Screening Task #6

Title: HEAO Attitude Ground Support System

Objective: The HEAO Attitude Ground Support System performs the spacecraft control functions of initial attitude acquisition, attitude reference updating and calculation of attitude detector misalignments and satisfies the definitive requirements of attitude solutions output at 1.28 second intervals. The HEAO-A mission in 1977 lasts 6 - 9 months and the HEAO-C support will extend for the same period in 1979.

Environment:
Target Computer - S/360
Development - S/360
Language - Structured FORTRAN using a precompiler
Level of Effort - 10 1/2 years
Program Size - 300 modules

Methodology:
Top down design, development and testing by subsystem.
Specification using pseudo-coding
Design Reviews
Some Card Reading
Master Library change control

PERTINENT INFORMATION: Information on the reasons for changing the source coding has been kept on a library modification request which was developed specifically for HEAO-A. Approximately 100 forms were collected.

Screening Task #7

Title: LANDSAT NAVPAK

Objective: The LANDSAT NAVPAK system is an orbit and attitude determination system which extracts landmarks from visible images taken by LANDSAT spacecraft, and then uses the extracted data as the primary data type for orbit and attitude determination process. The system consists of three major subsystems: Data Management, Preprocessor and Landmark Extraction/Estimation. The Data Management subsystem is intended to facilitate the manipulation of various data files by providing the users with such functions as delete, compress, insert, modify and the like for linked list files. The Preprocessor subsystem consists of three major modules: Full Scene Reconstruction, Search Area Extraction, and Momentum Wheel Data Reduction. The purpose of the first and the second modules is to prepare the raw image data for landmark extraction. Initially reading a set of four data tapes containing interleaved multiple band data and the associated annotation data, and then creating a single band file, these modules ultimately create a file containing image data from which a landmark is searched and extracted, based on the orbit in the scene annotation. The Momentum Wheel Data Reduction Module, on the other hand, prepares the raw wheel data, which contains the information about the history of the spacecraft attitude dynamics, for the use in propagating the attitude from one scene to another.

The third subsystem is the Landmark Extraction/Estimation Subsystem. After the image and wheel data have been preprocessed a user of the NAVPAK system is ready for the execution of this subsystem. Initially, one or a series of landmarks are extracted either automatically or manually. This

data is then fed into a sequentiol filter module to compute the corrections to the initial orbit and attitude states. This process of extraction/estimation is repeated until an input convergence criterian is satisfied.

Environment: Target Computer - PDP 11/70
Development Computer - PDP 11/70
Language - FORTRAN
Level of Effort - 2 or 3 programmer/analysts for 9-12 months
Program Size - 10,000 source statements

Methodology: None

PERTINENT INFORMATION; Forms Collected:

1. Resource Summary
2. Component Status Report
3. General Project Summary
4. Run Analysis
5. Component Summary

Screening Task #8

Title: MMS OBC Software

Objective: The objective of this project is to develop orbit and attitude software systems to be used on NASA's standard on-board computer (OBC) for the Solar Maximum Mission of the Multimission Modular Spacecraft series (SMM/MMS). The orbit software system, which was completed in April 1977, performs spacecraft ephemeris computation based on a Fourier-power data compression technique along with a 4-point Hermite interpolator. In the attitude area, however, the specific on-board algorithms have not been selected for implementation, although the candidate functions have been identified and implemented on a PDP 11/70 in FORTRAN. One portion of the attitude system will be the determination based on star positions using the direct match technique with a small star catalogue. Another portion of the attitude system will be in the area of attitude control. The specifications for this portion has not been developed.

The method of OBC software development and initial testing is slightly different from more commonly practiced methods. Because of the memory limitations and the lack of suitable peripheral devices on the OBC, the assembler resides on another computer, namely an XDS SIGMA 5, as a cross-assembler. In fact, along with the cross-assembler, there is a loader as well as a simulator, which resolves external references and then simulates the execution of the OCB code on a SIGMA 5.

The final testing of all on-board software will be performed on a bread-board OBC by an independent group of programmers and analysts.

Environment: Target Computer - NASA Standard Spacecraft Computer (NSSC-1)
Development Computer - SIGMA 5
Language - OBC assembly language
Level of Effort - 1 - 2 programmers for 3 months for the orbit computation software
Program Size - 2000 lines of source statements including comments for the orbit computation

Methodology: None

PERTINENT INFORMATION: Forms collected:

1. Resource Summary
2. Component Status
3. Run Analysis

Screening Task #9

Title: GESS-11/70

Objective: The objective of the Graphics Executive Support System for PDP-11/70 (GESS-11/70) is to provide a mechanism with which any applications system can be developed with extreme interavtice graphics capabilities. These capabilities include parameter and array displays and vector plotting onto a display device called model 70CS image display unit manufactured by International Imaging Systems. To utilize this graphics facility, a user must first learn the GESS language and code the desired displays in that language. The display source code is then assembled, and the executable code is generated by linking with the applications program and other system routines.

Currently all operational attitude determination systems are developed using GESS on the S/360's GESS-360 to generate parameter, array, and plot displays. In addition, the execution of such a system can be made under the control of a user through GESS. With the development of GESS-11/70, future attitude determination systems can then be developed on the PDP 11/70, since the GESS source code is totally transportable from one machine to the other.

As of April, 1977, Version 1 of GESS-11/70 has been completed and released to users. This version has the capability of displaying parameters, arrays and plots, however, the interactive facility is not included. Version 2 which is expected to be completed in June 1977 will have most of the capabilities which exist on the 360 version.

Environment: Target Computer - PDP 11/70
Development Computer - PDP 11/70
Language - FORTRAN, MACRO-11 assembly language
Level of Effort - 2 programmers for 18 months
Program size - 20,000

Methodology: None

PERTINENT INFORMATION: Forms Collected:

1. Resource Summary
2. Component Status Report
3. Change Report
4. Run Analysis
5. Component Summary

Controlled Experiments

Controlled Experiment 1

The initial controlled experiment was designed to be conducted on the PDP 11/70. Instead of one large software development task being used for this part of the project, several smaller tasks were created so that the design of the overall controlled experiment could best be satisfied.

Descriptions of the five tasks making up the first experiment follow. The organization and methodology to be used in this experiment was discussed in Section VI of this report.

CONTROL TASK #1 (SUBTASK #1)

Title: Human Resources Scheduling

Objective: The purpose of this program is to service the requirement of allocating manpower from a specific organization (Division) to a specific unique project when there is a specified upper and lower bound on manpower assignments to each of these unique projects.

This program will be interactive data base system by which one can enter, update and report human resource schedules. Capabilities will include the ability to enter overall constraints by projects, assign individuals to projects, and report on deviations from given constraints.

Environment: Target Computer - PDP-11
Development Computer - PDP-11
Language - FORTRAN
Level of Effort - 3 Man Months
Program Size - Undetermined

Methodology: See Section VI

Control Task #1 (Subtask #2)

Title: FORTRAN Source and Analyzer

Objective: This program is designed to be a static code analyzer for FORTRAN programs providing basic statistics typifying modules on the PDP-11. It is anticipated that such a capability will be implemented by developing a preprocessor, which will extract pertinent information from the source code and store it on some intermediate data set, and a post-processor which will use the data base created by the preprocessor. This is identical to the design of FACES.

During the first control experiment, the defined objectives will be limited in scope due to time constraints. Such information as type of module (I/O, control, computational, etc.) and level of complexity (derived by some predefined characteristics such as number of controls, amount of I/O, etc.) will be part of this first version.

Certainly, this program can be made as small or as complex as one desires.

Environment: Target Computer - PDP-11
Development Computer - PDP-11
Language - FORTRAN
Level of Effort - 2 Man Months
Program Size - Undetermined

Methodology: See Section VI

Control Task #1 (Subtask #3)

Title: PDP-11 NAMELIST Processor

Objective: In order to support attempts to make code developments on S/360 compatable with the PDP-11, the capability of using NAMELIST input and output is to be added to the PDP-11. This program will be a subprogram of an existing FORTRAN precompiler that will recognize NAMELIST specific syntax and will then generate equivalent FORTRAN non-NAMELIST type of I/O statements. The following is a list of statement types which must be recognized by this precompiler:

1. NAMELIST/NAME/variables
2. READ (m, NAME)
3. WRITE (m, NAME)

Environment: Target Computer - PDP-11
Development Computer - PDP-11
Language - FORTRAN
Level of Effort - 2 Man Months
Program Size - Undetermined

Methodology: See Section VI

Control Task #1 (Subtask #4)

Title: Contractor Financial Report Program

Objective: This program is required to generate a report of direct labor hours charged by the prime contractor. This information is combined with other pertinent data base data to generate reports and graphics tabular displays of such areas of concern as:

1. Work Authorization Document (WAD) category numbers associated with a specific task.
2. Off-site hours charged for a given time period.
3. On-site hours charged for a given time period.

Items 2 and 3 would normally be reported for the current month and for the cumulative charges for the current fiscal year.

Environment: Target Computer - PDP-11
Development Computer - PDP-11
Language - FORTRAN
Level of Effort - 1 to 3 Man Months
Program Size - Undetermined

Methodology: See Section VI

Control Task #1 (Subtask #5)

Title: IUE Control Monitor

Objective: This program is a real time, interactive attitude support subsystem designed to display various parameters pertinent to an Attitude reorientation. As telemetry data is made available to this program various meaningful pieces of information are computed, then automatically displayed on an interactive graphics device in tabular as well as vector type format. Upon completing a sequence of displays, more telemetry information is retrieved and the process continues until the operator terminates the sequencing.

One of the major requirements of this subsystem is that it must be "transportable" to the S/360 (from the PDP 11/70) and then run in place of an existing monitor system on that machine.

Several other pertinent pieces of information will be extracted from this subsystem development:

1. Can the Graphics Executive Support System (GESS) being developed on the PDP-11/70 (the system already is a standard utility on the S/360) support development of software for attitude systems in the same way it supports work on the S/360?
2. Can software targeted for the S/360 be realistically developed on the PDP-11, then transported to the S/360?
3. What relative design and development variations are apparent between systems developed on S/360 and the same system on the PDP-11?

Environment: Target Computer - S/360
Development Computer - PDP-11
Language - FORTRAN
Level of Effort - 6 Man Months
Program Size - 30 Modules

Methodology - See Section VI

Controlled Experiment 2

The second controlled experiment makes use of existing NASA/GSFC developments. However, instead of simply monitoring the progress of these developments, the projects will be impacted with specific methodologies - thus differing from the screening experiments. The projects to be developed as controlled experiment 2 are as follows:

Control Task #2 (Subtask #1)

Title: HCMM Attitude Determination System (AEM-A)

Objective: The Heat Capacity Mapping Mission (HCMM) - (also called the Atmospheric Explorer Mission - AEM). Attitude Determination System computes three-axis attitude from sun sensor and magnetometer spacecraft data and provides an estimate of sensor biases. The interactive system will be used throughout the one year mission.

Environment: Target Computer S/360
Development S/360
Language FORTRAN
Level of Effort 6 Man Years
Program Size 300 modules
30,000 lines of code

Methodology: Functional specifications
Strict requirements for design products

PERTINENT INFORMATION: The following forms will be used for information extraction:

1. General Project Summary
2. Component Summary
3. Component Status
4. Change Reports
5. Resource Summary
6. Run Analysis

Control Task #2 (Subtask #2)

Title: SEASAT - A Definitive Attitude Determination System

Objective: The SEASAT - A Definitive Attitude Determination System provides three-axis attitude results (pitch, roll, yaw) for all times when data has been received from the spacecraft. Yaw is interpolated when Sun sensor data is absent in telemetered data and attitude results are interpolated when data gaps occur.

Environment: Target Computer - S/360
Development - S/360
Language - FORTRAN
Level of Effort - 4 1/2 years
Program Size - 150 modules
10,000 lines of coding

Methodology: Functional Specifications
Top down design, development and testing by subsystem
Specifications with pseudo coding (POL)
Structured Code
Design reviews
Some code reading
Master library change control

PERTINENT INFORMATION: Gathering of information will depend on the usage of the following forms:

1. General Project Summary
2. Component Summary
3. Component Status
4. Change Reports
5. Resource Summary
6. Run Analysis

Control Task #2 (Subtask #3)

Title: ISEE-C Attitude Determination

Objective: This task is similar to the other attitude determination programs being used as screening tasks. The primary objective of the program is to extract sensor measurement information from telemetry data, then compute the vehicle orientation from this data.

The ISEE-C spacecraft will be launched in July of 1978, and this system must compute the vehicle orientation during the long transfer trajectory as well as during the final mission phase. The mission phase of the project is to be at the Earth-Sun Libration point.

The attitude which is determined from the telemetry information is merged with pertinent experimental data before being transmitted to the requesting project experimentors.

Environment: Target Computer - S/360
Developmental Computer - S/360
Language - FORTRAN
Level of Effort - 4 1/2 Man Years
Program Size - 250 modules
20,000 lines of code

Methodology: PDL
Structured Programming
Walk-throughs
Code Reading
Top Down

PERTINENT INFORMATION: Information is to be collected through the usage of the following forms:

1. General Project Summary
2. Component Summary
3. Component Status
4. Change Reports
5. Resource Summary
6. Run Analysis

Control Task #2 (Subtask #4)

Title: SMM Attitude Determination

Objective: Like the other attitude determination systems which are part of the screening tests, the Solar Maximum Mission (SMM) support software is the ground based package required to determine the vehicle orientation from telemetry information.

This support requirement begins at the time of launch (Mid 1979) and lasts through the life of the mission (1 year). This project is the first of a series known as the Modular Mission Spacecraft (MMS) series.

The resulting attitudes are merged with pertinent experimental data before being transmitted to the various SMM scientific investigators.

Environment: Target Computer S/360
Development S/360
Language FORTRAN
Level of Effort 6 Man Years
Program Size 350 Modules
25,000 lines of code

Methodology: Not yet determined

PERTINENT INFORMATION: Information is to be collected through the usage of the following forms:

1. General Project Summary
2. Component Status
3. Component Summary
4. Change Reports
5. Resource Summary
6. Run Analysis

Bibliography

[AMOR, 73] Amory, W., J. A. Clapp, A Software Error Classification Methodology, MTR 2648, Vol. VII, The Mitre Corporation, June, 1973.

[BAKE, 75] Baker, F. T., Structured Programming in a Production Programming Environment. International Conference on Reliable Software, Los Angeles, April 1975, (Sigplan Notices 10, 6, June 1, 1975, pp. 172-185).

[BASI, 75] Basili, V. R., A. J. Turner, Iterative enhancement: a practical technique for software development, IEEE Transactions on Software Engineering, 1, No. 4, December, 1975, pp. 390-396.

[BAUM, 63] Baumgartner, J. S., Project Management, Richard D. Irwin, Inc., 1963.

[BOEH, 75] Boehm, B. W., R. K, McClean, D. B. Urfrig, Some Experience Aids to the Design of Large Scale Reliable Software, IEEE Transactions on Software Engineering 1, No. 1, March, 1975, pp. 125-133.

[BROW, 73] Brown, J. R., A. J. De Salvia, D. E. Heine, J. G. Purdy, Automated software assurance, Program Test Methods, Prentice Hall, 1973, pp. 181-203.

[CAMP, 66] Campbell, D. T., J. C. Stanley, Experimental and quasi-experimental designs for research, Chicago, Rand McNally Publishing Co., 1966.

[CLAP, 74] Class, J. A., J. E. Sullivan, Automated Monitoring of Software Quality, 1974 AFIPS National Computer Conference Proceedings, pp. 337-342.

[DAHL, 72] Dahl, O., E. Dijkstra, C. A. R. Hoare, Structured Programming, New York, Academic Press, 1972.

[DOMO, 75] Domonic User Guide, Advanced Technology Group, Data Processing Center, Texas A&M University, 1975.

[ENDR, 75] Endres, A. B., An Analysis of Errors and Their Causes in System Programs, IEEE Transactions on Software Engineering 1, No. 2, June, 1975, pp. 140-149.

[FAGA, 74] Fagan, M. E., Design and Code Inspections and Process Control in the Development of Programs, IBM Technical Report TR 21.572, 17 December 1974.

[GANN, 75] Gannon, J. D., J. J. Horning, Language Design for Programming Reliability, IEEE Transactions on Software Engineering 1, No. 2, June, 1975, pp. 179-191.

[HAMI, 76] Hamilton, M., S. Zeldin, Higher Order Software - A Methodology for Defining Software, IEEE Transactions on Software Engineering 2, No. 1, March, 1976, pp. 9-32.

[HELD, 75] Held, G., M. Stonebraker, E. Wong, INGRES - A relational data base system, National Computer Conference, 1975, pp. 409-416.

[HELL, 72] Hellerman, L., A Measure of Computational Work, IEEE Transactions on Computers 21, No. 5, 1972, pp. 439-446.

[LING, 77] Linger, R. C., H. D. Mills, Structured programming theory and practice, Addison Wesley, 1977, (to be published).

[LITE, 76] Litecky, C. R., G. B. Davis., A Study of Errors, Error Proneness, and Error Diagnosis in COBOL, Communications of the ACM 19, No. 1 January, 1976, pp. 33-37.

[MILL, 72] Mills, H. D., Mathematical Foundations for Structured Programming, FSC 72-6012, IBM Corporation, Baithersburg, Maryland 20760, February, 1972.

[MILL, 73] Mills, H. D., The Complexity of Programs, in Program Test Methods, edited by W. C. Hetzel, Englewood Cliffs, Prentice-Hall, Inc., 1973, pp. 225-238.

[MILL, 75] Miller, E. F., Jr., Methodology for Comprehensive Software Testing, Interim Report, Rome Air Development Center, RADC-TR-75-161, June, 1975, AD# A013111.

[MYER, 75] Myers, G., Software Reliability Through Composite Design, New York, Mason Charter, 1975.

[RAMA, 74] Ramamoorthy, C. V., S. F. Ho., FORTRAN automatic code evaluation system (FACES), part I. Memorandum No. ERL-M-466, Electronics Research Laboratory, University of California, Berkeley, August, 1974.

[RAMA, 75] Ramamoorthy, C. V., S. B. F. Ho, Testing Large Software with Automated Software Evaluation Systems, IEEE Transations on Software 1, No. 1, March, 1975, pp. 46-58.

[REIF, 75] Reifer, D. J., "Automated Aids for Reliable Software," An Invited Tutorial at the 1975 International Conference on Reliable Software, 21-23 April 1975.

[RUBE, 75] Rubey, Raymond, J., Quantitative Aspects of Software Validation, Proceedings of the 1975 International Conference on Reliable Software, April, 1975, pp. 246-251.

[SHOO, 75] Shooman, M. L., M. I. Bolsky, "Types, Distribution, and Test and Correction Times for Programming Errors," Proceedings 1975 Conference on Reliable Software, April 21-23, 1975, pp. 347-362.

[SIEG, 56] Siegol, L. S., Nonparametric Statistics for the Behavioral Sciences, New York, McGraw-Hill Book Co., 1956.

[STAY, 76] Stay, J. F., HIPO and integrated program design, IBM Systems Journal 15, No. 2, 1976, pp. 143-154.

[SULL, 73] Sullivan, J. E., Measuring the complexity of computer software, Mitre Corp. Report MTR-2648, Vol. V, June, 1973.

[TAUS, 76] Tausworthe, R. C., Standard Development of Computer Software, Part 1 Methods, Jet Propulsion Lab, Calif. Inst. of Technology, Pasadena, Calif., July, 1976.

[TEIC, 77] Teichroew, D., E. Hershy, PSL/PSA: A Computer-aided Technique for Structured Documentation and Analysis of Information Processing Systems, IEEE Tranactions Software Engineering 3, No. 1, January, 1977, pp. 41-48.

[THAY, 76] Thayer, T. et.al. Software reliability study, TRW Defense and Space Systems Group, National Technical Information Services AD-A030-798, August, 1976.

[VANE, 70] Van Emden, M. H., The hierarchical decomposition of complexity, Machine Intelligence 5, 1970, pp. 361-380.

[VANL, 76] Van Leer, P., Top-down development using a program design language, IBM Systems Journal 15, No. 2, 1976, pp. 155-170.

[WALS, 77] Walston, C. E., C. P. Felix, A method of programming measurement and estimation, IBM Systems Journal, No. 1, 1977, pp. 54-73.

[WOLV, 72] Wolverton, R. W., The Cost of Developing Large Scale Software, TRW Software Series TRW-SS-73-01, March, 1972.

[YOUN, 70] Youngs, E., Error Proneness in Programming, Ph.D. Dissertation, University of North Carolina, Chapel Hill, 1970.