REPORT:

Marcy R. Braunstein *is a supervisor in the Advanced Services & Architecture Planning Department of AT&T Bell Laboratories in Holmdel, New Jersey.* ***Christopher L. Burton*** *is a member of the technical staff in the Planning Systems Department of Bell Laboratories in Holmdel.* ***Sandra D. McNabb*** *is a supervisor in the Network Services Concepts Department at Bell Laboratories in Holmdel. Ms. Braunstein is responsible for planning new customer control capabilities for AT&T network management products and services. She has a B.S. in applied mathematics from Brown University and an M.S. in operations research from Columbia University. Mr. Burton is working on advanced services prototyping and development. He has a B.A. in psychology from Williams College. Ms. McNabb is responsible for*
(continued on page 31)

ASQIC 800 CALL DATA MASTER

Motivation

The *advanced services quick implementation capability* (ASQIC) program provides for quick market tests of new AT&T service concepts. The Marketing Service Concepts group in AT&T Communications and Information Systems created the ASQIC program in 1985 to gain early customer feedback on sophisticated new service concepts. Service concepts that appear to be attractive to the marketplace but need further definition and evaluation are appropriate for an ASQIC trial. The concept must also be one that can be realized quickly.

The Marketing Service Concepts group selects a single, strategic AT&T customer to work with the ASQIC team and define the features that will be available during the trial. Trials run for about 90 days with customers evaluating trial features during and after the trial. All trial equipment is removed after the trial.

With ASQIC, AT&T hopes to increase the success of future services and, at the same time, create a strategic partnership by giving the participating customer information about our future plans and a chance to shape a future service.

First Application

The first concept tested via the ASQIC process was a proposed feature of AT&T advanced 800 service. This trial system, known as 800 *call data master* (CDM), collected call-attempt and completion data in real time for 800 service. It then processed the data and displayed information to the customer via an interactive, multicolor graphics interface.

The call data master consisted of three different user tools for: real-time monitoring and decision making, summary displays, and reports. The tools were jointly defined by the J. C. Penney Company, Inc., the AT&T Bell Laboratories system engineers, and AT&T Marketing Planning.

Real-Time Monitoring and Decision Making. The *real-time monitoring and decision-making tool* was a load-balancing tool displaying real-time traffic data in three ways:

- A map of the United States showed: (a) the number of call attempts terminating at each WATS band at each customer answering location and (b) the percent of unanswered calls at each band. (See Figure 1.) Each band is represented by a box. The number of call attempts to the answering location in the last fifteen minutes is shown above each box. The percent of calls in the last fifteen minutes *not answered* is shown by the height of the colored bar in the box and by the percent value inside the box. If all calls to a location were unanswered, the colored bar filled the box completely.
- A bar display showed the number of calls originating from each NPA ("numbering plan area" or area code) terminating at each WATS band at each answering location, and the percent of calls from each originating NPA to the answering band or location.
- A display of the customer's advanced 800 service call processing tree showed the number of attempts and the percent of attempts at each branch of the tree. The tree represented the software logic that determined how the customer's calls were routed through the AT&T network.

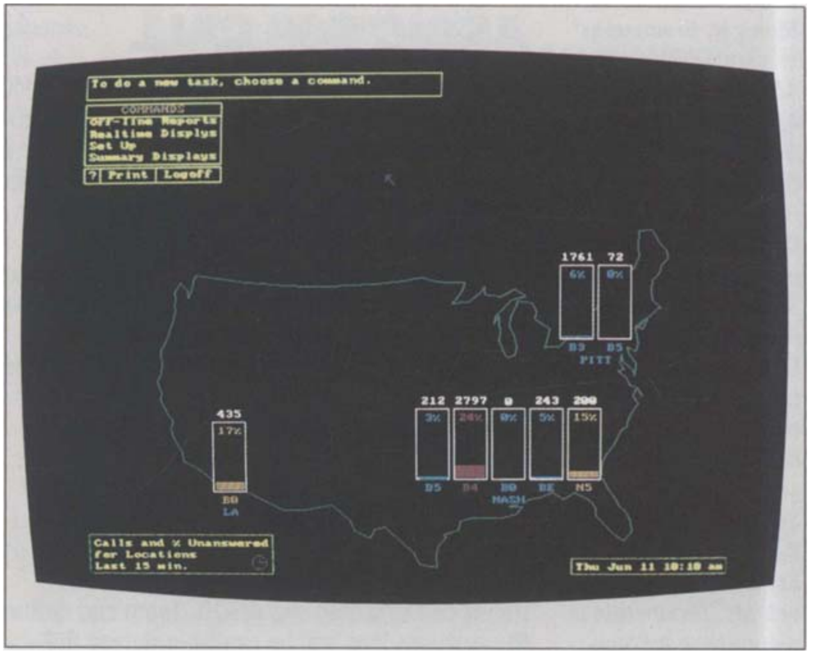

Figure 1. The U.S. map display used in the trial showing the number of call attempts terminating at each band at each customer answering location in the last fifteen minutes. The percent of unanswered calls at each band is shown both as a numerical percent value and in bar graph display. Red = 20 percent or more unanswered calls; yellow = 10 to 20 percent; blue = less than 10 percent. The user could change the threshold at which the display data changed colors.

All three real-time displays were updated every minute with the last fifteen minutes of traffic counts.

Summary Displays. The *summary displays tool* summarized traffic data over any customer-specified interval of time within the last 30 days. Data could be sorted by call termination, call origination, day of week, or time of day. The summary displays used line and bar graphics.

Reports. The *reports tool* provided a hard copy summary of the number of call attempts to the 800 service number for any 24 hours in the previous 30 days. The attempt data could be sorted by call origination or call termination. Reports could be printed or written to a PC hard disk on the customer's premises.

Customer Selection

The customer trial ran from November 1986 through January 1987. JC Penney is a large 800 service user with several applications that are served, depending on the size of the application, by three to fourteen telemarketing centers. A sophisticated user of advanced 800 service features, JC Penney makes frequent changes (sometimes 10 to 20 changes a day) to their advanced 800 service call processing tree and uses automatic call distributor (ACD) data to determine when answer locations are overloaded and, consequently, when changes should be made. However, JC Penney could not measure calls that are blocked from reaching the answering locations because of inadequate numbers of 800 lines.

The CDM provided JC Penney with

these data as well as the three tools described. JC Penney used the real-time monitoring and decision-making tool provided by the CDM to help determine what changes should be made to their call processing tree. For example, if the U.S. map display showed that a telemarketing center was receiving more calls than the attendants could handle, the CDM could help JC Penney determine how to off-load traffic to another telemarketing center. The bar display showed the customer what traffic (by originating NPA) could be off-loaded. The tree display showed what alternate routing schemes could be easily achieved by changes to the call processing tree.

JC Penney used the summary displays and reports tools to track promotional responses and to determine future staffing requirements. Staffing decisions had been based on the ACD data only.

Development Strategy

There were three major variations in the systems engineering approach in this project. The CDM architecture was designed specifically to meet an availability date 10 months after project kick off. Thus, architectural components that could not meet this availability date were eliminated. Secondly, a single systems engineer was responsible for functional requirements, detailed design requirements, and systems testing so that no start-up time was needed between the different stages of the project. Finally, requirements were written with significant input from the developers so that development difficulties could be assessed easily.

Some important characteristics of the CDM development that resulted from this short project time line were the need for a small, highly focused team of developers, phased (incremental) development and testing, and the use of an architecture that could be integrated into the existing AT&T network architecture with little or no effect on the network systems that support standard services. Because the CDM was only a prototype, nonstandard hardware and software were used to set up the trial quickly. The goal of the trial was to evaluate the service concept, not the service architecture.

The CDM development team was initially made up of three members; eventually there were five developers. Developers had to be willing to take risks in order to meet deadlines. At the start of the project, the CDM had no interfaces to any data collection systems. Clearly, it was technically feasible to develop these interfaces; however, there were major unknowns about the effect of these interfaces on production systems and whether they could be put into place in the required time period. In addition, support from several development organizations was needed on short notice to modify hardware and software for the trial. In particular, the AT&T Network Services, Signaling and Administration Laboratory in Columbus, Ohio and the AT&T 800 Service User Support System (USS) development group in Somerset, New Jersey provided support.

The CDM was developed in four consecutive software/hardware increments, known as the "red," "blue," "yellow," and "green" systems. Each development phase was followed by a testing period, which grew longer with each increment. The early phases were to prove the feasibility of the external interfaces to the main processing unit and to design internal interfaces between the major subsystems in the

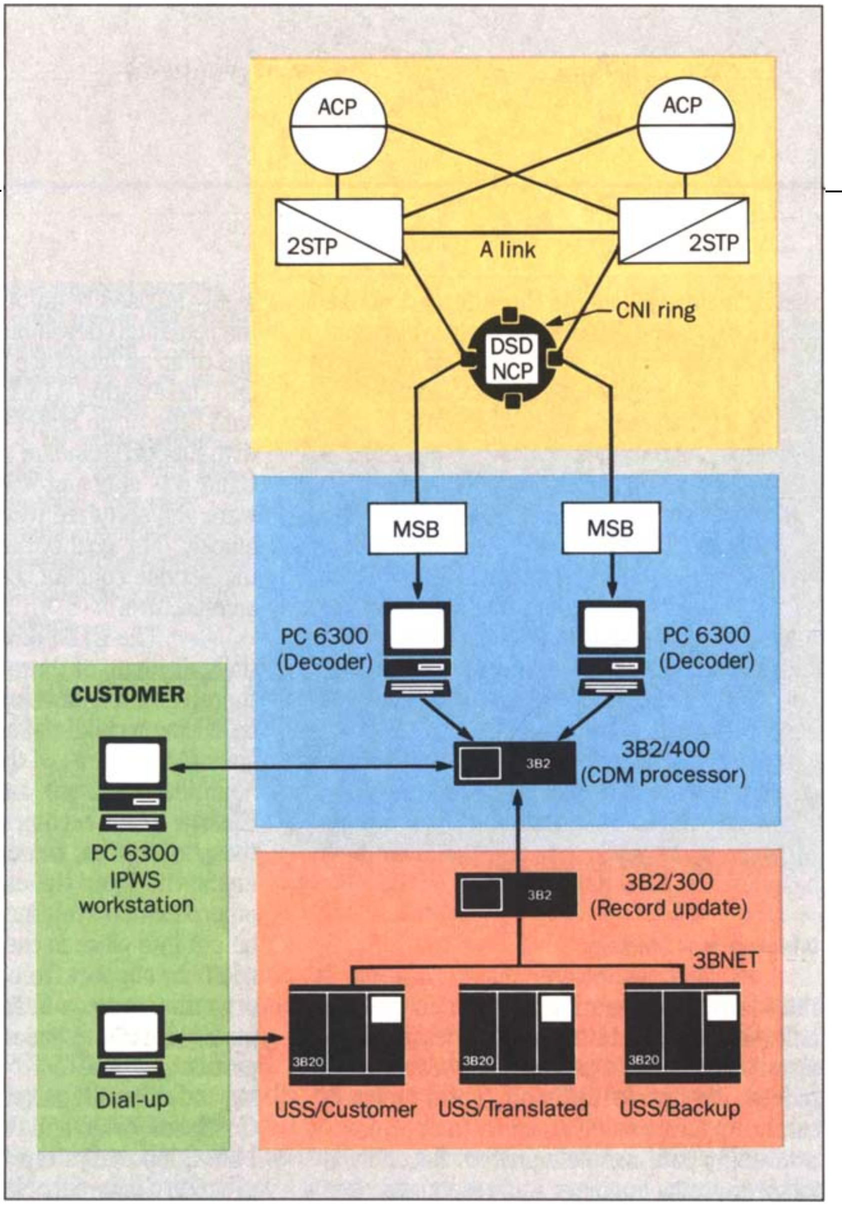

Figure 2. The call data master hardware architecture as it was superimposed on the advanced 800 service call processing architecture and the routing control service architecture. ACP = action control point; STP = signal transfer point; DSD = direct services dialing; NCP = network control point; CNI ring = common network interface; MSB = message sampling board.

software architecture.

There was only one designated system tester; however, the systems engineer and human factors expert helped in testing the later development increments. At the end of each development increment, the system tester moved the current system to a separate machine for testing. Formal modification requests were generated and tracked for each increment. Most of the call data master's code was maintained under the UNIX® source code control system (SCCS).

The hardware and software architectures of the CDM were designed for a single customer—not to be expanded to provide the same service for several AT&T customers. The CDM prototype was a "throw-away" system, designed simply to test a service concept.

Hardware Architecture

The CDM hardware architecture was superimposed on the advanced 800 service call processing architecture[1] and routing control service (RCS) architecture. The CDM architecture included the following components: message sampling boards (MSBs), AT&T PC 6300s, a CDM processor, a record update processor, and an interactive planning workstation (IPWS). (See Figure 2.)

The MSB was originally designed by the AT&T Network Services, Signaling and Administration Laboratory as a diagnostic tool for debugging common channel signaling system 7 (CCS7) messages in the network control point (NCP) link node. However, the MSB was unable to process the CCS7 messages quickly enough to handle the customer's call volumes. The MSB was enhanced for the trial, increasing the output capacity to 56 kilobits per second (kb/s) and reformatting the data in binary. As part of the CDM architecture, the MSB read all messages (in both directions) in the link node, copied the messages and transferred the copies to the AT&T PC 6300. The MSB was a read-only board; it could not write any messages onto the node processor bus. Two MSBs were required, one for each link node. The MSBs were located in the Dallas A3G NCP in Dallas, Texas.

The PC 6300s buffered and processed messages from the MSB and transmitted messages to the CDM processor. PC 6300s were selected because they were the smallest dedicated processors able to handle the 56-kb/s output from the MSBs. Two PC 6300s, also located in Dallas, were required for this function—one for each MSB.

The CDM processor was an AT&T 3B2 Model 400 minicomputer—the heart of the system. It had a variety of functions including generating call-attempt records from the direct services dialing (DSD) messages, emulating the NCP's routing decisions, generating reports and displays, storing all data for up to thirty days, and keeping a trouble log for maintenance purposes. The 3B2/400 machine was chosen because of its UNIX System V, version 2 (V.2) operating system, the availability of I/O expansion cards and its disk storage capacity. The CDM processor was located in Dallas.

When the customer made a change to the advanced 800 service call processing tree, the record update processor obtained an update from the advanced 800 user support system (USS). The record update processor was set up as a read-only node on the USS 3BNET (a local area network). It was an AT&T 3B2 Model 300 machine, the least costly machine that is supported by 3BNET. The record update processor received updated customer record files from the USS and transmitted them to the CDM processor via dedicated facilities. In this way, the security of the USS was not compromised and the CDM processor obtained an updated copy of the customer record as soon as it had been received by the NCP. The record update processor was located with the USS in Freehold, New Jersey.

An IPWS color graphics workstation was provided to the customer (in Milwaukee, Wisconsin) to present all real-time and historical displays of their advanced 800 service traffic and call processing logic. The workstation was composed of an AT&T PC 6300 (with a graphics board), a 19 1/2-inch color monitor, and the standard monochrome monitor. A color graphics printer was also attached to the PC so that the customer could produce hard copies of displays that appeared on the color monitor.

Using both the PC mouse and keyboard made the call data master an interactive system.

Color was used extensively to make the displays more informative. (See Figure 1.) The user could change the thresholds at which the displayed data changed color. The system had menus to access various displays and functions of the CDM.

Software Architecture

Figure 3 shows the call data master's software architecture mapped onto the hardware architecture. The CDM was a multiprocess UNIX-based system designed to:

- Interface to the message sampling boards
- Interface to USS's advanced 800 routing control service (RCS)
- Emulate the NCP's call routing process
- Provide a real-time interactive color graphics user interface
- Maintain a database of call-attempt and completion activity
- Generate daily reports
- Perform self-maintenance.

Most of the software development took place on an AT&T 3B20D machine. The software was then down loaded to the 3B2/400. Software was also developed for the PC 6300s and for the 3B2/300 (the record update processor). Both 3B computers ran UNIX System V.2, and the PC 6300s ran Microsoft Corporation's MS®-DOS operating system. All coding on the UNIX machines was done in C programming language. Code for the PC 6300s was written in assembly language to enable the 6300s to keep up with the 56-kb/s lines from the message sampling boards.

Certain features of the UNIX operating system were particularly useful in setting up the CDM. For example, shared memory on the 3B2/400 was used to hold the system's real-time data and also served as a medium for interprocess communication. The UNIX clock process or daemon, *cron*, scheduled reports, summarized data to disk from shared memory, and performed routine maintenance activities. The UNIX-to-UNIX copy command, *uucp*, transferred call processing trees from the record update processor to the 3B2/400.

The IPWS graphics workstation was used to develop the user interface. IPWS enabled the developers to perform device-independent graphics. In particular, it gave the developers the CORE graphics standard, extensive graphical menu management software, and a plots-and-charts subroutine package.

The following sections describe the basic subsystems of the CDM.

The Message Sampling Board Interface. The interface to the message sampling boards provided the CDM with all real-time call-attempt and completion data. CCS7 messages forwarded by the MSBs contained information for all 800 service calls routed to JC Penney by the NCP. Two processes called "decoders," which ran on the PC 6300s, filtered out messages that were not part of the advanced 800 service and recognized CCS7 messages relevant to JC Penney's traffic. The decoder processes condensed these messages, stripping out unnecessary fields in the CCS7 messages, and then buffered them on the PCs. Through the use of a polling protocol, these messages were forwarded over the 9.6-kb/s links to the 3B2/400.

The process on the 3B2/400 that polled the PCs for information, known as "read PCs," used a hashing algorithm keyed on a

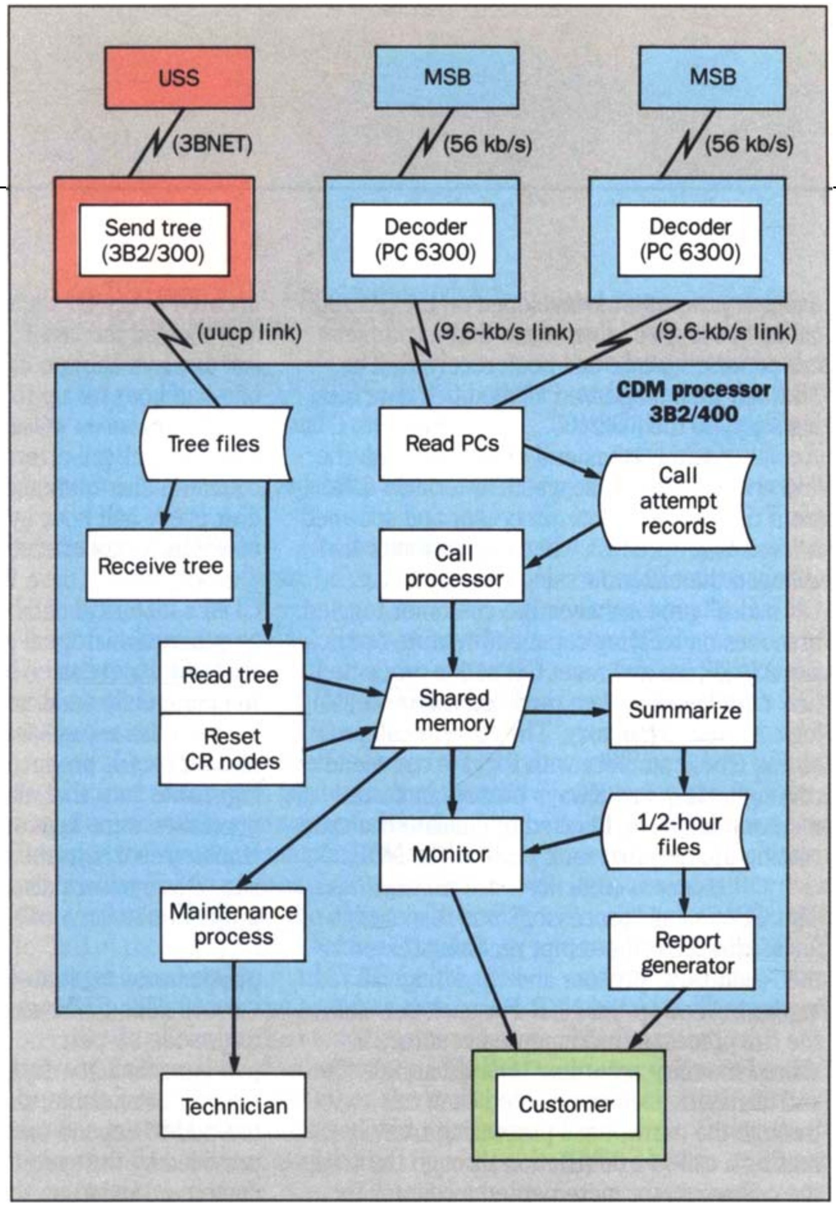

Figure 3. Call data master's software architecture mapped onto the hardware architecture.

CCS7 field called the conversation ID to build call-attempt records from individual messages sent by the decoder processes. Call-attempt records generated by this process included the following information: originating NPA, time of call, up to ten caller-entered digits, and call-completion status (answered or unanswered). Call-attempt records were written to disk, where they could be retrieved and processed.

The Interface to Routing Control Service. To display the customer's current call processing tree, the user support system (USS) developers modified the routing control service (RCS) pending record management system so that it copied several data files to the record update processors over the USS 3BNET. These files were copied each time the customer changed the call processing logic. A daemon process was developed on the 3B2/300 called "send tree" to confirm that a complete call processing tree had been received. The 3B2/300 then forwarded all modified data files, via *uucp*, to the 3B2/400.

On the other end of the *uucp* link the "receive tree" process watched for new data from the record update processor and spawned a "read tree" process when the customer had changed the call processing tree, or a "reset CR nodes" process when the customer toggled branches on existing command routing nodes. Both read tree and reset CR nodes confirmed new data files and then modified call processing logic in shared memory. The current call processing tree, complete with toggled command routing nodes, was always present in shared memory and could be used to emulate the call-routing process that took place at the NCP.

Emulation of the NCP's Call-Routing Process. The CDM's call "processor" was responsible for reading all call-attempt records created by the "read PCs" process and emulating call routing performed by the NCP. For each new call, the call processor incremented counters in shared memory according to originating NPA and destination. It then routed each call through the current call processing tree. In routing a call to a destination through the tree, the call processor incremented a counter for each branch taken so that the customer could analyze call processing strategy in real time. All real-time call-attempt and completion information was provided to the user interface through shared memory.

The User Interface. The user interface module, called the "monitor," generated real-time and historical displays of both the customer's call processing tree and call-attempt and completion data. Real-time displays were updated every 60 seconds from shared memory and showed the last 15 minutes of data. Historical displays showed data with a resolution of one-half hour for up to 30 days.

Database Management and Report Generation. Current call-attempt and completion counts in shared memory were written to hard disk every half hour by the "summarize" process, which was executed by the UNIX clock daemon, *cron*. These files represented the CDM's historical database. Besides being used to generate historical graphs via the monitor, these files were used by the report generator to create daily summaries of call activity.

Self-Maintenance. The system's maintenance process produced a system-wide error log, made sure that all the system's ongoing processes were kept alive, and sent alarms and trouble reports to the technicians at the Dallas site. Alarms were also sent to a member of the project team via a call-paging system.

Performance Objectives

The CDM architecture was designed to provide 95-percent availability and 99-percent reliability. If the system were frequently unavailable, the user would be less inclined to depend on and use it. If the data provided by the system were inaccurate, the customer might use the "bad" data to make "bad" decisions. For example, if the data showed that more calls were terminating at a particular location than was accurate, the user might decide to off-load that location, potentially overloading another location and thereby losing calls.

Call Data Master Availability. Availability was defined as the probability that all system features be accessible to the customer during the customer's normal operating hours (7:30

a.m. to 7:30 p.m. CDT, Monday through Friday). CDM availability was given by:

$$A\,[CDM] = \frac{\text{Operable Time}}{\text{Total Trial Time}}$$

To meet a 95-percent availability objective:

$$0.95 = [CDM] = (A\,[Hardware])\,(A\,[Software])(A\,[Network])$$

Hardware availability. Hardware availability was calculated by determining the expected *mean time between failures* (MTBF) and the *mean time to restore* (MTTR).

Mean time between failures (MTBF) was estimated for each hardware component based on manufacturer's specifications. Because the message sampling board (MSB) is not standard network equipment, no MTBF was available. However, the circuitry is similar to that of the encrypted version of the link interface board and so the MTBF for the message sampling board was based on this.

To minimize the mean time to repair, spares of some of the hardware (e.g., MSBs) were available on site. For other hardware, a service contract was purchased with the repair time specified to meet availability requirements.

In summary, the total estimated downtime because of CDM hardware failure was 14.37 hours out of a total of 2160 trial hours, or an availability of 99.3 percent.

Software availability. Software availability was defined as the probability that the program functioned successfully according to the requirements at a specific time.

Because all software was new and was developed specifically for the trial, there was no source of reliability data. To estimate the reliability of the software, the developers independently estimated the likelihood of each type of failure and the time to restore the system after the failure. Total estimated downtime because of software failures was 50.10 hours during the 2160 trial hours, amounting to an availability of 97.7 percent.

Network availability. A network failure was defined as a failure of the trial network control points (NCP) or a failure in either of the 2STPs (signal transfer points) and their associated A links. The CDM monitored only a single NCP of the NCP pair. If the trial NCP failed, no data would be collected. If a single 2STP or A link failed, all messages would be routed to the NCP by the secondary 2STP or A link. A CDM failure could result if the increased traffic volume over the remaining A link caused buffer overflows within CDM.

It was found that network failures could result in up to 22 hours of downtime during the trial and still meet the availability objective. The network availability would be 99.0 percent.

Based on the hardware, software, and network availability estimates, the total CDM downtime during the trial was expected to be 86.47 hours, or 96-percent availability.

Reliability of CDM Service. Based on the reliability constraint, the system was to capture accurate data on 99 percent of all 800 service calls processed by the NCP. There were two potential sources of unreliability in the trial architecture:

- Message sampling board (MSB) buffer overflows. If the traffic volume at the NCP exceeded a threshold value, then the buffers on the MSBs could overflow, losing all the call messages in the buffer.

- PC buffer overflows. If the PC received more data from the MSB than it could transfer to the 3B2/400, then the PC buffer could also overflow. This condition could occur if there was a failure in the data collection software of the 3B2/400.

A queuing model was used to analyze the buffer sizes required to process all the CCS7 messages under varying traffic loads. If the sustained traffic was greater than 40 calls per second per link, there was a small probability ($p < 0.0005$) that a buffer overflow condition in the PC or the MSB would result. While the probability of this event was slight, a decision was made to load the NCP pair with only the trial customer's record. This would ensure a light load on the NCP and no buffer overflows.

Trial Evaluation

The success of the CDM was measured by whether it increased the efficiency of the customer's operation. Four different methodologies were used to gather the data on which to base the evaluation: an audit trail of customer's interactions with the system, a customer journal, task analyses, and customer interviews.

An audit trail of customer interactions was provided by the CDM system. Each request for a different display or report and the display or report parameters were recorded by the CDM automatically. This audit trail was used to determine the customer's preferred display types, preferred data parameters, and preferred flow between screens. Relative frequencies of customer errors and "help" requests when interacting with the CDM showed potential flaws in the user interface design. The audit trail was also used to correlate CDM activity with the customer's advanced 800 service call processing tree changes.

As a part of normal operations, the customer logs all 800 routing tree changes in a journal. An additional column was added to the customer's journal for the trial. The column allowed the customer to show the reason for making the change and the CDM displays/reports used to decide to make the change. The journal was used during the customer interviews to guide the discussion. The journal was also used to track when the customer made tree changes to correlate this activity with CDM activity.

A task analysis was performed two times during the trial. During the first and last month, the ASQIC team observed the use of the CDM in the customer's operations center.

Customer interviews were conducted in conjunction with the task analyses to discuss how and why the CDM was used during the month and how effective the user interfaces were in conveying the desired information.

Results

JC Penney found most of the CDM features to be quite valuable. The real-time monitoring and decision-making tool was used to make routing decisions. In particular, the U.S. map display was used to identify congestion problems and to determine where to off-load traffic. The bar display showed which NPAs generated the unexpected call volumes, so that the appropriate changes could be made to the call processing tree. The CDM enabled the customer to make intelligent routing decisions rather than "running blind." Customer service was improved because queue times were shortened. Additional revenues were stimulated by decreasing the numbers of lost and abandoned calls.

The summary displays tool was used to study traffic trends, allowing more accurate forecasting procedures and thereby increasing the efficiency of the operation and decreasing the number of idle attendants at any time.

The graphics displays were found to be an important component of the CDM. On the U.S. map display, the customer could see at a glance which location and band was overloaded. The summary graphics were used to spot trends quickly, and hard-copies of these summary displays were used in upper management reports. The customer used the graphics representation of the call processing tree to decide what changes to the tree were needed. However, the number of call attempts to each branch of the tree was not used or needed for this decision; a simple graphics representation of the tree was enough.

The CDM helped determine how "real" real-time data must be for the customer. To respond quickly to traffic congestion and line failures, the screens needed to be updated every five minutes or less. If updates were longer than this, the data was too old to be used in making routing decisions. For planning purposes, half-hour summary data would be sufficient. The half-hour interval of time was selected because the customer's automatic call distributors (ACDs) produced half-hour reports, and it was important that the two reports be easily compared. Regardless of whether the CDM capabilities become commercially available, the JC Penney managers felt that their participation in the ASQIC process was very worthwhile. Through involvement in the CDM feature definition and the CDM evaluation, they analyzed their operation from a new perspective, learned new things about their operation, and identified new ways to increase the efficiency of their operation.

Acknowledgments

The authors thank P. Somers, the ASQIC 800 CDM human factors designer, for her input and careful review of this article. We also thank C. Richards, J. Kylin, and J. Danner and the entire ASQIC team for contributing to the success of the trial.

Reference

1. J. J. Lawser and P. L. Oxley, "Common Channel Signaling Network Evolution," *AT&T Technical Journal*, Vol. 66, No. 3, May/June 1987, pp. 13-20.

Biographies (continued)

technical planning to provide a unified network management and control system for AT&T customer control services. She has a B.A. in psychology from the University of Wisconsin and a Ph.D. in mathematical psychology from Indiana University.

(Manuscript received January 15, 1987)