Symposium on

Machine Processing of

Remotely Sensed Data

and

Soil Information Systems

and

Remote Sensing and Soil Survey

June 3-6, 1980

Proceedings

The Laboratory for Applications of Remote Sensing

Purdue University
West Lafayette
Indiana 47907 USA

IEEE Catalog No.
80CH1533-9 MPRSD

A TRANSPORTABLE EXECUTIVE SYSTEM FOR USE WITH REMOTE SENSING APPLICATIONS SOFTWARE

PETER VAN WIE AND DAVID FISCHEL AND DAVID HOWELL

NASA/Goddard Space Flight Center

I. ABSTRACT

This report describes a transportable software executive system under development at the Goddard Space Flight Center. This executive will facilitate user access to data and programs used in remote sensing applications and will provide the necessary system control, bookkeeping, operating system interface, man-machine communications and other services needed to make the computer an effective and economical tool for the remote sensing investigator.

II. INTRODUCTION

The Goddard Space Flight Center is currently utilizing and developing a number of interactive remote sensing applications computer systems for analysis of data collected by Landsat, SMS/GOES, Nimbus and other spacecraft. Table 1 gives a summary of these systems and the software packages installed on them. The primary function of these systems is image processing using color television image analysis terminals. These systems are used by investigators in agriculture, hydrology, meteorology, geology, land use and other disciplines. These applications systems have several common characteristics:

o They process digital image remote sensing data.

o They have interactive image analysis and display capabilities.

o They support multiple users.

o They are installed on minicomputers such as the DEC PDP 11/70 and the VAX 11/780.

o They require a collection of application programs to ingest, preprocess, analyze and display data.

Typical system users are experts in their discipline areas but are generally not computer systems specialists. The multi-system environment presents a problem for investigators who must learn to operate the variety of software packages and operating systems. Frequently a data analysis task cannot be fully supported on a single system and the user must transfer partially processed data between systems to accomplish his objectives. This environment is difficult for investigators who would prefer to concentrate on their discipline area and not get bogged down in the details of the computer. The challenge then is one of making the powerful capabilities of the computer available to these users without requiring them to go through extensive training in computer operation, programming, operating systems and specialized hardware.

A second major consideration is economics. Experience has shown that a large portion of the cost of Applications software systems is associated with control, communications, data management and other factors not directly involving applications programming. The availability of a transportable data analysis executive to which specialized applications modules are attached will reduce subsequent development costs of future systems.

Given the common characteristics of the remote sensing applications systems, the economic considerations and the need to simplify and standardize the user environment it was determined that a single software executive should be developed to service each of these systems and be amendable to future applications within NASA and outside the agency. The initial use for this executive will be with the Landsat-D Assessment System[1] (LAS).

III. APPROACH

A design team was formed to establish the system requirements and development methodology. The ground rules given to the team were that the executive (now called the Standard Applications Executive or SAE) was to be

(1) Transportable and

(2) VICAR compatible.

A. TRANSPORTABILITY

A high degree of portability is a key requirement since the SAE must operate on several different computers, e.g., the PDP 11/70 and the VAX 11/780. The issue of portability was studied and a report[2] written defining design factors favoring portability. Portability is viewed in a relative sense because absolute portability is not achievable. The following definition of portability is often quoted:

"Portability is a measure of the ease with which a program can be transferred from one environment to another: If the effort required to move the program is much less than that required to implement it initially, then we say that it is highly portable."[6]

Some specific recommendations made in the report are:

- o A top down software engineering methodology be used.

- o Functional capabilities of the SAE should be modularized so that those capabilities which cannot be implemented on a given target machine can be deleted with minimum effort. For example, scheduling of asynchronous tasks may not be supported on all target machines.

- o The interface of SAE to the host operating system should be narrow. Specifically, each operating system service call should occur only once in the executive and all operating system interfaces, including I/O, should be in executive modules and not accessed directly by Applications programs.

B. VICAR COMPATIBILITY

The VICAR (Video Image Communications and Retrieval) system, developed at the Jet Propulsion Laboratories, has demonstrated a highly successful structure for a modular, expansible image processing system. It is in wide use and contains a large library of applications programs. Due to the high degree of user familiarity and the availability of applications programs it was desired to make SAE VICAR compatible. This compatibility with VICAR encompasses

- o Use of the VICAR command language as a subset of the SAE command language.

- o Use of similar system structures for interfacing applications programs, such as parameter passing from the executive to applications programs.

- o Use of common data formats so that the output of one applications program can be input to the next and "scenarios" of sequential applications programs can be used to perform entire data analysis projects.

The degree of compatibility achievable between the SAE and VICAR is limited by some basic structural and environmental differences. VICAR is an IBM 360 single user batch oriented system while SAE is oriented toward minicomputer multiuser interactive systems.

C. METHODOLOGY

Software engineering techniques are being used in the development of SAE. A document was prepared[3] synthesizing the software engineering concepts presented in numerous other publications. Development of a software system is divided into nine life cycle phases

- o Requirements definition
- o Functional design
- o Detailed design
- o Test plan definition
- o Configuration control
- o Implementation
- o Testing
- o Operations
- o Maintenance

At this time (March 1980) the requirements definition phase is completed and an initial functional design has been prepared. This initial design lays out the basic system software elements and data structures. The remainder of this report details the results of the requirements[4] and functional design[5] phases.

IV. REQUIREMENTS

A. RESOURCE MANAGEMENT

At system start up the executive will allocate most system resources, such as memory, tape drives, disk drives, etc. Some devices such as the operator terminal and programmer/analyst terminals will not be allocated by SAE, but will remain under control of the host operating system. The executive will suballocate resources to users and batch tasks and will resolve contention problems through a priority allocation scheme.

B. ERROR MANAGEMENT

The SAE will provide a centralized error handling facility to intercept and handle errors arising from user input, applications software execution, data and I/O, the executive and the operating system. Applications programs will call a single error processor within the executive when error conditions are encountered. User input errors will be reported to the user and interactive (on-line) users will be given a choice of: recovery methods, optional HELP information, restart and return to input mode.

C. USER-MACHINE COMMUNICATIONS

User interface through menus and a command language will be available as well as the ability to dynamically switch between modes. Both modes will allow the user to specify the applications programs, data and program control parameters to use in their data analysis tasks. Default parameter values will be supplied by the executive to the maximum extent possible. A limited amount of parameter checking will be done to assure that required parameters are specified and that values are within an expected range. Menus will be structured in a hierarchy to organize the system capabilities and lead users in a logical manner to the applications programs they need. The command language will provide users with the same system capabilities as the menus, but with greater flexibility. For example, it will be possible within the command language to define procedures which can then be used to execute sequences of applications programs.

D. DISPLAY MANAGEMENT

The display manager will provide a number of functions to support image data transfer to and from the display refresh memories, display function control of look up tables, cursor and other program controllable features, image data manipulation (scrolling, shifting, mosaic construction), annotation and zoom.

E. DATA MANAGEMENT AND I/O

The SAE will manage image and non-image data by providing catalogs, data set labels, a data set naming convention, read/write access to data and some image data format conversion. Image data formats supported will be band interleaved by line (BIL), band sequential (BSQ) and band interleaved by pixel (BIP). Non-image data types include vector, network, polygon and tabular. Data set labels will contain information on ownership, creation date, data type, data source, size, format and data set history. Catalogs will be hierarchically structured directories of data on tape and disk.

F. TASK CONTROL AND MONITORING

The SAE will interpret user requests for application program execution and generate system service calls to invoke the specified programs. Program control information will be formatted into parameter packages and passed to the applications programs. An executive routine will be called by each application to read parameter packages and store the values into local variables. On line users will be able to start one synchronous and several asynchronous tasks and run them concurrently. The executive will support status reporting so that users can monitor the progress and condition of their applications tasks.

V. FUNCTIONAL DESIGN

A. OVERVIEW

The design of the SAE is oriented around the use of data structures which are independent from the code, that is, the system and its status are defined through the data structures. The system can be reconfigured by changing or adding control blocks without modifying code or retask building.

The term "process" is used in the functional design to refer to an applications program or a utility program. A "port" is any source or starting point for initiating input to the SAE such as a user terminal, an asynchronous process, a remote process and a batch process.

The following material provides an overview of some of the major aspects of the functional design.

B. USER CONTROL

Menu Mode - The main menu specifies the highest level breakdown of the systems capabilities. The user types the letter or number corresponding to the menu item they select and the next menu is presented. When a specific process has been selected the final menu(s) are used to specify control parameters for the process. Other options available while in the menu mode are:

COMMAND	enter command mode
! proc	skip intermediate menus and execute proc synchronously, remain in menu mode.
HELP[proc]	display general help information or proc specific information.
MENU	return to main menu.
<ESC>	return to previous menu.
LOGOFF	log user off
<CR>	take default
other	return to main menu.

Command Languages - More experienced SAE users will probably choose the command mode. The command syntax is

```
[command[(who,where,when)]]proc
[key1=value1,...]
```

where brackets denote optional fields. Commands tell the system how to perform a process (see command list). Command qualifiers (who, where, when) specify the library to use, the network node the command is directed toward and the start time for a batch job. The proc field specifies either a process or a command procedure to be run. The proc qualifiers (key 1 = value 1) define the parameters required by the process or procedure.

The SAE commands are LOGON/LOGOFF to a terminal, RUN/START a synchronous/asynchronous process, BATCH place a job in the batch queue, ABO abort a process or job, STAT provide status information, MENU/COMMAND to change modes, EXIT to release a terminal from SAE back to the operating system, ALLOC/DEALLOC resources for a user, REMOVE/ADDPORT to modify the port list, INSTALL a new process or procedure, DEFAULT to set or display values for system parameters.

The VICAR commands are supported and follow their natural syntax, not the SAE command syntax.

Command Procedures - A sequence of SAE commands, called command procedures, can be created, saved in a library and executed by SAE. Command procedures or procs can also be used to set groups of parameters and save them for reuse. Procs contain a header and text. The header is a command containing the proc name and its associated parameters. The text portion consists of commands which specify processes or other procedures to be executed. Symbolic parameters are used for value or character substitution from higher to lower level procs, or from the command invoking the proc to the proc itself. Procedures may contain expansion lines to permit modification of the proc. This capability allows for looping and condition testing within the procedure language.

HELP Facility and Tutored Input - The HELP facility can be reached from menu or command mode. Help can be requested to explain use of the menu system, command language syntax, procedures, processes and parameters. The tutored input (TI) mode is used to provide information on keyword parameters and the ability to specify keyword values. In the menu mode tutored input is used automatically to assist the user with parameter input. In the command mode TI is reached when the user requests help with parameter input or when a parameter input error has been detected by the executive.

C. DATA STRUCTURES

Disk Data Structures - Three types of control blocks are maintained on permanently open disk files to define all application and utility functions installed on the system. These control blocks can be dynamically changed to modify the Applications Configuration. The Menu Definition Block (MDB) contains all information about a given menu, including references to Picture and Message files, permissible responses and pointers to logically related MDBs or Process/Procedure Definition Blocks (PDBs). The PDB holds information needed to initiate a process or procedure. The PDB points to a Data Definition Block (DDB) which holds input parameter and keyword information for the process or procedure. The DDB is used in parameter checking by the SAE. If required parameters are not defined, or if parameter values are out of range, the tutored input mode is entered automatically. TI prompts for definition or correction of those parameters which are unspecified or misspecified.

Libraries - Command and procedure libraries are data files whose data entries can be accessed by name to avoid sequential searches. The command library (CMDLIB) is used by the command processor (CP) to interpret and execute commands. Procedure libraries (PROCLIBs) contain command procedures. A series of procedure libraries is maintained to support different classes of command procedures. User libraries contain user created special purpose procedures. Procedures with general usage are stored in the system library. Test and display libraries are used for procedures under test and for procedures tailored to specific display terminal hardware. A precedence scheme is used to specify the order the libraries are searched. Users may override this scheme by using LIB=SYS, for example, in the who field of the command invoking the procedure.

Processor/Procedure Installation - The INSTALL command is used to dynamically change the applications configuration. New processes and procedures are installed by making appropriate entries in the PROCLIB directory and the specified procedure library and creating or modifying control blocks (PDBs, DDBs and MDBs). Menu entries can be created which permit execution of command procedures as well as new processes.

D. EXECUTIVE STRUCTURE

The executive consists of the code for:

o Control and manipulation of data structures.

o User interface.

o Application program (process) interface.

o Application program services.

Conceptual Overview - The SAE is composed of a KERNEL and one or more PORT MONITORs. The KERNEL is a serially reusable task which controls memory resident data blocks, controls resources, handles requests for service and initiates and controls the PORT MONITOR. A PORT MONITOR exists for each PCB allocated by the KERNEL. For terminal ports the PORT MONITOR consists of a Terminal Monitor (TM) and a Command Processor (CP). For other types of ports it consists of an Independent Task Monitor (ITM) and a CP. As indicated by its name, the Command Processor interprets commands and initiates the designated process. The Terminal Monitor provides communications with the alphameric terminals and performs menu processing. It then passes commands on to the CP for handling. The ITM accepts procedures as input and passes commands on to the CP.

Resource Allocation - The KERNEL controls resources required by users and processes. The set of devices allocated to SAE at system initiation is contained in a Resource Availability Table (RAT). Resources required by users are entered in the User Reserved Resources Table (URRT) and process required resources are entered in the Process Resource Requirements Block (PRRB). Some user resource requirements are inferred, such as allocation of the terminal at LOGON and allocation of a disk pack associated with a specific user. Most user resource requirements must be explicitly requested. Process required resources are defined when the process is installed and maintained in the PRRB. When a device request is made the device is checked for availability. Generally, a process requesting an unavailable device will not start. For some non-reservable devices, such as array processors, a process may be kept in a wait state until the device becomes available or a time limit is reached.

Synchronous Process Management - At system initialization the operator allocates a user terminal to SAE. A PCB is created by the KERNEL and the terminal monitor task is started. When the terminal monitor recognizes a LOGON the users directory is opened and an initial environmental descriptor is read in. The environmental description contains information describing the environment a command is to be executed within, including, the user ID, default procedure library and I/O routing. After LOGON the terminal monitor calls PROCESS to run the session. PROCESS uses RDLINE to read tutored input lines from the terminal and RDCMD to read command lines from the terminal. PROCESS then uses EXPAND to expand the procedure specified in the user command. First, it reads the procedure, line by line and then passes each line to an EXEC procedure. EXEC handles each line as an internal command, determines if the line references a procedure or a process, checks parameters and accepts tutored input if required. If the current line specifies a process EXEC invokes the application program and waits for a request for services from the program or a termination message. On termination EXEC returns to EXPAND to look for the next line.

If the current line specifies a procedure EXEC saves the local parameter list and assigns the new values contained in this line. Then the appropriate procedure file is opened and EXPAND is called to expand this nested procedure. Multiple levels of procedure nesting may be encountered. EXEC and EXPAND operate recursively to fully expand and execute these nested procedures. When EXPAND encounters an end of file while looking for the next line the procedure is completed and control returns to PROCESS.

Display Management - The executive can support multiple display devices, each of which can control one or more image analysis terminals (IATs). A Display Terminal Control Table holds default information relating ITATs to CRTs and the number of refresh memories per IAT. Device Status Blocks exist for each refresh memory to identify which user has allocated the memory. A user may read/write to his refresh memories and read only from other memories. Allocation, data management and I/O access for refresh memories are handled the same as for any other data set except that special software is used at the device interface level.

Since display devices differ widely in their hardware capabilities each has its own display procedure library. A basic set of display functions (refresh memory load/unload, lookup table load/unload, bit plane load/unload, scroll, shift, mosaic, etc.) is defined for all display devices and the procedures account for which functions are implemented in hardware and which in software. The separate procedure library maintained for each display device allows default parameters to be tailored to the characteristics of the given display device.

Batch and Asynchronous Job Management - The START and BATCH commands are used to initiate synchronous and batch jobs. These commands are handled by EXEC which creates a request and sends it to the batch job manager (BATCH) or to the KERNEL (START) to satisfy the commands. The request contains the command and the environmental descriptor. The batch job manager places the job in the batch queue on disk. For asynchronous jobs the KERNEL creates a PCB and links it to the PCB queue and to the appropriate User Reference Table (URT). The URT contains the user ID and counters for items which are under quota restrictions, such as the number of batch jobs. The KERNEL than sends the command line and environmental descriptor to the Independent Task Monitor. Batch jobs can be released automatically by the batch job manager, manually by the operator or by the user. When a user releases one of his batch jobs it becomes an asynchronous job and is subject to asynchronous job constraints.

VI. SUMMARY

The initial functional design of the SAE is being reviewed and refined prior to beginning the detailed design phase. A first level (partial) implementation is planned for completion by mid 1981. The design is structured to permit a phased implementation. The SAE will ultimately be used in a multi computer network. Interprocessor communications capabilities are being incorporated to share data and resources. It is the hope of the authors and SAE development team that this executive will be widely disseminated and used in the remote sensing community.

REFERENCES

1. Bracken, P. A., G. G. Knoble, C. B. Howell and P. H. Van Wie, "The Design and Applications of the Landsat-D Assessment System", Harvard University International Users Conference on Computer Mapping Hardware, Software and Data Bases, July, 1979.

2. Rose, J., "Software Portability", Computer Sciences Corporation, Draft Document, Goddard Space Flight Center contract NAS5-24350/326, January, 1980.

3. Durachka, R., M. Banks, K. Lehtonen, K. Posey, G. Cyprych, "Software Engineering Standards and Software Development Life Cycle", Internal Document, Information Extraction Division, Goddard Space Flight Center, January, 1980.

4. Van Wie, P. H., D. R. Howell, J. Owings, D. Fischel, M. Szczur, "Functional Requirements for the Standard APplications Executive", Internal Document, Information Extraction Division, Goddard Space Flight Center, December, 1979.

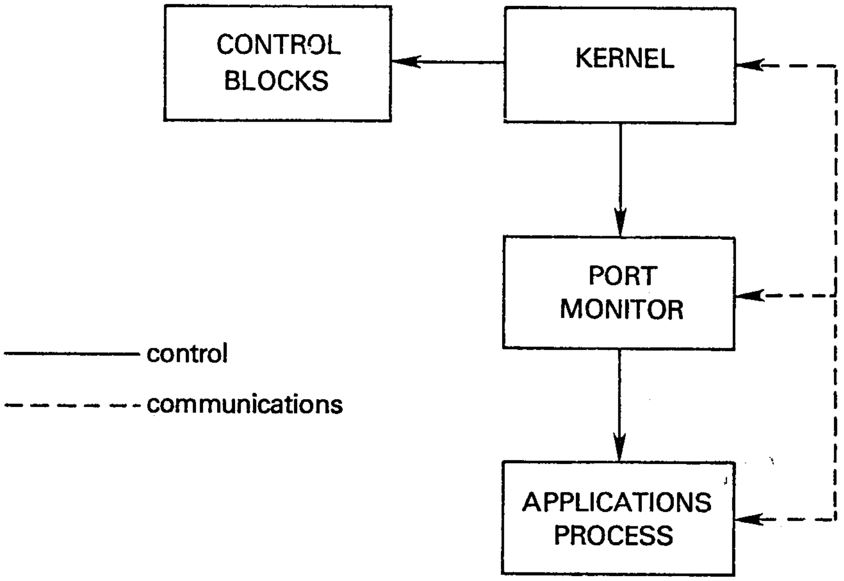
5. Howell, D. R., J. Owings, M. Szczur, "SAE System - - A Conceptual Design", Internal document, Information Extraction Division, Goddard Space Flight Center, February, 1980.

6. Poole, P. C. and W. Waite, "Portability and Adaptability", chapter on Advanced Course in Software Engineering, P. 183, 1973.

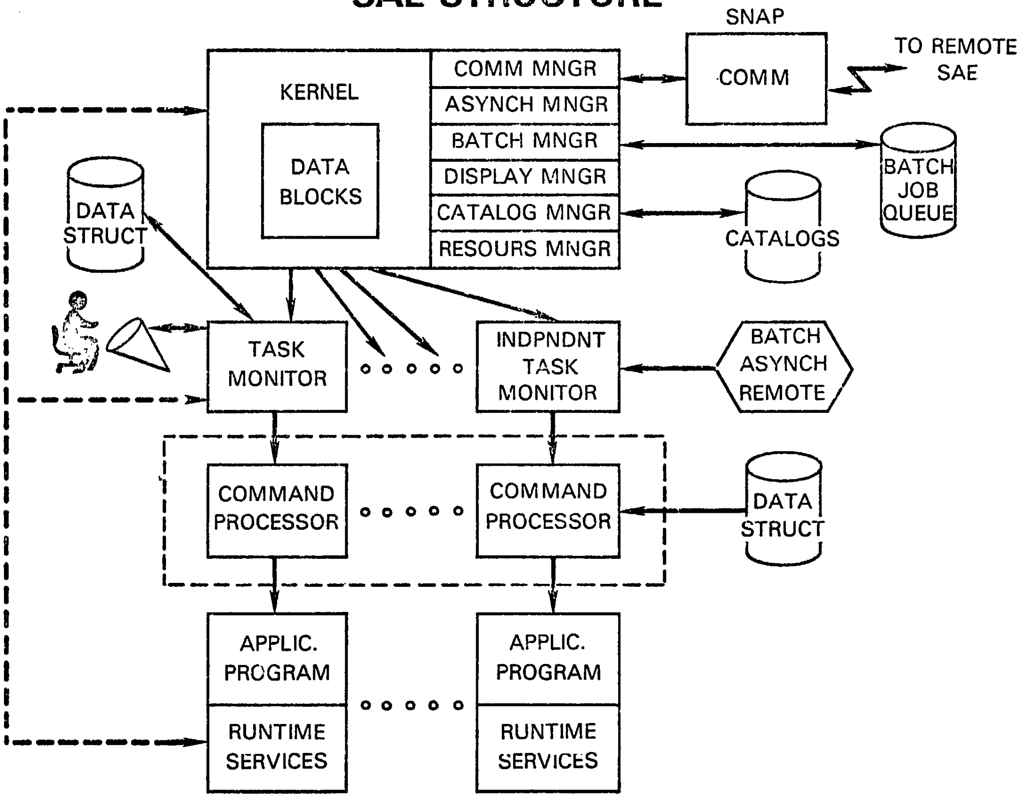
Table 1. Interactive Applications Systems

SYSTEM NAME	PROCESSOR	DISK SPACE	TAPE	IAT	SOFTWARE
Atmospheric and Oceanographic Information Processing System AOIPS	PDP 11/70	616 Megabytes (MB)	3-9track 1-HDTR	2-Hazeltine DDG-17 5 channel	METPAK-Meteorological Analysis Package
Image 100	PDP 11/45	2 MB	2-9track	GE 5 channel	Earth resources applications software.
Visable and Infrared Spin Scan Radiometer Atmospheric Sounder (VAS) Processor	PDP 11/70	528 MB	2-9track	IIS 3 channel	VAS demonstration. Extract temperature and humidity profiles.
Domestic Information Display System (DIDS)	VAX 11/780	352 MB	2-9track	De Anza 1 channel	Display Census data at state, county, and census tract level.
Landsat-D Assessment System (LAS)	VAX 11/780 + AP120B	1408 MB	2-9track 2-HDTR	3-Hazeltine DDG-17A 8 channel	Earth resources applications software. Evaluate Thematic Mapper digital imagery.
Pilot Climate Data Base Management Systems PCDBMS	TBD	704 MB	3-9track	2 IATS TBD	Integrated data base management system.
VAS Assessment Processor	TBD	704 MB	3-9track 2 shared with AOIPS	1-4 IATS TBD	Multisource data integration for meteorology/forecasting

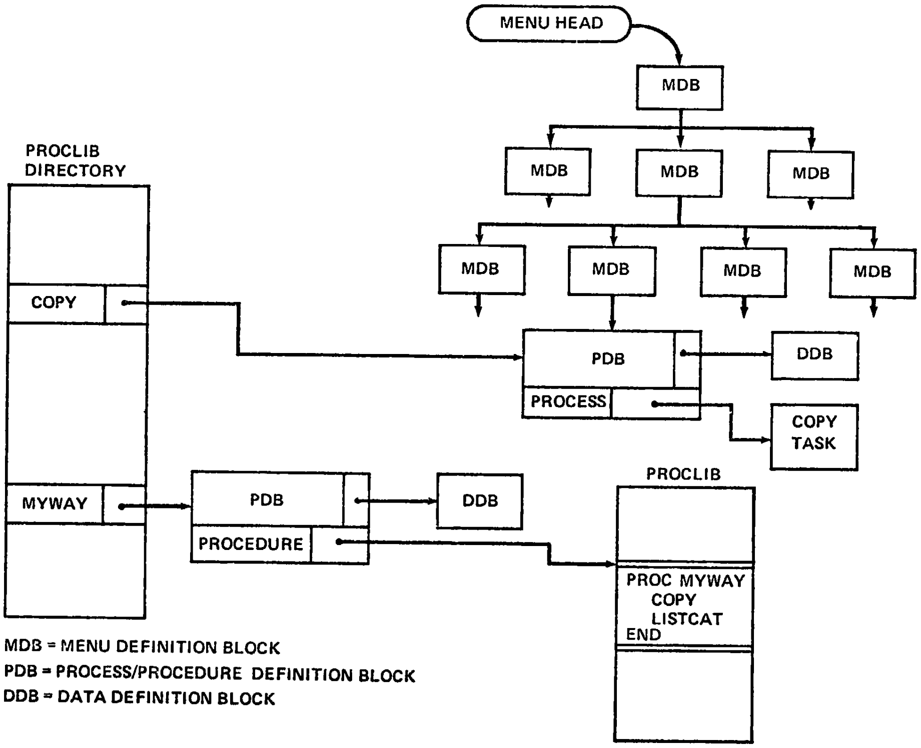
Conceptual Organization of Executive



SAE STRUCTURE



DISK DATA STRUCTURES



MDB = MENU DEFINITION BLOCK
PDB = PROCESS/PROCEDURE DEFINITION BLOCK
DDB = DATA DEFINITION BLOCK

Peter Van Wie received his Bachelor of Science degree in mathematics from Wayne State University in 1965. He has worked for Goddard Space Flight Center since 1966. From 1966 to 1972 he worked on the development of spacecraft trajectory analysis programs. Since 1972 he has worked in the area of remote sensing software development, primarily for Landsat multispectral scanner image processing. In June 1979 he was appointed to head the Systems Development Section in the Interactive Systems Development Branch. His current assignments involve development of the Standard Applications Executive and applications software for the Landsat-D Assessment System.

David Howell received his Bachelor of Science degree in physics from New Mexico University in 1968. He came to Goddard Space Flight Center in 1964 as a co-op student developing applications software for Earth orbiting satellites and deep space probes. He has over 15 years experience in the design and development of space applications software, graphics programming system, process control, real time analysis and instrument control, communications and executive control systems. In December 1979 he was appointed to head the System Applications Section in the Interactive Systems Development Branch. He is currently working on developing the Standard Applications Executive and a real time image analysis system for the VAS Demonstration Project. He is a member of the ACM and the American Institute of Physics.

David Fischel received his Bachelor of Science degree from Brown University in 1958 and a PhD from Indiana University in 1963. He was a research scientist at NASA/Ames Research Center from 1963 to 1965. He joined Goddard Space Flight Center in 1965 and has conducted research in stellar physics, plasma physics and astronomical image processing. He has fifteen years of experience in the design, development and implementation of application software systems, including the FORTH image processing system using a PDP 11/40 computer. He is a member of the ACM, Sigma Xi and SPIE. He is currently a senior member of the Interpretive Techniques Branch in the Applications Directorate at Goddard Space Flight Center.