JANUARY 1981

HEWLETT-PACKARD JOURNAL

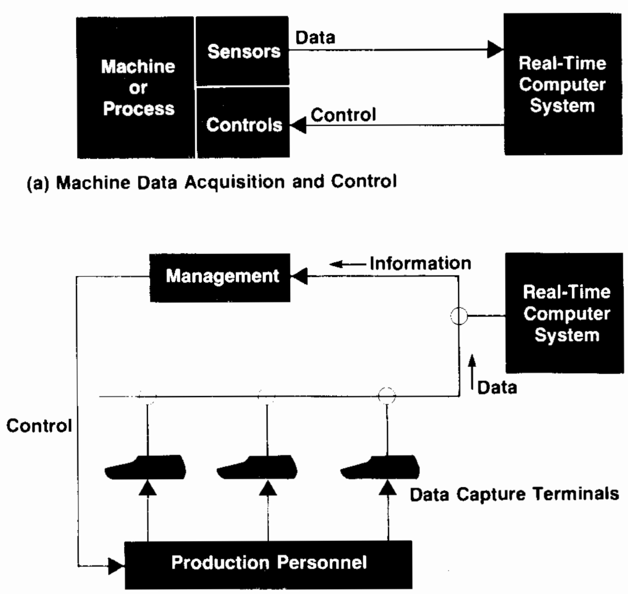
Programming for Productivity: Factory Data Collection Software

DATACAP/1000 is a software tool for designing and managing data collection networks. Running on an HP 1000 Computer System, it is flexible, easy to use, and compatible with user-written routines.

by Steven H. Richard

WHAT IS FACTORY DATA COLLECTION? For many people the whole concept of collecting or capturing data at its source is an unfamiliar one. Those who have had exposure to real-time factory automation are familiar with process or machine data acquisition and control loops (Fig. 1a). In factory data collection we are dealing with logistical control rather than machine control (Fig. 1b). The term factory data collection encompasses acquisition methods normally associated with manual input.

Examples of the kinds of data normally collected in the factory data collection environment include labor information (time and attendance), work-in-process tracking, inventory control, component/product testing, and so on. At the present time, most manufacturing companies keep track of this sort of information through paperwork systems or combinations of paperwork and batch computer systems.



(a) Machine Data Acquisition and Control

(b) Logistical Data Acquisition and Control

Fig. 1. *DATACAP/1000 is a software package for HP1000 Computer systems. It collects factory data such as time and attendance, work in process, inventory, and test statistics, and presents it to management. It is a system for logistical data acquisition and control (b) rather than the more familiar form of factory automation shown in (a).*

The result is that plant control information (inventory levels, order status, and so on) is at least several hours and often days old before management has the information in hand and can act on it. Without a comprehensive, real-time data capture system, the factory information system (management reports, graphs, bar charts) can be neither timely nor complete.

A Factory Data Collection System

A computer based system is particularly well suited to the relatively simple task of collecting data from the factory floor. The most powerful capability of the computer in helping to solve the data capture problem is that it can instantly catch errors made in submitting the data. There are frequently transposition, transcription, and other errors, which can be easily caught and corrected as they occur, using simple range checks, mask checks, or table lookups. This ability to validate data before it is accepted for further processing makes subsequent management reports more timely and accurate because the time-consuming and often impossible task of tracking down and correcting errors is substantially reduced.

Hewlett-Packard is one of several manufacturers of computer systems aimed at solving the factory data collection problem. The HP 1000 Computer System for factory data collection includes the following major components:

Hardware

HP 1000 E-Series Computer
HP 7906/20/25 Disc Drive
HP Factory Data Link Terminal Connection
HP Data Capture Terminals (Fig. 2) with optional
- -Displays (LED or CRT)
- -Keyboards (Numeric or Alphanumeric)
- -Card Readers (Type III, V, Magnetic)
- -Bar-Code Reader
- -Electrical I/O (RS-232C or IEEE-488)

Software

RTE-IVB Operating System
Standard Languages and Utilities
IMAGE/1000 Data Base Management System
DS/1000 Distributed Systems Software
DATACAP/1000 Factory Data Collection Software

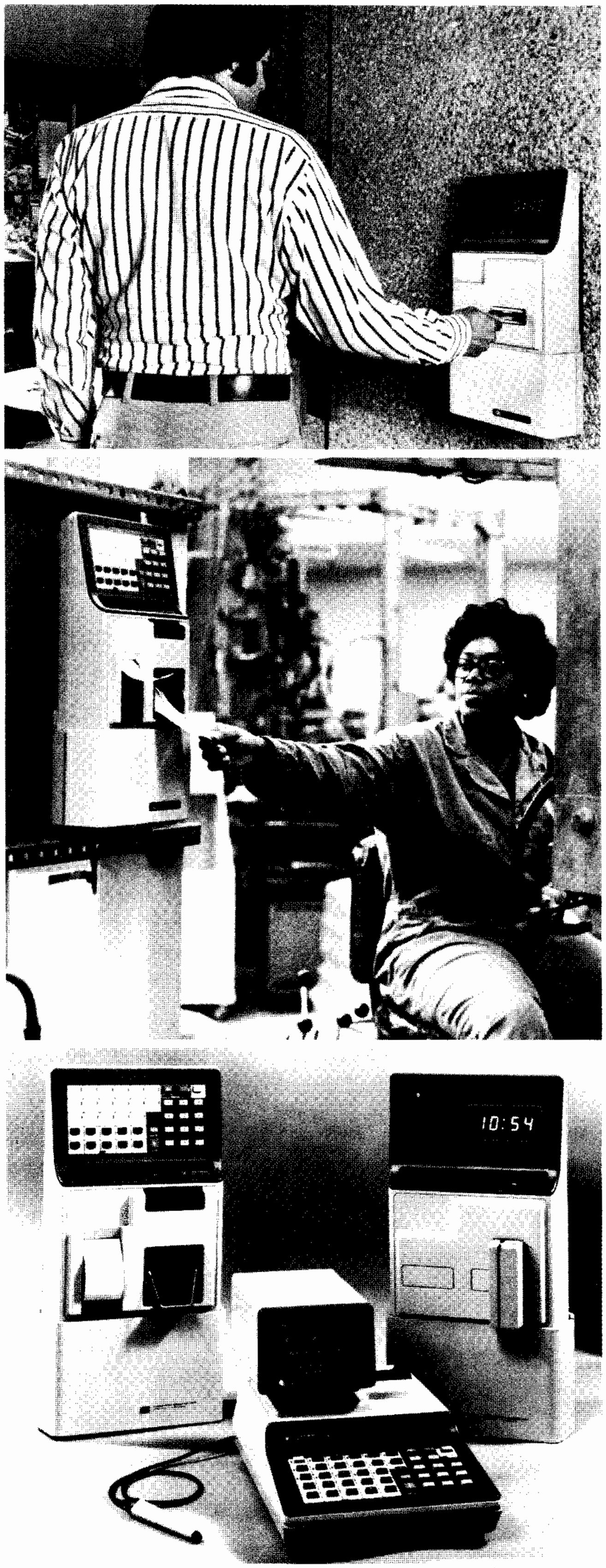


Fig. 2. *Data collected by DATACAP/1000 is entered manually by factory personnel using a variety of HP data capture terminals like these.*

DATACAP/1000

The key software element in this application is DATACAP/1000, a new software application tool that is used to customize the general-purpose HP 1000 System to the factory data collection needs of a manufacturing facility. An important feature of DATACAP/1000 is that it is *not* an application program that makes the HP 1000 System a turnkey data collection system. Instead, it is a program that helps the user develop an application program suited to a particular factory's needs. In other words, it is a *program that generates a program*.

Basically, DATACAP/1000 generates a generalized program that operates under the control of a parameter table. This program, the transaction monitor program (TMP), uses parameters supplied by the table (the transaction specification) to control an easy-to-understand dialogue at the HP data capture terminal. Additional parameters are used to direct interaction with an IMAGE/1000 data base, data storage devices, various optional input and output devices available with HP data capture terminals, and any customized source-coded subroutines that the user might wish to use to extend the basic table-driven capabilities. A block diagram of the major components of DATACAP/1000 is shown in Fig. 3.

The major advantage of the table-driven approach is that the development of a particular application (transaction specification) involves only the construction of a new table, rather than writing, keying in, editing and compiling a source program and loading the resultant object code. The transaction generator program (TGP), a friendly interactive program, prompts the user through the process of table building. Included in TGP are sophisticated edits that prevent the user from building a table with any logical inconsistencies that might cause problems during the execution of the application. A typical screen from the transaction generator program is shown in Fig. 4.

Similarly, the transaction monitor program generator (TMPGN), is used to develop the tables needed for the TMP to manage the aspects of the application that are less dynamic, that is, the data capture terminals, user-provided extensions (subroutines), and IMAGE/1000 data bases. a typical screen from the TMPGN is shown in Fig. 5.

Finally, a very simple monitor program (DCMON), is used to provide access to the generation and operation of DATACAP/1000 through the programmable function keys of the HP 2645A System Console. This relieves the system designer and operator of ever having to remember program names or run string commands. Fig. 6 shows a typical DCMON screen.

The Transaction Generator Program (TGP)

TGP consists of one main program and 15 segments. The main program is very short and is used only to initialize some variables and to call the first segment. Thereafter, the segments perform all functions and control is never returned to the main program. All communication between segments is accomplished through variables declared in common.

A basic overview of the path of TGP operation can be seen in the flowchart of Fig. 7. In addition to the formal reference manual for DATACAP, a brief tutorial is built into the trans-

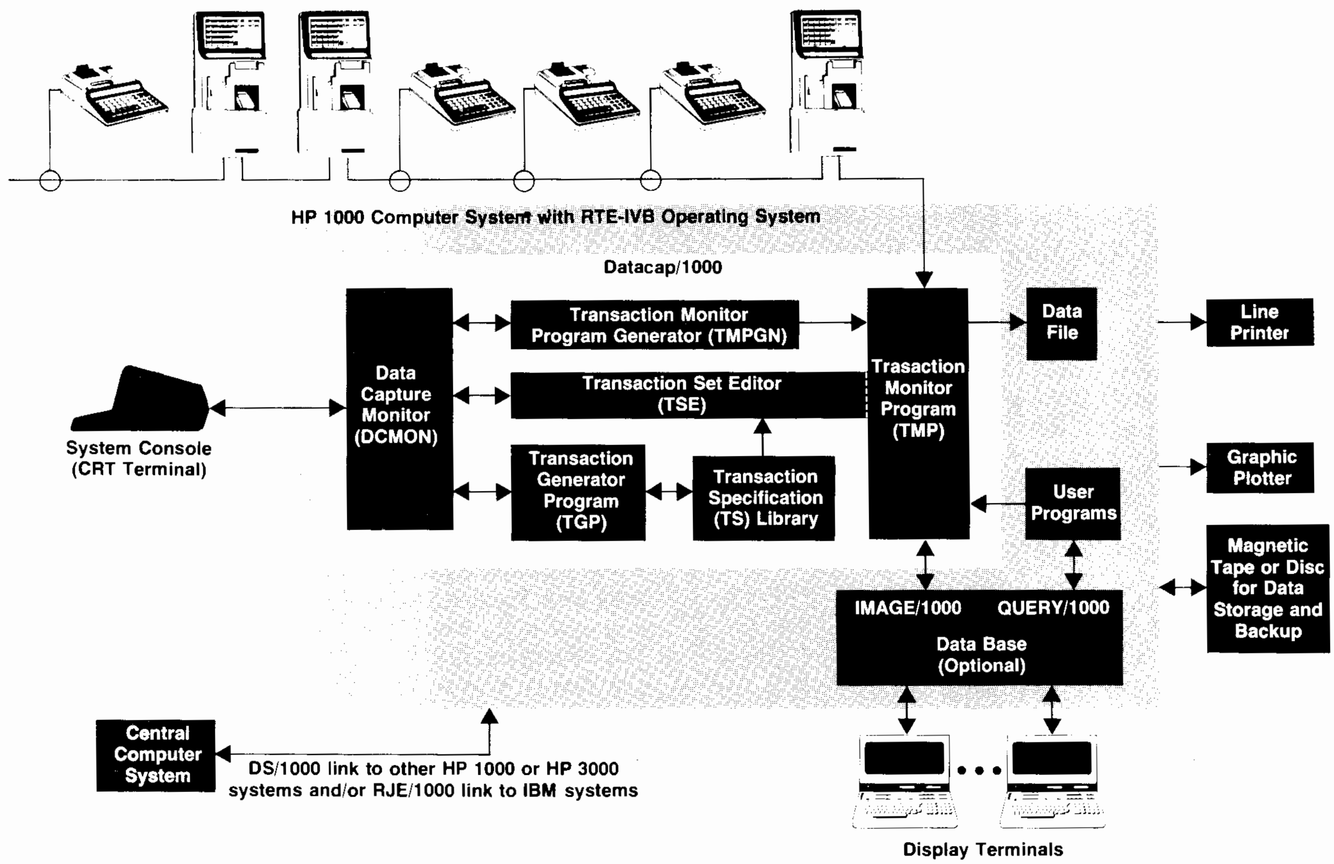


Fig. 3. *Major components of DATACAP/1000. The transaction generator program (TGP) and transaction monitor program generator (TMPGN) interactively lead the user through the process of building transaction specifications and the transaction monitor program (TMP). The TMP then manages the collection of factory data and creates data files or enters the data into the IMAGE/1000 data base, if present.*

action generator program (a sample screen in shown in Fig. 8), and each field of each screen is provided with further clarification if the HELP key is pressed.

As each screen is processed, the answers are reviewed for consistency with all previously provided answers to ensure that no illogical constructs are placed in the parameter table. Finally, after all questions have been answered, the answers are converted into the executable parameter table and written to a transaction library, a device file, or a disc file for later use by the transaction monitor program. The final operation performed by TGP is to provide the system designer with a listing of the transaction generated, including a model of the data capture terminal prompting-light label.

Fig. 9 shows the executable parameter table structure. Each executable transaction consists of a transaction header followed by up to twenty states consisting of three components each. Each component is a variable-length record with a fixed minimum length. The minimum-length component contains information about optional extra members of the component. The three components and their contents

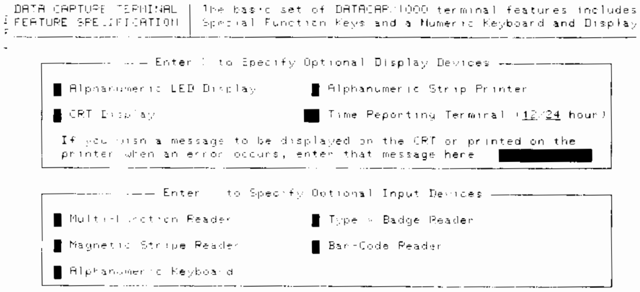


Fig. 4. *Typical transaction generator program (TGP) screen.*

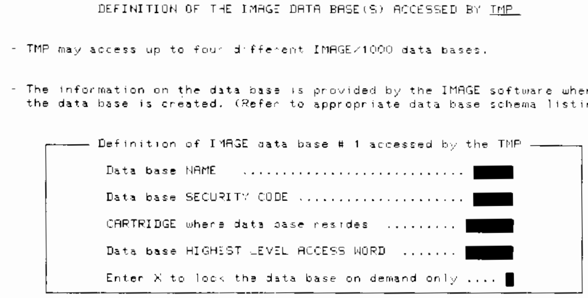


Fig. 5. *Typical transaction monitor program generator (TMPGN) screen.*

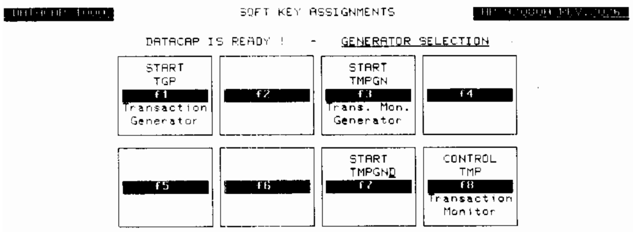


Fig. 6. *Typical data capture monitor (DCMON) screen.*

are:

State General Information
- Forward pointer to the next state
- Answer data type
- Data capture terminal configuration information
  - Prompting light
  - Alternate input device configuration
  - Display and printer use

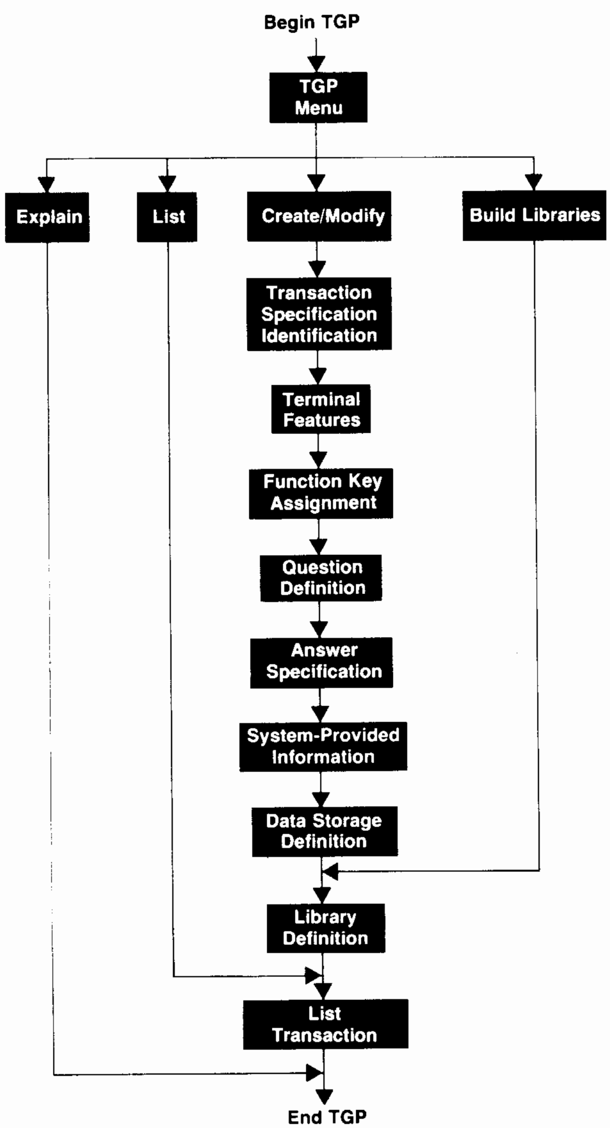


Fig. 7. *Basic transaction generator program flow.*

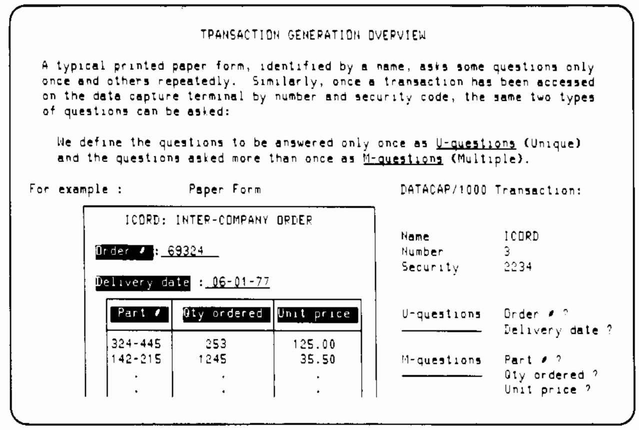


Fig. 8. *Tutorial screens like this one are built into the transaction generator program to provide the user with information about the system. A* HELP *key provides further clarification.*

Display Information
- User-written subroutine name, if any
- Display item and set name, if any
- Data capture terminal configuration information
  - Prompting light
  - Display and printer use

Edit Specification
- Range or mask depending on data type
- Default value
- User-written subroutine name, if any
- Automatically generated IMAGE edits, if any

In addition to the states consisting of these three components, there are three other states. The special function key state provides information about the data capture terminal special function keys. The storage state includes the disc file storage name, the device file storage name, the name of the user-written subroutine name, if any, and a list of IMAGE/1000 operations to be carried out, if any. The off-line printout state includes information that allows a summary of the transaction to be printed on the data capture terminal printer upon completion of the transaction.

Following this executable form of the transaction are all of the original answer buffers. This allows later modification of the transaction by TGP without uncompiling the executable form.

The Transaction Monitor Program Generator (TMPGN)

The TMPGN provides the definition of data capture terminal logical unit numbers, user provided subroutine relocatable names, and IMAGE/1000 data base names. Like TGP, TMPGN prompts the user through a set of interactive displays on the HP 2645A System Console. Again like TGP, the answers are maintained in a file for later modification of the TMP and a convenient listing is available to the system designer to review the application. Once all the answers have been provided, TMPGN generates the appropriate relocatable members of the DATACAP/1000 package and calls the RTE-IVB loader to relocate them into the structure shown in Fig. 10.

Taking advantage of the environment provided by a specially developed terminal management tool (see box,

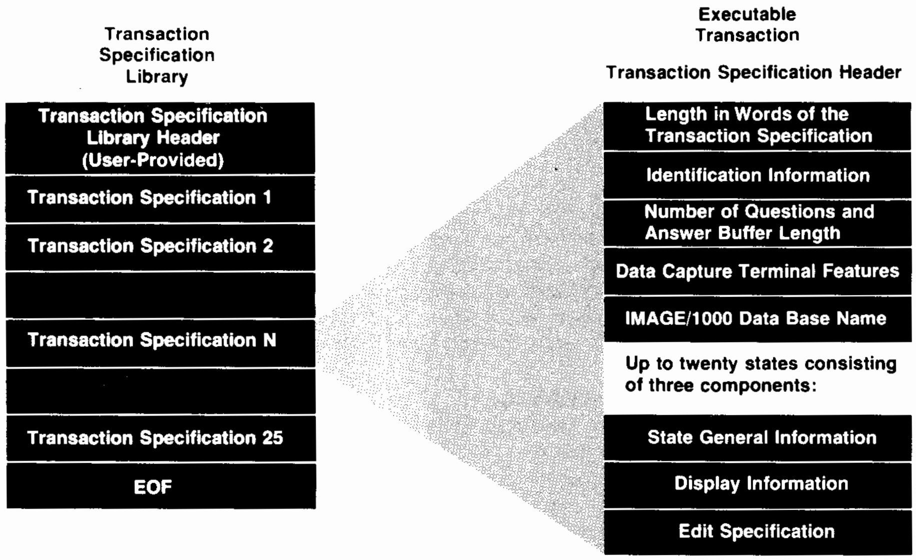


Fig. 9. *Each transaction specification in the TS library consists of a transaction header followed by up to 20 states consisting of three components each.*

page 30), each function of DATACAP/1000 can be handled by a separate subroutine:

ZTMP This module contains the main logic of the TMP. When the operator of a data capture terminal requests a specific transaction, ZTMP calls the transaction specification manager (TSMG) subroutine to return the first state of the transaction specification. ZTMP then uses information from the first state to configure the special function keys, prompting lights, input devices, and so on. Once the appropriate input has been received from the operator (as determined from the edit component of the state), ZTMP calls TSMG for the next state, and so on, progressing to the storage state upon completion of the transaction by the operator.

TSMG This module maintains the working set of transaction specifications while the TMP is running. TSMG uses its end-of-partition space as a buffer to hold several transactions, each of which can be called by any of the terminals on-line to the TMP (given that the operator knows the transaction identification number and security code). This buffer space is set at a minimum of 7500 words. Since the typical length of the executable form of the transaction is 250 words, about 30 transactions can be managed by TSMG simultaneously.

TSE The transaction set editor is used by the system operator to load and unload transaction specifications from the working set. It also can be used to check on the status of the working set (that is, which transactions are active), the transaction library from which they were loaded, and so on.

IOM7X The two modules IOM75 and IOM70 are functional drivers used by ZTMP to interface to the RTE system's terminal drivers DVR07 and DVA47, respectively. These functional drivers relieve ZTMP of the need to concern itself with the construction of escape sequences and the specifics of the two different drivers' calling sequences.

STORX The two modules STORA and STORB are used to handle the two distinct types of data storage available to the DATACAP/1000 system designer. STORA handles the passing of data and offset information to the DATACAP/1000 data base hand-

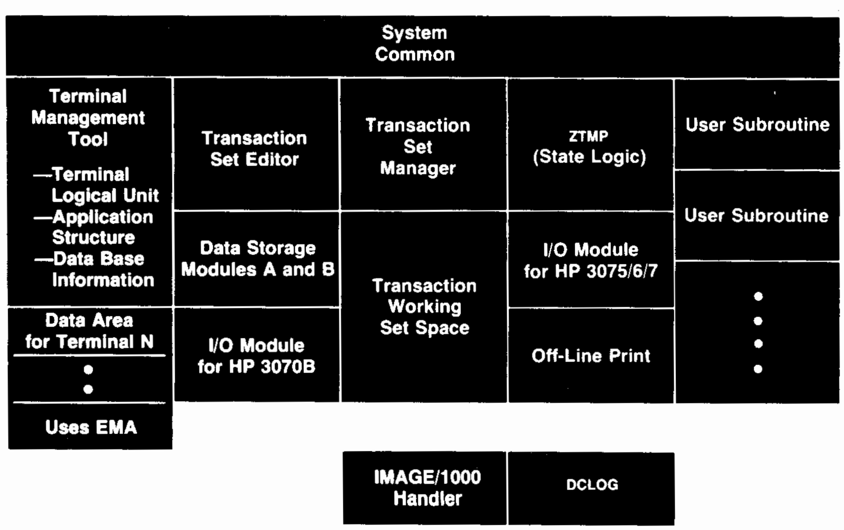


Fig. 10. *Once it has all of the answers it needs, the transaction monitor program generator generates the relocatable members of DATACAP/1000 and calls the RTE-IVB loader to relocate them into the memory structure shown here.*

A Terminal Management Tool

by Francois Gaullier

A user faced with the problem of running the same application concurrently on a number of terminals in the RTE-IVB environment may use the following techniques:

1. Write the program for one terminal, duplicate it for N terminals and run these programs concurrently. The amount of main memory available limits the number of terminals that can be supported with an acceptable response time, since swapping to the disc may be necessary.
2. Write the program for N terminals. In this case, the user must keep track of the program flow for each terminal, must access most variables through arrays, and cannot easily segment the program or spread processing among several programs. Enough memory must be provided for the maximum number of terminals even if only a few are actually being used. The development and maintenance of this type of program will be more complicated because of the bookkeeping required to keep track of each user's data.

The ideal solution to the problem of developing and maintaining large, multi-user applications is to write a single set of reentrant programs to handle all users. In a reentrant environment, applications program development and maintenance are reduced to the single-user case. Program swapping is minimized because only the users' data needs to be swapped. (Instruction spaces do not need to be swapped out, just overlaid.)

Since no facilities for large-scale reentrancy existed in RTE-IVB, a primary objective of the DATACAP/1000 project was to develop a terminal management tool to provide a reentrant environment for RTE. It was hoped that this tool would both ease the task of developing a multiterminal application like DATACAP and provide on-going benefits by reducing the amount of main memory required to support the application. DATACAP/1000 was developed using the features provided by this terminal management tool. The tool is also supplied as part of the DATACAP software, so that the final application running on a customer's system takes advantage of and depends upon the environment provided by the terminal management tool. Features of this environment are:

1. Code and data separation in FORTRAN with the capability of dynamically specifying which blocks of data are to be saved and restored when the program is reentered.
2. An easy means of spreading the processing among several program units in different partitions, that is, an easy means to do subroutine calls across partition boundaries. This eliminates the need for segmentation, and if sufficient main memory is available, lets the entire application reside in memory.
3. Recursive capabilities for specialized needs.

Programming Conventions

To use the features of the terminal management tool, the programmer must observe the following conventions:

1. I/O requests to the interactive terminals must be done through a special set of subroutines, TMLIB.
2. Code and data must be separated by declaring the latter to exist in the area of main memory designated as common.
3. I/O requests that wait for completion should be avoided whenever possible.
4. Calls to subroutines and processes that issue service requests must be done through TMLIB.
5. All programs in an application must be coded as subroutines.

The type of reentrancy provided by the tool can be termed "breakpoint reentrancy," because the programs that run in its environment can be reentered only when they issue a service request through TMLIB. Breakpoint reentrancy is quite suitable for large, multiterminal, interactive applications, because most of the execution time is spent issuing and waiting for I/O operations, which are performed by the tool's service requests.

Data Segments

The most important requirement for reentrant programming is the separation of code and data. The most convenient facility for accomplishing this in the RTE-IVB languages (FORTRAN, Assembly) is the common memory area. An added benefit of having data exist in common memory is that subroutines are provided with a very efficient means of communicating. Unlike accessing arrays, accessing variables in common is without any overhead. These two reasons are why common memory was chosen as the vehicle for separating code and data in the terminal management tool.

The primary restriction, then, for use of the terminal management tool is that a tool subroutine must assign all of its reentrant variables to common memory. Not all variables must be assigned to common memory, just those whose values are to remain intact after service requests, because it is only at these points that the subroutine can be reentered.

Since it is not necessary to preserve the integrity of all reentrant variables at all times (there is some overhead associated with this), the data segment is divided into six common blocks, which may be enabled or disabled according to the user's needs. When a common block is enabled, the data that it contains will be saved when a tool service request is issued and restored after the request completes. If a block is disabled, the integrity of its data is not preserved across the service request. By grouping the reentrant variables into common blocks and selectively enabling them, the user can reduce the overhead required by a program and the amount of main memory used at any given time. Only the data needed for the portion of code that is currently executing will be saved.

The library of subroutines, TMLIB, was designed to provide an easy interface between the user application program and the terminal management tool. TMLIB includes the following services:

- TMDFN To manage memory allocation
- TMCBL To manage memory allocation
- TMCBE To manage access of data blocks
- TMCBD To manage access of data blocks
- TMSUB To call a tool subroutine
- TMPRO To launch a son process from an executing one
- TMLUL To lock a logical unit
- TMIO To perform
- TMRD
- TMWR
- TMCTL I/O
- TMBWR
- TMBCT
- TMWRD operations
- TMPZ To suspend execution of a process for a given period of time
- TMSCH To schedule a non-tool program
- TMCST To place a process in critical/normal state
- TMSTP To stop the application in an orderly way

- TMSAB To abort the application

IMAGE/1000 Access

To access an IMAGE/1000 data base with the full capabilties (GET, ADD, DELETE, UPDATE) the programmer must open it in mode 3 (exclusively) or in mode 1 (shared read/write) and lock it, which prevents other programs from concurrently accessing (writing to) the data base. Under these conditions one way to have multiprogram access to a data base is to centralize all requests to IMAGE in a single program. IMAGE/1000 access through the tool uses this concept.

Given the above feature, the programmer is faced with the difficulties of managing all the different operations that may be performed simultaneously on the data base by all the programs (update of the same entry by more than one program at the same time, deletion of a chain entry while another program is going through the chain, etc.). One way to solve these problems is to implement a locking mechanism. The most sophisticated locking mechanism is to lock at the entry level in the IMAGE data base. The terminal management tool offers this capability to the user.

The tool also automatically performs the DBINF calls to save and restore the run table when necessary for each process, thereby saving the user the worry of keeping track of the run table when repeatedly accessing the data base (chain or serial read).

All calling sequences to the DATACAP/1000 IMAGE handler implemented using the terminal management tool are compatible with the IMAGE/1000 calling sequences with the addition of a single-word parameter to the calls analogous to DBFND and DBGET, which allows specification of the record locking action to be taken. These calls are accessible by the user through a library of routines, %XMLIM, whose members are the same as full IMAGE/1000 DBMS calls (DBYYY(parameters)) but whose entry points are XBYYY(parameters). To keep the demand of the terminal management tool IMAGE/1000 handler on system resources to a reasonable level, the following limitations are imposed on the user:

1. Calls for locking and unlocking the data base (DBLCK and DBUNL) are not provided.
2. Positions in a data set (current record, chain, etc.) are not remembered when a different data set is accessed. It is necessary for the user to save the run table for a data set before accessing a second data set, then restore the run table for the first data set to access it starting at its current position.
3. The maximum entry length allowed is 512 words instead of 2048 words.
4. The maximum number of data bases that can be opened simultaneously is eight (combined total in DATACAP/1000).
5. All data bases that are to be accessed must be defined in TMPGN.

Francois Gaullier

Francois Gaullier received his diploma—equivalent to a master's degree—in 1971 from Ecole Speciale de Mecanique et d'Electricite in Paris. He joined HP that same year as a systems engineer. In 1975 he became a software design engineer, developing RTE drivers and modules of DATACAP, and he's now a lab project manager at HP France in Grenoble. Francois was born in Paris and now lives in St. Egreve near Grenoble. He's married and has two daughters. He enjoys hiking, skiing, and listening to classical and country music, and he recently designed and built a microprocessor-controlled model railroad.

ler, while STORB is responsible for generalized data storage to disc files, device files (magnetic tape), or user storage subroutines.

IMAGE/1000 Handler This module provides a front end to the standard IMAGE/1000 DBMS, specifically establishing a record locking mechanism and a link to the DATACAP/1000 transaction logging module.

DCLOG provides a means for writing standard, recoverable records and prevents the terminal user from proceeding with a second transaction until the first is safely logged to a disc or device file. DCLOG is complemented by the recovery utility DCRCV that can be used to recover an IMAGE/1000 data base that has been corrupted.

Acknowledgments

The DATACAP/1000 product was conceived at Hewlett-Packard's division in Grenoble, France in 1975 and was originally released for sale from Hewlett-Packard's Data Systems Division in 1978. It has undergone several revisions to incorporate new data capture terminals, revisions to the RTE operating system and IMAGE/1000 DBMS, and flexibility enhancements. The design team has included Marc Brun, Elizabeth Clark, Francois Gaullier, Scott Gulland, Ben Heilbronn, Tom Hirata, Jean-Charles Miard, Miles Nakamura, Daniel Pot, and Steve Witten. Production personnel have included Chengwen Chen, Ron Schloss, and Alice Woo. Product assurance was the responsibility of Clemen Jue, Julie Knox, Doug Larson, and Elizabeth Gates.

Steven H. Richard

Steve Richard graduated from Stanford University in 1969 with BS and MS degrees in industrial engineering, and joined HP the same year. After four years of production engineering, followed by four more years of programming and EDP management, he became project manager for DATACAP/1000, a post he still holds. Steve is a member of AIIE and ACM and is working towards his MS degree in computer science at the University of Santa Clara. An Arizona native, he is married, has two sons, and lives in Los Altos, California. His interests include church activities, bicycling, downhill and cross-country skiing, sailing, and cooking breakfast.

SPECIFICATIONS

HP Model 92080A DATACAP/1000 Data Capture Software

MAXIMUM NUMBER OF TERMINALS: An HP 1000 Computer or System can support up to 56 HP data capture terminals. The number of terminals supportable by DATACAP/1000 depends upon the number of transactions per hour to be processed, i.e., response time required, on transaction complexity, such as IMAGE/1000 data base activity, user subroutines, etc., and on system main memory size.

COMMUNICATION WITH IMAGE/1000 DATA BASE: Transaction Monitor Program (TMP) can retrieve data from, and record data in up to four IMAGE/1000 data bases. No transaction specification can access more than one IMAGE/1000 data base. Any user program has read/write access to a data base that is under DATACAP control.

COMPATIBILITY WITH SESSION MONITOR: DATACAP/1000 does not use or require the Session Monitor in RTE-IVB and in certain respects may not be compatible with the Session Monitor. Where multi-user access to the RTE-IVB system is required concurrently with DATACAP/1000 operation, the Multi-Terminal Monitor is recommended instead of the Session Monitor. The DATACAP Configuration Guide (92080-90003) documents the particulars regarding DATACAP/1000 compatibility with Session Monitor.

DATA CAPTURE SYSTEM REQUIREMENTS: HP 92080A DATACAP/1000 is supported only on the 92068A RTE-IVB operating system, and therefore has the same minimum system requirements as the 92068A, plus additional requirements listed below.

DATA BASE SOFTWARE: If communication with an IMAGE/1000 data base is desired, 92069A IMAGE/1000 is required.

DATA CAPTURE TERMINALS: At least one (and up to 56) of any of the HP 3075A, 3076A, or 3077A terminals.

MAGNETIC TAPE: A magnetic tape unit is strongly recommended if IMAGE/1000 is used.

SYSTEM CONSOLE: HP 2645A, 2647A, 2648A, 2649B, 2649C, or 2649G system console on which DATACAP/1000 is readied for use must connect to the system via the 12966A+001 interface and must have the following options and accessories.

OPTION 007: Mini cartridge I/O (not required on 2647A/49G).

OPTION 032: Substitutes Extended Async Comm. Card for standard comm. card.

13231A Display enhancements with line drawing character set.

FAST FORTRAN PROCESSOR FIRMWARE: 12977B or 13306A, depending on computer model.

RTE-IVB SYSTEM MEMORY REQUIREMENTS: The RTE-IVB system configured to support DATACAP/1000 has the following memory requirements.

SYSTEM CODE AND TABLES: 56K bytes.

SYSTEM AVAILABLE MEMORY: 12K bytes.

SYSTEM COMMON FOR DATACAP USE: 4K bytes.

DATACAP/1000 MEMORY PARTITION REQUIREMENTS: The following partitions are required for DATACAP/1000:

DATACAP/1000 APPLICATION CODE: 122K bytes.

BUFFERING FOR DATA CAPTURE TERMINALS: 54K-40K bytes (see Table 1), using RTE-IVB mother partition.

TABLE 1. Memory buffering requirements by number of Data Capture Terminals

No. of Terminals	Buffer Size	No. of Terminals	Buffer Size	No. of Terminals	Buffer Size
1- 5	54Kb	21-25	194Kb	41-45	334Kb
6-10	89Kb	26-30	229Kb	46-50	369Kb
11-15	124Kb	31-35	264Kb	51-56	404Kb
16-20	159Kb	36-40	299Kb		

OPTIONAL MEMORY PARTITIONS: The following partitions are required for optional DATACAP/1000 capabilities.

EACH IMAGE/1000 DATA BASE ACCESSED: 56K bytes.

FOR INTERFACING WITH USER SUBROUTINES: 10K bytes, variable, for the user subroutines.

TRANSACTION GENERATOR PROGRAM: 40K bytes, to avoid swapping if creating transactions concurrently with real-time data capture operations.

DS/1000 NETWORK COMMUNICATIONS: 6-25K bytes, depending upon remote access capabilities supported.

PRICES IN U.S.A.:

92080A DATACAP/1000 Data Capture Software Package, $5000.

92080R Right to Copy DATACAP/1000 for use on an additional computer system, $2000.

MANUFACTURING DIVISION: DATA SYSTEMS DIVISION
11000 Wolfe Road
Cupertino, California 95014 U.S.A.

Hewlett-Packard Company, 1501 Page Mill Road, Palo Alto, California 94304

Bulk Rate
U.S. Postage
Paid
Hewlett-Packard
Company

HEWLETT-PACKARD JOURNAL

CHANGE OF ADDRESS: To change your address or delete your name from our mailing list please send us your old address label. Send changes to Hewlett-Packard Journal, 1501 Page Mill Road, Palo Alto, California 94304 U.S.A. Allow 60 days.