Usability Analysis of a Personal Digital Assistant Based Data Collection Tool for the Shipboard Training Environment

Dr. Robert C. Allen
NAVAIR Orlando Training Systems Division
12350 Research Parkway
Orlando, FL 32826-3275
USA

Mr. Paul J. Hession
Sonalysts, Inc.
12501 Research Parkway
Orlando, FL 32826-3224
USA

Ms. Eleni D. Kring
Dynamics Research Corporation
3505 Lake Lynda Dr.
Suite 100
Orlando, FL 32817-
USA

Summary

Researchers for the U.S. Navy have developed multiple instructor aides for performance measurement hosted on hand-held computers such as pen tablet computers. This technology provides a potential solution to the challenge of supporting training in complex, data intensive shipboard environments. However, Hand-held computers are relatively expensive and can be cumbersome in the confined spaces found in such environments. Therefore, the U. S. Navy is investigating the use of more portable, lightweight data collection tools such as Personal Digital Assistants (PDAs). Hardware and software limitations associated with these devices exist, including limited screen real estate and memory. The challenge for the Naval training and human factors communities is to develop training applications for PDAs that are relevant to shipboard users and that also apply sound human factors and usability principles. The primary purpose of this paper is to describe a usability analysis of a training application loaded onto a Pocket PC. The analysis included heuristic evaluations, user testing sessions, and redesign recommendations. The target audience of the application is U.S. Navy shipboard instructors, who would use the application to prebrief a training audience, to collect data during an exercise, and to debrief the training audience.

Background

In order to fully understand the usability evaluation data reported in this paper, it is necessary to first describe to the reader the forces that led to the development of the subject of this paper, that is, the Personal Digital Assistant (PDA) training application. These forces include the changing training environment in the U.S. Navy, the Navy's implementation of a training system and methodology (Battle Force Tactical Training or BFTT and Objective Based Training or OBT, respectively), the development of software that supports this training methodology (the Afloat Training Exercise and Management System or ATEAMS), and the development of hand-held computer devices that support shipboard data collection. Note that an acronym list is provided at the end of this document to help the reader more easily understand the terms contained herein.

The U.S. military is attempting to reduce both cost and manning while concurrently maintaining operational readiness. In order to meet these conflicting goals, the U.S. Navy is investigating various strategies to augment or supplement training. For example, one approach uses embedded training systems to both simulate a theater of war and stimulate trainees' instruments to display operationally realistic data. Such systems are capable of capturing data that was not possible to capture through previous data collection methods (e.g., paper and pencil). However, the amount of data that can be displayed/given to an instructor, either real-time or during an After Action Review, can be overwhelming. To help aid the instructor capture

Paper presented at the RTO HFM Symposium on "Advanced Technologies for Military Training", held in Genoa, Italy, 13 – 15 October 2003, and published in RTO-MP-HFM-101.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **00 APR 2004**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE
Usability Analysis of a Personal Digital Assistant Based Data Collection Tool for the Shipboard Training Environment

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NAVAIR Orlando Training Systems Division 12350 Research Parkway Orlando, FL 32826-3275 USA; Sonalysts, Inc. 12501 Research Parkway Orlando, FL 32826-3224 USA

8. PERFORMING ORGANIZATION REPORT NUMBER

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

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM001667, NATO RTO-MP-HFM-101 Advanced Technologies for Military Training (Technologies avancées pour lentraînement militaire)., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**50**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

training-relevant data in highly complex and data-rich training environments, the U.S. Navy has been developing automated data collection tools. For example, automated performance measurement is being implemented under the Navy's Battle Force Tactical Training (BFTT) system. BFTT immerses trainees in a controlled, interactive Synthetic Theater of War environment. The BFTT system is designed to allow trainees to train as they fight, using their operational equipment during a training exercise. The trainees can be onboard ship or in a schoolhouse (see Figure 1) (RCI, 2000).

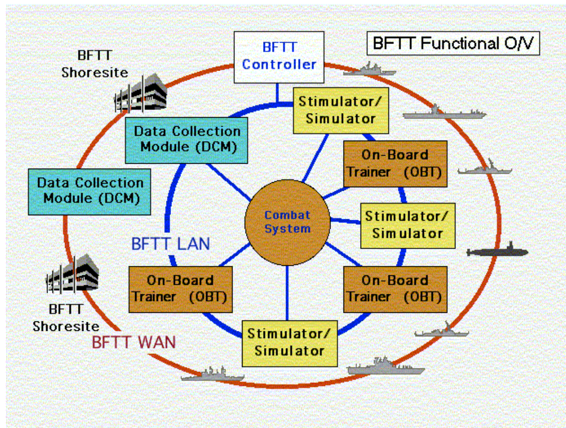

Figure 1. BFTT functional overview. From Battle Force Tactile Training (BFTT) System Overview CD-ROM.

The BFFT effort has defined a 'train by objective' strategy, to be used to improve training onboard ship. This strategy requires the identification of quantifiable training objectives that can be used to accurately rate performance (Lyons & Allen, 2000; Stretton, 2001). Therefore, Commander in Chief, U.S. Pacific Fleet approved a training concept labeled Objective Based Training (OBT). OBT defines the tasks that must be performed, either at the individual, team, or ship level, how these tasks must be performed, and the standards that must be achieved (Lyons & Allen, 2000).

Objective Based Training

The OBT strategy employs Terminal Objectives (TOs), Enabling Objectives (EOs), and Measures of Performance (MOPs) of various warfare areas, such as Combat Systems, Engineering, and Damage Control. Overarching the TOs, EOs, and MOPs are Training Events (TEs). A Training Event is a very high level description of an event that will occur during a training exercise - for example, Employ Firepower - and is composed of one or more TOs. TOs are objectives to which the ship, team or crewmember train. They are high-level objectives that, when achieved, indicate satisfactory accomplishment of the task (e.g., accurately classifying aircraft). EOs are lower level tasks or actions that, when performed correctly, allow the ship, team or crewmember to meet the terminal objective (e.g., was the IFF utilized to classify aircraft?). Multiple EOs may be needed to meet a given TO. MOPs are still lower level tasks or actions that the ship, team or crewmembers need to execute, which then determine whether an EO was performed correctly (e.g., was IFF utilized and all modes challenged to identify air contacts?). Multiple MOPs may need to be executed to meet the EO (see Figure 2; Lyons & Allen, 2000; Stretton, 2001).

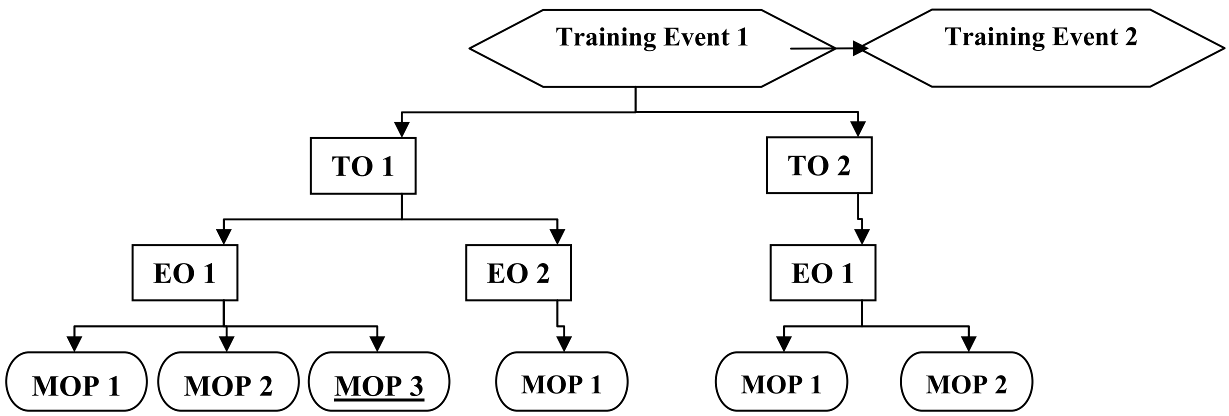

Figure 2. Example of hierarchy of OBT process. Note that TEs are not necessarily sequential.

Afloat Training Exercise and Management System (ATEAMS)

The OBT process was originally implemented manually (paper-based), but this method proved to be labor intensive. Therefore, a requirement to develop an automated tool dubbed the Afloat Training Exercise and Management System (ATEAMS) tool was issued. ATEAMS is a PC-based software application created with Embedded Visual Basic© (eVB). It is designed to manage data relating to basic training of teams and individual Naval crewmembers in both live and simulation-based training exercises, while adhering to the OBT process (Lyons & Allen, 2000). Stretton (2001) notes that ATEAMS

> "provides the capability to conduct training based on pre-defined objectives that are both measurable and traceable. Commands can use several paths for selecting objectives that include: Universal Naval Task List, Fleet Exercise Publications, mission selection, training team selection, watchstation selection, watchteam selection, and querying individual operator performance from previous exercises. This selection process … provides a simplified means to develop training scenarios that are traceable to selected objectives, as well as providing standardized methods to measure team and individual performance. ATEAMS capabilities include:
>
> • Support shipboard training teams
> • Plan training events
> • Generate objective-based training scenarios
> • Identify and generate data collection requirements
> • *Provide the means to gather performance data*
> • *Retrieve and integrate collected data and support debrief*
> • Provide the mechanism to conduct trend analyses
> • Provide feedback to chain of command
> • Provide feedback to schoolhouses
> • Provide feedback to Systems Commands
> • Support the administration of data for the above functions" (emphasis added, p. 1430).

The steps of the OBT process that ATEAMS supports are as follows. First, ATEAMS provides a historical database of trainee and team performance. This data can be used by a Naval command to help determine the training requirements for an upcoming training exercise. It can also assist the Naval command in determining the training audience for a given training exercise, for example the Combat Systems Training Team and/or the Damage Control Training Team. Once the training audience is identified, the command can then select training objectives from the Universal Naval Task List and/or Fleet Exercise Publications databases stored on ATEAMS or they can create their own training objects (Stretton, 2001). ATEAMS can then be used to create OBT Events (TEs) that are embedded within a training scenario package. The package would include elements such as TEs, TOs, EOs, MOPs, exercise location, environmental factors, geopolitical factors, and opposition force composition. The current intention is to then take the data generated by ATEAMS and feed it into the BFTT simulation system, which would use the information to simulate the exercise and stimulate trainees' shipboard instruments. ATEAMS would also provide a means by which instructors can collect data during a training exercise. Once the data is collected, the instructor can use it for debriefing purposes and/or later analysis. The data can also be used to update the individual and/or team's performance history. This history will help commands identify teams' strengths and weaknesses, providing them with a means of more precisely identifying training requirements (see the OBT cycle in Figure 3).

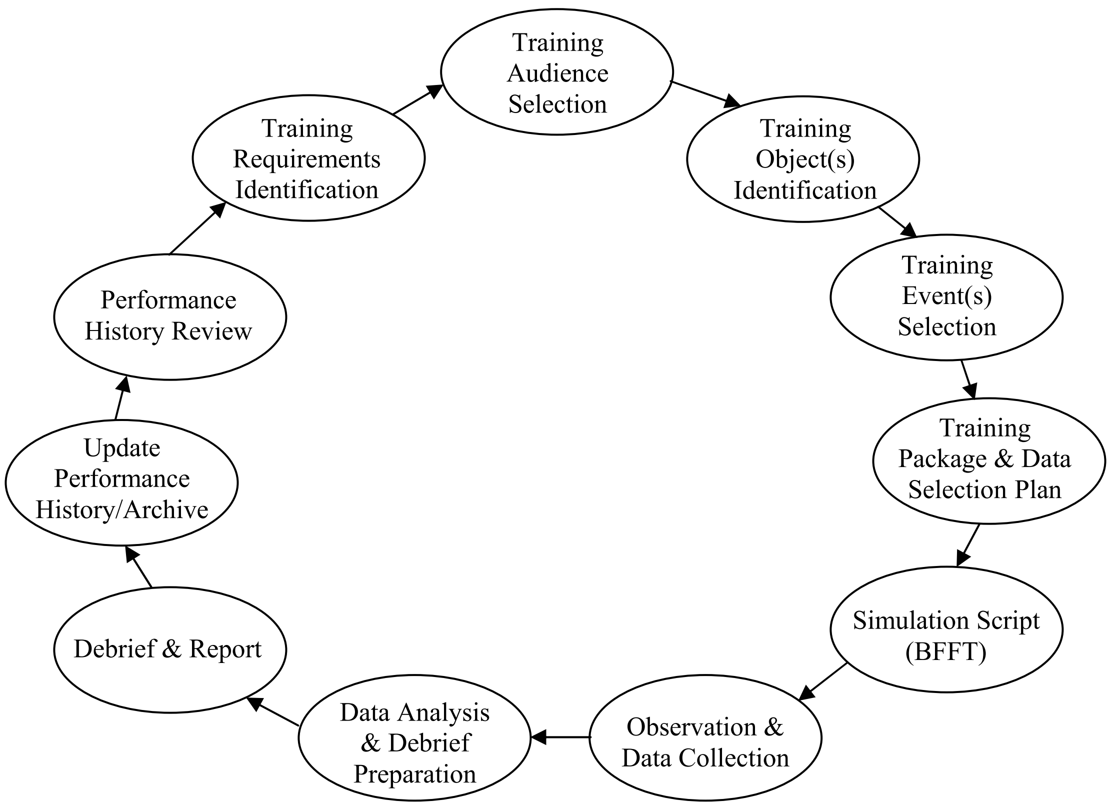

Figure 3. OBT cycle. BFTT drives the simulation system. Adapted from Stretton (2001).

As mentioned, paper-based data collection methods are extremely labor-intensive, both during and after training exercises. In addition, paper-recording methods can lead to inefficient scoring and errors. In the former case, the instructor must take time to flip through sheets of paper during a training exercise, time that could be used to observe performance or provide feedback to the trainee. In the latter case, the instructor may misplace one or more scoring sheets, leading to incorrect data analysis. Therefore, one requirement associated with ATEAMS was the development of a hand-held automated performance assessment tool (Lyons & Allen, 2000). The concept was to link the ATEAMS databases to a hand-held device and download instructor-relevant data (e.g., TOs, EOs, and MOPS) onto this device. An application loaded on the hand-held device would provide an interface that would allow the shipboard instructor to easily capture performance, and other, data. Once captured, this data would be uploaded back into the ATEAMS database to be used for team debriefs, trend analysis, as well as updating ATEAMS' performance history database.

Hand-held Devices

There are important factors that must be taken into consideration when selecting a hand-held device, especially when applied to the military training environment. These factors include memory capacity, the operating system, screen size and type, battery type and storage capacity, ruggedness, and data import and export capabilities (i.e., docking station, cable, modem, and/or infrared connection) (Weber & Roberts, 2000). Initially, pen tablet computers were considered as the platform of choice for the ATEAMS data collection tool. However, pen tablet computers are relatively expensive, are heavy, weighing up to four pounds, and can be cumbersome in the confined spaces found in shipboard environments. Therefore, the Navy is currently investigating the use of more portable, lightweight electronic tools for data collection. One such tool is the Personal Digital Assistant (PDA).

PDA's fall into two general categories: Palm-style organizers and Pocket PCs such as Compaq's iPAQ™ (Consumer Reports, 2001). Pocket PC's are more like mini-computers than traditional PDAs. For example, Pocket PCs have much faster processors, are loaded with familiar applications like Excel™, and, unlike many Palm-style organizers, have color displays.

Within each category, key aspects differentiate one hand-held device from another. Some of these factors include processor speed, memory capacity, battery convenience, display quality, and ease of use. Depending on the model, memory can range from 2 to 32 MB. The battery should be examined for two factors: the life of the battery before replacement or recharging is necessary and the method by which replacement is made. That is, some batteries are rechargeable while others are replaceable. If rechargeable, some PDA manufactures require that the unit be returned to the factory for battery replacement. The quality of the display also differentiates PDAs. Factors in this area include display size, resolution, and color capability.

PDAs and Pocket PCs now have the capability to incorporate multiple tools and functionalities into one relatively lightweight and affordable unit. Many of today's smaller hand-held devices, such as cell phones, have capabilities that include Internet access, digital camera and video/audio recording, ebook, MP3 recording, in addition to cell phone capabilities. Many have the capability to easily record, import, export, and manipulate data. These capabilities are being incorporated into hybrid units (i.e., cell phones with PDA functionalities). Although still requiring usability improvements, hand-held devices can be used for flexible and non-intrusive data collection. Data can be entered through an on-screen electronic keyboard, through handwriting (either natural or script) and/or by voice recording. For recording longer responses or detailed observations, Gravlee (2002) recommends that an external keyboard be used. Hand-held devices can also be loaded with custom data collection software, greatly expanding their utility in training and data collection settings. However, the more functionality added, the higher the associated costs will be to memory, battery life, weight, affordability, and perhaps durability and usability. Nonetheless, the number of hand-held devices with high functionality as well as affordability is growing.

Compared to pen tablet computers, PDAs are lightweight and inexpensive. However, the pen tablet computer has higher computational power, storage capacity, and a larger screen. If the PDA is the platform of choice for the Naval training community, then the challenge for the training application developer is to develop PDA applications that can compensate for the devices' weaknesses and/or capitalize on its strengths. One way to help ensure this is through the use of sound human factors and usability analyses during the development of the training application. For example, a usability analysis can determine whether a PDA-based training application is easy to use or if the displayed graphics and text are readable. Ease of use and readability are important; else the instructor, while attempting to step through a complex application or decipher graphical/textual data, may miss critical events occurring during a training exercise.

To test the feasibility of using a PDA for shipboard training exercises, researchers at Sonalysts, Inc., in conjunction with NAVAIR Orlando Training Systems Division, developed a prototype data collection application ATEAMS PDA (APDA). Through this application ATEAMS, hosted on a PC, could be synched with a PDA. Once synched, training-relevant data such as TOs, EOs, MOPs, and Rules of Engagement (ROE), could be downloaded from ATEAMS to the PDA.

For shipboard exercises, the current intention is that each instructor would have a PDA containing data that would be applicable only to the team that the instructor was evaluating (e.g., engineering or the combat information center). The instructor could use the information stored on the PDA for briefing purposes, for referencing scenario-related material during the training exercise, as well as for capturing data through APDAs' Graphical User Interface (GUI). Once collected, the instructor could use the tool to access the captured data for debriefing purposes. The data could then be uploaded back into ATEAMS databases for later analysis and for updating a team's performance history.
Once the ADPA application prototype was completed, researchers at NAVAIR Orlando Training Systems Division subjected it to a usability evaluation. Usability evaluations can reveal critical and non-critical design flaws in hardware and software and, therefore, was chosen as one method for evaluating the APDA application.

Usability

Usability reflects the extent to which users of a given system can use the functionality of that system. Usability has many components, including how easily a system can be learned (learnability), how easy a system is to remember (memorability), the degree of efficiency that can be obtained after the user has learned the system (efficiency), the error rate (errors), and the subjective satisfaction of using the system (satisfaction) (Nielsen 1993). A usability evaluation generally consists of four phases: a task analysis, a heuristic evaluation, user testing session(s), and design or redesign recommendations. The current evaluation falls primarily under the last three phases of the usability process. That is, due to resource limitations, a task analysis was not conducted. However, SMEs were consulted regarding the nature of training onboard ship. This knowledge was applied to the design of the usability testing sessions.

Method

The current evaluation included two heuristic evaluations, user testing sessions, and redesign recommendations. The evaluation was conducted at NAVAIR Orlando Training Systems Division, located in Orlando Florida, in February 2001.

Participants

Three participants took part in the evaluation. All were male and had prior Navy service (two retired Chiefs, one retired Captain) with backgrounds similar to APDA end-users. All are from the surface community with 15 to 25 years of experience in the training community (mean = 20 years). All participants indicated a high or very high level of experience with Windows®-based applications.

Materials and Equipment

Four PDA models were considered for the current evaluation. The iPAQ™ H3650, manufactured by Compaq®, was chosen as the test bed for the ATEAMS PDA software (see Figure 4). This device was

Figure 4. Compaq® iPAQ™ with and without slipcase.

selected because, of four devices considered, it was the only Pocket PC. Pocket PC's are more like mini-computers than traditional PDAs. For example, Pocket PCs have much faster processors, are loaded with familiar applications like Excel™, and, unlike many Palm-style organizers, have color displays. The iPAQ™ H3650 runs on Microsoft Windows® CE with an ARM SA1110 processor with 31.15 MB of main memory.

APDA Application Windows. There are three main windows associated with the APDA application – these are labeled Prebrief, Collecting Data, and Debriefing. Four additional windows can be activated through the Collecting Data window. These include a Timeline window, an Assessment window, a window labeled Incomplete?, and a Comment window. A brief description of the functionalities associated with each window follows.

Prebrief Window. This window contains information that an instructor may want to access before or during a training exercise. This includes Training Team Assignment (e.g., Combat Systems Training Team), Trainees (names & rates), Timelines (displays the onset time of each TE), Lessons Learned (from previous training exercises), Safety, Rules of Engagement (ROE), Scenario Summary (displays the Training Package name, Mission Statements, Current Situation, and Tactical Objectives), and Objectives (displays the TEs, TOs, EOs, and MOPs, formatted in a tree view control) (see Figure 5).

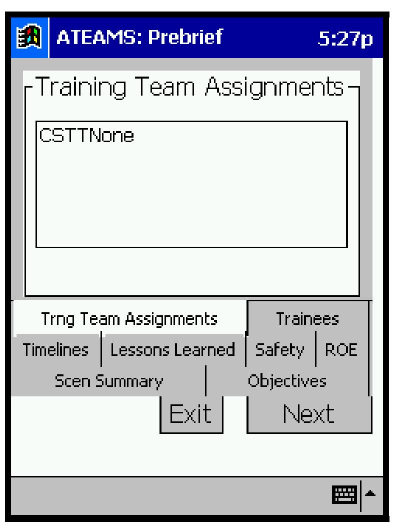

Figure 5. Prebrief window.

Collecting Data Windows. At the top of the Collecting Data window are two radio buttons labeled Timeline and Assessment, which allow the user to toggle between a window displaying the timeline of TEs and a data collection window. The Timeline window contains a list of the start time of all TEs, given in scenario time, using an hours:minutes:seconds format. The instructor can use the timeline information to better manage his time, allowing him to focus on the trainee(s) whose performance will be affected when the start time of a given TE is reached. The Assessment window is used to collect performance data during a scenario run. The Assessment window displays the current TO, EO and associated MOPs. The TO or EO can be viewed by clicking a gray-colored toggle command button (currently set at TO as seen on the middle window in Figure 6).

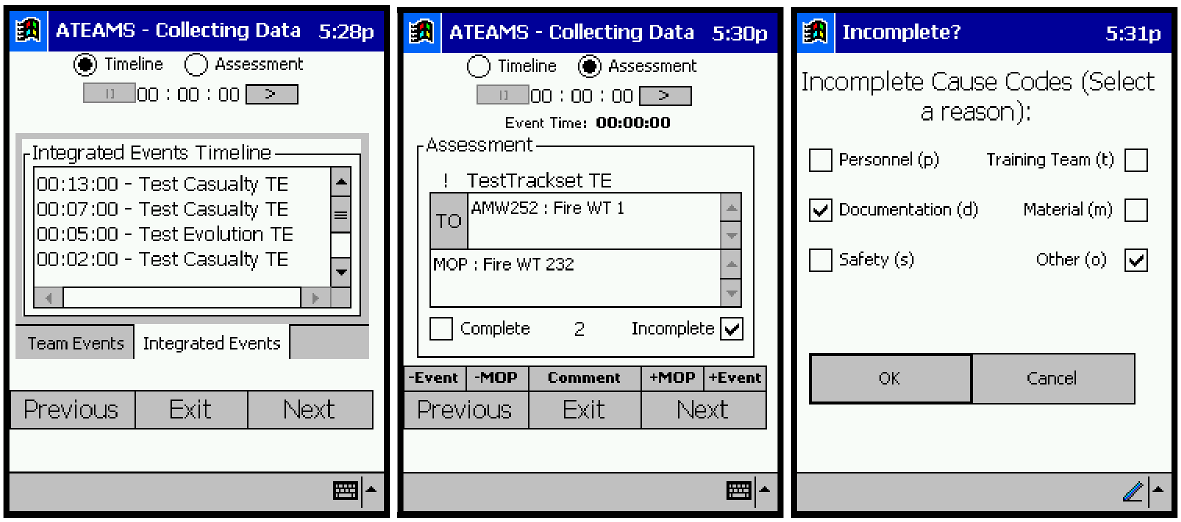

Figure 6. Left to right – Timeline, Assessment and Incomplete Cause Codes windows.

The instructor uses the Assessment window to rate trainee and team performance. To accomplish this, checkboxes labeled Complete or Incomplete are used. If the Complete box is checked, the instructor is indicating that an MOP, relating to a given EO and TO, has been completed. If the Incomplete box is checked, a window labeled Incomplete? appears. Through this window, the instructor can check up to six cause code checkboxes to explain why the MOP was not completed (see Figure 6). Therefore, the performance measurement currently provided by ATEAMS is essentially dichotomous – either a trainee did, or did not, complete an MOP task. After checking the appropriate cause code checkbox(es), and clicking the OK button, the instructor is returned to the Assessment window. Command buttons, located near the bottom of this window and labeled with a plus (+) or a minus (-) sign, are used to display the next or previous TO's, EO's and MOPs. Scroll bars can be used to view the text of long TO's, EO's or MOP's. Navigation buttons, labeled Previous, Next, and Exit, are located at the bottom of both Collecting Data windows. These buttons are used to close one main window and open another (e.g., when the Prebrief window is open, clicking Next will close the Prebrief window and open the Collecting Data window).

When the Comment button, located on the Assessment window, is clicked, a Comment window appears. A comment can be entered through the Pocket PC keyboard or Microsoft's® block or letter recognizer software. The latter method requires the instructor to learn how to write alphanumeric characters based on the software's rules. Each character is written, one at a time, in an input panel area (see Figure 7). When text

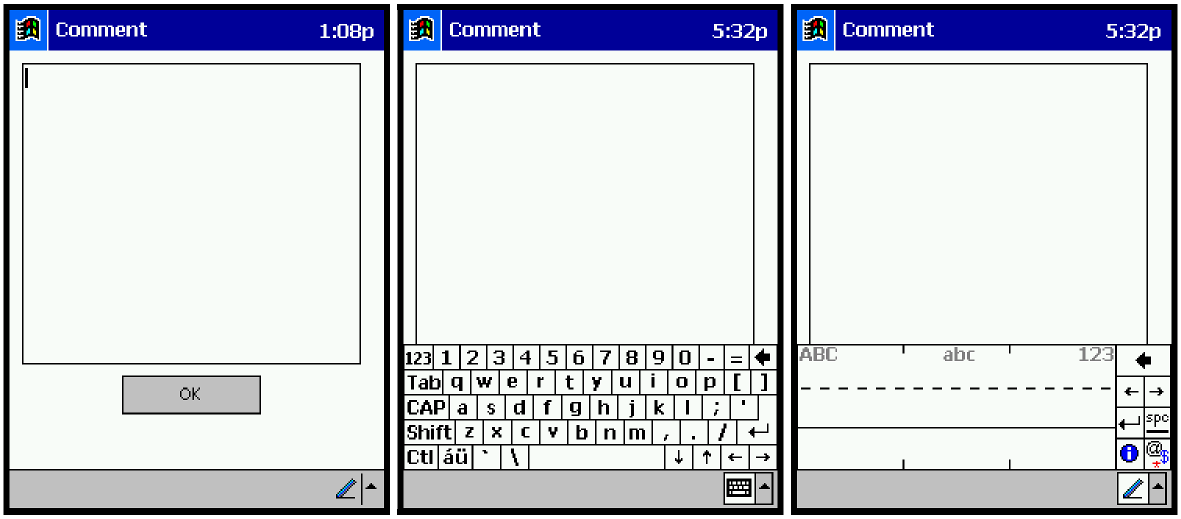

Figure 7. Comment window: with keyboard (lower center) and input panel area (lower left).

is entered, through either method, it is displayed as typed characters in a large text box. Clicking the command button labeled OK will save the comment, time-stamp it, and link it to the active TO/EO/MOP. The instructor is also returned to the Assessment window. The Comment window provides a method by which the instructor can explain and/or augment the selected cause codes measures.

Debriefing Window. The Debriefing window lists all TE's, TO's, EOs, and MOP's in a tree view control. The words complete and incomplete are used to indicate whether the procedure/task described by a MOP was completed or not (see Figure 8).

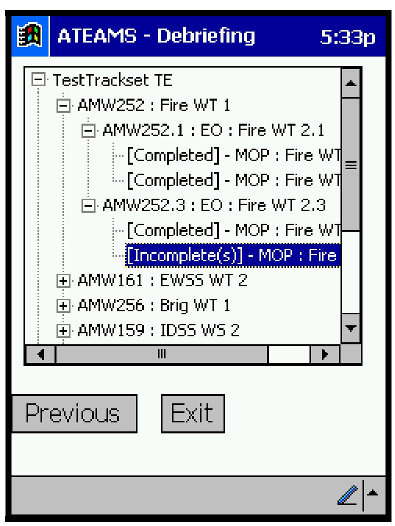

Figure 8. Debriefing window. Top eight lines illustrate a TE, one TO, two EOs, & two MOPs for each EO.

Heuristic Evaluation Procedure

The heuristic design principles applied to the evaluation of the APDA were derived from various sources (Eberts, 1994; Hamel & Clark, 1986; Lynch & Horton, 1999; Mandel, 1997; & Nielsen, 1993). Seven heuristic principles were employed: speak the user's language, minimize users' memory load, provide consistency, prevent errors, provide adequate help and documentation, simplicity, and progressive disclosure (see Appendix A for definitions of these heuristics).

Two human factors experts examined the APDA, evaluating it against the seven guiding principles. Violations of the heuristics were identified and categorized under the three main APDA windows (Prebrief, Collecting Data, and Debriefing) as well as the Collecting Data sub-windows (Timeline, Assessment, Incomplete Cause Codes, and Comment). Note that a heuristic evaluation conducted by one usability expert will reveal approximately 35% of the usability flaws of a given system (Nielsen, 1993). Therefore, between 35% and 70% of APDA usability flaws may have been detected through the current evaluation.

Although it is advisable to fix all identified violations, due to time and financial constraints it is not always feasible. The usability process can be adjusted for this problem by prioritizing each heuristic violation. For example, a priority level of high, medium or low can be assigned to each violation. Violations in the high priority category are issues that are assessed as severely hampering usability; thus, these violations should be resolved through redesigns prior to fielding the system. Medium priority violations should be addressed through redesigns, but are less critical than high priority violations. Low priority violations violate heuristic guidelines, but it is uncertain whether they would impede practical use of the system. Therefore, low priority violations should be addressed, but not if doing so delays the release of the system. The assignment of a priority level to an identified heuristic violation is based on the usability professionals' judgment as to the expected impact each violation will have on user performance. User testing is then employed to confirm or disconfirm the professionals' judgments. User testing sessions can also identify violations missed by the usability professional during the heuristic evaluation process.

<u>User Testing Procedure</u>

The participants in this study had taken part in a previous evaluation of the ATEAMS software (see Ricci, Allen, Reynolds, Daskarolis-Kring, & Hodak, 2001). During that evaluation, the participants were asked to develop a training scenario involving the Combat Systems Training Team and Damage Control Training Team and to perform specified tasks using the APDA software. The objective of the current evaluation was for participants to use and evaluate the GUI, as well as various functionalities, in each APDA window. The evaluation tasks consisted of opening and examining prebriefing material, collecting MOP data, and reviewing the collected MOP data in the Debriefing window (see Appendix B for the instruction set). Additionally, the participants were asked to write a comment on the PDA using the two methods that could be used to input comments. The two methods were the iPAQ's ™ keyboard and Microsoft's® block or letter recognizer software. The participants were asked to verbalize their thoughts about the usability of the PDA and the APDA throughout the evaluation.

APDA Usability Questionnaire. An APDA questionnaire was constructed for this evaluation. This questionnaire was administered immediately after user testing. The questionnaire consisted of four fill-in-the-blank questions, 32 five-point likert-scale questions, and three open-ended questions. Questions 1 – 20 were adapted from the 5NINES Usability Survey by Motorola®. These questions are converted and summed, yielding a score that ranges between 0 – 100 points: 0 equaling very low usability and 100 equaling very high usability. Questions 21 – 32 were adapted from the writings of various usability experts (Eberts, 1994; Hamel & Clark, 1986; Lynch & Horton, 1999; Mandel, 1997; and Nielsen, 1993).

The APDA usability evaluation produced subjective data only. This data consisted of the results of the heuristic evaluation and of the usability survey as well as user's comments. An attempt was made to collect objective data (i.e., number of errors made and elapsed time taken for a given task). However due to equipment limitations, it was quickly determined that this data could not be accurately collected because the researchers did not have access to videotaping equipment. An attempt was made to record time and error data by hand but this proved to be too difficult, given the size of the Pocket PC display.

Results Part 1: Heuristic Violations

A total of 37 heuristic violations were detected. Of these, five were discovered during the user testing sessions with the remainder detected by the Human Factors experts during the heuristic evaluations. A summary of the violations can be found in Table 1. Of the 37 heuristic violations, 25 were categorized as either medium or high priority violations. Note that all of the violations identified under the Prebrief window were rated as low priority violations.

	Users Language	Minimize Memory Load	Consistency	Prevent Errors	Help	Simplicity	Progressive Disclosure	Total: Rows
All Windows		1	2	3	1	2		9
Prebrief Window	1		1	1				3
Data Collection Windows		4	8	4		5	1	22
Debrief Window		1	1			1		3
Total: Columns	1	6	12	8	1	8	1	37

Table 1. Summary of heuristic violations, by window and heuristic.

The following three sections delineate the results of the usability evaluation process. The three sections list the heuristic violations that were common to two or more windows of the APDA application or violations that were specific to a given window (i.e., the Collecting Data windows or the Debriefing window). Within each of these three sections, each violation is number and described. It is then categorized under one of the seven heuristics used during the evaluation and is also assigned a priority level (i.e., High or Medium). User testing data (i.e., user comments and results from the usability questionnaire) that validated the observed heuristic violation, if any, is then given. Finally, redesign recommendations are given. These recommendations are examples of how a violation *may* be addressed, but are not necessarily the *best* solution. That is, the developer may have a more in-depth understanding of the capabilities and limitations of the system being evaluated relative to the usability professional's knowledge of the system. Therefore, the developer may know of a more elegant solution to a given violation (or may know that fixing a violation may not be possible, given system limitations, time constraints and/or budgetary constraints).

<u>Violations Common to Two or More APDA Windows</u>

1. Clicking the Exit button immediately closes the APDA software. No error check message is provided. This increases the possibility that a user may accidentally exit the program.

Heuristic Violation (High): Preventing Errors

User Testing: While looking for a button to back out of the Collecting Data window, one user accidentally exited. The same user accidentally exited from the Debriefing window. His comments include "…oops, where did it go? (accidentally hit Exit). The tendency is that you hit right there first. It would be nice to be able to give confirmation (in wanting to exit)." A second user also accidentally exited from this window. His comments include "That was not good. I hit the wrong exit button I think. Whatever I hit was wrong."

Recommendation: Use the standard Windows® error-checking pop-up window with appropriate text. For example, "Do you want to Exit ATEAMS?" OK/Cancel.

2. The designers of APDA may have purposely employed large command buttons to make it easier for the user to click. The proximity of many buttons could lead to errors produced by mis-clicks (see Figure 9). However, even the smallest buttons used in the APDA software seem to be easy enough to

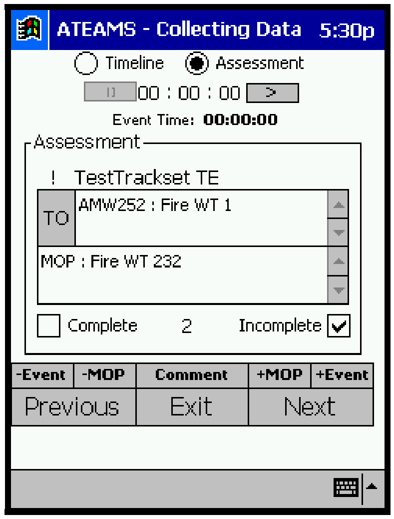

Figure 9. Close proximity of command buttons, like the Exit and Comment buttons, could lead to mis-clicks.

activate; for example, the Timeline/Assessment radio buttons.

Heuristic Violation (Medium): Preventing Errors

User Testing: The mean rating for question 22 of the usability survey, “It was easy to select the checkboxes, buttons, tabs, etc. of ATEAMS HHD”, was 4.0. This indicates that the users, on average, agreed with this statement (4.0 = agree with statement). This rating suggests that reducing the size of the command buttons may have no adverse impact on usability since other APDA buttons are already relatively small.

Recommendation: Reduce the size of the Exit/Next/Previous buttons and insert some space between all buttons. This is especially critical for the Exit button to help prevent the user from accidentally clicking this button and exiting the application.

3. APDA provides no access to a help function. This is inconsistent with other applications. A simple help function may be useful to the user, providing guidance when the user needs help with the application.

Heuristic Violation (High): Providing Adequate Help and Documentation and Consistency

User Testing: The mean rating for question 31 of the usability survey, “I need no help when using ATEAMS HHD”, was 1.33. This indicates that the users, on average, strongly disagreed with this statement (1.0 = strongly disagree with statement).

Recommendation: Add a help function. Due to limited screen space and memory, the help function will likely need to be simpler than help functions found in standard Windows®-based applications.

4. Users wanted the APDA screens to be as simple and efficient as possible.

Heuristic Violation (High): Simplicity

User Testing: One user, commenting on the method used to scroll through the MOPs/TEs on the Assessment window, stated, "Given that I've developed these MOPs, I'd like to go directly to an MOP. Is it the case that I have to scroll through five or ten of these and the sub-elements beneath? I don't like that. If it's taking a while I'm going to miss a lot of activities that are taking place, missing a lot of behavior that should be observed or recorded." Commenting on the Debriefing window tree view control, another user stated, "Gee, I have to hit plus, plus everywhere".

Recommendation: Adding a drop-down menu may make it easier for the user to find/select the TEs, TOs, EOs, and MOPs from the Assessment window. A drop-down menu may also help the user more quickly locate a given TO, EO, or MOP when using the Debriefing window.

5. On both the Prebrief and Debriefing windows, the user must click on the plus signs used in the tree view control to expand the hierarchy of TEs, TO's, EO's, and MOPs. This is time consuming and may frustrate the user.

Heuristic Violation (Medium): Simplicity

User Testing: One user commented, "Gee, I have to hit plus, plus everywhere."

Recommendation: Include 'Expand all' and 'Collapse all' options. Use a drop-down menu that allows the user to filter the displayed data (e.g., display MOPs only).

6. The timeline in both the Debriefing and Collecting Data windows displays time in scenario time, given in hours, minutes, and seconds (see Figure 10). The time is associated with the point within the scenario that a given TE will begin (this follows the OBT model, i.e., using pre-scripted events in order stimulate and then rate specific areas of trainee performance).

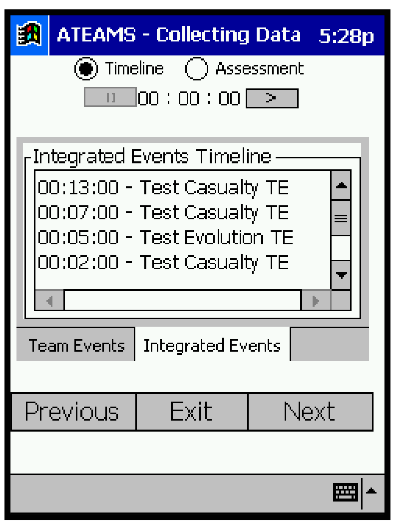

Figure 10. Integrated Events Timeline from Collecting Data window.

Heuristic Violation (Medium): Speak the User's Language

User Testing: One user stated, "It gives me 00:17:00 – that means 17 minutes to analyze and plan? …I'm not used to seeing it like that. 00:17:00 - to me I would read it as seventeen hundred. I would make it plain and simple. Make it 17 minutes."

Recommendation: Use the words 'TE Start Time (hours:minutes:seconds)' as a descriptor of the displayed time.

Violations Relating to the Collecting Data Windows

1. The user should be able to quickly access the data collection functionality, as this is the primary purpose for APDA. When APDA is started, the initial window is the Prebrief window. However, there is no clear indication that the next window is the Collecting Data window, nor is it clear that the user must click the button labeled 'Next' to open that window. This is also the case when a user is finished with data collection and wants to open the Debriefing window. During user testing, it was discovered that several of the users were confused by the textual descriptors or symbols associated with the command buttons (i.e., by the textual descriptors of Next and Previous and by the symbols of + and – associated with the MOP and TE buttons). For example, while on the Assessment window one user clicked the button labeled Previous. This act closed the Assessment window and opened the Prebrief window. The user stated that he thought clicking the Previous button would take him to the previous MOP.

Heuristic Violation (High): Simplicity

User Testing: After exhibiting confusion on how to move forward through the MOPs, a user commented, "Do I hit 'Next' and the next guy comes up…I put complete, and I was saying next, next MOP. Right? That doesn't really make sense." He then tapped the button labeled Next instead of + MOP, which took him to the Debriefing window. Another user also experienced difficulty on how to move from a completed TO to the next TO. His comment was, "Do I go to the next event here?" (He was then directed to the ± MOP/Event buttons).

Recommendation: The Next and Previous buttons can be relabeled to reflect the window that they will open, for example, Prebrief or Collect Data. These labels will need to be abbreviated. To capitalize on a users experience with symbols (e.g., used on a video recorder), an arrow scheme may also be employed for the MOPs and TE's. An example would be as follows: ⇦ MOP. ⇨

2. When the user opens the Collecting Data window, the initial window is the Timeline window. This seemed to confuse the users. During testing, the users often searched for the data collection/assessment portion of the program and had to be provided hints on how to find it.

Heuristic Violation (Medium): Simplicity

Recommendation: Consider making Assessment the default window of the Collecting Data window. If this is not possible, consider making the font for Timeline or Assessment larger or darker (bold).

3. Proper use of screen real estate is critical in any text-intensive PDA application. Two factors affected by PDA screen real estate are readability and comprehension. Readability refers to how easily a user can read displayed text. Comprehension refers to how well a user can comprehend and remember what is/was displayed. Font size, font type, illumination, glare, contrast, and distance from the reader's eye to the text, among other things, affect text readability. In APDA, some of these elements also likely affect reading comprehension. However, comprehension is also affected by the simplicity of the text/sentence and the amount of text that is viewable at once. For example, the text that is displayed in the Debriefing window does not wrap. The result is that the user may have a difficult time in comprehending/remembering long sentences, that is, the user may have to scroll left and right several times to view and comprehend the sentence.

Heuristic Violation (High): Consistency and Minimize User's Memory Load

User Testing: In terms of text scrolling off screen, one user commented, "Well I have to scroll everything, but that's real estate. I'd rather see it wrapped." In addition, the mean rating for question 30 of the usability survey, "I did not have a problem with the length of some of the text, i.e., text that required scrolling", was 2.33. This indicates that the users, on average, disagreed with this statement (2.0 = disagree with statement).

Recommendation: The best solution may be to wrap all text. Wrapping text provides an additional benefit. That is, by wrapping text the horizontal scroll bar found on the Timeline window can be eliminated. The resulting empty space could be used to enlarge the text box vertically, allowing at least one more line of text to be displayed within the text box.

4. The formatting of the text boxes reduces the amount of space available for text, both on the Timeline and Assessment windows. The formatting between the two windows is also inconsistent (see Figure 11).

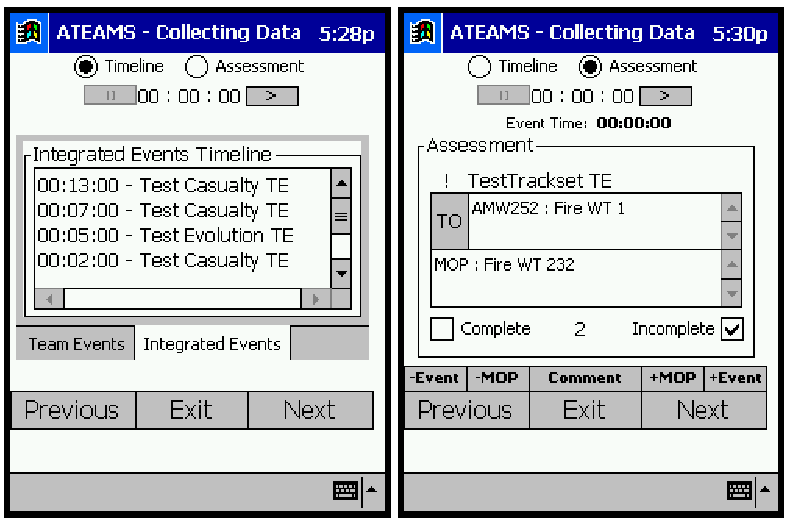

Figure 11. Examples of wasted screen space. Shrink & move tab/command buttons. Reformat textbox area.

Heuristic Violation (High): Consistency and Minimize User's Memory Load

Recommendation: The gray borders of the text boxes should be removed with the text box enlarged horizontally such that its borders touch the left/right edges of the Pocket PC window. The controls tabs (Team Events, Integrated Events) and command buttons (Previous, Exit, Next) can be reduced in size and moved downward, increasing the amount of vertical space available for the text box. Finally, the textual labels of Team Events, Integrated Events Timeline, and Assessment, which appear just above the text boxes, are redundant (e.g., to open the Assessment window, the user clicks the radio button labeled 'assessment'). This text can be eliminated, increasing the amount of vertical space available for the text box.

5. The clock control buttons consist of a reset button and a button that toggles between start and pause. No error check message is provided when the reset button is clicked. Thus the user can accidentally click this button and reset the scenario clock to zero.

Heuristic Violation (High): Prevent Errors

Recommendation: Provide an error check message when the reset button is clicked; for example, "Reset scenario clock to zero?" OK/Cancel.

6. The Comment window is composed of an input panel area, located on the lower portion of the Comment window, and a text box display area. The input panel area is used to input text into the Pocket PC; the textbox area displays the text. The user can choose from one of two methods to input text - the user can tap on a virtual keyboard or use Microsoft's® block or letter recognizer software. Once a comment is written, the user saves it by clicking a command button labeled 'OK'. The keyboard seems to be easier to use, but also seems to take more time to input characters. The block/letter recognizer method forces the user to learn how to write letters based on the rules of the software. During the heuristic evaluation, it was noted that this method seemed to lead to more input errors compared to the number of input errors made with the keyboard.

Heuristic Violation (High): Simplicity and Prevent Errors

User Testing: While using the keyboard, a user commented, "It's hard to type." He then began trying to write (erroneously) on the blank comment window. A second user commented about block/letter recognizer method, "Well gee, I wrote on an 'n' and I got a w, l, and v. Let's say that part is not very good, now you're going to train them to write script again and that's not good. When you're putting a training team together you don't have a lot of time". This user was then asked to comment about the keyboard, "The keyboard is OK". He was then asked to comment about the use of digital ink, which had been used for capturing text in other training applications developed at NAVAIR. Digital ink is essentially a drawing application that can capture text exactly as it is written and then store it as a bit map image. "That would be cool, because if I could go to comments and start scribbling straight in there. It would be cool and a lot more useful, instead of me going to this one little keyboard, and having to do the little typing and all that stuff. And it's rapid. You have to realize that as soon as he (a trainee) doesn't do something we might need to stop the drill and I want to make comments about that." A third user commented, "With Palm® you have what they call Graffiti. Some of those characters I have trouble remembering. But I would much prefer being able to write it out".

Recommendation: Rather than forcing the user to learn the format of the block/letter recognizer method or tap on a small keyboard, employ a digital ink method for collecting notes. This method would be the easiest and most accurate method to use. Several software applications have been developed that can be used to create digital notes. For example, Gonna Software® has created a Pocket PC application called PocketStickey, which allows the user to write notes directly on the Pocket PC screen and then link the note to a designated file. Determine if this, or another, application can be used to link a digital ink note to the MOP data collected through APDA. In addition, Microsoft® has created a handwriting software application for Pocket PCs called Microsoft's® Transcriber Version 1.1 for Windows® CE. This application can be tuned based on an individual user's writing style. That is, a user views a representative set of each letter of the alphabet and numbers 0 – 9 and selects, from six different examples of each letter or number, the style that most closely matches his/her own writing style. Investigate this application to determine its error-rate. Military personnel are unlikely to accept error rates higher than 0.5% - 1.0%.

7. To enter a comment, the user selects one of the text entry methods mentioned above. However, all methods cover the OK button of the Comment window, which is used to save the text of a comment (see right two images in Figure 12). To uncover the OK button while in the keyboard mode, the user must click

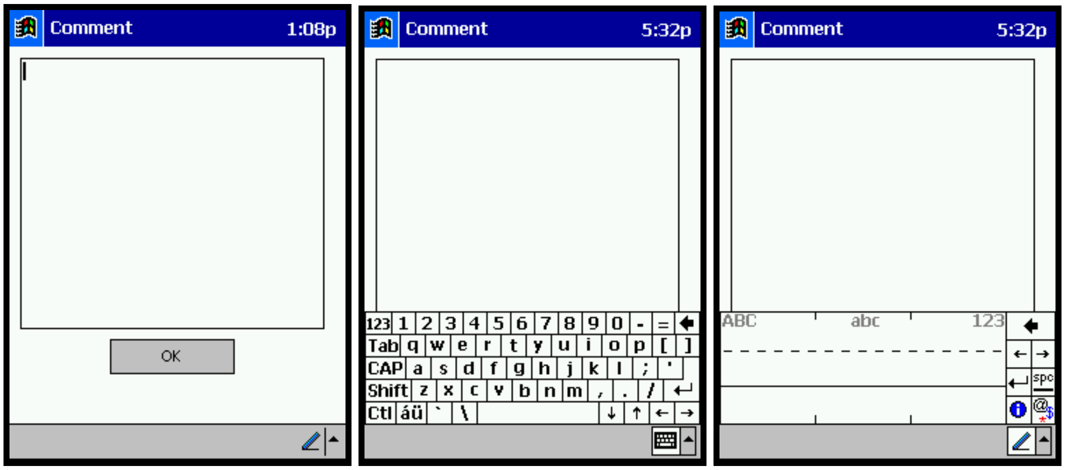

Figure 12. OK command button (left) covered by keyboard (center) and input panel area (right).

on the keyboard icon. There is no indication on screen that this is the method used to minimize the keyboard. If in the block or letter recognizer mode, the user must click a green pen icon to minimize the input panel area. In this case, the background color of the icon changes, depending on the state of the input panel area (gray for inactive, white for active). It is unlikely that the user will recognize this fact. More seriously, if the user accidentally clicks another button – for example, the demo button located to the right of the blue information button - it is quite easy for the user to get lost in the resulting windows and not be able to find his way back to the comment window.

Heuristic Violation (High): Simplicity and Prevent Errors

User Testing: After first using the keyboard, then the block/letter recognizer software, one user had to be shown how to find his way back to the Comment window. That is, he became lost and was unable to save his comment without assistance.

Recommendation: Ensure that the OK button is always displayed. In addition, rename the 'OK' button to 'Save', since that is the operation the user is performing.

8. There is no Cancel button on the Comment window (see Figure 12). This is inconsistent with most Windows®-based applications and could lead to errors during debrief. That is, the user may wish to close the Comment window without leaving a comment. If no comment is entered, the only option the user currently has is to click the button labeled 'OK'. Once the OK button is clicked, a blank comment is inserted into the Debrief database.

Heuristic Violation (High): Consistency and Prevent Errors

Recommendation: Add a Cancel button.

9. The user may wish to edit and/or delete comments before data collection has stopped (see Lyons & Allen, 2000). In APDA, once comments are entered there is no method to recall and edit them. This is inconsistent with other Windows®-based applications.

Heuristic Violation (High): Consistency

Recommendation: Provide a method through which the instructor can recall and edit notes during a training exercise.

Usability Issues Relating to the Debriefing Window

1. The Debriefing window uses a tree view control to display all TE's, TOs, EOS, and MOP's and text to indicate whether the MOP was completed or not. About 40 alphanumeric characters can fit on the window. Unfortunately, the tree view control structure, combined with the length of the MOPs, means that most of the MOP text is off screen and can only be viewed through the use of a horizontal scroll bar. For example, some MOPs are 19 words long. As noted under the Collecting Data section, this formatting may increase the time it takes for the user to view/comprehend the displayed TEs, TOs, EOs, and/or MOPs.

Heuristic Violation (High): Minimize User's Memory Load

User Comment: "Gee, I have to hit plus, plus everywhere." Another user commented, "Well I have to scroll everything, but that's real estate. I'd rather see it wrapped." In addition, the mean rating for question 30 of the usability survey, "I did not have a problem with the length of some of the text, i.e., text that required scrolling", was 2.33. This indicates that the users, on average, disagreed with this statement (2.0 = disagree with statement).

Recommendation: Here we see one of the key limitations of a PDAs, that is, the size of its display screen. Based on the text length that appeared on the APDA, it would seem that scrolling horizontally would tax the users memory load to a greater degree than if the text wrapped. However if the length of the text is long, then wrapping the text may also unduly burden the user's memory load because the tree view control displays all events and objectives, which could number into the hundreds. Ideally no scrolling should be used for an individual MOP. For this to occur in the current configuration, however, the font size would have to be severely reduced, perhaps to an unreadable level. In addition the readability of a given display depends on, among other factors, the amount of stress the user is under, the viewing conditions (light levels, vibration), and quality of the display (Wickens, 1992). No data could be found that directly examined the effects that such factors as text size, scrolling method, stress and/or viewing conditions had on memory or what tradeoffs might be made. Related data would seem to indicate that the use of wrap-around text, drop-down menus, or some other method or combination of methods might be in order (The Windows® Interface Guide, 1995). Additionally, during debriefing the MOP is likely to be the main data point that the instructor would access. If so, it may be useful to separate the TO/EO from the MOP. That is, place the former in a drop-down window, time sequenced, and let the instructor select the TO/EO from this menu (the TO/EO could also be sequenced alphabetically). Then, display the MOPs for the selected TO/EO in a tree view control. It may be useful to provide both options to the instructor (e.g., a 'display all' option), because some instructors may prefer to see the TO/EO/MOP relationships.

2. The MOPs displayed in the Debriefing window have text that indicate whether an MOP was completed or not and, in the latter case, small alpha characters that represent the cause code(s) associated with the reason(s) an MOP was incomplete. However, there is no template to guide the user for deciphering the cause codes. In addition, there is no indication if a comment had been entered for a given MOP nor is there a method for the user to access the comment associated with an MOP (see Figure 13).

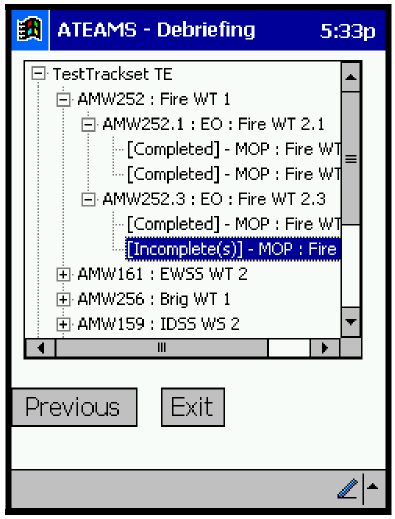

Figure 13. Debriefing window with complete/incomplete text and cause code (s in highlighted example).

Heuristic Violation (High): Consistency and Simplicity

User Comment: One user commented, “I guess would want to have some indication as to where I've made a comment - without having to look through every one of them.” A second user commented, “Say there is one incomplete, where do I see the comment?” (explained that he can't from here). "Why not? I went through all that process of writing stuff in so that in debrief I can see it. No, it should have a link right here. Put a little tab, something so that the comment opens up. If you say incomplete and there are twelve people, they want to know why it's incomplete. How are you going to learn if you don't know why it wasn't complete? To me that's really big fault there." A third user commented, "Let me see if I can find one that has an incomplete. That's all it's going to tell me? Where are my personal notes? That's got to be in there. Otherwise, I just might not remember, especially if there are a number of incomplete."

Recommendation: Provide a cause code template to the user (perhaps through a help button). Add a comment code to the MOP so the user will know which MOP has a comment associated with it. Provide a method through which the user can display his comments; for example, by tapping on the MOP.

Results Part 2: APDA Usability Survey

An APDA usability survey was constructed for the evaluation. It was administered immediately after user testing. The results of the survey can be found in Table 2.

Question (based on a five-point likert scale: 1 = Completely Disagree; 3 = Neutral; 5 = Completely Agree).	**Mean**
Years of Experience in Training Community	20.00
1. ATEAMS HHD is difficult to use.	2.00
2. I have a good understanding of how ATEAMS HHD features are organized.	3.33
3. I would not want to use ATEAMS HHD every day.	2.33
4. The layout of the buttons made sense.	3.00
5. The physical layout of ATEAMS HHD make it difficult to use.	3.33
6. The steps to complete tasks followed a logical sequence.	2.67
7. I had to tap too hard on the buttons to operate them.	1.33
8. The icons and the graphics on the buttons/keys were clear.	3.00
9. I was confused by the organization of information on the display.	3.33
10. It was easy to learn how to use ATEAMS HHD.	3.33

11. I could not relate the words on the screen to the tasks.	3.00
12. It was easy to explore ATEAMS HHD features by trial and error.	3.67
Question (based on a five-point likert scale: 1 = Completely Disagree; 3 = Neutral; 5 = Completely Agree).	**Mean**
13. It took too many steps to complete tasks.	3.00
14. It was easy to read the words on the screen.	4.33
15. I could not tell when I had completed an action correctly.	3.67
16. It was easy to correct mistakes.	2.67
17. I was confused by the terms on the screen.	3.33
18. It was easy to find the features I wanted.	2.67
19. I often felt lost and did not know how to proceed.	3.00
20. The display size was not too small.	4.00
Motorola 5NINES Score: Questions 1-20 (0 = very low usability; 100 = very high usability)	55.67
21. The information provided on the prebrief screen would be useful during a training exercise.	3.67
22. It was easy to select the checkboxes, buttons, tabs, etc. of ATEAMS HHD.	4.00
23. The terms used on ATEAMS HHD were understandable, i.e., ATEAMS HHD 'spoke my language'.	
24. I found the interface of ATEAMS HHD to be highly intuitive.	3.00
25. It was easy to navigate through ATEAMS HHD.	3.00
26. It was not easy to recover from mistakes made with ATEAMS HHD.	3.67
27. The information provided on the debrief screen was useful.	3.67
28. The performance rating system of the data collection screen made sense.	2.33
29. Useful feedback was provided by the computer whenever I completed an ATEAMS HHD task.	2.67
30. I did not have a problem with the length of some of the text, i.e., text that required scrolling.	2.33
31. I needed no help when using ATEAMS HHD.	1.33
32. ATEAMS HHD does not need much improvement.	2.00

Table 2. APDA Usability Survey Results.

An examination of the data found in Table 2 seems to indicate that the users felt that the APDA software had poor usability. This is indicated by the 5NINES summary score of 55.67 (out of a possible 100) as well as by the answer to question 32, "ATEAMS HHD does not need much improvement", which garnished a mean rating of 2.00 indicating that the user disagreed with this statement. However, they also disagreed with the statement that the ATEAMS HHD was difficult to use. It may be that the users found the *iPAQ™ hardware* easy to use *physically* (e.g., button clicks, simplicity of text) but the *process* of using the *APDA software* difficult (e.g., moving from one screen to another, entering a comment). For example, questions 1, 6, and 16 are process questions and indicate that the users experienced some difficulty with the APDA process associated with each question. Alternatively, questions 4, 7, and 8 are questions relating to the physical features of iPAQ™ H3650 and seem to indicate that the users did not have much difficulty with the physical feature associated with each question. The difficulty with the APDA process was born out by user comments:

1. "When you are grading an exercise you are not going to have a lot of time really to switching back and forth."
2. "Everything needs to be as expeditious as possible."
3. "Given that I've developed these MOPs, I'd like to go directly to an MOP that - is it the case that I have to scroll through five or ten of these and the sub-elements beneath."
4. "I'm sort of lost on how to get to the debrief section."
5. "I guess would want to have some indication as to where I've made a comment - without having to look through every one of them."
6. "Do I hit 'Next' and the next guy comes up? Right? That doesn't really make sense."

In addition to the process questions, the users also did not feel that the APDA performance rating system made sense (question 28). This was confirmed by one user's comment, "These gross classifications, I'm not sure if they would be much value to me." Whether more complex performance assessment methodologies can be placed on a PDA with its limited screen real estate remains to be seen.

Summary

Overall, 37 violations were identified during the heuristic evaluation and user testing sessions. The breakdown of these violations can be found in Table 3.

Violated Heuristic	Number of Occurrences
Speak the user's language	1
Minimize users' memory load	6
Consistency	12
Prevent errors	8
Provide adequate help	1
Simplicity	8
Progressive disclosure	1

Table 3. Breakdown of Heuristic Violations.

As can be seen in Table 3, the consistency heuristic was most frequently violated. This is not surprising, given that APDA is prototype software. This is not to say that these violations are not important and should not be fixed. For example, one violation that fell under the consistency heuristic was that comments were not accessible from the Debriefing screen. This was a major source of frustration for the users and would severely reduce the utility of APDA. The number of violations that fell under the simplicity heuristic indicates that the APDA process should be simplified. The number of violations that occurred under the memory load and prevent errors heuristics also reinforces this notion.

Several correctable errors occurred during the evaluation, two of which should be corrected immediately. First, the user can accidentally exit the system without receiving an error-checking message. Second, the user can accidentally reset the scenario clock without an error-checking message appearing. The users also expressed concern about the errors committed when using two of the methods for inputting comments via the iPAQ™ (i.e., the scripting and electronic keyboard methods). The use of digital ink would correct this problem. Several software applications have been developed that can be used to create digital notes. For example, Gonna Software® has created a Pocket PC application called PocketStickey, which allows the user to write notes directly on the Pocket PC screen and then link the note to a designated file. Seiko Instruments USA Inc. has developed a unique input method that can be used for hand-helds, laptops, and PCs. It is called the InkLink™ handwriting system. To use this system, the user attaches a large clip, dubbed a data clip, to a pad of ordinary paper. The InkLink™ system tracks the movement of an electronic pen and sends this data to the computer device, which faithfully displays what the user has written and/or drawn.

Presentation and readability of displayed text were also areas of concern, both for the users and evaluators. Although the users indicated that they thought that the size of the text was adequate, they also noted that they had a problem with the length of the text (i.e., text that required scrolling). Because text length is affected by both the size of a font and its type, a font size and type/number of characters tradeoff occurs for PDAs. That is, the size and type of font used affects the number of characters that will fit in a given text box. Currently between 30 and 34 characters (including blank spaces) can fit into the Team Events Timeline and Assessment text windows before the text either disappears off screen (true for the Team Events Timeline window) or wraps (true for the Assessment window). The most common font sizes used in newspaper text ranges between nine and 12 points (Sanders & McCormick, 1987). This size can be used as a baseline when trying to apply standards to a computer display. For such displays, the font should subtend a visual angle of at least 12 minutes of arc (Sanders & McCormick, 1987). This would mean that at a viewing distance of 18 inches, the font size should be at least six-points. Such a small font size is in agreement with the user's assessment of the readability of the text on the iPAQ™, as indicated by the usability survey. However other factors, such as lighting conditions, vibration, criticality of information, visual acuity of the user, and amount of stress the user is under can also affect the readability of a display (Wickens, 1992). Under these circumstances, the font size should probably be increased beyond the minimum six-point font size. Some investigators indicate that legibility and readability would be increased with larger font sizes, sizes that subtend up to 25 minutes of visual angle (Wickens, 1992). At a viewing distance of 18", a 25-minute viewing angle translates into a 13-point font size. Based on the above data, it seems reasonable to suggest that the APDA font size match the font size range found on newspaper print (i.e., from a minimum of nine to a maximum of 12-points).

The key to the above is to ensure that all information displayed on APDA, especially the most important information, is readable and easily comprehensible. For the Timeline and Assessment windows, the most important information is found within each text box (e.g., the timeline of TEs and the MOPs, respectively). Therefore, the size of the text boxes should be maximized. The size of this display area also affects how much scrolling a user must perform. Unfortunately, in text-heavy applications, a PDA may lead to the type of frustration expressed by the participants in this evaluation. Therefore it is recommended that the Navy test the usability of each class of computer device (e.g., PDA and pen tablet computer) onboard ship during live training exercises with instructors, before committing to a given type or brand of device. Testing under real training conditions is critical in any evaluation. Such tests could reveal flaws in the design of the application and/or equipment that would not be found under more sterile testing conditions.

Other recommendations regarding usability issues have been outlined throughout the document. Specific recommendations include:

1. Error-checking when a user clicks the exit button or reset clock button.
2. Allowing the user easy access to comments.
3. Simplifying the APDA process. One way to do this would be to use clear textual descriptors to help indicate how a user moves from one screen to another.
4. Reducing the need for scrolling (e.g., by providing wrap-around text).
5. Reducing the size of the buttons. This should help reduce the potential for accidental activation and also provide additional space for the text box display area.
6. Using digital ink for comments.

Several of the users indicated that the information collected via the data collection screen may not be of much value. The simple data collected through APDA may be a limitation of the size of Pocket PC screens. It is recommended that a task analysis be conducted to define the type of data that shipboard instructors need. If an application with a more complex interface, or that is more text-intensive in nature, is required then the U.S. Navy might consider investing in pen tablet computers for its instructor data-collection requirements.

Implementing the APDA Redesign Recommendations

The developers of APDA received a comprehensive version of the data and results presented in this paper. The first step taken by the developers was to consolidate the identified design issues, isolating those situations in which solving one design issue would eliminate other design issues. The design violations were organized by APDA window or affected windows and then by the related heuristic. Doing so provided a framework that was used to reference specific design issues and to identify actions to be taken to correct the issue (see Appendix C).

Once the recommended modifications were categorized and actions needed to implement the modifications identified, a prioritization process was initiated. Three categories were used to help prioritize a given redesign recommendation.

1) How the modification would affect the user's understanding of how to use the APDA
2) The required level of effort to implement a modification.
3) How the modification would affect the functioning of the APDA

Two SMEs assessed the recommendations in the framework, determining which modification(s) would lead to the most significant improvement(s) for the APDA end-user. The SMEs rated each modification as important, nice to have, or not important. In parallel to the SME assessments, software designers evaluated the level of effort that a given action (i.e., implementing the modification) would require. Embedded within the level of effort assessment was an evaluation of how an action would affect the functioning of the APDA application. The factors that were considered for assigning level of effort included time, cost, and supportability. Based on the relative level of effort required, the actions were then classified as easy, medium, or hard. An easy action would require minimal time and/or research effort to accomplish. A hard action might require writing new controls for an APDA function, conducting an extensive search for existing software solutions, or identifying an issue that could only be resolved by significantly reorganizing how the ATEAMS database was arranged. A medium action would fall somewhere between an easy and a hard level of effort. The results of the SME and developer assessment efforts was a prioritization matrix, with each action falling somewhere on the important – not important and easy – difficult continuums (see Appendix D).

The final step of this process was to develop an implementation schedule. Each of the actions was reviewed based on the prioritization matrix and available resources to determine which action would be implemented first. From this review, each action was prioritized as a Priority 1, 2, 3, or 4 action. All Priority 1 actions were to be completed prior to beginning Priority 2 actions, all Priority 2 before Priority 3, and so on. This prioritization scheme enabled Sonalysts to efficiently allocate programmer time to the APDA redesign (see Appendix D).

It should be noted that several of the identified redesign recommendations were treated as special cases. For example, the recommendation to provide help functionality and documentation was an effort that was well beyond the funding and time requirements available for implementation and thus was treated as a separate project. Several other actions were also classified as "not to be implemented in the prototype". These actions were so classified due to certain functionalities that were not supported by either the APDA operating system or the programming environment or due to resource requirements that would not be available during the project performance period (e.g., evaluation by actual end-users. See Appendix D).

<u>Implementation Examples</u>

The prioritization scheme meant that modifications categorized as easy-to-implement and important would be given a priority 1 rating. This scheme is reasonable, but could mean that critical errors would not be corrected. For example, the usability evaluators determined that accidentally exiting APDA was a high priority design violation. Had the developers and SMEs rated this as a medium-level effort and a nice-to-have or as a high-level effort, but important, it would not have been given a priority 1 rating. Yet, this violation was critical, given that the user would have to restart the application, losing valuable assessment time and potentially missing key observations.

What follows are examples of APDA windows, both before and after recommendation(s) from the usability evaluation were implemented. This section applies an abbreviated format to the one used above. First the violation that was addressed is numbered and described. The violations are grouped under violations that were common to two or more windows, to the Collecting Data windows or to the Debriefing window. The redesign recommendation(s) to the violation(s) is (are) then noted. The original window, and the redesigned window, is then shown.

Violations Common to Two or More APDA Windows and Implemented Solutions

1. Clicking the Exit button immediately closes the APDA software. No error check message is provided. This increases the possibility that a user may accidentally exit the program.

Solution: An error-checking window now appears when the Exit button is clicked (see Figure 14).

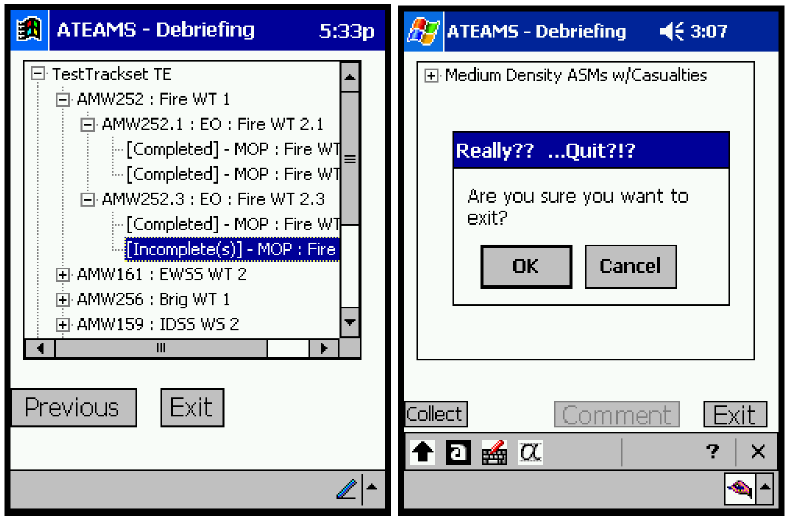

Figure 14. Original (left) and redesign. Note error-checking message.

2. To make it easier for the user to click, the designers of APDA may have purposely employed large command buttons. The size and location of many of the buttons could lead to errors produced by mis-clicks.

Solution: Command buttons were resized, moved, and/or separated in space (see Figure 15).

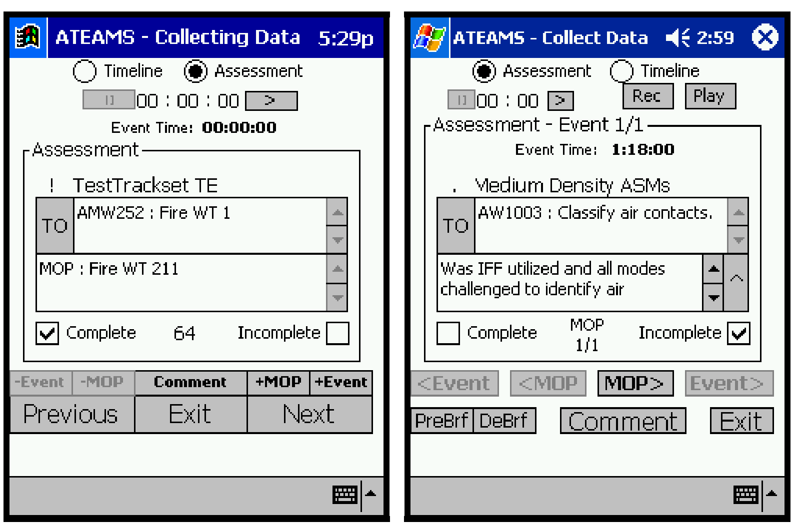

Figure 15. Command button location increases potential for mis-clicks (left). Note size/spacing changes on right.

Violations from the Collecting Data Windows and Implemented Solutions

1. When APDA is started, the initial window is Prebrief. However, there is no clear indication that the next window is Collecting Data, nor is it clear that the user must click the 'Next' button to open that window. This

is also the case when a user is finished collecting data and wants to open the Debriefing window. During user testing, it was discovered that several of the users were confused by the textual or symbolic descriptors associated with the command buttons (i.e., by the textual descriptors of Next and Previous and by the symbolic descriptors of + and – associated with the MOP and TE buttons). For example, while on the Assessment window one user clicked the button labeled Previous. This act closed the Assessment window and opened the Prebrief window. The user stated that he thought clicking the Previous button would take him to the previous MOP.

Solution: The navigation command buttons were renamed, reflecting the name of the window that they open when clicked. For example, on the Collecting Data window, the Previous and Next buttons were relabeled to PreBrf and DeBrf, respectively. Additionally, the + and – symbols found on the MOP and Event buttons were replaced by VCR-like control symbols. The idea is that these symbols will tap into a users experience with symbols found on VCRs, and other similar devices, making it easier for them to interpret the meaning associated with these symbols (move forward, move back). These changes can be seen in the right window in Figure 15. Also, when the user moves from the Prebrief window to the Collecting Data window, the Assessment window is now the first window to open. Previously, the Timeline window was the default window.

2. To enter a comment, the user selects one of the text entry methods. However, all methods cover the OK button of the Comment window. The user also cannot exit out of this window without saving a blank comment.

Solution: The command button was moved above the display area. Additionally, the command button was renamed Save and a Cancel button was added (see Figure 16).

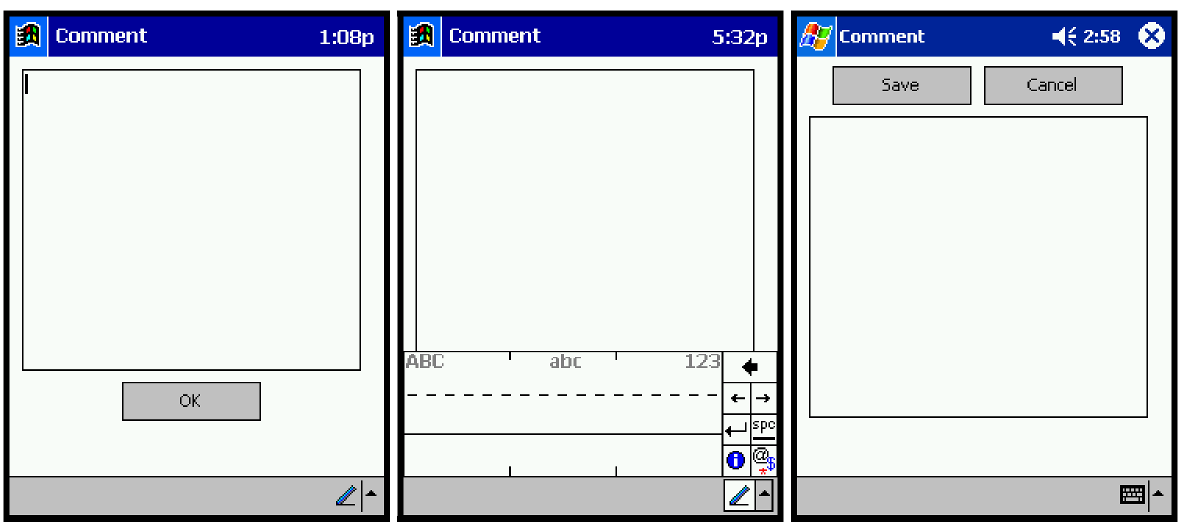

Figure 16. Command button (left), hidden (center). Solution: moved above display area (right).

3. The user may wish to edit and/or delete comments before data collection has stopped, but APDA does not provide this capability.

Solution: A comment command button was added to all windows. Additionally, the user can edit comments during data collection (see Figure 17).

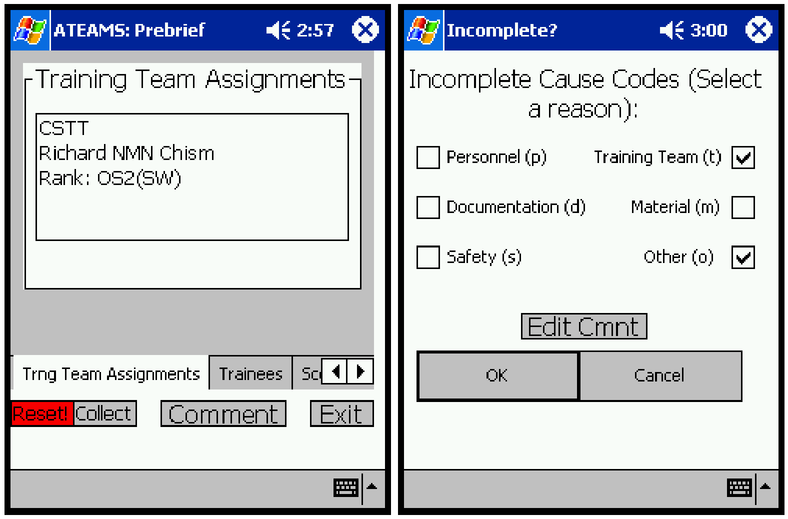

Figure 17. Comment link added to Prebrief (left). Comment can be edited in Incomplete? window.

<u>Violations from the Debriefing Window and Implemented Solutions</u>

1. The text that is displayed in the Debriefing window does not wrap. The result is that the user may have a difficult time in either comprehending/remembering long sentences or may take more time to read the text compared to the time it would take to read text that was viewable at once.

Solution: The user can double-tap on a given MOP in the Debriefing window, which will then open a separate window that displays the MOP in a large font (see Figure 18).

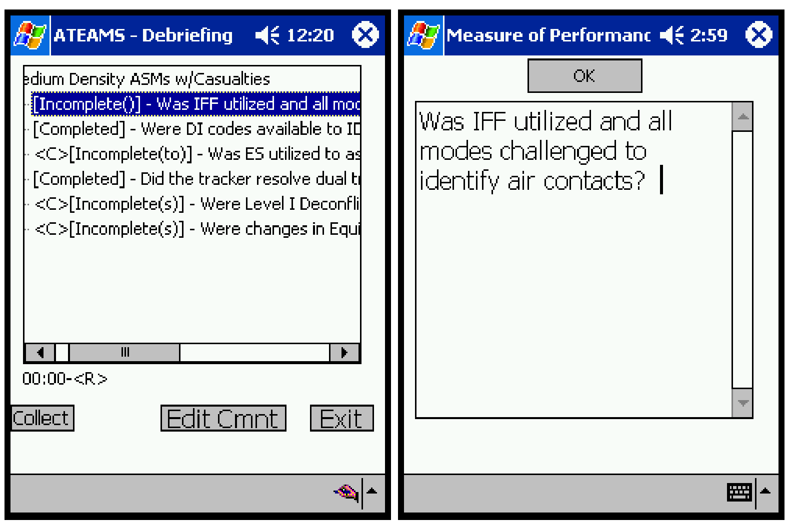

Figure 18. Double-tapping on an MOP listed in the Debriefing window now opens a display window.

3. The Debriefing window displays whether an MOP was complete or incomplete, with cause codes listed with the latter. However, there is no indication if a comment is attached to an MOP.

Solution: A comment code has been inserted into the MOP list. The code is displayed as <C> (see Figure 19).

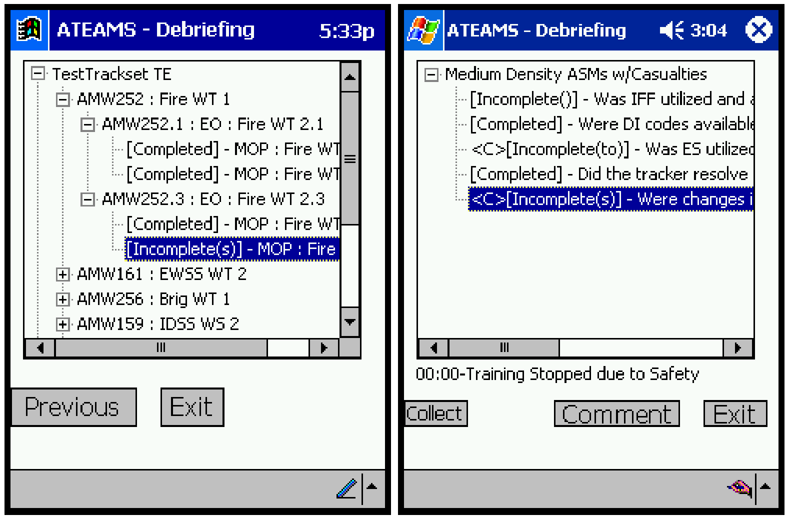

Figure 19. Initial version of Debriefing window (left). Redesign with comment code <C> (right).

Current and Future Efforts

The APDA usability evaluation was a preliminary evaluation of prototype software. The results have been used to drive the redesign of the application's windows. One key to increasing the usability of a product is iteration, meaning the system should go through continual refinements until pre-established acceptability criteria have been met. Therefore, the redesigned APDA will also be subjected to a usability evaluation. The usability process is also meant to be inclusive; meaning the usability professional, the system developers and the end-users should be involved from design conception to completion. So far, the APDA process has been iterative. However, the process also needs to be inclusive. Therefore, an attempt will be made to contact end-users and involve them in the design process.

There are several new efforts underway towards implementing data collection capabilities through PDAs. Two examples follow. Each of these efforts has applied knowledge gained in the APDA usability evaluation process to the design of the newer applications.

Coalition Readiness and Exercise Management System (CReaMS)

The Coalition Readiness and Exercise Management System (CreaMS) is a project designed to develop efficient and effective training situations for individuals and teams that are distributed around the globe. The targeted training audience is a coalition Naval Task Group and its members. Participants include Australia, the Netherlands, and the United States. Observations of past exercises identified areas of coalition operations that could be significantly improved with the employment of proper training. The CreaMS effort identified the need for an effective debrief between coalition partners at the conclusion of an exercise (i.e., timely and meaningful interactions between coalition partners that would review the strengths and weaknesses of each partner's efforts in the exercise).

One of the requirements for an effective exercise debrief in this context is that debriefing data be available to all coalition partners. ATEAMS and the APDA were used to provide exercise coordinators a means by which they could rapidly develop a training scenario and create a data collection plan tailored to the particular watchstations that an instructor would be observing. They also were used to quickly distribute the data collection plan and to compile the collected data fast enough for a timely debrief. The developers of the CreaMS APDA applied lessons learned from the APDA usability evaluation to the CreaMS application, likely reducing development time that might have otherwise been needed for redesign efforts.

Sonalysts has also created an ATEAMS-independent hand-held data collection tool that is currently being used with revised CReaMS databases. Antidotal data suggests that incorporating the results of the APDA usability process early in the design phase produced a product that was easy for the end-user to learn.

Battle Stations 21

The Battle Stations 21 project is designed to produce a capstone, 24-hour Navy Basic Training experience. All recruits must complete Battle Stations 21 prior to graduating from Boot Camp. Battle Stations 21 will present realistic scenarios that challenge the recruit's determination, endurance, and core Navy values of honor, courage, and commitment. During Battle Stations 21 recruits will be provided an opportunity to demonstrate a variety of basic Navy skills acquired during the previous seven weeks in recruit training. These skills cut across a variety of domains from protocol and etiquette to Seamanship and Damage Control. An overarching goal of the events and scenarios of Battle Station 21 is to foster a team environment in which recruits must work together to complete exercises or overcome obstacles.

Battle Stations 21 will be an intensive experience that introduces a number of challenges for the instructor. Instructors must provide a realistic training experience in a training environment that is highly constrained, in terms of both time and space, and that also has a large student throughput requirement. To achieve the goal of Battle Stations 21, a state-of–the-art Training Management System will be developed. A hand-held data collection tool similar to the APDA will be part of the Training Management System. The following sections describe how the hand-held device will be implemented within the Training Management System.

Data Collection Module. The Training Management System will be used to create the exercises' data collection packages, which will then be downloaded to a hand-held device. The hand-held device's software will enable instructors to collect performance data and to make observations that are related to safety, the scenario story, and scenario integrity.

Debrief Support Module. Once performance data is collected, the hand-held device will be used to present debriefing material (e.g., the hand-held device will be connected to a projector that will project the debriefing material onto a screen). This requirement means that the software must allow instructors to quickly summarize and prepare data for debrief. In a similar vein, debriefing support software will also be implemented on the hand-held device. Instructors can use the debriefing support software to guide them through a pedagogically sound debrief.

Instructor Communication Module. With the advent of reliable and affordable wireless networks, the hand-held device's software will also support exchange of information between instructors and between instructors and exercise operators/controllers. This software must allow personnel to communicate with minimum effort. Ease of use is a key concern of instructors, whose primary focus must be on what is happening with the recruits during a training exercise.

Conclusion

To test the feasibility of using a PDA for data collection during shipboard training exercises, researchers at Sonalysts, Inc. developed a prototype data collection application ATEAMS PDA (APDA). The initial application could be used to synch the Afloat Training Exercise and Management System (ATEAMS), hosted on a PC, with a PDA. Once synched, training relevant data could be downloaded from ATEAMS to the PDA. Researchers at NAVAIR Orlando Training Systems Division subjected the APDA to a usability evaluation. Usability evaluations can reveal critical and non-critical design flaws in hardware and software and, therefore, was chosen as one method for evaluating the APDA application. This evaluation consisted of heuristic evaluations, user testing sessions, and redesign recommendations. Three users that had taken part in the ATEAMS software evaluation were asked to perform specific tasks using the APDA software. The user testing sessions were designed to validate the heuristic violations or find problem areas not identified by the usability evaluators.

Thirty-seven violations were identified during the heuristic evaluation and user testing sessions. Redesign recommendations were provided to the developers of APDA. The developer used the APDA report to prioritize the heuristic violations. Two SMEs determined which modification(s) would lead to the most significant improvement for the APDA end-user. The SMEs rated each modification as important, nice to have, or not important. In parallel to the SME assessments, software designers evaluated the level of effort that a given action (i.e., implementing the modification) would require. The factors that were considered for assigning level of effort included time, cost, and supportability. Based on the relative level of effort required, the actions were then classified as easy, medium, or hard. The result of the evaluation process was a prioritization matrix. The data within this matrix drove the redesign effort.

The APDA evaluation demonstrated that the usability process could be applied to small form factors like Pocket PCs. The evaluation also indicated the need to develop tools that could be used to collect objective data such as time and error rates, when evaluating such small form factors. Videotaping equipment could be employed only if the PDA were mounted, which would introduce artificiality in the evaluation because users generally hold PDAs. However, Noldus Information Technology® has developed a wireless camera that can be mounted on a PDA. This device is relatively unobtrusive and would be well suited for videotaping such devices (Noldus Information Technology, 2003). There is a need for a PDA software application that can collect time and button-click data, which could then be analyzed for error events. Such applications currently exist for standard PCs. The Group for Interface Research at the University of California at Berkley has developed an application that records button clicks for web-enabled hand-held devices called WebQuilt. This application records the web links that a user clicks on, the web pages visited, as well as time spent on a page. However, it cannot record other onscreen clicks (e.g., when a user clicks on a scroll bar) or user comments (Waterson, Matthews, & Landay, 2002).

The APDA evaluation revealed limitations in PDAs when they are applied to U. S. Naval data collection efforts. The primary limitation is the size of the display screen. Because of this limitation, a PDA may not provide the best solution for text-intensive data collection applications. A secondary limitation relates to the input methods described in the paper. The Pocket PC does not provide an acceptable solution when quick and accurate note taking is required. Microsoft® has introduced handwriting recognition software, both for Pocket PCs and pen tablet computers. Preliminary evaluations of both applications are not encouraging. The former does not seem to provide the accuracy needed for training environments. The latter's method of converting handwritten notes to text seems to be extremely labor intensive and non-intuitive. Although the recommended method may tax the storage capacity of a Pocket PC, digital ink may be the best solution for electronic note taking for a PDA in a shipboard training environment.

Appendix A: Heuristic Definitions

1. Speak the user's language (rather than computerese). Text displayed to the user should be expressed in words, phrases, and concepts familiar to the user. Input and output must conform to population stereotypes.
2. Minimizing users' memory load. The user should not have to remember information from one part of an application to another. Instructions for use of the system should be visible or easily retrievable. Methods of chunking and focusing information aid in reducing mental workload.
3. Consistency. Users should not have to wonder whether different words, situations, or actions mean the same thing. If the user develops a correct mental model of the system, there will be a dramatic reduction in cognitive processing expended on understanding how the system works. One of the major benefits of consistency is that users can transfer their knowledge and learning to a new program if it is consistent with other programs they already use.
4. Preventing errors. Even better than good error messages is a careful design that prevents a problem from occurring in the first place.
5. Providing adequate help and documentation. Even though it is better if the system can be used without documentation, it may be necessary (and is advisable to accommodate all users types) to provide help and documentation. Any such information should be easy to search, be focused on the user's task, list concrete steps to be carried out, and not be too large.
6. Simplicity User's are not impressed with complexity that seems gratuitous, especially those who may be depending on the application for timely and accurate work-related information.
7. Progressive Disclosure. Users should not be overwhelmed by what they can do in a product. You don't need to show users all of the functions the product offers. The best way to teach and guide users is to show them what they need, when they need it, and where they want it. This is the concept of progressive disclosure. New technology such as wizards and assistants use progressive disclosure to guide users through common tasks. Wizards guide users through steps in a progressive manner where each step is simple and meaningful for even casual users.

Appendix B: APDA Scenario Instructions

Last time we worked together, we asked you to develop a training scenario using the ATEAMS software. The training teams in this scenario were the Combat System Training Team (CSTT) and the Damage Control Training Team (DCTT). The Mission Area was Under Sea Warfare (USW). The idea was that you were onboard ship, developing a training scenario to be used for shipboard training. After completing the scenario, the next step would be to download the trainee performance evaluation sheets provided by ATEAMS to a Personal Digital Assistant (PDA). We have done this for you. This is the purpose of the current evaluation, i.e., you are now going to use and evaluate the data collection portion of ATEAMS that has been loaded onto the PDA.

As you use the PDA and the ATEAMS software, we again ask that you "think aloud" so that we can accurately interpret misunderstandings, opinions, and expectations you may have related to ATEAMS. We may prompt you at times for such information. During this exercise we will be unable to assist you unless absolutely necessary. You can refer to the information below while using the PDA. If you have any questions, please ask them now. Do not start ATEAMS until the experimenter gives you the go-ahead.

Tasks:

1) Turn on the PDA and start ATEAMS HHD (ATEAMS Hand-Held Device).
2) Using the provided hand-held electronic device containing ATEAMS HHD:
    a. Confirm the Training Team Assignment as CSTT
    b. Confirm the Trainee(s) as John Hodak.
3) Look at the Timelines and Scenario Summary and share your thoughts on what information is provided and how it is provided.
    a. The Lessons Learned, Safety, and ROE screens are empty, but feel free to explore and comment on those screens.
4) Review the Objectives and share your thoughts on what information is provided and how it is provided.
5) Document the given Measures of Performance (MOPs) for the named trainee. You will be acting as an instructor assessing the trainee, John Hodak, on the listed MOPs. He is a trainee for the watchstation Time/Brg/Time Freq Plot. When you begin the assessment,
    a. Start the timing clock.
    b. Rate the listed MOPs as complete or incomplete (see attached form).
        i. If the MOP is incomplete, document the Cause Code. All the information you need to enter is listed in the table below.
6) While working with ATEAMS HHD, share your thoughts on what information is provided and how it is provided.

Measures of Performance (MOPs) USW		Complete	Incomplete: Cause Code(s)
201	*Analyze and Plan for a USW Mission or Task*		
	Incidents of conflict not resolved.	C	
	Time to prepare/promulgate plan.		T (training team)
205	*Search, Detect, Localize, and Track*		
	Range from own unit/forces contacts(s) detected versus predicted.		P (personnel, O (other)
	Time to fix contact's position.	C	
101	*Detect, Classify, and Track Subsurface Contacts.*		
	Was SITSUM display monitored?	C	
	Were assigned verniers replicated?		S (safety)
103	*Control Aircraft in a USWW Role.*		
	Was classification of the target updated?		M (material)
206	*Classify and ID Subsurface Contacts.*		
	Range from own unit/force contact classified/identified.		D (documentation) S (safety)
	Time to classify/identify contacts.	C	
207	*Engage to Achieve Mission*		
	Time to develop firing solution(s) or engagement plan.	C	
	Time to conduct attack.		T (training team M (material) O (other)
102	*Engage Subsurface Threat with Antisubmarine*		
	Were re-attack procedures conducted?		D (documentation)

6) When finished, submit the data for debriefing and look over the debriefing material provided for the assessed objectives. Again, please share your thoughts on what information is provided and how it is provided.

Appendix C: Prioritized Actions to be Taken on Report Recommendations

Recommendations from the Initial Usability Evaluation of the APDA Software and Hardware.

General

1. Preventing accidental exiting from the system or resetting of the system clock.
2. Allowing the user easy access to comments.
3. Simplifying the APDA process, e.g., using clear textual descriptors indicating how a user moves from one screen to another.
4. Reducing the need for scrolling, e.g., by providing wrap-around text.
5. Reducing the size of the buttons. This should help reduce the potential for accidental activation and also provide additional space for text display.
6. Using digital ink for comments.
7. Conducting a task analysis. Several of the user's indicated that the information collected via the data collection screen may not be of much value. The simple data collected via the APDA may be a limitation of the size of PDA screens. Conduct a task analysis to try and discover the type of data that the shipboard evaluator needs. If more data, or data of a screen-intense nature, is required then consider investing in a larger handheld or wearable computer device.

Usability Issues Relating to All Screens

A. Heuristic Violation: Minimizing the User' Memory Load

Issue

Text readability/Font size concerns.

Recommendation

1) Use a font size no smaller than 9-point
2) Reduce amount of text displayed in an individual box
3) Increase text box size

Action

1) Add a form that maximizes MOP. Form will be displayed at the touch of a button that is to be part of the "smaller" MOP frame.
2) Increase font size in maximized window for ease of reading.
3) Add an OK button to maximized window to return to Assessment screen.
4) Ensure functionality of the maximized screen and the access button are detailed in training/help materials.

B. Heuristic Violation: Preventing Errors

Issue

When a button/tab is clicked, it turns white (Figure 5 in report)

Recommendation

1) Adding a thick black border around the selected button may make it easier for the user to remember which button is selected. Color could also be used as a highlighting method.

Action

1) Ensure there is a mention that when selected, tabs and buttons will turn white in the training/help material. This is a basic Windows CE function. There is no provision for changing the tab color to something else other than white when selected.

C. Heuristic Violation: Consistency

Issue

Neither the Prebrief nor the Debrief screens have a link to a comment page. However, the user may wish to add a comment while on either of these screens.

Recommendation

1) Adding a comment link to all screens may increase the usability of the software.

Action

1) Add a link. Note: Comments added from the Pre-brief and De-brief pages will not be directly related to a particular MOP as comments added from the Assessment page are. ATEAMS currently has no means for handling comments that are input in either case.

Issue

The titles of the three screens are inconsistently phrased

Recommendation

1) Change the phrasing of the titles of the last two screens to Data Collection and Debrief

Action

1) Rename offending screens.

D. Heuristic Violation: Preventing Errors

Issue

Clicking the Exit button immediately closes the APDA software. No error check message is provided

Recommendation

1) Use the standard Windows error-checking pop-up window

Action

1) Add error-checking functionality.

Issue

The size and location of many of the buttons could lead to errors, i.e., mis-clicks (Figure 6 in report)

Recommendation

1) Reduce the size of the Exit/Next/Previous buttons and insert some space between all buttons.
2) Place the Next/Previous buttons next to each other.
3) Move the Exit/Next and Previous buttons to the extreme lower left/right of the screen.
4) Increase the size of the +/- Event and MOP buttons.
5) Increase space that holds the MOP listing.
6) Make MOP listing area the visual focal point of the user (Consider raising event box higher and enlarging the MOP box).
7) Consider raising the Event box higher and enlarging the MOP box. If possible make the font in the MOP box the largest on the screen so that the user's attention is drawn there.

Action

1) Reduce size of Exit, Next, and Previous buttons. Reposition buttons so that the Exit button is always in the lower right corner of all screens and the Next and Previous buttons are in the lower left of all screens.
2) Increase the size of the +/- Event and MOP buttons
3) Add a form that maximizes MOP. Form will be displayed at the touch of a button that is to be part of the "smaller" MOP frame.

E. Heuristic Violation: Providing Adequate Help and Documentation and Consistency

Issue

APDA provides no access to a help function

Recommendation

1) Add a help function

Action

1) Add a help function

F. Heuristic Violation: Simplicity

Issue

Users want screens to be as simple and efficient as possible

Recommendations

1) Adding a drop-down menu may make it easier for the user to find/select the TE/MOP from the Assessment screen or from the Debrief screen.
2) Comment screen should pop up when as the user clicks the button Other (on the Incomplete screen), since the trainer should indicate what happened to cause a given MOP to be incomplete.
3) Comment screens should be accessible from both the data collection and debriefing screens directly.
4) Users wanted some coding method to be used on the Debriefing screen, indicating where they had made comments related to a given MOP and a means to access those comments directly

Action

1) Add a comment button to the cause code screen to provide the user the option to comment a cause code. Note: there is no ATG requirement to add comments to the cause codes and forcing a trainer to the comment screen each time the cause code "other" is selected will result in unnecessary screen taps. Providing the option to add a comment however is a good idea.
2) Add a comment button to the Pre-brief and Debrief screens
3) Add an indicator that a comment has been entered for a particular MOP (the character C displayed under the MOP in the Assessment and Debrief screens may be a good way to indicate this. Another possibility might be to change the text that appears in the Assessment screen comment button from "Comment" to "Edit Comment" when an MOP that has a comment entered is displayed).
4) Add a Review Comments button to the Debrief screen. The Comment Review screen would display the MOPs commented and the comment text that has been entered for each. The screen would also provide an edit comments button that will allow for editing/addition of comments from the debrief screen.

Issue

On both the Prebrief and Debriefing screens, the user must click on the pluses to open up all the objectives

Recommendation

1) Include 'Expand all' and 'Collapse all' options for objectives

Action

1) eVB does not include "Expand All" and "Collapse All" functionality.

Usability Issues Relating to Prebrief Screen

G. Heuristic Violation: Prevent Errors

Issue

Potential for activating one function over a desired one to crowded positioning of buttons/tabs.

Recommendations

1) Reduce the size of the buttons and insert some space between all buttons

2) Move the Exit/Next and Previous buttons to the extreme lower left/right of the screen

Action

1) Reduce size of Exit, Next, and Previous buttons. Reposition buttons so that the Exit button is always in the lower right corner of all screens and the Next and Previous buttons are in the lower left of all screens.
2) Unable to reduce size of buttons while maintaining readability of button label.

Issue

Abbreviations are inconsistently created

Recommendation

1) Abbreviation method chosen should be consistently applied, with truncation being the preferred method

Action

1) Need fleet feedback to provide labels that will be most meaningful and understandable to end users. Abbreviations and truncations currently used on existing screens are meaningful to training team members however improvements may be gained through fleet feedback.

H. Heuristic Violation: Consistency

Issue

Position of the three aforementioned rows of tabs change, depending on which is clicked

Recommendation

1) Keep the position of each button the same no matter which button is clicked

Action

1) This is standard Windows functionality. No change to the screen will be made. Add a reference regarding the changing tab positions to the training/help material.

I. Heuristic Violation: Speak the User's Language

Issue

Scenario Time displayed for a selected TE is difficult to understand

Recommendation

1) Use the words "Scenario Time" as a descriptor of the displayed time

Action

1) The Scenario Time is currently displayed in a form that is inherently understood by training team members - Start+HH:MM:SS. Add a reference regarding the format of the Scenario Time to training/help material.

Usability Issues Relating to the Collecting Data Screen

J. Heuristic Violation: Consistency

Issue

Relative to the Prebrief and Debrief screens, the size of the Exit button on the Collecting Data screen is inconsistent (larger and conjoined with the Next/Previous buttons)

Recommendation

1) Standardize the size and placement of the common command buttons

Action

1) Standardize the size and placement of the common command buttons (Exit, Next, Previous) between all screens

Issue

There is an inconsistency in the way text is displayed between the Team Events Timeline and Assessment screens. Text in the former wraps. Text in the latter does not.

Recommendation

1) Force the text to wrap to eliminate need for horizontal scroll bar

Action

1) Create a word wrapping function for the Events Timelines.
2) Integrated Events timeline window displays events backwards. This needs to be fixed!

Issue

All caps are used for the text of the TEs when the Assessment button is clicked but when the Timeline button is clicked mixed case is used for the text of the TEs

Recommendation

1) Change the former to mixed case

Action

1) This is not currently a problem. The situation may have been a function of the way the original scenario was developed.

K. Heuristic Violation: Simplicity

Issue

Several users were confused by the textual descriptors of the Next, Previous, + MOP – MOP, + TE, and – TE buttons

Recommendation

1) Next and Previous are frequently used as textual descriptors of navigational aids in web pages as well as in computer based training software. Therefore they should not be changed. Reducing the size of these buttons, while simultaneously increasing the size of the ± MOP and TE buttons may help. Color coding the latter buttons may also help. To capitalize on a users mental model of tape controls, e.g., of a video player, an arrow scheme may also be employed, e.g., $\Leftarrow$ MOP $\Rightarrow$.

Action

1) Reduce size of Exit, Next, Previous buttons and reposition as previously stated.
2) Enlarge size of +/- Event and MOP buttons.
3) Modify label on the +/- Event and MOP buttons to resemble that of a VCR (i.e. <Event, Event>, <MOP, MOP>) to indicate direction of navigation.

Issue

When the user enters the Data Collection screen, the initial screen seen is the Timeline. This seemed to confuse the users.

Recommendation

1) Consider making Assessment the default screen of the Data Collection screen

Action

1) Need fleet feedback to determine most desirable order for presenting screens.

L. Heuristic Violation: Simplicity and Progressive Disclosure

Issue

It should be made obvious to the user how to quickly access the data collection component of the system

Recommendation

1) Label the 'Previous' and 'Next' buttons with the actual screen name that will appear when the button is clicked

Action

1) Changing the button labels contradicts previous recommendation to keep these the same. No change will be made, however comments regarding the order of screen appearance and meaning of navigation buttons will be included in the training/help material when developed.

M. Heuristic Violation: Consistency and Minimize User's Memory Load

Issue

Two of the main issues associated with a PDA are readability and comprehension. The designers of the APDA reduced the font size used in the Assessment screen, relative to the Team Events Timeline screen, presumably to reduce the need to scroll through the text (i.e., so the user can read an entire TE and/or MOP without having to scroll). However, this is inconsistent use of font size and may reduce the readability of the text.

Recommendations

1) The best solution may be to wrap the text and, if necessary, reduce the font size since the user can always move the PDA closer to his eyes

2) Consider raising the Event box higher and enlarging the MOP box

2) If possible make the font in the MOP box the largest on the screen so that the user's attention is drawn there

Action

1) Maximized screen discussed earlier will fix these issues.

Issue

Text box layout used reduces the amount of space available for text, both on the Timeline and the Assessment screen

Recommendations

1) Box can either be eliminated or enlarged such that its borders touch the edge of the PDA screen

2) Control buttons (Team Events, Integrated Events, Previous, etc.) can be reduced in size and moved away from the text box, increasing the amount of available space for the textbox

Action

1) Box border cannot be eliminated. It is part of the basic layout of the screen.

2) Control button issues will be addressed as previously discussed.

Issue

When the Assessment button is clicked, the TE title is duplicated

Recommendation

1) Redundant text can be eliminated, again increasing the amount of available space for the text box

Action

1) The TE name is presented on both Timeline and Assessment pages to reduce the need to cycle between the Timeline screen and Assessment screen. Presenting the TE, TO, EO, and MOP on one page maintains the referential integrity that provides the context needed to evaluate the trainees performance on a particular MOP. This feature will not be changed.

N. Heuristic Violation: Prevent Errors

Issue

The proximity of the Previous, Exit, and Next buttons, relative to the Comment and the ± Event and MOPs buttons, may increase the possibility of mis-clicks

Recommendation

1) Reduce the size of the Previous/Exit/Next buttons and insert some space between all buttons

2) Place the Next/Previous buttons next to each other

3) Move the Exit/Next and Previous buttons to the extreme lower left/right of the screen to help prevent accidental activation

4) Include an error-check for the exit button

Action

1) Add an error check for the Exit button.
2) Previously discussed actions will correct the other issues.

O. Heuristic Violation: Consistency

Issue

Same information is displayed when either the Team Events or Integrated Events button is selected in the Timeline screen. The order of TEs changes depending on which button is clicked.

Recommendation

1) Unless there is a clear purpose for the Integrated Events button, eliminate it. Make sure the order of the events (as displayed to the user) is proper.

Action

1) The evaluated scenario contained only CSTT events. In a case like that, the team timeline and integrated timeline will be identical. The two displays are important for maintaining the overall context of the scenario when more than one training team is participating. No action will be taken.

P. Heuristic Violation: Minimize User's Memory Load

Issue

There is no indication of the total number of TE's or MOPs. There is no indication of the number of TE's/MOP's completed (e.g., percent completed).

Recommendation

1) Attempt to include the aforementioned counters. User should be provided with an indication of the number of complete (or incomplete) TE's and MOPs.

Action

1) Add a counter to indicate TE/MOP completion.

Issue

The position of the two pairs of TE and MOP buttons is mirrored as follows (see Figure 8 above):

- Event – MOP Comment + MOP + Event

This mirroring may confuse the user.

Recommendation

1) A better layout may be as follows:

+ MOP - MOP Comment + Event - Event

An alternative method would be to use toggle buttons, e.g.,

MOP [+/–]   [+/–] Event   Comment

or a symbol system, e.g.,

⇦ MOP ⇨   ⇦ Event ⇨   Comment

Action

1) Button positioning and Text labels will be modified as previously described.

Issue

Text labeled Event Time is somewhat distant from the selected TE text. Meaning of the text label Event Time may not be clear.

Recommendation

1) Moving the label/event time closer to the TE text may make it easier to find/use.
2) Replace text label of Event Time with the text label of Time of Event - may make the meaning of the TE time a little clearer.

Action

1) Attempt to place the Event Time indicator inside the box that lists TE, TO, EO and MOP.
2) Determine whether Time of Event label will fit in space available.

Q. Heuristic Violation: Consistency and Prevent Errors

Issue

Clock control buttons consist of a reset button and a button that toggles between start and pause. No fast forward, rewind, or stop buttons are provided.

Recommendation

1) May be useful to include rewind, fast forward, and stop buttons.

Action

1) VCR type controls not necessary. An explanation of clock operation will be included in the training/help materials.

Issue

No error check is provided when the reset button is clicked. Thus the user can accidentally click this button and reset the scenario clock to zero.

Recommendation

1) Provide an error check message when the reset button is clicked

Action

1) Provide error check for clock.

R. Heuristic Violation: Simplicity and Prevent Errors

Issue

Method of text input not simple nor intuitive.

Recommendation

1) Use digital ink note text input method.

Action

1) Provide training/help information on text input methods. An ink note capability exists as included functionality w/in the iPaq. Another useful function that has been added is the transcriber that converts handwriting to text.

Issue

Comment screen consists of a blank text box and an OK button. When entering in a comment, the keyboard or the writing surface area covers the OK button of the Comment screen. To reveal the OK button, the user must click either a keyboard or pen icon (located to the immediate left of the aforementioned arrow). If the user accidentally clicks another button, e.g., the demo button found on the handwriting screen, it is quite easy to get lost.

Recommendation

1) Make sure the OK button is always displayed.
2) Rename the OK button to Save.

Action

1) Reposition the OK button.
2) Change the label to Save vice OK

S. Heuristic Violation: Consistency

Issue

No Cancel button available on Comment screen. If no comment is entered, the only option the user has for returning to the Collecting Data screen is to click the OK button. Once the OK button is clicked, it is not known if blank comments are being inserted into the debrief database**.**

Recommendation

1) Add a Cancel button.

Action

1) Add a Cancel button

T. Heuristic Violation: Minimize User's Memory Load

Issue

When a comment has been entered and saved by the user, the user may wish to edit and/or delete comments before data collection has stopped. Once comments are entered there is no way to edit them.

Recommendation

1) Provide a method through which the user can recall and edit notes.

2) The presence and the method for accessing related MOP comments in debrief should be obvious. In other words, provide a symbol near a MOP where a comment has been made. Allow access to that comment by clicking on that symbol.

Action

1) When a comment has been entered, an indicator will appear a discussed above. When the Comment button is selected the software should retrieve the previously entered comment for editing.

<u>Usability Issues Relating to the Debriefing Screen</u>

U. Heuristic Violation: Minimize User's Memory Load

Issue

About 40 characters can fit on the screen. Unfortunately, the menu structure, combined with the length of the MOPs, means that most of the MOP (and sometimes part of the TE) descriptors are not viewable without using the horizontal scroll bar (some MOPs are 19 words long).

Recommendation

1) Based on the TE's and MOPs viewed during this evaluation, it would seem that scrolling horizontally would tax the users memory load to a greater degree than if the text wrapped. However if the length of the text is long, e.g., 19 words, then wrapping the text may also unduly burden the user's memory load. Ideally no scrolling should be used, e.g., a 19 word MOP should fit on the display. For this to occur, however, the font size would have to be reduced, perhaps to an unreadable level. In addition the readability of a given display depends on, among other factors, the amount of stress the user is under, the viewing conditions (light levels, vibration), and quality of the display. Related data would seem to indicate that the use of wrap-around text, drop-down windows, or some combination of the two might be in order (The Windows® Interface Guide, 1995). It is important that some (or all) of the above factors be examined in an operational setting so that the usability of a PDA/pocket PC can be thoroughly evaluated.

Action

1) The Maximized Screen will correct these issues for the MOPs being evaluated. The TE, TO, and EO are merely provided as a fallback in case the trainer cannot remember the

context of the MOP. Since those areas will not be critical to the evaluation of the MOP, they will not be included in the Maximized Screen.

V. Heuristic Violation: Consistency and Simplicity

Issue

There is no template to guide the user for deciphering the MOP code

Recommendation

1) Provide a template to the user (perhaps through a help button).

Action

1) In the Debrief Screen, the software currently provides the first letter of each of the cause codes selected as a reason for the MOP being evaluated as incomplete. This is a good way to concisely indicate which cause codes apply to the MOP and should be readily apparent to any training team member. Add comments explaining the structure of the debrief and the meaning of the incomplete cause codes to the training/help material to avoid any confusion for the user.

Issue

There is no indication if a comment had been entered for a given MOP

Recommendation

1) Add a comment code to the MOP so the user will know which MOP has a comment associated with it.

Action

1) This issue will be resolved as discussed above.

Issue

There is no method for the user to display the comments associated with a given MOP

Recommendation

1) Provide a method through which the user can display his comments.

Action

1) This issue will be resolved as discussed above.

W. Actions Not A Part of the Usability Report

1) Filtering/Displaying Incomplete MOPs

2) Modify Pre-brief to provide the IITSEC specific pre-brief materials.

3) Develop functionality that allows 1 trainer to collect data on more than 1 trainee.

Appendix D: Actions to be taken for HHD – 29 Oct 2001

All actions listed as Priority 1 will be worked together as the top priority. Priority 2 will be worked when we finish the Important to Do Priority 1 actions. We need to move quickly on the Priority 1 items so that we can use the HHD software for the CReaMs test events that are planned for next week.

Important Things to Do

Priority 1

A1 Add a form to the Assessment Screen that maximizes MOP. Form will be displayed at the touch of a button that is to be part of the "smaller" MOP frame.
Addresses: D4, M1, M2, U1

Important - Easy

A2 Increase font size in maximized window for ease of reading.
Addresses: M1, M2

Important - Easy

A3 Add an OK button to maximized window to return to Assessment screen.

Important - Easy

D1 Add error-checking functionality when Exit button is pushed and when resetting clock.
Addresses: N1, Q2

Important - Easy

J3 Integrated Events timeline window displays events backwards. This needs to be fixed!

Important – Easy/Medium

Priority 2

C1 1) Add a comment link to prebrief and debrief screens. 2) Store Comments. Note: Comments added from the Pre-brief and De-brief pages will not be directly related to a particular MOP as comments added from the Assessment page are. 3) ATEAMS currently has no means for handling comments that are input in either case.
Addresses: F2

1) Important – Easy
2) Important – Medium
3) Important – Impossible for now

F1 Add a comment button to the cause code screen to provide the user the option to comment a cause code. Note: there is no ATG requirement to add comments to the cause codes and forcing a trainer to the comment screen each time the cause code "other" is selected will result in unnecessary screen taps. Providing the option to add a comment however is a good idea.

Important - Easy

F4 1) Add a Review Comments button to the Debrief screen. The Comment Review screen would display the MOPs commented and the comment text that has been entered for each. 2) The screen would also provide an edit comments button that will allow for editing/addition of comments from the debrief screen.

Addresses: V3

1) Important - Medium
2) Nice to have - Hard

Priority 3

R2 Reposition the OK button on the Comment Screen to remain clear of pop-up keyboard at all times.

Important - Easy

W1 Filtering/Displaying Incomplete MOPs

Important - Medium

<u>Nice to Have Things to Do</u>

Priority 1

D2 Reduce size of Exit, Next, and Previous buttons. Reposition buttons so that the Exit button is always in the lower right corner of all screens and the Next and Previous buttons are in the lower left of all screens.

Addresses: G1, G2, J1, K1, M3, N2, P2

Nice to have - Easy

D3 Increase the size of the +/- Event and MOP buttons.

Addresses: K2, P2

Nice to have - Medium

K3 Modify label on the +/- Event and MOP buttons to resemble that of a VCR (i.e. <Event, Event>, <MOP, MOP>) to indicate direction of navigation.

Addresses: P2

Nice to have - Easy

Priority 2

F3 Add an indicator that a comment has been entered for a particular MOP (change the text that appears in the Assessment screen comment button from “Comment” to “Edit Comment” when displaying an MOP that has had a comment previously entered. If the “Edit Comments button is selected the button will retrieve the previously entered comment).

Addresses: T1, V2, V3

Nice to have - Medium

F4 1) Add a Review Comments button to the Debrief screen. The Comment Review screen would display the MOPs commented and the comment text that has been entered for each. 2) The screen would also provide an edit comments button that will allow for editing/addition of comments from the debrief screen.
Addresses: V3

1) Important - Medium
2) Nice to have - Hard

Priority 3

None.

Priority 4

C2 Rename inconsistently named screens.

Nice to have - Easy

J2 Create a word wrapping function for the Events Timelines.

Nice to have – Easy/Medium

P1 Add a counter to indicate progress in TE/MOP completion.

Nice to have - Hard

P3 Attempt to place the Event Time indicator inside the box that lists TE, TO, EO and MOP.

Nice to Have - Medium

R3 Change the label on the Comment Screen OK button to Save vice OK.

Nice to have - Easy

S1 Add a Cancel button to the Comment Screen.

Nice to have – Easy/Medium

W2 Modify Pre-brief to provide the IITSEC specific pre-brief materials.

Nice to have - Medium

W3 Develop functionality that allows 1 trainer to collect data on more than 1 trainee.

Nice to have - Hard

<u>Not Important Things to Do</u>

Priority 4

P4 Determine whether Time of Event label will fit in space available.

Not Important - Easy

Create a Help/Training Function – Important for Integrated. Will not be done for I/ITSEC.

A4 Ensure functionality of the maximized screen and the access button are detailed in training/help materials.

B1 Ensure there is a mention that when selected, tabs and buttons will turn white in the training/help material. This is a basic Windows CE function. There is no provision for changing the tab color to something else other than white when selected.

E1 Add a help function.

H1 Positioning a row of tabs at the front of the grouping when selected is standard Windows functionality. No change to the screen will be made. Add a reference regarding the changing tab positions to the training/help material.

I1 The Scenario Time is currently displayed in a form that is inherently understood by training team members - Start+HH:MM:SS. Add a reference regarding the format of the Scenario Time to training/help material.

L1 Changing the button labels contradicts previous recommendation to keep these the same. No change will be made, however comments regarding the order of screen appearance and meaning of navigation buttons will be included in the training/help material when developed.

Q1 Changing button labels on the clock to VCR type controls not necessary. An explanation of clock operation will be included in the training/help materials.

R1 Provide training/help information on text input methods. An ink note capability exists as included functionality w/in the iPaq. Another useful function that has been added is the transcriber that converts handwriting to text.

V1 In the Debrief Screen, the software currently provides the first letter of each of the cause codes selected as a reason for the MOP being evaluated as incomplete. This is a good way to concisely indicate which cause codes apply to the MOP and should be readily apparent to any training team member. Add comments explaining the structure of the debrief and the meaning of the incomplete cause codes to the training/help material to avoid any confusion for the user.

No Action to be Taken

F5 eVB does not include "Expand All" and "Collapse All" functionality for treelists.

G3 Need fleet feedback to provide labels that will be most meaningful and understandable to end users. Abbreviations and truncations currently used on existing screens are meaningful to training team members however improvements may be gained through fleet feedback.

J4 No action.

K4 Need fleet feedback to determine most desirable order for presenting screens when Assessment screen is initially called up.

M4 No action.

O1 No Action.

References

Data to go. (2001, May). *Consumer Reports*, 20-23.

Eberts, R. E. (1994). *User interface design*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Gravlee, C. C. (2002). Mobile computer-assisted personal interviewing with handheld computers: The Entryware System 3.0. *Field Methods* 14(3), 322-336.

Hamel, C. J. & Clark, S. L. (1986). CAI evaluation checklist: Human factors guidelines for the design of computer-aided instruction. *Technical Report: NAVTRASYSCEN TR86-002*. Navy Training Systems Center – Human Factors Division: Orlando, FL.

Lynch, P. J., & Horton, S. (1999). *Web style guide: Basic design principles for creating web sites*. New Haven, CT: Yale University Press.

Lyons, D. M., & Allen, R. C. (2000). Mobile aid for training and evaluation (MATE): A hand-held, configurable set of team performance measurement tools. *Proceedings of the 2000 Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)*, (pp. 661-671).

Mandel, T. (1997). *The elements of user interface design*. New York: John Wiley & Sons, Inc.

Nielsen, J. (1993). *Usability engineering*. Boston: Academic Press.

Noldus Information Technology (2003, July 10). *Noldus products*. Retrieved July, 10, 2003, from http://www.noldus.com/products/index.html

RCI (Producer). (2000). Battle force tactical training (BFTT) system overview [CD ROM]. Virginia Beach, VA: Combat Direction Systems Activity, Dam Neck Naval Surface Warfare Center.

Ricci, K., Allen, R.C., Reynolds, A., Daskarolis-Kring, E., & Hodak, J. (2001). *An evaluation of the AFLOAT training exercise and management system (ATEAMS) software*. Orlando, FL: Naval Air Warfare Center Training Systems Division.

Sanders, M. S. & McCormick, E. J. (1987). *Human factors in engineering and design*. New York: McGraw-Hill, Inc.

Stretton, M. (2001). Afloat training, exercise, and management system(ATEAMS): Enabling objective-based training. *Proceedings of the 2001 Interservice/Industry Training, Simulation and Education Conference (I/ITSEC),* (pp.1429-1519).

The Windows®Iinterface Guidelines for Software Design. (1995). Redmond, WA: Microsoft Press.

Weber, B. A., Roberts, B. L. (2000, May/June). Data collection using handheld computers. *Nursing Research, 49*(3), 173-175.

Waterson, S., Matthews, T., & Landay, J. A. (2002). In the lab and out in the wild: Remote web usability testing for mobile devices. *Extended Abstracts of Human Factors in Computing Systems: CHI 2002*, Minneapolis, MN, April 20-25.

Wickens, C. D. (1992). *Engineering psychology and human performance* (2nd ed.). New York: HarperCollins Publishers, Inc.

Acronym List

APDA – ATEAMS Personal Digital Assistant
ATEAMS – Afloat Training Exercise and Management System
BFTT – Battle Force Tactical Trainer
EO – Enabling Objective
GUI – Graphical User Interface
MOP – Measure of Performance
OBT – Objective Based Training
PDA – Personal Digital Assistant
ROE – Rules of Engagement
TE – Training Event
TO – Terminal Objective