Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers & Education 44 (2005) 53–68

COMPUTERS & EDUCATION

www.elsevier.com/locate/compedu

Mobile educational features in authoring tools for personalised tutoring

Maria Virvou *, Eythimios Alepis

Department of Informatics, University of Piraeus, 80, Karaoli and Dimitriou Street, Piraeus 18534, Greece

Received 5 August 2003; accepted 9 December 2003

Abstract

One important field where mobile technology can make significant contributions is Education. In the fast pace of modern life, students and instructors would appreciate using constructively some spare time that they may have, in order to work on lessons at any place, even when away from offices, classrooms and labs where computers are usually located. In this paper, we describe a mobile authoring tool that we have developed and is called Mobile Author. Mobile Author can be used by human instructors either from a computer or a mobile phone to create their own Intelligent Tutoring Systems (ITSs) and to distribute them to their students. After the ITSs have been created, students can also use any computer or mobile phone to have access to theory and tests. The tutoring systems can assess the students' performance, inform the databases that record the students' progress and provide advice adapted to the needs of individual students. Finally, instructors can monitor their students' progress and communicate with their students during the course. The mobile features of both the authoring tool itself and the resulting ITSs from it have been evaluated by instructors and students, respectively. The results of the evaluation showed that mobile features are indeed considered useful.
© 2004 Elsevier Ltd. All rights reserved.

Keywords: Authoring tools and methods; Distance education and telelearning; Intelligent tutoring systems; Teaching/learning strategies

1. Introduction

There are many virtues of web-based educational software, which have been recognised by educators and educational institutions. Some important assets include platform-independence and

* Corresponding author. Tel.: +30-1-414-22-69; fax: +30-210-4142264.
E-mail addresses: mvirvou@unipi.gr (M. Virvou), talepis@unipi.gr (E. Alepis).

0360-1315/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compedu.2003.12.020

the practical facility offered to students of learning something at any time and any place. In many situations this means that learning may take place at home or some other site, supervised remotely and asynchronously by a human instructor but away from the settings of a real class.

However, in many cases it would be extremely useful to have such facilities in handheld devices, rather than desktop or portable computers so that users could use the software on a device that they can carry anywhere they go. Handheld devices render the software usable on every occasion, even when one is standing rather than sitting. Among handheld devices, which include palm or pocket PCs and mobile phones, the mobile phones provide the additional very important asset of computer-device independence for users. This is so because unlike mobile phones, palm-top PCs have to be bought by a person for the special purposes of computer use. On the other hand, mobile phones are very wide spread devices, which are primarily used for speaking purposes. However, they may also be used as computers. Thus, prospective users of handheld devices do not need to buy extra computer equipment since they can use their mobile phone, which they would buy and carry with them anyway. In this sense, using the mobile phone as a handheld computer is a very cost-effective solution that provides many assets. Such assets include device independence as well as more independence with respect to time and place in comparison with web-based education using standard PCs. Indeed, there are situations where students and instructors could use some spare time constructively to finish off their homework and lesson preparation, respectively, in situations where no computer may be available. Such situations may occur in trains, buses and coaches while commuting, in long queues while waiting or when unexpected spare time comes up. In the fast pace of modern life such situations can be very frequent.

In view of these compelling needs, the research work described in this paper has dealt with the problem of enriching existing educational software technology with mobile aspects. In particular, it has dealt with authoring tools and Intelligent Tutoring Systems (ITSs). This work resulted in the development of an authoring tool that can generate ITSs of multiple domains. The authoring tool is called Mobile Author.

Authoring tools in general are meant to be used by human instructors (prospective authors) to build tutors in a wide range of domains, including customer service, mathematics, equipment maintenance, and public policy; these tutors have been targeted toward a wide range of students, from grade school children to corporate trainees (Murray, 1999). More specifically, authoring tools that specialise at ITSs aim at providing environments for cost-effective development of tutoring systems that can be intelligent and adaptive to individual students. The main goal of ITSs as compared to other educational technologies, is to provide highly individualised guidance to students. It is simple logic that response individualised to a particular student must be based on some information about that student; in ITS technology this realisation led to student modelling, which became a core or even defining issue for the field (Cumming & McDougall, 2000).

Mobile Author allows instructors to create and administer data-bases concerning characteristics of students, of the domain to be taught and of tests and homework. The creation and administration of these data-bases can be carried out through a user-friendly interface from any computer or mobile phone. In this way the creation of mobile ITSs is facilitated enormously and a high degree of reusability is ensured.

Similarly, in the resulting tutoring applications, students can answer test questions and can read parts of the theory from any computer or mobile phone. The underlying reasoning of the tutoring systems is based on the student modelling component of the resulting educational applications.

The student modelling component monitors the students' actions while they use the educational system and tries to diagnose possible problems, recognise goals, record permanent habits and errors that are made repeatedly. The inferences made by the system concerning the students' characteristics are recorded in their student model that is used by the system to offer advice adapted to the needs of individual students. Moreover, the students' characteristics can be accessed by human instructors who may wish to see their students' progress.

2. Related work

Computer Assisted Learning (CAL) has grown enormously during the past decades and has been enhanced by the recent advances in web-based applications, multimedia technology, intelligent systems and software engineering. CAL may be used by instructors in a complementary way for their courses. Students may use educational software inside and outside classrooms in order to learn, practice and consolidate their knowledge. They may also use software from remote places in cases where the instructor is far from the student.

However, many researchers (Salomon, 1990; Welch & Brownell, 2000) point out that technology is effective when developers thoughtfully consider the merit and limitations of a particular application while employing effective pedagogical practices to achieve a specific objective. For example, Hasebrook and Gremm (1999) argue that learning gains are mainly due to instructional methods and thus many researchers aim at making their tutoring systems more effective using "intelligent" software technologies to adapt to the learners' demands, abilities and knowledge. The same applies to web-based educational applications, which are often limited to the capabilities of "electronic books" with little scope for interactivity with the student.

Intelligence can be added to educational software if the technology of ITSs is used. ITSs have been designed to individualise the educational experience of students according to their level of knowledge and skill (Du Boulay, 2000). It has been widely agreed that an ITS should consist of four components, namely the domain knowledge, the student modelling component, the tutoring component and the user interface (Self, 1999; Wenger, 1987). The domain knowledge consists of a representation of the domain to be taught (e.g., Biology, Chemistry, etc.). The student modelling component involves the construction of a qualitative representation that accounts for student behaviour in terms of existing background knowledge about the domain and about students learning the domain (Sison & Shimura, 1998). Student modelling may model various aspects of a student, such as what his/her knowledge level is in the domain being taught (e.g. Sison & Shimura, 1998), whether s/he has misconceptions (e.g., in previous research of our own (Virvou, 2002; Virvou & Kabassi, 2001)), what the associated emotions with the process of learning are (e.g. Elliott, Rickel, & Lester, 1999) etc. The tutoring component contains a representation of the teaching strategies of the system. Finally the user interface is responsible for translating between the system's internal representation and an interface language understandable by the student.

However the development of such educational programs is a hard task that needs quite a lot of effort from domain and computer experts. One solution to this problem is given by ITS authoring tools, which provide user-friendly environments for human instructors to develop their own ITSs in an easy way. Naturally, the reasoning abilities of the resulting ITSs have to be provided by the authoring tools. Therefore the authoring tools incorporate generic and domain-independent methods that can be customised to the particular tutoring domain of each instructor–author. In this sense authoring tools are harder to develop than ITSs but they provide a high degree of reusability. Therefore they are worth the extra effort.

Recently, there have been many research efforts to transfer the technology of ITSs and authoring tools over the Internet. A recent review (Brusilovsky, 1999) has shown that all well-known technologies from the areas of ITS have already been re-implemented for the Web. This could be expected since Web-based education needs intelligence to a greater extent than standalone applications. There are two important reasons for this: First, most Web-based educational applications are targeted to a much wider variety of users than any standalone application and thus there is a need for greater personalisation; second, students usually work with Web-based educational systems on their own (often from home) and cannot benefit from personalised assistance that an instructor or a peer student can provide in the normal classroom settings (Weber & Brusilovski, 2001).

Web-based ITSs and ITS authoring tools can be enhanced significantly by the incorporation of mobile features in them. Many researchers point out that the basic approach to mobile education should be integrative combining a variety of devices (mobile and non-mobile) via a variety of transmitting techniques (wired and wireless) (Farooq, Shafer, Rosson, & Caroll, 2002; Lehner & Nosekabel, 2002). At the current state, there are not many such mature systems since the technology of mobile computing is quite recent and has not yet been used to the extent that it can. However, there have been quite a lot of primary attempts to incorporate mobile features to this kind of educational technology and the results so far confirm the great potential of this incorporation.

In particular, Ketamo (2003) reports on an adaptive geometry game for handheld devices that is based on student modelling. Ketamo admits that the system developed was very limited and the observed behaviour was defined as very simple. However, an evaluation study that was conducted concerning this system showed that the learning effect was very promising with the handheld platform. A quite different approach is described in the system called KleOS (Vavoula & Sharples, 2002) which allows users to organise and manage their learning experiences and resources as a visual timeline. The architecture of KleOS allows it to be used on a number of different platforms including handheld devices. However, unlike Mobile Author, none of the above systems deals with the problem of facilitating the human instructor in the educational software development and maintenance. This problem is dealt with to a certain extent by Moulin, Giroux, Pintus, and Sanna (2002). The educational software they have built has addressed both the edition of a mobile lesson content and the management of the students on the field. However, their approach depended heavily on geo-referenced data. Therefore, unlike Mobile Author, it was beyond the scope of that system to achieve domain-independence and sufficient generality for the creation of an authoring tool that could provide an environment to instructors for them to create an ITS in any domain they are interested in.

3. Mobile authoring architecture and procedure

Mobile Author allows human instructors to create their own ITS in the domain they are interested in. For this purpose, human instructors have to insert domain data through a user-friendly interface from any computer or mobile device they wish to use. Then Mobile Author provides the reasoning mechanisms needed for the creation of a complete ITS.

The overall architecture of Mobile Author is illustrated in Fig. 1. Instructors can communicate with the system through IIS Server. This is how they can insert domain data in order to author their ITS. All domain data that is inserted by instructors is kept into the Tutoring Domain DataBases, which communicate with the Educational Application Reasoner and thus can form the ITS to be delivered to the students. The Educational Application Reasoner provides domain-independent reasoning mechanisms for monitoring the students' actions while they interact with the resulting ITS. Thus, the Educational Application Reasoner can perform student modelling and render the interaction with the students more individualised to their needs. The observations and inferences that have been made by the Educational Application Reasoner about individual students are kept in the Long Term Student Models. These models represent long term characteristics of students and the way they learn, for example their persisting misconceptions. Finally, in the resulting ITSs, the students can communicate with the system in typing through a Graphical

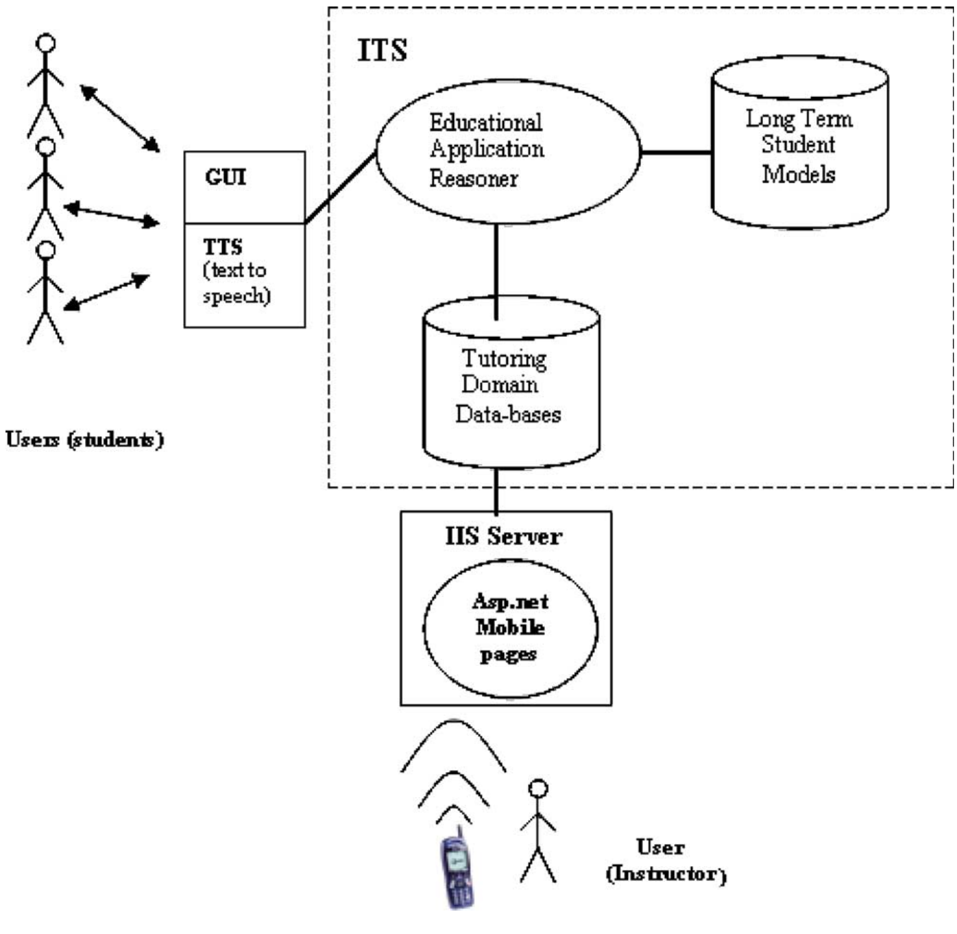

Fig. 1. The architecture of mobile author.

User Interface (GUI) via a computer or a mobile device. In addition there is a facility that allows students to interact with the application orally and a text-to-speech (TTS) and speech-to-text (STT) processor can make the appropriate transformations. However, this facility is only available when users use a computer and not a mobile device.

3.1. Tutoring domain

During the authoring procedure, the main task of the author (instructor) is to insert domain data into the application. Domain data may consist of lessons that the instructors wish to teach, and student assessment tests accompanying these lessons. In the lessons the instructors have the ability to create hypertext documents. In the tests, each assessment question may be associated with a part of the theory that students should know so that they may answer the question correctly. Tests may consist of questions of the following types: Multiple choice questions, fill-in the blank space and true/false questions.

Each type of question is associated with certain facilities that Mobile Author may provide to instructors for the creation of a sophisticated educational application. In multiple choice and true/false questions instructors have the ability to provide a list of frequent errors for each question or they may type explanations of errors. More specifically, they may associate erroneous answers to particular causes and explanations of errors so that these may be used by the system to give more detailed and informative feedback to students. Moreover, these explanations are used to create each student's profile, which is recorded permanently in his/her long term student model and is updated after each interaction of the student with the educational application. For example, the same explanation error may apply to more than one faulty answers of the student. In this case the long term student model holds the number of occurrences of the same type of explanation in the interactions of the student with the ITS. Then it compares the number of occurrences of different explanations and finds the student's most common mistakes and proneness to errors. These numbers are also used to find out whether students have made any progress since the last time they interacted with the educational application or whether they have forgotten parts of the syllabus that they seemed to have known in previous interactions.

An example of the creation of a new chapter entitled "Head and neck" during the mobile authoring procedure is illustrated in Fig. 2. An example of creation of tests during the mobile authoring procedure is illustrated in Fig. 3. In this example, an author (instructor) has started creating a multiple choice test of medium difficulty in medicine and has started editing the database questions.

3.2. Interface with mobile devices

The actual way that the instructor can communicate with the authoring tool is through the IIS server. In particular, instructors are able to connect to the system databases with their wireless device, either mobile phone or mobile Pocket PC simply by entering the corresponding URL into their device, which consists of the IP of the server computer and the name of the mobile ASP.NET page (example: http://195.252.225.118/mobilepage.aspx). In order to do that there are mobile web pages using ASP.NET mobile controls with Visual Basic.NET as the programming language. The server computer of Mobile Author requires using Windows 2000 or higher as the operating

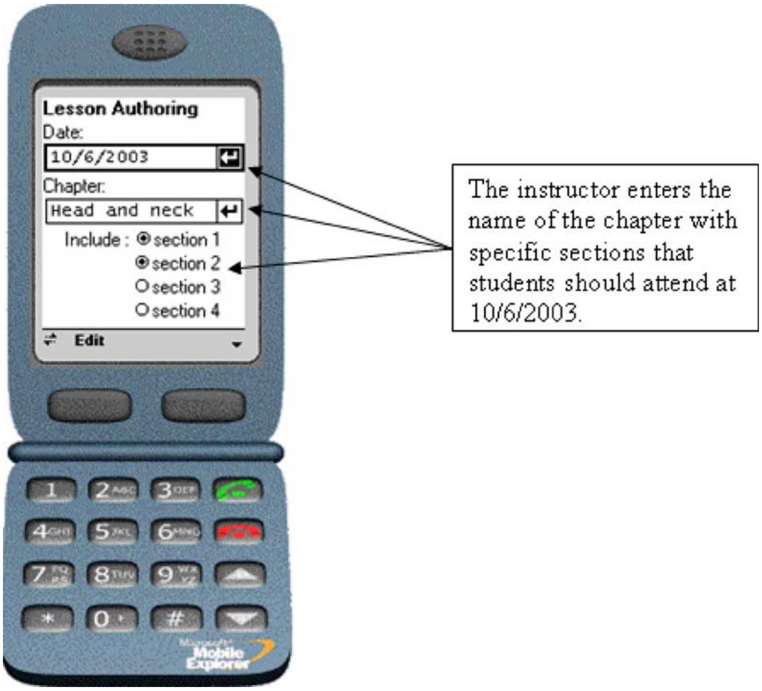

Fig. 2. The instructor is authoring a lesson.

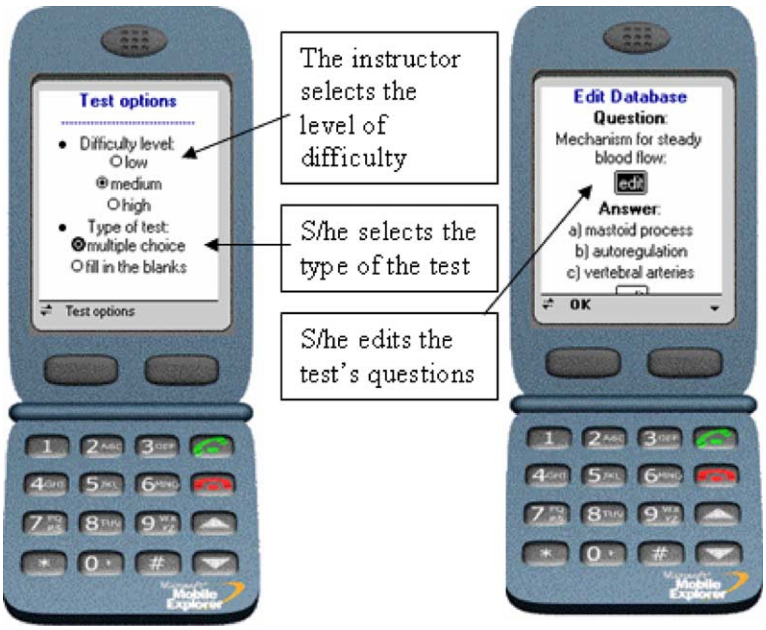

Fig. 3. Example of test creation by the author (instructor).

system, with IIS server installed. For the processing of the mobile web pages the .NET Framework SDK has been used and additionally the Mobile Web SDK. Any user can connect to these pages simply using any device that supports the Wireless Modelling Language (WML) or HTML, although some devices might differ in presentation. All these are supported by the Wireless Application Protocol (WAP) which is an open, global specification that empowers mobile users with wireless devices to easily access and interact with information and services instantly.

Loading new pages often takes a very long time due to the limited bandwidth of mobile devices. To address this problem, mobile pages contain as many server-side forms as necessary, whereas ordinary web pages can contain only one. Each form can contain different information and controls to display. This minimises the need to load new pages into the device. Moreover our mobile pages are "programmed" to automatically handle the rendering of each mobile control depending on the device being used. For this we used <DeviceSpecific> elements, which depend on the parameters specified by the device of the user.

The mobile ASP.NET pages consist of Web forms, server controls and script blocks. ASP.NET Mobile controls are designed to detect the client type and render the correct markup code so our mobile pages are capable of rendering to a variety of supported mobile devices using one set of code. This provides more functionality to our mobile pages, since they can be accessed either from a mobile phone or a mobile palmtop PC. Depending on the capabilities of the devices we define "device filters" for our application and then specify the corresponding filter for each type of device. We should also emphasise two aspects of the architecture of our mobile pages, which give them more "power" and (also very important) reduce costs. Mobile pages can contain as many server-side forms as necessary, whereas normal web pages can contain only one. Mobile controls automatically paginate content according to the device keeping the paged data on the server until requested by the user. Image controls allow us to specify multiple image files for one page image, using multiple formats and the correct image file is selected based on device characteristics. These foregoing aspects are very important considering the limited network bandwidth of mobile networks.

4. Mobile tutoring and course management

When an ITS has been created by an author (instructor), it may be used by students as an educational tool while instructors can be assisted in the management of the course and the assessment of their students. As a result, at this stage both kinds of user (students and instructors) can use the application to cooperate in the educational process. The instructor and the students are not only able to have easy access to the data-bases of the application but they can also "communicate" with each other.

The communication between instructors and students can be realised in many ways. By using a mobile phone (and of course connecting to the application's mobile pages) instructors can send short messages via the short message service (SMS), either directly to their students (if they also have mobile phones) or by e-mail. Alternatively instructors can "write" the message to the application's data-base. In this case, instructors have to declare the name of the receiver and the application will use its audio-visual interface to inform him/her as soon as s/he opens the application.

In the first case (e-mail or SMS message) the message is written to the data-base and then is sent to an internet service that provides SMS message and e-mail sending. Instructors are also able to send an e-mail directly through their mobile phones but it may be preferable for them to use the application to do that. The main reason for this is the fact that mobile networks are considerably slow and cost much and thus they may not be very convenient for the application's users. Thus the application is expected to send an e-mail to the internet service. The body of this e-mail will be the body of the short message and the "subject" of the e-mail will specify the receiver by his/her mobile telephone number. Messages (if sent from the instructor) can contain information about which test the student should visit next or about anything that the instructor thinks that the student should pay attention at. Messages can also be sent by students who can very easily send reports of their progress to their instructors.

Instructors have easy access to the "master" data-base of the application through their mobile phone, which means that they can stay informed of the progress of their students wherever they may be and whenever they want. An example of an instructor monitoring the progress of a student called John, through a mobile phone and sending an SMS to him is illustrated in Fig. 4, while the whole process of the interaction is illustrated in Fig. 5. Naturally, students also have the "privilege" to access the data-base by their mobile phone. Students would access the data-base for different reasons from instructors. For example, if they want to see which the next test is, or to read the remarks of their instructor about previous tests.

Additionally both instructors and students are able to send short messages (SMSs) containing remarks and additional information. The body of the SMS is entered in the "enter message" field and the name of the receiver is written in the "enter student name" field. Finally after the "direct

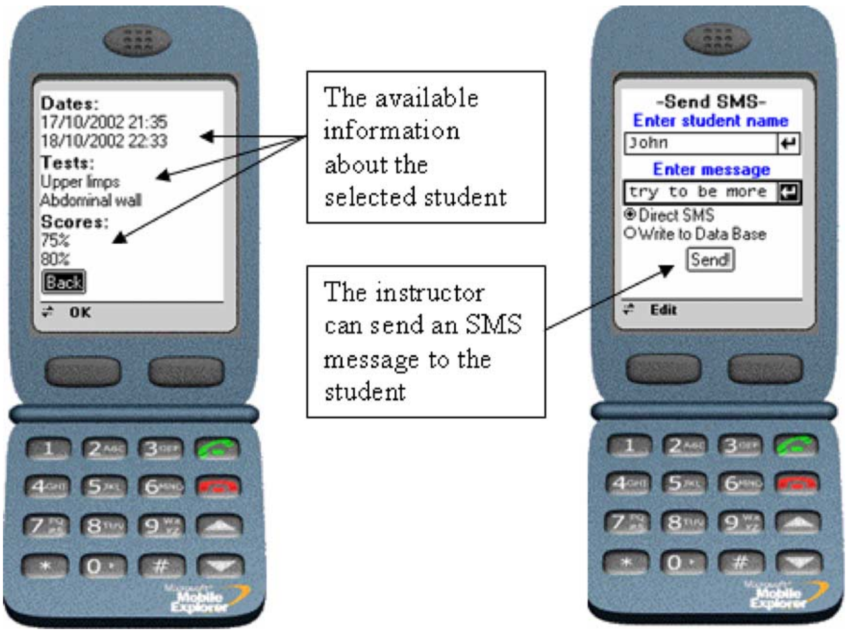

Fig. 4. Example: An instructor monitors the progress of a student and sends SMS.

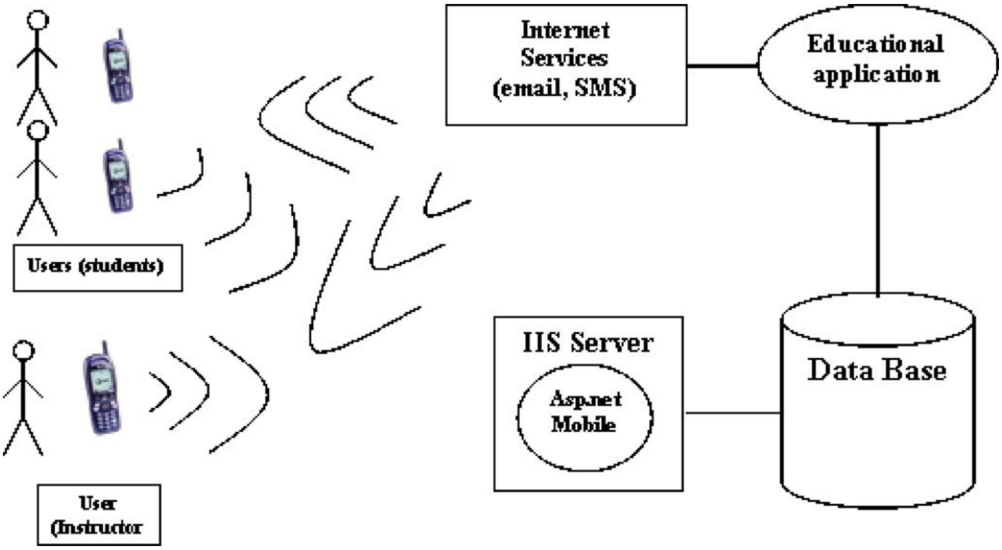

Fig. 5. Communication between instructors and students.

SMS or Write to Database" choice the message is sent by pressing the "Send!" button. If the user selects "Direct SMS" then the SMS is delivered directly to the mobile device of the receiver through an Internet service described earlier. Only the username is required since all the mobile phone numbers are stored in the system's database.

The discrimination between the instructor and the simple users is done by the application (installed in the server) and for each different user a personal profile is created and stored in the database. In order to accomplish that, user name and password is always required to gain access to the mobile web pages. On the first level of authentication the application determines whether there is a valid username–password or not. On the second level of authentication users are identified as simple users (students) or supervisors (instructors).

Mentioning the security and privacy aspects of our mobile application, it is also able to use the Secure Sockets Layer (SSL) in order to encrypt data during important data transactions. Both the IIS server and the. NET Framework for mobile devices support this operation simply by inserting a valid certificate signed by a certificate verifying authority (such as VeriSign). The user can easily use this "provision" by inserting an "s" in his/her URL: https://195.252.225.118/mobilepage.aspx.

As mentioned earlier, the ITS that has resulted from the authoring process, uses the Educational Application Reasoner to reason about the students' way of learning. Then all long-term characteristics inferred about a particular student are stored in his/her long term student model so that they can be used by the application itself to be more adaptive to the student's needs or by the instructor so that s/he can provide tutoring advice tailored to each individual student. Thus, the educational application monitors the students' actions while they work with the system. It keeps a record of how many times (if any) a student has visited a particular web page of the theory and then it keeps a record about how the student performs in tests. Then all of the information that has been gathered is used by the system so that a personal student profile is created and his/her progress is monitored.

Students may take tests through their mobile phones. An example of a student taking a test is illustrated in Fig. 6. When students answer questions, the tutoring system tries to perform error diagnosis in cases where the students' answers have been incorrect. Error diagnosis aims at giving an explanation about a student's mistake taking into account the history record of the student and the particular circumstances where the error has occurred. Giving a correct explanation of a mistake can be a difficult task for a tutoring system. One problem that complicates this task further is ambiguity, since there may be different explanations of observed incorrect users' actions. For example, in a fill-in the blank space question a student may give an incorrect answer simply because s/he has mistyped the answer. However, this may well appear as a lack of knowledge in the domain being taught.

Mobile Author always checks the students' answers with respect to spelling or typing errors. In particular, spelling mistakes may be identified by the following procedure. Each time that an incorrect answer is typed by a student, this answer is converted to the actual pronunciation sound of the word typed and is then compared to the pronunciation sound of the correct answer that the student should have given. If these answers are found similar then perhaps the student has made a spelling mistake. If this is the case, it does not mean that s/he does not know the answer at all. If the student has typed a word, which is completely different from the correct one then s/he has made a domain error. Such information about the cause of an error is then recorded to the student model. Thus, the tutoring application records both domain-dependent and domain-independent

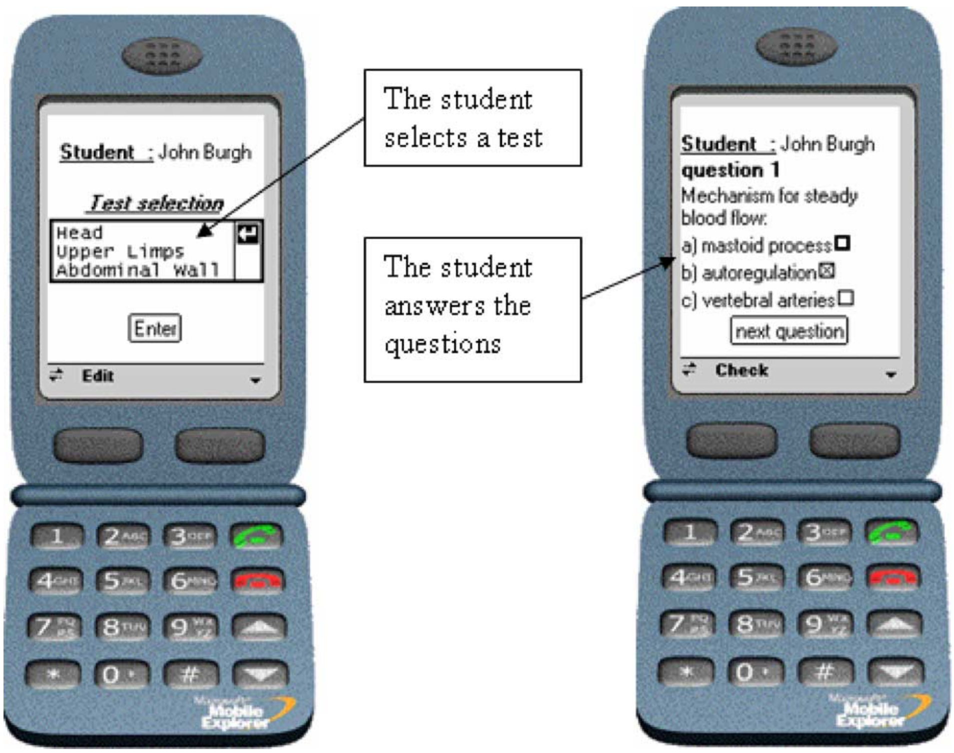

Fig. 6. A student is taking a test.

information about particular students to their long-term individual student models. For example, a student may be consistently making a lot of spelling mistakes when s/he is typing answers to questions posed by the tutoring application. This is a domain-independent feature of the student concerning the student's spelling skills rather than domain knowledge.

Domain errors may be examined further for the identification of a deeper cause of error. For example, the instructor may have provided a list of frequent errors and each of them may have been associated with an underlying cause of error. In this way the instructors may create a bug-list, which is based on their experience about students' making errors. Such lists may be used for further classification of domain errors and the student model is updated.

5. Evaluation

Software that is meant to help the educational process can be considered successful if it is approved by human instructors and is educationally beneficial to students. Otherwise it may not even be included in the educational process and may not be accepted by its targeted users. Thus, evaluation of this kind of software is an important phase that has to follow development at all times. In particular, formative evaluation is one of the most critical steps in the development of learning materials because it helps the designer improve the cost-effectiveness of the software and this increases the likelihood that the final product will achieve its stated goals (Chou, 1999).

Educational applications constitute a special category of software that needs to model and assist many aspects of the cognitive processes of humans whether these are learners or instructors. Therefore, in the literature there are evaluation methods that are completely specialised to educational software. One such evaluation framework outlines three dimensions to evaluate: (i) context; (ii) interactions; and (iii) attitudes and outcomes (Jones et al., 1999). The context determines the reason why the educational software is adopted in the first place, i.e., the underlying rationale for its development and use; different rationales require different evaluation approaches. Students' interactions with the software reveal information about the students' learning processes. The "outcomes" stage examines information from a variety of sources, such as pre and post-achievement tests, interviews and questionnaires with students and tutors. This framework has been used for the evaluation of Mobile Author.

The underlying rationale of Mobile Author involves offering more convenience with respect to time, place and kind of device to its users (instructors and learners), therefore the context of the evaluation required an emphasis on the mobile aspect of the application. Then, students' interactions with the software were evaluated with respect to the students' learning processes while they used mobile devices. Finally, the "outcomes" stage involved pre and post-achievement tests before and after the use of a mobile device. In addition, it involved many interviews of students and instructors, which focused mainly on evaluating the use of mobile devices. In view of these, the evaluation of Mobile Author involved both instructors and students and was conducted in two different phases. At the first phase, the authoring procedure was evaluated by instructors, who were interviewed after they had developed an ITS. The second phase concerned the evaluation of the resulting educational applications and involved both instructors and students. The instructors of the second phase were exactly the same as in the first phase, so that they could have a complete experience with Mobile Author, both for the creation of an ITS and the management of their course.

At the first and second phase, 10 instructors participated in the evaluation. Six of them were secondary school instructors and were asked to prepare lessons and tests in geography, history, biology, physics, chemistry and English, respectively, depending on their expertise domain. The rest of the instructors were University instructors. Two of them were Medical Science instructors and two of them were Computer Science instructors. All of the instructors who participated in the experiment were familiar with the use of computers. In addition, they had been trained for the use of Mobile Author before the experiment.

When interviewed, all of the authors confirmed that Mobile Author had a user-friendly interface and that the mobile facilities were either useful or very useful. More specifically, 7 of them stated that they found the mobile facilities of Mobile Author either useful or very useful both for the creation and the maintenance of their courses whereas 3 of them said that they had not used the mobile features at all during the creation of the course but they found them useful during the maintenance of the course. The exact answers of instructors to questions about the mobile features of the authoring tool are illustrated in Tables 1 and 2. As expected, all of the 7 instructors who found useful the mobile features of the application for both phases, made clear that they had used the mobile facilities in a complementary way with a desktop computer, since the authoring process involves inserting a lot of data. Thus it would have been difficult for anyone to develop the whole course using a mobile phone.

It must be noted that among the most enthusiastic instructors about the mobile features were the University instructors who were Computer Science and Medical instructors. To some extent, it was expected that Computer Science instructors would probably like Mobile Author more than the other experts due to their familiarity with technology. Indeed they rated the usefulness of the mobile features very highly. However, the application was even more appreciated by Medical Science Instructors. This was probably due to the fact that Medical Science Instructors usually

Table 1
Instructors' opinions about the usefulness of mobile creation of courses

	Instructors	Instructors' answers to the question: "How do you rate the usefulness of the facility of mobile creation of a course"		
		(a) Not useful at all	(b) Useful	(c) Very useful
	University level			
1	Computer Science		√	
2	Computer Science		√	
3	Medicine			√
4	Medicine		√	
	School level			
5	Geography		√	
6	History	√		
7	Physics	√		
8	Biology		√	
9	Chemistry			√
10	English	√		

Table 2
Instructors' opinions about the usefulness of mobile maintenance of courses

	Instructors	Instructors' answers to the question: "How do you rate the usefulness of the facility of mobile maintenance of courses"		
		(a) Not useful at all	(b) Useful	(c) Very useful
	University level			
1	Computer Science			√
2	Computer Science		√	
3	Medicine			√
4	Medicine			√
	School level			
5	Geography		√	
6	History		√	
7	Physics		√	
8	Biology			√
9	Chemistry			√
10	English		√	

depended a lot on their mobile phones due to the heavy responsibilities that they had and the fact that they had to be in many different places during the day (hospital, college classrooms, visits to patients etc.).

The second phase involved in total 50 students, 5 students from each of the respective classes of the 10 instructors who participated in the evaluation. The underlying rationale of mobile ITSs lies on the hypothesis that these applications are more convenient and flexible to use while they retain the educational quality. At a first glance, the validity of this hypothesis might look obvious. However, there may be students who are not familiar with educational software in general and thus might not like the particular applications. On the other hand, there may be students, who are very familiar with computers and mobile phones and are very happy to use them for educational purposes. Hence, one important aspect of the evaluation was to find out whether students were indeed helped by the mobile environment. Another very important aspect was to find out whether students had gained educational benefits from the ITSs.

Students were asked to use the ITSs as part of their homework for their courses. After the courses were completed, the students were interviewed. A part of the questions that were asked to the students is illustrated in Table 3. 76% of the students found the whole educational software application useful or very useful. An even greater percentage (82%) found the instructor–students mobile communication facilities useful or very useful during the course. Pie-charts representing these results are illustrated in Fig. 7. The students who were more enthusiastic about these facilities were either these students who attended many lectures during the day and thus they had a very "mobile" timetable, or students who liked to use SMS in their everyday lives for all types of their communication with other people, whether these were friends, class-mates or instructors. In particular, for this category of students who constituted 34% of the total students, the facility of using mobile devices rendered the whole educational software application more attractive and engaging simply because they liked the mobile phone as a medium.

Table 3
A part of the questions that were asked to the students

1	How do you rate the usefulness of the whole application? (a) Not useful (b) Useful (c) Very useful
2	How do you rate the usefulness of the instructor–students mobile communication facilities? (a) Not useful (b) Useful (c) Very useful
3	Did you consider the application attractive? If yes, what did you like about the application?

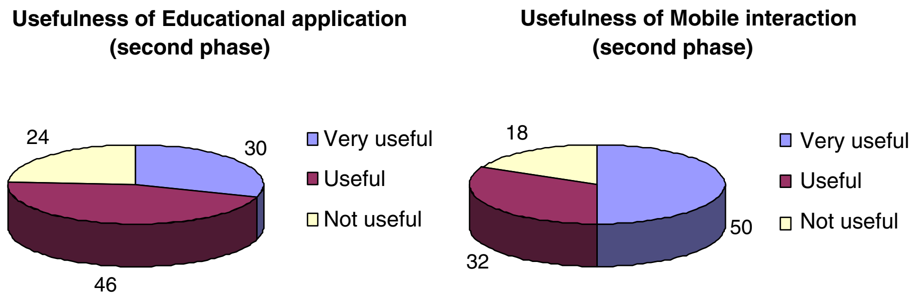

Fig. 7. Rate of usefulness of educational application and mobile interaction according to students.

6. Conclusions

Mobile phones have already become very popular among people and thus they are imposing a new culture. As a result, their use in education as a new tutoring and communication medium can be very useful. However, in the case of education, many design issues have to be taken seriously into account so that the resulting applications can be educationally beneficial to students and be included in the educational process. Among these important design features, is the high degree of adaptivity and personalisation that has to be achieved in tutoring systems. These features need student modelling facilities that are mainly used by ITSs. Moreover, instructors have to be included in the educational design of tutoring applications, therefore authoring tools can provide a good solution for non-computer experts to create cost-effective tutoring applications of high quality.

In this paper, we have shown how a mobile ITS authoring tool can be designed to provide the relatively new mobile facilities to both instructors and students while retaining high quality of the educational application with respect to interactivity, adaptivity and personalisation. The research conducted, has resulted in the development of Mobile Author that has been evaluated among instructors and students. The evaluation results were very encouraging and showed that the contribution of mobile software features to education can be appreciated by both instructors and students.

References

Brusilovsky, P. (1999). Adaptive and intelligent technologies for web-based education. In C. Rollinger, & C. Peylo (Eds.). Intelligent systems and teleteaching [Special issue]. *Künstliche Intelligenz, 4* (pp. 19–25). Available: http://www2.sis.pitt.edu/~peterb/papers/KI-review.html.

Chou, C. (1999). Developing CLUE: A formative evaluation system for computer network learning courseware. *Journal of Interactive Learning Research, 10*(2).

Cumming, G., & McDougall, A. (2000). Mainstreaming AIED into education? *International Journal of Artificial Intelligence in Education, 11*, 197–207.

Du Boulay, B. (2000). Can we learn from ITSs?. In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *LNCS: 1839. ITS 2000* (pp. 9–17). Berlin Heidelberg: Springer-Verlag.

Elliott, C., Rickel, J., & Lester, J. (1999). Lifelike pedagogical agents and affective computing: An exploratory synthesis. In M. J. Wooldridge & M. Veloso (Eds.), *LNCS: 1600. Artificial intelligence today* (pp. 195–212). Berlin: Springer-Verlag.

Farooq, U., Shafer, W., Rosson, M.B., Caroll, J.M. (2002). M-Education: Bridging the gap of mobile and desktop computing. In *IEEE international workshop on wireless and mobile technologies in education (WMTE'02)* (pp. 91–94).

Hasebrook, J. P., & Gremm, M. (1999). Multimedia for vocational guidance: Effects of individualised testing videos and photography on acceptance and recall. *Journal of Educational Multimedia and Hypermedia, 8*(4), 377–400.

Jones, A., Scanlon, E., Tosunoglu, C., Morris, E., Ross, S., Butcher, P., & Greenberg, J. (1999). Contexts for evaluating educational software. *Interacting with Computers, 11*(5), 499–516.

Ketamo, H. (2003). An adaptive geometry game for handheld devices. *Educational Technology & Society, 6*(1).

Lehner, F., Nosekabel, H. (2002). The role of mobile devices in e-learning – first experiences with a wireless e-learning environment. In *IEEE international workshop on wireless and mobile technologies in education (WMTE'02)* (pp. 103–106).

Moulin, C., Giroux, S., Pintus, A., & Sanna, R. (2002). Mobile lessons using geo-referenced data in e-learning. In S. A. Gouarderes, & F. Paraguacu (Eds.). *Intelligent tutoring systems 2002, LNCS* (Vol. 2363, p. 1004).

Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education, 10*, 98–129.

Salomon, G. (1990). Studying the flute and the orchestra: Controlled vs. classroom research on computers. *International Journal of Educational Research, 14*, 521–532.

Self, J. (1999). The defining characteristics of intelligent tutoring systems research: ITSs care, precisely. *International Journal of Artificial Intelligence in Education, 10*, 350–364.

Sison, R., & Shimura, M. (1998). Student modelling and machine learning. *International Journal of Artificial Intelligence in Education, 9*, 128–158.

Vavoula, G., Sharples, M. (2002). KleOS: A personal, mobile, knowledge and learning organisation system. In *IEEE international workshop on wireless and mobile technologies in education (WMTE'02)* (pp. 152–155).

Virvou, M. (2002). A cognitive theory in an authoring tool for intelligent tutoring systems. In *Proceedings of IEEE international conference on systems man and cybernetics 2002 (SMC'02)* (Vol. 2, pp. 410–415).

Virvou, M. & Kabassi, K. (2001). F-SMILE: An intelligent multi-agent learning environment. In *Proceedings of IEEE international conference on advanced learning technologies 2002 (ICALT'02)*.

Weber, G., & Brusilovski, P. (2001). ELM-ART: An adaptive versatile system for web-based instruction. *International Journal of Artificial Intelligence in Education, 12*, 351–384.

Welch, M., & Brownell, K. (2000). The development and evaluation of a multimedia course on educational collaboration. *Journal of Educational Multimedia and Hypermedia, 9*(3), 169–194.

Wenger, E. (1987). *Artificial intelligence and tutoring systems*. Morgan Kaufmann.