AN AUTONOMOUS INTELLIGENT DRIVING SIMULATION TUTOR FOR DRIVER TRAINING AND REMEDIATION: A CONCEPT PAPER

Matthew Romoser
University of Massachusetts, Amherst
Amherst, Massachusetts, USA
Email: mromoser@ecs.umass.edu

Summary: An intelligent tutoring model for use in a driving simulation training platform is proposed. Driving simulators by themselves cannot teach and staffing driving simulators with live trainers limits their ability to reach a wide audience. Research has shown that customized feedback, coupled with active practice in a simulator is very effective in changing a driver's behavior for the better. A driving simulation training program which utilizes an intelligent tutoring system (ITS) can diagnose driver errors, tailor feedback to the student's specific needs, determine when a student has mastered a specific skill set and can provide remediation as necessary. A brief discussion of basic ITS architecture is provided. An ITS model that has been successful in teaching individual skills in other domains (such as mental rotation) is applied to driving simulator instruction. The various critical components of the ITS, including the domain model, student model and tutoring model, are discussed in detail and a working example provided.

INTRODUCTION

A properly designed Intelligent Tutoring System (ITS) can provide instruction, diagnose errors on behalf of the student, modify instruction to address errors and individual differences, and provide customized feedback. There are four basic components to an ITS – the student, domain, and tutoring models and some sort of overall system control or management (Siemer and Angelides, 1998). Such ITS models could be used to create an autonomous driving instruction simulator (or online tutor) which could diagnose student errors, assess progress toward desired behavior, and then provide customized feedback to the driver. In the remainder of this article, we will begin with an introduction to ITS and then propose an architecture that could be utilized for autonomous, stand-alone simulator-based driver training.

INTELLIGENT TUTORING SYSTEMS (ITS)

Basic ITS Architecture

Before proposing an ITS model for driving simulator instruction, it would be beneficial to first quickly describe the "typical" ITS. General agreement in the literature is that an ITS usually should contain the following: 1) The Domain Model, 2) The Tutoring Model, 3) The Student Model, and 4) Overall System Control / Management (Woolf, 2002; Siemer & Angelides 1998; Sizoneko, 2010). A typical ITS architecture is demonstrated in Figure 1.

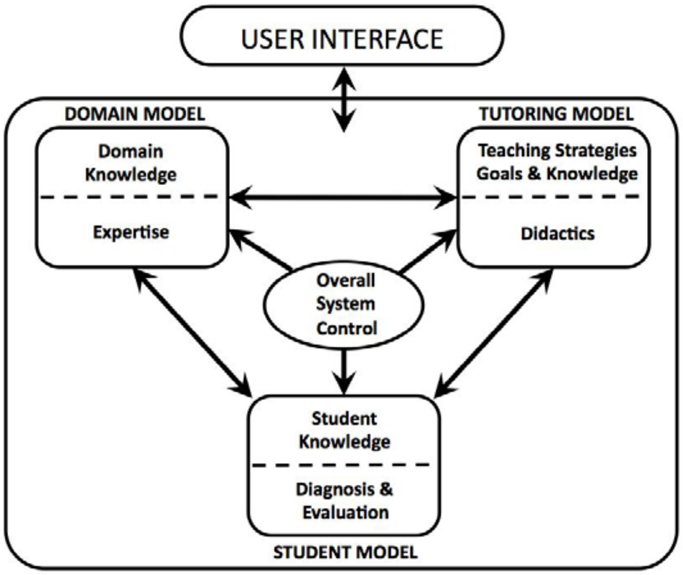

Figure 1. Typical ITS architecture; modified from Siemer & Angelides (1998)

The domain model contains the domain knowledge and expertise. Domain knowledge could be modeled in the form of expert responses, facts, procedures, or solution paths. The tutoring model contains methodology for teaching strategies, remedial strategies, and the overall instructional goals. The student model contains the tutor's representation of what the student does or does not know, a diagnosis strategy or method, and perhaps even a representation of individual student differences. In practice, there is typically some overlap between the models. For instance, the domain model might overlay the performance of an expert directly on the student's performance protocol to determine what deviations have occurred from "how the expert would respond." The overall system control does exactly what its name implies – it coordinates the efforts of the three ITS models, ensures they communicate, and oversees the flow of the learning experience.

According to Siemer & Angelides (1998), the student model can be divided into two key areas of responsibility – knowledge representation and diagnosis. Individual learning differences and preferences would also be stored in the student model. This information would be used by the tutoring model and would drive pedagogical strategy and interface presentation. The diagnosis portion of the student model allows the ITS to construct the knowledge representation of the student.

AN INTELLIGENT DRIVING SIMULATOR INSTRUCTION SYSTEM

A simulator that utilized an ITS would be capable of automatically diagnosing driver performance, providing feedback, and customizing the curriculum to fit the learning needs of the student driver. Research has shown that active practice in a simulator coupled with customized feedback on errors is effective in improving driver performance (Pradhan, Pollatsek, Knodler & Fisher, 2009; Romoser & Fisher, 2009; Wang, Zhang & Salvendy, 2010). Very little research has been done in the area of ITS for driving instruction using simulators. Only one article was found. Weevers et al. (2003) created the Virtual Driving Instructor (VDI), a multiagent ITS system for simulator-based driving instruction. The VDI utilized separate agents for things such as situational awareness, presentation of material and curriculum management and a tree-structure for managing individual skills and driver behavior. The authors also propose the use of neural networks to help the tutor guess the intentions of the driver. Unfortunately, it is not clear if the model was ever implemented into a driving simulator, as no further information is available in the literature.

However, it is possible to implement an ITS for a driving simulator without the need to design high-end artificial intelligence components (such as neural networks) to manage them. In Figure 1, the basic components of an ITS model were introduced: Domain Model, Tutoring Model and Student Model. Below a way in which each of these components might be designed for an ITS-based driving simulator trainer is discussed.

The Domain Model - Modeling Domain Knowledge

The first step is to determine the critical skills in the domain. The domain model is the portion of the tutor in which the "expert" representation of the knowledge to be taught is stored. One approach to modeling the domain is to create knowledge tiers. A general conceptual model for knowledge tiers in an ITS is outlined in Figure 2.

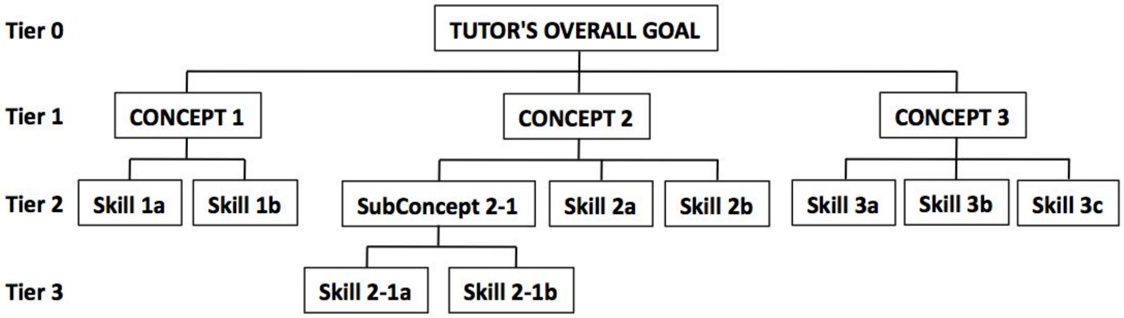

Figure 2. Conceptual Tiered Domain Model for ITS Driving Simulator

Topics such as "Attention Maintenance," "Speed Management," "Older Driver Issues," or "Hazard Anticipation" are some examples of top-level goals for driver training. Each would be incorporated into its own tutor. For the purposes of this article, we will use the example of "Hazard Anticipation Training" as the tutor's overall goal. Therefore Tier 0, the topmost level, contains the tutor's overall instructional goal – "Hazard Anticipation Training." The Tier 0 objective is then further broken down into individual concepts. Tier 1 might include concepts such as "Hidden Pedestrians," "Hidden Turning Vehicles," "Hidden Cross Traffic," "Sudden Lane Changes," "Lane Changes," and "Hard Braking / Headway." Each Tier 1 concept is then further decomposed into its respective skills. In the case of skill-based tutors, these should be expert representations of each skill. For example, the Tier 1 concept of "Hidden Pedestrians" might be broken down into skills such as "Identify Obscuring Object," "Speed," "Lane Position," and "Scan for Pedestrian." The ideal performance of the driver on each of these parameters is stored here. In the case of driving tutors, ideal performance might be derived from the baseline performance of drivers with the lowest crash rates – namely adults twenty-five to fifty-five years of age with at least ten years driving experience. Ideal performance might include information on proper eye fixations, target speed, braking and ideal lane position. Skills in Tier 2 could be further broken down into sub-skills – in which case the Tier 2 skills could also be thought of as "sub-concepts." For instance, the Tier 2 skill of "Indentify Visual Cues" might include the Tier 3 sub-skills to identify "Signs," "Crosswalk," and "Stopped Vehicles." The model lends itself to any number of concepts and tiers and as a result is extremely flexible.

The Student Model – Diagnosing and Evaluating Student Knowledge

The student model is the heart of the ITS. It is responsible for tracking the student's progress relative to the skills defined for each concept. It is also responsible for deciding when a student has mastered the required skills and when extra remediation is needed. A well-designed student model should quickly graduate students who have already mastered or have quickly mastered the material and provide extra remediation and practice for those who are struggling with certain skills. An example of an adaptable tutoring model for teaching individuals skills was developed by Shute (1995) called the SMART model (Student Modeling Approach for Responsive Tutoring) and later modified by Romoser, Woolf, Bergeron & Fisher (2004) with the introduction of a remediation criterion and curve adjustment based on learning style.

$P(S_i)$ is defined as the probability that the student has mastered the i^{th} skill of the concept. The probability that the student had mastered the skill is adjusted based on the number of hints or interventions required to arrive at the correct answers using a series of quadratic equations. The curves used are shown in Figure 3. After the student answers correctly, a New $P(S_i)$ is determined using the curve corresponding to the level of hints used. As an example, using Figure 2 as a guide say a student's current $P(S_i)$ on skill *i* was 0.50. If the student required zero hints to solve the problem, then the new $P(S_i)$ would be *adjusted up* using the L_0 (level zero) Hint curve to approximately 0.70 and a more difficult problem on skill *i* could be chosen to be answered next. If the student required two hints to solve the problem, then $P(S_i)$ would be *adjusted down* using the L_2 (level two) Hint curve to approximately 0.35 and a simpler problem on skill *i* chosen to be answered next. The model contains a mastery threshold set at 0.90, at which point the student would graduate from that concept (having mastered it). It also contains a remediation threshold set at 0.10 at which point the student would be removed from the tutor to receive remedial instruction on that particular skill. A sufficient library should be developed for each concept that contains problems varying in difficulty on each of the concept skills. Based upon the student's calculated probability of having learned each of the skills, the tutor selects a problem that is appropriate for the student's level of aptitude.

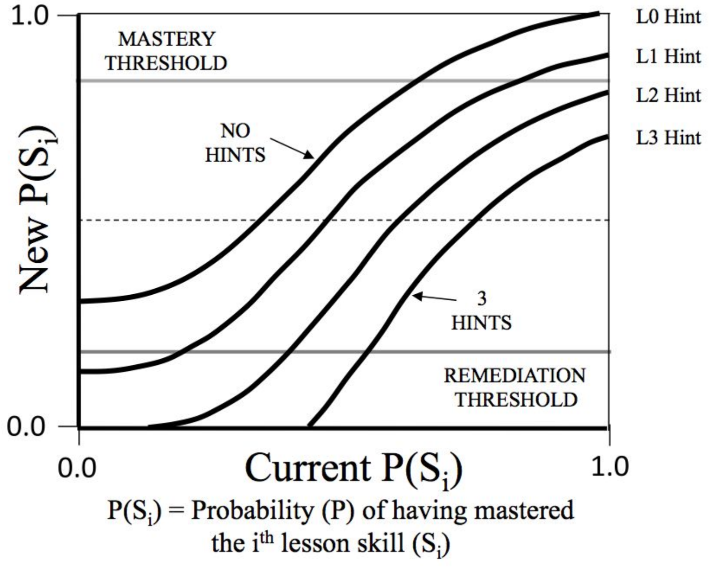

Figure 3. SMART model for intelligent mastery determination (Shute, 1995); modified by Romoser et al. (2004)

Again, let us fit our working example of the hazard anticipation training ITS to the model in Figure 3 using the first concept of the hazard anticipation tutor – "Hidden Pedestrians." In order to evaluate the student's ability on each of the skills, a library of hidden pedestrian scenarios would need to be developed for the simulator. As the student drove a scenario, the ITS would evaluate the student on each of the module's skills – "Identify Visual Cues," "Speed," "Lane

Position" and "Scanning for Pedestrians." An eye tracker could be used for identifying visual cues and scanning for pedestrians. Percent deviation from the ideal could be used for speed and lane deviation performance. Ideally, in a hidden pedestrian situation, the student should fixate the visual cues (such as a truck parked in front of a crosswalk and the crosswalk itself), slow down, move slightly over in his lane to allow a clearer line of sight, and then scan far to the right for potential pedestrians. Should the student not meet the requirements (not look, not slow, not move over slightly) on one or more of these skills, then the ITS would generate feedback (a hint) for the student. For instance, an automated voice might say, "You should slow down to give yourself more time to react to an unexpected pedestrian" and then repeat the scenario. Feedback and hints could also come in the form of a replay of the student's drive with automatically generated commentary to draw attention to errors. After the scenario was complete, the $P(S_i)$ would be adjusted up or down according to the student's performance on that skill and number of hints required to perform the skills correctly. Each time, the simulator would choose a different scenario from the library to aid in skill generalization. If at any point the student's $P(S_i)$ on a skill fell below the remediation threshold, they would be directed to a remedial lesson on the skill and be given the opportunity to review the primary lesson. Once the student is able to perform such that the $P(S_i)$ fell above the mastery threshold for all the skills, the student would be graduated from the module.

The Tutoring Model – Teaching Strategies & Course Management

The tutoring model is responsible for the overall flow and management of the course material, lessons, evaluations and remedial components. It is flexible in nature with each course concept contained within its own module. Because they are self-contained, the modules can be delivered serially, with the student required to finish one module before being allowed to continue on to the next module, or in parallel, with the student given the choice of which modules to complete first. Modules for sub-concepts could even be nested within the module belonging to its parent concept. A generalized Conceptual Tutoring Model for the ITS is outlined in Figure 4. The below generalized model corresponds to Concept 1 from Figure 2.

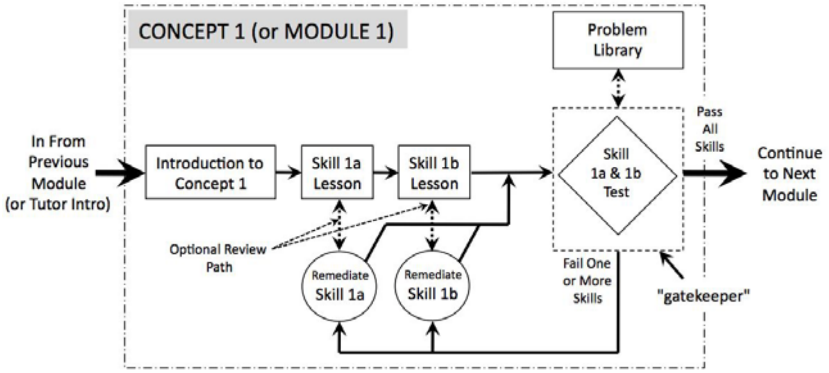

Figure 4. Conceptual Tutoring Model for ITS Driving Simulator

Applying this concept again to our working example of hazard anticipation, the hidden pedestrian concept would be contained within its own module, as would each of the other concepts ("Hidden Turning Vehicles," "Hidden Cross Traffic," "Sudden Lane Changes," "Lane Changes," and "Hard Braking / Headway"). The student would enter the module either from a lesson index (if the student is allowed to choose) or from the previous module (if arranged serially). The student would first receive an introduction to the module, outlining what the goals of the hidden pedestrian module were and what they should expect to learn. They would then receive lessons on each of the module's skills. These lessons could contain any number of media for teaching including text, audio, video, and interactive learning modules. The goal of the lessons is to provide information on how the student should drive in a situation in which pedestrians might be hidden. For example, in the "Identify Visual Cues" lesson, students might be shown plan views of various situations in which pedestrians might be hidden by vehicles, vegetation, or buildings and be asked to click on the area in which pedestrians might be obscured. In the lesson on speed, they may be shown animations of how and when to slow down when approaching these types of hazard to give themselves a safety buffer should a pedestrian appear.

After the student has finished all the lessons, they are sent to the "gatekeeper." The gatekeeper is part of the student model, which was discussed in the previous section. The gatekeeper contains some means of evaluating the student on the skills just learned. In the case of our driving simulator example, the student would be tested by having him drive a series of hidden pedestrian scenarios. These scenarios would be contained in the library that would be accessed by the gatekeeper. If, according to the student model, the student has demonstrated mastery of each of the skills taught – the $P(S_i)$ for all skills fall above the mastery criterion – then he would be graduated from the module. If, on the other hand, the student fails one or more of the skills – the $P(S_i)$ falls below the remediation criterion – then he would be sent by the gatekeeper to one or more remediation lessons, which would be designed to provide additional feedback and guidance on that particular skill. Afterward, the student would then have the option to revisit the individual lessons, or to return to the gatekeeper for a new test. Before returning to the gatekeeper (student model), after remediation the $P(S_i)$ would be reset to 0.50.

CONCLUSIONS

A model for an autonomous intelligent driving simulation tutor for driver training was presented here. The model is extremely flexible and modular in nature allowing for easy expansion, modification and rearrangement of learning concepts as needed. The system is capable of diagnosing the errors of drivers within individual concepts and providing them with customized feedback and replay of their drives.

The development of an ITS for driving instruction would not be a trivial undertaking. The complexities of which cannot be address adequately in a concept paper such as this. The strategies that a designer could use to implement such a system are equally diverse. For instance, what constitutes ideal performance and how is it defined? One strategy would be to use parameters based on experienced, adult drivers and compare those to the performance of the student. This method could be employed not only for speed, braking and lane position, but also road scanning using fixation coordinates versus time as determined by an eye tracker. How hints and remediation should be delivered is another challenge. Should interventions be delivered live while the student is driving or saved for the end of the drive? How many scenarios should be developed for the ITS to select from? How long should each drive take? How should information flow between modules? These are all questions that need to be answered by the designer. The purpose of this article was to introduce a straightforward conceptual model that could be employed to begin tackling these types of design issues.

Too often, in their efforts to develop a course for online delivery, institutions publish their material for delivery using the easiest means possible. In other words, they *make the instructional systems to fit the technology*. This flies in the face of traditional human factors principles. The real challenge for instructors and ITS developers is to *make the technology fit the instructional systems.* This means constructing the technology such that it adheres to the best pedagogical methods available for the target subject. By themselves, driving simulators cannot teach. Given the research that shows that customized feedback and replay of errors is extremely effective in changing driver's behavior for the better, tutors such as the one described in the current article can deliver automated instruction to a much wider audience and at a much lower cost than can be achieved with a live instructor. An ITS-based driving simulator can provide the customized learning necessary for positive learning transfer – and, by making training available to more individuals, result in saving more lives than instructors alone.

REFERENCES

Federico, P-A. (1999). Hypermedia environments and adaptive instruction. *Computers in Human Behavior, 15,* 653-692.

Lawless, K. A., & Brown, S. W. (1997). Multimedia learning environments: Issues of learner control and navigation. *Instructional Science, 25(2),* 117-131.

Pradhan, A.K., Pollatsek, A., Knodler, M., & Fisher, D.L. (2009). Can younger drivers be trained to scan for information that will reduce their risk in roadway traffic scenarios that are hard to identify as hazardous? *Ergonomics, 52(6),* 657-673.

Romoser, M.R.E., & Fisher, D.L. (2009). The effect of active versus passive training strategies on improving older drivers' scanning in intersections. *Human Factors, 51(5),* 652-668.

Romoser, M., Woolf, B.P., Bergeron, D., & Fisher, D.L. (2004). The mental rotation tutors: A flexible, computer-based tutoring model for intelligent problem selection. *Proceedings of the Human Factors and Ergonomics Society's Annual Meeting,* New Orleans, 1025-1029.

Shute, V.J. (1995). SMART evaluation: Cognitive diagnosis, mastery learning & remediation, *Artificial Intelligence in Education Conference (AI-ED) 1995,* 123-130.

Siemer, J. & Angelides, M.C. (1998). Towards an intelligent tutoring system architecture that supports remedial tutoring. *Artificial Intelligence Review, 12,* 469-511.

Sizoneko, R. (2010). The concept of the model for building e-learning system. *Aict2010 – Applied Information and Communication Technologies, Proceedings of the 4th International Scientific Conference,* Jelgava, Latvia, 169-175.

Wang, Y.B., Zhang, W., & Salvendy, G. (2010). Effects of a simulation-based training intervention on novice drivers' hazard handling performance. *Traffic Injury Prevention. 11(1),* 16-24.

Weevers, I., Kuipers, J., Brugman, A.O., Zwiers, J., van Dijk, E., Nijholt, A. (2003). The virtual driving instructor: Creating awareness in a multiagent system. *AI'03 Proceedings of the 16th Canadian Society for Computational Studies of Intelligence Conference on Advances in Artificial Intelligence,* Halifax, Canada, 596-602.

Woolf, B. P., Hart, D., Reece, P., Reid, J, Stillings, N., Bruno, M., and Murray, D. (2002). Expanding a general platform for inquiry learning. *Position paper for CSCW '02 Workshop: Learning Environments for Inquiry Skills.*

Woolf, B. P., and Hall, W. (1995). Multimedia pedagogues: Interactive systems for teaching and learning. *IEEE Computer (Special issue on multimedia), 28(5),* 74-80.