Authoring Educational Games Through Affective Teachable Agent

Ailiya¹, Chunyan Miao² and Zhiqi Shen³

School of Computer Engineering,
Nanyang Technological University, Singapore
¹ai0001ya@e.ntu.edu.sg, ²ascymiao@ntu.edu.sg, ³zqshen@ntu.edu.sg

Abstract—To encourage teenagers to learn in their virtual life, more and more educational researchers realize the importance of virtual learning communities. To ensure the long term running of the community, teachers should be involved to the design of educational features in an iterative way. To this end, this paper proposes an authoring tool which extends the teachable agent in Virtual Singapora project with authoring capabilities. In this way, not only educators can easily bring their ideas into the game design, but also the game developers can effectively implement the design to the real game scenarios. A preliminary study illustrates the effectiveness of this approach.

Keywords—Affective Teachable Agent; authoring tool; learning goals

I. INTRODUCTION

In this day and age, young generations have a twofold life. One is in real world, the other in virtual communities. To adapt this emerging life style, various researches integrate educational features into virtual games for making full use of students’ interest and motivate them to learn more things. In light of the cyber game communities, we work on build up a virtual learning space which aims to encourage teenagers to learn in their virtual life and let the playing of educational games become part of their everyday life.

To achieve this, the game design should fulfill two main requirements. First, concerning the game players who may come from different schools and in different grades, the game system should have the flexibility to personalize the learning content for different students. Second, the designing of learning content is not a process which puts things right once and for all. Instead, the learning content should be able to easily update regularly. For these requirements, the game developers need to coherently work with educational experts throughout the game design and long term maintenance. Thus, an authoring tool [1] is indispensable to unite teachers and game developers. Therefore, our research utilizes one of our previous research work, Affective Teachable Agent (ATA) [2], and extended it as an authoring tool to assist teachers to design and timely update the learning materials through an iterative authoring process.

Unlike the traditional pedagogical agents which perform as a virtual teacher to “spoon-feed” knowledge to students, ATA is designed as a “naïve” learner who needs to be taught by our students. This reversed design is inspired by an interesting phenomenon that when students teach their peers, they learn much better than they learn for themselves[3]. This area has been studies in many researches [4-6]. The Learning-by-teaching [7] effect is represented by ATA through its “Teachability”, the capability to be taught. Through teaching the agent, students no longer passively “receive fruits picked by others” but take charge of their learning progress and think about how to “grow their own plants”.

Owing to the characteristic Teachability, ATA can be used in different ways according to different “teachers”: when a student teaches the agent, the agent is performed as a “naïve” learning companion, but when a teacher teaches the agent during the designing process, the agent is worked as an authoring tool which can bring teachers a natural way to convey domain knowledge to agent’s knowledge base. In this way, our teachers can be easily involved into the designing process. The preliminary feedbacks show that the authoring process can encapsulate the game programming details from educators, and the setting of learning goals contributes to a well-defined game structure since it works as a map that tells developers what contents should be involved.

II. THE AGENT MODEL AND SYSTEM DESIGN

In [2], we have introduced three life cycles of the ATA, which are learning cycle, reasoning cycle, and emotion cycle. As the proposed ATA in this paper is an extension of the previous ATA with authoring capability, the extended features only affect the learning cycle. The learning cycle represents agent’s learning process, which includes two components, Perceiving and Knowledge Storing. With the feature of involving teacher into the game design, current ATA not only perceives the knowledge taught by students, but also perceives the authored content from the Goal Net Designer interface as Fig. 1. Current teachable agents set all the perceived knowledge from Goal Net Designer as default correct knowledge.

Our system involves three parties: the educational designers (teachers) identify learning goals and define tasks in transitions between goals; the program developers realize the tasks in transitions via editing transition functions; and the students customize their ATA’s learning path through selecting existing learning goals in the system at the beginning of each game scenario, and conducting their teaching tasks via multiple ways. The details are summarized as Table 1.

III. TEACHABLE AGENT AS AN AUTHORING TOOL

In this section, we will use a part of Virtual Singapora project which is related to ATA as an example to illustrate the authoring process. The example focuses on a specific subject, transport in plants in secondary science curriculum.
TABLE I. THE SUMMARY OF VS SYSTEM WITH DIFFERENT PARTIES

<table>
<thead>
<tr>
<th>Virtual Singapore</th>
<th>Educational Designer (Teachers)</th>
<th>Program Developer</th>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Authoring learning content</td>
<td>Do the game programming based on the authoring results</td>
<td>Teach ATA through:</td>
</tr>
<tr>
<td></td>
<td>1. Identify learning goals (first by topic, then by difficulty level)</td>
<td>1. Build up game flow based on learning goals</td>
<td>1. Tutoring ATA via Concept Map, demonstrative experiment, etc.</td>
</tr>
<tr>
<td></td>
<td>2. Define task list in transitions for detail</td>
<td>2. Realize game based on the task list in transitions</td>
<td></td>
</tr>
</tbody>
</table>

**Goal Net Designer**

Generate goal net structure to depict learning materials

Merge the transition functions into game system

Connect the user interface with goal net and trigger game scene

---

With the authoring interface Goal Net Designer (details in our previous work [8]), the authoring of learning content is structured based on learning goals (Fig. 1). Through specifying learning goals of the curriculum, teachers can deliver their ideas. All the learning goals can be drawn as round dots in Goal Net Designer which are defined as states in a Goal Net structure. The relationships between learning goals can be drawn as lines with arrow, which are defined as transitions of Goal Net. If the fulfillment of a learning goal needs several tasks, we can add a task list to the subsequent transition of the current goal state. The transitions with task list are presented as rounded corners rectangles.

The buttons for drawing Goal Net are listed on the left panel (Fig. 1A). Teachers can edit the learning contents through dragging nodes to the canvas. The dot with “S” is used to add states; the dot with “A” is for adding transition without task list; and the dot with “T” is for transition with task list. On the upper right panel all the used elements such as goal states, transitions, and tasks in transitions are listed in Fig. 1C with a catalogue as Fig. 1B. The corresponding properties of the activated element will be displayed in Fig. 1D. Users can edit the details through fill in the property blanks.

When teachers do the authoring, they can follow and repeat the two steps below.

1) **Identify the skeleton of learning content through setting learning goals:** To identify the underpinning educational purpose – what students are going to learn, the author should capture all the knowledge key points from the corresponding curriculum and represent the key points through drawing the hierarchy of learning goals. To be practical, teachers can follow two sub-steps.

   a) **Specify learning topic:** **Learning topic** refers to the knowledge correlation which divides learning materials into subjects and themes. Each learning topic is a learning goal which serves the overall goal, to teach the knowledge on “transport in plants”. The teacher can add the learning goals through dragging the round dot with “S” to the canvas, and edit the description through the property window in Fig. 1D. Then the teacher can add the transitions between learning goals through dragging the round dot with “T” to the canvas. At this time the system will pop up a editing window for entering the start and end dots of the transition. The whole structure of learning goals can be completed by repeating this process. As the dots with number 89, 82, and 83, the three learning topics in our example are: Xylem and Phloem, Root, Stem and Leaf; Osmosis and Diffusion; and Photosynthesis.

   b) **Depict each learning goal from different difficulty levels:** **Difficulty level** refers to the difficulty of a learning content which divides learning materials into different teaching sequence, such as concepts, examples, transferred situations, etc. For each learning topic, designers can edit its sub-goals according to the learning levels. Taking the sub-goal “Diffusion and Osmosis” (the node 89) as an example, the authors set three different learning levels – Basic Concepts, Concrete Understanding, and Operational Experiments (as the nodes 85, 86, and 87).

Once the setting from both learning topic and difficulty level has been finished, the system will provide the option to add “details”, which comes to the next step for game scenario design as the following.

2) **Fill up the detailed domain knowledge for each identified learning goals:** This process is achieved within the Goal Net transitions. Teachers can set the scenario plots and manage the game progress by designating the actions in transitions through double clicking the transition rectangle, and there will be a small editing window for user inputs. Currently, we support two means of detail input.

   a) **Upload Concept Maps related to the key knowledge:** Using Concept Map to present knowledge is an popular approach. The system supports teachers to upload the Concept Maps, and the agent system will revise the Concept Map to a blank-filling problem for students to use.
b) Design the detailed experimental procedure through editing task list: in our system the authoring of detailed design is organized as the editing of task list within the transition rectangle. An tasklist editing window can pop up if the user double clicks any rounded corners rectangle.

All these tasks in transition will be organized in a task library in game engine side. The game developers will do the program based the task lists to realize the game scenario in 3D virtual world. The build-in tasks will be recorded as functions in the task library, and prepare for reuse in the future. The detailed scenario implementation still highly depends on the game engine program, but with the help of Goal Net, educational experts can brief their ideas through repeating above procedures and deliver the domain knowledge related information to the system designer and game developers.

IV. DISCUSSION AND CONCLUSION

For the time being, the proposed system is under a small-scale testing to get indications of the system efficiency. We interviewed three teachers, two game designers and two program developers. Due to the small numbers of interviewees, the evaluation does not allow for statistically significant results. But as a formative evaluation, it is valuable to summarize the useful features and identify areas for improvement. Each interview lasted around two hours. The feedbacks are summarized as following.

<table>
<thead>
<tr>
<th>Advantages (from teacher side):</th>
<th>Areas for improvement (from teacher side):</th>
</tr>
</thead>
<tbody>
<tr>
<td>• The drawing with Goal Net is easy to learn and convenient to use.</td>
<td>• Providing more supportive guidance</td>
</tr>
<tr>
<td>• The graphical presentation of learning goals is clear and direct to depict the domain knowledge.</td>
<td>• Providing basic structures for teachers to choose and fill in the blanks rather than totally drawing by themselves</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Advantages (from game developer side):</th>
<th>Areas for improvement (from game developer side):</th>
</tr>
</thead>
<tbody>
<tr>
<td>• The structured presentation is similar to Object-Oriented design which assists programmers encapsulate different parts and functions into small groups.</td>
<td>• If the authoring advances to the game development, it is ok for teachers to design whatever they want; but if the authoring is for regular updating of learning content, teachers should consider existing 3D models and game logic. The system should provide the information.</td>
</tr>
<tr>
<td>• The strategy for function reuse can reduce workload of coding.</td>
<td></td>
</tr>
</tbody>
</table>

From the results we may find that the problems are focused on the tradeoff between the flexibility of user control and the setting of supportive constraints. On one side, more flexibility of user control may allow teachers to author the learning content with less rules and constraints. They may express their ideas more freely. But the cost is more workload for programmers and more expense on game maintenance. Meanwhile, some of the teachers who are not very familiar with the authoring interface may feel too few guidance or rules to follow. On the other side, more settings of constraints may hedge the author’s creativity in, but on the contrary the programmers can regulate the game with a well organization.

A way to solve the problem is to design a general framework as knowledge carrier. For example, in VS we have a mission system which is used to assign tasks for students to complete. The general framework for loading learning content can be set within the mission format, such as to complete a mission the student has to collect several stones with certain concepts, which need to be fully understood through successfully filling in all the blanks of a certainly designed concept map. The mission of finding “concept stones” is a general framework, but the concept maps embedded in with domain knowledge are authored by teachers. With the framework, the learning content updating can be easily achieved without a big change from the game side.

In this new version of VS, we involve teachers into the game design. Through authoring the learning content, teachers abstract the key points of the domain knowledge and make the educational features deliverable. The system developers can subsequently build up the game flow based on these learning goal threads. Meanwhile, the involving of teachers also provides a guarantee of the long term running of the virtual learning space, as teachers can continuously update the learning content and collect student data and feedback as a bonus. Then students can continuously learn new knowledge in the project. In our prospect, this type of teacher-student interaction will achieve a win-win situation.

REFERENCES


