HEWLETT·PACKARD JOURNAL

HEWLETT-PACKARD JOURNAL

Technical Information from the Laboratories of Hewlett-Packard Company

Contents:

APRIL 1981 Volume 32 ● Number 4

An Interactive Material Planning and Control System for Manufacturing Companies, *by Nancy C. Federman and Robert M. Steiner*   *Complex manufacturing operations are aided by this software package, which is fully supported by HP even after the user modifies it.* — 3

A Novel Approach to Computer Application System Design and Implementation, *by Loretta E. Winston*   *The Application Customizer helps designers construct generalized application systems and gives users tools to tailor these systems to their own needs.* — 13

Automating Application System Operation and Control, *by Barry D. Kurtz*   *The Application Monitor schedules, initiates, and controls all interactive and background activities in an application system.* — 19

Precision DVM Has Wide Dynamic Range and High Systems Speed, *by Lawrence T. Jones, James J. Ressmeyer, and Charles A. Clark*   *Many bench and system voltage and resistance measurements are made easier by this versatile digital voltmeter.* — 23

In this Issue:

Without computers, much of our modern society would cease to function. Most of us wouldn't receive our paychecks—or bills, for that matter—and governments and corporations would choke on the vast amount of information they must process to survive.

For every application in which a computer is effective, a programmer or group of programmers has had to write an application program, a detailed set of instructions that tells the computer how to handle every conceivable situation. These programs are also called application software packages or application systems. The subject of this month's cover and the article on page 3 is an application system that runs on an HP 3000 Computer. Called Materials Management/3000, it's designed to help manufacturers improve productivity as they deal with the challenges of cost pressures, shipment schedules, and production line efficiency. For example, this system helps optimize a company's investment in inventory—too little leads to late shipments and unhappy customers; too much means higher carrying costs and money tied up unproductively.

Although more and more business operations are being put under the control of computer application systems, installing these systems isn't always a routine job. All too common are stories of how this firm or that has installed a computer and then gone back to a manual system after months of frustration. Either the application system was badly designed, or it couldn't be adapted to a particular firm's operations, or the using firm found that it needed a staff of programmers to modify the system and maintain it on a day-to-day basis. In many cases the costs can exceed the benefits.

Materials Management/3000 is innovative because it's designed to be tailored easily to a particular company's operations without a staff of computer professionals. What's more, after the system is customized by the using manufacturer, HP can and will continue to maintain and support it. This is possible because the basic program instructions are kept separate from the customizable parts, so they don't change when the system is customized. Two special software utilities called the Application Customizer (page 13) and the Application Monitor (page 19) are largely responsible for these benefits. They're parts of Materials Management/3000, and they'll also be used by HP designers to create new application systems to help companies operate more productively.

On page 23 you'll find an article about a new high-performance digital voltmeter, Model 3456A. It's characteristic of digital voltmeters that they take a long time when they're making relatively precise voltage measurements, say six digits, but they're much faster on less precise measurements, say three or four digits. The 3456A's designers have developed a new measurement technique that gives more precision in less time than previous designs. They've given it a name—Multi-Slope II. It'll give you 330 four-digit readings per second or 48 six-digit readings per second. The 3456A is accurate enough to be considered a transfer standard, with accuracy traceable to the United States National Bureau of Standards. But it's not just a laboratory instrument, it's also designed for use in automatic computer-controlled systems. It can accept, remember, and execute a measurement program loaded into it by a computer and signal the computer when it's ready with the result. There are many other nice features in the 3456A, such as the ability to take measurements from a thermistor and display temperatures in degrees, and compensation for voltage offsets in resistance measurements. Read the article for the full design story.

-R. P. Dolan

Editor, Richard P. Dolan ● Associate Editor, Kenneth A. Shaw ● Art Director, Photographer, Arvid A. Danielson
Illustrator, Nancy S. Vanderbloom ● Administrative Services, Typography, Anne S. LoPresti ● European Production Manager, Dick Leeksma

© Hewlett-Packard Company 1981 Printed in U.S.A.

An Interactive Material Planning and Control System for Manufacturing Companies

Drawing on HP's own experience, this powerful software for the HP 3000 Computer makes it easier to deal with the complexities of operating a manufacturing company.

by Nancy C. Federman and Robert M. Steiner

S O YOU WANT TO BE AN ENTREPRENEUR. Let's assume that you have already invented that hot item and you have procured sufficient venture capital. Do you really know what is involved in manufacturing a product? The first step is to specify the engineering data. What is the structure of the product? You need to determine the parts and subassemblies that go into the product and the quantities of each required per unit. Which of these components will you manufacture? Which will you subcontract to other manufacturing companies, and which will you purchase?

Now that you have some of the engineering specifications the next step is to determine the manufacturing procedure. How is the product to be made? You will need to decide the manufacturing operations necessary to construct the product and then specify the locations in your shop where the work will be done (i.e., the workstations). The sequence of these production steps, called the routing, plus the labor and material required in each operation, must be detailed.

The product is specified and the manufacturing plant and process are established. You are now ready to begin production. How many products should you build? You need to consider customer demand, current and forecast, as well as the capacity constraints of the factory. Once your production plan is established you need to determine your material requirements. How many component parts do you need to meet your production schedule? You also need to determine the schedule of production for the manufactured components. When do the purchased parts have to be ordered? You need to balance the desire to have components always available with the economic necessity of keeping inventory levels as low as possible.

Controls are needed to monitor the flow of materials in the stockroom and on the factory floor. When do the component parts need to be issued to the production lines and what quantities are required? You will want to monitor the shop floor and track the work in process. How much material is wasted? How efficient is your work force? When the products are finally completed you need to keep them in a finished goods inventory and track their sales.

Finally, you will be concerned with calculating the costs—labor, material and overhead. The production costs will help you decide your pricing policies and determine your profitability.

From this brief and simplistic overview of a manufacturing operation it should be clear that there are many intricate relationships to deal with. Effective and efficient management of a manufacturing operation is difficult to achieve, and many neighborhood businesses as well as sophisticated manufacturing companies are turning to computers for help.

Materials Management/3000

Hewlett-Packard is among the vendors providing application systems for manufacturing management—a manufacturing company offering a solution to the problems of manufacturing companies. Materials Management/3000 is an interactive material requirements planning and control system designed to make it easier to deal with the complexities of operating a manufacturing company. It is primarily designed for manufacturers who build standard products to stock in discrete manufacturing steps (fabricators and assemblers). These companies have a significant investment in inventory. Materials Management/3000 can help them balance their inventory levels with customer demands for timely shipments to optimize their dollar investments.

The complexity of the manufacturing environment can also be seen in the internal complexity of the software product that provides solutions to manufacturing problems. Materials Management/3000 consists of over 400,000 lines of SPL* code which make up 161 transactions using nine data bases. There are 291 screens, both transaction screens and menu screens. There are also 168 batch programs. The application data bases and screens can be customized by the user. Materials Management/3000 operates on any of the HP 3000 family of computers and uses the 26XX family of terminals. The system uses HP's IMAGE data base management system and V/3000 screen handler.

Materials Management/3000 consists of ten major modules (see Fig. 1):

- Master Production Scheduling: MPS is an on-line management planning and production scheduling tool. It is used by the master scheduler to generate a production schedule for the plant's marketable products and to set the mix for the product options. Input to the module includes current customer orders, forecast customer or-

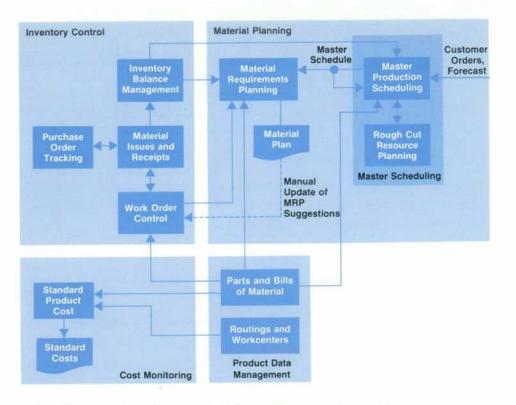*SPL is the HP 3000 Systems Programming Language

Fig. 1. Materials Management/3000 consists of ten major modules. It is primarily designed for manufacturers who build standard products in discrete manufacturing steps.

ders, the current production schedule, and the current level of product inventory. The output of the MPS calculations is called the master production schedule. This schedule contains suggested manufacturing orders including quantities and starting dates. The schedule is then input to the Materials Requirements Planning (MRP) module to plan the manufacture and purchase of the required component parts. MPS also includes a "what if" simulation capability which allows the user to generate tentative schedules and view the impact of modifications on the current schedule (Fig. 2).

■ Rough Cut Resource Planning: Rough Cut Resource Planning (RRP) is a management tool used by the master scheduler to help produce a realistic master schedule by

Fig. 2. This is a sample report from the Master Production Scheduling module. It compares product demand (actual and forecast orders) to the current production schedule (in units, dollars, and labor hours). The effect on inventory is noted. The report also compares scheduled production to factory capacity and flags (OL) production periods that are overloaded. Production schedules for individual parts can also be viewed on-line.

```
        PART NUMBER:  07              DESCRIPTION:  LEVEL01                 CONTROLLER:  39

--------------------------------------------- CRITICAL RESOURCE --------------------------------------

        RESOURCE ID: MACHINEA                   DESCRIPTION: FORK LIFT

        UNTS OF MEASURE: HI       CAPACITY PER DAY:   50      UNIT REQUIREMENTS:    0     SHOP CALENDAR DEPENDENT? Y

--------------------------------------------- PART REQUIREMENTS ---------------------------------------

RESOURCE DUE DATE:  02/04/81  02/11/81  02/18/81  02/25/81  03/04/81  03/11/81  03/18/81  03/25/81  04/01/81
WORKING DAYS:            5         5         5         5         5         5         3         1         5
MAXIMUM CAPACITY:      250       250       250       250       250       250       150        50       250
TOTAL REQUIREMENTS:     96       192       288       288       192        96       192       192       288
PERCENT USAGE:          38        76       115*      115*       76        38       128*      384*      115*

RESOURCE DUE DATE:  04/08/81  04/15/81  04/22/81  04/29/81  05/06/81  05/13/81  05/20/81  05/27/81  06/03/81
WORKING DAYS:            5         5         5         5         5         5         5         5         5
MAXIMUM CAPACITY:      250       250       250       250       250       250       250       250       250
TOTAL REQUIREMENTS:    192        96        96       192       192        96        96       192       192
PERCENT USAGE:          76        38        38        76        76        38        38        76        76

RESOURCE DUE DATE:  06/10/81  06/17/81
WORKING DAYS:            5         5
MAXIMUM CAPACITY:      250       250
TOTAL REQUIREMENTS:     96         0
PERCENT USAGE:          38         0
```

Fig. 3. *A sample report of the Rough Cut Resource Planning module. This report compares the current scheduled production plan to the available production capacity.*

comparing the resources needed to implement the master schedule with the available critical resources. The user specifies the critical resource requirements for each master schedule product and the maximum capacity of these resources. Examples of critical resources are labor hours, floor space, investment in work-in-process inventory, and material supplies. The RRP reports highlight the capacity constraints and help the user resolve competing demands for critical resources. An on-line RRP report can also be used to help evaluate simulated master schedules by comparing their resource requirements to the requirements of the current schedule (Fig. 3).

■ Material Requirements Planning: MRP simulates the complex flow of materials required to manufacture products and generates a material plan. MRP planning starts with up-to-date information about current inventory levels and the planned production requirements. Using part and bill-of-material information the material requirements for each part are calculated. The plan starts with the highest-level assemblies and proceeds through the lowest-level parts. MRP will reschedule current work and purchase orders and suggest new orders as necessary to meet the demand. MRP is a regenerative system; a complete material plan is generated every time MRP is run (Fig. 4).

■ Parts and Bills of Material: This module provides on-line maintenance of engineering, accounting, and planning information about each part and product, and information on how the parts relate to one another to form the product structure (bill of material). Responsibility for maintaining this data will normally be shared among several departments—accounting, engineering specifications, and planning. Part and structure data can be reviewed on-line or through printed reports. The part and structure data is used by many of the other modules in Materials Management/3000, including MPS and MRP (Fig. 5).

■ Routings and Workcenters: The bill of material defines the parts and subassemblies that go into a product but doesn't document how the various components are assembled. The Routings and Workcenters module maintains information that describes the locations where the

```
10          LEVEL02                                                      ORDER POLICY:  LOT FOR LOT
     3536 QTY ON HAND      0 QTY IN INSPECTION    LEAD TIME:      13      6 MONTH REQUIREMENT:      10110
        0 SAFETY STOCK  3500 TOTAL AVAILABLE   SAFETY LEAD TIME:  10      UNIT COST:            473.7900
   --DUE-DATE--  --PROJ-AVAIL-- ---QTY---  -----PARENT-----  ----DEMAND-----  -----SUPPLY----- --ITEM- ---ACTION--- =START-DATE-
     12/29/80*        2489      -1011  02                    ALLOC 23202
     01/19/81*        1478      -1011  02                    ALLOC 23203
     01/28/81*         467      -1011  02                    ALLOC 23204
     02/05/81*        -544      -1011  02                    ALLOC 23205
                       -33        511                                          PURCH 39120005    1                  01/19/81*
                       978       1011                                          PURCH 39120006    1
     02/17/81*         -33      -1011  02                    ALLOC 23206       PURCH 39220001    1                  01/29/81*
                       978       1011
     02/23/81*         -33      -1011  02                    ALLOC 23207       PURCH 39220002    1     PULL IN      02/04/81*
                       978       1011                                                                  CANCELLED?
     03/02/81*         -33      -1011  02                    EXPLO 23208       PURCH 39220003    1     PULL IN      02/11/81*
                       978       1011                                                                  CANCELLED?
     03/10/81*         -33      -1011  02                    EXPLO 23209       PURCH 39320012    1     PULL IN      02/19/81*
                         0         33                                                                  CANCELLED?
     03/20/81        -1011      -1011  02                    EXPLO 23210       PURCH 39320013    1     PULL IN      03/03/81*
                         0       1011                                                                  CANCELLED?
     04/06/81        -1011      -1011  02                    EXPLO 23211       PURCH 39320014    1                  03/10/81*
                         0       1011

11          LEVEL02                                                      ORDER POLICY:  LOT FOR LOT
        0 QTY ON HAND      0 QTY IN INSPECTION    LEAD TIME:       9      6 MONTH REQUIREMENT:          0
        0 SAFETY STOCK     0 TOTAL AVAILABLE   SAFETY LEAD TIME:   0      UNIT COST:          10306.3487
   --DUE-DATE----  --PROJ-AVAIL-- ---QTY---  -----PARENT-----  ----DEMAND-----  -----SUPPLY----- --ITEM- ---ACTION--- -START-DATE-
     12/24/80*         270        270                                          RUN   00000096       FIRM   12/11/80*
     01/28/81*         540        270                                          RUN   00000097       FIRM   01/15/81*
     01/04/81*         810        270                                          RUN   00000099       FIRM   02/19/81*
     03/31/81         1080        270                                          RUN   00000100       FIRM   03/10/81*
     04/15/81         1350        270                                          RUN   00000063       FIRM   04/02/81
     05/13/81         1620        270                                          RUN   00000064       FIRM   04/30/81

12          LEVEL02                                                      ORDER POLICY:  LOT FOR LOT
        0 QTY ON HAND      0 QTY IN INSPECTION    LEAD TIME:      15      6 MONTH REQUIREMENT:       3220
        0 SAFETY STOCK     0 TOTAL AVAILABLE   SAFETY LEAD TIME:  15      UNIT COST:              1.7500
   --DUE-DATE----  --PROJ-AVAIL-- ---QTY---  -----PARENT-----  ----DEMAND-----  -----SUPPLY----- --ITEM- ---ACTION--- -START-DATE-
     01/15/81        -520        -520  03                    EXPLO 39010001
                    -1420        -900  03                    EXPLO 39010002
                     -520         900                                          WKORD 11202                PULL IN      03/17/81
                      380         900                                          PURCH 39120011    1       PULL IN
     04/22/81        -520        -520  03                    EXPLO 39010003
                    -1420        -900  03                    EXPLO 39010004
                     -900         520                                          SUGG  39020001            W/I WINDOW   04/01/81
                        0         900                                          SUGG  39020002            W/I WINDOW

                  * = PAST DUE   + = PAST START DATE
```

Fig. 4. *A sample action report produced by the Material Requirements Planning module.*

Fig. 5. A sample on-line review of the bill of material for a part.

products are made (workcenters) and the proper sequence of manufacturing steps (routings). This information is used to generate cost information for the Standard Product Cost module, and to help develop detailed production schedules. Responsibility for this data usually resides with manufacturing specifications personnel.

■ Standard Product Cost: SPC provides manufacturers with the capability of accurately calculating the standard costs associated with the manufacture of each product. All current cost information may be edited and reviewed on-line. The standard cost of a product is determined by accumulating all relevant material, labor, and overhead costs for the components of the product as well as the costs associated with the construction of the finished product. These standards can be used to determine product pricing and profitability. Marketing personnel and the material manager use this data.

■ Material Issues and Receipts: This module helps control stockroom inventory by maintaining timely and accurate records of all actions that affect inventory balances. The data includes receipts of work orders or purchase orders, material issues from stock to a particular work order, filling of a back order, or an unplanned issue. All record keeping and updating is done on-line and a record of all inventory activity is kept on-line for a user-specified period of time as an audit trail (Fig. 6). Stock room personnel are the primary users of this system.

■ Inventory Balance Management: The Inventory Balance Management module maintains information about inventory balances and the warehouse locations where the inventory is stored. The current inventory status can be affected by three types of transactions: material movement, inventory counts, and stock adjustments. All three types of transactions will trigger an immediate update of the inventory counts as well as create an audit trail record. All updates are done automatically in an on-line mode. Current inventory balance data from this module is used by MPS and MRP to determine the next master schedule and the next material plan. All activity that affects inventory status can be reviewed on-line. An inventory value report is also available. Materials Management/3000 also allows for multiple stock locations; a separate on-hand balance can be maintained for each stock location in each warehouse. This system also helps with the counting of inventory, which is periodically used to verify inventory totals.

■ Work Order Control: A work order is an internal factory authorization to build a specified quantity of a particular subassembly by a specified date. All work orders require the issue of on-hand inventory for their completion. Prior reservation of on-hand inventory is the best method of preventing shortages at the time of issue. Allocation, or logical reservation of on-hand inventory helps predict and prevent these shortages. The timely notification of exceptions to the material plan can allow corrective action before the results become disastrous. The outputs of this tracking system are reports noting exception conditions. The materials manager can act on these reports. The actual issuing of parts and work orders, and the actual receipt of finished products is accomplished using the Material Issues and Receipts module. MRP is a prime user of information from this system.

■ Purchase Order Tracking: A purchase order represents a scheduled receipt for purchased items. Entering a purchase order requires the entry of more information than that required on a work order—for example, vendor information, shipping information, and price information. It is also possible to group multiple delivery dates and/or multiple parts on the same purchase order. The Purchase Order Tracking module monitors these scheduled receipts and maintains vendor information. Users can get a report on current orders for a particular vendor or the value of outstanding purchases by scheduled receipt date. The purchasing department and the materials manager normally use this module.

What distinguishes Materials Management/3000 from other materials management systems is not just its features but the design and implementation philosophy behind it. This philosophy evolved from previous experience with application systems, a knowledge of the competitive situation, and first-hand experience with manufacturing company operations. The design philosophy can be summarized as providing a functionally complete solution that fits the business practices of the user, is friendly and easy to use, and is supportable by HP. As the philosophy evolved into more concrete design goals the project team realized that certain functional capabilities were required but were unavailable in the current set of tools provided on the HP 3000. The team realized that it was not possible to anticipate all the detailed and distinctive capabilities peculiar to any particular customer. The implementation strategy was to include standard and accepted functions in the program code and provide software tools to allow customers to tailor the system to fit their own requirements. The application designers thus precipitated the development of the required new tools, the Application Customizer and the Application Monitor described in the articles on pages 13 and 19.

Fig. 6. An on-line record of inventory activity produced by the Material Issues and Receipts Module.

Design Principles

The development of Materials Management/3000 was guided by a set of design principles that can be grouped into four categories: contribution, simplicity, versatility, and supportability.

Contribution. All product teams strive to meet HP corporate goals by producing products that are contributions in the marketplace and provide lasting value to the customer. Materials Management/3000 was designed to provide a contribution unusual in the industry—customizability with full vendor support. With other application system products, customers must face the dilemma of accepting a standard package and adjusting their business practices, or modifying the application source code to better suit their businesses and losing the support and enhancements available from the vendor (or paying extra for it). Materials Management/3000 eliminates the need to choose between function and support. The user can customize the application system without jeopardizing the ability to get full support from HP and to receive factory enhancements and upgrades.

Simplicity. Materials Management/3000 embodies simple sophistication. The product provides a full set of functions, yet is friendly and easy to use. It represents a major step towards producing a manufacturing application system that requires no data processing staff and assumes the user has no knowledge of computers.

The user interacts with the product using functional terminology, not computerese. Materials Management/3000 is menu-driven—the user simply selects the desired function from a list of choices. On-line help screens assist the novice user in stepping through the functions. User-oriented documentation, organized by functional area, is provided. Skilled users are not frustrated, however, since they can traverse the screen hierarchies at will simply by entering the name of the desired screen in the command window, a special input field for control functions. They can even restructure the entire screen sequence to better correspond to their operations.

The application control functions are automated (e.g., initiating batch jobs). Instead of the MPE operating system, the user can interact with a monitor that is optimized for non-computer professionals.

Supportability. A major design goal was to develop a product that could be tailored by the user and still be supported by HP. This goal was achieved by separating the application code, which contains the standard product, from the parameters that characterize each particular instance of the application. The parameters are maintained in tables by the Application Customizer. This design allows the user to alter the application without modifying the application code. All customers, then, have the same program code installed, so HP has only one version of the product to support, regardless of how different each installation may appear.

Versatility. Materials Management/3000 was designed to have functional generality. It can be tailored to fit each installation. Users can specify the parameters that characterize their business practices, such as the order policy and the length of the planning horizon, and can specify the parameters that indicate how they will implement the computer solution: the terminal configuration, user passwords, and capabilities. Customers can also modify the face the application presents to the user. Field labels, screen names, screen contents, even the screen sequence can be modified, and the user can add fields to screens and data bases. The product is also versatile enough to allow customers to use the product as installed, without requiring customization. This versatility implies that the application will be easy to use for the novice, yet rich and sophisticated enough for the mature user.

Implementation

After determining the design goals for Materials Management/3000, the design team turned to the problem of implementing an application that would meet the criteria established during the design phase, yet would be efficient enough to provide good response time during heavy use. Some of the tools were already available. IMAGE/3000, the data-base subsystem, eliminates data redundancy and resulting maintenance problems. V/3000, the forms data entry subsystem, made it easy to design and implement a friendly, consistent user interface. The MPE message system provides a facility for creating customizable report headings and user error messages.

To meet the ease-of-use objectives it was necessary to develop two more tools, the Application Customizer (page 13) and the Application Monitor (page 19). The Customizer provides a method for the customer to tailor Materials Management/3000 to fit an individual environment, and the Monitor automates many of the day-to-day administrative functions usually performed by an operations staff. The Monitor accomplishes its function by starting and stopping terminals at predetermined times and scheduling background jobs such as MRP to be run on a regular basis. System security is controllable because users may not use the application (i.e., Materials Management/3000) unless the system administrator has instructed the Monitor to start the application on a specific terminal. The Monitor also includes review capability of application-generated error messages and other system activity, such as the background job schedule or current terminal activity. To the application program, the Monitor provides many services normally associated with operating systems. The application programs may request services such as process initiation, inter-process communication, and resource allocation for on-line terminals and printers. The application designer can concentrate on solving application-oriented problems and call on the Monitor to provide other functions that are necessary but not directly involved with materials management functions.

The key component of a customizable application is the application data dictionary, which serves as a repository for application-dependent information such as data item characteristics, data base schemas, V/3000 form descriptions, security passwords, terminal configurations and background job schedules. The Application Customizer was designed to maintain the data dictionary, and it performs two major functions. The first is a facility for customers to alter or customize the application system using a simple menu-driven fill-in-the-blanks sequence of forms. Since this is the part of the Customizer most visible to the

customer, the bulk of the design effort went into making customization functions simple and easy to understand by nonprogrammers. The second function performed by the Customizer is to transform the information present in the application data dictionary from data structures suitable for efficient on-line update to structures suitable for run-time access by the application programs. These transformed data structures, collectively known as the run-time application data dictionary, are used by the application programs to determine the values of all customizable parameters in the system.

Fig. 7 shows how the Customizer, the Monitor, IMAGE/3000, V/3000 and the application software interact.

Design Methods

The rest of this article describes some of the methods used by the designers of Materials Management/3000 to design programs that can operate efficiently in a customizable environment. Because Materials Management/3000 is a customizable application, the customer has the ability to change many of the characteristics of the system by modifying items in the application data dictionary, rather than using the time-consuming and error-prone traditional method of modifying source code and compiling programs. Designing customizable applications is therefore complicated by the fact that many assumptions traditionally made by application programmers are not true. Customers may modify data item characteristics, add and delete items, modify the on-line user interface, and define additional processing.

Changing Data Item Characteristics

An assumption traditionally made is that once a data item is defined, its characteristics will not change. In a customizable environment that assumption is no longer valid. Because it is possible for the customer to alter the length, type and precision of any field, the application program has no idea what the characteristics of fields will be until the program is executing. For example, there are three broad categories of data type used by Materials Management/3000: alphanumeric strings, numeric fields, and date fields. An application designer may assume a data item is one of

these three general types, but cannot know the specific format of the field. Numeric fields may be any of five numeric data types: display numeric (with explicit sign and decimal point), zoned numeric (with implicit decimal point and sign overpunch), packed decimal, 16-bit integer, and 32-bit integer. Any numeric field may be changed to any other numeric type and the length and the precision (number of decimal places) of display numeric, zoned numeric, and packed decimal numbers may also be altered by the customer.

The solution is to place field definitions in tables that are accessed by the application program at execution time. These tables form the run-time application data dictionary generated by the Application Customizer and are accessed only by a set of Application Customizer routines called intrinsics. This enables the designer to code the application without specific knowledge of the structure of the tables. As the Application Customizer is enhanced, the tables may change, but the application programs will not have to be modified because the intrinsics insulate the application from the Application Customizer.

A field may have several occurrences in an application, each having slightly different characteristics. For example, a numeric field may be present on an IMAGE data set, and also on a data entry screen defined for a transaction that updates the data set. The item on the screen will be defined as being display numeric type, with a length of ten digits including two decimal places. The same item on the data set will be defined as being packed decimal, with a length of 15 digits including four decimal places. The designer can develop customizable programs without concentrating on these differences because of the intrinsics provided by the Application Customizer to handle all arithmetic and data movement operations.

A table lookup is required every time any data item is manipulated by the application. Materials Management/3000 is structured to provide the best response time for users who perform the same transaction many times, using few or no other transactions. An example is loading dock personnel who perform "receive stock" transactions almost exclusively. When a transaction is entered, only that por-

Fig. 7. The designers of Materials Management/3000 used many services in designing transactions. V/3000 intrinsics (routines) communicate with the user terminal. IMAGE/3000 intrinsics store and retrieve data. Application Customizer intrinsics retrieve data item definitions, screen formats, data set formats, and customer-added processing specifications. Customizer intrinsics also manipulate any data items whose characteristics are unknown to the application designer and must be looked up in the Customizer tables.

tion of the customizer tables that contains data item definitions used by the transaction is moved to the program data area. The data item definitions remain in memory until the user branches to another transaction. With the needed data item definitions in program data memory, Customizer intrinsics may access data definitions with a minimum of overhead. This conserves memory and provides fast response time for subsequent executions.

Since there are Customizer intrinsics that perform data movement and arithmetic operations, instead of coding SPL statements to manipulate data, the application designer codes calls to intrinsics that add, subtract, multiply, or divide numeric data items, and move numeric or alphanumeric items. These intrinsics reference the data item definition tables, performing data validation, decimal point normalization, data type conversion, and security checking. If an error prevents proper processing, the intrinsic returns an appropriate error code, and the user can be informed.

Modifying Fields

In addition to changing data item characteristics, it is possible for the customer to add and delete some fields appearing on screens and data sets. Materials Management/3000 is designed to perform specific inventory control functions, so a working set of data items must be present for the application to perform its function properly. These data items are defined as critical to the application and may not be deleted by the customer. Other data items in the released product are included for optional processing and may be deleted by the customer for reasons of efficiency or to prevent user confusion. On the other hand, a customer may want to adapt the application to perform additional functions not anticipated by the application designers. This will require the addition of data items to data entry screens and data sets. A method must be used to represent the association of data items with screens and data sets to the application programs.

Fortunately, much of the processing in Materials Management/3000 and many other data processing applications involves the movement of complete records from place to place. For example, "add" transactions simply construct a record from the data items entered on a data entry screen, and after appropriate validation edits, move the record to an IMAGE data set. "Change" transactions retrieve a record from a data set, update it with fields entered from the screen, and then move the record back to the data set. When adding or deleting a data item on a data set or screen both the designer and the customer must associate the item with a specific record format. Record formats are nothing more than collections of data item definitions that correspond to the fields on a data set or data entry screen record. A data entry screen record and the corresponding data set record it will update will contain many of the same data items, although they may have different characteristics. Since it is unknown until execution time exactly what items will be present on a given record, the Application Customizer provides an intrinsic that moves corresponding data items from one record to another.

The operation of the MOVE CORRESPONDING intrinsic is very simple. The intrinsic is passed the record format defi-

nitions that describe the source and target records. For each item definition present in the source format, the intrinsic searches the target format for a corresponding item definition. If a match occurs, the data is moved from the source to the target record, changing the data type, length, and precision if necessary. This process continues until all corresponding fields have been moved from the source to the target record. The MOVE CORRESPONDING intrinsic allows the designer to think on a record level, not being concerned with individual data items. This makes it possible for the customer to add and delete noncritical data items at will.

Fig. 8 shows an example of MOVE CORRESPONDING operation. Each record is described by a format maintained by the Application Customizer. Every item is assigned a unique item number by the Customizer. This item number is used to identify all occurrences of an item. Each format consists of a format header, which contains pointers and information concerning other control structures, and a collection of item definitions, organized in ascending item-number order. The MOVE CORRESPONDING intrinsic performs its function for each item in the source format (in this case the screen format) which has a matching item definition in the target format (the data set format). The intrinsic locates the field and determines its length, type, and precision, using information stored in the item definition. In this example, the source field for item 135 is located at byte 0 and is ten bytes long. An item type code of 3 indicates that the field is in display numeric format and the precision is
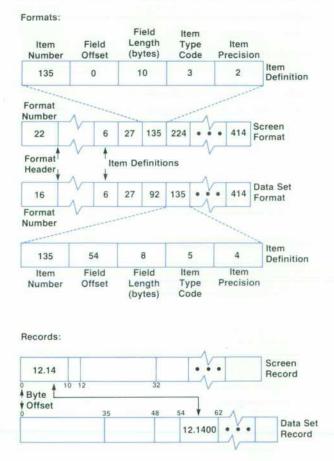

Fig. 8. *An example of the operation of the MOVE CORRESPONDING Customizer intrinsic. See text for details.*

two decimal places.

The target field is located at byte 54 of the data set record and is eight bytes in length. An item type code of 5 indicates that the field is in packed decimal format, and the precision is four decimal places. As shown at the bottom of Fig. 8, MOVE CORRESPONDING copies the field from the screen record to the data set record, changing the type, length, and precision of the data according to the item definition.

Changing Screens

In addition to changing field and record characteristics, the customer has the ability to modify the appearance of the application itself. Data entry screen appearance, and even the sequence of screens may be altered by the customer.

V/3000 provides a relatively simple method for altering screen appearance. Screens may be redesigned by repainting them using a few control character sequences on HP's 26XX series terminals. This gives the customer the power to alter screens so they look like forms that are presently in use, lessening the technology shock that many users experience. Screen alterations are then entered into the run-time application dictionary via the Customizer and translated into updated record format definitions. The application program is thereby insulated from cosmetic changes to screens. The MOVE CORRESPONDING and other Customizer intrinsics handle changes in data field order as easily as additions and deletions.

In Materials Management/3000, screens corresponding to transactions are at the bottom of a large tree of menus. The 26XX terminal series has eight dynamically definable softkeys. These keys are used by the application as the primary method of moving from screen to screen. The top of each screen in Materials Management/3000 contains eight labels, each corresponding to a data entry screen or a menu. The user may navigate through the menu tree by pressing a softkey that will cause the application to transfer to the desired transaction, or to a menu that will list seven other choices. The eighth function key is always labeled EXIT and takes the user to the screen's parent.

The customer has the ability to modify these labels through the Customizer, creating subtrees for different users. For example, security reasons may require that a customer prevent stockroom personnel from altering any engineering data. By removing any labels that identify transactions dealing with engineering data, it is possible to restrict the stockroom personnel to a closed set of transactions.

The application determines softkey definitions by looking up values in a screen sequence table, which is part of the run-time application dictionary and is accessed by Customizer intrinsics. An entry in the screen sequence table is associated with every screen. Before displaying a screen, the corresponding entry is moved to the program data area. If the user presses a softkey, the application looks up the value that corresponds to the key pressed and transfers control to the appropriate screen or menu. This allows the customer to be very flexible in tailoring the system and relieves the designer of the burden of determining the screen structure while coding.

An additional feature becomes very powerful for experienced users of Materials Management/3000. A 16-character command window is present on all menu screens. If the function desired by the user is not directly accessible from a menu, the function name may be entered into the command window and the corresponding screen will be accessed directly, eliminating the need to navigate through the menu tree. Whenever the application detects an entry in the command window, a Customizer intrinsic retrieves the appropriate value, effectively providing a ninth softkey, which the application program is not aware of. The command window may be altered via the customizer and V/3000 to accept only selected labels. This provides an additional measure of security, while providing the means for the experienced user to travel rapidly from screen to screen.

Processing Logic Customization

It is impossible for the designers of a general-purpose application to anticipate the needs of every customer. Customers will almost always want the application to do some additional processing, beyond the capabilities of the standard product. With noncustomizable applications, the customer would either have to purchase source code and modify it, or live with the standard product. Materials Management/3000 provides two methods of modification. The first involves V/3000 and the second involves the Application Customizer. V/3000 provides a set of powerful functions, including: checking for minimum length, data type checks, range checks, pattern checks, and data formatting. However these functions apply only to data entered on the screen records. To allow customer-defined manipulation and movement of data between screens and data sets, a set of functions called processing specifications may be entered using the Customizer.

Processing specifications are defined by the customer for each transaction where additional processing is desired. Simple commands allow the user to add, subtract, multiply, divide, and move data items. These commands are compiled and placed in tables that are accessed by Customizer intrinsics at execution time. In most of the product, each transaction is structured so that after all normal processing occurs but before any data sets are updated, the processing specification interpreter is called. This is a Customizer intrinsic that performs the operations indicated by the customer-entered statements. It is possible to alter almost any data item on any data set that is to be updated by a transaction. This tool allows the customer to extend the usefulness of the application program to areas that were not originally anticipated by the designer.

Fig. 9 shows how customer processing specifications are implemented. The format header of the screen format contains a pointer to any processing specifications the customer may have defined for the transaction. All processing specifications are generated by the Customizer and placed in a processing specification table, which resides in an extra data segment. The processing specification interpreter uses the pointer and length fields in the format header to locate and move the processing specifications defined for this transaction to the stack. The Customizer generates an intermediate language in the form of triples, which consist of an operation code and two operands. Each operand field is either a constant, a register, or a format/field number com-
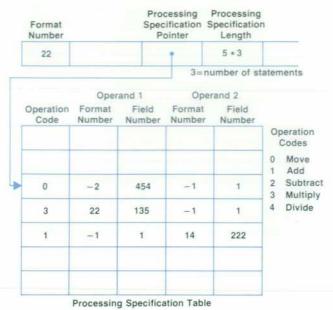
Format Header

	Format Number		Processing Specification Pointer	Processing Specification Length
	22		•	5 * 3

3 = number of statements

Processing Specification Table

Operation Code	Operand 1		Operand 2	
	Format Number	Field Number	Format Number	Field Number
0	−2	454	−1	1
3	22	135	−1	1
1	−1	1	14	222

Operation Codes

0 Move
1 Add
2 Subtract
3 Multiply
4 Divide

Fig. 9. *An example showing how customer-specified processing is implemented. See text for details.*

bination. In this example, the customer wishes to convert the value entered in field 135 from pounds to grams and accumulate the result in field 222, which is described in format 14. This might occur in the situation where the customer wishes to record the year-to-date quantity ordered for management reporting. Field 135, described in format 22, corresponds to the quantity-ordered field on the data entry screen. Field 222, described in format 14, corresponds to the year-to-date quantity ordered, which is accumulated on some other record for use in preparing periodic management reports. The normal unit of measure for ordering is pounds, but for some reason, management has decided to accumulate the total quantity in grams. The first triple moves the constant 454 to register 1. The second triple multiplies the contents of field 135 by the contents of register 1, and places the result back in the register. This converts the value of the field from pounds to grams. The contents of the register are then added to the contents of field 222 in the third triple. Upon returning from the processing specification interpreter, the transaction will update all of the affected data sets. This method of implementation allows the customer to add to or override the processing specified by the application designers.

Local Languages

HP's market for manufacturing applications is worldwide. The application designer cannot assume that the users of an application understand the English language. Materials Management/3000 is designed to be completely localized to any language supported by the 26XX series terminal without reprogramming. Localization may be accomplished by translating the screens using V/3000, by modifying report headings and error messages stored in message catalogs, and modifying other literals maintained in the application data dictionary. Materials Management/3000 uses many single-word literals to control processing.

For example, a user may enter engineering information about a part, such as whether it is normally purchased or fabricated. The English version of Materials Management/3000 codes this information as P or F on the data base. The literals P and F will have different interpretations in other languages. Therefore the customizer maintains another table containing all literals defined by the application designer. When manipulating literals entered by users, the application must first look up the current value of the literal. The table is loaded into the program data area and accessed by Application Customizer intrinsics. Because the table is located in the program data area and accessed directly, there is very little additional overhead. Non-English-speaking customers have an application product that is easily understandable by their users, and the support burden is minimized for HP because only one version of an application system needs to be supported instead of one for each language.

Security Checking

An advantage of manipulating data in an interpretive mode is that other functions may be added with a minimum of effort by the designer. One example is security checking. Many auditors demand that security access be carried down to the data item level. In Materials Management/3000, each user is assigned a password that will grant that user access to only the data items that he or she is expected to review or update. The password is entered only once on a special security screen. The user may view only screens that contain data items for which that individual has access, and on screens that may be accessed, not all data items may be reviewed or updated. This allows the user to see and manipulate only the authorized items.

This type of security would require a lot of design and coding effort in a conventional system. In Materials Management/3000 the Customizer intrinsics that manipulate data also perform a security check. The Customizer maintains a table containing all valid passwords along with a list of data items to which that password grants access. Each time a Customizer intrinsic accesses a data item, a table lookup is performed. If the user does not have access to the item, an error message is displayed on the terminal. This powerful feature is implemented with a minimum of overhead and design effort.

Conclusion

Materials Management/3000 is HP's first user-customizable, factory-supportable application system. It is a tool developed to help manufacturing management plan and control material requirements and inventory levels. The product may be tailored by the customer to fit individual circumstances without modifying source code. The Application Customizer, used by application designers and customers, allows extensive data item and cosmetic customization. The Application Monitor handles many day-to-day operating functions traditionally performed by an operations staff, making the design and use of Materials Management/3000 easier. Built on the Customizer and Monitor, Materials Management/3000 is another step toward providing a total solution for manufacturing companies.

REFERENCE INFORMATION
HP Model 32260A Materials Management/3000
Manufacturing Systems

Materials Management/3000 is a user customizable, interactive system for managing the materials planning and control function of a manufacturing operation. Materials Management/3000 consists of ten application software modules:

- Master Production Scheduling
- Rough Cut Resource Planning
- Parts and Bills of Material
- Routings and Workcenters
- Material Issues and Receipts
- Inventory Balance Management
- Work Order Control
- Purchase Order Tracking
- Material Requirements Planning
- Standard Product Costing

A discrete manufacturer who assembles standard, multi-piece products in lots represents the ideal candidate for Materials Management/3000. However, the software is applicable to most manufacturing operations. Hewlett-Packard Sales Representatives and Manufacturing Industry Specialists are available for consultation regarding the suitability of Materials Management/3000 for a specific application.

Materials Management/3000 is available in a form which meets the input, output, and processing requirements of most manufacturing companies. Manufacturing personnel may, however, modify data entry screens, data edits, and information retrieval screens to suit their specific needs. They may also add, delete, and/or modify data items in the Materials Management/3000 data base. All of these changes can be accomplished easily and without the need for computer programming.

Materials Management/3000 is a standard (object code) application product which is fully supported by Hewlett-Packard. A comprehensive support package is available and includes:

- 11 user reference and installation manuals.
- 2 weeks of customer training.
- Planning, implementation, and customization consulting by Hewlett-Packard manufacturing industry specialists.
- A wide range of software support services.

FEATURES

- On-line IMAGE data base update.
- Customizable user interface and data base.
- On-line terminal data entry, field editing, and error correction using the capabilities of HP's Data Entry and Forms Management System, V/3000.
- Easy to use, on-line transaction menus.
- Automatic transaction logging.
- Tailorable data entry screens and retrievals.
- On-line assistance via help screens.
- Use of intelligent HP CRT terminals.
- Predefined materials data base.
- Use of proven materials management techniques.
- Advanced security capabilities.
- Automated operator functions.

HARDWARE ENVIRONMENT

For Materials Management/3000, the minimum system recommended is an HP 3000 Series 44, III, 33, or 30 with 512K bytes of memory. However, the amount of memory actually needed will depend on expectations regarding terminal response time and transaction volume. A minimum of 100M bytes of disc storage is required; in any specific situation the exact amount of disc storage will be determined by an HP manufacturing industry specialist. The interactive terminals must be HP26XX series CRT units with at least 4K bytes of terminal memory (8K bytes is recommended). Display enhancements are also recommended.

SOFTWARE ENVIRONMENT

Materials Management/3000 requires the HP 3000 Computer System Fundamental Operating Software.

ORDERING INFORMATION

Product Number	Description	Price in U.S.A.
32260A	Right to use Materials Management/3000. Includes one complete manual set.	$31,500
32260R	Right to copy and sublicense one Materials Management/3000. Includes one complete manual set.	$26,250
32260M	Right to copy one Materials Management/3000 without sublicense. Includes one complete manual set.	$10,500

MANUFACTURING DIVISION: MANUFACTURING SYSTEMS OPERATION
19447 Pruneridge Avenue
Cupertino, California 95014 U.S.A.

Nancy C. Federman

Nancy Federman was born in Cleveland, Ohio and attended the University of Michigan, graduating with a BS degree in physics in 1970 and an MS degree in mathematics and computer science in 1971. During the next six years she did systems programming, computer security research, consulting, documentation, and further graduate study at the University of Chicago. She joined HP in 1977. She's helped develop HP 300 business application systems and the MPS module for Materials Management/3000. She's also done college recruiting and served on many HP task forces, and she's now a project manager with HP's Manufacturing Systems Operation. A member of the Society of Women Engineers. Nancy is married and lives in Saratoga, California. Her leisure activities include gardening, running, weight training, needlepoint, tennis, and piano.

Robert M. Steiner

A native of Jefferson City, Missouri, Bob Steiner received his BS degree in Computer Science in 1978 from the University of Missouri at Rolla. As a development engineer with HP's Manufacturing Systems Operation since 1978, he's helped develop the Application Customizer and various application modules for Materials Management/3000. He's a member of the IEEE Computer Society and the American Production and Inventory Control Society. He lives in San Jose, California and enjoys backpacking, skiing, and photography.

A Novel Approach to Computer Application System Design and Implementation

by Loretta E. Winston

TRADITIONALLY, BUSINESSES wanting to computerize their operations have been faced with two alternatives: develop an in-house solution using their own programming resources or purchase a standardized solution from an OEM or independent software house. Both alternatives have serious drawbacks. The development and continuing maintenance and enhancement of in-house application systems are expensive and very time-consuming. Furthermore, the initial time required to develop an in-house system delays the realization of any benefits from a computerized solution. Purchased software systems, on the other hand, are for the most part relatively inflexible. Businesses must either adapt their operations to fit the specifications of these predefined solutions or undertake the costly programming required to achieve a satisfactory solution. If they choose the latter course, they risk losing the support of the system vendor.

Hewlett-Packard's Application Customizer and Application Monitor, new software systems for the HP 3000 Computer, offer users a third alternative. These systems allow application designers to construct generalized application solutions and provide users with tools to tailor applications to meet existing needs and to modify these applications to meet the ever-changing requirements of business. These systems also provide tools for the user to define, initiate, and control an entire application system. All of this may be accomplished by personnel who are not data processing (DP) professionals.

This major contribution to application system design and implementation drastically reduces the need for businesses to maintain a large staff of DP professionals to supervise in-house computerized applications.

The first HP software product to make use of the Application Customizer and the Application Monitor is Materials Management/3000, a comprehensive materials planning and control system for manufacturers (see page 3). The entire application consists of the Application Monitor, the Application Customizer, and a set of customizable application systems. Fig. 1 illustrates the relationship between the various components. The Application Customizer and Application Monitor are basically tools to support the application software systems. Neither the Application Customizer nor the Application Monitor is itself customizable. This article describes the Application Customizer. The Application Monitor is described in the article on page 19.

Customization Capabilities

The Application Customizer offers an unconventional approach to constructing tailored application solutions. It is a compromise between the traditional approach—supplying source code application packages to end users and allowing them to reprogram these solutions using a general-purpose programming language to meet their specific requirements—and providing users with a specialized applications language to develop their own customized application systems.

Customizable application systems are table-driven so that the costly programming and debugging efforts normally associated with implementing tailored application solutions are eliminated. For each application the Application Customizer maintains a set of tables that define the application and its operational environment. These tables are collectively referred to as the application data dictionary. Tailoring of the application system is accomplished by modifying these underlying tables; no source program code is ever modified.
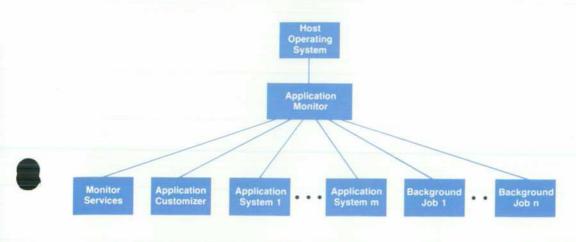
The application data dictionary differs from conven-

Fig. 1. The structure of a customizable application system, such as Materials Management/3000, based on the Application Customizer and the Application Monitor.
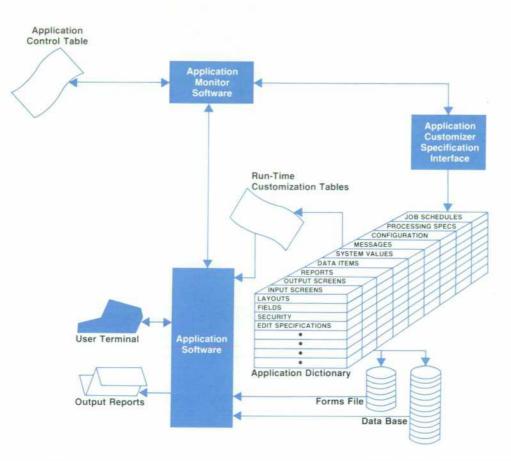
Fig. 2. A summary of the customization features provided by the Application Customizer.

tional data dictionaries in several respects. Conventional data dictionaries are basically repositories of information about the data bank, containing information such as data item names and characteristics, alias definitions, security specifications, edit information, and relationships between data items (i.e., data structures). The application data dictionary contains information not only about the data bank, but about the application system as well, including application data structures such as screen and report layouts, internal data structures, and application configuration. In traditional dictionary-based applications, if the characteristics of an existing data item are changed, or a new data item is added, or an existing data item is deleted, all programs that use the data item must be changed also. In Application Customizer dictionary-based systems, no reprogramming is necessary. Further, most data dictionary systems are passive in nature. They basically serve as documentation facilities and are unaware of the data base management systems and application processes that operate within the same overall environment. Information in the application data dictionary, on the other hand, is used during the actual execution of application programs.

There are two classes of Application Customizer users: application designers, who develop customizable application solutions, and system administrators, who tailor these generalized application solutions to meet the specific needs of their business environments. System administrators are roughly equivalent to systems analysts in terms of computer problem-solving skills and expertise. They are not necessarily computer programmers. System administrators have the responsibility not only for customizing application sys-

tems but for supervising and controlling the daily operation of these application systems as well. Fig. 2 summarizes the capabilities provided by the Application Customizer. The entire set of customizable capabilities is available to application designers. Only a subset of these capabilities is available to system administrators.
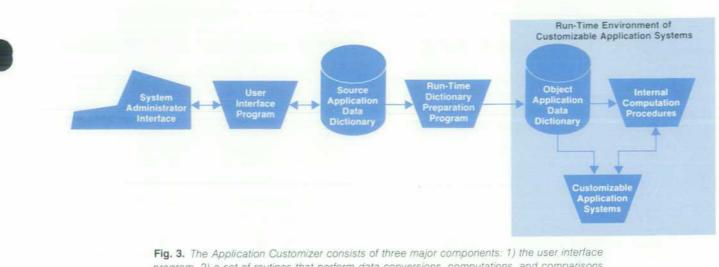
Generally when we speak of the Application Customizer, we mean the user interface to the Application Customizer system, which is an on-line, interactive program. However, the entire Application Customizer system consists of three major components: the user interface component, a set of routines (called intrinsics) that perform data conversions, computations and comparisons on customizable data fields and data structures, and a program that prepares a run-time (object) version of the application data dictionary from the compile-time (source) data dictionary (see Fig. 3). Two versions of the application data dictionary are maintained to allow one class of users to customize the application systems at the same time that a second class of users is operating or executing these applications.

The next section of this article outlines the steps an application designer takes in implementing a customizable application system and the final section illustrates the steps a system administrator might take in tailoring a customizable application system to fit specific needs and requirements.

Implementing a Customizable Application System

An application subsystem is a logical grouping of on-line and interactive transactions, background jobs, and reports that provide a total solution to some well-defined user problem or task. An application system is a group of one or more

Fig. 3. *The Application Customizer consists of three major components: 1) the user interface program, 2) a set of routines that perform data conversions, computations, and comparisons on customizable data fields and structures, and 3) a run-time application data dictionary preparation program.*

interrelated and interdependent application subsystems. There are two major tasks involved in implementing a customizable application system: 1) defining the overall application and its operational environment and 2) specifying the application data bank and set of functions that make up each application subsystem. These two tasks are further divided into several subtasks. To define the overall application and its environment requires the specification of the following entities: the application system, application subsystems, device and terminal configuration, and terminal security configuration. The complete specification of the application subsystems involves the actual coding of the transactions, background jobs, and reports, the definition of application data bases, and the specification of user interfaces for the various application functions (e.g., user messages, screen and report layouts).

To illustrate the steps involved in developing a customizable application system, the implementation of a hypothetical Inventory Management System will be presented. The Inventory Management System consists of a number of subsystems, one of which is the Inventory Stock Status Subsystem. This subsystem consists of transactions to adjust inventory balances, receive and disburse goods from inventory, transfer stock from one location to another, return goods to inventory, review stock status, and indicate recent stock activities. The Inventory Stock Status Subsystem provides a report called the Inventory Count Analysis Report that lists the current inventory status for all warehouses and inventory locations. This report is generated by a background job that is scheduled for execution on a demand basis. The subsystem contains a data base that consists of two major data sets called Stock Activity and Inventory Master. The Stock Activity data set contains basically an audit trail of transactions against inventory. The Inventory Master data set contains stock information by warehouses and inventory/bin locations within these warehouses. In defining the application and its operational environment, the first step is to define the name and characteristics of the application system as illustrated in Fig. 4. Second, the names and characteristics of the application subsystems are specified. Fig. 5 shows one of the subsys-

tems, namely Inventory Stock Status, being defined. Third, terminals for operating the application subsystems are specified, and finally, passwords to access these terminals are specified. This completes the specification of the application and its operational environment. The next major task is to define the data bases, functions and interfaces for each of the application subsystems. Figs. 6 and 7 show two steps in the definition of the application data bank. Figs. 8 and 9 illustrate typical steps involved in adding a new transaction to an application subsystem. The pseudo-code for the new transaction called "Adjust Quantity on Hand" is given below:

Step   1. Read user-entered input from screen.
Step   2. Lock INVENTORY—MASTER data set.
Step   3. Read INVENTORY—MASTER record for specified PART—NUMBER, WAREHOUSE and INVENTORY—LOCATION.
Step   4. Add ADJUSTMENT—QUANTITY to QUANTITY—ON—HAND.
Step   5. Update INVENTORY—MASTER record.
Step   6. Unlock INVENTORY—MASTER data set.
Step   7. Move corresponding fields from input screen buffer to STOCK—ACTIVITY record buffer.
Step   8. Lock STOCK—ACTIVITY data set.
Step   9. Add record to STOCK—ACTIVITY data set.
Step 10. Unlock STOCK—ACTIVITY data set.
Step 11. Exit.

Fig. 4. *An Application Customizer screen for specifying a new application system.*
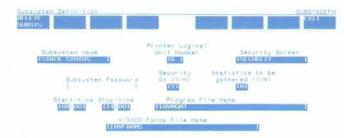
Fig. 5. An Application Customizer screen for defining a new application subsystem.

Fig. 7. An Application Customizer screen for defining a new application data set.

Fig. 8 shows segments of this program written in the HP 3000 Systems Programming Language, SPL/3000. The segments shown correspond to steps 4 and 7, respectively, in the pseudo-code program. The major coding difference between this and a conventional approach is that data manipulations (data transfers, conversions, comparisons, and arithmetic/logical operations) are not hard-coded; rather they are performed via specialized Application Customizer routines (intrinsics). To illustrate the power and usefulness of this feature, consider the following example. Suppose that transaction X adds records to data set Y and further suppose that a user wants to add a new field Z to data set Y and initialize it via transaction X. To accomplish this in customizable application systems, the user simply adds the new field Z to both data set Y and the screen for transaction X. The data handling routines will automatically move this new field from the screen to the data set. No reprogramming is necessary.

To better understand how this works, the contents of the application data dictionary must be examined in greater detail. The Application Customizer maintains a description of every application structure in its data dictionary. An application structure is a general term used to describe any collection of related fields. These include application screens, reports, data sets, and internal structures. For each application structure the Application Customizer maintains a format that contains the set of fields in the structure, their data types, their lengths, and other pertinent information. The application designer defines the default set of application fields and application structures. End users may add and delete field specifications from the entire application, add and delete fields from specific application structures, and change the lengths and data types of fields within application structures. Hence, specialized data manipulation and arithmetic/logical routines are provided to accommodate the customization of these application structures.

Basically, arithmetic/logical routines work as follows. Information on the data types and lengths of the source and destination fields is retrieved from the application data dictionary. The source fields are converted to an appropriate intermediate data format. The arithmetic operation is then performed. Finally the result of the computation is converted to the appropriate output data type and stored in the destination field. The other operations work similarly.

To complete the specifications of the Adjust Quantity on Hand transaction described above, the user interface must be designed and user messages entered. Fig. 9 illustrates the definition of a new form. A general-purpose screen-design/forms-management facility, V/3000, is provided for painting screens. At present there is no automated mechanism for creating screen formats for the application data dictionary from screens designed using the forms management facility. This must be done manually. However, automation of this step is anticipated in a future release of the product.

Customization of Processing Logic

Customization capabilities may be divided into four logical categories: application specification, user interface, application structures, and processing logic. Examples of three of these general capabilities have been demonstrated. The fourth capability, processing logic customization, is briefly described below and an example is given in the next section.

Customization of processing logic allows end users to tailor the functioning of application procedures. There are three different mechanisms for customizing processing logic: by selecting an algorithm from among a predefined set of algorithmic choices or techniques, by specifying a customizable parameter value, and by adding new process-

```
<< INCREMENT QUANTITY ON HAND BY ADJUSTMENT QUANTITY >>
AC'ADD (FORMAT'ADJUST'QTY'ON'HAND,
       FIELD'ADJUSTMENT'QTY,
       BUFFER'ADJUST'QTY'ON'HAND,
       FORMAT'INVENTORY'MSTR,
       FIELD'QTY'ON'HAND,
       BUFFER'INVENTORY'MSTR,
       FORMAT'INVENTORY'MSTR,
       FIELD'QTY'ON'HAND,
       BUFFER'INVENTORY'MSTR,
       AC'STATUS);


             .
             .
             .

<< MOVE CORRESPONDING FIELDS FROM THE DATA ENTRY     >>
<< SCREEN BUFFER TO THE STOCK-ACTIVITY RECORD BUFFER >>
AC'MOVE'CORR (FORMAT'ADJUST'QTY'ON'HAND,
              BUFFER'ADJUST'QTY'ON'HAND,
              FORMAT'STOCK'ACTIVITY,
              BUFFER'STOCK'ACTIVITY,
              AC'STATUS);
```


Fig. 6. An Application Customizer screen for defining a new application data base.

Fig. 8. Program segments from a customizable application transaction, showing calls to Application Customizer routines (intrinsics), e.g. AC'ADD.

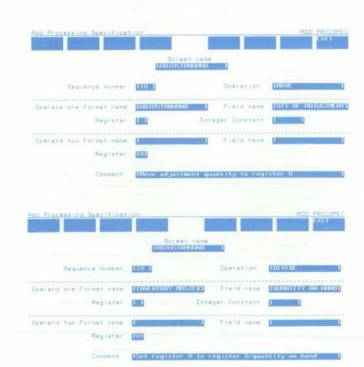**Fig. 9.** *Designing a V/3000 screen for a new transaction.*

New function key label

Fig. 12. *Changing a function key label on a screen form (V/3000 screen).*

ing steps. The first two mechanisms must be provided for by the application designer in coding the application functions. In instances where there are several recognized techniques employed by businesses, the application designer may elect to make all techniques available and allow the system administrator to select the particular technique or algorithm to be used. The concept of customizable parameter values is a slight variation of this same principle. In this case a single method for performing an application function is provided, but the input parameter(s) to the function may be changed dynamically by the system administrator. The third mechanism is by far the most powerful. It allows the system administrator to add special processing steps to existing application functions.
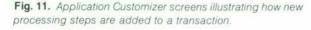

Fig. 10. *An Application Customizer screen for defining a new data item.*

Fig. 11. *Application Customizer screens illustrating how new processing steps are added to a transaction.*

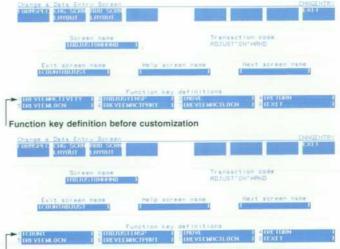## Tailoring an Application System

The following case study illustrates the ease with which an application may be tailored, without programming, to meet the requirements of a particular environment. Using the hypothetical Inventory Stock Status Subsystem described previously, several examples of customization are presented.

A clear distinction is made between customizing the application data dictionary and activating those changes. A special background job that prepares a run-time version of the application data dictionary is executed whenever customization changes are to be made available to the application subsystems. This allows application operators or data entry clerks to operate the application subsystems at the same time that the system administrator is customizing these subsystems.

The first example illustrates the simple steps necessary to add a field to the application.

Problem Statement. A user wants to keep track of the name of the warehouse foreman whenever an adjustment of inventory at a location is made.

Solution. The first step is to define the new data item to the application dictionary as shown in Fig. 10. The second step is to add the new field to the appropriate form or screen via the screen-design/forms-management system and then recompile the forms file. The third step is to add the new field

Function key definition before customization

Function key definition after customization

Fig. 13. *Changing a function key definition in the application data dictionary (Application Customizer screens).*

```
** MESSAGE CATALOG EDITOR **
/REPLACE 100

100 ───► 5  No record found for warehouse/inventory location

100 ───► 5  Either an invalid warehouse or location has been entered


              User message after customization

User message before customization
```

Fig. 14. *Customizing user messages.*

to the application data dictionary's screen layout, and the fourth step is to add the new field to the data base. The final step after customization is complete is to make the revised transaction available to application operators. This is accomplished by executing the background job that prepares the application data dictionary.

The second example illustrates the ease with which special calculations may be added to an existing transaction.

Problem Statement. A user wants to display the percentage of the adjustment whenever the inventory balance is adjusted.

Solution. The customization changes required are very similar to those outlined in the previous example and may be entered at the same time. In addition, special processing steps to calculate the percentage adjustment must be added to the application data dictionary for this transaction. Two steps are illustrated in Fig. 11.

The next example demonstrates the ease with which function key definition may be changed to access different application functions from a particular transaction screen.

Problem Statement. A user wants to be able to access the "Review Inventory Activity" menu screen from the "Adjust Quantity at an Inventory Location" screen.

Solution. The first step is to modify the function key label on the form or screen via the screen-design/forms-management facility and to recompile the forms file. The next step is to change the function key definition for this transaction screen in the application data dictionary. This two-step process is shown in Figs. 12 and 13.

The fourth example demonstrates the ease with which user messages may be customized.

Problem Statement. A user wants to change the error message that is displayed when an invalid warehouse or inventory location is entered.

Solution. The user simply invokes the editor system from the Application Customizer, modifies the message as desired, and then activates those changes. Fig. 14 shows the error message before and after customization.

The final example illustrates the simple steps necessary to change the schedule for a background job.

Problem Statement. A user wants to schedule the Inventory Count Analysis Report job to be executed automatically on a weekly basis rather than on demand.

Solution. As shown in Fig. 15, the user simply makes the necessary changes to the application data dictionary's schedule for this background job and then activates those changes.

Conclusion

Customizable application systems offer customers the best of both worlds. They make it easy for users to construct tailored application systems without costly and time-consuming programming, debugging, and maintenance efforts, and they permit on-going vendor support of these tailored systems. The HP 3000 Application Customizer provides many tangible benefits to application designers as well. It provides specialized routines and utilities that facilitate the implementation of customizable application systems and enhance maintainability and supportability.
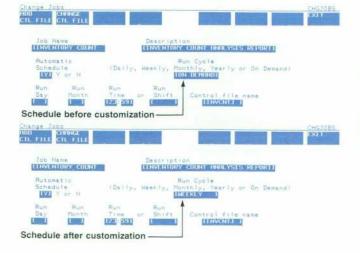
Acknowledgments

Fig. 15. *Customizing the schedule for a background job (Application Customizer screens).*

Loretta E. Winston

Loretta Winston first joined HP as a student in 1976, implementing mathematics library routines for the HP 300. In 1977, after receiving her MS degree in computer science from Iowa State University, she returned to HP and helped develop the MPS module for Materials Management/3000. She's now project leader for Application Customizer development. A native of Vicksburg, Mississippi, she received her BS degree in computer science from the University of Southern Mississippi in 1975. She's active in her church, plays racquetball, and regularly participates in conferences and workshops aimed at attracting minority groups and women to careers in science and technology. She lives in Santa Clara, California.

Automating Application System Operation and Control

by Barry D. Kurtz

APPLICATION SYSTEM OPERATION and control means the daily use and supervision of application system functions, including initiating programs that provide application functions to the user and scheduling and monitoring background jobs performing batch processing.

Until now, application system operation and control have required considerable user intervention. Also, since most business computing machines are general-purpose in nature, the task of initiating application programs is frequently left to personnel who are not computer professionals. Thus the user of an application (a clerk, receiving dock worker, etc.) may have to learn the host machine's command language and error codes, which are optimized for general machine use instead of individual applications.

Recent developments in operating system software have lessened the burden on the non-computer professional. These include user-defined commands, comprehensive "help" facilities, and so on. However, these have solved only part of the implementation problem. The user should perceive the computer system as a comprehensive solution to an application problem and not as a set of unrelated application tools.

Hewlett-Packard is dedicated to increasing the ease of use of application systems. A project that has made a great contribution to the ease of control and operation of application systems is called the Application Monitor, and was developed as a component of the current Materials Management/3000 application system offered by HP (see page 3). The Application Customizer described in the preceding article is another part of this effort.

The Application Monitor controls the execution of applications running under its control and provides services to them. It schedules, initiates and controls all interactive and background job activities in an application system. A friendly fill-in-the-form CRT interface is used by the user or manager of the system to review and control system activity. The monitor takes a major step toward operatorless,

abortless application systems by providing automatic application scheduling, control, and recovery services.

User Interface

The user interface of the Application Monitor, a friendly fill-in-the-form CRT interface, allows the user or manager of the system to perform comprehensive system control without having to learn a complex command language. Each screen presented uses the special function keys of HP 26XX series CRT terminals. The functions are described in eight shaded boxes at the top of the screen. Each box corresponds to a single function key (softkey).

A command window is also presented on Monitor menu selection screens. This area is used for special system control functions that are not executed by a simple function key signal.

Since the functions made available through the user interface are critical to system operation, the user may not add, delete, or change Application Monitor screens.

Traditional Application System Activities

Most application systems involve the following activities:

- Initiating on-line applications on user terminals
- Scheduling and monitoring background job processing
- Running recovery and cleanup jobs
- Supervising system operation.

Traditionally, most of these activities were executed manually and required frequent human intervention. All control was dependent upon the attention of the operator. This practice has led to inconsistent data processing activities, sometimes resulting in the loss of critical management reports.

The Monitor allows the user or manager of an application system to automate all of the above activities, thus providing an environment in which consistent data processing may take place. This reduces the frequency and severity of errors and security breaches, and reduces the training re-
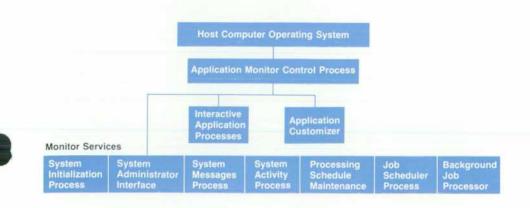

Fig. 1. *The application system environment, showing the relationship of the Application Monitor to other programs. The Application Monitor initiates, monitors, and controls run-time application activities. Monitor services are provided to application programs and users to initiate additional processes and supervise system activity. The Application Customizer may be initiated to alter the tables that define the system environment.*

Fig. 2. The System Administrator Interface is the system administrator's window to the application system. The functions that may be initiated from this screen are represented in eight shaded boxes at the top of the screen. These functions are initiated by pressing one of eight terminal function keys. This relieves the system administrator of the need to memorize a system command language, and allows functions to be initiated quickly and accurately.

quired for administration of the application system.

Classes of Application System Users

Two main classes of users make use of the services provided by the Monitor: system administrators and end users. The system administrator is an individual who has global responsibility for the application system. This individual supervises and controls all application system activities (i.e., background job scheduling, on-line application scheduling and control, etc.).

The end users of an application system include all those who use application program functions (i.e., clerks, managers, receiving dock personnel, stores or inventory personnel, etc.). End users benefit from good application system management, which leads to consistent data processing, reports that are on time, and other benefits. However, they do not usually get involved in actual operation and control.

Application System Operational Environment

The Application Monitor consists of eight separate programs. These programs run simultaneously and work together to monitor and control the application system. Fig. 1 depicts the application system environment.

Fig. 3. The System Activity Process user interface. The System Activity Process allows the system administrator to review current system activity. Interactive application transactions may be tracked and background job execution may be controlled: in this example, BOB'S TERMINAL is executing the REVIEW PART transaction (ten transactions have been executed, and the terminal has been active since 8:02 AM.). A background job (DAILY REPORT) is running and is presently executing the SORT INPUT step.

Fig. 4. The Processing Schedule Maintenance user interface. The Processing Schedule Maintenance process allows the system administrator to review the background job schedule for the current day. In this example, three jobs are scheduled to be run. The first job (SUMMARIZED BILL) has its AUTO START flag set to N. This means that it will require manual intervention to run. The system administrator must set the flag to Y before the scheduled run time. The other jobs will run automatically when their run time comes up.

The following is a brief description of the major processes that make up the Application Monitor.

Monitor Control Process (MCP). The Monitor Control Process is the parent of all programs in the application system environment. Once the application system is active, it is this program that initiates, monitors and controls application program activities.

The MCP uses a memory table to track program activity throughout the system. This table is called the application control table (ACT) and is constructed during system initialization. Each program in the system has an entry in this table called a process information block. The information kept in each block is sufficient for the MCP to monitor the current status of each program and provide any services that are determined necessary.

System Initialization Process (SIP). The system initialization process is initiated at system startup time and executes once a day. This program reads a prepared application data dictionary generated by the Application Customizer (see article, page 13). Based on the information read from this dictionary, the SIP initializes global tables to be used by the application system.

System Administrator Interface (SAI). This program is the system administrator's window to the Application Monitor. The SAI allows the user to select various functions to review and control application system operation. Fig. 2 depicts the

Fig. 5. The Processing Schedule Maintenance process allows the system administrator to modify the background job schedule. In this example, a job (PART REPORTS) is being added to the schedule. If this job is to be run at regular intervals, the system administrator may modify the permanent job schedule in the application data dictionary via the Customizer. The job will then automatically be scheduled to run daily, weekly, monthly, or yearly as specified.

Fig. 6. *The System Messages Process User interface. The System Messages Process allows review of system informational messages and messages requiring action from the system administrator. In this example, a background job has been restarted and has failed on the restart. The Background Job Processor is asking the system administrator if the job should run again. The system administrator may initiate the reply action to this message by pressing the appropriate terminal function key corresponding to a shaded box at the top of the screen.*

user interface for the system administrator process.

Job Scheduler Process (JSP). The job scheduler process automatically schedules background jobs that were specified to be run on the current day. These jobs are predefined by the system administrator through the Customizer.

Background Job Processor (BJP). A background job that has been scheduled for execution will be processed by the background job processor. This program executes background job commands and provides comprehensive job restart and recovery capability.

System Activity Process (SAP). This program allows the system administrator to review and control system activities. Through the SAP user interface the system administrator can review the current activity of all interactive and background job processes and control background job processing concurrency. Fig. 3 depicts the system activity user interface.

Processing Schedule Maintenance (PSM). The processing schedule maintenance process allows the system adminis-trator to review and modify the background job processing schedule for the current day. Figs. 4 and 5 depict the user interface for the processing schedule maintenance process.

System Messages Process (SMP). Messages informing the system administrator of the specifics of system activity can be reviewed by means of the system messages process. Certain messages may require a reply from the system administrator. The reply can be processed using the SMP user interface, which is illustrated in Fig. 6.

System Startup

The monitor control process is the initial program to be run in the application system. The MCP is initiated by a user-defined command supplied with the installation software. This relieves the user of having to know the physical file name of the MCP and any parameters that must be supplied.

The MCP launches the system initialization process. The SIP reads the prepared application data dictionary generated by the Customizer. This dictionary contains the following information critical to system operation and control:

- Terminal configuration
- Device configuration
- Schedule of interactive applications and their associated CRT terminals
- Schedule of all background jobs.

Using this information, the system initialization process builds global tables to be used during that day's operation of the system. When initialization is complete, the MCP begins normal execution.

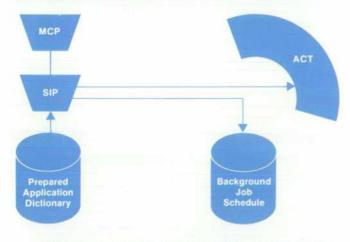Fig. 7 depicts the application system environment during system startup.

Fig. 7. *Application system startup. The Monitor Control Process (MCP) is the first program to run in the application system. Its first step is to launch the System Initialization Process (SIP). The SIP reads a prepared application data dictionary generated by the Application Customizer. From this dictionary, the SIP builds system-wide tables used in operation and control of the application system. The two main tables generated are the application control table (ACT) and the background job schedule. Once the tables are initialized, the MCP begins normal operation and continues execution indefinitely.*

Fig. 8. *Interactive application management. The Monitor Control Process (MCP) reads the application control table (ACT) built by the System Initialization Process. According to values initialized in the ACT the MCP launches application programs for presentation of application functions to users. The application programs use Monitor intrinsics to request services and allow the system administrator to review and control application activities.*
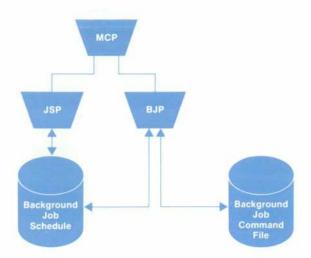
Fig. 9. *Background Job Management. The Job Scheduler Process (JSP) reads the Background Job Schedule to determine which jobs should be scheduled for execution at the current time. All jobs that have run times on or before the current time will be scheduled for execution. The Background Job Processor (BJP) executes a job by processing that job's command file. Up to three BJPs may run concurrently, allowing concurrent execution of up to three background jobs. Each job may have recovery procedures and checkpoints defined. If a job fails, the recovery procedures are automatically executed and the job is restarted at the appropriate checkpoint or step.*

Interactive Application Process Management

Application processes are launched according to values initialized in the application control table. When the MCP initiates an application program it sends the program its process information block entry number. This entry number corresponds to a physical process information block in the application control table. By using this number in Monitor-supplied intrinsics, the application may perform the following functions:

- Obtain information regarding which interactive terminal to use
- Start and communicate with a concurrent process to facilitate simultaneous processing of data
- Send a message to and start a successive process, suspending execution of the application until the successive process completes
- Log transaction response time for review by the system administrator.

Once the application system is in operation, interactive applications are automatically launched according to the schedule defined by the system administrator via the Customizer. If the schedule defined for application initiation is accurate, no user interaction is necessary to gain access to application program functions. The appropriate application will be presented to the appropriate set of users at the appropriate time.

If special needs arise, the system administrator may modify interactive application activity through the system administrator interface.

Fig. 8 depicts the MCP's management of interactive applications.

Background Job Management

Background jobs are scheduled for execution according to a list created by the system administrator via the Customizer. This list is read each day by the system initialization process to determine the jobs to be run for the current day.

The jobs are automatically executed at the time specified in the list. If a job fails, predefined recovery procedures are executed. When intervention is necessary, the system administrator may modify the execution sequence of background jobs via the processing schedule maintenance process.

Fig. 9 depicts the Monitor's management of background jobs.

Conclusion

The Application Monitor is a major step in HP's long-term commitment to increase the ease of use of application systems. The automation of an application system greatly reduces the time the system administrator must spend supervising operations and training operators. This allows more time for management of external functions in the environment where the applications are used.

Acknowledgments

Several people have contributed to the successful development and support of the Application Monitor project. The original design team included the author and Harry Sterling. Charlie Sum joined the team following project completion and has assisted in support of the current system and research of future application management systems. Steve Goldsworthy is the section manager with Harry Sterling as project manager.

Barry D. Kurtz
Originally in HP 3000 manufacturing, development engineer Barry Kurtz later joined the application development team at HP's Manufacturing Systems Operation and designed many components of the Application Monitor for the HP 3000. Barry came to HP in 1976 with a background in digital electronics. He holds an AS degree in electronics and is continuing his course work for a BS degree in computer science. Born in Richmond, Indiana, Barry is married and lives in Campbell, California. He's a data processing instructor at a local community college, and his interests include church activities, microcomputers, playing guitar, and writing songs.

Precision DVM Has Wide Dynamic Range and High Systems Speed

This digital voltmeter makes precision laboratory measurements with 100-nanovolt dc resolution and two-ppm linearity. Variable integration time allows four-digit measurements at 300 readings per second.

by Lawrence T. Jones, James J. Ressmeyer, and Charles A. Clark

THE MODEL 3456A DIGITAL VOLTMETER (Fig. 1) is designed to handle most bench and system voltage and resistance measurement problems. For the bench user, a calculator-style keypad, nine math functions (including conversion of thermistor resistance to °C or °F) and reading storage facilitate many measurements formerly done with much effort, if at all. For systems use, the standard 3456A is compatible with the HP-IB.* Variable speed and resolution allow the user to tailor the voltmeter to the job at hand. Reading storage and program memory allow an HP-IB controller to perform other tasks while the voltmeter is running without the need for time-consuming communications between them.

Most of the versatility and improved speed of the 3456A comes from its integrating analog-to-digital (A-to-D) converter using the Multi-Slope II technique developed for this product. It can run in five different modes, trading off speed

* Hewlett-Packard's implementation of IEEE Standard 488 (1978).

against resolution and precision. This tradeoff is made by the user specifying how long the input is to be integrated. The unit of time for this specification is one power line cycle (16.67 milliseconds for 60-Hz operation) and the integration times available are 100, 10, 1, 0.1 and 0.01 power line cycles. The performance parameters affected by changing the integration time are reading rate, line frequency rejection, accuracy, number of digits converted, and voltmeter electrical noise (see Fig. 2 and Fig. 3). Power line noise is rejected by integrating for a period equal to an integral number of power line cycles.

For the six-digit resolution of the A-to-D converter to be truly useful, it needs to be coupled with good linearity (± 2 ppm). Fig. 4 shows the linearity of a typical unit. This level of performance makes the 3456A an extremely useful tool in a standards laboratory environment.

The 3456A Digital Voltmeter measures dc volts on five ranges with 100-nV resolution on the lowest range and

Fig. 1. *The HP Model 3456A Digital Voltmeter is a 3½-to-6½-digit, fully-guarded integrating voltmeter with flexibility and precision, designed for both bench and system measurement applications.*
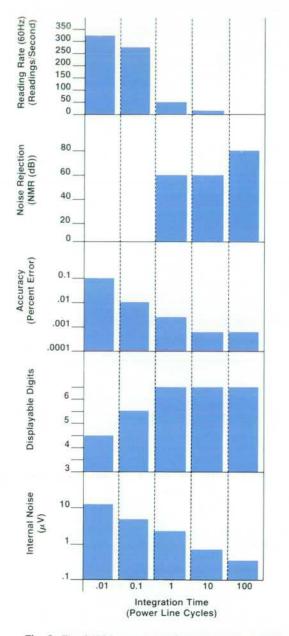
Fig. 2. *The 3456A user can select measurement integration time to obtain the optimum combination of reading rate, noise rejection, accuracy, internal noise, and displayable digits for a particular application. The relationship of integration time to these parameters is shown above.*

1000V full scale on the highest. The accuracy over a 24-hour period at 10V is 10 ppm. For ac volts, the 3456A has a true-rms converter that measures on four ranges from 1.2V full scale to 700V full scale, all within 0.05% of midband reading. Resistance can be measured using either two-wire or four-wire measurements. Both functions have eight ranges with 100-μΩ resolution on the lowest and 1 GΩ full scale on the highest. Basic accuracy on the 1-kΩ and 10-kΩ ranges is 24 ppm at full scale. All functions are selectable from the front panel or from the I/O bus.

Using the **SHIFT** key gives the operator five more functions. The **RATIO REF** terminals are used to measure a dc voltage for use as the denominator for three calculated ratio functions. These ratio functions first measure the voltage on

the volts terminals (dc, ac, or ac + dc). Next, they measure the dc voltage on the **RATIO REF** terminals and do a software division to arrive at the answer. Ratio reference measurement autoranges between the 0.1V, 1V and 10V ranges.

The last two functions are activated by shifting the ohms functions. These new voltmeter functions are called offset-compensated ohms. Resistance measurements, both two- and four-wire, can be made with these functions in the presence of offset voltages, such as thermal voltages. The voltmeter automatically takes the error voltages into account and displays the correct resistance value. The offset-compensated ohms functions have four ranges: 100Ω to 100 kΩ full scale.

The **AUTO ZERO** key toggles the digital automatic zeroing function of the 3456A on and off. In stable environments the reading rate can be increased by a factor of almost two if **AUTO ZERO** is turned off.

The **FILTER** key turns the active input filter on and off in dc voltage measurements. The filter has three poles at 5 Hz, giving 60 dB of rejection at 50 Hz. This key can also be used in ac volts measurements to extend the low-end frequency from 400 Hz down to 20 Hz. In ac measurements, the **FILTER** key turns on the input filter to filter the output of the ac converter, and turns on a lower-frequency filter in the ac converter itself. A delay for filter settling is automatically added between readings when the filter is on.

The **TEST** key puts the 3456A into a mode that cycles the instrument through a number of internal tests. If any of the tests fails, the failed test number is displayed. Any individual test can be run and the readings observed by storing the appropriate test number in the count register while in the test mode. The tests exercise all circuits in the 3456A including the various crossguard circuits, A-to-D converter, input amplifier, input switching, input filter, ohms measurement function and the ac converter.

The systems user is given a new feature with the front panel **SRQ** key. This key activates the service request function on the HP-IB. When supported by software in the system controller, this new channel of communication can initiate complex measurement tasks in any sequence. For example, in a production test application a technician may have five distinct test sequences. Using the **SRQ** key and a numerical entry, the technician can tell the controller which sequence to run. This feature lets a controller handle a number of 3456A test stations from a central location.
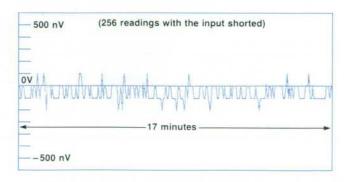

Fig. 3. *A typical 3456A DVM exhibits excellent noise performance with a selected integration time of 100 power line cycles (sampling every three seconds).*
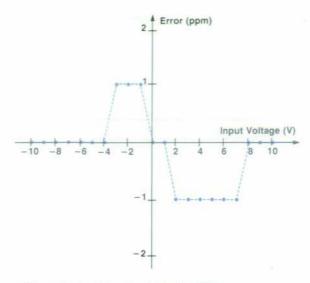
Fig. 4. *Typical linearity plot for the 10V range.*

The numeric keypad controls registers used in the math functions (see box below) and in voltmeter control. Voltmeter control registers are N (number of readings per trigger), G (number of digits displayed), D (delay from trigger to sample start), and I (number of power line cycles integrated over). Each of these registers can be set by the user according to the user's needs.

Multi-Slope II

To maintain the linearity and low-noise advantages of integration while decreasing the time needed to do a measurement, an improved A-to-D conversion technique (Multi-Slope II) was developed for the HP Model 3456A Digital Voltmeter. It was also highly desirable that the cost
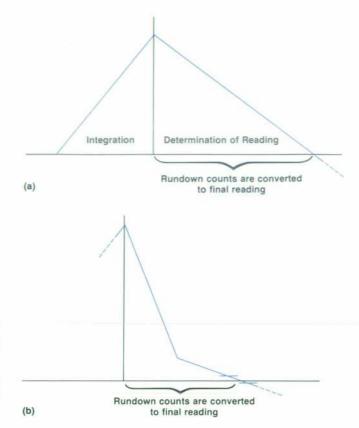

Fig. 5. *The basic dual-slope A-to-D technique (a) has limited-conversion speed because the final reading is not known until the ramp crosses zero during rundown. In the 3455A DVM multislope A-to-D technique (b), two rundown slopes are used to reduce the time required to determine the final reading. The initial steep rundown slope reduces conversion time and is switched to a more gradual slope just prior to the zero crossing to provide a more accurate reading using a conventional-speed zero comparator.*

of this circuit be kept relatively low.

In addressing this problem, two principles were developed:

- Do as much processing of the answer as possible during the measurement time.
- Microprocessors can add and subtract much faster than they can multiply. Therefore, potential gain errors should be converted into offset errors that can be subtracted out.

To understand the 3456A's A-to-D technique, consider the traditional dual-slope technique (Fig. 5) used in several of HP's earlier DVMs. In this technique the unknown voltage causes the integrator to ramp for a fixed time (t_{runup}). Then a known voltage is applied to the integrator until the output crosses through zero. The time required for this crossover ($t_{rundown}$) is proportional to the unknown voltage and can be measured by a digital timing circuit (see Fig. 5a).

This basic approach was refined during the evolution of HP system voltmeters. The A-to-D conversion technique used by the 3455A Digital Voltmeter[1] reduces rundown time by first discharging the integrator at a fast rate and then switching to a slower rate until zero crossover (Fig. 5b).

3456A Math Functions

1/NULL: The first reading V_1 is subtracted from each successive reading V_n and the difference $V_n - V_1$ is displayed for each reading.

2/STAT: The statistics function accumulates the number of readings taken, the maximum and minimum readings, and the mean and variance of the readings in recallable registers. The voltmeter displays each reading normally while this function is running.

3/PASS/FAIL: The user enters the upper and lower limits. Any reading outside of these limits will display HI or LO and pull the service request line. Readings inside the limits will be displayed normally.

4/dBm(R): The user enters the resistance R. All readings are then displayed as power dissipated in the resistor in units of dB referenced to one milliwatt.

5/THMS °F: Thermistor linearization in degrees Farenheit. With the voltmeter in ohms and a thermistor (HP part number 0837-0164 or equivalent), the reading is displayed in °F.

6/THMS °C: Same function as **THMS °F** but reading is displayed in degrees Celsius (°C).

7/(X−Z)/Y: This linearization function allows the user to make both a zero correction and a slope or gain correction 1/Y.

8/100(X−Y)/Y: Percent deviation of the reading X from the stored value Y.

9/20 log X/Y: The reading X is displayed in dB with respect to the stored reference Y.
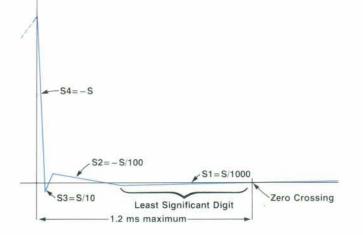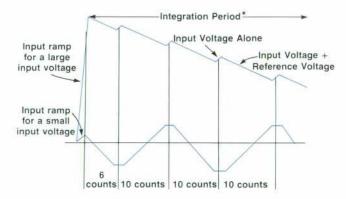

Fig. 6. *The Multi-Slope II A-to-D technique used in the 3456A DVM uses four slopes during the rundown period to determine, in succession, the four least-significant digits of the final reading.*

The 3456A DVM takes this improvement one step further. Initially, a steep slope S4 is applied until a zero crossing is detected. Next, a slope S3, one-tenth the magnitude of S4 and of opposite polarity, is applied until the integrator crosses back through zero. This process is repeated two more times, reducing the slope magnitude by 90% and changing polarity each time, until the final slope S1 passes through zero. Fig. 6 shows a typical rundown sequence.

The multiple crossings at first may seem strange. The important point is to return the integrator to the same state at the end of the measurement as it was at the beginning (i.e., zero volts). The fact that S4 overshoots zero is ac-

* (Number of power line cycles selected by user)

Fig. 7. *During the integration time used in the Multi-Slope II method the reference voltage ramp is applied periodically at a fixed rate independent of the unknown input voltage level. The polarity of the reference ramp during each period is dependent on the value of the sum of the previous ramp and the input voltage just before the previous ramp was turned off. This approach reduces the voltage level at the end of the integration time, thus reducing the demand on the integrator capacitor for low dielectric absorption. The number of positive reference ramps used during the integration time minus the number of negative ramps determines the most-significant digits of the reading.*

counted for in slope S3. Since each successive slope is exactly one decade smaller than the one preceding it, the microprocessor can accumulate the answer as each slope is applied. Each digit of the answer is stored in a register and each slope corresponds to a digit (S1 corresponds to the least-significant digit (LSD), S2 to the next highest digit, and so on). The time that each slope was on is accumulated in its register while the next slope is being counted. By the time slope S1 is crossing zero the higher-order digits have been determined.

The idea of accumulating the answer during the measurement can be extended to the runup portion as well. By periodically applying a reference slope during runup (Fig. 7) the most-significant digits can be converted during the time the integrator is being charged. Increased dynamic range without sacrificing noise performance is another benefit. Fig. 7 shows the integrator output waveforms for two different voltage input levels to the 3456A A-to-D converter. During runup the reference slope is turned on periodically. The reference slope polarity used during each of the periods depends on the state of the comparator just before the previous slope was turned off. Subtracting the number of times that one reference slope polarity is on from the number of times that the other is on determines the most-significant digits of the answer.

Increasing the length of runup allows more digits to be converted at the cost of decreasing conversion speed. The 3456A uses four integration times to obtain various speed/resolution combinations. In its fastest mode the integration time is 167 microseconds (0.01 power line cycles (PLC)). In this mode the reference is never turned on during runup, allowing all four digits to be converted during rundown. The five-digit mode integrates for 1.67 milliseconds (0.1 PLC) so that the most-significant digit (MSD) is converted during runup with the four LSDs converted during rundown. The longest integration time is 10 PLC. While only six digits are displayed, seven digits are actually converted with the LSD used as a guard digit to give a very accurate six-digit measurement. Digitally averaging ten of these 10-PLC measurements (equivalent to 100 PLC) allows very-low-level signals to be measured with less noise.

Much effort was spent in the development of the 3456A A-to-D converter to identify and eliminate major errors. A-to-D errors can be grouped in two classes. One is offset errors, that is, the A-to-D does not read zero when the input is zero volts. The second is gain errors, that is, a 10-volt input does not give a 10-volt output.

Offset errors are corrected by taking two measurements. The unknown voltage is measured and then followed by another measurement with the input shorted. Subtracting the second measurement from the first eliminates offset errors. For maximum speed the offset of the voltmeter can be measured only once and stored in a register. At the beginning of a reading the complement of this value is stored in the answer accumulator registers. As the measurement of the unknown voltage proceeds, each digit is added to its appropriate register. When the measurement cycle is complete the answer accumulator contains the measurement with the offset error voltage subtracted out.

Small variations in the switching time of the reference voltage can cause errors in the final answer. In the A-to-D
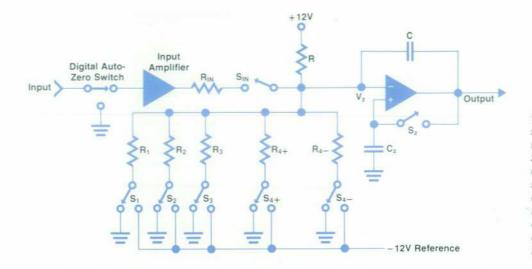
Fig. 8. A highly simplified schematic of the integrator and surrounding circuitry. Two auto-zero systems are used—a digital method using a separate measurement cycle to determine the offset error and an analog technique where the autozero capacitor C_z is used to store any offsets produced at the input to the integrator.

converter of the 3455A DVM the number of times that the reference is turned on depends on the magnitude of the input voltage. This is a potential gain error. In the 3455A such gain errors are handled by measuring a known 10 volts, calculating the error, and multiplying subsequent measurements by a correction factor. It was felt that this procedure takes too long to be used in the 3456A. Every attempt was made to convert any potential gain errors into offsets.

The Multi-Slope II technique holds the number of reference voltage switch transitions constant regardless of the magnitude of the input. Timing errors are therefore the same for all inputs and can be subtracted out.

The automatic zeroing scheme of the 3456A uses a single reference to achieve both positive and negative-going ramps. A schematic of the integrator is shown in Fig. 8. When switches S_1, S_2, S_3, S_{4+} and S_{4-} are switched to ground the current I from the +12V supply goes through the summing node into the integrating capacitor C (the summing node is at virtual ground). This generates a negative ramp at the integrator output.

If both S_{4+} and S_{4-} are switched to the −12V reference, twice the current (2I) supplied by the +12V supply alone is drawn through the parallel paths of R_{4+} and R_{4-}. The extra current must be drawn from the integrating capacitor to the summing node, which causes a positive ramp to be generated at the output of the integrator.
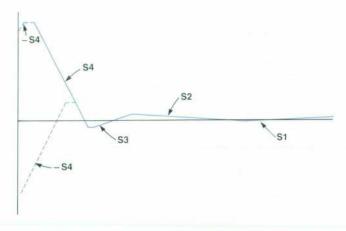
Between measurements an automatic zeroing operation is performed. S_{4+} and S_2 are switched to the −12V reference and S_1, S_3 and S_{4-} are connected to ground. Because the parallel combination of resistors R_2 and R_{4+} is approximately equal to resistor R, any difference Δ between the +12V and the −12V references will appear at the summing node as Δ/2 when switch S_z is closed. This error term Δ/2 is stored on the autozero capacitor C_z when S_z is opened. During ramp generation, S_z is open and the positive terminal of the operational amplifier is at Δ/2, which corrects for the Δ error in the positive supply. The +12V supply needs only to have good short-term stability since the +12V error is converted to an offset and subtracted out.
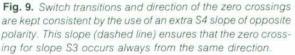
Another potential source of error is hysteresis in the zero comparator. To combat this problem the rundown slopes are arranged so that the final zero crossing always occurs in the same direction. To discharge integrator voltages of both

polarities, two S4 slopes of the same magnitude, but different polarities, are used. The length of time that each S4 polarity is on depends on the magnitude and polarity of the voltage to be discharged (Fig. 9). Note that both positive and negative-polarity slopes are only needed for the most-significant slope S4. These ramps are generated by closing the appropriately identified switches in Fig. 8. The negative-going slope S2 is obtained by connecting switch S_2 to ground because it was connected to −12V during the automatic zeroing cycle.

To minimize cost, one of the design goals was to devise an algorithm requiring a minimum of precision parts. The −12V reference, the set of tantalum-nitride resistors that produces the slopes, and a low-noise FET input for the integrator are the only precision components required. The critical slope switches are a digital CMOS inverter pack connected as shown in Fig. 10. Only one adjustment to the −12V reference is needed to calibrate the A-to-D converter.

What's Inside

Fig. 11 shows a block diagram of the paths taken by various signals. The input switching block has two basic functions. One is to complete the signal paths for a given voltmeter configuration. The second is to multiplex the

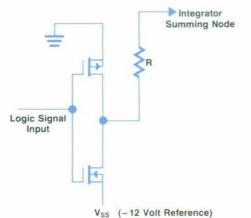**Fig. 9.** Switch transitions and direction of the zero crossings are kept consistent by the use of an extra S4 slope of opposite polarity. This slope (dashed line) ensures that the zero crossing for slope S3 occurs always from the same direction.

Fig. 10. *The slope switches shown in Fig. 8 are CMOS inverters.*

measuring terminals for the measurement cycle. For each signal condition, a correctly scaled, low-ripple voltage is presented to the A-to-D converter. For dc volts the signal conditioner is the dc input amplifier, and for dc signals that have large ac components, the low-pass filter is added to attenuate them. For ac volts the scaled dc voltage is supplied by the ac converter. For frequencies below 400 Hz the low-pass filter is also connected. The gain in the input amplifier is set at one. For resistance measurements, correctly scaled current is passed through the unknown resistor. The dc input amplifier is set at a gain of 100 for the 100Ω range, a gain of 10 for the 1-kΩ and 10-kΩ ranges, and a gain of 2 for the remaining resistance ranges. The current source used in the 3456A is referenced to the voltmeter's **LO** terminal. This eliminates the separate floating power supplies that would be required if the current source were floating. When lead impedance is a concern, the 3456A can be used
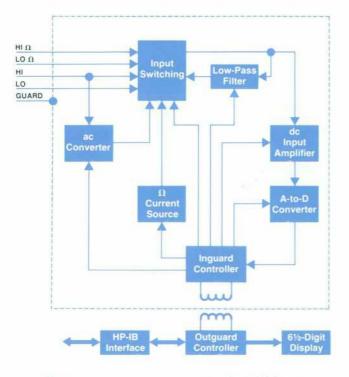
in a four-wire mode. In this mode the current is passed through the unknown resistor via the **HI** and **LO** terminals and the voltage drop is sensed via the **HI** and **LO** Ω sense terminals, which float with respect to the voltmeter **LO** terminal and therefore the current source. These terminals are also used for the ratio reference when the 3456A measures voltage ratios.

Autocalibration schemes are often used to correct for internal errors in voltmeters. These errors or autocalibration constants are measured between readings. The sequence is measure unknown, measure constant, measure unknown, measure constant and so on. In the HP Model 3455A Digital Voltmeter[1] there are 13 autocalibration constants that are stored in instrument memory. This takes time

Table I

Range Switch State Table
(0=OFF, 1=ON, see Fig. 12)

Range	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
1V	1	1	0	0	0	1	0	0	1	0
10V	0	0	1	1	1	0	0	0	1	0
100V	0	0	1	0	0	1	1	1	0	0
1000V	0	0	1	0	0	1	0	1	0	1

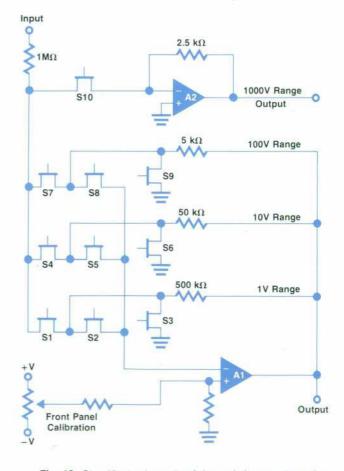**Fig. 11.** *Block diagram of the 3456A Digital Voltmeter.*

Fig. 12. *Simplified schematic of the resistive ac attenuator. Virtual-ground switching at the amplifier inputs provides good range isolation and eliminates the need for a gate bias. Table I above indicates the switch closures required for each range.*

and puts some rigorous requirements on the input switching. For instance, 1000Vdc must be isolated while measuring the accuracy of the high-voltage attenuator.

In the 3456A, gain errors due to the instrument reference and dc input amplifier are adjusted out by the front-panel calibration procedure. The error left is the offset of the dc input amplifier. This is removed by an automatic zeroing scheme. With **AUTO ZERO** on, each measurement really consists of two measurements. The first measurement is the unknown voltage plus the dc input amplifier offset (voltmeter **HI**). The second measurement is the dc input amplifier offset (voltmeter **LO**). These two measurements are subtracted, leaving only the unknown voltage as the answer. With **AUTO ZERO** off, instrument **LO** is measured and stored by the inguard controller. The measurement cycle then consists only of measuring the **HI** terminal voltage and subtracting the stored value. Note that for any instrument reconfiguration the offset must be updated and this is done automatically. With automatic zeroing the input switching becomes much simpler and more reliable. For instance, only five relays are required for voltage and/or function block isolation, compared with as many as thirteen for the 3455A instrument using **AUTO CAL**.

ac Resistive Attenuator

A resistive attenuator, Fig. 12, is used in the 3456A ac converter. Using fineline thin-film resistors, high-gain negative feedback and virtual-ground switching at the input to the operational amplifiers, the attenuator has a −3-dB point at 1 MHz and range isolation greater than 88 dB at 100 kHz. Besides low cost, the big advantage of the attenuator is that it has only one adjustment. This adjustment is available at the front panel; it corrects for the attenuator offset.

Two types of adjustments generally needed in ac attenuators are those for low-frequency gain and high-frequency flatness. If the thin-film resistors are trimmed properly, the scaled output of each range at low frequencies can be made to agree within ±0.01%. Other components affect full-scale accuracy. However, they drift with time at a rate at least ten times greater than the resistor ratio, so there is no need to provide periodic correction of this ratio. To obtain range-to-range tracking with frequency, high-gain negative feedback is used. Each range has the same loop gain characteristics. Therefore, when the loops are closed their frequency responses are equal.

Because of the number of switches used to achieve range selection, there is a capacitance of 27 to 33 pF at the summing junction of amplifier A1 in Fig. 12. To get and maintain good frequency response with the impedance levels used, amplifier A1 has only one pole below 5 MHz and has switchable gain-bandwidth. For instance, when used in the 1V range, a minimum gain-bandwidth product of 100 MHz is required.

Since the attenuator is uncompensated (resistive-only), the resistor ratios must hold not only at dc and low frequencies, but at high frequencies also. To insure this, the capacitance across the one-megohm input resistor must be kept as low as possible. To get a one-megohm resistor with flat-enough frequency characteristics, a three-wire technique is employed. A third wire is held at ground potential and brought into close proximity to the input and output terminals of the 1-MΩ resistor. This shunts some of the capacitance to ground that would otherwise shunt the resistor. With this technique the transfer capacitance is 0.15 pF, ten times better than that for two-wire techniques. This resistor also has the capability of handling high-voltage pulses up to 3 kV without damage. See Fig. 13 for a photograph of the fineline thin-film resistor chip used in the 3456A's ac converter.

Wide Dynamic Ohms Measurement

Using the input switching and arithmetic capability of the outguard processor, the ohms function achieves a wide dynamic range. The measurement of low resistance values can be enhanced by using the offset-compensated ohms function. This function takes two separate samples of the unknown. The first sample is taken with the current source turned on and the second is taken with the current source turned off. The second sample measures the voltage on the unknown not caused by the current source. This voltage adds to the voltage generated by the current source and causes an error. This error voltage is subtracted from the first sample to give the true voltage caused by the current source in the resistor.

The importance of this function is demonstrated when the magnitudes of some signal and error voltages are compared. One type of error voltage is generated thermally by dissimilar metals in the measurement circuit. A metal film resistor can generate voltages on the order of many microvolts where the resistive metal film contacts its own copper leads. On the 100Ω range the current source puts out 1 mA. The resolution in the six-digit mode is 100 $\mu\Omega$ which

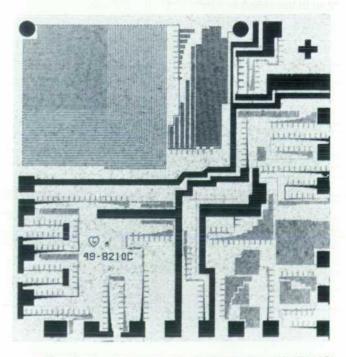**Fig. 13.** *Magnified view of the fineline resistor chip (8210C) used in the ac converter. The 1-MΩ input resistor is in the upper left corner. Note the contact pads rounded for high voltage and larger than usual area for heat dissipation. The third wire mentioned in the text for reducing transfer capacitance is located on the printed circuit board on which this 5-mm-square chip is mounted.*

Voltmeter Stores Measurement
Instructions and Data

The HP Model 3456A Digital Voltmeter has an internal memory (RAM) that can be used to store the results of measurements and the instructions for taking the measurements. This capability allows an HP-IB controller to input a short data-gathering program to the 3456A and then be able to address other instruments on the bus while the 3456A takes and stores readings. A typical set of ASCII alphanumeric strings output on the HP-IB to the 3456A by a controller is given below (the spaces are optional and included here to separate individual commands for discussion):

```
"L1 F2 Z0 1STI D0 P1 30STN 0.05STD RS1 T3"
"F1 10STN 0STD T3"
"F3 R4 20STN 0.01STI T3 RS0 P0 Q"
```

This program can be executed at any time by sending the alphanumeric string "HX1" on the HP-IB to the 3456A. The first line indicates the beginning of the program and tells the voltmeter to take 30 ac-volt measurements and store them in packed format in the internal memory. The second line tells the 3456A to take and store 10 low-level dc-volt readings with normal-mode rejection (NMR). The last line takes 20 thermistor resistance measurements at 300 readings per second and indicates the end of the program.

Most of the commands are related to the location of the appropriate key on the front panel of the 3456A. For example, dc volts is the first function key, hence F1 is the command. Similarly F2, F3,

F4, and F5 indicate the second, third, fourth and fifth function keys corresponding to ac volts, ac-plus-dc volts, two-wire kilohms and four-wire kilohms, respectively. Code definitions for other commands in the above program are:

L1 - Marks the beginning of a sequence to be stored in program memory.
Z0 - Turns **AUTO ZERO** off.
1STI - Selects one power-line-cycle integration time.
D0 - Turns the display off.
P1 - Turns the packed data mode on.
30STN - Thirty readings to be taken per trigger.
0.05STD - Store 50-ms delay between measurement cycles.
RS1 - Turns readings stored on.
T3 - Triggers the number of readings previously programmed.
0STD - Removes delay between readings.
R4 - Selects the 10-kΩ range.
RS0 - Stop storing data.
P0 - Packed mode off.
Q - Marks the end of the program to be stored in the internal RAM.
H - Software **RESET** (autoranging on, dc-volts function, ten power-line-cycles integration time, **AUTO ZERO** on).
X1 - Executes program stored in the internal RAM.

corresponds to measuring 100 nV across the unknown resistor. Thus, thermally-generated error voltages can cause large errors on the more sensitive resistance ranges. By using the offset-compensated ohms function these errors are removed so that low resistance can be measured accurately.

On the other end of the dynamic range, high-resistance measurements have been expanded through use of a calculated-resistance technique. The 100-MΩ and the 1-GΩ ranges use the same signal current as the 10-MΩ range. The unknown resistor is paralleled with a known 10-MΩ resistor in the input switching. Hence, the resistance measured is the parallel combination of the unknown resistor R_x and 10 MΩ. The value of the unknown can then be computed from the formula:

$$R_x = \frac{R_m \times 10\,M\Omega}{10\,M\Omega - R_m}$$

Fig. 14. *All functions and ranges of the 3456A DVM can be completely calibrated from the front panel of the instrument by removing the cover at the right as shown.*

where R_m is the measured parallel resistance.

System Planning for High Speed

Numerous features have been designed into the 3456A specifically for systems use. Included in these is a voltmeter-complete output. This rear-panel digital output switches from a low to a high logic level (from 0 to 5V) when the voltmeter is finished measuring the input signal. The voltmeter-complete signal can be used in conjunction with external triggering and a channel-closed signal from a scanner to make a high-speed scanning loop that is independent of the controller. Also useful in this type of application are a programmable delay between readings and a programmable number of readings per trigger.

HP-IB is standard in the 3456A. All functions available from the front panel are also available under remote control. Some systems features that would not be useful to the bench user are accessible only under remote control. A DISPLAY OFF command allows a faster reading rate because the processor does not have to format the data for the display. A packed mode output speeds up the system reading rate because only four bytes of packed information are sent out, compared to 14 bytes in the unpacked mode.

An internal RAM is available to the systems user to be used for reading storage and/or program memory. Although reading storage can also be accessed from the front panel, program memory is a remote operation only. When there is no program in memory, 350 readings can be stored. Using program memory, any sequence of functions, including reading storage, can be performed without intervention by the controller. This allows readings to be taken and stored in different configurations when the controller is busy doing other tasks. An example of how program memory is used is shown in the box on the left.

Maintenance and Calibration

Extra design effort was expended on the 3456A to simplify maintenance and calibration both electrically and mechanically. Calibration of the complete voltmeter can be accomplished by removing the front-panel calibration plate (see Fig. 14), connecting the correct standards, and adjusting the front-panel settings. There are 14 adjustments: five for dc volts, four for ac volts and five for ohms.

All printed circuit boards containing active circuitry, except the ac board, are readily accessible by removing four screws. To access the ac board, the four screws holding its shield on must also be removed.

Acknowledgments

The authors wish to thank our fellow development team members. Not only did their individual contributions make the project successful, but the group camaraderie made the work fun. Mike Gravitz and Vicky Sweetser developed the logic hardware. They were joined by Gary Ceely in developing the extensive software. Much of the friendly operator interface is due to the mechanical design of Hudson Grotzinger. Grant McFetridge worked long and hard on the reference and fineline resistor circuits.

Special thanks must go to Joe Marriott, our project manager. We would be remiss if we didn't thank Roy Barker, our section manager, for schedule guidance and an environment to get the job done. To the above people and the others who made the production of the 3456A possible, many thanks.

References

1. A. Gookin, "A Fast-Reading, High-Resolution Voltmeter that Calibrates Itself Automatically," Hewlett-Packard Journal, February 1977.

Charles A. Clark

Chuck Clark joined HP in 1977 after earning the BSEE degree from the University of Colorado. The 3456A project was his first assignment and he was responsible for developing the true-rms converter. Then he worked in production engineering and is currently involved with developing new voltmeter products. He is a native of Wichita, Kansas and now lives in Loveland, Colorado. Chuck's interests are live and recorded music, stereo, occasional water skiing, and current literature, which includes some science fiction.

James J. Ressmeyer

Jim Ressmeyer was born in Pipestone, Minnesota and attended the University of Minnesota where he was awarded the BSEE degree in 1972. Jim came to HP that same year and has worked on thin-film hybrids and analog design for voltmeters, including the 3455A DVM. He was responsible for the ohms, input filter, and front-end switching sections in the 3456A DVM and is now a project manager for new voltmeters. Between work and his outside activities, Jim attended Colorado State University as part of HP's SURGE program, earning an MSEE degree in 1975. He and his wife and two daughters live in Loveland, Colorado. Jim enjoys bowhunting, reading, playing racquetball, and backpacking and camping with his family.

Lawrence T. Jones

Larry Jones joined HP in 1970 and worked on MOS development until 1977 when he transferred to voltmeter development and worked on the dc section for the 3456A. Larry attended the University of Colorado where he received a BS degree in physics in 1968 and a BS degree in electrical engineering in 1970. Outside of work he serves as an assistant Scoutmaster and enjoys fly fishing, cross-country and downhill skiing, woodworking, backpacking, and winter camping. Larry is a native of Denver, Colorado and now lives in Loveland, Colorado with his wife and two boys. He and his wife just finished an addition to their home, doing everything from the foundation to the roof.

ABRIDGED SPECIFICATIONS
HP Model 3456A Digital Voltmeter

dc Voltage

INPUT CHARACTERISTICS (6½-digit mode, 10 PLC integration time):
RANGE/MAXIMUM READING/RESOLUTION:

0.1V	0.1199999V	100 nV
1.0V	1.199999V	1 µV
10.0V	11.99999V	10 µV
100.0V	119.9999V	100 µV
1000.0V	1000.000V	1 mV

*Power line cycles

RANGE/MEASUREMENT ACCURACY/TEMPERATURE COEFFICIENT PER °C (±(% of reading + number of counts)): AUTO ZERO on, FILTER off, 24 hours, 23°C ±1°C.

0.1V	0.0022 + 24	0.0002 + 2.0
1.0V	0.0009 + 4	0.0002 + 0.2
10.0V	0.0006 + 2	0.0002 + 0.02
100.0V	0.0011 + 3	0.0002 + 0.2
1000.0V	0.0011 + 2	0.0002 + 0.2

FILTER ON: Rejection is +60 dB at 50 Hz. Add 2 µV for 0.1V, 1.0V, and 10V ranges and 200 µV for 100V and 1000V ranges.

INPUT RESISTANCE:
0.1V, 1.0V, AND 10V RANGES: >$10^{10}\Omega$.
100V AND 1000V RANGES: 10 MΩ ±0.5%.

NOISE REJECTION (1 kΩ unbalance in LO):
ac NMR: 0 dB (0.01 or 0.1 PLC), 60 dB (>1 PLC), 120 dB (>1 PLC with filter).
ac CMR: 90 dB (0.01 or 0.1 PLC), 150 dB (>1 PLC), 160 dB (>1 PLC with filter).
dc CMR: 140 dB.

ac rms Voltage

INPUT CHARACTERISTICS 6½-digit mode, 10 PLC integration time):
RANGE/MAXIMUM READING/RESOLUTION:

1.0V	1.199999V	1 µV
10.0V	11.99999V	10 µV
100.0V	119.9999V	100 µV
1000.0V	700.000V	1 mV

FREQUENCY (FILTER OFF)/FREQUENCY (FILTER ON)/MEASUREMENT ACCURACY (±(% of reading + number of counts)): AUTO ZERO on, >1% of full scale, dc component <10% of ac component, 24 hours, 23°C ±1°C.

—	20 Hz to 30 Hz	0.33 + 300	
400 Hz to 20 kHz	30 Hz to 20 kHz	0.05 + 440	
20 kHz to 50 kHz	20 kHz to 50 kHz	0.15 + 1500	
50 kHz to 100 kHz	50 kHz to 100 kHz	0.53 + 2700	
100 kHz to 250 kHz*	100 kHz to 250 kHz*	5.00 + 6300	

*Frequencies ≥100 kHz are specified for 1.0V and 10V ranges only.

TEMPERATURE COEFFICIENT (±(% of reading + number of counts)): (0.008 + 60)/°C for dc component <10% of ac component. Otherwise use (0.008 + 120)/°C.
DC COMPONENT >10% OF AC COMPONENT: Add (0.05% of reading + 500 counts) to accuracy. For signals with no ac component, use the 1-kHz ac specification.
CREST FACTOR: >7:1 at full scale.
COMMON MODE REJECTION (1 kΩ unbalance in LO): >90 dB, dc to 60 Hz
INPUT IMPEDANCE: 1 MΩ ±0.5% shunted by <90 pF.

Resistance

INPUT CHARACTERISTICS (6½-digit mode, 10 PLC integration time): Filter is not operational in resistance measurements.
RANGE/MAXIMUM READING/RESOLUTION/CURRENT THROUGH UNKNOWN:

100Ω	119.9999Ω	100 µΩ	1 mA
1 kΩ	1199.999Ω	1 mΩ	1 mA
10 kΩ	11.99999 kΩ	10 mΩ	100 µA
100 kΩ	119.9999 kΩ	100 mΩ	50 µA
1 MΩ	1199.999 kΩ	1Ω	5 µA
10 MΩ	11.99999 MΩ	10Ω	500 nA
100 MΩ	119.9999 MΩ	100Ω	<500 nA*
1 GΩ	1000.000 MΩ	1 kΩ	<500 na*

*Ohms source is a 500-nA current source in parallel with a 10-MΩ resistor.

RANGE/MEASUREMENT ACCURACY/TEMPERATURE COEFFICIENT PER °C (±(% of reading + number of counts)): AUTO ZERO on, FILTER off, four-wire ohms, 24 hours, 23°C ± 1°C.

100Ω	0.003 + 24	0.0004 + 2.0
1 kΩ	0.002 + 4	0.0004 + 10.2
10 kΩ	0.002 + 4	0.0004 + 0.2
100 kΩ	0.002 + 2	0.0004 + 0.2
1 MΩ	0.006 + 2	0.0004 + 0.04
10 MΩ	0.041 + 2	0.001 + 0.04
100 MΩ	1.3 + 1	0.16 + 0
1 GΩ	11 + 1	1.6 + 0

NONDESTRUCTIVE OVERLOAD: 350V peak.
TWO-WIRE OHMS ACCURACY: Same as four-wire ohms except add a maximum of 0.2Ω offset.
OFFSET COMPENSATED OHMS ACCURACY: Same as two-wire and four-wire except maximum reading may be reduced by 9% for large offset voltages

Ratio

TYPE: dc/dc, ac/dc, or (ac + dc)/dc. Range is automatically selected.
METHOD: Four-wire with VOLTS LO input common. FILTER is off.
RATIO: Signal Voltage/(Reference HI Voltage - Reference LO Voltage).

MAXIMUM REFERENCE VOLTAGES:
REFERENCE HI: ±12V.
REFERENCE LO: ±9% of reference HI.
REFERENCE HI – REFERENCE LO: ±11.9999V
PROTECTION: ±350V peak.
ACCURACY: Total % signal error plus total % reference error (same as 0.1V, 1.0V, or 10V dc volts).

Reading Rate

INTEGRATION TIME/AUTO ZERO OFF/AUTO ZERO ON: Reading rates per second are with autoranging, math, display, and FILTER off. Output is to internal memory using internal trigger and packed mode. Packed output in place of internal memory adds 0.35 ms to each reading; ASCII output adds 2.3 ms. Values below apply to dc volts and 100Ω through 10-kΩ ranges with preprogrammed settling times. Also, ac or ac + dc volts and 100-kΩ through 10-MΩ ranges with 0.1 second delay.

Power Line Cycles (PLC)	60 Hz	50 Hz	60 Hz	50 Hz
0.01 (4½ digit)	330	290	210	180
0.1 (5½ digit)	210	180	120	100
1.0 (6½ digit)	48	40	25	20
10.0 (6½ digit)	5.8	4.8	2.9	2.4
100.0 (6½ digit)	0.57	0.47	0.29	0.24

Math Functions

GENERAL: Math function specifications do not include error in X (instrument reading) or in entered values (R, L, U, Y, Z). Range of values input or output is 0.000000×10^{-9} to $\pm 1999999 \times 10^{9}$. Out-of-range values send "OL" to the display and +19999999 × 10^{9} to HP-IB.
FUNCTIONS AVAILABLE (for further description see box on page 25): Pass-fail, statistics, null, dBm(R), thermistor (°F), thermistor (°C), scale, % error, and dB.

General

MAXIMUM INPUT VOLTAGE (all dc and ac ranges): ±1000V peak (700Vrms), 10^{8} VHz.
GUARD TO CHASSIS: ±500V peak.
GUARD TO LO: ±200V peak.
MEMORY SIZE: Total size = 1400 bytes. Memory used is one byte per ASCII character plus four bytes per reading stored.
VOLTMETER CONTROL FUNCTIONS: Number of digits displayed (3 to 6), number of readings per trigger, delay time between measurements (0 to 999.999 s in 1-ms increments), and measurement integration time (0.1 to 100 power line cycles).
OPERATING TEMPERATURE RANGE: 0 to 50°C.
WARMUP TIME: One hour to meet all specifications.
HUMIDITY RANGE: 95% relative humidity, 0 to 40°C.
STORAGE TEMPERATURE RANGE: −40 to 75°C.
POWER: 48 to 66-Hz line operation, 60 VA, 100/120/240 Vac +5% − 10% or 220 Vac ±10%.
SIZE (HWD): 88.9 × 425.5 × 527.1 mm.
WEIGHT: Net 10.49 kg, shipping 13.35 kg.
PRICE IN U.S.A.: Model 3456A Digital Voltmeter, $3500.00
MANUFACTURING DIVISION: LOVELAND INSTRUMENT DIVISION
815 Fourteenth Street, S.W.
Loveland, Colorado 80537 U.S.A.

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.