AL-TP-1991-0006

ARMSTRONG LABORATORY

A PILOT STUDY OF THE NAMING TRANSACTION SHELL

Ann Marie Canfield

Utah State University
Logan, UT 84321

J. Michael Spector

Department of Computer Science
Jacksonville State University
Jacksonville, AL 36265

HUMAN RESOURCES DIRECTORATE
TECHNICAL TRAINING RESEARCH DIVISION
Brooks Air Force Base, Texas 78235-5000

June 1991

Interim Technical Paper for Period August 1990 - March 1991

Approved for public release; distribution is unlimited.

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235-5000

NOTICES

This technical paper is published as received and has not been edited by the technical editing staff of the Armstrong Laboratory.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Office of Public Affairs has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.

DANIEL J. MURAIDA
Contract Monitor

HENDRICK W. RUCK, Technical Director
Technical Training Research Division

RODGER D. BALLENTINE, Colonel, USAF
Chief, Technical Training Research Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** June 1991	**3. REPORT TYPE AND DATES COVERED** Interim Technical Paper - August 1990 - March 1991

4. TITLE AND SUBTITLE

A Pilot Study of the Naming Transaction Shell

5. FUNDING NUMBERS

PE	-	62205F
PR	-	1121
TA	-	10
WU	-	66

6. AUTHOR(S)

Ann Marie Canfield
J. Michael Spector

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Utah State University
Logan, UT 84321

Department of Computer Science
Jacksonville State University
Jacksonville, AL 36265

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

Armstrong Laboratory
Human Resources Directorate
Technical Training Research Division
Brooks Air Force Base, TX 78235-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

AL-TP-1991-0006

11. SUPPLEMENTARY NOTES

Armstrong Laboratory Technical Monitor: Daniel J. Muraida, Ph.D. (512) 536-2981

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200-words)

The many hours required to develop quality computer-based instruction make it difficult to produce instruction that is both current and adaptable. Transaction Shells (Merrill, Li, and Jones, 1990) provide novice instructional designers and computer users with tools to aid in the design and delivery of meaningful and easily modifiable courseware while dramatically decreasing the development time. This study was an initial pilot study designed to evaluate the effectiveness of the Naming Transaction Shell developed by M. David Merrill and Zhongmin Li at Utah State University. The major finding was that the predicted improvement in development time did occur. The software was tested at the Air Force Academy in Colorado by a subject matter expert with little computer and instructional design expertise. The software was found to decrease development time and produce instructionally sound courseware. Recommendations from the subject include that future versions of the software contain more feedback parameters. Student record management, a feature not yet implemented, was strongly encouraged to be included in the next version.

14. SUBJECT TERMS

computer-based instruction
courseware authoring
instructional design
transaction shell
transaction theory

15. NUMBER OF PAGES 32

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

TABLE OF CONTENTS

LIST OF FIGURES

PREFACE

The many hours required to develop quality computer-based instruction make it difficult to produce instruction that is both current and adaptable. Transaction Shells (Merrill, Li and Jones, 1990) provide novice instructional designers and computer users with tools to aid in the design and delivery of meaningful and easily modifiable courseware while dramatically decreasing the development time. This study was an initial pilot study designed to evaluate the effectiveness of the Naming Transaction Shell developed by M. David Merrill and Zhongmin Li at Utah State University.

The Armstrong Laboratory and Dr. Merrill have signed a Memorandum of Agreement wherein Merrill allows AL/HRTC use of the software for purposes of formative evaluation. Test sites include USAFA, Lowry AFB, Maxwell AFB, and Randolph AFB.

This initial study was conducted at USAFA. Dr. Mary Marlino and Maj. Milt Neilsen provided valuable assistance with design and implementation. Capt. Kevin Crenwelge proved to be a cooperative and insightful subject.

We wish to thank all the USAFA personnel who helped make this study possible. In addition we appreciate the support of Col. Ballentine, and Drs. Dan Muraida, Scott Newcomb, and Henk Ruck at AL/HRT. We also wish to thank Universal Energy Systems for partial support of the Naming Transaction Shell Pilot study under Air Force Office of Scientific Research Contract Number F49620-88-C-0053.

SUMMARY

The development and production of computer-based instruction is a time-consuming and difficult process. Dr. M. David Merrill and colleagues at Utah State University have developed a Transaction Shell approach to developing computer-based instruction (Merrill, Li, and Jones, 1990). A Transaction Shell approach provides designers with tools that decrease the development time. This study was an initial pilot study designed to evaluate the effectiveness of the Naming Transaction Shell developed by Merrill and Li. The study was conducted at the Air Force Academy in Colorado Springs. The Transaction Shell approach parallels the Air Force's objective of designing an Advanced Instructional Design Advisor (AIDA) and in the future may be incorporated into the AIDA system. The major finding of the study was that the predicted improvement in development time did occur. The software was found to decrease the development time and produce quality computer-based instruction. Recommendations that future versions of the software be able to track and record students progress were noted. Limitations in the software were also recorded.

I. INTRODUCTION

The need for corporate, military and industrial computer-based training (CBT) continues to rise. As a result, courseware developers and managers are confronted with the time-consuming aspect of the CBT development process (Faiola, 1989). Programming, debugging, and testing courseware is such a time-consuming aspect that efficient courseware development is an immediate concern (MacKnight and Balagopalan, 1988-89). The time required to develop one hour of computer-based training has been estimated to take anywhere from 200 to over 6000 labor hours (Carter, 1990; Lippert, 1989). These numbers make it difficult and expensive to produce current and timely CBT.

The Armstrong Laboratory is in the process of assessing the feasibility and effectiveness of developing a Transaction Shell approach to courseware authoring (Merrill, Li and Jones, 1990). This approach involves providing subject matter experts with tools to aid them in the design and delivery of effective, reusable and easily modifiable courseware. Transaction shells provide a novice designer with the ability to produce effective computer-based instruction (CBI) while dramatically decreasing the development time. Transaction Theory is a relatively new and innovative approach to courseware authoring develcped by Dr. M. David Merrill and colleagues at Utah State University. The Navy has incorporated a somewhat similar idea in the Computer-Based Educational Software System (CBESS) (Wetzel and Wulfeck, 1987). Tennyson also has used a transaction-shell-like approach in Multi-International Language Instruction Module (MILIM) (Goldenberg and Turnure, 1989). However, Merrill's Transaction Theory provides a basis for generating a comprehensive library of transactions appropriate to courseware delivery (Merrill, Li and Jones, 1990).

The Armstrong Laboratory also has, as a long range goal, the development of an Advanced Instructional Design Advisor (AIDA). AIDA is a courseware authoring advisor which will assist and guide military trainers through the process of effective computer-based instructional design. AIDA will take established theories of knowledge, learning, and instruction and incorporate the theories into course authoring (Muraida & Spector, 1990). The use of the Naming Authoring Transaction Shell is a first step in the implementation of the theories evolving in the AIDA research effort.

Merrill and Li at Utah State University have developed a prototype Naming Transaction Shell for designing and delivering courseware for naming the parts and functions of a device. This transaction shell provides the user, a subject matter expert, with an environment which contains all of the relevant instructional strategies and expertise. The user enters the appropriate content and the programmed strategy configures and delivers the instruction. Default parameters provide for delivery of the instruction with appropriate instructional values. However, the author can manually override the defaults for individual cases. For instance, the author may want to limit the response time during practice to five seconds instead of using the default setting, which in this case is learner control.

The Naming Transaction Shell was the focus of this initial pilot study. The purpose was to determine the effectiveness of the environment and the feasibility of incorporating the approach into the AIDA project. Results from this pilot study will be used by AFHRL to determine whether further and more comprehensive research should be planned in this area. The results indicate that Transaction Theory is a promising technology.

II. OBJECTIVES OF THE RESEARCH EFFORT

This study represents the first in a series of studies evaluating the usability of Merrill's transaction shell technology in Air Force technical training settings. In order to establish the kind of data gathering techniques appropriate for subsequent studies, this initial study was confined to a single transaction shell, the Naming Transaction, and a single subject author, an Air Force Academy navigation instructor assigned to the study less than half-time over a two week period.

Specific questions addressed by this study include the following:

1) What length of time should we expect a novice to require to become adept at using transaction shells?
2) What specific training will be required to instruct users about authoring with transaction shells?
3) What length of time should we expect a novice to require to author a simple nomenclature lesson module?
4) What critical times and factors should be observed in subsequent studies?

The Armstrong Laboratory is interested in determining whether this type of system is a viable alternative to traditional courseware authoring (e.g., Merlin, ISS, TenCORE, QUEST, etc.), and whether this technology should be incorporated into the AIDA project.

III. METHODOLOGY

This initial study was designed to have one subject matter expert design and develop instruction for a portion of an aviation course using the Naming Transaction Shell software. Broader and more conclusive studies will be scheduled based on the outcome of this initial study. The experiment took place at the Air Force Academy in Colorado. The instructional goal was to teach the names and functions of the instruments on the T-37 instrument panel. A two week period was allotted for the development of the instruction, with the subject matter expert working approximately half-time on the project. The subject was observed during his development time. The subject also kept a personal record of his thoughts and observations (see Appendix A). At the end of each week, during the two week evaluation, a video-taped interview also recorded the subjects progress and thoughts.

The study was designed to test the usability, generalizability, executability and productivity of the Transaction Shell approach to courseware authoring. The development time required to produce courseware, both on-and off-line, and developmental obstacles were observed and recorded during the study. Observations were made with regard to the following:

1) How easily the user could adapt to the transaction shell environment.
2) The perceived versatility and utility by the user.
3) The user's assessment of student and instructor acceptability and user difficulties with the product.
4) Recommendations from the user for future versions of the software were also noted.

The software, Authoring Naming Transaction, was designed for use by subject matter experts who are novices in the area of CBI design. The user imports a graphic representation of the device to be taught. The individual parts are then identified and labeled. The function of each part is also defined by the user. At this point, the module can be delivered to the student since all of the remaining parameters of a naming lesson contain default values. Figure 1 illustrates a sample student lesson screen.

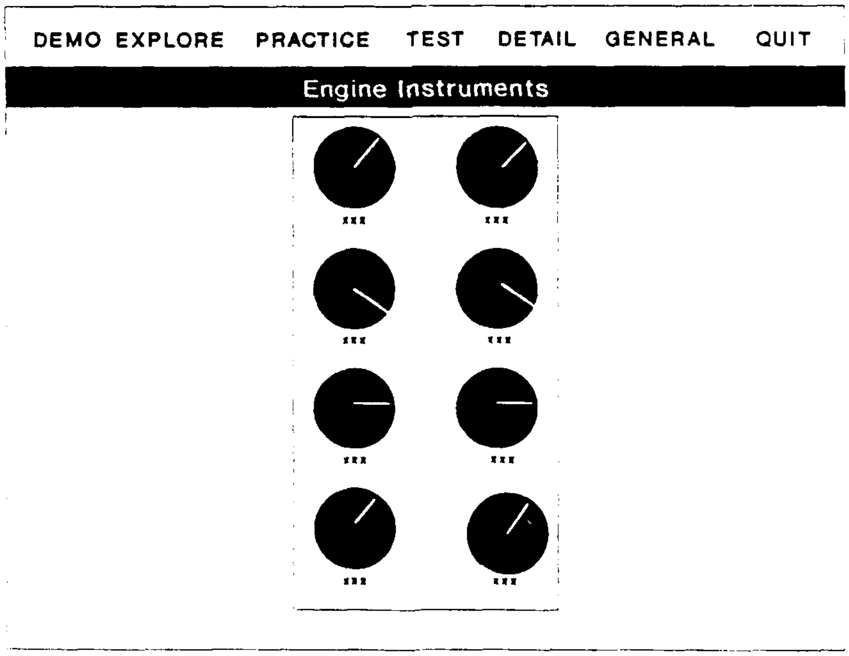
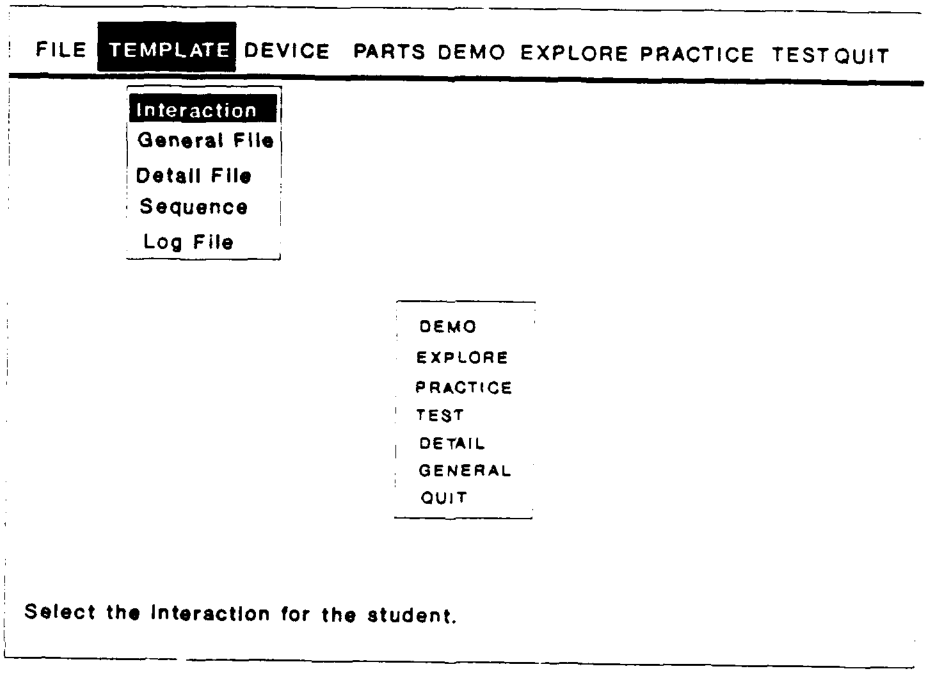


FIGURE 1. Student Screen

Default values are provided in order to automate delivery. A user, however, may desire to customize a lesson for a particular device or audience. The user can manually set all of the delivery parameters using the pull down menus. The software allows the user to select which student interaction modes to include in the delivery of instruction (see figure 2). In some cases, testing may not be required, or the instructor may not want students to interact in a particular mode.



Select the Interaction for the student.

FIGURE 2. Author mode

Interaction parameters for the student during delivery can be modified and include the following:

Interaction	Description
Demo	System presents all of the parts and functions.
Explore	Allows the learner to control the presentation sequence of parts and functions.
Practice	The learner is presented a series of inquisitory instances with feedback.
Test	Learner is tested and scored on parts and/or functions names and locations.
Detail	Moves the learner to a more detailed lesson of a particular part.

General	Moves the learner to a more general or overview lesson.

Additional values the designer may wish to manipulate are the timing (learner control vs. timed response) or sequence values (i.e., part name followed by function, part only, function only, or function followed by part name) for the Demo, Explore and Practice options of delivery. There are also performance feedback parameters which can be modified. In the Test mode, the designer can select the sample size and criterion for success (e.g., 75%, 80%, 90% correct). The sample size refers to the number of times the student will be queried on each part during the test mode.

The modules can be linked together by way of general and detail files at any point during development. The user enters the file name to which the current file should be linked by a general or detail file. 'General' refers to moving up the hierarchical associations, for example an overview lesson. 'Detail' refers to lower in the hierarchical association and may include a close-up view of a particular part with more 'details' nested beneath it. Linking detail files to a general file allows for lesson modularity.

The background of the subject was of particular importance in this study. The subject was selected based on his expertise in the subject matter. Captain Kevin Crenwelge was an instructor for the AV480 course which teaches the T-37 instrument panel used in this study. The subject was a computer and instructional design novice, which was the desired target audience for the study. His computer experience was limited to basic word processing. His instructional experience consisted of his having taught the course in a classroom setting on several occasions. In addition he had developed the course materials currently being used in the classroom. He is presently responsible for several other AV480 instructors.

IV. RESULTS

At the end of the first week of work, the subject had completed four mini-lessons designed to teach and explore various aspects of the Authoring software (see Appendix B). This introduction to the system took approximately five hours. The subject had also completed nine and one-half hours of off-line design and development. This portion of the experiment included designing what would be taught, how the material would be grouped and a handwritten documentation of all design decisions. The subject relied on his own knowledge and Technical Order 1T-37 B-1 (T-37B Flight Manual) to determine lesson content. At the end of the first week of the experiment, the subject had dedicated fourteen and one-fourth hours to the project.

The second week of the study began with completing the hand written design of the lesson content, which required two and one-half hours. The subject then selected the graphics he wanted developed for his instructional modules (this process will be detailed in section V). With the written design and development completed, the on-line authoring began. After fourteen and one-fourth hours of on-line development time, the subject had completed the entire lesson package as designed. Twenty individual picture files and lesson sub-modules had been designed with ten detailed groupings which teach 125 parts and their functions.

The package was ready for delivery to students after a total of 31 hours of training, design and development time. A complete CBT package was completed in 31 hours by a subject matter expert with little instructional design or computer expertise. The package included the delivery of the instruction with testing capabilities.

Figure 3 shows the ratio of on-line to off-line development time. See Appendix C for summaries of development time.

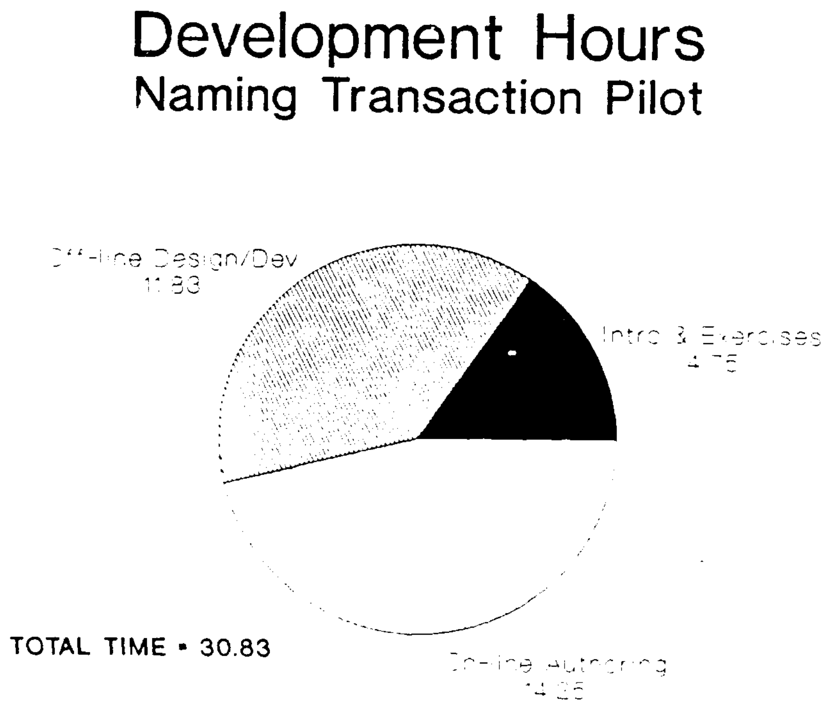


FIGURE 3. Proportions of Developmental Time

Individual module development time ranged from 5 to 25 minutes and included approximately five minutes of debugging or correcting time. Modules ranged in size from 2 parts to 10 parts (see Appendix D), ten being the maximum allowed by the system.

Linking the files together was also completed within the individual module development time.

Figure 4 illustrates development time in minutes versus the number of parts in an individual module.

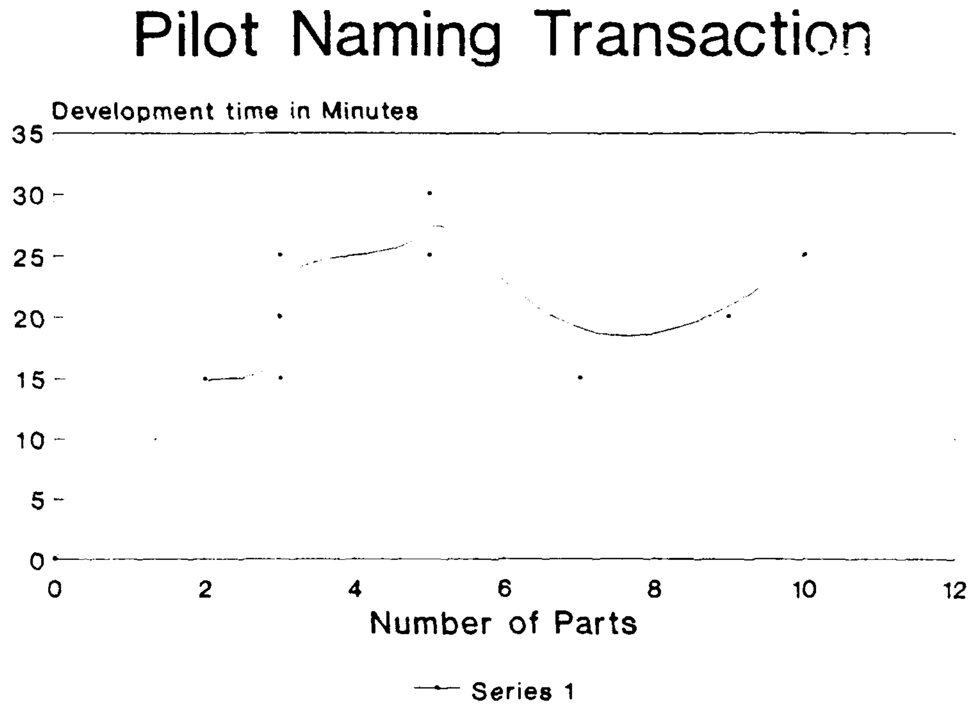


FIGURE 4. Number of Parts vs. Developmental Time

The student instructional time developed is estimated to be in excess of three hours of instruction. Actual data are currently being collected to confirm this estimate. Development time was 31 hours. The development to instruction ratio, therefore, is approximately ten hours of development per hour of student instructional time.

A reasonable comparison for development time per hour of instruction is 200 hours under traditional software development methods (Lippert, 1989). Many authors cite development hours into the thousands (cf., Carter, 1990). The 200:1 ratio, from the low end of the scale, is used for comparison because we are omitting the graphics production on both ratios. The Transaction Shell development time is a significant decrease from the reasonable comparison of 200 hours identified with traditional software development. Although the ratio of 10:1 does not include graphics production, it does include selecting and directing the graphics development. Future versions of the Transaction Shell approach will put the development of graphical materials within the hands of the targeted user population.

After completing an introduction to the system the subject was enthusiastic. He could see many applications for this type of software within the Academy. He felt the environment was easy enough that anyone would be able to use it and was excited to see what he could produce with the system. Traditional course development (non-computer) was much more laborious, he stated, and traditional CBT development was beyond the scope of his abilities. This new approach, according to the subject's observations, seemed more efficient and would improve student performance as well as motivation. There was some apprehension as to whether this approach to instruction would eliminate the jobs of some instructors.

At the end of the experiment the subject remained enthusiastic. He felt that anyone could learn to use the system both in design and delivery modes. Traditional course development (non-computer instruction) for the same material, in the subject's estimation, would have required at least two times the amount of development time. With the computer environment, he felt that the students would have higher motivation. The instructor could ensure that a proficiency level had been reached before the student entered the simulator which would, as a result, decrease the amount of time in the simulator. It was difficult under the non-computer system to ensure that students were studying and learning the material in the text. This often resulted in wasted effort in the simulator. By ensuring that the students knew the components and functions before they arrived, simulator time could be reduced and more students would be provided opportunities for training.

The subject enjoyed the time he had spent developing the instruction and was enthusiastic to share his accomplishments with co-workers and supervisors, who also had immediate positive reactions. Apprehensions about the computer taking over the role of instructors had been dropped as the software was now viewed as a tool to enhance the role of the instructor and would make their time more efficient. From the comments of supervisors, it was apparent that they were interested in the finished product. Every course taught in their division had a portion which could use this approach to instruction and benefit from the naming transaction. They were anxious for the next phase of the study and wanted to be included in the study. These reactions support the direction of AIDA research and development and indicate that further development of the Transaction Shell approach is warranted.

The default parameters for delivery seemed appropriate to the subject in most cases. He varied the parameters only slightly. The subject prefered to use timed presentations in the Practice interaction rather than the default of learner control. He also elected to modify the Testing parameters. He prefered a sample size of 3 and criterion level of 75% as opposed to the defaults of 2 and 90%. Interaction parameters were modified as appropriate to individual lessons. For instance, an overview module was designed to aquaint the learner with what lessons were available on the system. This module did not require testing, fo the Test interaction was disabled. Appendix D shows a hierarchy of the lesson modules the subject created.

V. SOFTWARE LIMITATIONS

Several limitations in the software were discovered and recommendations were made for future versions. The software had been developed as a prototype for demonstration purposes only. This study provided an opportunity to identify vital factors necessary for refining the robustness of the program for future studies. There were some problems with occasional mouse failure. This obstacle needs to be resolved before implementation can occur. The subject recommended that, in future versions, the author be given options to include different types of feedback during the test mode. At present there is no feedback during testing, only during the practice mode. Student record management was initially a feature planned for implementation in the next version. The subject strongly encouraged this addition which he felt would enhance the software. This feature will be implemented in the next version. In addition, the software is currently being modified to include data collection of time on tasks for both authors and students.

Limitations on descriptor size often made the development more difficult than necessary. Part names had a maximum length of 30 characters and the subject often had difficulty abbreviating the names of parts to fit this limitation. Part names often consisted of five or more labels. The function description was limited to five lines and this was thought to be occasionally restrictive. Instructional principles prescribe limiting the amount of knowledge placed in short term memory at any one point. Good instructional principles have been purposely built into the Transaction Shell so that instructionally sound CBT will be developed without the user having to be knowledgeable in the area of instructional design. However, the military technical orders the subject worked from to generate the lesson material dictated longer descriptors than would normally be recommended. This conflict can be addressed by allowing larger descriptors than are recommended with defaults that encourage good instructional practices.

An additional discovery which prolonged the development was that each time a detail or general file is linked, the current file must be saved. Errors surfaced when all links within one file were made at one time. The subject found a solution to the problem, which added time to the development process. The subject's problem-solving approach in dealing with this problem, demonstrated his easily acquired knowledge and understanding of the system. This problem can be eliminated readily through more thorough preliminary system training and checks within the software to assure that the instructions are followed.

Several limitations in the graphic identification of parts were also discovered which may necessitate changes in the future. Some instances require identifying parts that are inside of other parts. Information nested for identification was not accessible during the Explore mode. However, this information could be accessed during the Demo, Practice and Test modes. The obvious, immediate solution is to omit the Explore interaction in these cases.

Identifying parts by location presented a difficulty for the subject when he wanted to group items that functioned similarly but were located in separate areas of the screen. This limitation was overcome by grouping items by area instead of by function.

The subject was concerned also with how a student would know when all of the detail files available on the system had been completed. A student, if not careful, might miss some of the additional files. The subject felt some way of identifying what the student should accomplish or a map of where the student was in the system would be an important feature.

Graphics development was a notable problem during this experiment. The subject directed the graphic development. However, he was kept at a distance from these problems as they were not the focus of the study and could easily be corrected by specialists. The plan was to have the subject direct a photographic session of the instrument panel for the instruction. The next step was to scan the pictures into the computer and have them available for the subject. The scanner software however, presented unexpected difficulties. This resulted in many of the graphics being hand rendered by the design team.

The graphics obstacles should and will be addressed in future versions of the study and software development. Recommendations include a different software package for the scanner and/or a different file format for the transaction software. Either alternative may eliminate the graphics problem. Digitized photographs may also be an alternative. The real problem with graphics involves the lack of genuine graphic standards for PCs.

VI. CONCLUSION

The conclusion of this experiment is that the Transaction Shell environment is an appropriate alternative to courseware authoring and further studies should follow. Transaction Shells will implement AIDA's objectives effectively. The Transaction Shell provides an efficient and effective tool for subject matter experts who have little instructional design or computer expertise and produces quality CBI. Our subject, Captain Crenwelge, became an expert user in a matter of hours. He remained enthusiastic throughout the experiment and was anxious to see more developments and implementation of the system. Development time was decreased by a factor of 10 using the authoring system developed by Dr. Merrill. Quality courseware could be efficiently produced using a Transaction Shell Environment. Excellent suggestions for future development and enhancements of the system also emerged, as expected.

VII. RESEARCH RECOMMENDATIONS

It is recommended that the transaction shell approach to courseware authoring receive further attention and research. After correcting the minor mouse and graphics problems, it is suggested that an experiment be conducted that involves more subjects, including both instructors and students, and a variety of content domains. Factors that should be evaluated in future studies include learner outcomes, learner performance and courseware effectiveness. Traditional instruction and instruction generated by use of the Naming Transaction Shell for this study should be examined for their effect on the length of time required in follow-up simulator sessions.

Worksheets were used to help the subject in this study plan and organize his lessons (see Appendix E). He noted that the worksheets were very beneficial. Automation of the worksheets would be an appropriate follow-up project which would advance the AIDA concept further. The worksheets and the exercises were developed as the experiment progressed. Future experiments should allow for job-aid and training development preceding the experiment, with ample time for revision in order to produce higher quality job-aids and training. A future study should also include the Checklist Procedure Transaction Shell currently being developed at Utah State University for the Air Force Human Resources Laboratory.

REFERENCES

Carter, J. (1990). The ICW Decision Handbook. Randolph AFB, TX: Headquarters, Air Training Command Technical Report in press as AFP-50-xx.

Faiola, T. (1989). Improving Courseware Development Efficiency: The Effects of Authoring Systems on Team Roles and Communication, Educational Technology, Vol. 29(8), pp. 16-19.

Goldenberg, T.Y. & Turnure, J.E. (1989). Transitions between short-term and long-term memory in learning meaningful unrelated paired associates using computer based drills, Computers in Human Behavior, Vol. 5, pp. 119-135.

Lippert, R.C.(1989). Expert systems: tutors, tools, and tutees. Journal of Computer-Based Instruction, Vol. 16(1), pp. 11-19.

MacKnight, C.B. & Balagopalan, S. (1988-89). Authoring Systems: Some Instructional Implications, Journal of Educational Technology Systems, Vol. 17(2), pp. 123-134.

Merrill, M. D., Li, Z. & Jones, M.K. (1990). Limitations of first generation instructional design. Educational Technology, Vol. 30(1), pp. 7-11.

Merrill, M.D., Li, Z. & Jones, M.K. (1990). Second generation instructional design (ID2). Educational Technology, Vol. 30(2), pp. 7-14.

Muraida, D.J., & Spector M. (1990). The Advanced Instructional Design Advisor (AIDA): An Air Force Project to Improve Instructional Design. Educational Technology, Vol. (30)3, pp. 66.

Wetzel, D. & Wulfeck, D. (1987). Computer-Based Educational Software System. San Diego, CA: NPRDC TR-87-25.

<u>APPENDIX A</u>

SUBJECT DIARY

6 Aug 90

Watched demonstration of author and student program. After lunch returned and ran exercise of setting up a basic lesson. Very straight forward. Few minor glitches - mouse - but no big deal. Ran easy and learned a lot by trial and error. The longer I worked with it, the less errors there were. In student Run under Test (or?practice?) need some feedback to let the them know what questions they got wrong and what the correct answer was. As of present only give total % at end of test.

Worked small amount on picture grouping and breakdown.

7 Aug 90

Detail file - everything came back and lesson 2 was smooth. Worked on groupings for photo's - only allowed 10 groups. <u>Difficult</u>, if want to group items by similar function, etc. - ie. fuel system or elect pwr supply system!! Decided to group by sections since it is only to learn the switches and functions not all the interactions.

Got photo session done. Problem shooting a link simulator but wanting to teach the real T-37 cockpit layout. Decided to cut and paste the photographs to get the real look of the T-37 instrument panel.

8 Aug 90

Exercise in detail files. Was confused at first in how to tie all the files back together. Was frustrating when I did get it correct, the machine had a glitch that would call up the wrong detail file.

9 Aug 90

Start on new exercises, Good!! Must remember to SAVE. Made more sense - easy to get around.**

Worked on layout of lesson plans - name files, functions, putting detail files together.

Feels good to have something concrete on paper.

10 Aug 90

Laid out more lesson plan files.

Was interviewed.

13 Aug 90

Finished lesson plan files - finally!!

Started putting one lesson file onto computer. Picture needs to be better.

14 Aug 90

More lessons into computers - losing lines on the function - tried to repair by modify but would not work - had to redefine the entire piece. Many mouse errors.

Seems 5 lines in the limit to the function code.

15 Aug 90

Finished 1st level files. Small problem when label boxes overlap. In Explore mode it is difficult to call some of the boxes up because the boxes are hidden in larger boxes. Fix by changing the sizes of boxes and making smaller overlap.

No mouse problems.

16 Aug 90

Tie my first level files into the one main general T-37 cockpit file and then work on make the 2nd level detail files. Problems were encountered when trying to link the main general file to the detail files, it wouldn't always take and when it did most of the time the wrong picture would be displayed in the detail file. Found the correction by saving after connection each detail file. Takes more time but works.

Definitely need the worksheet to keep everything organized (names of files, which ones linked to each other) also helps when wanting to EDIT the program. You have the text in front of you.

APPENDIX B

AUTHORING NAMING TRANSACTIONS

EXERCISE #1

1. The point of this lesson is to gain some familiarity with AUTHOR'ing transactions. Use AUTHOR to creat a lesson calls LESSON1 which uses the MADAR.PIC file to teach four parts. You can make up the part names, locations and functions.

2. Enter START to load graphics driver.

3. Enter AUTHOR to start author mode.

4. select FILES, then CREATE
   use LESSON1 as coursename
   use MADAR.PIC as picture file
   use 4 parts

REMEMBER: FREQUENT SAVES!

AUTHORING NAMING TRANSACTIONS

EXERCISE #2

OBJECTIVE - Additional practice with AUTHOR, especially using a detail file and altering default parameters.

1. Create LESSON2 using MADAR for the top level picture file and CONTROL for the detail file:

```
START
AUTHOR
FILE CREATE LESSON2
4  parts - 1 of which is the keyboard
        in the lower rignt corner
```

2. Limit student interactions to EXPLORE, PRACTICE, TEST AND DETAIL (AND QUIT).

3. Set EXPLORE timing to 3 seconds.

4. Set TEST sample size to 2 and criterion to 80%.

(REMEMBER - FREQUENT SAVES!)

QUESTION: When you reference the DETAIL file CONTROL, what assumptions were being made? What was the implication of using CONTROL as a detail file for the keyboard?

AUTHORING NAMING TRANSACTION
EXERCISE #3

OBJECTIVE: Additional familiarization with AUTHORing transaction shells -- especially using a GENERAL file, a DETAIL file, and additional control of default parameters.

1. Create a file called LESSON3 which will serve as the general file. Use MADAR for the picture file. LESSON3 will have three parts.

2. Identify two detail files -- one for the keyboard area (bottom right corner) which already exists in a file called CONTROL, and a second for the left half of the MADAR device which already exists in a file called MADAR2.

LESSON3 has 3 parts (2 of which have detail files) -- one detail of the keyboard (bottom right corner), another detail of the on/off switch panel area (left half of the picture), and a third part of your choosing (there will be no detail file for the third part).

3. Set the interactions for LESSON3 to include EXPLORE, PRACTICE, TEST, DETAIL, and QUIT. Set parameters for EXPLORE. PRACTICE, and TEST as you see fit. Save and test your work frequently.

4. Edit CONTROL and provide a GENERAL file called LESSON3. Change other parameters as you like.

5. Edit MADAR2 and provide a GENERAL file called LESSON3. Change other parameters as you like.

6. Run LESSON3 in the student mode -- at the DOS prompt enter:

```
NAME LESSON3.DAT
```

Test the lesson as if you were a student. Be sure to try DETAIL when in LESSON3. Also try GENERAL when in one of the DETAIL files.

7. Terminate the student mode when you are satisfied and run AUTHOR to edit LESSON3 if you want to make changes.

8. QUESTION: Describe what DETAIL and GENERAL files are and how they can be used.

9. Record any observations or unanswered questions on the back:

AUTHORING NAMING TRANSACTION
EXERCISE #4

OBJECTIVE: Additional familiarization with AUTHORing transaction shells -- especially using a GENERAL file, a DETAIL file, and additional control of default parameters.

In this Exercise you will create two lesson files, LESSON4 and DETAIL1. Create each module and the appropriate links with the information provided. You may wish to create each of the modules first, and then make the neccessary links to connect them.

FILE NAME: DETAIL1
Picture: Control
Number of Parts: 5
General File: LESSON4

FILE NAME: LESSON4
Picture: Madar
Number of Parts: 4
Detail File: DETAIL1

Change the parameters as you like for each module.

APPENDIX C

Pilot Naming Transaction Study
Summary of Development Hours

6 Aug 90		
	Overview and Exercises	2 hrs 15 min
	Offline Design & Development	2 hrs 15 min
7 Aug 90		
	Exercises	40 min
	Offline Design & Development	2 hrs 40 min
8 Aug 90		
	Exercises	45 min
9 Aug 90		
	Exercises	1 hr
	Offline Design & Development	2 hrs 15 min
10 Aug 90		
	Offline Design & Development	2 hrs 10 min
13 Aug 90		
	Overview	5 min
	Offline Design & Development	2 hrs 30 min
	Online Author	35 min
14 Aug 90		
	Online Author	6 hrs 10 min
15 Aug 90		
	Online Author	3 hrs
16 Aug 90		
	Online Author	4 hrs 30 min

TOTAL HOURS = 30 hrs 50 min

APPENDIX D
LESSON HIERARCHY

GENERAL<-------------------------------------->DETAIL

	ENGINE INSTRUMENTS 10 parts & functions	Tachometers 2 parts & functions		
	FUEL PANEL 7 parts & functions	Fuel System Switch 2 parts & functions		
	NAVIGATION INSTRUMENTS 10 parts & functions			
	COMMUNICATION AND AC 9 parts & functions	IFF Transp 5 parts & function	UHF 7 parts & func	NAV/DME 9 parts & func
COCKPIT OVERVIEW 10 PARTS				
	POWER SWITHCES 10 parts & functions	Starter Switches 3 parts & functions		
	WARNING LIGHTS 5 parts & functions	Heading Cut Out and Fast Slave 3 parts & functions		
	GEAR AND LIGHTS 10 parts & functions	Nav lights 3 parts & functions		
	INSTRUCTOR PILOT PANEL 7 parts & functions	Oxygen Regulator 5 parts & functions		
	LEFT CONTROL QUADRANT 7 parts & functions	InterPhone Control Panel 3 parts & functions		
	CENTER CONTROL QUADRANT 8 parts & functions			

APPENDIX E

LESSON MODULE PLANNING SHEET

FILE NAME ____________

DEVICE NAME ____________

NUMBER OF PARTS ________

PICTURE FILE NAME ________

Circle the desired options below.

TEMPLATE

Interaction

Demo Explore Practice Test Detail General Quit

General File ________________

Detail File(s) ______________________________

Sequence (sequence parts for presentation):

____

DEVICE

Background (0-15)____

DEMO

Sequence

Lab->Func

Mixed/Sep

Func->Lab

Mixed/Sep

Lab Only

Func Only

Time

0

Sec___

EXPLORE

Sequence

Lab->Func

Mixed/Sep

Func->Lab

Mixed/Sep

Lab Only

Func Only

Time

0

Sec___

PRACTICE

Format

L/P

F/P

P/L

Present Mode

Mixed/Sep

Timing

0

Sec___

Feedback

Rt&Wr

Score&Timer

Score only

Timer only

Spelling

Approx

Ext Lab

Ext Func

Ext Both

TEST

S Sz_____(2)

Crit___(90)%

Names of Parts	Function	Key Words &=and,!=or