The Financial Management Environment (FaME) – A Prototype Interactive Hypertext-based Financial Planning and Reporting System

Kimberly D. Barnes Deborah M. Flanagan Nancy W. Grady
Judy A. Green Forrest M. Hoffman James A. Kohl
Michael R. Leuze Philip M. Papadopoulos Richard F. Sincovec *†

Abstract

It is critical in every government, research, and industrial organization that accurate and timely financial information be made available at all levels so that project and business decisions can be made within funding constraints. In the current climate of budget austerity, front line researchers and project leaders are being given more responsibility for managing their own finances. As a result, financial officers are now being called on to help more people understand the details of an organization's finances and therefore have less time to accomplish their primary tasks of budget planning and management. Large organizations are often characterized by legacy financial systems with out-of-date user interfaces, limited functionality, and late or difficult to comprehend reports. These systems are ill-suited to the task of easily extracting and presenting the necessary information to front-line researchers, project leaders and others with financial responsibility. Furthermore, these systems are usually too costly to replace.

The FaME prototype implemented at Oak Ridge National Laboratory extracts information from a legacy system, builds easy-to-understand reports and graphs, and presents them on-line so that people at all levels in an organization can assess the financial status of individual projects or entire organizations. Reports are presented in hypertext and graphical formats that can be read with popular worldwide web browsers such as NCSA Mosaic or Netscape. All reports are hyper-linked in a natural way to simplify navigation and information retrieval. To protect potentially sensitive information, FaME provides access control so that individuals can retrieve only the information that is required for them to carry out their financial duties. In addition to the reports and graphs, FaME includes budget building tools to provide for future financial planning. The overall goal of the FaME prototype is to give users the information they need with the equipment that already exists on their desktops.

*All, with exceptions noted below, are with the Computer Science and Mathematics Division, Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831. F.M. Hoffman is with the Environmental Sciences Division. K.D. Barnes was formerly with the Financial and Business Management Division.

†The FaME team can be reached at fame@msr.epm.ornl.gov or with URL http://www.epm.ornl.gov/fame

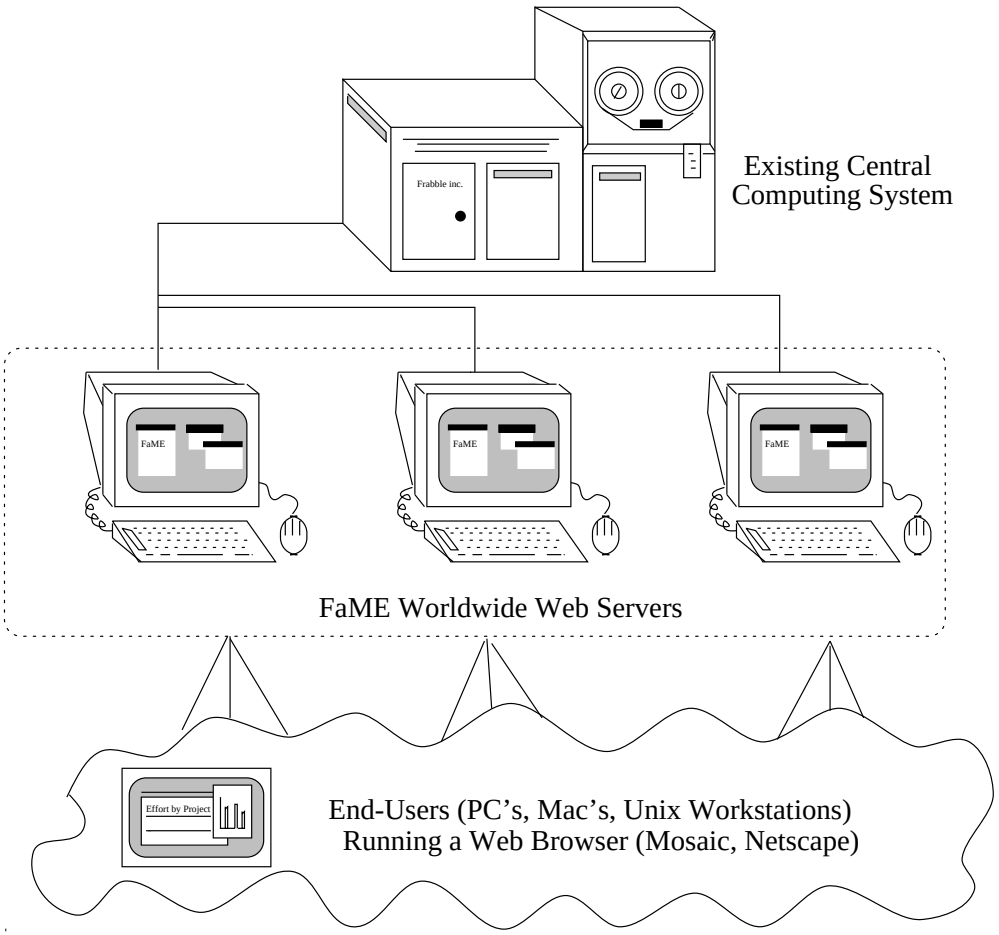

Figure 1: Logical Overview of Network Computing Environment

1 Introduction

The objective of the FaME is to show the financial status of an organization or project at a glance with the innovative use of the latest information technology. This prototype demonstrates the possibility of extracting information from a legacy system and presenting useful cross-sections of the data in both hyper-text and graphical formats. This report presents some of the details that allow the presentation financial data in the context of the National Information Infrastructure and the worldwide web. Figure 1 shows the logical layout of information transfer from an existing legacy system to several FaME servers and finally to end-users running a Web browser.

The overall implementation strategy of FaME is relatively straightforward. Figure 2 outlines the steps that are involved to bring the final report to the user's desktop. Data subsets from a legacy system are downloaded to a more modern relational database that runs on a server workstation. Custom "paper" reports using a structured query language (SQL) are then created. The reports are processed by a set of programs to convert the paper reports into an online browsable format that can be viewed from a personal workstation

using NCSA Mosaic or Netscape. This conversion creates hyper-text links between the paper reports so that more detail or related material can be retrieved with a single mouse click. At the time of the conversion, graphs are created to aid the user in understanding the tabular data. From the end-user's point of view, this process is completely automatic. Once a month (or whenever official cost accounting is available), the online reports are updated and the user only sees the custom reports. FaME should not be regarded as a replacement for a complete accounting system but rather as a value-added feature to an existing system that adds better reports, variance analysis and budget building tools.

Although, automation is used as much as possible to facilitate the transfer from the legacy system, there must be an administrator to manage access control, data selection criteria, and the monthly update of the online reports. FaME provides interactive interfaces to modify the selection, mapping, and downloading of data and interface for budget building. A graphical interface to manage access restrictions is currently under development. data needed for the divisions, sections and projects under a a directorate are downloaded to the workstation.

FaME implements access security to the server (a valid account and password is needed view information). Since, financial information for any organization is considered to be "business-sensitive" and given the ease with which financial data can be accessed when using FaME , tight controls are kept on who may read certain reports. For example, section heads may only view the financial information that is germane to their section and projects under that section. ORNL, like other large organizations, have program managers who need data that crosses these usual boundaries. FaME can handle the access needs for this type of crosscut without compromising the security of others in the organization.

This report details the methods used in selecting data, transferring data from the central system, creating paper reports and the final conversion to online hypertext and graphical formats. Several items refer to the specific prototype implementation at Oak Ridge. The conclusions section outlines areas in the prototype that need to be generalized for transferring FaME to other agencies.

2 Selecting Information for Downloading and Reporting

The FaME servers are workstations with large disks (9 Gigabytes) and expanded memory (128 Megabytes). Each server processes information for a logical unit of the overall organization. For example, two servers are currently in place, one serves the Physical Sciences and Advanced Materials Directorate (PSAM), and the other serves the Computer, Networking, Informatics, and Education Directorate (CNIE). An individual FaME server downloads only the information that is needed to process the reports for the logical business unit that it serves. To accomplish this goal, several selection tables are maintained by the unit's finance officer. These selection tables are termed control tables. The control tables provide enough flexibility to so accounts that are not associated within the natural hierarchy of an organization can be accessed. At ORNL, for example, LDRD (Laboratory Directed Research and Development) funds, are always associated with the Central Management Offices, rather than with the divisions to which the funds have been allocated.

The control tables are used so the at reports that are constructed on each FaME server can be made generic with respect to each organization. The control tables simply fill in

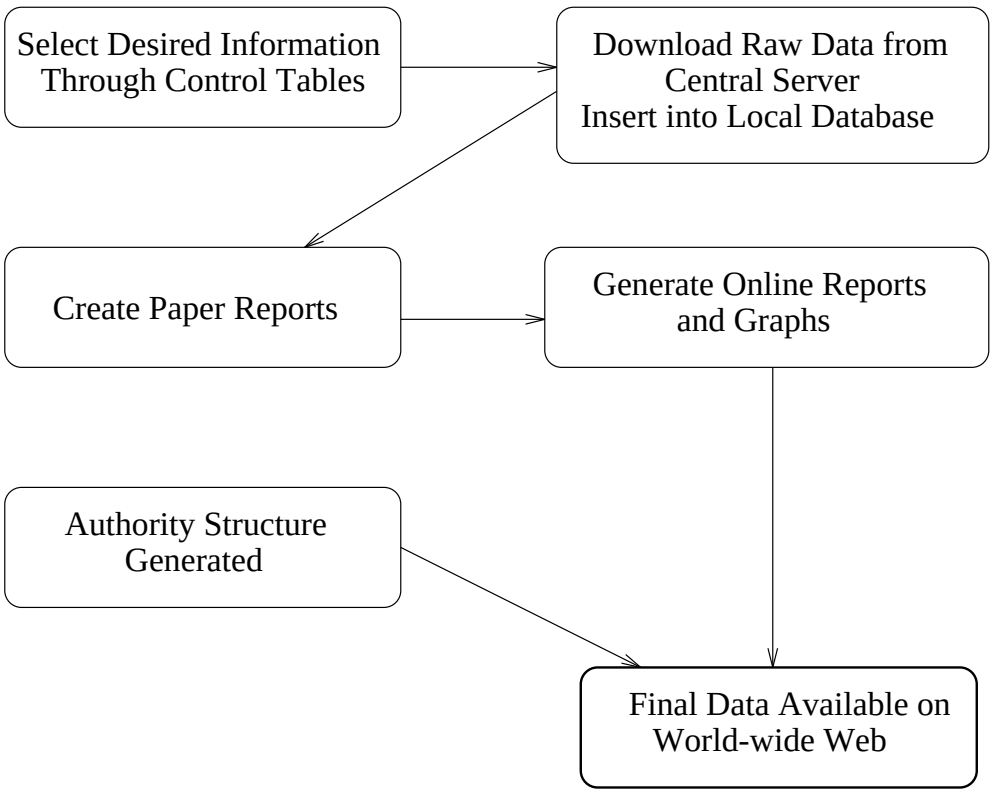

Figure 2: Flow Chart for Creating Online Reports

the specifics of which directorate, division, section, and projects are to be processed on a particular server.

The prototype server maintains four control tables with an interface for a unit's finance officer to interactively modify these tables. The next several subsections define the control tables that are used in FaME and shows some illustrative examples.

2.1 The SELECT Control Table

The SELECT control table specifies which information in the CAS (Cost Accounting System) is of interest to the directorate. Although the data is from CAS, all information is retrieved from the Corporate Information Center (CIC) and uses their copies of the CAS tables. It is used only by the automatic data transfer component of FaME. An example of a SELECT control table for the Computing, Networking, Informatics, and Education (CNIE) directorate is shown in Figure 3. Each line specifies a set of records to be fetched from

```
AC        ACDE  MAC  OUDI  XACTV      XADS  XPROG
--        ----  ---  ----  -----      ----  -----
                     X005
          05-%       X020
                     X010
          10-%       X020
                     X012
          12-%       X020
                     X055
          55-%       X020
                           KC07%
                           KT%
                           KV%
                           72STX3901
                           72STX3902
32000007
32030724
32030725
32090114
32120013
```

Figure 3: SELECT *control table for the CNIE directorate.*

the CAS databases. For example, the first line indicates that all records for which the OUDI (Operating Unit/Division) attribute has value X005 (Science Education and External Relations Division) are of interest to the CNIE directorate. Line two is used to pull records for which attribute ACDE (Account Description) begins with the characters 05- AND the OUDI attribute has value X020. These records pertain to LDRD funds and seed money that the Central Management Offices have allocated to the Science Education and External Relations Division. Other lines in the table indicate activities and accounts of interest to the CNIE directorate. Note that attribute values in different lines are logically OR-ed; attribute values in the same line are logically AND-ed. To be fetched from the CAS databases, a record must match the criteria of at least one line in the SELECT control table. Not all attributes of the SELECT control table are shown in Figure 3. To allow greater flexibility in specifying records for a directorate, "negative" attributes can take on values to indicate

which records are not of interest. For example, if a directorate is responsible for all projects in program KC except those projects in subprogram KC07, an entry in the SELECT control table with attribute XPROG set to KC and attribute NXACTVO set to KCO7 would specify the appropriate records. It was determined empirically that, in most cases, two sets of negative attributes will be sufficient to characterize a group of desired records. The names of these attributes are derived from the names of the positive attributes with N added as a prefix and 0 or 1 added as a suffix.

2.2 The DIVISION Control Table

The DIVISION control table associates the records of a directorate with divisions in that directorate. Its structure is identical to that of the SELECT control table, with the addition of a DIVISION attribute. The value of the DIVISION attribute is the division's OUDI value. As an example, the DIVISION control table for the CNIE directorate is illustrated in Figure 4. Note the use of negative attributes in the DIVISION control table. The Engineering Physics

DIVISION	AC	ACDE	MAC	OUDI	XACTV	XADS	XPROG	NACO	NOUDIO	NXADSO
X005				X005						
X005		05-%								
X005						ERKCL14				
X010				X010						
X010		10-%								
X012				X012						
X012		12-%						32090144		
X012						ERKCE%				ERKCE12
X012	32030724									
X020					72STX3901				X012	
X020					72STX3902					
X020	32030725									
X055				X055						ERKCL14
X055		55-%								
X055	32090114									

Figure 4: DIVISION *control table for the CNIE directorate.*

and Mathematics Division (X012), for example, is responsible for all FWPs (Field Work Proposals) with names beginning ERKCE except ERKCE12. The Center for Computational Science (X055) is not responsible for FWP ERKCL14, even though its OUDI value is X055. Not all negative attributes are shown in Figure 4; only those required by the CNIE directorate are included.

2.3 Other Control Tables

The COST CENTER and PROGRAM control tables are similar to the DIVISION control table, with the DIVISION attribute being replaced by the CC or the PROGRAM attribute, respectively. The values of the CC attribute are the eight-digit cost center account numbers. For example, 34100010 is used to specify the Mathematical Sciences Section cost center. The values of the PROGRAM attribute are an eight-digit account number followed by a two-character

program descriptor. For example, the KC07 "program" administered by the Center for Computational Sciences is designated by `34500001KC`. The Advanced Neutron Source program in the Physical Sciences and Advanced Materials (PSAM) directorate is designated by `32120007KE`.

2.4 Control Table Manipulation Interfaces

FaME supports a uniform user interface to all the control tables in the database. This interface allows users to create and modify the control tables, and their corresponding system functionality, in a consistent and efficient manner. The user can select any of the four control tables and then edit the rules that make up that table using a common "rule editor." Each rule represents a high-level description of some database selection criteria, and translates to a single line in the control table. All of the rules are logically OR-ed within each table to generate a complete selection statement to be passed to the database.

Rules are directly specified by the user as combinations of specific attributes and their desired values, rather than by filling in a complicated control table structure. Each attribute can be assigned to be "equal" to the given value, or "not equal" as needed for the negative attribute fields in certain control tables. All user input for the control tables is done via buttons and menus, requiring only mouse clicks, aside from some occasional typing to actually enter an attribute value. The user can either edit existing rules or create new rules from scratch to complete the desired control table. The user can then save the control table back to the database with the click of a single button.

3 Downloading Selected Data with Automatic Transfer

Wherever possible, the FaME downloads data from the central databases on system 61 (ksv2.priv.ornl.gov). System 61 runs the VMS operating system and is equipped with Multinet to allow Internet communication with other systems. The FaME servers are all Sun Microsystems Sparc stations running SunOS, a version of Unix. A variety of programs were written to bridge the gap between these two operating systems.

At the highest level, a finance manager determines the criteria that will select the appropriate subsets of data to be downloaded from system 61. This selection is made through an interactive interface to the selection control table (Figure 3). The user interface allows for only a small number of attributes to be used as selection criteria. The underlying software takes care of mapping these fields to table-specific attribute names. For example, most of the system 61 tables support the attribute OUDI (operating unit division). However, some tables name this field DI (e.g., the commac table). Our software allows only the attribute OUDI and will map OUDI to DI for the comacc table. The mapping relieves a FaME administrator from worrying about the specific names of attributes in the wide variety of central system tables. Once the selection criteria has been determined the following steps are taken to get the data from system 61 into the local Informix database:

- The selection criteria is output in a control table format
- If necessary, database fields are mapped to table-specific fields

- A DEC Command Language File (DCL) is generated from the mapped control table parameters. The DCL file does the following
  - Start ALPHA and select the data
  - Create a flat-file database that includes only the selected criteria
  - Remote copy the flat-file database to the FaME server
- The DCL file is submitted to system 61 using the utility *rsubmit*.
- An SQL and a database loading command file is automatically generated to load the flat-file database into Informix on the local FaME server.

Each of these major items will be discussed in some detail.

3.1 Selection Criteria Control Table Format

To keep the control selection interface separate from programs that use the selection criteria, an Informix table is used to keep the current selection criteria. An ACE-formatted report (DUMPctl) is run to output the current criteria. This report is never explicitly seen by the user, but is kept as an intermediate format. The first non-blank line of the report indicates the field names with an optional "NOT" prepending the field name to indicate a negative selection criteria. Fields are separated by a vertical bar. The following lines specify the selection. Fields are anded across a single line and multiple lines have indicate a logical or. An example is the simplest way to interpret the control table output

```
ACDE       | OUDI |  NOT ACDE      | NOT OUDI |
05-%       |  %   |      %         |     %    |
10-%       |  %   |    10-01%      |     %    |
  %        | X010 |      %         |     %    |
```

The field names are OUDI and ACDE. The first line indicates that all accounts descriptions beginning `05-` should be included. The second line says that *all* accounts beginning with description `10-` but not those beginning `10-01` should be included. The last line states that all records with `OUDI` equal to `10` should be included.

3.2 Control Table Mapping

The control table reported by DUMPctl has a set collection of attributes that may be used in selection. Sometimes, these attributes do not match those on the central system because of an inconsistent naming scheme used on the central system. The program *mapctl* reads a database attribute file and the generic control table to write a new control table with the renamed headings. The mapping constructs are kept in a file and are specific to each table that is to be downloaded. For example, if the `OUDI` attribute is named `DI` in a specific table, then *mapctl* will take the following directive to perform this mapping.

```
map: OUDI -> DI
```

The control table from above would then look like,

```
ACDE       |  DI   |  NOT ACDE      | NOT DI    |
05-%       |  %    |      %         |     %     |
10-%       |  %    |    10-01%      |     %     |
  %        | X010  |      %         |     %     |
```

Unfortunately, all mappings are not quite this simple. *Mapctl* handles a variety of mapping specifications and has a special construct that allows the mapping of an attribute to nothing when an appropriate attribute does not exist in a certain table.

3.3 Creating The DCL File

At ORNL, the central database system has a character-based interface ALPHA that allows a user to select tables, search criteria, canned reports, and user-defined reports. ALPHA contains some rudimentary relational capabilities, but was designed well before the advent of structured query languages SQL. The interface runs on a single central machine and is an access bottleneck. The ALPHA database selection criteria is based upon the control table criteria as output by the program *mapctl*. ALPHA's ability to join databases is used so that the selection criteria from the original FaME control table can be simplified. Since, attributes are not consistent across all tables on the central system, a configuration file for each central table is maintained. This configuration file contains the name of the ALPHA database, control table mappings, which attributes should be saved when a data subset is created, and information on when a join should be performed. The following is a subset of the configuration file for the travel table:

```
localtable: travel
database: travel
preselect: JOIN AC ACMSTR
map: OUDI -> ACMSTR.OUDI
report: ACWO
report: BEGDT
report: CITY
report: NUMDA
report: NAME
report: PUR
report: STATE
report: TVLNO
```

The format of the file is very simple. Each line contains a keyword and the data associated with that keyword. The above configuration file indicates that the name of the local (Informix) table is travel and the name of the ALPHA database is also travel. The keyword `preselect:` indicates one or more ALPHA commands that should be executed before the selection based upon the control table is performed. Here, we are joining to the database `ACMSTR`. There is the corresponding keyword `addselect:` that indicates ALPHA commands to be performed after the selection. The keyword `report:` lists the attributes that are to be included in the flat-file database. The map construct indicates that the we want to select records based upon the `OUDI` attribute in `ACMSTR` instead of the one in `TRAVEL`.

Based upon the information in the configuration file and the control table, a DCL file is generated that starts up ALPHA, selects the data, writes out a flat file database, and then transfers the database to the FaME server.

3.4 Remote Retrieval

The process of selecting data, writing out the table and copying it back to the local FaME server can be time consuming (several hours in some cases). It is important to make the process tolerant of failures and to be able to monitor failures. Unix and the Multinet extension to VMS support to methods for network logins. One is rlogin (remote login) and the other is rsh (remote shell). Since we have written a file of DEC commands to access ALPHA, we must be able to determine the status of the submission. The typical method in Unix is to redirect the standard input to be a file instead of a user's terminal. This does not work for rlogin because rlogin expects and interactive terminal device. Rsh does not work because the Multinet implementation does not read all the system variables properly to start ALPHA. We needed a program like rlogin but one that would take a file as input. Our program is named *rsubmit* and performs just like rlogin, but will take a file as input. It monitors the network connection and has a timeout mechanism if data does not return within a specified time. We have a wrapper around rsubmit that will provide multiple retries of the remote submission in case of network or remote node failure. This automatic retry feature is important due to the length of time ALPHA takes to process select statements.

When the remote submission is complete, a copy of the flat-file database is on the local Unix machine. All that remains is to load the database into the local Informix database.

3.5 Loading data into Informix

The flat-file database that is written out by ALPHA also includes a separate structure file that can be interpreted to build SQL and command files to automatically load the data into the Informix database. Once these files are built, execution results in the flat-file database being loaded into the Informix database

3.6 Interface

There is an interface that brings all the above pieces together. The interface allows the user to update a single local table or all of the tables. The user selects the table (or all tables) to be updated and clicks on one button. For user's familiar with ALPHA and DCL, the command file that has been generated by the software is displayed. If the user is satisfied with how his/her selection criteria has been interpreted, then a submission to system 61 requires one mouse click and a long wait.

4 Inserting Retrieved Data into the Local Database

Each structure file accompanying a flat file from the CAS databases is used to create two other files. The first is an SQL file used to create a table in the local Informix database. For example, the structure file created when fetching information for the `ACCOUNTS` table is illustrated in Figure 5, and the resulting SQL file is illustrated in Figure 6.

```
ATTR AC_DESCRIPTION               OR  ACDE     TEXT     32              -
    KEYED     LENGTH    32 MISSING PROHIBIT INITIALLY " "
ATTR AC_OPEN_DATE                 OR  ACODT    DATE                     -
    KEYED     LENGTH    10
ATTR AC_CLOSE_DATE                OR  ACCDT    DATE                     -
    KEYED     LENGTH    10
ATTR EFFORT_OPEN_DATE             OR  EFODT    DATE                     -
    KEYED     LENGTH    10
ATTR EFFORT_CLOSE_DATE            OR  EFCDT    DATE                     -
    KEYED     LENGTH    10
ATTR MANAGEMENT_CODE              OR  MAC      TEXT      3              -
    KEYED     LENGTH     3 MISSING PROHIBIT INITIALLY " "
ATTR X_ACTIVITY_NO                OR  XACTV    TEXT      9              -
    KEYED     LENGTH     9 MISSING PROHIBIT INITIALLY " "
ATTR X_ADS_NUMBER                 OR  XADS     TEXT     15              -
    KEYED     LENGTH    15 MISSING PROHIBIT INITIALLY " "
ATTR X_PROGRAM_NO                 OR  XPROG    TEXT      2              -
    KEYED     LENGTH     2 MISSING PROHIBIT INITIALLY " "
        END_DATASET;
```

Figure 5: *Structure file for* `ACCOUNTS` *table.*

```
CREATE TABLE accounts
  (
          AC  char(8),
        ACDE  char(32),
       ACODT  date,
       ACCDT  date,
       EFODT  date,
       EFCDT  date,
         MAC  char(3),
       XACTV  char(9),
        XADS  char(15),
       XPROG  char(2)
  );
```

Figure 6: *SQL file to create local* `ACCOUNTS` *table.*

A second file created from the structure file is the command file required by the Informix `dbload` facility to load the information directly from the flat file into the local database table. The command file that results from the structure file of Figure 5 is shown in Figure 7.

```
FILE "accounts.dmi" (
         AC  1 -  8,
       ACDE  9 -  40,
      ACODT  41 -  50,
      ACCDT  51 -  60,
      EFODT  61 -  70,
      EFCDT  71 -  80,
        MAC  81 -  83,
      XACTV  84 -  92,
       XADS  93 -  107,
      XPROG  108 -  109
);
INSERT INTO accounts ;
```

Figure 7: *Command file used by* `dbload` *to load information into* `ACCOUNTS` *table.*

The information fetched from the CAS databases is augmented with tables that contain information not available in the central system about the personnel and budgets of a directorate. This information is accumulated in flat files and then loaded into the local Informix database in a manner similar to the information fetched from the CAS databases.

5 Creating Paper Reports

FaME integrates data in the form of precise, meaningful reports and graphs that allow managers to understand the status of finances at a glance. Reports are programmed using the Informix ACE report specification language, and exist for different levels of line management from the Associate Director and Program Director overviews to the detailed project view. The reports are created in a "paper" format without cross references. A subsequent step converts the paper reports into on-line, web-browsable, hyperlinked pages with links for graphs.

Several reports have been defined and are described below.

DIRsub is a report summarized at the directorate level by subprogram. For each subprogram, the Budget Authority (BA) Carryover, New BA, Total BA, Budget Operating (BO), Current Month Cost, Fiscal Year Cost, Average Monthly Cost, BO Remaining, Percent Spent, Commitments (including overhead), and BA Remaining are reflected.

DIRsubdiv is a report summarized at the directorate level by subprogram and by divisions within subprograms. The same information as in the DIRsub.ace report is presented at a more detailed level to reflect division status within a subprogram.

DIRactiv is a report summarized at the directorate level by activity. For each activity in a directorate, regardless of division, the BA Carryover, New BA, Total BA, BO, Current Month Cost, Fiscal Year Cost, Average Monthly Cost, BO Remaining, Percent Spent, Commitments (including overhead), and BA Remaining are reflected.

DIVactivcc is a report summarized at the division level by activity and by cost centers participating in the activity. The same information as in the directorate activity report is presented in a more narrow scope to include only a specific division's information rather than information for an entire directorate. Within an activity, the division director can quickly view the status of funding for participating sections.

CCactiv is a report summarized at the section level by activity. The information is more narrow in scope than the division activity report, since the report reflects only a specific section's information rather than information for an entire division. For all activities within a section, the BA Carryover, New BA, Total BA, BO, Current Month Cost, Fiscal Year Cost, Average Monthly Cost, BO Remaining, Percent Spent, Commitments (including overhead), and BA Remaining are reflected at the detailed account level and summarized at the activity level. The section activity report is more detailed than the division activity report, since the section report reflects cost at the account level.

CCproj is a report summarized at the section level by projects (FWPs). For all projects within a section, regardless of activity code, the BA Carryover, New BA, Total BA, BO, Current Month Cost, Fiscal Year Cost, Average Monthly Cost, BO Remaining, Percent Spent, Commitments (including overhead), and BA Remaining are reflected at the detailed account level and summarized at the project level.

PROJsumary is a project report. For each project within a section, regardless of activity code, the BA Carryover, New BA, Total BA, BO, Current Month Cost, Fiscal Year Cost, Average Monthly Cost, BO Remaining, Percent Spent, Commitments (including overhead), and BA Remaining are reflected at the detailed account level and subtotaled for the project in the Budget Section of the report. In addition, effort, services, materials, travel, and commitment information is provided at the detailed transaction level in their respective sections.

Effort Section: The effort information includes the account charged, the account description, the name of the person charging the account, the source account from where the charges came, and the source account description, the current month hours charged, the current month cost, the fiscal year hours charged, and the fiscal year cost.

Services Section: The services information includes the account charged, the account description, the name of the person charging the account, the source account, the source account description, the service provided, the current month cost, and the fiscal year cost.

Materials Section: The materials information includes the account charged, the account description, the type expense code, the line identification (for the OCR report summary information), the purchase order number, the item on the purchase order, the description of the item, the current month cost, and the fiscal year cost. A description is always reflected in the description column if a description exists in the central system. The terms "accrual" and "reverse accrual" have been replaced with the item description. The materials information is sub-totaled by line identification for each account within the project.

Travel Section: The travel information includes the account charged, the account description, the name of the person charging the account, the travel number,the destination, the purpose of the travel, the date travel began, the number of days traveled, the current month cost, and the fiscal year cost.

Commitment Section: The commitment information includes the account charged, type expense, purchase order number, purchase order item number, delivery date, person to whom delivery is made, item description, original obligation, current obligation, estimated taxes, total commitment, estimated overhead, and total commitment with overhead.

CCadmin is a cost center, division, or directorate administration report that provides almost the same information as the project summary reports with the effort, services, materials, travel, and commitment sections. For the administration accounts, the Budget Section reflects current month cost, fiscal year cost, average monthly cost, and commitments.

EFFpers is a report compiled at the section level to reflect effort information for all persons within a section. The purpose of the report is to allow a section head to quickly view how the personnel for which he/she has responsibility are charging time. The effort information includes the person's name, project charged, account number/description charged, current month hours, current month cost, fiscal year hours, and fiscal year cost. The information is sub-totaled by person, and totaled by section. At a glance, for any given month, a section head can see the "income" generated from personnel charges.

EFFproj is a report compiled at the section level to reflect effort information for all projects within a section. The purpose of the report is to allow a section head to quickly view how staff both inside and outside the section are charging the projects for which he/she has responsibility. The effort information includes the project charged, account number/description charged, person charging the account, the source account from where the person charged, current month hours, current month cost, fiscal year hours, and fiscal year cost. The information is sub-totaled by project and totaled by section. At a glance, for any given month, a section head can see the effort charged to his/her projects.

Hours is a report compiled at the section level to reflect retained versus charged time for personnel within the section. The information includes the person's name, vacation, personal, sick, holiday, "other time burdened," total retained time, hours charged, grand total of all hours, and the efficiency rate. In addition, the individual information is totaled at the section level to allow the manager to view total vacation, personal, sick, holiday, "other time burdened," retained, and charged time for the scientific personnel.

Variance is a report that reflects the status of variance information for cost centers within a section. The variance information is in three sections entitled Variance Report, Historical Expenses and Income, and Projected Expenses and Income.

Variance Report Section: This information includes the current month, standard rate, actual rate, average actual rate, gross current variance, special rated account effects, variance distribution, net current variance, and net cumulative variance.

Historical Expenses and Income: Monthly as well as cumulative historical expenses and income are reflected.

Projected Expenses and Income: Information is divided into income and expenses. The expense information labeled "MINIMUM REQUIRED" indicates the minimum required expenses to cover payroll, fringe, and division administration costs to date as well as in the future. In addition, expected expenses derived from the standard rate budget and labeled "CURRENT PLAN" are reflected. The income information entitled "AT CURRENT CHARGE RATE" indicates the amount of income generated to date as well as expected income generation in the future at the current charge out rate. Maximum future income is reflected at the maximum charge out rate (100%) and is labeled "AT MAXIMUM CHARGE RATE." The charge out rate necessary to meet minimum costs or current planned costs is reflected. At a glance, a manager can determine if enough income has been and can be generated to meet minimum requirements, and if enough income has been and can be generated to meet planned expenditures.

Note that standard rate and personnel information must be entered by the financial personnel before the third section of the variance report can be generated.

PRproj is a report summarized at the program level by project regardless of activity. For each project, the Budget Authority (BA) Carryover, New BA, Total BA, Budget Operating (BO), Current Month Cost, Fiscal Year Cost, Average Monthly Cost, BO Remaining, Percent Spent, Commitments (including overhead) and BA Remaining are reflected by account and sub-totaled by project.

PRprojdiv is a report summarized at the program level by projects regardless of activity and by divisions participating on those projects. The report contains the same information as the program report by project, but the view of information is not at the account level. The information is summarized at the division level within a project.

PRdivproj is a report summarized at the division level within a program. Within a division, the information is summarized by project. The view allows the program director to enter a specific division and view all the projects within a division that are funded by his/her program at the detailed account level.

PRsubactiv is a report summarized at the program level to provide a program director with an overview of budget, cost, and commitment information sub-totaled by activity and subprogram to determine if a more detailed investigation is necessary.

PRsummary is a program summary report containing the same information as the project summary report containing the budget, effort, services, material, travel, and commitment information described earlier in this report. The information is at the program rather than the project level.

6 Conversion of Paper Reports for On-line Browsing

The final goal for the project is to insure the timely delivery of cost information to the managers who are in charge of the spending decisions. This clearly requires an on-line, multi-platform presentation solution for existing desktop systems. Mosaic and Netscape provide such an immediate solution for information presentation. Several steps are taken to convert the static financial information pages to interconnected on-line access, and to provide coherent access into the information.

A perl script is used to parse the ascii report files. Keying on cost center titles allows the division of the sequential ascii files into separate HTML (hypertext mark-up language) pages which are stored in directories specific to that cost center and month. Two passes are made through the ascii files. The first pass collects people and project titles within the directorate, and data for graphical representation (see below). In the second pass hyperlinks are inserted to jump to appropriate references to persons, projects and graphs . This cross checking insures that no links are created for inappropriate or non-existent connections.

The use of the mosaic interface allows feedback from the users, and connectivity to external sources of information. References to people can be linked to individual home pages. Likewise project references can be made to project pages. Icons are used to reference phone and address listings, and organizational charts. Two perl mail routines are used to send email suggestions to the FaME implementation team, or to report errors. To assist in identification the URL of the report being viewed is coded into the icon anchor to automatically identify the source document the comment is originating from.

6.1 Graphs

The `perl` script that converts ASCII reports to HTML serves an additional function. At each point that a graph is to appear in a report, it dumps information required to produce that graph into a file with a `.dat` suffix. It then links to a Postscript file with the same name, except for a `.ps` suffix.

The Postscript files are produced by a C program that uses the `.dat` files as input. A typical input file is illustrated in Figure 8. In Figure 8, `LBA` indicates that a layered

```
LBA
3
ENG. PHYSICS & MATHEMATICS
SUBPROGRAM 11A3
FOR BUSINESS MONTH ENDED 07/24/1994
247841 161099 408940 213887 24458
52.3%
```

Figure 8: *Input file for C program that produces Postscript graphs.*

bar graph is to be produced. There are three title lines. The five numerical values are BA carryover, new BA, BO, FY cost, and commitments. The final line in the input file indicates that FY cost amounts to 52.3% of the BO. The resulting graph is shown in Figure 9. For

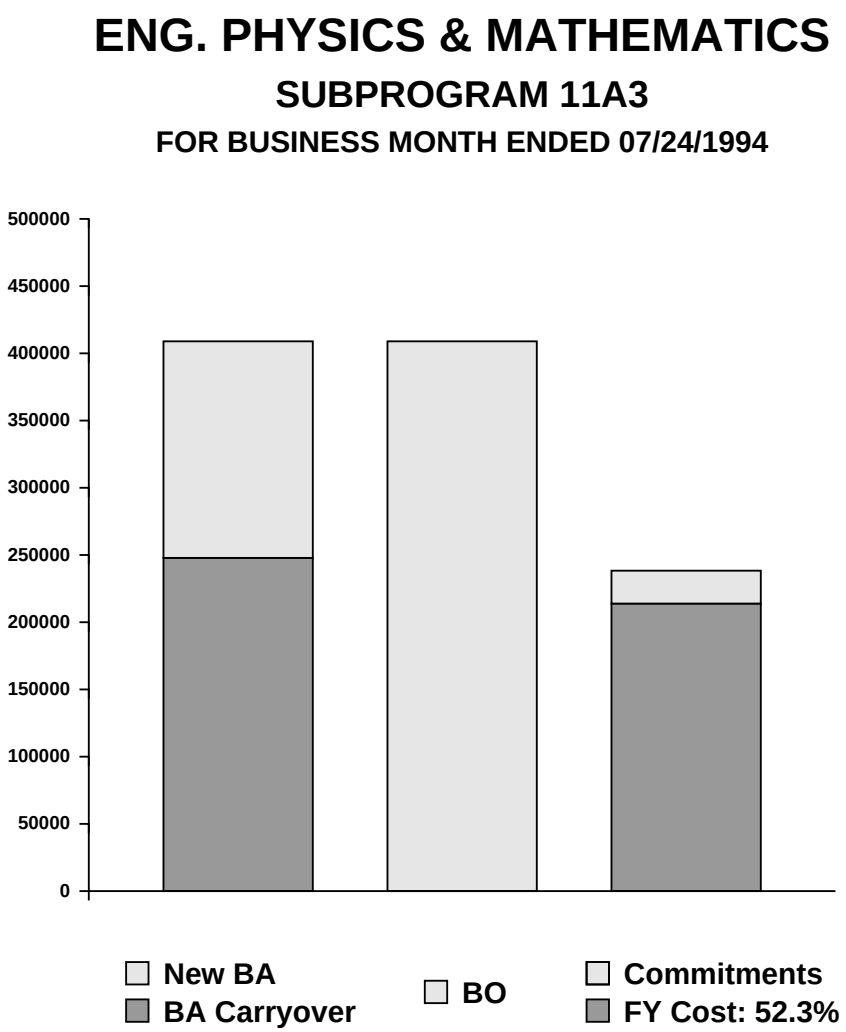

Figure 9: *Graph in Postscript file produced by C program.*

this report, the full color figure supported by the FaME system has been modified to a gray scale version.

The FaME system produces a large number of graphs in a variety of types, including pie charts, line graphs, and bar charts. Data from the highest level, funding levels for a directorate by program, for example, to the lowest level, the cost of an individual researcher's effort to specific accounts, for example, are illustrated graphically.

7 Authority Structure

Access to FaME is handled through user password authentication. A table is maintained to describe the organizational relationships and to determine access. This *orgstructure* table contains a line for each organizational unit and cost center. The first string determines the hierarchy and the rest the three character id of managers or finance officers who should have access.

```
/directorate/division  division_director finance_officer
/directorate/division/section  section_head
/directorate/division/section/costcenter  manager1  manager2
```

A perl script processes this file to automatically generate the `.htaccess` and `.htgroup` and `access.conf` files NCSA mosaic uses to define access. A separate program grabs the standard system password for the users. Using this mechanism allows the use of proper passwords, definition of the members of groups, and groups which can access cost center information.

The same `orgstructure` table is read to dynamically create a home page that shows only the levels in the organizational tree that the user is authorized to see. Selection of an entry in the organization tree calls a second perl script which reads the chosen directory structure and groups the files into categories based on their naming convention. This scheme provides quick, authorized access to any information in the full tree after no more than three mouse clicks, for efficient navigation.

8 Budget Manipulation Interface

In addition to the control table interface described in a previous section, there is also one special user interface for creating and manipulating budgets. A budget can be efficiently created with a few mouse clicks using scrolling menus. One menu allows the user to select any desired cost center for a budget item, and another menu allows selection of the desired line item ID. After entering a dollar amount for the line item, the interface automatically inserts the line item into the budget, removing any previous amount for the given line item ID, and then also provides the resulting overhead costs. The budget is then displayed back to the user. If the changes to the budget are correct then the budget can be saved back into the database.

9 Future Work

FaME is a useful tool for easily bringing financial information to those who need to track accounts. The overall system works well but should be regarded as experimental. FaME started from a small core of useful paper reports that had been developed over several years. Several different reports were added during the implementation. The system is fragile in the sense that each new report requires the program that converts to the online format to be modified. The online format was almost an afterthought, but has proved to be a timely and inexpensive method for distributing information. The online graphs add a much needed dimension for analyzing financial information. The overall system needs to be simplified with a single report specification that can generate both paper and online reports. A good specification would make it easy for users to download a particular page of data in a spreadsheet format for further, specialized queries.

Access control is essential, but the proliferation of networks means that data is no longer secure while in-transit from the FaME server to the web browser. Encryption of data is the proper route, but it will take several more months before secure servers and clients that can understand the encrypted data streams are prevalent throughout the laboratory.

For FaME to exist outside of the laboratory, experimental implementations at other organizations are essential to discover similarities and differences among financial structures. FaME works well at ORNL, but it is unclear how well it would work for Energy Systems or even the Department of Energy. Certainly the specifics of access to central data will differ in detail from ORNL's. But how does their financial organization differ? Can the existing reports be easily adapted to their structure?

10 Conclusions

This report has outlined the steps that are performed each month for on-line reports to reach end-users. The data initially resides on disparate tables on a central legacy system. Specific subsets of data selected and download onto FaME workstation servers. Reports using a commercial database system are created based upon selection criteria kept in several control tables. The paper reports then go through an online conversion with the addition of useful graphs. The final "product" may be viewed by authorized users from any Web browser running on PC's, Mac's or Workstations.

FaME is a useful tool for analyzing the existing financial status of an organization. The overall concepts and implementation are straightforward. Work needs to continue to determine if a generic set of programs that implements FaME capabilities can be developed for use at other Government and business sites.

References

[1] D. Dougherty, *sed & awk*, O'Reilly & Associates, Sebastopol, CA, 1992.

[2] Informix Software, Inc., *Informix Guide to SQL*, Dec. 1991, Menlo Park, CA.

[3] Peters, J.F., III and P. Holmay, *The VMS User's Guide*, Digital Press, 1990.

[4] Wall, L. and R.L. Schwartz, *Programming perl*, O'Reilly & Associates, Sebastopol, CA, 1991.