ABSTRACT

This paper reviews the development efforts used to create a Tracking, Scheduling, and Reporting System to assist in the management of in-house training at Virginia Power.

VIRGINIA POWER INFORMATION SYSTEMS TRAINING

Virginia Power Information Systems Training provides both mainframe, word processing and personal computer training to the corporation's 13,000 employees whose job responsibilities require the use of computer resources. A training facility, equipped to support mainframe, personal computer, and IBM 8100 training activities, is maintained near corporate headquarters in Richmond.

SCHEDULING PROJECT OVERVIEW

Information Systems Training began discussing the need to enhance its scheduling and tracking system during the fourth quarter of 1984. The system in use at that time, which consisted of a combination of manual and automated procedures, was not capable of addressing the set of scheduling goals given to the department by management. While there were several enhancements and maintenance changes that the department wanted to make to the system, the most pressing problem was the way the existing system handled scheduling. The system required that training requests be logged in at the time of receipt, but a request that required prerequisites would not appear in the 'to be scheduled' group until all prerequisites were met. The user community had registered concerns over this practice because it made it difficult for them to schedule training at times when they were sure they could attend. To address this need, Information Systems Training was instructed to schedule and notify all individuals requesting training of their placement within 30 days of receipt of a request.

The automated portion of the existing system did not support these changes and therefore, much of the system was not being used.

OVERVIEW OF THE OLD SYSTEM

The old system required that each training request be entered on the automated portion of the system and then be posted manually on a roll sheet for the class. When a course filled up, a waiting list was started on the bottom of the roll sheet. If someone cancelled, which happened frequently, a name was taken from the waiting list and the opening was filled. If a waiting list was considered full, names were transferred to the waiting lists for other classes. Occasionally, a name was entered on two or more waiting lists because of prerequisites. These entries usually had a variety of hand-written notes that detailed the reason the entry was on the list. When a person was scheduled for a class, a notification card was written by hand and mailed to confirm placement. Approximately two days before the class was to meet, a "reminder phone call" was made. All things considered, this system worked well, as long as the cancellation and rescheduling load remained light.

The automated portion of the old system used several different SAS/FSP(r) screens to handle data entry and queries. Moving from screen to screen required exiting from the current screen and restarting the TSO CLIST. This requirement placed the burden of remembering "what to do when" on the non-technical clerical users.

In the second quarter of 1985 a project team was assembled to create a new scheduling system that would address both the goals set by management and other new reporting requirements. Training elected to develop the system itself because it had the required skills in the department. A further motivation to keep the design effort within the
Training Department was the long development time required to have applications development design and implement the system. The need to frequently revise the input screens to accommodate changes in course offerings also figured in to the decision to develop the application in-house.

THE DEVELOPMENT EFFORT

The team began by reviewing the existing system's procedures and programs. Both were largely undocumented. Over time, the programs had been enhanced, maintained, and patched by several people, most of whom were no longer employed by the corporation, and very little documentation was done. Large blocks of code were found that had been by-passed with comments. After this review, a decision was made that the existing programs would not be used as a base for the new system. The development effort then turned to evaluating the contents of the system master files.

A study of the master file revealed that data entry had not been consistent, and there were large discrepancies in the codes used to identify specific courses. Many fields in the record were not being used, but data had been entered in some records and omitted from others. This situation had not created problems under the manual system, but it would be difficult to handle these anomalies in an automated system. A decision was made not to use the existing master file in its current state, but rather to convert as much of its data as possible to whatever format the new system might require.

When the development effort was started, the team prepared an initial set of objectives for the system. As the evaluation of the existing system proceeded, design objectives were added that were felt to be necessary. The final set of design objectives stated that the system must:

1. support time of receipt to time of class reporting to provide data on how long individuals had to wait for classes.

2. use a single screen format for all screen dependent transactions. (The old system required that several screens be used for some transactions.)

3. produce rosters.

4. produce instructors' copies of rosters for use as turn-around documents to verify student data and attendance.

5. produce notification and reminder letters.

6. include strategies to minimize data entry errors and assist in error detection.

7. permit changes to be made readily and have the changes reflected in the reports.

8. be menu driven where possible to simplify access for the users.

The review of the existing scheduling system and file structure indicated that major changes were required to meet the objectives established for the new system. Because a complete re-design was to be done, it was decided to evaluate alternate methods of implementation rather than to continue working in SAS(r).

Over a 5 week period other programming techniques were tested. Each evaluation consisted of prototyping a small section of the system in the method under consideration. Each method was evaluated for response time, disk usage, perceived difficulty of development and maintenance. Because this system may have to be maintained by users other than the development team, ease of future maintenance was an important feature. As each method was tested, portions of the existing master file were converted to a format consistent with the processing method. As the evaluations proceeded, it was found that the existing data structure would not readily support some of the features that were desired in the new system. Concurrent with evaluation of other programming methods, alternative data structures were produced and tested.

As a result of the prototyping efforts, it was determined that SAS/FSP(r) would be the easiest and most cost-effective method to use for development of the new system. Also, as a result of the prototyping experiences, a new
data structure was defined that would help to:

- simplify data entry,
- minimize data entry errors,
- limit the way course requests were defined to the system.

After the development product was selected, the development effort continued to prototype program modules to accomplish each of the identified tasks the system was to perform. SAS/FSP(r) was found to work well in the prototyping environment under which this application was developed. Prototyping allowed the system to undergo many of the initial activities as it was developed. When a set of screens and accompanying programs had been designed, it was turned over to the users for evaluation. The user feedback identified oversights and shortcomings. This helped avoid many design problems, and cut development and initial system modification time considerably.

OVERVIEW OF SYSTEM OPERATION

The scheduling system was designed as a menu-driven application, and it was interfaced with ISPF/PDF menus on a departmental level. Panel libraries and supporting data sets were created and loaded for all individuals involved with the system.

Requests for training are identified on the data entry menus by entering the dates associated with the request. Initially, the date the request was received by Information Systems Training is entered in the column headed Date Rec'd. When an opening in a class is available, an entry is made in the Date Sched. column. Openings are determined manually by consulting the Space Available report. This entry consists of the date on which the first day of class falls. The system uses the date fields as flags to indicate the status of training requests. A date entry in a specific Date Rec'd column indicates that the person has requested training in the subject listed to the left of that row. A corresponding entry in the Date Sched. column indicates that the request has been filled. These fields are the only method used to determine what course has been requested and the status of the request. Data entry errors result in erroneous rosters, incorrect space available counts, and incorrect notification letters. Errors of this type will appear first in internal departmental reports, which are used to assist in error detection and correction.

DATA STRUCTURE

As mentioned above, the old scheduling system used an array as its data structure. Each course was identified by a code entered into a text variable. These variables were named COURSE1, COURSE2, etc., and there was no direct connection between the variables and a specific course. Any course identifier could be entered in any variable. A corresponding set of variables, DATE1, DATE2, etc., contained the date on which the course was scheduled. Across time, several persons performed data entry, and because no input editing was performed, courses were coded differently. This data inconsistency did not impact the manual scheduling system because this data was not used. However, automated reporting was severely hampered, and manual counts were used in many cases to arrive at realistic figures.

For these reasons, and others, the new system discarded the old data structure and implemented one based on two parts:

The first part of each record consists of student data. It contains variables for uniquely identifying each student: name fields, social security number, charge-back system account number, telephone number, etc.

The second part of the record contains the variables for actual course registrations. This section consists of a set of five [5] variables for each of the 35 courses offered. The variables contain:

a. the date the request was received by Training; Due to a reporting requirement, it is necessary to know when the request was received.

b. the date that a requested course starts,

c. the time the course will begin,

d. whether or not a notification
letter has been generated, e.g. the status of the request.

The record layout follows:

<table>
<thead>
<tr>
<th>SSN</th>
<th>LNAME</th>
<th>FNAME</th>
<th>COSTCNTR</th>
<th>DEPT1</th>
<th>LOCATION</th>
<th>GROUP</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDP_STAT</td>
<td>IDP_DTRQ</td>
<td>IDP_DTSC</td>
<td>IDP_NOTE</td>
<td>IDP_TIME</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Identifying Data

Course Data

Note: Course Data is repeated once for each course in the schedule. Identifying Data is entered once for each student.

In addition to providing data on the status of a request, these variables are used to control the processing that generates reports. A discussion of each processing module follows.

OVERVIEW OF PROCESSING MODULES

The scheduling system was designed as a set of program modules that permitted each unit to be tested and placed in production independent of the others. After a module was completed and tested, it was placed in use and run in parallel with its manual system counterpart. The prototyping approach used during the development effort supported the modular installation very well. Modules were developed to handle student originated cancellations, notification letter generation, waiting list generation, roster production, and space available reports generation.

Each module is discussed below:

CANCELLATIONS

Cancellation processing was one of the major problems with the manual system. To address this problem, a change was made to departmental policy to require those who cancel to resubmit a Request for Training. This change simplified cancellation processing.

The system processes student originated cancellations by resetting the five variables associated with the course. The complete record was not deleted because many individuals will take several courses over time, and the student identification data remains on the system. Student originated cancellations are defined to the system by entering a cancellation code in the course status field. The daily backup procedures include routines that process the cancellations. A report is generated listing the name of the person and the class that was cancelled.

Cancellations originated by the training department are processed by entering a cancellation code in the course status field for each person registered for the class. End-of-day processing for these records resets the necessary control variables but leaves the Date Rec'd field intact. These entries are reported separately from the student originated cancellations. These records will appear on the waiting lists in Date Rec'd order; student originated cancellations do not appear on the waiting lists.

NOTIFICATION LETTERS

A module was developed to generate two notification letters for each person who has been scheduled for a class. Both letters are produced on standard white paper. One letter is the Notification of Scheduled Training; the other is a reminder. Both letters contain the person's name, company mailing address, the date their request was received, and name, date, and time for the course. The reminder letter is dated three days before the course is to meet, and it is filed in a tickler file by this date. Under the old system, a hand-written notification form was completed and mailed to each person. In addition, each person was called the day before the class. Under the new system, only the two letters are used for notification and reminder. No data is yet available to indicate if the current system's no-show rate is better or worse than before. The new procedure does, however, require significantly less time to administer.

UNSCHEDULED REQUESTS

Under the old system, the waiting lists were kept manually. The
new system uses the Date Rec'd field to determine if a person has requested a course and has not been scheduled. If there is no opening for the course requested, an entry will be made in the Date Rec'd field, but the Date Sched. field will be left blank. The unscheduled request program looks at the relationship between these two fields to determine the status of the request. If both are filled in, the request has been scheduled; if only the Date Rec'd field is completed, the request is reported on the waiting list. To help meet the requirement that all persons who request training be notified within 30 days, the value of TODAY() is compared to the Date Rec'd value and if the request is more than 20 days old, the record is flagged. Waiting list entries are printed in chronological order.

ROSTERS
The roster program provides a roster for each course in which at least one person is registered, and the program is run as required. The program determines enrollment by checking the Date Sched. field. The system does not perform course assignments. This was not implemented due to the changes that occur in the rosters. The roster program does not process any requests for courses that start on dates less than the value of TODAY().

SPACE AVAILABLE
This program creates reports for each course in which at least one person has been scheduled. It reports the number of persons enrolled. It also flags any course that is over-booked.

PROCESSING RESPONSIBILITIES
What: Entering Training Requests
Who: Department Administrative Assistant (Scheduler)
Procedure:
Completed Requests for Training are processed as follows:

1. The request is checked to insure that it has been properly completed. If the request is determined to be complete, it will be date stamped to indicate the date it was received in Information Systems Training.

Incomplete or incorrect requests will not be date stamped. These are returned for correction.

2. Checked requests will be entered into the Training Request Processing System in the following manner:

The Request file will be queried to determine if an entry exists for the individual named on the request.

If an entry in the master file does not exist, an entry will be made and the required personnel information entered.

If an entry exists, the date the request was received will be entered in the appropriate row under the Date Rec'd heading. If it is known when the request will be scheduled, the date of the first class meeting will be entered under the Date Sched. heading. No other entries need to be made.

If it is not possible to schedule the request when received, it is entered under the appropriate course(s) by making an entry in the Date Rec'd column, leaving the Date Sched. field blank.

What: Roster Generation Procedure:
The roster generation program will produce two sets of class rosters - one for the scheduler, and one for the instructor for each class in which one or more students are registered.

Roster generation ability is limited to the administrative assistant and a designated backup person to prevent multiple copies from proliferating in the department.

What: Class Enrollment Counts Procedure:
This report generation program will produce one set of reports for use by the scheduler in assigning personnel to classes. The report will include each class in which there is one or more students.

What: Unscheduled Requests Report Procedure:
This report will list by class the names of persons who have requested training, but who have
not been placed in a class as of the date that the report is run.

What: Entering Completion Data
Procedure:
Class completion data is entered by instructors. Instructors then find each person named on the roster and change his/her course status from P, for pending, to C, for complete. If a person does not complete a class, an I, for incomplete, is entered.

During this process, the instructor enters, or corrects, any data that the participant has indicated is in error. The information on the instructors' copies of the rosters is verified by the students when the class begins.

What: Student Cancellations
Procedure:
Upon receipt of a request to withdraw, the administrative assistant posts the request on the master file by entering an X in the Status column for the course.

The cancellation program is an integral part of the daily backup and housekeeping routines run when the master file is moved to a backup storage device. This program updates the master file and resets the control variables to insure that the individual's record will be processed properly. All fields concerning the cancelled class will be updated, and subsequent rosters will reflect the change.

Students who cancel must resubmit a Request for Training form to be rescheduled.

What: Training Department Originated Cancellations
Procedure:
When it has been determined by the training department that a class must be rescheduled, the administrative assistant indicates this on the master file by entering an R in the Status column for the course.

Cancellation processing is an integral part of the daily backup and housekeeping routines run when the master file is moved to backup storage. This program updates the master file and resets the control variables to insure that the individual's record will be processed properly. All fields concerning the cancelled class will be adjusted, and subsequent rosters will reflect the change.

Contact Author:
J. M. Plunkett
Virginia Power
OJRP-04
P.O. Box 26666
Richmond, VA 23261
Telephone: 804/771-3543