2

UCRL-92089
PREPRINT
CONF-8504108--1

EDS OPERATOR AND CONTROL SOFTWARE

Linda L. Ott

This paper was prepared for submittal to
Interex/HP-1000 Users Conference,
Antwerp, Belgium
April 15-19, 1985

April 1985

Lawrence
Livermore
National
Laboratory

UCRL--92089

DE85 010470

EDS Operator and Control Software*

Linda L. Ott

Lawrence Livermore National Laboratory
P. O. Box 808 L-441
Livermore, California 94550
U.S.A.

ABSTRACT

The Enrichment Diagnostic System (EDS) was developed at Lawrence Livermore National Laboratory (LLNL) to acquire, display and analyze large quantities of transient data for a real-time Advanced Vapor Laser Isotope Separation (AVLIS) experiment. Major topics discussed in this paper are the EDS operator interface (SHELL) program, the data acquisition and analysis scheduling software, and the graphics software. The workstation concept used in EDS, the software used to configure a user's workstation, and the ownership and management of a diagnostic are described. An EDS diagnostic is a combination of hardware and software designed to study specific aspects of the process. Cverall system performance is discussed from the standpoint of scheduling techniques, evaluation tools, optimization techniques, and program-to-program communication methods. EDS is based on a data driven design which keeps the need to modify software to a minimum. This design requires a fast and reliable data base management system. A third party data base management product, Berkeley Software System Database, written explicitly for HP1000's, is used for all EDS data bases. All graphics is done with an in-house graphics product, Device Independent Graphics Library (DIGLIB). Examples of devices supported by DIGLIB are: Versatec printer/plotters, Raster Technologies Graphic Display Controllers, and HP terminals (HP264x and HP262x). The benefits derived by using HP hardware and software as well as obstacles imposed by the HP environment are presented in relation to EDS development and implementation.

*This work was performed under the auspices of the U.S. DOE by Lawrence Livermore Nat'l. Lab under contract No. W-7405-Eng-48.

1. INTRODUCTION

The AVLIS process uses finely tuned frequencies of laser light to photoionize, or electrically charge, atoms of a particular isotope of uranium. The photoionized atoms are collected on charged plates. Reactor grade uranium can be produced by separating uranium-235 atoms from other naturally occuring isotopes. The purpose of developing the AVLIS process is to produce reactor fuel at a lower energy consumption and at lower capital and operating costs than other processes (gaseous diffusion or centrifugation). An experimental facility (Fig. 1) was constructed at LLNL to study laser isotope separation. The project is divided into two areas: the laser facility and the separator facility. EDS was written to support the separator portion of the process.

Two types of data were identified as being necessary to collect: the conventional, relatively slowly changing scalar values (temperatures, pressures, etc.) and large vector arrays of fast transient data. Many commercial systems are available to monitor the slowly changing data and the decision was made to use the HP product PMC/1000. To collect the highly specialized types of data, the data acquisition system (EDS) was developed in-house. EDS is functionally divided into three parts: data acquisition hardware and software, control and graphics software, and analysis software. This paper describes the second function: control and graphics.

The general philosophy behind EDS is to create a diagnostic to study specific aspects of the process and then, once it is sufficiently characterized, to replace the diagnostic with a "black box" to provide the condensed data to the plant process and control system. For example, during one vapor diagnostic sweep cycle, 64K words of data are reduced to approximately 10 values (e.g. temperatures, velocities, densities). The natural

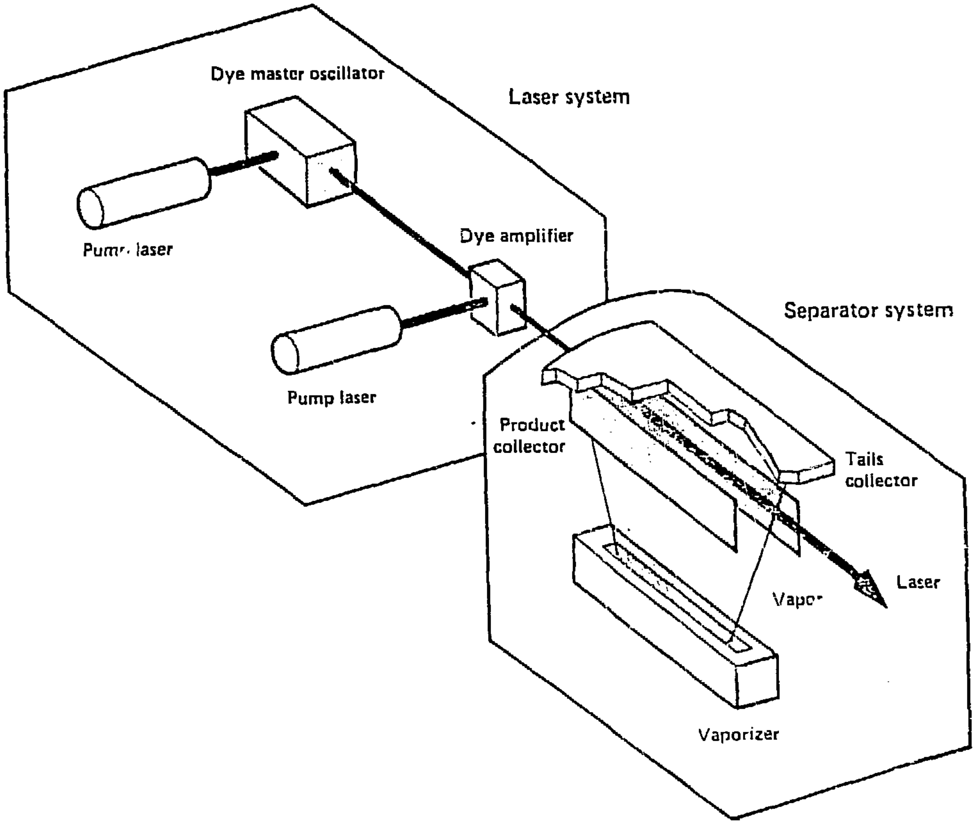

Figure 1. Atomic Vapor Laser Isotope Separation Diagram.

progression for an EDS diagnostic is to migrate from an interactive experiment to a passive instrument.

2. DESCRIPTION

The Separator Instrument and Control system is based on a network of five HP1000's:

Node 1:	Post Run Analysis	(F series)
Node 2:	PMC/1000	(A700)
Node 3:	EDS	(A700)
Node 4:	Feeder system	(A700)
Node 5:	Modelling	(A900)

EDS consists of over 100 programs written primarily in FORTRAN. EDS has evolved over a number of years from a one user, one diagnostic system to a multiple user, multiple diagnostic system. EDS currently manages four diagnostics and eight users. The four diagnostics are:

* Vapor characterization by absorption spectroscopy
  (64 channels, 1024 words per channel)

* Gas analysis by mass spectrometry
  (10 channels, 1024 words per channel)

* Extractor Performance using transient recorders
  (20 channels, 1024 words per channel)

* Process Laser Characterization
  (32 channels, 1024 words per channel)

A complete diagnostic cycle consists of data acquisition, display, analysis and transmission of key process parameters to the process monitoring/control system, PMC/1000, and to the process modelling system.

3. GENERAL REQUIREMENTS FOR EDS

The requirements fo- EDS are listed below:

* Acquire large quantities of transient data.

* Create a convenient user interface
  - Provide means to configure workstation displays
  - Provide simple commands to manipulate diagnostic

* Provide guarantee that only one user is controlling a diagnostic at a given time.

* Design interface to incorporate new commands easily.

* Allow several levels of command execution to handle a range of user sophistication.
  - no interaction
  - detailed interaction

* Permit different modes of diagnostic operation.
  - Automatic Mode: Used when system is stable. Scan at specific intervals.
  - Manual Mode: Used during critical transition periods. User has complete control.

* Ensure that the first priority is data acquisition. Graphics and analysis are to be done as time allows.

- * Provide immediate results of scan.
  - Graphical output should be displayed within ten seconds of the scan's completion.
  - Analytical capabilities should be user defined and modifiable during the run.

- * Make data available for display at any workstation.

- * Need ability to isolate or coordinate diagnostics.

- * Need convenient means of adding new diagnostics.

- * System needed immediately. The deadline was moved from 1987 to April 1985.

4. EDS DESIGN ELEMENTS

After studying the general requirements for EDS, the following principle design elements emerged:

- * Ability to share large amounts of data among many programs (data acquisition, graphics, analysis codes).

- * Make codes data table driven. Use data bases and interpretative codes.

- * Make data acquisition, graphics and analysis independent of each other. The system should be uncoupled.

- * Need a fast system.

- * Need a standard user interface.

- * Need a general graphics package.

5. EMA DEFINITION

Shared extended memory area (EMA) was selected as the fastest means of sharing large data arrays among programs. Twelve named common blocks were created using a total of 778 pages (1024 words per page) of memory area. The common blocks are listed below:

- Vapor diagnostic raw data and parameters (65 pages).
- Mass spectrometer diagnostic raw data and parameters (33 pages).
- Extractor diagnostic raw data and parameters (21 pages).
- Integrated separator diagnostic raw data and parameters (33 pages).
- Vapor analysis work area (77 pages).
- Mass spectrometer analysis work area (9 pages).
- Extractor analysis work area (41 pages).
- Analysis matrix (165 pages).
- Graphics data area (202 pages).
- Graphics user tables (97 pages).
- Graphics viewport tables (9 pages).
- Graphics element tables (26 pages).

Each diagnostic has its own common area which contains the raw data and the parameters necessary to control the diagnostic. The raw data is duplicated in both the graphics and the analysis work common areas. The cost of keeping graphics and analysis independent of data acquisition and of each other is large memory usage.

6. EDS INTERFACE PROGRAM (SHELL)

The SHELL is the primary interface between the experimenter and the rest of the EDS software. The main purpose of the SHELL is to provide the user with a mechanism to programmatically control a diagnostic, analyze the data and graphically display the raw and reduced data. Basically, the SHELL schedules programs and passes commands to active programs. It isolates the user from the HP operating system. All commands to EDS are executed through the SHELL. System log-on files are created for each diagnostic which set the working directory and run SHELL with the appropriate diagnostic name. The first prompt the user sees is the SHELL prompt for his diagnostic.

Commands are grouped in the SHELL by category. Each diagnostic has its own category. Once a user has obtained ownership of a category, he has exclusive access to the commands in the category.

Commands within a category are two characters followed by parameters. If parameters are required but not entered, a menu is presented to prompt the user for input. The parameters are ranged checked and validated before passing on to the appropriate program. The SHELL is not connected with EMA.

All commands are contained in the SHELL database. When new commands are needed, they are entered into the data base. The SHELL program itself is not modified. The following elements define a command in the data base:

Category (two characters)
Subcategory (two characters)
Purpose (forty characters of text)
Execution or communication mode (two charcter action selector):

Use class I/O
Single copy per user, run with wait
Single copy per user, run without wait
Single copy per system, run with wait
Single copy per system, run without wait

Name of program to execute or program to receive class I/O message
Specify if any parameters are required for the command
Number of parameters associated with the command
Parameter type (integer, real, ASCII)
Specify if individual parameters are required
Parameter definition (text description)
Parameter key words
Parameter default values
Parameter lower limits
Parameter upper limits
Specify if default values can be used
Forms library name
Menu number

The standard format for a message from the SHELL is an integer ASCII string. The string contains the SHELL data base manager name, the SHELL workstation, the command, and the parameters.

7. SHELL HELP LEVELS

Several levels of help are available. Commands within a category are grouped by function into subcategories. Two question marks (??) will list all the categories and subcategories in the SHELL. If a category is owned, the owner will also be listed. A question mark followed by a category (e.g. ?VC) will list all the commands available in the category; a question mark followed by a subcategory (e.g. ?V8) will list the commands in the subcategory with a short description of the command. A question mark followed by the command (e.g. ?VCSW) will list a detailed explanation of the command. Users may select different levels of command execution from immediate execution when the command is entered to stepping the user through each phase of command execution with a prompt.

8. WORKSTATION CONFIGURATION

Through the SHELL, the user configures the workstation's graphics device, typically either an HP2627 display terminal or a CONRAC monitor driven by a Raster Technologies Graphic Display Controller. A versatec printer/plotter may be selected as the graphics display device although this is usually not done in the real-time mode. For the Raster Technologies device, there are generally two graphic planes defined with either four or eight viewports per plane. In this manner, the user has eight to sixteen graphs available at one time. Plots may be superimposed; line types, colors, and symbols are selectable. The user also has the capability of plotting one data set against another. The data may be reduced in a number of ways before being plotted by defining functions and coefficients in the data base. The in-house Device Independent Graphics Library (DIGLIB) has been very valuable in meeting the EDS graphic requirements.

9. TYPICAL DIAGNOSTIC PROCEDURE

Perhaps the best way of illustrating the operation of EDS is to describe a typical diagnostic run session. The vapor diagnostic will be used for the example. Be aware, that the other three diagnostics could be running on separate workstations at the same time. It would be possible for one user to "own" and control all four diagnostic at once. The basic sequence of events for a diagnostic is:

- Set up instrument(s)
- Get data
- Plot raw data or slightly transformed data
- Condense the vector data to scalar parameters
- Plot the reduced data
- Send the reduced data to other nodes

A typical procedure is outlined below:

1. Log-on to the computer and EDS.

   The user types VAPOR
   The SHELL prompt (VC>) appears for the vapor diagnostic.

2. The user turns "ON" his diagnostic.
   a. CAMAC instruments are initialized.
   b. Update EMA with the data base values (Fig. 2)
      - Defines active channels
      - Labels
      - Engineering conversion algorithm and coefficients
      - Analysis parameters

   c. Set up Class I/O communication link between the SHELL and the message handling program (VMESS).

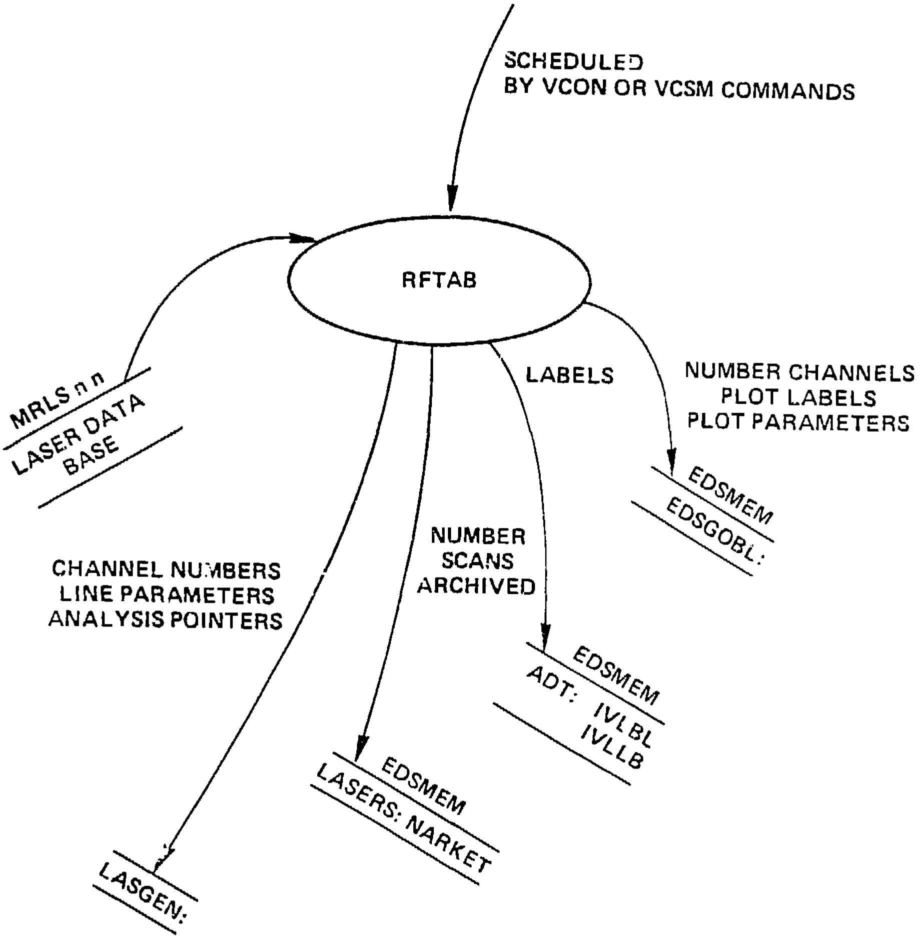

Figure 2. Update EMA with Data Base Values.

3. The user selects the graphs to be displayed.

4. The user selects analysis options:
   * Don't do any analysis
   * Calculate reduced parameters
   * Calculate and plot reduced parameters

5. Initiate data acquisition.
   The user may select to do sweeps one at a time, as rapidly as possible, or a set intervals in a batch mode.

6. The user may adjust a number of parameters during the experiment to tailor the diagnostic to changing needs.

7. End data acquisition.
   The data file is closed; Class I/O's terminated.

10. GENERAL SWEEP SEQUENCE

Continuing with the vapor diagnostic example, the sweep command processing is initiated when the user enters a sweep command to the SHELL (Fig. 3). If the user wants to take a number of sweeps as rapidly as possible, SW followed by the number of sweeps to do is entered.

The SW command causes the program VSWEEP to be executed. VSWEEP passes the sweep command and parameters to the data acquisition program, VCDL, via an EMA command stack. VCDL checks EMA for commands to process.

VCDL accepts the sweep command (Fig. 4) and sends the appropriate commands to the CAMAC instruments. When the data is ready to offload from the data logger, VCDL uses the HP routine VMAIO to transfer 32K words of data to EMA. Two VMAIO calls are

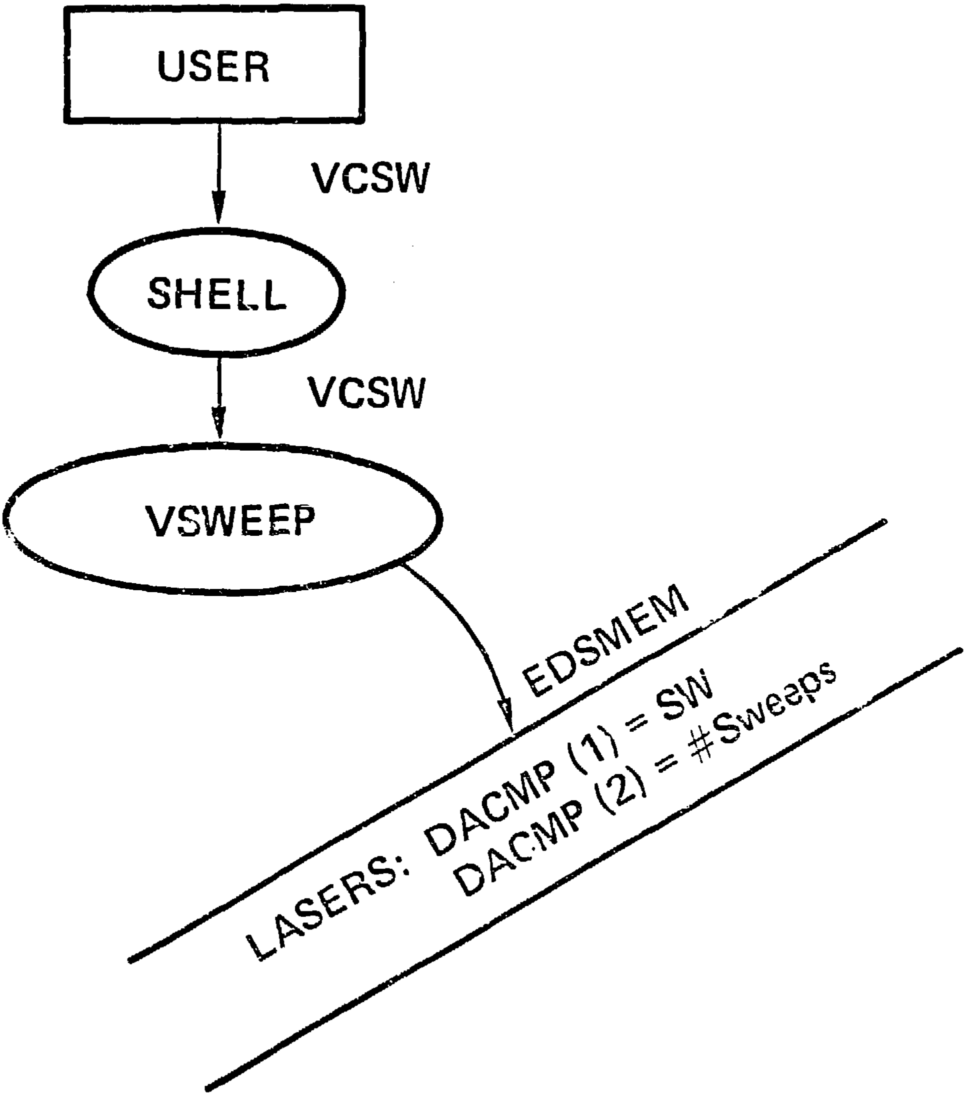

Figure 3. Execution of a Sweep Command.

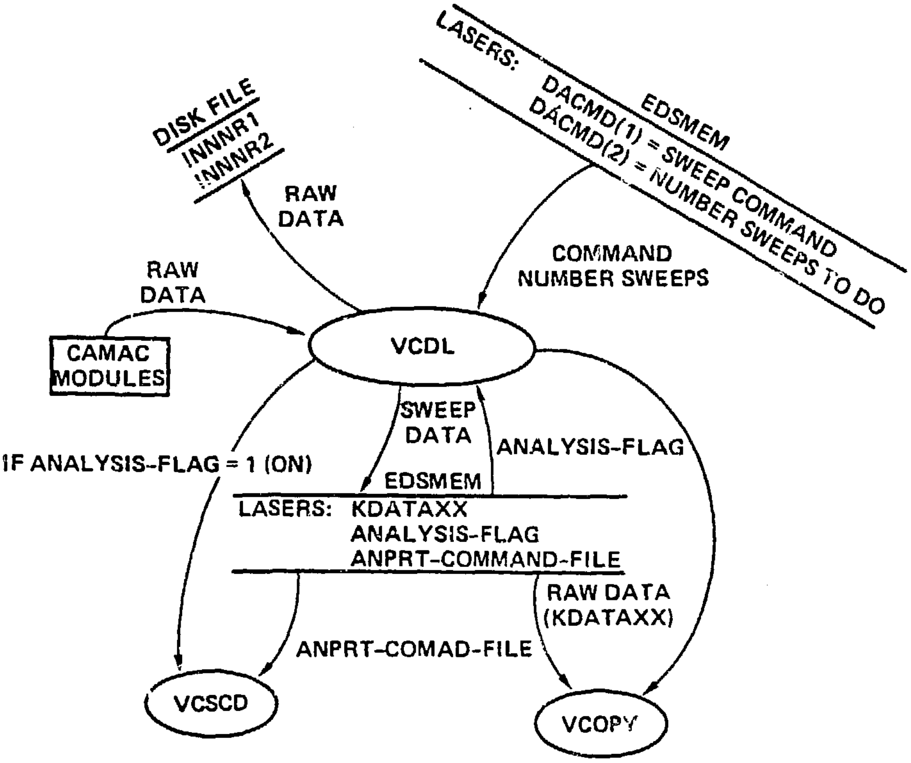

Figure 4. EDS Vapor Sweep Flow.

used to tra-sfer all 64K words of data. If the data is to be archived, it is written to disk and the data base is updated with the sweep information (time of sweep, file name, number of channels, and other parameters).

The graphics sequence is started by scheduling the graphics copy program, VCOPY, with an EXEC 10 call (immediate execution, without wait). In a similar manner, the analysis sequence is started by scheduling the program, VCSCD, with an EXEC 10 call if analysis has been requested.

The "without wait" option permits VCDL to continue without waiting for the son program (VCOPY or VCSCD) to complete. This was one of the requirements, namely, that graphics or analysis would not interfere with data acquisition. By requesting "immediate" execution, the graphics and analysis are skipped if either is still actively processing previous sweep data.

II. GENERAL GRAPHICS SEQUENCE

The graphics sequence is started when VCDL schedules the copy program, VCOPY (Fig. 5). VCOPY checks a flag to make certain that the data from the previous copy has been plotted. If it has not been plotted, VCOPY ends. If the flag is clear, the data is copied into the graphics area for the vapor diagnostic. The scan number is checked before and after the copy to ensure that the data is completely from one sweep. If the data is complete, the plot executive program, PLTEX is scheduled. PLTEX scans the graphics tables to determine which users have requested data from the vapor diagnostic. PLTEX schedules the plotting program, QUICK_LOOK_DRAW, for each user requesting vapor plots. When the data has been plotted, the flag is cleared so that VCOPY is allowed to overwrite the data the next time it is called.

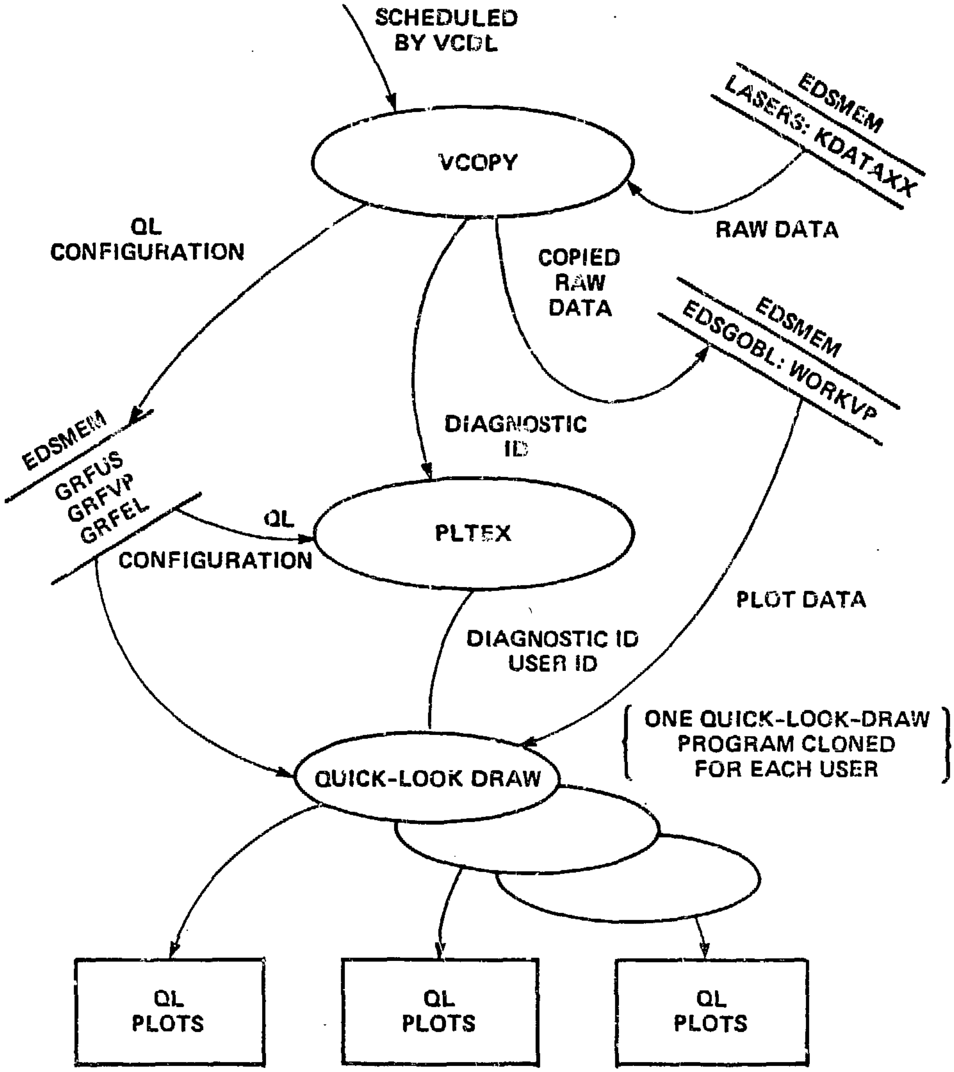

Figure 5. EDS Graphics Flow.

12. GENERAL ANALYSIS SEQUENCE

The analysis sequence is started when VCDL schedules the program, VCSCD (Fig. 6). VCSCD checks a flag to make certain that the data from the previous sweep can be overwritten. If it cannot be overwritten, VCSCD ends. If the flag is clear, the data is copied by ANCPY into the vapor analysis EMA work area. The scan number is checked before and after the copy to ensure that the data is completely from one sweep. If the data is complete, the program RTGEN is scheduled. RTGEN does the calculations requested and puts the reduced values in the analysis spreadsheet matrix. The data base contains directives of what to compute, how to compute it, and where to store the results in the spreadsheet matrix. These directives are stored in EMA for RTGEN to use. RTGEN clears the overwrite flag to permit new data to be copied. If plots have been requested of the reduced values, ANPRT is scheduled. ANPRT is capable of sending data to other nodes in the system. ANPRT is an interpretive code which receives its input from a file. The user can change the analysis parameters during the experiment by altering the input file to ANPRT. VCSCD schedules ANCPY, RTGEN, and ANPRT with wait. In this way, a new analysis cycle cannot be started until the previous one is completed.

13. BERKELEY SOFTWARE SYSTEMS DATABASE MANAGEMENT SOFTWARE

Since EDS is designed to be data base driven, it was essential to have a quick and reliable data base management system. The Berkeley Software Systems (BSS) hash database management software written by a consultant, Richard Lawhorn, is used for all EDS databases. BSS has been used in a number of projects at LLNL. It uses Class I/O to communicate between the application program and the data base manager. A hashing algorithm is used to extract data from a type 1 disk file by the data base manager program. The disk file is built by a structure initialize program (BSHSI) from a text file, the blocklist. A value initialize program (BSHVI) is used to initialize data cells

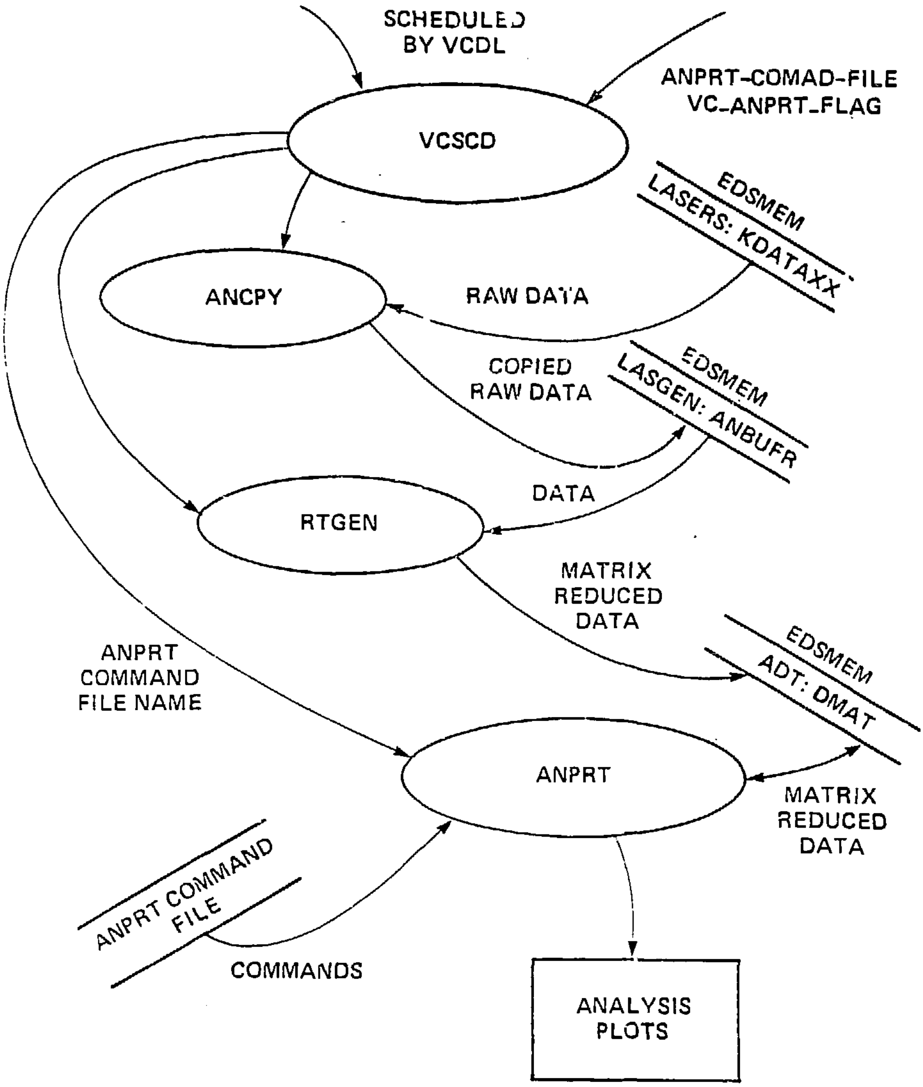

Figure 6. EDS Vapor Analysis Flow.

to defined values. A data base of a thousand blocks takes about two minutes to build and initialize.

14. SYSTEM PERFORMANCE

The use of shareable EMA was the first step taken to increase overall system speed. All frequently used parameters are kept in EMA. If data is to be used in detailed calculations, it is often beneficial to move it from EMA to local buffers. The double word addressing used for EMA variables is slower than the addressing scheme used for local variables.

Critical and frequently used codes are locked into memory and ended with the "saving resources" option. Priorities are adjusted so that the data acquisition codes have the highest priority. Priorities of other EDS codes are set according to their importance.

The buffer limits on I/O devices were increased to speed data throughout. Disk accesses are kept to a minimum. Class I/O was used extensively for program-to-program communication. Programs posting a Class I/O call can suspend themselves without CPU overhead and are awakened only when there is a message present. FMP file reads and writes are used instead of FORTRAN I/O since the FMP calls are much faster.

15. HP EXPERIENCE, GOOD AND BAD

Overall, the use of HP hardware and software has been a good solution. The system has meet the major requirements. It was discovered early in the project that the phone-in-consulting service (PICS) was not adequate in resolving many of the problems in a timely manner. A contract was made with HP to provide an on-site consultant one day a week to resolve problems, answer questions, perform system generations and write performance tools. This proved to be a very useful service in terms of resolving specific problems. Turning over system generations to HP was less successful in that problems would arise several days after the new system was installed. It was usually necessary to call HP back in to fix the problems. After several episodes of this, the system was frozen and no further updates were done.

Another troublesome area centered around the use of EMA, critical to EDS. The fact that HP provided a mechanism such as shared EMA to permit many programs to have access to large data arrays was very valuable. The restriction that there could only be one EMA declared per program was a problem. Separate common block were desired since each program needed only a subset of the total 778 pages of EMA. Two diagnostics, vapor and mass spectrometer, use VMAIO calls to transfer 32K words of data from the I/O device to their respective EMA raw data buffers. Depending upon the name of the common block, the transfers would either work or not. The errors for unsuccessful transfers were undefined.

After many months of investigation by the on-site consultant, it was discovered that LINK was the program that arranged the common blocks in memory and that the user had no control over the arrangement. LINK uses a hashing algorithm on the block names to define memory. The other crucial information obtained was that for a VMAIO call to successfully transfer 32K words of data, the EMA array must begin on a page boundary. In order to work under these restrictions, each common block is padded with the appropriate number of spares to make the block an even multiple of a page. Careful system management is necessary, since the redefinition of any common block requires the recompiling and linking of all codes using shared EMA.

Since the EDS design specifies large data array transfers, a fast copy routine was needed. The VIS EMA move routines seemed to be likely candidates, but unfortunately were not suitable. The

VIS routines only work on floating point values, whereas the EDS values are integer.

EDS depends upon having only one copy of certain programs active at one time in the system. Past operating system (RTE-6VM and RTE-4B) have included a "don't copy" bit in the program's ID segment to manage cloning. RTE-A has dropped this bit making it much more difficult to control cloning. It was discovered that by linking a code as a system utility and by RP'ing the code with the "don't clone" option, cloning could be inhibited.

Two performance tools provided by the HP consultant were very helpful in finding system bottlenecks. The program CPU displays a CPU utilization profile, and ACT is a detailed system activity monitor. The consultant also provided utilities to programatically obtain the state of any program in the system.

16. CONCLUSION

EDS has been a successful integration of work done by many people over several years. It is unique at LLNL in the amount of data acquired, the graphics displayed and the analysis done in real-time. The EDS design has withstood the test of time. While continually incorporating new commands, diagnostics, and capabilities, it has been run at regular intervals, once to three times a month. At the present time, EDS has been used in fifteen experiments with a total running time in excess of six hundred hours.

17. ACKNOWLEDGEMENTS

I wish to express my appreciation to my co-workers: Steve Telford, Dan Schneberk, Mathilde Killian, Tom Treadway, and George Miller. I thank Jay Ackerman, Joe Brandt, Jim Held and Lloyd Hackle for their support and for the opportunity to participate in the HP1000 User's Group Conference.