DEVELOPMENT OF A FRONT END ANALYSIS TOOL

Jeffery L. Maxey
Advanced Technology, Inc.
Orlando, Florida

Robert H. Ahlers
Naval Training Systems Center
Orlando, Florida

ABSTRACT

After almost 20 years, deficiencies still exist in ISD and the products prepared for training. Analysts attribute many ISD shortcomings to compromises made in its implementation and limitations in the available tools. To improve the quality of ISD products, the Naval Training System Center is developing a family of software tools to support the ISD front end analysis process. The first of these is VISTA, the Visual Interactive System for Task Analysis.

This paper describes VISTA's design philosophy which incorporated the application of basic human factors data in concert with refinement through iterative test and evaluation. The design process is discussed to include results from system usability tests. The paper also describes VISTA's present state and plans for future development and applications.

INTRODUCTION

The procedures, models, standards, and decision aids for instructional systems development (ISD) comprise the tools available for designing and implementing military instruction. But, after almost 20 years of ISD applications, deficiencies still exist in ISD and in the products prepared for training (Montague & Wulfeck, 1986).

Analysts attribute many ISD shortcomings to compromises made in its implementation and limitations in applying available tools. Inexperienced personnel are often used to complete much of the analytical and developmental work required for training development. Their inexperience can lead to errors in applying ISD guidance and procedures.

Further, ISD tools are typically not user friendly. Inexperienced personnel find them difficult to apply, especially in complex circumstances where many different factors influence how best to proceed. The tools are difficult to use because just the "what to do" is specified in any detail. Little attention is directed at the "how to do it". Experience and judgment, which the inexperienced users lack, are required to bridge the gap between "what" and" how".

To improve the quality of ISD products, the Naval Training System Center (NTSC) is developing a family of graphic interface tools designed to aid the completion of the front end analysis phase of ISD, i.e., Phase I: Analysis. The first of these is the Visual Interactive System for Task Analysis (VISTA). This tool is designed to support the rapid development of task lists and hierarchies; the identification of tasks meeting specified selection criteria; and the documentation of training data by both inexperienced and proficient ISD practitioners.

VISTA DESIGN PHILOSOPHY

Many other attempts to provide a computer-based tool for assisting the ISD Analysis phase have been made in government and industry (e.g., Jared, 1987; Kribs, 1989; Marcue, Blaiwes, & Bird, 1983). Typically, the problem has been viewed as a structured database application. The focus has been on design of the data base with little attention paid to ease of use issues. While there are certainly gains to be made by providing a tool with database functions, the VISTA designers felt that a tool's ease of use would be the primary determinant of its benefit.

A usable tool must have an interface which requires little effort to use and which has functions that are relatively obvious to access (Norman, 1988). The volume of task and training data, along with the many forms in which it must be reported, also requires that the user interface be as efficient as possible. These design tenets dictated that the tool development process should involve human factors and user inputs throughout the development cycle. These data would be used to resolve user interface issues.

Interface Issues

Probably the biggest criticism of complex (as well as many simple) systems is that "You need to be an engineer to figure out how to work this." Certainly, computer based systems are some of the biggest offenders. Why? Poor interface design.

Recognizing interface design as a major determiner of whether a system will be easy to use, the decision was made at the outset of the VISTA project to promote ease of use by incorporating a set of basic design goals in the interface. These goals were driven by two basic human factors. First, humans interact with "things" primarily through the visual modality. Second, humans are limited in their ability to process information.

Two other ideas were integrated into the design goals. Since typing skills are not universal, keyboard input should be kept to a minimum. Because most human information processing is serial, the user's attention should be focused on one objective at a time.

These goals yielded the following design characteristics:

- o Multiple views of task data through both graphic and text displays
- o Incremental refinement of task data bases
- o Separate entry of procedural and declarative information
- o Operating cues integrated into the interface
- o Maximum use of contextual information
- o Rapid data input with minimal keyboard interaction

Selection of a windows-icon-mouse-pointing (WIMP) interface as the basic means of interacting with ISD task analysis and training data facilitated achievement of these goals. This was for several reasons. First, because it is a visual interface, WIMP provides a convenient mechanism for connecting to a graphic data environment. This type of representation would in turn support realization of the first design goal, i.e., multiple data views.

Accessing a graphic data environment, a user should be able to acquire both graphic and text representations of task data. Graphic views could be a full top-down graphic display of the task analysis data to provide context and location within the confines of the data. It could also be displays of task elements within specific segments of the environment, showing structural relations among elements. In contrast, text views, representing standard task listings with task statements referenced by identification code numbers, could be created.

A network formalism was chosen to create the graphic environment. The network provided for definition, modification, and real time visualization of structural information about tasks. It also provided a mechanism for attaching qualitative and quantitative data to tasks.

Thus, in interacting with analysis data via the WIMP interface, definition and refinement of task relationships could be accomplished as a first step to set the scope of the analysis. These could be refined as dictated by new information. Then, qualitative and quantitative data could be added to individual elements to enrich the data base captured by the network. This would realize the second and third design goals.

The WIMP interface would also allow users to rapidly select and execute system functions. Obvious icons and menus could be created. Contextual information could be embedded in the interface to lighten the memory burden imposed on new or casual users. This would allow users to create tasks, move them around in the data base, and define intertask relationships quickly with minimal effort. This would effectively achieve the fourth and fifth design goals.

Rapid data input would also be supported by allowing users to import task and data representations created by other software through the WIMP interface. Minimal keyboard input would also be promoted (sixth design goal) through data importation and, as well, by the creation of a reusable task grammar stored in networks and accessible through special grammar windows.

Hardware Issues

The software from which VISTA was derived had been developed on an expensive, limited availability LISP machine. Past experience has shown that one impediment to the use of software is the inability of users to gain access to the machines on which it is hosted (Panel on Information Technology, 1989). Conversely, the more machines that can host a piece of software, the more likely and frequently the software will be used.

One consequence of the personal computer revolution of the last 10 years has been the widespread availability of fast powerful systems having abundant working and storage memory, high quality color graphics, and a variety of programming languages, some especially suited for rapid prototyping. The result is the availability of computing environments capable of hosting applications that could previously be implemented only on minicomputers and high end workstations.

Thus, to insure the general availability of VISTA, the decision was made to rehost the LISP based software that would form the core of VISTA on a 286 machine. Further, this would be accomplished using an interpretive, object oriented programming language designed to support the WIMP interface.

This approach had several advantages. First, 286 machines are very common within the military, thus ensuring the practicality of widespread test and evaluation at candidate user sites as well as the widespread distribution of VISTA once it is ready for general use.

Second, using interpretive software to code VISTA ensured that prototyping could be accomplished very rapidly. This would accommodate the iterative development-evaluation-development cycle which was planned. Following development of basic system structures, bugs and design flaws could be remedied efficiently while identified enhancements could be easily implemented.

VISTA DESCRIPTION

VISTA is designed to facilitate the creation of task lists and hierarchies, the searching for task elements meeting specified criteria, and the outputting of training data to support the preparation of ISD materials. Both graphic and text representations of the data base can be output to support other portions of ISD Phase I: Analysis or other ISD Phases involving task or training analysis.

As discussed later in this paper, support for other ISD activities (e.g., media analysis) could be provided. The only limitations are the resources available for tool development and the creativity of the tool developer. Based on experience to date, the VISTA interface should be suitable for most ISD analytical activities.

Features And Operation

VISTA's capabilities are accessed via a set of mouse-sensitive icons and menus. Figure 1 shows the layout of the primary VISTA screen containing an example task network. An identification message area naming the current Job (RELIEVE WATCH AS TAO in Figure 1), the network level (TOP LEVEL), and the Job number (JOB 1) reside along the screen top. The job identifies which of two parts of the network data base is active (JOB 1 or JOB 2), while level identifies the network location shown in the Work Area for the VISTA data base that has been loaded or created.

Task elements are represented graphically as nodes within the network. The JOB name is the apex of the network. "Duties" which define the JOB comprise the TOP LEVEL of the net. One or more tasks comprise each duty. Tasks can be divided into subtasks. In turn, these can be divided into steps which can be split into substeps. Task nodes which have associated lower levels (e.g., tasks which have subtasks under them in the net) are differentiated from those without lower levels via color coding.

Figure 2 shows the multiple levels a network can have. This is the full representation of the network whose TOP LEVEL is shown in Figure 1. Referring to

Figure 1, two TOP LEVEL duties (the first at the *left and the fourth at the bottom of the screen)* have been decomposed into at least one or more tasks. This is indicated by the darker coloring of these duties in the figure.

Referring back to Figure 2, the first duty has been broken into five tasks which have been further broken into subtasks and steps. The second duty from the TOP LEVEL has been broken down only to the task level.

Using the Figure 2 network representation, the user can always determine the current location. This is indicated by a text message in the bottom left hand corner of the screen work area and by the position of the "cross-shaped" cursor. The user can move to any other part of the network by moving this cursor to the appropriate point within the network *representation* and *designating* the *new* location through the mouse.

Icons, representing high use functions, reside along the left-hand side of the screen next to the Work Area. Menus reside between the ID and Work Area. Along the bottom of the Work Area, a second message area displays text statements designed to aid the user in exercising icon or menu functions.

Seven system functions are accessed via icons:

- o Create a task (represented by a green rectangle)
- o Indicate a temporal sequence between tasks (represented by a black arrow)
- o Move one or a group of tasks from one location to another (represented by an orange truck)
- o Delete one or a group of objects, either tasks or arrows (represented by a green trashcan)

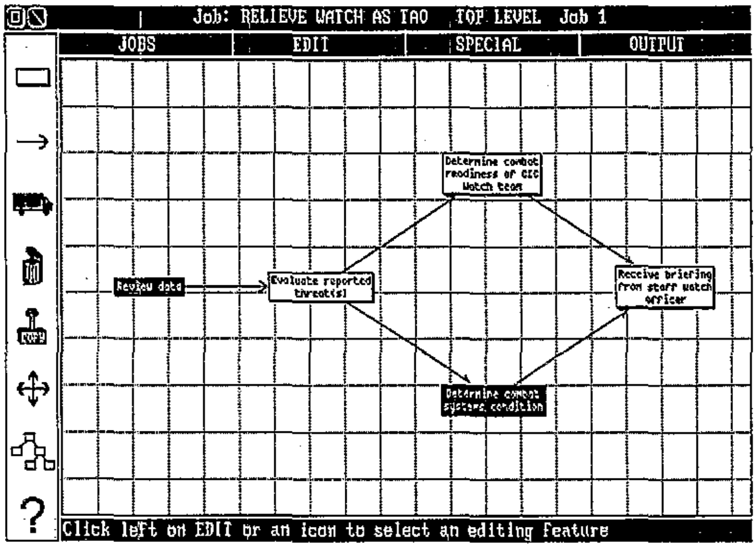


Figure 1. Primary VISTA screen with an example task network.

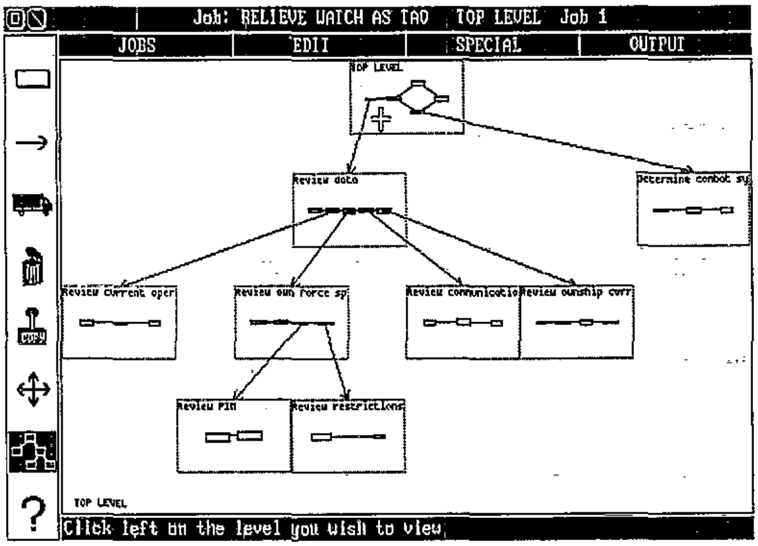


Figure 2. Full graphic representation of the Figure 1 task network.

- o Copy one or a group of tasks from one location to another (represented by a purple rubber stamp with the label COPY)
- o Scroll within a level or move to another level (represented by a black set of Up/Down/Left/Right arrows)
- o Identify current position in the hierarchy and accommodate movement among levels (represented by a network structure)
- o Help (represented by a blue question mark)

The four menus support additional system functions. These are expected to be used less frequently than the functions accessed via the icons. Figures 3 through 6 show primary VISTA screens with VISTA menus pulled down.

Figure 3 shows the VISTA screen just after the system has been accessed and the JOBS menu has been pulled down. A VISTA data base has not been created or loaded at this point, so the screen work area is empty except for the gridded background. The user has five options at this point: create, load, copy, remove, or rename a VISTA data base.

Figure 4 shows the primary VISTA screen after a network data base has been loaded and the EDIT menu has been pulled down. The network in this figure is the same as the one shown in Figure 1. Two kinds of options are available to the user here: make structural changes to the task data base or edit a task name.

Figure 5 shows the same screen as before except that the SPECIAL menu is pulled down. This collection of functions allows the user a variety of eclectic options from leaving VISTA to conducting a search of the VISTA data base to finding task elements meeting specific criteria.

Finally, Figure 6 shows the primary screen with the OUTPUT menu pulled down. The options shown in this menu allow the user to output descriptive information about the network loaded in VISTA, a text listing of the task hierarchy represented by the network, training data associated with the task data, or lists of tasks meeting specific criteria.

Figure 7 (top) illustrates a TOP LEVEL VISTA task hierarchy. This hierarchy comprises four tasks. The analogous text version of the hierarchy, as it would be typically presented, is shown at the bottom of Figure 7.

There is an important observation to be made from comparing the graphic and text representations of Figure 7. The graphic representation provides a quick and much more readily apparent display of the task structure for the job. In the textual representation, this structure is just not present.

Also from the outline, there is no way of determining which (if any) tasks branch (the decision task is indicated by the hexagonal box) or if tasks "loop" (the "Go To" condition is designated by a circular task icon). Any structure implied by the numerical task codes must be abstracted from the codes before it can be comprehended. The graphic representation of task relationships is superior because patterns can be recognized much faster from visual displays than from conventional formats involving columns of numbers in a list or table.

The left mouse button is used to initiate all icon and menu functions used to create and edit a task hierarchy. The middle and right buttons are used to implement two high frequency operations, movement between hierarchy levels and task documentation.

When the user places the cursor on a task and clicks the middle mouse button, the system shifts to the next level below the task, displaying within the Work Area any task elements defined at this level. When the user points to empty space within the Work Area, the system automatically shifts to the next higher level in the task net. Thus, the middle mouse button provides a very quick and easy way to move between levels within the net.

Pointing to a duty, task, subtask, step, or substep and clicking the right mouse button yields a window for entering task data. These data may be for just the task element selected or may include all associated subordinate elements. The data can consist of text recommending instructional media, defining performance conditions and/or standards or numerical data reflecting dimensions such as percent of personnel performing the task, task criticality, training priority, etc.

Hierarchies can be created from scratch or from existing task lists or hierarchies. In both cases a grammar (task verbs and verb-objects) must be defined from which new tasks can be created. Creating a grammar from scratch involves creating and naming a number of tasks. The grammar is automatically extracted from the tasks and organized into a network, which can be accessed by pointing and clicking with the mouse each time a new task is created and named. Much time and typing can be saved by transferring task elements from the grammar, rather than developing each task statement from scratch. However, it is still time consuming, and does require typing skills by the user, in order to build an adequate grammar.

To expedite the development of task lists/hierarchies, VISTA has the ability to import exiting lists or hierarchies stored as disk files. Using the "Create a new job diagram" function from the Jobs Menu, the user can import a task list to define a grammar from which tasks can be constructed. Also, the user can import an existing hierarchy. This not only yields a grammar, but also the task structure which can then be modified. In this way, the user can start with a hierarchy similar to what is desired as an end product and modify it to include structural information not present in a standard text description.

Jobs are graphically displayed in one of two working areas, each of which may have up to seven levels. The user can have two Jobs "active" at the same time, transferring data from one to the other. This facility allows, for example, parts of one job hierarchy to be copied to another.

Within a level, tasks can be arranged in the sequence in which they are to be performed, including parallel activities. Looping processes are indicated with "Go To" statements. Decisions are represented with a multiple branching structure.

Tasks (individually or in groups) may be added and removed, linked and unlinked, moved and/or copied from one location to another or from one level to the next. Task data is tied to individual tasks via notecards. Information about conditions and standards of performance as well as data reflecting percent personnel performing, task criticality,

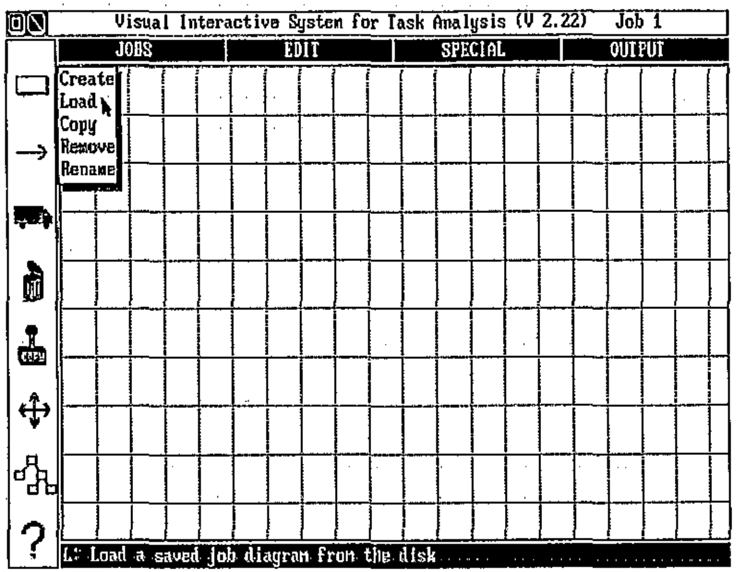


Figure 3. Primary VISTA screen with JOBS menu pulled down.

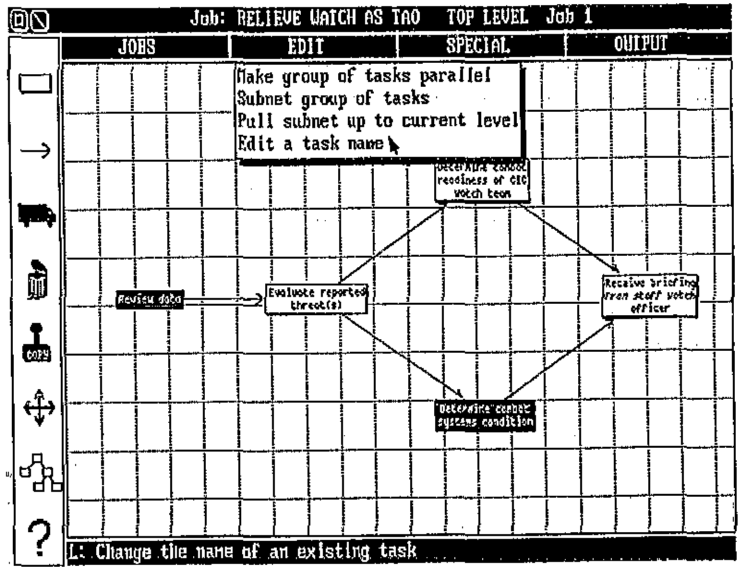


Figure 4. Primary VISTA screen with the EDIT menu pulled down.

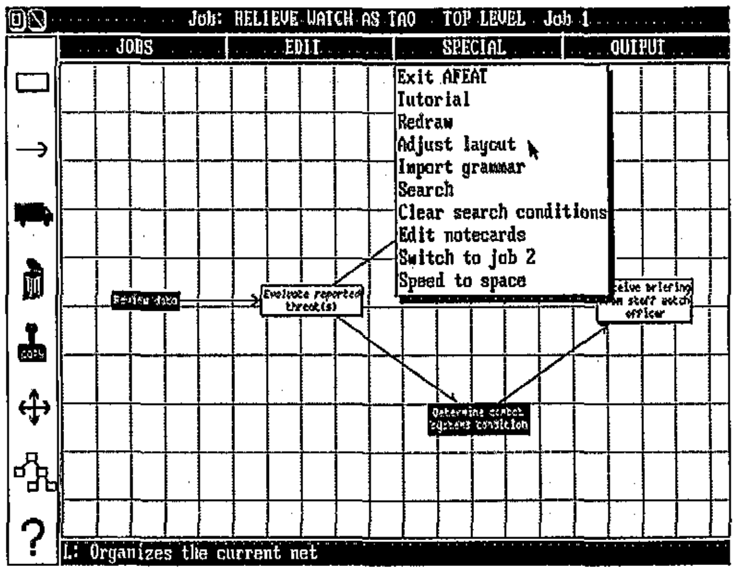


Figure 5. Primary VISTA screen with the SPECIAL menu pulled down.

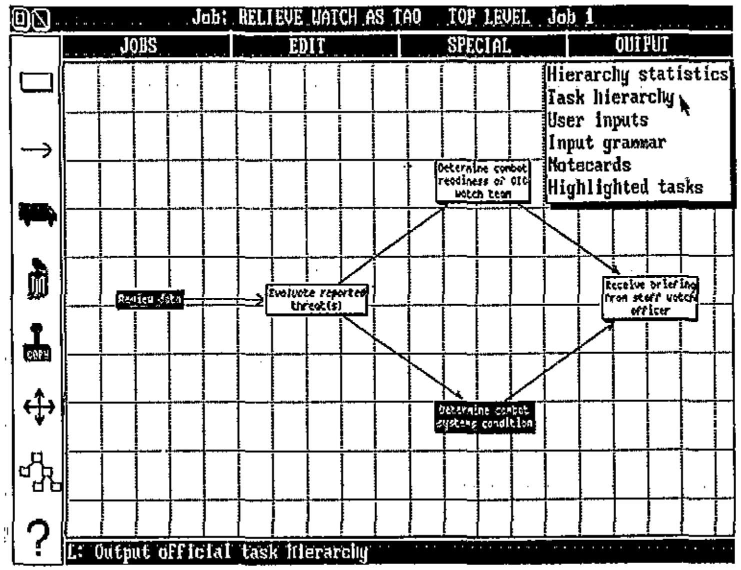
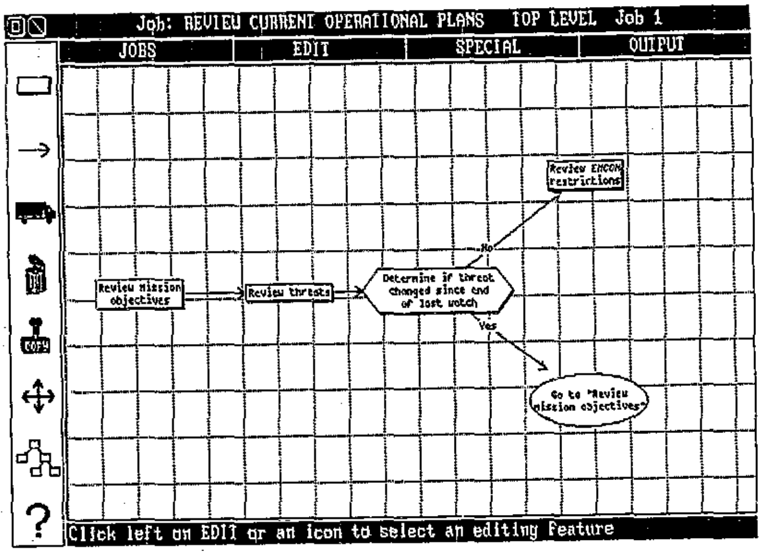


Figure 6. Primary VISTA screen with the OUTPUT menu pulled down.



1.0 Review current operational plans
1.1 Review mission objectives
1.2 Review threats
1.3 Determine if threat changed since end of last watch
1.4 Review EMCON restrictions

Figure 7. Descriptions for Review Current Operational Plans task. The VISTA graphic representation is at top and a typical text listing is at bottom.

performance frequency, task difficulty, recommended media or other relevant data can be entered on separate cards. The cards for a task may be displayed by positioning the cursor over the task and clicking the right mouse button.

Outputs include task hierarchies, training objective hierarchies, hierarchies comprising tasks that meet specific sets of criteria, and task grammars. These may be printed to the printer, a disk file, or the screen.

Hosting

VISTA requires a 80286-based, MS-DOS computer with an EGA system, 3.1 MB of extended RAM, a 3-button mouse, and 2 MB of disk space. VISTA is written in Smalltalk/ V 286 R1.00, an object oriented language.

USER EVALUATION

The user has frequently been left out of the software development process. In recent years, however, it has become more commonplace for software developers to have typical users try out a system, perform benchmark tasks, and suggest improvements (Schneiderman, 1987).

Because of the benefits expected from user participation (e.g., quality feedback about ease of use, obviousness of features, and apparent and subtle bugs), user evaluations were made a part of the VISTA design process. Evaluation was designed to serve three functions: 1) to define the rate with which potential users learn to apply the tool, 2) to assess the ease with which the users apply the tool once familiar with its operation, and 3) to identify specific user needs and interests not obvious from standard task analysis documentation and for product improvement.

Four formal evaluations were planned during the one-year VISTA development process, each to be tailored to the current development stage:

- Evaluation I was to address human factors issues related to the user interface.
- Evaluation II was to provide guidance from a typical user sample concerning desirable features.
- Evaluations III and IV were to provide summative data to allow cost and effectiveness comparisons.

The first two evaluations have been completed. The first verified the functionality of the initial VISTA prototype and identified ways in which its interface and features could be streamlined. This was completed by a human factors psychologist experienced in front end analysis and software development. The approach systematically exercised the tool's features, noting apparent bugs, mismatches between expected and actual outcomes, and difficulties encountered in exercising features. Findings were reported (Maxey, 1989a) and provided to the developer to be used for refining the tool.

The second evaluation (Maxey, 1989b) determined the capabilities of the revised tool in preparing and validating task lists and hierarchies. Potential users (two subject matter experts involved in training analysis) applied the revised tool under two conditions. They first explored use of the tool under "free play" conditions with limited instruction. This provided an opportunity for them to learn how to use the tool to create and modify task lists and hierarchies.

Next, the two subjects used the tool to complete two benchmark tasks. The first was to create a task list (given an operational scenario and a disk file containing candidate tasks), and then used the task list to create a task hierarchy. The second task was to validate a previously developed task list and associated task hierarchy.

Data were obtained from three sources. First, performance was observed and documented by the experimenter. Secondly, users responded to a questionnaire designed to elicit opinions about VISTA's design and use. This questionnaire contained 54 questions which covered 13 basic system attributes (Table 1) using a 10-point rating scale. Each question was rated on an appropriate dimension, such as: "Difficult" - " Easy", "Never" - "Always", "Confusing" - "Clear". Numerically higher ratings indicate more desirable characteristics. Finally, users were queried in a *structured interview to* determine:

- o their opinions about use of the tool to complete the benchmark tasks
- o the quality of the interface and features
- o the usability of the tool
- o planning needed to use the tool
- o training needed to use the tool

TABLE 1

System Attributes Evaluated Via Questionnaire And Mean Responses Of Users

Attribute	Number of Questionnaire Items	Mean Response
Initial Operation	4	8.50
Display Quality	4	8.31
Interaction Pace	2	8.13
Task Facilitators	4	8.13
Terminology	3	7.75
Sense of Position	2	7.63
Screen Characteristics	5	7.57
Feedback	4	7.53
Information Processing	6	7.36
Instructions	7	7.21
Task Operations	4	6.48
Error Correction	3	6.17
Error Messages	6	4.13
TOTALS	54	7.30

The user data were summarized and reviewed to determine VISTA's usability, problems and system bugs, and potential enhancements to the system.

Usability

Attribute ratings were created for each user for each benchmark task session. This involved averaging the responses to the questionnaire items comprising each attribute category. As shown in Table 1, the number of items per category varied from two to seven items.

The four ratings for each system attribute category (i.e., two per user *times two users)* were then averaged to produce the mean ratings in Table 1. These ratings were inspected to identify attributes that received Low (zero to three), Midpoint (four to six), and High (seven to ten) level ratings. Low and Midpoint ratings would indicate possible design flaws or bugs, while High ratings would point to those aspects of VISTA that required no further development.

Ten of the thirteen attribute categories had High ratings. Only one category (Error Messages) was below Midpoint. Questionnaire items addressing this attribute concerned communications about user errors and the actions required to correct them.

Problems and Bugs

The data on user problems and "bugs" were collected to identify and guide the redesign of poorly designed parts of the software. User problems were assumed to be caused by design flaws and not by user carelessness. The software "bugs" were due to faulty program development, but most of the design flaws were caused by violating a user's expectations. Frequently reported problems included:

- o Confusion from multiple options for implementing a system feature
- o Imprecise terminology
- o Mismatches between stated instructions and the actual actions required to perform an operation
- o *Incomplete instructions*
- o Misleading error messages
- o Complex procedures for accomplishing an apparently simple task
- o Exceeding a user's memory limitations

Suggested Enhancements

The observation data and the questionnaire and debrief comments were reviewed to identify enhancements that the users thought would make VISTA easier to use. The major recommendations addressed the system interface, especially icon and menu interactions, and error message content.

Enhancements included a new graphic representation to designate job tasks normally performed by a human that are now performed by a machine. A single operation for creating a lower hierarchy level by grouping related tasks at a higher level was also suggested. This was a function which the users performed quite frequently but required chaining together a number of elemental operations.

Menu enhancements were suggested to simplify or improve the efficiency of menu features, reduce possible user confusion about system operations, and make system operations more obvious.

Users were unanimous in suggesting that error messages should not only indicate that an error condition was present but also instruct the user how to correct the abnormal situation.

Evaluation Summary

The findings from the second VISTA user evaluation clearly showed that while the system was sufficiently mature to give an indication of its potential, additional resources had to be invested in its development. The second evaluation provided guidance to select the most cost effective use of the resources.

In the last two evaluations, users will perform two tasks: 1) develop a task hierarchy, and 2) select tasks for training. These will be done manually and by using VISTA. Records of the time spent performing each step in hierarchy development and task selection will provide the raw data for determining the relative effectiveness of VISTA compared to the manual method. Additionally, difficulties and problems experienced in developing the hierarchy and selecting tasks will be documented for both manual and VISTA supported performance. The intent of the evaluations is to quantify the effectiveness of VISTA and to identify problems and difficulties inherent in the manual method that VISTA remedies or that could readily be eliminated through further modifications.

FUTURE DEVELOPMENTS

Several directions have been identified for future development. In its present form, VISTA is a qualitative analysis tool in that its focus is to support decomposition of tasks into their constituent elements in a top-down manner. The decomposition is performed by the user. VISTA provides an environment in which the decomposition can be organized and recorded as it is being accomplished.

One direction being explored is the integration of VISTA into a network with other application programs to include statistical analysis programs for summarizing job analysis data, database programs for report generation, word processing programs for preparing initial task hierarchies or grammars for VISTA analysis, and spreadsheet programs for analysis of numerical data.

Another possibility is to modify VISTA so that users can play "what if" to investigate trading-off different sets of circumstances to determine their influence on instructional design. For example, an analyst might like to know how different sets of assumptions about the tasks selected for training affect required media and thus the cost of the instructional delivery system. If one or two tasks drive the selection of an expensive system, this can be readily and quickly identified.

Support for other ISD functions (e.g., media analysis) is being considered. This will involve consulting first with ISD practitioners and determining the data normally used and the steps typically taken to acquire and process this data to create ISD products. Frequently performed analytical functions will be identified and implemented within VISTA either as new icons or menu selections. The intent will be to make their performance quick and efficient. Outputs will be displayed graphically to take advantage of the human's ability to detect patterns in pictures faster than through the inspection of columns of numbers or lists of words and phrases. Such displays are receiving increased attention. The Panel on Information Technology (1989) has concluded that: "Scientists need an alternative to numbers. A technical reality and a cognitive imperative tomorrow are the use of images. The ability of scientists to visualize complex computations and simulations is absolutely essential to ensure the integrity of analyses, to provoke insights, and to communicate those insights with others" (p. 32).

Expert systems could help novices make decisions they do not have the experience to make. An expert system is well suited to evaluate the many factors involved in ISD. But a problem may result from using a expert system to substitute for a novice's inexperience; the novice will remain inexperienced. In work by Whitaker, Taynor, & Wiggins (1989) novice engineers "see" how experienced engineers solved a problem similar to one they have to solve. A similar capability could be designed for ISD decision support tools to show the novice how and why experts made certain decisions so the novice VISTA user may become more experienced.

VISTA should be viewed as a flexible user interface design (a philosophy of user interaction) which allows a very complex database to be input, maintained, and searched. The preceding "wish list" of future extensions is purposely incomplete. The users will provide the details concerning how they would like VISTA to grow to meet their needs.

REFERENCES

Jared, J. (1987). Computer aided training development system (CATDS). In Proceedings of the Ninth Interservice/Industry Training System Conference (pp.205-211). Washington, DC: American Defense Preparedness Association.

Kribs, D. (1989). Automated instructional system development. San Diego, CA: Instructional Science and Development, Inc.

Montague, M. & Wulfeck, W. (1986). Instructional systems design. In J. Ellis (Ed.). Military contributions to instructional technology (pp. 1-19). New York, NY: Praeger Press.

Marcue, N., Blaiwes, A., & Bird, R. (1983). Computer aided system for developing aircrew training (CASDAT) (Technical Report NAVTRAEQUIPCEN 70-C-0076-1). Orlando, FL: Naval Training Equipment Center.

Maxey, J. (1989a). Technical report of NTSC Automated Task Analysis Tool evaluation (Interim Report, Contract No. N61339-87-0006). Orlando, FL: Advanced Technology.

Maxey, J. (1989b). Technical report of the second user evaluation of the NTSC Automated Task Analysis Tool (Interim Report, Contract No. N61339-87-0006). Orlando, FL: Advanced Technology.

Norman, D. (1988). The psychology of everyday things. New York, NY: Basic Books.

Panel on Information Technology (1989). Information technology and the conduct of research: The user's view. Washington, DC: National Academy Press.

Schneiderman, B. (1987). Designing the user interface. Reading, MA: Addison Wesley.

Whitaker, L., Taynor, J., & Wiggins, S. (1989). Case-based reasoning systems for computerized training of novice engineers in the area of survivability/vulnerability (Letter Report). Yellow Springs, OH: Klein Associates.