We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Avatar Augmented Annotation Interface for e-Learning

Won-Sung Sohn[1] and Jae-Kyung Kim[2]

[1] *Department of Computer Education, Gyeongin National University of Education,*
Gyesan-Dong, Gyeyang-Gu, 407-753, Incheon, Korea
sohnws@ginue.ac.kr

[2] *Dept. of Information Science and Telecommunications, School of Information Sciences*
University of Pittsburgh
Ki187cm@gmail.com

Abstract

Creating avatar-augmented presentations that extend gesture and annotation to distance education is laborious and time-consuming. We present Avatar Augmented Annotation (AAA), a project to design and share web based e-Learning materials using an animated avatar and digital inking. Using AAA, the instructor can create a presentation, using free-hand annotations that include high-level teaching behavior without doing any programming. The AAA then generates a script expressed in XML and augments the presentation with an animated avatar following each annotation. We found that AAA provides positive educational effectiveness and usability compared to previous online courseware.

1. Introduction

Today e-Learning, web-based teaching (WBT), and distance learning techniques are common in university classrooms[1],[2],[3]. These approaches enable students to learn in their own spaces and times[2],[3]. However another issue offset this advantage: interactions between students and teachers provide more educational effect[2],[3],[7]. If online teaching systems can't provide any interactions, students become bored with online coursework. Many students who take online coursework print out all web materials as exam period approaches in our experiences.

To address this problem, some interaction techniques are proposed. Many commercial online coursewares provide video and animation clips that have advantages over static web courseware. Several recent systems provide a "cyber teaching assistant" to improve the educational effect of static WBT material. The main advantage of classroom instruction over distance education is that students can easily observe gestures and annotations performed in live instruction. Shindo[2] developed CTA (Cyber Teaching Assistant) for a programming class that includes an animated avatar and scenario markup language. CTA responds to student questions and calls using previously defined scenario files from a database. Ray[3]

makes a course that uses an animated agent to guide students and emphasize specific teaching material. Ray found that students overwhelmingly preferred the animated lecture courses over other online courses they had taken. Mash[4] generates a script to control an MS-Agent avatar[6] without requiring the user to do any programming. In Mash, the user simply chooses gestures, types in the accompanying script, and saves the file.

Surprisingly, avatars may be a better approach than live video[7] – for example, this approach enables an instructor to update a presentation over time, without needing to keep the same suit of clothes in the closet! However, a key limitation of such systems is that the content provider must build avatar movement and interaction into web material manually using a script or other programming language. Creating these avatar-augmented presentations is usually laborious and time consuming. Mash used a parameter box to control the avatar, in which the user enters the physical parameter in the 'X', 'Y' field even though it doesn't require programming. Mash generates script for MS product and mash's format which can't be shared in other system.

Annotation can minimize such effort. Annotation (digital inking) is an important feature of a digital lecture or presentation system[1],[9]. Digital inking can capture the knowledge between student and teacher and easily share and reuse material[1],[9]. Webtour[5] allow to web document designer augmenting the contents with digital ink. Webtour records a user's drawing and supports playback with web media. This scenario is useful and cost effective. But, Webtour can only synchronize annotation itself without any animated agent.

Thus we believe that avatar augmented annotation tool for web-based teaching, can enhance learning effects of distance education and usability. We present the avatar augmented annotation (AAA) interface, which supports a method to design teaching scenarios and augment them using an animated avatar in web based learning materials. The instructor first creates a presentation, and then marks up the presentation using freehand annotations such as underlining, circling, and selects teaching behavior such as introduce, exit which be done by animated avatar. The AAA then generates XML based script to control the avatar movement. Finally AAA augments the presentation with an animated avatar that follows the annotations sequentially, drawing attention to the content that the instructor indicated.

2. Overview of Avatar Augmented Annotation

The overview of AAA interface shown in Figure 1.

As shown in Figure 1, AAA includes an authoring interface to make a scenario with avatar and annotation interface to allow a student to add his or her own knowledge or perspective. On the authoring interface side, the key components are:

- •• Annotation recognizer to allow freeform drawing from scenario designer and determine its type.
- •• Script generator to control avatar movement and create avatar control markup language (XML) that includes high-level teaching behavior.
- •• Synchronizer to interpret the script and map avatar behavior to low-level avatar movement. Then synchronizer plays avatar lecturing on the HTML/XML document.

On the annotation interface side, the key modules are:

- •• Annotation recognizer to determine marking type such as line, circle, note annotation from the student.
- •• Annotation integrator to store marking as the markup language and print out annotation to the screen.

We describe each interface in detail in the following sections.

Fig. 1. Overview of AAA interface.

3. Avatar Annotation Markup Language

We define Avatar Annotation Markup Language (AAML) to express annotation and control animated avatar on web document. In AAA, instructor draws freehand annotations and adds comments. Then system creates annotation and avatar motion information represented in XML form. This approach allows users to more easily control the avatar behaviors, compare to previous efforts [2],[3],[8]. Creating scenarios using digital inking in the HTML/XML document requires that graphic information of the markings and context information of the original document be presented in the form of external links [9]. Through external linking, markings are saved independently from the original document and can be shared by multiple users. The overall structure of AAML is shown in Figure 2.

AAML includes 'meta' and 'annotationList' elements. A 'meta' element includes attributes for lecturer, lecture name, date and note. The annotationList element includes each annotation elements (Figure 3). Annotation has 'type', 'behavior', 'context', 'comment' and 'avatar' elements. The type element of an annotation expresses one of nine marking types, for example 'line', 'ellipse', 'highlight' and others. The behavior element describes a high-level motion set for describing avatar behavior in the teaching material.
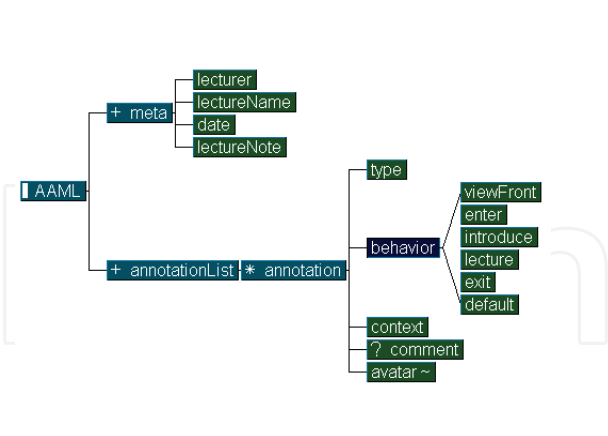
```
<annotationList>

  <annotation>
   <type areaSelect="false">
    <eiilipse color="16744576" page="1"/>
   </type>
   <behavior></viewfront></behavior>
   <context> <nonStructural>pop</nonStructural>
    <structural>
     <startingElement
      offset="138"path="/body[2]/chapter[0
      ]/p[3]/#text[0]"totalOffset="266"/>
     <endingElement
      offset="-43" ath="/body[2]/chapter[0]/p[3
      ]/#text[0]"totalOffset="264"/>
    </structural>
   </context>
   </avatar name="santa">
  </anntation>

</annotationList>
```

Fig. 2. Overall structure of AAML.   Fig. 3. An example of AAML document.

We define 5 kinds of behavior including 'viewFront', 'enter, 'introduce', 'lecture' and 'exit' behavior. If the instructor does not select any behavior, the system selects 'default' behavior. Context elements include 'nonstructural' and 'structural' sub elements. A non-structural context has target (anchor) text. A structural context describes a point and offset within the annotated part such as Xlink and Xpointer.

4. Authoring Interface

In the authoring interface, the instructor first marks on the web content and then selects the teaching behavior of AAML such as introduce, exit, and others. The system generates a teaching scenario with AAML and web contents. The interpreter analyzes scenario informa-tion saved as XML and maps high-level motion of teaching behavior to low-level motion of avatar movement. Lastly, the authoring interface augments the animated teaching avatar to follow the annotation.

Figure 4 and shows a scenario for a data-structure course lesson on simple stack operations. Our prototype is based on a Window XP and XML document. This supports 3D-based ava-tar motion renderer (Figure 4(A)) or 2D-based MS-Agent[6] (Figure 4 (B)) and we will de-scribe each interface in detail next.

Figure 4 shows a screen shot of marking procedure by the instructor. The instructor draws free-form annotation and selects teaching behavior defined in AAML. In Figure 4 (A), in-structor draws annotation on his interesting point. User draws marking on 'push' and 'pop' context with 'Viewfront' behavior selection in Figure 4 (B). He can add a comment that the avatar will pronounce when the avatar presents the lecture on the push and pop concepts.

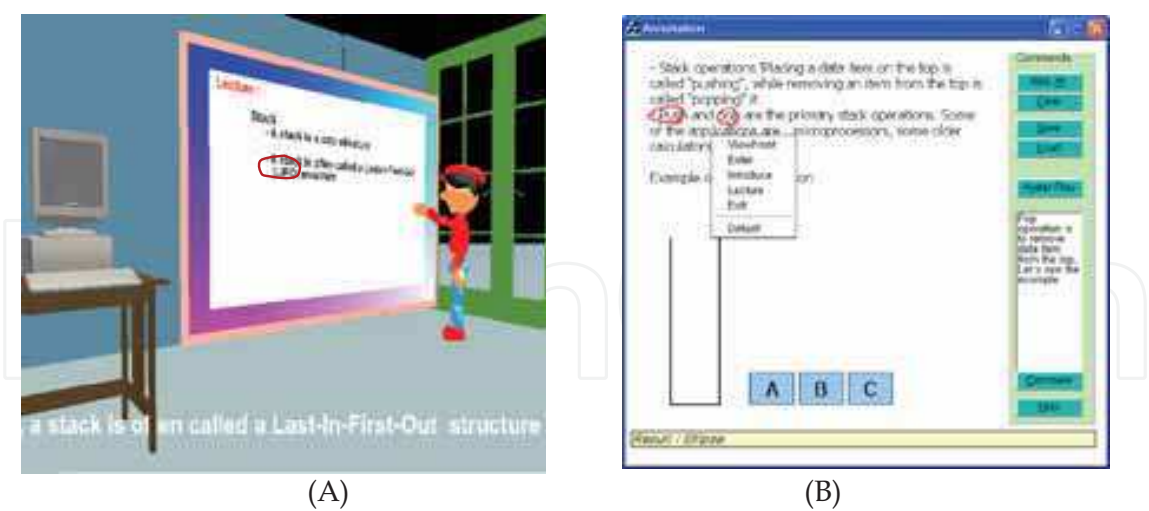(A)   (B)

Fig. 4. Screen shot of authoring interface.

The AAA system generates a scenario script based on AAML after the user pushes the save button. The system recognizes the free-form marking type and analyzes the context behind the marking, which consists of element and physical position.

Figure 5 shows the avatar augmenting procedure after the user pushes the 'avatar play' button. Subsequently the system interprets high-level behavior of AAML and sends it to the avatar motion generator to generate low-level avatar movement. The motion-generator traces the positions of annotation anchors and creates a physical path to each annotation, in order to play back the avatar animation. This approach enables the instructor to design a teaching scenario without having to know a script language or to understand the system architecture in detail.
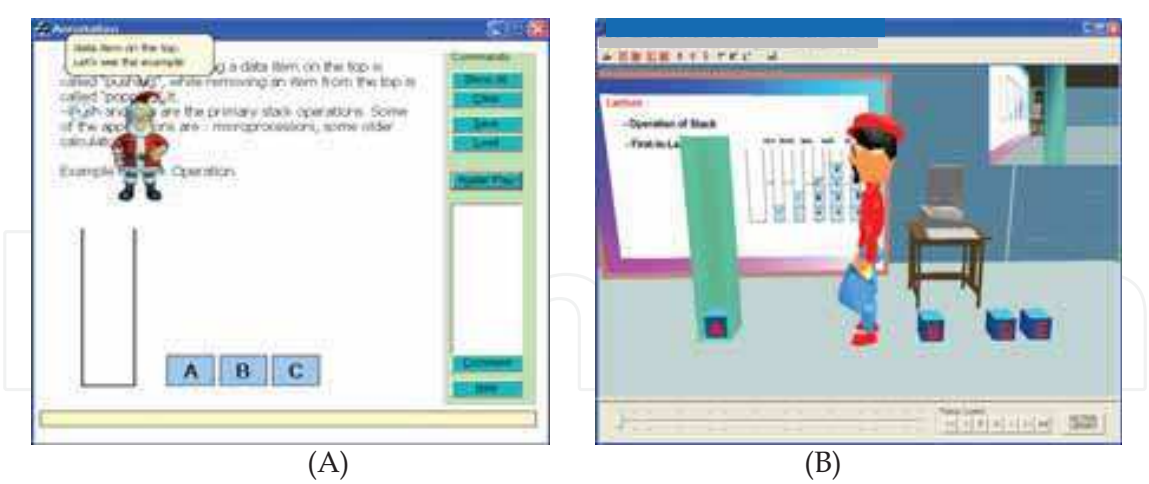

(A)   (B)

Fig. 5. Avatar presents 'push' and 'pop' concept.

AAA supports synchronization between an animated avatar and a multimedia object. In Figure 6 (A), instructor moves each box to show the concept of pushing an element onto the stack. After that, AAA synchronizes the movement between a box and an avatar using in-ternal rules (Figure 6 (B)).
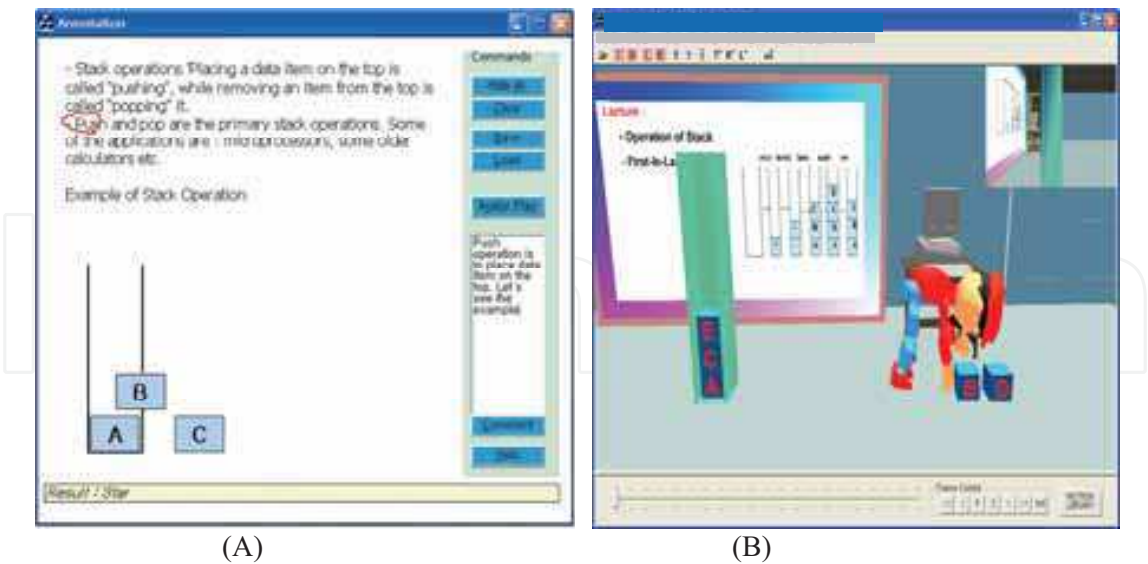
(A)   (B)

Fig. 6. An example of synchronization process.

5. Annotation Interface

AAA's Annotation interface is enables students to add questions, comments, critiques, and more using several annotation types. We used a context-based annotation engine[9] which include reorganization and integration modules for the HTML/XML document.
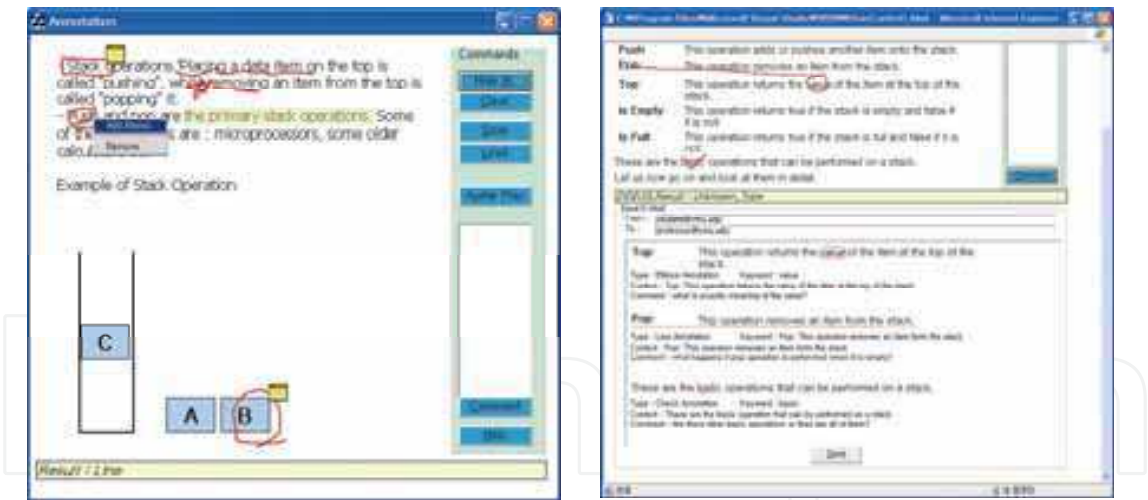

Fig. 7. Annotation Interface in AAA.

Fig. 8. Interaction techniques between instructor and student in AAA.

Figure 7 shows our annotation interface; students have added line, circle, and note annotations.

AAA provides interaction techniques that enable students to ask questions instructors to reply (Figure 8). To ask questions, students draw marks and type their question in a text input box. The system recognizes the marking type and extracts the context under the mark-

ing area to produce an image screen and send it to the instructor along with the question. Hence instructor doesn't need to check the annotation notes on the web site.

6. Summary of Usability Inspection

We conducted a usability inspection on our prototype using heuristic evaluation[11] with five instructors and seven university students. We built a small data-structure lesson using AAA to augment existing university online coursework[10]. We surveyed the instructor and students who had previously taken online data structure courses. The instructors were asked to design scenario. We demonstrated the AAA lecturing to the students and ask to add annotation. After the tests, a survey was done with 5 principles and 6 heuristics which include severity from 1 to 6. For items with severity less than 3, an empirical study was performed through separate survey. In the chapter, only the most important items from test results are summarized.

All five instructors commented that our approach was very interesting and they were satisfied with AAA's easy authoring interface. Four instructors suggested incorporating more real-world teaching behavior into AAA. One instructor pointed out if we would support synchronization functions between annotation and multimedia objects, it would takes too a great deal of time to build on existing material. For this reason, they recommended adding or substituting Flash or Java applets in AAA rather programming the synchronization.

All seven students preferred animated online courses to previous static web courses. Most students were satisfied that they could clearly understand the key point of the page they were shown. However some students disliked the necessity of animated teaching assistant in online courseware. They have to watch the avatar lecturing even though they can grasp the key point quickly. They commented that they would prefer to be able to choose watching the avatar or not.

7. Conclusions and Future Works

We have built an avatar augmented annotation (AAA) interface and investigated its effectiveness and usability. Our authoring interface enables an instructor to create a teaching scenario using digital inking annotation on 2D and 3B based web material. AAA generates a script for avatar lecturing which includes high-level teaching behavior. To do this, we define an avatar annotation markup language and support annotation techniques to lecture material. We found that AAA presents positive educational effectiveness and usability compared with previous static online courseware.

We acknowledge that there are significant technical issues with some of the features and methods as indicated from the user study. Our study applies to e-Learning, IETM (Interactive Electronic Technical Manual), Intelligent Tutoring System, and eBook. In future work we plan to develop a plug-in program to add our lecturing interface on existing web document.

8. References

Richard J. A., Crystal H., Steven A. Wolfman, and Ruth A.: A study of digital ink in lecture presentation. Proceedings of the Human factors in computing systems, Vienna Austria (2004) 567-574

Shindo, Y., Matsuda, H.: Prototype of Cyber Teaching Assistant. Proceedings of IEEE International Conference on Advanced Learning Technology (ICALT2001), IEEE, USA (2001) 70-73

Ray A.: Animating Web Lectures with Agent Technology. http://jurist.law.pitt.edu/ lessons/lesfeb01.htm.

Mash, http://www.bellcraft.com/mash.

Chellury R. S., Darrin P. L., and Arturo P.: Webtour: a system to record and playback dynamic multimedia annotations on web document content. Proceedings of the seventh ACM international conference on Multimedia (Part 2), Orlando Florida(1999) 175-178

MS Agent, "http://www.microsoft.com/msagent"

Elisabeth A., Thomas R. and Jochen M., :WebPersona: A Life-Like Presentation Agent for Educational Applications on the WWW. Proceedings of the workshop "Intelligent Educational Systems on the World Wide Web", Kobe Japan(1997)

Kshirsagar, S., Guye-Vuilleme, A., Kamyab, K., Magnenat-Thalmann, N. D., Thalmann, E., and Mamdani,: Avatar Markup Language", Proceedings of the Eurographics Workshop on Virtual Environments (EGVE'02), Spain(2002) 169-177

Sohn, W. S., et al.: Context-based Free-form Annotation in XML Documents. In International Journal of Human Computer Studies, Vol. 59. 3. Academic Press, (2003) 257-285

YSCEC, http://yscec.yonsei.ac.kr/

Useit.com: Jakob Nielsen's Website http://useit.com.

E-learning

Edited by Maja Jakobovic

E-Learning is a vast and complex research topic that poses many challenges in every aspect: educational and pedagogical strategies and techniques and the tools for achieving them; usability, accessibility and user interface design; knowledge sharing and collaborative environments; technologies, architectures, and protocols; user activity monitoring, assessment and evaluation; experiences, case studies and more. This book's authors come from all over the world; their ideas, studies, findings and experiences are a valuable contribution to enriching our knowledge in the field of eLearning. The book is divided into three sections. The first covers architectures and environments for eLearning, while the second part presents research on user interaction and technologies for building usable eLearning environments, which are the basis for realizing educational and pedagogical aims, and the final last part illustrates applications, laboratories, and experiences.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Won-Sung Sohn and Jae-Kyung Kim (2010). Avatar Augmented Annotation Interface for e-Learning, E-learning, Maja Jakobovic (Ed.), ISBN: 978-953-7619-95-4, InTech, Available from: http://www.intechopen.com/books/e-learning/avatar-augmented-annotation-interface-for-e-learning

INTECH
open science | open minds