The Virtual Driving Instructor
Creating Awareness in a Multiagent System

I. Weevers[1], J. Kuipers[2], A.O. Brugman[2],
J. Zwiers[1], E.M.A.G. van Dijk[1], and A.Nijholt[1]

[1] University of Twente
Enschede, the Netherlands
i.weevers@tm.tue.nl
{zwiers, bvdijk, anijholt}@cs.utwente.nl
[2] Green Dino Virtual Realities,
Wageningen, the Netherlands
{jorrit, arnd}@greendino.nl

Abstract. Driving simulators need an Intelligent Tutoring System (ITS). Simulators provide ways to conduct objective measurements on students' driving behavior and opportunities for creating the best possible learning environment. The generated traffic situations can be influenced directly according to the needs of the student. We created an ITS - the Virtual Driving Instructor (VDI) - for guiding the learning process of driving. The VDI is a multiagent system that provides low cost and integrated controlling functionality to tutor students and create the best training situations.

Introduction

Driving simulators, such as the Dutch Driving Simulator developed by Green Dino Virtual Realities, offer great opportunities to create an environment in which novice drivers learn to control and drive a car in traffic situations. Although simulators still show some problems, such as simulator sickness [1], their main advantages are the objective measurements that can be carried out on the user's driving behavior and the creation of situations that suits the current student's skill level.
Driving instructors guide the students individually in acquiring the complex skills to become a proficient driver. In driving simulators, a student needs also this guidance. Since a simulator is capable of measuring the driving behavior objectively, the integration of an intelligent tutoring system with the driving simulator becomes a cheap and innovative educational technique. Accordingly, the system will evaluate the driving behavior in real-time and adapt the simulated environment to the student's needs, and a human driving instructor does not need to assist the student most of the time. In this paper, we present the Virtual Driving Instructor (VDI) - an intelligent tutoring multiagent system that recognizes and evaluates driving behavior within a given context using a hybrid combination of technologies. We will discuss driving education, awareness as the design principle for the system, and the architecture of the system.

Driving education and instruction

Driving involves carrying out driving tasks that suit the current situation. Driving education focuses on learning these tasks. Michon [2] discerned three driving task levels: strategic (route planning, higher goal selection), tactical (short-term objectives, such as overtaking and crossing an intersection) and operational (basic tasks, such as steering and using the clutch). McKnight and Adams [3] conducted an extensive task analysis on driving. Since this listing also includes tasks at all three levels, we used this listing for embedding driving knowledge into the VDI. Driving education not only implies knowing how to execute driving tasks, but also involves the evaluation and feedback processes. We carried out a two days empirical research on the practical experience of professional driving instructors at the Dutch national police school. This research provided insights into instruction aspects, such as feedback timing and the formulation of utterances. The most important results were that (1) the feedback usually is positively expressed; (2) that the student is being prepared for approaching complex situations by feedback; and (3) that the instructor mainly focuses on the aspects the exercise is meant for.

Awareness in education

One of our design questions concerned the knowledge of the instructor. For several reasons it is important that the instructor has different types of common and specific knowledge. There has to be a mutual understanding between teacher and student, the instructor should know how to drive, how to apply a driving curriculum, and so on.
A driving instructor needs to possess situational awareness for a good understanding of and application of expert knowledge in traffic situations. In addition, driving instruction involves more than only situational awareness and therefore we defined more awareness types. According to Smiley and Michon [5], awareness is the domain-specific understanding to achieve goals for this domain. This definition shows that an instructor should not only have knowledge for different driving education aspects, but also has to be aware of achieving goals within those knowledge domains. Probably the most important is situational awareness; the VDI needs to recognize and evaluate the student's driving behavior in relation to the current situation. Subsequently, the VDI determines the best piece of advice for this behavior and presents it to the student. We decided to divide this knowledge into two awareness types: First, the adviser awareness concerns the feedback directly related to the situation element or driving task on which the feedback is generated. Second, the presentation awareness relates to the context in which the feedback may be provided. This context depends on former feedback, the current situation and the student. Third, we identified curriculum awareness for dealing with the structure and management of the driving program. We used the different awareness types for the design. The situational, adviser and presentation awareness types include the VDI's core functionality. We chose to add curriculum awareness, since the recent introduction of the new standard for the new Dutch driving program the RIS (Driving Education in Steps) makes the integration of this aspect attractive to the market.

Developing the multiagent system

A recent approach to the design of intelligent tutoring systems is the multiagent system. We developed an agent for each awareness type: The Situation Agent implements situational awareness, the Presentation Agent implements presentation awareness and the Curriculum Agent implements curriculum awareness. The VDI's application domain is complex, unpredictable and uncertain. By using agents, we modularize the functionality of the design. In this way, the design becomes more flexible, easily changeable and extendible. The agents need to communicate for realizing intelligent tutoring behavior. We divided the agent's design into two layers. The communication layer deals with the communication with other agents. The agent layer implements the specific agent functionality and therefore differs for each agent.

Understanding and evaluating the situation and driving behavior

Situational awareness, as defined by Sukthankar [6], is one of the most fundamental awareness types for realizing the VDI. It involves recognition and evaluation of the driving behavior and the corresponding situation. Since both processes are closely related, we decided to combine them into one awareness type and thus into one agent: the Situation Agent. The VDI needs to perform a driving task and situation analysis. The VDI is only capable of accomplishing this when it knows the feasible driving tasks and situation elements. Sukthankar [6] decomposed these elements into three groups, which are (1) the road state, (2) traffic, speeds, relative positions and hypothesized intentions, and (3) the driver's self state. Since the groups concern only the situation and not driving tasks, we extended the knowledge of the driver's self state with these driving tasks. We used the task analysis conducted by McKnight and Adams [3], which is probably the most extensive driving task listing, for this purpose. Although the descriptions in the listing are sometimes too vague to express computationally, we used some empirically based parameters to apply to the description. By integrating the listing's tasks with the situational elements, we created relations amongst the elements and tasks. These are needed to understand the contextual coherence in the situation. We decided to integrate a continuous, dynamic and static driving task within the first analysis functionality to show that our design principle works for different situation types. We selected for speed control, car following and intersections.

Tree-like structures, as shown in figure 1, suit the integration of driving tasks with the situation elements. We adopted this idea from Decision Support Systems, which use the knowledge-based approach to declare the task structure. By defining the several tasks as different nodes in the structure, these tasks can be addressed separately. The nodes also represent the situation elements. When a situation element is present in the current situation or the student carries out a driving task, the corresponding node becomes active. Vice versa, when the element or task does not apply for the situation anymore, the node will be deactivated. The VDI recognizes the current situation and driving tasks by the activity status of the tree nodes. The VDI then is capable of generating rational feedback, since the structure allows evaluating whether the student performed or should have performed certain driving tasks in relation to the situation.

No matter what situation, a driver should always maintain an acceptable speed. The speed depends on current situational elements. We integrated some influencing situational elements that often occur in the simulator situations. These are the speed limit, acceleration or deceleration, turning intentions and the lead car's presence. Figure 1 shows the tree-like structure that combines the situation elements and driving tasks.

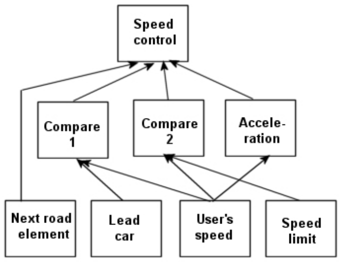

Fig. 1. Tree structure for speed control and car following

We discuss the structure by the components:

1. Next road element: Checks the next road element type.
2. Lead car: Checks whether there is a car in front of the driver.
3. User's speed: Determines the driver's speed. Speed limit: Determines the allowed speed for the current road.
4. Compare-1: Compares the user's speed to the distance to the lead car.
5. Compare-2: Compares the user's speed to the speed limit.
6. Acceleration: Checks whether the student is accelerating or slowing down.
7. Speed control: Determines which situational elements to consider as most important for the current situation.

We used arrows to indicate that one component (the speaker) might tell the other component (the listener) that its activity has changed. This speaker-listener principle - an event mechanism -has two advantages: (1) the speaker does not know what components are its listeners. In this way, the tree can be extended or changed easily, mostly without changing functionality of other parts of the tree. (2) The speaker only notifies its listeners when its activity state has changed. Therefore, the statuses of the components need not to be conveyed every update cycle, which will benefit the overall performance.

In all situations, the speed control component uses the compare-2 component for evaluating the user's speed in relation to the speed limit. However, in case there is a lead car (which is shown by the activity of the relating component) the relation of the user's speed to the lead car's distance is usually more important. Therefore, the VDI also considers the acceleration or deceleration by the student before evaluating the relation to the speed limit. By changing the speed, the student may be trying to achieve a higher or slower speed.

After the VDI conducted the recognition process for a given situation, the uppermost active component in the tree initiates the evaluation process. It coordinates the process by telling its speakers when to start their evaluation process. Subsequently, those speakers start their own evaluation process. In this case, the speed control component tells the compare-2 component (Figure 1) to evaluate, because the compare-2 component is active. If the compare-1 component is also active - because of an active lead car component - the speed control also tells that component to start evaluating.

Adviser Awareness Adviser awareness is embedded into the tree components. Each component evaluates a driving task or situation element and decides if it is important to provide feedback on that task or element. It measures the performance for that task by the current level and the progress, which both are classified in a local ?level x progress' matrix. The component calculates the level by using the deviation between the range of best values and the student's value. It determines the progress by comparing a range of previous levels and the current level. The matrix holds records for each field that maintain how much feedback is actually provided to the student on the specific component's status (level and progress). In this way, comments on a component can be chosen carefully with respect to a former status.

Each component may provide and time advice that is related to the driving task or situation element. After a component determines which piece of advice is currently needed, it passes it to its listeners. Some components receive pieces of advice from different speakers at the same moment. Since only one piece of advice can be provided at the same time, that component uses several methods to decide amongst those pieces of advice. First, predefined parameters assign the components a priority, which it uses to classify the pieces of advice. Second, the component knows the activities of the speakers' components and uses a simple rule-based choice algorithm to identify the most important piece of advice in case of a given component activity structure. A piece of advice is passed up through the tree. The highest coordinating component finally has the last judgment for the pieces of advice and puts forward the best overall piece of advice.

Evaluation phases A major difference between different trees is the duration. Speed control applies all the time, while an intersection is a periodic event. We decompose the latter events into three phases: the motivating, mentoring and the correcting phase. The VDI uses the motivating phase to prepare the student for approaching the situation. This may be an introduction or a reminder of former task performances. The mentoring phase deals with evaluating the task behavior while the student is conducting that task. The correcting phase evaluates the task performances afterwards. This evaluation may be in the short term - how did the student perform the task this time - as well as in the long term - how does the last performance compare to previous performances.

A hybrid tree structure Most tree components that recognize the presence of situational elements are straightforward, such as a lead car. However, the VDI also has to be capable of recognizing elements or driving tasks that are more vague, unpredictable and uncertain. For example, the other road user's intentions influence the situation intensively. These events are not easily captured by some parameters and depend on a variety of fuzzy data. Neural networks probably will help to guess such intentions. We can easily integrate another technique - such as a neural network - into the tree by creating a component that implements the technique internally, but externally works according to the speakers-listeners principle. This will result in a hybrid tree with the most suitable techniques for the related situational elements and driving tasks.

Contextual adaptive presentation of feedback

Presentation awareness concerns the provision of natural feedback. This involves formulating natural utterances and timing the utterances both naturally and educationally. We implemented this awareness by creating the Presentation Agent. This agent receives advice information about what to present from the Situation Agent. The Presentation Agent schedules, formulates and presents the feedback. Scheduling involves ordering different pieces of advice according to their priority and possibly ignoring them if they are outdated. Furthermore, it decides on the timing of the next piece of advice. For example, pieces of advice should not follow each other too quickly, since this will cause an information overload to the student. However, when the piece of advice is about dangerous behavior, the VDI has to tell that right away. Scheduling also depends on the phase - motivation, mentor or correction - of the situation elements or driving task. Since the mentor phase concerns the current context, which may change immediately, feedback in this phase should not be delayed. However, feedback in the motivation and correction phase may be provided within a short time range.

A flexible architecture

One of the main design principles was to design a system that uses a flexible architecture, such that future changes and extensions can be carried out without changing the VDI's basis. The multiagent approach in combination with our common communication channel realizes this flexibility. Existing functionality may be changed or extended, which only causes internal agent adjustment. New functionality can be added by adding new agents. Another opportunity within the current architecture is to develop an instructor for another application domain. Apart from adaptations to the simulator, we can create a motorcycle instructor by adjusting and replacing some agents. The driving tasks almost equal those of car driving, except for operational tasks. This also counts for the driving curriculum. These aspects require some adjustments. Student awareness creates a student profile and can probably be reused. Another application domain of the VDI may be another country. Apart from adapting the language, traffic rules and driving program, nothing needs to be changed.

Conclusions

We have presented the Virtual Driving Instructor, a multiagent system that realizes different awareness types in order to create an intelligent learning environment. It achieves different learning objectives and provides ways for an adaptive teacher-student relationship. We used a flexible and easily extendible architecture for integrating the awareness types by agents.

We created situational awareness. The VDI conducts driving behavior analyses with respect to the current situation. It recognizes and evaluates speed control, car following and intersection. Within the three evaluation phases, motivation, mentor and correction, it provides feedback on the level and progress of the student's performances. We created a tree structure that follows a speaker-listener principle. Dependency is reduced in this way, which benefits the process of changing or extending the tree structure.

With adviser awareness, we added advice knowledge that depends on a situation element or driving task. It deals with relating the piece of advice to the current level and progress of the student's performance. We developed presentation awareness to make feedback provision context aware, well-timed and with adaptive expression.

Finally, we added curriculum awareness to the system. It implements elements of the new Dutch standard for driving curricula, relating to the driving tasks, which the Situation agent evaluates. It saves the current student's performance.

The first results are promising. The provided feedback has a high contextual dependency and we achieved the integration of important driving educational aspects. These include different phases of feedback provision, priority classification for tree components in a given situation and the use of a driving program.

Acknowledgements

We thank Rob van Egmond and Ronald Docter of the Dutch national police driving school, LSOP, for their support with the research. We would also like to thank the colleagues of Green Dino Virtual Realities.

References

1. Casali, J.G. Vehicular simulation-induced sickness, Volume 1: An overview. IEOR, Technical report No. 8501, Orlando, USA (1986)
2. Michon, J. A critical view of driver behavior models: What do we know, what should we do?. In Evans, L., and Schwing, R. (eds.), Human Behavior and Traffic Safety, Plemum (1985)
3. McKnight, J., Adams, B. Driver education and task analysis volume 1: Task descriptions. Technical report, Department of Transportation, National Highway Safety Bureau (1970)
4. Pentland, A., Liu, A. Towards augmented control systems. In Proceedings of IEEE Intelligent Vehicles (1995)
5. Smiley, A. and Michon J.A. Conceptual framework for generic intelligent driving support. Deliverable GIDS/I, Haren, The Netherlands, Traffic Safety Centre (1989)
6. Sukthankar, R. Situational awareness for tactical driving. Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (1997)