Threat Simulation -- A COMPARISON OF TWO TECHNIQUES

Nathaniel R. League
Electronic Combat Operations
AAI Corporation
PO Box 126
Hunt Valley, MD 21030-0126

ABSTRACT

In an Electronic Warfare (EW) training environment it is necessary to model threat radars and weapons systems in order to set up specific training missions. These threats react to the student's actions throughout the training session. This paper discusses what threat reactions are and why they are required. The paper then presents two different techniques for controlling the threat's reactions to student actions in a training environment. One technique is more complex but offers greater flexibility. The other technique is more generic and less complex for the user but offers less flexibility.

The first technique uses an interpretive language that allows the user to program the reactions of each threat in the training environment. Examples of how this language is used to program threats are given. With this technique, the threats are completely flexible. However, in addition to understanding EW systems, the user must be a capable programmer in order to properly code and debug the threats.

The second technique uses generic threat reaction algorithms. Examples of these reaction algorithms are presented. This technique is less flexible than the first technique because it has a limited number of algorithms which are utilized repeatedly to simulate all of the threats in the training environment. However, the user need only fill in the blanks on a preprinted CRT display in order to define the threat reaction decision data. This technique requires that the user have a basic understanding of EW systems. But, the user does not need to be a programmer.

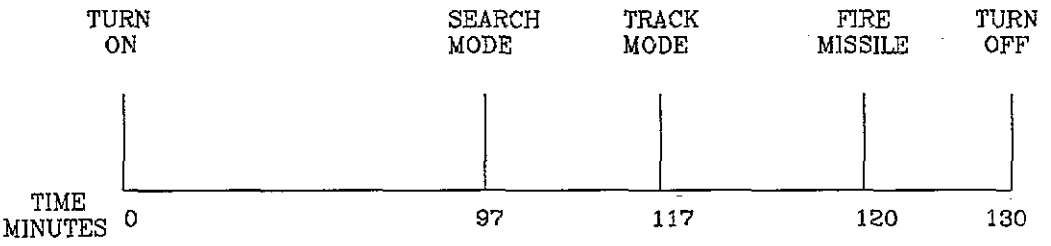
The two techniques are presented individually, then compared to highlight the differences in cost, lines of code, memory and CPU time.

INTRODUCTION

In an Electronic Warfare (EW) trainer, there are one or more students being trained to defeat threat radars and weapon systems in order to accomplish a specific task, such as bomb a site. The student is being taught how to defeat these weapon systems in order to protect himself or some other vehicle. The student typically attempts to defeat a threat radar by using one of several Electronic Countermeasures (ECM) techniques or through maneuvering. Electronics countermeasures, or jamming, is a technique whereby noise or false targets are transmitted into a threat's receiving equipment to prevent detection of one or more approaching vehicles.

In order to provide realistic training, the EW training device must allow threat radars and weapon systems to react to the student's ECM actions. For instance if a student is jamming a threat with a certain jamming technique, the threat radar should invoke any countering technique it has to defeat the jamming technique being used. These countering techniques for defeating jamming are called Electronic Counter Countermeasures (ECCM). They include techniques such as changing frequency, changing PRF, automatic gain control (AGC), fast time constant (FTC), moving target indicator (MTI) as well as others. The student detects these weapon system ECCM changes on whatever monitoring equipment he has available. These reactions of the threat to the student's actions are called threat reactions and are the subject of this paper.

It is important for the student to see these threat reactions because he needs to know what to expect of the threat in a real-world situation. In the past, threats where modeled using a technique called "time triggered threat moding". With this technique, threats changed there signal characteristics based on a preprogrammed scenario. For instance, a certain threat may have become active in its search mode at 97 minutes into the training mission. Then 20 minutes later it may have changed from a search mode to a tracking mode. It was up to the instructor to program the time line and make sure that the student's vehicle was at the right place at the right time so that the threat would appear to be acting properly. There were two problems with this approach. First, it took many hours for the instructor to put together the threat time line and get everything coordinated. Second, and more important, a threat would react in a given way regardless of whether the student performed a correct action or an incorrect action.



TIME TRIGGERED THREAT MODING

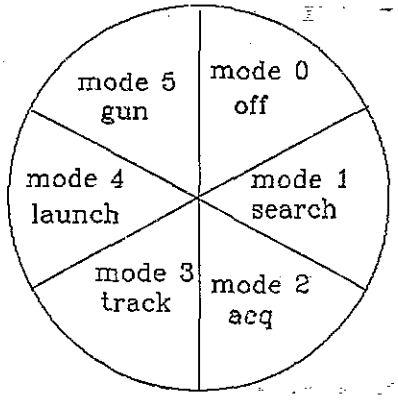
With the two techniques being presented, threats react to a student depending on the combination of his position, rate of angular change, speed and ECM activity(jamming). Two of these techniques are presented in the following paragraphs. After the explanation of the two techniques an example is discussed comparing the two techniques as to capability. The next section of this paper will then contrast the two techniques as to CPU time, memory utilization and ease of use. The two techniques being described will be referred to as technique A and technique B throughout this paper. Technique A was used by AAI on the B-52 WST trainer and on the EF-111 OFT trainer. Technique B was used on the A-10, F-4D, F-16 and EA-6B trainers.

A threat is modelled by breaking it into several subsections. For instance if a threat has a search radar and an associated track radar, it would be modelled as having three subsections: off, search and track. With technique A, these subsections are called states. When using technique B, they are called modes. The two terms are used throughout the remainder of this paper. They can be thought of as interchangeable.

So, a mode (or state) is one of the subsections of the threat. The number of modes required to model a threat is dependent on the threat complexity. For every combination of signal parameters a new mode may need to be defined. The number of modes required is usually dependent on the number of different signals that the threat can generate serially. The first mode, or state, in each algorithm is the "off" mode. The "off" mode occurs when a threat is first loaded into the environment but has not passed its criteria to start radiating signals.

Technique A

Threat reaction technique A is typically used on systems where more complex threat reaction modeling is required. Technique A uses a threat reaction algorithm language which allows the user to program the reaction of the threat to virtually any action which may occur in the training exercise. This technique generates a unique program for each threat to be modelled in the training mission. An example of the language follows:



BOOMER MODE BREAKOUT

```
SEARCH                                    STATE IDENTIFIER

          ON.SIGNAL (1)          ---
          SET.SCAN (1,1)           1    INITIAL CONDITIONS
          SET.PRF (1,1)          __1

     TGT.RNG LT 20 NM            ---
     TGT.RNG GT 8 NM               1
     J/S.1 LT 20 DB                1    TACTICAL OPERATION
          GO.TO ACQUISITION      __1
```

The threat reaction algorithm consists of three parts. The state identifier is the beginning address for the particular state being processed. Technique A breaks an algorithm into a maximum of ten subsections called states. These states represent the different operating states or modes of the radar system being modelled. There may be one state for a search mode of a radar, one for the acquisition mode of a radar and so forth.

The initial conditions are used to set up any signals as required. The tactical operation attempts to transition the threat to the next state as appropriate. Each time the reaction algorithm runs it executes only its current state. Every statement in the current state may be executed each iteration except the initial conditions. The initial conditions are run only the first iteration after changing to a particular state. When the "GO.TO" statement is executed, the statement processing for the current state is terminated. The functioning of the entire threat reaction algorithm is to, based on the geometric relationship and the EW environment, transition the threat to its next logical state. Every state in the threat has a unique set of signal characteristics which are generated and presented to the trainee on his equipment.

The language used for technique A was first developed for the B-52 WST defensive station. The language has over 120 test/commands which can be used to program a threat. Each of these tests/commands are called statements. Each threat reaction algorithm can contain up to 200 statements. However, the typical threat reaction algorithm contains approximately 80 statements.

This approach, technique A, is extremely flexible. The user can generate a very accurate and "intelligent" threat reaction model. However, it is obvious that the person building these threat reactions must have a significant amount of programming capability.

Technique B

The second approach is to have a number of canned algorithms built and let the user just fill in the blanks. This approach shall be called technique B. In an off-line editor, the user must call up one of the available algorithms, usually around thirty are available. The algorithm is then presented to the user in a structured format on a CRT display. The user need only fill in the blanks to provide the pertinent information.

Below is an example of one of these algorithm formats:

```
IF RANGE LT _____ NM
     THEN
          IF RANGE GT _____ NM
               THEN
                    GO TO MODE _____
          ENDIF
ENDIF
```

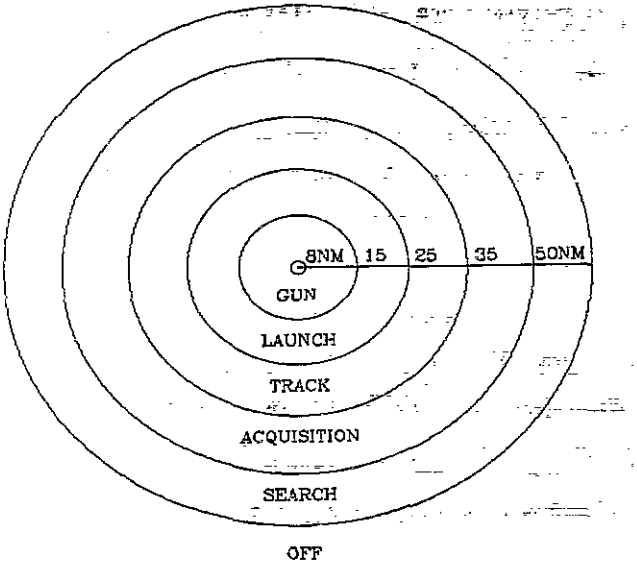
If the user selects the above algorithm, he needs to provide three pieces of information: the first or outer range, the second or inner range and the designation for the next mode. If the two tests are successfully passed, the algorithm will transition control to the specified mode for the subject threat. The user will need to script one algorithm for each mode defined for the subject threat. With this system the user can have up to seven modes per threat. Each mode contains its own reaction algorithm and signal set. Seven modes was chosen as the limit because it provides enough detail to properly simulate most threats. With this approach, the user has less flexibility, only that designed into the thirty or so algorithms. But, his work load is considerably reduced. He no longer needs to be a programmer.

Again, the functioning of the threat reaction algorithm is to transition the threat to its next logical mode. Every mode in the threat, with this technique, also has a unique set of signal characteristics which are generated and presented to the trainee on his equipment.

The method used for technique B was first developed for the A-10 EW trainer. Technique B, is easier to use than technique A. The user does not need to be a programmer. The user can still generate an accurate and "intelligent" threat without having any programming capability.

Threat Control, A Comparison of Techniques

In the example shown below, the infamous threat Boomer has five modes of operation plus an "off" mode. In this example the threat transitions between most modes based purely on range. When the ownship enters the threat's detection range (50 nm), the threat will turn on its "search" mode. When the ownship gets within 35 nm the threat transitions to its "acquisition" mode. The following diagram shows at what ranges each of the five modes is active:



Following are two different ways of scripting this threat to react as indicated in the table above. The first one presented is technique A.

Technique A

The technique A reaction algorithm for the subject threat is contained in Figure one. The threat will enter the environment based on a scripted activation range which is external to the threat reaction process. Once in the environment, the threat will stay in an "off" state, not radiating, until the ownship gets within 50 nm. In the "off" state, the reaction algorithm turns off all four signals. These four commands are called the initial conditions of the state. The initial conditions can exist for every state. They start with the first command after the state name and end with the first test in the state. The initial conditions are executed only when a state change occurs.

Each time the algorithm runs, once per second, it will test to determine if the ownship is within the 50 nm range. If and when the ownship penetrates this range, the threat will change to its "search" state and turn on signal number one. Signal number one for this threat has been defined via another editor and is resident on the disk. The parameters for signal number one are loaded into the simulation hardware and will show up on the trainee's ownship equipment. The threat will stay in its "search" state until the ownship gets within 35 nm or goes outside of 50 nm. If the ownship gets within 35 nm, the threat will transition to its Acquisition state. If the ownship goes outside of the 50 nm range, the threat will go to its "off" state.

The reaction algorithm is coded to attempt to transition to the proper state either moving up or down in lethality. The example shown is intentionally simple. A typical reaction algorithm would include tests for jamming, range rate, angular position or rate of change as well as other geometric or environmental checks. In the example, if the trainee gets within 25 nm, the threat will transition to its "track" state. Once in the "track" state, this threat checks to determine if it can launch a missile. In order to launch a missile, the ownship must be within the firing range of the missile. It must also be within the angular limits of the missile's firing cone. If these conditions are met and if the trainees jamming effort is inadequate, the threat will transition to its "launch" state. In the "launch" state, a missile launch will be simulated. The missile launched in this case is a guided missile. Therefore, the trainee will receive a missile guidance signal on his equipment and the tracking signal will also remain on.

While in the "launch" state, the threat will launch a missile as long as the ownship is within the firing cone, there are less than three missiles in flight and it has been 10 seconds or more since the last missile launch. The threat will stay in this state until the trainee gets to within eight (8) nm, or goes outside of the fifteen (15) nm firing range. If the threat gets within eight (8) nm, the threat will fire its guns at the trainee. The guns will only show up on trainers with active radars to pick up the gun burst. If a trainee flies close enough to the target, he will see the threat transition from its

```
BOOMER

    OFF
                    OFF.SIGNAL (1)
                    OFF.SIGNAL (2)
                    OFF.SIGNAL (3)
                    OFF.SIGNAL (4)
                TGT.RNG LT 50 NM
                    GO.TO (SEARCH)

    SEARCH
                    ON.SIGNAL (1)
                TGT.RNG LT 35
                    GO.TO (ACQUISITION)
                TGT.RNG GT 50
                    GO.TO (OFF)

    ACQUISITION
                    OFF.SIGNAL (1)
                    ON.SIGNAL (2)
                TGT.RNG LT 25 NM
                    GO.TO (TRACK)
                TGT.RNG GT 35
                    GO.TO (SEARCH)

    TRACK
                    OFF.SIGNAL (2)
                    ON.SIGNAL (3)
                TGT.RNG LT 15
                TGT.EL  LT 20
                TGT.AZ  LT 37
                J/S.3 LT 20 DB
                    GO.TO (LAUNCH)
                TGT.RNG GT 25
                    GO.TO (ACQUISITION)
                J/S.3 GT 20 DB
                    WAIT (5) SEC
                    CHANGE.FREQ (3)

    LAUNCH
                    ON.SIGNAL (4)
                TGT.RNG LT 15
                TGT.EL  LT 20
                TGT.AZ  LT 37
                J/S.3 LT 20 DB
                SAMS.IN.AIR LT 3
                LAST.LAUNCH MT 10 SEC.AGO
                    FIRE.SAM
                TGT.RNG LT 8
                    GO.TO (GUN)
                TGT.RNG GT 15
                    GO.TO (TRACK)

    GUN
                    OFF.SIGNAL (4)
                    FIRE GUN
                TGT.RNG GT 8
                    GO.TO (LAUNCH)
```

TECHNIQUE A

FIGURE 1

"off" state down to its "gun" state and then, on the way out, back to its "off" state.

The threat reaction algorithm language developed for technique A has over 120 tests/commands. These test/commands can be concatenated in any order to program a threat to react as desired.

Technique B

The same threat situation can be programmed using technique B. For this paper a system was chosen using only ten (10) canned reaction algorithms. For each mode of Boomer, there will be one reaction algorithm. However, the reaction algorithms will be reused with different data, as can be seen in Figure 2.

Notice that there are two "track" modes scripted for the threat, modes 3 and 4. Technique A allowed the user to change frequency within the algorithm for the "track" state. With technique B, the user needs to request a mode change to bring in a different track mode with different signal parameters (ie: frequency). The mode change thus causes a change to the signal transmitter frequency.

The reaction algorithm for the "off" mode, which is mode zero, is reaction algorithm ten (10). This generic reaction algorithm tests to see if the ownship is within the scripted range from the threat. If it is, a mode change to the specified mode will be requested. Notice the "minimum time in mode" field. The value in this field determines the minimum time estimated to perform a function, in the real world, such as go from track to launch. It essentially forces the reaction algorithm to slow down to a realistic speed rather then change through the modes very rapidly. Once the minimum time has expired, the threat is evaluated each second for the possibility of a mode change.

The reaction algorithm for the "off" mode will be loaded whenever the threat first becomes active. The algorithm will be processed to capture the "minimum time in mode" value. In this case, 10 seconds is scripted. Therefore, no more processing will be performed for this threat for 10 seconds. After the ten (10) second period has expired, the algorithm will be processed once per second. As long as the ownship stays outside of the 50 nm range, the threat will stay in its "off" mode. Once the ownship gets inside of the 50 nm range, the threat will request a mode change to mode 1, its "search" mode. Thus the signal data scripted for mode 1 of this threat will be loaded and displayed on the trainees equipment.

The reaction algorithm for the "search" mode (mode 1), the "acquisition" mode, (mode 2) and both track modes (modes 3 and 4) is reaction algorithm eight (8). This reaction algorithm first tests to determine if the subject threat is being successfully jammed. If so, a mode change to the scripted mode is requested and processing ends. If the threat is not jammed, the algorithm will check to determine if the ownship is outside of, or beyond, some scripted limit. If so, a mode change to the specified mode is requested and processing terminates. If the ownship is not beyond the scripted range, a check is made to determine if the ownship is within some scripted

```
BOOMER

     MODE: OFF (0)        REACTION ALGORITHM: 10

     IF OWNSHIP RANGE IS WITHIN __50___
          THEN
               SELECT MODE __1___
     ENDIF

     MINIMUM TIME IN THIS MODE IS __10___ SECONDS.


     MODE: SEARCH (1)          REACTION ALGORITHM: 8

     IF JAMMING EFFECTIVENESS IS ABOVE __100___ THEN
          SELECT MODE __0___
     ELSE IF OWNSHIP RANGE IS BEYOND __50___ THEN
          SELECT MODE __0___
     ELSE IF OWNSHIP RANGE IS WITHIN __35___ THEN
          SELECT MODE __2___
     ENDIF

     MINIMUM TIME IN THIS MODE IS __20___ SECONDS.


     MODE: ACQUISITION (2)          REACTION ALGORITHM: 8

     IF JAMMING EFFECTIVENESS IS ABOVE __100___ THEN
          SELECT MODE __0___
     ELSE IF OWNSHIP RANGE IS BEYOND __35___ THEN
          SELECT MODE __1___
     ELSE IF OWNSHIP RANGE IS WITHIN __25___ THEN
          SELECT MODE __3___
     ENDIF

     MINIMUM TIME IN THIS MODE IS __10___ SECONDS.


     MODE: TRACK1 (3)          REACTION ALGORITHM: 8

     IF JAMMING EFFECTIVENESS IS ABOVE __60___ THEN
          SELECT MODE __4___
     ELSE IF OWNSHIP RANGE IS BEYOND __25___ THEN
          SELECT MODE __2___
     ELSE IF OWNSHIP RANGE IS WITHIN __15___ THEN
          SELECT MODE __5___
     ENDIF

     MINIMUM TIME IN THIS MODE IS __10___ SECONDS.


     MODE: TRACK2 (4)          REACTION ALGORITHM: 8

     IF JAMMING EFFECTIVENESS IS ABOVE __60___ THEN
          SELECT MODE __3___
     ELSE IF OWNSHIP RANGE IS BEYOND __25___ THEN
          SELECT MODE __2___
     ELSE IF OWNSHIP RANGE IS WITHIN __15___ THEN
          SELECT MODE __5___
     ENDIF

     MINIMUM TIME IN THIS MODE IS __10___ SECONDS.


     MODE: LAUNCH (5)          REACTION ALGORITHM: 3

     IF JAMMING EFFECTIVENESS IS ABOVE __60___ THEN
          SELECT MODE __2___
     ELSE IF CHAFF EFFECTIVENESS IS ABOVE __100___ THEN
          SELECT MODE __0___
     ELSE IF MANEUVER EFFECTIVENESS IS ABOVE __100___ THEN
          SELECT MODE __0___
     ELSE IF OWNSHIP
          (RANGE IS BETWEEN __8___ AND __15___ NM) AND
          (ELEV ANGLE IS BETWEEN __0___ AND __20___ DEG) AND
          (ALTITUDE IS LESS THAN __200___ METERS) THEN
          SET LAUNCH ENABLE FLAG TO TRUE
     ELSE SET LAUNCH ENABLE FLAG TO FALSE
          IF OWNSHIP RANGE IS BEYOND __15___ THEN
          SELECT MODE __3___
     ELSE IF OWNSHIP RANGE IS WITHIN __8___ THEN
          SELECT MODE __6___
     END IF

     MINIMUM TIME IN THIS MODE IS __10___ SECONDS.


     MODE: GUN (6)        REACTION ALGORITHM: 3

     IF JAMMING EFFECTIVENESS IS ABOVE __100___ THEN
          SELECT MODE __0___
     ELSE IF CHAFF EFFECTIVENESS IS ABOVE __100___ THEN
          SELECT MODE __0___
     ELSE IF MANEUVER EFFECTIVENESS IS ABOVE __100___ THEN
          SELECT MODE __0___
     ELSE IF OWNSHIP
          (RANGE IS BETWEEN __0___ AND __8___ NM) AND
          (ELEV ANGLE IS BETWEEN __0___ AND __90___ DEG) AND
          (ALTITUDE IS LESS THAN __1000___ METERS) THEN
          SET LAUNCH ENABLE FLAG TO TRUE
     ELSE SET LAUNCH ENABLE FLAG TO FALSE
          IF OWNSHIP RANGE IS BEYOND __8___ THEN
          SELECT MODE __5___
     ELSE IF OWNSHIP RANGE IS WITHIN __8___ THEN
          SELECT MODE __6___
     END IF

     MINIMUM TIME IN THIS MODE IS __10___ SECONDS.
```

FIGURE 2

range. If it is, a mode change is requested to the specified mode.

Since these algorithms are generic, there are times when the user may want some reaction algorithm test(s) to always be false, or possibly, always true. Notice in the algorithm for the "search" mode that the first test checks for a value of greater then one-hundred. Since jamming is always a positive number, from zero to 100, the test result will always be false. The user needed to "deactivate" the first test because he is not interested in checking for jamming in this mode. Since there are a limited number of canned algorithms, they are often tailored to deactivate any undesired tests.

After changing to the "search" mode, the *algorithm processing is suspended for 20 seconds.* However, the signal data is generated immediately and sent out to the trainee's equipment. After the 20 second period has expired, the reaction algorithm will check for a jamming value above 100%. This test will always fail. Next, a check is made to see if the ownship has gone out of the 50 nm limit. If so, the algorithm transitions the threat back to its "off" mode. If the ownship is within 35 nm, the algorithm transitions the threat to its "acquisition" mode. If the ownship is between the 35 and 50 nm area, the threat will stay in its "search" mode "radiating" its search signals.

The processing for the "acquisition" mode is virtually identical to the processing for the "search" mode. If the ownship goes outside of the 35 nm range, the threat transitions back to its "search" mode. If the threat gets to within 25 nm, the threat transitions to its "track1" mode. Otherwise, the threat stays in its "acquisition" mode, "radiating" its acquisition signal(s).

The "track1" mode does use the first test of the reaction algorithm. If the trainee's jamming level exceeds 60%, the threat will transition to its "track2" mode which is identical to the "track1" mode except that it radiates at a different frequency. This mode change is performed in an effort to reduce the trainee's jamming effectiveness to something below 60%. If the jamming is thus defeated, one of the track mode algorithms will test to determine the range of the ownship. If it is outside of 25 nm, the threat will transition back to its "acquisition" mode. If the ownship comes within 15 nm, the threat will transition to its "launch" mode.

A new reaction algorithm, number 3, is used for the "launch" mode. This reaction algorithm allows the user to check for three different ECM techniques and branch to a new mode as required. If the threat is not jammed, tests are made to determine if the ownship is within the threat's missile cone. If so, a missile launch is requested. If a launch cannot be made, the algorithm will go on to its two range checks and branch to the requested mode.

For this particular threat, the user is checking only for jamming effectiveness. The tests for chaff and maneuver effectiveness have been "deactivated". If jamming effectiveness is above 60%, the threat will go back to tracking the ownship. If the jamming is below 60%, the algorithm will test to determine if the ownship is within the threat's firing cone. If the ownship is within the firing cone a missile launch will be attempted. If a launch cannot be attempted, the algorithm goes on to check on ownship range. If it is outside of the 15 nm firing range, the threat transitions back to its "track1" mode. If the ownship is too close for a missile launch (within 8 nm) the threat transitions to its "gun" mode.

The algorithm for the "gun" mode is the same as for the "launch" mode. In this case all three ECM tests are "deactivated". A check is made to determine if the ownship is within the gun's firing range. If it is, gun firing is requested. When the ownship gets outside of the gun's firing range, the threat transitions back to its "launch" mode in an effort to fire a missile at the ownship *as it is on its way out. Notice that the last test* has been deactivated. As long as the ownship is within 8 nm, this test will just cause the threat to stay in the "gun" mode.

Following is a comparison of the two threat reaction techniques. The comparison addresses cost, flexibility, CPU time, memory usage, and disk space.

Comparison of Resource Utilization

The data presented in table 1 was obtained from the B-52 program for the technique A data and from the F-16 program for the technique B data. The data is representative of the average sizes and time for the various threats of the two systems. As can be seen, technique A requires slightly more memory than technique B. In addition, technique A uses overlays such that only

	TECHNIQUE A	TECHNIQUE B
memory used (bytes)	30k *	25k
CPU time used avg. (msecs)	100	37
CPU time used WC (msecs)	400	100
Disk space (bytes) required for 1000 copies of BOOMER	380,000	18,000
Flexibility	HIGHER	LOWER
Ease of use	HARDER	EASIER
Cost in hours to script and test BOOMER	40	20

* uses overlays so that only 1 threat is in memory at a time.

TECHNIQUE A/B COMPARISON CHART

TABLE 1

one (1) threat is in memory at a time. With technique B, more threats, from 20 to 100 depending on the system, are memory resident *simultaneously*.

The reaction algorithms for technique B take much less CPU time than those for technique A. Technique A takes the additional execution time because of the time it takes to step through the algorithm.

Technique A takes much more disk space than technique B. Technique B uses 18 bytes per threat. Technique A uses 380 bytes per threat on the average.

Technique A is more flexible than technique B. But technique B is easier to use. Consequently, on the average, it takes longer to script a threat using technique A than it does using technique B.

Summary

Technique A is a more flexible scripting technique. However, across the board it costs more in memory utilization, CPU time, disk space and cost to script. In addition, using Technique A is more complex than Technique B.

It turns out that both approaches have their place. On more complex systems where the trainee is an accomplished EWO or ECMO with lots of signal monitoring equipment, technique A would most likely be used. If the trainer is being designed to train new EWOs or ECMOs or if there is only one or two pieces of ECM gear, technique B may be used.

One must also consider resources when deciding which of the two techniques to use. On smaller systems with less memory and slower processor speeds, technique B is sometimes dictated.

Another important consideration is who will be changing or adding new threats to the trainer. Some instructors only serve at a training site for one year. If that person is doing the threat scripting, the data entry method should be as simple as possible. In this case, technique B is again recommended.

ABOUT THE AUTHOR

Nat League is a Senior Design Analyst with the Training and Simulation Division of AAI Corporation. Mr. League has been assigned as the project engineer on various simulation devices at AAI. Mr. League holds a B.S. in Computer Science from The University of Maryland. Mr. League has also completed work for an M.S. degree in Engineering Sciences from Loyola College. Mr. League has 18 years of experience working with real-time software in the fields of signal processing and training and simulation.