The Virtual Driving Instructor: A Multi-agent Based System for Driving Instruction

I. Weevers[1], J. Kuipers[2], A. Brugman[2], J. Zwiers[1], B. van Dijk[1] & A. Nijholt[1]

[1]Center for Telematics and Information Technology

TKI-Parlevink Research Group
University of Twente
PO Box 217, 7500 AE 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 multi-agent system that provides low cost and integrated controlling functionality to tutor students and create the best training situations. Its architecture offers flexibility for adding and extending several awareness types that are required for a good driving instructor. In our design, we added situational, adviser, presentation and curriculum awareness. The design provides a simple interface for future extensions, such as the addition of student, environmental and scenario awareness.

The Virtual Driving Instructor: A Multi-agent Based System for Driving Instruction

I. Weevers[1], J. Kuipers[2], A. Brugman[2], J. Zwiers[1], B. van Dijk[1] & A. Nijholt[1]

[1]Center for Telematics and Information Technology

TKI-Parlevink Research Group
University of Twente
PO Box 217, 7500 AE 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 multi-agent system that provides low cost and integrated controlling functionality to tutor students and create the best training situations. Its architecture offers flexibility for adding and extending several awareness types that are required for a good driving instructor. In our design, we added situational, adviser, presentation and curriculum awareness. The design provides a simple interface for future extensions, such as the addition of student, environmental and scenario awareness.

1 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 [2], 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. With the increasing emphasis on the reduction of traffic accidents, handling more crowded roads, and the reduction of CO_2, the use of driving simulators as learning environments becomes more attractive.

Many European countries require novice drivers to learn driving at driving schools. 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 multi-agent system that recognizes and evaluates driving behavior within a given context using a hybrid combination of technologies. We will discuss driving education and subsequently awareness as the design principle for the system, stressing situational, adviser, presentation, and curriculum awareness. The first awareness type deals with recognizing and evaluating driving behavior within an arbitrary situation. The second and third types deal with feedback and relate this to the previous and current context in which feedback will be provided. The fourth type concerns managing and structuring the driving program. We implemented the awareness types by looking at theoretical knowledge and practical experience in driving education.

Subsequently, we will discuss the architecture of the system that integrates the different awareness types and offers flexible opportunities for future extensions and changes. Although we are still in the process of evaluation and development, the first results suggest that the integration of the techniques offers support for the further development of the system.

Interests in the field of artificial intelligence, driving simulators and agents prompted Green Dino Virtual Realties and the Parlevink Group of the University of Twente to explore collaboratively the opportunities in this area.

2 Driving education and instruction

Driving involves carrying out driving tasks that suit the current situation. Driving education focuses on learning these tasks. Michon [6] 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 [7] 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.

Bloom et al. [1] provided a taxonomy of learning objectives by which an education program should be influenced. These objectives include factual, conceptual, procedural, and meta-cognitive knowledge. We used these knowledge types for understanding the driving information that the VDI needs to convey to the student.

Gardner [5] mentioned four important properties of a student-teacher relationship, which are (1) tutoring, (2) encouraging, (3) advising, and (4) monitoring. A student requires feedback presentation to be matched to the context and personality. The teacher uses the four properties to realize this. Therefore, we integrated them into our design.

2.1 Awareness in education

One of our design questions concerned the knowledge of the instructor. For several reasons it is important 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. The driving instructor should not only have this knowledge, but also use it. He has to apply the knowledge to determine what is best for the current context and how he can achieve a good learning environment.

Sunkthankar [10] carried out research on situational awareness. He looked at many systems that use this awareness type to indicate the understanding of dynamic entities in an environment and their impact on someone's decisions within that environment. Sukthankar [10] used Endsley's [4] definition for situational awareness: An expert's perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future.

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 [9], 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. For example, he has to achieve educational based feedback.

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. Furthermore, 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.

At first, we left out student awareness and environmental awareness. Although they have an important added value for a learning environment, we considered them less important for our design, which focuses on the core of the driving tutoring system. Student awareness deals with creating a student profile, based on behavioral and psychological aspects of the student. The VDI will use this profile to adapt the feedback and curriculum structure to the student's needs. Environmental awareness concerns the knowledge about the roadmap structure, the city, rural area and so on. It strives to create and adapt the driving route for the lessons to the needs of the student. In a more advanced version, this awareness type may even create situations, including other road-users with preferred driving behavior, dynamically in real-time.

We designed an architecture in which we can add or adapt awareness types easily.

3 Developing the multi-agent system

A recent approach to the design of intelligent tutoring systems is the multi-agent system. Weiss [11] stated that multi-agent systems offer a promising and innovative way to understand, manage, and use distributed, large-scale, dynamic, open, and heterogeneous computing and information systems. The driving simulator with its situations relates to this domain. Each agent within the system tries to accomplish its own goals and by cooperating, the system's behavior shows the overall design principles. We developed an agent for almost 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. We did not use an Agent Communication Language (ACL) for the agents' communication, but applied the communication layer, which is the same for all agents, in a straightforward way. We did not use a popular ACL, such as KQML or FIPA ACL, since we did not want to restrict the information exchange by using the defined speech acts. Furthermore, we avoided the use of another higher-level language, such as XML. The VDI works in a real-time environment and therefore delays are highly undesirable. Wrapping the contents into or parsing it from a high-level message representation only causes unnecessary delays. All agents run on the same processor and therefore a low-level common message structure (sender, receiver, message type and contents) suits the design.

We will discuss the agent layers individually in the following sections, since they differ for each agent.

3.1 Understanding and evaluating the situation and driving behavior

Situational awareness, as defined by Sukthankar [10], is one of the most fundamental awareness types for realizing the VDI. It involves recognition the driving behavior and the corresponding situation. In addition, situational awareness deals with evaluating the driver's behavior. Since both processes are closely related, we decided to combine them into one awareness type and thus into one agent: the Situational 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 [10] 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 [7], 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 structure. Tree structures 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 tree nodes, these tasks can be addressed separately. Tree 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 tree 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.

Speed control and car following. 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 structure that combines the situation elements and driving tasks.

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

We discuss the tree 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.

When a situation element is valid or driving task applies for the current situation, the related tree component becomes or stays active. 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.

In addition, the VDI also has to consider 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.

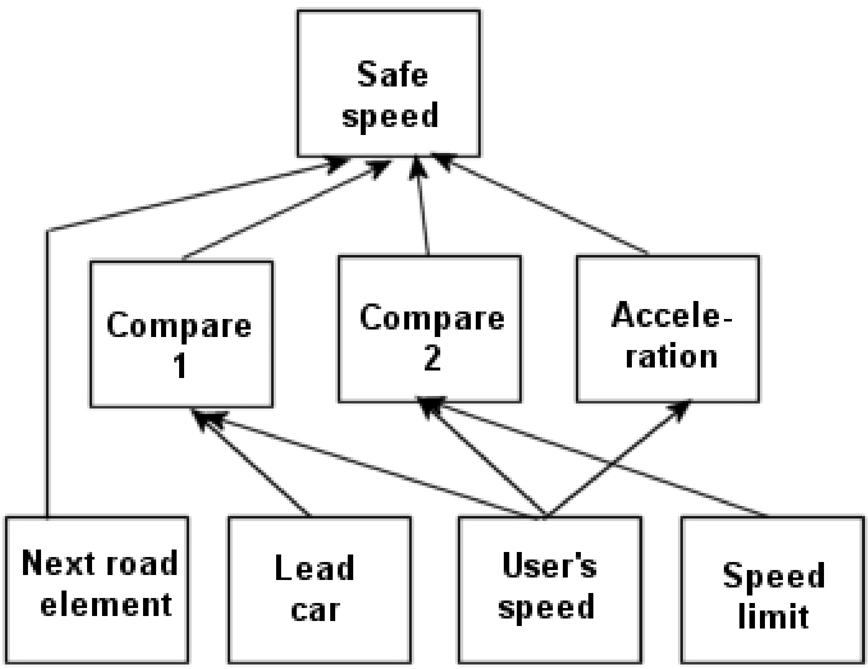

Figure 1 - Tree structure for speed control and car following.

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 matrix (Figure 2). 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.

Progress Level	Decrease	Constant	Increase
Bad			
Average			
Good			

Figure 2 - Progress/Level table

Each component has a specific implementation, such that the advice provision is related to the driving task or situation element. The number of times for and timing of provision of a piece of advice, depends on the driving task's status, which is represented by a tree component.

After a component determines which piece of advice is currently necessary, 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 (determined by priority values assigned to driving tasks by McKnight and Adams) assign the components a priority. The component classifies the pieces of advice according to their assigned priority. 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.

Intersections. We used the presented technique - tree structure with components for driving tasks and situational elements - for the situational elements relating to intersections. The focus lies on approaching the intersection, and more specifically the approaching speed. Figure 3 shows the corresponding tree-like structure.

Again, we discuss this structure by its components:

1. Traffic lights: Checks if there are traffic lights at the intersection
2. Turning intentions: Checks if the driver is planning to turn
3. Right of way: Checks whether the student has right of way.
4. Users speed: Determines the users speed.
5. Next road element: Checks the next road

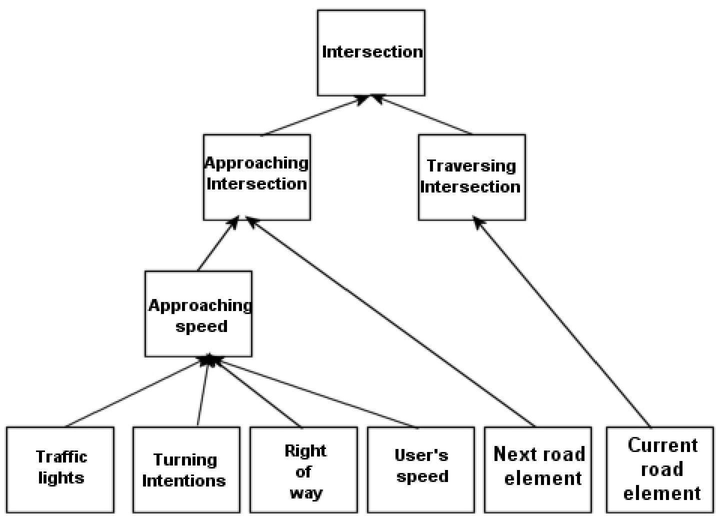

Figure 3 - Tree-like structure for intersection

element type

6. Current road element: Checks the current road element type.

7. Compare-1: This component evaluates the current students (users) speed to the status of traffic lights.

8. Compare-2: This component evaluates the current students (users) speed to the intended turning.

9. Compare-3: This component evaluates the current students (users) speed to the fact of having or having no right of way.

10. Approaching speed: Evaluates the approaching speed to the intersection according to turning intentions, right of way and the presence of traffic lights.

11. Approaching intersection: Checks when the driver is approaching the intersection

12. Traversing intersection: Checks when the driver is traversing the intersection

13. Intersection: Is the most abstract component that coordinates the complete intersection process.

The tree has two mutually exclusive processes: approaching and traversing the intersection. By structuring the tasks in such a way, we can evaluate the order of execution. Suppose we integrate the use of the indicator into the approaching intersection component. If the traversing intersection component becomes active, the VDI evaluates the approaching intersection component. When the indicator is not turned on, when the student has to turn, the VDI can notify this failure in time.

Evaluation phase. The main difference between the intersection and the speed control component is the duration. Speed control applies all the time, while the intersection is a periodic event. We decompose those 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 an intersection or 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.

3.2 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.

After the Presentation Agent has decided on the next piece of advice, it determines the corresponding utterance. The formulation follows some educational principles: The utterances:

1. Have a positive expression.
2. Are not too long, since this will also cause mental overload.
3. Have to be informative and explain why behavior is good or bad.
4. Have to change according to the number of times feedback is provided on the same given topic.
5. Are adapted to the current task or situational element phase motivation, mentor or correction.

We recorded all utterances in Dutch and show some of them in Table 1 (English translation). The first one shows encouraging tutoring, the second shows encouraging by notifying progress, the third is feedback in the evaluation phase (after the student finished a task - in this case approaching the intersection), the fourth concerns advice on repeating bad behavior and the last one shows an explanation.

The presentation is realized by a software-based interface to the loudspeaker. Before a piece of advice is actually provided to the student, it went through some internal processes of the VDI. The internal representation of advice in these processes differs from the external representation - the actual utterance.

The conversion from internal to external representation happens just before the actual provision to the student. Therefore, there is not many work involved in integrating another interface - such as a graphical display. In that case, only a new conversion algorithm has to be implemented.

3.3 Managing the driving curriculum

An intelligent tutoring system has to provide the learner various training exercises and adequate instruction in order to benefit the acquisition and training potential of skills [3]. Recently, the new Dutch standard for driving curricula - RIS - was introduced and we integrated its elements that match the situational elements and driving tasks of situational awareness. The standard consists of modules that keep track of current performances, level and progress of the student for the related driving tasks. The Curriculum Agent was developed to embed curriculum awareness into the VDI and it communicates with the Situation Agent to update and to synchronize the information on the student's performances.

The Curriculum Agent writes the student's information to a file and reads it when the student continues his or her driving lessons. The VDI saves individual data, and in this way, the student is able to share his results with other people after the driving sessions. The agent also will be responsible for selecting training goals according to the student's progress and personality. Currently, the VDI only evaluates a few driving tasks, which leaves no space for selecting amongst a collection of tasks.

4 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 multi-agent 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. The language then requires a replacement of the pre-recorded utterances, another national driving program requires an adjustment of the Curriculum Agent and the traffic rules require some small tree component adjustment in the Situation Agent.

5 Conclusions

We have presented the Virtual Driving Instructor, a multi-agent 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. The amount of conveyed information should not overload the student and the utterances of feedback have to be natural and well timed. However, the VDI has to comment on dangerous behavior at once. We used knowledge of the adviser awareness, timing algorithms and pre-recorded utterances to present the feedback to the student.

Although this feedback was positively expressed, it was directed towards both good and bad driving behavior. 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 situational 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.

Most problems occur with traffic situations that are not covered by situational awareness. For example, the presence of other road users at an intersection affects the student's behavior, but this is not taken into account by the VDI. Another problem relates to feedback timing. The VDI times events by looking at the previous context and not by looking at the expected next context. A human instructor will refrain from providing less important feedback when more important feedback possibly has to be given within a couple of seconds. We think however, that those problems can be overcome in future developments.

6 Future directions

The VDI integrates functionality, which shows that the design principles offer a basis for future developments. A full application requires a more developed version. First, the greatest priority lies with the extension of the situational awareness for conducting a more advanced driving task and situation analysis. In relation to that, the Presentation and Curriculum Agent have to be extended to deal with the extended situational awareness.

The Presentation Agent will also influence the utterance formulation by the created student profile, such as the student's personality. For example, a shy student requires relaxed feedback. Another improvement is real-time speech synthesis, since this will realize more dynamically and context adapted utterances.

Other possible directions include the development of other agents, such as the Student, Environmental and Scenario Agent. The first concerns the creation of a student profile that will be used to adapt the driving program and the feedback provision. The second deals with route planning to find the best suitable route according the training goals. The third is a more advanced agent that will create situations - with other traffic participants - to achieve the best possible training exercises. Dynamic scenario generation probably will be an innovative alternative for lesson construction.

In the current simulator, there is just a monolog, and no dialog between teacher and student. With the empirical research session, we noticed there is intensive interaction between them. Therefore, for creating a better VDI we need to integrate advanced speech recognition and to produce additional functionality.

An important aspect of driving is eye gaze behavior. The student has to learn to look at all important aspects of traffic situations. Furthermore, this eye gaze is important for the VDI for other reasons. The driver's gaze seems to be strongly correlated with the driver's mental state in real life driving [8]. The eye possibly conveys relevant information in the driver's maneuver intentions and how he feels while driving. The Situation Agent can use this to respond better to the student's behavior. The Student Agent can use this for creating the student profile.

7 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 the software development group of Green Dino Virtual Realities.

8 References

1. Bloom, B.S., Englehart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R. Taxonomy of educational objectives. The classification of educational goals, Handbook 1: Cognitive Domain. New York: Longmans, Green, Co. (1956).
2. Casali, J.G. Vehicular simulation-induced sickness, Volume 1: An overview. IEOR, Technical report No. 8501, Orlando, USA (1986).
3. Dong Mei Zhang From task sequencing to curriculum planning, In Proceedings AI ED 97, (1997).
4. Endsley, M., Towards a theory of situation awareness in dynamic systems, Human Factors (1996).
5. Gardner, T. The TEAM program: Enlisting teachers to aid students with behavioral problems. Commission scolaire Sault-St-Louis.
6. 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).
7. McKnight, J., Adams, B. Driver education and task analysis volume 1: Task descriptions. Technical report, Department of Transportation, National Highway Safety Bureau (1970).
8. Pentland, A., Liu, A. Towards augmented control systems. In Proceedings of IEEE Intelligent Vehicles (1995).
9. Smiley, A. and Michon J.A. Conceptual framework for generic intelligent driving support. Deliverable GIDS/I, Haren, The Netherlands, Traffic Safety Centre (1989).
10. Sukthankar, R. Situational awareness for tactical driving. Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (1997).
11. Weiss, G. Multi-agent Systems A Modern Approach to Distributed Artificial Intelligence. Massachusetts Institute of Technology (1999).