the key scenarios of human attention during driving in more detail and transfer these results into the model code, breaking down the overall complex task into smaller subtasks in specific situations.

Conclusion and Outlook

We hope that this research will motivate more members of the computational modeling community to study human attention during driving a car and to overcome the practical limitations. Modeling of such complex tasks holds great promise for meeting the modeling challenges.

The progress to date in the development of cognitive architectures has been impressive, yet scientific gaps, technical challenges and practical issues remain. On one hand, cognitive models help to develop an understanding of driver behavior and aim to provide a theoretical account for human attention while driving. On the other hand, they are powerful and practical tools when implementing human-centered design and real-world applications. First steps towards the examination of the source of human mistakes through distraction from the primary driving task through secondary tasks like dialing a phone haven been taken (Salvucci, 2001) showing the feasibility of the architecture for these task and possible extensions.

The ACT-R architecture enables to elucidate interesting aspects and provides a theory of human attention while

driving. At the same time, human attention during driving is a challenging task for the ACT-R cognitive architecture. It shows the still existing limitations beyond basic laboratory tasks and pushes the research community to expand the architecture towards more complex and finally real-world tasks.

Acknowledgments

References

Aasman, J. (1995). Modeling driver behavior in Soar. In: Leidschendam, The Netherlands: KPN Research

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C. & Qin, Y. (2004). An Integrated Theory of the Mind. Psychological Review, 111, 1036

Anderson, J. R. & Lebiere, C. (1998). The atomic components of thought. Mahwah, NJ: Lawrence Erlbaum.

Bellet, T., Bailly, B, Mayenobe, P., & Georgeon, O. (2007). Modelling Driver Behavior in Automotive Environments. Critical Issues in Driver Interactions with Intelligent Transportation Systems. Cognitive Modelling and Computational Simulation of Driver Mental Activities.

Benmimoun, A. (2004). Der Fahrer als Vorbild für Fahrerassistenzsysteme? Ein fahrermodellbasierter Ansatz zur Entwicklung von situationsadaptiven FAS. 13. Aachener Kolloquium Fahrzeug- and Motorentechnik

Boer, E. R. (1996). Tangent point oriented curve negotiation. IEEE Proceedings of the Intelligent Vehicles 96 Symposium

Byrne, M. D. (2001). ACT-R/PM and menu selection: Applying a cognitive architecture to HCI. International Journal of Human-Computer Studies, 55, 41-84

Land, M., & Horwood, J.(1995). Which part of the road guide steering? Nature, 3, 77, 339-340

Liu, Y. (1996). Queuing network modeling of elementary mental processes. Psychological Review, 103, 116-136

Liu, Y., Feyen, R., & Tsimhoni, O. (2006). Queuing Network-Model Human Processor (QN-MHP): A computational Architecture for Multitasking Performance in Human-Machine Systems. ACM Transactions on Computer-Human Interaction 13, 37§70

Llewellyn, L. (1971). Visual guidance of locomotion. Journal of Experimental Psychology , 91, 245-254

Michon, J. A. (1985). A critical view of driver behavior models: What do we know, what should we do? Human behavior and traffic safety, 485–52, Plenum Press

Pomerlau, D., & Jochem T. (1996). Rapidly adapting machine vision for automated vehicle steering. IEEE Expert, 112, 19-27

Reid, L. D., Solowka, E. N.,& Billing, A. M. (1981). A systematic study of driver behavior steering control models. Ergonomics, 24 , 447-462

Salvucci, D. D. (2001). Predicting the Effects of In-Car Interface Use on Driver Performance: An Integrated Model Approach. International Journal of Human-Computer Studies, 55, 85-107

Salvucci, D. D. (2006). Modeling Driver Behavior in a Cognitive Architecture. Human Factors, 48, 362-380

Salvucci, D. D., Liu, A. , & Boer, E. R. (2001). Control and monitoring during lane changes. Vision in Vehicles, 9

Salvucci, D. D., & Gray, R. (2004). A Two-Point Visual Control Model of Steering. Perception, 33, 1233

Statistisches Bundesamt (online 20.12.2011) www.destatis.de