

# A Laboratory Information Management System (LIMS) for an Academic Microchip Fabrication Facility

Robert W. Hendricks<sup>1</sup> and Matthew R. Learn<sup>1,2</sup>

<sup>1</sup> The Bradley Department of Electrical and Computer Engineering

<sup>2</sup> Entomology Department

Virginia Polytechnic Institute and State University  
Blacksburg, Virginia 24061

**ABSTRACT:-** A relational database laboratory information management system for managing the personnel, facilities, equipment, and instruments associated with a university microelectronics facility is described. Emphasis has been placed on assuring personnel safety. The system includes capabilities for managing training and access to facilities, instruments and tools, for monitoring task assignments to personnel, for tracking inventories of supplies and capital property, and for handling maintenance of both facilities and equipment. There is a built-in mail system that allows communications with personnel based on a wide range of sorting choices including user classification and facility and instrument access rights. As is unique to university facilities, the system is also capable of tracking courses that may be taught in different laboratories.

The LIMS gathers data from an unlimited number of computers in an unlimited number of widely separated facilities over a distributed computer network using TCP/IP communications over the University LAN and a Microsoft SQL 2000 Server. We have used coding technologies that allow LIMS administrators to add and delete an unlimited number of users, facilities, instruments, inventory items, and courses. The system is thus expandable and capable of handling the smallest to the largest facilities without any recoding.

User access is via several custom programs coded in VB and via protected WWW access on pages generated in HTML and with active server pages (ASP) from a Microsoft IIS server. Both the VB programs and the ASP pages are developed in a highly modular form with security controls that allow the laboratory administrators to control access to each module at three levels—none, read only, and read/write. The code makes use of a large number (currently over 100) SQL 2000 stored procedures. These procedures make it easy to perform complex SQL operations.

Data integrity is maintained using a strategy of using multiple servers to separate various server functions (FTP, HTTP, and SQL) and by using multiple levels of backup. The SQL 2000 database is backed up dynamically and daily to a separate PC, and the other servers (FTP and HTTP) are backed up daily over the University LAN using Tivoli.

## 1. INTRODUCTION

The operation of a university microelectronics fabrication and characterization facility that is used for both education and research by a wide range of faculty and students with quite disparate backgrounds is a complex task. Typical tasks that must be monitored and information that must be acquired include: data concerning the management of people and their access to facilities, instruments, and tools; the management of facilities, including access, cleaning

and maintenance; the management of instruments and tools, including user access and instrument calibration and repairs; and information about inventories of supplies and spare parts. In addition, special consideration must be given to procedures for managing laboratory courses that may be taught in the facilities. Primary emphasis must at all times be directed at maintaining a safe workplace for our students, and at maintaining data integrity concerning our facilities, equipment, instruments, and supplies.

We are developing a laboratory information management system (LIMS) to assist us in the operation of our microelectronics cleanroom, packaging laboratory, and device characterization laboratories as well as their supporting infrastructure such as storerooms and gas bottle cages. The system described in this paper is an outgrowth of work reported previously by Eckerman and Hendricks [1] in which we attempted to perform these tasks using a commercial LIMS designed for data acquisition in analytical laboratories. We found that such systems, although ideal for their stated purpose, did not work well in our academic environment. We have chosen to develop a custom system using distributed PCs operating in a purely Microsoft software environment in order to make use of the very attractive site licenses available under University contract and the enormous background in developing and operating such systems.

## 2. STRATEGY

There are innumerable ways in which this management task may be undertaken. Of course, one could develop a very simple system based on handwritten information recorded in various paper logs maintained at each facility and instrument. We believe that such a system, although the cheapest and simplest to develop in our environment, is inadequate to meet many of our needs. First, such information is often lost because system log books are either poorly maintained or misplaced. Second, there is little discipline for maintaining the necessary information because of the wide range of users who access each facility or instrument. Finally, it is virtually impossible to track permissions for access to instruments or facilities, or to track inventories of supplies and parts. Thus, we desired a computer-based system that could record all of these tasks.

Following the recommendations of King [2], we have separated the wide range of functions that must be

accommodated into several stand-alone modules and have carefully considered the functionality that should be part of each component of our larger system and of each module of the LIMS. We have determined that it is essential to separate the operational aspects of the facilities from those associated with monitoring wafers. Analysis of our user needs revealed that we required a LIMS that could support tracking people, places, and things. It must also support purchasing and inventory functions. Finally, as an academic facility with both research and teaching needs, it must support the presence of students involved in formal laboratory courses. We explicitly eliminated gathering data regarding the wafers—this information will be acquired through our electronic wafer traveler program [3]. Furthermore, we relegated all cleanroom monitoring (particle counters, temperature, and humidity) to Facility Net, a commercial system from Particle Measuring Systems [4], and all toxic gas monitoring for our MOCVD and various etchers to a dedicated system provided by Zellweger Analytics. The LIMS database and the wafer traveler program are being integrated as each will use much of the same information concerning people and instruments. To date, we have found no reason to integrate data acquired with LIMS and that acquired with either Facility Net or the toxic gas system.

### 3. SYSTEM DESIGN

Our LIMS is implemented with distributed workstations and servers running Windows 2000 (Server and Pro) that are interconnected via the campus Ethernet. The relational DBMS is Microsoft SQL-2000. Although not required for correct operation of our LIMS, we have chosen to include all on-campus computers in a single domain that is managed by two dedicated domain controllers. Because the campus network does not support virtual private networks (VPNs), it is not possible to include our off-campus computers in this domain. However, we have no problems gaining full access to all functions of the LIMS from such off-campus computers. In addition to the domain controllers, we use three separate machines for data storage—one for the SQL database, one for the IIS WWW server, and one for a student FTP file server. The physical separation of these functions on different machines provides greater security and integrity of our data. The SQL database is mirrored on the WWW server to provide a layer of redundancy for this critical function. All five machines are backed up daily over the campus network to remote machines in our computing center with Tivoli. For additional security, we have created a CD ROM image of each machine using Ghost from Symantec [5]. We are thus able to recreate any failed computer rapidly from our server room in a remote location.

Although we considered developing a system in which user access to each data acquisition computer is controlled from the primary domain controller, we found that the

complexities of adding and deleting the rapidly changing number of authorized users using Active Directory was more time-consuming than our development schedule could allow at the present time. Therefore, for the time being we developed a process in which each laboratory has a dedicated login computer that runs a custom program called Lab Login where users identify themselves on entry into the facility and where they record their use of various tools and equipment. In addition, Lab Login checks a reservation system that allows remote scheduling of instruments and tools. This scheme allows us to manage all facility entrances and instrument accesses from data within the LIMS with no complex interaction with the Active Directory features of the Windows operating system.

With computers distributed in numerous laboratories and facilities in several buildings within the College, software maintenance and upgrades become a problem, both for our LIMS developers and administrators, as well as for our Systems Administrators. To ease their burden, we have installed Remote Administrator [6] on each of our computers. This allows much of the routine system administration to be performed from the administrator's workplace, including home computers.

It is critical that all of the servers and the login computers are operational at all times. Because of the dispersion of our facilities in three separate buildings within the College, it is impractical that such services be checked manually on a 24 × 7 basis. We have thus automated this process using Ping Monitor from JC Software [7]. This program pings each of our machines as well as some of our processes every fifteen minutes and, if there is no response within a set number of tries (usually 5), issues an e-mail of the specific failure to the relevant facility and system administrators via the University Listserv. Thus, the relevant personnel are informed of a system failure within 15 minutes on a 7 × 24 basis.

We now have approximately two years of experience with this network topology and have found it to be robust and to serve us well.

### 4. SOFTWARE DESIGN

The heart of our LIMS is a relational database built in SQL 2000. However, for ease of user access, it is important to have simple graphical user interfaces (GUI's) that hide the details and complexities of the DBMS. Furthermore, with students and faculty users scattered throughout the campus (and some off-campus users as well), we believed that it was important that the general user interface be implemented using a WWW browser rather than through a compiled Visual Basic (VB) program. Such a strategy would eliminate all of the headaches of tracking version numbers of DLLs and worries of corrupting the registration database of a user's computer. However, there were many administrative functions that we determined

Table 1: Information stored in the various tables of the relational database. Data are entered into and/or are read from the fields by both Active Server pages executed by WWW browser access or via two custom programs written in Visual Basic 6.0.

Entities Tracked	Typical Information Acquired
People	Name; Address; Phone; E-mail Address; UserId; Classification (faculty, student, TA); Academic Advisor; Lab and Instrument Approvals; Lab Access; Instrument and Facility Training; Task Assignments
Places	Facility Name; Facility Manager; Hours of Operation; Tools, Access Authorization; Instruments, Equipment and Supplies; User Access and Equipment Used; Cleaning Schedule and Tasks
Things	Instruments; Tools; User and Service Manuals; Equipment; Computers; Supplies; Locations; Authorization for Use; MSDSs; Toxicity and Hazard; Inventory; Property Tags; Repair Record; Status; Mentor; Master; Purchase Price; Warranty Dates
Purchasing	Vendor; Address; Contact; PO Number; Items
Courses	Instructor; TAs; Course and Lab Section; Semester Offered; Students; Team Members; Lab and Equipment Use

would be much more efficiently implemented with VB code. Such programs, written in VB 6.0, would be installed only on computers in the microelectronics domain, or on computers of faculty and staff closely related to the center where our system administrator would have easy access to assist with installation and operational problems when they arose. We describe each of these aspects of our LIMS in the following paragraphs.

**4.1:- RDBMS Design** The database has been developed in third normal form through the use of over 50 tables [8, 9]. The various tables are set up on the SQL 2000 Server [10]. Typical information acquired is outlined in Table 1. The structure of the tables is always available through entity—relationship (E-R) diagrams [8] which may be created automatically by SQL 2000. In addition, the fields (and their formats) for the tables are available through the web browser.

The systems programmer uses the main SQL 2000 access program for program development such as adding/modifying tables, fields and stored procedures. However, we have developed the system around the philosophy that no user should ever access any information in this manner. Thus, we have created a two-tiered graphical user interface (GUI) for controlling the flow of data into the system and the sorting and display of data emanating from the system. This GUI involves data access by both a web browser and by custom programs written in Visual Basic 6.0.

**4.2:- Browser Design** With several years of experience, we have found that it is essential from a software maintenance point of view that all non-administrative

access to the database should be through a standard WWW browser. This decision eliminates the problems associated with installing an executable program on widely varying configurations of PCs. In order to simplify the web page coding, we have developed most of the data access pages using active server pages (ASP) and SQL 2000 stored procedures. We have found that the development of stored procedures has been critical to the ease of managing and maintaining the system. In addition, we have made extensive use of JavaScript, VBScript, Cascading Style Sheets (CSS2), and the document object model (DOM) in programming our web pages. Here, we have found inconsistencies between the implementation of CCS2 by IE and Netscape to be such that, at least for now, our code will execute correctly only on IE 5.5 and above.

In order to protect user-proprietary information, we believe it is essential to store public information about our facilities and the LIMS data on two completely separate WWW sites. For improved security these two web sites are maintained on two separate machines, each running IIS Server.

When the user first accesses the LIMS home page, she is given the choice of either joining MicrON<sup>1</sup> or logging in to the password-protected web site. On logging in, she is presented with a list of operational functions that may be performed as shown in Table 2. This list is customized for each user based on user classification and permissions (see next section for details). We provide access to all information required by any class of user and user

<sup>1</sup> Virginia Tech's Center for Microelectronics, Optoelectronics and Nanotechnology

**Table 2:** Representative information available via a web browser. Access to each item is controlled by user-specific permissions.

User Profile	Advisees	Who Reports to Whom	Authorization Inconsistencies	Cleanroom Tasks
Instruments and Tools	People	All the Types of Users	Courses I Teach	Task Assignments
Facilities	Mail	Can Use Instrument	Lab Fees	
Schedule Instrument	Add New Instrument	Can Use Facility	Database Tables	
Cleaning Teams	Instruments You Mentor	Is in Facility	Facility Contents	
Cleaning Team Administration	Instruments You Master	Lock Functions	Create Cleaning Tasks	

permission via the WWW page. No access via the VB code described in the next section is required for access to and operation of the facilities and instruments.

Most of the functions listed in Table 2 are self explanatory. However, a few warrant mention. The “Is in Facility” is particularly important. The Lab Login program described in the next section records every entrance and exit from each of our facilities. These data are not stored locally, but are stored in the central database. This function allows faculty and facility administrators to track all personnel who are present in the facilities at any time from anywhere on the internet.

The “Mail” function allows users with mail privileges to send e-mail to users selected from different categories: facility users, instrument users; user classification; individually selected users, etc. The mail system is one-way—it can only send mail and has no capabilities for receiving mail. We find this feature critical for informing selected member users about facility operations and instrument status that are of direct concern to them without inundating other users with “spam” messages that are irrelevant to them.

The “Authorization Inconsistencies” function checks to determine if users have been granted access to instruments but have not been granted appropriate facility access. The inverse capability of determining if users have access to facilities for which they have no instrument access will be added soon. We have found that this information is critical for system maintenance.

The users particularly like the “Schedule Instrument” feature. Here, a user is allowed to reserve an instrument or tool for future use. Of course, one can reserve only those instruments for which one has been trained and has received permission to use.

We have included a complete task scheduling system in which approved users may assign tasks to other users. These tasks are logged and sorted by due date, priority, and task assignee. The system checks the task log each evening and sends e-mail to the assignees informing them of any new tasks. On completion, the assignee makes any necessary comments and marks the assignment as “Completed.” The system checks for overdue tasks weekly and sends the assignee gentle reminders of outstanding work. Although originally developed as a simple methodology for the authors to keep track of the bugs and features of the LIMS system, we have found that this is an invaluable method of tracking the wide range of tasks that must be accomplished by our students, faculty and support staff to keep our Center operating.

A particular advantage of this GUI is that data are available to users and to administrators from anywhere on the internet without installation of proprietary code. One of us (RWH) has found this particularly valuable while on travel status when system functions were accessed from an internet café.

**4.3:- Visual Basic Design** There are many operations that are not easy to implement in ASP. After considerable thought and experimentation, we determined that it was wiser to develop such functions in custom code. We selected VB 6.0 for these programs because of its excellent IDE and interface with SQL 2000. For the system described here, we have found no need to use the more complex VB.NET.

We have developed two separate applications: the main LIMS program and the Lab Login program. Each is described below.

The LIMS program provides administrators and facility managers with capabilities for managing Users, Facilities,

Instruments, Computers, Classes, Inventory, Inventory, and System Administration functions. Each of the operations has been developed in a separate module and is stored as an OCX program. Most data entry operations that fill the database tables are performed through this LIMS program. Of course, LIMS uses many of the stored programs that are also used by the browser access version, thus leading to some duplication of function. We have found that this duplication is, in fact, rather nice because it allows the experienced operator to use the most convenient method of data access and entry for the job at hand.

We have developed access functions that allow the system administrator to grant every user in the database Read, Write, or Read/Write access to each module. The default access is "None." In addition, the access functions may also be used to grant permissions for some of the browser-only features such as mail, authorization inconsistencies and database table structures.

The main screen for LIMS (following the splash screen and login) is a series of buttons that take the user directly to the module of interest. The security system "gray's out" all those functions for which the user does not have permission to access. This program is distributed only to those faculty and administrators who need R or R/W access to the data in the DBMS. With this tight control, problems associated with module installation problems are confined for the most part to computers under our direct control.

We have found this strict control of data access to be extremely valuable in working with a wide range of support personnel who assist us with inventory control, computer systems administration, and purchasing. We are able to grant such personnel access to all the features they need to perform their work, but do not provide access to those features for which they have no administrative responsibility.

The Lab Login program is installed on computers located at the entrance to every laboratory and storage facility. This program verifies that a user has authorization for access to the facility and that she is permitted to use instruments and tools located therein. On login, the program presents the user with a list of tools and instruments she is authorized to use and checks the instrument reservation system to verify that the requested instrument is available. Algorithms have been designed to handle no-shows for reservations, and to resolve conflicts of overlapping use requests. On logout, the program records the instrument use time. Provision has been made to allow a user to start a long run (e.g., a long diffusion or oxidation run), leave the lab for a break, and maintain control of the instrument.

We operate our laboratory with experienced graduate students (called instrument masters) who have

responsibility for training new students in the proper use of each tool. Each instrument also has a "faculty mentor", a faculty member who is experienced with the tool who can assist the master with more difficult problems. The masters and mentors have access to functions that allow them to view all user accesses to both the facilities and to the tools for which they have responsibility. Such data are critical in sharing the expenses of operating the facilities and repairing instruments.

Lab Login has been coded to respond to a ping on a port selected by the Systems Administrator. Thus, as noted in Section 3, we are able to verify that Lab Login is always available on the login computers. Due to power failures and the usual inexplicable system restarts, our administrative staff find it very useful to know when they must manually restart Lab Login. We are finding that by putting Remote Administrator on auto start, we are usually able to fix problems with Lab Login remotely.

Finally, we have developed a "Smart Updater" that makes it very simple to update any of our VB modules over the internet. This program, when run on any computer on which our programs have been installed, checks the version number of each VB module on that machine versus the current version numbers of the latest modules stored on a distribution computer. If there are version discrepancies, the newer modules can be downloaded and installed automatically. Such updates do not require that the OS be restarted. We have found that it is easy to update all of the Lab Login computers via Remote Administrator.

## 5. DISCUSSION

The LIMS described in this paper has been under development for approximately two years. During that time, the faculty members have changed the policies and procedures by which we operate our Center to meet their evolving needs. A major feature of our development is the use of pull-down menus for defining new entities such as facilities, instruments, users, etc. We have taken great care to avoid hard coding variables into the programs. Thus, it is easy to add, change, or delete such entities at any time. Although it took time to develop the methodology by which such code was written, the effort has more than paid for itself as we changed the system to track the changing needs of our faculty and staff. This feature also makes it very easy for this extensive and complex system to be ported to other microelectronics facilities, or, for that matter, to other university research laboratories in general.

We are indebted to Andrew Duncan and John Harris for Systems Administration support for the over 30 computers installed in the Center. We are indebted to the ECE and MSE Departments for their financial support of the development of the LIMS. Finally, we wish to thank Bryan Dickerson, Charlie French, Brad Heath, Bill Houck, and

David Gray for their continued interest in improving the LIMS and for their comments, notes and observations when our code sometimes goes awry.

## 6. REFERENCES

- [1] P. D. Eckerman, and R. W. Hendricks, "*A Laboratory Information Management System (LIMS) for an Undergraduate Microchip Fabrication Facility*," Proc. 2001 ASEE Annual Conference, Albuquerque, NM, (2001), available from ASEE, Washington DC on CD ROM.
- [2] J. G. King, "*The Design of Cleanrooms for the Microelectronics Industry*," Ch 3 in Cleanroom Design (2E), W. Whyte, ed., John Wiley & Sons: Chichester, UK (1999), pp 51–78.
- [3] A. Shiekh, S. D. Sheetz, and R. W. Hendricks, "*An Electronic Wafer Traveler System for an Academic Microchip Fabrication Facility*," (this conference)
- [4] <http://www.pmeasuring.com/particle/products/software/docs/facilityNet>
- [5] <http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=3>
- [6] <http://www.radmin.com>
- [7] <http://www.jcsoftware.net/pingmoni.htm>
- [8] P. Robb and C. Coronel, Database Systems: Design, Implementation, and Management (5E), Course Technology: Boston, MA, (2002)
- [9] R. M. Riordan, "*Designing Relational Database Systems*," Microsoft Press: Redmond, WA, (1999)
- [10] R. M. Riordan, "*Microsoft SQL Server 2000 Programming*," Microsoft Press: Redmond, WA, (2001)