Proceedings of the Fifth IASTED International Conference
WEB-BASED EDUCATION
January 23-25, 2006, Puerto Vallarta, Mexico

CONDUCTING SITUATED LEARNING IN A COLLABORATIVE VIRTUAL ENVIRONMENT

Yongwu Miao*, Niels Pinkwart**, and Ulrich Hoppe*

* Institute for Computer Science and Interactive Systems
The University of Duisburg-Essen, Germany
(miao, hoppe)@collide.info

** Human Computer Interaction Institute
Carnegie Mellon University, USA
nielsp@cs.cmu.edu

ABSTRACT
In the view of situated learning theory, knowledge and understanding is fundamentally a product of a learning situation. Learning is situated and takes place by means of legitimate peripheral participation within the context of a community of practice. Based on the theory of situated learning, we develop conceptual and technical approaches to build a web-based collaborative 3D car-driving simulation environment that requires only low computing and development resources. Rather instructing individuals on a formal, structured, intensive and programmed base, this simulation environment supports members of virtual communities of practice to perform informal, unstructured, spontaneous, collaborative, situated learning.

KEY WORDS
Situated learning, computer simulation, collaborative virtual environment, car driving

1. Introduction

Constructivist conceptions of learning assume that knowledge is individually constructed (radical constructivism) and socially co-constructed (social constructivism) by learners based on their interpretations of experiences in the world. Since knowledge cannot be transmitted but is actively built, instruction should consist of creating situations that provide interpretable experiences. In particular, to train students to perform a certain task, students learn best when they are actively engaged in acquiring and constructing knowledge in a learning-by-doing situation.

However, sometimes the instruction method of learning-by-doing is difficult to apply, because of costs, safety requirements, and resource unavailability. Many researchers and educational practitioners believe that 3D simulation systems offer strong benefits that can support education, especially, facilitate constructivist learning activities [16]. Nowadays, many 3D simulation systems for training purposes have been developed and are in use, allowing students to experience a real life situation in an artificial environment. The use of simulation systems demonstrates several practical advantages and offers enhanced training possibilities [11, 12].

Many educational simulation systems emphasize the importance of physical characteristics to facilitate the training process. For example, a typical car-driving simulator contains all necessary hardware and software modules, an advanced visual and audio system, a motion system, and a fully functional cab with instrumentation to provide a "real-life" driving environment. Furthermore, to achieve authentic traffic conditions, a series of traffic scenarios are elaborately designed and programmed [15]. Because these kinds of simulation systems are very expensive, the training programs involving these simulators are usually formal, structured, and intensive courses.

Today however, for some training purposes, low-cost simulation systems can be developed and used as well, as the performance of personal computers increases. Recently many companies have acknowledged this and have developed low-cost simulation systems for training. For example, a 3D driving school [1] has been developed as a software application running on a single PC with a simple user interface. Although such a low-cost training simulation system clearly has a different focus, the training procedures are similar. An individual trainee interacts with a simulation system through standard, pre-defined driving scenarios. The system or an instructor gives commands and monitors the performance of the trainee. If the trainee has driven through a set of scenarios without mistakes, the system will present to him scenarios with increasing difficulty. If an error is committed, feedback will be presented to the trainee during the lessons, and the trainee may need to drive through the same series of scenarios repeatedly until success.

Our research work described in this paper follows a different educational paradigm. We try to provide a collaborative virtual simulation environment for multiple users to perform an informal, unstructured, spontaneous, collaborative, situated learning. The first attempt in this direction has been described in [9], which focuses on discussing a "high-dimensional data object" method to represent situated knowledge. This paper focuses on presenting technical approaches to a web-based, low-cost 3D collaborative virtual simulation environment for car driving. Rather than adopting a "drill and practice" approach for teaching specific driving skills, our environment fosters users to learn traffic rules and basic knowledge about car-driving in virtual communities of practice.

2. Requirements Derived from Situated Learning Theory

Situated learning places the learner in the centre of an instructional process consisting of *content* - the facts and processes of the task; *context* - the situations, values, beliefs, and environmental cues by which the learner gains and masters content; *community* - the group with which the learner will create and negotiate meaning of the situation; and *participation* - the process by which learners working together and with experts in a social organization solve problems related to everyday life circumstances [3, 6, 13]. It implies that situated learning in a virtual learning environment should integrate these four elements: content, context, community, and participation. In this section, we discuss the first two elements in subsection 2.1 and the other two elements in the remaining two subsections, respectively.

2.1 Supporting Contextualised Learning

There is an increased emphasis on practice and teaching of the field in regard to learner experience, learner control and learner interpretation as well as an emphasis on authentic environmental and contextual factors and their impact on teaching and learning [10]. Knowledge similarly indexes the situation in which it arises and is used. The embedding circumstances efficiently provide essential parts of its structure and meaning. Learners interact with the world not following preconceived plans but related to the situations encountered. Learning the situatedness of knowledge is as important as learning knowledge itself. An approach to a virtual learning environment should therefore provide an authentic context and embed content in authentic situations

2.2 Supporting Collaborative Learning

Lave and Wenger [7] have proposed situated learning as an analytical perspective on learning. In situated learning, learning takes place within the context of a *community of practice*. Lave and Wenger are reacting to modern-day schooling with its emphasis on academic achievement and its aloofness from activities as they actually take place within a community: 'More importantly, the organization of schooling as an educational form is predicated on claims that knowledge can be decontextualised, ...' [7]. Through involvement in a community, learners interpret, reflect, and form meaning. Community provides the setting for the social interaction needed to engage in dialogue with others to see various and diverse perspectives on any issue [2, 7]. Community is the joining of practice with analysis and reflection to share the tacit understandings and to create shared knowledge from the experiences among participants in a learning opportunity. An approach to a virtual learning environment should provide shared activity spaces for members of the community of practice to interact with each other and to socially co-construct knowledge.

2.3 Supporting Participation and Scaffolding

Participation is a process by which learners working together with peers and with experts in a social organization solve problems related to everyday life circumstances [3, 6, 13]. While working on these tasks, a learner may receive instruction as needed to complete tasks or understand aspects of the domain knowledge. Scaffolding allows students to perform tasks that would normally be slightly beyond their ability without that assistance from the teacher. An approach to a virtual learning environment should focus on creating authentic activity and on posing problems to the students to solve, including the necessary scaffolding and instruction to let the student engage the problem as early as possible.

3. Technical Approach to a Collaborative Virtual Environment for Situated Learning

Having outlined the pedagogical background of situated learning theory and its general implications on the design of collaborative simulation environments, this section presents a concrete collaborative car driving simulator that takes into account these considerations. In particular, we set the focus on the aspects of "content and context" and "social interaction". On the technical level, these are partially in conflict with each other: a realistic content and context requires a rich data model, while in turn the desired fast interaction with peers in a distributed system demands a reasonably small amount of data to transmit in order to reduce the network load.

3.1 Embedding Content in Situations

The minimum requirements needed for a low-cost but still "authentic" collaborative car driving simulator can be subdivided into two categories: static objects (e.g., houses or trees) and dynamic objects (other cars, pedestrians, etc.). The model of a virtual driving place that we employed consists of a grid of small cells, each cell containing a list of objects of different types (static and dynamic) together with some generic parameters (position of the cell in the grid, etc.). This abstract representation format was chosen for two reasons:

- Extensibility – it is easy to incorporate new object types (e.g., bicycles) into the architecture

- Communication – the subdivision of the whole place into smaller cells can be used to reduce the data that needs to be transmitted over the net (cf. 3.2)

To facilitate the development of different driving places, a simple grid editor has been developed which allows an easy and fast construction of basic grids. This editor is shown in Figure 1.

Figure 1. The map editor

Maps created with this tool can contain static objects like houses, signs, traffic lights and trees. System internally, these maps are transformed to a VRML representation, which is then used by a Java 3D engine to finally show the driving place to the learner.

Apart from these rather static elements, two different aspects of dynamicity are to be considered. First, traffic lights change their light over time. As such a change is happening on a recurring base, this was adopted using a time based control component acting on the abstract data model of the driving place. For the cars, a predefined abstract data model together with its transformation to VRML representation was employed. The parameters incorporated in this model were: direction and speed of the car, turning angle, acceleration/braking information (needed for the break lights), indicator status, and sector information.

Like in a realistic car, the user can manipulate certain of these parameters directly (e.g., braking information or indicator status) and others indirectly (like speed or position information). The simulation updates the car data model frequently according to the user actions and updates the central data storage (cf. next subsection) as well as the local view.

Conceptually, the described static and dynamic objects in the driving place constitute the "situated knowledge" embedded in a concrete situation in a road network that is used by multiple cars. We have chosen Jess [4], a rule-based logic programming system that bases on inferencing mechanisms and the Rete algorithm, as a technical platform to encode these objects. The advantage of Jess as opposed to other solutions like Prolog is its elaborated interface to Java, allowing for a smooth transition between the different (graphical and logical) representations.

While the facts used within the Jess engine are immediate consequences of the atomic objects and their properties, the production rules serve different purposes:

- A number of rules serve the purpose of *situation* recognition. Here, location-bound situations never change their position (e.g., a curve), while non-location-bound situations (e.g., a car approaching another car) can happen anywhere. These situations serve as the key for all the other rules and also for the feedback mechanism, as they represent content and context of a task the student has to solve. E.g., speeding in a curve is associated to the "curve" situation, and the safe distance is related to the "approaching a car" situation. Situations can be nested and concurrent, as it is of course possible that a car is, e.g., approaching another car and a crossing.
- *Target* facts and rules describe how the outcome of specific situations should be. For instance, they encode that the indicator should be used in the "crossing with turning" situation. If a target state is reached, this can be noted in the user model to represent the learner's growing ability to deal with the related situation.
- Finally, *control rules* are used to represent whether the student has actually reached any of his targets or not (in either case, different consequences apply).

Figure 2 shows a simplified version of a situation for the violation of the safe distance.

```
(defrule safe_distance_violation
  (vpcar (position ?pos)
  (direction ?dir)
  (speed ?speed))
(car_in_lane
  (car_position ?carpos)
  (car_direction ?cardir)
  (car_speed ?carspeed))
(not (target_state
  (desc safe_distance_violation)))
(test (violated_safe_distance
   ?pos ?speed ?carpos ?carspeed))
=>
(bind ?list (create$ "distance"))
  (?*guidance* addInstruction 6 ?carpos ?list
   ?pos)
(assert (target_state
  (situid 6)
  (checkpoint ?carpos)
  (chkpt_passed FALSE)
  (targets ?list)
  (desc safe_distance_violation)))
(?*guidance* addMistakes ?list 6))
```

Figure 2. A Jess rule example

The condition part of the situation states that this situation has not been recognized yet and that the distance to a car driving in front is too small compared to the speed. The result part of the rule builds a target state (which is input for target rules) and creates a list of instructions.

3.2 Enabling Social Interaction

While the architecture as described up to here essentially enables single users to conduct driving simulations, an important part (as identified in section 1 and 2) for adopting the situated learning approach is collaboration support. This can naturally be done by allowing multiple users to simultaneously drive on a shared driving space. There are several groupware libraries that can support this task. We have chosen the TSpace system [8], a tuplespace based solution, due to its proximity to the Jess based fact/rule architecture (tuples can simply be conceived as fact lists). Figure 3 illustrates how the students and the teacher, each of them on their local PCs, interact with the remote space. It is visible that, e.g., the manipulation of traffic lights is done on a dedicated teacher PC, whereas the collision detection (as a specific case of a situation) is done in a distributed manner. The "moving car" tuples are created by the student's actions (or by artificial driver agents on the teacher machine) – furthermore, the tuple space also serves statistical purposes in error counting and thereby updating the user models.

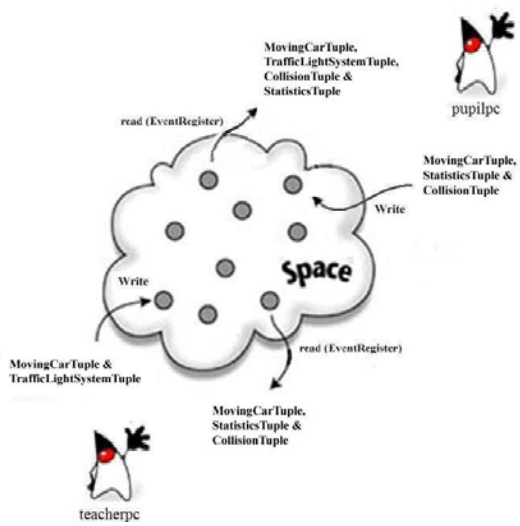

Figure 3. Interactions with the tuplespace

Apart from this conceptual interaction design, an important issue to consider in this real-time collaborative simulation is of course time. A real social interaction in the sense of multiple learners using the system together, each with their own car, could be hindered by insufficient solutions in which, e.g., the positions of cars on different computers differ. Here, we have made use of sector arithmetic. As the grid is made up of single cells, it is only necessary for an application to know the states of his cell and the states of its eight surrounding cells (dark-gray area in Figure 4). This is of course specific to the requirements of the driving simulator: situations which have factors spanning over great distances (i.e., three or more cells) do not occur here.

With this approach, the network traffic is reduced, as only the relevant context is received by an application. Sector changes can be easily transmitted to the tuple space and result in a subscription for different "fields of interest".

To furthermore reduce the network traffic, we designed the system in a way that only car status changes (braking, turning, accelerating, indicator status) as opposed to raw position changes are transmitted, and the peer clients extrapolate the exact positions of cars. Test runs yielded that this approach allowed for both a smooth interaction also in slower networks, and also for good congruence actual position and remote extrapolations.

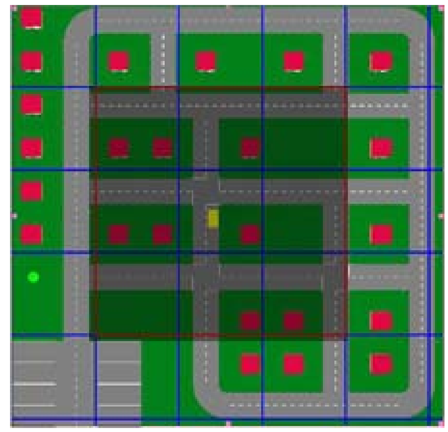

Figure 4. Neighbour cells of a car

3.3 Providing Situated Guidance

The educationally oriented collaborative car driving simulator as described before can already be used in a situated learning approach. Users of the system can drive a virtual car in virtual driving places in a way similar to driving in the real world. They will encounter unexpected and rich situations and meanwhile they are involved in creating training scenarios for others. They can create audio connections with the other users who can be seen in the virtual driving places. They can also discuss the situations in which they are involved and analyse the responsibilities and explore solutions if they have problems (e.g., collision). If a teacher is available, a learner can ask for help and the teacher can "sit" in his/her car and coach him/her in the same way as they are in the real world. Otherwise, the user can require guidance service.

The Jess based representation invites guidance (or tutoring) add-ons. To allow for personalized feedback, these rely on a situation-based user model, which captures the user's performance records concerning his dealing with situations. Whenever a situation is encountered, the system will examine whether the user has knowledge and skills to handle this situation by investigating the statistic records indexed by this situation. According to the user's preference information, the system can provide guidance in an appropriate mode and an appropriate form of media (e.g., textual, verbal, and augmented graphics). Three modes of providing guidance are currently foreseen:

- Forewarn or hint: if a situation occurs that the user can not correctly deal with, the system actively hints the user what he should take care of and how to behave.

- Simple feedback: the system monitors the performance of the students. If any error is committed, the system will show the feedback to the user and update the user statistic date.
- Guidance on demand: a user, who may have little experience in a situation, can ask for help on demand by clicking a key. Then, the system will provide guidance.

4. Prototype System

As described in the previous sections, our design of the 3D simulation system for car-driving training is guided by the theory of situated learning. In order to demonstrate the feasibility of our approach, we have implemented a prototype system. This prototype is completely in line with the research goals and theoretical framework, but (due to resource limitations), has some restrictions. For example, the driving place model and the car model are graphically rather simple. Furthermore, the number of the modelled situations is currently limited (though, of course, the system is open to additions here). The user can currently act by keyboard input only – more advanced steering options are not implemented yet. In addition, the level of detail in the graphics and the rendering quality can be improved. However, the main purpose of the system is fulfilled: we try to support learning-by-playing-games. The users not only entertain with driving in a collaborative virtual environment, but also learn driving knowledge and apply knowledge in virtual communities of practice.

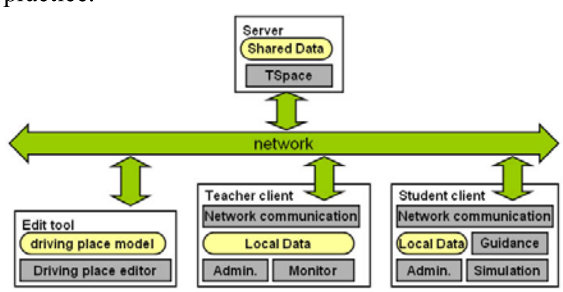

Figure 5. System Modules and Components

Figure 5 illustrates the general software architecture we employed. The figure contains the communication modules (white), the main software components (grey) and the data components (yellow) of the prototype system. There are four communication modules: the map editing tool, the teacher client, the student client, and the server. These modules communicate in a client-server architecture via a network (green). Driving place models can be edited and saved in the database. Teachers can set up a collaborative virtual environment by creating a session and loading a desired driving place model. Each student then registers and joins the session by using an administration component. The student client will then connect with the groupware server via a communication component. After creating a connection, the student client will get data updates from the server. Then the student can see the view of the driving place from the perspective of his/her car. The student controls the virtual car by input actions via the keyboard; each user action is handled by the simulation component. The state of the car in the local data will be changed and communicated to the server. The latter broadcasts changes to those student clients that contain cars which are near to the student's car in the shared driving place (cf. section 3.2). The view of the driving place with the changed state of car is then rendered by the simulation component in each student client. The teacher client has more options: the teacher can monitor all student cars in the driving place. On the teacher client, it is possible to find any student car and couple the view with the view of the corresponding student. Currently, audio connections between clients have not been implemented. This is planned, as such a function will be very useful, allowing learners to communicate when they are in a situation that involves multiple cars. We are also planning to relate multiple users to a single virtual car (then, the users within this car can permanently communicate). In such a "shared car" setting, users can have different roles (driver, coach, and observer) with different rights to control the car. When a student violates traffic rules, the system will provide feedback in a textual form (see Figure 6). The other forms of guidance as outlined in section 3.3 have not been implemented in the current version of the prototype yet.

Figure 6. Screenshot of a View with Guidance

Although the system is still under development, the developed functions have demonstrated the feasibility of our approach. We have tested the system with multiple clients - technically, the results were promising. The interaction worked smoothly, and also the guidance system was fully functional (see Figure 6).

5. Conclusion and Future Work

Learning to drive a car in a real world is an unsafe process. Training via a fully functional car-driving simulator can be helpful, but is usually associated to high costs for most learners. Furthermore, a lot of conventional driving simulators either give no guidance at all, or adopt a “drill and practice” approach. A PC-based car-driving simulation system makes it possible for most users to learn driving knowledge anytime anywhere. Because of the limitation of the PC power and the lack of some devices like motion equipments and large 360 degree screens, it is unrealistic to train all relevant skills in the low-cost simulation systems. However, if we focus on training some important aspects of car-driving such as recognising situations, learning traffic rules, the low-cost simulation systems are useful as well. According to [14], a considerable part of training tasks (about 60%) can be done with a low-cost simulator.

Guided by situated learning theory, we recognised that at minimum, the following three requirements should be taken into account while developing a simulation system: supporting contextualised learning, supporting collaborative learning, and supporting participation and scaffolding. In the design of our low-cost 3D simulation system, we included technical and conceptual approaches to meet these three requirements by embedding content in situations, enabling social interactions, and providing situated guidance, respectively. A prototype system has been developed by adopting these approaches. Although the prototype is still under development, the current version of the system has demonstrated the feasibility of our approaches. We have tested the system in a multiple users setting and found that the system works well and the guidance service can provide situated instructions timely and correctly.

We will continue our work on the system development and implement more functions in the near future. In particular, we will develop automatic agents which can create appropriate scenarios for learners. More situations and alternate forms of guidance will be designed and implemented. Furthermore, an audio communication mechanism with flexible channel controls is planned to be integrated into the system. With these additions, serious evaluations will be conducted.

Acknowledgements

Thanks to Hasna Boussairi, Michael Nahhas, Sebastian Ruiling, Oliver Schilbach, Tobias Schloesser, and Sabine Zill for their development of the prototype systems and the provision of materials.

References

[1] 3D-Fahrschule home page: retrieved from http://www.3d-fahrschule.de/

[2] A.L. Brown, "The Advancement of Learning." *Educational Researcher 23, no. 8* (November 1994): 4-12. (EJ 495 748)

[3] J.S. Brown, A. Collins, and P. Duguid, "Situated Cognition and the Culture of Learning." *Educational Researcher Vol. 18, no. 1* (1989): 32-41. (EJ 386 603)

[4] E. Friedman-Hill: Jess, The Rule Engine for the Java Platform. http://herzberg.ca.sandia.gov/jess, Sandia National Laboratories, Livermore, CA, Version 6.1p8, 23. March 2005.

[5] D. Kirschner, and J. Whitson, eds. *Situated Cognition: Social, Semiotic, and Psychological Perspectives*. (Mahwah, NJ: Lawrence Erlbaum Associates, 1997).

[6] J. Lave, *Cognition in Practice*. New York: Cambridge, 1988.

[7] J. Lave, and E. Wenger, *Situated Learning*. New York: Cambridge, 1991.

[8] T. Lehman, S. McLaughry, P. Wyckoff (1999) TSpaces: The Next Wave, *Proc. Hawaii International Conference on System Sciences (HICSS-32)*

[9] Y. Miao (2004). Supporting Situated Learning for Virtual Communities of Practice: Representation and Management of Situated Knowledge. *Proc. of the 4th IEEE International Conference on Advanced Learning Technologies*, pp. 490-494, Joensuu, Finland.

[10] R. Richey, Agenda-Building and its implications for theory construction in instructional technology. *Educational Technology, 37(1),* 34-41. 1997.

[11] J.C.G.M. van Rooij. Research on simulator-based training and instruction strategies, *Proc. of the HFES Europe Chapter annual Meeting,* Dortmund, Germany, 1994, 73-81.

[12] W. Schneider, Training High-performance Skills: Fallacies and Guidelines., *Human Factors, 27,* 285-300.

[13] I. Shor, *Critical Teaching and Every Day Life*. Chicago: University of Chicago Press, 1987.

[14] W. van Winsum, and J.E. Korteling, *Low-cost simulators 3D: Task analysis for driving simulation.* (Report No. TM-98-A041). Soesterberg, The Netherlands: TNO-Human Factors. 1998.

[15] P. Wolffelaar, S. Bayarri, and I. Coma, Script-based definition of complex scenarios. *Proc. of the 4th Driving Simulation Conference,* Paris, France, July 7-8, 1999.

[16] C. Youngblut, Educational Uses of Virtual Reality Technology, Institute for Defense Analyses: (1998) http://www.hitl.washington.edu/scivw/youngblut-edvr/D2128.pdf