

**Development of Data Store Search Engine
for Surface Water Pathway Criteria List**

by

S. Rony Mukhopadhyay
S.B. Civil Engineering
Massachusetts Institute of Technology, 1996

Submitted to the Department of Civil and Environmental Engineering
In Partial Fulfillment of the Requirements for the Degree of

MASTER OF ENGINEERING
IN CIVIL AND ENVIRONMENTAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 1997

© 1997 S. Rony Mukhopadhyay
All rights reserved

*The author hereby grants to M.I.T. permission to reproduce and distribute publicly paper
and electronic copies of this thesis document in whole or in part.*

Signature of the Author _____
~~Department of Civil and Environmental Engineering~~
May 9, 1997

Certified by _____
Patricia J. Culligan-Hensley
Assistant Professor
Civil and Environmental Engineering
Thesis Supervisor

Certified by _____
Feniosky Peña-Mora
Assistant Professor
Civil and Environmental Engineering
Thesis Supervisor

Accepted by _____
Professor Joseph Sussman
Chairman, Department Committee on Graduate Studies

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

JUN 24 1997

Eng.

Development of a Data Store Search Engine for Surface Water Pathway Criteria List

by

S. Rony Mukhopadhyay

Submitted to the Department of Civil and Environmental Engineering
on May 9, 1997 in partial fulfillment of the requirements for the degree of
Master of Engineering in Civil and Environmental Engineering

ABSTRACT

In 1980, the United States Congress established the Superfund Program as a process to identify and remediate hazardous waste sites. The process consists of a number of phases, each successive stage examining the site under more stringent requirements to determine whether remediation is indeed necessary. One of these phases is the Preliminary Assessment (PA) phase, and that is, in part, the focus of this thesis.

The purpose of this work is to describe an Information System that was developed by the author and four colleagues. The system is designed to assist Environmental Engineers in one of the steps of the Preliminary Assessment phase; filling out the PA Scoresheets. The scoresheets are filled out by gathering commonly available data from a variety of different data sources. The system assists in this process by making many data sources available at a single (electronic) location, and recording the information that was gathered from these sources. Though it is not yet fully developed, a demonstration system has been implemented that incorporates several necessary aspects.

This thesis is divided into three sections. Section I describes the Superfund process and the PA scoresheets, as well as the motivation and general architecture of the information system. Section II gives a detailed description of the portion of the system that was developed by the author, the Data Store Search Engine. This includes a description of the architecture implemented, and a technical analysis of the chosen configuration. This part of the system is currently able to answer questions on the scoresheets that relate to the Surface Water Pathway. Section III concludes the work, describing both the Data Store Search Engine and the overall system in terms of the future of the technology used and its expandability. Lastly, there is a discussion of the anticipated impact of the system on the current Preliminary Assessment process.

Thesis Supervisors: Patricia Culligan-Hensley
Assistant Professor

Feniosky Pena-Mora
Assistant Professor

Acknowledgments

There are a number of people I would like to thank for their help:

Feni and **Trish**, my thesis advisors, for their patience and advice all year long.

Joel, **Kai**, **Anna** and **Carrie**, my group members, with whom the concept for the system implemented for this thesis was developed.

The many people in the MIT community, especially in the Intelligent Engineering Systems Laboratory who helped me greatly, taking time out of their busy schedules, and answering my questions at all hours of the day and night.

Most of all, **Ma** and **Baba** for their infinite love and support throughout.

Table of Contents

ABSTRACT	2
ACKNOWLEDGMENTS	3
TABLE OF CONTENTS	4
LIST OF FIGURES	6
SECTION I MOTIVATION AND DESCRIPTION OF THE INFORMATION SYSTEM	7
1 INTRODUCTION	7
1.1 OVERVIEW OF SUPERFUND (CERCLA) PROCESS	7
1.1.1 History.....	7
1.1.2 Current Superfund Process	8
1.2 PRELIMINARY ASSESSMENT UNDER CERCLA	10
1.2.1 Objective	10
1.2.2 Scope.....	10
1.2.3 The Preliminary Assessment Scoresheets - Current Procedure	11
1.2.4 The Preliminary Assessment Scoresheets - Anticipated Future Procedure.....	13
2 INFORMATION SYSTEMS	14
2.1 DEFINITION OF AN INFORMATION SYSTEM.....	14
2.2 APPLICATIONS OF INFORMATION SYSTEMS IN THE ENVIRONMENTAL FIELD	14
2.3 PROPOSED INFORMATION SYSTEM FOR PRELIMINARY SITE ASSESSMENT UNDER CERCLA	15
2.3.1 Proposed System Architecture - the User's Point of View	15
2.3.2 Future Architecture of the System	17
3 EIDSS SECTIONS.....	19
3.1 GRAPHICAL USER INTERFACE FOR EXISTING PRELIMINARY SITE ASSESSMENT SCORESHEET.....	19
3.1.1 Overview of the Graphical User Interface of the Electronic Scoresheet	19
3.1.2 Structure of the Electronic Scoresheet	19
3.1.3 Implementation of the Electronic Scoresheet (The Graphical User Interface).....	20
3.2 DEVELOPMENT OF INTERNET SEARCH ENGINE FOR GROUNDWATER PATHWAY CRITERIA LIST.....	21
3.2.1 Availability of Information on the Internet for Decision Support of PA Scoresheets.....	21
3.2.2 Accessing Internet Search Tools Through the Use of Common Gateway Interface.....	22
3.2.3 Multi-Keyword Searching	22
3.3 AN INTERACTIVE DATABASE OF PA SCORESHEETS FOR CROSS-SITE COMPARISON	24
3.3.1 Motivation	24
3.3.2 Design and Population of Data Storage	24
3.3.3 Database Queries and Comparison Criteria	25
3.3.4 Using the System	26
3.3.5 Making Use of the Results	26
3.4 EXECUTIVE INFORMATION SYSTEM.....	27
SECTION II DEVELOPMENT OF DATA STORE SEARCH ENGINE FOR SURFACE WATER PATHWAY CRITERIA LIST.....	28
4 THE DATA STORE SEARCH ENGINE	28
4.1 THE DATA STORE SEARCH ENGINE'S ROLE IN THE SYSTEM.....	28
4.2 DESCRIPTION OF USGS DATA SOURCE	29
4.3 MATCHING EXISTING DATA SOURCE INFORMATION WITH SCORESHEET QUESTIONS	30
5 IDEAL DATA STORE SEARCH ENGINE ARCHITECTURE - EXCLUSIVELY JDBC	33

5.1	JDBC SET-UP.....	33
5.1.1	<i>java.sql.DriverManager</i>	34
5.1.2	<i>java.sql.Connection</i>	35
5.1.3	<i>java.sql.Statement</i>	36
5.1.4	<i>java.sql.ResultSet</i>	36
5.2	PROBLEMS WITH JDBC.....	37
5.2.1	<i>General JDBC Problems</i>	37
5.2.2	<i>JDBC Problems and their Implications to EIDSS</i>	38
6	ALTERNATIVE SOLUTION - ODBC	39
6.1	ODBC	39
6.2	PROBLEMS WITH ODBC	39
7	THE IMPLEMENTED SOLUTION: USING THE JDBC-ODBC BRIDGE	41
7.1	CURRENT SYSTEM ARCHITECTURE.....	41
7.2	DETAILING ASPECTS OF THE SYSTEM	42
7.2.1	<i>The Web Page Itself</i>	42
7.2.2	<i>The Perl Script</i>	43
7.2.3	<i>The Java Application with the JDBC-ODBC Bridge</i>	45
7.2.4	<i>ODBC</i>	46
	SECTION III DATA STORE SEARCH ENGINE AND EIDSS CONCLUSIONS	47
8	DATA STORE SEARCH ENGINE CONCLUSIONS	47
8.1	ELECTRONIC INFORMATION.....	47
8.1.1	<i>Availability</i>	47
8.1.2	<i>Standardization</i>	48
8.2	TECHNOLOGY - DATA SOURCE CONNECTION.....	48
8.3	EXPANDABILITY	49
9	GENERAL SYSTEM CONCLUSIONS	51
9.1	TECHNOLOGY - LINKING THE EIS WITH THE DSS	51
9.2	EXPANDABILITY - BEYOND THE PRELIMINARY ASSESSMENT PHASE.....	52
9.3	IMPACT OF THE SYSTEM ON THE PRELIMINARY ASSESSMENT PROCESS.....	52
	REFERENCES	54
	APPENDICES.....	55
	APPENDIX A: USGS DATA FILES	55
	APPENDIX B: COMPLETE LIST OF WATER-USE DATA ELEMENTS	56
	APPENDIX C THE PERL SCRIPT	63
	APPENDIX D THE JAVA CODE.....	64

List of Figures

Figure 1.1.2-A Diagram of the Superfund Process	9
Figure 2.3.1-A Proposed EIDSS set-up	17
Figure 4.1-A Diagram of Data Store Search Engine	28
Figure 5.1-A JDBC Class Diagram	34
Figure 7.1-A Diagram of Current Data Store Search Engine Architecture	42

Section I Motivation and Description of the Information System

1 Introduction

1.1 Overview of Superfund (CERCLA) Process

1.1.1 History

In the past, there was little understanding of what effect certain wastes have on human health and the environment. Consequently, numerous abandoned hazardous waste sites contributed to the pollution of the earth's soil, water and air. Some common hazardous waste sites include abandoned warehouses, manufacturing facilities, processing plants and landfills. In 1980, Congress established the Superfund Program to clean up these sites in response to a growing concern over the health and environmental risks posed by hazardous wastes. The Superfund program was created as part of the Comprehensive Environmental Response, Compensation, and Liability Act (CERCLA) which established broad authority for the government to respond to problems posed by the release, or threat of release, of hazardous substances, pollutants, or contaminants. In 1986, CERCLA was amended by the Superfund Amendments and Reauthorization Act (SARA) and by the National Contingency Plan (NCP). At present the U.S. Environmental Protection Agency (EPA), in cooperation with individual states and tribal governments, administers the Superfund Program.

The Superfund Trust Fund was established to support the cost of cleanup of hazardous waste sites under the Superfund program. The Trust Fund is supported from taxes on the chemical and petroleum industries and is used primarily when those responsible for contamination at Superfund sites cannot be found, or cannot pay for the cleanup work.

1.1.2 Current Superfund Process

The Superfund process consists of two main phases: site assessment and remedial response action (see figure 1.1.2-A). Site assessment is the evaluation of all sites to determine those for which some response action may be required. If appropriate, the result of the site assessment process is the listing of a hazardous waste site on the National Priorities List (NPL). For sites that are placed on the NPL, the second phase of the superfund process, the remedial response action, is performed. During this phase, the nature and extent of contamination is determined, followed by the selection and implementation of any necessary cleanups at the site. If threats to human health are imminent, immediate or short-term responses may be performed during either of these two main phases.

The site assessment phase begins with notification to EPA of possible releases of hazardous substances. Sites are then entered into the Comprehensive Environmental Response, Compensation, and Liability Information System (CERCLIS), which is the EPA's computerized inventory of potential hazardous substance release sites. The site assessment phase continues with Preliminary Assessment (PA) and Site Inspection (SI) phases. The Preliminary Assessment phase uses relatively limited data that are readily available to identify sites that may pose a threat to human health and the environment, and therefore require further investigation. Only if the Preliminary Assessment phase recommends further investigation, is the Site Inspection performed. The purpose of the Site Inspection is to determine which sites have a high probability of qualifying for the NPL. Once a site has been placed on the NPL, the site will undergo the remedial response action phase, as explained previously.

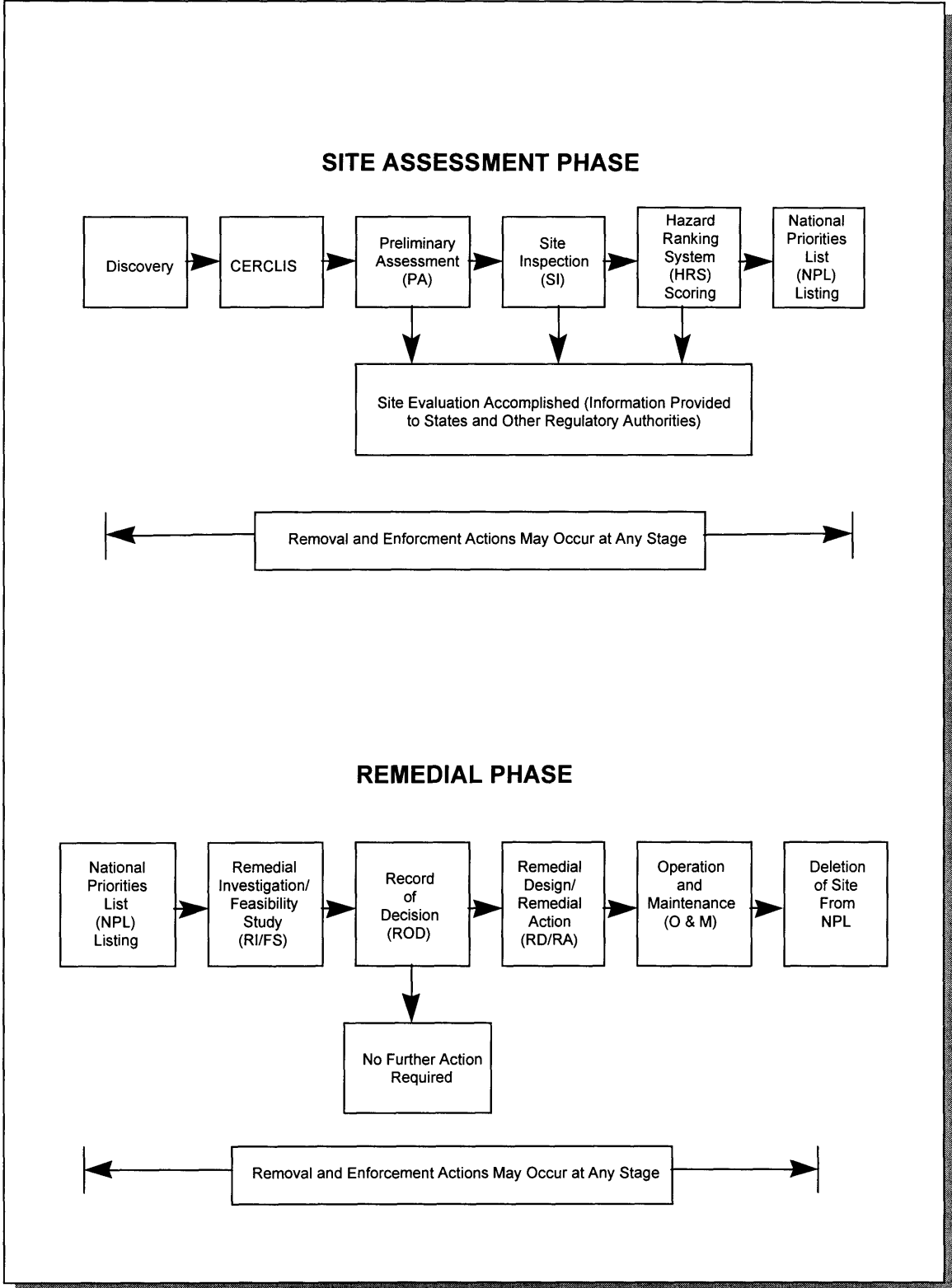


Figure 1.1.2-A Diagram of the Superfund Process

Because the Executive Information / Decision Support System (EIDSS) is designed to help with the PA phase of the Superfund process, this phase is discussed in more detail in the next section.

1.2 Preliminary Assessment under CERCLA

1.2.1 Objective

Based on limited data, the Preliminary Assessment (PA) phase is designed to recommend whether or not a site should undergo further investigation.

1.2.2 Scope

The Guidance for Performing Preliminary Assessments under CERCLA (U.S. EPA 540/G-91/013, Sept. 1991) defines the scope of the Preliminary Assessment as sufficient to complete the following tasks:

- Review existing information about the site.
- Conduct a site and environs reconnaissance.
- Collect additional information about the site with an emphasis on target information.
- Evaluate all information and develop a site score.
- Prepare a brief site summary report and site characteristics form.

The report generated from the PA phase summarizes the information gathered, and based on this information, suggests that either (i) the site poses no threat to human health or the environment; (ii) there is a potential threat and the site needs further investigation; or (iii) emergency actions are necessary. If the site is determined to be potentially hazardous, the PA report will often be referred to throughout successive stages of the Superfund process.

The PA report consists of three parts: the data and site characteristic form, the narrative report and the PA Scoresheets. Filling out these three parts takes an average of 120 hours for each site, and the information can be presented informally (i.e. legible handwriting as opposed to typewritten) (U.S. EPA 1991). The data and site characteristics form, entitled “Potential Hazardous Waste Site Preliminary Assessment Form,” is a four page summary of the PA scoresheets and the narrative report. The narrative report summarizes all of the information researched and the most important characteristics of the site. It also explains the major points of concern. EIDSS was specifically designed to assist with filling out the final part, the PA scoresheets, which are detailed in the next section.

1.2.3 The Preliminary Assessment Scoresheets - Current Procedure

The PA scoresheets are distributed as a workbook made up of checklists, worksheets, factor value tables, and scoring forms, each with brief instructions and guidelines for scoring. Some regions in the United States may require additional scoresheets, but there is a set of standard scoresheets that must be filled out for all regions.

The scoresheets are divided into six sections; General Site Information, Source and Waste Evaluation, and four more sections corresponding to the four hazardous substance exposure routes called pathways; Ground Water Pathway, Surface Water Pathway, Soil Exposure Pathway and Air Pathway. Each pathway section is loosely divided into three sections based on factor categories; likelihood of release (relative likelihood of a hazardous substance migrating from the site through the specific pathway), target (presence of people, physical resources or environmental resources that may be threatened by release of a hazardous material from the site), and waste characteristics (an estimation of the type and quantity of the wastes at the site). The particular importance of each factor can vary with the pathway, but, for example, primary targets

(such as presence of a school nearby) are weighed heavily in the score regardless of pathway.

The left-hand pages of the scoresheet workbook provide instructions for filling out the right-hand pages. The left-hand pages also often explain the questions in greater detail, or provide tables and formulas to transfer the data obtained into a numerical score for a particular section. There is also a review for internal consistency included in the workbook, which is designed to eliminate inconsistencies between sections of the report. The EPA stresses, however, that the reviews and guidelines are merely to assist the environmental engineers in the scoring process, and much of the time the engineer will be expected to use his or her professional judgment in the actual scoring.

In this manner, many sections or pages are assigned a total score, which is combined at the end to determine the overall score of the site. Many of the pages, however, simply ask for an explanation of certain aspects of the site in paragraph form, rather than request a numerical score. The total time to research the information and score a site averages about 100 hours, and writing the reports averages about 20 hours (U.S. EPA 1991). Sites determined to be ineligible for CERCLA response (i.e. sites where there is no danger of hazardous waste leakage; not simply a lack of targets) may submit abbreviated PA reports. The scoresheets need not be submitted for CERCLIS analysis. However, the first two pages of the Potential Hazardous Waste Site Preliminary Assessment Form and the narrative report remain a requirement.

Finally, the decision (i.e. further action or no action) made on the outcome of the PA is usually based on the overall site score. In general, a score of 28.50 or higher receives a recommendation for further investigation, while a score of less than 28.50 receives a “No Further Remedial Action Planned” (NFRAP) recommendation.

1.2.4 The Preliminary Assessment Scoresheets - Anticipated Future Procedure

To aid in the process of gathering data and storing decisions made during the preliminary assessment phase, data sources in electronic format may be used. There are many advantages to be gained by converting the traditional paper PA scoresheets to electronic forms on networked computers. Multiple engineers can work on the same scoresheet without the delay of paperwork transfer from one office to the other. Completed and partially completed scoresheets may be stored for the use of updating information or retrieving records of past decisions at any point in the future. Also, electronic scoresheets can easily be linked to the Internet or databases where large amounts of relevant information may be found. Lastly, once the scoresheet is completed and stored in a database, an executive summary report may easily be generated.

2 Information Systems

2.1 Definition of an Information System

An information system is simply a system for retrieving appropriate and relevant data from a source and transferring it to a designated target using a different format. A more advanced information system would also consist of a unit for processing the data and adding meaning to it.

A Decision Support System (DSS) is an information system designed to provide users access to information crucial to their decision-making processes. The scope of a DSS is rather broad; any system providing its user knowledge can be categorized as a DSS. An EIS is an information system designed for executives, or others in the organization that may wish to access the DSS to, for example, see why a particular decision was made, or by whom and when.

2.2 Applications of Information Systems in the Environmental Field

Environmental decision making involves understanding not only the immediate impact of human activity on the environment, but also issues like human health, current and future costs, environmental regulation and fairness. In principle, all of these interrelated factors have a bearing on any decision made relating to the environment.

To deal with these complex problems, the environmental engineering industry could greatly benefit by utilizing information technology. The system developed for this project, called the EIDSS seeks to help with information management during the Preliminary Assessment phase of the Superfund process by using a combined Decision Support System (DSS) and an Executive Information System (EIS). The EIDSS simplifies the gathering of information by allowing the

access of many sources of information relevant to the PA process from one location (the EIDSS web site). In future, an “executive information summary” will also be accessible from this web site, which can be used to provide, for example, accountability and correctability to the process of filling out the scoresheets.

2.3 Proposed Information System for Preliminary Site Assessment under CERCLA

The global objective of this project was to develop an information system that can support decision making during the complex process of hazardous waste site remediation. This section describes the proposed Information System for Preliminary Site Assessment under CERCLA, which was used as a focus for the project objective. The DSS and the EIS components of the system, which are currently not linked, are described in the following two sections.

2.3.1 Proposed System Architecture - the User's Point of View

The existing portion of the DSS section developed to date has been implemented on the World Wide Web. Each of the Preliminary Assessment scoresheets is in electronic form at the EIDSS web site. Currently, anyone can access and use the system. However, in the future the system will require a username and password, so that only registered users (paying subscribers) can have access to the PA information of different sites.

After logging in, the user must enter the desired Superfund site to assess or start a new site assessment. Then, the user selects which question to answer, and the system accesses the data store that is relevant to that question and returns the answer using the Data Store Search

Engine (the Data Store Search Engine is the part of the system developed by the author and is discussed in great detail in Section II). Often, the answer will be accompanied by a recommendation for evaluation and a confidence level assigned to the data source thus accessed. The user will then set his or her own confidence level to the answer. In the future, when the DSS and the EIS information systems for the PA are connected, the results of the query will be recorded in a database within the system, along with the user that made the query, the source of the query information and what question it was used to answer. If no answer is available in standard electronic form, the system will access its own Intelligent Search Engine (discussed further in Section 3.2) to point out possible storage locations for relevant information on the Internet.

For example, one of the (multiple choice) questions on the scoresheet is, “Is precipitation in the area heavy?” with possible answers, “Yes,” “No,” and “Unknown.” Asking for the answer to this question will send the system to a database that has precipitation information for the area. In this case, the system will return a numeric value for the average amount of yearly rainfall in the area of the previously entered site. This value may be accompanied by a confidence rating for the source and/or a recommendation (in this case a statistic saying what level of precipitation is considered heavy). The user can take this recommendation or not, thereby answering the question, and then assign a confidence level to the answer. The user may choose not to answer the question at this particular time by indicating that no decision is made, and go on to the next question or exit the system.

When the end of the scoresheets is reached, the user will be able to ask the system to calculate a score for the site based on the questions answered. The user can obtain a more permanent copy of the work by printing out the scoresheets. Figure 2.3.1-A is a diagram of the

proposed EIDSS set-up.

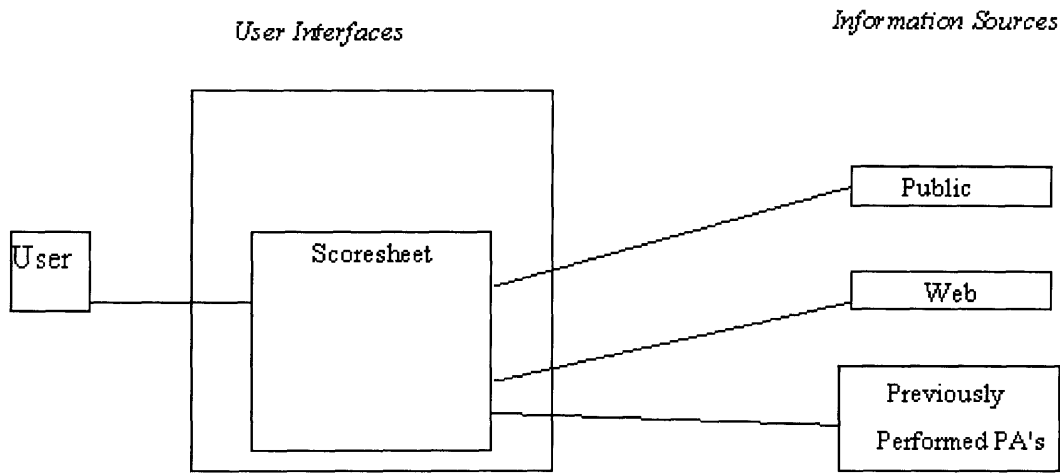


Figure 2.3.1-A Proposed EIDSS set-up

2.3.2 Future Architecture of the System

While the proposed information system will greatly improve the current PA process by making information more reliable and readily available, the potential exists for future enhancements to both the PA process and the information system supporting it. Under the current system, information from a variety of electronic sources facilitates the question-by-question completion of the PA scoresheets. Calculating a final numerical score that determines the next stage of the CERCLA process could be made easier, and more consistent, by somehow reusing the completed site/facility assessments. Once the proposed information system has been used to complete one or more PA's, a database of the results could be created. This additional data source could provide valuable comparisons for performing new studies. If a new site has characteristics that are in some way similar to sites that have been studied previously, completing the PA process may be facilitated by referring to these previous studies. A detailed description of system that would incorporate previously conducted site assessments is in Section 3.3.

Expanding beyond the scope of aiding engineers and scientists, the Decision Support System can incorporate the concept of an Executive Information System to provide better information for senior management. For example, a senior manager could use the system to manage the individual progress of PA's of the various sites that he or she oversees. An EIS also opens new opportunities in document tracking and decision management. These new functions allow senior managers, especially the Chief Executive Officer and the Chief Financial Officer, to better monitor and control the company performance and growth rate. The EIS could also be accessed from the EIDSS web page, but by users with the proper permission and/or authority.

3 EIDSS Sections

This chapter provides a description of each part of the proposed system, except for the Data Store Search Engine. The Data Store Search Engine (the section of the system that was implemented by the author) is left for discussion in Section II, where its architecture is described in great detail. Here, the Graphical User Interface, the Internet Search Engine, the Interactive Database for cross-assessment comparison, and the Executive Information System are discussed under separate headings.

3.1 Graphical User Interface for Existing Preliminary Site Assessment Scoresheet

3.1.1 Overview of the Graphical User Interface of the Electronic Scoresheet

The Graphical User Interface for the electronic scoresheet presents information filtered out from the system that is determined to be useful to the user. The interface for the electronic scoresheet provides (1) better definition of fields to make it easier for the user to fill out the PA scoresheets over the web, (2) integration between the parts of the system that provide supporting information, and (3) automation capabilities for calculating the score of the site.

3.1.2 Structure of the Electronic Scoresheet

Since the current procedure for preliminary site assessment undergoes constant change, the format of the scoresheet is also subject to changes. When implementing the current format of the scoresheet document, flexibility was a major consideration in determining the structure and technology used to implement the system.

Currently, the scoresheets are broken into sections corresponding to physical aspects of the environment: namely, ground water, surface water, soil, air, two sections about the site and the source of contamination, and a final conclusion section. Each one of the sections of the electronic scoresheet corresponds to one section of the existing scoresheet document. The first part asks for the name and address of the site and the investigator. Based on this information, both external and internal databases are searched for relevant information that could be used to fill out subsequent parts of the scoresheet.

The next two pages contain more detailed information about the site and the possible source of the contamination. Those pages will eventually be filled out partially by the system, based on the name of the site, and partially by the investigator. Based on this information, an internal database is created that will help with filling out the following parts of the scoresheet.

The next four parts provide the score for individual aspects of the environment, including ground water, surface water, soil pathway, and air pathway. Here, the user will be asked to fill out all of the information that is not filled out automatically by the system. For some of the questions, the user can perform an additional search for information. For each of the pages, the calculation of the score will be automated, based on information provided by the answered questions.

The last part integrates all the information from the individual parts and the final score is calculated automatically by the application.

3.1.3 Implementation of the Electronic Scoresheet (The Graphical User Interface)

The electronic scoresheet is a web-based application whose pages are created dynamically to allow (1) filling out default values relevant to the site that are a result of the initial and sequential searches, and (2) to provide connectivity between the pages. Each of the

eight parts explained in the previous section is implemented as a separate web page.

In its present state, the user is allowed to fill out the scoresheet only sequentially page by page. The values from the previous sections are carried over to the following sections. At a later stage the system can be updated to allow the user to browse back and forth between the pages in any order. For example, HTML frames could be used to implement this feature.

Upon pressing the “Next Page” button, the values of the fields are stored in temporary files, and a new page is created. Writing information into a temporary file is necessary because the web page itself has no capabilities of storing any information. Currently, upon exiting a page, all the information is lost and can not be recovered. For more information on the Graphical User Interface of EIDSS, see Lukasiak, 1997.

3.2 Development of Internet Search Engine for Groundwater Pathway Criteria List

3.2.1 Availability of Information on the Internet for Decision Support of PA Scoresheets

For the purpose of completing a PA scoresheet for an environmental clean up, there are many Internet web sites available to aid in the decision making process. These include both general websites, that contain information which can be applied to virtually any clean-up site, and specific websites, that provide information pertaining to only one particular clean-up location. It is important to note that in both cases, documents contained in the website are maintained by the party owning that particular domain and thus the reliability of information found is often indeterminate.

One example of a general website were non-site-specific information pertaining to environment clean up can be found is, “<http://www.epa.gov/>”, of the US Environmental

Protection Agency (EPA). Here, one can find information varying from state and local environmental protection laws, to educational resources, to links to specific clean-up sites. Because the EPA, a well credited source, maintains this website, the information found here could be assumed accurate and reliable. However, the final decision with respect to data reliability is up to the engineer completing the PA scoresheet. Therefore, it may be useful to contact the webserver administrator to verify the status of the information found at any specific website.

3.2.2 Accessing Internet Search Tools Through the Use of Common Gateway Interface

As mentioned earlier, there are many Internet search utilities available free to the public. Some examples of these include Excite (<http://www.excite.com>), Yahoo! (<http://www.yahoo.com>), and Alta Vista (<http://www.altavista.com>). Each of these companies has developed programs that search their extensive databases of URL's (Universal Resource Locators) to return a series of web pages that contain the search string queries entered by users. In each case, the pages returned may vary due to differences in the databases maintained and the search programs created by the different companies. Thus, it may be desirable to use multiple search engines in order to increase your chances of finding exactly the information you require. This is known as "metasearching". (An example of this technique can be found at "<http://metasearch.com/>".)

3.2.3 Multi-Keyword Searching

As anyone who has used an Internet search engine most likely knows, the search results returned are sometimes not exactly the results that were being sought. To limit the pages

returned to only those pertaining to the exact topic being searched, it is often useful to add keywords to the search.

Multi-keyword searching involves adding a series of search words to a search string variable that will help to better describe the information desired. It is helpful if the words used are not likely to be found on any site pertaining to a different subject matter. An example of such multi-keyword searching would be to add the word “groundwater” to a search for the word “environment”. Sites pertaining to topics such as “political environment” or “social environment” will most likely not contain the word “groundwater”. Thus, this fact would prevent thousands of unrelated sites appearing at the top of the search results window and the user will not to bother wading through countless irrelevant sites.

3.3 An Interactive Database of PA Scoresheets for Cross-site Comparison

3.3.1 Motivation

The feasibility of using databases located on the Internet and from local CD-ROM to complete preliminary site assessment scoresheets is limited by the actual availability of information from these data sources. One of the most useful sources of data may be previously completed PA scoresheets. Once the proposed information system discussed in this project has been used to complete one or more PA's, a database of previous scoresheets will be available for site comparisons. If a new site under investigation has characteristics that are in some way similar to sites that have been studied previously, completion of the new PA may be facilitated by referring to the database of these previous studies. Comparing new sites with completed studies will provide an additional check for reliability and support for the current site under assessment. This section summarizes the proposed architecture for an interactive database of PA scoresheets for cross-site comparison. This architecture will enhance the utility of the currently proposed decision support system. The database of scoresheets will additionally be used for the future implementation of the executive information system under development as well.

3.3.2 Design and Population of Data Storage

In order to take full advantage of the completed PA scoresheets, some data storage method must be developed to keep track of the completed studies. A relational database (Hawryszkiewicz, 1990), which is a set of data tables linked by common data fields, is well suited for this purpose.

Any number of commercially available relational databases can be used to develop the tables that store the completed PA scoresheets. As more PA scoresheets are completed and

entered into the database, more information will be available for site comparisons and therefore for support of completion of other PA scoresheets. The best way to populate the database is to provide a direct connection between this component of the Information System the rest of the DSS, allowing storage of user input over the Internet. The ease of accomplishing this task will depend on development of current and new technologies. For example, both Microsoft Access '97 and Oracle provide interfaces to allow connection to their databases through the Internet, but these interfaces are limited by the amount of database manipulation they allow and also by the fact that they are difficult to set up.

3.3.3 Database Queries and Comparison Criteria

The database should be designed with particular queries or retrieval of information from the interactive database in mind. The database should at least allow easy viewing of the various PA scoresheet sections (General Site Information, General Facility Information, Source Evaluation, Ground Water Pathway, Surface Water Pathway, Air Pathway, Soil Pathway) and easy mechanism for overall scoring. Ideally, the user should be able to select a facility and site and then be able to view any of the mentioned sections. The real benefit of the database, however, will be derived from the ability to perform comparisons between sites based on specific criteria such as geographic location, types of hazardous wastes, type of facility or operations, and types of waste sources.

The most flexible option will be the selection of any type and number of parameters for performing site comparisons. For example, sites could be ranked by both geographic location and the type or number of similar hazardous wastes. An additional option of comparing sites by individual sections (ground water, surface water, air, and soil pathways) will allow ranking of sites based on answers within these sections of the PA.

3.3.4 Using the System

To illustrate how the interactive database could be utilized, consider the case where the geographic location is used as a parameter for cross-comparison of sites. A geographic location specified by latitude and longitude coordinates would first be entered. A query would then be run, outputting a list of sites ranked by proximity to the specified geographic location. Alternatively, the user could select to first choose the geographical location of a particular site. The resulting query would return a specific number of sites ranked by proximity to the chosen site. The use of the type of facility or operations, types and sources of wastes could be used in similar fashion. The user could explicitly input a value for which the sites are ranked or select a site for which to perform a cross-comparison. In the case where comparisons are based on individual sections of the PA scoresheet such as the ground water pathway or surface water pathway, a completed or partially completed PA will have to be selected first as the basis for the comparison. The result of all of these queries could be a listing of the most similar PA scoresheets. The user will then be able to select any of the sites returned by the query and view the contents of the completed PA.

3.3.5 Making Use of the Results

If a facility is located very near another, or has been contaminated by the same hazardous substances or is similar in any other way as other sites, the user may find it useful to peruse the information that was used to fill out the completed scoresheets of the previous site. The most useful information may actually be found in the “memo” sections found in each of the major sections of the PA scoresheet. These “memo” sections give further explanation of the rationale behind some of the answers given on the scoresheet, and remind the user of some factors that are not explicitly covered by the PA scoresheet questions. Additionally, where sites share particular

characteristics, the scoresheets can be compared to see if the data are reasonable. For example, if two sites were located very near each other, it would be expected that the population distribution recorded in both scoresheets should be similar as well. For more information on this section of EIDSS, see Guzman, 1997.

3.4 Executive Information System

In addition to these base tables, a series of EIS tables has been build to record information about the data in the base tables. These EIS tables contain information about the confidence level, the person who filled out the field, the date, the information source and any comment related to the field. Indeed, the EIS could simply be an extension of the Database for cross-Preliminary Site Assessment Comparison (described in section 3.3), except that the aforementioned fields of assessor, date, time and source of decision along with comments are added to each table.

Once an executive information system architecture has been designed, a company can implement it using any relational database management system such as Microsoft Access 97, or Oracle7. However, implementation is not simple. Careful selection between different hardware and software packages is very important. Of course, a well design database structure will help speed up the implementation. Again, the same factors regarding the ease of linking the database to the web must be considered when choosing the platform for the EIS section of the system. For more information on the EIS section of EIDSS, see Kuo, 1997.

Section II Development of Data Store Search Engine for Surface Water Pathway Criteria List

4 The Data Store Search Engine

This section describes the part of the system that was developed by the author in detail. Section 4.1 provides an overview of the Data Store Search Engine itself, while the remaining sections provide technical descriptions of the development process.

4.1 The Data Store Search Engine's Role in the System

The data store search engine contains static links to data sources that are in a known, standard format. This section of the system is diagrammed in figure 4.1-A which is a cropped

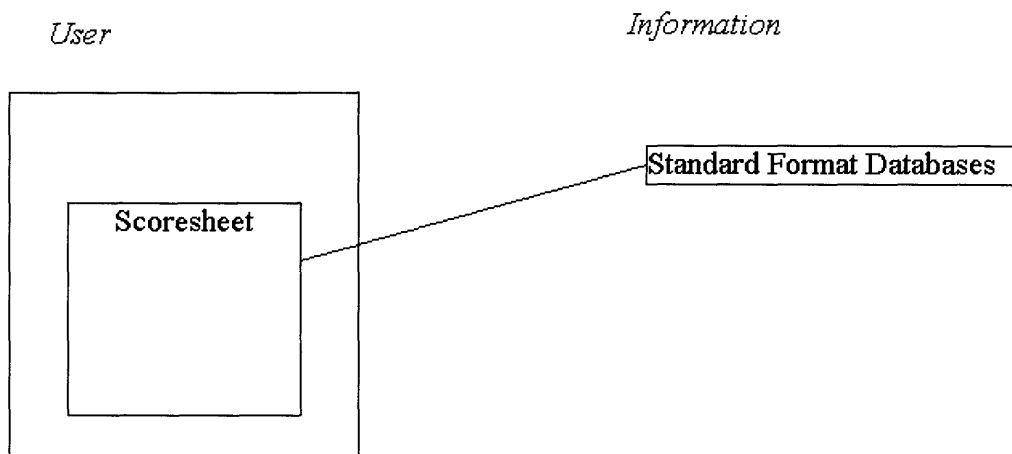


Figure 4.1-A Diagram of Data Store Search Engine

section of figure 2.3.1-A. At the EIDSS web site, a Query button next to a PA scoresheet question indicates that it can be answered by a known, standard format data source. This button initiates the search of the appropriate data source for that particular question. These include data sources that are in a parsable standard format on the World Wide Web, on a connectable CD-ROM, or any other source where the information is in a format that allows the computer to extract specific information from the database without the user's help. Generally, this means that the data is held in a spreadsheet or database format, as opposed to a written document, or a less formatted information list.

The implementation of the data store search engine for this version of EIDSS was limited by the current availability of documents containing PA Scoresheet information in a standard format. The environmental consultant on this project, Kenneth Till was able to locate one such data source; a United States Geological Survey (USGS) web site that provides water-use information for fifty states in the US. For this project, the USGS data source file for the state of Massachusetts was connected to the PA scoresheets used to assess the existing Massachusetts Military Reservation (MMR) Superfund site.

4.2 Description of USGS Data Source

Each of the available USGS data source files contains information for a particular state in the United States. The data were gathered in 1990 and placed in standard text files written in spreadsheet format, with each row corresponding to a county in that state. Although the file currently connected to the PA scoresheet is for the state of Massachusetts, combining the different state files into one countrywide file for performing queries on different Superfund tasks nationwide would not be difficult.

The column headings of the data file are codes for the water-use data elements present. A brief description of the file format is given in Appendix A and a complete list of the water-use data elements along with their codes and descriptions is given in Appendix B. The elements used in the current system, along with the PA scoresheet questions that they can be used to answer are as follows:

- ps-popgw (total population served by the ground water in the area), used to answer questions 3 and 4 on page 8 of the PA scoresheet
- to-totsw (total surface water used in millions of gallons per day), used to answer the first question in the “Suspected Release” column of page 11 of the scoresheets
- do-sstot (total domestic water withdrawals), used to answer the first question in the “Primary Targets” column of page 11 of the scoresheets
- ps-popsw (total population served by the surface water in the area), used to answer questions 4 and 5 on page 12 of the scoresheets

4.3 Matching Existing Data Source Information with Scoresheet Questions

Many of the questions on the scoresheets are currently unanswerable from existing data sources. However, much of the information required is such that it could one day be compiled into a spreadsheet or a database format.

As previously noted, the scoresheet questions that have been linked to the existing data stores will be distinguishable by a “Query” button located next to the question. Pushing the button will open a second browser window that will display the results of the query. The results of any queries run from that point forwards will also appear in this second browser window.

Information received from querying the Data Store Search Engine often requires interpolation on the part of the user to transfer the query result into an answer to a PA scoresheet question. For example, with the question “Is surface water nearby,” (question 1, column 1, page 11), the system currently returns the amount of surface water used in the area. If this number is greater than zero, the user will answer, “Yes,” and if not, “No.” Obvious answers such as this are not automatically filled in by the system in order to ensure that the user plays an active role in answering any question or making any recorded decision.

When the user pushes the “Query” button, he or she is actually executing a program that is on the same server as the web page that they are viewing. As mentioned in Section 4.2, the system currently accesses a USGS file containing Massachusetts’s water-use information. It is envisioned that when the system is complete, the user will log on to a particular account and select the Superfund site that they wish to assess. This selection will automatically tell the system which data files to access. The particular button pushed will tell the system which question is being asked, and the program will determine how to run the query for that particular piece of information. The query will be run on the data source (also currently located on the server of the web page). The information will be parsed into HTML so the web browser can read it, and then post it for the user.

It is important to note that for a particular scoresheet question, the program used to execute the query and the data source components of the system must be on the same server. However, this server does not need to be the same server as that used for the main system user interface. This means that if there is an organization that maintains a standard format data source, the program required to access their data will need to be on their server, otherwise their data files will need to be downloaded to another server. If the source is public, there should not

be a problem bringing data into a server controlled by EIDSS administrators. If it is a private data source, however, permission must be obtained to access it. Once such permission is obtained, the relative locations of the data source and the program can be easily placed as required.

5 Ideal Data Store Search Engine Architecture - Exclusively JDBC

The Data Store Search Engine was originally going to be constructed with Java Database Connectivity (JDBC) alone. JDBC is a group of Java classes and drivers that was created for connecting existing databases of different formats over the web (Patel, 1996). JDBC allows users to perform queries on remote databases through its own structured protocol. It has only been available since the end of 1996, and as a result, there has been little published literature available on it, and some parts of the package are not at all reliable. Further explanation of any of the terms or concepts discussed in this section can be found in Patel, 1996 or in Lemay, 1996.

5.1 JDBC Set-Up

There are a number of steps that must be followed for a user to remotely query a database. The JDBC code must call a number of Java interfaces (that are included in the Java Developer's Kit (JDK) version 1.1) in a particular order to be able to establish a connection to a remote database, create a query that can be executed on that database, execute the query on the database, and finally process the results. As mentioned, JDBC is just a collection of Java classes; it is essentially the subset of Java that is used for database programming. Thus, the terms Java and JDBC will be used interchangeably when referring to the code that was implemented here.

The interfaces (all part of the java.sql library) used to query a remote database are:

java.sql.DriverManager

java.sql.Connection

java.sql.Statement

java.sql.ResultSet

Figure 5.1-A is a diagram of the above interfaces and their relationships, while the next four subsections offer an individual explanation of these relationships.

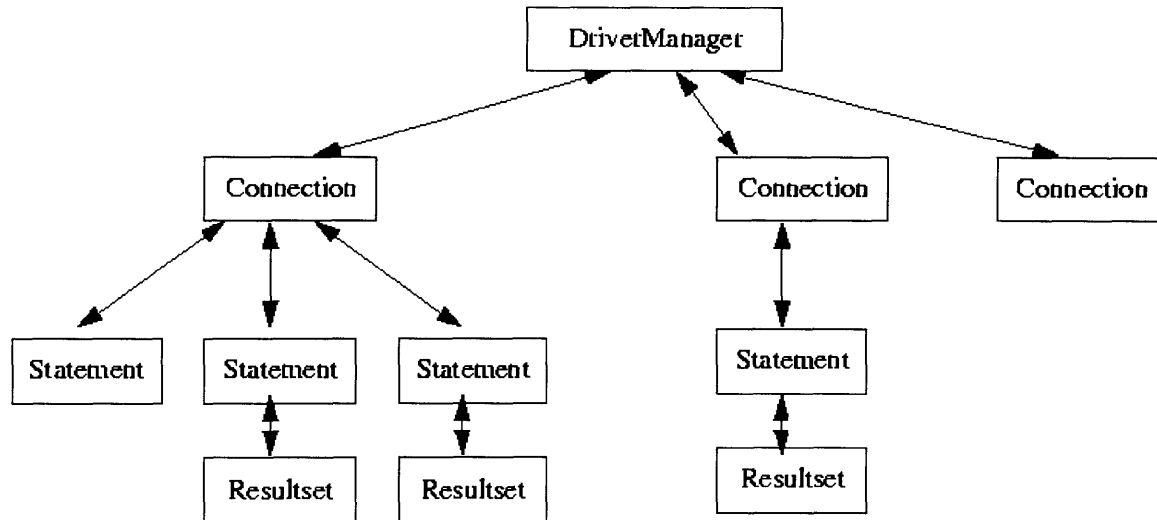


Figure 5.1-A JDBC Class Diagram

It should be stated here that creating a “connection to a database” is the process of establishing the necessary context so that an application can send information to, and receive information from, a data source. The “JDBC driver” is simply a small Java program that has been written to make these connections. Different drivers must be used for different databases. The purpose of JDBC and other database connectivity tools is to shield application developers from much of the complexities of creating this connection. The significance of the complexities is increased by the fact that each database type has a specific protocol for establishing a connection. With JDBC, the application programmer does not need change his or her program because of this.

5.1.1 *java.sql.DriverManager*

This interface is acts as an overseer to the connections that are made to different databases. With the `DriverManager` interface, the programmer will not need to know the specific

location of each of the different database drivers (also included in JDK 1.1), or their names. The programmer must give the location of the Universal Resource Locator (URL - essentially just the location) of the database in standard JDBC compliant format. This is in the form:

```
jdbc:<subprotocol>:<subname>
```

where “jdbc” is the standard base, “subprotocol” is the particular data source type and “subname” is usually the name of the database itself. The subname can be the hostname of the computer where the database resides followed by the name of the database, so that it can be accessed over a network.

5.1.2 java.sql.Connection

An instance of the Connection class must be declared for each database that a connection is being made to. The Connection class represents the actual connection to the driver that will be used to send the user’s queries to the database. The JDBC programmer must set an instance of the Connection class equal to the value returned by the DriverManager’s getConnection method.

The arguments sent to the getConnection method are the URL of the database and, if necessary, a username and password for an account on the database server. From the URL, JDBC determines which database driver is required, and connects to it. This connection remains open until it is explicitly closed, which should be done at the end of the session, when no more queries will be performed on the database. Query statements for each database must be created using the Connection object for that particular database, so if multiple connections are created, the JDBC programmer must keep track of them.

5.1.3 java.sql.Statement

The Statement class is initialized with the createStatement method of the Connection class. The particular instance of Connection that is used will determine which type of database the new Statement can perform queries upon. The Statement class contains the method executeQuery, which actually executes the queries on the databases. Each query that is to be performed must have a Statement object associated with it.

5.1.4 java.sql.ResultSet

The ResultSet class is the class that contains the results of the queries. It must be set equal to the value returned by the executeQuery method of the Statement class. The executeQuery method must have the actual Structured Query Language (SQL - the standard language for communicating query statements to databases (Hawryszkiewicz, 1990)) statement (in correct syntax) of the query sent to it as an argument. This SQL statement can be created by the user dynamically and read in by the Java code, or it can be hard-coded into the program itself. For the system developed here, it makes far more sense to hard-code the statements to the scoresheet questions. Otherwise, the user (typically an environmental scientist or engineer) will have to know the correct SQL syntax and the structure of each of the connected databases well enough to formulate the queries. Also, answering any particular question on the database will always require the SQL statement to be executed on the same database, so hard-coding the SQL is also the best solution from this point of view. In addition, if the database used for a particular question is changed, then the code needs to be changed only one time in the Java program by the

system administrator, rather than having to inform all of the users of the system about the change and teaching them the new query syntax they will need to enter to answer a question.

Once the values returned by `executeQuery` are assigned to the instance of `ResultSet`, the `ResultSet` must be parsed to interpret the results. The `ResultSet` comes in the form of an array of strings, with the elements returned from the query in proper order. The format of the commands that print the results must match the number of columns, fields, etc. returned by the query. Thus, this section must also be coded into the Java program, and processed separately for each query, so the user does not have to worry about the syntax and format of the results.

5.2 Problems with JDBC

The explanation of JDBC architecture given in the previous section describes how this application should theoretically work; however, as mentioned before, the JDBC architecture is very new, and there are many problems associated with a JDBC-only system. Because of these, the Data Store Search Engine was not implemented as a stand-alone JDBC application.

5.2.1 General JDBC Problems

The major problem with JDBC is that the drivers that have been created for the database connection still have bugs. Some of them work properly (as will be described in Section 7.1), but others are still not able to make the connections that they claim to be able to. There are also some classes that support the `executeQuery` method, and should remain invisible to even the JDBC programmer that also have bugs. In developing this system, a complete JDBC solution was attempted, but increasingly back-end classes and interfaces turned out to give compile and run-time errors. Thus, a complete JDBC solution could not be implemented, although it is

anticipated that it probably can be within the next year, when the bugs in the current version are expected to be fixed.

Another significant problem with JDBC, alluded to in the first part of this section, was that because it is relatively new, there is very little documentation or support available on the application. When the development for the Data Store Search Engine began in late January, there were only four books on the subject, essentially only one of which was commonly available. Four more books were due to come out on the subject in February, but only one of those actually had appeared in press by the time of this printing.

5.2.2 JDBC Problems and their Implications to EIDSS

The inability to create a complete JDBC solution required a fundamental change in the intended structure of the system itself. JDBC was the only tool which would allow connection to remote databases; thus, the Java program that was accessing the database and the database itself did not need to be on the same computer. For now, however, this has become necessary. With the current architecture of the Data Store Search Engine that was used to overcome the problems with JDBC (see section 7), every remote database will either have to be downloaded to the server that EIDSS itself is on, or the administrators of the remote data source will have to install a program that can access their database on their own server, and give anyone accessing it through the EIDSS web site permission to run the program. This should not be a problem in implementing the full system, however, since the use of data sources maintained by a third party will require that party's permission regardless of the location of the program. Thus, if permission is granted, it will not matter where exactly the program is located.

6 Alternative Solution - ODBC

6.1 ODBC

The current database connection standard is ODBC (Open Database Connectivity), which is much the same as JDBC, except with C++ classes and functions instead of Java. ODBC, like JDBC, prevents the database application developer from having to deal with the complexities of connecting to a database. ODBC was first developed in the early 1990's, and has become the industry standard for database connectivity (Geiger, 1995). Because of its age and widespread use, its drivers are very reliable and well-supported. As section 6.2 explains, however, it is not well suited for EIDSS, so a complete ODBC solution could not be implemented either.

6.2 Problems with ODBC

The first problem with ODBC is that it is a much more difficult tool to learn than JDBC. Creating ODBC applications to connect to data sources requires more steps, and more lines of code than JDBC. This means that each section of the program that connects to a data source needs to be more complicated, and thus more difficult to develop and maintain.

The major problem with ODBC in terms of EIDSS, however, is that it cannot be implemented over the Internet. The ODBC program itself must reside on the same machine as the database. Thus, the EIDSS administrators would have to make sure that every data source that the system connects to resides on the same server as EIDSS itself, or they would have to implement a small program that connects to each data source on the server of that particular data source. This compounds the previously mentioned problem of ODBC being more complicated

than JDBC, because an ODBC-only solution may entail installing a more difficult to maintain program on someone else's server.

Furthermore, an ODBC-only solution is extremely short-sighted. Once the JDBC drivers are improved (it is anticipated that this will happen within a year), whatever solution is currently implemented should be upgraded to a JDBC-only solution, because that is indeed the ideal architecture for the system. Thus, at whatever time the JDBC drivers are improved, an entire ODBC-only Data Store Search Engine would have to be rebuilt.

For these reasons, an ODBC-only solution was not implemented either.

7 The Implemented Solution: Using the JDBC-ODBC Bridge

7.1 Current System Architecture

Because the Data Store Search Engine for EIDSS could not be implemented using exclusively JDBC, there are a number of extra steps the system currently goes through when querying a database. As mentioned in Section 4.1, only one suitable database was found, which was not specifically a database, but rather a textfile in spreadsheet format. However, since both ODBC and JDBC have Text drivers, this was an allowable data source format for the system.

The current set-up of the system is depicted in figure 7.1-A. The EIDSS web site has all the scoresheet questions and the space to answer them. Because the system could not be implemented using only JDBC, many tools were required, of which JDBC was one. Pushing the “Query” button runs a Perl script that is located on the Common Gateway Interface (CGI) bin (Tittel, 1995) Web server. This Perl script calls a Java application which is set up exactly like a stand-alone JDBC program. However, this application connects to the ODBC driver manager (through the JDBC-ODBC Bridge), which accesses the proper ODBC driver instead of connecting to the JDBC driver. The JDBC-ODBC Bridge can be thought of as another database driver (only the current version of this driver works properly); it is a small Java program that enables the JDBC application to communicate with the ODBC DriverManager.

The ODBC driver is the driver that actually makes the connection to the database, and then executes the query that is sent to it by the Java program. The results are then returned back through to the Perl script, which displays them in a new web browser on the user’s machine. Some additional information on each of these steps is given in the following sections.

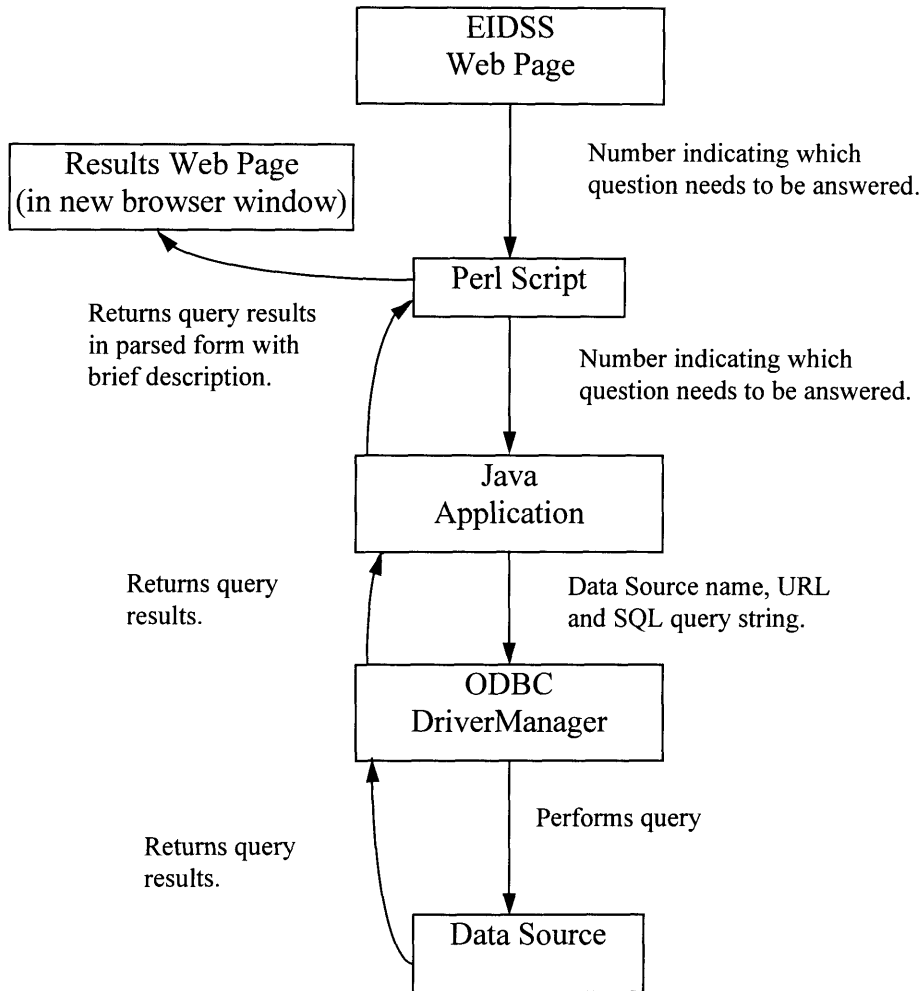


Figure 7.1-A Diagram of Current Data Store Search Engine Architecture

7.2 Detailing Aspects of the System

7.2.1 The Web Page Itself

All of the different "Query" buttons call the same Perl script, but each button passes to the script an integer variable with a different value. From this value, the system determines what query to perform on the database. Currently, the system is only able to fill out PA scoresheet questions for the MMR Superfund site. In the future, however, it may be used for many sites, so the program in the CGI bin must know which site the assessment is being performed on, in order

to determine which database to extract the information from. This will probably be done by setting the user's "cookies". When the user enters later versions of the EIDSS, he or she will first need to enter their username and password, as well as which site they would like to perform an assessment on. When this information is entered, it will be stored in a file on the user's own machine (called "cookies") which any CGI program linked to the web page can check at any time until the web browser is closed.

Also, later versions of the Data Store Search Engine will have links to different data sources, for different questions on different scoresheet pages for different sites. Currently, using an integer variable to distinguish between questions was feasible because there are only a handful of questions to be answered, and they are all being asked of the same site. In future versions, however, the increasing number of variables within the system will make this method rather confusing. Instead, the "Query" buttons will have to send more obvious indications of the questions which they are linked to, such as perhaps text values relating to the question itself. The structure of the programs will be the same, however; it will simply be a matter of the Java program comparing text values rather than numbers to determine which query to perform. This will make it easier for the developers to expand the system, as more data sources become available.

7.2.2 The Perl Script

The Perl script (Tittel, 1995) itself is relatively simple (the commented code for the script is given provided in Appendix C), and serves two basic functions. First, it informs the browser that called it that any output from the Perl script will be in HTML form, so the user's web browser can read it. This is done through a number of "print" statements, which set the title and

headers of the web page. These print statements could have been placed in the Java application, but including them in the Perl script enables a cleaner allocation of the roles in each part of the system.

Secondly, the Perl script calls the Java application, which actually performs the queries, and sends it the variable that indicates which query to perform. The button is not able to call the Java program itself, because of a problem that occurs in the syntax of the command used to run a Java application. When a Java program is written, it must be given a “.java” extension. Java compilers then recognize this file as compilable Java code. The command to compile such a file at the DOS prompt is:

```
javac <filename>.java
```

When this code is compiled, an executable file “<filename>.class” is created. To run this executable file, however, the command is:

```
java <filename>
```

WITHOUT the “.class” extension. In registering executable programs with the CGI bin, however, the filename (including extension) must match the command that executes the file. Because in this case, it does not, Java applications cannot be called directly from a web page.

7.2.3 The Java Application with the JDBC-ODBC Bridge

The code for the Java application (commented code is provided in Appendix D) is exactly as described in Section 5.0 - with sections for connecting to the database, creating and executing a query statement and processing the results. The only difference between this Java code and the code that would be used for a JDBC-only solution is instead of calling the JDBC DriverManager, the application calls the "JDBC-ODBC Bridge". This Bridge acts as a normal JDBC driver as far as the Java application is concerned, in that ensures that a connection is made to the database, and then allows the program to execute queries, and it places the results in the JDBC compliant ResultSet format. However, as the bridge is actually connecting the Java application to the ODBC DriverManager, so that the proper ODBC driver can be implemented. The application calls the bridge and sends the name of the database to be connected to. As with JDBC, the ODBC DriverManager finds the database, determines its type, and then connects using the proper driver. As mentioned in previous sections, the JDBC Drivers are new, and not yet reliable or well supported. ODBC, however, has reliable, widely used and well supported drivers.

Section 6.2 notes that the ODBC-only solution was also avoided because of lack of expandability. A fully developed EIDSS will connect to a number of databases remotely over the Web. Currently, the program must reside on the same server as the database, but this will not be true in the future, when the JDBC drivers are improved. Thus, implementing an ODBC-only solution would mean that when the JDBC drivers are improved, all of the code for the Data Store Search Engine would have to be re-written and re-compiled in a new language (because JDBC is Java and ODBC is C++). This would entail a significant amount of re-development time, and would effectively double the total amount of time that was spent on developing this section of the system. Creating the Search Engine in JDBC and using the Bridge, however, means that later

incorporating the JDBC drivers will only require changing two lines of code. Instead of the Java application calling the Bridge and then the ODBC DriverManager, the application will call the JDBC DriverManager.

7.2.4 ODBC

Because the program calling the drivers that connect to the database must reside on the same server as the database, so must the drivers themselves. Thus, the machine with the databases will need to be equipped with ODBC drivers. This should not be a problem, as all of the most commonly used ODBC drivers come with the Windows 3.1, 95 and NT operating systems. It is unlikely that a party that maintains a database with PA Scoresheet relevant information will use another operating system. However, if this is the case, the ODBC drivers can be downloaded from the Internet.

SECTION III Data Store Search Engine and EIDSS Conclusions

This thesis presented the development of an Information System to support decision making during the Preliminary Assessment (PA) phase of the Superfund process. Concluding thoughts on the system have been divided into two sections; conclusions about the part of the system that was developed by the author, the Data Store Search Engine; and conclusions about EIDSS as a whole.

8 Data Store Search Engine Conclusions

The three major issues that require resolution with respect to the Data Store Search Engine deal with the availability and standardization of electronic information, the technology available, and the expandability of this section of the system.

8.1 Electronic Information

8.1.1 Availability

As mentioned in previous sections, the EIDSS was severely limited by the lack of suitably stored information (environmental consultant Kenneth Till was only able to locate one suitable data source). There is currently very little information relevant to the PA Scoresheets in an electronic format, and even less in a standard format. Thus, the scope of both the Internet Search Engine and the Data Store Search Engine is currently limited. Many of the questions on the scoresheets are conducive to being answered with the assistance of electronic media, but the

questions of when the information will be standardized and who will actually set the standards remain unanswered. It is likely that compiling such data into an electronic form will be done by a non-profit organization, as there appears to be little commercial profitability linked to such a task. Until more standard data sources appear, filling out the relevant sections of the scoresheets electronically is more difficult than filling them out manually.

8.1.2 Standardization

It should also be noted that the information required to complete the PA Scoresheets must be from sources reliable enough that a potential user of the system will be comfortable with the data. If the system is to eventually be set up so that users pay for accounts to use the system, the sources must conform to industry and countrywide reliability standards. This problem is best illustrated by considering the case where information is duplicated in two or more data sources. Under such circumstances, there must be a method to determine which source will be used. Possible determination methods include surveying the current clients to find their preferences and conducting a thorough investigation on the origins of the data sources in order to determine which source is more reliable.

8.2 Technology - Data Source Connection

There were many obstacles encountered in trying to connect the scoresheet pages on the web to the data source itself. A large part of this was because the development of the Data Store Search Engine was first attempted completely using Java Database Connectivity (JDBC), a new tool that is not yet very reliable. As mentioned in Section 7.2.3, the system currently requires that the program executing the query on the data source and the data source itself be on the same

server. So when more sources are gathered, either a program must be installed at the location of each data source, or all of the data sources must be brought on to a single server.

Using JDBC for the entire Search Engine, however, would eliminate this requirement. Thus, there could be one query program located on a server that is easily accessed by system administrators, and the data could be stored anywhere, as long as it is in the correct format and the data source owners give the system access permission. JDBC will likely be more reliable within a year, so the possibility of using a more robust system is conceivable in the future.

8.3 Expandability

Expandability was a large factor in selecting the architecture for this demonstration Data Store Search Engine. Currently, this section simply performs a handful of queries on a local textfile. This is a relatively simple task on its own, and could have been performed with a short Perl script in the CGI bin of the EIDSS web server. This, however, would create a system that would not be at all expandable. Simply using a Perl script to perform the queries would not be possible if the database type changes, or if the relative locations of the database and web site change. The current architecture allows the system to be expanded relatively easy.

In general, the nature of the information requested on the scoresheets is not conducive to being placed in a database. It is far more likely that if data sources are created in the future, whether they are text files on the web, or CD-ROMs, they will be in spreadsheet format, as is the one that EIDSS currently uses to demonstrate the data source search engine. The demonstration system that has been implemented in this project could be expanded to access other text data sources very easily, with the addition of approximately ten lines of code. This expansion will be simple, because the entire framework required to make this type of connection has been

completed. If data sources of other types are found, it is estimated that the amount of code required to incorporate them will be similar. However, there may be some additional configuring that the system administrator would also need to perform. The difficulty of this process will vary with the type of data source to be included in the search engine.

9 General System Conclusions

The conclusions about the EIDSS system itself deal with the technology used, the expandability of the system, and the anticipated impact that a fully developed system will have on the Preliminary Assessment process.

9.1 Technology - Linking the EIS with the DSS

Currently, the system is a separated Executive Information System (EIS) and Decision Support System (DSS). The integration of these two parts was greatly restricted by available technology, although, these restrictions will likely disappear in the near future, as the technology matures.

The basis of the EIS is a local database, which resides on the same machine as the web pages that the user accesses. Each user's decisions are recorded on this database for later examination. The DSS will link data sources in three different forms: an Internet Search Engine locating possible relevant data locations in non-standard form; an Data Store Search Engine retrieving data that is known to exist from standardized data sources; and a Database for Cross-Site Comparison, which allows retrieval of data from previously filled out PA Scoresheets. More database types, however, are becoming easier to connect to the Internet through the use of tools that easily display query results or table values on a web page. This facility makes the inclusion of EIS access from a DSS link very easy to implement. However, the versatility and reliability of these web connections is not yet assured, although they are expected to be so in the near future.

9.2 Expandability - Beyond the Preliminary Assessment Phase

Due to the scope of the project, the information system is currently limited to the Preliminary Assessment phase. Using the EIDSS principle of combining a document-like user-interface, a search engine and a database management system, however, a similar information system could be designed and used in other steps of the Superfund process. This expanded system would then allow users to easily refer to, and use, information from various Superfund process steps throughout their work on a particular site.

9.3 Impact of the System on the Preliminary Assessment Process

This system provides numerous advantages to its users at all levels. First of all, the documentation needed to keep track of the PA process will be greatly reduced. The questions that were filled out electronically will be stored electronically, along with the bibliographical information. In addition, if there is more than one person scoring a certain site, much of the confusion accompanying trading the papers and other documentation will be eliminated by the accessibility of the forms over the Internet. Anyone with permission who needs to access the forms can do so from any office using a desktop computer, or from the field using a laptop. Furthermore, this eliminates the need to remember to bring the proper forms to the proper meetings or to give them to the proper person before they leave for the site, etc. It is also much faster to answer questions using the EIDSS, than it is to manually search out and go through, data sources stored in many different locations.

On a management level, this system provides many advantages to executives as well. The EIS portion provides greater accountability both to the people filling out the scoresheets and the sources being used for the information entered into these scoresheets. Having the exact

source, date and time of a particular decision can greatly simplify the assessment process if particular decisions or sources need to be later examined. Indeed, the system can even be used to identify the engineers that are more efficient at filling out the scoresheets and reward or promote them accordingly.

Though this system is still limited by certain constraints, overall it has great potential to assist in the Superfund Preliminary Site Assessment process. With backing from a few Environmental Engineering firms, data standards can be set and reliable data sources compiled so that future preliminary assessments will be far less error-prone, time-consuming and costly than at present.

References

- Bilicki, Ed. *Environmental Decision Support System*, May 28, 1996
(<http://www.iceis.mcnc.org/EDSS/EDSSPage.html>)
- Dickson, John N. *Environmental Search Engine*, March 10, 1997
(<http://www.webdirectory.com/>)
- Geiger Kyle. *Inside ODBC*, Micorsoft Press, 1995.
- Guzman, Joel. *Interactive Database of Preliminary Assessment Scoresheets for Cross-Site Comparisons*, Master's of Engineering Thesis, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology: Cambridge, 1997.
- Hamilton, Graham and Cattell Rick. *JDBC 1.20 API Specification*, Sun Microsystems, Inc., January 10, 1997 (<http://splash.javasoft.com/jdbc>)
- Hawryszkiewicz. Igor T. *An Introduction to Relational Database Design*, Sydney: Prentice Hall, 1990.
- Kuo, Kai. *Web-based Database-Enabled Executive Information System for Preliminary Site Assessment under CERCLA*, Master's of Engineering Thesis, Department of Civil and Engineering, Massachusetts Institute of Technology: Cambridge, 1997.
- Lemay, Laura and Perkins Charles L. *Teach Yourself Java in 21 days*, Sams.net Publishing, 1996.
- Leidner, Dorothy E. and Elam, Joyce J. "Executive Information System: Their Impact on Executive Decision Making," *Journal of Management Information Systems*, Winter 1993-94, 10:3, page139-155.
- Lukasiak, Anna. *Internet-Enabled Integrated Presentation System for Site Remediation Preliminary Assessment*, Master's of Engineering Thesis, Department of Civil Engineering, Massachusetts Institute of Technology: Cambridge, 1997.
- Patel, Pratik and Moss, Karl. *Java Database Programming with JDBC*, Coriolis Group, Inc., 1996.
- Stanczyk, Stefan. *Theory and Practice of Relational Databases*, London: Pitman Publishing, 1990.
- Tittel, Ed and Gaither, Mark and Hassinger, Sebastian and Erwin, Mike. *Foundations of World Wide Web Programming with HTML and CGI*, IDG Books Worldwide, Inc., 1995.

Appendices

Appendix A: USGS Data Files

All of the fields in the USGS data files were reformatted in May 1996. They are now in a 'spreadsheet' format, in which all fields are separated by a TAB character. This method should allow the data to be easily imported into a spreadsheet program on your computer.

The first line in each data file contains the 'headers', which is an 8 (or so) letter code describing each data element. For example, the code for 'Commercial withdrawals, ground-water, fresh' is 'co-wgwfr'. Each line (record) in the data files begins with some descriptive information, such as the state, year, and county or water-resources region code. There are headers for these items at the beginning of the first line (the header line). These first few header codes vary by file type.

NOTE: The initial header information (first 4 or 5 items in each line) varies by file. The actual code headers for the data elements are described in Appendix B.

The data elements with a "Entered by user" under the "HOW ELEMENT IS COMPUTED" heading indicate data that has been compiled and entered manually into the files. A data element that is not "Entered by user" was calculated by using other data values. For example, 'ps-popgw' (index #3) and 'ps-popsw' (index #4) are summed to create 'ps-popto'. So, total public supply population served is computed by summing the population served by ground water and the population served by surface water.

A table of data elements, element codes and method of computation is in Appendix B.

Source: <http://h2o.usgs.gov/public/watuse/data/wudict.txt>

Appendix B: Complete List of Water-Use Data Elements

WATER-USE DATA ELEMENT		CODE	INDEX	HOW ELEMENT IS COMPUTED
=====	=====	=====	=====	=====
TOTAL POPULATION (in AREA, in thousands)		po-total	2	Entered by user

PUBLIC SUPPLY:				
Population served (thousands):	Ground Water	ps-popgw	3	Entered by user
" "	Surface Water	ps-popsw	4	Entered by user
" "	Total	ps-popto	5	3+4
Water withdrawals, fresh	Ground Water	ps-wgwfr	6	Entered by user
" " "	Surface Water	ps-wswfr	7	Entered by user
" " "	Total, Fresh	ps-wtofr	8	6+7
Water withdrawals, saline	Ground Water	ps-wgwsa	9	Entered by user
" " "	Surface Water	ps-wswsa	10	Entered by user
" " "	Total Saline	ps-wsato	11	9+10
Water withdrawals, total	Total Total	ps-total	12	6+7+9+10
Water deliveries:	Public Use and Losses	ps-loss	13	12-21-30-44-71-93
" "	Total Deliveries	ps-deliv	14	13+21+30+44+71+93
Per capita use:	Total, Gal/d	ps-prcap	15	12*1000/5
Number of facilities:	In Area	ps-facil	16	Entered by user
" " "	Water-Use Database	ps-facdb	17	Entered by user

COMMERCIAL:				
Self-supplied withdrawals:	Ground Water	co-wgwfr	18	Entered by user
" "	Surface Water	co-wswfr	19	Entered by user
" "	Total	co-wtotl	20	18+19
Deliveries from water supply:	Fresh	co-psdel	21	Entered by user
Total:	Withdr. + Deliveries	co-total	22	18+19+21
Consumptive use:		co-cuse	23	Entered by user

DOMESTIC:

SELF SUPPLIED:

Population:	Thousands	do-sspop	24	2-5
Water withdrawals:	Ground Water	do-ssgwf	25	Entered by user
"	Surface Water	do-ssswf	26	Entered by user
"	Total	do-sstot	27	25+26
Per capita use:	Gal/d	do-sspcp	28	$(25+26)*1000/24$

PUBLIC SUPPLIED:

Population:	Thousands	do-pspop	29	5
Deliveries from water supply	Fresh	do-psdel	30	Entered by user
Per capita use:	Gal/d	do-pspcp	31	$30*1000/29$

TOTAL:

Withdrawals + deliveries:		do-total	32	$25+26+30$
Consumptive use:		do-cuse	33	Entered by user

INDUSTRIAL:

Self-supplied withdrawals:	Ground Water, fresh	in-wgwfr	34	Entered by user
" "	Ground Water, saline	in-wgwsa	35	Entered by user
" "	Ground Water, total	in-wgwto	36	$34+35$
" "	Surface Water, fresh	in-wswfr	37	Entered by user
" "	Surface Water, saline	in-wswsa	38	Entered by user
" "	Surface Water, total	in-wswto	39	$37+38$
" "	Total, Fresh	in-wtofr	40	$34+37$
" "	Total, Saline	in-wtosa	41	$35+38$
" "	Total, Total	in-wtotl	42	$34+35+37+38$
" "	Reclaimed Sewage	in-recww	43	Entered by user
Deliveries from water supply	Fresh	in-psdel	44	Entered by user
Total, withdrawal+deliveries	Fresh	in-total	45	$34+37+44$
Consumptive use	Fresh	in-cufr	46	Entered by user
" "	Saline	in-cusal	47	Entered by user
" "	Total	in-cuse	48	$46+47$
Number of facilities:	In Area	in-facil	49	Entered by user

" " "	In Water-Use Database	in-facdb	50	Entered by user

THERMOELECTRIC POWER				
(ELECTRIC):				
All thermoelectric water use:	Ground Water, fresh	pt-wgwfr	51	65+79+87
" " " "	Surface Water, fresh	pt-wswfr	52	66+88
" " " "	Surface Water, saline	pt-wswsa	53	67+89
" " " "	Surface Water, total	pt-wswto	54	66+67+88+89
" " " "	Total, Fresh	pt-frtot	55	65+66+79+87+88
" " " "	Total, Total	pt-wtotl	56	65+66+67+79+80+87+88+89
Deliveries from water supply	Fresh	pt-psdel	57	71+93
Total: Withdrawal + deliv	Fresh	pt-total	58	65+66+71+79+87+88+93
Consumptive use	Fresh	pt-cufr	59	73+82+95
" "	Saline	pt-cusal	60	74+83+96
" "	Total	pt-cuse	61	73+74+82+83+95+96
Total power generated:	Gigawatthours/year	pt-power	62	76+84+98
Number of facilities:	In Area	pt-facil	63	77+85+99
" " "	In Water-Use Database	pt-facdb	64	78+86+100

THERMOELECTRIC POWER

(ELECTRIC), FOSSIL FUEL

Fossil fuel: Withdrawals	Ground Water, fresh	pf-wgwfr	65	Entered by user
" " "	Surface Water, fresh	pf-wswfr	66	Entered by user
" " "	Surface Water, saline	pf-wswsa	67	Entered by user
" " "	Surface Water, total	pf-wswto	68	66+67
" " "	Total, Fresh	pf-frtot	69	65+66
" " "	Total, Total	pf-wtotl	70	65+66+67
Deliveries from water supply	Fresh	pf-psdel	71	Entered by user
Total: Withdrawal + deliver	Fresh	pf-total	72	65+66+71
Consumptive use	Fresh	pf-cufr	73	Entered by user
" "	Saline	pf-cusal	74	Entered by user
" "	Total	pf-cuse	75	73+74

Power generation	Gigawatthours/year	pf-power	76	Entered by user
Number of facilities:	In Area	pf-facil	77	Entered by user
" " "	In Water-Use Database	pf-facdb	78	Entered by user

THERMOELECTRIC POWER

(ELECTRIC), GEOTHERMAL

Geothermal: Withdrawals	Ground Water, fresh	pg-wgwfr	79	Entered by user
" "	Ground Water, saline	pg-wgwsa	80	Entered by user
" "	Total	pg-wtotl	81	79+80
Consumptive use	Fresh	pg-cufr	82	Entered by user
" "	Saline	pg-cusal	83	Entered by user
Power generation	Gigawatthours/year	pg-power	84	Entered by user
Number of facilities:	In Area	pg-facil	85	Entered by user
" " "	In Water-Use Database	pg-facdb	86	Entered by user

THERMOELECTRIC POWER

(ELECTRIC), NUCLEAR:

Nuclear: Withdrawals	Ground Water, fresh	pn-wgwfr	87	Entered by user
" "	Surface Water, fresh	pn-wswfr	88	Entered by user
" "	Surface Water, saline	pn-wswsa	89	Entered by user
" "	Surface Water, total	pn-wswto	90	88+89
" "	Total, Fresh	pn-frtot	91	87+88
" "	Total, Total	pn-wtotl	92	87+88+89
Deliveries from water supply	Fresh	pn-psdel	93	Entered by user
Total: Withdrawal + deliveries	Fresh	pn-total	94	87+88+93
Consumptive use	Fresh	pn-cufr	95	Entered by user
" "	Saline	pn-cusal	96	Entered by user
" "	Total	pn-cuse	97	95+96
Power generation	Gigawatthours/year	pn-power	98	Entered by user
Number of facilities:	In Area	pn-facil	99	Entered by user
" " "	In Water-Use Database	pn-facdb	100	Entered by user

MINING:

Withdrawals:	Ground Water, fresh	mi-wgwfr	101	Entered by user
"	Ground Water, saline	mi-wgwsa	102	Entered by user
"	Ground Water, total	mi-gwtot	103	101+102
"	Surface Water, fresh	mi-wswfr	104	Entered by user
"	Surface Water, saline	mi-wswsa	105	Entered by user
"	Surface Water, total	mi-swtot	106	104+105
Withdrawals, total:	Total, Fresh	mi-frtot	107	101+104
" "	Total, Saline	mi-satot	108	102+105
" "	Total, Total	mi-total	109	101+102+104+105
Consumptive Use:	Fresh	mi-cufr	110	Entered by user
" "	Saline	mi-cusal	111	Entered by user
" "	Total	mi-cuse	112	110+111

LIVESTOCK:

Stock: Withdrawals:	Ground Water	ls-gwtot	113	Entered by user
" "	Surface Water	ls-swtot	114	Entered by user
" "	Total	ls-total	115	113+114
Stock, consumptive use:		ls-cuse	116	Entered by user
Animal specialties, withdrawals:	Ground Water	la-gwtot	117	Entered by user
" " "	Surface Water	la-swtot	118	Entered by user
" " "	Total	la-total	119	117+118
Animal specialties, consumptive use:		la-cuse	120	Entered by user
Total livestock: Withdrawals:	Ground Water	lv-gwtot	121	113+117
" " "	Surface Water	lv-swtot	122	114+118
" " "	Total	lv-total	123	113+114+117+118
Total livestock, consumptive use:		lv-cuse	124	116+120

IRRIGATION

Withdrawals, fresh	Ground Water	ir-wgwfr	125	Entered by user
" "	Surface Water	ir-wswfr	126	Entered by user
"	Reclaimed Sewage	ir-recww	127	Entered by user

" "	Total, Fresh	ir-frm	128	125+126
Irrigated land, in 1000 acres:	Sprayed	ir-spray	129	Entered by user
" " " "	Flooded	ir-flood	130	Entered by user
" " " "	Total	ir-irrig	131	129+130
Conveyance losses:	Total	ir-convy	132	Entered by user
Consumptive use:		ir-cuse	133	Entered by user

HYDROELECTRIC POWER:				
Water use:	Total	hy-total	134	Entered by user
Power generation:	Gigawatthours/year	hy-power	135	Entered by user
Number of facilities:	In Area	hy-facil	136	Entered by user
" " "	In Water-Use Database	hy-facdb	137	Entered by user

SEWAGE TREATMENT:				
Number of facilities:	Public	ww-facpu	138	Entered by user
" " "	Industrial + Other	ww-facot	139	Entered by user
" " "	Total Number	ww-facil	140	138+139
Municipal system returns:		ww-retrn	141	Entered by user
Number of facilities:	In Water-Use Database	ww-facdb	142	Entered by user
Reclaimed waste water (WW) from pub.		ww-recww	143	Entered by user
WW facilities				

RESERVOIR EVAPORATION				
Amt evaporated, (1000 acre ft):	Fresh	re-evap	144	Entered by user
Surface area, in 1000 acres:		re-area	145	Entered by user

TOTAL WATER USE:				
	Ground water, fresh	to-gwfr	146	6+18+25+34+65+79+87+101+113+117+125
	Ground water, saline	to-gwsal	147	9+35+80+102
	Ground water, total	to-totgw	148	6+9+18+25+34+35+65+79+80+87+101+ 102+113+117+125
	Surface water, fresh	to-swfr	149	7+19+26+37+66+88+104+114+118+126

Surface water, saline	to-swsal	150	10+38+67+89+105
Surface water, total	to-totsw	151	7+10+19+26+37+38+66+67+88+89+104+ 105+114+118+126
Total, fresh	to-totfr	152	6+7+18+19+25+26+34+37+65+66+79+87+
Total, saline	to-totsa	153	9+10+35+38+67+80+89+102+105
Total, total	to-total	154	6+7+9+10+18+19+25+26+34+35+37+38+65+ 66+67+79+80+87+88+89+101+102+104+105+ 113+114+117+118+125+126
Reclaimed sewage	to-recww	155	43+127
Consumptive use, fresh	to-cufr	156	23+33+46+73+82+95+110+116+120+133
Consumptive use, saline	to-cusal	157	47+74+83+96+111
Consumptive use, total	to-cuse	158	23+33+46+47+73+74+82+83+95+96+110+ 111+116+120+133
Conveyance losses	to-convy	159	132

Source: <http://h2o.usgs.gov/public/watuse/data/wudict.txt>

Appendix C The Perl Script

```
# The lines that start with the '#' key are comments.

# Sets the default output device to not be buffered
$| = 1;

# The following print statements say that the output will be in
# HTML format so the Web browser that calls this program can read it.
print "Content-Type: text/html\n\n";
print "<HTML>\n";
print "<HEAD>\n";
print "<TITLE>Hello World</TITLE>\n";
print "</HEAD>\n";
print "<BODY>\n";

# This parses the command that calls this program in order to
# get the number that will determine which query to perform
$ENV{QUERY_STRING} =~ /([^\=]+)=(.*)/;

print "<P>";

# The next line calls the Java program and send the number
# which indicates which query to perform.
system("java bonch $2");

# This is the HTML code that closes out the Web page that
# that will open up with the query results.
print "</PRE>\n";
print "</BODY>\n";
print "</HTML>\n";
```

Appendix D The Java Code

```
// The lines that start with '//' are comments

import java.sql.*;
import java.io.*;
import java.util.*;
import java.lang.*;
import sun.jdbc.odbc.*;

class bonch {

public static void main (String argv[])
{
String str=null;
int s=0;

// Make sure that the number indicating which query is to be performed has been passed to this
// Java program. If the argument has been passed, store the value in the variable 's'.
if (argv.length != 1) {
    System.out.println("There is a problem with the number of arguments.");
}
try {
    str = argv[0];
    s = Integer.parseInt(str);
}
catch (Exception e) {
    e.printStackTrace();
    System.out.println (e.getMessage()+"<P>");}

    int count = 0;

java.sql.Connection con = null;
try {
    // Create an instance of the driver

    java.sql.Driver d = (java.sql.Driver) Class.forName("sun.jdbc.odbc.JdbcOdbcDriver").newInstance();

    // URL to use to connect

    String url = "jdbc:odbc:ma90co";

    // Connect to the Text driver

    con = DriverManager.getConnection(url);
}
// In case it was not able to connect to the database
catch (SQLException ex) {
    System.out.println("\n*** SQLException caught ***\n<br>");

    while (ex != null) {
        System.out.println("SQLState: " + ex.getSQLState()+"<br>");
    }
}
}
}
```



```

        System.out.println("Message: " + ex.getMessage()+"<br>");
        System.out.println("Vendor: " + ex.getErrorCode()+"<P>");
        ex = ex.getNextException();
    }
}
catch (java.lang.Exception ex) {
ex.printStackTrace();
}

try {
    ResultSet rs = null;
    Statement stmt = null;
    System.out.println("<b>Query Results:<br>");
    // Determine which query to perform by checking the
    // number that has been passed to this program
    if (s == 1) {
        stmt = con.createStatement();
        rs = stmt.executeQuery ("SELECT pspopgw FROM ma90co.txt WHERE area=15");
        System.out.println ("Total population served by the public supply of groundwater
(thousands):<br>");
    }
    if (s == 2) {
        stmt = con.createStatement();
        rs = stmt.executeQuery ("SELECT toswtot FROM ma90co.txt WHERE area=015");
        System.out.println ("Total surface water used in the area (Million Gal / Day):<br>");
    }
    if (s == 3) {
        stmt = con.createStatement();
        rs = stmt.executeQuery ("SELECT lvttotal FROM ma90co.txt WHERE area=015");
    }
    if (s == 4) {
        stmt = con.createStatement ();
        rs = stmt.executeQuery ("SELECT dossswf FROM ma90co.txt WHERE area=015");

        System.out.println ("Total domestic water withdrawals (Million Gal / Day):<br>");
    }
    if (s == 5) {
        stmt = con.createStatement ();
        rs = stmt.executeQuery ("SELECT pspopsw FROM ma90co.txt WHERE area=015");

        System.out.println ("Total population served by the public supply of surface water
(thousands):<br>");
    }
}

int columns, rows;
columns = (rs.getMetaData()).getColumnCount();
rows = (con.getMetaData()).getMaxRowSize();
boolean more;

more = rs.next ();
// Parse and print the results of the query
while (more) {
    if (s != 3){
        System.out.print (" "+rs.getString(1)+" <br>");
    }
}

```

```

    }
    if (s == 3) {
        String temp = rs.getString(1);
        System.out.println ("Total water withdrawals in area for livestock: "+temp+".<br>");

        if (temp != "0.0"){
            System.out.println ("There is livestock in the area, so there must be some
vegetation<br></b>");
        }
        else {System.out.println ("There is no livestock in the area; evidence is
inconclusive<br></b>");}
    }
    more = rs.next();
} // this is the close of the while (more)

stmt.close();
}
catch (Exception e) {
    e.printStackTrace();
    System.out.println (e.getMessage()+"<P>");}
try{con.close();}
catch (Exception f){
    f.printStackTrace();
    System.out.println (f.getMessage()+"<P>");}
}
}

```