Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 00 (2013) 000–000

Procedia Social and Behavioral Sciences

www.elsevier.com/locate/procedia

4th International Conference on New Horizons in Education

Developing an Intelligent Tutoring System for Vehicle Dynamics

Carlos Huertas*, Reyes Juárez-Ramírez

Autonomous University of Baja California, Calzada Universidad 14418, Tijuana 22390, Mexico.

Abstract

Intelligent Tutoring Systems (ITS) represent a substantial portion of knowledge from human tutors, therefore they are able to provide a better learning guidance than Computer Assisted Instructions (CAI). One of the most important goals of an ITS is to provide feedback tailored to the learner unique needs. There are several educational activities that are supported by these systems such as: problem solving, example review, educational games, etc. This research focus on interactive simulation to improve the learning curve of vehicle dynamics topics as it has been shown that engineer students usually find math and physics too boring and hard to understand when they are unable to map a topic to a real life problem, therefore a most interactive approach seeks to catch student attention while providing a sandbox to experiment and learn. In this paper we present our first prototype, discuss its design and provide test results based on a two year evaluation process.

Selection and peer-review under responsibility of The Association of Science, Education and Technology-TASET, Sakarya Universitesi, Turkey.

Keywords: Intelligent tutoring systems; Interactive learning environments; simulation; applications in mechanical engineering;

1. Introduction

The Computer Assisted Instruction (CAI) were introduced by Patrick Suppes at Stanford University since early 1960, these systems have evolved as a learning tool providing students with encoded set of exercises and associated solutions. CAI system usually provides problems with a single correct answer, hence here is where the problem arises as more complex scenarios cannot be completely evaluated, as is the case of Vehicle dynamics topics where a single problem may have a lot of potential correct answers, and since we talk about nonlinear problems, it is usually required to perform the whole process to find out if the answer is correct or not. From several researches (Alexander, 1999; Leung, 2003; Jain & Getis, 2003) it has been revealed that there is indeed a difference in students achievements between a CAI assisted group and a conventional group, these differences are

* Carlos Huertas. Tel.:+52-664-199-27-28.
E-mail address: nxgtrsim@gmail.com

mostly positive as shown by Owens and Waxman (1994), Teh and Fraser (1995), Yalcinalp, Geban and Ozkan (1995) and Leung (2003). Even when it has been show the effectiveness of CAI systems, there is still a gap to be filled for more complex topics, and there is where the Intelligent Tutor Systems (ITS) proved to match and even surpass CAI system performance as reviewed by Glickman and Dixon (2002).

Among goals in ITS research are how to create computer based tutors more flexible, autonomous and adaptive for particular student needs. There are three types of knowledge that any tutor, being human or artificial needs to have in order to aid student learning (Conati, 2009):

- About the target instructional domain
- About the student
- About the relevant pedagogical strategies

Besides, an ITS needs to have communication knowledge in order to present the required useful information via the computer and the available channels. These different types of knowledge contribute to define the body of an ITS that is integrated basically by four main models according to a general consensus research (Nwana, 1990; Freedman, 2000; Nkambou et al., 2010):

- Domain model
- Student model
- Tutor model
- User interface model

The *domain model*, also called expert model, it is usually built on a theory developed by John Robert Anderson from Carnegie Mellon University that is called Adaptive Control of Thought—Rational (ACT-R), this theory tries to take into account all the possible steps required to solve a problem, this model has the knowledge to evaluate student performance and detect errors. The *student mode*l can be seen as overlay on the domain model, and it is usually considered a core element in a ITS, this model is responsible of tracking the student progress and raise flags or warnings if the student deviates from the domain model. The *tutor model* is the glue that holds the domain and student models together and is responsible of making decisions about adequate strategies and actions for better learning process. It is required to take actions when the student models set deviated flags (Anderson, H. & Koedinger, M). The *user interface model* finally integrates the required information for a successful communication, such as: knowledge about patterns of interpretation, domain knowledge needed for communicating content and knowledge needed for communicating intent (Padayachee, 2002).

Intelligent Tutoring Systems (ITS) may have discrepancies in their architectures, however, they are still much the same as any other instructional design process, Corbett et al., summarizes ITS design and development in four iterative stages:

- Needs assessment
- Cognitive task analysis
- Initial tutor implementation
- Evaluation

The *needs assessment*, is a common stage of an instructional design process, this analyses the learner based on experts and teacher experience, therefore this is the first step to develop the domain model (Corbett et al., 1997). The *cognitive task analysis* involves the detailed approach for developing a computational model for problem solving, observation of human tutor interaction with students provides invaluable information to develop this stage (Farhana et al., 2002). The *initial tutor implementation* involves setting up a problem solving environment following for a series of evaluation activities, which according to Corbett et al. (1997) are very similar to any other software development. The fourth and final stage called *evaluation* perform studies to find out the usability and impact of the ITS, finding learning rates and achievements level (Corbett et al., 1997).

Our current research focus on the domain model, in this paper we present the framework that is being developed to support the first model and that will serve as the expert system for problems solving and will determine the correctness of student results, this is a very important and vital part of the system, because as we can learn from Conati (2009) researches, one main key difference between a CAI and ITS is that the latter, needs to be able to generate real-time solutions not previously defined by a human tutor. The remaining of this paper goes as follows: In section 2, we provide details for the domain model design. In section 3, the current work status of development is discussed. In section 4, an evaluation of domain model accuracy is compared with real life vehicle dynamics scenarios. In section 5, we provide statistical data about current prototype usage. In section 6, we present our conclusions and future work.

2. Domain Model Design

A physics engine is a computer software that provides an approximate simulation of certain physical systems, such as rigid body dynamics and it is the core element in our domain model as our ITS will focus on vehicle dynamics, specifically evaluating vehicle parameters to output performance characteristics under testing conditions, theoretically an endless amount of problem generation can be provided for students as long as the system is capable of process real-time unassisted results on its own.

The model is constructed mainly by three modules:

- Engine Model
- Driveline Model
- Dynamics Model

The specific details of each module are discussed below.

2.1. Engine Model

In order to be able to estimate vehicle performance, it is important to determine how this behavior can be computed. The maximum achievable acceleration of a given vehicle is limited by two main factors, (1)Maximum torque at driving wheels, and (2) Maximum traction force at wheels.

The common vehicle engine to be modeled in this research are internal combustion, and it maximum attainable power *Pe* is a function of engine angular velocity *We*, this need to be determined experimentally by the student or, the system can provide an expected torque curve for all operative Revolutions per Minute (RPM) using a third order polynomial function, as can be seen in (1):

$$P_e = \sum_{i=1}^{3} P_i W_e^i \tag{1}$$

With this approach the students can develop their own formula for determining engine torque curve or simplify the problems using the provided polynomial approach.

The output of this computation is then feed into the driveline module for further analysis as explained below.

2.2. Driveline Model

We use the driveline term to represent the whole system responsible to transfer the torque being generated by the engine(1) to the tires(7), among the components being modeled we have: clutch(2), gearbox(3), differential(5), axles(6) and shaft (4), since the power needs to be transferred from system to system, the power at wheels is always less than the power at engine, this is called driveline efficiency and opens a lot of possibilities for problem solving. The overview of the vehicle power transfer can be seen in Fig 1.

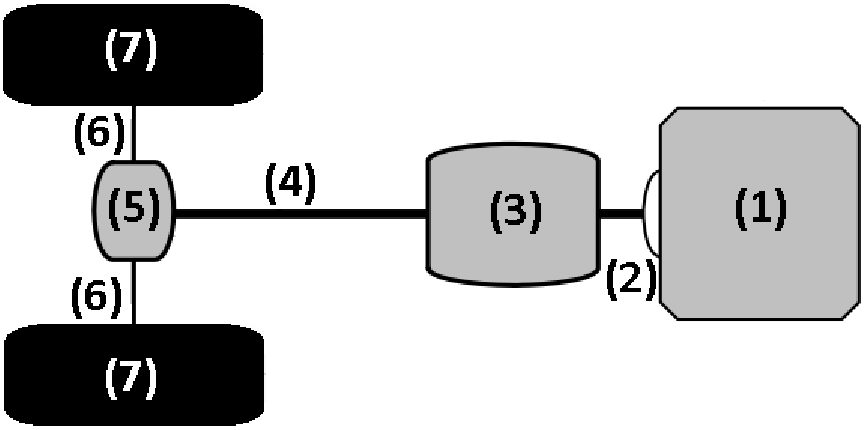

Fig. 1. Vehicle power transfer overview

To find the power at wheels student can develop their own model based on the power laws or run experimental tests varying the whole driveline losses to match specific vehicle characteristics, then the torque finally reaching the wheels can be calculated as in (2):

$$T_w = \eta T_e \tag{2}$$

Where the Greek letter Eta represents the overall system efficiency applied to the engine torque *Te* to get the resultant torque at wheels *Tw*. Once we find out the force moving the tires we feed the dynamics model for final analysis.

2.3. Dynamics Model

This module is responsible for integrating all the vehicle modeling to find out the maximum possible acceleration that can be reached, the limits of this acceleration can be defined by power limitations, i.e. the engine cannot provide enough power to keep accelerating, or traction limited, i.e., the tires are unable to transfer the power to ground and spins, therefore wasting energy instead of moving the car. To evaluate the required time to reach a desired speed it is necessary to find the traction force *Fx* and integrate. This is being done based on the Newton equation of motion and the resultant integral is presented in (3).

$$m\int_0^s \frac{1}{F_X - F_R} dv_X \tag{3}$$

With this motion integral with limits from a standstill 0 up to the goal speed *s*, we consider the vehicle mass *m*, and the traction force *Fx* minus the total resistance force *Fr* which is caused by several sources but being the main source the air drag forces, these forces are evaluated considering different parameters of vehicle body structure that students can design on their own. The dynamics model is far more complex however, more in deep analysis would fall out of the scope of this paper, all these models were developed as part of the first stage of the ITS, in the following section more details about development are provided.

3. ITS Development Status

The first challenge in ITS development was to determine what could be the most adequate development framework to use, since vehicle dynamics modeling requires intensive calculations, framework performance was a major decision point and, since a server controlled internet application would give us much more control and feedback about the ITS usage, we decided to develop a Rich Internet Application (RIA), to determine which framework would provide enough performance a series of frameworks were evaluated with a Bubble render animation benchmark developed by Alexey Gavrilov (2009). In the following Table 1. we provide our test results.

Table 1. Frameworks Benchmark Results

Framework Name	FPS
DHTML JavaScript	115
Silverlight 3.0	199
Flash Flex	55
Java Swing	144
Adobe Air	56

Based on benchmarking results, it was decided to use Silverlight as the development framework as it proves to be up to 38% faster than the second fastest evaluation, even after considering the Silverlight inability to run on Linux machines, the Windows and Mac market share is over 92% as reported by W3C (2013), these results coupled to the fact that it is a Freeware framework made the decision easier. By the time of this paper, the newest prototype ITS version is being developed in the most recent version of Silverlight 5.1.20125.0, which provides numerous features such as: increased isolated storage, improved HTTP stack latency, improved graphics stack with 3D support, real time audio and 64bit architecture support that open possibilities for improving student learning curve by taking advantage of the new features.

The first public prototype was released on June 2011, followed by a refresh in early 2012 after extensive user feedback and usability analysis, the Graphical User Interface (GUI) was redesigned to improve the *user interface* *model*, in the following Fig. 2 we can see two main screens where the student can interact with the ITS, on top it is shown the vehicle management screen, on bottom the real-time vehicle simulation test.

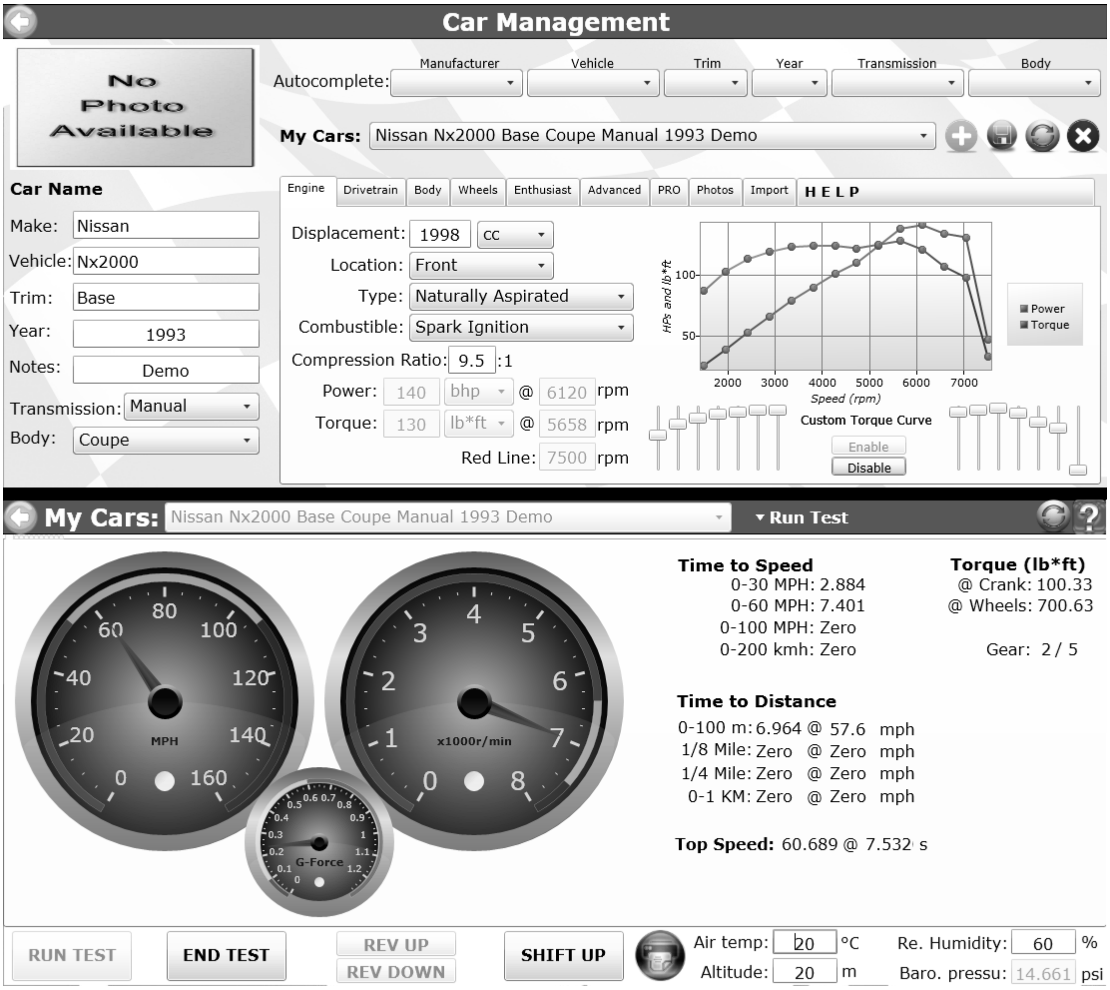

Fig. 2. Main ITS screens for student interaction

On car management screen (Fig. 2, top), the student can design their own vehicle modeling, characteristics are organized in 7 different tabs that provides the student a total of 91 different parameters and configurations to model the vehicle as the problem to be solve demands, while in the real-time test (Fig. 2, bottom), the student can see the effect of different vehicle properties in a non-linear physics problem and therefore confirm or deny his proposed solution for a given problem.

In the following section we show our current experimentation of the physics engine and its accuracy to support the domain model.

4. Domain model accuracy evaluation

As reviewed in introduction, it is important that the tutor, in this case an ITS to have different knowledge, in particular in this section we test the domain knowledge, since an ITS must be able to solve problems in real-time, and this problems do not necessary comes from a pre-defined source or defined by a human tutor as in CAI systems, we need to perform exhaustive testing to prove the ITS capability to solve non-linear problems regarding vehicle dynamics domain. To perform the accuracy comparison, a series of experiments with real cars were taken as the gold benchmark, the source of these experiments were provided by the Road&Track Magazine, the parameters to evaluate were the vehicle capacity to achieve a speed of 60 mph (96.5 km/h) and 100 mph (160.9 km/h) from a standstill, and find out how much time is required to travel a distance of a quarter mile (402.3 meters). The main idea is to see how the physics engine works and how close is to real-life experiments, in Table 2, we provide our accuracy results for a partial list of evaluated cars.

Table 2. Domain Model (Physics engine) accuracy benchmark

Tested Vehicle (Year, Brand, Line, Trim)	Real 0-60	Sim 0-60	Real 0-100	Sim 0-100	Real 1/4	Sim 1/4	Res. Acc
2013 Hyundai Genesis Coupe 2.0T	6.3	5.7	14.8	14.3	14.5	14.4	95%
2012 Mazda Mx5 Miata GT	6.6	6.8	17.8	17.7	15.0	15.2	98%
2013 Subaru Brz Premium	6.6	6.6	17.1	16.7	14.9	15.0	99%
2003 Audi TT 225Q	6.7	6.7	16.4	17.1	14.8	15.2	98%
2004 Chrysler Crossfire Base	6.7	6.5	16.2	16.0	14.9	14.9	99%
2003 Mitsubishi Lancer Evo	4.8	5.0	12.7	12.7	13.4	13.6	98%
2004 Subaru Impreza Sti	4.9	4.8	12.6	12.3	13.3	13.3	99%
2003 Nissan 350z Track	5.8	5.4	14.5	13.5	14.4	14.1	95%
2005 Acura RSX TypeS	6.7	6.9	16.6	17.0	15.0	15.3	98%
2005 Chevrolet Cobalt SS	6.2	6.7	15.9	16.5	14.8	15.2	95%
2006 Honda Civic Si	6.6	6.9	16.3	17.1	15.0	15.3	96%
2006 Volkswagen GTI 2.0T	6.3	6.6	16.9	17.5	14.8	15.2	96%
2007 MazdaSpeed 3 GT	6.2	6.3	16.0	14.1	14.5	14.6	95%
2010 Chevrolet Camaro SS	4.6	4.5	10.5	9.2	12.9	12.6	94%
2009 Dodge Challenger RT	5.8	5.6	13.8	13.4	14.2	14.2	98%
2010 Ford Mustang GT	5.3	5.2	12.7	13.0	13.8	13.8	99%
2006 Mitsubishi Eclipse GT	5.8	6.4	14.4	14.7	14.4	14.7	95%

The average accuracy percentage found the Table 2. was calculated from the average of the individual accuracy of the previous 3 tests, as it can be seen, the physics engine which is powering the domain model has an overall accuracy of over 96.8%, it is important to notice that some differences does not necessary mean a problem in the simulated environment as the Road&Track magazines suffer from result inconsistency, which can vary from different causes, e.g., driver weight and skills which were not modeled on the system. For instance, the same 2007 MazdaSpeed 3 tested on different Road&Track reviews, shows differences of 8% in 0-60 test, 14% in the 0-100 test and 3% in the 1/4 mile test, these inconsistencies are not present in our physics engine which allows to be confident in the results, and even when they differ with real-life tests, we can expect the real-life tests to vary greatly from source to source. The first ITS prototype it is available worldwide for free for basic usage or paid for advanced modeling simulation and can be accessed at *www.nxgtrsim.com*, the specific details about the statistics related to the usage are discussed in next section.

5. ITS prototype statistics data

The first prototype version v11.x was launched in June 15, 2011 and primary goal was to find users to provide feedback about the capabilities for the system, it had a successful response from community and by January 2012, it already had 1340 users from 30 different countries, being the most notable United States with 840 users, Canada 84 and Mexico 64, and over 1000 modeled vehicles. With the provided feedback from users around the world the next prototype version was released in mid January 2012 as version v12.x, in this second prototype the research interest was to get user usage analysis and data about how users interact with the system to be able to have enough information to feed the *student model*, in the Fig. 3 we show the prototype usage by quarters since v12.x up to current version v13.4.19.

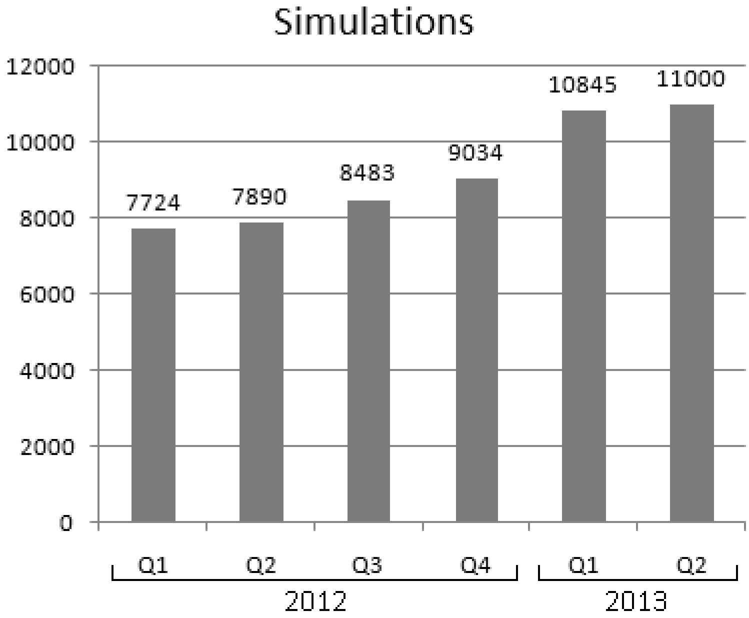

Fig. 3. ITS Prototype 2012-2013 Usage Statistics

As it can be seen in Fig 3., the prototype have had a usage positive increase with a 40.4% increase from 2012 Q1 as compared with 2013 Q1, at the time of this paper, the 2013 Q2 has not finished and the presented data was the result of the current trend. From the sample evaluated we have that users around the world have solved at least 54,976 problems until May 12, 2013 and the numbers keep growing at a ratio of ~800 per week. The countries which are more engaged with the system are United States which usage is over 27,617 simulations which represents the 50.2% of all traffic, in second place we have Canada with 4492 (8.2%), Philippines with 2501 (4.5%), Mexico with 2075 (3.8%), Poland with 1261 (2.3%) and finally United Kingdom with a total of 1162 for a reach of 2.1%.

The data shows that the system has been getting more mature and more people are getting used to the system, in the following section we discuss our conclusions and plans for future work

6. Conclusions and future work

Since the first prototype release v11.6.15 to the actual version v13.4.19, there has been a lot of changes, these changes were only possible by the community feedback, after over 50,000 problem solved with an accuracy over 96% we think the physics engine is stable enough to be integrated in the *domain model.*

As reviewed in introduction, a common ITS is composed by four models, in this paper we have focused in the initial *domain model* which have been completed with the development of the physics engine, and in a smaller scale, the *user interface model* which is still on very early stages as only one model has been completed.

As technology advanced so does the potential problems to be solved which creates a never ending cycle of *domain model* improvements, so we are constantly developing new modeling, however our future plans involves focus more on the next model which would be the *student model*. With all the usage analysis of the past years, we have a very good understanding about how to detect when the user needs help and which methods have result in a more positive engagement experience. The *student* and *tutor models* needs to be develop at the same time as they are usually linked by an action-reaction event.

Our roadmap would be a next stage for *student* and *tutor models*, followed by an updated of the currently *user interface model*, after these two stages we can have a complete ITS out of prototyping stage, we do expect to have some 3D environments for improvement user engagement.

Acknowledgements

I would like to thank to my professor Reyes Juárez-Ramírez for all the experience and support he has provided for this research.

Without any particular order I would like to thank some of the users that have made this possible, with the nicknames: erwin, Ali, My Name, Adam Rod, Hvbert, patrick nyhan, Josh, Mitch (Canada), Bimmer, Bryan, christian fatzinger, Marcus, Corey Young, Jake Bridges, al fresco and all the thousand more that I am unable to thank here.

References

Alexander, M. 1999. "Interactive Computer Software in College Algebra",Retrieved on Dec 20, 2005, from http://archives.math.utk.edu/ICTCM/EP-11/C1/pdf/paper.pdf

Alexey Gavrilov. Bubblemark animation test. 2009. From: http://bubblemark.com/

Anderson, H. & Koedinger, M.. 1997. "Intelligent tutoring goes to school in the Big City". International Journal of Artificial Intelligence in Education, 30-43.

Conati, C., 2009. "Intelligent Tutoring Systems: New Challenges and Directions". National Sciences and Engineering Research Council of Canada (NSERC)

Corbett A. T., Koedinger, K. R., & Anderson, J. R. 1997. "Intelligent tutoring systems: Handbook of human-computer interaction", pp. 849–874. Amsterdam: Elsevier.

Freedman, R. 2000. "What is an intelligent tutoring system? Intelligence", 15–16.

Glickman, C. & Dixon, J. 2002. "Teaching intermediate algebra using reform computer assisted instruction". The International Journal of Computer Algebra in Mathematics Education, vol. 9, no. 1, pp. 75-83

Jain, C. & Getis, A. 2003. "The effectiveness of internet-based instruction: An experiment in physical geography". Journal of Geography in Higher Education, vol. 27, no. 2, pp. 153-167.

Leung, K. 2003. "Evaluating the effectiveness of e-learning". Computer Science Education, vol. 13, no. 2, pp.123-136.

Nkambou, R., Mizoguchi, R., & Bourdeau, J. 2010. "Advances in intelligent tutoring systems". Heidelberg: Springer.
Nwana, H. S. 1990. "Intelligent tutoring systems: An overview. Artificial Intelligence Review", 251-277.

Owens, E. & Waxman, H. 1994. "Comparing the effectiveness of computer-assisted instruction and conventional instruction in mathematics for African-American postsecondary students". International Journal of Instructional Media, vol. 21, no. 4, pp. 327-393.

Padayachee I., 2002. "Intelligent Tutoring Systems: Architecture and Characteristics".

Shah, Farhana; Martha Evens, Joel Michael, Allen Rovick. 2002. "Classifying Student Initiatives and Tutor Responses in Human Keyboard-to-Keyboard Tutoring Sessions". Discourse Processes

Teh, G. & Fraser, B. 1995. "Gender differences in achievement and attitudes among students using computer-assisted instruction". International Journal of Instructional Media, vol. 22, no. 2, pp. 111–120.

W3C Schools OS Platform Statistics and Trends. 2013. from: http://www.w3schools.com/browsers/browsers_os.asp

Yalcinalp, S., Geban, O., & Ozkan, I. 1995. Effectiveness of Using Computer-Assisted Supplementary Instruction for teaching the mole concept. Journal of Researchin Science Teaching, vol. 32, no.10, pp.1083-1095.