ERDC/CRREL TR-03-18

Cold Regions Research and Engineering Laboratory

US Army Corps of Engineers®
Engineer Research and Development Center

Assessing Hydraulic Modifications on *Vallisneria americana* in Peoria Lake, Illinois

A Pilot Study Using Data Sharing Protocols to Integrate Legacy Models

Patrick B. Black, Elly P.H. Best, E. Allison Newcomb, Terry J. Birkenstock, William L. Boyt, Ronald Heath, and William F. James

September 2003

ERDC/CRREL TR-03-18
September 2003

Assessing Hydraulic Modifications on *Vallisneria americana* in Peoria Lake, Illinois

A Pilot Study Using Data Sharing Protocols to Integrate Legacy Models

Patrick B. Black*, Elly P.H. Best†, E. Allison Newcomb**,
Terry J. Birkenstock*, William L. Boyt‡, Ronald Heath‡,
and William F. James†

**Cold Regions Research and Engineering Laboratory*
72 Lyme Road
Hanover, New Hampshire 03755

†Environmental Laboratory
3909 Halls Ferry Road
Vicksburg, Mississippi 39180

***Information Technology Laboratory*
73909 Halls Ferry Road
Vicksburg, Mississippi 39180

‡Coastal and Hydraulics Laboratory
3909 Halls Ferry Road
Vicksburg, Mississippi 39180

Prepared for OFFICE OF THE CHIEF OF ENGINEERS

ABSTRACT

A pilot modeling study was conducted to assess the utility of implementing the Land Management System's (LMS) Level II Protocols for the efficient sharing of data among legacy models and GIS tools. The modeling was directed toward an investigation of potential population increase of the submersed macrophyte *Vallisneria americana* as a consequence of the construction of a hypothetical levee in the north–south direction in Upper Peoria Lake, Illinois. Numerical models were used to simulate the behavior of the lake in terms of hydrodynamics, sediment concentration, and aquatic plant biomass production for one growth season in 1996 from 1 May to 31 October. Two scenarios were modeled: 1) the current condition without levee, and 2) a hypothetical situation with a levee created in the north–south direction. The model results indicated that the presence of the levee would increase the maximum plant biomass of Vallisneria moderately to greatly in the northern part and along the shores of Upper Peoria Lake, but decreased it in the southern part and along the edges of Lower Peoria Lake. It also would allow the presence of considerably higher tuber numbers at the end of the growth season. The latter would enable a larger plant population to form the subsequent year. Thus, the results of this pilot study indicate that the creation of a levee in the north–south direction may lead to an increase in the Vallisneria population in Peoria Lake. It must be noted, however, that without the levee the water column remains turbid longer than in this modeled situation, because the predominant sediment class has a lower fall velocity and the sediment is resuspended in this wind-exposed lake. The integrated models and tools used in the study were readily available and merely required specific data in appropriate, model-specific formats. A Level II architecture was implemented that promoted efficient data exchange between models and GIS tools.

CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

PREFACE

This report was prepared by Dr. Patrick B. Black, Cold Regions Research and Engineering Laboratory, Dr. Elly P. H. Best, Environmental Laboratory, E. Allison Newcomb, Information Technology Laboratory, Terry Birkenstock, Cold Regions Research and Engineering Laboratory, William L. Boyt, Coastal and Hydraulics Laboratory, Ronald Heath, Coastal and Hydraulics Laboratory, William F. James, Environmental Laboratory, U.S. Army Engineer Research and Development Center.

Funding was provided by ERDC Land Management System Initiative, Work Item 007FB3, *Protocol Development for CW Geospatial Data Management*.

Technical review was provided by Andrew J. Bruzewicz and Dr. Charles H. Racine, both of ERDC-CRREL.

The Commander and Executive Director of the Engineer Research and Development Center is Colonel James R. Rowan, EN. The Director is Dr. James R. Houston.

Assessing Hydraulic Modifications on *Vallisneria americana* in Peoria Lake, Illinois
A Pilot Study Using Data Sharing Protocols to Integrate Legacy Models

P. B. BLACK, ELLY P.H.BEST, E. ALLISON NEWCOMB, TERRY J. BIRKENSTOCK, WILLIAM L. BOYT, RONALD HEATH, WILLIAM F.JAMES

1 INTRODUCTION

This pilot study implemented Land Management System Level II Protocols using an ecological resource management feasibility study as an example. The test example relates to ongoing research at Peoria Lake, Illinois. Since the initial work for this study was started, new technology developed by the LMS group has been incorporated in this write-up to demonstrate how the Level II Protocols are currently implemented.

Submersed vegetation disappeared from the shallow Peoria Lake, Illinois, in the 1950's. Because different aquatic plant species play important roles in providing suitable habitat for epifauna, fish, and waterfowl, great benefit can be derived from promoting their growth. In this pilot study, the indirect effects of building a north–south oriented levee on the plant biomass of the desired species, *V. americana,* were explored using a modeling approach.

The current distribution of submersed vegetation is limited largely because of high turbidity of the water column. Turbidity must be reduced to increase plant growth. Two common methods to reduce turbidity in water are 1) the physical removal of particles, and 2) the application of chemicals to promote flocculation. For this study, a physical method was considered because the application of chemicals is potentially harmful. The standard physical and mechanical methods that can be considered in later studies are dredging and hydraulic modification.

The results of this pilot study reflect the potential changes in plant distribution attributable to changes in turbidity that might take place after the hydraulic behavior of the lake is modified by the modeled levee. Numerical models were employed to simulate these changes relative to the present situation with no levee. The models applied were existing, numerical models developed by the

U.S. Army Engineer Research and Development Center (ERDC). All models and the required data for their operation were controlled by an architecture constructed following the guidelines of the Land Management System (LMS) Level II Protocols.

The LMS is an initiative of the ERDC that is focused on improving analytical and management capabilities on a landscape scale in several of the major mission areas of the USACE. These mission areas include the U.S. Army Civil Works programs (navigation, flood control, water supply and quality, recreation, etc.), operations and management of military installations (specifically military land management), and military engineering and terrain-related operations (trafficability analysis, military hydrology, littoral operations, line-of-site analysis, etc.).

LMS was established to improve cooperative technology development in these mission areas, to improve the USACE's and the Department of Defense's (DOD's) ability to analyze and visualize processes and other features on a landscape scale, and to forecast future conditions as a consequence of alternative management decisions.

The LMS initiative had its roots in a study by the USACE's laboratories in autumn 1995 on modeling and simulation capabilities developed or used by the USACE for landscape or geological processes (e.g., hydrologic, geomorphic, biochemical transport, or ecological processes). A report released summer 1995 by the Defense Science Board titled, *Modeling and Simulation for Environmental Quality*, helped provide impetus for LMS. The USACE laboratory study group, led by Dr. Richard Price and Dr. David Tazik, composed a catalogue of simulation and visualization technology capabilities, and reported on opportunities to the U.S. Army Corps of Engineers Research and Development (CERD) Director in December1996. Based on this study and in consultation with laboratory directors and others, the CERD Director decided to establish the LMS initiative.

The LMS Protocols are procedures and standards built into a framework that allows the flow of data between concurrently running models and various GIS systems to be handled in a standardized manner. The protocols focus on the implementation of modern software technology that facilitates the interoperability of data among legacy software. Level II Protocols provide a convenient and extensible framework to standardize data sharing, and to maintain the persistence of a data-sharing scheme for future reuse.

The problem solving approach of Level II Protocols is to reuse legacy programs by automating data sharing, rather than modifying code to include additional features found in other programs. Thus, the problem in question is analyzed to determine its subcomponents, which are mapped to existing models. An architecture is then employed that allows those models to share the data required

by each model. This architecture-centric approach to computer-based problem solving is used for most modern software programming, but it has scarcely been employed in the environmental modeling domain. Institutional inertia, therefore, may make the implementation of this technology challenging, although it is available for use. These protocols also provide the mechanisms to automatically transfer data among different operating systems, platforms, and programming languages. These characteristics force modelers to use a formal and very rigorous process that clearly defines the data fields and formats required for input and output files for each program. This process will promote the persistence of these links. Thus, once the data fields of a specific model have been fully fitted into this process, the protocol development for these models does not have to be repeated. Future users can reuse the model rapidly without tracking down all the details on data requirements.

Higher Level Protocols focus on different approaches to interoperability and new approaches to technical problem solving in a geospatial domain. In the current pilot study, Level II Protocols were implemented. Problem solving under Level II Protocol involves five steps:

- Define problem requirements.
- Select team of subject matter experts.
- Identify legacy programs and detail their data requirements.
- Build architecture to share data between codes.
- Solve problem.

2 MODELING SYSTEM

Plant Model

This pilot modeling study is aimed at simulating the growth of the submersed macrophyte *V. americana* in Peoria Lake, Illinois, and to explore the potential beneficial impact of the construction of a levee in the north–south direction in Upper Peoria Lake on the growth of this plant. This requires the capability to calculate plant biomass as a function of parameters that govern the behavior of the lake in terms of hydrodynamics and sedimentation related to underwater light climate. The mechanistic model VALLA (Best and Boyd 2001a,b) meets these requirements. The model simulates the carbon flow through the vegetation in a 1-m^2 column of water. It includes descriptions of several factors that affect biomass dynamics, such as site-characteristic changes in climate, water temperature, water transparency, pH and oxygen effects on CO_2 assimilation rate at light saturation, wintering strategies, grazing and mechanical control (removal of shoot biomass), and latitude. The characteristics of community and site can be easily modified by the user.

VALLA incorporates insights into the processes affecting the dynamics of an American wild celery community in relatively shallow, hard water (0.2–6 m depth; dissolved inorganic carbon [DIC] concentration > 0.8 mmol and pH ranging from 7.6 to 9.4), under ample supply of nitrogen and phosphorus in a pest-, disease-, and competitor-free environment under the prevailing weather conditions. The growth starts from the subterranean tubers alone. Plant biomass usually peaks once a year, in July, and intensive downward transport of soluble carbohydrates, used for the formation of tubers that grow into the sediment, occurs after anthesis.

For this study, the primary factor impacting the viability of *V. americana* is light extinction, with the remaining environmental factors assumed constant to reasonable values. Future studies will expand on the results of this work and include dynamic behavior for several other environmental factors. The first order effect that determines light intensity through a column of water in Peoria Lake is sediment concentration; an increase in light availability results from a decrease in sediment concentration. The light extinction coefficient (L) is calculated using the following regression equation, derived from measured data on total suspended sediment (TSS) concentration, turbidity, and light extinction coefficients at sites located outside the main channel (Best et al., in prep.).

$$\ln(L) = (0.585[\ln TSS]) - 0.614$$

To apply this equation to different locations in the lake, the total suspended sediment concentration at any of these locations had to be known. These data had to be obtained from another source than the always scarce field observations. The LMS provides the mechanism to locate models and simulations to assist in land management decisions. Figure 1 is the opening screen to the LMS 2000 client that provides a link to the Protocol Level I LMS model catalog site. A search of the catalog for hydrologic/hydrodynamic models that also deal with sedimentation, revealed five possibilities (Table 1). A comparison of the characteristics of these models indicates that SED2D is a suitable candidate (Appendix A), as it pertains to lakes and is designed for the casual user. Additional advice is required to operate this model, and, therefore, the appropriate points of contacts are provided (TABS support group of the Coastal and Hydraulic Laboratory [ERDC-CHL]).

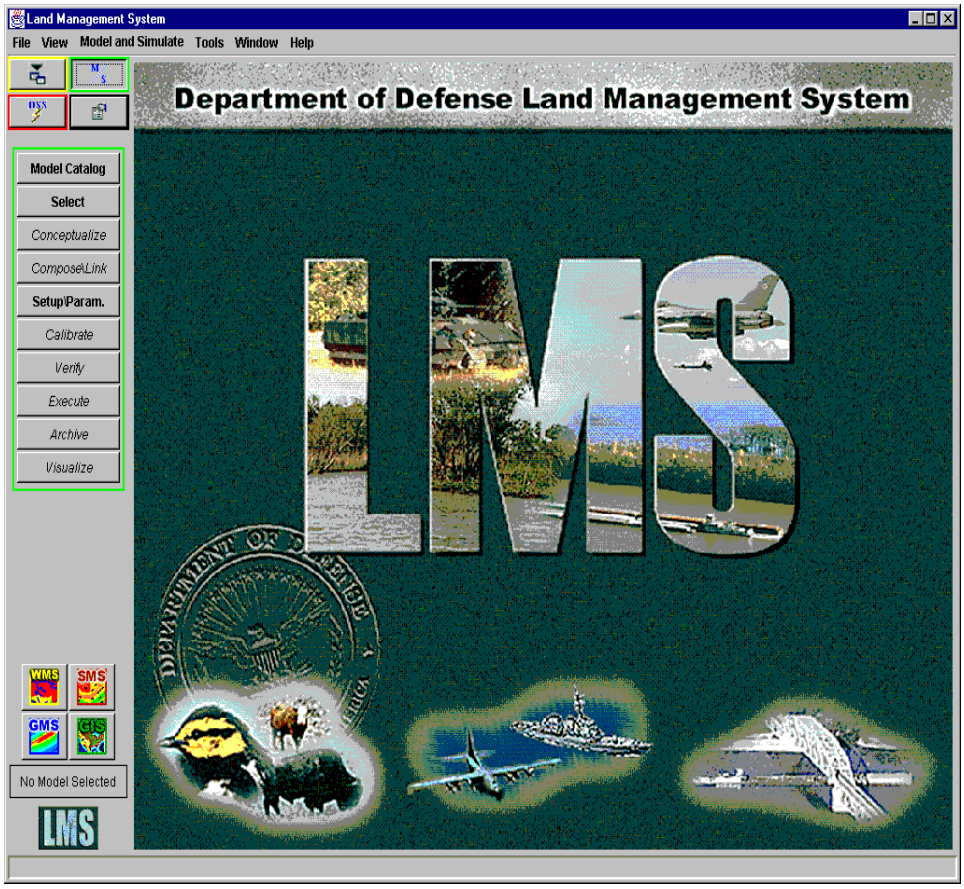

Figure 1. LMS2000 client showing catalog link.

Table 1. Choices of hydrodynamic models that also deal with sedimentation provided by the Level I Model Catalog.

Product Category Search Results

CatalogHome
SearchPage
Information
Future Help

Your seach retrieved 5 items. The keywords you selected looked in the product/model Name, Description, Problem, and Benefit statements. The keyword(s) used in the search were: **' sediment '**. Your search was limited to those products/models that provide: **'Hydrologic/Hydrodynamic Modeling'** . You may also run the ***draft comparison report***.

CH3D-SED - a three dimensional model for hydrodynamics, salinity, and non-cohesive transport: CH3D-SED is a boundary-fitted, non-orthogonal finite difference model for hydrodynamics, salinity and non-cohesive sediment transport. catalog details web site (last updated 09-Aug-00)

HEC-6 (Hydrologic Engineering Center (HEC-6) Scour and Deposition in Rivers and Reservoirs).: This one-dimensional sediment transport model calculates water surface and sediment bed surface profiles by computing the interaction between sediment material in the streambed and the flowing water-sediment mixture. catalog details web site (last updated 10-Aug-00)

RECOVERY - A contaminant model for surface water: Contaminant fate model for estimating long-term chemical concentrations in surface waters and their bottom sediments resulting from contaminant loadings and/or contaminated sediments. catalog details web site (last updated 16-Aug-00)

RMA10-WES - a multi-dimensional hydrodynamic numerical model: RMA10-WES a multi-dimensional hydrodynamic numerical capable of steady state or dynamic simulation of three dimensional hydrodynamics, salinity, and sediment transport. catalog details web site (last updated 16-Aug-00)

SED2D - two dimensional sediment transport model: SED2D a two dimensional sediment transport model for both deposition and erosion of sand or clay bed sediments. catalog details web site (last updated 07-Aug-00)

Geographical Information System (GIS)

A crucial component to modeling plant viability in Peoria Lake is the ability to visualize model output spatially. To accomplish this, we employed a GIS component in our modeling effort. The GIS provides a common framework for analyzing and displaying all data distributed over the lake, and enables the end user to visualize spatial changes. An important aspect of a GIS is that it reduces the workload of the computational system, by accomplishing spatial analyses and visualization of the outcomes.

A GIS representation of Peoria Lake is presented in Figure 2. The lake is characterized by its great length relative to its width and by its narrow channel dissection, both oriented in the north–south direction.

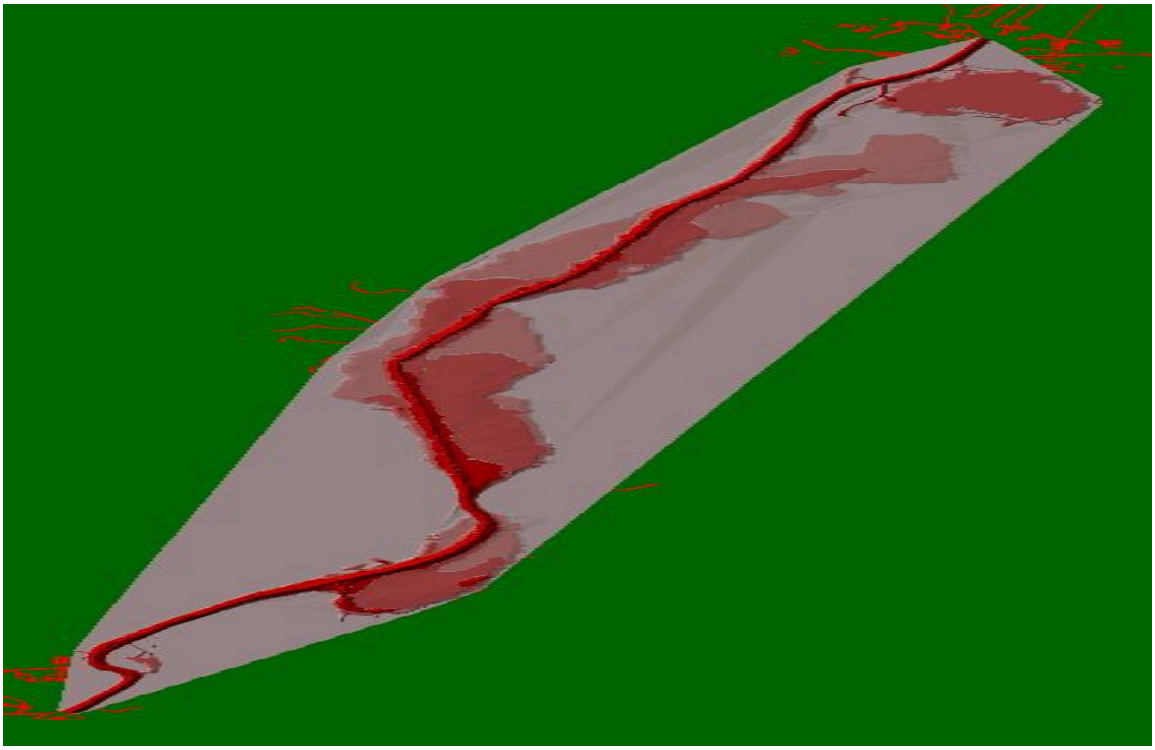

Figure 2. Grand view of Peoria Lake, Illinois.

Hydraulic Models

VALLA requires as input light extinction coefficients that can be derived from sediment concentration throughout the lake. The determination of both longitudinal and lateral distributions of suspended sediment concentrations requires the use of two- or three-dimensional hydrodynamic and sediment transport models. A one-dimensional model would not provide information on lateral variations in stage and current speed or induced circulation within the upper- and lower-turning basins and therefore could not be used to compute the lateral distribution of suspended sediment concentrations. The bathymetry of Peoria Lake in 1998–99 is presented in the maps of Figure 3. It illustrates that the lake is long and shallow, closely approximating a two-dimensional geometry. Based on this information, we assumed that two-dimensional models would be appropriate to model hydrodynamics and sediment transport in this lake. To further simplify the modeling process, the redistribution of bed sediments by wind-driven waves, considered a significant process in Peoria Lake, is not addressed in this pilot study.

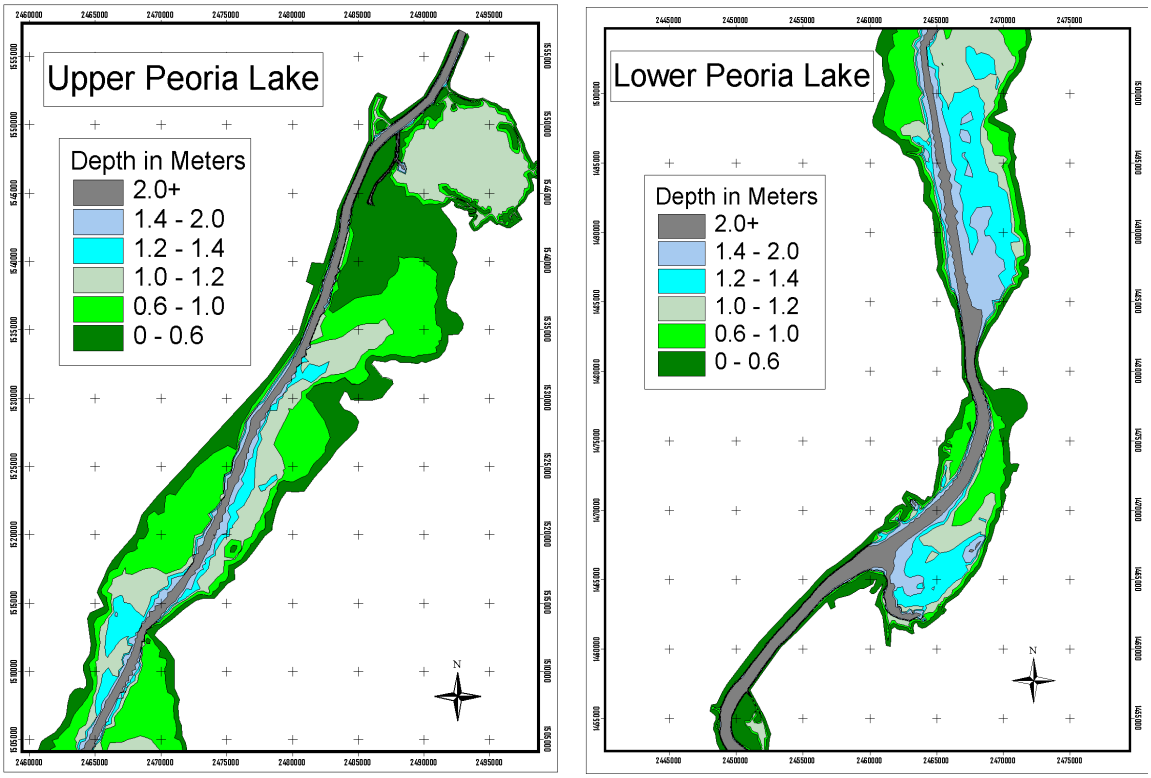

Figure 3. Bathymetry 1998-99 GIS for Peoria Lake.

The ERDC TABS-MD modeling system was selected to calculate suspended sediment concentrations. This modeling system is a family of numerical models that provides multidimensional solutions to open-channel flow and sediment transport problems. The TABS-MD modeling system is the Corps of Engineers' standard for general purpose modeling of two-dimensional, depth-averaged open-channel flow and sediment transport problems and has been supported by the ERDC since the mid-1980's.

The depth and longitudinal and lateral distribution of scour-induced suspended sediment concentrations within the study area were computed using the TABS-MD models, RMA2 and SED2D. RMA2 is a two-dimensional, depth-averaged hydrodynamic model, used to compute water levels and current patterns.* RMA2 employs finite element techniques to solve the Reynolds Form of

* Two-dimensional depth averaged finite element hydrodynamic numerical model. Original developed by Norton, King and Orlob (1973), Water Resources Engineers, for the U.S. Army Corps of Engineer Walla Walla District. Further developments have been made concerning the marsh porosity option. The current version of the code is supported

the Navier-Stokes equations for turbulent flows. SED2D is a companion two-dimensional, depth-averaged sedimentation model, used to compute the scour erosion, transport, and deposition of bed sediments.* SED2D is an enhanced version of the previous TABS-MD sedimentation model, STUDH.

A common finite-element mesh is used by both models to describe the geometry and spatial distribution of various physical characteristics of the study area, such as Manning's roughness coefficients and bed sediment characteristics. During a simulated flood event, RMA2 computes a time-history of water depth and depth-averaged velocity at each node in the mesh. These data are used by SED2D to compute the transport of suspended sediments through the mesh and a bed shear stress at each node. SED2D compares the bed shear stress to the measured or estimated physical properties of the bed material to compute the sediment transport rate and cumulative scour erosion or deposition of bed material at each node and the corresponding changes in suspended sediment concentrations.

The sedimentation model uses a layered bed structure to characterize the spatial distribution of bed material properties, e.g., density, critical shear stresses, and erodibility, horizontally and with depth in the bed. Erodibility of the bed depends on the threshold or critical shear stress for erosion of the material, and a corresponding erosion rate constant. The erosion threshold controls at what level of applied bed shear stress erosion begins. The applied bed shear stress together with the erosion rate constant controls how rapidly sediment layers erode with shear stress in excess of the threshold value. Deposition occurs when the applied bed shear stress is less than the critical shear stress for deposition. The sediment particle fall velocity and the local suspended sediment concentration determine the deposition rate. These parameters are determined by erosion experiments and characterization tests on material from the site. For a certain erosion rate (mass per unit area), the model computes the change in bed elevation based on the dry density of the bed layers and the computed erosion or deposition rate.

---

in TABS-MD by the U.S. ERDC, Coastal and Hydraulics Laboratory. Version 4.52 was used for the current Peoria Lake pilot study.

* A generalized computer program for two-dimensional, vertically averaged sediment transport. SED2D-WES is a rewrite of the program STUDH, developed by R. Ariathurai (1972), Univ. of California, Davis, California. Further developments are still being made. The current version of the code is supported in TABS-MD by the U.S. ERDC, Coastal and Hydraulics Laboratory. Version 4.50 was used for the current Peoria Lake pilot study.

Level II Protocol Architecture

A very important objective of the Peoria Lake Pilot study was to show the current implementation of Protocol Level II. The approach used for this protocol is an architecture based upon a set of client-server applications that will allow the legacy programs' data files to be shared through an Oracle database. The database consists of a set of data files and their field mappings between the various input/output data fields of the legacy programs. All programs interoperate through sharing data with the common Oracle database.

The pilot was built using geographically dispersed teams. The numerical models RMA2 and SED2D were managed by the ERDC-CHL. The aquatic plant viability model was provided by the Environmental Laboratory (ERDC-EL). The Oracle database was developed and maintained at the Information Technology Laboratory (ERDC-ITL). The GIS was developed and maintained at the Cold Regions Research and Engineering Laboratory Remote Sensing/GIS Center (ERDC-CRREL). All work for the pilot was coordinated from Picatinny Arsenal (TACOM-ARDEC).

The design and implementation of a suitable modeling strategy to address ecological resource management questions in the Peoria Lake Pilot Study were greatly facilitated by diagrams. These diagrams help promote common understanding among the team members, as well as the users of the models. Therefore, diagrams are used as much as possible to illustrate the technical concepts involved in this demonstration. 'Use-case analysis' is also used frequently to promote common understanding. 'Use-case analysis' is short narrative of tasks and processes that the modeling strategy must complete. Appendix B contains those use-cases employed to assist the team in defining the requirements and assigning tasks.

A Unified Modeling Language (UML) depiction of the software components involved in the current model study is given in Figure 4. The *User* is the District planner who seeks a solution to the water quality problem in Peoria Lake. To assist in the planner's problem solving process, the distribution of aquatic plants throughout the lake is required along with simulated changes resulting from the construction of a levee. The *User* creates two scenarios: one is a representation of a typical plant biomass distribution in any given year, and the other one is the plant biomass distribution in the same year—but with the levee present. A GIS database is constructed for the lake to present all spatial data. The data required for the GIS are obtained from an Oracle database. The data contained within the tables of this database are selected from the output files of RMA2, SED2D, and VALLA. Additional data, required to run the models, are obtained from the team

members from ERDC-CHL and ERDC-EL, and added to the georeferenced data for the lake from ERDC-CRREL.

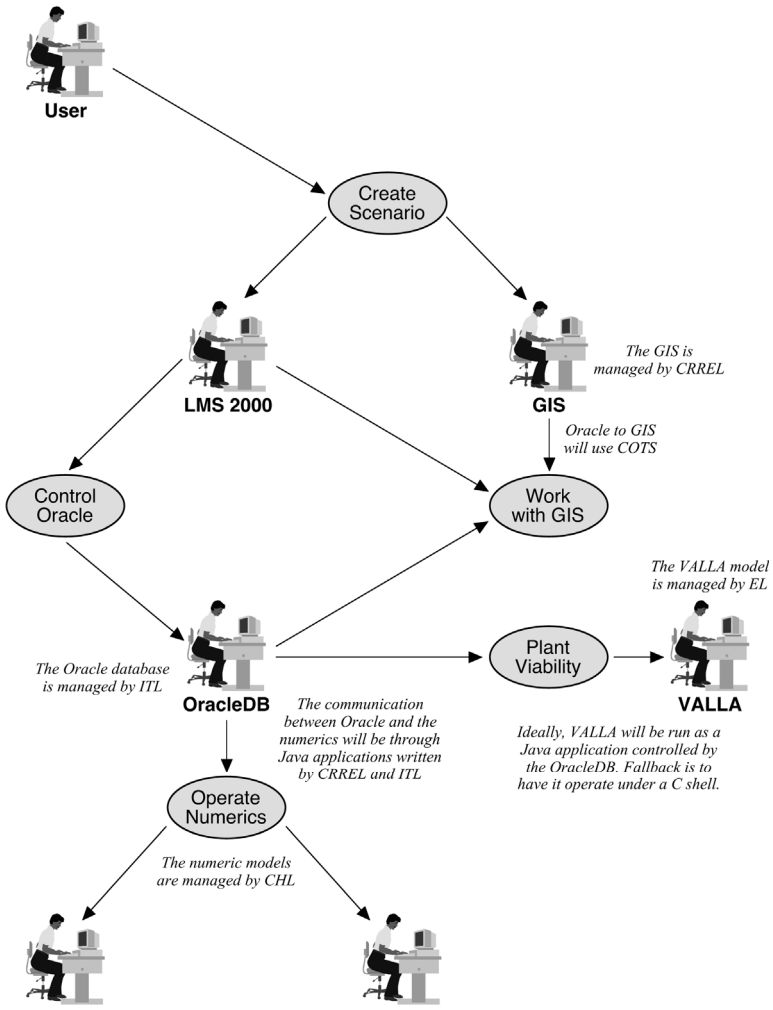

Figure 4. Unified Modeling Language (UML) diagram of the Peoria Lake Pilot Study.

Level II protocol architecture provides the flexible programming environment for numerical model results to interact with the analysis capability of the GIS. The *User* is not concerned with the operation of these models, but rather with the data presented through the GIS views. Under the client-server architecture of Protocol Level II, data flow between models and GIS is automated so that the *User* only needs to concentrate on the problem solution.

First, an attempt is made to run each model as it was originally programmed, i.e., the architecture is modified to suit the models rather than the models being modified to fit the system. Data sharing is automated as much as possible. However, some time-consuming activities remain, such as transforming data into the specific data formats or manipulations required by the legacy codes that are not readily automated, and the creation of computational grid generation required to run RMA2 and SED2D.

The sequence of events in the normal data flow between the programs is presented in Figure 5. The process begins with the *User* creating a modeling scenario through LMS2000. LMS2000 then controls the request to Oracle to begin coordinating the creation of data files and running of programs. Events involve obtaining GIS data, creating input files for all models, as well as reading all required output files from those model runs. In this pilot study, many of these transfers take place over the Internet as the models, database, and GIS are all geographically dispersed. This automatic coordination of distributed programming is a feature provided by the Level II protocols that enhance problem-solving capabilities. Once a model is connected through a Level II protocol, that connectivity can be easily reused for future solution of new problems.

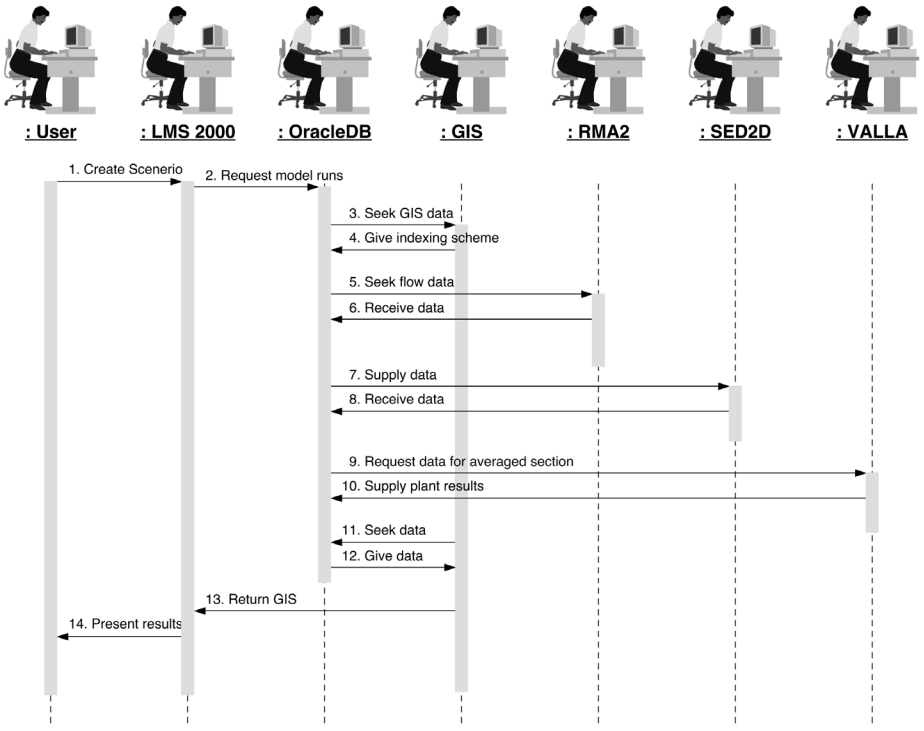

Figure 5. Peoria Lake Pilot Study sequence diagram.

3 PEORIA LAKE PILOT PROJECT STEPS

A series of multiple tasks that were to be performed by the team members has been presented in Figures 4 and 5 and they are briefly explained as Use-cases in Appendix B. These tasks and their mutual relationships, along with the laboratory entities responsible for them, are presented in Figure 6. For this pilot study, the tasks are distributed over four laboratories: ERDC-CHL, ERDC-CRREL, ERDC-EL, and ERDC-ITL. Defining requirements and creating the scenario are tasks in which all team members participated.

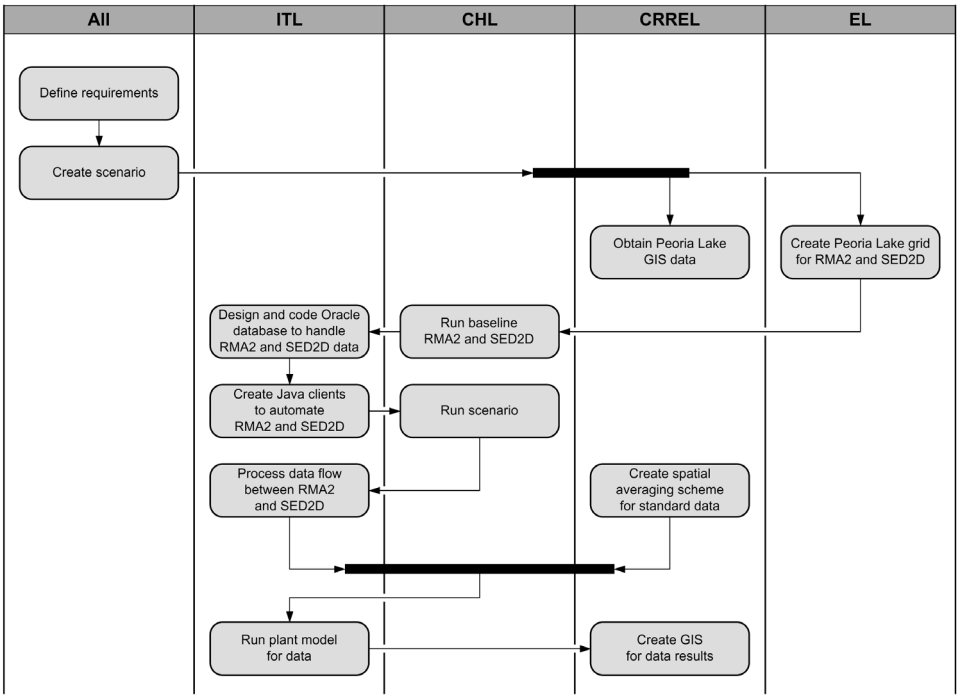

Figure 6. Peoria Lake Pilot Study tasks.

After the problem is defined, the data required by the models were collected by ERDC-EL, in collaboration with ERDC-CHL and ERDC-CRREL. Subsequently, the hydraulic models were run and tested using existing linkage methods to serve as a baseline reference for comparison with similar runs that used the Level II connection. Concurrent with the model testing, the GIS and Oracle database were constructed. The degree to which the plant viability model input data could be obtained from the GIS was determined by ERDC-CRREL and ERDC-

CHL. Communication between the plant viability model, and other models and tools, was explored by ERDC-EL and ERDC-CRREL, and established by ERDC-CRREL. The appropriate data fields were designed and mapped by ERDC-CHL and ERDC-ITL. The Oracle database was constructed, a linkage between database and GIS was established, and the required Java connections between the GIS, models, and database were programmed by ERDC-ITL and ERDC-CRREL. After completing all connections, the team tested the scenario. Results were collected, discussed, and iterative improvements made until all team members were satisfied with the validity of this linkage method for Level II Protocols.

4 PROBLEM SOLUTION

The Level II Protocol approach to problem solving is focused on data requirements. The approach uses existing models to obtain more (or other) data and facilitates data flows among models and tools to solve a problem. During the construction and running of the models, each model expert communicates with other team members to determine data requirements. Once the models are selected through careful analysis, the data requirements are all that need further discussion. It is up to the domain experts to assure that their models are operated correctly with the provided, requested data.

Plant Model

VALLA was operated in a mode that set all input parameters constant except for daily water depth and light extinction coefficient at noon. It was run in the default mode, i.e., starting from 233 tubers per m^2 lake surface, an initial tuber weight of 0.09 g DW per tuber, and 5.5 tubers concurrently formed per plant (Best and Boyd 2001a). The model was automatically run for each spatial zone of interest (node category; see Appendix C) using input files provided by the Oracle database described below. Simulated data on plant biomass and tubers resulting from these daily runs for each zone were collected and stored in the database for spatial analysis by the GIS.

Hydraulic Models

The scenario representing current conditions was obtained with a finite element mesh, shown in Figure 7, consisting of 5885 elements and 14,045 nodes. It was developed from hydrographical survey data collected by the Rock Island District in 1998–99. Nodal coordinates were referenced to the Illinois state plane coordinate system, West zone, NAD 1983. Bathymetry was referenced to the NGVD 1929 datum. The second scenario was created by modifying the existing conditions with insertion of a 10,000-m-long dike along the left descending side of the navigation channel (Fig. 8), to create a hypothetical plan condition intended to reduce suspended sediment concentrations in the southeastern two-thirds of the upper portion of Peoria Lake.

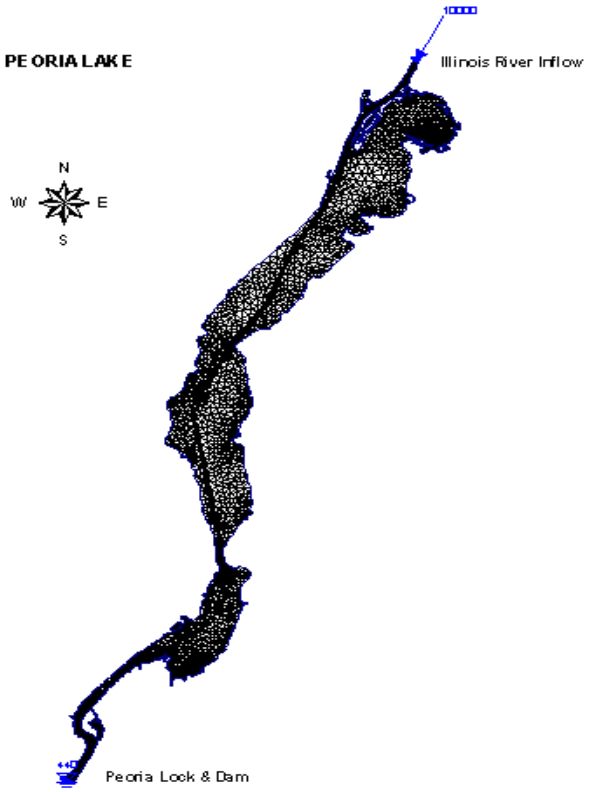

Figure 7. Peoria Lake Pilot Study RMA2/SED2D mesh.

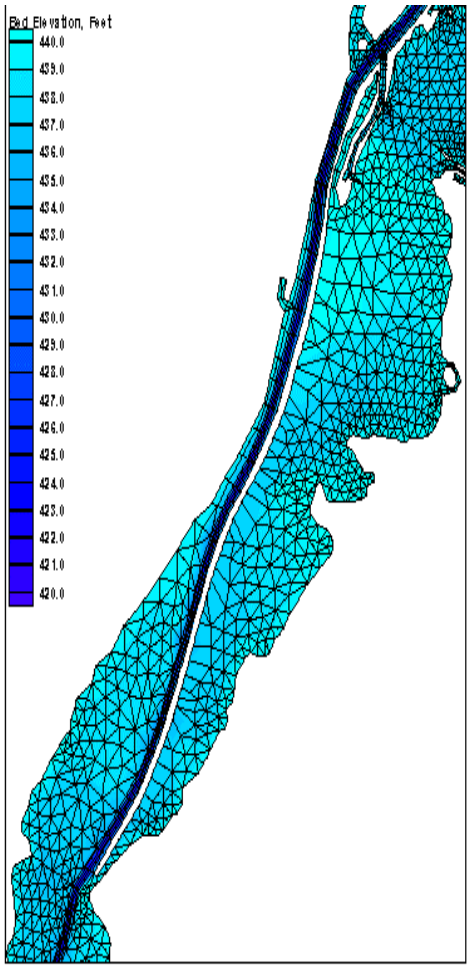

Figure 8. Grid of Upper Peoria Lake with levee, including bathymetry data used by RMA2 and SED2D.

In addition to mesh geometry, the hydrodynamic model requires input data describing roughness and turbulent exchange coefficients and initial and boundary conditions. A review of available field data did not produce sufficient information to estimate the roughness coefficient, Manning's n-value. A global Manning's n-value of 0.025 was selected as representative of the relatively smooth, fine-grained bed observed in the river navigation channel. A Manning's n-value of 0.04 was selected as representative of the shallow waters outside the navigation channel. Turbulent exchange coefficients were automatically set during the simulation to approximate a Peclet number of 10. This value provided a reasonable balance between model stability and numerical diffusion.

For hydrodynamic model simulations of the summer 1996 hydrograph, two types of boundary conditions were required. At the downstream boundary of

Peoria Lake, an observed stage hydrograph (Fig. 9) was specified. At the upstream boundary, water flow and sediment concentration were specified (Fig. 10). The initial flow-field was developed by computing a steady flow solution corresponding to the first set of inflow boundary conditions used in flood simulation. Hydrodynamic and sedimentation calculations were performed using one hour time-steps. The TABS-MD marsh porosity option was used to simulate wetting and drying of the mesh during the simulation.

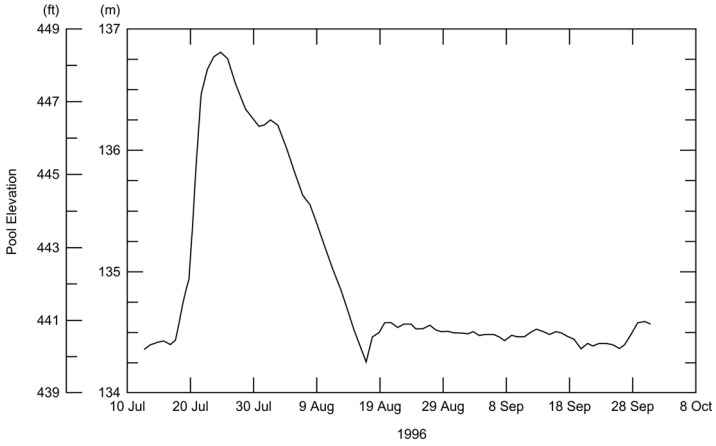

Figure 9. Observed downstream 1996 hydrograph.

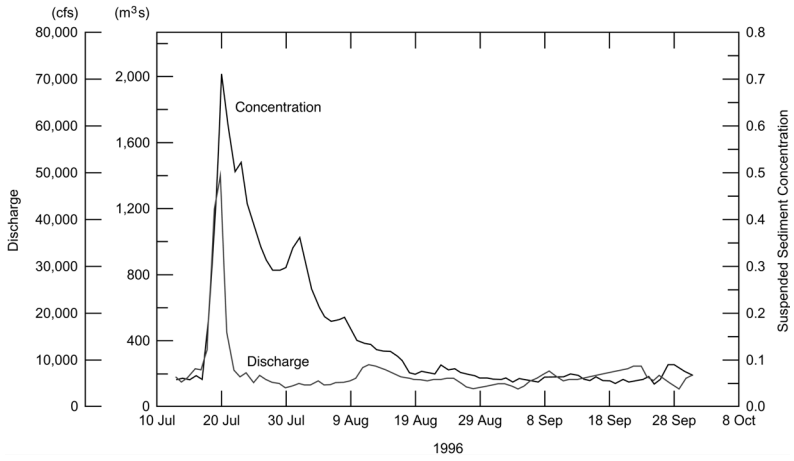

Figure 10. Hydrograph for water discharge and sediment concentration at the point of inflow.

GIS Model

GIS data were developed for Peoria Lake. Base data were provided by the Rock Island District and included shoreline, river miles, ortho-photos, bathymetry, and land cover themes. The general procedures described below were used to access, manipulate, and return data within the Peoria Lake Pilot Study framework. Detailed steps used to process the data are provided in Appendix C. However, it should be noted that there may be several variations to the process described here that could be used to achieve the same end result. Once the data were loaded into the GIS, its spatial analysis capabilities were used in three different ways to:

- Assist in data reduction activities.
- Visualize model input/output.
- Analyze results.

Table 2. Node subsection.

Attributes of Nodes1.shp

Shape	LOCATION_ID	X_COORD	Y_COORD	Z_COORD	NODE_NUM	CONCEN	DELBED	WATER_DEPTH	x_feet	y_feet
Point	1	745936.875000	440766.593750	134.112000	1	0.058121	-0.002036	0.262468	2447292.0776063	1446080.2637728
Point	1	745948.312500	440755.906250	134.075424	2	0.058120	-0.002034	0.297547	2447329.6020994	1446045.1999022
Point	1	745959.750000	440745.218750	134.038849	3	0.058117	-0.002032	0.332627	2447367.1265925	1446010.1360316
Point	1	745970.750000	440735.406250	133.276855	4	0.058113	-0.002029	1.094347	2447403.2157225	1445977.9428872
Point	1	745981.750000	440725.562500	132.514847	5	0.058106	-0.002032	1.856068	2447439.3048525	1445945.6472169
Point	1	746002.750000	440713.375000	131.184402	6	0.058093	-0.002028	3.186599	2447508.2022825	1445905.6621012
Point	1	746023.750000	440701.156250	129.853943	7	0.058081	-0.002029	4.517129	2447577.0997125	1445865.5744597
Point	1	746044.812500	440688.906250	129.389130	8	0.058072	-0.002029	4.981971	2447646.2021944	1445825.3842922
Point	1	746065.750000	440676.718750	128.924316	9	0.058068	-0.002029	5.446812	2447714.8945725	1445785.3991766
Point	1	746086.687500	440664.531250	127.795029	10	0.058069	-0.002029	6.576116	2447783.5869506	1445745.4140609
Point	1	746107.687500	440652.343750	126.665741	11	0.058076	-0.002029	7.705420	2447852.4843806	1445705.4289453
Point	1	746128.625000	440640.156250	128.467102	12	0.058088	-0.002029	5.904108	2447921.1767587	1445665.4438297
Point	1	746149.625000	440627.968750	130.268478	13	0.058104	-0.002029	4.102796	2447990.0741887	1445625.4587141
Point	1	746170.687500	440615.750000	132.190247	14	0.058119	-0.002029	2.191519	2448059.1766706	1445585.3710725
Point	1	746191.625000	440603.531250	134.112000	15	0.058131	-0.002046	0.280242	2448127.8690488	1445545.2834309
Point	1	746204.500000	440593.906250	134.075424	16	0.058137	-0.002040	0.307278	2448170.1097350	1445513.7054422
Point	1	746217.312500	440584.250000	134.038849	17	0.058141	-0.002036	0.334314	2448212.1453694	1445482.0249275
Point	1	746235.437500	440579.406250	134.075424	18	0.058144	-0.002037	0.298679	2448271.6104131	1445466.1334072
Point	1	746253.500000	440574.625000	134.112000	19	0.058144	-0.002039	0.263043	2448330.8704050	1445450.4469387
Point	1	746023.312500	440816.000000	134.038849	20	0.058093	-0.002033	0.333615	2447575.6643494	1446242.3572800
Point	1	746086.937500	440886.843750	134.038849	21	0.058053	-0.002033	0.334603	2447784.4071581	1446474.7835803

Because the "nodes" were the link to all attribute information used in all models and the GIS, the NODE locations (x, y) and associated attributes were obtained from the Oracle database, which had been populated following the RMA2 and SED2D model runs. The ArcView SQL Connect option was used for making the database connection. The data were then queried from the appropriate table with the SQL statement 'select * from table_name'. The selected set was saved as a DBF file in ArcView and imported as a "Theme" by using the "add event theme" option. These processes required the Oracle 8i Client software (Net8), and ArcView 3.X. Table 2 shows a subset of the nodes attribute table.

The hydrodynamic models, like many finite elements models, generate vast quantities of data. We decided to perform a spatial analysis of the lake and subdivide it into a few similar zones, based on assumed large-scale spatial homogeneity (patchiness). This was done for two reasons. First, the precision of the VALLA model did not merit the fine spatial resolution required for the hydrodynamic models. Second, nodes defined by the SED2D model are optimized for hydraulic and sediment computations (and the data are two-dimensional) and might not be optimal for the VALLA model (which is zero-dimensional).

The sediment concentration information at the 14,045-node mesh was redefined to a smaller number of cells by a four-step GIS spatial analysis process:

1. Water depth classes were generated for the study area to help in reducing the initially large amount of data and to improve data visualization. First, a water depth surface was interpolated by generating a triangular area network (TIN) using the depth attribute. The TIN was then converted into a GRID and the data classified into depth categories (in this case n=6). This step required ArcView 3D Analyst and Spatial Analyst.

2. Next, a depth category (1–6) was assigned to each NODE by performing a Spatial Join of the node attribute table and the depth category attribute (Gridcode) in the depth GRID. Table 3 shows a subset of the node data referenced by depth category.

Table 3. Subset of the node data referenced by depth category.

Attributes of Nodedep_intersect.shp

Shape	NODE_NUM	Id	Gridcode
Point	1	5070	6
Point	2	5070	6
Point	3	5070	6
Point	4	5084	3
Point	5	5084	3
Point	6	5115	1
Point	7	5115	1
Point	8	5115	1
Point	9	5115	1
Point	10	5115	1
Point	11	5115	1
Point	12	5115	1
Point	13	5115	1
Point	14	5115	1
Point	15	5118	5
Point	16	5122	6
Point	17	5122	6
Point	18	5122	6
Point	19	5122	6

3. "Spatial averaging zones" were then generated. These zones were defined relative to large-scale environmental homogeneity (main channel, right

over-bank, and left over-bank) in approximately 2- to 4-mile-long (3.2- to 6.4-km-long) reaches for a total of 25 zones. This process assumed that the local environment for points in a given spatial zone within a given depth category would be similar enough that the VALLA model could be averaged across that set of points. The main benefit of this process was to reduce the amount of data for the VALLA model. Figure 11 shows the resulting 25 zones obtained by this spatial stratification.

Figure 11. Twenty-five zones subsection of Peoria Lake from GIS analysis of shared depth and spatial characteristics.

Table 4. Sample of the nodes assigned to a zone.

Attributes of Nodedepzones_inters.shp

Shape	NODE_NUM	Gridcode	zone
Point	1	6	4
Point	2	6	4
Point	3	6	4
Point	4	5	4
Point	5	1	10
Point	6	1	10
Point	7	1	10
Point	8	1	10
Point	9	1	10
Point	10	1	10
Point	11	1	10
Point	12	1	10
Point	13	1	10
Point	14	1	10
Point	15	5	3
Point	16	6	3
Point	17	6	3
Point	18	6	3
Point	19	6	3

4. The final step in the data reduction process was to assign nodes from the numeric mesh to the appropriate cell (assign zone category [1–25] to each NODE). This was done by doing a Spatial Join of the node attribute table and the spatial zone category attribute (zone) from the "Spatial Averaging Zone" layer. Table 4 presents a sample of the nodes assigned to a zone.

Additional spatial analyses were performed for the two model scenarios (base condition and with levee). The first run (LOCATION_ID =1) established the base conditions of Peoria Lake. The second run (LOCATION_ID=2) considered the case where the levee was present.

As described above, six water depth categories were defined by the GIS, ranging from 0 to 5 ft (0 to 1.5 m) and each node was assigned a depth category. Sediment concentration, water depth, and bed elevation were then spatially averaged to generate a single set of VALLA input for each cell. The VALLA model was then run to provide data on plant biomass and number of tubers. Following the VALLA run the GIS again queried the database to access the newly populated plant data. The GIS was then used to reveal the spatial changes that occurred in sediment concentration and plant distribution.

A final two-step procedure was followed to analyze the hydrodynamic data from each scenario and generate a smaller number of data runs for VALLA. The first step was to generate an attribute for each node that identified unique value combinations (case ID) of the depth and spatial zone attributes. This creates a scheme to classify points according to shared depth and spatial characteristics (e.g., Depth Category 6 and Spatial Averaging Zone 19 form the unique combination identified as Case 78, of which there are 219 such points—see Table 5). The XTools Extension to ArcView was used to generate this new attribute using the "Frequency Table" option. The unique combinations of depth and spatial zone (82 cases in this example) reduced the number of points that needed to be run through the VALLA model from 14,045 to 82. Table 5 shows the unique spatial attributes based on depth and zone.

The second step was to generate an ASCII attribute file containing node number, depth category, spatial averaging zone, and case ID. This file was then provided to the Oracle database for use there and by the VALLA model.

Database

The Level II Protocol is concerned more with data modeling than numerical coding. The database is the central hub through which data flow is directed to and from the appropriate models. It handles the transport of data across the Internet and the communication among programs running on different platforms and operating systems. It provides a tool for verifying the format of the data files used by the models, as well as rapidly manipulating data among files. It also offers a convenient storage capability for the data mapping that can be reused for future problem solving tasks.

Table 5. Points classified according to shared depth and spatial characteristics.

Attributes of Nodedepzones_inters.shp

Shape	NODE_NUM	Gridcode	zone	sum_zone
Point	1	6	4	70
Point	2	6	4	70
Point	3	6	4	70
Point	4	5	4	56
Point	5	1	10	6
Point	6	1	10	6
Point	7	1	10	6
Point	8	1	10	6
Point	9	1	10	6
Point	10	1	10	6
Point	11	1	10	6
Point	12	1	10	6
Point	13	1	10	6
Point	14	1	10	6
Point	15	5	3	55
Point	16	6	3	69
Point	17	6	3	69
Point	18	6	3	69
Point	19	6	3	69

freq_img.dbf

Case	Frequency	Depth_cat	Zone
1	292	1	1
2	272	1	2
3	4	1	5
4	653	1	6
5	30	1	9
6	840	1	10
7	4	1	11
8	1	1	13
9	211	1	14
10	9	1	17
11	243	1	21
12	285	1	24
13	58	2	3
14	43	2	4
15	258	2	7
16	69	2	8
17	25	2	12
18	52	2	13
19	18	2	15
20	22	2	16
21	50	2	18
22	65	2	19
23	55	2	20
24	193	2	22
25	61	2	23
31	11	3	12
76	705	6	16
77	107	6	18
78	219	6	19
79	208	6	20
80	94	6	22
81	59	6	23
82	128	6	25

The first of two tasks handled by the relational database management system (RDBMS) is data modeling. Figure 12 is the entity relationship (ER) diagram of the Oracle database developed for the Peoria Lake Pilot Study. It contains eight tables from which all required analyses can be readily and remotely done by the GIS. The database performs data manipulation for required data input to external models, storage of model results and model data flow control. For example, the NODES table references all nodes used in the mesh for the hydrodynamic models (Fig. 7). The SPATIAL_INDEX table does the same for the scheme developed by the GIS (Fig. 11). The Oracle database can then perform rapid spatial averaging of values corresponding to the nodes in NODES that belong to a particular cell in SPATIAL_INDEX.

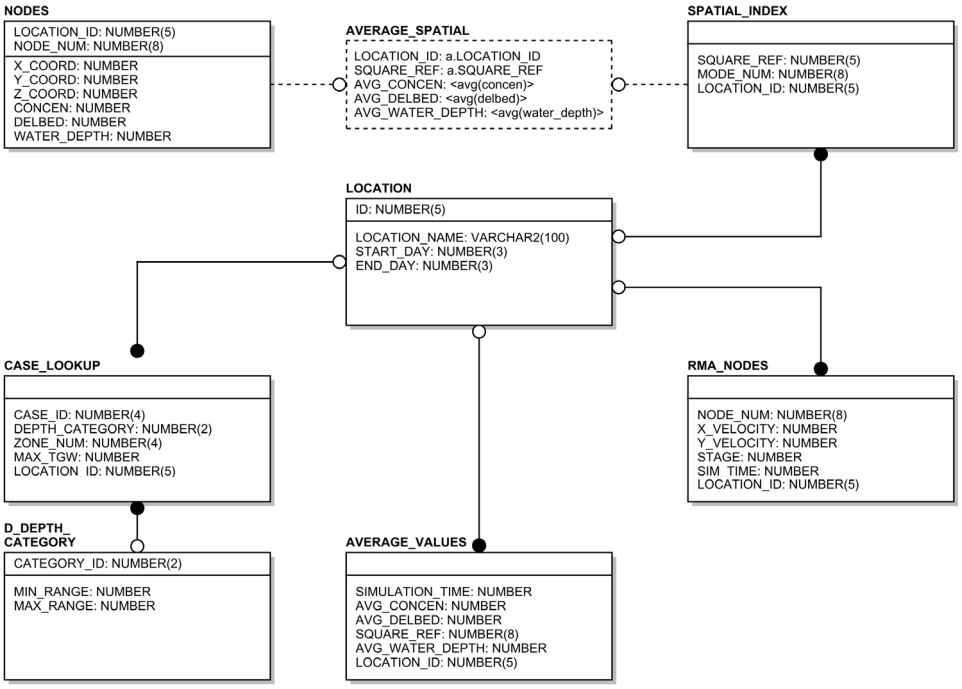

Figure 12. The entity relationship (ER) of data for the Peoria Lake Pilot Study.

The other task managed by the RDBMS is coordination of data flow and model execution. When dealing with legacy modeling systems, it is often necessary to provide tools to initiate and control operation of those programs. The Peoria Lake Pilot Study required several external Java applications that are controlled by the Oracle database and perform tasks that help automate the operation of those legacy systems. Appendix D lists the code for those models. They are the controller for RMA2 and SED2D called pltest, ReadRMADays to convert the raw RMA data into convenient vector data, and ThinDataSource that operates on the large numeric data files to thin the data according to the GIS averaging scheme. The other required external control code is in UploadFiles, which loads the input files on a remote system for VALLA, RunVALLA to run VALLA, and UploadVALLAFiles to capture the output from the VALLA program.

In operation, the Oracle database would read the appropriate data from the hydrodynamic codes and then create the necessary data input files for VALLA. It would then run VALLA for those files and retrieve the resulting output files for use by the GIS.

5 RESULTS

Following runs of RMA2, SED2D, and VALLA for the two scenarios representing the current and levee conditions, the output data for maximum biomass and number of vegetative plant propagules, or tubers, attributes were selected from the Oracle database as input to the GIS. To visualize these data and detect changes for the levee and non-levee runs the following procedure was used:

- Table from VALLA output was queried from Oracle.
- Attributes were JOINED using the common case_id attribute to assign VALLA attributes to the Node Attribute Table.
- A TIN was generated for each time period of interest for each attribute.
- The TIN was converted to a GRID.
- The GRID was reclassified into categories.
- The GRIDS for each date were overlaid and the difference between each grid cell was calculated.
- The differences were quantified.
- Output maps were produced.

The potential beneficial impact of any management scheme on the population of *Vallisneria americana* is indicated by both increased plant biomass and increased tuber population. An increased tuber population would tend to result in a larger plant population the following year. Figure 13 is the GIS representation of the predicted plant biomass distribution at the end of the growing season with and without a levee. Visual observation of the maps produced from both scenarios shows no significant improvement by the levee. However, the third map greatly assisted the interpretation by displaying the differences between the scenarios. It demonstrates a potential improvement in Upper Peoria Lake, particularly around the point of inflow of the Illinois River and close to the shoreline, and a concurrent, smaller, decrease in the southern part and along the shoreline of Lower Peoria Lake.

Figure 14 shows the three GIS views of tuber population. The behavior of the end-of-year tuber numbers confirmed the positive effects of the levee in Upper Peoria Lake, and the smaller negative effects in Lower Peoria Lake.

The GIS is capable of displaying the absolute changes in plant biomass and tuber population. However, it is important to realize that this analysis is not exact because neither the hydrodynamics from RMA2 nor the sediment results from

SED2D were verified against detailed field data. Nevertheless, this type of analysis does provide a convenient initial indication of trends for this pilot study. Given this big qualification to the reliability of these numbers, the GIS results from Figures 13 and 14 are listed in Table 6. The simulations indicate a small 6% increase in plant biomass and a large 58% increase in tuber population with the levee. The overall predicted tuber population increase is beneficial. It suggests that the following year should have an increase in plant biomass at the start of the growing season. That increased initial condition should also lead to an overall increase throughout the growing season.

Additional inspection of tuber population shows a decrease in the area near the river channel in the central and lower regions indicates with a levee. This implies that the following season would have fewer plants and, thus, lower biomass. There are some possible explanations for this trend. One is that the levee is slowing the water outside so that the sediment is not carried away. In time, sediment in these areas could settle and improve plant growth. Another is that, because the precision of these predictions is low, these small trends are really below reliable prediction limits. It must be stressed that this study was implemented to test the Level II Protocol approach to problem solving. It was not intended to be a thorough feasibility study for Peoria Lake. A more detailed sensitivity analysis of the decisions made and procedures adopted in this study could help answer the precision/reliability question.

The results of this Peoria Lake Pilot study indicate that a levee may offer improvement in the *Vallisneria americana* population resulting from increased tuber density at the end of the test year's growing season. It has to be noted, though, that in reality the water column remains turbid longer than in this modeled situation, because the predominant sediment class has a lower fall velocity and the sediment is resuspended in this wind-exposed lake. The significance of this tentative improvement must be further evaluated by the decision makers. Further simulation studies will require more detailed input data, such as hydrographs, sediment, climate, and the availability of plant propagules. Connectivity of the models used for the current pilot study remains, because of the existence of the Level II Protocols, and, thus, future analyses will be easier to do.

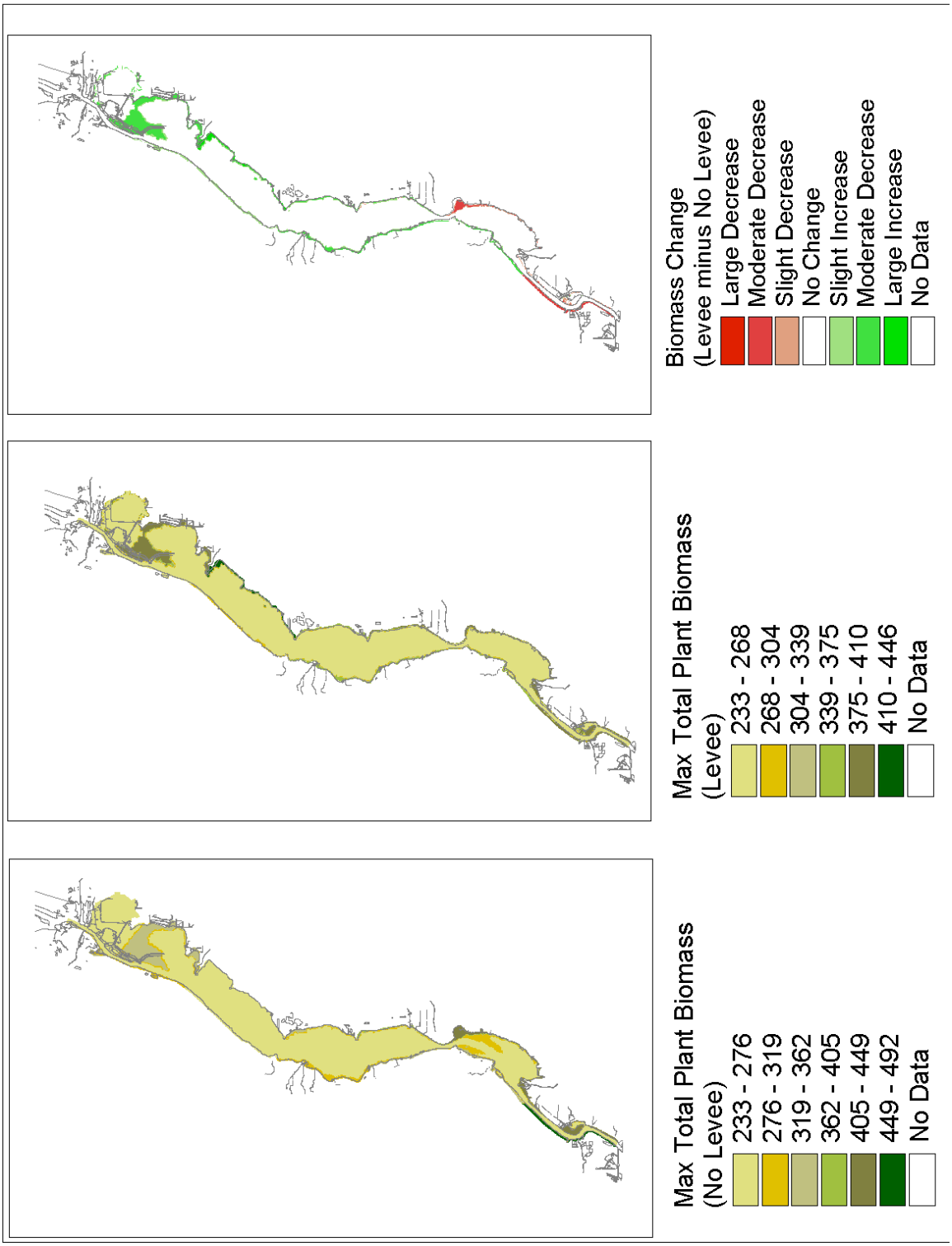

Figure 13. GIS representation of the simulated plant biomass distribution at the end of the growing season.

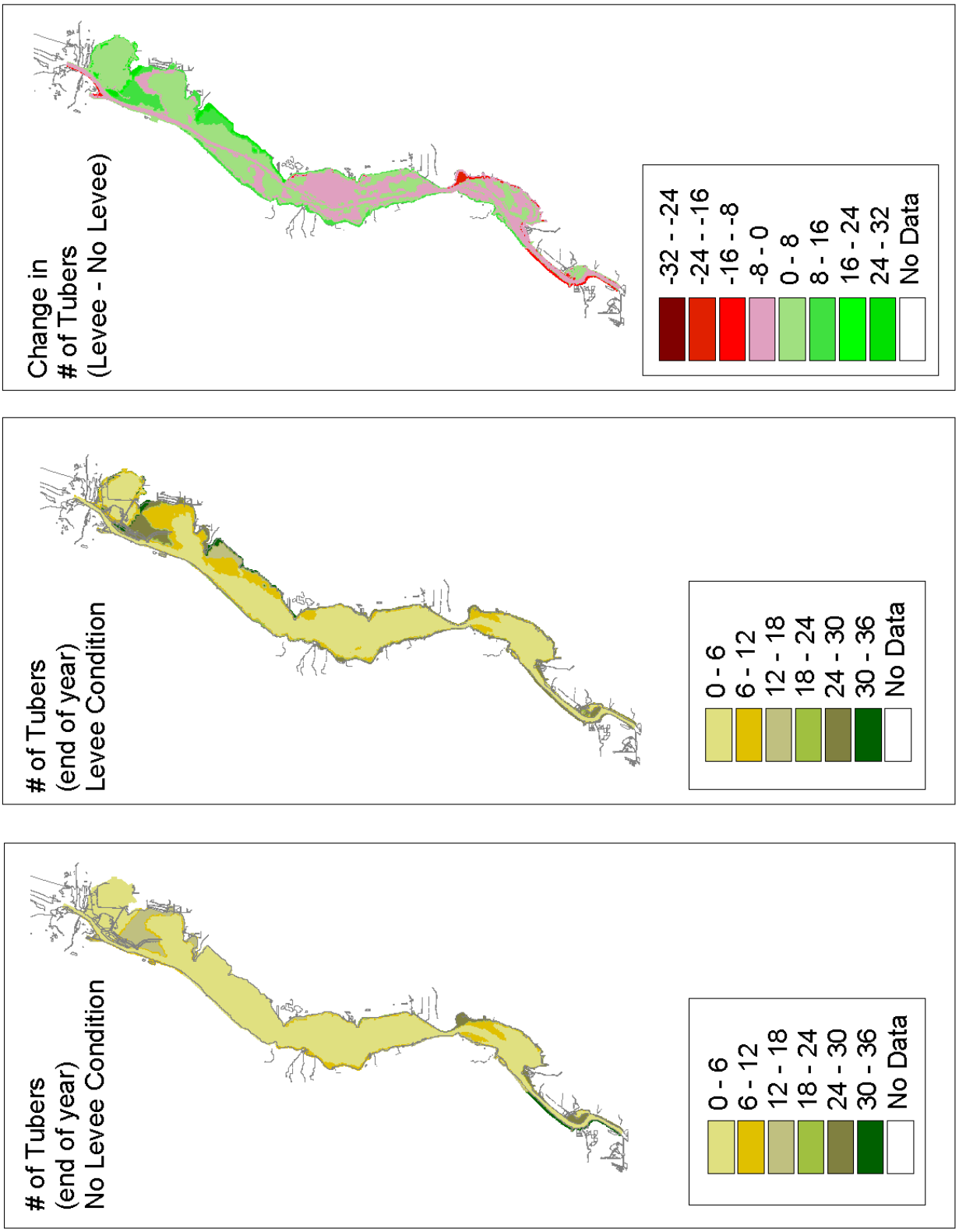

Figure 14. Simulated end-of-year tuber numbers for the Peoria Lake Pilot Study.

Table 6. GIS summary analysis of plant biomass and tuber population changes resulting from inclusion of a levee

TUBERS WITH NO LEVEE		TUBERS WITH LEVEE		Change
Sum:	416,711	Sum:	666,579	58%
Mean:	176	Mean:	205	16%
Maximum:	67,358	Maximum:	78,986	17%
Minimum:	0	Minimum:	0	
Variance:	3,210,110	Variance:	3,407,766	
Standard Deviation:	1,792	Standard Deviation:	1,846	
BIOMASS WITH NO LEVEE		**BIOMASS WITH LEVEE**		
Sum:	19,588,281	Sum:	20,883,021	6%
Mean:	6,902	Mean:	2,576	-62%
Maximum:	17,167,906	Maximum:	14,061,317	-18%
Minimum:	233	Minimum:	233	
Variance:	103,984,300,817	Variance:	24,645,288,502	
Standard Deviation:	322,466	Standard Deviation:	156,988	

6 SUMMARY

This pilot study demonstrated a method to address management questions about water quality, incorporating the Land Management System's Level II Protocol approach. Models, GIS tools, and data were integrated, and handled in a way that promoted data sharing as opposed to model recoding. This benefits end users in that they may use any model required to answer any question posed. It also allows rapid graphical visualization through GIS.

Level II Protocols require a team approach to problem solving. The team members were selected by matching subject matter experts to the problem's requirements. Consequently, their overlap in technical knowledge was small. Probably the most difficult part of the project was the ability to clearly communicate needs and issues among members specializing in different disciplines. In several instances, it appeared that similar functions could be conducted by a specific model, the GIS, or the Oracle database, and it often was not obvious which "tool" would process the specific request most efficiently. These areas of overlap will likely become fuzzier as technological advances incorporate greater functionality into each of these "systems."

Key to the success of this team approach was that all members succeeded in communicating terms of data needs, behaved as data consumers or producers, and were capable of efficiently communicating the details of their data in terms of fields and formats. This allowed the practical discussions on data flows required for the Protocol Level II approach.

REFERENCES

Best, E.P.H., and W.A. Boyd (2001a) A simulation model for growth of the submersed aquatic macrophyte Vallisneria americana Michx. US Army Engineer Research and Development Center, Vicksburg, Mississippi, ERDC/EL TR-01-5.

Best, E.P.H., and W.A. Boyd (2001b) VALLA (version 1.0): a simulation model for growth of Wild celery. US Army Engineer Research and Development Center, Vicksburg, Mississippi, ERDC/EL SR-01-1.

Best, E.P.H., W.A. Boyd, and W.F. James (in prep.). Simulation of potential persistence of Sago pondweed and American wildcelery in Peoria Lake, Illinois, using recent and historical data on light availability and suspended sediments. Hydrobiologia.

APPENDIX A: COMPARISON OF HYDRODYNAMIC MODELS THAT ALSO DEAL WITH SEDIMENTATION

Instructions:

You are comparing **5** products. The answers to each yes/no or choice question in the survey data for the selected product are provided in a side-by-side comparison format. Numbers listed with each item allow you to cross reference the data presented here with detailed survey data.

Product Key: Click the product name to go to the long report for that model

1. CH3D-SED - a three dimensional model for hydrodyna
2. HEC-6 (Hydrologic Engineering Center (HEC-6) Scour
3. RECOVERY - A contaminant model for surface water
4. RMA10-WES - a multi-dimensional hydrodynamic numer
5. SED2D - two dimensional sediment transport model

Part 1. General Information

Product:	1	2	3	4	5
1.A.3. Product Category:					
Decision Support System:	☐	☐	☐	☐	☐
Ecological/Landscape Model:	☐	☐	☐	☐	☐
Economic/Planning:	☐	☐	☐	☐	☐
Erosion/Sediment Transport:	☐	☐	☐	☐	☐
Hydrologic/Hydrodynamic:	☐	☐	☐	☐	☐
Model Development Environment:	☐	☐	☐	☐	☐
Water Quality/Contaminant Transport:	☐	☐	☐	☐	☐
Other:	☐	☐	☐	☐	☐
1.A.4. Applicable Ecological Regions:					
Arctic:	☐	☐	☐	☐	☐

Coastal:	☐	☐	☐	☐	☐
Desert:	☐	☐	☐	☐	☐
Estuary:	☐	☐	☐	☐	☐
Flood Plain:	☐	☐	☐	☐	☐
rassland:	☐	☐	☐	☐	☐
Ground Water:	☐	☐	☐	☐	☐
Lake:	☐	☐	☐	☐	☐
River:	☐	☐	☐	☐	☐
Forest:	☐	☐	☐	☐	☐
Savannah:	☐	☐	☐	☐	☐
Shoreline:	☐	☐	☐	☐	☐
Tundra:	☐	☐	☐	☐	☐
Wetland:	☐	☐	☐	☐	☐
Other:	☐	☐	☐	☐	☐
1.A.5 Model Components:					
Animal Behavior:	☐	☐	☐	☐	☐
Animal Population:	☐	☐	☐	☐	☐
Climate:	☐	☐	☐	☐	☐
Economics:	☐	☐	☐	☐	☐
Ground Water:	☐	☐	☐	☐	☐
Human Activities:	☐	☐	☐	☐	☐
Over Land Water:	☐	☐	☐	☐	☐
Plants:	☐	☐	☐	☐	☐
Surface Water:	☐	☐	☐	☐	☐
Weather:	☐	☐	☐	☐	☐
Other:	☐	☐	☐	☐	☐
Part 2. Product Applicability and Audience					
Product:	1	2	3	4	5

2.1 Target Audience:					
Decision Makers:	☐	☐	☐	☐	☐
Engineers:	☐	☐	☐	☐	☐
Model Developers:	☐	☐	☐	☐	☐
Operations Personnel:	☐	☐	☐	☐	☐
Planners:	☐	☐	☐	☐	☐
Policy Makers:	☐	☐	☐	☐	☐
Scientist:	☐	☐	☐	☐	☐
Software Specialists:	☐	☐	☐	☐	☐
Other:	☐	☐	☐	☐	☐
2.2 Computer Knowledge Required:					
Casual User:	☐	☐	☐	☐	☐
Experienced User:	☐	☐	☐	☐	☐
Programmer:	☐	☐	☐	☐	☐
2.4 Product Cost:					
Purchase Cost:	$0.00 __	$0.00 __	$0.00 __	$0.00 __	$0.00 __
Annual Cost:	$0.00 __	$0.00 __	$0.00 __	$0.00 __	$0.00 __
Part 3. General Product Features					
Product:	1	2	3	4	5
3.1 Assessment:	☐	☐	☐	☐	☐
3.2 Predictions:	☐	☐	☐	☐	☐
3.3 Alt. Analysis:					

	☐	☐	☐	☐	☐
3.4 Alt. Testing:					
	☐	☐	☐	☐	☐
3.5 Consensus:					
	☐	☐	☐	☐	☐
3.6 Comparisons:					
	☐	☐	☐	☐	☐
3.7 Adjacencies:					
	☐	☐	☐	☐	☐
3.8 Audit:					
	☐	☐	☐	☐	☐
Part 4. Technical Product Features					
Product:	1	2	3	4	5
4.1 Releveant Technologies:					
Deterministic:	☐	☐	☐	☐	☐
Empirical:	☐	☐	☐	☐	☐
Fuzzy Logic:	☐	☐	☐	☐	☐
Inductive Reasoning:	☐	☐	☐	☐	☐
Knowledge-Based System:	☐	☐	☐	☐	☐
Optimization:	☐	☐	☐	☐	☐
Simulation:	☐	☐	☐	☐	☐
Stochastic:	☐	☐	☐	☐	☐
Symbolic Logic:	☐	☐	☐	☐	☐
Other:	☐	☐	☐	☐	☐

4.4 Distributed Processing:					
	☐	☐	☐	☐	☐
4.5 Error Detection:					
	☐	☐	☐	☐	☐
4.6 Sensitivity Analysis:					
	☐	☐	☐	☐	☐
4.7 Incomplete Data:					
	☐	☐	☐	☐	☐
Part 5. Scale of Input & Output					
Product:	1	2	3	4	5
A. Inputs:					
5.A.5 Input Frequency:					
Each Seconds:	☐	☐	☐	☐	☐
Minutes:	☐	☐	☐	☐	☐
Hourly:	☐	☐	☐	☐	☐
Daily:	☐	☐	☐	☐	☐
Monthly:	☐	☐	☐	☐	☐
Annually:	☐	☐	☐	☐	☐
Other:	☐	☐	☐	☐	☐
B. Model Runs/Execution:					
5.B.2 Spatial Scale:					
Centimeters. or smaller:	☐	☐	☐	☐	☐

Meters:	☐	☐	☐	☐	☐
100's of Meters:	☐	☐	☐	☐	☐
Kilometers:	☐	☐	☐	☐	☐

5.B.3 Temporal Scale:

Seconds:	☐	☐	☐	☐	☐
Minutes:	☐	☐	☐	☐	☐
Hours:	☐	☐	☐	☐	☐
Days:	☐	☐	☐	☐	☐
Months:	☐	☐	☐	☐	☐
Years:	☐	☐	☐	☐	☐

5.B.4 Ecological Scale:

Individual:	☐	☐	☐	☐	☐
Species:	☐	☐	☐	☐	☐
Group:	☐	☐	☐	☐	☐
System:	☐	☐	☐	☐	☐

C. Outputs:

5.C.6 Graphical Formats:

Charts:	☐	☐	☐	☐	☐
Graphs:	☐	☐	☐	☐	☐
Maps:	☐	☐	☐	☐	☐
VRML:	☐	☐	☐	☐	☐
Other:	☐	☐	☐	☐	☐

5.C.7 Reports Available:

	☐	☐	☐	☐	☐

Part 6. System Status

A. Detailed Status:					
6.A.1 Status:	Operational	*(not provided)*	*(not provided)*	Operational	Operational
6.A.2 Release Date:	01-Mar-98	*(not provided)*	*(not provided)*	01-May-98	20-Apr-98
6.A.3 Next Release:	01-Jun-99	*(not provided)*	*(not provided)*	01-Jun-99	*(not provided)*
6.A.4 Source Code:	☐	☐	☐	☐	☐
6.A.5 Public Domain:					
6.A.6 Y2K Compliant:	☐	☐	☐	☐	☐
B. Platform Requirements:					
6.B.1 Operating System:	LINUX	*(not provided)*	*(not provided)*	UNIX	UNIX
6.B.2 Speed:	200 Mhz	*(not provided)*	*(not provided)*	Above 300 Mhz	200 Mhz
6.B.3 Min. RAM: :	128 MB	*(not provided)*	*(not provided)*	Above 128 MB	128 MB
6.B.4 Min Disk Space:	10 GB	*(not provided)*	*(not provided)*	Above 10 GB	1 GB

Part 7. System Documentation

A. Documentation:

7.A.1 On-Line Help:

☐ ☐ ☐ ☐ ☐

7.A.2 Reference Matl.:

☐ ☐ ☐ ☐ ☐

7.A.3 Other Support:

☐ ☐ ☐ ☐ ☐

B. Technical Support:

7.B.1 Hot-Line:

☐ ☐ ☐ ☐ ☐

7.B.2 Bulletin Board:

☐ ☐ ☐ ☐ ☐

7.B.3 Support Available:

☐ ☐ ☐ ☐ ☐

C. Training:

7.C.1 Classroom training:

☐ ☐ ☐ ☐ ☐

APPENDIX B: USE CASE ANALYSIS

Use Case Specification <Create Scenario>

Brief Description

This Use Case enables the *User* to select the numeric models from a list (an Oracle Master Form) to identify data sources and select the appropriate GIS implementation for the site.

Flow of Events

1. The Use Case begins with the *User* selecting the option to run a scenario from the Oracle Master Form

2. The *User* selects 1 of 2 options: (I) Use a previous scenario, or (II) Create a new scenario

I. Use Previous Scenario

I.1. The system retrieves and displays a list of existing scenarios

I.2. The *User* selects a scenario and receives a list of required resources

I.3. The system verifies that all required resources are available and submits the request to proceed to the GIS and Oracle Database

II. Create New Scenario

II.1. The system retrieves and displays a list of available models

II.2. The *User* selects the required models and the system displays a list of required resources

II.3. The *User* supplies the system with the required resources

II.4. The system verifies that all required resources are available and submits the request to proceed to the GIS and Oracle Database

Associations

Actors

The actor starting this Use Case is: *User*

Other Actors involved in this Use Case are: Oracle Database, GIS

Associations with other Use Cases: None

Association from other Use Cases: None

Preconditions

The *User* must be logged into the system for this Use Case to begin

Special Requirements

None

Use Case Specification <Operate Numerics>

Brief Description

This Use Case controls the communication between the Oracle Database and the numeric models RMA2 and SED2D.

Flow of Events

I. RMA2

I.1. The Use Case begins with the request reaching RMA2 to run the RMA2 Process Data

I.2. The RMA2 data input file has to be available either from a table in Oracle or from the *User*

I.3. RMA2 starts running either upon a request from Oracle Database call or *User*

I.4. RMA2 generates its output file

I.5. The Oracle Database receives a results file

I.6. The Oracle Database processes data (in the case where a previous scenario is run, no data are processed)

II. SED2D

II.1. The Oracle Database generates a request that reaches SED2D to run SED2D Process Data

II.2. The Oracle Database posts the appropriate data file to the proper URL

II.3. The Oracle Database posts a request to begin processing, which is in the form of a notification to the SED2D *User* that the data input file is available

II.4. The Oracle Database queries the appropriate URL for a results file

II.5. The Oracle Database receives the results file

II.6. The Oracle Database processes data

Associations

Actors

The actor starting this Use Case is:	Oracle Database
Actors also involved in this Use Case are:	RMA2, SED2D, *User*
Associations with other Use Cases:	None
Association from other Use Cases:	None

Preconditions

Required data files must be available on numeric clients or in Oracle

SED2D and RMA2 must be accessible to Oracle Database either through clients or entities on the Oracle server

Special Requirements

None

Use Case Specification <Working with GIS>

Brief Description

This Use Case coordinates the data exchange between the numerical results managed by the Oracle database and the GIS.

Flow of Events

1. The Use Case begins when the GIS requests action from the Oracle Database

2. A computational grid, that pertains to the demonstration site, with a suitable number of nodes to allow an acceptable (reasonably short) computational time, is selected. The selection is based on discussions between all team members. The nodes of this grid are grouped into a relatively low number of node categories, to each of which an average value of the SED2D output data will be assigned (after having been calculated). The average values equal the value of the geographic center of the grid cell category

Oracle compute plant biomass

Using the averaged sediment value obtained from 2.1.1, the plant biomass regression is applied to each daily value for each grid and another field or table is generated in Oracle

GIS Requests Data from Oracle

GIS sends request over Internet to Oracle database for data operation

Oracle receives request

Oracle processes query

Oracle returns results over the Internet to the GIS

Associations

Actors

The actors starting this uses case are:

OracleDB

GIS

Associations with other Use Cases

None

Association from other Use Cases

None

Preconditions

A scenario must be started in order for this use case to begin

Special Requirements

None

APPENDIX C: GIS DATA PROCESSING STEPS

NODE locations and attributes

Obtain NODE locations (x,y) and attributes from the Oracle database described below following the RMA2 and SED2D model runs.

Purpose

Input data into ArcView.

Requirements

Oracle 8i Client software (Net8), ArcView 3.X,

- Use ArcView SQL Connect option for database connection
- Query data from table with statement 'select * from table_name'
- Save as DBF file in ArcView
- Import table as a "Theme" by using "add event theme" option

Redefine sediment concentration

The sediment concentration information at the 14,000 nodes mesh was redefined to a smaller number of cells by a 4 step GIS spatial analysis process:

Generate water depth classes for the study area

Purpose. Data reduction and improved visualization

Requirements. ArcView 3D Analyst and Spatial Analyst

- Interpolate a water depth surface by generating a TIN using the depth attribute
- Convert to a GRID
- Classify into depth categories (in this case n=6)

Assign depth category (1-6) to each NODE.

- Purpose of this is data reduction, summarization, pattern detection

- Spatial join of Node attribute table and depth category attribute (Grid-code) in depth GRID
- Table 3 shows a subset of the node data referenced by depth category

Generate "spatial averaging zones"

There will be 25 zones: main channel, right overbank, left overbank in approximately 2–4 mile long reaches

Purpose. Data reduction for VALLA model.

Assumption: We are assuming that the local environment for points in a given spatial zone within a given depth category will be similar enough that the VALLA model can be averaged across that set of points.

- Make polygon theme to edit 'active'
- Go to Theme Menu and select Start Editing
- Digitize lines perpendicular to flow to split polygon features at desired locations
- Go to Theme Menu and Stop Editing (save results)

Figure 11 shows the resulting 25 zones obtained by the spatial analysis. The final step in the data reduction is then to assign nodes from the numeric mesh to the appropriate cell.

Assign zone category (1-25) to each NODE

- Purpose of this is data reduction, summarization, pattern detection.
- Spatial join of Node attribute table and spatial zone category attribute (zone) from Spatial Averaging Zone layer.
- Table 5 presents a sample of the nodes assigned to a zone.

Additional spatial analysis

Additional spatial analysis was performed for the two model scenarios:

- The first run (LOCATION_ID =1) established the base conditions of Peoria Lake.
- The second run (LOCATION_ID=2) considered the case of the levee being present.

Six water depth categories were defined by the GIS ranging from zero to five feet. Each node was then assigned a depth category. Sediment concentration, water depth, and bed elevation were then spatially averaged to generate a single set of VALLA input for each cell. The VALLA model was then run to provide data on plant biomass and the number of tubers. The GIS then accessed the database to visualize the changes in sediment concentration and plant distribution.

Final two-step procedure

A final two-step procedure was followed to analyze the hydrodynamic data from each scenario and generate a smaller number of data runs for VALLA.

Generate an attribute for each node

Generate an attribute for each node that identifies unique value combinations (case ID) of the depth and zone attributes.

Purpose. Create a scheme to categorize points that share depth and spatial characteristics. (e.g. Depth Category 6 and Spatial Averaging Zone 19 form the unique combination identified as Case 78, of which there are 219 such points – see appropriate table)**.**

Requirements. XTools Extension to ArcView

- Open the Node attribute table.
- Select the "Table Frequency" choice in the Table XTools menu.
- In the first dialog box, select the field or fields for which you wish to compute a frequency (Gridcode and Zone).
- In the second dialog box, select the field or fields that you wish to summarize (Optional). (This option was not used.)
- The third dialog box asks if you wish to add the case item to the current table (Optional). Answer "Yes", and enter the new field name that you wish to give the case item.
- The last dialog box will ask you for the name and location that you wish to specify for the frequency table.
- The unique combinations of depth and spatial zone (82 cases in this example) will reduce the number of points that need to be run through the VALLA model from 14,049 to 82. Table 5 shows the unique spatial attributes based on depth and zone.

ASCII attribute file

An ASCII attribute file containing node number, depth category, spatial averaging zone, and case ID was then provided to the DBA for use in Oracle and by the VALLA model.

APPENDIX D: JAVA PROGRAMS USED FOR PEORIA LAKE PILOT STUDY

pltest

The primary program is pltest which controls the operation of both RMA2 and SED2D.

Program: pltest

Purpose: Process RMA2 binary solution file

Command: java -cp CLASSPATH\classes111b.zip pltest –a *filename.sol*

Open input file passed in command line.

Read RMADAYS.DAT. This is a text file created by the user to indicate the number of days in the simulation. These data are stored in the table LOCATION.

Reads binary file produced by RMA2. Extracts VELOCITY and STAGE data and stores the values in the table RMA_NODES.

Program: pltest

Purpose: Process SED2D binary file

Command: java -cp CLASSPATH\classes111b.zip pltest –a *filename.cd*

Opens input file passed in command line.

Reads binary file produced by SED2D. Extracts geometry and node data and stores the values in the table NODES.

For each time step and node number, the CONCENTRATION, BED ELEVATION, and STAGE are extracted and stored in the table NODES.

The average values for CONCENTRATION, BED ELEVATION, and WATER DEPTH are calculated using the view AVERAGE_SPATIAL. The resulting averages are stored in the table AVERAGE_VALUES.

For every record for a case in CASE_LOOKUP, a file is created that contains the WATER DEPTH and LIGHT EXTINCTION COEFFICIENT. These are input files for VALLA and must be stored in the VALLA program directory.

The CONTROL.DAT file contains the files that serve as input and output for VALLA.

The VALLA program is then executed.

11. For each case, the file is read to extract MAXIMUM PLANT BIOMASS and the NUMBER OF TUBERS at the end of the year. These values are then stored in the table CASE_LOOKUP.

```
/**
   CREATED      DATE         COMMENTS
   GDR        27-MAR-00       CREATION
   GDR                        24-APR-00                    added code to pull number of nodes
                              and elements from RMA file
            GDR        25-APR-00       changed some hard coded node numbers to
                                                                        the variable tnNodes
   GDR                        04-MAY-00                    code clean-up, position, comments, etc.
   GDR                        17-MAY-00                    added code to run VALLA model
   GDR                        18-MAY-00                    added code to output a MODEL?.DAT file
   GDR                        18-MAY-00                    added code to output a CONTROL.DAT file
   GDR                        23-MAY-00                    added code to update database with VALLA OUTPUT

DESCRIPTION: THIS PROGRAM INPUTS TWO PARAMETERS FROM THE COMMAND LINE
            -a <filename> <location_id>
            THE a PARAMETER SPECIFIES A FILENAME TO OPEN AND READ FROM
  AND a LOCATION_ID to add
*/
import java.lang.*;
import java.sql.*;
```

```
import java.io.*;
import java.math.*;
import java.util.*;
import oracle.sql.*;
import oracle.jdbc.driver.*;
public class Pltest {
public static int LOCATION_ID=1;
static final String newline = System.getProperty("line.separator");

public static void main(String[] args) {
  try {
    if (args.length != 3) {
       System.out.println("usage: ");
       System.out.println("  PlTest -a [filename] [location_id]");
       System.out.println("  -a adds file to database");
       System.out.println("  [filename] name of file (with path)");
       System.out.println("  [location_id] location number");
    } else if (args[0].equals("-a")) {
      // add File
      String tsInfile = args[1];
      LOCATION_ID = Integer.parseInt(args[2]);
```

```
int tiStrLen = tsInfile.length();
if (!((tsInfile.substring(tiStrLen-3, tiStrLen)).equalsIgnoreCase("sol") ||
        (tsInfile.substring(tiStrLen-3, tiStrLen)).equalsIgnoreCase(".cd")) ) {
    System.out.println("This Function only uses .cd or .sol files.");
    System.exit(0);
}
int tiSrchRec;
int tiLastRec;
         Class.forName("oracle.jdbc.driver.OracleDriver");
ThinDataSource toDataSource = new ThinDataSource();
toDataSource.setServerName("wescs11.wes.army.mil");
toDataSource.setPortNumber(1521);
toDataSource.setDatabaseName("MYTEST");
Connection toCon = toDataSource.getConnection("plowner", "renwolp");
toCon.setAutoCommit(false);
if ((tsInfile.substring(tiStrLen-3, tiStrLen)).equals("sol")) {
        tiLastRec=7;
                                int tiInt, tnElements, tnNodes;
        int tiOutint;
        int tiRecord=1;
        tiSrchRec=5;
```

```
        boolean tbeof=false;
        FileInputStream tofistream = new FileInputStream(tsInfile);
        BufferedInputStream tobistream = new BufferedInputStream(tofistream);
        DataInputStream todata = new DataInputStream(tobistream);
    //read record 1 length
    tiOutint =todata.readInt();
    // read model flag
    tiOutint =todata.readInt();
                                    // read version
    tiOutint =todata.readInt();
                                    // read number of nodes
    tnNodes = convertBytes(todata.readInt());
    // read number of elements
    tnElements = convertBytes(todata.readInt());
    //read record length
    tiOutint =todata.readInt();
    tiRecord=2;
        while (tiSrchRec != 7) {
                            while (tiRecord < tiSrchRec) {
                    tiOutint = todata.readInt();
                    tiOutint = convertBytes(tiOutint);
```

```
            todata.skipBytes(tiOutint+4);
            tiRecord++;
          }
          // Skipped to record of interest
               float tnSimTime=0.0F;
                    float[] tnXVelocity = new float[tnNodes];
                                   float[] tnYVelocity = new float[tnNodes];
                                   float[] tnStage = new float[tnNodes];
ReadRMADays reader = new ReadRMADays();
                                   Vector rmaDays = reader.readDays();
Enumeration e = rmaDays.elements();
                    PreparedStatement toPrepStmt = toCon.prepareStatement("INSERT INTO RMA_NODES " +
   " (LOCATION_ID, NODE_NUM, X_VELOCITY, Y_VELOCITY, STAGE, SIM_TIME) " +
                          " VALUES (?, ?, ?, ?, ?, ?) ");
                    int bytesRead;
int tiTmpInt;
                    while (e.hasMoreElements()) {
                      // input record length field
  String str = (String) e.nextElement();
  try {
               float nextDay = Float.parseFloat(str);
```

```
                    bytesRead=0;
// input # of bytes for this record
                                        tiInt = todata.readInt();
                    tiInt = convertBytes(tiInt);
tnSimTime = Float.intBitsToFloat(convertBytes(todata.readInt()));
                                        bytesRead+=4;
                                        while (nextDay != tnSimTime) {
          todata.skipBytes(tiInt+4);
                                                tnSimTime = Float.intBitsToFloat(convertBytes(todata.readInt()));
                                        }
System.out.println("Simulation Time: " + tnSimTime);
 System.out.println("");
//read number of nodes
tiTmpInt=convertBytes(todata.readInt());
//System.out.println("Number of Nodes: " + tiTmpInt);
                                        bytesRead+=4;
                    for (int t=0; t<tnNodes; t++) {
                                                tnXVelocity[t] = Float.intBitsToFloat(convertBytes(todata.readInt()));
                                                tnYVelocity[t] = Float.intBitsToFloat(convertBytes(todata.readInt()));
                                                tnStage[t] = Float.intBitsToFloat(convertBytes(todata.readInt()));
                                        }
```

```
                    bytesRead += tnNodes*4*3;
                  for (int t=0; t<tnNodes; t++) {
                        toPrepStmt.setInt(1, LOCATION_ID);
                        toPrepStmt.setInt(2, t+1);
                                                toPrepStmt.setDouble(3, tnXVelocity[t]);
                        toPrepStmt.setDouble(4, tnYVelocity[t]);
                                                toPrepStmt.setDouble(5, tnStage[t]);
                                                toPrepStmt.setDouble(6, tnSimTime);
                     toPrepStmt.execute();
                    }
                    toCon.commit();
    // input record length field
                                                todata.skipBytes((tiInt-bytesRead)+4);
    } catch (NumberFormatException ex)  {
                                                // catch a bad number in RMADAYS.DAT
   }
}
toPrepStmt.close();
tiSrchRec=7 ;
    todata.close();
    tobistream.close();
```

```
    tofistream.close();
            break;
    }
} // if for file type
        else {
    // .cd file
                        Statement tstStmt = toCon.createStatement();
                        ResultSet tstRS = tstStmt.executeQuery("SELECT * FROM NODES WHERE LOCATION_ID=" +
        LOCATION_ID);
    if (tstRS.next() == false) {
            tiLastRec=7;
                                int tiInt;
                int tiOutint;
                int tiRecord=1;
                tiSrchRec=6;
                boolean tbeof=false;
                FileInputStream tofistream = new FileInputStream(tsInfile);
                BufferedInputStream tobistream = new BufferedInputStream(tofistream);
                DataInputStream todata = new DataInputStream(tobistream);
                while (tiSrchRec != 8) {
                        while (tiRecord < tiSrchRec) {
```

```
                tiOutint = todata.readInt();
                tiOutint = convertBytes(tiOutint);
              todata.skipBytes(tiOutint+4);
              tiRecord++;
            }
            // Skipped to record of interest
                PreparedStatement toPrepNodes = toCon.prepareStatement("INSERT " +
   "INTO NODES (LOCATION_ID, X_COORD, Y_COORD, Z_COORD, NODE_NUM)" +
  " VALUES (?,?,?,?,?)");
                    int bytesRead;
int tiTmpInt;
                int tnNodes, tnElements;
bytesRead=0;
// input # of bytes for this record
tiInt = todata.readInt();
tiInt = convertBytes(tiInt);
//input # of nodes
   tnNodes = convertBytes(todata.readInt());
System.out.println("# of nodes=" + tnNodes);
bytesRead+=4;
//input # of elements
```

```
    tnElements = convertBytes(todata.readInt());
System.out.println("# of elements=" + tnElements);
bytesRead+=4;
                        float[] tnXCoord = new float[tnNodes];
                                                float[] tnYCoord = new float[tnNodes];
                                                float[] tnElev = new float[tnNodes];
for (int t=0; t<tnNodes; t++) {
    tnXCoord[t] = Float.intBitsToFloat(convertBytes(todata.readInt()));
    tnYCoord[t] = Float.intBitsToFloat(convertBytes(todata.readInt()));
                                                }
bytesRead += tnNodes*4*2;
//now skip over the next bytes until we get to the elevations
todata.skipBytes(tnElements*8*4 + tnElements*4);
bytesRead += tnElements*8*4 + tnElements*4;
for (int t=0; t<tnNodes; t++) {
    tnElev[t] = Float.intBitsToFloat(convertBytes(todata.readInt()));
                                                }
bytesRead += tnNodes*4;
for (int t=0; t<tnNodes; t++) {
    toPrepNodes.setInt(1, LOCATION_ID);
                toPrepNodes.setDouble(2, tnXCoord[t]);
```

```
                toPrepNodes.setDouble(3, tnYCoord[t]);
                toPrepNodes.setDouble(4, tnElev[t]);
                toPrepNodes.setInt(5, t+1);
                toPrepNodes.execute();
   toCon.commit();
}
toCon.commit();
// input record length field
todata.skipBytes((tiInt-bytesRead)+4);
//now at end of record 6
    // now start inputting record 7
                        float tnSimTime;
                        float[] tnConcen = new float[tnNodes];
                                        float[] tnDelBed = new float[tnNodes];
                        float[] tnWatDepth = new float[tnNodes];
                        PreparedStatement toPrepStmt = toCon.prepareStatement("UPDATE " +
    "NODES SET CONCEN=?, DELBED=?, WATER_DEPTH=? WHERE LOCATION_ID=?"
   + " AND NODE_NUM=? ");
                        PreparedStatement toCopyView = toCon.prepareStatement(" INSERT INTO"
    + " AVERAGE_VALUES (LOCATION_ID, SIMULATION_TIME, AVG_CONCEN, " +
   "AVG_DELBED, AVG_WATER_DEPTH, SQUARE_REF) SELECT ?, ?, " +
```

```
"AVG_CONCEN, AVG_DELBED, AVG_WATER_DEPTH, SQUARE_REF FROM " +
"AVERAGE_SPATIAL WHERE LOCATION_ID=" + LOCATION_ID ) ;
                int y=0;
                int tnCount=0;
                        // changed loop to run indefinitely    09-May-2000  GDR
                try {
 while (true) {
                                y++;
                            // input record length field
                            bytesRead=0;
                            System.out.println("starting day=" + y);
                                                tiInt = todata.readInt();
                            tiInt = convertBytes(tiInt);
                                                tnSimTime = Float.intBitsToFloat(convertBytes(todata.readInt()));
                            System.out.println("Simulation Time: " + tnSimTime);
                            for (int t=0; t<tnNodes; t++) {
                                                        tnConcen[t] = Float.intBitsToFloat(convertBytes(todata.readInt()));
                                                }
                            bytesRead += tnNodes*4;
                            for (int t=0; t<tnNodes; t++) {
                                                        tnDelBed[t] = Float.intBitsToFloat(convertBytes(todata.readInt()));
```

```
                                        }
                    bytesRead += tnNodes*4;
                    //now skip over the bytes to get to the water depth
                    todata.skipBytes(tnElements*4 + tnElements*4 + tnNodes*4);
                  bytesRead += tnElements*4 + tnElements*4 + tnNodes*4;
                                        //skipped to the water depth
                    for (int t=0; t<tnNodes; t++) {
                                                tnWatDepth[t] = Float.intBitsToFloat(convertBytes(todata.readInt()));
                                        }
                    bytesRead += tnNodes*4;
System.out.println("before update");
                    for (int t=0; t<tnNodes; t++) {
                        toPrepStmt.setDouble(1, tnConcen[t]);
                                                toPrepStmt.setDouble(2, tnDelBed[t]);
                                                toPrepStmt.setDouble(3, tnWatDepth[t]);
                                                toPrepStmt.setInt(4, LOCATION_ID);
                            toPrepStmt.setInt(5, t+1);
                      toPrepStmt.executeUpdate();
                    }
toCon.commit();
System.out.println("after update");
```

```
                                                    toCopyView.setInt(1, LOCATION_ID);
                    toCopyView.setFloat(2, tnSimTime);
                                    toCopyView.execute();
                                    toCon.commit();
                                // input record length field
                                                    todata.skipBytes((tiInt-bytesRead));
                                    // puts byte pointer at end of record
                                    System.out.println("Finished day: " + y);
                                    tnCount++;
                                                }
                            } catch (EOFException e) {
        } catch (Exception e) {
        }
                    tiSrchRec=8;
        todata.close();
        tobistream.close();
        tofistream.close();
                break;
    }
// done importing data, now process VALLA files
  Statement simdays = toCon.createStatement();
```

```
                ResultSet simdaysRS = simdays.executeQuery("SELECT * FROM LOCATION " +
                        " WHERE ID=" + LOCATION_ID);
                int start_day=0;
                int end_day=0;
                if (simdaysRS.next()) {
                                        start_day = simdaysRS.getInt("START_DAY");
                  end_day = simdaysRS.getInt("END_DAY");
                }
                simdaysRS.close();
                simdays.close();
                                String DPTT;


// PatLookHere - LT = Light Extinction Coeff calculation
    String LES;
                PreparedStatement watstmt = toCon.prepareStatement("SELECT * FROM " +
                        "AVERAGE_VALUES WHERE LOCATION_ID=? AND SQUARE_REF=?"
                  + " ORDER BY SIMULATION_TIME");
              Statement depthcat = toCon.createStatement();
                // get resultSet with max number of indices of different cases
                Statement numIndices = toCon.createStatement();
                ResultSet numIndicesRS = numIndices.executeQuery("SELECT MAX(CASE_ID) " +
```

```
                    " AS MAX_INDICES FROM CASE_LOOKUP WHERE LOCATION_ID=" + LOCATION_ID);
int maxIndices=0;
if (numIndicesRS.next()) {
                        maxIndices = numIndicesRS.getInt("MAX_INDICES");
}
                    numIndicesRS.close();
numIndices.close();
for (int tiIndex=1; tiIndex<=maxIndices; tiIndex++) {
        DPTT = " DPTT=" + newline;
        LES = " LT=" + newline;
        // create control.dat file with new resX.dat and modelX.dat lines
                        writeControlFile( Integer.toString(tiIndex));
        // create modelX.dat from base file
        FileReader basefile = new FileReader("c:\\Wildcel\\model.tpl");
        BufferedReader inbuf = new BufferedReader(basefile);
                        FileWriter outfile = new FileWriter("c:\\Wildcel\\model" + tiIndex + ".dat");
        BufferedWriter outbuf = new BufferedWriter(outfile);
        boolean eof=false;
        while (!eof) {
                                String line = inbuf.readLine();
        if (line == null) {
```

```
                eof = true;
 } else {
outbuf.write(line);
outbuf.newLine();
 }
 }
 // now write DEPTH AND LES data
 watstmt.setInt(1, LOCATION_ID);
 watstmt.setInt(2, tiIndex);
 ResultSet watDepthRS = watstmt.executeQuery();
 ResultSet depthcatRS = depthcat.executeQuery("SELECT * FROM " +
                "D_DEPTH_CATEGORY A, CASE_LOOKUP B WHERE A.CATEGORY_ID=B.DEPTH_CATEGORY" +
 " AND B.CASE_ID=" + tiIndex + " AND B.LOCATION_ID=" + LOCATION_ID);
                        depthcatRS.next();
                        if (start_day>1) {
                                DPTT = DPTT + "   1., " + depthcatRS.getFloat("MAX_RANGE") + ", " + newline +
                                        "   " + (start_day-1) + "., "+ depthcatRS.getFloat("MAX_RANGE") +
", " + newline;
                                LES = LES + "   1., 2.8, " + newline +
                                        "   " + (start_day-1) + "., 2.8, " + newline;
                        }
```

```
// changes the LES line to multiply                                    06/19/2000                    GDR
//  the SED2D conecntration by 1000 to
//  convert it to parts per million
            while (watDepthRS.next()) {
            DPTT = DPTT + "   " + (start_day + (watDepthRS.getInt("SIMULATION_TIME")/24)) + "," +
                        watDepthRS.getFloat("AVG_WATER_DEPTH") + ", " + newline;
            LES = LES + "   " + (start_day + (watDepthRS.getInt("SIMULATION_TIME")/24)) + "," +
                        + (float) Math.exp(
                        (0.585*Math.log(watDepthRS.getFloat("AVG_CONCEN")*1000.0))-0.614) +
            ", " + newline;
            }
            if (end_day<365) {
                                    DPTT = DPTT + "   " + (end_day+1)+ "., " +
                        depthcatRS.getFloat("MAX_RANGE") + ", " + newline +
                                                "   365., "+ depthcatRS.getFloat("MAX_RANGE") +
           ", " + newline;
                                    LES = LES + "   " + (end_day+1)+ "., 2.8, " + newline +
                                                "   365., 2.8, " + newline;
                              }
            outbuf.write(DPTT.substring(0, DPTT.length()-2-newline.length()) + " ");
            outbuf.newLine();
```

```
            outbuf.write(LES.substring(0, LES.length()-2-newline.length()) + " ");
                                    outbuf.close();
            inbuf.close();
            // execute VALLA for these new files
            //RunVALLA valRun = new RunVALLA();
            //valRun.execute();
    }
    depthcat.close();
    watstmt.close();
    // write regular control file
    writeControlFile("");
/*
                        UploadVALLAFile toUpload = new UploadVALLAFile(toCon);
    // now import RES?.DAT files
            for (int tiIndex=1; tiIndex<maxIndices; tiIndex++) {
                                                toUpload.setFileName("c:\\Wildcel\\res" + tiIndex + ".dat");
        toUpload.setAreaNum(tiIndex);
                                                toUpload.importFile();
        // uncomment next two lines to
        // delete model?.dat files
        // File delFile = new File("c:\\Wildcel\\model" + tiIndex + ".dat");
```

```
                // delFile.delete();
                    }
            */
                                toCon.commit();
            toCon.close();
            System.out.println("Done.");
                    } else {
                                        System.out.println("Data for this Location already exists.");
            }
        tstRS.close();
                                tstStmt.close();
                    }
        } else {
                    // invalid switch
                                System.out.println("  " + args[0] + " is an invalid switch.");
          }
        }
        catch (EOFException eofex) {
                    // catches EOF
}
        catch (Exception e) {
```

```
            System.out.println(e);
            }
   // uncomment this line for production
   //System.exit(0);
}

            static int convertBytes(int piConvert) {
            /**
               CREATED       DATE          COMMENTS
               GDR         29-MAR-00         REVERSE BYTE ORDER OF piConvert
            */
            int tiOutInt;
            tiOutInt = (((piConvert << 24) >>> 24) << 24) | (((piConvert << 16) >>> 24) << 16) |
                                 (((piConvert << 8) >>> 24) << 8) | (piConvert >>> 24);
            return tiOutInt;
            }

            static void writeControlFile(String psCase) throws IOException {
            /**
   CREATED       DATE          COMMENTS
   GDR         16-MAY-00         write CONTROL.DAT FILE
```

```
*/
            FileWriter tofileout = new FileWriter("c:\\Wildcel\\CONTROL.DAT");
   tofileout.write("*-------------------------------------------" +
            "-------------------------*" + newline);
                        tofileout.write("* CONTROL.DAT file generated by Pltest.java " +
            "                        *" + newline);
                        tofileout.write("* File names to be used by FSE 2.1          " +
            "                        *" + newline);
                        tofileout.write("*                                           " +
            "                        *" + newline);
                        tofileout.write("* The input files (except FILEIR) may may used " +
            "in reruns.              *" + newline);
                        tofileout.write("* Up to five input data files may be used " +
            "(FILEI1-5)                  *" + newline);
                        tofileout.write("*-------------------------------------------" +
            "-------------------------*" + newline);
   tofileout.write(" " + newline);
                        tofileout.write("   FILEON = 'RES" + psCase + ".DAT'" + newline);
                        tofileout.write("   FILEOL = 'MODEL.LOG'" + newline);
                        tofileout.write("   FILEIR = 'RERUNS.DAT'" + newline);
                        tofileout.write("   FILEIT = 'TIMER.DAT'" + newline);
```

```
                        tofileout.write("    FILEI1 = 'MODEL" + psCase + ".DAT'" + newline);
    tofileout.write(" " + newline);
    tofileout.write("* FILEI2 = ' '            ! Second input data" +
            " file (not used)" + newline);
    tofileout.write("* FILEI3 = ' '            ! Third input data" +
            " file (not used)" + newline);
    tofileout.write("* FILEI4 = ' '            ! Fourth input data" +
            " file (not used)" + newline);
    tofileout.write("* FILEI5 = ' '            ! Fifth input data" +
            " file (not used)" + newline);
    tofileout.close();
    tofileout.close();
            }
}
```

Additional code to process numeric data

Two additional programs are required to process data from the numeric codes. The first is ReadRMADays that converts the raw RMA data into a more convienient vector. The other is ThinDataSource that performs the spatial decomposition of the large set node based data.

ReadRMADays

```
import java.util.*;
import java.io.*;
        /**
          CREATED      DATE        COMMENTS
          GDR        25-MAY-00       Class to load RMADAYS.DAT and return a
                                                  Vector of Days
        */
public class ReadRMADays {
        public ReadRMADays() {}
  public Vector readDays() throws IOException {
        Vector daysVector = new Vector();
    String str;
                try {
                FileReader fread = new FileReader("RMADAYS.DAT");
                        BufferedReader bread = new BufferedReader(fread);
              while ((str = bread.readLine()) != null) {
                                daysVector.add(str);
      }
    } catch (FileNotFoundException ex) {
                System.out.println("Error. RMADAYS.DAT not found.");
```

```
    }
    return daysVector;
            }
     }
```

ThinDataSource

```
import java.sql.*;
import oracle.jdbc.driver.*;
import oracle.sql.*;
public class ThinDataSource {
  protected String serverName;
  protected int portNumber;
  protected String databaseName;
  public ThinDataSource() {
  }
            public String getServerName() {
            return serverName;
  }
  public void setServerName(String psName) {
            this.serverName = psName;
            }
  public int getPortNumber() {
```

```
        return portNumber;
        }
  public void setPortNumber(int psPort) {
        this.portNumber = psPort;
  }
        public String getDatabaseName() {
        return databaseName;
  }
  public void setDatabaseName(String psDatabase) {
        this.databaseName = psDatabase;
        }
        public Connection getConnection() throws SQLException {
        return getConnection(null, null);
  }
  public Connection getConnection(String psUserid, String psPassword)
        throws SQLException {
    String url = "jdbc:oracle:thin:@" + getServerName() + ":" +
        getPortNumber() + ":" + getDatabaseName();
    return DriverManager.getConnection(url, psUserid, psPassword);
}
}
```

VALLA Control Code

The final set of external code required for the Peoria Lake Pilot operate VALLA. They are UploadFiles which load the input file required to run VALLA in the database, RunVALLA and UploadVALLAFiles which captures the output of VALLA.

UploadFiles

```
import java.sql.*;
import java.io.*;
import oracle.sql.*;
import oracle.jdbc.driver.*;
/**
   CREATED      DATE         COMMENTS
   GDR                          26-MAY-00                                creation

DESCRIPTION: THIS PROGRAM UPLOADS VALLA RESXX.DAT FILES TO DATABASE
*/
public class UploadFiles {
            public static void main(String args[]) {
     try {
                        Class.forName("oracle.jdbc.driver.OracleDriver");
       ThinDataSource toDataSource = new ThinDataSource();
```

```
toDataSource.setServerName("wescs11.wes.army.mil");
toDataSource.setPortNumber(1521);
toDataSource.setDatabaseName("MYTEST");
Connection toCon = toDataSource.getConnection("plowner", "renwolp");
            toCon.setAutoCommit(false);
                                int tiLocationID;
int tiNumberOfCases;
                                if (args.length == 2) {
      tiLocationID = Integer.parseInt(args[0]);
                                        tiNumberOfCases = Integer.parseInt(args[1]);
                        UploadVALLAFile toUpload = new UploadVALLAFile(toCon, tiLocationID);
                        // now import RES?.DAT files
                                for (int tiIndex=1; tiIndex <= tiNumberOfCases; tiIndex++) {
                                                        toUpload.setFileName("c:\\Wildcel\\res" + tiIndex + ".dat");
                             toUpload.setAreaNum(tiIndex);
                                                        toUpload.importFile();
                                }
                        toCon.commit();
```

```
                        toCon.close();
        } else {
                                System.out.println("UploadFiles: java UploadFiles <LOCATION_ID> <NUMBER_OF_CASES>");
        }
                                System.exit(0);
                        } catch (Exception e) {
            System.out.println(e);
     }
            }
}
```

RunVALLA

```
import java.io.*;
import java.net.*;
import java.sql.*;
public class RunVALLA {
String command;
            /**
```

```
   CREATED       DATE          COMMENTS
   GDR         18-MAY-00         Class to Run VALLA
*/
public RunVALLA() {
command = "";
}
public void execute() {
try {
String str;
                        Process p = Runtime.getRuntime().exec(command);
            BufferedReader br = new BufferedReader
                                    (new InputStreamReader(p.getInputStream()));
            BufferedWriter bw = new BufferedWriter
                                    (new OutputStreamWriter(p.getOutputStream()));
                        BufferedReader stdin = new BufferedReader
(new InputStreamReader(System.in));
            BufferedReader stderr = new BufferedReader
                                    (new InputStreamReader(p.getErrorStream()));
```

```
    char[] cbuf = new char[1];
boolean stdout=false;
while (true) {
                                        try {
    cbuf = new char[1];
                    // may need to input a line from stdin and write to bw
    try {
                    if (p.exitValue()==0) break;
      } catch (Exception e) {
      }
   if (stdout==true) {
    try {
                    if (p.exitValue()==0) break;
      } catch (Exception e) {
      }
      str = stdin.readLine();
      bw.write(str);
      bw.newLine();
```

```
            bw.flush();
          }
          if (br.ready()) while (br.ready() && (br.read(cbuf)) != -1 ) {
           System.out.print(cbuf);
            stdout=true;
          } else {
           stdout=false;
          }
                                                if (stderr.ready()) while (stderr.ready() && (stderr.read(cbuf)) != -1 )
                    System.out.print(cbuf);
                                                } catch (Exception e) {
           System.out.println(e);
                                        }
      }
                    }
                    catch (Exception e) {System.err.println("Valla Error: "+e.toString());}
           }
}
```

UploadVALLAFiles

```
import java.util.*;
import java.io.*;
import java.net.*;
import java.sql.*;
        /**
           CREATED      DATE         COMMENTS
           GDR        23-MAY-00      Class to upload a VALLA output file to db
                                     which saves the maximum value of the
                TGW and the end of year value for NDTUB
        */
public class UploadVALLAFile {
private String tsFileName;
private FileReader fread;
private BufferedReader bread;
private Connection toCon;
private int tiAreaNum;
private int tiLocationID;
```

```
        public UploadVALLAFile(Connection poCon, int piLocationID) {
    toCon = poCon;
    tiLocationID = piLocationID;
}
        public void importFile() throws FileNotFoundException, IOException,
                SQLException {
                boolean eof=false;
    PreparedStatement updateValues=null;
                fread = new FileReader(tsFileName);
    bread = new BufferedReader(fread);
    String str="";
    float tnTGW, tnNDTUB;
    while (str.indexOf(".")==-1) str=bread.readLine();
    tnTGW=0;
    String tsValue="";
    updateValues = toCon.prepareStatement("UPDATE CASE_LOOKUP " +
        "SET MAX_TGW=?, NDTUB=? WHERE CASE_ID=? AND LOCATION_ID=?");
                while (str != null) {
```

```
                                        // need to input the header of the file
                                        StringTokenizer token = new StringTokenizer(str, " ");
  int tiCount=1;
  while (token.hasMoreTokens()) {
                                                tsValue = token.nextToken();
                                                if (tiCount==2) {
                                                        if (Float.parseFloat(tsValue)>tnTGW)
                                tnTGW=Float.parseFloat(tsValue);
      }
    tiCount++;
  }
str = bread.readLine();
}
tnNDTUB = Float.parseFloat(tsValue);
// update database with values
                                updateValues.setFloat(1, tnTGW);
                                updateValues.setFloat(2, tnNDTUB);
updateValues.setInt(3, tiAreaNum);
```

```
        updateValues.setInt(4, tiLocationID);
                                        updateValues.execute();
        System.out.println("Area Num: " + tiAreaNum + "   TGW:" + tnTGW +
              "   NDTUB=" + tnNDTUB);
        updateValues.close();
                                bread.close();
                                        fread.close();
    }
    public void setFileName(String psFileName) {
                                tsFileName = psFileName;
    }
    public void setAreaNum(int piAreaNum) {
                                tiAreaNum = piAreaNum;
    }
}
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
September 2003	Technical Report	

4. TITLE AND SUBTITLE

Assessing Hydraulic Modifications on
Vallisneria americana in Peoria Lake, Illinois:
A Pilot Study Using Data Sharing Protocols to Integrate Legacy Models

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Patrick B. Black, Elly P.H. Best, E. Allison Newcomb, Terry J. Birkenstock,
William L. Boyt, Ronald Heath, and William F. James

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Engineer Research and Development Center
Cold Regions Research and Engineering Laboratory
72 Lyme Road
Hanover, NH 03755-1290

8. PERFORMING ORGANIZATION REPORT

ERDC/CRREL TR-03-18

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of the Chief of Engineers
Washington, DC

10. SPONSOR / MONITOR'S ACRONYM(S)

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

Available from NTIS, Springfield, Virginia 22161.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A pilot modeling study was conducted to assess the utility of implementing the Land Management System's (LMS) Level II Protocols for the efficient sharing of data among legacy models and GIS tools. The modeling was directed toward an investigation of potential population increase of the submersed macrophyte *Vallisneria americana* as a consequence of the construction of a hypothetical levee in the north–south direction in Upper Peoria Lake, Illinois. Numerical models were used to simulate the behavior of the lake in terms of hydrodynamics, sediment concentration, and aquatic plant biomass production for one growth season in 1996 from 1 May to 31 October. Two scenarios were modeled: 1) the current condition without levee, and 2) a hypothetical situation with a levee created in the north–south direction. The model results indicated that the presence of the levee would increase the maximum plant biomass of Vallisneria moderately to greatly in the northern part and along the shores of Upper Peoria Lake, but decreased it in the southern part and along the edges of Lower Peoria Lake. It also would allow the presence of considerably higher tuber numbers at the end of the growth season. The latter would enable a larger plant population to form the subsequent year. Thus, the results of this pilot study indicate that the creation of a levee in the north–south direction may lead to an increase in the Vallisneria population in Peoria Lake. It must be noted, however, that without the levee the water column remains turbid longer than in this modeled situation, because the predominant sediment class has a lower fall velocity and the sediment is resuspended in this wind-exposed lake. The integrated models and tools used in the study were readily available and merely required specific data in appropriate, model-specific formats. A Level II architecture was implemented that promoted efficient data exchange between models and GIS tools.

15. SUBJECT TERMS

Aquatic plant growth
Biomass dynamics
Environmental monitoring
Geospatial
Interoperability
Simulation modeling

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
U	U	U	U	97	