INTEGRATING HYDROLOGIC MODELS AND SPATIAL DATA IN A DISTRIBUTED INTERNET APPLICATION

Averill Cate, Jr., Research Associate, University of Arizona, Tucson, AZ, acate@email.arizona.edu; David C. Goodrich, Research Hydraulic Engineer, USDA/SWRC, Tucson, AZ, dgoodrich@tucson.ars.ag.gov; D. Phillip Guertin, Associate Professor of Watershed Management, University of Arizona, Tucson, AZ, phil@nexus.srnr.arizona.edu

Abstract Planning and assessment in land- and water-resource management are evolving from simple, local-scale problems toward complex, spatially explicit regional ones. Such problems have to be addressed with distributed models that can compute runoff and erosion at different spatial and temporal scales. In addition, to evaluate management practices and their impacts on water quality, land and resource managers need to describe and simulate the impacts of land use and best management practices (BMPs) on watershed response. The extensive data requirements and the difficult task of building input parameter files have long represented an obstacle to the timely and cost-effective use of such complex models by resource managers. To address this problem an Internet-based hydrologic modeling tool, DotAGWA, is being developed. This paper will review the DotAGWA design and features.

Introduction Effective watershed decision-making requires the integration of knowledge, data, simulation models, and expert judgment to solve practical problems and provide a scientific basis for decision-making at the watershed scale (NRC 1999). User-friendly decision support tools are needed to help different stakeholder groups develop, understand, evaluate and share alternative watershed management strategies. The tools should integrate a suite of components consisting of database management systems, geographic information systems, simulation models, decision models, and user-friendly interfaces that could then be available to different stakeholder groups. The Internet provides a great opportunity for sharing information and applications with decision-makers.

The Automated Geospatial Watershed Assessment tool (AGWA, see: http://www.tucson.ars.ag.gov/agwa) was developed jointly by the USDA Agricultural Research Service, the U.S. Environmental Protection Agency, the University of Arizona, and the Univ. of Wyoming to conduct hydrologic modeling and watershed assessments at multiple time and space scales (Miller et al., 2002; Goodrich et al., 2006). AGWA is a standalone, desktop application that uses widely available standardized spatial datasets. The required data sets include topography (DEM data), soils, 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 is currently being used nationally and internationally. International applications include projects in Israel, Kenya, and Mexico. In 2004, AGWA was added to the EPA BASINS toolbox (BASINS 3.1 – www.epa.gov/OST/BASINS/), a multi-purpose environmental analysis system to support environmental assessments. AGWA is also recognized by NASA (i.e. “Coin Chart”, http://wec.gsfc.nasa.gov/index.html) as an environmental modeling tool and decision support system.

The development and subsequent support of AGWA has illustrated that the application is useful to many stakeholders and organizations. However, not all of these people have access to the geospatial data or software required to process the geospatial data. While some people recognized the value of the application, they did not have adequate scientific training to understand the inner-workings of the hydrologic models, nor had access to other tools such as geographic information systems. Recognizing these issues lead to the initial development of DotAGWA, an Internet version of the AGWA tool (Miller et al. 2003). This paper will review the current status and design characteristics of DotAGWA.

DotAGWA was designed as a shared application used to assist in the decision-making processes and to offset the software and data requirements typically required in a desktop application (Figure 2). The features available in DotAGWA help users share and visualize data. Different stakeholder groups can also interact with the application to help facilitate the communication and decision making processes. Users access the application through an Internet browser interface. Application servers connect the user-interface to the database servers (both spatial and non-spatial). The application servers also connect the user and the 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. The user can specify different output formats (i.e., XML, Word doc, HTML) for the resulting simulation output. Finally, the DotAGWA architecture provides access to some of the individual components of the application. For example, the Web Services component of Figure 1 shows a watershed delineator (WShedDelin). The delineator has been implemented as a web service, which makes it available for consumption by other applications over the Internet. Consequently, other application developers could extend these components.

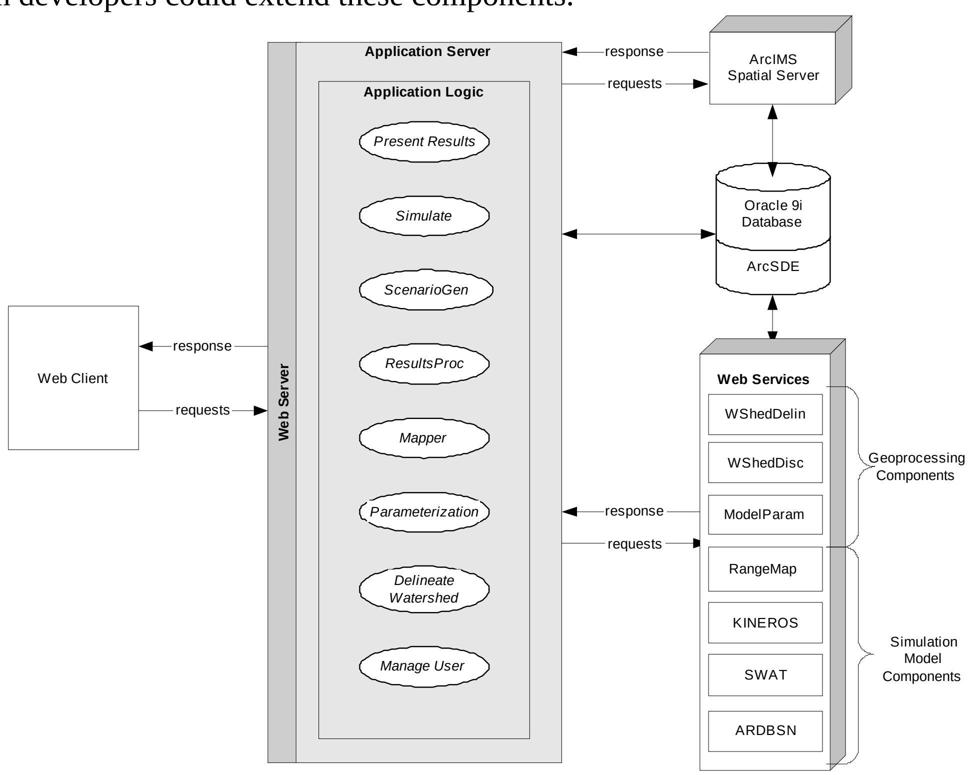

Figure 1 DotAGWA Conceptual Model

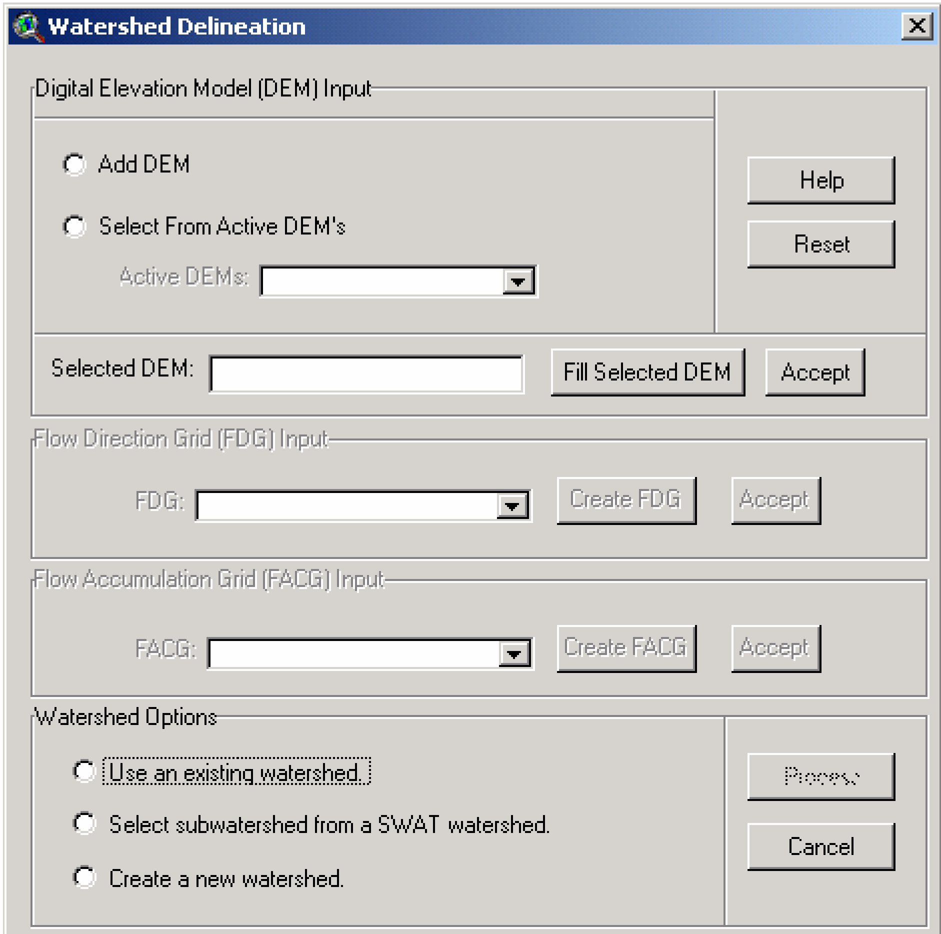

Figure 2 AGWA Desktop Application Example

System Architecture DotAGWA was based on the Model View Controller (MVC) design pattern. The MVC design pattern is useful in 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, the View, or the Controller) changes or needs to be replaced the alterations do not cause major changes in other parts of the system. For example, if the supporting database used in the system implementation is MySQL and after an extended amount of usage MySQL proves to be inadequate, then another database like PostgeSQL can be plugged into the system with minimal effort and interruption to the existing system.

Model: The model component of the architecture also allows the different system components to be represented as individual entities. For example, one of the identified system needs was permitting multiple users to access the system. In order to maintain user state, Internet applications require session tracking. The way this was accounted for in DotAGWA was by designing a representation of a user or a user object (Figure 3). The user object models the Internet web application user. The user object has some attributes (e.g., user id, first name, last name, email, and plans) and methods or functions that are used to access the information stored in a user object. Another, object modeled in the application is the management plan. A management plan has points, lines, polygons, a name, a model, and model output. The points can represent features like water points or outlets. Lines can represent fence lines or other linear features and polygons can represent features like pastures or exclusion zones. The model is the name of the surface water runoff model the user has decided to run for the current management plan and the output from the model is stored in the user's plan.

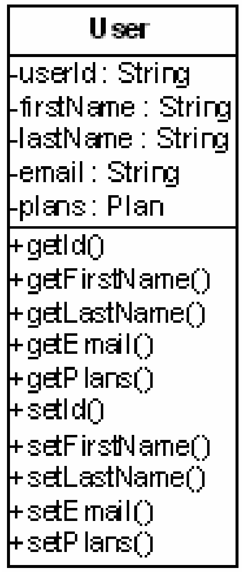

Figure 3 User Object Model

View: The view (Figure 4) is basically the user interface and all of the components associated with the interface. The MVC architecture allows user to interact with all the features of the application without knowing how the different components interact. The user is allowed to create a management plan by adding points, lines, and polygons and selecting models to run for a particular management plan. If the functionality of how the mapping component of the application changes, then this component can be updated without changing how the backend database saves information related to a management plan.

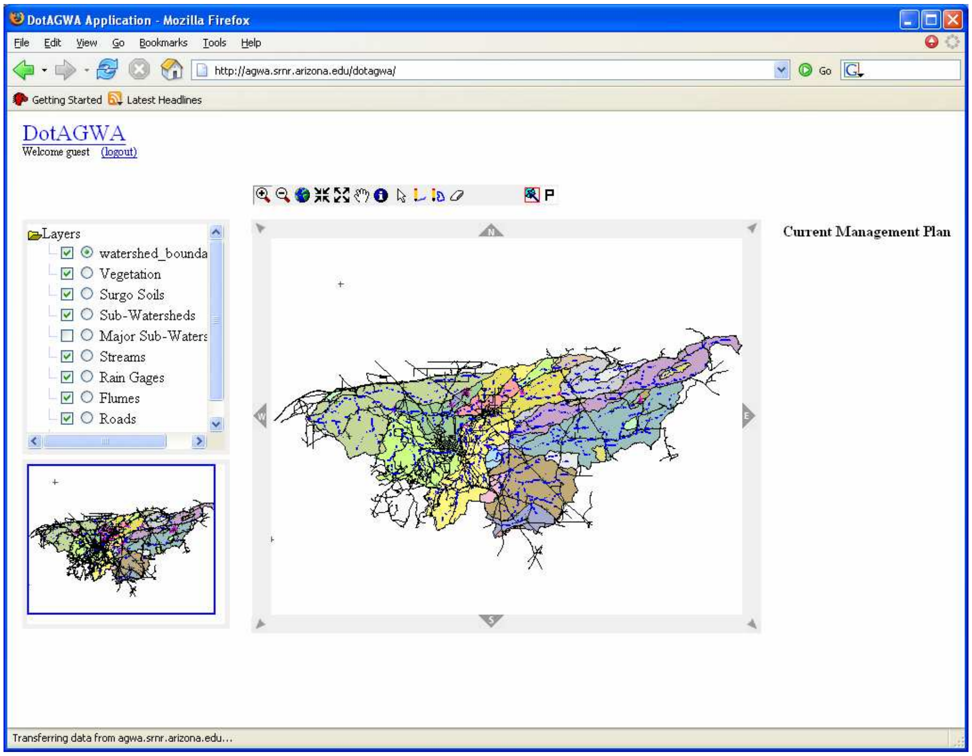

Figure 4 DotAGWA user interface

Controller: The controller ties the Model and View components together. The controller maintains the current state of the system and responds to user interactions or application requests. The controller maintains state information related to the user and the web server application as well as state information related to user defined features (i.e., points, lines, polygons). The controller also provides communication between the mapping component of the application and the supporting databases. The controller manages the transfer of user state information into and out of the supporting database system.

Functionality: The core functionality of DotAGWA is watershed delineation, watershed discretization, model parameterization, model execution, output management, and plan management. Watershed delineation consists of having the user click on a point (an outlet) and the delineation module defines a basin based on this point. Watershed delineation uses a digital elevation model (DEM), a flow accumulation grid, a flow direction grid, and the outlet point to produce a shape file representing the delineated watershed. The shapefile (Figure 5) is added to the map display and the map is refreshed allowing the user to see the results of the delineation process. Discretization defines sub-basins in the delineated watershed based on the user selected model and contributing source area. A discretized watershed can then be parameterized. Parameterization uses the sub-basins to gather soils, vegetation, precipitation, and other data that are required by the selected model to create an input data set for the model. The input data sets can be altered by the user before being used in a model. After parameterization DotAGWA will take the created input data set and pass it to the model, run the model, and display the model output for the user. The user can download the output data set in different file formats (i.e., xml, spreadsheet). Finally, DotAGWA allows the user to save a defined management plan, which consists of user-defined points, lines, polygons, delineated watershed, model input, and model output into a database. Users can load existing plans for comparison or modification.

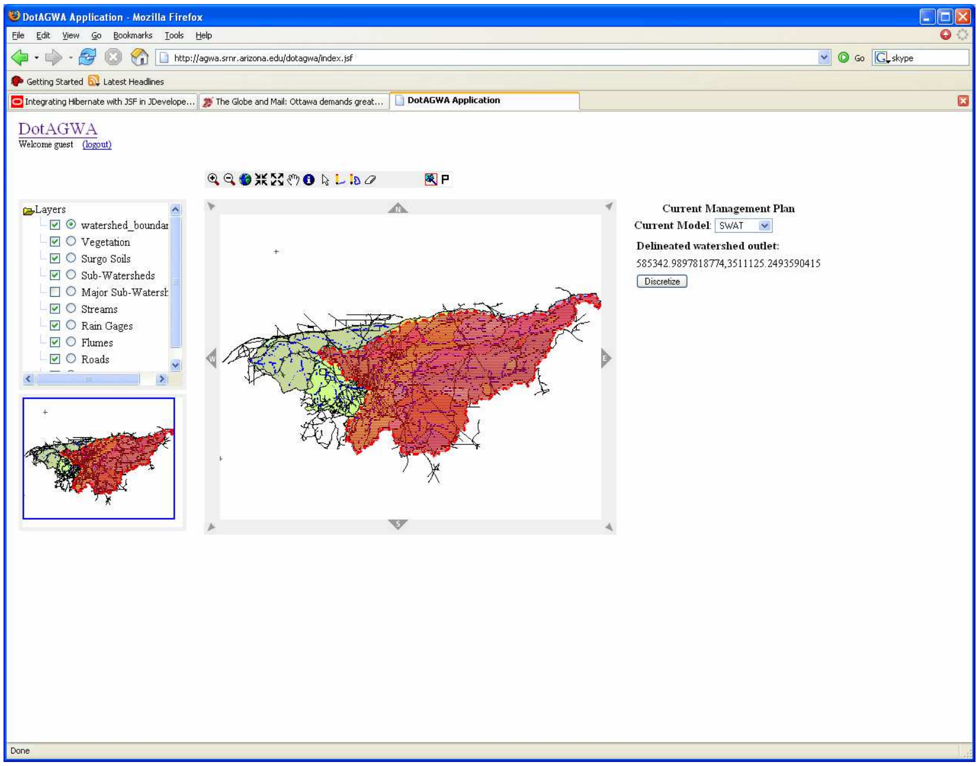

Figure 5 Watershed Delineation

Findings/Conclusions DotAGWA was designed to leverage functionality built into the AGWA desktop application. The new releases of AGWA and DotAGWA applications were developed in tandem to maximize code re-use. Many of the geo-processing functions built into AGWA are re-used in DotAGWA by writing wrapper code around these functions that make them available as web-services. This fits into the MVC architecture in that it allows developers to change how any particular function may behave without affecting how the output of the function is used.

Even though AGWA and DotAGWA are almost mirror applications, many interesting issues have emerged as more of the AGWA functionality is incorporated into DotAGWA, including tools to simulate common best management practices. One issue that must be addressed in the future is data management. AGWA is dependent on spatial data that has to be provided by the user for each project and stored on the desktop in a specified file structure. DotAGWA is currently connected to only one spatial data set for the Walnut Gulch Experimental Watershed. However, the long-term goal is for DotAGWA to be a GIS-service with the capability of accessing Geo-libraries to obtain input data. Since there is a separation between model, view, and controller different spatial data sets could be used with the current version of DotAGWA as long as the data is structured the same way as the Walnut Gulch data set. This would require the development of tools to support the gathering, integration and organization of data obtained from the Geo-libraries. Another approach would be to design DotAGWA to utilize the data structures for common Geo-libraries, or have DotAGWA support its own data archive. Other technical issues include how and where user created information will be stored, security and visualization needs for different user groups.

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.

Goodrich, D.C., Scott, S., Hernandez, M., Burns, I.S., Levick, L., Cate, A., Kepner, W.G., Semmens, D.J., Miller, S.N., Guertin, D.P., (2006). Automated Geospatial Watershed Assessment (AGWA): A GIS-based hydrologic modeling tool for watershed management and landscape assessments, Proceedings of the Third Federal Interagency Hydrologic Modeling Conference, April 3-6, 2006, Reno, NV, CD-ROM, 8 p. (this issue)

Miller, R., D.P. Guertin and P. Heilman. (2003). An Internet-based spatial decision support system for rangeland watershed management. In: First Interagency Conference on Research in the Watersheds, Proceedings. Renard, K.G, S.A. McElroy, W.J. Gburek, H.E. Canfield, and R.L. Scott, editors. U.S. Department of Agriculture, Agriculture Research Service. pp. 725-730.

Miller, S.N., D.J. Semmens, R.C. Miller, M. Hernandez, D.C. Goodrich, W.P. Miller, W.G. Kepner, and D. Ebert, (2002). GIS-based hydrologic modeling: the Automated Geospatial Watershed Assessment tool, In: Proceeding of the Second Federal Interagency Hydrologic Modeling Conference, July 28 - August 1, 2002, Las Vegas NV, CD, 12 p.

National Research Council (NRC), Committee on Watershed Management, (1999). New Strategies for America's Watersheds. National Academy Press, Washington, D.C.

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