Web-Based Tools for Assessment Automation

Jim C. Harden, Michael G. Lane

Department of Electrical and Computer Engineering,
Mississippi State University, Mississippi State, MS 39762

Abstract

In this paper we describe a computer-assisted automated system for EC2000 assessment. Calendar scheduling is used in conjunction with web-based assessment instruments using web scripts to funnel data into an SQL database. Though we offer no easy, all-encompassing solution, this paper does share our experiences in moving toward a maintainable system for supporting course evaluations, ethical and contemporary technical issues documentation, and a number of constituent surveys.

Ideally, the infrastructure needed to enable program improvement would support the extraction of pertinent data from a common database, conveniently placed there on time by willing constituents. What may have been a relatively infrequent compilation of materials under the old criteria has been transformed into a process requiring continuous support. The acquisition and management of this data can be a concern for otherwise busy faculty members who may assume a responsible role for shepherding the assessment process. Though assessment events occur regularly, in some cases once or twice a year, they are not always so frequent as to make them routine. Since they are a part of a dynamic process of program improvement, the criteria and assessment methods are prone to change over time. It is clearly a non-trivial process and maintaining the quality of faculty and other constituent participation can require additional effort as the time becomes further removed from an accreditation site visit. We present a framework, primarily based on freeware, that promotes the goals of methodical, convenient, and secure data entry. This web-based system also supports data extraction, and outcome and objective analysis to reinforce the goals of program improvement under EC 2000.

I. Introduction

A review under ABET EC 2000 is based on the continuous collection of diverse assessment data[1]. Faculty and staff will secure, analyze, and accumulate data to assess how students are meeting various criteria. These data form the basis for the updating of educational objectives and

the determination of progress in meeting measurable outcomes. The timeliness and ease of data entry and extraction are critical to obtaining results needed to make continuous improvement.

An effective assessment data acquisition scheme must support the compilation of data from diverse activities and constituents, and do this at intervals spaced over months or years. Problems can arise when the data gathering is loosely supervised and unorganized. While the mechanics of the implementation of data collection processes do not drive assessment itself, they are nonetheless a key and nontrivial component. A self-organizing, time-efficient approach is needed since faculty and staff are likely to have many other activities vying for their attention.

Although benefits of on-line, computer-assisted assessment for specific activities such as collaboration and teamwork, portfolios, and surveys have been reported[3,6,9],the collective load of data collection and analysis is motivating a look at more encompassing assessment management tools[5]. For those responsible for the administration of assessment, there is a need for an integrated, general framework providing systematic coordination and maintenance of the assessment process. For those participating in assessment, such a system can provide a central, familiar interface. However, because of the diversity of the assessment methods, a goal of a totally comprehensive system may be elusive. Nevertheless, important organizational and access advantages can be gained by a more inclusive view of computer-assisted assessment.

Combining the use of the World Wide Web with office related software can accommodate substantial elements of the process. The web is a widely accessible portal for data gathering and communication, an important feature since a program's constituency may be distributed and diverse. A web-based scheme can use common types of applications available such as web pages developed for gathering survey or other evaluation data, database software for storing and accessing data, calendars for triggering events, and spread sheets for data analysis and presentation.

II. Example Assessment Environment

This paper uses the example of a specific assessment environment developed for the Electrical Engineering and Computer Engineering programs at Mississippi State University. The implications for other programs are left to the reader but the Computer Engineering example of Figure 1 surfaces a number of the key issues. Four constituents are identified: alumni, employers, students and faculty. Various assessment tools are used to obtain constituent input including an array of surveys and questionnaires, course evaluations of specific outcomes, professional membership statistics, and ethical and contemporary technical issues documentation. Furthermore, core course instructional objectives are monitored and evaluated in connection with measurable outcomes. This information must be processed by the ECE and CS staff and the Computer Engineering program coordinator. Consideration of program assessment materials involves the interaction of the ECE Undergraduate and Computer Engineering Steering Committees with an Industrial Advisory Board, the ECE and CS faculties, and focus groups.

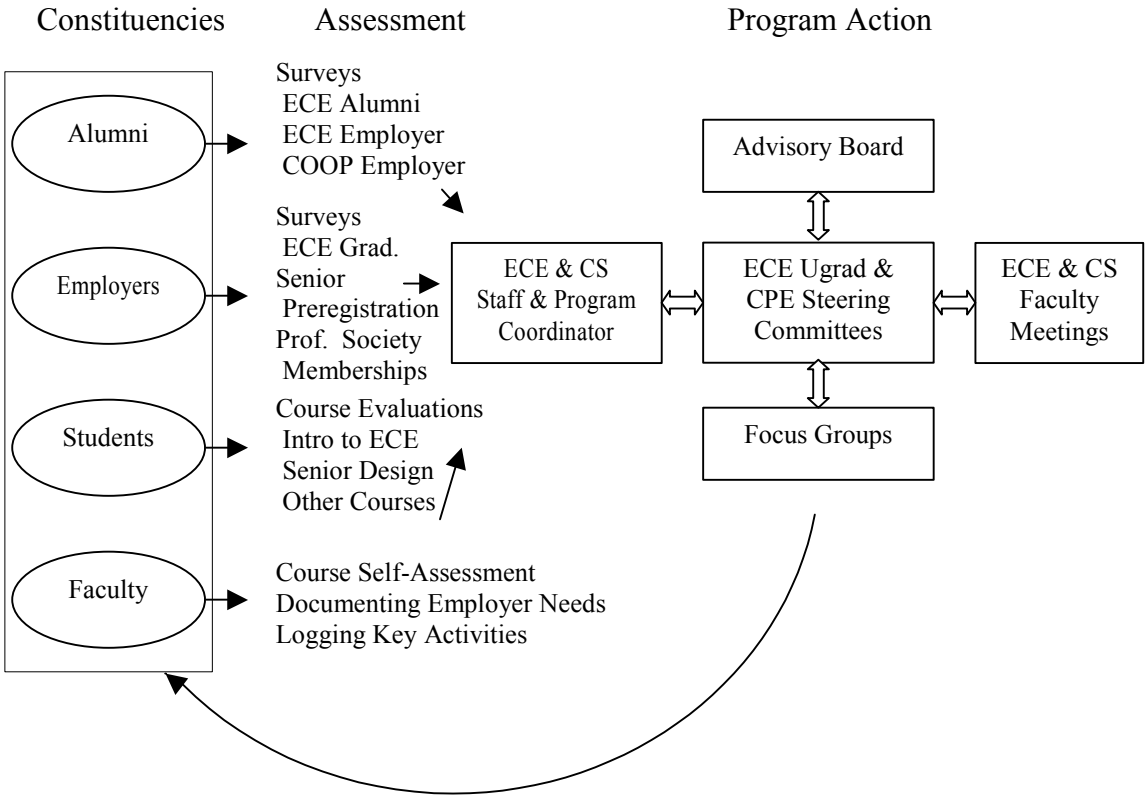

Figure 1. Example Infrastructure Supporting the Assessment Process

Perhaps the most critical component of an assessment data management system is data entry. Without the availability of the raw data, securely and unambiguously preserved, no presentation of results may be assured. As can be seen from Figure 1, data entry occurs at the interface between the constituents and the assessment tools. The goal is to directly capture the assessment event, thereby avoiding unnecessary administrative handling of raw data. This means linking a diverse group of constituents in various locations directly to the assessment database, an obvious match for web-based acquisition.

Unfortunately, not all constituents will appreciate the need for their web-based participation. This can occur for a number reasons. Convenience (perceived or otherwise) can play a role. A professor or employer may have an existing method of assessment that they prefer to retain. In most cases web entry can be made attractive, so this will not be a major problem. Caution, however, should be used when releasing a web-based instrument to the constituents. If the system has even a single bug, possibly only revealed by a particular user, it could spell the end of direct entry by that constituent.

Compatibility issues can also be important when designing web interfaces for survey use. Although the prevalence of a number of web clients removes several barriers, such as operating

system dependence and development time, their differences in interpreting and displaying HTML code sent from survey scripts must be carefully taken into account. Differences in appearance or responses between web browsers can have a negative effect in the overall acceptance of this medium for data collection.

Access to Assessment Instruments

Two primary web pages provide access to the system – one for constituents listing the on-line instruments, and another for administration. Neither is linked to the department web site. Constituents normally gain access for assessment data entry by being supplied the web address in an email. Others may be given the page address for the on-line list of instruments. Users of various instruments are authenticated through an LDAP (Lightweight Directory Access Protocol) server. By using an LDAP server, consolidation of maintenance can be achieved since many new authentication schemes have support for LDAP. Students use the same ID they use to log into the department's computer systems. For employers and alumni, a common user name and password is supplied to them. This gives the coordinator the flexibility of having static or dynamic passwords for these surveys.

The administration features are more closely guarded and help key faculty and staff monitor the status of data entry, tabulate results, and make corrections. It has proven helpful to monitor the level of constituency participation. In addition to providing a history of event-by-event participation, tracking the number of respondents *during* an assessment event enables corrective action to encourage a good sample. Although the evaluation of results can occur in a number of ways including the importing of data into Excel spreadsheets for tabulation and graphical display, one administrative tool uses Perl scripts to automatically provide results. This is the more time efficient way of tracking assessment results by semester or year. A maintenance feature allows for the deletion of entries. This capability is not generally available to the constituent since good data could be lost by mistake or through mischief. Should, for example, a faculty member providing course evaluation information wish to remove an entire entry, he or she may request that this be done by the assessment coordinator.

System Software Used in Assessment Automation

Figure 2 provides an overview of the software components employed in the assessment system. Each of the major components used in the implementation is demonstrated in the figure.

Apache

The Apache[2] web server is one of the most popular web servers being used on the Internet. In addition to its stability, it is cross platform and offers a number of extensions to enhance the server functionality. For our environment, only three extensions were needed: a Perl, PHP, and auth_ldap module.

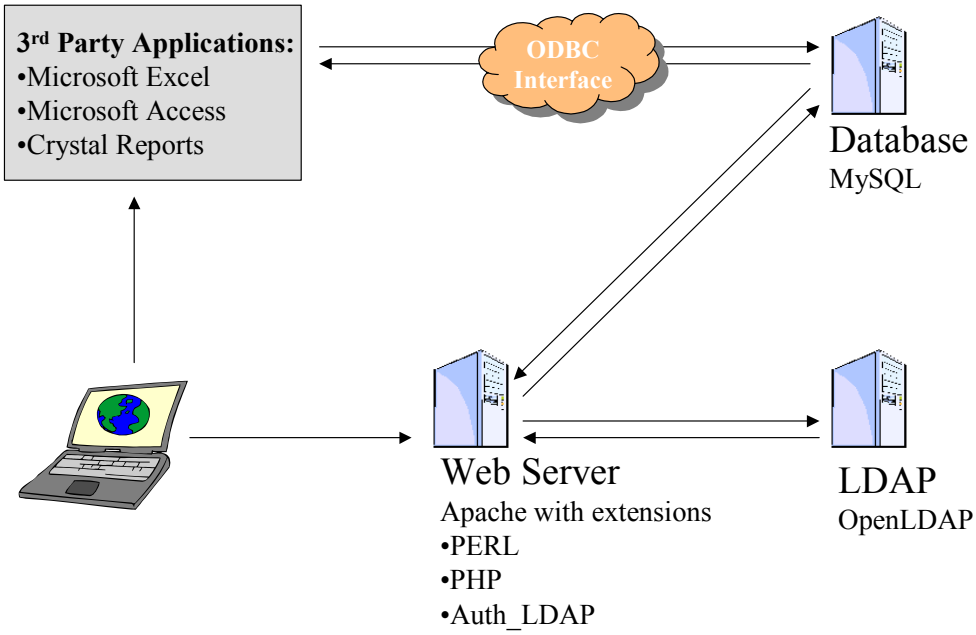

Figure 2. Client-Server Implementation of Assessment Automation System

MySQL

For this particular project, the department chose to use a publicly available SQL database called MySQL. As Open Source Software, MySQL is available for downloading at no charge from the Internet[10]. Use of the software is controlled under GPL (GNU General Public License)[7].

A project of this nature is not limited to this particular database though. MySQL was chosen for the merits that fit the department's needs, however, the scripts and tools used are compatible with a vast majority of SQL compliant databases that have an ODBC (Open Database Connectivity) driver.

Scripts – Perl and PHP

The scripts that generate the surveys in this project were written in Perl and PHP. Perl and PHP are also Open Source Software, and available for downloading at no charge. Information about these scripting languages can be obtained from their respective web sites [4,11].

Once again, the use of Perl and PHP, as with MySQL, was chosen because of prior departmental usage. Everything currently done in the department's surveys could easily be done in another

language such as Microsoft's ASP extensions, assuming they can interface with the particular database being used.

Perl is a flexible language that offers a great deal of control with all aspects of the survey development and presentation. It is also extensible through the use of reusable modules. Perl has support for many databases natively, as well as through a programming interface called DBI/DBD. This project was developed using the DBI/DBD interface because it does not require the same database to be used should it fail to accommodate the department's needs in the future. With minimal code changes, a new database such as Oracle could be used.

PHP is a convenient and powerful language to quickly develop web-based content. As with Perl, PHP offers a convenient programming interface to interact with a variety of databases. PHP also is supported on webservers other than Apache, including Microsoft IIS and Netscape.

LDAP

An LDAP server is used to provide authentication to each survey. The particular server used in the department is OpenLDAP. The web server is coupled with a module called auth_ldap to provide a very elegant method for providing a simple security model. Security of survey information should be a concern to ensure the integrity of the information obtained. LDAP was used for this project because it was already in use and allowed the surveys to be secured without adding additional maintenance. There are many other ways to secure web pages and scripts. Using the Apache web server for example, participants could be authenticated from NIS servers, NT domains, Novell NDS, or from flat files just by adding modules to the server.

Automated Reporting Using Perl

Easy access to results of web-based assessment tools is essential. Although entering data into the database is a critical step, spending extensive amounts of time inefficiently extracting the results must be avoided to make the system truly workable. It is only natural for the initial focus to be on getting the assessment instruments developed, but the benefits of streamlining reporting will be appreciated in years that follow.

Fortunately, web scripts can be designed to provide periodic reports online. An assessment instrument and reporting script will share common content, so there is also a developmental efficiency in creating them together. Since assessment tools may change, it may be desirable to identify those segments of the instrument that is relatively static for automated reporting.

An example based on Perl scripts is presented in Figure 3. It shows a web interface for generating an employer survey report and a section of the resulting report.

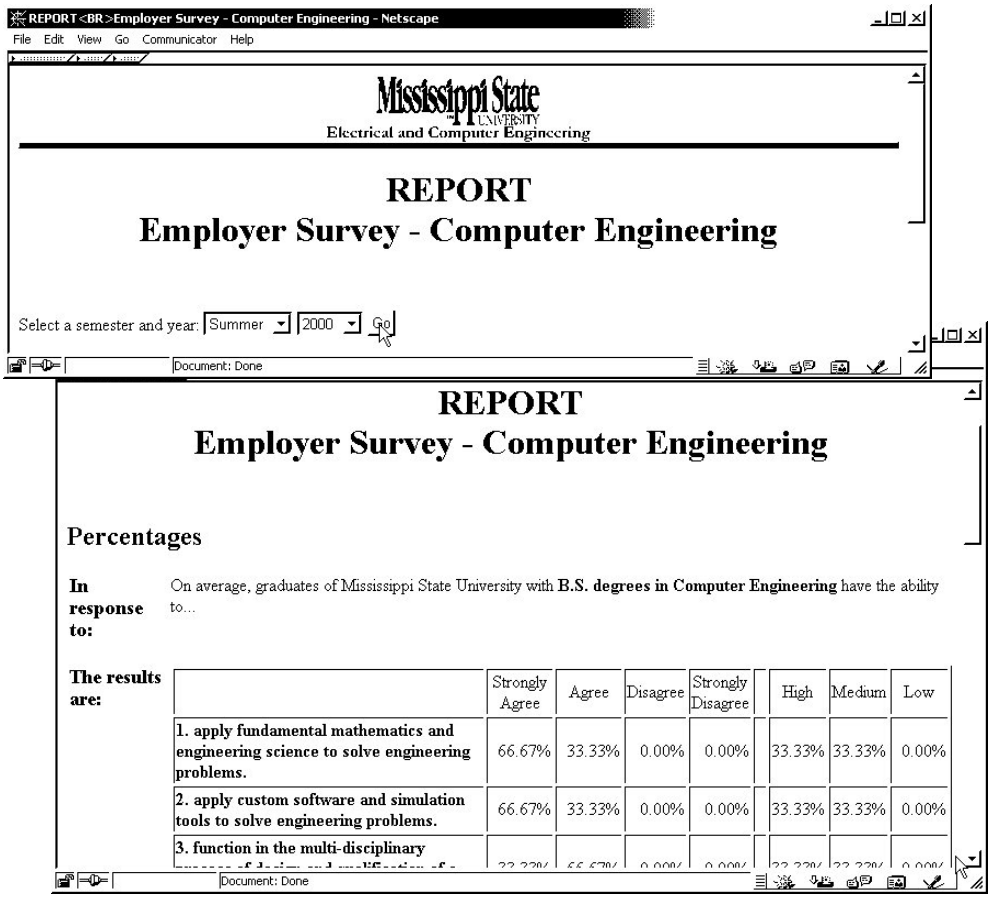

Figure 3. Report Selection Menu (upper window) and Portion of Resulting Report (lower window)

Viewing Results Using Excel

The most flexible way to tabulate, analyze, and present results is through a database query using Excel. As a change in academic program is considered, quick feedback on particular issues can be critical to the decision making process. With web-based surveys, questions may be added at the last minute and results returned in a matter of days. By acquiring this data in a spreadsheet, the results can be conveniently manipulated and graphically displayed to address key questions. This enables a more responsive system where issues raised by one constituent can be considered expeditiously by the rest.

In the case of MySQL, the MySQL ODBC client driver is installed on the computer system gathering the information. Once installed, a system DSN (Data Source Name) is created for each database that gives the pertinent information for connecting and authenticating. Once this is

done, any reporting tool using ODBC interfaces can be used to collect and manipulate information, including Excel.

To capture database information requires the identification of the data source and specifics about the query used to return the desired data. For Excel, a typical selection sequence requires *Data → Get External Data → Create New Query* followed by the selection of New Data Source where the query is specified for the MySQL database under password protection. Specific questions within an assessment instrument are identified using IDs associated with *fields*. The administration web page includes a link to a Perl script that relates these field IDs to assessment entries so the Excel headings can be identified properly. Figure 4 shows a portion of the Graduating Senior Survey and the corresponding Survey Format window identifying the question IDs. Once it is finalized, saving each query enables its reuse and supports documentation of the process. Details for external data access operations can be found in Excel references[8].

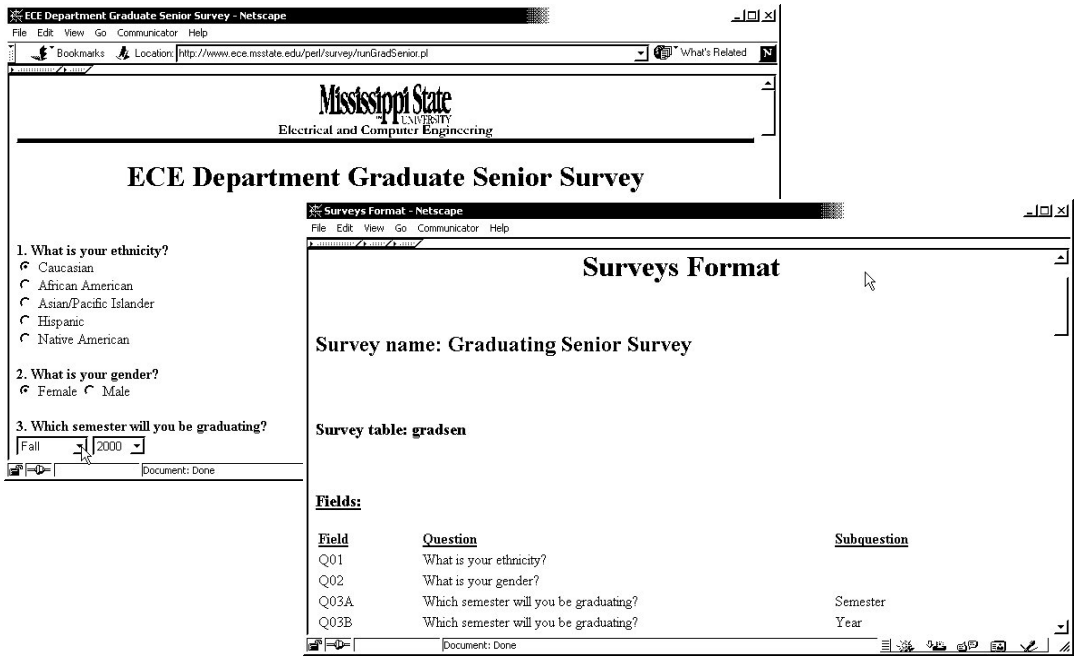

Figure 4. Portions of Example Survey (upper window) and Corresponding Format File (lower window)

Calendar

Assessment sequences typically involve a number of steps distributed over a period of a single semester to several years. A UNIX- or Windows-based desktop calendar tool is useful in coordinating these assessment events since it provides timely notification of key activities and documents a schedule of systematic action. Beyond the reassurance it provides to the assessment

coordinator that key activities will not be forgotten, it also is a convenient way to share tasks with other faculty and staff through common calendar access. With sufficient detail for each calendar item, it is possible for staff personnel to take the required actions or work with the coordinator as needed.

An example sequence of calendar entries for an event may include messages and dates for

1. reexamination of the assessment instrument prior to reuse,
2. modification of the assessment instrument and reporting tool,
3. selection and notification of the constituents expected to respond to the event,
4. follow up message(s) to ensure participation,
5. compilation of results,
6. evaluation and presentation of results, and
7. determination of actions.

Features of the desktop calendar tool facilitate the periodic placement and editing of action items over multiple years.

III. Summary and Future Work

For all participating in the assessment process, EC 2000 raises a number of procedural and data managerial issues. Conserving time, insuring data integrity, effectively reaching a diverse and widely distributed constituency, and documenting regular assessment events are all concerns. In this paper we have provided examples of computer- and web-based methods that may be useful in addressing these issues. These methods employ a database as an inclusive repository for assessment data, web-based interfaces for entry and reporting, and a shared calendar for triggering time-critical events.

This web-based system for managing assessment data is an ongoing project. Further efforts will focus on reporting, integration, and improving user interfaces. This will include the use of Excel for integrated reports, simultaneously drawing from a number assessment instruments and periods. We also plan to extend our efforts to generalize both the Perl and PHP scripts for automating reporting for new or changing instruments. Other reporting tools with graphical capabilities will be investigated such as Crystal Reports. Another area of interest is incorporating additional assessment activities into the database. This includes the importing of externally generated data such as Co-op evaluation data and standard exam results. By generating a few basic scripts, we have been able to incorporate numerous data sets, supplied as comma delimited files, from other entities. We will also continue to address any user interface issues that surface as the tools gain wider utilization. To expedite data entry, for example, we will expand the ability to enter new information by editing prior entries.

Bibliography

1. Accreditation Board for Engineering and Technology, Inc. (2000). *Criteria for Accrediting Engineering Programs*, Baltimore, MD.
2. Apache information, (2002). http://www.apache.org
3. Chung, G., Harmon, T., Baker, E., (2001). ," Potential Uses of On-Line Performance Assessments in Engineering Education: Measuring Complex Learning outcomes and Processes, *Proceedings of the American Society for Engineering Education*, Albuquerque, NM.
4. CPAN: Comprehensive Perl Archive Network for distribution and documentation of Perl, (2002). http://www.cpan.org.
5. Diefes-Dux, H., Haghighi, K., (2001). ," Web-Based Technology for Long-Term Program Assessment, *Proceedings of the American Society for Engineering Education*, Albuquerque, NM.
6. Estell, J., (2000). "On-Line Assessment for Web-Based Programming Portfolios," *Proceedings of the American Society for Engineering Education*, St. Louis, MO.
7. GNU Project web server providing software and documentation, (2002) http://www.gnu.org.
8. Ivens, K., Carlberg, C., (1999). *Excel 2000: The Complete Reference,* Osborne/McGraw-Hill.
9. McGourty, J., Sebastian, C., and Swart, W., (1998). "Developing a Comprehensive Assessment Program for Engineering Education," *Journal of Engineering Education*, vol. 87, no. 4, pp. 355-361.
10. MySQL source of information and downloads, (2002). http://www.mysql.com/downloads/index.html.
11. Perl Mongers: A Perl advocacy web site, (2002). http://www.perl.org.

JIM C. HARDEN

Jim C. Harden received his B.S. degree from Mississippi State University in 1965, his Masters from Georgia Institute of Technology in 1966, and his Ph.D. for Texas A&M University in 1985. He joined the faculty of Mississippi State University in 1985. His research interests include parallel and distributed computing. He served as computer architecture thrust leader in MSU's NSF Engineering Research Center before becoming coordinator for Computer Engineering at Mississippi State University.

MICHAEL G. LANE

Michael Lane received his B.S. degree in Computer Science/Mathematics from Francis Marion University in 1996. He joined the staff of Mississippi State University in 1998 where he is currently the systems administrator for the Department of Electrical and Computer Engineering.