ORNL/TM-13139

Computer Science and Mathematics Division

Mathematical Sciences Section

THE FINANCIAL AUTOMATED MANAGEMENT ON-LINE USER SYSTEM (FaMOUS): A PROTOTYPE INTERACTIVE HYPERTEXT-BASED FINANCIAL PLANNING AND REPORTING SYSTEM

K.D. Barnes[†] J.M. Donato D.M. Flanagan N.W. Grady J.A. Green
F.M. Hoffman[‡] J.A. Kohl M.R. Leuze P.M. Papadopoulos R.F. Sincovec

Computer Science and Mathematics Division
Mathematical Sciences Section
Oak Ridge National Laboratory
P.O. Box 2008, Bldg. 6010
Oak Ridge, TN 37831-6414

[‡]F.M. Hoffman is with the Environmental Sciences Division.

[†]K.D. Barnes, formerly of the Financial and Business Management Division, is now with the Computer Science and Mathematics Division.

Date Published: November 1995

Funded by Operating Funds, Oak Ridge National Laboratory, U.S. Department of Energy.

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
managed by
Lockheed Martin Energy Systems, Inc.
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-84OR21400

Contents

1 Introduction 1
2 Overview of the FaMOUS System 4
3 Description of Current Reports 6
3.1 Basic Report Templates 6
3.2 Current Reports 7
3.3 Report Traversal 9
4 Authority Structure and Access Permissions 9
4.1 Organization Structure Table 9
4.2 Three Character User IDs and Passwords 10
4.3 Group Access Files 10
4.4 Access via the Graphical User Interface 11
5 Conversion of Reports for On-line Browsing 11
6 Selection of Information for Downloading 13
6.1 The `SELECT` Control Table 14
6.2 The `DIVISION` Control Table 15
6.3 Other Control Tables 16
6.4 Control Table Manipulation Interfaces 16
7 Downloading Selected Data with Automatic Transfer 16
7.1 Selection Criteria Control Table Format 17
7.2 Control Table Mapping 18
7.3 Creating the DCL File 18
7.4 Remote Retrieval 19
7.5 Inserting Retrieved Data into the Local Database 20
7.6 Interface 22
8 Budget Manipulation Interface 22
9 Future Work 22
10 Conclusions 23
11 Acknowledgments 23
12 References 24

Abbreviations and Acronyms

ASCII	American Standard Code for Information Interchange
BA	Budget Authority
BACOV	Budget Authority Carryover
BO	Budget Operating
CAS	Cost Accounting System
CC	Cost Center
CIC	Corporate Information Center
CRE	Computing, Robotics, and Education Division
FaMOUS	Financial Automated Management On-Line User System
FWP	Field Work Proposal
FY	Fiscal Year
HTML	HyperText Markup Language
ORNL	Oak Ridge National Laboratory
OTB	Other Time Burdened
OUDI	Operating Unit/Division Identifier
SGML	Standard Generalized Markup Language
SQL	Structured Query Language
URL	Uniform Resource Locator

List of Figures

1 *Logical Overview of Network Computing Environment* 2
2 *Overview of FaMOUS Information Flow* 3
3 *Flow Chart for Generating Online Reports* 5
4 *Example* `orgstructure` *Table* 9
5 *Example* `.htpasswd` *file* 10
6 *Example* `.htgroup` *file* 10
7 *Input file for C program that produces Postscript graphs.* 12
8 *Postscript Graph as produced by C program.* 13
9 `SELECT` *control table for the CRE directorate.* 14
10 `DIVISION` *control table for the CRE directorate.* 15
11 *Structure file for* `ACCOUNTS` *table.* 20
12 *SQL file to create local* `ACCOUNTS` *table.* 21
13 *Command file used by* `dbload` *to load information into* `ACCOUNTS` *table.* 21

THE FINANCIAL AUTOMATED MANAGEMENT ON-LINE USER SYSTEM (FaMOUS): A PROTOTYPE INTERACTIVE HYPERTEXT-BASED FINANCIAL PLANNING AND REPORTING SYSTEM

K.D. Barnes† J.M. Donato D.M. Flanagan N.W. Grady J.A. Green
F.M. Hoffman‡ J.A. Kohl M.R. Leuze P.M. Papadopoulos R.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. The FaMOUS[1] prototype implemented at Oak Ridge National Laboratory extracts financial data 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 World Wide 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, FaMOUS 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, FaMOUS includes budget building tools to provide for financial planning. Another primary feature is that the prototype utilizes equipment that already exists on the user's desktop. The overall goal of the FaMOUS system is to provide users with precise and meaningful information on the financial status of an organization or project at a glance.

[1] Except as noted below the authors are in the Computer Science and Mathematics Division, Mathematical Sciences Section, Oak Ridge National Laboratory, P.O. Box 2008, Bldg. 6010, Oak Ridge, TN 37831-6414.
The FaMOUS development team can be reached by electronic mail at fame@msr.epm.ornl.gov or on the World Wide Web via the URL http://www.epm.ornl.gov/fame.
‡ F.M. Hoffman is with the Environmental Sciences Division.
† K.D. Barnes, formerly of the Financial and Business Management Division, is now with the Computer Science and Mathematics Division.

1. Introduction

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 thus 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 objective of the Financial Automated Management On-Line User System (FaMOUS) is to provide users with precise and meaningful information on 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 of financial data in the context of the National Information Infrastructure and the World Wide Web. Figure 1 shows a logical layout of the network computing environment consisting of an existing legacy system connected to several FaMOUS servers that then connect to end-user workstations.

The FaMOUS prototype implemented at Oak Ridge National Laboratory (ORNL) extracts financial data 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 World Wide 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, FaMOUS 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, FaMOUS includes budget building tools to provide for financial planning. Another primary feature is that the prototype utilizes equipment that already exists on the user's desktop. Figure 2 depicts a FaMOUS system from the viewpoint of information flow.

The next section presents a brief overview of the FaMOUS system. Later chapters will provide background information necessary for selecting data, downloading data from the central system, creating reports, and converting the reports to online hypertext and graphical formats. Several items refer to the specific prototype implementation at ORNL. The section entitled "Future Work" outlines areas in the prototype that need to be generalized for porting FaMOUS to other agencies.

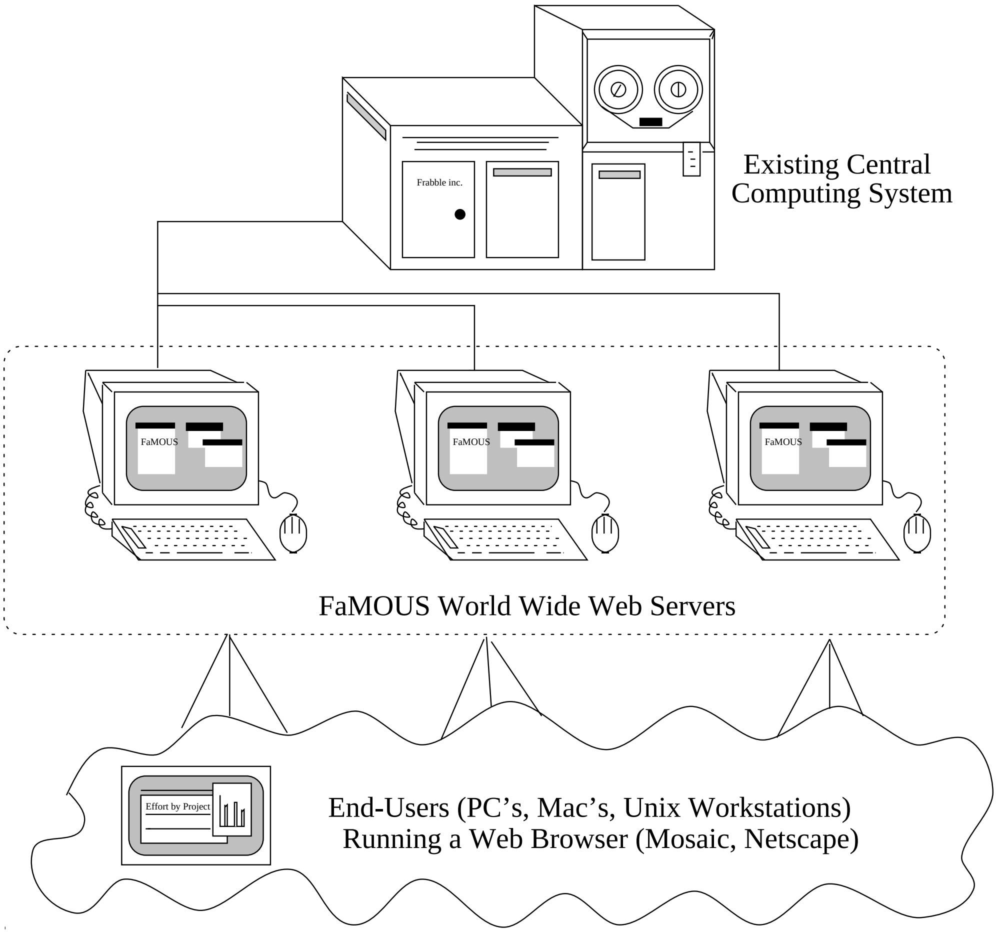

Figure 1: *Logical Overview of Network Computing Environment*

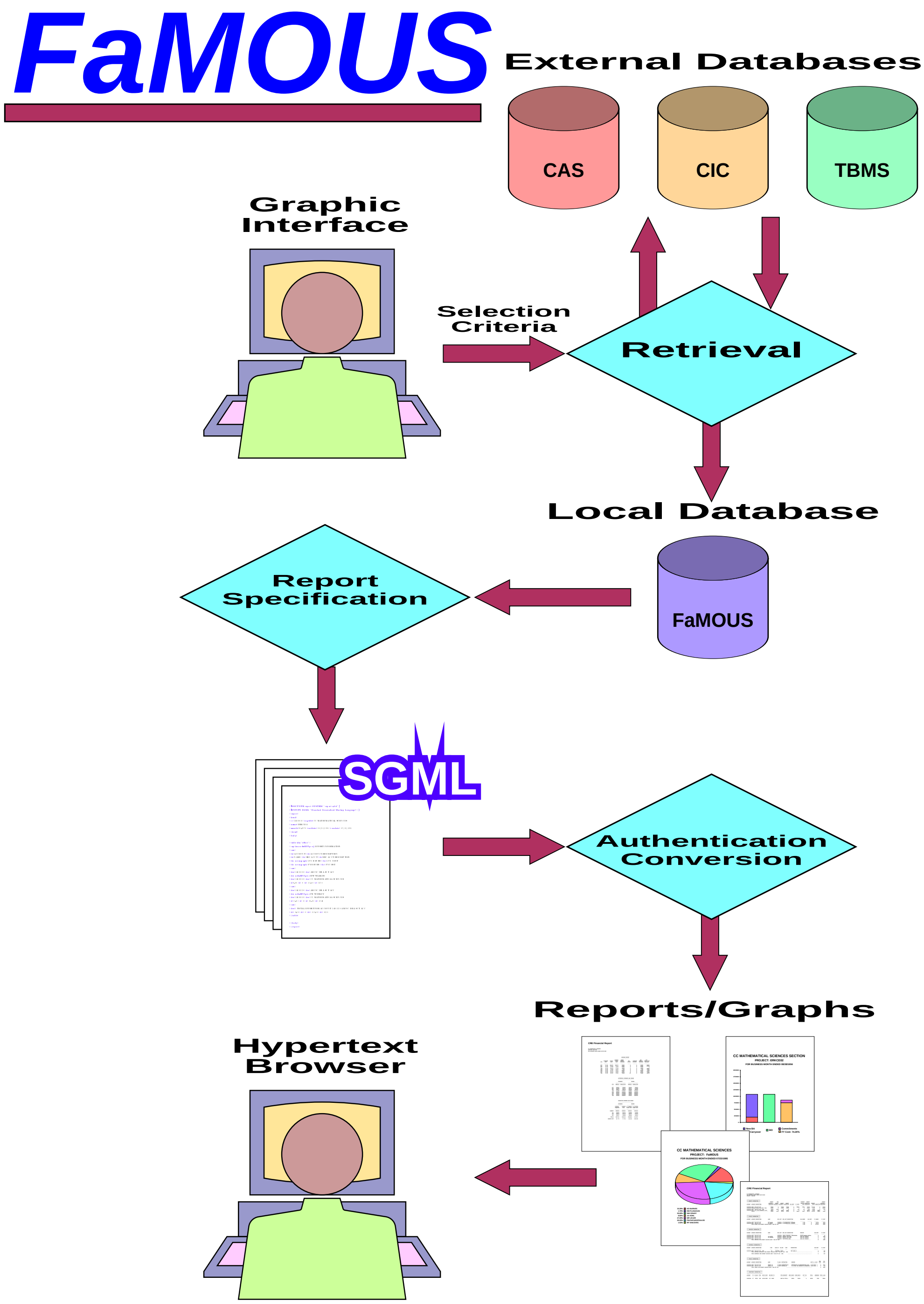

Figure 2: *Overview of FaMOUS Information Flow*

2. Overview of the FaMOUS System

The overall implementation strategy of FaMOUS is relatively straightforward. 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 sees only the appropriate custom reports. However, FaMOUS 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, information connectivity and navigation, variance analysis, and budget building tools.

Figure 3 depicts the the stages involved in generating reports accessible at the user's desktop.

- Selection tables are used to specify the data subsets to be downloaded from a legacy system to a more modern relational database. This local database runs on an FaMOUS workstation server.
- The specified reports are then created via SQL (Structured Query Language[2]) commands.
- The reports are processed by a set of programs to convert the reports into an online HTML format that can be viewed from a personal workstation using a browser (such as NCSA Mosaic or Netscape). This conversion creates hyper-text links between the 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.
- The reports and graphs may now be viewed via a World Wide Web browser at the user's workstation. An Authority Structure (a table specifying the organization structure and access permissions) is used to restrict user viewing information pertinent to that user position in the organization's hierarchy.

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. FaMOUS provides interactive interfaces to modify the selection, mapping, and downloading of data and for budget building. A graphical interface to manage access restrictions is currently under development. Data needed for the divisions, sections, and projects under a directorate are downloaded to the FaMOUS server workstation.

FaMOUS provides access security on the server (a valid account and password is needed to 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 FaMOUS, tight controls are kept on who may read certain reports. For example, section heads may view only the financial information that is germane to their section and projects under that section. ORNL, like other large organizations, has program managers who need data that crosses these line boundaries. FaMOUS can handle the access needs for this type of crosscut without compromising the security of others in the organization.

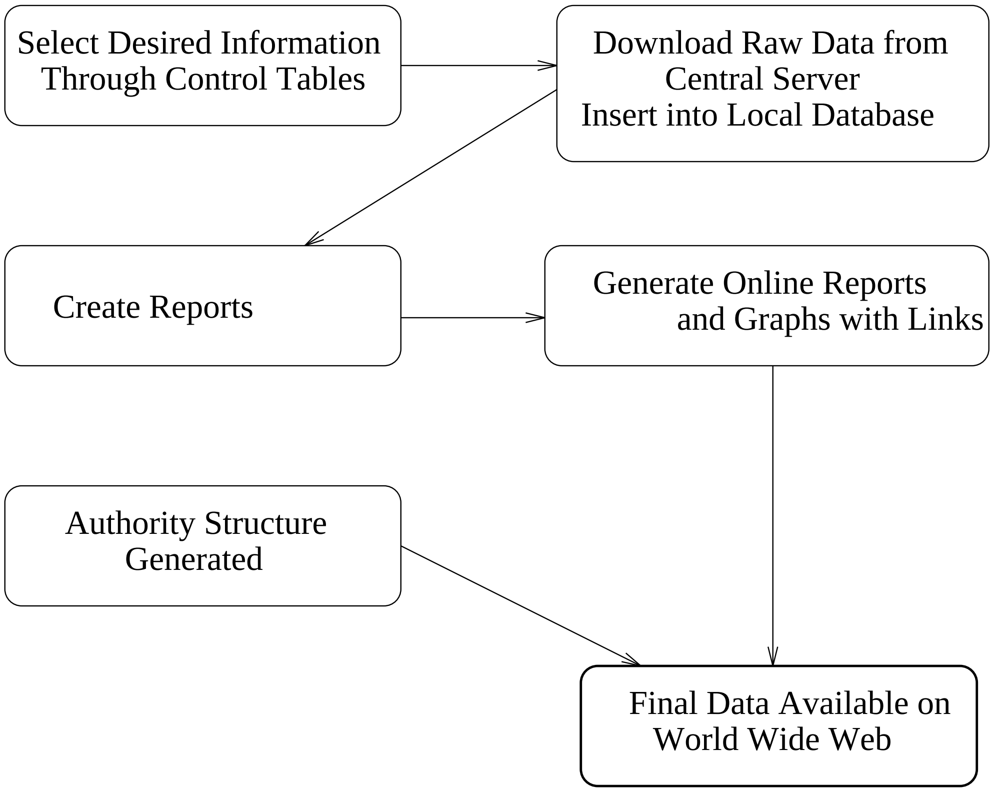

Figure 3: *Flow Chart for Generating Online Reports*

3. Description of Current Reports

FaMOUS integrates data in the form of precise meaningful reports and graphs allowing managers financial status information at a glance. Reports are generated using the Informix ACE report specification language. Different levels of reports correspond to different levels of management, from the Associate Director and Program Director level overviews, to more detailed Program/Project and Section level views. The reports are created in an ASCII computer format without cross references. A subsequent step converts these reports into on-line, Web-browsable, hyperlinked pages with links for graphs (pie or bar charts). Authorization for different levels of access is described in Section 4. The generation of hypertext links and graphs for online browsing are described in Section 5.

Before giving brief descriptions of some of the current reports, we describe the basic templates upon which most of the reports are based.

3.1. Basic Report Templates

The basic **"Budget"** template is comprised of the Budget Carryover, New Budget, Total Budget, Budget Operating, Current Month Cost, Fiscal Year Cost, Average Monthly Cost, Budget Operating Remaining, Percent Spent, Commitments, and Budget Authority Remaining.

The **Hours** template contains, for each person, the person's name, holiday, vacation, sick, and other retained hours, total hours retained, hours charged, and total hours.

The **Effort-by-Person** template contains, for each person, the person's name, the project code to which the person charged, the account within a project to which a person charged, account description, current month hours, fiscal year hours, current month cost, and fiscal year costs.

The **Effort-by-Project** template contains, for each person that charged to the project, the account number charged, the account description, the name of the person, the source account from where the person charged, current month hours, fiscal year hours, current month cost, and fiscal year costs.

The **Travel** template contains, for each travel entry, the account to which the travel was charged, the account description, the name of the person in charge of the travel, the travel number, destination, purpose, date and days of travel, current month cost, and fiscal year costs.

The **Materials** template contains the account to which the material was charged, account description, type expense code, line identifier, purchase order number, purchase item number, item description, current month cost, and fiscal year costs.

The **Services** template contains the account number to which services were charged, account description, the name of the person providing the service, source account from where the person charged, source account description, the service provided, current month cost, and fiscal year costs.

The **Commitment** template contains the account number, the type expense code, purchase order number, item number, delivery date, the person to whom (or the place to which) to deliver the item, item description, original obligation, current obligation, estimated tax, total cost before overhead, overhead amount, and total cost with overhead.

3.2. Current Reports

We now describe a number of the current reports below. The report title is given in **boldface** followed in parentheses by the name of the program file within the FaMOUS system. The template upon which the report is based is given in *italics* within the description.

There are no explicit section codes. Hence, to delineate projects by section, the major Cost Center account code within each section is used as that section's identifier. In addition, the terms "Project" and Field Work Proposal ("FWP") are used interchangeably.

The reports provide managers with an overall summary and access to the desired level of detail.

Directorate Cost by Subprogram (DIRsub)

Provides a Directorate level summary of *Budget* information for the directorate sub-divided by program and then by sub-program with a program.

Directorate Cost by Subprogram/Division (DIRsubdiv)

Provides a Directorate level *Budget* summary for the directorate sub-divided by sub-program and then by division.

Directorate Cost by Activity (DIRactiv)

Provides a Directorate level *Budget* summary for each activity within the directorate.

Program Cost by Project (PRproj)

Provides a Directorate level *Budget* summary for each program managed by the Directorate sub-totaled by projects within that program. It is further sub-totaled by the accounts within the projects funded by the program.

Program Cost by Project and Division (PRprojdiv)

Provides a Directorate level *Budget* summary for each program by sub-program and by each project with accounts funded by the sub-program. It is then further sub-divided by the divisions that manage accounts within the project.

Program Cost by Activity (PRsubactiv)

Provides a Directorate level *Budget* summary for each program managed by the Directorate sub-divided by sub-programs and activities.

Program Cost by Division and Project (PRdivproj)

Provides a Directorate level *Budget* summary for each program. It is sub-totaled by each division within the program and further sub-divided by projects and accounts.

Division Cost by Activity/Section (DIVactivcc)

Provides a Division level *Budget* summary organized by sub-programs and by activities within the sub-programs. It also reflects the Sections within a division that participate in the activity.

Cost Center Administration (CCadmin)

Provides a *Budget* summary of the administration accounts at the appropriate level (Directorate, Division, Section, or Program). This reports also utilizes the *Effort-by-Person, Effort-by-Project, Services, Materials, Travel,* and *Commitments* templates. This report uses similar templates as the **Project Summary** report. The distinction is that this report is for administration accounts, whereas the **Project Summary** report is for research and development final cost objective accounts.

Cost by Activity (CCactiv)

Provides a Section level *Budget* summary sub-divided by program, then by activity, project, and account.

Cost by Project (CCproj)

Provides a Section level summary by program sub-divided by project and then by accounts within that project.

Effort Report by Person (EFFpers)

Provides a Section level *Effort-by-Person* summary for all members of the Section.

Effort Report by Project (EFFproj)

Provides a Section level *Effort-by-Project* summary for each project. This information is sub-divided by each account within a project and then by each person that charges to that account.

Retained vs. Charged Hours (Hours)

Provides a Section level *Hours* summary for each member of the Section.

Project Summary (PROJsumary)

Provides a Project level *Budget* summary. This report also utilizes the *Effort-by-Person, Effort-by-Project, Services, Materials,* and *Travel* templates. This report is the at the finest level of detail for capturing information at the transaction level.

Variance Analysis (Variance)

Provides a *Variance* summary for each administration account within the research and development organizations.

3.3. Report Traversal

Traversal to other reports through links is possible. For example, within a section report by activity, a manager can link to an Effort-by-Person report, an Effort-by-Project report, or a Project-Summary report. From the Effort-by-Person or Effort-by-Project reports, a manager can link to an Hours report, a Project-Summary report, or back to the Cost by Activity report. We've described only a couple of traversal paths; however, many exist. In addition, a manager may link to any report by clicking on the report name listed in an index rather than traversing from report to report.

4. Authority Structure and Access Permissions

Access to FaMOUS is controlled through user password authentication and a restricted menu based system. A table is maintained to describe the organizational relationships and to determine access at each organizational level. The files needed on the FaMOUS server system to implement these access controls are described in the following subsections.

4.1. Organization Structure Table

This `orgstructure` table, shown in Figure 4, contains a line for each organizational unit and cost center. The first string determines the access hierarchy (or groups) and the remaining strings are the three character user IDs of managers or finance officers who should have access.

```
/fame                                              fame_developers
/fame/fmb                                          finance_manager
/fame/fmb/directorate                              associate_director
/fame/fmb/directorate/division                     division_director finance_officer
/fame/fmb/directorate/division/section             section_head
/fame/fmb/directorate/division/section/costcenter manager1 manager2
/fame/fmb/directorate/division/section2            section_head2
/fame/fmb/directorate/division/section3            section_head3
/fame/fmb/directorate/division2                    division_director2 finance_officer2
```

Figure 4: *Example* `orgstructure` *Table*

In the example, the groups are fame, fbm, directorate, division, section, costcenter, section2, section3, and division2. Each FaMOUS user listed in the `orgstructure` table is a member of groups at that organizational entry level and below. So, for example, a division director can see any report associated with the director's division. The Associate Director sees the divisions as well as directorate reports. The Finance Manager would further expect to access the control tables as well as all reports in the directorate. The FaMOUS developers can access everything needed by the finance manager, in addition to any items needed specifically for the development group.

4.2. Three Character User IDs and Passwords

A `perl` script *orggroup.perl* processes this `orgstructure` table to automatically generate the files used to determine access. The `.htpasswd` file contains three character user IDs and passwords, shown in the sample in Figure 5.

```
aaa:password_for_aaa
bbb:password_for_bbb
ccc:password_for_ccc
```

Figure 5: *Example* `.htpasswd` *file*

The systems administrator uses a separate workstation password server program to grab the standard system password for each user. Each user must also have an entry in the standard /etc/password file. Access to FaMOUS for the end user is provided only via the Web browser interface. Direct logins to the FaMOUS servers are disabled for all but the FaMOUS developers.

4.3. Group Access Files

Access to the reports directories is governed by the `.htgroup` file which is determined from the `orgstructure` table. Figure 6 shows a partial `.htgroup` file based on the `orgstructure` example in Figure 4. The first string on each line of an `.htgroup` file is a group name which is taken from the last level of the organizational structure in the column on the left of the `orgstructure` table. The access strategy is as described in the section on the `orgstructure` table.

```
fame        fame_developers
fbm         fame_developers finance_manager
directorate fame_developers finance_manager associate_director
division    fame_developers finance_manager associate_director division_director
```

Figure 6: *Example* `.htgroup` *file*

The `access.conf` file associates the groups in the `.htgroup` file with directories on the disk. Each directory is restricted in both the the domain `.ornl.gov` as well as to a specific group. The restriction to `.ornl.gov` provides a secondary level of security.

These three files allow the use of proper passwords for user authentication and access authorization to reports and to selection interfaces. As an additional bonus, the `orgstructure` table contains the three character user IDs of all users, which can be written as a `.mailrc` alias. This allows easy electronic mail communication with all users of the system.

4.4. Access via the Graphical User Interface

The same `orgstructure` table is read to create dynamically a home page table of contents that shows only the levels in the organizational tree that the user is authorized to see. A link is placed on the table of contents only if files exist within that directory. 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 upon a naming convention. Available subordinate directories are shown, and reports are grouped into administrative, programmatic, summary, or individual projects. This scheme provides quick, authorized access to any information in the full tree after no more than three mouse clicks and provides a logical grouping for efficient navigation.

The use of the Web browser interface also allows easy feedback from users through electronic mail and allows connectivity to external sources of information. References to people can be linked to individual home pages. Likewise project references can be made to Web project pages. Icons are used to reference phone and address listings and organizational charts. A `perl` routine is used to send electronic mail suggestions to the FaMOUS 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 from which comment originated.

5. Conversion of Reports for On-line Browsing

The goal for the FaMOUS 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. Web browsers such as NCSA Mosaic or Netscape provide an immediate solution for information presentation. Several steps are taken to convert the static financial information pages to interconnected on-line access.

A `perl` script is used to parse the ASCII report files using a two pass process. Keying on report titles allows the division of the sequential report files into separate HyperText Mark-up Language (HTML) pages which are stored in directories specific to the appropriate cost center and month. The first pass through the report files collects people and project titles within the directorate, and data for graphical representation (see below). In the second pass, hyperlinks are inserted to associated references to persons, projects and graphs. This cross checking insures that no links are created for inappropriate or non-existent connections. The links are inserted around predetermined elements within the headers or within the data records, chosen to provide a logical association between HTML reports. An HTML `name` anchor is also inserted so that links can be made to specific locations within reports.

A flat directory structure is used to store the reports, with subdirectories under the report's directory for each cost center and division, and (as will be described below) for sections and for the directorate. Within each of these subdirectories are further subdirectories for each month. Each report is separated into pages at each new title heading and is stored within the date subdirectory of its organizational unit. There are two exceptions. The first is for administrative reports. The reports dealing with cost center administration, for example, are provided for section heads who must monitor the costs and rates for each cost center within their section. Sections are not, however, present in any way in the cost accounting system. For this reason, an `orgstructure` table (shown in Figure 4) had to be generated to define the hierarchical organizational structure to assign cost centers to sections and sections to divisions. A second `sections` table provides the section titles corresponding to the abbreviation used in the `orgstructure` table and as the reports subdirectory name. The second exception is for the directorate reports. Similarly, information organizing divisions by directorate are not represented in the cost accounting system. These are added to the FaMOUS system in the same way as the sections are added as described above.

Several presentation constraints were imposed in the conversion to HTML. For example, implementations of HTML at the time FaMOUS was developed did not allow for tables or tabular data. The only option for presenting columnar information is using the $< PRE >$ tag for preformatted text. Browsers implement this with a monospaced font. However, on the Macintosh a bold font is wider than a normal font even for fixed-width fonts. For this reason labels could not be rendered in bold to visually separate them from the data. These restrictions will not be present in future browsers that implement HTML tables. There will, however, potentially be users who view the data with a text-based browser. Either the use of graphical browsers will have to be required, or the above restrictions will have to be observed.

Presentation considerations such as font size and hyperlink color are the responsibility of the browser and cannot be specified in the stored HTML report pages. A start-up script was written for NCSA Mosaic or Netscape to choose fonts and link colors. For the PC and MAC platforms, suggestions are given on the FaMOUS project page for browser preferences.

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, the `perl` script dumps information required to produce the graph into a file with a `.dat` suffix and 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 7.

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

Figure 7: *Input file for C program that produces Postscript graphs.*

In Figure 7, `LBA` indicates that a layered bar graph is to be produced. There are three title lines. The five numerical values are New BA, BA Carryover (BACOV), BO, Commitments, and FY cost. 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 8. For this report, the full color figure supported by the FaMOUS system has been modified to a gray scale version.

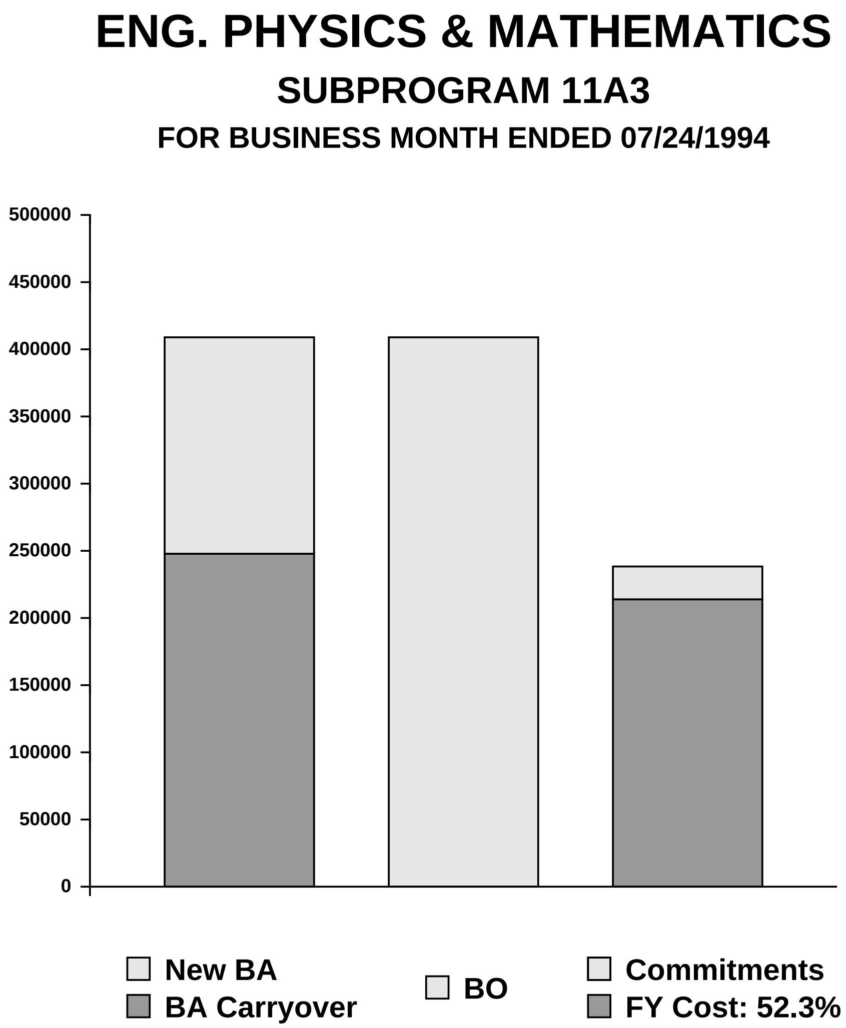

Figure 8: *Postscript Graph as produced by C program.*

The FaMOUS 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 (e.g. funding levels for a directorate by program) to the lowest level (e.g. the cost of an individual researcher's effort to specific accounts) are illustrated graphically.

6. Selection of Information for Downloading

The FaMOUS servers are workstations with large disks (9 Gigabytes) and expanded memory (128 Megabytes). Each server processes information for a specific logical unit of the overall organization. For example, two servers are currently in place, one serves the Advanced Materials, Physical, and Neutron Sciences (AMPANS) Directorate, and the other serves the Computing, Robotics, and Education (CRE) Directorate. An individual FaMOUS server downloads only the information that is needed to process the reports for the specific unit it serves. To accomplish this goal, several selection tables (designated "control tables") are maintained by the unit's finance officer. The control tables provide enough flexibility so that accounts associated with the natural hierarchy of an organization can be accessed. At ORNL, for example, LDRD (Laboratory Directed Research and Development) funds are associated with the Central Management Offices rather than with the divisions to which the funds have been allocated.

The FaMOUS system generates reports in a generic manner for each organization. The control tables simply fill in the specifics of which directorate, division, section, and projects are to be processed on a particular server.

Each 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 FaMOUS and show some illustrative examples.

6.1. The `SELECT` Control Table

The `SELECT` control table specifies which information in the Cost Accounting System (CAS) 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 during the automatic download of a FaMOUS server database. An example of a `SELECT` control table for the CRE directorate is shown in Figure 9.

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 9: `SELECT` *control table for the CRE directorate.*

Each line specifies a set of records to be fetched from the CAS databases. For example, the first line indicates that all records for which the Operating Unit/Division attribute, OUDI, has value X005 (which is the OUDI attribute value for the Office of Science Education) are of interest to the CRE directorate. Line two is used to pull records for which the Account Description attribute, ACDE, 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 Office Science Education. Other lines in the table indicate activities and accounts of interest to the CRE 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 9. 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 NXACTV0 set to KC07 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.

6.2. The DIVISION Control Table

The DIVISION control table associates the records of a directorate with divisions in that directorate. Its structure is analogous to 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 CRE directorate is illustrated in Figure 10.

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 10: DIVISION *control table for the CRE directorate.*

Note the use of negative attributes in the DIVISION control table. For example, the Computer Science and Mathematics Division, X012, is responsible for all Field

Work Proposals (FWPs) 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 10; only those required by the CRE directorate are included.

6.3. Other Control Tables

The COST CENTER (CC) 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 CC attribute is an eight-digit account number. The PROGRAM attribute is an eight-digit account number followed by a two-character program descriptor. Here are some examples. Account number 34100010 specifies the Mathematical Sciences Section cost center. Account number 34500001KC designates the KC07 program which administered by the Center for Computational Sciences. The Advanced Neutron Source (ANS) program in the Advanced Materials, Physical, and Neutron Sciences (AMPANS) directorate is designated by 32120007KE.

6.4. 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.

7. Downloading Selected Data with Automatic Transfer

Wherever possible, FaMOUS downloads data from the central databases on System 61 (ksv2.priv). System 61 runs the VMS operating system and is equipped with Multinet to allow Internet communication with other systems. The FaMOUS 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 user interface to the selection control table (Figure 9). This 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 OUDI attribute. However, some tables name this field DI (e.g., the COMMAC table). The FaMOUS software currently allows only the attribute OUDI and will map OUDI to DI for the COMACC table. The mapping relieves the FaMOUS administrators 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.
- Database fields are mapped to table-specific fields as necessary.
- A DEC Command Language (DCL) File is generated from the mapped control table parameters.
- The DCL file is submitted to System 61 using the utility *rsubmit*. 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 FaMOUS server.
- An SQL and a database loading command file is automatically generated to load the flat-file database into Informix on the local FaMOUS server.

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

7.1. Selection Criteria Control Table Format

An Informix table, called a control table, holds the current selection criteria. This provides independence from the programs that use the selection criteria. In this manner, changes to the selection criteria table are propagated throughout associated programs. An ACE-formatted report, DUMPctl, is run to generate the selection criterion from the `SELECT` control table. The DUMPctl output is used internally by the system and need never be seen by the user. The first non-blank line of the DUMPctl output indicates the field names selected. A prepended "NOT" before a field name indicates a negative selection criteria.

Fields are separated by a vertical bar. The subsequent lines in the DUMPctl output specify the actual selections. Field entries occurring within the same line specify a database query based on all the the entries (e.g. logical AND operations). Each line represents a separate query. Hence, multiple lines represent queries combined via logical OR operations. The following example illustrates the DUMPctl output:

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

The field names are OUDI and ACDE. The first line selects all account descriptions beginning with `05-`. The second line selects all records with account descriptions beginning with `10-` but not those beginning with `10-01`. The last line selects all records with `OUDI` value of `X010`.

7.2. Control Table Mapping

The control table generated by DUMPctl has a 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 specified control table to write a new control table with the renamed headings. The list of mappings 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. In the case of a specified attribute for which there is no appropriate corresponding attribute in the the table, *mapctl* effectively ignores (maps the attribute to "nothing") such an attribute.

7.3. 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 FaMOUS 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 FaMOUS 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, attributes that should be saved when a data subset is created, and information that indicates 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`. A corresponding keyword `addselect:` exists that indicates the 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 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 transfers the database to the FaMOUS server.

7.4. Remote Retrieval

The process of selecting data, writing out the table, and copying it back to the local FaMOUS 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 those failures. UNIX and the Multinet extension to VMS support two methods for network logins. One is by use of the "rlogin" (remote login) command and another is by the use of the "rsh" (remote shell) command. A DCL file is generated to access the ALPHA database. During the execution of these DEC commands, it is important that we be able to determine the status of these "batched" commands. The typical method in UNIX is to redirect the standard input to a file instead of a user's terminal. This does not work for rlogin because rlogin expects an interactive terminal device. The UNIX "rsh" command does not work because the Multinet implementation does not read all the system variables properly to start ALPHA. Further, 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. Rsubmit 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. The final rsubmit step loads the flat-file database into the local Informix database.

7.5. Inserting Retrieved Data into the Local Database

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. After the SQL and command files are built, execution results in the flat-file database being loaded into the Informix 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 11, and the resulting SQL file is illustrated in Figure 12.

```
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 11: *Structure file for* `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 11 is shown in Figure 13.

```
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 12: *SQL file to create local* ACCOUNTS *table.*

```
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 13: *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 loaded into the local Informix database in a manner similar to the information fetched from the CAS databases.

7.6. Interface

An interface exists that integrates all the rsubmit pieces together. The interface allows the user to update single or multiple tables within Informix. The user selects the table (or all tables) to be updated and clicks on one button. For users familiar with ALPHA and DCL, the command file that has been generated by the software is displayed. If the user is satisfied with how the selection criteria have been interpreted, then a submission to System 61 requires one mouse click. Writing the flat-file databases on the central system requires time. However, after the flat-file databases are generated, the ftp and Informix database load processes are fairly quick. After the Informix database is loaded, the reports are generated and converted to HTML. From this point, the user has quick access to the entire financial status of the organization at the click of a button.

8. Budget Manipulation Interface

In addition to the control table interface described in a previous section, one special user interface for creating and manipulating budgets also exists. A budget can be efficiently created with a few mouse clicks using scrolling menus. One menu allows the user to select the 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. The fiscal year's costs, entered for any line item ID, are automatically distributed across the fiscal months. This distribution is based upon the number of employees and the number of work days during those months. Division burden and fringe benefit costs are allocated based upon labor costs. When the line item ID labor is changed for any given cost center within a division, the division rate is recalculated and reallocated to all cost centers automatically. After changing any information, the budget is 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

FaMOUS has already become a useful tool for easily bringing financial information to those who need to track accounts. The overall system works well, but it should be regarded as experimental. FaMOUS started from a small core of useful 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 online format to be modified. The online format 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 also make it easy for users to download a particular page of data into a spreadsheet format for further specialized queries.

Access control is also essential, but the proliferation of networks means that data is no longer secure while in-transit from the FaMOUS 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 FaMOUS to exist outside of the laboratory, experimental implementations at other organizations are essential to discover similarities and differences among financial structures. FaMOUS 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. It will be necessary to see how these financial organization differ, and to determine how existing reports be easily adapted to a new structure.

10. Conclusions

This report has outlined the steps that are performed each month for on-line financial reports to reach end-users. The data initially resides on disparate tables on a central legacy system. Specific subsets of data are selected and downloaded onto FaMOUS workstation servers. Reports using a commercial database system are created based upon selection criteria kept in several control tables. The reports 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.

FaMOUS is an advanced tool for analyzing, presenting, graphing and tracking the status of an organization's existing financial status. It has a number of advantages over direct access to ALPHA. FaMOUS's powerful, yet straightforward graphical user interface saves management and financial officers time when tracking budgets and cost detail by providing such information at a glance. It frees time for the finance officers to do crucial planning and analysis. And, it provides an interface for communication throughout management. It is a step toward the paperless office.

Work needs to continue to determine if a generic set of programs that implements FaMOUS capabilities can be developed for use at other government and business sites.

11. Acknowledgments

The authors would like to acknowledge all those who have provided information and support necessary for the completion of this report. In particular, we acknowledge the helpful comments on this document from Mark T. Elmore and J. Mark Hamby.

12. 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.

ORNL/TM-13139

INTERNAL DISTRIBUTION

1. K. D. Barnes
2. T. S. Darland
3. J. M. Donato
4. M. T. Elmore
5. D. M. Flanagan
6. G. A. Geist
7. N. W. Grady
8. J. A. Green
9. J. M. Hamby
10–11. M. A. Harris
12. H. R. Hicks
13. F. M. Hoffman
14. L. L. Horton
15. J. A. Kohl
16. M. R. Leuze
17. M. G. Manes
18. N. P. Norton
19. C. E. Oliver
20. P. M. Papadopoulos
21–25. S. A. Raby
26. J. W. Reed
27. J. A. Rome
28–32. R. F. Sincovec
33. R. J. Verastegui
34. Central Research Library
35. ORNL Patent Office
36. K-25 Appl Tech Library
37. Y-12 Technical Library
38. Lab Records Dept - RC
39–40. Laboratory Records Dept

EXTERNAL DISTRIBUTION

41. Charles Aulds, US Army Missile Command, Corporate Information Center, Redstone Arsenal, Alabama
42. Bruce K. Behn, University of Tennessee Accounting Department, 621 Stokely Management Center, Knoxville, TN 37922
43. John Cavallini, Deputy Director, Scientific Computing Staff, ER-7, Applied Mathematical Sciences, Office of Energy Research, U. S. Department of Energy, Washington, DC 20585
44. Ginger A. Caldwell, National Center for Atmospheric Research, P. O. Box 3000, Boulder, Colorado 80307-3000
45. Maureen Davidson, Director of Technology, Albuquerque Public Schools, 220 Monroe, SE, Albuquerque, NM 87108
46. Donald J. Dudziak, Department of Nuclear Engineering, 110B Burlington Engineering Labs, North Carolina State University, Raleigh, NC 27695-7909
47. Amy W. Gatian, University of Tennessee, Accounting Department, 620 Stokely Management Center, Knoxville, TN 37922
48. Dr. Jose'-Marie Griffiths, Vice Chancellor for Information Infrastructure, 507 Andy Holt Tower, University of Tennessee, Knoxville, TN 37996-0157
49. Daniel A. Hitchcock, Scientific Computing Staff, ER-30, Applied Mathematical Sciences, Office of Energy Research, U.S. Department of Energy, Washington, DC 20585
50. Fred Howes, Mathematics, Information, and Computational Science Division, ER-31, Applied Mathematical Sciences, Office of Energy Research, U. S. Department of Energy, Washington, DC 20585

51. Carol L. Hunter, Lawrence Livermore National Laboratory, P.O. Box 808, L-301, 7000 East Avenue, Livermore, CA 94551
52. John D. Jarrard, Director, Computing & Administrative Systems, 2417 Dunford Hall, University of Tennessee, Knoxville, TN 37996-4050
53. Dr. Gary Johnson, Mathematics, Information, and Computational Science Division, ER-31, Applied Mathematical Sciences, Office of Energy Research, U. S. Department of Energy, Washington, DC 20585
54. Pamela K. Kappelhof, Lawrence Livermore National Laboratory, P.O. Box 808 L-319, Livermore, CA 94551
55. James E. Leiss, Route 2, Box 142C, Broadway, VA 22815
56. Jacob V. Maizel, Jr., Chief, Laboratory of Mathematical Biology, Division of Basic Science, National Cancer Institute, Building 469, Room 151, Frederick Cancer Research and Development Center, Frederick, MD 21702
57. Neville Moray, Department of Mechanical and Industrial Engineering, University of Illinois, 1206 West Green Street, Urbana, IL 61801
58. David Nelson, U.S. Department of Energy, Director Scientific Computing Staff, ER-7, Applied Mathematical Sciences, Office of Energy Research, U. S. Department of Energy, Washington, DC 20585
59. Arvo W. Schoen, PlainFieldGroup, 24570 Road 95, Davis, CA 95616
60. Bob Ward, Head Department of Computer Science, Ayres Hall, University of Tennessee, Knoxville, TN 37996-1301
61. Mary F. Wheeler, Rice University, Department of Mathematics, P.O. Box 1892, Houston, TX 77251
62. Aaron Wisner, Department of Accounting and Analysis, CR-42, Department of Energy, Washington, DC 20585
63. Office of Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations Office, P.O. Box 2001 Oak Ridge, TN 37831-8600

64–65. Office of Scientific & Technical Information, P.O. Box 62, Oak Ridge, TN 37831