

An ITS Authoring Tool: Experimental Advanced Instructional Design Advisor

Carol Luckhardt Redfield, Ph.D.

Mei Technology Corporation

8930 Fourwinds Dr., #450
San Antonio, TX 78239
(210) 655-8911
carol@meitx.com

Abstract

The eXperimental Advanced Instructional Design Advisor (XAIDA) is one of the intelligent tutoring system (ITS) authoring tools presented at the AAAI 1997 Fall Symposium. This paper briefly describes the system including the Deliver (student mode) and Develop (authoring mode) components. Four shells are described although some exist only in a preliminary prototype. XAIDA is best used for creating tutors to teach maintenance of physical devices. Maintenance training is split into teaching the physical characteristics of the device, the theory of operation of the device, the procedural operations of the device, and troubleshooting problems with the device. Each of these shells is discussed below.

Overview

Wouldn't it be great to have an automated instruction machine that could take someone's expertise and automatically generate a computer-based course or an intelligent tutoring system (ITS)? This paper describes one such system that will be demonstrated at the American Association of Artificial Intelligence's 1997 Fall Symposium on ITS Authoring Tools. The system described here is called the eXperimental Advance Instructional Design Advisor (XAIDA¹), an ITS authoring tool developed by Mei Technology through funding from Brooks Air Force Base. XAIDA has been developed as a part of the AIDA (Advanced Instructional Design Advisor) program that is studying the overall usability and effectiveness of the types of tutors that are produced with XAIDA.

An intelligent tutoring system looks like a computer-based training (CBT) system in some respects, but extends the traditional CBT with some ideas and techniques from artificial intelligence. Typically in an ITS, there is an interface component, a model of the expertise, a student model and an instructional module. The instructional

module communicates to the interface what is to be shown or done next in conjunction with the expert and student models. The tutor is then responsive to an individual student's performance.

Examples of XAIDA's interface for each component are shown later in this paper. The student sees presented material and has practice sessions within a lesson. The expert and student models are contained in database tables where the structure of the knowledge varies according to the shell or component of XAIDA being used. The instructional module presents material in the order in which the instructor describes and provides practice of the material based on the student's performance.

Compared to many other existing ITS authoring tools (see other papers in this Symposium; Frasson, Gauthier & Lesgold, 1996), XAIDA is very easy for a subject matter expert to use for creating courseware and does not require any programming knowledge nor a programmer to support the development. A developer need only know about the subject matter to be taught and how to use a computer for running tools and applications. In looking at other similar tools, it appears that the more knowledge that is required of the user or developer, the more sophisticated the tool and resulting tutor, as one might expect.

XAIDA

It was determined early on in the AIDA project that a good way to teach maintenance of physical devices is to consider the concepts to be learned in four separate but related components (Spector, Polson & Muraida, 1993). The physical characteristics of a device should be taught and understood first, followed by the theory of how the device operates. Then operational procedures can be learned, completing with troubleshooting methods. Also determined early on, XAIDA has been developed by first programming and testing the student portion called Deliver and then the authoring or developer portion called Develop. It was anticipated that a Deliver and Develop for

¹ XAIDA was developed under Air Force funding, Contract No. F1624-93-C-5002.

each of the four shells would be developed. Although two shells have been completed, the other two exist as a combination of prototypes and design documents.

Basically a developer sitting at an XAIDA interface inputs expertise about a device or topic, filling in templates and using various XAIDA editors. With the information provided by the developer, XAIDA produces an intelligent tutoring system that teaches that expertise. The effectiveness of these lessons has been shown in numerous studies, and lessons of many and varied topics have been developed, including ones that were not intended to be the type produced by XAIDA such as customer satisfaction and parabolic equations (Wenzel, Half, Richardson, & Gibson, 1996).

The following sections describe the current components of XAIDA including Deliver and Develop for Physical Characteristics, Deliver and Develop for Theory of Operation, Deliver for Procedures, and Deliver for Troubleshooting. The Develop portions of each shell have been designed to be a what-you-see-is-what-you-get interface, so the Deliver portion shows what Develop would look like with some additional menu options, tool bars, and editors.

Physical Characteristics

Both Deliver and Develop are fully operational for this component. The physical attributes and characteristics of a device are easily and quickly input into the Develop component. The data is stored as a tree structure of parts of the device with media and facts attached to each part as needed. The developer must provide a picture or background graphic for each part that has parts underneath

it. The student is asked questions about the location of parts, the part hierarchy, and facts about parts. This practice is done after each set of subparts.

Deliver of Physical Characteristics for a coffee maker is shown below. The student can use the Next and Back buttons to navigate or select an item in the Contents or Outline windows. The student can also click on a part inside the Portrayal window and go the portion of the lesson on that selected part.

Coffee Machine

File GoTo Help

Portrayal

Reservoir

Filter Basket

Switch

Coffee Pot

The capacity of the Coffee Machine is 2 to 12 cups.

Press Next to view Parts.

Back Next

Outline

- Coffee Machine
 - Switch
 - Reservoir
 - Filter Basket
 - Coffee Pot

Contents

- Coffee Machine
 - Coffee Video
 - capacity
 - Parts

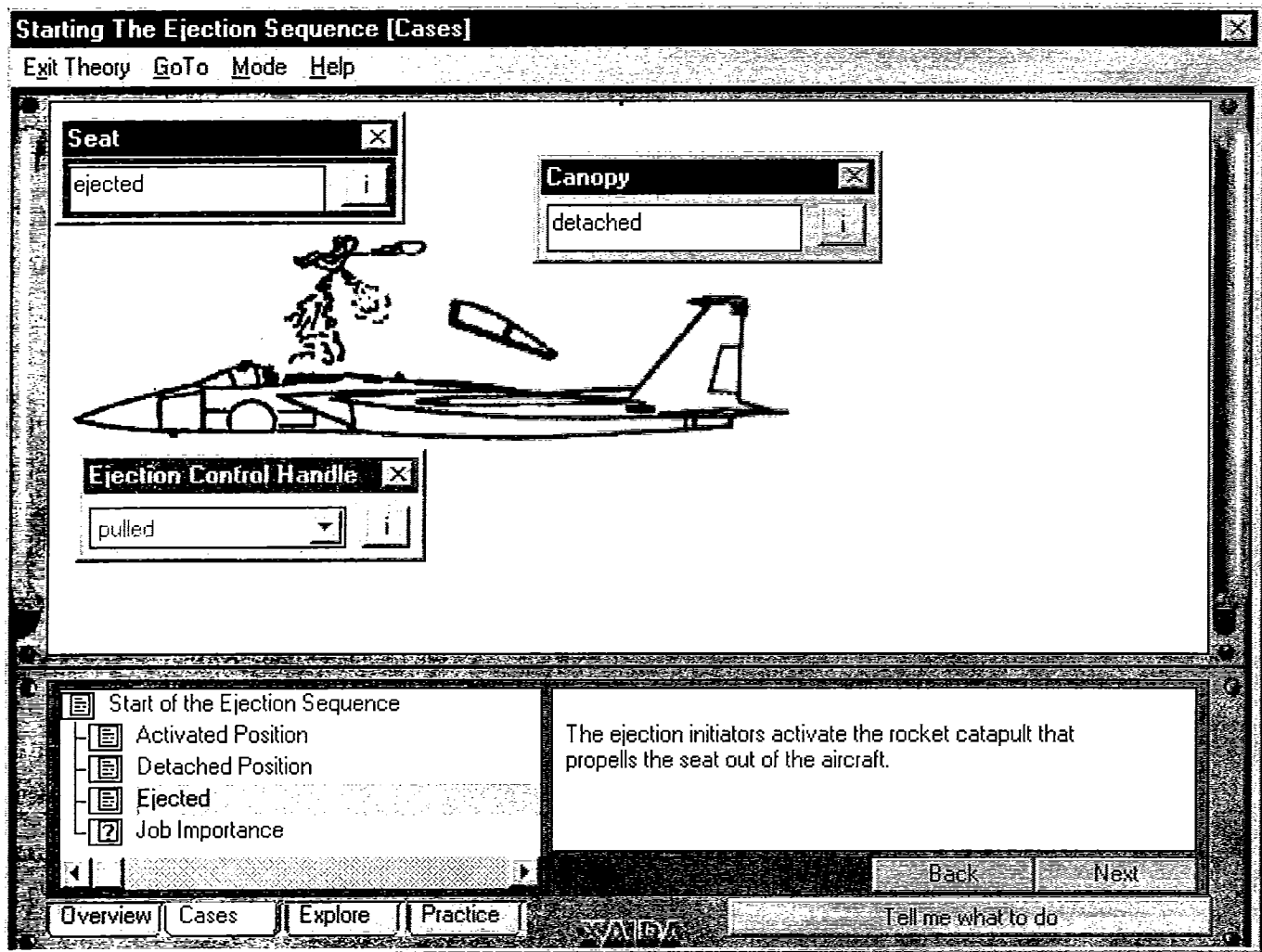
XALPA

Theory of Operation

The theory of how a device works and operates is taught with a theory of operation lesson. This component also exists in a fully developed Deliver and Develop form. The knowledge for the Theory of Operation is stored as rules of how the values of variables or attributes relate with specified cases. The student sees the cases determined by

the developer, and the student can manipulate the values in what is called a user-defined case. The learns the interactions and relationships of the operations of a device by studying these cases and associated rules.

One screen of Deliver of a Theory of Operation lesson on an ejection seat in an Air Force aircraft is shown below.



Procedures

Procedures are represented as a series of steps with associated information such as pictures, mistakes, tools, mishaps, notes, warnings and cautions. A developer inputs the correct steps in addition to typical or common problems that arise in performing the procedures, both those caused by the user and those resulting from outside forces. A lesson consists of presenting and practicing the steps of the procedure and answering questions. The

student's goal is to perform the entire procedure correctly. Practicing a step consists of selecting an object, an action, and an optional tool. Potentially a simulation can be put in place of this simplistic functional simulation.

Currently, the Deliver portion of Procedures is in prototype form and the Develop portion is only partially designed on paper. Below is an example of the Deliver portion of Procedures.

The screenshot shows a software window titled "Solving a Story Problem". The window has a menu bar with "File", "View", "Go", and "Help". Below the menu bar is a toolbar with icons for "Title Page", "Back", "Next", "View Menus", "Font", "Search", "Print", and "View Manual".

The main content area is divided into two columns. The left column contains a "Procedure Overview" section with links for "Description", "Equations", and "View Procedure". Below this is a "Lesson Contents" section with links for "Introduction", "Procedure Overview", and a numbered list of five steps: "1. draw a picture", "2. set up variables", "3. find equations", "4. solve the equation", and "5. answer the question". There is also a link for "Potential Problems".

The right column contains the main problem text: "Find the length of the side of a square that has a perimeter of 24 cm." Below the text is a diagram of a square with a label "24 cm" next to its top side. Underneath the diagram is the text "s = length of the side". To the right of the diagram is the text "formula for perimeter of a square is P = 4s where s is the length of the side". Below this is a series of equations: "24 cm = 4s", "4s = 24 cm", "4s/4 = 24/4 cm", and "s = 6 cm". At the bottom of the right column, the text "The length of the side is" is followed by "6 cm" which is circled.

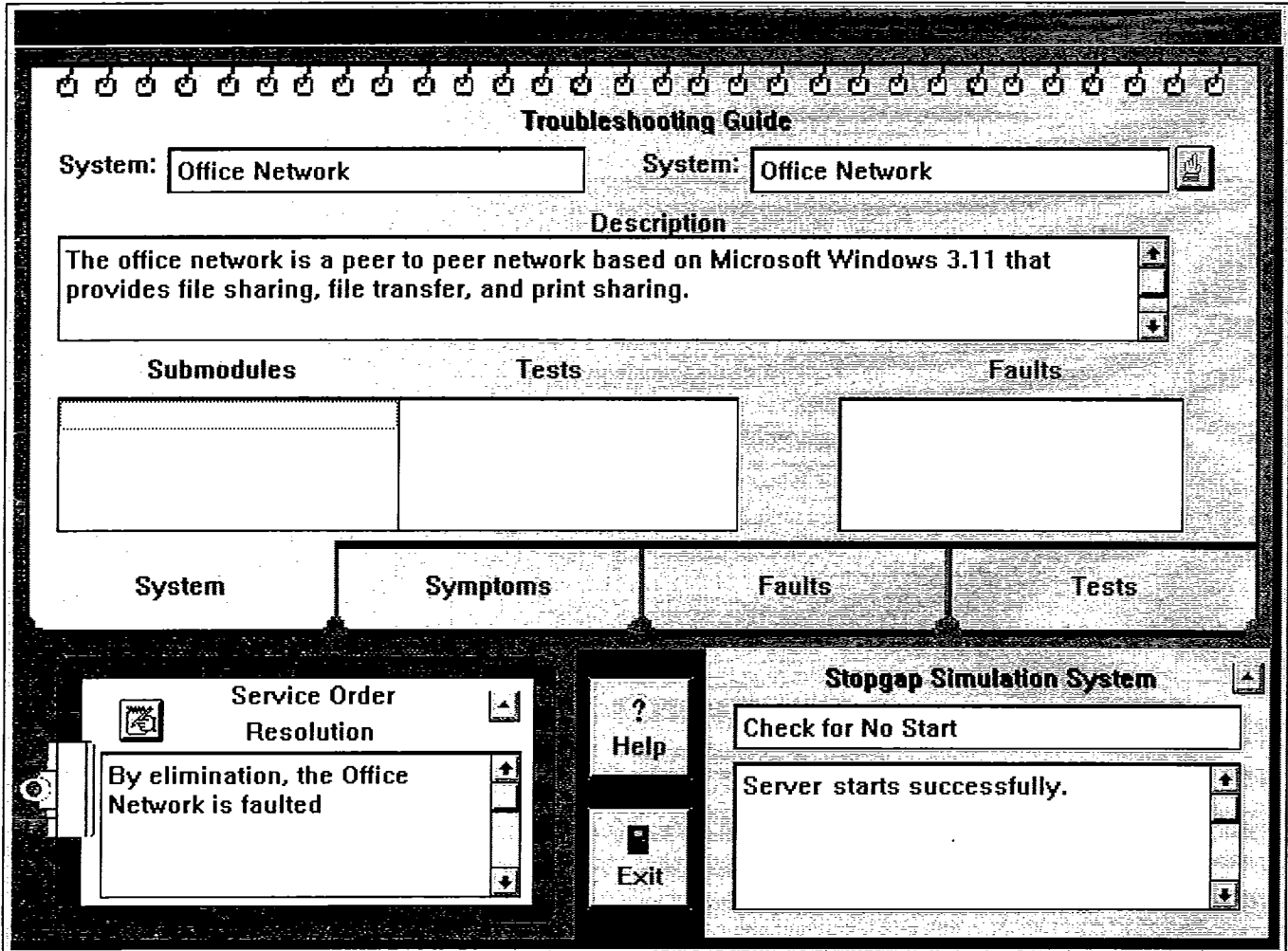
At the bottom of the window, there is a "GO TO" button with an arrow pointing to the "Equations" link.

Troubleshooting

The Troubleshooting shell teaches a student how to do fault isolation. The knowledge of the expert is stored as a fault tree and how to traverse that tree. The instructional module presents new and previously seen faults as a student progresses through a troubleshooting lesson. A student moves between viewing the Service Order, Troubleshooting Guide, and Stopgap Simulation. The

student isolates and repairs single faults. Advice and error correction is provided immediately based on the student's actions.

This troubleshooting shell exists as a very primitive prototype where a database can be manually entered to be presented as a lesson to students. The Develop component has not been fully defined.



Conclusions

XAIDA has proven to be effective at quick development of useful ITS material. Many subject matter experts have used the system and been surprised at how fast they can develop lessons. In many cases, finding and putting together the graphics and other media such as sound, video, and animation takes up much of the time in creating a course. Organizing and inputting the content required is accomplished in a short period of time especially compared to current method of CBT and ITS development. Many courses have been developed in a matter of a few hours (Walsh, 1994).

References

- Frasson, C., Gauthier, G., & Lesgold, A. (Eds.). 1996. Intelligent Tutoring Systems: Third International Conference Proceedings. ITS '96. Berlin: Springer-Verlag.
- Spector, J.M., Polson, M.C., & Muraida, D.J. (Eds.) 1993. Automating Instructional Design: Concepts and Issues. Englewood Cliffs, NJ: Educational Technology Publications.
- Walsh, W.J. 1994. Automated Authoring: Some Preliminary Results, 16th Interservice/ Industry Training Systems and Education Conference, Orlando, FL
- Wenzel, B.M., Halff, H.M., Richardson, K., and Gibson, E.G. November, 1996. Assessing the instructional effectiveness of the Experimental Advanced Instructional Advisor (XAIDA). Interim technical report for U.S. Air Force Armstrong Laboratory.