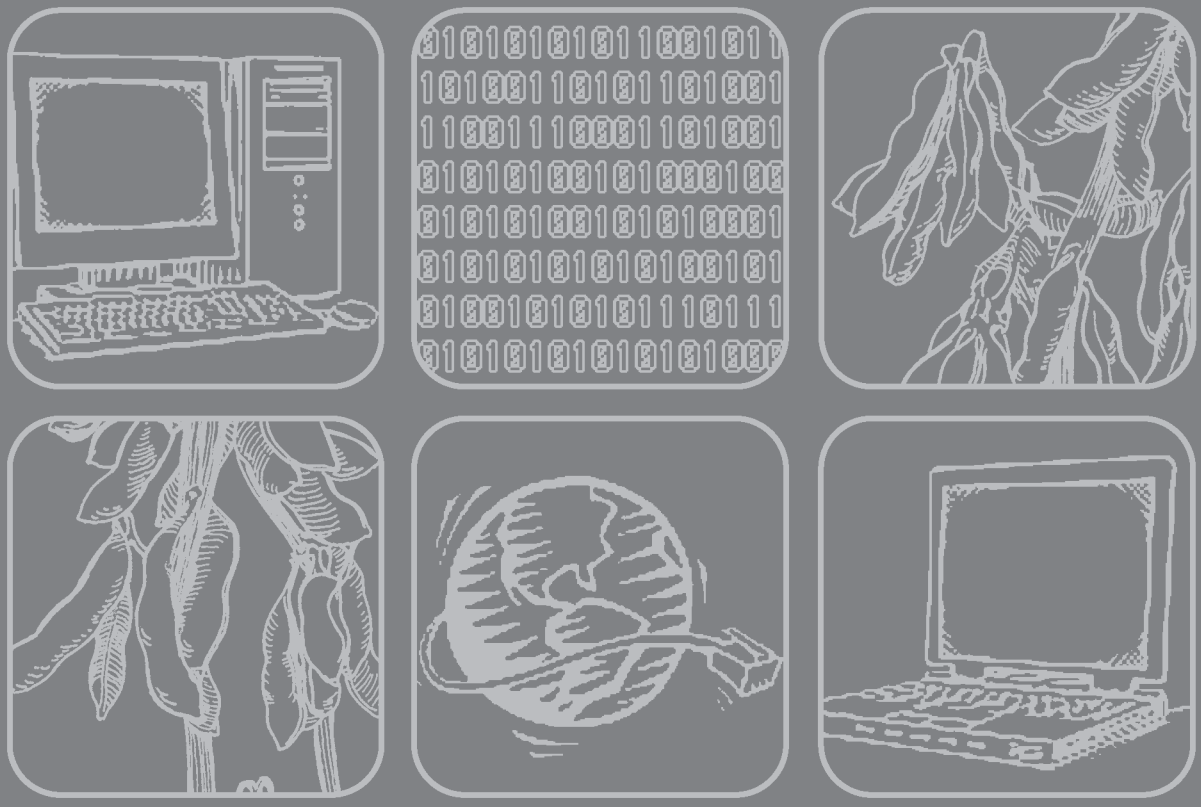
BULLETIN 1131 MAY 2003

Structure and Technical Compositions of an Internet-Based Soybean Database



Experiment Station
Vance H. Watson, Director

Mississippi Agricultural & Forestry Experiment Station
J. Charles Lee, President • Mississippi State University • Vance H. Watson, Interim Vice President

Structure and Technical Compositions of an Internet-Based Soybean Database

Lingxiao Zhang
Associate Research Professor
Delta Research and Extension Center

Wanwen Qi
Microsoft Corporation
(Formerly of the MSU Department of Computer Science)

Special thanks go to Dr. S. Bridge at the Computer Science Department at Mississippi State University for her technical corrections and comments on this manuscript. This project is fully supported by the Mississippi Soybean Promotion Board. For more information, contact Dr. Zhang by telephone at (662) 686-3215 or by e-mail at lzhang@drec.msstate.edu. This bulletin was published by the Office of Agricultural Communications, a unit of the Division of Agriculture, Forestry, and Veterinary Medicine at Mississippi State University.

Abbreviations: API = Application Program Interface; CGI = Common Gateway Interface; CPU = Central Processing Unit; DAO = Database Access Object; DBMS = Database Management System; DSL = Digital Subscriber Line; JDBC = Java Database Connectivity; LAN = Local Network; HTML = Hypertext Markup Language; ODBC = Open Database Connectivity; OS = Operation System; PC = Personal Computer; SQL = Standard Query Language; OVT = Official Variety Trial; PHP = Hypertext Preprocessor.

Structure and Technical Compositions of an Internet-Based Soybean Database

Abstract

An Internet-based database, Deltasoy, has been built to assist soybean producers in better use of information from official Mississippi soybean variety trials. This bulletin discusses related background information and presents a description of technical aspects of database construction, including issues of software, hardware, database design, and programming. Issues regarding administrative tools and database maintenance, such as input and output formats, are also discussed. This detailed technical information will be valuable for people who are interested in building other new databases for similar purposes. It will also help users to effectively utilize Deltasoy.

Introduction

Deltasoy is an Internet-based database system used to assist soybean growers in obtaining information from official variety trials more efficiently. The initial work of building Deltasoy started in August 1998, and the framework of the database was completed in August 1999. Deltasoy has been available on the Internet since October 1999. It has been modified and new functions have been added continuously. A previous paper discussed agronomic aspects of the applications of this database (Zhang et al., 2002). However, a detailed description of the structure and functions of the database were not discussed in that paper. This information is important to people who are interested in an in-depth understanding of the database. The objective of this bulletin is to provide technical details about the structures, functions, programming, and maintenance procedures of Deltasoy to serve as a model system for other parties interested in building a system for similar purposes.

Background Information

A Web database is a collection of data accessible via a query language or an Application Programming Interface (API) (Hobuss, 1998). A Web-based database system includes two components — the Web interface and the database. A Web interface includes two distinguishing parts — hardware and software. The hardware consists of a server machine, network facilities, power, and a database server. The software includes the system architecture, the operating system, Web server, interface design, database managing system (DBMS), and development tools. Hardware and software can communicate with each other through a Common Gateway Interface (CGI). These three parts are connected in a systematic way that forms a unit to function as an Internet database. Figure1 shows a typical three-tier system, which will be discussed further in section 4 (system architecture). In the workflow of a typical Web-database session, these are the main components that affect the developmental cost and the performance of a Web database system.

Key words: soybean, variety trial, database, Internet, information system

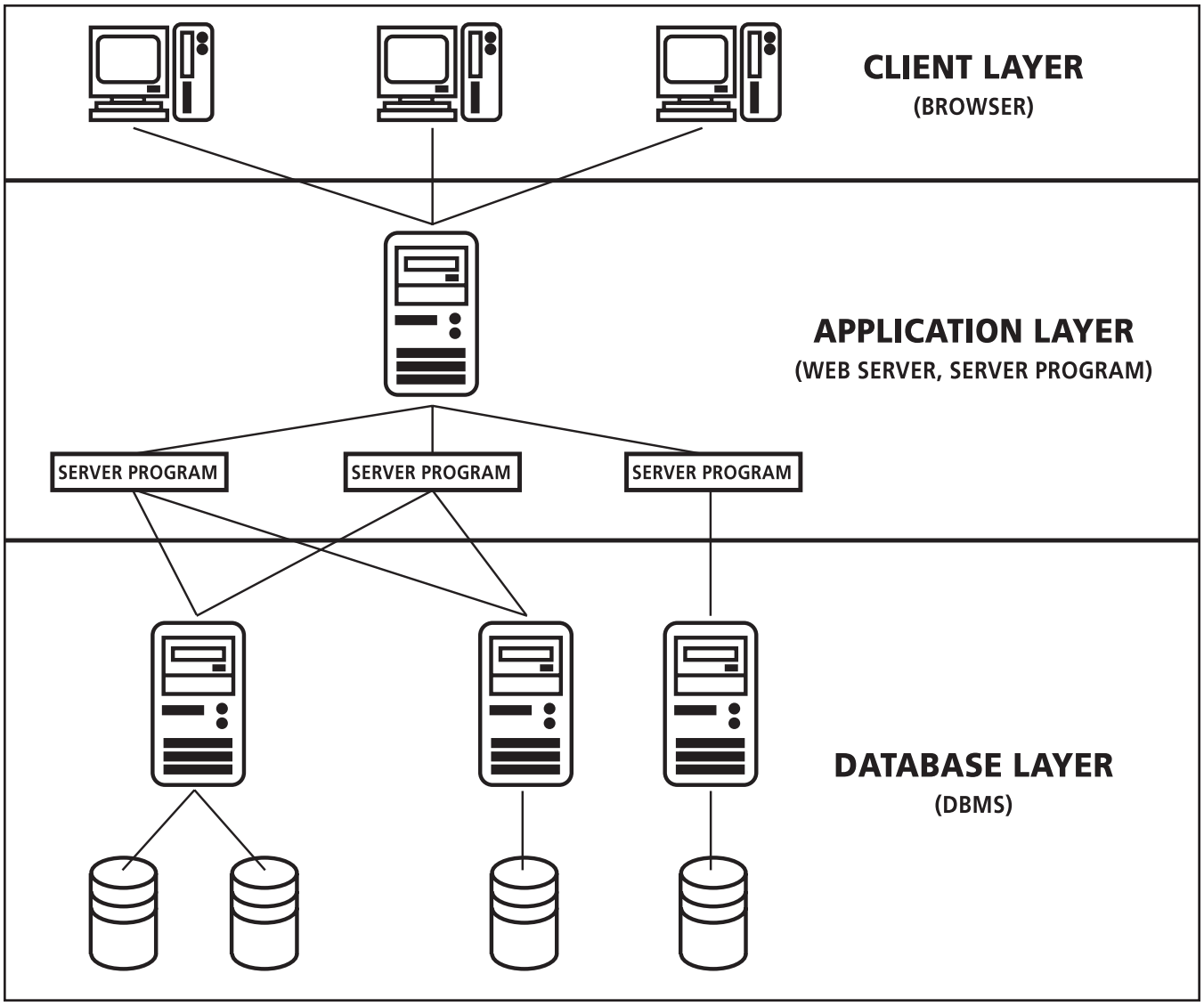


Figure 1. Diagram of the three-tier architecture of a Web database.

Hardware

Web Server — A Web site can be established by storing all Hypertext Markup Language (HTML) files on space allocated by service providers. This strategy works very well when the Web site is not enormously complicated and no interaction occurs between existing files. It is also very convenient and low cost (no computer is needed). Most Web sites are built this way. However, if flexibility is the first priority and files need to be updated frequently, such as in a Web database, an independent computer should be devoted as the Web site server. Storing the whole database on a server elsewhere would not be convenient to manage. A personal computer usually is capable of hosting a Web database, especially when it is not expected to have an unusually large number of accesses in the same time period. Otherwise, a workstation should be considered.

Database Server — The database server is used to store and run the database management system. A cost-effective way to build a Web database is to use the Web server as the database server. The advantage of sharing the Web server with the database server is that the costs of hardware and communication are both lower than they would be if separated. However, the disadvantage is a decrease in speed. Therefore, if high-volume access is expected, a set of multiple central processing units (CPUs), or clustered workstations, with high-speed connections should be used.

Network Facility — The Web server must be connected to the Internet, which can be achieved by telephone line modem, cable modem, digital subscriber line (DSL), fiber optical line, satellite, etc. The server may connect to the Internet directly or via Local Networks (LAN). The costs of these methods vary with bandwidth requirements. With high-volume access, a high-speed, permanent Internet connection is required. A Web server usually has been assigned to a simple or meaningful domain name so that a user can access the site easily.

Backup Device — A database system needs a backup device for file safety, data consistency, and preservation of data in case of natural disasters and human mistakes. The most common backup devices are hard disks and tapes. There are several ways to back up a Web site. One way is to run two of the same types of machines with the same software at the same time but at different places. All the transactions will be committed to both units. Failure from one of them will not cause the break down of the whole system. This way, the system provides high availability; however, it costs more and it may also increase processing time for each transaction. The alternative is to copy the data or make a log on another device, such as a tape, a spare hard disk, etc. If the designer selects this method for the system, a spare server is needed. When a failure occurs, the spare server will be used and the most recent valid state of the Web server and database will be restored on the backup machine. With this method, it may cause the Web site to be off-line temporarily. However, restoration is done incrementally while a backup is being made; therefore, the off-line time for the Web site should be limited.

Software Aspects

Operating System and Server Software — These two aspects are also related to hardware. If a Web server machine is built with a personal computer (PC), either Windows or Linux may be chosen as its operating system (OS). If it is built with a workstation, then a Unix system, such as Sun Solaris, may be used as the OS. Before the computer can be used as a Web server, some server software must be installed. The Windows OS can be divided into two classes according to functionality: server and client. The server class includes the Windows NT Server and Windows 2000 Server. The client class includes Windows 95, Windows 98, Windows NT Workstation, and Windows 2000 Professional. The choice of OS is dependent on the hardware being used.

The server program is the kernel of the middle layer in a three-tier structure system (Figure 1). Two main issues involved in the design of the server program are the kind of database interface to be used and the way that the client will interact with the server program. The interface between the server program and the database depends on the language that the server program uses. Furthermore, if a native database interface is used, then the database management system used will also decide what kind of native interface will be used. The only advantage that a native interface has over the standard interface, such as Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC), is the speed. However, those standard interfaces can achieve the independence of a DBMS, which means the server program can still work if the DBMS has been changed for any reason.

Interface Software Development Kit — For the Web page design, some HTML files need to be created first. To make the Web site more dynamic and attractive, CGI, or an integrated Java applet, and JavaScript with HTML are often used.

Database Management System — The database management system (DBMS) is used to store and manage all the data. It also provides an interface for the user to access the data. There are different kinds of DBMS available for use. The cost of a DBMS ranges from free for small applications to thousands of dollars for a large data center. Most database systems are standard query language (SQL) databases, such as Microsoft Access, Oracle, Sybase, IBM DB2, SQL server, and so on.

Development Tools — Many tools, such as Netscape Communicator, Microsoft FrontPage, and Adobe PageMill, can be used for Web page development.

CGI Programming

There are many protocols and standards that are used over the Internet. CGI, Java Servlet, and Java Applet are among those used most frequently. Each protocol has its own specific functionality and capacity. They all can be used to control the interaction between the Web browser and the server program. The functions, advantages, disadvantages, and languages of the protocols are summarized in Table 1.

The CGI approach is one of the most widely used standards. CGI is a protocol that defines how the program is to be invoked on the server side, how the parameters are passed from the Web server to the program, and how the feedback is retrieved. Programming of CGI is a key component in a Web database system. It communicates with the Web server and database as shown in Figure 1. Through the CGI, the end user can query the remote database and get the results from the query.

The first step in the procedure is to find the C library of the CGI program. In this library, many useful functions, including the functions accessing the database, have been defined with certain computer languages. Then, the computer program needs to be written to execute the main functions. These procedures include several steps. First, the program receives the form parameters from the Web server and organizes the form into a SQL language. Then, it submits the information to a database, such as Microsoft Access, and gets the results and stores them. After this process, the program creates a dynamic format of HTML code for the results. Finally, actions are added and the system returns it to the Web server, which activates the CGI program to the *html* file, so that the user can review the query results.

Table 1. Comparisons of three basic protocols used for controlling the interaction between the Web browser and the server program.

Protocol type	CGI	Java Servlet	Java Applet
Function	Server program, works between Web server and server program, use process	Works between Web server and server program, uses threads	Execute within Web browser
Advantage	Use a fast language, C++	Platform independence, serve more sessions at one time, suitable for large-scale applications	Flexibility and control power, computing ability in Web browser
Disadvantage	One-time action	Slower language, Java,	Slow speed, complexity
Language	C/C++, Perl, Java, and others	Java	Java

System Architecture

The early architecture of a Web sever was a two-tier system. The client (user) was in charge of controlling the logic and coordinating with other clients. This structure limited the performance and the flexibility of the system. The three-tier software architecture emerged in the 1990s to overcome the limitations of the two-tier architecture. A middle layer was added between the user interface and the database server. This middle layer incorporates the business logic and coordinates hundreds of transactions. The modification of the underlying database required adjustment or modification of the existing client. The middle layer in this three-tier software architecture can shield the details of the database by providing abstract transactions, etc., so that modification of the database will not affect the client. If more database units are added into the system, the middle layer can also adjust to accommodate them. Another advantage of having a middle layer is that the coordination of the clients can be carried out in this layer, which can improve the performance of the system. The three-tier architecture increases the performance, flexibility, maintainability, reusability, scalability, and cost of the system. The three-tier architecture is appropriate for the Web database system. The browser serves as a client, the Web server and the server program form the middle layer, and the database management system is the last layer.

Programming and Maintenance of Deltasoy

Issues related to agronomic aspects and elements about structural design, software, and hardware involved in the Deltasoy have been reviewed in detail in another paper (Zhang et al., 2002). Agronomic aspects of database design and standard selections followed the descriptions in Mississippi Agricultural and Forestry Experiment Station Information Bulletin 363 (White et al., 1999). This bulletin focuses on the administrative tools and the practical issues related to input and output formats and database maintenance.

CGI Programming

C++, which is one of the most popular programming languages, was the primary language in programming Deltasoy. A server program written in C++ retrieves user commands by CGI in the backend and accesses the database using a Microsoft database access object (DAO) interface. The DAO interface is easy to use. Because Microsoft DAO was used to access the database, two specific files — dbdao.h and ddao35.dll — were included. These define the interface that sends the query to a Microsoft Access database and retrieves the result.

The C library for the CGI program in this database includes two files: cgi.c and cgi.h. They contain the functions that define the standard of communication between the Web browser and the server. These also include common components of the form used for queries so that the server program can easily get the value of a string that a user enters in the form.

Administration Tools and Procedures

Administration tools are needed for authorized persons to use and maintain the database. Using Microsoft Access, an application was built that includes the following components.

Main Switchboard — The main switchboard lists all the functions as buttons that, once clicked, will execute the associated functions. It is loaded by default once the database system is opened. It is implemented using an autoexec macro and Deltasoy form.

New Data Input — Variety trial data are collected on an annual basis, thus data must be input once per year. Instead of using a data sheet view for data entry, an input form is provided that is associated with the new table. The user inputs one tuple (a set of data) at a time and, by clicking the *next* button, continues the data entry process. This avoids accidental mistakes of modification caused by seeing the whole datasheet. By using an embedded Visual Basic script that defines the correct action, the input form can remember the data most recently entered, thus avoiding repeated data entry. This capability is implemented using an input form (Figure 2). The new data are stored in the table named *new*.

Import Text File — Another method, which is more likely to be used, imports data from other places in other formats, such as Microsoft Excel files. This process occurs when data are obtained from other states. Because of the variety of sources, a format that is consistent with the database is necessary. The format is a delimited text file, where columns correspond to the attributes. The first line of the text file is the attribute name. The data are appended to the new table. Before importing, there will be warnings, requiring the converter to confirm the correct format and check. This avoids errors caused by importing data in the wrong format. The procedure is implemented by an *import_verify* form and an *import_text* macro.

Data Clean — New data are not directly entered into the yield table because that would increase the risk of putting poor-quality data, or even wrong data, into the database. Before it goes into the main table, the data needs to be checked. This process is also called data scrubbing. It is a technique that uses pattern recognition and other artificial intelligence techniques to upgrade the quality of the raw data before transferring it to the database (McFadden et. al, 1999). Common errors are misspelled names, impossible or erroneous dates, fields used for purposes for which they were never intended, missing data, etc. Some of these errors can be detected by defining integrity constraints or business rules in Microsoft Access, but others need data cleaning. This capability is implemented by a *data_clean* macro and several update queries. The queries recognize some patterns and perform some modifications.

Merge New Table into Old Table — After data cleaning, the new data are merged into the old table. Before merging, the program will remind personnel to confirm that the data have been cleaned. Then the new data are appended to the previous table, and all the tuples in the new table will be deleted. This is to avoid

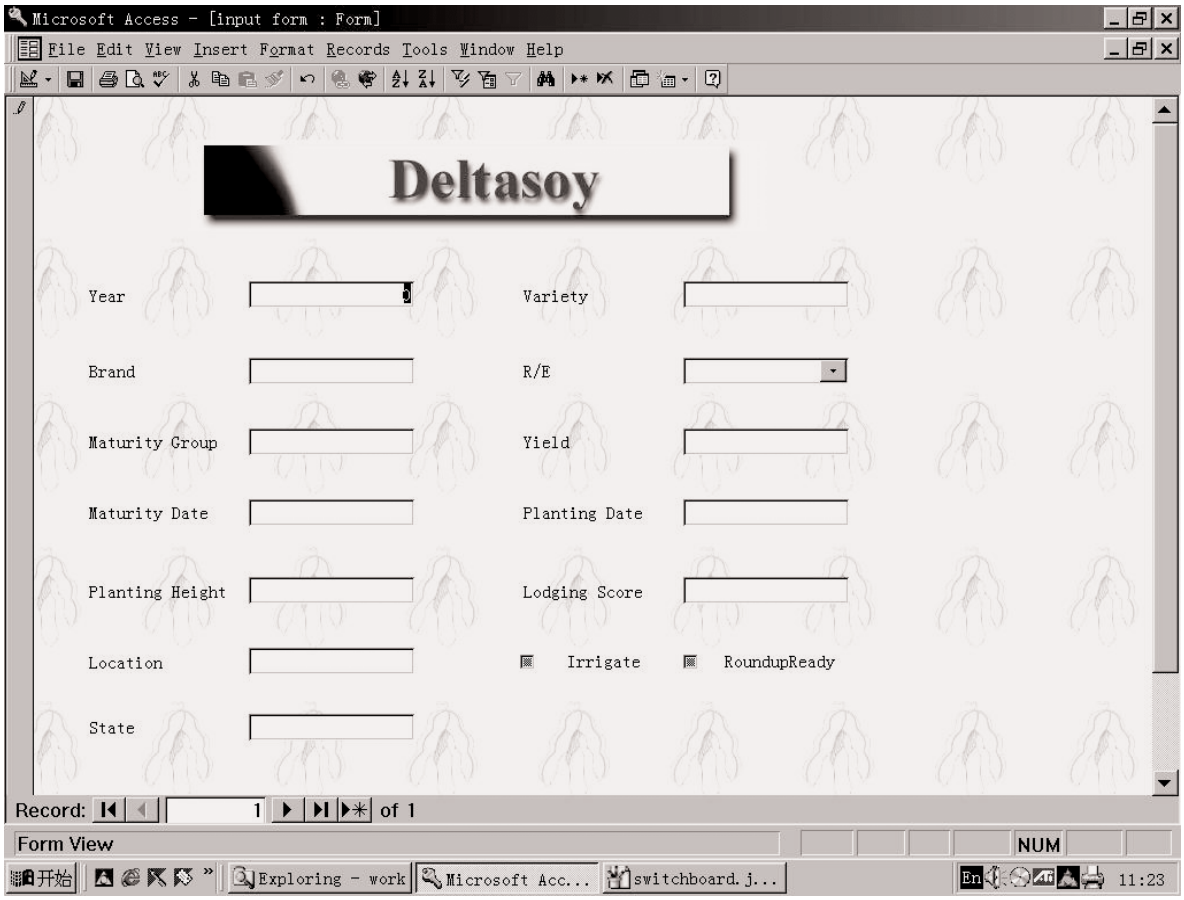


Figure 2. Input form for Deltasoy.

repeatedly adding the same tuples. It is implemented by the *merge_verify* form, a merge macro that runs SQL.

Data Table Backup — Data need to be added into the previous table in such way that, if the table is mistakenly changed, it can be changed back to its previous status. If a copy of the table is not made prior to a deletion, modification, or addition, the correct data may be lost. The backup job is done whenever there is an action about to change the content of the yield table. It is implemented by a backup macro, which deletes the *yield_backup* table, then copies the yield table to the *yield_backup* table.

Restore Yield Table — When an error is made in updating the yield table, the backup file is used to recover the database. However, caution must be used; unlike the backup, which is never harmful, the restore can cause the loss of the current version of the yield table. So there is also a verify form that gives a warning. It is implemented by a restore macro, which deletes the yield table, then copies the *yield_backup* table to the yield table.

All parts of the above administration tools are gathered by a macro-function in Microsoft Access software (Figure 3).

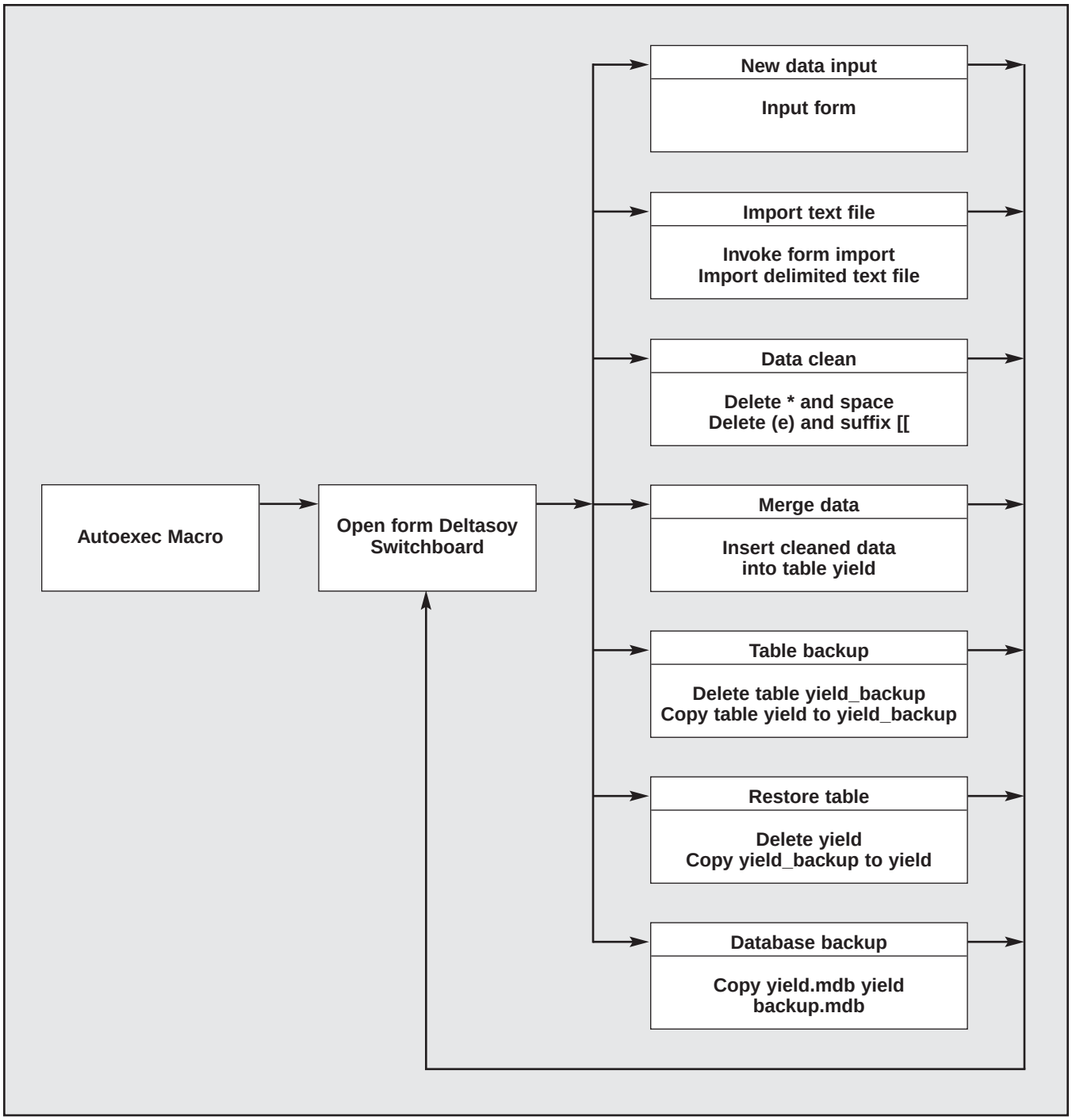


Figure 3. Diagram of relationships among administrative tools of Deltasoy.

Input and Output Format

The database was built with two main CGI programs. These programs answer two different user requests for two forms based on search function: basic search or advanced search. The basic search form is designed for the users who do not request disease information. Parameters being used in the advanced search are exactly the same as for the basic search, except options were added to include disease information. With the advanced search function, users may exclude those varieties that are sensitive to certain diseases (up to three at once). After proper query from the database, some parameters that have been listed as dependent variables, such as planting date and plant height, will be displayed in the result tables.

Basic Search — This program implements a normal search function by constructing the appropriate query. Following is an example of a query:

```
select * from yield
where year=1999
and variety="DP3478"
and location="Stoneville"
and [maturity group]= 4.0-5.0
and [roundup ready]=yes
and irrigation=yes
order by yield asc
```

Figure 4 shows an example of the output results of a query from the basic search with the conditions all the same except with no specific variety selected. If the mean yield over several years is the main item of interest, the query must include a "group by" or "average" clause. When the result is displayed, the query conditions are also listed.

Advanced Search Program — If any disease information is selected, the variety names are underlined, which means the user can click on the variety name to find out more detailed disease information. Following is an example:

```
select * from disease
where variety="DP3478"
```

Figure 5 shows a sample result of an advanced search function. The difference of the output result from the basic search is that the listed variety names are underlined. Clicking an underlined variety name leads to a new table that shows information about the resistance and susceptibility of this variety to some major diseases.

The most important outcome in this database is yield. All yield data are from current Mississippi official variety trials. Later, it should be possible to include the data from other individual credible experiments and from other states. When the search involves multiyear results, mean values can be calculated. In addition to the search functions, this database also has many links set up for general soybean production research. A comment page is provided to allow users to submit comments for the further improvement of the database.

Query Condition:

1. Year: 1999
2. Location: Stoneville
3. Soil Type: All
4. Maturity Group: from 4.0 to 6.0
5. Roundup Ready: No
6. Irrigation: Yes
7. List in order of descent by Yield
8. List 10 records

Result:

Year	Variety	Brand	MG	Yield	Maturity Date	Plant Height	Lodging Score	Irrigation	RR	Planting Date	Location	State	Soil Type
1999	DK5850	DELTAKING	5.8	80.60	9/30/1999	28.00	1.00	Yes	No	5/10/1999	Stoneville	MS	Clay
1999	HBK4890	HORNBECK	4.8	80.00	9/25/1999	39.00	3.00	Yes	No	5/10/1999	Stoneville	MS	Clay
1999	DT96-16809	PUBLIC	5.8	79.80	9/28/1999	32.00	2.00	Yes	No	5/10/1999	Stoneville	MS	Clay
1999	DK5995	DELTA KING	5.9	79.60	10/ 2/1999	25.00	1.00	Yes	No	5/10/1999	Stoneville	MS	Clay
1999	A5959	ASGROW	5.9	79.00	9/30/1999	26.00	1.00	Yes	No	5/10/1999	Stoneville	MS	Clay
1999	TVX4881	TERRALSEED	4.8	78.60	9/24/1999	33.00	3.00	Yes	No	5/10/1999	Stoneville	MS	Clay
1999	A5547	ASGROW	5.5	78.50	9/29/1999	24.00	1.00	Yes	No	5/10/1999	Stoneville	MS	Clay
1999	A5944	ASGROW	5.9	77.90	9/29/1999	29.00	1.00	Yes	No	5/10/1999	Stoneville	MS	Clay
1999	BOLIVAR	PUBLIC	5.8	76.50	9/26/1999	28.00	1.00	Yes	No	5/10/1999	Stoneville	MS	Clay
1999	DK4711	DELTA KING	4.7	76.50	9/23/1999	33.00	3.00	Yes	No	5/10/1999	Stoneville	MS	Clay

Mean Yield=78.70

[Back to search form][Back to Home] [Variety Trial Publication][Location Information]

Figure 4. Example of outcome result table from normal search of Deltasoy.

Query Condition:

1. Year: 1999
2. Location: Stoneville
3. Soil Type: All
4. Maturity Group: from 4.0 to 6.0
5. Resistant Disease: Virus SMV/BPMV
6. Roundup Ready: Yes
7. Irrigation: Yes
8. List in order of descent by Yield
9. List 5 records

Result:

Click variety name to view disease information

Year	Variety	Brand	Maturity Group	Yield	Maturity Date	Plant Height	Lodging Score	Irrigation	Roundup Ready	Planting Date	Location	State	Soil Type
1999	H4994RR	HARTZ	4.9	74.30	9/20/1999	23.00	1.00	Yes	Yes	5/10/1999	Stoneville	MS	Clay
1999	H5999RR	HARTZ	5.9	72.30	9/28/1999	29.00	2.00	Yes	Yes	5/10/1999	Stoneville	MS	Clay
1999	SG498RR	SUREGROW SEED	4.9	68.80	9/21/1999	31.00	1.00	Yes	Yes	5/10/1999	Stoneville	MS	Clay
1999	DP5644RR	DELTAPINE SEED	5.6	67.80	9/27/1999	26.00	1.00	Yes	Yes	5/10/1999	Stoneville	MS	Clay
1999	TS556RR	TERRA	5.5	67.60	9/20/1999	26.00	1.00	Yes	Yes	5/10/1999	Stoneville	MS	Clay

Mean Yield=70.16

[Back to search form][Back to Home] [Variety Trial Publication][Location Information]

Figure 5. Example of outcome result table from advanced search of Deltasoy.

SUMMARY

The applications of computer technology and information systems in agricultural production have increased greatly in recent years. The need for having a multistate variety trial database for soybean is apparent. A well-designed information system can be a very helpful tool for soybean growers, seed producers, extension specialists, and researchers. Internet-based databases provide people with more flexibility and convenience. However, up-to-date applications of these technologies have not been used as extensively as they should. This bulletin describes the technical aspects of constructing a Web-based database and can serve as a sample in helping to apply modern technology in applied agricultural research.

REFERENCES

Hobuss, J.J. 1998. *Building Oracle Web Sites*. Upper Saddle River, NJ: Prentice Hall PTR.

McFadden, F.R., J.A. Hoffer, and M.B. Prescott. 1999. *Modern Database Management*. Reading, MA: Addison-Wesley Publishing Company.

White, B., A. Blaine, R. Ivy, W. Maily, W. Marlow, A. Smith, C. Watson, M. Young, and L. Zhang. 1999. Mississippi soybean variety trials, 1999. Mississippi Agricultural and Forestry Experiment Station Information Bulletin 363.

Zhang, L., W. Qi, L. Su, and F. Whisler. 2002. Deltasoy — An Internet-based soybean database for official variety trials. *Agron. J.* 94(5):1164-1171.

Mississippi State University
125 YEARS
1878—2003

Printed on Recycled Paper

MSUcares.com

17429