Using ArcGIS to Support a Regional Hydrologic Model

Author: Richard R. Miessau II, PMP
South Florida Water Management District
Hydrologic & Environmental Modeling Department

Abstract

The South Florida Water Management District (DISTRICT) is responsible for the operation and management of the Central & South Florida (C&SF) system of canals, structures, pump stations, dikes, and levees in order to balance the needs for flood control, water supply, agriculture, and environmental protection. Hydrologic simulation models are used for analysis of the performance of the hydrologic system. At the regional scale (several counties in size), the DISTRICT has been developing a new, generic model, called the Regional Simulation Model (RSM). RSM is a finite volume code, based on an irregular mesh of triangles. It is an object-oriented design, implemented in C++ and makes use of a variety of commonly available file formats, including NetCDF and HEC-DSS. RSM has two major components, HSE (Hydrologic Simulation Engine) and MSE (Management Simulation Engine). HSE is responsible for simulating the natural movement of water through overland flow, groundwater flow, and in canals and other conveyances, while MSE simulates the management decision process, using operational rules.

This paper discusses how GIS was utilized in the design of the RSM Graphical User Interface (GUI). The GUI has been developed for the RSM, incorporating a collection of tools used to generate documented input files. It provides a streamlined workflow for users of the RSM and automates documentation of the modeling scenarios. It also provides an interface to run the model and provides a collection of tools to post-process output from the model. The GUI uses a combination of a geodatabase, custom geographic information system (GIS) applications and Python applications. The highly detailed geodatabase represents the complex physical hydrologic water features and relationships that are simulated in the RSM. Examination of these elements in the GUI and the resulting files offer a look into the model and the new approach being taken to solve real world problems.

When fully developed, the RSM and accompanying GIS tools are expected to be one of the primary DISTRICT modeling platforms for the next decade.

Introduction

Developing a Graphical User Interface (GUI) to a hydrologic model requires knowledge of how the model functions and an understanding of how users are expected to utilize and interact with the model. The Graphical User Interface (GUI) to the Regional Simulation Model (RSM) provides users a means to assemble their model, execute the model and visualize their output. Some of the early development was undertaken to support developers of the model. As the model matured and moved to a calibration and implementation phase the GUI evolved to include a wider selection of features to support implementation of the model. From the beginning a traditional project management approach was adhered to. Standard processes taken from the Project Management Institute (PMI©) Body of Knowledge were used to guide the work through planning, scoping, execution, and control of the project. Software development processes were adopted from the Carnegie Mellon, Software Engineering Institute, Capability Maturity Model (CMM©). The overall RSM development effort employs both PMI© and CMM© methodologies. These highly disciplined methodologies prescribe standard processes that can be monitored, measured and fine tuned to ensure successes can be repeated and improvements can be applied at the end of each development cycle. To these ends, the GUI captured user requirements, documented development efforts, and built upon each success to assemble a comprehensive user interface and integrated set of tools.

The RSM was envisioned as being a model to be made accessible to a wide variety of clients. To minimize barriers imposed by hardware and software costs, a strategic decision was made to utilize an OpenSource approach. The model has been written in an object oriented C++ framework and has been compiled to run on a Linux platform. Linux affords implementers the opportunity to select hardware based on their needs and resource availability. OpenSource Linux GNU compilers have been used to ensure the widest implementation across platforms. And finally, it was decided the GUI would be developed using Python, a 4th generation, polymorphic scripting language, which would also provide the widest distribution and platform independence.

Project Initiation

Having selected an approach, methodology and OpenSource platform the RSM GUI development effort was initiated during the first quarter of 2005. A project manager, application programmer, geographer and hydrologic engineer were assigned to the effort. The first order of business was to assess the current tools being used by the model developers and to collect requirements. This assessment set the stage for the GUI development and indicated a set of constraints. Leveraging the extensive enterprise Geographic Information System (GIS) at the DISTRICT, ESRI© GIS had already been established as the source for geographic spatial data pertaining to the model mesh, canal network features, hydrologic structures and other geographic features.

Developers of the Regional Simulation Model at first relied on manually assembling input data sets, hand writing XML files, and using custom written scripts for data visualization. The number of manual steps currently required to perform a modeling scenario from beginning to end and the number of systems a modeler was required to interact with impeded the pace at which model scenarios could be completed. A preliminary inventory identified 18 pre- and post-processing tools being used in addition to numerous manual processes in need of automation.

A prioritization of needs was established to help guide the development of the first GUI tools:

- HIGHEST NEED: processes where only a manual process exists
- MEDIUM NEED: processes where automated, but not standardized, methods exist
- LOW NEED: processes where a satisfactory automated (nut not integrated) method already exists

After assembling the requirements of the modelers, a prioritized list of needs was created and used to scope the GUI project. A charter was signed by the Program Director for the RSM and work was initiated to develop the framework for the GUI.

GUI Programming Framework

Selection of a programming framework sets the basis for how all applications will be developed to ensure integration and compatibility. The framework chosen to support development of the RSM GUI consisted of Python ver.2.4 and a collection of companion libraries to extend the capabilities of the scripting language. TkInter was chosen as the Python GUI programming interface based on its wide usage, portability among computer platforms and content of GUI features. Within the GIS community, C# and Microsoft .Net© programming platform were the dominant standard and were chosen for development of custom GIS tools within the ESRI© GIS. For archiving and version control Subversion (SVN) by Tigris.org was chosen based on compliance with DISTRICT enterprise standards, usability and being and OpenSource product. With the pieces in place work was started to develop the first pieces of the structure that would define a flexible application interface to develop and implement the GUI tools

Data Architecture

Data used by the RSM comes in many standards and formats. Primarily, the GUI was expected to aid in generation of the XML input files for the model. Additionally the model utilizes and generates ASCII, DSS and netCDF files. A geodatabase schema was designed as the main repository for the spatial data. Database experts were consulted in designing the geodatabase and establishing relational tables to organize the extensive attributes needed to produce the XML files used as input to the model. The resulting database incorporated the underlying principles of the ArcHydro geodatabase schema but the addition of significant custom feature classes,

relational tables and dynamic attribute generation precluded the schema from being ArcHydro compliant. The final result was a highly attributed project level database schema optimized to contain hydrologic relationships and features suitable for the RSM. It was a major concern that the final schema took into consideration the data needed to be included, as well as the functionality needed to be supported in order to automate generation of the XML files.

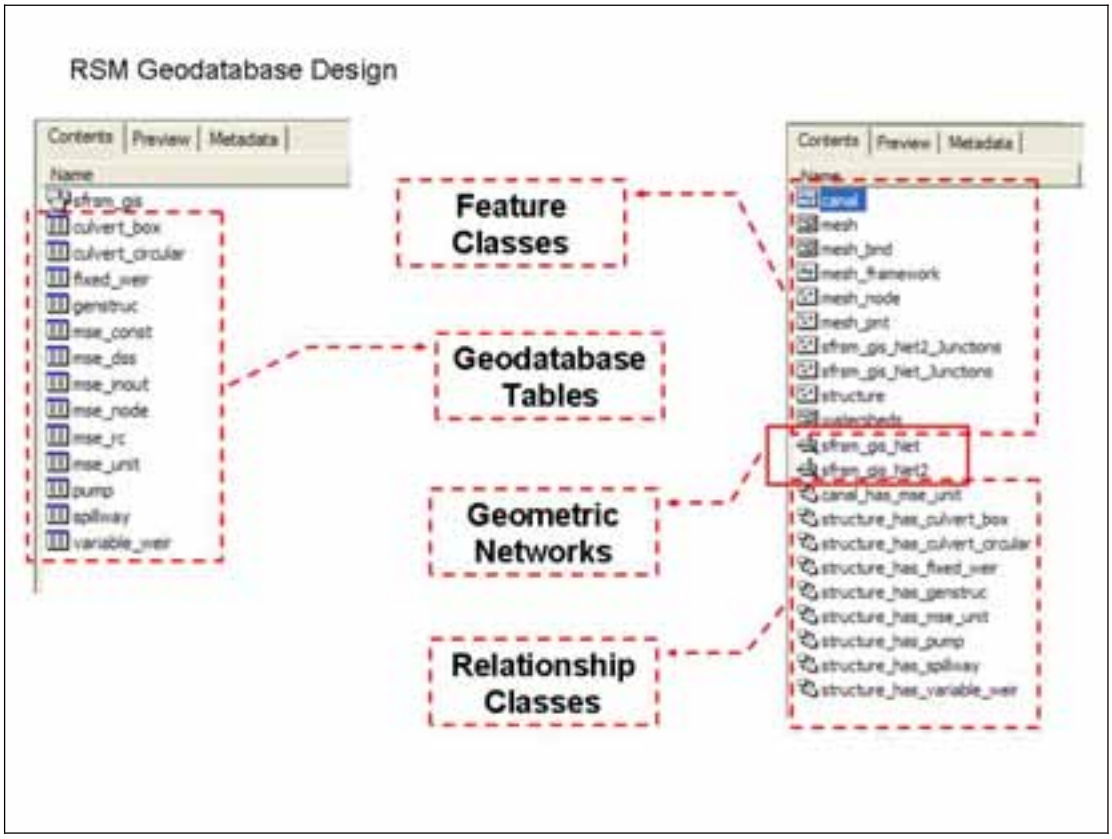

Figure 1. RSM Geodatabase Design

Geodatabase Featureclasses and Attributes

The RSM geodatabase consists of Feature classes for the mesh, canals and structures. The mesh feature class is intersected with several data layers to acquire key physical information such as topo, landuse, hydrologic and conductivity. The structure feature class contains subtypes to help distinguish between diversion structures, inline structures and junctions. Diversion structures divert or re-direct water. Inline structures impact the flow of water but do not divert the water from the canal on which it resides. Junction (structures) are used to represent logical breakpoints along the canals where structures reside, intersections occur or where there are significant characteristics or attribute changes. The canal feature class consists of complex edge polylines and two subtypes to help represent canal segments and watermover segments. Canal segments are polylines which collectively represent physical properties of canals in the hydrologic network such as bottom width, slope, depth, and mannings coefficient. Individual segments are organized into canal reaches and a canal reaches are assembled into canal stage-reaches. The canal segments are used to

build the canal geometric network. Watermover segments are used to represent the movement of water from one source to a destination but they do not participate in the hydrologic network and they are not reflected in the model. They primarily are used to ensure water comes only from one source and goes to only one destination, thus they help move water through a canal intersection.

Relational tables have been incorporated as part of the geodatabase to organize the structure information for individual units at each physical structure. Structures can consist of one or multiple culverts, pumps, weirs or spillways. Relationship classes keep track of the one-to-many relationships for each type of unit at each structure. And finally domains have been used to limit the type of data which can be input for several of the attributes to ensure consistency where automation relies on key data fields such as enabled (yes, no).

A key feature of the geodatabase is the ability to enable or disable any feature which enables modelers to turn-off features in the database. Disabled features are then ignored by the preprocessing XML tools.

Tool Design & Development

RSM GUI was organized into three categories: pre-processing, model execution, and post-processing. These three groupings of the tools allowed developers to optimize and reuse functions within each category as functionality was repeated within several tools. For example, there are several post-processing tools which read a netCDF input file. Reading a netCDF only needed to be programmed once and then reused by each tool needing to read a netCDF file.

Training & Support

The simplified logic flow for setting up, executing and post-processing a model run is as follows:

- Generate an irregular triangular mesh
- Intersect the mesh with other spatial features (topo, landuse, etc.)
- Discretize the canal network (segmentation)
- Assign and adjust spatial attributes pertaining to each feature
- Generate XML files
- Run the MODEL
- Post-process and generate output graphics for analysis

Pre-processing

The pre-processing portion of the RSM GUI enables users to create, read and modify data directly in a geodatabase. The GUI tools don't require the users be proficient with an endless assortment of applications, platforms and data sources necessary to link and view the desired objects. The tools within GIS enable users to graphically reconfigure the model for different simulations. New modeling databases can be generated by starting with a template database and "cookie cutting" a section to meet specific modeling needs. New meshes can be created, imported and a blank schema is created, allowing users to then input their data relevant to their model application.

Canal discretization (segmentation) is carried out automatically against the geodatabase preserving the physical characteristics of the canal network but allowing modelers to select an optimal segment length to coincide with the density of their mesh.

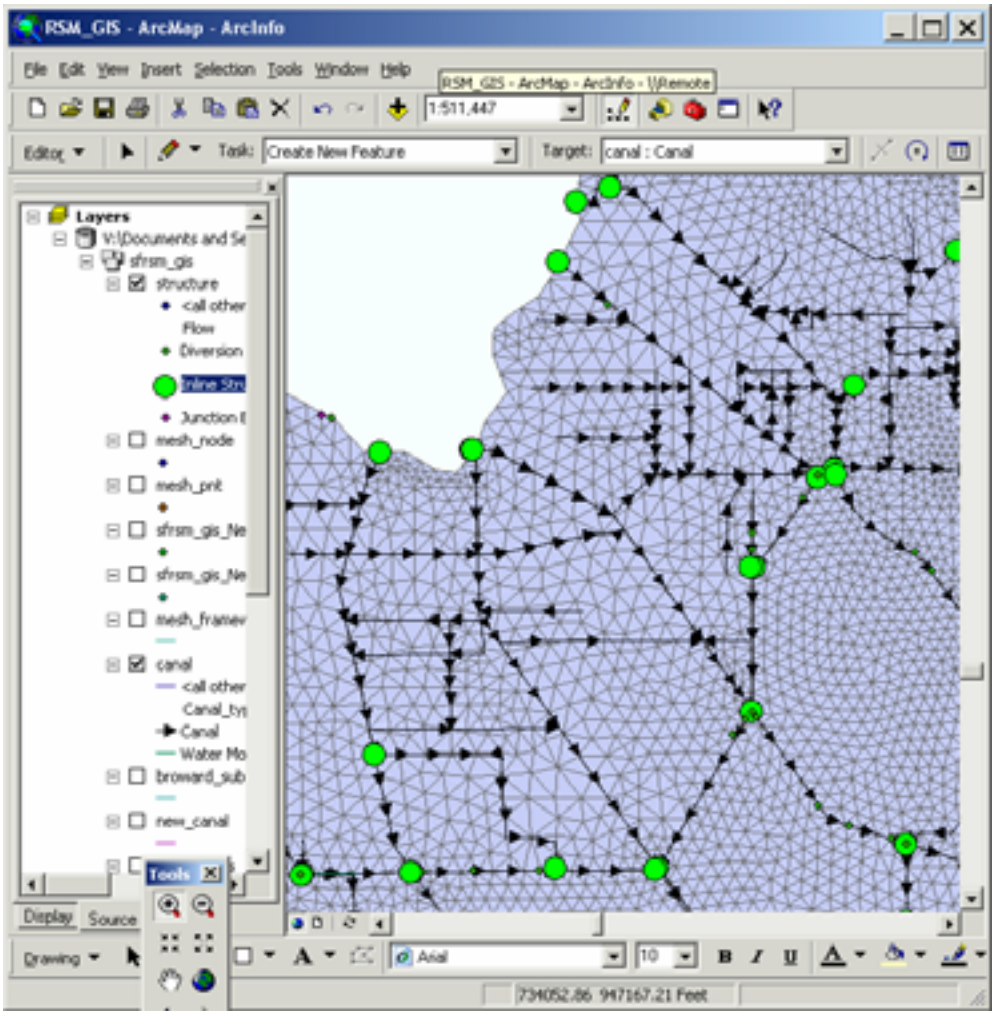

Figure 2. GIS Interface showing canal network, structures and mesh

Once the geodatabase has been configured to represent the model being simulated a set of XML generating tools can be used to generate the XML input files needed to run the model. These highly productive tools automate the XML generation based on a document type definition (DTD) saving hundreds of hours and produces a consistent set of input files that can be tracked to a geodatabase source. The modelers currently implementing the RSM use and reuse a number of templates editing them to test and calibrate the model. To aid with editing existing XML files a set of Python tools have been created to generate the most common XML blocks. Blocks of XML can be copy/pasted to existing XML's or assembled to create new XML's. To the extent possible, a primary objective of the GUI is to minimize the need and/or desire for the modelers to hand-edit their XML files. Once the XML files are assembled the model can be executed.

Figure 3. Relational table and visual check of geodatabase

Generating XML files

Model Execution and Post-processing

A "Main XML" is used to configure the model run. This control XML defines the input parameters, location of linked files and specifies the output options selected for the run. The compiled model has been implemented to run standalone on a Linux computer. A Linux supercomputer cluster has been created by the DISTRICT to meet the production needs for the volume of models run by the agency.

While the model can be easily executed at the command line level the RSM GUI offers added value features such as:

- Capturing a details pertaining to each run in a searchable "model log"
- Capturing statistics to assess performance of the hardware being used to execute the model
- Simplifying execution locally or on the supercomputer cluster

Post-processing

Post-processing of the model output from the RSM is the current focus of the RSM GUI effort. The primary output data files are DSS and netCDF binary data formats. Data output by the model can best be understood by examining relationships of waterbodies and watermovers. Every body containing water is categorized as a waterbody. Connections between waterbodies fall into a wide variety of watermovers. Waterbodies and an initial volume, ending volume and delta volume at each timestep. Watermovers can be examined to analyze the influences for the delta volume. Waterbodies can represent canal segments, mesh cells, lakes, basins and any body that can contain water. Watermovers represent recharge, runoff,

evapotranspiration, rainfall, seepage and any means to transport water or exchange water among the waterbodies. The GIO tools directly read the binary output files, assemble data into collections of waterbodies, generate statistics and generate graphics for analysis. A variety of methods are used to parse the data into arrays facilitating graphical and tabular examination of the data. As usage of the model increases and applications for projects becomes the primary focus the GUI development team expects to continue expanding the selection output options and refining the graphics capabilities.

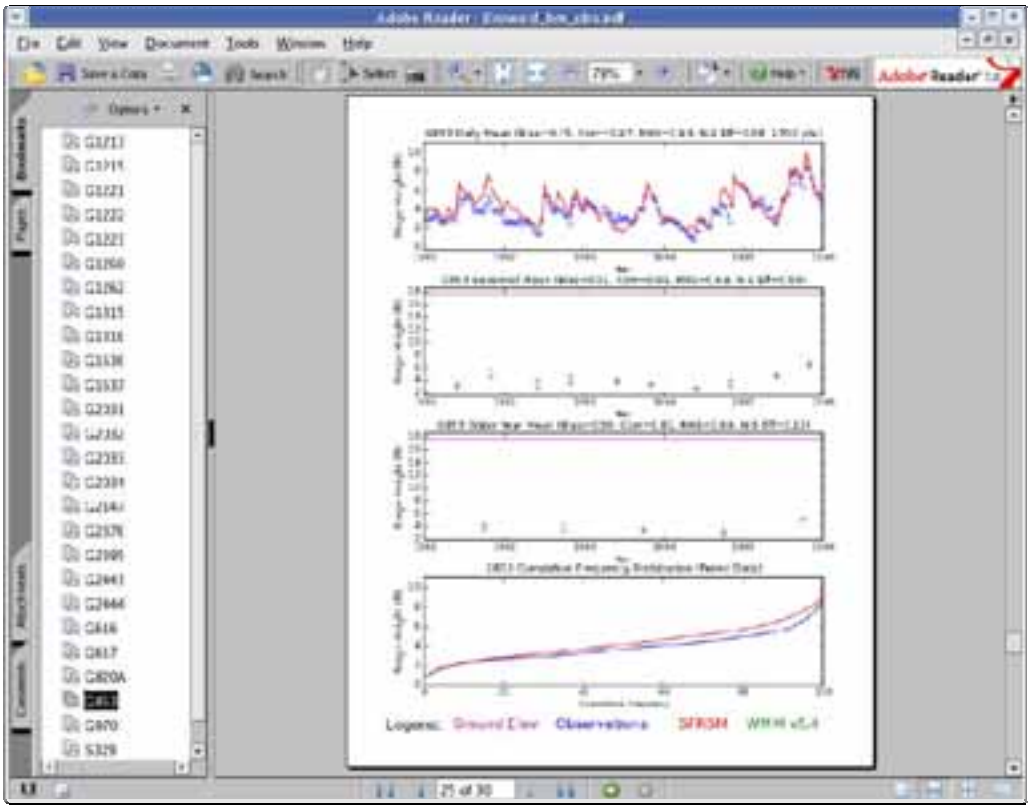

Figure 4. Post-analysis charts generated directly from netCDF into PDF format.

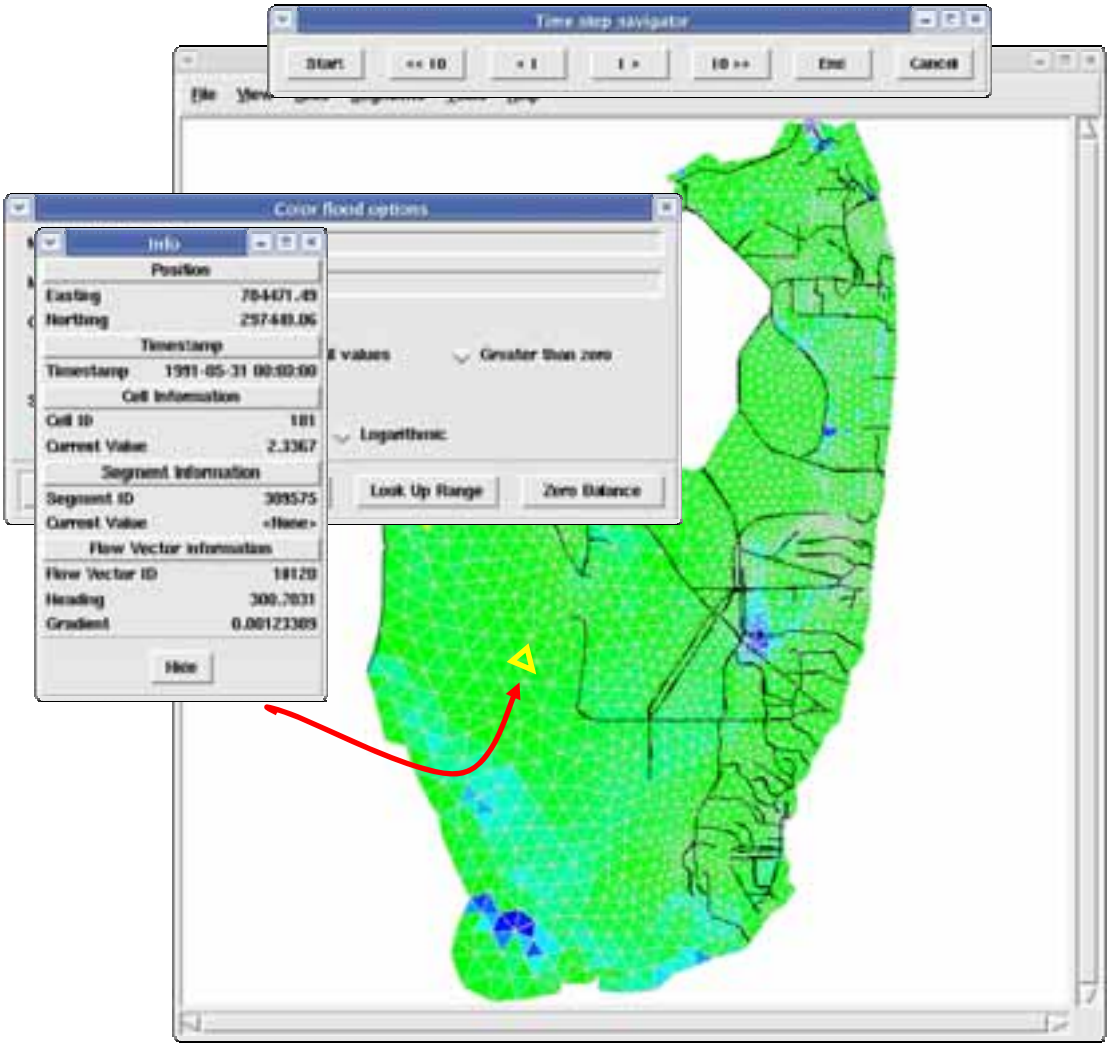

Figure 5. 2D Animation and timestep viewer, Python Post-processing tool.

Conclusion

It's more common to expect the creation of a graphical user interface to an application such as a hydrologic model to not be undertaken until the full scope of the application itself is completed. By initiating the development of the RSM GUI and establishing a database schema earlier in the model development process, measurable productivity gains have been realized and a 360 degree feedback has netted changes to the model that may not have been as easy to make once the model was competed. If the two efforts can be managed in a symbiotic manner, the two dissimilar efforts will both benefit from one another. If a more mutually beneficial approach can be realized earlier on in the development, then the two efforts can adopt a relationship where it actually benefits both efforts by working together. Being able to function side-by-side, testing real end-user applications of the model, the criteria expected by the final clients can be addressed earlier on. Unneeded optimization can be eliminated if end results can be proven to meet user expectations and developers can benefit by having a better interface to aid with model development and testing.

With a sound project management approach and agreement on processes to ensure a trend toward improvement, both efforts can be undertaken in parallel. Convergence of the two efforts, when completed, will realize benefits much earlier on and return on investment for both efforts can be substantially higher.

References

The Project Management Institute, *Guide to The Project Management Body of Knowledge*, http://www.pmi.org/info/pp_pmbok2000welcome.asp *(Jan 12, 2007)*

Carnegie Mellon, Software Engineering Institute, *Capability Maturity Model Integration Version 2.0 overview*, http://www.sei.cmu.edu/cmmi/general/general.html (Jan 12, 2007)

PYTHON.ORG, TkInter Documentation, http://wiki.python.org/moin/TkInter (Jan 12, 2007)

South Florida Water Management District, Regional Simulation Model Documentation, http://xstage.sfwmd.gov/site/index.php?id=342 (Jan 12, 2007)

Author Contact Information:

Richard Miessau, PMP
Project Manager
South Florida Water Management District
Hydrologic & Environmental Modeling Department
3301 Gun Club Rd.
West Palm Beach, FL 33406
(561)686-8800