SWORD, an Information Server for Juvenile Justice

Daniel Bond[1] [Andrew Strelzoff[2], Tim Rehner[3], Ray Seyfarth [4]]

Abstract – Historically the juvenile justice system in southern Mississippi kept records on cards which had to be indexed, filed and then retrieved to move children through the system. University of Southern Mississippi's SWORD system is a client-server database which was developed to improve the quality and reliability of child information. SWORD currently serves six southern Mississippi counties with 16 client sites and 2 servers with information on over 8,000 juveniles cataloged in the database. The system provides a way to store, retrieve, and modify information about clients for the detention center, juvenile court, and youth counselors.

This paper outlines the SWORD architecture and discusses the technological parameters and practical considerations which contributed to its design. Also discussed are major challenges to further development of SWORD including legacy code and equipment, varying agency requirements, improved research data access and interface development with other state and federal level data banks.

Keywords: database, Juvenile Justice, Software Architecture, network, Visual Basic.

INTRODUCTION

The project was named for the "SWORD of Justice" and is used to track juvenile offenders through the juvenile justice system. SWORD provides key database functions to the detention center and youth court for eight counties in southern Mississippi. Currently, the system maintains juvenile data for approximately 8,000 juveniles including personal contact, offense, detention, and court information. Prior to the implementation of SWORD, juvenile data was tracked using paper files which posed a logistical problem for court officers. Duplicate records were often kept between the youth court and the detention center resulting in data redundancy. It was virtually impossible to provide data consistency using the manual method of record keeping. With the implementation of SWORD, data is consistent and can be used in a concurrent manner. The ability to easily access and modify juvenile data in real time generates more current and reliable data.

The system was conceptualized in 1990 by Doctors Tim Rehner and Mike Forster at the University of Southern Mississippi as an information management system designed to aid in the prevention of juvenile delinquency. Advocated by Forrest County Judge Micheal W. McVail, SWORD is funded by the Juvenile Accountability Incentive Block Grant from the City of Hattiesburg (located in Forrest County) and is coordinated by the Family Network Partnership. The current computer science faculty investigators are Doctors Ray Seyfarth and Andrew Strelzoff who provide guidance and problem solving for the graduate student developers. The goals of the project are to provide a common database for Forrest County, promote the interdepartmental (court, detention, counseling) sharing information about juvenile offenders, provide easy access for authorized users to juvenile information, automate report generation for the youth court and detention center, and improve the ability of

[1] Graduate Student University of Southern Mississippi, 42 Bridgefield Turn Hattiesburg, MS 39402, djaybond@gmail.com

[2] Assistant Professor, University of Southern Mississippi, College of Science and Technology, School of Computing, 118 College Drive Hattiesburg MS 39406-0001 strelz@orca.st.usm.edu

[3] Professor University of Southern Mississippi, College of Health, School of Social Work, 118 College Drive Hattiesburg MS 39406-0001 *tim.rehner@usm.edu*

[4] Associate Professor, University of Southern Mississippi, College of Science and Technology, School of Computing, 118 College Drive Hattiesburg MS 39406-0001 ray.seyfarth@gmail.com

Forrest County to provide juvenile services. With the expansion of the project to include other counties, the goals have been expanded to include inter-county information sharing.

Evolution

SWORD 1.0 consisted of a client-server architecture using Active Server Pages (ASP) and Microsoft Access database. ASP is a Microsoft product that functions inside Internet Information Services (IIS). When a browser requests an ASP file, IIS passes the request to the ASP engine. The ASP engine parses the ASP file, line by line, and executes the scripts in the file and then finally returns to the browser the plain HTML, thereby eliminating the need to have a client application. The user's web browser became the client application. This thin client structure provided ample function for a simple query and view model but was limited when more complex behaviors were needed because after the ASP engine delivers the HTML, there is no control by the sever over the rendered page. Although the ASP approach had many advantages including the use of HTML for submission of queries and viewing of results, the system response was slow due to the server side execution of the scripts and the need to transfer more data via the network under normal operating conditions.

The current structure of the system is a client-server database supporting multiple clients. As noted in figures 2 and 3, the client machines use Microsoft Windows XP/2000 Operating System (OS) while the server uses Linux OS (Ubuntu). The client software ran on the Windows OS, is a Visual Basic (VB) .NET application that uses the .NET framework 2.0 [4]. The .NET framework is a collection of pre-coded classes that manages the execution of programs developed using the framework. The server consists of a Linux OS, server interface code (SIC), and a MySQL 5.0 database. It is important to note the server OS and database are freely available and are open-source software.

The need to integrate functions that could not be supported by ASP and the desire to change the server OS to a Linux derivative, motivated the revision of the structure of SWORD. Although it is still a client-server system, the second version (2.0) had two major revisions implemented. First, the thin client structure was replaced with a Visual Basic .NET (VB.NET) client. The addition of the VB client provided the ability to perform "client side" functions and reduce server work load. Since all of the client machines use Windows OS and the .NET framework and the client executable code are freely redistributable, the VB.NET client did not increase the overall project cost. Secondly, the OS on the server was changed to Linux thereby reducing the number of Windows licenses. The change in server OS and replacement of ASP did create the need for additional sever communication programming as shown in Figure 1 but an overall reduction in server work load was realized.

The SIC is written in C language and acts as communications control and database interface. Via sockets, the VB client makes a request for information from the database using the SIC. For an example, when the client needs the personal information for a specific child the client sends a "getpersonalinfo" request and with the associated youth identification number. The SIC receives the request, constructs the appropriate query for the database, executes the query against the database using open data base connectivity (ODBC), and returns the selected data. The SIC includes the type request in the returned data so the client can confirm the response prior to using the data. After confirmation, the client can then populate the appropriate forms and display the results.

Current Structure

Due to the incremental development path the project has taken, SWORD 2.0 currently includes two different client programs and databases, one for the youth court and the other for the detention center. Ideally, a single client program and database for both departments should have been provided. This was not the case due to the need to deploy a system before completing both systems (ideally one system). Once the first system was functional and deployed, attention shifted to completing the second system while maintaining the first. This caused the evolution of two systems. A benefit of having two systems is there is an obvious separation of functions allowing each to be customized to provide specific departmental functions. For example, the detention center has no need for court documents, therefore this feature is omitted. While the youth court may need to see which children are detained, therefore this function is included. An effort to correct this deficiency and "merge" the systems is currently underway.

Figures 2 and 3 illustrate the overall network topology used to implement SWORD. In the first configuration shown in figure 2, the database server is located on the local network. In this instance since all equipment is secure by controlled access and the network is protected by an appropriate firewall, the internal network communications is unencrypted. All external communications is encrypted using OpenVPN [1]. This is in contrast for the topology shown in figure 3. In this configuration all client machines communicate with a remote server via the Internet and all network communication with the server is encrypted. OpenVPN is open source software that provides a virtual private network (VPN) using secure socket layer (SSL).

As noted in Figures 2 and 3, multiple instances of client software are located at various locations around the region. The logistics of maintaining current copies of client software became a problem. Since the developers on the project are students, they were not always able to respond in a timely manner when distributing updates. This problem was solved using Unison [2]. Unison is file synchronization software that runs on both UNIX and Windows. It allows the developers to propagate changes to multiple clients without the need physically visit the machine. When the SWORD client software is started, Unison determines if a newer version exists on the project server. If it does, the new version is copied to the client machine and executed. Otherwise, the current copy on the client is executed.

The issue of consistent data backup exists in any database system. It is imperative for the system to provide timely backups of all data to guarantee recovery from failures. In the SWORD data backup is accomplished using a combination of MySQL commands and Linux scripts. MySQL database provides a database "dump" function (MYSQLDUMP) that can accessed from the OS command line [3]. The dump creates a complete copy of the database and can be equated to a instantaneous "snap shot" of the database that includes the database schema and all data contained in the database tables. The dump file is formatted to allow easy restoration of the database by allowing the dump file to be executed against the database. See the appendix for a simple example of the format of a MySQL dump.

Back up of systems that have remote servers (remote from the University) is completed using a scheduled database dump. The dump is scheduled by using the Linux "cron" system that allows execution of programs based upon time [5]. The database dump file is then copied to the server located at the University development lab via the Internet using secure file copy (SCP). For systems that use the server located at the University, the dump is made locally. The data is then copied to a second server located at the University using Dirvish [6]. Dirvish is a disk based virtual image network backup system and allows the complete back up of the development code, the deployed code, and the relevant data. Backup is completed once a week with the complete state of the SWORD project is copied to long term storage such as CD or DVD. This approach prevents the loss of more than a single day's worth of data in the event of a server failure.

EXTENSIONS

As is common with many programs, SWORD is under constant development. The need to improve current features and introduce new ones keeps the project in a constant state of flux. Currently four aspects are being developed.

Biometrics: The need to accurately identify juveniles entering the detention center has created a need to include the ability to scan finger prints. An extension is currently underway to provide digitally stored finger prints and confirm juvenile identities by searching the database. This feature is also under consideration for integration with the Hattiesburg Police Department and other regional police agencies.

Photograph Documentation: One recent extension to SWORD included adding the capability to store a child's photograph in the database and use it for identification during in processing at the detention center. A caveat to this extension is the use of photographs to document identifying marks on the child. With the nationwide increase in gang activity the desire to accurately document gang marks and affiliations has moved to the forefront. This extension will aid the Juvenile Justice System in tracking gang members and affiliations.

System Merging: As noted above, SWORD currently consists of two separate systems each consisting of a custom VB client, SIC, and database. Merging of the two clients into a single system is currently underway. Many times during maintenance the same changes are being made to both systems. By merging the systems into a single system a reduction of labor should be realized.

Statistical Capabilities: Efforts to expand SWORD to provide increased statistical information is underway. Currently, SWORD provides all required reports for the youth court and the detention center. These reports provide some demographic and statistical information concerning the youths but only as it applies to the respective departments. An effort to expand the statistical capabilities is underway which will provide more statistical information to the Department of Human Services and other state agencies.

Conclusion

SWORD provides key database functions to the detention center and youth court for eight counties in southern Mississippi. Currently, the system maintains juvenile data for approximately 8,000 juveniles which includes personal contact, offense, detention, and court information. Increased demand for accurate and timely data about juvenile offenders has motivated the development of SWORD and continues to drive it's evolution.

References

[1] Author Unknown, "OpenVPN - An Open Source SSL VPN Solution by James Yonan," retrieved November 10, 2006 from http://openvpn.net/.

[2] Author Unknown, "Unison File Synchronizer," retrieved November 4, 2006 from http://www.cis.upenn.edu/~bcpierce/unison/index.html.

[3] Author Unknown, "Database Backups," retrieved November 4, 2006 from http://dev.mysql.com/doc/refman/5.0/en/backup.html

[4] Author Unknown, ".Net Development Framework Center," retrieved November 4, 2006 from http://msdn2.microsoft.com/en-us/netframework/ default.aspx

[5] Author Unknown, "Using Cron," retrieved November 4, 2006 from http://www.scrounge.org/linux/cron.html

[6] Author Unknown, "Welcome to Dirvish," retrieved November 4, 2006 from http://www.dirvish.org/

[7] Mandeville, Meghan, "SWORD: A Useful Weapon in Juvenile Information Management," 2003 retrieved August 1, 2006 from http://www.corrections.com/news/article.aspx?articleid=516

[8] Middaugh,Kim "SWORD of Justice," Government Technology, 2004 retrieved August 1, 2006 from http://www.govtech.net/magazine/sup_story.php?id=90807&magid=8&issue=7:2004

[9] Rehner, Tim and Forster, Micheal, et al. "An Integrated 'Between-Agency' Information Sharing System for Juvenile Offenders: SWORD," proceedings of the 31st Annual Alabama/Mississippi Social Work Education Conference, 2002

[10] Rehner, Tim and Forster, Micheal, et al. "SWORD: An Integrated Web Access Juvenile Delinquency Database," proceedings of the 30th Annual Alabama/Mississippi Social Work Education Conference, 2001

FIGURES

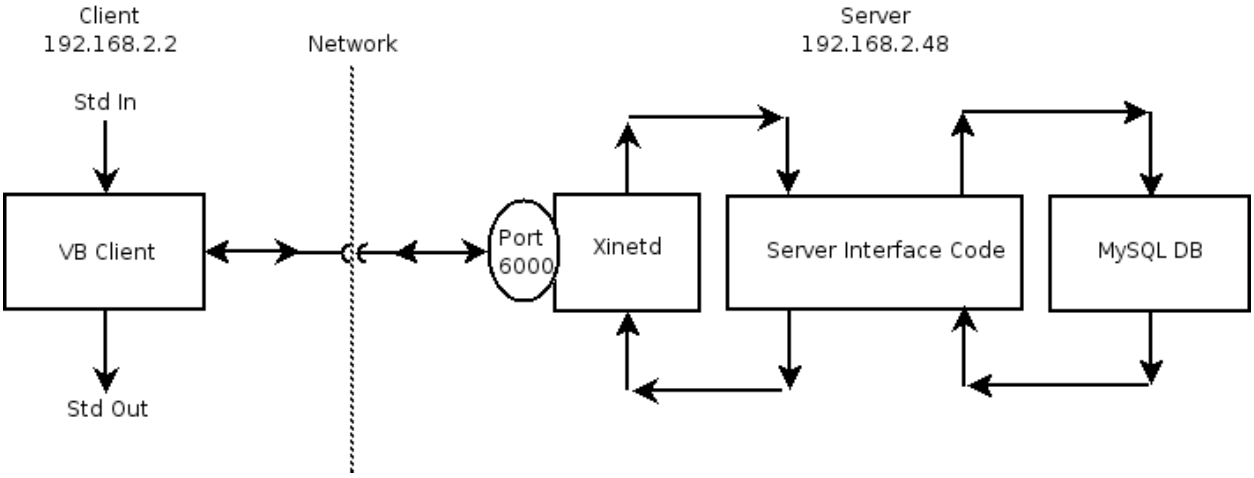

Figure 1 - Communication Path between Client and Database.

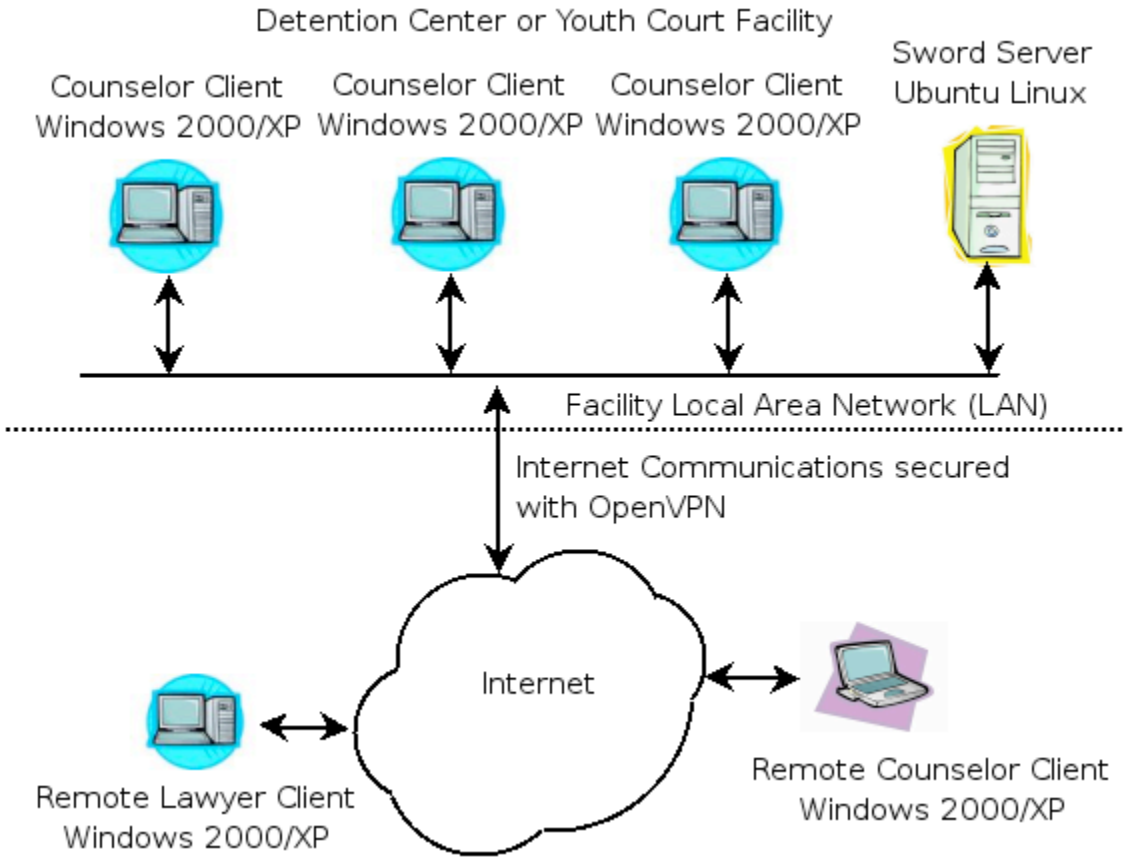

Figure 2 - Network Topology with Local Server.

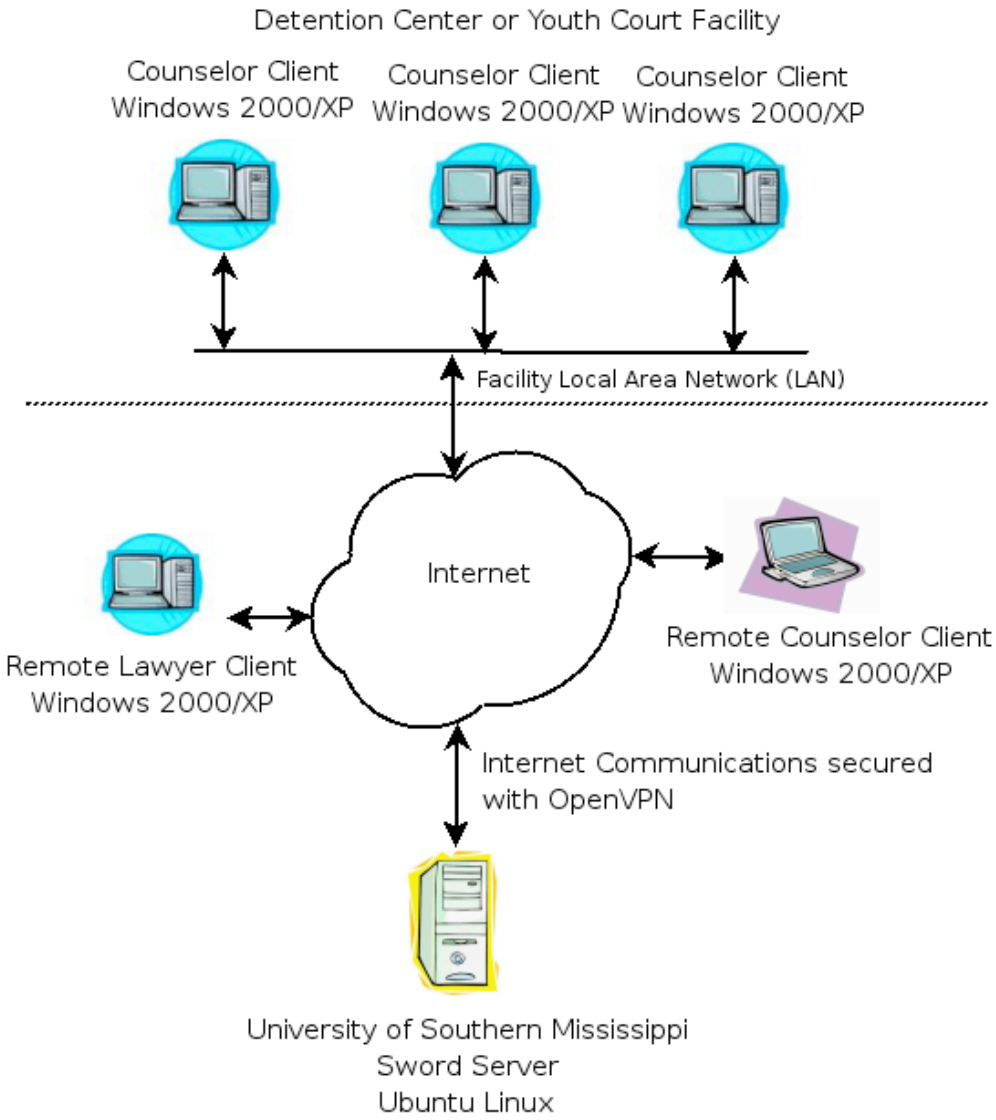

Figure 3 - Network Topology with Remote Server.

APPENDIX

This dump was made on a database called "test" which contains a single table called "customers."

-- MySQL dump 10.10

--

-- Host: localhost Database: test

-- --

-- Server version 5.0.17-max

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;

```
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
--
-- Table structure for table `customers`
--
DROP TABLE IF EXISTS `customers`;
CREATE TABLE `customers` (
  `c_id` int(11) NOT NULL auto_increment,
  `c_company` char(60) NOT NULL,
  `c_name` char(60) NOT NULL,
  `c_street` char(60) default NULL,
  `c_state` char(2) default NULL,
  `c_zip` char(9) default NULL,
  PRIMARY KEY  (`c_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
--
-- Dumping data for table `customers`
--
/*!40000 ALTER TABLE `customers` DISABLE KEYS */;
LOCK TABLES `customers` WRITE;
INSERT INTO `customers` VALUES (1,'c_company1','c_name1','c_street1','c1','c_zip1'),
(2,'c_company2','c_name2','c_street2','c2','c_zip2'),(3,'c_company3','c_name3','c_street3','c3','c_zip3'),
(4,'c_company4','c_name4','c_street4','c4','c_zip4'),(5,'c_company5','c_name5','c_street5','c5','c_zip5');
UNLOCK TABLES;
/*!40000 ALTER TABLE `customers` ENABLE KEYS */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

Daniel Bond
Daniel Bond is a graduate student at University of Southern Mississippi and is a member of the SWORD development team. He is a student member of IEEE and ACM. His current interests include database and information retrieval systems.

Andrew Strelzoff PhD

Andrew Strelzoff received his Ph.D. in Computer Science, University of California, Santa Barbara in 2004. He is an Assistant Professor, Computer Science, University of Southern Mississippi. Areas of Research: FPGA computing, Radio Frequency Identification (RFID)

Tim Rehner PhD

Tim Rehner is currently assistant director of the School of Social Work at the University of Southern Mississippi. He received his PhD University of Alabama, Tuscaloosa, Alabama in 1994. He is the Co-Founder & Director, Family Network Partnership - a university affiliated community based delinquency prevention agency.

Ray Seyfarth PhD

Ray Seyfarth is currently an Associate Professor of Computer Science in the School of Computing at the University of Southern Mississippi. Prior to his current appointment, he completed a Ph.D. in Computer Science at the University of Florida in 1989. His teaching interests include UNIX, algorithms, graphics, compilers, programming languages, operating systems, formal languages and databases. His research has spanned many disciplines through collaboration with polymer scientists, physicists, oceanographers, biologists and mathematicians. His current funding is in juvenile justice software and image mosaicking.