DOTAGWA: A CASE STUDY IN WEB-BASED ARCHITECTURES FOR CONNECTING SURFACE WATER MODELS TO SPATIALLY ENABLED WEB APPLICATIONS

Averill Cate, Jr., USDA Agricultural Research Service, University of Arizona, acate@email.arizona.edu
Darius Semmens, Hydrologist, USEPA Office of Research and Development, semmens.darius@ epa.gov
D. Phillip Guertin, Professor of Watershed Management, University of Arizona, phil@nexus.srnr.arizona.edu
David C. Goodrich, Research Hydraulic Engineer, USDA/SWRC, Tucson, AZ, dgoodrich@tucson.ars.ag.gov

Keywords: Spatially enabled web application, surface water, decision support systems, environmental science, best management practices, landscape change and monitoring, land-use change

Abstract

The Automated Geospatial Watershed Assessment (AGWA) tool is a desktop application that uses widely available standardized spatial datasets to derive inputs for multi-scale hydrologic models (Miller et al., 2007). The required data sets include topography (DEM data), soils, climate, and land-cover data. These data are used to develop input parameter files for two USDA-ARS watershed runoff and erosion models: the Kinematic Runoff and Erosion Model (KINEROS2, Smith et al., 1995; http://www.tucson.ars.ag.gov/kineros) and the Soil and Water Assessment Tool (SWAT, Arnold et al., 1994; http://www.brc.tamus.edu/swat/). AGWA has proved to be a useful tool for many different applications. Not all potential users, however, have access to the geospatial data or software required to process it and run tools like AGWA. In addition, some potential users recognized the value in being able to use the application, but did not have adequate technical training to gather and process the necessary data and run the application through a geospatial information system (GIS) software platform. A Web-based version of AGWA, DotAGWA, was developed to address these issues and cater to a wider potential user audience. This paper describes the design and structure of the DotAGWA application and discusses important findings related to issues and problems that emerged during application development. In particular, important issues emerged related to configuring a system that would connect surface water models, originally intended as desktop applications, to a spatially enabled web application. Some of these issues include input and output file management for model runs when models are executed from the web-client to the server's operating system, configuring the systems spatial and non-spatial data requirements in a web server environment, and designing an extensible or at least re-usable system architecture.

The United States Environmental Protection Agency through its Office of Research and Development partially funded and collaborated in the research described here under assistance agreement DW12922094-01-0 to USDA-ARS. It has been subjected to Agency review and approved for publication. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

1. INTRODUCTION

AGWA, originally released in 2002, was developed using the Environmental Systems Research Institute's (ESRI) ArcView 3.x and its Avenue scripting language. An updated version for ESRI's ArcGIS 9.x platform, AGWA 2.0 is being released in 2007. DotAGWA was designed in concert with the AGWA 2.0 desktop version and these two versions share the core GIS functionality. The goal of this paper is to provide information documenting the DotAGWA development process and findings that were and are being discovered as the application develops.

DotAGWA is an Internet based application connecting hydrologic models to a map-based interface. Users can develop management scenarios or define land use characteristics which are combined with supporting data sources to parameterize and run the hydrologic models connected to the application. Applications similar to DotAGWA do exist. The Spatial Decision Support System for Watershed Management developed at Purdue University has watershed processing features similar to those used in DotAGWA. The Purdue System also allows users to run hydrologic impact and sedimentation models for a delineated watershed.

Researchers at Purdue University have also developed an online web application that allows users to run the Watershed Erosion Prediction Project (WEPP, http://milford.nserl.purdue.edu/wepp/weppV1.html). WEPP is a simulation that estimates erosion using climate, land management, soils, and topography as inputs. Once inputs are defined through web-based input forms the user can run the model and download the output. However, this application does not allow the user to develop input data by letting the user define a project area through a map-based interface. DotAGWA has unique features that facilitate interaction with hydrologic simulation models. By obviating the need for data compilation and processing and enforcing a controlled process for user interaction with the model, DotAGWA creates an environment where people with little or no training in hydrologic modeling and GIS can use this powerful technology for practical decision support. Via a spatially enabled interface, users can develop management scenarios, parameterize and run the hydrologic models, and compare model results.

The features available in DotAGWA are also designed to help users share and visualize data. Groups can interact with the application to facilitate the communication and decision-making processes. Users access the application through an Internet browser interface. Application servers connect the user-interface to database servers (both spatial and non-spatial). The application servers also connect the user and data to the hydrologic models. Users can define management scenarios, attach models to a plan and have the application parameterize and run the models for the defined management plan.

2. APPLICATION DESIGN

The early stages of the DotAGWA project required developing a design document using the Unified Markup Language (UML) and other software programming techniques (Cate et. al., 2005). Important reasons for writing a design document included providing a roadmap for developers with clear application requirements. In addition, the design document also served as a bridge between AGWA 2.0 and DotAGWA. AGWA 2.0 has existing functionality that was identified as being reusable in the DotAGWA application. Since, re-use is an important tenet in software development; AGWA 2.0 became a foundation for DotAGWA, providing a significant portion of the geo-processing tools needed in the DotAGWA application.

2.1. Application Users

Like AGWA, DotAGWA's end users were identified as resource planners, developers, watershed management professionals, and many other interested parties (Cate et al., 2005). However, the scenario for application usage for the users had different requirements. A web-based application allows the user community to access the application from any computer connected to the Internet. Users can work together in activities like defining spatially explicit best management practices (BMP), test the effectiveness of BMPs in areas of interest and share the results with others via the Internet.

2.2. System Requirements

A system like DotAGWA has many hardware, software, programming tool, and data requirements. The development of the design document and knowledge gained from development of AGWA provided a solid foundation of information that could be used for DotAGWA application development, but because DotAGWA is an Internet-based application many new issues needed to be addressed.

2.2.1. Software

Like most other web applications DotAGWA has many different software requirements. Programmers need an integrated development environment (IDE). Data used to support the development and deployed versions of the application need to be stored in a common, easily accessible database. Web-server software is needed to make the application accessible over the Internet. Finally, and most importantly in the case of DotAGWA, a spatial data server is required to make spatial data available to the application.

2.2.2. Programming Environment

DotAGWA was developed using Microsoft Visual Studio 2003 (Visual Studio). In this environment programmers can develop in different programming languages that are supported in Visual Studio. In the case of DotAGWA, C# was chosen as the primary programming language.

2.2.3. Database Server

The application stores and serves both spatial and non-spatial data. The spatial data include digital elevation models (DEM), shape files, and attribute data that describe features like soil types, land-cover information, and meteorological information. The non-spatial data include, user information (e.g., user id, email), session information (e.g., model name, model output), and parameter look-up tables. The current version of the application uses a personal geodatabase (Microsoft Access file) for storing all of this information.

2.2.4. Web Server

Microsoft's Internet Information Services (IIS) was selected as the web application server. The main reason IIS was chosen was because it is the web server that Visual Studio works with. IIS also comes installed on most Windows operating system servers. Consequently using IIS required no extra cost, and imparted no limitations to the developers or end users.

2.2.5. Spatial Data Server

The Web Server also has an additional spatial data server that allows the application to display and manipulate spatial data required by the project. The spatial data server is the ESRI's ArcGIS Server version 9.1. ArcGIS Server was chosen because it allows programmers to build web-based applications that can both serve and manipulate spatial data. Earlier versions of DotAGWA used ESRI's ArcIMS map server, which only displays spatial data through a web interface and required using ESRI's ArcSDE application programming interface (API) to manipulate data. ArcGIS Server provides tools to the programmer through its API that allow spatial data manipulation.

2.3. Data

Spatial data from the USDA-ARS Walnut Gulch Experimental Watershed was utilized in the testing and development releases of DotAGWA. Ultimately, DotAGWA can be implemented on a national or regional scale, but at present it requires pre-loading data for project areas on a case-by-case basis. The base data sets include a DEM (raster/gird), land-cover/land-use (raster/grid), flow direction (raster/grid), flow accumulation (raster/grid), watershed boundary (vector/shape file), sub-watersheds (vector/shape file), stream network (raster/grid), soil maps (vector/shape file), climate (database), and land-cover/land-use parameter lookup database tables.

2.4. Hardware

Intel Pentium PCs running Windows XP Professional are used for application development. Any Windows-based PC can be used as long as the computer is able to run Visual Studio and ArcGIS Server. The deployment server is currently running Windows XP Professional, but in mid-2007 a new PC running Windows 2003 Server is going to be used to host the application. Because the spatial data server software can consume a great deal of system memory all computers have a minimum of 2 gigabytes (GB) of memory and at least 100 GB hard disk storage. The current development configuration, where there are very few web users accessing the application, is adequate for the current version of DotAGWA. However, when DotAGWA is opened to the general public, it will be hosted on the Windows 2003 server making it capable of supporting enterprise-level user requests.

3. APPLICATON DEVELOPMENT FINDINGS

3.1.1. Application Development

One of the most important requirements for the DotAGWA project was to re-use as many features as possible that were available from the AGWA 2.0 project (Cate et. al., 2005). AGWA 2.0 has functions that automate watershed delineation, watershed discretization, model parameterization, and model execution. These are the primary functions that DotAGWA relies on. These functions allow a user to delineate a watershed, discretize the delineated watershed into model elements, derive parameter inputs for the selected hydrologic model, and finally to run the model using the parameterized input data set. Figure 1 is ?????

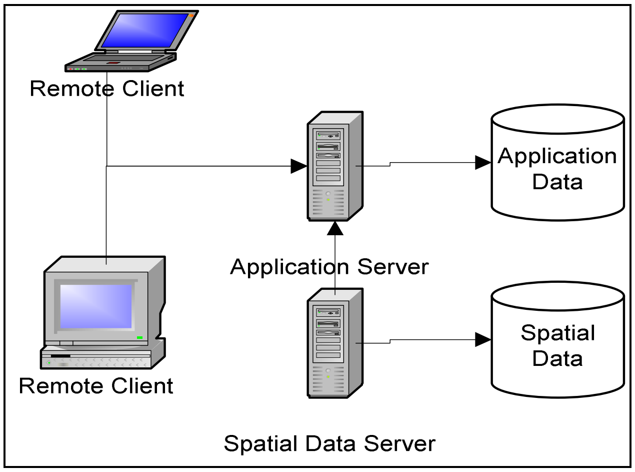


Figure 1 Basic System Architecture

DotAGWA's system architecture. The application is hosted by an Internet application server. The Internet application server presents the web interface to the client(s). The web interface has a map component that allows the user to develop application input by interacting with a map as well as typical web-based input forms. The application server process input and handles interaction between a supporting spatial database server as well as an application data server. Separating the two server types makes it possible to swap servers with minimal impact to the entire application. The spatial data server performs in a supporting role. The two servers are currently not two separate computers. However, the system is flexible enough to allow the database servers to be two physically different systems.

3.1.2. Interface Development

One of the first steps in application development was to construct web interfaces that were similar to the AGWA 2.0 application and its user interfaces. Figure 2 presents DotAGWA's basic work-flow model. The work-flow model represents the user's activities from login through delineation, adding management-plan features (optional), model selection, discretization, parameterization, model execution and result/output processing. Because the application focuses on providing a tool that allows people to build and compare management scenarios the system has to track individual users and management plans created by those users. A user logs in and an application session variable is instantiated and used for the entire time that the user interacts with the application. The session variable that describes the current user also stores data that tracks information related to the user's current management plan and has functions that allow the user to manipulate that plan and/or previously created management plans.

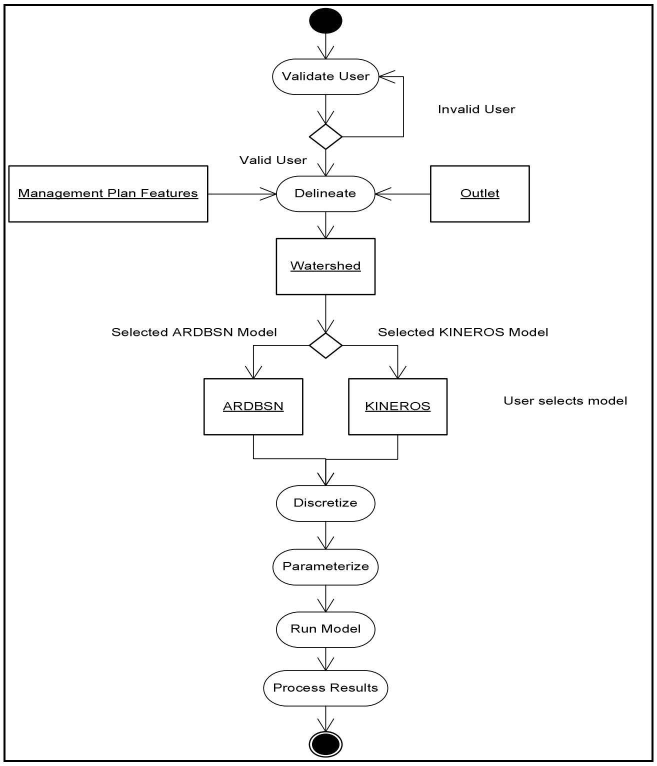
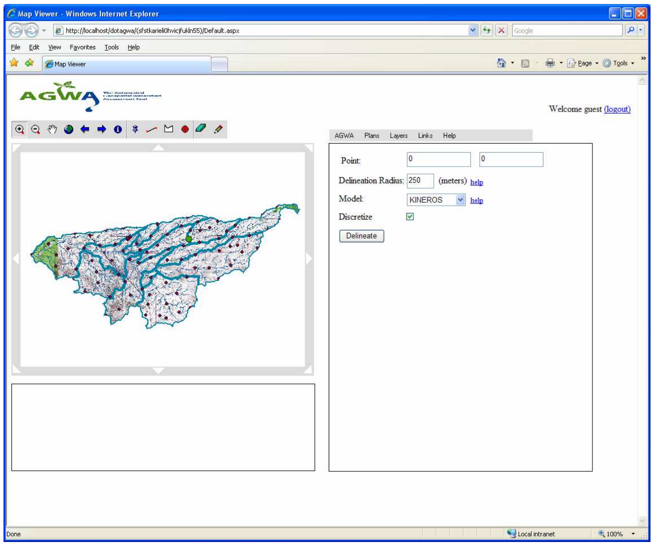


Figure 2 Basic Workflow

To begin, the user is presented with a map of the project study area (Figure 3), in this case the Walnut Gulch Experimental Watershed (WGEW) in southeastern, Arizona. At this point the user has initiated a new management plan that is ready to be developed for whatever management scenario the user would like to create.



Point: 0 0

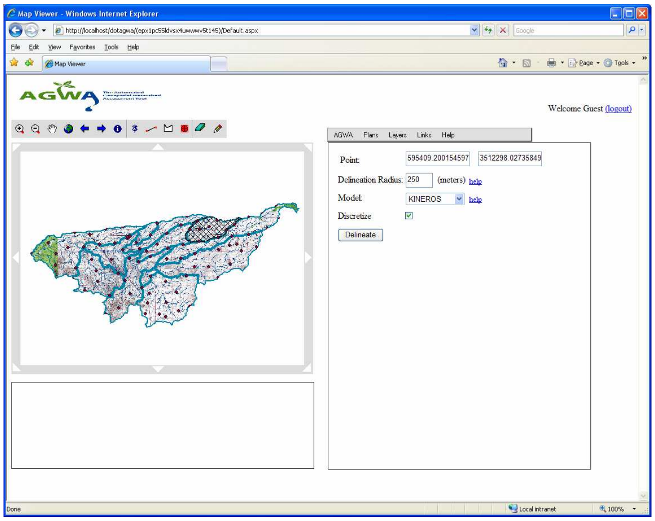
Delineation Radius: 250 (meters)

Model: KINEROS

Discretize ☑

Figure 3 DotAGWA Main Interface

Figure 4 presents an example of the user selecting a delineation point and creating the watershed delineation, which is shown on the map figure as the grey hash-lined area.



Point: 595409.200154597 3512298.02735849

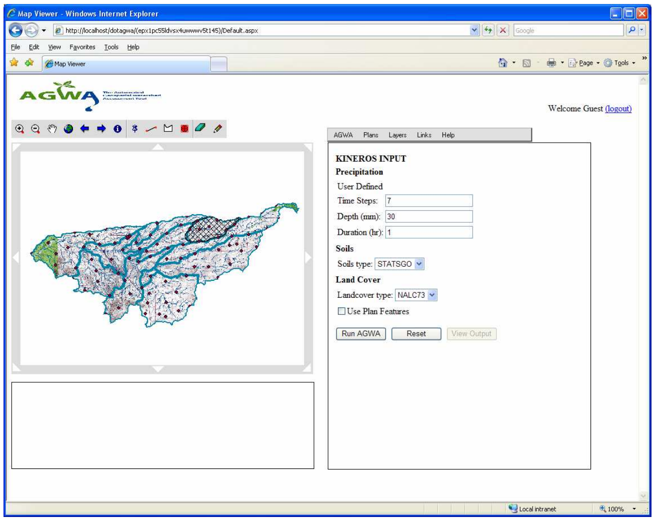
Delineation Radius: 250 (meters)

Model: KINEROS

Discretize ☑

Figure 4 User Selected Point for Watershed Delineation

After delineation (Figure 5) the user can select a model to run against the new watershed and select precipitation input for that model. The current state of the application allows for user-defined precipitation events (design storms) in KINEROS2, or a pre-loaded rainfall record for SWAT. The user also selects land-cover and soil maps to use for model parameterization. With all input parameters set, the model can be run. Running the model also executes the discretization process, during which the watershed is subdivided into model elements (subwatersheds and stream reaches) based on the model type that is selected. The two hydrologic models that are currently incorporated in the application each require a different discretization procedure.


Map Viewer - Windows Internet Explorer
AGWA
Welcome Guest (logout)
AGWA Plans Layers Links Help

KINEROS INPUT

Precipitation

User Defined

Time Steps: 7

Depth (mm): 30

Duration (hr): 1

Soils

Soils type: STATSGO

Land Cover

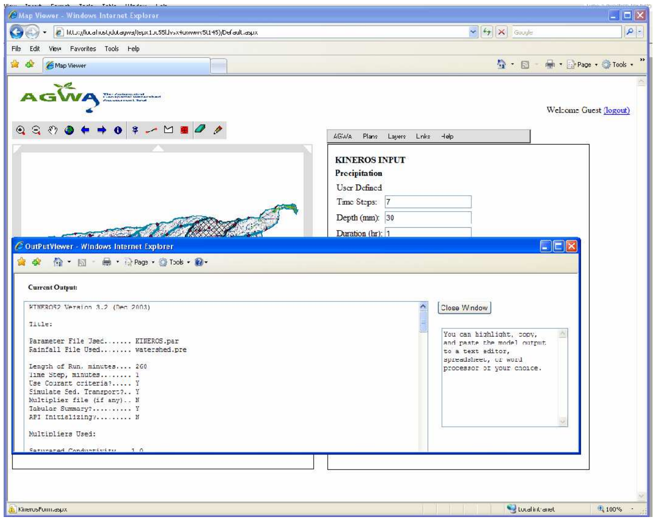
Landcover type: NALC73

□ Use Plan Features

Run AGWA | Reset | View Output

Figure 5 Initializing Model Run

After the model has finished running the user can click a button on the interface that loads the model's output file into a web page. The user can review the output and/or cut and paste it into another document for further analysis or use in a report. Also, the user can select to have the map display update to create a visual representation of an output parameter (i.e., peak flow).


Map Viewer - Windows Internet Explorer
AGWA
Welcome Guest (logout)
AGWA Plans Layers Links Help
OutPutViewer - Windows Internet Explorer
Close Window
KinerosForm.aspx
Local intranet

KINEROS INPUT

Precipitation

User Defined

Time Steps: 7

Depth (mm): 30

Duration (hr): 1

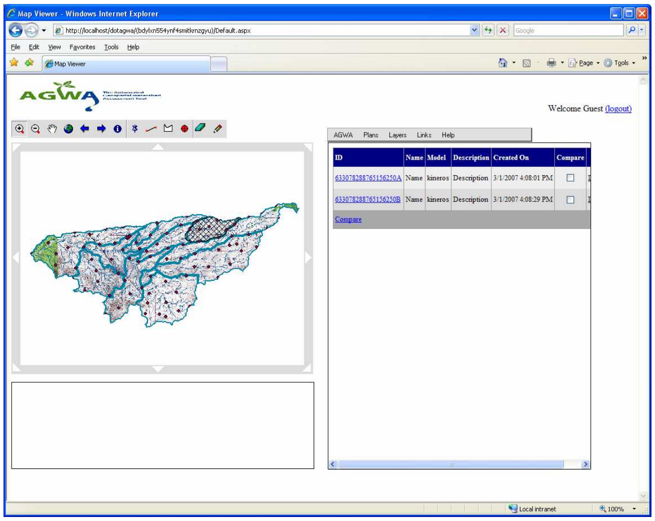
Current Output:

```
KINEROS2 Version 3.2 (Dec 2003)

Title:

Parameter File Used....... KINEROS.par
Rainfall File Used........ watershed.pre

Length of Run, minutes.... 260
Time Step, minutes........ 1
Use Courant criteria?..... Y
Simulate Sed. Transport?.. Y
Multiplier file (if any).. N
Tabular Summary?.......... Y
API Initializing?......... N

Multipliers Used:

Saturated Conductivity     1.0
```

You can highlight, copy, and paste the model output to a text editor, spreadsheet, or word processor of your choice.

Figure 6 Model Output

Finally, DotAGWA allows for users to manage BMP plans or scenarios. A user can develop a delineation, discretization, and define spatial features that represent a BMP. The BMP is stored in a supporting database. A model can be selected and run against a user-defined BMP and that output can then be stored as part of the BMP. Each plan can be reviewed or deleted from the user's profile. As of the writing of this paper, development is proceeding in adding functionality that lets a user compare (non-spatially) BMPs. Comparing BMPs will require the watershed to be near identical and model output for each scenario should originate from the same model. The comparison will be simplified. It will compare some of the most common output parameters like peak flow and sediment yield.


Map Viewer - Windows Internet Explorer
AGWA
Welcome Guest (logout)
AGWA Plans Layers Links Help
Local intranet

ID	Name	Model	Description	Created On	Compare
633078288765156250A	Name	kineros	Description	3/1/2007 4:08:01 PM	□
633078288765156250B	Name	kineros	Description	3/1/2007 4:08:29 PM	□

Compare

3.1.3. System Configuration

Developing the application has revealed many issues that are worth noting. An earlier version of the application was Java based and relied on calling the AGWA 2.0 functions as web services. Although an interesting concept, this approach proved to be problematic. AGWA 2.0 creates personal geodatabases (Microsoft Access databases) to store user output for each watershed. Every time a watershed delineation request was executed the delineated watershed was stored in a personal geodatabase. The ArcGIS server had to access the geodatabase in order to display the delineation in the web interface. This worked for the delineation process, but because the personal geodatabase, Microsoft Access, is not really designed to be a multi-user database accessed from an Internet application, subsequent calls to the personal geodatabase for discretization caused file locking and security issues with the IIS web application server. The solution to this involved porting the entire Java application to the Visual Studio C# language and connecting the AGWA 2.0 tools to DotAGWA as a dynamic link library (dll). Because AGWA 2.0 is accessed under the same IIS user account on the server no web-based security rules were broken and the output personal geodatabase could be accessed as needed.

3.1.4. Data Integration

Programming DotAGWA began by using the same standard data set available to AGWA 2.0 desktop users and used in development of the AGWA 2.0 utilities. The base data set covers the ~150 km^2 WGEW in Tombstone, AZ. When using AGWA 2.0 users are required to set up a project, which involves path names to input data sets like the DEM, land-cover grid, etc. In AGWA 2.0 each user and each project can have different base input data. However, because DotAGWA is a web-based application that could be accessed by multiple users, DotAGWA is currently not configured to let users define their own input data sets; all data must be pre-loaded by an administrator. Because the application is a multi-user web application, each installation is unique to the project study area. However, the application was developed with enough flexibility to allow administrators to develop a base data set that is compatible with both AGWA 2.0 and DotAGWA, deploy a new version of the web application by installing to an IIS enabled folder, and configuring the application to use the new data by changing only one configuration text file.

DotAGWA setup requires pre-processing of the base DEM to create the flow-direction grid, flow-accumulation grid and the stream-network grid. The AGWA 2.0 desktop application allows the user to derive these data sets from the DEM. The pre-processing is required because developing the dependent raster data sets is computationally intensive and slow. The current version of the application also uses default database look-up tables deriving model input parameters from land-cover classes and soil types. AGWA 2.0 allows users to edit these tables if needed. An enhancement currently in progress allows users to delineate and discretize a watershed, and subsequently change the land cover classes or types within that watershed. This feature will allow users to define future land-cover changes to watersheds and run the models against these changes to make estimates about the impacts the changes might cause. As stated previously, precipitation data is handled by letting the user create his/her own design storm or by using pre-loaded daily rainfall data. These methods represent base-case implementations for the two hydrologic models supported by DotAGWA. More technically sophisticated options will be added later, such as directly interfacing with National Weather Service web services

Finally, the application needs a way to track the development of a BMP, and to track user interaction with the application. The following data model shows how users and BMPs are accounted for in the supporting application database. Figure 7 shows the database table structures and relationships used to manage supporting application data. An application user has a unique identifier (id), first name (fname), last name (lname), password (pword) and email address (email). Each user can have zero or more management plans. A management plan consists of a unique identifier (id), a unique user id (user_id), a plan name (name), a plan description (description), creation date (created_on), the model name used on the plan (model), and the model output (output). Each plan has zero or more points that consist of a unique identifier (OBJECTID), spatial point object (SHAPE), a user identifier (user_id), a plan identifier (plan_id), and a point type (type). Each plan also has zero or more plan lines that consist of unique identifier (OBJECTID), spatial line object (SHAPE), a user identifier (user_id), a plan identifier (plan_id), and a line type (type). A plan can also have zero or more polygons, which have a unique identifier (OBJECTID), spatial polygon object (SHAPE), a user identifier (user_id), a plan identifier (plan_id), and a line type (type). Finally, each plan has a watershed delineation and a watershed discretization. A watershed is composed of a unique identifier (OBJECTID), spatial polygon object (SHAPE), a user identifier (user_id), a plan identifier (plan_id), and a point type (type) and a discretization is made up of a unique identifier (OBJECTID), multiple spatial polygon objects (SHAPE), a user identifier (user_id), a plan identifier (plan_id), and a point type (type).

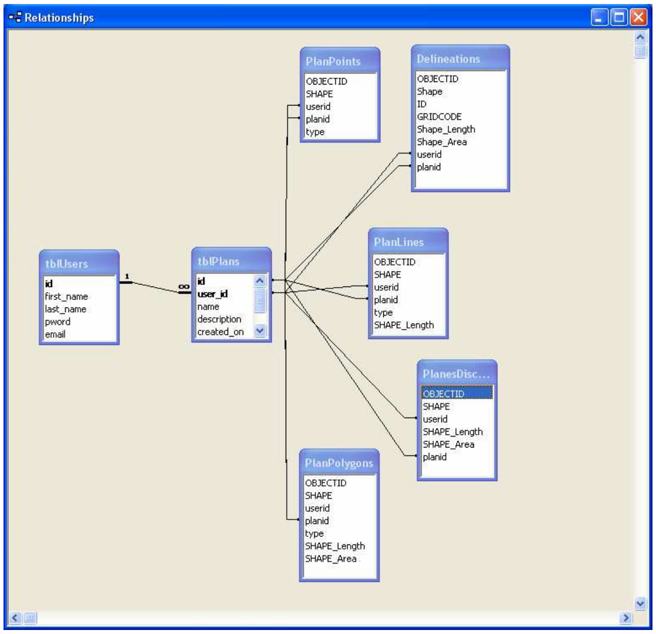


Figure 7 Database Models and Relationships

3.1.5. Model View Controller Design Pattern

The MVC design pattern is useful for separating the presentation components of the system architecture from the data storage and processing components of the system. The reason this separation is useful is if one of the three components the Model, View, or Controller changes or needs to be replaced the alterations do not cause major changes in other parts of the system like the interface. In the case of DotAGWA, web forms and pages serve as the view or user interfaces. Requests generated by user interaction with the view are handled by controllers with specialized functionality. For example, the application has a controller for handling interactions with the map and a second controller that handles interactions between user interfaces and the surface-water models. The processing, transfer and storage of data are handled by the model component, which serves as a bridge between the application and its supporting databases. The most important reason for using the MVC design pattern is to maximize cohesion and minimize coupling between application components. Striving for this minimizes the negative impacts to major changes in the software. If, for example, the application needed a completely new interface it could be developed and have almost no impact on the rest of the application. Or, if the GIS server software changes then minimal changes to the application code controlling communication between the map server and the view or model would be necessary.

4. FINDINGS AND SUMMARY

The DotAGWA project has involved integrating five major components: spatial databases; Internet web servers; spatial data servers; relational databases; and standalone surface-water hydrology models. The overall goal has been creating an application that addresses the growing need to build tools that increase our knowledge and understanding of landscape management and change. The Internet web server, the relational databases, and the spatial data servers are all designed to work together. Consequently, integrating these components has required the least amount of effort.

There have been some issues related to connecting the web application server, the spatial data server, and the AGWA 2.0 geo-processing tools. AGWA 2.0 uses personal geodatabases to create and store new spatial data sets. Personal geodatabases are a proprietary data format created by ESRI, but they are basically Microsoft Access database files. As mentioned in section 3.1.3 (System Configuration), the AGWA 2 functions were originally accessed as web services. However, this proved problematic because AGWA 2.0 produces and uses a personal geodatabase for storage. Because the personal geodatabases are Microsoft Access databases they are not well suited for Internet-based applications. The particular issues that arose were related to file locks on the database during geo-processing and scale issues related to the number of users accessing the web application. Two things were done to address these issues. First, the original version of DotAGWA was re-programmed from Java to C# using Microsoft Visual Studio. Second, DotAGWA has special operating-system level user permissions and utility functions that let it access personal geodatabases. In the very near future, the personal geodatabase usage in DotAGWA will be phased out and spatial data creation and storage will be handled by either Microsoft's SQL Server or Oracle's Enterprise database server, each with spatial data features enabled.

Connecting the surface-water models has proved to be one of the most challenging parts of the application. Each model has a unique way of defining a watershed and its sub-basins. Each of these basin definitions requires a unique parameterization method, which in turn requires application code to be created that transforms the spatial information into an input text file that the model can read. The surface-water models are technologically older, created using FORTRAN and meant to run as standalone desktop applications. Consequently, a strategy for running these models in a shared web environment has been developed whereby a model wrapper object is created that reads and writes the input data for the model. Each model is executed in a unique file-system folder. The folder is created based on the current user id and a time stamp. The model executable and input are stored to this temporary location, the model is run, and the model output is read from that location and stored in session variables in the web application. This method has so far been successful with one minor negative consequence: the need for removal of temporary files and folders. Lastly, open-source GIS tools were considered during the design phase. DotAGWA uses time and resource intensive raster geo-processing tools. The open-source community has a limited number of tools that can handle the type of geo-processing functions needed. However, the Federal agencies supporting DotAGWA development have negotiated licensing agreements providing access to ESRI's geo-processing tools. This, together with extensive documentation and very large user community for ESRI's products, the use of proprietary GIS products was a logical choice.

In conclusion, the goal of this paper has been to describe the methods and procedures used to construct a web-based application that can be used in decision making processes related to water resources, land-use / land-cover change and best management practices. With rapidly increasing demands on our natural resources and the potential for system-altering climate change, there is a heightened need for tools that exploit new technologies to make complex process models available to a wide audience, namely local governments, NGOs, and resource managers.

References

Arnold, J.G., J.R. Williams, R. Srinivasan, K.W. King, and R.H. Griggs. 1994. SWAT: Soil Water Assessment Tool. U. S. Department of Agriculture, Agricultural Research Service, Grassland, Soil and Water Research Laboratory, Temple, TX

Cate, A., Semmens, D., Burns, I.S., Goodrich, D.C., Kepner, W.G. 2005. AGWA Design Documentation: Migrating to ARCGIS and the Internet. EPA/600/R-05/056 and ARS/181027 Technical Reports, 77 p. (671 KB PDF)

Cate, A., Goodrich, D.C., Guertin, D.P. 2006. Integrating hydrologic models and spatial data in a distributed internet application. Proc. 3rd Fed. Interagency Hydrologic Modeling Conf., April 2-6, 2006. Reno, Nevada. 2006 CDROM

Miller, S.N., Semmens, D.J., Goodrich, D.C., Hernandez, M., Miller, R.C., Kepner, W.G., Guertin, D.P. 2006. The Automated Geospatial Watershed Assessment Tool. J. Environmental Modeling and Software. 22:365.377 (948 KB PDF)

Smith, R.E., D.C. Goodrich, D.A. Woolhiser, and C.L. Unkrich. 1995. KINEROS: A kinematic runoff and erosion model. Pages 697-732 in V. P. Singh, editor. Computer Models of Watershed Hydrology. Water Resources Publications, Highlands Ranch, Colorado.

Biography

Averill Cate, Jr. is a PhD candidate in Natural Resources at The University of Arizona. He has a B.S. in Hydrology from the University of New Hampshire, an M.S. in Watershed Management from The University of Arizona, an M.S. in Management Information Systems for The University of Arizona. Current research projects include the AGWA/AGWA 2.0 and DotAGWA projects, as well as work related to integrating spatial information systems, natural resources, and watershed management.

Darius J. Semmens is a Research Hydrologist with the U.S. EPA Office of Research and Development in Las Vegas, NV. He received his Ph.D. in Watershed Management from The University of Arizona, M.S. in Geology from Northern Arizona University, and B.S. in Geology from the University of Minnesota. Current research efforts are directed at landscape characterization, management and climatic scenario development, alternative futures, and cumulative impacts assessment in environmental modeling. He is also an Adjunct Professor of Landscape Architecture at the University of Nevada, Las Vegas.

David C. Goodrich is a Research Hydraulic Engineer with the USDA-Agricultural Research Service-Southwest Watershed Research Center, in Tucson, AZ (since 1988). He received his Ph.D. from the Hydrology and Water Resources Department at the University of Arizona and his B.S. and M.S. degrees in Civil and Environmental Engineering from the Univ. of Wisconsin-Madison. Current research efforts are directed to scaling issues in rainfall-runoff modeling, identification of dominant hydrologic processes over a range of basin scales, climatic change impacts on semiarid hydrologic response, incorporation of remotely sensed data into hydrologic models, and the functioning of semiarid riparian systems. He Co-Led the interdisciplinary multi-agency Semi-Arid Land-Surface-Atmosphere (SALSA) Research Program and is an Associate Adjunct Professor in the Dept. of Hydrology and Water Resources of the University of Arizona. He is also an Executive Member of the NSF SAHRA (Sustainability of semi-Arid Hydrology and Riparian Areas) Science and Technology Center in charge of river systems research and has worked closely with policy and decision-makers within the Upper San Pedro Partnership.

Dr. D. Phillip Guertin, is an Associate Professor of Watershed Management and Leader of the Advanced Resource Technology Group in the School of Renewable Natural Resources. Dr. Guertin received his B.S. in Forest Hydrology from Utah State University, M.S. in Watershed Hydrology from Colorado State University and Ph.D. in Forest Hydrology from the University of Minnesota. Dr. Guertin was also a National Research Council Postdoctural Fellow in Hydrology with the U.S. Geological Survey. Dr. Guertin's recent research has focused on the use of geographic information systems for natural resource and watershed management in the southwestern United States. He currently has GIS-based research projects with the U.S. Forest Service, U.S. Army, U.S. Navy, U.S. Fish and Wildlife Service, National Park Service, U.S. Geological Survey, and Agricultural Research Service.