This paper discusses system considerations of error recovery, prevention of data loss, and protection against loss of function. Back-up techniques, problem determination procedures, maintenance procedures, and system features provided to facilitate their respective uses are discussed. Avoidance of the interruption of store operation is emphasized.

Reliability, availability, and serviceability design considerations for the Supermarket and Retail Store Systems

by R. O. Hippert, L. R. Palounek, J. Provetero, and R. O. Skatrud

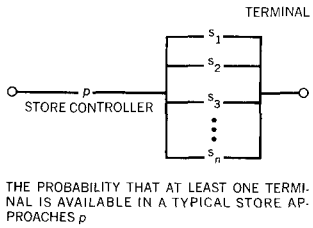
During the past several years, electronic point-of-sale systems have appeared in retail stores and supermarkets in many parts of the world. An important function of such systems is to perform the task of customer checkout, replacing traditional cash registers. Store managers had not concerned themselves with the availability[1] of mechanical or electromechanical cash registers to perform customer checkout because of the number of registers in each store. This high availability, guaranteed in the past by using multiple cash registers throughout the store, presents the designer of the Supermarket and Retail Store Systems with a very formidable problem: how can their availability be made consistent with the satisfactory performance of the tasks described in Reference 2? System design that strives to achieve this goal must consider availability at three levels: strategically, in the design of the distribution of functions and their communication links; tactically, in the design of system components to diagnose and recover from error conditions; and reparatively, in the design of diagnostic and repair procedures in the event of component failure.

This paper covers these three aspects of availability and the design concepts by which the problem can be solved economically. The basic strategic problem is the location of the system controller. The tactical problem is associated with data integrity. The reparative problem has many aspects, which are covered in detail, all concerned with the speed with which operation can be restored after an interruption due to a failure in some component.

McEnroe, Huth, Moore, and Morris have discussed the commonality between, and the uniqueness of, the Supermarket and the Retail Store Systems.[2] Our treatment of system availability is consistent with their description – where areas of commonality are concerned, we consider both systems, and where areas of uniqueness are concerned, we indicate the relevant system.

Reliability and availability of system function

Figure 1 Dependency of checkout availability of a single store on the controller

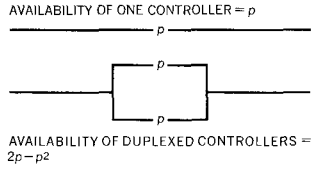


THE PROBABILITY THAT AT LEAST ONE TERMINAL IS AVAILABLE IN A TYPICAL STORE APPROACHES p

The checkout of shoppers entails an interactive communication between the checker, the terminal, and the controller.[3] Failure of any one of the three will interrupt the checking process. Adequate system design requires that these disruptions be of little consequence to the shopper.

While the potential level of checker productivity is a system concern, the level of competency of the individual checker (speed, accuracy, and efficiency) is more a management concern than a system problem, provided the system does not limit the checker's performance.[4] To reduce managerial concerns about the checker, the manager can provide training until the checker has a satisfactory level of competency. The systems described in this paper provide training modes of operation to allow checker training which includes the handling of error-recovery procedures discussed in the last section of this paper.

Figure 2 Parallel controllers



AVAILABILITY OF DUPLEXED CONTROLLERS = $2p-p^2$

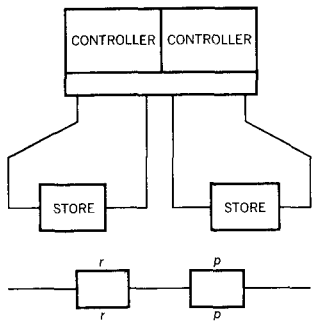
A shopper's transaction is processed by the checker at one of many point-of-sale terminals in a store. Thus, if one point-of-sale terminal fails, another terminal is available to handle the transaction. In the case of the Supermarket System, the controller can be instructed by the system operator[5] to move a transaction from one terminal to another if a terminal fails in a transaction. This procedure is required for a supermarket because the average transaction item count is large. The operation is done without requiring reentry of items previously entered at the first terminal. The principal inconvenience to the shopper in the event of failure of the checkstand is the physical relocation of the unchecked portion of the order to another checkstand.

The terminals are connected serially on loops attached to the controller.[2] Few types of failures will cause a loop disruption, and for most of these, the system will automatically recover rapidly by detecting the terminal that is causing the loop disruption and allowing a bypass branch around it. The remainder require some operator action for recovery. This operation is more fully explained in a subsequent section of this paper.

Controller failure disrupts checkout on all its dependent terminals. The dependence of the supermarket terminals on the con-

troller is strong because the prices of the items sold are stored there.[2] Retail terminals are less dependent on the controller for support of the sales function;[2] the retail point-of-sale terminal can operate off-line in a stand-alone mode as described later in the discussion on retail back-up. The supermarket terminal, however, needs a controller to function; the availability of the system in the supermarket depends on the availability of the supermarket controller: if p is the availability of one controller and the supermarket depends on one controller, then p will be the availability of the system in that store. While p is high, it was determined that some form of controller redundancy is required since an even higher availability was considered necessary for the Supermarket System.

Figure 3 Duplexed remote controllers



r=AVAILABILITY OF ONE PHONE LINE
p=AVAILABILITY OF ONE CONTROLLER
AVAILABILITY OF TWO CONTROLLERS SEPARATED BY TWO PAIRS OF PHONE LINES FROM STORE FUNCTION = $(2r-r^2)(2p-p^2)$

Store sales function availability can be represented by a reliability network. Figure 1 shows the network for a single store controller. In a reliability network, each component is represented by an edge labeled with the component availability. Failure of a component can be represented by the deletion of the respective edge from the network. Component failures are assumed to be independent. The probability of such a deletion is the complement of the component availability. System availability is the probability of the existence of at least one path through the network, where, for such a path to exist, none of its constituent edges may be deleted.

duplexing store controllers

In Figure 2, it can be shown that parallel controllers increase the availability from p to $2p - p^2$. This assumes complete redundancy and isolation including separate power supplies, files, loop adapters, and processors. No cross-dependencies of data required for processing a transaction could be tolerated. Availability would be satisfactory by duplexing controllers. The cost is that of a normally unproductive second controller.

duplexed remote controllers

Another method is to have duplexed remote controllers servicing multiple stores (shown in Figure 3), a configuration that provides redundancy of phone lines and controllers for the store function. If phone lines were completely independent of each other, we would have r representing the availability of each phone line. This configuration would yield store availability of $(2r - r^2)(2p - p^2)$. Unless $(2r - r^2)$ is equal to one, availability is less than that with duplexed controllers but at less cost per store.

It was discovered however, that phone lines do have an element of commonality. Dependencies between two sets of lines can occur in routing. Alternate routing appeared limited and, in some cases, could remove one commonality only to create another. Diversification proved to be expensive in those locations

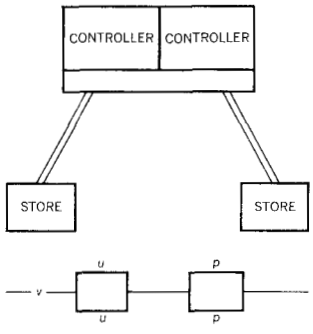
where it could be used and, in many locations, right-of-way problems could prevent its use altogether.[6]

Therefore, the availability of each of a pair of telephone lines, r, had to be equated to vu where u was independent and v was the dependent component. Now the availability (shown in Figure 4) becomes $v\ (2u - u^2)\ (2p - p^2)$. We have replaced a dependent mode of failure by two independent modes, one entailing a line and the other entailing the pair of lines.

paired store controllers

An earlier study on long-haul telephone lines[7] intimated that the component of dependent failure might be too large for a duplexed remote controller system to provide adequate availability. To obviate this problem, the base operation of the store controller was moved in-store with no external phone line dependencies. To improve availability, a paired sister store would have a controller function available via a switched network telephone line connection. This connection would be used only for back-up. Availability, now dependent on a second controller and a phone line with availability of r is $p + pr\ (1 - p)$ as shown in Figure 5.

Figure 4 Availability of remote controllers



IF AVAILABILITY r OF PARALLEL PHONE LINES IS DECOMPOSED TO v FOR DEPENDENT AVAILABILITY AND u FOR INDEPENDENT AVAILABILITY, THEN $v(2u-u2)\ (2p-p2)$

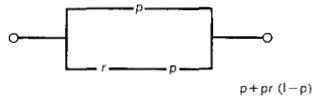
If $r = 0$, the system is equivalent to the single store controller. If $r = 1$, the system is equivalent to duplexed controllers. Clearly, both controllers are productive in normal use. This arrangement was implemented as the supermarket back-up technique. Telephone line availability appeared adequate for the store controller with paired back-up to yield availability sufficiently close to that of a duplexed controller system.

Techniques of implementing back-up procedures are discussed later in a separate section.

Data integrity

Data integrity is the accurate transfer, retention, and manipulation of data. Preservation of data integrity entails protection against both data error and data loss.

Figure 5 Availability of paired store controllers



$p+pr\ (1-p)$

If data is unintentionally modified during manipulation or transfer, an error may occur unless this modification is both detected and corrected by error-checking procedures. These procedures are automatic system functions requiring no human intervention. Internal correction is normally by repetition of the operation, although erroneous characters in scanned data entry[2] may be corrected directly.

If data in storage is unintentionally modified, it may be lost to the system in the store. However, this data is normally recover-

able by retransmission from the host, by manual reentry at a terminal, etc. Some human intervention may be necessary to recover such data.

Correction of errors and recovery of loss cause some interference with the store operation. The design and implementation of correction and recovery techniques determine the magnitude of the interference. In many cases, the interference is imperceptible and detected after the fact only by the presence of an error log that noted the disturbance. The techniques discussed in the remainder of this paper serve to hold the effect and duration of this interference to a satisfactorily low level.

The following discussion will reveal some of the capabilities provided in the systems with respect to data integrity.

power-on diagnostics

If a fault,[8] which may cause a failure, exists in the system, that failure can often be avoided by finding the fault before data is to be transferred. Extensive testing of the Supermarket and Retail Store System hardware is performed at power-on time, and if deficiencies are found, the operator can take the necessary action to correct the problem. A hierarchy of tests is performed to isolate and define faults.

Initial diagnostics are contained in the read only storage (ROS) of the point-of-sale terminals and the store controller. The ROS microcode provides the basic testing capability to determine that the critical paths necessary to allow more complete testing are functional. Diagnostic ROS microcode in the store controller tests portions of the basic machine hardware including data flow registers, read/write storage, and disk read capability and performs a check of the microcode contained in ROS. Here the primary function of the ROS microcode is to test the capability to read from the disk into storage and to execute microcode. The disk contains additional diagnostic microcode, that once loaded, can provide a more complete test of the hardware.

The diagnostic ROS microcode contained in the point-of-sale terminals perform much the same function as in the store controller. If the ROS tests do not complete successfully, the terminal will disconnect itself from the store loop and become passive. The method of disconnection (bypass) is described in a subsequent section. Observe that the primary function of the diagnostic ROS microcode contained in the terminal is to determine if the hardware required to communicate with the store controller is operational.

After sucessful completion of the ROS diagnostics, the point-of-sale terminal will request that the "extended initial microcode load tests" be sent from the store controller. Here again, the

purpose is to provide more complete testing than can be provided by the ROS diagnostic tests. When all diagnostic testing is completed successfully, the system microcode is loaded into each on-line terminal, and the system is then ready for operation.

levels of checking

In the Supermarket and Retail Store Systems, several different data flow checking methods are used. The amount of checking on any type of data transfer depends both on the likelihood of the modification of data and on the effectiveness of checking methods to detect such modification. For instance, the checking of a data transfer over a store loop between the store controller and a terminal is more powerful than that of a storage data transfer. Data checking methods include use of cyclic redundancy checking (CRC), parity, sequence numbers, length checks, and read after write.

Three data checks are performed on the data transfers between the store controller and the attached point-of-sale terminals. The first of these is CRC. On receiving the data, the CRC is again generated and compared to the two CRC characters included at the end of the transfer. Sequence numbers are also provided with each message transferred. The sequence numbers are checked by the receiver to allow detection of missed data blocks. If a sequence error is detected, the sending end is notified, and the last valid sequence number is indicated. This allows the data flow to be resynchronized at the point where the error occurred. Length testing is performed to ensure that the message received is not too long or short. On all detected errors except sequence number errors, the message in error is discarded. Because there will be no response to the message, the sending hardware will retransmit the original message.

Messages sent from a terminal are not discarded until the appropriate response is received from the controller. If the store loop fails during a transaction, the pending message in a terminal will be held until the loop is restored. If the controller fails and back-up is initiated, the pending message will be transmitted to the back-up controller (supermarket). Two input buffers are provided at each terminal. No further data may be input to a terminal until the contents of a buffer are discarded.

When the store controller is communicating with the host system over common carrier lines, all of the checks previously described for the store loops are performed. Some additional format checking is also provided. The store controller does not discard any data that has been sent to the host system until an acknowledgment of good receipt is returned, and if an error is detected, the data is retransmitted. When the store controller is receiving data from the host system, an acknowledgment for the

complete data chain is not sent until all data checks are satisfied and the data is on disk. The error recovery is handled by the line control code or by a higher-level code depending on the type of error.

Data transfers to and from the disk use CRC checking of data fields. In addition, on all write operations, the data is read back and CRC compared to verify that the data is written correctly on the disk.

There are instances where data transfer errors that are not correctable can occur in the controller. For example, a permanent read error from the disk, which occurs when data cannot be read from a sector after it had previously been written successfully, or a parity error during a memory transfer will result in data loss. Retry capability is provided for both types of error, but if all retries fail, the store controller will either stop with a machine check indicated or post a message to the operator that will indicate the action to be taken for recovery. In both cases, the operator will initiate appropriate actions to recover both system operation and the data lost to the system.

power line disturbances

The external supply of power, provided by the power utility to the store, is frequently subject to disturbances that can be due to planned decreases in voltage (brown-outs), momentary drops in power (often discerned by a flicker in the lights), or complete power failures. In areas where power failures are common, a store might provide its own auxiliary power source. Even in such cases, there would be a short disturbance during switchover. If the system were merely to come to a stop on a power line disturbance without some protection designed into the system, the contents of volatile storage would be lost which would both inhibit restart and destroy the integrity of transactions in process.

When system design requires that data be stored in a volatile random access storage, there is a potential risk of data loss. Power line variations can cause the power system to go outside its specified limits, resulting in a modification or loss of the stored data. Power line disturbances severe enough to cause data loss in the typical supermarket and retail environment are expected to occur with sufficient frequency to require protective measures.

Data is protected by means of a dump of volatile storage to disk in the store controller. The line voltage is constantly monitored by the controller. When a power line disturbance occurs, causing the line voltage to drop below a threshold, it is detected by the monitor. Thereupon all in-process operations are halted, and data in volatile storage necessary for the error-free resumption

of controller processing is written onto the disk before the store controller power supplies go out of tolerance. Sufficient capacitance is provided in the power supplies to maintain necessary output voltages for the length of time needed to write the data onto the disk. Once the procedure is initiated, it will continue even if the line voltage again rises above the threshold.

When power is restored, the saved information is retrieved during the initial microcode load process and is written back to volatile storage from the disk. System microcode is reloaded from the disk, and the store controller is allowed to continue processing where it left off.

Terminals in the same store as the store controller will probably also have experienced the same power disturbance. In that event, after power is restored, the terminal will execute the power-on diagnostics contained in its ROS and then obtain from the store controller an initial microcode load, together with current data on any in-process checkout operation. Thus, the checkout process can resume where it left off. Supermarket terminals in back-up mode that did not experience the disturbance, supported by a remote controller that did experience the disturbance, will wait out the disturbance and then resume operation when the controller is ready. Retail terminals on a remote loop that did not experience the disturbance will continue to operate off-line until the controller is ready.

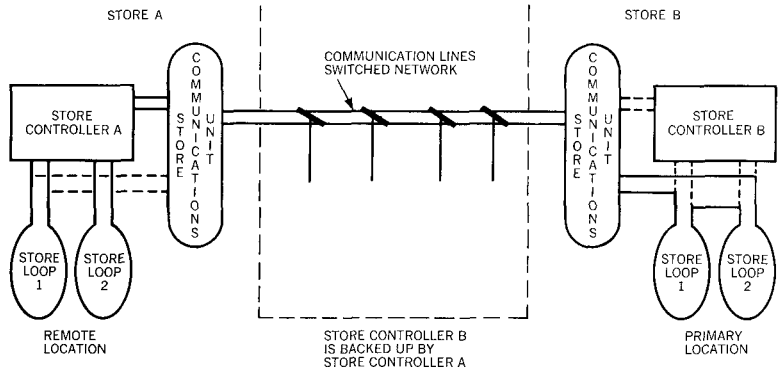
Back-up procedures

supermarket back-up

The paired back-up method discussed in the first section is the form of redundancy chosen for the Supermarket System. The *primary* store receives the back-up; the *remote* store provides it. Obviously, roles are interchangeable when and as necessary. (Refer to Figure 6 for the back-up configuration.)

To establish the back-up mode in the Supermarket System, the operator changes a switch setting on the store communications unit attached to the primary controller and dials the remote store. This switch setting will disconnect all terminals from the primary controller and reconfigure the two separate loops into one loop connected directly to the communications unit. Entry into back-up mode is automatic after dialing the call. When the call is received at the remote store, the remote controller assumes it was a call from the host and waits for a message. Since the call was not from the host, there will be nothing initially received, and by virtue of a time-out mode, with no data received, the controller now assumes it is a request for back-up. It will reconfigure for the additional terminals and then poll[2] them requesting all the necessary totals that the remote controller needs to continue their processing. When the controller has all the totals, it will start polling normally to allow the checkout process

Figure 6 Supermarket back-up configuration



to continue. The transactions in process are picked up at the point of interruption and continued on the remote controller.

To come out of the back-up mode, the store operator restores the back-up switch to the normal position without regard to any activity on the loops. The local store controller then polls the terminals for all of its totals and establishes a reference point for continuing operation. At this time, however, the telephone line remains connected to the remote controller for reconciliation purposes. This reconciliation is actually a transfer between remote and primary controllers of totals accumulated during the back-up period. With this data, the repaired controller can now handle all the store support procedures required for normal store accounting functions (operator reconciliation, tender reconciliation, closing reports, etc.).

retail back-up

Retail store back-up differs from supermarket back-up. Due to limited price look-up procedures in the Retail Store System, off-line use of the point-of-sale terminal in the event of controller failure is adequate for most installations. This method, therefore, will be the usual back-up scheme with the terminal basically assuming the role of a cash register. Other system functions normally provided by the controller are not supported in back-up mode. During back-up operation, the terminal prints all transactions on the journal tape, one of three print stations on the terminal printer. The journal tape is marked for each off-line transaction recorded. In order to maintain data integrity, all marked data should be subsequently manually entered into the system, either keyed in through the terminal or keypunched and entered at the System/370 host.

The retail terminal automatically detects the need to go into back-up either when it sends a transaction message to the controller for which it expects an answer and gets none or if it receives no

polls at all. Then all terminals will go into the time-out mode and will be ready to process transactions off-line until recovery on-line can occur. The terminal in off-line mode continues to monitor the store loop for a poll. The first poll recognized after the off-line transaction is completed will induce it to go back on-line.

Serviceability considerations

hardware servicing philosophy

Maintenance and repair of the system by traditional methods would require a high level of skill by the serviceman[9] and very lengthy repair operations in an environment where rapid recovery from failure is necessary.

Three basic elements of the hardware servicing philosophy are:

1. The system detects its own failures whenever practical and directs store personnel, who need not be skilled in data processing operations, to the appropriate recovery technique.
2. The system contains hardware and functions to assist the same store personnel to identify the failing system unit using problem determination procedures so that the appropriate service organization can be called.
3. The system contains hardware and capabilities to permit the serviceman to repair faults, using a maintenance package described later, without being involved in details of hardware or code.

The techniques of diagnosis and recovery from various component failures are discussed below.

recovery from loop failure

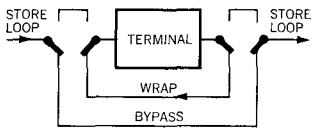
Since terminals are located serially on the loop, methods are provided to bypass a terminal on the loop when necessary without opening the loop. Normally a terminal will be so bypassed when it is switched off (Figure 7). The terminal bypass is also used to automatically locate faults that cause loop disruption.

Figure 8 illustrates a terminal operating normally in the loop. Two bypasses are shown in an open position, set thus by two pairs of relays that also maintain the terminal on the loop. If a fault causes disruption at some point on the loop, no signals will be received by terminals below the fault. Furthermore, since the controller will also not receive the end of the message sequence, it will not put out a new message sequence or poll to that loop until sufficient time has elapsed for the terminals to perform the tests to be described shortly. Consequently, after a short time, the terminals above the break will receive no more signals. The quiescence of the loop will induce every terminal to switch its relays into the position of Figure 7, in which the loop is closed

with the terminal bypassed and the terminal itself is on a closed wrap circuit.

The terminal will test its ability to pass signals through itself on the wrap circuit. If it succeeds, it will restore its configuration to that of Figure 8 and transmit its own identification on the loop (beacon) until it receives some signal from a terminal above that is doing the same thing. Thus, if the failure occurred in a terminal and is discernable by the wrap test, the loop will be restored with the failed terminal bypassed, each beacon silenced by the beacon from the preceding terminal, and the first beacon silenced by a message sequence from the controller. If the failure is not discernable by the wrap test and the loop remains open after the tests, the controller will receive a beacon from the first terminal below the break. This beacon will cause the identification of the terminal issuing the beacon to be displayed at the controller for the benefit of the system operator. In the Retail Store System, all the terminals on the loop will go into off-line mode if the loop is not restored (see discussion on retail backup). In both systems, if the loop is not restored, the system operator will be alerted so as to be able to follow an established problem determination procedure, discussed next, using the beacon identification as a guide to restore the loop.

Figure 7 Bypass and self-wrap test

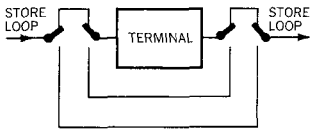


problem determination procedures

The methods provided to the user for isolation of problems are called problem determination procedures. They are used in conjunction with system functions provided specifically for their use, enabling the system operator to isolate a problem rapidly and efficiently to the point where the user can call the appropriate service organization. Since other companies such as telephone companies or manufacturers of ancillary equipment may have equipment attached to the systems, the identification of the responsible service organization is important. After a fault is isolated, the procedures assist the operator in bypassing the failure and recovering system operation whenever possible in the event that automatic system recovery does not take place.

A problem determination procedure is a set of keyed procedural steps used by the system operator at the controller or at a terminal. The controller and terminals contain functions that the operator can execute. The results of function execution indicate the next step to be taken. The last step of any sequence of steps is an instruction to call a specific service organization to repair a located fault.

Figure 8 Relays set for normal operation



Typically, a problem determination procedure used on a failed terminal works in a prompting mode through the terminal printer. Both guidance messages and queries or instructions are printed, and the operator responds using the keyboard. For example, suppose that a supermarket terminal failure is suspected with re-

spect to a coin dispenser, which may be ancillary equipment provided by a manufacturer of coin dispensers. While executing the problem determination procedure, routines are loaded from the controller disk into storage of the failed terminal. One such routine, for example, tries to dispense $1.17 and initiates another routine that asks the operator (via the printer) whether $1.17 has been dispensed. Depending on the operator's response (keyed on the keyboard), the next routine is called, which may print a message that informs him what his next action should be. Finally, the message printed on the printer informs him whom to call for repair.

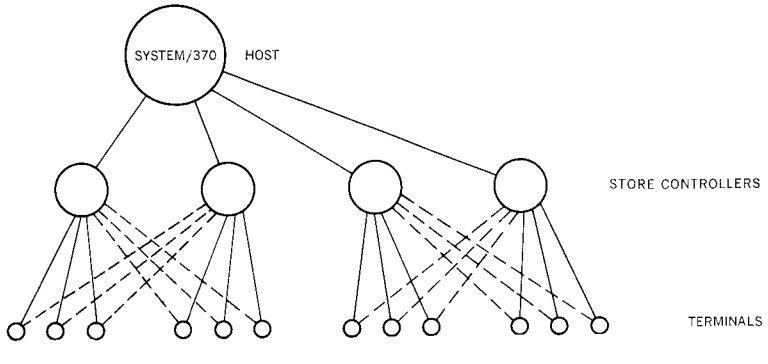
service procedures

The maintenance package contains a set of procedures and a set of functions: MAPs (Maintenance Analysis Procedures) and diagnostic microcode. The MAPs instruct the serviceman when and how to call for the particular module of the diagnostic microcode and how to use it once he has it. The maintenance package tests the hardware of the systems rather than the execution of their functions—the former is a more efficient procedure, [10,11] The diagnostic microcode is used not only for failure detection but also for fault location.

The diagnostic and maintenance package was designed for both solid (hard) and intermittent failures and faults. An important feature of the diagnostic microcode is that it contains provisions for looping. When the serviceman is locating a fault causing an intermittent failure, the MAPs instruct him to loop the proper portion of the diagnostic microcode. In some cases, they specify the number of times the loop should be executed, in others, the microcode loops until terminated by the serviceman. If the looping diagnostic code detects the failure or locates the fault, it stops and provides the serviceman the answer to a question asked by the proper block in the MAPs. An analytical indicator card that has been designed for the point-of-sale terminals and store controller is used to indicate results of diagnostic tests. A sequence of steps will finally lead the serviceman to the step that specifies which field replaceable unit contains the faulty component. Because the Supermarket and Retail-Store Systems use a system maintenance philosophy approach to the maintenance of the in-store hardware, the MAPs for all elements of the in-store system use consistent conventions and the same structure.

To assist in isolating the source of intermittent errors in the system, error logging is provided. Whenever an error is detected, it is stamped with the date and time and logged on the controller's disk. Entries into the error log are grouped into several "buckets," and errors are recorded "first in—last out." Therefore, a very frequent intermittent fault or a solid fault causing an intermittent failure will not fill the whole disk space reserved for error logging while a record of the most recent failures is always

Figure 9 Node structure of Supermarket System



available. When the serviceman is trying to locate a fault causing an intermittent failure, the MAPs instruct him how to call for the proper portion of the error log and how to interpret the output.

maintenance support strategy

The diagnostic microcode is, of course, contained in the system. A difficulty may arise if the serviceman needs a function that is inaccessible because of machine failure. This difficulty is avoided in the following way. The whole Supermarket System may be pictured as a structure of nodes representing the distribution of microcode as shown in Figure 9. The Retail Store System is essentially similar except for back-up connections, but there the terminal is a lowest-level element. The maintenance support strategy is to minimize the dependency on the computing power of a higher-level node. If a node can function without the higher-level node, the diagnostic code for the lower-level node function is available even without communication with the higher-level node. This holds even if the function is that of communication with a higher-level node. For example, because the printer of the supermarket terminal cannot function without both the terminal and the controller, the diagnostic microcode for the printer and printer adapter can reside on the controller's disk.

Some faults prevent the serviceman from calling diagnostic microcode from the controller. In this event, that particular portion of the terminal maintenance package is designed such that the serviceman can service failures caused by such faults by following the MAPs only, without microcode support. For example, some faults within the keyboard could prevent the serviceman from typing the sign-on procedure, required to call diagnostic microcode from the controller.

preventive maintenance

The system was designed such that neither scheduled preventive maintenance nor unscheduled preventive maintenance are required. This design increases the overall system availability while decreasing the service cost.

Summary comment

Adequate system availability is provided by suitable redundancy of components. The general form of redundancy chosen is a composite form in which the reserve component is operating on its own behalf while ready to take over the tasks of a failed component. In particular, the point-of-sale terminal can pick up the operation of another terminal if failure occurs in midtransaction, and the supermarket controller can assume the load of a failed controller in another store. The retail point-of-sale terminals can operate off-line in the event of controller or loop failure. Each supermarket is served by two loops so that half the number of terminals in the store will be unaffected by a loop failure.

Techniques of preserving data integrity are discussed. Particular emphasis is placed on the avoidance of interruption of store operation. Preoperative techniques (power-on diagnostics, avoidance of effects of power line disturbances), multiple levels of checking, automatic methods of error correction, and procedures for rapid recovery of data loss are used.

Three levels of functions are contained in the system hardware and microcode to assist in rapid recovery from component failure: automatic testing and system recovery, functions to assist the system operator using problem determination procedures to locate faults, and functions to assist the serviceman using MAPs to diagnose and repair faults. Examples are given of their use.

CITED REFERENCES AND FOOTNOTES

1. J. E. Hosford, "Measures of dependability," *Operations Research* **8,** No. 1, 53 (1960): "Interval availability is the expected fraction of a given interval of time that the system will be able to operate . . ." Limiting interval availability is the long-run expected fraction; availability in the context of this paper is limiting interval availability.
2. P. V. McEnroe, H. T. Huth, E. A. Moore, and W. W. Morris, III, "Overview of the Supermarket System and the Retail Store System," in this issue.
3. D. C. Antonelli, "The role of the operator in the Supermarket and Retail Store Systems," in this issue.
4. W. C. Metz, Jr. and D. Savir, "Store performance studies for the Supermarket System," in this issue.
5. The system operator is one of the store personnel responsible for loading the system, changing operating modes, and handling exceptional procedures.
6. *Bell System Data Communications—Technical Reference*, "Transmission specifications for voice grade private line data channels" (March 1969).
7. J. Provetero, "Availability of voice grade private wire telephone lines," *Proceedings of the IEEE Electronics Conference*, 392 (October 1971).
8. A failure is an occurrence of an inability of an item to perform its function when called upon to do so. Failure is an event, not a physical state of an item. It is solid, or hard, if it occurs each time the component is called upon to perform its required function; otherwise, it is intermittent. A fault is that physical condition or state of the component that may cause failure.
9. Serviceman as used here refers to the maintenance personnel for the equipment installed. Usually each vendor would have their own personnel; for IBM, they are the customer engineers.

10. R. D. Eldred, "Test routines based on symbolic logic statements," *Journal of the ACM* **6,** No. 1, 33–36 (1959).
11. R. A. Marlett, *On the Design and Testing of Self-Diagnosable Computers,* Report R-293, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois (1966).

Reprint Form No. G321-5006