TRAINING DEVELOPMENT SOFTWARE TOOLS TO SUPPORT SYSTEM ACQUISITION MISSIONS

Chia-jer Tsai, Ph.D.
Southwest Research Institute
San Antonio, Texas

ABSTRACT

Development of the training system for a major defense system generates a tremendous workload for system acquisition and training professionals. Inadequate management and development of the supporting training system hinder the deployment and operation of a defense system significantly. The armed services and the software industry have developed software tools to assist system acquisition and training professionals. This paper reports the results of evaluating selected tools including GTET (Guidelines for Transportable Education and Training System), TRACES (Training Cost Estimator Systems), TASCS (Training Analysis Support Computer System), and JS ISD/LSAR DSS (Joint Service Instructional Systems Development/Logistic Support Analysis Record Decision Support System). The functions of the tools and considerations for making adoption decisions are described. A global comparison matrix of the tools' functions is presented, and recommendations for improving the tools are provided.

ABOUT THE AUTHOR

Dr. Chia-jer Tsai is a Senior Research Scientist in the Training Systems and Simulators Department of Southwest Research Institute (SwRI). He has worked extensively in needs assessment and technical training development with the Air Force and the manufacturing industry. Dr. Tsai received Ph.D. and M.S. degrees in instructional systems from Florida State University and a B.S. in electrical engineering from the National Taiwan University.

TRAINING DEVELOPMENT SOFTWARE TOOLS TO SUPPORT SYSTEM ACQUISITION MISSIONS

Chia-jer Tsai, Ph.D.
Southwest Research Institute
San Antonio, Texas

INTRODUCTION

The information in this paper is from an evaluation conducted as part of a Southwest Research Institute (SwRI) project to revise the current Air Force instructional systems development (ISD) process. A major premise identified by the revision is that ISD is critical to the success of defense system acquisition. Without effective and concurrent ISD effort, adequate training would not be available when a defense system is ready for deployment. This deficiency would hamper the performance of the defense system.

The complexity of modern defense systems often demands complex training programs and results in long and complicated acquisition processes. The defense systems usually undergo a number of changes during the development process, thus mandating corresponding changes in the training systems being developed simultaneously. These factors make development of the training systems a time-consuming, iterative process which is difficult to manage.

Several Air Force agencies have developed training development software tools. Four of these tools, because of their potential to facilitate system acquisitions, were selected by the sponsors of the SwRI project for evaluation. The tools are GTET (Guidelines for Transportable Education and Training System), TRACES (Training Cost Estimator Systems), TASCS (Training Analysis Support Computer System), and JS ISD/LSAR DSS (Joint Service Instructional Systems Development/Logistic Support Analysis Record Decision Support System).

The software tools were developed for IBM-compatible personal computers. GTET was developed with SuperProject Expert and 1-2-3, and TRACES was developed with Excel. These application programs are required to run GTET and TRACES, respectively. TASCS and JS ISD/LSAR DSS are compiled programs and run independently.

The evaluation focused on both software performance and utility. The latter includes training development support functions, user friendliness, and documentation. The report of the results of evaluating a tool is divided into four sections—functional purpose, target users, description, and adoption and modification considerations of a tool.

EVALUATION OF THE SOFTWARE TOOLS

GTET (Guidelines for Transportable Education and Training System)

Purpose - To assist in the conversion of existing resident courses into transportable versions.

Target Users - Training managers.

Description - The GTET system consists of three components: the GTET model, the GTET Management Information Support System (GMISS), and the GTET Cost Analysis Support System (GCASS).

The print-based GTET model provides a hierarchical set of diagrams that depict conversion project activities. The Top Level Process Model shows the five sequential project phases: determining feasibility, evaluating design alternatives, producing the prototype course/lesson, producing and finalizing course material, and transition.

The next level of the model is expansion of the phases into functions. The last level of

diagrams further expands the functions into tasks. Detailed guidance for managing project activities is provided for the tasks.

GMISS is a generic file used with the commercial project management software, SuperProject Expert, to produce a GTET project schedule. Based on decisions made with the GTET model, SuperProject Expert is used to modify the default data in GMISS and produce project schedules and resource allocation reports.

GCASS is a Lotus 1-2-3 worksheet containing macro commands. Input tables are provided for general activity and three educational activities: transmission, presentation, and evaluation. Users enter their instructional plan data into GCASS. Up to five alternative plans can be entered in a single file for analysis and comparison. GCASS then calculates the cost estimates using default cost factors which can be modified by users. GCASS output includes worksheets and graphs of life cycle cost, annual cost flow, and cost drivers.

The scope of the GTET program is limited to the individual course level. The course must be classroom-oriented, i.e., the program is not applicable to training complex psychomotor skills. Research and development of specialized or unique hardware or of new educational/training techniques is also beyond the scope of GTET.

Considerations - Users of the tool must be knowledgeable and skilled in ISD, training project management, SuperProject Expert, and Lotus 1-2-3. Because it is unlikely that users will have all of these prerequisites, time investment in learning is required. The learning curve for the GTET model is steep because it is paper-based; the software version evaluated does not provide a tutorial or an on-line version of the model.

The guidance included in the GTET model consists of abstract statements of principles and procedures. More examples or, better still, working templates should be provided to increase the usefulness of the model.

The GCASS costing model is unsophisticated. Its advantage is that it does not require a great deal of data. Its output is not very accurate, however, and can only serve for general guidance.

The internal working logic of GCASS is not transparent and is not documented, which precludes understanding of and confidence in the costing model. Many users may also want to modify the logic to suit their own methods or local environment. Providing modifiable logic will increase the level of user acceptance.

TRACES (Training Cost Estimator System)

Purpose - To support resource allocation decisions by examining life cycle costs of various media mixes for a proposed course of instruction.

Target Users - Training managers and developers.

Description - Similar to GCASS, TRACES is a spreadsheet costing model for training projects. The model is more detailed than GCASS; therefore, more preparation and data are required to use the tool, and the output should be more accurate. The user interface is well designed. Users are prompted by dialogue boxes to enter data. The mouse and Windows' graphical user interface (GUI) are standard parts of the user interface.

Users enter course-related and site-related specifications into TRACES modules: main, personnel, equipment, supply, and facility. These modules feed data into the report generator module.

Considerations - TRACES appears to be a well-conceived and well-designed program. Unfortunately, the original document is not very structured, and the manual for the Excel version is not available yet. A good manual and tutorial will increase the usefulness of this sophisticated spreadsheet program significantly.

Similar to GCASS, TRACES' internal working logic is not transparent and is not documented. Improvements in this area will further improve the program's usefulness.

TASCS (Training Analysis Support Computer System)

Purpose - To help instructional designers perform Air Force-related training design.

Target Users - Instructional designers.

Description - Unlike the on-line tools of GTET and TRACES, TASCS is an instructional design tool. It uses a database-oriented approach to build databases of tasks, training objectives, media, and units/lessons. This process is reflected in the main menu which provides choices of tasks, objectives, media, and syllabus analyses.

The default data contained in this tool are clearly developed for the Air Force. The first steps in using the tool are creating a task database and then entering task statements and numbers that indicate the task positions in the instructional hierarchy.

After the job tasks are identified, they are analyzed with a set of characteristics which are also used in objectives, media, and syllabus analyses. Examples of these characteristics are level of proficiency, area of operation, reason for difficulty, mission criticality, and frequency. Although the characteristics are fixed, users can modify the characteristic values, which are specified for the tasks in the characterization analysis.

Most values of task characteristics for tasks lower in the instructional hierarchy can be automatically collected for higher-order tasks. Once the tasks are characterized, a database of tasks for which training is necessary must be extracted from the master task database. TASCS provides a filtering rule to perform the extraction automatically. The rule uses task characteristics such as criticality and difficulty for criteria. Users can also make up their own rules. The tasks selected for training are then used to create an objectives database.

Steps in the next stage, objectives analysis, are similar to those in task analysis. Task statements become behavioral statements of corresponding objectives. A hierarchy number field, a condition field, and a standard field are also provided for each objective record.

The characteristic values of tasks to be trained are carried over to corresponding objectives. These values can be edited at this time, and two new characteristics, instructional methods and evaluation methodology, are provided for objectives.

Most characteristic values for lower-order objectives can also be collected for higher-order objectives. After the objectives are characterized, they are ready for syllabus analysis.

The third item on the main menu, media analysis, does not have to be specific to a program. The purpose of the analysis is to give media attributes weights which are used in an automated selection of media for training. For each medium, users can assign attribute weights to each characteristic's value. A weighing factor of 0 to 5 indicates the degree of appropriateness of a medium for a characteristic's value. For example, a weight of 5 for CBT against "knowledge—forming associations" means CBT is very effective for teaching association formation.

Syllabus analysis is the last phase of the TASCS process. Each objectives database can be converted into (at most) nine syllabus databases. A lesson is created for each second-level objective. Lessons supporting the same first-level objective form a unit.

Users can request automated media assignment to the lessons. TASCS does this by matching media attribute weights with each objective's characteristics. A primary medium and an optional medium are identified for each lesson.

TASCS generates various tasks, objectives, and syllabus reports to aid the instructional design process. The final product, the syllabus report, lists such items as objectives, time to train, media, instructional methods and strategies, evaluation methods, and proficiency levels.

Considerations - Besides being an information organization tool, TASCS automatically performs three functions for instructional designers: it keeps track of the hierarchical structure of objectives, selects tasks that require training, and selects training media.

Although certain default data in TASCS are modifiable, many factors are fixed. For example, the number and type of characteristics cannot be changed. Therefore, the tool is most useful for people who use the instructional design model underlying the program. The usefulness for others is limited. Since the characteristics are designed for Air Force operations, Air Force trainers may find the tool particularly useful.

Hierarchical analysis is an important aspect of TASCS, but its text-based user interface does not support the analysis well. This task is best supported by a user interface that supports drawing of hierarchical block diagrams.

TASCS does not allow characteristic concurrency to be maintained easily. For example, modifying an objective's characteristics does not update corresponding task characteristics automatically because the objective database is a derivative of the task database. This limitation is a significant drawback of the tool since the design process is iterative.

The user's manual is clear and understandable. However, explanations and examples of the underlying instructional design model are missing. Also, the manual furnished was for version five of the software, whereas the current software is version seven.

JS ISD/LSAR DSS (Joint Service Instructional Systems Development/Logistic Support Analysis Record Decision Support System)

Purpose - To support service-specific ISD methodologies and integrate training system development with other defense system design activities. This PC-based multi-user system's automated LSAR-to-ISD data interface permits training system development to maintain concurrency with the evolving defense system.

Logistic Support Analysis Record (LSAR) contains design and logistics information for a defense system. Logistic Support Analysis (LSA) is the iterative process that regularly updates the system's design and supportability information through all phases of acquisition.

Target Users - Training development managers and instructional designers.

Description - The fundamental design of the DSS is using databases to help military training professionals perform LSAR and ISD analyses to make job performance aid (JPA) and training decisions.

A fundamental feature of DSS is its interface with and use of LSAR data. Another significant feature of DSS is that it accommodates several different task selection models and media selection models. This flexibility allows the tool to be used by different services.

DSS consists of LSAR data input routines and Joint Service ISD analysis processes. The system includes utility functions that provide system security, database administration, report generation, and ISD analysis functions.

The procedures classify users into five categories: Database Administrator, Training Development Manager, ISD Analyst, Quality Assurance Reviewer, and Reference File Maintainer. DSS provides facilities for each user category except the Reference File Maintainer, whose work is performed manually. Except for the Database Administrator, each user category has both administrative and ISD analysis responsibilities. The same person can be in more than one user category. Depending on which category a user assumes when using the software, it makes available the functions that are applicable to the category.

The procedures are a top-down approach starting from a defense system down to tasks and learning objectives. At different stages of the procedures, users analyze a defense system downward in the following sequence: defense system, subsystems and their associated skill specialties, tasks, task elements, terminal learning objectives, and enabling learning objectives. During this elaborative process, training personnel are assigned responsibility for the items at different levels. The process ensures that needed training components for a defense system are covered.

A large proportion of data needed for analysis can be imported from LSAR. If the LSAR data

are not available, users can also enter the data directly into DSS.

DSS supports a number of ISD functions. With the help of automated logic, users select tasks that require training, select instructional settings and training media, sequence instruction, and identify training equipment fidelity requirements. DSS presents LSAR and other analysis-related data, previously entered or generated by the system, to assist users in making ISD decisions. Context-sensitive help is available throughout the software.

Users can overwrite LSAR and DSS decisions. The DSS allows users to document their *rationale for the overwriting decisions. Quality* assurance reviewers are encouraged to pay particular attention to these decisions.

DSS documents all analyses on automated worksheets. Users can print various worksheets for quality control, communication, and documentation purposes. Examples of worksheets are the task instructional setting report, course outline report, and task element report.

Considerations - The ISD/LSAR DSS is specifically designed to meet the training development needs of large-scale defense system acquisition projects. It is similar to TASCS but is more sophisticated. The process is more complex; extensive on-line help is provided; reports can always be sent to a file instead of to a printer; the manual is more thorough; and the software can be used in a multi-user LAN environment. Furthermore, continuous software and training support is available from the contractor who develops the tool.

The concurrency of databases is easier to maintain in DSS than in TASCS. Updated versions of data are consistently presented at different parts of DSS.

Similar to TASCS for ISD analyses, DSS covers only instructional design. The supported procedures end at identifying instructional settings and media, and assigning learning objectives to courses, annexes, and lessons. The text-based user interface also does not support instructional hierarchy analysis in a graphical manner. It is obvious that tools supporting the middle and the rear of the ISD process must be identified to complement DSS.

DSS also does not deal with time and financial management aspects of training development. Training managers can get help for these important decisions from other tools such as GCASS and TRACES.

The DSS manual is fair, but the ISD methodology is not well documented. Although the manual includes a long list of references, the sources of specific procedures are not identified.

RECOMMENDATIONS FOR IMPROVEMENT

User Manual

The lack of a good user manual significantly reduces the value of software regardless of how powerful the software may be. It is therefore puzzling that these software tools developed for the training community are not complemented with good user manuals or training materials. Up-to-date user manuals should be developed to include underlying training development methodology, internal logic, and software functions and procedures.

On-line Help

A comprehensive, context-sensitive, on-line help system would further enhance the usability of the software tools. Except for JS ISD/LSAR DSS, none of the tools provide this facility yet.

Graphical Interface and Tools

Instructional analysis and design require the production of graphical working documents such as block diagrams and flow charts. For example, hierarchical drawings greatly facilitate analyses of intellectual skill objectives[1,2]. Without diagrams, it is almost impossible to depict and evaluate the relationships among the training objectives of a complex training system.

Although adoption of the IBM-compatible platform may have made it more difficult to provide graphical interface and tools, they still

need to be programmed into the software for the tools to be effective. Presenting instructional hierarchies in an outline format instead of a block diagram format simply does not provide adequate support for instructional analyses.

Internal Logic

Internal logics of the automated tools are implementations of specific instructional design models. To allow training developers to use the tools correctly, the logics should be transparent to the users. The logics must be either documented or made accessible in the software. Furthermore, users often need to modify the logics to meet local requirements or accommodate advancements in instructional design methodology. Making the logics modifiable would facilitate adoption of the tools.

Integration with Authoring Tools

Table 1 is the analysis result of the tools' functions which cover training management and instructional design only. Instructional development is not supported by the tools. Integrating the tools with authoring tools, by automatically incorporating the output of the former as input for the latter, would improve the usefulness of the tools.

DOD Clearinghouse

The fact that a number of Department of Defense agencies set out to develop the tools evaluated in this paper shows there is a widely recognized need for such tools. Development of sound and comprehensive tools is not an easy task and requires many resources. Establishment of a clearinghouse to coordinate efforts would save resources and help to create more useful tools.

CONCLUSION

The Air Force has made some headway into building training development software tools. Based on the analysis reported in this paper, obviously there is still a long way to go to maximize both efficiency and quality benefits such tools may produce. Given the task's tremendous demand on resources, agencies interested in undertaking the endeavor should coordinate with each other to optimize the use of limited resources. Only with this approach can the formidable task of creating a truly comprehensive, flexible, and user-friendly tool be accomplished in the foreseeable future.

Note: This paper is based on a study conducted for the Aeronautical Systems Division and the Air Training Command of the U.S. Air Force, contract number F42650-87-D-0026, delivery order number 5141. The opinions expressed in the paper are the author's and may not be endorsed by the Air Force.

Table 1.
Functions of Training Development Software Tools

FUNCTIONS	TOOLS			
	GTET	TRACES	TASCS	ISD/LSAR
TRAINING MANAGEMENT				
Cost estimation	X	X		
Time management	X			
Resource management	X			
ID team assignment management				X
Defense system data management				X
INSTRUCTIONAL DESIGN				
Job task management			X	X
Training task selection			X	X
Instructional setting selection				X
Instructional hierarchy analysis			X	X
Instructional objective management			X	X
Media selection			X	X
Course structure management			X	X
Training device fidelity determination				X

REFERENCES

1. Dick, W. and Carey, L. *The Systematic Design of Instruction* (2nd ed.). Glenview, Illinois: Scott, Foresman and Co., 1985.

2. Gagné, R. M., Briggs, L. J., and Wager, W. W. *Principles of Instructional Design* (4th ed.). Fort Worth, Texas: Harcourt Brace Jovanovich, 1992.