The 4th International Scientific Conference
eLearning and Software for Education
Bucharest, April 17 - 18, 2008

# HEURISTIC EVALUATION OF WEB-BASED INTELLIGENT TUTORING SYSTEMS

Ioan I. ANDONE *
Napoleon-Alexandru Al. SIRETEANU**

***Abstract**: Intelligent Tutoring Systems (ITSs) might improve learning and teaching in two distinct ways. Just as they have made teachers and students more productive, they could reduce teaching costs or increase the speed with which learners acquire knowledge/of knowledge acquirement. Alternatively, such systems might help improve the quality of learning, rather than simply making it faster and cheaper. In this paper we first review evidence that heuristic evaluation can improve the efficiency of Web-based Intelligent Tutoring Systems; then we consider how they might lead to better educational outcomes.*

***Keywords**: online learning systems, intelligent tutoring systems, heuristic evaluation.*

## I. INTRODUCTION

Online Learning Systems[1] enable students to use tutoring resources anytime and anywhere. They are often used in the corporate field for knowledge management purposes. Among various possibilities for implementing Online Learning Systems, supported ones are nowadays the most popular. Intelligent Tutoring Systems (ITSs) are, in many respects, very similar to human tutors. Intelligent Tutoring Systems have been around since the late 1970s, but increased in popularity in the 1990s. Based on cognitive science and Artificial Intelligence, these systems have proven their worth in various ways in multiple domains of Education [1-15].

Therefore, a web-based intelligent tutoring system, broadly defined, is educational software containing an artificial intelligence component that provides direct customized instruction or feedback to students, without the intervention of human beings in the online learning process[2]. The software tracks each student's work, tailoring feedback and hints along the way. By collecting information on a particular student's performance, the software can make inferences about strengths and weaknesses, and can suggest additional work. A Web-based ITS may employ a host of different technologies. However, usually such systems are more narrowly conceived of as artificially intelligence systems[3], more specifically expert systems[4] used for tutoring.

The traditional instructional methods present learners with facts and concepts followed by the test questions. These methods are effective in exposing to large amounts of information and testing their recall. However, they often instill "inert knowledge" that learners can recall but may not apply correctly when needed. By contrast, ITSs use simulations and other highly interactive learning environments that require people to apply their knowledge and skills. These active, situated learning environments help them retain and apply knowledge and skills more effectively in operations settings.

---

[1] Any form of educational material which is readily available for distribution on the Web or an internal network.
[2] Electronic learning or eLearning is a general term used to describe computer-enhanced learning.
[3] A computer program that has parts for each of the functions described in the intelligent system document.
[4] An intelligent system which in an interactive setting asks a person for information and, based upon the response, draws conclusions or gives advices.

This paper discusses the ITSs and Web-based ITS's fundamentals, then heuristic evaluation's characteristics, and a new International Standard for improving the efficiency of Web-based ITSs usability.

## II. FUNDAMENTALS OF ITSs AND WEB-BASED ITSs

In order to provide hints, guidance, and instructional feedback to learners, ITSs typically rely on three types of knowledge, organized into separate software modules as shown in Figure 1.

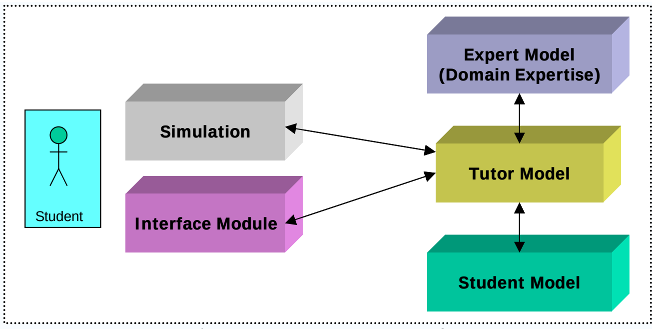



Figure 1 Components of an ITS

The *expert model* represents subject matter expertise or domain expertise and provides the ITS with knowledge of what it's teaching. The *student model* represents what the user does and doesn't have. This knowledge lets the ITS know who it's teaching. The *tutor model* enables the ITS to know how to teach, by encoding instructional strategies used via the *Interface Module*.

Here's how each of these components works. An expert model is a computer representation of domain's expertise and problem-solving ability. This knowledge enables the ITS to compare the learner's actions and selections with those of an expert in order to evaluate what the user does and doesn't know. Some ITSs captures the domain's expertise in rules (expert systems), and enables to generate problems on the matter expertise, combine and apply rules to solve problems, asses each learner's understanding by comparing the expert system's reasoning with theirs, and demonstrate the software's solutions to the participant's. Though this approach yields a powerful tutoring system, developing and expert system that provides comprehensive coverage of the subject material is difficult and expensive.

The student model evaluates each learner's performance to determine the knowledge, perceptual abilities, and reasoning skills or monitor a learner's sequence of actions to infer his or her understanding. The tutor model encodes instructional methods that are appropriate for the target domain and the learner. Based on the knowledge of a person's skill, strengths and weaknesses, participant expertise levels and student learning styles, the tutor model selects the most appropriate instructional intervention. Simulation without a human tutor could be almost useless. ITS is still perceived by many as a technology of the future, but rapid growth of learning software and artificial intelligence is making it a viable option, and Web-based ITSs will soon ad substantial improvements to that popular training medium, because the World Wide Web facilitates co-operative teaching and learning, and an exciting potential for sharing both the learning resources as well as the tasks of creating new learning resources through international collaboration. Since the current web based learning systems cannot adequately provide for interactive learning that is supported by dynamic feedback, it is important to extend the benefits of traditional ITSs to the web based systems through a modular architecture supported by authoring tools and an indexing mechanism for the repository of the various tutoring components.

With the development of WWW technology, web-based ITSs are becoming mainstream area of research and development. The web-based ITSs installed and supported in one place can be used by thousands of learners all over the world. Although many web-based educational systems appeared recently, most of them emerged from their predecessor legacy stand-alone systems. Therefore, they not only restrict themselves in functionality, but also fail to take advantage of distributed nature of WWW [Dinamet, 2005, p.30]. These systems are usually dedicated to a restricted expert system's knowledge base and have closed architecture with little possibility of modifications once released.

Recently a lot of web-based ITSs appeared on the Internet. The famous example include *CALAT* [Nakabayashi et al, 1997], *InterBook* [Brusilovsky et al, 1997], *VC PROLOG* [Peylo et al, 2000], Desire2Learn, and many others [Koedinger, 2006].

Most of these systems used a client/server architecture that places shared resources and functionality on servers, and uses internet to deliver student interfaces on a wide variety of client platforms at any location with internet access. This architecture (see Figure 2) aimed to improve the deployability and interoperability of knowledge-based educational software without sacrificing advanced functionality.

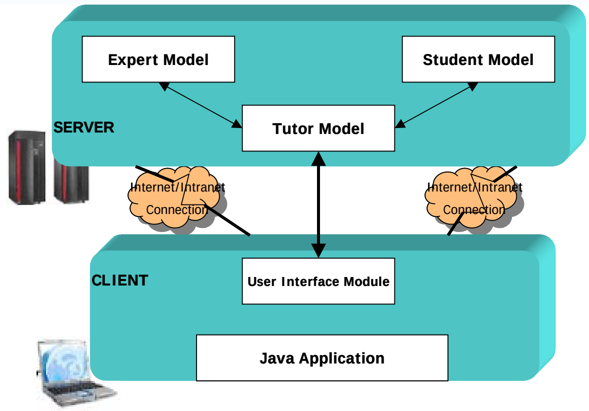



Figure 2 *Architecture of Web-based ITSs*

As we know, the main characteristic of a client/server application is that a part of application (Client) requests the execution of some job, while the other part of the application (Server) executes this job. All the functions of the application must be split into the two programs: the client program, which is executed on the user's computer and the server program, which is executed on the host hardware platform. The client program provides an interaction with the user, the input of the data, and the presentation of the results; it is able to communicate with the server program, it sends the requests and receives the results. The server program, which is the heart of a client/server application, receives requests from the client, executes them and sends the results back to the client, through the Interface module. All the main and computationally expensive operations are implemented in the server program.

In these architectures of the Web-based ITSs, the student model is used where the student's knowledge is assumed to be a subset of the expert's knowledge and the goal of tutoring is to enlarge this subset. When students learn the topic and navigate within the learning material, the system stores the progress in the database. When students log in, system starts where the previous session was stopped. If students finish one unit (lesson), system informs them that the unit is finished and post-test is available. While students take tests, system's inference mechanism determines the knowledge acquired by the student in that particular unit, and stores the results in the database. Both human tutor and students may check the learning progress.

There are still the questions of how theory, strategies, and methods from the learning sciences are applied to the design of online learning systems and how the use and evaluation of those systems inform the learning sciences.

## III. CHARACTERISTICS OF HEURISTIC EVALUATION

According to some definitions, teaching is about learning to know, and training is about learning to do. E-learning has the potential to encompass both. The concern of teaching, by that definition, is efficient information transfer. The methods of training, by contrast, focus on the activation of cognitive and behavioral strategies that allow learners to reproduce not just ideas but operational competence. As such, teaching is fundamentally algorithmic: learn this information and demonstrate understanding in standard applications. Training, however, is fundamentally heuristic (discovery-based): find or build your own best internal strategies and demonstrate behavioral competence in situations that extend beyond the learning environment.

The concept of heuristics has an important and long history, spanning the fields of philosophy, law, psychology, and human-computer interaction among others. We try to define the term, present the alternative usability heuristics, and show a new international standard for improving the efficiency of Web-based ITSs usability.

### 3.1. Definitions and alternative usability heuristics

R. Molich and J. Nielsen argue that the heuristic evaluation is a form of usability inspection where usability specialists judge whether each element of a user interface follows a list of established usability heuristics [Nielsen, 1994]. These authors stated that, heuristic evaluation is a usability engineering method for finding the usability problems in a user interface design so that they can be attended to as part of an iterative design process. Heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics"). Usually two to three analysts evaluate the system with reference to established guidelines or principles, noting down their observations and often ranking them in order of severity. The analysts are usually experts in human-computer interaction (HCI) or user interfaces, but others, less experienced have also been shown to report valid problems [7].

We agree this opinion, which is valuable also for the heuristic evaluation of a Web-based ITS. A heuristic or expert evaluation can be conducted at various stages of the Web-based ITS development lifecycle, although it is preferable to have already performed some form of context analysis to help the experts focus on the circumstances of actual or intended product usage.

This method is to identify usability problems of a Web-based ITS based on established human factors principles. The method will provide recommendations for design improvements. However, as the method relies on experts, the output will naturally emphasize interface module functionality and design rather than the properties of the interaction between an actual user and the Web-based ITS.

David Travis [2007] stated that an usability expert is an experimented evaluator which uses the Jakob Nielsen's ten usability 'heuristics' (see Figure 3), but many of these guidelines are common sense, and they are not based on substantive research. “These heuristics, which are widely used, have never been validated. There is no evidence that by applying these heuristics in the design and development of user interfaces that it will improve the interface.” [Bailay, 1999]

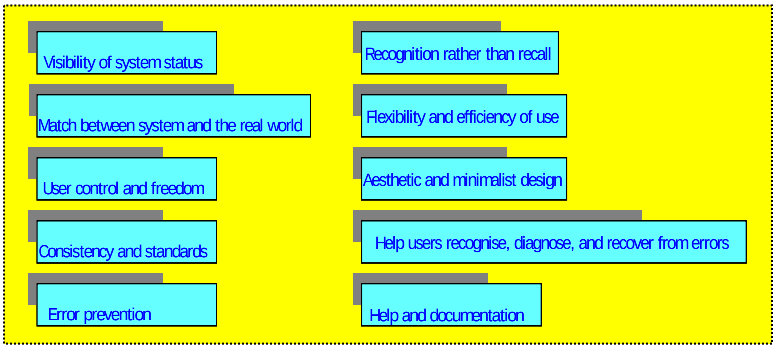



Figure 3 *J.Nielsen’s usability heuristics*

From R. Molich and J. Nielsen’s works, lots of people had derived guidelines and principles for usability but there were often so many guidelines that an expert review could take many days to complete. Nowadays, most usability professionals have their own favorite sets of guidelines or checklists but these suffer from the same problem: they are not research-based and they are often a matter of opinion. Consequently, they are subject to the biases and whims of the reviewer. In our opinion, we need a set of guidelines that are based on research and have some international consensus. For example, Mosier and Smith's Guidelines for Designing User Interface Software has 944 guidelines and remains the largest collection of publicly available user interface guidelines in existence. [Mosier & Smith, 1986]. Because the limitations of Nielsen’s usability heuristics and many collections of user interface guidelines, the International usability standard, BS EN-ISO 9241-110 proposes an alternative set of seven guidelines. These guidelines have the benefit of international consensus and they can be applied to any interactive system for improving the efficiency of usability (Figure 4).

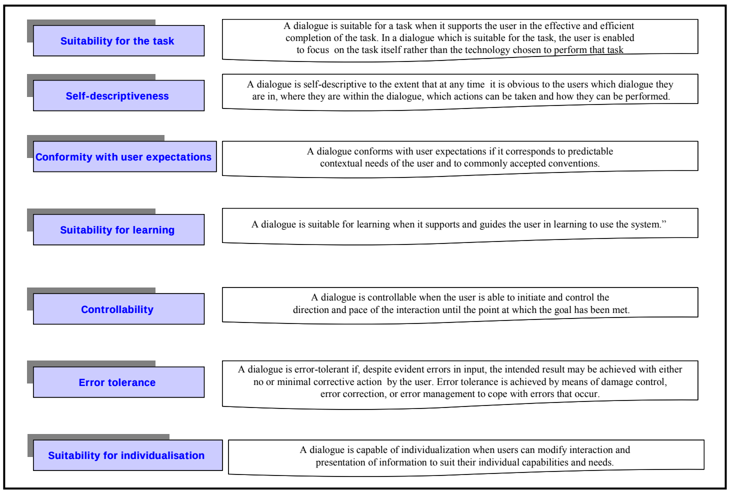



Figure 4 *ISO 9241 usability heuristics*

### 3.2. A New International Standard for Web-based ITSs usability

Most standards activity relevant to the web has emerged from the W3C but they have little to say about usability. ISO are engaged in developing a new standard titled ISO/AWI 23973 "Software ergonomics for World Wide Web user interfaces". ISO has been developing ergonomics standards for over 20 years and one of their sub-committees (SC 4) is responsible for standards in the field of human system interaction.

In our experience, developers are more likely to fix an issue with a user interface when we point to an international standard than when we base our judgment on personal opinion. They may not be as well known, but they do have the authority and credibility of an international standard.

ISO 9241 is this new international standard for web usability and for Web-based ITSs usability also. This new standard will be of interest to anyone who designs, evaluates or commissions web sites for Web-based ITSs and it is likely to have a significant impact in improving the overall usability of the Web applications**.** This new web usability standard hasn’t been approved yet — it is based around a reference model that distinguishes the three domains within which design work is carried out (Figure 5).

ISO uses a reference model for web design comprising design, process and evaluation. The Standard itself is focused on the design domain aspects only. These are: purpose and strategy, content and functionality, navigation and interaction, and presentation and media design. These aspects can be seen as different levels of the overall design process. The other two parts of the model represent the process domain and the evaluation domain. These constitute important aspects for the user-centered development of Web-based ITSs applications but they are not elaborated in this International Standard.

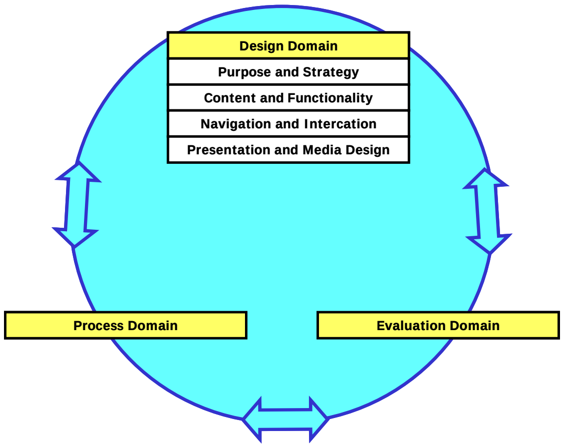



Figure 5 *ISO 9241 Reference Model*

As we see in Figure 5, this emerging standard addresses the design domain only, so this isn't the place to go if you are interested in usability evaluation or design process (although it does contain a handy list of references to these other areas).

- *The process domain.* This domain describes the design process used by the organization, such as the one described in ISO 13407:1999 "Human-centered design processes for interactive systems".
- *The evaluation domain.* This domain contains the tools and techniques used to assess the final design, such as *usability testing*: a method for uncovering the problems that real users experience with the ITS web site under actual conditions of use. The process involves asking participants to carry out realistic tasks and observing where they experience problems.
- *The design domain.* This is the domain within which the designer develops the Web-based ITS User Interface.

### 3.3. Heuristic evaluation with Techsmith's User Vue

In our experience, UserVue makes usability testing straightforward and easy – there's no excuse for not seeing how users use the software or features. UserVue [30] is an online service that lets you remotely observe and record users' desktops as they navigate applications and sites:

- Perform user testing
- Conduct user research
- Collect design feedback
- Collaborate on projects.

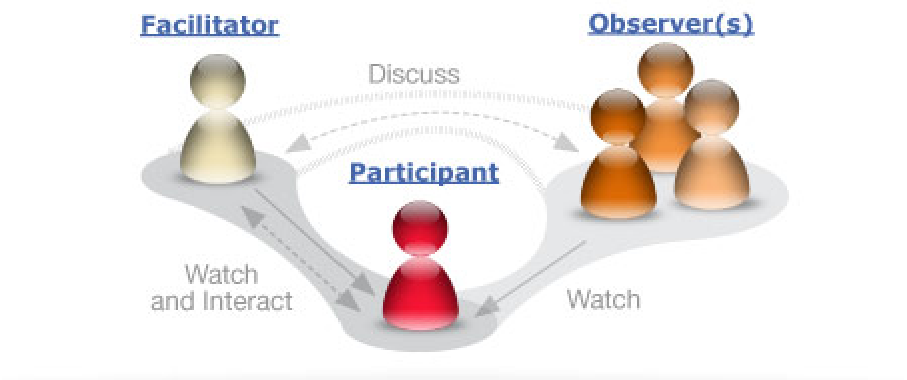
In short, the stages of the method are the following:

*Planning.* The panel of experts must be established in good time for the evaluation. The material and the equipment for the demonstration should also be in place. All analysts need to have sufficient time to become familiar with the product in question along with intended task scenarios. They should operate by an agreed set of evaluative criteria.

*Running.* The experts should be aware of any relevant contextual information relating to the intended user group, tasks and usage of the product. A heuristics briefing can be held to ensure agreement on a relevant set of criteria for the evaluation although this might be omitted if the experts are familiar with the method and operate by a known set of criteria. The experts then work with the system preferably using mock tasks and record their observations as a list of problems. If two or more experts are assessing the system, they should not communicate with one another until the assessment is complete. After the assessment period, the analysts can collate the problem lists and the individual items can be rated for severity and/or safety criticality.

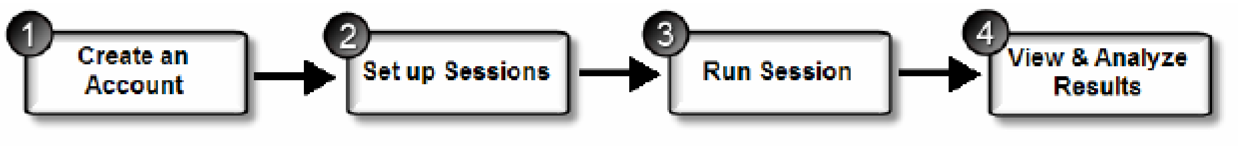
*Reporting.* A list of identified problems, which may be prioritized with regard to severity and/or safety criticality, is produced. In terms of summative output the number of found problems, the estimated proportion of found problems compared to the theoretical total, and the estimated number of new problems expected to be found by including a specified number of new experts in the evaluation can also be provided. A report detailing the identified problems is written and fed back to the development team. The report should clearly define the ranking scheme used if the problem lists have been prioritized.

UserVue allows the remotely connect to, interact with, observe and record users as they navigate through a Web-based ITS application or Web site. The Help system of this software contains help specific to each role in a session—facilitator, participant, and observer (Figure 6).




Source: [30]
Figure 6 *UserVue Architecture*

There are four main steps in UserVue (Figure 7):




Source:[31]
Figure 7 *UserVue steps of work*

A session is a scheduled instance of remote desktop screen sharing run by the facilitator in UserVue. The facilitator is responsible for running the session. The facilitator guides the participant and observer(s) through the process of starting a session. During a session, the facilitator asks the participant to perform tasks while the facilitator records the participant's desktop. The participant's screen is recorded on the facilitator's computer and shared with observers. UserVue allows setting up multiple sessions over a period of time and customize the session settings to meet the testing needs. This process is completed on the UserVue Web site. As we see in Figure 6, UserVue is a Web-based service that allows you to remotely connect to, interact with, and observe users as they navigate an application or Web site. It is secure, easy-to-use, and works through firewalls.

Connecting to a UserVue session is easy for participants because there is nothing for them to install or configure. UserVue simplifies collecting user experience information so anyone can use it to identify design problems and make critical improvements. Whether you are doing remote user testing, validating a design, or gathering qualitative feedback, UserVue gives you accurate user feedback, rapidly and repeatedly. For the cost, UserVue provides great flexibility in promoting the value of user testing to team members and stakeholders. It lets us get feedback to the design team – fast (figure 7).


Figure 7 UserVue's Event Capture

We tried some of UserVue functionalities and we observed that it allows to:

• *Focus on the user's experience.* Unlike Web conferencing tools that share the moderator's screen, UserVue shares the user's screen, providing a truly accurate view of the user experience.
• *View and share results immediately.* UserVue automatically creates a Windows Media Video (.wmv) file of the recording which can be viewed and shared immediately following a session.
• *Invite stakeholders to observe.* Multiple stakeholders can view a UserVue session live and collaborate using chat without disturbing the user.
• *Seamlessly import recordings into Morae.* Import recordings into Morae Manager version 1.3 for easy analysis and sharing, creating a complete solution for user experience research.

## IV. THE BENEFITS OF THE HEURISTICS EVALUATION

The benefits from heuristic evaluation are mainly due to the finding of usability problems, though some continuing education benefits may be realized to the extent that the evaluators increase their understanding of usability by comparing their own evaluation reports with those of other evaluators [Nielsen, 1994]. The benefits of the heuristics evaluation method are:

- There is a general acceptance that the design feedback provided by the method is valid and useful. It can also be obtained early on the design process, whilst checking conformity to established guidelines helps to promote compatibility with similar systems.
- It is beneficial to carry out a heuristic evaluation on early prototypes of Web-based ITS before actual users are brought in to help with further testing.
- Usability problems found are normally restricted to aspects of the interface module that are reasonably easy to demonstrate: use of colors, layout and information structuring, consistency of the terminology, consistency of the interaction mechanisms. It is generally agreed that problems found by inspection methods and by performance measures overlap to some degree, although both approaches will find problems not found by the other.
- The method can seem very critical as designers may only get feedback on the problematic aspects of the interface module as the method is normally not used for the identification of the 'good' aspects.

## V. USABILITY METRICS OF HEURISTIC EVALUATION

Usability metrics are precise, quality measures used to evaluate the system. Their purpose is to produce a Web-based ITS that is neither under- nor over-engineered. Usability metrics are useful, because of their helping for outrun the competition. The specialists recommend collecting usability metrics in the three Es: effectiveness, efficiency and emotion (Table 1).

- *Effectiveness*. The accuracy and completeness with which users achieve specified goals.
- *Efficiency*. The accuracy and completeness of goals achieved in relation to resources.
- *Satisfaction*. Freedom from discomfort and positive attitudes towards the use of the Web-based ITSs.

**Table 1 Usability metrics of Web-based ITS heuristic evaluation**

| **Effectiveness measures** (the accuracy and completeness with which users can achieve their goals) | **Efficiency measures** (the amount of effort users need to put in to achieve their goals) | **Emotional measures** (how users feel about the system) |
|---|---|---|
| Number of power tasks performed | Time to execute a particular set of instructions | Ratio of positive to negative adjectives used to describe the Web-based ITS |
| Percentage of relevant functions used | Time taken on first attempt | Per cent of customers that rate the product as "more satisfying" than a previous product |
| Percentage of tasks completed successfully on first attempt | Time to perform a particular task | Rate of voluntary use |
| Number of persistent errors | Time to perform a particular task after a specified period of time away from the Web-based ITS | Per cent of customers who feel "in control" of the product |
| Number of errors per unit of time | Time to perform task compared to an expert | Customer rating on a 7-point scale anchored with "makes me more/less productive"; |
| Per cent of users able to successfully complete the task | Time to learn the criterion | Per cent of customers who would recommend it to a friend after two hours' use |
| Number of errors made performing specific tasks | Time to achieve expert performance | Per cent of customers that rate the product as "easier to use" than a key competitor |
| Number of requests for assistance accomplishing task | Number of key presses taken to achieve task | - |
| Objective measure of quality of output | Time spent on correcting errors | - |
| - | Number of icons remembered after task completion | - |
| - | Time to install the Web-based ITS | - |
| - | Per cent of time spent using the user guide manual | - |
| - | Time spent relearning functions | - |

Source: [Travis, 2007]

## Conclusions

Designing open, flexible and distributed intelligent tutoring systems for globally diverse learners is challenging. As more and more institutions offer intelligent tutoring systems to students worldwide, we will become more knowledgeable about what works and what does not work. Case-based learning with ITSs is a special learning style, where the learner takes over a certain role in the training case and should act and interact in the training case scenario. The training case is usually provided in a narrative manner. Two cognitive processes that can be trained in the case-based scenario are the process of general knowledge application and the process of diagnostic reasoning. To optimally support the training of these two processes, the ITS should adapt the training case at runtime to the learner's expertise, his knowledge, and his decision about how to proceed in each step.

## BIBLIOGRAPHY


**[1]** Alpert, S.R., Singley, M.K., Fairweather, P.G., (1999), *Deploying Intelligent Titors on the Web: an Architecture and an Example*, International Journal of Artificial Intelligence in Education, 10, 183-197.

**[2]** An example of ITS authoring tool (2008), *Cognitive Tutoring Authoring Tools*, <http://ctat.pact.cs.cmu.edu/>

**[3]** Bailey, B., (1999), *Insights from Human Factors International*, UI Design Newsletter – May.

**[4]** Dinamet, P., (2005), A Web-Based Adaptive Learning System to Assist Learners and Teachers, in Montero, M., Gaudioso, E., Knowledge-Based Virtual Education, Volume 178, Springer Berlin / Heidelberg, pp. 23-48.

**[5]** Discenza, R., Howard, C., Schenk, K.D. (Eds) (2002), *The Design and Management of Effective Distance Learning Programs*, Idea Group Pub., PA.

**[6]** Gerhardt-Powals, J., (1996), *Cognitive engineering principles for enhancing human - computer performance*, International Journal of Human-Computer Interaction, 8(2), 189-211.

**[7]** Graeser, A.C., Person, N.K., Harter, D., (2001), *Teaching tactics and dialog in autotutor*, International Jourmal of Artificial Intelligence in Education, 12, 12-23.

**[8]** Grigoriadou, M., Tsaganou, G., (2007), Authoring Tools for Structuring Text-Based Activities, Lectures Notes in Computer Science, Volume 4556, pp. 319-328.

**[9]** Human-Computer Interaction Resources on the Net:<www.ida.liu.se/~miker/**hci**/>

**[10]** ISO 9241-3:1992/Amd 1:2000 Visual performance and comfort test.

**[11]** ISO 13406-1:1999 Ergonomic requirements for work with visual displays based on flat panels -- Part 1: Introduction.

**[12]** ISO 13406-2:2001 Ergonomic requirements for work with visual displays based on flat panels -- Part 2: Ergonomic requirements for flat panel displays.

**[13]** ISO 9241-3:1992 Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 3: Visual display requirements.

**[14]** Kinshuk, Patel, A., Russell, D., (1999), *Hyper-ITS: A Web-Based Architecture for Evolving and Configurable Learning Environment*, in Staff and Educational Development Journal, 3(3), 265-280.

**[15]** Koedinger, K.R., Corbett, A. (2006), "Cognitive Tutors: Technology bringing learning science to the classroom", in Sawyer, K., *The Cambridge Handbook of the Learning Sciences*, Cambridge University Press, pp. 61-78.

**[16]** Lim, K.Y., Long, J.B., (1995), *The Muse Method for Usability Engineering*, Cambridge Series on Human-Computer Interaction, Cambridge University Press.

**[17]** Morae Usability Testing for Software and Web Sites: http://www.techsmith.com/morae.asp

**[18]** Nielsen, J., (1994), *Usability Engineering*, Morgan Kaufmann Pub., San Francisco.

**[19]** Pecheanu, E., Stefãnescu, D., Istrate, A., (2004), *An Extended Structural Model for Intelligent Instructional Systems*, The Romanian Symposium on Computer Science – ROSYCS 2004, July 2004, Iasi, Scientific Annals of the "A.I. Cuza" University of Iasi, Tome XV, pp. 167-177.

**[20]** Peylo, C., Thelen, T. Et al, (2000), A Web-Based Intelligent Educational System for PROLOG, http://virtcampus.cl-ki.uni-osnabrueck.de/its-2000/paper/peylo/ws2-paper-9.html

**[21]** Psotka, J., Mutter, S.A., (1988). *Intelligent Tutoring Systems: Lessons Learned*. Lawrence Erlbaum Associates.

**[22]** Smith, S. L., Mosier, J.N., (1986), *Guidelines for Designing User Interface Software* (ADA 177-198), Natl Technical Information.

**[23]** Students and tutors work one-to-one, communicating using instant messaging and drawing problems on an interactive whiteboard (2008):<http://www.tutor.com/>

**[24]** The E-Learning Community (2008):<http://www.astd.org/content/education/eLearning/>

**[25]** Thomas, C., Bevan, N., *Usability Context Analysis*:*A Practical Guide,* Version 4.04: <http://www.usabilitynet.org/papers/UCA_V4.04.doc>

**[26]** Travis, D., (2007), Bluffers' Guide to ISO 9241, Downloadable eBook, UserFocus, London, www.userfocus.co.uk

**[27]** Truemper, K., (2007), *Design of Logic-based Intelligent Systems*: <http://www.iasi.cnr.it/DLIS/iasi_lecture_1_04.pdf>

**[28]** UserVue Website:https://uservue.techsmith.com

**[29]** UserVue Online Help: http://download.techsmith.com/uservue/docs/onlinehelp/uservue_help.pdf

**[30]** WebCT: <http://www.webCT.com/>

**[31]** Woolf, B.P., Beck, J., Eliot, C., Stern, M., (2001), *Growth and maturity of intelligent tutoring systems: A status report*, In Forbus, K.D., Feltovich, P.J., (Eds), Smart machines in education, MIT Press, Cambridge, pp.100-144.

**[32]** Inquestra Learning Tool:<http://www.inquestra.com>

---

* *„Alexandru Ioan Cuza" University, Dept.of Business Information Systems, 22, CAROL I Boulevard, 700505, Iasi, ROMANIA, Phone: +40-232-201413*

** *„Alexandru Ioan Cuza" University, Dept.of Business Information Systems, 22, CAROL I Boulevard, 700505, Iasi, ROMANIA, Phone: +40-232-201413*