AUTOMATIC AUDIT INFORMATION SYSTEM FOR SOFTWARE DEVELOPMENT

Gary A. Brown
AAI Corporation
Cockeysville, Maryland

ABSTRACT

Management of a software development project is typically characterized by a lack of control and poor projections. Status reports are notoriously inaccurate; worse yet, the prerequisite software audits drain development resources from the design effort.

This paper describes an automated procedure for performing software audits and generating status reports. This procedure reduces the time required for both tasks significantly, and makes status reports available upon demand. Timely status reports furnish management with an early warning of problem areas so that project control can be exercised. For example, resources may be reallocated or additional resources employed where these problems are identified.

The Automatic Audit Information System for Software Development (AAIS) procedure has been implemented by AAI Corporation for the development of Device 20B5. It is based upon the following concepts:

* A central software development library
* Software development milestones and criteria
* Functional hierarchies
* A development scoreboard.

AAIS provides the 20B5 management with close project control by means of timely audits and concise status reporting.

Introduction

The technological revolutions that have yielded more powerful and sophisticated computer systems have encouraged defense contracts for more complex and higher fidelity training systems. As trainer specifications incorporate the new technology, designs such as those incorporated in Device 20B5 are often required for effective simulation and stimulation for team training. Team training implies a multi-task environment to support several operational equipments which are typically employed in cooperative roles. In this case, the simulation software developed exceeds the scope of medium-complexity software systems of the past, and becomes one of the complex software systems of the present. The development effort for complex systems is larger, by an order of magnitude, and must be carefully controlled in order to ensure its successful completion. This control must be implemented with concise and timely status reporting. However, the preparation of status reports must not interfere with the critical milestones established for design personnel. In addition, the information retrieved, analyzed and presented must accurately reflect the current status of the software development effort.

The Automatic Audit Information System for Software Development (AAIS) procedure has been developed to efficiently track software development with a minimum of design personnel interaction. The progress reports which are generated, provide both management and design personnel the information needed to monitor software progress and identify schedule and cost variances. The various report formats allow the manager to selectively scan the information and extract those items which are pertinent for analysis. The reports highlight milestone completion with regard to budget and schedule, which is of primary concern to management personnel.

The automated procedures incorporated in the AAIS program are controlled by criteria established during the infancy of the software development effort. These criteria are derived from contractual requirements such as those presented in MIL-STD-1644. In addition, documentation defined by the Data Item Descriptions (DIDs), in particular, the Program Design Specification (PDS), assist in formulating the Software Work Breakdown Structure (SWWBS) which becomes an integral foundation for the command and control facilities used to drive the AAIS program.

Design of the AAIS

The Automatic Audit Information System for Software Development (AAIS) procedure has evolved as a result of continuing efforts at AAI to track and control software development for training devices. The salient features which have emerged are:

* Software hierarchies
* Milestone completion criteria
* Software development scoreboard
* Centralized on-line development library

* Software module formats

* Automated audit and report procedure.

Each of the above features is described below in the context of its application to the development of Device 20B5. Figure 1 presents a diagram of the AAIS concept.

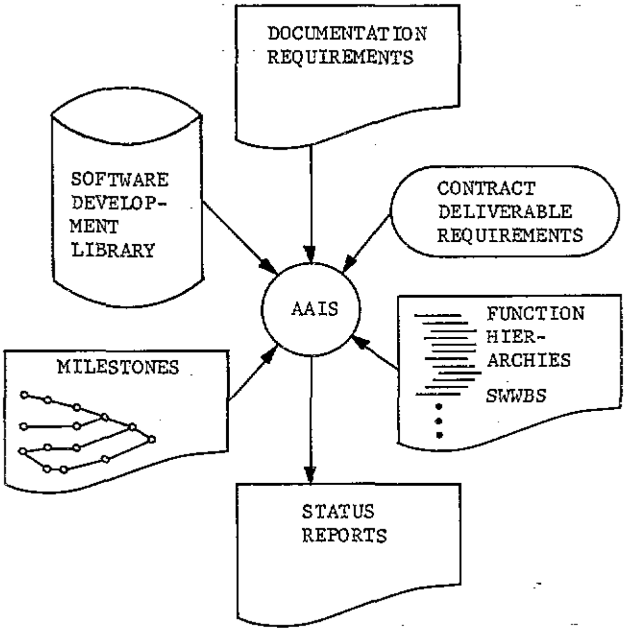

Figure 1. The AAIS Concept

The contract sponsor requires a hierarchal listing of software modules for each functional area defined in the Program Performance Specification (PPS) and detailed in the Program Design Specification (PDS). The sample hierarchy given in Figure 2 indicates naming conventions and the control structure defined for the function. These hierarchies are entered and maintained in an encoded form, known as the Software Work Breakdown Structure (SWWBS), on the computer storage media. The stored files represent a means of addressing a functional area down to the module level.

Milestones are provided by the customer in the form of contractual delivery and report dates. These dates imply that software managers must develop the necessary and detailed software development milestones required to meet the contract delivery dates. This effort amounts to critical path scheduling of available personnel within time and computer availability constraints. Five (5) software development milestones are defined as follows:

* Module design

* Module coding

* Module test driver design

* Module testing

* Function testing.

```
3.2.2.5.2     ASW ENVIRONMENT (SONOBUOY &
              AN/SQS-56 SONAR) FUNCTION SWWBS
              (PDS SECTION:  3.3.2.5.2)

SEL TASK:  NEW             COMPUTER:  OWNSHIP

NO2NEXEC                   NEW TASK EXECUTIVE
   NE4EXEC                 ASW ENVIRONMENT (SONOBUOY
                             & SQS-56) EXEC
      NE5WKMSK             WAKE MASKING CONTROL
         NE6WMREF          WAKE MASKING DATA FORMAT
         NE6WHERE          WAKE SEGMENT POSITIONS
         NE6STATS          SENSOR STATUS AND WAKE
                             POSITION
         NE6STREN          WAKE SEGMENT STRENGTH
      NE5DOPRT             DOPPLER RATIO
      NE5OPL56             SQS-56 OCEAN PROPAGATION
                             LOSS CONTROL
         NE6INTRP          PROPAGATION CONTROL
            NCZINTP2       TWO-WAY INTERPOLATION
      NE5SNBWO             SONOBUOY WASHOVER
      NE5OPRLS             OCEAN PROPAGATION LOSS
                             CONTROL
         NE6OPLRT          OCEAN PROPAGATION LOSS
                             RETRIEVAL
            NCZINTP2       TWO-WAY INTERPOLATION
      NE5AMBNS             AMBIENT NOISE
      NE5KBSHD             KELP BED SHADOWING
      NE5TCNIM             TCNI MANAGEMENT CONTROL
         NE6HARME          HARMONIC FAMILY DATA
            XDZDSCIO       DISC I/O QUEUING
         NE6RECRD          RECORD NUMBER
            XDZDSCIO       DISC I/O QUEUING
         NE6VALID          VALIDITY CHECK
            XDZDSCIO       DISC I/O QUEUING
            NCZBSRCH       BINARY SEARCH
      NE5RVERB             REVERBERATION
```

Figure 2. Sample Function Hierarchy

The completion criteria for each milestone is provided by project management and software team personnel. These criteria identify development requirements for the completion of each milestone. Completion of each milestone is predicted upon the existence of a critical item in the module file, e.g., the existence of the high level language (code) satisfies the coding completion criteria.

A development scoreboard is formed by the combination of a function hierarchy and software development milestones. A matrix of values can be obtained if a value is assigned to each milestone and to each module in the hierarchy. A milestone value represents a specific development weighting factor whereas a module value can be thought of as a design complexity factor for the design of the module. The resulting matrix elements are defined by computing the product of each combination of milestone value and module value. Milestones which are not completed are assigned a value of zero. The significance regarding the implications of this approach should not be overlooked. Applying a set of weighted values to milestones is in effect a means for budgeting the time required to fulfill each milestone as a fraction of the total time for an average module. At a modular level, this budgeting effort is much more precise than at a functional level. Typical estimates are available from the Pie chart illustrated in Figure 3 and in terms of industry standards for lines of code per programmer day. On the other hand, module weights are derived by allocating a total value to each function based upon perceived difficulty, and in turn, rationing this total among all modules of the associated function. The total value attributed to the function represents the amount of effort budgeted for the function. The ratio of the sum of all the matrix values to the total possible sum for the matrix is interpreted as the reportable completion percentage of the function at the time of audit.

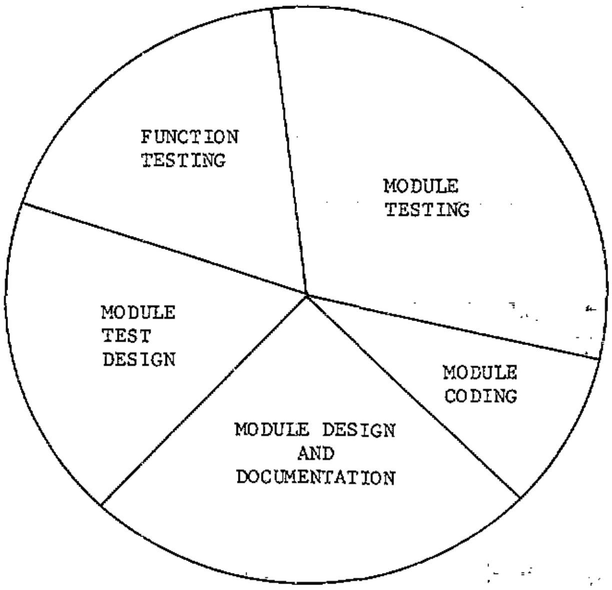

Figure 3. Function Development Costing

A centralized, computer-resident software development library makes possible automatic auditing of the software inventory. All modules are readily accessible and maintainable. Although modules are developed by individual designers, a module belongs to the project and resides in the control library. This feature is essential to project control, since all modules must be available for auditing purposes.

All modules in the central library are entered according to the format developed for Device 20B5. Templates are provided to designers and data entry personnel to simplify the process and insure standardization. The object of the format is to capture the high level language along with design summaries and internal documentation for the module, as depicted in Figure 4. These module files contain sufficient information for automatic generation of contractual reports such as the PDS. Module files represent self-contained repositories of information concerning the embedded execution language (FORTRAN) routine. This information also includes a module test procedure (a driver description) as well as configuration management data for the code and test procedure.

```
*************** 20B5 MODULE FILE ****************
*************** MODULE CONTROL DATA *************

*  MODULE NAME:  SA6SAADJ
*  MODULE TITLE:  SONOBUOY SONAR AMPLITUDE
                  ADJUSTMENT
*  MODULE SECURITY CLASSIFICATION:  U
*  MODULE PART NUMBER:  58181-7-0S1-36405-U
*  PROGRAMMER:  GABROWN

*************** PDS DATA ***********************

*  MODULE PROCESSING: SA6SAADJ WILL COMPUTE THE *
                      VESSEL SONAR AND INTER-
*                     FERING SONOBUOY AMPLITUDE
*                     DEGRADATION FACTORS OF THE
*                     RESULTING SIGNAL ENVELOPE
*                     WITH RESPECT TO EACH OF
*                     FOUR POSSIBLE SONOBUOYS ON
*                     CHANNEL.  THE DEGRADATION
*                     FACTORS ...

************** PROGRAM DESIGN LANGUAGE *********

*PDL- 1  DO FOR EACH VESSEL TARGET SLOT
*PDL- 2     IF THE VESSEL SONAR PING INDICATOR IS
            SET TO
*PDL- 3      TRUE THEN
*PDL- 4        DO FOR EACH SQR-17 CHANNEL

*************** MODULE REVISION HISTORY *********
*  0000 06/18/83 GAB INTEGRATED
*************** TEST PROCEDURE REVISION HISTORY *
*  0000 06/18/83 GAB INTEGRATED
*************** MODULE TEST PROCEDURE ***********

C      PROGRAM DRIVER
C      REAL * 4  ERVAL11  /-2.0/, TOLER11 /0.001/
C      CALL BEGINMTP (MODULE)
C      OINSHIPB = 1
C      AASPINGL = .TRUE.
C      CALL SA6SAADJ

*************** CODE ****************************
      SUBROUTINE SA6SAADJ
      INCLUDE SONOB
      INCLUDE SONOBP
*
*************************************************
*PDL- 1           DO FOR EACH VESSEL TARGET SLOT
*************************************************
*
      DO I = 1, OINSHIPB
*
*************************************************
*PDL- 2     IF THE VESSEL SONAR PING INDICATOR IS
            SET TO
*PDL- 3      TRUE THEN
*************************************************
*
       IF ( AASPINGL (I) ) THEN
```

Figure 4. Module File Template

Finally, based upon all of the above features, an automated auditing and status reporting procedure is employed to control software development. This procedure is driven by the manager's selection of one or more function hierarchies. For each function selected, the associated hierarchy is used to provide the module names and corresponding weights. The actual modules in the central library are automatically inspected and the criteria is applied to the retrieved information such that a set of reports can be prepared. Both module and function summations are computed, along with their respective percentages of completion. Discrepancies in module file formats are discernable from the resulting reports, providing quality control for both audits and module files. History files are updated to automatically record the current audit statistics. In addition to producing status reports, information reports are conveniently produced for review by software designers. These summaries provide highly useful organizational information.

AAIS Outputs: Status Reports

The AAIS program generates seven (7) distinctive report formats for review by both managers and design personnel. The report formats contain the following information:

(1) Functional Design History and Resource Utilization by module

(2) Functional Development Status by module

(3) Functional Milestone Status by module

(4) Functional Hierarchy by module

(5) Functional Configuration Status by module

(6) Milestone Summary by function

(7) Cost Performance Summary by work order (charge) number

Report formats (3), (6) and (7) are provided as the basic set of status outputs to the manager. These summaries describe software development status in terms of milestones and cost. Report formats (1), (2), (4) and (5) provide additional organizational information to the designer. These reports highlight module design status in terms of documentation requirements and configuration management data.

The Functional Design History and Resource Utilization report lists, on a module basis, development and testing revision histories. Figure 5 depicts a sample report format.

3.3.2.5.2 ASW ENVIRONMENT (SONOBUOY & SQS-56) -- FUNCTION DEVELOPMENT HISTORY -- 6/13/83 06:18

COMPUTER	SEL TASK	MODULE	PROG.	DISC	ENTRY DATE	MODULE REV. DATE	MODULE REV. LEVEL	TEST REV. DATE	TEST REV. LEVEL	WORST CASE TIME	WORST CASE MEM.
OWNSHIP	NEW TASK	NCZBSRCH	FINLEY	STS20B5E	4/12/82	12/14/82	0000	12/14/82	0000	30	280
OWNSHIP	NEW TASK	NCZINTP2	FINLEY	STS20B5E	4/13/82	06/03/82	0001	06/03/83	0001	8	496
OWNSHIP	NEW TASK	NE4EXEC	FINLEY	STS20B5E	3/4/82	03/10/83	0000	03/10/83	0000	816	1160
OWNSHIP	NEW TASK	NE5AMBNS	FINLEY	STS20B5E	2/22/82	05/11/83	0001	ERR	0001	300	208
OWNSHIP	NEW TASK	NE5DOPRT	FINLEY	STS20B5E	2/22/82	01/11/83	0002	01/11/83	0002	5	200
OWNSHIP	NEW TASK	NE5KBSHD	FINLEY	STS20B5E	2/19/82	03/10/83	0000	03/10/83	0000	250	712
OWNSHIP	NEW TASK	NE5OPL56	FINLEY	STS20B5E	04/13/82	03/02/83	0000	03/02/83	0000	250	1088
OWNSHIP	NEW TASK	NE5OPRLS	FINLEY	STS20B5E	4/14/82	03/10/83	0000	03/10/83	0000	100	608
OWNSHIP	NEW TASK	NE5RVERB	FINLEY	STS20B5E	09/27/82	05/19/83	0001	05/19/83	0001	100	576
OWNSHIP	NEW TASK	NE5SNBWO	FINLEY	STS20B5E	2/19/82	03/01/83	0000	03/01/83	0000	30	144
OWNSHIP	NEW TASK	NE5TCNIM	FINLEY	STS20B5E	03/05/82	04/11/83	0003	04/11/83	0003	300	840
OWNSHIP	NEW TASK	NE5WKMSK	FINLEY	STS20B5E	07/26/82	02/22/83	0000	02/22/83	0000	100	1040
OWNSHIP	NEW TASK	NE6HARME	FINLEY	STS20B5E	16JUN82	04/12/83	0002	04/12/83	0002	10	432
OWNSHIP	NEW TASK	NE6INTRP	FINLEY	STS20B5E	04/15/82	06/02/83	0001	06/02/83	0001	2	368
OWNSHIP	NEW TASK	NE6OPLRT	FINLEY	STS20B5E	04/14/82	03/21/83	0000	03/21/83	0000	30	336
OWNSHIP	NEW TASK	NE6RECRD	FINLEY	STS20B5E	05/24/82	05/12/83	0002	05/12/83	0002	100	312
OWNSHIP	NEW TASK	NE6STATS	FINLEY	STS20B5E	07/23/82	02/21/83	0000	02/21/83	0000	100	192
OWNSHIP	NEW TASK	NE6STREN	FINLEY	STS20B5E	07/27/82	02/21/83	0000	02/21/83	0000	100	328
OWNSHIP	NEW TASK	NE6VALID	FINLEY	STS20B5E	05/21/82	04/11/83	0001	04/11/83	0001	200	256
OWNSHIP	NEW TASK	NE6WHERE	FINLEY	STS20B5E	07/23/82	02/22/83	0000	02/22/83	0000	300	560
OWNSHIP	NEW TASK	NE6WMREF	FINLEY	STS20B5E	07/28/82	02/22/82	0000	02/22/83	0000	100	656
OWNSHIP	NEW TASK	TC4EINIT	FINLEY	STS20B5E	09/27/82					100	632
OWNSHIP	NEW TASK	XDYTAMID	FINLEY	STS20B5E	6/3/82	03/22/83	0002	03/22/83	0002	100	528
OWNSHIP	NEW TASK	XDYTCNID	FINLEY	STS20B5E	09/27/83					100	260
OWNSHIP	NEW TASK	XDZTCN1D	FINLEY	STS20B5E	01/20/83	06/08/83	0000	06/08/83	0000	100	1736
OWNSHIP	NEW TASK	XDZTCN2D	FINLEY	STS20B5E	01/20/83	06/07/83	0000	06/07/83	0000	100	840
OWNSHIP	NEW TASK	XDZTCN3D	FINLEY	STS20B5E	03/14/83	06/01/83	0000	06/02/83	0000	100	136
OWNSHIP	NEW TASK	XDZTCN4D	FINLEY	STS20B5E	01/15/83	05/31/83	0000	05/31/83	0000	100	928
OWNSHIP	NEW TASK	TN2TCNIX	FINLEY	STS20B5E	09/27/82					100	1000
OWNSHIP	NEW TASK	TNZTCN1D	FINLEY	STS20B5E							
OWNSHIP	NEW TASK	TNZTCN2D	FINLEY	STS20B5E							
OWNSHIP	NEW TASK	TNZTCN3D	FINLEY	STS20B5E							
OWNSHIP	NEW TASK	TNZTCN4D	FINLEY	STS20B5E							
OWNSHIP	NEW TASK	TC6GLB55	FINLEY	STS20B5E	05/19/83					100	100
OWNSHIP	NEW TASK	TC6GLB53	FINLEY	STS20B5E	05/19/83					100	100
OWNSHIP	NEW TASK	AMBNS2	FINLEY	STS20B5E	05/18/83					100	100

Figure 5. Function Design History and Resource Utilization Report

The *Functional Development Status* report denotes, on a module basis, the current state of design for a function. The design states are categorized by the major milestones assigned to a module. In this way, the designer can quickly assess the overall development state of a function hierarchy and identify incomplete or incorrect data entries. Figure 6 shows a *Functional Development Status* report.

3.3.2.5.2 ASW ENVIRONMENT (SONOBUOY & SQS-56) -- FUNCTION COMPLETION STATUS -- 06/13/83 06:18

COMPUTER	SEL TASK	MODULE	PROG.	DISC	CONF CONT	FILE STATUS	COMPLETION STATES HEADER	PDL	CODE	MTP	MOD REV	TEST REV	I
OWNSHIP	NEW TASK	NCZBSRCH	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0000	0000	G
OWNSHIP	NEW TASK	NCZINTP2	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0001	0001	S
OWNSHIP	NEW TASK	NE4EXEC	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0000	0000	G
OWNSHIP	NEW TASK	NE5AMBNS	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0001	0001	S
OWNSHIP	NEW TASK	NE5DOPRT	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0002	0002	G
OWNSHIP	NEW TASK	NE5KBSHD	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0000	0000	G
OWNSHIP	NEW TASK	NE5OPL56	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0000	0000	E
OWNSHIP	NEW TASK	NE5OPRLS	FINLEY	STS20B5E	NO	ENTERED	H 1	P	C	M	0000	0000	S
OWNSHIP	NEW TASK	NE5RVERB	FINLEY	STS20B5E	NO	ENTERED	H 1	P	C	M	0001	0001	S
OWNSHIP	NEW TASK	NE5SNBWO	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0000	0000	G
OWNSHIP	NEW TASK	NE5TCNIM	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0003	0003	S
OWNSHIP	NEW TASK	NE5WKMSK	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0000	0000	G
OWNSHIP	NEW TASK	NE6HARME	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0002	0002	R
OWNSHIP	NEW TASK	NE6INTRP	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0001	0001	S
OWNSHIP	NEW TASK	NE6OPLRT	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0000	0000	C
OWNSHIP	NEW TASK	NE6RECRD	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0002	0002	S
OWNSHIP	NEW TASK	NE6STATS	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0000	0000	G
OWNSHIP	NEW TASK	NE6STREN	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0000	0000	G
OWNSHIP	NEW TASK	NE6VALID	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0001	0001	S
OWNSHIP	NEW TASK	NE6WHERE	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0000	0000	G
OWNSHIP	NEW TASK	NE6WMREF	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0000	0000	G
OWNSHIP	NEW TASK	TC4EINIT	FINLEY	STS20B5E	NO	ENTERED	H 1	P	C	M			S
OWNSHIP	NEW TASK	XDYTAMID	FINLEY	STS20B5E	YES	ENTERED	H 1	P	C	M	0002	0002	S
OWNSHIP	NEW TASK	XDYTCNID	FINLEY	STS20B5E	NO	ENTERED	H 1	P	C	M			
OWNSHIP	NEW TASK	XDZTCN1D	FINLEY	STS20B5E	NO	ENTERED	H 1	P	C	M	0000	0000	S
OWNSHIP	NEW TASK	XDZTCN2D	FINLEY	STS20B5E	NO	ENTERED	H 1	P	C	M	0000	0000	S
OWNSHIP	NEW TASK	XDZTCN3D	FINLEY	STS20B5E	NO	ENTERED	H 1	P	C	M	0000	0000	S
OWNSHIP	NEW TASK	XDZTCN4D	FINLEY	STS20B5E	NO	ENTERED	H 1	P	C	M	0000	0000	S
OWNSHIP	NEW TASK	TN2TCNIX	FINLEY	STS20B5E	NO	ENTERED	H 1	P	C	M			
OWNSHIP	NEW TASK	TNZTCN1D	FINLEY	STS20B5E	NO	ABSENT							
OWNSHIP	NEW TASK	TNZTCN2D	FINLEY	STS20B5E	NO	ABSENT							
OWNSHIP	NEW TASK	TNZTCN3D	FINLEY	STS20B5E	NO	ABSENT							
OWNSHIP	NEW TASK	TNZTCN4D	FINLEY	STS20B5E	NO	ABSENT							
OWNSHIP	NEW TASK	TC6GLB55	FINLEY	STS20B5E	NO	ENTERED	H 1	P	C	M			
OWNSHIP	NEW TASK	TC6GLB53	FINLEY	STS20B5E	NO	ENTERED	H 1	P	C	M			
OWNSHIP	NEW TASK	AMBNS2	FINLEY	STS20B5E	NO	ENTERED	H 2		C	M			
						MODULE TOTALS:	32	31	32	32	26	26	

Figure 6. Function Development Status Report

The Functional Milestone Status report depicts, on a module basis, the milestone matrix for a function hierarchy. This matrix represents the reportable status of the function at the time of audit. The matrix values are displayed for each milestone and each module. Figure 7 illustrates a Functional Milestone Status report.

20B5 SOFTWARE STATUS REPORT 06/13/83 06:18
MODULE STATUS
FOR
3.3.2.5.2 ASW ENVIRONMENT (SONOBUOY & SQS-56)

MODULE	WT	PROG.	1* DESIGN (12)	2* PDL ENTRY (1)	3,4,5* DESIGN W/T CODING ENTRY (6)	6* TEST PROC ENTRY (6)	7,8* CODE WALK THRU/ MOD TEST (9)	9* FUNCTION TEST (12)	MOD STAT CODE	MOD SCORE	PERCENT COMPLETE
NCZBSRCH	1	FINLEY	12	1	6	6	9	0-	8	34	100.0
NCZINTP2	1	FINLEY	12	1	6	6	9	0-	8	34	100.0
NE4EXEC	2	FINLEY	24	2	12	12	18	0	8	68	73.9
NE5AMBNS	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE5DOPRT	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE5KBSHD	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE5OPL56	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE5OPRLS	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE5RVERB	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE5SNBWO	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE5TCNIM	3	FINLEY	36	3	18	18	27	0-	8	102	100.0
NE5WKMSK	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE6HARME	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE6INTRP	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE6OPLRT	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE6RECRD	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE6STATS	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE6STREN	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE6VALID	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE6WHERE	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
NE6WMREF	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
TC4EINIT	2	FINLEY	24	2	12	12+	0	0-	6	50	73.5
XDYTAMID	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
XDYTCNID	3	FINLEY	36	3	18	18	0	0-	6	75	73.5
XDZTCN1D	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
XDZTCN2D	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
XDZTCN3D	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
XDZTCN4D	2	FINLEY	24	2	12	12	18	0-	8	68	100.0
TN2TCNIX	2	FINLEY	24	2	12	12+	0	0-	6	50	73.5
TNZTCN1D	2	FINLEY	0	0	0	0	0	0-	0	0	.0
TNZTCN2D	2	FINLEY	0	0	0	0	0	0-	0	0	.0
TNZTCN3D	2	FINLEY	0	0	0	0	0	0-	0	0	.0
TNZTCN4D	2	FINLEY	0	0	0	0	0	0-	0	0	.0
TC6GLB55	2	FINLEY	24	2	12	12	0	0-	6	50	73.5
TC6GLB53	2	FINLEY	24	2	12	12	0	0-	6	50	73.5
AMBNS2	2	FINLEY	24	0	12	12+	0	0-	6	48	70.6
TOTALS:	72		768	62	384	384	459	0		2057	83.2

* CORRESPONDS WITH MODULE STATUS CODE.

Figure 7. Functional Milestone Status Report

The Functional Hierarchy report reiterates the module breakout defined for the PDS and associates module completion codes with each module name. The module completion code represents an index of the current reportable status of the module. Figure 8 exemplifies the Functional Hierarchy report.

3.2.2.5.2 ASW ENVIRONMENT (SONOBUOY & AN/SQS-56 SONAR) FUNCTION SWWBS
(PDS SECTION: 3.3.2.5.2)

SEL TASK: NEW COMPUTER: OWNSHIP

NO2NEXEC	NEW TASK EXECUTIVE	*
NE4EXEC	ASW ENVIRONMENT (SONOBUOY & SQS-56) EXEC	5
NE5WKMSK	WAKE MASKING CONTROL	2
NE6WMREF	WAKE MASKING DATA FORMAT	5
NE6WHERE	WAKE SEGMENT POSITIONS	2
NE6STATS	SENSOR STATUS AND WAKE POSITION	5
NE6STREN	WAKE SEGMENT STRENGTH	4
NE5DOPRT	DOPPLER RATIO	5
NE5OPL56	SQS-56 OCEAN PROPAGATION LOSS CONTROL	4
NE6INTRP	PROPAGATION CONTROL	6
NCZINTP2	TWO-WAY INTERPOLATION	4
NE5SNBWO	SONOBUOY WASHOVER	5
NE5OPRLS	OCEAN PROPAGATION LOSS CONTROL	6
NE6OPLRT	OCEAN PROPAGATION LOSS RETRIEVAL	4
NCZINTP2	TWO-WAY INTERPOLATION	4
NE5AMBNS	AMBIENT NOISE	4
NE5KBSHD	KELP BED SHADOWING	0
NE5TCNIM	TCNI MANAGEMENT CONTROL	4
NE6HARME	HARMONIC FAMILY DATA	6
XDZDSCIO	DISC I/O QUEUING	5
NE6RECRD	RECORD NUMBER	4
XDZDSCIO	DISC I/O QUEUING	5
NE6VALID	VALIDITY CHECK	4
XDZDSCIO	DISC I/O QUEUING	5
NCZBSRCH	BINARY SEARCH	6
NE5RVERB	REVERBERATION	2

Figure 8. Functional Hierarchy Report

The *Functional Configuration Status report* illustrates the current integration state of each module of the function. The module is listed *under the appropriate control library designation*. Figure 9 provides a sample report.

```
            20B5 SOFTWARE_STATUS REPORT                  06/13/83  06:18
                        FOR
      3.3.2.5.2  ASW ENVIRONMENT (SONOBUOY & SQS-56)

                 CONFIGURATION STATUS

         MODULES            MODULES            MODULES            MODULES
          UNDER            CONFIGURED           UNDER              ABSENT
       DEVELOPMENT                            REVISION
       -----------        ----------         ----------          -------

                           NCZBSRCH
                           NCZINTP2           NCZINTP2
                           NE4EXEC            NE4EXEC
                           NE5AMBNS           NE5AMBNS
                           NE5DOPRT
                           NE5KBSHD
                           NE5OPL56           NE5OPL56
       NE5OPRLS
       NE5RVERB
                           NE5SNBWO
                           NE5TCNIM           NE5TCNIM
                           NE5WKMSK
                           NE6HARME           NE6HARME
                           NE6INTRP           NE6INTRP
                           NE6OPLRT           NE6OPLRT
                           NE6RECRD           NE6RECRD
                           NE6STATS
                           NE6STREN
                           NE6VALID           NE6VALID
                           NE6WHERE
                           NE6WMREF
       TC4EINIT
                           XDYTAMID           XDYTAMID
       XDYTCNID
       XDZTCN1D
       XDZTCN2D
       XDZTCN3D
       XDZTCN4D
       TN2TCNIX
                                                                 TNZTCN1D
                                                                 TNZTCN2D
                                                                 TNZTCN3D
                                                                 TNZTCN4D
       TC6GLB55
       TC6GLB53
       AMBNS2
```

Figure 9. Function Configuration Status Report

The Milestone Summary report lists, by function, the actual and scheduled budget values and associated variances from subsequently scheduled audit dates. This report provides the manager with a snapshot of the current software design status in terms of the milestones established. Figure 10 presents a sample Milestone Summary report for Device 20B5.

20B5 SOFTWARE STATUS REPORT 01/11/83 07:33

FUNCTION STATUS

SECTION	DESCRIPTION	CHARGE NO.	TOTAL POINTS	POINTS TO DATE	PER CENT COMPLETE	01/01/83 BUDGET PER CENT	02/01/83 BUDGET PER CENT	DEFICIT/ SURPLUS POINTS	POINTS TO NEXT MILE-STONE
3.3.2.1.1	VEHICLE DYNAMICS	125611	11784	1652	92.6	100.0	100.0	-132	132
3.3.2.1.2	OWNSHIP ENVIRON-MENT	125611	692	563	81.4	94.0	100.0	-87	129
3.3.2.1.3	MK-15 CIWS	125621	432	156	36.1	100.0	100.0	-276	276
3.3.2.1.4	MK-12 AIMS IFF/SIF	125611	160	88	55.0	100.0	100.0	-72	72
3.3.2.1.6	NAVIGATION SYSTEM	125611	92	92	100.0	92.0	100.0	8	0
3.3.2.1.7	HAND-HELD UNIT	125611	92	92	100.0	100.0	100.0	0	0
3.3.2.1.8	LINK-14	125611	500	338	67.6	100.0	100.0	-162	162
3.3.2.1.11	MK-75 GUN	125621	228	100	43.9	100.0	100.0	-128	128
3.3.2.1.13	LAMPS NON-ACOUSTIC SENSORS	125641	352	172	48.9	60.0	72.0	-38	81
3.3.2.1.14	OWNSHIP WEAPONS DAMAGE ASSESSMENT	125611	636	414	65.1	33.0	66.0	205	5
3.3.2.1.15	HARPOON MISSILE	125641	1498	1075	71.8	62.0	73.0	147	18
3.3.2.1.16	STANDARD MISSILE	125641	1124	782	69.6	51.0	66.0	209	0
3.3.2.1.17	MK-46 TORPEDO	125641	1192	976	81.9	60.0	72.0	261	0
3.3.2.1.18	COMMON MODULES	125641	408	360	88.2	70.0	79.0	75	0
3.3.2.2.1	PASSIVE ACOUSTICS (PASS. EFFECTS)	125641	9328	6876	73.7	81.0	90.0	-679	1519
3.3.2.3.1	ASW ENVIRONMENT (SQS-56)	125611	296	272	91.9	100.0	100.0	-24	24
3.3.2.3.2	ACTIVE ACOUSTICS (SQS-56)	125641	636	534	84.0	78.0	93.0	38	57
3.3.2.3.3	PASSIVE ACOUSTICS (AN/SQS-56 TASK)	125641	1814	1424	78.5	81.0	90.0	-45	208
3.3.2.3.4	PASSIVE BUFFER	125641	46	0	.0	92.0	100.0	-42	46
3.3.2.3.5	ASW SONAR I/O	125611	204	0	.0	92.0	100.0	-187	204
3.3.2.4.2	ACTIVE ACOUSTICS (SONOBUOY)	125641	908	650	71.6	74.0	93.0	-21	194
3.3.2.4.3	LAMPS SONOBUOY ACOUSTICS	125641	1872	1315	70.2	100.0	100.0	-557	557
3.3.2.4.4	LAMPS SONOBUOY ACOUSTICS I/O	125611	204	0	.0	92.0	100.0	-187	204
3.3.2.5.1	SONOBUOY AND SQS-56 RELATIVES	125611	1158	1110	95.9	95.0	100.0	10	48
3.3.2.5.2	ASW ENVIRONMENT (SONO & SQS-56)	125611	1724	1185	68.7	100.0	100.0	-539	539
3.3.2.6.1	OWNSHIP DISC I/O	125611	468	399	85.3	94.0	100.0	-40	69
3.3.2.7.1	OWNSHIP MONITOR	125611	204	204	100.0	93.0	100.0	15	0
3.3.2.7.2	REAL-TIME EXECUTIVES	125611	476	476	100.0	93.0	100.0	34	0
3.3.2.8.1	DDL I/O	125611	432	144	33.3	93.0	100.0	-253	288
3.3.2.8.2	AN/SQS-56 PING PROCESSING	125641	1978	1252	63.3	80.0	92.0	-330	567
3.3.2.8.3	INTERBUS LINK HANDLER	125611	1104	1104	100.0	89.0	100.0	122	0
3.3.3.2.3	DISC FILE TRANSFER	125611	160	106	66.2	93.0	100.0	-42	54
TOTALS:			32202	23911	74.3	82.7	91.0	-2718	5384

Figure 10. Milestone Summary Report

The Cost Performance Summary report provides a tally of the milestone matrix values in terms of the assigned work order numbers and displays the variances associated with each tally. The variances identify the difference in expected and actual design completion for a work order. In effect, a work order represents the allocated budget for a particular design effort, while the variance represents the additional effort required or surplus effort expended in meeting the budgeted milestone. Figure 11 illustrates a sample Cost Performance report.

20B5 SOFTWARE STATUS REPORT 01/11/83 07:33

POINT NUMBER STATUS

CHARGE NO.	TOTAL POINTS	POINTS TO DATE	PER CENT COMPLETE	01/01/83 BUDGET PER CENT	01/01/83 EXPECTED POINTS	02/01/83 BUDGET PER CENT	POINTS REQUIRED	STATUS
125511	0	0	.0	.0	0	.0	0	INACTIVE
125521	0	0	.0	.0	0	.0	0	INACTIVE
125522	0	0	.0	.0	0	.0	0	INACTIVE
125611	10386	8239	79.3	92.1	9570	97.9	1331	TROUBLE
125621	660	256	38.8	100.0	660	100.0	404	TROUBLE
125622	0	0	.0	.0	0	.0	0	INACTIVE
125632	0	0	.0	.0	0	.0	0	INACTIVE
125634	0	0	.0	.0	0	.0	0	INACTIVE
125641	21156	15415	72.9	77.5	16399	87.3	983	TROUBLE
125711	0	0	.0	.0	0	.0	0	INACTIVE
SURPLUS POINTS:							0	
TOTALS:	32202	23911	74.3	82.7	26629	91.0	2718	TROUBLE

Figure 11. Cost Performance Summary Report

In addition to the standard reports generated by the AAIs program, the manager can direct the program to accumulate milestone statistics over auditing periods to graphically construct progress plots. This accumulation is accomplished by the use of the history files described in the previous section. These progress plots have been effectively presented to the contract sponsor by the 20B5 management during periodic Program Progress Reviews (PPRs). One such plot is presented in Figure 12.

AAIS Inputs: Control Data

Software managers must define seven (7) input data sets to drive the AAIS program. These data sets are as follows:

(1) Baseline starting dates

(2) Software work order numbers

(3) Plotting options

(4) Function hierarchies

(5) Function milestone weights

(6) Function milestone dates

(7) Module weights.

Data input sets (1), (2) and (4) apply to the entire software effort and are specified for each function hierarchy defined. The remaining input data sets are unique to each function hierarchy.

As shown previously in Figure 2, each function hierarchy must be established when the design is initiated. The hierarchy is subsequently maintained in a file on the computer storage media. Note that module values are specified according to the level of complexity. It should be recognized that the status reports will incorporate any design changes as performed on the appropriate data files. However, the manager must understand that this flexibility allows for a rolling baseline for audits. The particular function or functions audited are specified when performing an audit.

AAIS Processing

AAIS processing is given by the functional flow diagram presented in Figure 13. For Device 20B5, a full audit involving approximately 800 modules requires about 45 minutes of computer execution and report generation time. The entire process does not interfere with software development activities.

Organizing the control inputs presented in the previous section into files on the computer storage media requires approximately one (1) man-month of effort. These files are subsequently updated by individual designers as software designs are expanded or modified. An additional man-month of effort is required to produce the AAIS program in the desired high level language.

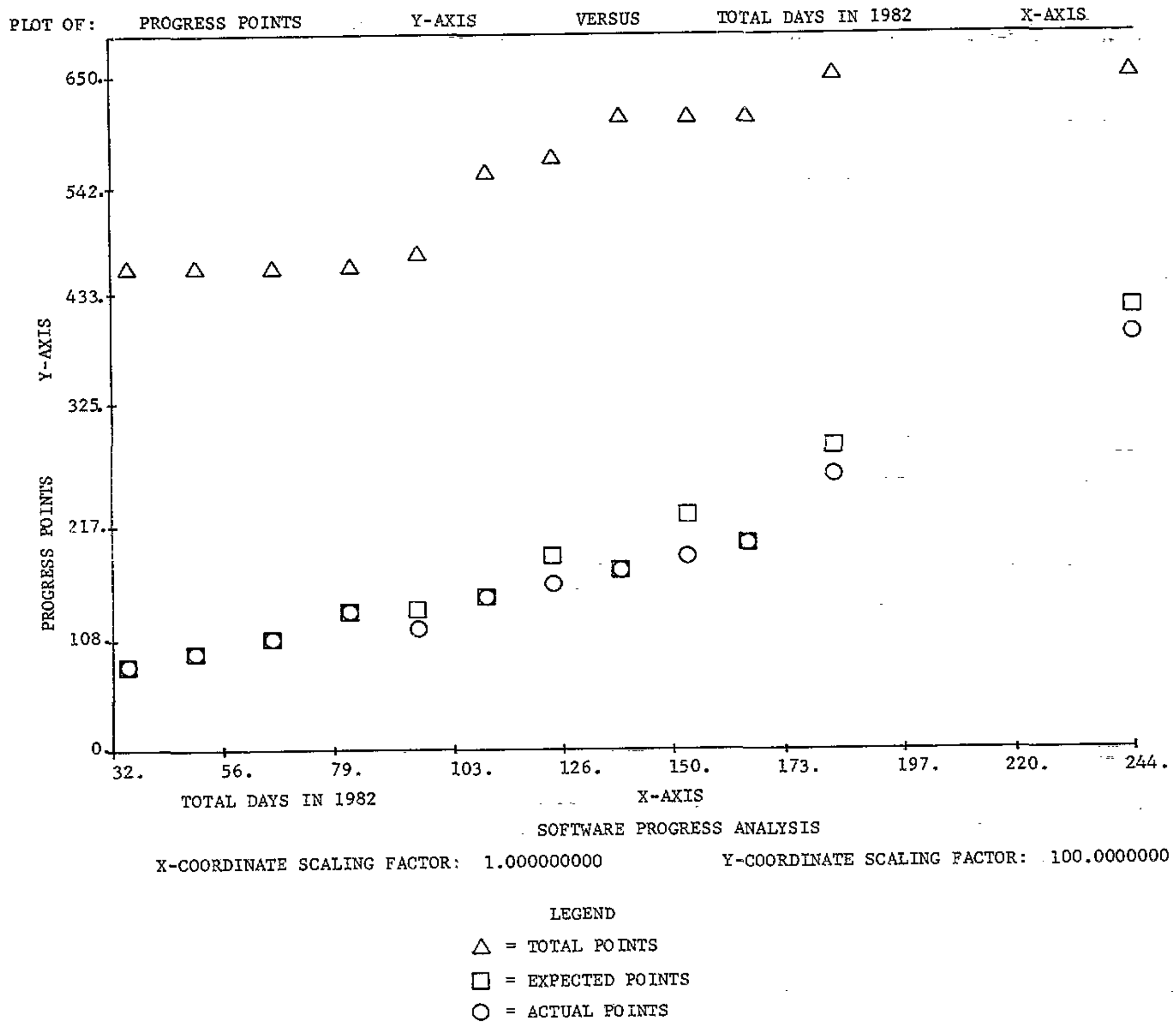

Figure 12. Historical Plot of Software Progress

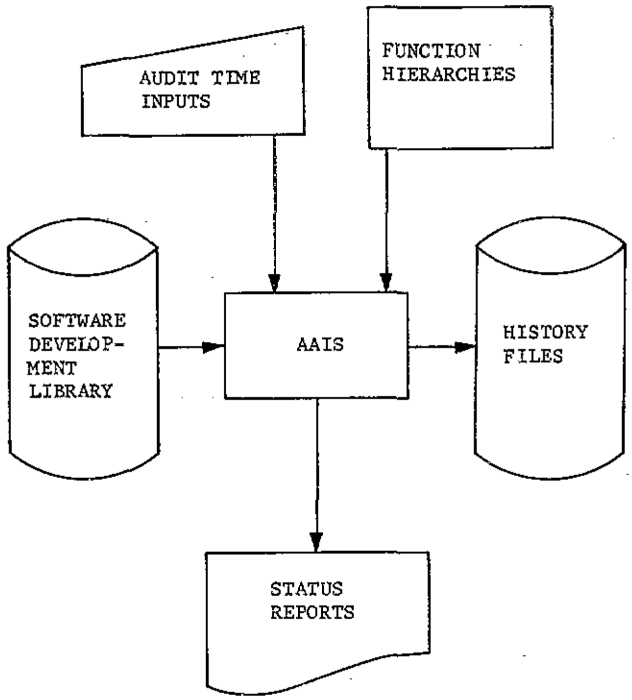

Figure 13. The AAIS Program Functional Flow

Summary

The AAIS procedure performs audits of the 20B5 software development library upon demand. The audit information provides managers with an improved capability to control software development activities and to improve projections for completion. The system is adaptable to other projects if similar organization and procedures are applied.

ABOUT THE AUTHOR

MR. GARY A. BROWN is a Senior Engineering Analyst in the Training and Simulation Department of the Electronic Engineering Division of AAI Corporation. He is currently responsible for the design, development, and implementation of microprocessor based software and diagnostic programs for the Device 20B5 Pierside Combat System Team Trainer. He is the principal designer and implementer of the AAIS and other 20B5 unique software tools. He holds a Bachelor's of Science degree from Rensselaer Polytechnic Institute in Computer and Systems Engineering. Gary has formerly developed system software for Device 15F12, a Radar Landmass Team Trainer.