Abstract

Web-based education is particularly good for remote teaching and learning at any time and place, away from classrooms and without necessarily the presence of a human teacher. However, this independence from real teachers and classrooms may cause emotional problems to students who may feel deprived of the benefits of human-human interaction. In this paper we address this problem by rendering web-based human-computer interaction more human-like. This is achieved by an authoring tool that human teachers may use to create their own educational software for their remote students. Through this authoring tool, human teachers may also create their own tutoring characters for the user interface of the educational application. These characters may be parameterised in many aspects, the way they speak, the pitch, speed and volume of their voice, their body-language and the content of their messages. In this way they may reflect the human teachers’ vision of teaching behaviour.

Keywords: Web-based education, animated agents, authoring tools, tutoring characters.

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 the practical facility 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 teacher but away from the settings of a real class. However, this may also create some emotional problems to students who may be deprived of the advantages of human-human interaction between them and a human teacher. This may affect the educational process in a negative way because as Goleman [1] points out, how people feel may play an important role on their cognitive processes as well.

A remedy for this problem may lie in rendering human-computer interaction more human-like and affective in educational software. To this end, the incorporation of speaking, animated agents in the user interface of the educational application can be very important. Indeed, the presence of animated, speaking agents has been considered beneficial for educational software [4, 2]. Hence, there have been many educational applications that incorporate animated pedagogical agents in their user interfaces [9, 5, 8].

However, imitating the behaviour of a human teacher may take as many forms as the number of teaching strategies and characters of human teachers. Canned patterns of behaviour incorporated into a tutoring system may not be acceptable by human teachers who are in charge of the remote teaching and may not be appropriate for their remote students.

In view of these, we have developed a web-based authoring tool for educational software that may be used by human teachers to create their own tutoring systems and their own virtual tutoring characters. 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 [7]. The human instructors that will use authoring tools should not necessarily be computer experts and should be helped to develop sophisticated educational applications in an easy and cost-effective way. Thus authoring tools may provide reasoning mechanisms and user interface components, useful for the resulting tutoring systems, that would be difficult to create from scratch [12].

However, as yet there are no authoring tools that provide parameterisation in user interface components such as speech-driven, animated agents. The present authoring tool provides the facility to authors to develop tutoring systems that incorporate speaking, animated...
agents who can be parameterised by the authors-teachers in a way that reflects their own vision of teaching behaviour in the user interface of the resulting applications.

2 Animated agent

In the educational applications that result from the authoring process described in this paper, the animated agent of the user interface is a cartoon-bear. The reason why we have selected a cartoon is the fact that in a study that was conducted previously in the context of another tutoring system it was shown that speaking cartoons are more likeable by students than speaking anthropomorphic agents [6]. This is probably due to the fact that cartoons look more amusing and less stressful than anthropomorphic agents.

2.1 External features of the animated agent

The cartoon-bear is an animated agent who can move around the tutoring text and can show parts of the theory that a student should read (Figure 1). It has also incorporated features of human body-language. It shows patience while the student reads the theory, boredom if the student is not responding to the system, wonder if the student makes an unexpected move etc.

Moreover the author-instructor may choose from 27 speech engines that the system incorporates. These speech engines are synthesisers that produce different voices. Finally the system offers the facility of parameterising these voices by changing the pitch, speed and volume. Thus, the resulting tutoring system may use the voices differently in different contexts to show enthusiasm, when the student is doing particularly well, to imitate whisper, when it judges that the student needs help, or even to show anger when the student is consistently careless and does not pay any attention to the system. However, the exact way that the system is going to respond to such situations is going to be determined by the authors-instructors themselves.

2.2 Underlying reasoning

The animated agent’s behaviour in the application is determined by the way that it has been programmed to respond on particular tutoring situations. This is how the animated agent turns into a tutoring character. However, the tutoring character needs to employ some reasoning mechanisms so that it may be able to identify different tutoring situations for each individual student.

The underlying reasoning of the tutoring characters is based on the student modelling component of the resulting educational applications. In general, the student-modelling component of an educational application 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 [10].

In the present system the student modelling component monitors the students’ actions while they use the educational system and tries to diagnose possible problems, recognise goals and record permanent habits and errors that are made repeatedly. The student modelling component employs a combination of the overlay [11] and stereotype technique [3] for finding out the knowledge level of the student in the particular domain being taught. Moreover, it records inferences concerning the long-term characteristics of students in their long-term student model. Based on the inferences about students, the user interface agents provide individualised advice concerning each situation.

For example, in cases of a student’s error in an electronic exam question, the system tries to find out whether the error was due to an accidental slip or a persisting misconception. This procedure is assisted by long-term information about each individual student that has been recorded in his/her long-term student model. Then depending on the situation and the known features of each student, the tutoring character forms a response tailored to the particular circumstances using the appropriate voice, pitch, speed, volume and body movements according to the way it has been programmed by the human teacher.

3 Creation of tutoring characters

Instructors are given many facilities to create the tutoring characters through a user-friendly interface. The creation process can be carried out either from a standalone version of the authoring tool or from a web-based one. Instructors can program the movements of the animated agent and its voice so that they may change in
the way that the instructors wish, depending on particular tutoring situations.

3.1 Basic facilities

The animated agent comprises a quite large set of programmatically controlled commands. These commands deal not only with the “on the screen” movement of the agent but also control its hands, face, feet and make it act like speaking. Its face movements in conjunction with the “speech synthesis system” give the impression of a quite realistic human-like assistant.

Figure 2: Setting parameters for the voice of the tutoring character

Instructors are offered the facility to select from a number of different speech engines that are incorporated into the system. The exact number of speech engines depends also on the pre-installed software of the particular computer that the teacher is using. This means that although the authoring tool always checks and installs a set of speech engines, the user may also install and use other speech engines. These speech machines as part of the operating system are controlled through the authoring tool. In particular, the authoring tool has special configurations in order to produce desirable results of human-like speaking.

The basic step though is to control the speed, volume and the pitch of the sound. Figure 2 illustrates three track bars that can be used to change speech attributes. With these attributes the instructor may create and store for future use many kinds of voice tones such as happy tone, angry tone, whisper and many other depending on the application’s needs.

Figure 3: Detailed controls for the voice of the tutoring character

For example, by increasing the pitch and also the speed and volume of the speech we have the effect of a more “angry” tone of speaking. This may also be achieved by selecting the appropriate speech engine. We may have a special speech engine that always synthesises speech with an angry tone. Additionally, the instructor may use the form of Figure 3 that provides more specific and detailed controls. In this form the instructor may also have the ability to set the exact pronunciation of a word by using phonemes.

The effects created by the changes in speech attributes may be used in the application for educational purposes. For example, Figure 4 illustrates a form with four tabs, where each tab corresponds to a particular student’s situation while s/he is taking a test. In this form the instructor can specify the behaviour of the animated agent when the student gives an answer, which is either correct or completely wrong, or when the answer is wrong.
but close to the correct one. The closeness of an answer to a correct answer is determined by the phonetical similarity of the answer that the student has given to the correct answer. If the correct answer is phonetically the same as the answer that the student has given but the spelling is not the same then the student has probably made a spelling mistake rather than a domain error. Finally the fourth tab corresponds to the situation when a student ends the test having a very good score. Easily in each tab the instructor can add text to be spoken, enriched with commands that are separated by slashes. As one can see in Figure 4, the instructor can insert a command by selecting it and writing its value in the text box. In this way, the oral effect in conjunction with a programmed graphical animation of the agent becomes part of the educational process of the system.

In our example where the text to be spoken is “Very good! You are doing great!” the teacher has also entered two commands. The first command “\Chr=Happy\” tells the speech synthesis system that for this situation a pre-programmed character will be used. In this case the character will have a happy tone. This is achieved by changes to the speech attributes as described before. These special values for pitch, volume and speed, combined with an animation of the agent (movements of hands, feet and head) are stored in the system’s database as the “happy” character. Using the authoring tool many other “characters” may be created and stored for future use. The second command “\pau=1000\” inserts a pause of 1000 milliseconds in the speech. There are many other commands that can be selected from the drop down list box and inserted into the text after having specified a value. All these options aim to make the system usable by teachers who are not programmers and thus should not be obliged to enter the commands manually by writing them.

There is also a large set of over sixty commands that refer to the animation controls of the agent. These, including “on screen” movements provide the instructors with the facility of making the behaviour of the agent as much human-like as possible. The cartoon-bear of our example can move its hands, feet head and mouth, obtaining many facial expressions particularly useful for educational purposes. The agent also supports a set of reading, writing congratulating, disapproving and approving animations that are frequently used in tests. The teacher can also choose whether the spoken text can additionally be read from a balloon tip appearing near the agent. Finally, after the modifications of the character have been completed, all the work can be saved by pressing the “Save” button. At any time the instructors can also use the authoring tool to open and modify work they did and saved in the past or even load programmed behaviours of agents from other machines.

Figure 4: Programming the behaviour of animated agents depending on particular students’ situations

The application has built in tools, to which only the instructors have access. These tools help the instructors modify the behaviour of the characters further. Not only can the instructor “command” the assistant to say something under certain circumstances but s/he can also add commands in the text that will be spoken. These commands are understood by the “speech synthesis system” and are interpreted into changing speech attributes.

3.2 Web-based features

The web-based facility has been provided to instructors so that the system is more flexible and accessible from everywhere. In the case of the web-based approach, the instructor can use any personal computer or handheld PC or even a mobile phone that supports HTML, that is connected to the Internet. In such cases, the instructor can easily modify the behaviour of the tutoring characters by the use of a web page connected to the educational application and its database. With this approach the instructor has the advantage of having
multiple access to many students’ PCs and to make particular modifications depending on each occasion. Of course the “IP address” of each computer that is connected to the network must be known and this is simply the only limitation. Through a web page, the instructor has access to the active application and to the database of the educational system.

Figure 5: Example of a web page of the authoring tool

In Figure 5, an example of a web page is illustrated. The web browser used is the Internet explorer and IIS 5.0 is used as the application’s server. Moreover as illustrated in the example, the instructor is connected locally (http://localhost/…) to a web page that interacts with the educational application. The interaction is based on the system’s database where the instructor has read and write rights. In the web page, changes can be made not only to the speech engine but also to the speech attributes described above. Then pressing the “Send to the application” button sends the information. Specifically, the information is first written in the database and then transferred to the basic application. With this facility, instructors have “remote access” to all the installed applications and that is why user authentication foreruns are needed to prevent undesirable use form other users. Another important reason why the authentication process is fact that necessary is the web-based version of the system may be used for the creation of many different exam tests.

Figure 6: Example of an authentication process

Figure 6 illustrates an example of an authentication process. The instructor enters his/her user name and password and if the authentication process is positive s/he may enter the system. With this kind of access the instructors can make on line changes to the applications concerning special characteristics and needs of the users that run them. They can also modify the tests and of course the behaviour of the animated agents that accompany these tests.

Teachers can modify their own tests leaving other colleagues do their own work (changes, modifications) as well. For the web-based part of the tutoring system the only requirement, as emphasised before, is a machine connected to the Internet with Internet explorer working. This means that teachers do not need to have access to their students in order to check and/or modify the tutoring system. Furthermore it is a process that can be done at any place and any time, saving teachers’ precious time.

The fact that the web pages have a very simple lay out and contain basic functions of selections as well as database access, makes them quite practical.
Technologically improved mobile devices that are already in the market can easily view web pages and interact with them. Basic features such as HTML and java support are the only prerequisites for a mobile phone to connect to a web page of the authoring tool.

4 Conclusions

In this paper we have presented an authoring tool that may be used by instructors to generate Web-based educational applications. Both the authoring tool itself and the resulting educational applications may be used remotely by instructors and students respectively.

The authoring tool provides an important facility to instructors for the creation of their own tutoring characters for the user interface of the resulting applications. In this way, the tutoring characters that are speaking, animated agents may represent the teaching behaviour of the human instructor who is in charge of the remote lessons. The tutoring characters may be programmed to produce different movements and have different voices, which may change speed, pitch and volume depending on the tutoring situations. Thus, the characters' body-language and the way they speak may create the impression of pedagogic teachers’ emotions, such as compassion and empathy, when students are having justifiable problems, enthusiasm when students are doing particularly well or even anger when students seem not to pay any attention.

On the other hand, the students, who are going to use the educational applications remotely, will have a user interface that is more human-like and affective. Thus they may feel less deprived of the human-human interaction between them and a human teacher that would take place in the settings of a real classroom.

References


