We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Spatial Decision Support System (SDSS) for Stormwater Management and Water Quality Assessment

Nally Kaunda-Bukenya, Wubishet Tadesse, Yujian Fu,
Teferi Tsegaye and Mezemir Wagaw
Alabama A&M University & City of Huntsville, AL, Planning Division
USA

1. Introduction

Land use policy in the United States is a predominantly local issue (Giannotti & Arnold, 2002). The challenge is that land use policies and decisions are made by elected and appointed municipal officials (Stocker et al., 1999) whose training may not necessarily be in environmental management. Because of the critical importance of their work, and because they deal with land-use planning and regulation on a daily basis, local officials need decision tools that can allow them to place case-by-case land-use decisions within the broader context of the watershed. These land use managers need tools for assisting them to evaluate environmental impacts of their land-use decisions, visualize alternative scenarios, and educate their constituency (Arnold, 2000). Historically, decision makers have indicated that inaccessibility of required geographic data and difficulties in synthesizing various recommendations are primary obstacles to spatial problem solving (Ascough et al., 2002). Indeed studies have shown that the ability to produce meaningful solutions can be improved if these obstacles are lessened or removed through an integrated systems approach, such as a Spatial Decision Support System (SDSS). As Ascough et al. (2002) have observed, a SDSS makes a positive contribution to decision-makers' task if it enables them to reach: (i) a more accurate solution, (ii) a faster solution to a given problem, or (iii) both of these.

The driving force for developing this SDSS is the limited use of Geographic Information Systems (GIS) for environmental planning in municipalities. This limitation is due to the fact that, even though GIS software is available in most municipal land management data centers, it is too complex for policy makers and environmental officials to use "out of the box" without acquiring expertise in GIS. Thus, there is a need to develop custom tools that are less intimidating to non-GIS audience or users, but robust enough to perform complex geoprocessing tasks and hydrological models in the background. The goal of this Chapter is to develop an adaptive and customer-driven environmental SDSS to assist municipal officials fulfill environmental legislations and minimize the impact of pollution resulting from urban development. There are two specific objectives that we will address. The first objective is to develop a front-end graphical user interface (GUI) that is robust

enough

enough to quantify and geolocate pollution hot-spots in an urban area, but simple enough for use by land use decision-makers whose expertise is neither GIS nor water quality assessment. The second objective is to demonstrate the use of the SDSS for generating environmental compliance reports and for assessing pre-development and post-development conditions of a land use change. The rest of the chapter is organized into four sections. First, a review of previous studies is presented in section 2, the methods for developing the SDSS components in section 3. and the last two sections are the results and summary.

2. Literature review

A spatial decision support system (SDSS) is an interactive, computer-based system designed to support a user or group of users in achieving a higher effectiveness of decision making while solving a spatial problem (Sprague, 1982). According to Sprague (1982), a SDSS has three primary components: a geographic database management system for handling geographic data; a number of potential models that can be used to forecast the possible outcomes of decisions; and a user interface to provide interaction of the user to model scenarios. Similarly, Armstrong and Densham (1990) suggest that five key modules are needed in a SDSS: (i) a database management system (DBMS), (ii) analysis procedures in a model base management system (MBMS), (iii) a display generator, (iv) a report generator, and (v) a user interface.

Purdue Research Foundation (2010) developed a model called Long Term Hydrologic Impact Analysis (L-THIA). This model takes land use, soil, and long-term precipitation data as input and computes changes in recharge, runoff, and nonpoint source pollution resulting from past or proposed development as an output. The L-THIA modeling program is available in three forms as an online spreadsheet, as Avenue scripts that run as an extension of ArcView 3.x, or as an interactive mapping application developed using Java programming (Purdue Research Foundation, 2010). Although L-THIA was originally developed for municipal planners, its main focus seems to be non-point source (NPS) pollution. A similar tool is needed for addressing point source pollution in addition to NPS pollution, and for assisting municipalities with tools for environmental legislation compliance. The current research fills this gap by providing the ability to quantify point source pollution and use the newly developed user interface to generate reports for environmental compliance. The idea is that, having an environmental compliance tool that produces qualitative pollution hotspot maps or charts in addition to quantitative outputs (such as tabular data), enables decision-makers to track the environmental status of their watersheds and monitor the long-term effect of land use on the environment.

Wilkerson et al. (2010) developed a SDSS that allows users to balance watershed protection with smart growth/low-impact site development strategies. The SDSS was developed to calculate: time-varying runoff and water quality as a function of rainfall, site characteristics, and BMPs for development sites within the Southeastern U.S; and BMP cost, and compare various scenarios for effectiveness and cost. The authors used an existing Hydrological Simulation Program-FORTRAN (HSPF) for computing movement of water through a complete hydrologic cycle—rainfall, interception, evapo-transpiration, runoff, infiltration, and flow through the ground. HSPF runs on Better Assessment Science Integrating point

and Nonpoint Sources (BASINS) interface called WinHSPF. A simplified windows interface called Latis was developed by the authors as a simpler replacement of BASINS. (Wilkerson et al., 2010) The Latis model involves using specific rain event to model and compare scenarios under pre-development, and "as-built" with BMP options, and even worst case scenario (100 percent impervious). Later on, Latis, was further improved into Latis-LIDIA to estimate runoff based on pre- and post-developed site conditions using the widely-used Soil Conservation Service (SCS) runoff curve number (CN) method. The first step in the model requires user input of project information, site dimensions, and precipitation data. Precipitation data are automatically generated by selecting state and county, or manually entered by user-defined values. The precipitation database is tailored for sites within Alabama, Louisiana, and Mississippi (Wilkerson et al., 2010). The user then characterizes land use and land cover for each respective hydrologic soil group (HSG), cover type, and size. The model then generates runoff coefficients. Although this is significant contribution to the existing body of knowledge, the authors admit that the model needs further development to accommodate pollutant loading computation. This current research addresses this deficiency.

In another instance, the Decision Evaluation in Complex Risk Network Systems (DECERNS) is a similar SDSS tool which focuses on land use planning and management (DECERNS - Team, 2006). DECERNS provides spatial data visualization for vector and raster models. It is used in the development of alternatives and criteria specification; implementation of the basic and advanced multi-criteria decision; and generation of various reports, including text descriptions, tables, diagrams, and maps (DECERNS -Team, 2006). It is a powerful commercial SDSS that caters for a larger community of state and regional officials, educators, and researchers. However, local governments who make critical land-use decisions are not direct beneficiaries because of the costs associated with this commercial software and also because this system lacks specific direct benefits to municipalities such as environmental legislation compliance tools. Thus, justifying the need to develop a municipal SDSS targeted for local communities to visualize the impacts of their decisions and simultaneously fulfill environmental legislation.

In another example, Rodman and Jackson (2006) used Python programming and the ArcGIS geoprocessor to develop a standalone spatial application, for the US Army Corps of Engineers (USACE), which performs data mining in geographic datasets. Python was selected as the language of choice because it is a powerful open source cross-platform programming language, that can run on Windows, Mac, or Unix and has a wealth of available code and tools that connect to databases for developing graphical user interfaces (GUI). The authors' goal was to create an application that relies on ArcGIS for handling spatial data formats, geographic coordinate system transformation, mapping, and geoprocessing. The resulting data mining application known as *Aspect*, is used to discover association rules that describe spatial relationships between geographic features. This allows decision makers to have tools that combine knowledge of terrain, travel routes, structure, land use/cover to improve situational awareness (Rodman & Jackson, 2006).

In this research, several implementation options were examined to determine the best approach to develop the graphical user interface. The option selected would need to take

advantage of the already established ArcGIS geoprocessor for handling spatial data processing, coordinate system, and data editing. For this reason, some of the options examined for use include Esri's ArcGIS Add-ins, Tkinter/python-based GUI, a web-mapping application, and a Java-based custom desktop application. Each of these applications is briefly described below while highlighting possible shortcomings as they relate to the current research.

2.1 ArcGIS Add-ins

In the 2011 version of ArcGIS 10 software, the concept of Add-ins was introduced to expand ArcGIS desktop's functionality and extend the interface (Burke and Elkins, 2010). Using Esri's ArcObjects, one can create new custom functionality using Add-ins to create a button or tool that a user interacts with to do something with the map or with GIS data. There are several types of Add-ins ranging from menus, buttons, toolbars, dockable windows, tool palettes, or applications and extensions. Add-in extensions are invisible to the user, but are event listeners that react to events by running code attached to them. The advantage here is that the complexity of the code is hidden from the end user, making complex processes more user-friendly. Add-ins can be built with C++ programming and often require a lot less code since the programmer simply adds more functionality instead of creating a new software package. They are also portable as they can be easily shared by email or file transfer from one user to another using a few steps to install (Burke & Elkins, 2010).

There are several ways to create Add-ins. Burke and Elkins (2010) use Microsoft Visual Studio 2008 and .NET to create a button Add-in using a wizard, assign an image icon, a name, and add reference to it. Microsoft Visual Studio 2008 Express is a free version that can be used to get started with Add-in development. Add-Ins can also be built with Java, for example using Eclipse development environment. To do any Add-in development, ArcObjects Software Development Kit (for .NET if using Visual Studio) needs to be installed. An Add-in is just one file with *EsriAddin* file extension, a folder-list container that contains everything needed for it to work without much setup. Add-ins can be placed in a specific directory where ArcGIS checks every time the software is launched. In summary, the Add-in creation process in Visual Studio involves making a Visual Studio project, creating an add-in, adding type to it, writing the business logic (what it does, and code behind it), and then testing it (Burke & Elkins, 2010).

Although Add-Ins seems straightforward, a Java or C++ programming skill is required to get the full effects (Burke & Elkins, 2010). Also, they are relatively new in ArcGIS and do not have a large user community yet, thus they were not selected for implementation in this research. Alternatively, Esri's ArcGIS Engine software can be used with ArcObjects to add dynamic mapping and GIS capabilities to existing applications, build custom mapping applications, or add geoprocessing scripts using application programming interfaces (APIs) for COM, .NET, Java, and C++ (Burke & Elkins, 2010). This option was not utilized in the current research because ArcGIS Engine software package was not available. The second option that was considered using a python-based module called Tkinter.

2.2 Tkinter/python-based graphical user interface

Tkinter Tool Command Language (Tk/Tcl) is an integral part of Python that provides a platform-independent windowing toolkit that is available to Python programmers using the Tkinter module (Official Tk/Tcl website, 2010). The Tkinter module (renamed to tkinter in Python version 3.x) is the standard Python interface to the Tk GUI toolkit (Official Python Website, 2011). Tkinter is basically a set of wrappers that implement the Tk widgets as Python classes. Tcl (Tool Command Language) is a dynamic programming language that is suitable for a very wide range of uses, including Web and desktop applications, network programming, embedded development, testing, general purpose programming, system administration, database work, and many more (Official Tk/Tcl website, 2010). (Official Tk/Tcl website, 2010).

Tk, a graphical user interface toolkit can be used in Tcl or in Perl language to create a number of GUI components such as buttons, labels and canvas. These components are known as widgets (Official Tk/Tcl website, 2010). Once these widgets have been created, three geometry managers are used to display them in relation to each other: pack, grid, and place. The "pack" geometry manager allows one to place your widgets in a row or column. "Grid" geometry manager allows the placement of widgets in a matrix. The third geometry manager, "place" provides the ability to place widgets by pixel or by the proportion of the way across the window that the programmer wants them to appear. More complex windows can be built using frames or nested frames (Well House Consultants, 2006). The python-based Tkinter appears to be a good option since the underlying geoprocessing scripts developed for this research are also Python-based. The limitation with this application is that the spatial components of the GUI such as coordinate system handling and spatial analysis would need to be programmed, which could be time-consuming. The third option examined for the GUI was a web-mapping application.

2.3 Web mapping application

Esri's software, ArcGIS Server enables the user to create, manage, and distribute GIS services over the Web (Esri, 2011). Different Application Programming Interfaces (API) are available for web application development on various platforms. The Esri APIs include JavaScript, Flex, Silverlight, and Java Web Application Development Framework (ADF), and the NET Web ADF (Esri, 2011). ArcGIS Explorer Online is another option to present web map services, add other content to it, navigate, present and share the map. Explorer Online makes it possible to disseminate work on the Web and integrate map services from various sources (Esri, 2011). ArcGIS Explorer Online allows one to save maps to ArcGIS.com and choose to save them privately, share with a group or share over the Internet (ArcGIS Explorer Online Team, 2011). While the web-mapping approach is not emphasized in this research, a prototype web-mapping application was developed for spatial data editing and for interactive mapping. The next section further describes the SDSS development methodology.

3. Spatial Decision Support System methodology

The overall methodology for this research encompasses a three-tier approach that leads to the development of a spatial decision support system. The three levels are the data level, the

model development level, and the development of a graphical user interface. The data level was implemented primarily for creating a comprehensive database that stores all the data needed for water quality evaluation. Modeling the effects of land use change on water quantity and quality requires a multidisciplinary approach which incorporates many different data types. Therefore, several data sets such as sewer infrastructure data, rainfall data, soils/curve number, and pollutant sampling data were compiled into a central geodatabase (geographic geodatabase).

3.1 Database component

Geodatabase development for stormwater management in particular can be challenging due to the multitude of different types of stormwater features involved, and the complex topological relationships that exist between them. In this research a geodatabase schema was developed that can be adopted by other municipalities as a template for stormwater mapping. Specifically, ArcGIS software suite by Esri was used to develop an enterprise level geodatabase for managing the watershed and subsurface infrastructure data. Geodatabases are needed for the successful storage, access, retrieval, manipulation, and management of massive data sets typical of municipalities. The most critical data set emphasized in this research was the sewer infrastructure data. Specifically, two separate datasets were developed for stormwater networks and sanitary sewer (wastewater) networks. A dataset is a set of georeferenced data layers that are topologically related and are in the same spatial extent (Esri, 2010). The stormwater dataset includes locations of stormwater inlets, pipes, headwalls, and culverts captured using mapping-grade Global Positioning Systems (GPS), as well as creeks, rivers, and delineated drainage basins from topographic mapping. Similarly, the sanitary sewer database consists of GPS locations and dimensions of manholes, sewer lines, pump stations, wastewater treatment plants, and sewer drainage basins. Network topology rules were developed to enforce the connectivity of the sewer features such that all features that participate in a network are topologically connected. For example, if an inlet is not connected to any pipe or waterway, it would be marked as an error. Similarly, attribute validation rules are also established for the sewer features to minimize errors when editing the sewer data. After the database design the next step was to populate the "spatial container" with data from multiple sources including GPS, aerial photos, engineering design drawings, and digital elevation models. Once all the data had been collected, the next step in the SDSS development was model development.

3.2 Model development component

The model development stage is critical to the full-functioning of any SDSS because data has to be processed to make it meaningful for decision-making. The general methodology of the model development stage is as follows:

- Stormwater outfalls were extracted by querying for stormwater pipes (diameter of at least 12 inches in industrial areas and at least 36 inches in all other land uses) that empty into major rivers and creeks. This query is based on the definition of an outfall by the United States Environmental Protection Agency (EPA, 1992).
- Drainage boundaries were delineated using topographic elevation and the spatial and topologic locations of underground and above-ground stormwater infrastructure. The

result was a delineation of basins, subbasins, and mini-basins. It is important to note that mini-basins are categorized into outfall basins (point source of pollution) and diffuse basins (non- point source of pollution).

- Land use characterization was achieved by performing a spatial union of an existing land use layer with the mini-basin polygons. This resulted in one of the parameters in the hydrological model; area values of each land use type in each mini-basin.

- A geospatial approach for hydrological modeling was developed using python programming to determine the water quality and runoff effect of land use change.

The uniqueness of the model component of the SDSS is in the integration of Java and Python programming languages. The user interface was written using Java, but an existing hydrological model was programmed into a geospatially-enabled hydrological model using Python programming. The hydrological model adopted is a series of three equations based on "The Simple Method" by Schueler (1987), often called the Curve Number Method. The equations were spatially enabled by encoding them into ArcGIS scripting environment called arcpy (python for ArcGIS). First, an area-weighted runoff coefficient (weighted C-value) script was written using Python's mathematical operations and program looping to calculate weighted runoff coefficient (Rvi) for each outfall basin (**equation 1**).

$$Rv_i = \sum (A_i * Rv) / \sum A_i \tag{1}$$

Where *Rv*= Runoff coefficient for each land use within the outfall drainage area.

A_i=Land area of each land use within the drainage area of the major outfall. Next, Event Mean Concentration (EMC) of each pollutant was calculated using the result from equation 1 above.

$$EMC = \sum (A_i / A_T) * (Rv / Rv_i) * C_i \tag{2}$$

Where *A_i*=Area of a specific land use within the outfall drainage basin

- *A_T*=Total land area within the drainage area of the major outfall
- *Rv*=Runoff coefficient (C-value) for a particular land use
- *Rv_i*=weighted C-value for the drainage area of the major outfall
- *C_i*=measured pollutant concentration in each land use

Finally, the pollutant loadings (weight of pollutant per season) were calculated using the following equation:

$$Li = [(P * CF * Rvi) / 12](EMC * Ai) \tag{3}$$

Where *Li*= Seasonal pollutant load in pounds per outfall/season

- *P*= Precipitation (inches/season)
- *CF*=Correction factor that adjusts for storms which produce no runoff
- *Rvi*=Weighted runoff coefficient for the area drained by each outfall
- *EMC*=Event Mean Concentration of pollutant in milligram/liter
- *Ai*=Land area drained by each major outfall (acres)
- 12 and 2.72 are unit conversion factors.

These three equations have been sequentially executed and the resulting table would have the output that is used to create thematic maps that show pollutant hotspots in the study

area. The equations would be encapsulated in the user interface and simplified as a one-button click. The next section describes the user interface in more detail.

3.3 SDSS GUI implementation

The Java-based custom desktop application was selected for the development of the SDSS user interface. GUI programming is a complicated task involving multiple steps, because many interface widgets need to be generated and specified. Furthermore, the alignment and appearance of these interface widgets also required a lot of programming. The GUI was developed using version 1.0.23 of Java Software Development Kit (JDK) and version 6.9.1 of NetBeans Integrated Development Environment (IDE). NetBeans and JDK are free programs obtained from NetBeans' website (NetBeans official website, 2011). NetBeans is an IDE that simplifies the programmer's task and ease the burden of GUI programming. To make the GUI components produce results that react to user input, many kinds of events must be handled properly and programmed to make the widgets functional. Almost every GUI widget is a generator that is capable of initiating multiple kinds of events. Java uses "event listener" to capture events that are generated from user interactions. A "Listener" is a an interface that a programmer can put all information that is needed to make the buttons functional. An event object is a holder of generated events. The "Listener" can only capture the registered event handler. By this observer pattern, a Java GUI program can properly handle and process user interactions, providing a flexible and extendable programming strategy.

In our GUI development, the typical events generated were action events and mouse events. An action event is generated when a button is pressed, or by pressing *Enter* in a text field, or when a menu item is selected. (Schildt, 2001). The programmer has to implement an action listener to define what methods are invoked when a user performs certain operations. With registered necessary event handlers, an "actionPerformed" message is implemented to handle all generated events on the relevant component. For example, the *Save/Add* button on the user interface listens for the user to click the button, then performs the save operation to save the user input from the text field into the corresponding table in the basins ArGIS personal geodatabase (Microsoft Access database).

Swing components were developed by SUN Microsystem's, to provide a more sophisticated and user-friendly GUI programming paradigm, including frames, buttons, panels, text fields, and labels (NetBeans official website, 2011). **Figure 1** shows the component hierarchy of the Java Swing as illustrated by Reddy (2007). Other modules used include input/output (java.io), abstract windowing toolkit event (java.awt.event), and java.lang.reflect methods. Due to the feature of not invoking OS resource, Swing is considered as lightweight component that is extended on top of many widgets of an AWT packet.

Figure 2 illustrates a use-case scenario that shows how a user would interact with the system and how the system responds to the user's interaction. In general, the SDSS allows the users to:

1. input data into simplified forms on the interface and save edits at the click of a button,
2. use one-button click to run complex geoprocessing in the background, and
3. use one-button click to generate maps and output reports that are needed for informed decision-making and for environmental legislation compliance.
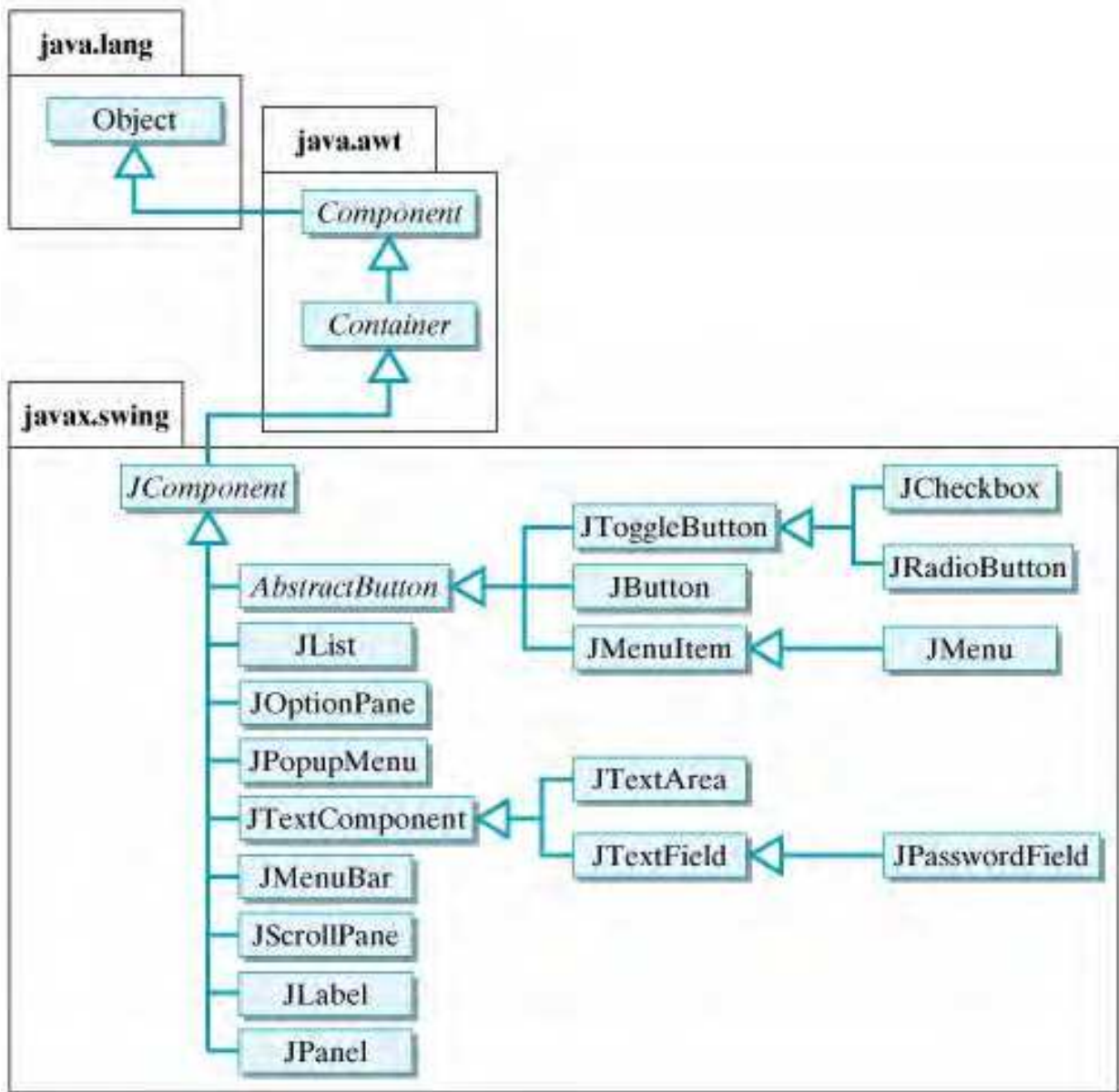
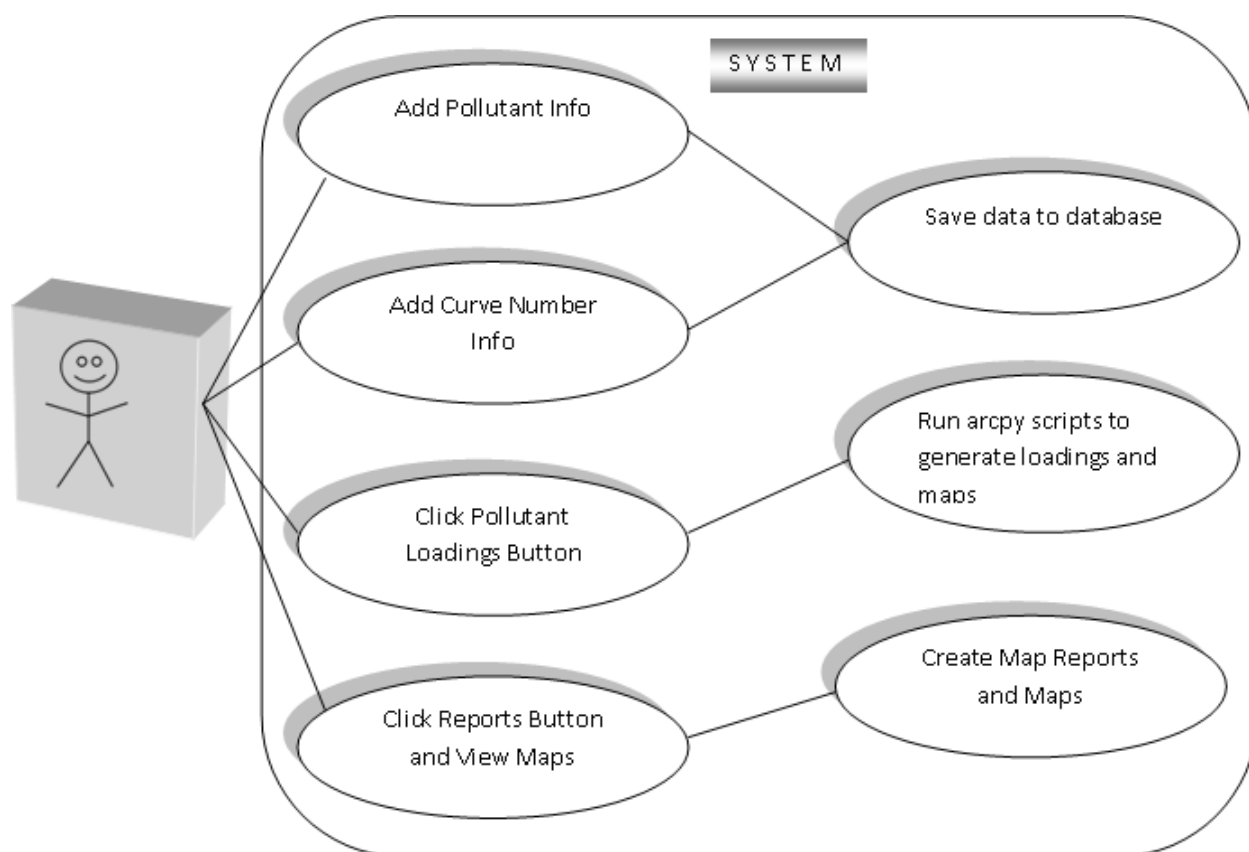Fig. 1. The hierarchy of Java Swing Components

Fig. 2. Use Case Diagram showing how a user interacts with the interface

4. Results

A graphical user interface for the SDSS was developed for non-GIS professionals as a front-end for data collection as well as for executing hydrological modeling scripts. The developed geospatially-enabled hydrological model can optionally be executed in ArcGIS as a Python script tool, but the complexity of the ArcGIS interface **(Figure 3a)** has been simplified for non-GIS users into a simple, straightforward interface as shown in **Figure 3b.** The resulting GUI allows the complex pollutant loadings scripts to be executed in a less intimidating environment.

The *Calculate Loadings* in Figure 3b button runs a Java command that executes arcpy scripts (ArcGIS 10 Python scripts), which then progress by invoking classes and methods on the ArcGIS geoprocessor object. Specifically, the *Generate Loading* scripts fetch pollutant data such as pollutant concentration and curve number information entered by the user through the Java interface and use that as input for calculating the Estimated Mean Concentrations and pollutants loadings. Pollutant hotspot maps indicating the spatial distribution of outfall basin pollutant loadings are also generated upon pressing the *Calculate Loadings* button. The *Loadings Maps* button fetches the generated maps and displays them using a PDF reader such as Adobe Reader, ready for inclusion in reports, and for immediate decision-making. The user also has the option to press the *Generate Report* function on the file menu executes Java commands to generate water quality reports based on EMC's and Pollutant loadings fields from the ArcGIS geodatabase. When clicked, the *Open Online Maps* button opens a

web browser showing a FlexViewer web map developed by the authors for viewing and editing geographic data associated with the SDSS. Finally, the *Open GTViewer* button opens an internal GIS desktop application (by Graphics Technologies, Inc.) that the City of Huntsville uses.

With this application, non-GIS-expert users can collect pollutant data such as pollutant name, concentrations, etc., and save into the master database tables via the *Pollutant Editor* form. The data entry is developed in such a way that the pollutant data can be edited by adding new records or editing existing records, **Figure 4** shows the pollutant editor form for this purpose.

Since different municipalities use different land use classification schemes, they may find it easier to manually enter literature-based runoff coefficients from a land use lookup table. For this reason, a runoff coefficient editor was created for manually populating C-Values/runoff coefficients; **Figure 5** shows the *Curve Number Editor* form.

The core of the model component is encapsulated in the "*Calculate Loadings*" Button on the GUI. When this button is pressed, the hydrological modeling equations (1-3) described in the model development stage would be executed. **Figure 6** shows the two buttons for processing pollutant loading calculations and for viewing output maps. To demonstrate the SDSS application, two applications were illustrated: a report generation function and a comparison of pre-development and post-development pollution contributions.
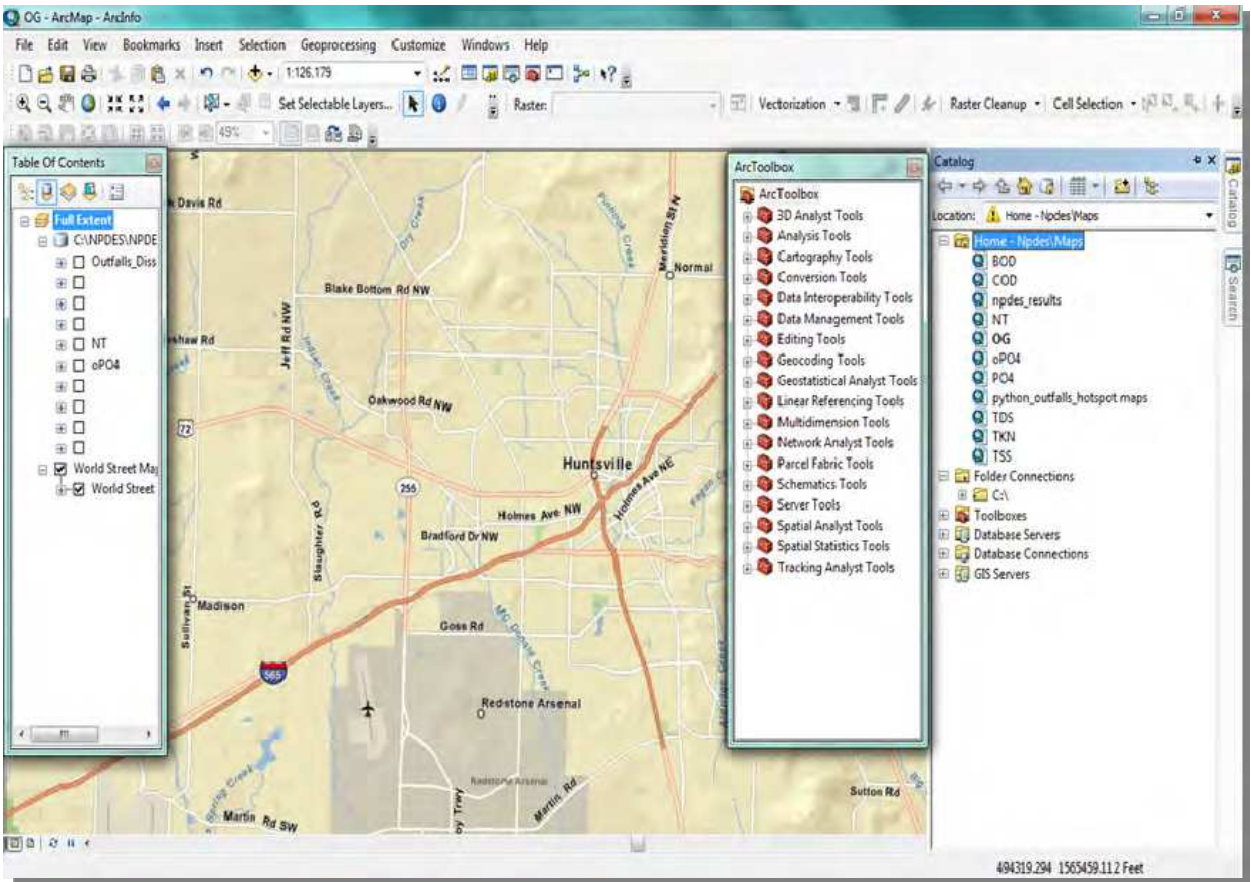

Fig. 3a. ArcGIS 10 interface

Fig. 3b. The newly developed, less intimidating graphical user interface

Fig. 4. Pollutant Editor Form

Fig. 5. The Curve Number Editor

Fig. 6. Pollutant loadings and output map button

4.1 SDSS Demonstration 1: Report generation and pollution contributions of land use change (Bridge Street Town Center)

First, a report generation function for municipal environmental compliance and pollution contributions of a land use change are illustrated. To test the report function, a report was generated for the City of Huntsville stormwater outfalls using 2004 data. The *Generate Report* function on the file menu generates a report ready for submission to the United States Environmental Protection Agency (US EPA) for stormwater regulation compliance. Stormwater pollution is regulated under the Clean Water Act (CWA) of 1972. Under the CWA, the US EPA has implemented pollution control programs and set standards that make it unlawful for industries, municipalities, and other facilities to discharge any pollutant into navigable waters, without a permit (US Congress, 1972). US EPA's National Pollutant Discharge Elimination System (NPDES) permit program controls these point source discharges and has put specific regulation in place as a guide for NPDES permit *applicants* (US Congress, 1972). **Figure 7** illustrates the report generation workflow. The user interface simplifies this process for land use decision-makers by using just a few clicks to quantify pollution and generate maps for decision support or reports for environmental legislation programs such as NPDES permit compliance. Instant report generation saves time for municipal officials so that they focus more on decision-making instead of technical setbacks. The added value of the geospatially-enabled hydrological model is the ability to produce pollutant hotspot maps that unveil spatial trends, allowing land use policy makers to visualize the environmental impact of their decisions.
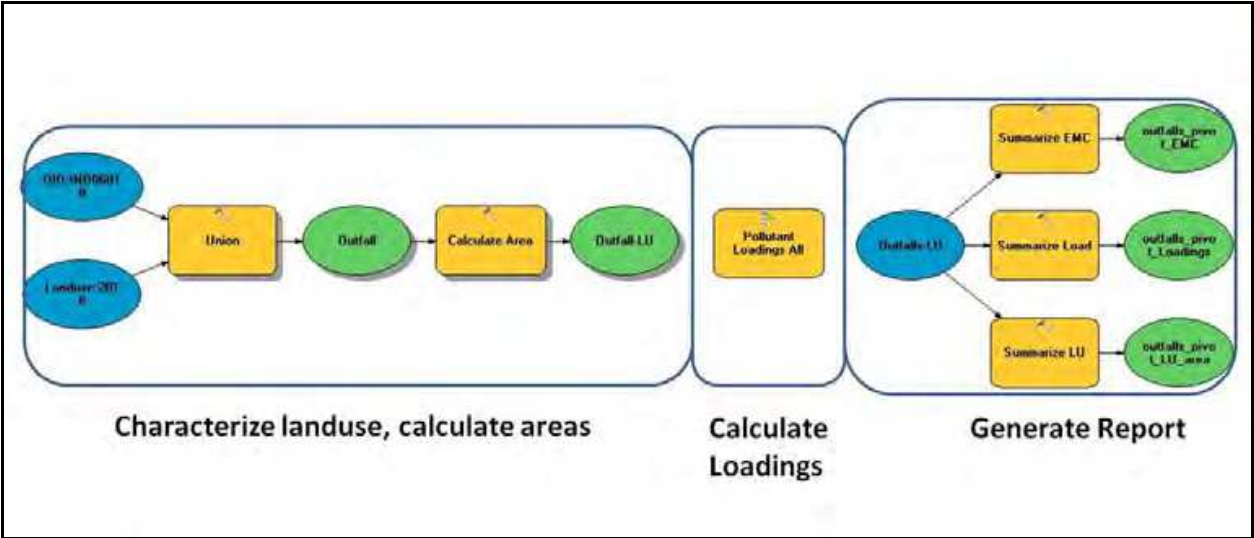

Fig. 7. Complete workflow showing land use characterization, model execution, and report generation

The first set of processes in **Figure 7** calculate the land use acreage in each mini-basin, the second calculates EMC's and pollutant loadings, and the last step selects outfall polygons and generates summarized Pivot tables with a report output. The report includes the stormwater outfall ID, weighted runoff coefficients, event mean concentration of pollutants, and pollutant loading in each outfall basin. **Tables 1-3** are respectively, land use summaries, Event Mean Concentrations, pollutant loadings generated as part of the report. **Figure 8** is an example of pollutant hotspot maps generated using the tools from the user interface.

OUTFALL_ID	Tot_Area	Rvi	Low_Density	Forest	Recreation	Transportation	Urban_Open
ALD03001	6.828	0.40	6.83	0.00	0.00	0.00	0.00
ALD03002	25.045	0.39	23.19	1.85	0.00	0.00	0.00
ALD04001	44.683	0.37	34.44	10.23	0.01	0.00	0.00
ALD04002	27.586	0.39	25.26	0.36	0.00	0.49	1.48
ALD05001	101.876	0.27	16.90	84.97	0.00	0.00	0.00
ALD05002	44.346	0.27	7.26	37.09	0.00	0.00	0.00
ALD06001	27.607	0.52	0.75	0.00	0.01	2.06	10.38
ALD06002	179.886	0.46	97.83	16.47	4.02	18.06	8.28
ALD06003	6.501	0.60	0.08	0.00	0.00	0.00	0.00
ALD06004	8.751	0.53	0.01	0.00	1.67	0.00	0.00
ALD06005	5.738	0.59	0.00	0.00	0.20	0.00	0.00
ALD07001	14.053	0.39	13.57	0.00	0.00	0.00	0.00
ALD07002	53.039	0.42	41.64	2.00	0.00	0.00	0.00
ALD07003	141.464	0.44	76.32	24.35	0.06	0.00	0.00
ALD07004	36.601	0.49	18.33	0.00	0.31	4.90	0.00
ALD07005	95.606	0.39	86.50	5.65	0.00	0.00	1.17
ALD07006	79.904	0.39	75.48	0.00	0.00	0.00	2.36
ALD07007	20.849	0.39	19.51	0.00	0.00	0.00	1.22
ALD07008	11.493	0.57	0.00	3.11	0.00	1.27	0.00
ALD07009	68.361	0.40	51.92	1.29	4.69	1.43	3.25
ALD07010	136.089	0.38	67.64	33.98	0.00	2.36	16.01
ALD07011	20.149	0.52	12.01	0.00	0.00	0.17	0.00

Table 1. Land use area of each mini-basin in acres. (Rvi = weighted runoff coefficient)

OUTFALL_ID	Tot_Area	Rvi	EMC_BOD	EMC_COD	EMC_TSS	EM
ALD03001	6.83	0.40	13.50	19.38	42.25	
ALD03002	25.05	0.39	13.13	20.36	43.57	
ALD04001	44.68	0.37	12.27	22.61	46.60	
ALD04002	27.59	0.39	12.89	20.45	42.41	
ALD05001	101.88	0.27	7.56	35.02	63.30	
ALD05002	44.35	0.27	7.54	35.08	63.39	
ALD06001	27.61	0.52	7.69	35.27	63.12	
ALD06002	179.89	0.46	9.95	26.71	46.96	
ALD06003	6.50	0.60	8.37	18.51	13.90	
ALD06004	8.75	0.53	8.10	20.42	18.74	
ALD06005	5.74	0.59	8.29	18.82	14.51	
ALD07001	14.05	0.39	13.50	19.38	42.25	
ALD07002	53.04	0.42	12.19	20.57	39.51	
ALD07003	141.46	0.44	10.49	25.24	47.18	
ALD07004	36.60	0.49	9.52	22.42	31.59	
ALD07005	95.61	0.39	13.00	20.41	43.18	
ALD07006	79.90	0.39	13.35	19.77	42.78	
ALD07007	20.85	0.39	13.21	20.15	43.29	
ALD07008	11.49	0.57	7.40	36.18	65.65	
ALD07009	68.36	0.40	11.88	24.62	50.71	
ALD07010	136.09	0.38	10.28	27.33	51.52	
ALD07011	20.15	0.52	10.82	34.49	76.71	

(Rvi = weighted runoff coefficient, EMC= Event Mean Concentration, BOD=Biological Oxygen Demand, COD=
TSS= Total Suspended Solids, TDS = Total Dissolved Solids, PO4 = Phosphates)

Table 2. Annual Event Mean Concentration of each pollutant in each mini-basin

OUTFALL_ID	Tot_Area	Rvi	BOD_Load	COD_Load	TSS_Load	TDS_Load	PO4_Load	oPO4_Load
ALD03001	6.8	0.4	424.6	609.4	1328.9	2815.0	27.4	23.1
ALD03002	25.0	0.4	1472.4	2283.3	4887.1	9975.3	93.9	78.7
ALD04001	44.7	0.4	2308.9	4253.3	8766.9	16486.9	142.0	117.4
ALD04002	27.6	0.4	1615.2	2562.5	5315.1	11339.0	102.3	85.8
ALD05001	101.9	0.3	2438.3	11293.9	20414.0	25952.4	97.5	63.6
ALD05002	44.3	0.3	1056.6	4918.8	8886.8	11277.4	42.1	27.3
ALD06001	27.6	0.5	1272.8	5839.1	10449.3	12595.8	66.8	69.8
ALD06002	179.9	0.5	9417.9	25294.8	44464.3	83974.1	551.2	486.9
ALD06003	6.5	0.6	374.5	827.9	621.9	2902.0	19.0	23.8
ALD06004	8.8	0.5	434.9	1096.9	1006.5	3536.8	21.1	26.1
ALD06005	5.7	0.6	321.9	730.9	563.6	2518.8	16.1	20.3
ALD07001	14.1	0.4	844.0	1211.3	2641.5	5595.5	54.5	46.0
ALD07002	53.0	0.4	3092.6	5218.3	10025.7	21275.2	192.8	171.4
ALD07003	141.5	0.4	7466.8	17965.6	33583.7	55109.0	456.6	395.0
ALD07004	36.6	0.5	1969.3	4641.0	6537.6	18070.8	114.5	102.7
ALD07005	95.6	0.4	5577.3	8753.6	18519.9	37862.9	354.9	298.5
ALD07006	79.9	0.4	4732.6	7009.2	15167.2	31648.1	304.2	255.7
ALD07007	20.8	0.4	1233.3	1881.9	4043.2	8316.6	78.8	66.2
ALD07008	11.5	0.6	554.3	2709.7	4916.4	5692.8	31.2	29.4
ALD07009	68.4	0.4	3774.8	7823.7	16111.5	27751.5	235.4	200.3
ALD07010	136.1	0.4	6155.1	16354.0	30834.3	49845.4	352.0	303.1
ALD07011	20.1	0.5	1316.2	4196.0	9332.4	9573.9	85.4	79.1
ALD07012	73.7	0.4	3852.1	7301.4	13761.6	29609.7	233.1	196.4
ALD07013	36.9	0.4	2252.1	3232.2	7048.4	14930.9	145.6	122.6
ALD07014	15.3	0.4	933.1	1339.2	2920.2	6186.0	60.3	50.8
ALD08001	104.7	0.4	4847.8	10505.1	20265.3	38260.8	285.1	232.4
ALD08002	140.0	0.4	7269.9	13911.8	25757.2	52707.2	435.1	380.6
ALD08003	63.6	0.4	3952.2	5672.1	12368.8	26201.3	255.4	215.2
ALD08004	31.7	0.4	1747.0	3057.1	6056.4	12992.0	108.3	90.2

Table 3. Annual pollutant loadings for each pollutant in each mini-basin in pounds/year. (oPO4_Load = orthophosphates, other pollutant acronyms as described in Table 2)

4.2 SDSS Demonstration 2: Pollution contribution of land use change

The second application of the SDSS is the assessment of pre-development (2004) and post-development (2010) conditions of an existing commercial establishment to evaluate the pollution contribution of Bridge Street Town Center in Huntsville, Alabama. Bridge Street is located in Cummings Research Park and is mostly a commercial development that also has condominiums, hotels and recreational facilities. TThis section demonstrates the use of the SDSS by comparing Pre and Post Development conditions of mini-basin IND06018 which encompasses Bridge Street Town Center. In 2004 Bridge Street did not exist, but in 2010 the land was highly developed as shown in **Figure 9.**

The land use changes shown in **Figure 10** indicate that cropland (-38%) and campus/institutional (-30) land use areas decreased while all other land uses increased. The highest land use change after the development was commercial, at 41% increase. **Figure 11** shows that runoff coefficients increased by 12%, and the event mean concentration for all pollutants increased, with Oil and grease showing the highest increase (50%). The increase

in runoff coefficients can be attributed to an increase in impervious surface as the area was mostly cropland in 2004 and in 2011 it is mostly urban. The higher oil and grease increase can be attributed to the increase in parking areas where oil leaks are possible from parked vehicles, and grease from the commercial establishments in the town center are inevitable. The percent change in pollutant loadings for oil and grease were also analyzed and shown in **Figure 12.** Consequently, the loadings for oil and grease also increased for the dry, wet, transitional and annual seasons.
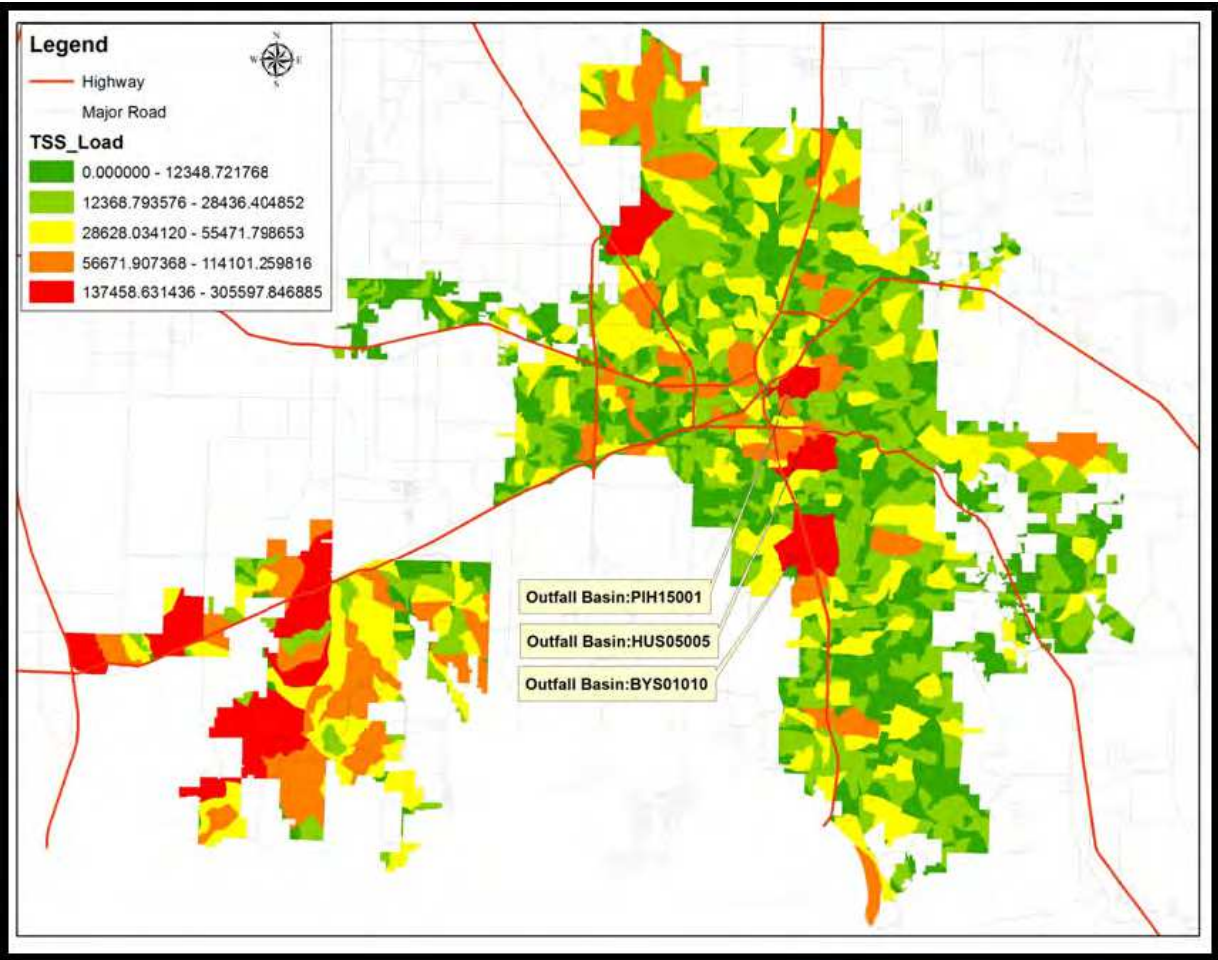
Fig. 8. Example of hotspot map for Total Suspended Solids (TSS) loads in Huntsville, Alabama (units in pounds/year) The map is presented in State Plane Coordinate System, North American Datum of 1983, Alabama East FIPS 0101.
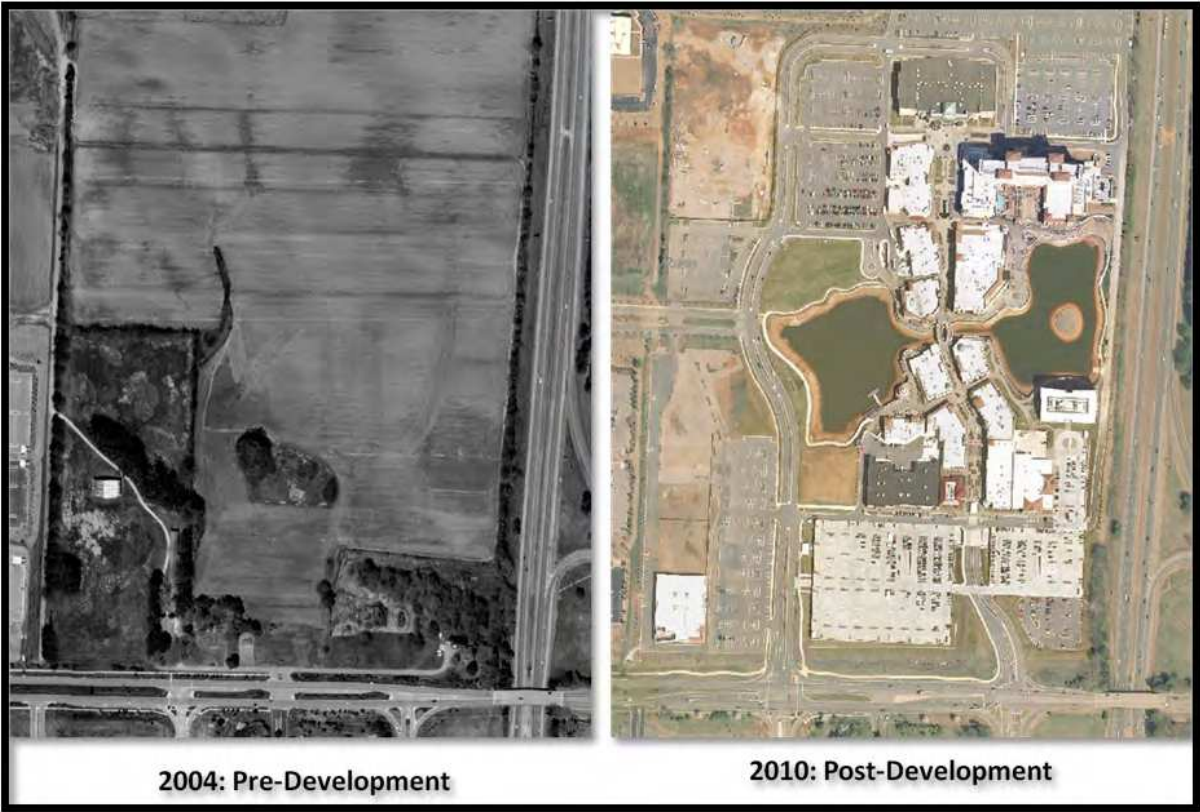
Fig. 9. 2004 and 2010 Orthophotos at the Bridge Street Town Center location: pre and post development
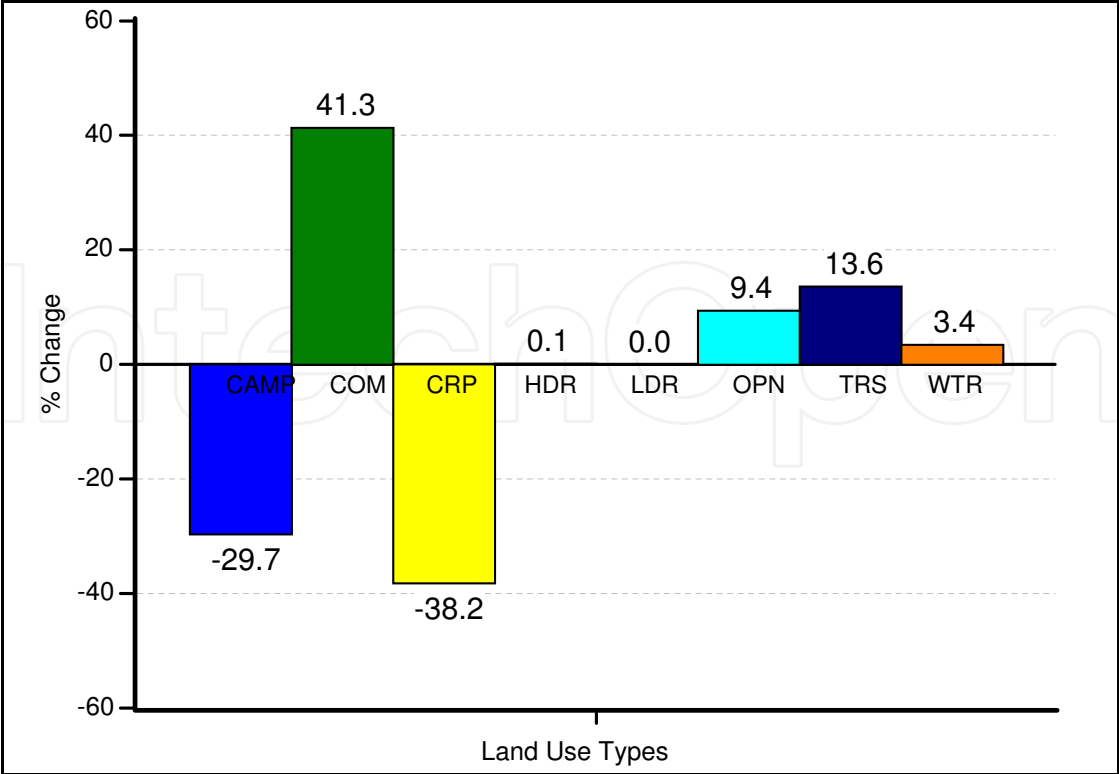
Fig. 10. Percent Change in Land use Acreage after Bridge Street Development
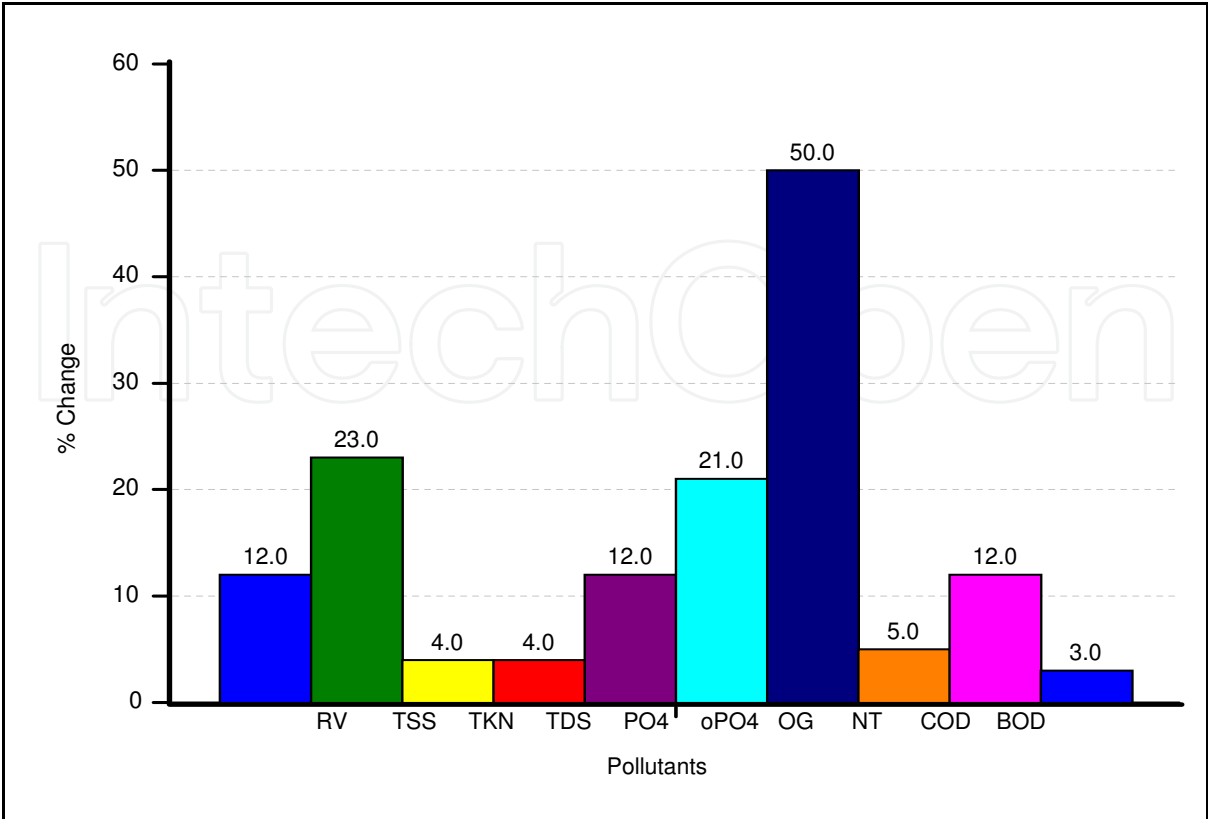
Fig. 11. Percent change in runoff coefficients and Event Mean Concentrations
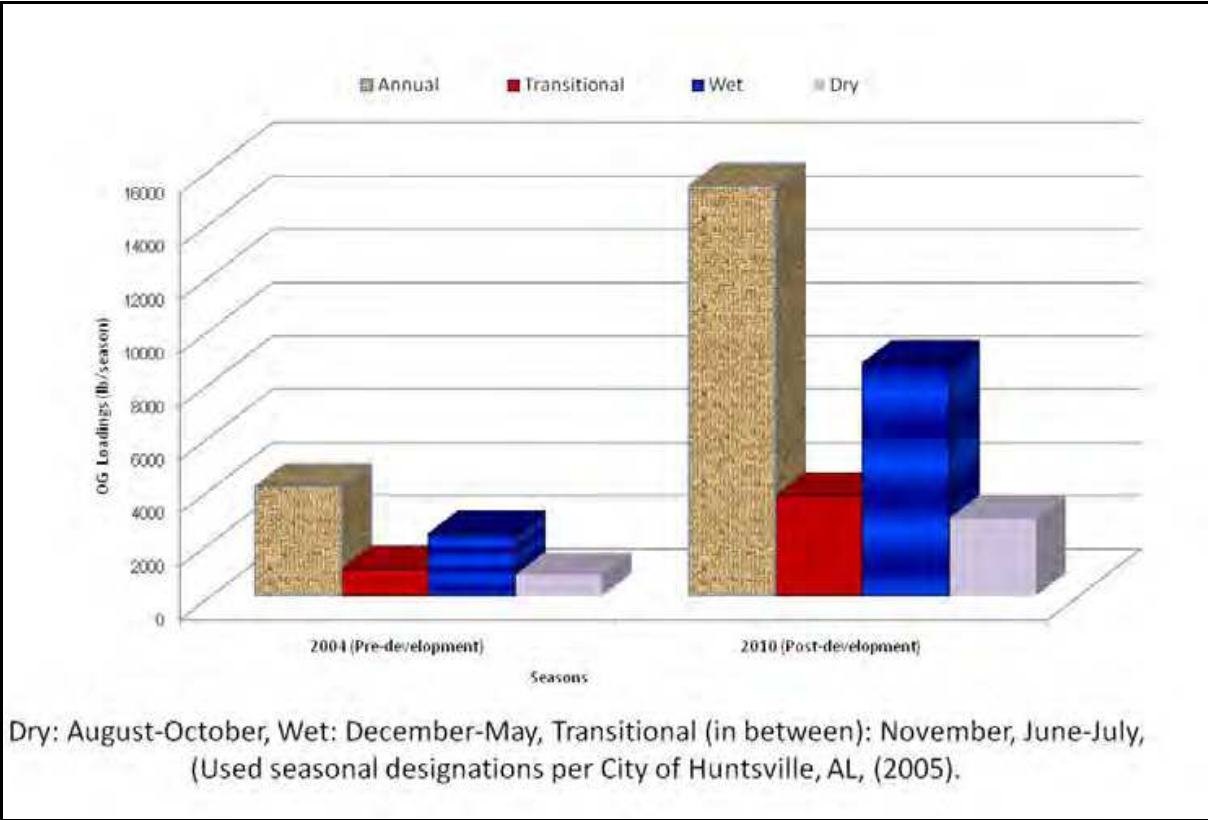
Fig. 12. Change in Seasonal Pollutant Loadings for Bridge Street area

5. Summary and conclusions

The first objective of this study was to develop a front-end graphical user interface (GUI) that is robust enough to facilitate data collection, quantify and geolocate urban pollution, but simple enough for land use decision-makers, whose expertise is neither GIS nor water quality assessment. The second objective was to demonstrate the use of the SDSS for generating reports and assessing pre-development and post-development conditions of a land use change. A desktop application has been designed and implemented using Java programming in NetBeans IDE. The GUI is a user-friendly interface that conceals program details, saving the user valuable time from focusing on technical complications, while still getting a powerful tool for the intended needs. The GUI provides custom tools for quick data input, spatial analysis, report generation, and environmental regulation compliance. The output is a graphical user interface for municipal officials or other land-use decision-makers and watershed managers for visualizing and quantifying the effects of land use on the environment. A robust, but user-friendly custom interface for local land-use officials is necessary for decision-makers to be more environmentally aware and to channel resources where they are needed the most.

One shortcoming of the desktop application is that it only allows for pollutant and curve number editing, but does not support editing GIS data. As a result, a web-mapping application is under development to address this limitation. A link to the website was established in the user interface using the *Open Online Maps* button. The web mapping application was developed using Esri's FlexViewer API and hosted on the City of Huntsville's ArcGIS server as a prototype that is not yet available for public view. The web map can be loaded by field inspectors onto a mobile device or compatible smart phones to edit and modify GIS data, sending data back to the master database via a cell phone network or the Internet. Similarly, it can be used by local citizens to pinpoint incidents such as locations of illegal pollutant discharge, sanitary sewer overflows, or flooding complaints that may result in pollution.

6. References

ArcGIS Explorer Online Team. (2011). Online Electronic Documentation. Accessed February 4, 2011. Available from http://explorer.arcgis.com/

DECERNS–Team. (2006). Decision Evaluation in Complex Risk Network Systems. Accessed 2/4/11. Available from http://195.112.127.216/?q=rd

EPA. (1992). Guidance Manual for the Preparation of Part 2 of the NPDES Permit Applications for Discharge from Municipal Separate Storm Systems. *EPA Office of Water*. Washington, DC.

Esri. (2010). ArcGIS Server Technology. Accessed February 10, 2010. Available from http://www.Esri.com/

Esri. (2011). Add-ins for ArcGIS Desktop. Accessed March 26, 2011. Available from http://blogs.esri.com/Dev/blogs/arcgisdesktop/archive/2010/05/05/Add_2D00_ins-for-ArcGIS-Desktop-10.aspx

Giannotti, L A. & Arnold, C L. (2011). Changing Land use Decision Making One Town at a Time. *North Carolina State University.* Accessed March 1, 2011. Available from http://nemo.uconn.edu/publications/about_nemo/changing_land use_10year.pdf

NetBeans Official Website. (2011). NetBeans Integrated Development Environment. Accessed January 25, 2011. Available from http://NetBeans.org/

Official Python Website. (2011). Python Programming Language. Accessed January 25, 2011. Available from http:// www.python.org

Official Tk/Tcl website. (2011). The Tcl Developer Exchange. Accessed January 25, 2011. Available from http://www.tcl.tk/

Purdue Research Foundation. (2010). Long Term Hydrologic Impact Analysis. Accessed December 8, 2011. Available from
https://engineering.purdue.edu/~lthia/

Reddy, D. (2007). UI's and Java Swing -Java Swing Components Hierarchy. Accessed Feb. 4, 2011. Available from
http://www.google.com/imgres?imgurl=http://jroller.com/DhilshukReddy/reso urce/JavaSwing/JavaSwingCompoentsList.PNG

Rodman L C. & Jackson, J. (2006). Creating Standalone Spatially Enabled Python Applications using the Arcgis Geoprocessor. Accessed Dec. 8, 2010. Available from http://proceedings.esri.com/library/userconf/proc06/papers/papers/pap_1091. pdf

Schildt, H. (2001). *The Complete Java Reference: Java 2 Fourth Edition*. New York: Osborne/McGraw-Hill.

Singh,V P. & Frevert, D.K. (2010). Hydrologic Modeling Inventory. Texas A&M University and the Bureau of Reclamation. Accessed December 8, 2010. Available from http://hydrologicmodels.tamu.edu/

Sprague, R. H. & Carlson, E.D. (1982). Building effective Decision Support Systems. Englewood Cliffs, N.J.:Prentice-Hall, Inc.

Stocker, J., C., Prisloe, A.S. & Civco, D. (1999). Putting Geospatial Information into the Hands of the Real Natural Resource Managers. *Proceedings of the 1999 ASPRS Annual Convention*, Portland, Oregon. 1070-1076.

U.S. Congress. (1972). Federal Water Pollution Control Act (33 U.S.C. §1251 et seq. Accessed September 18, 2010. Available from http://epw.senate.gov/water.pdf

Well House Consultants. (2006). Tk - Laying Out Your GUI with Frames, Pack And Grid. Accessed Feb. 1, 2011. Available from
http://www.wellho.net/mouth/787_Tk-laying-out-your-GUI-with-frames-pack-and-grid.html

Wilkerson, G W., McAnally, W.H., Martin, J.L. Ballweber, J.A., Pevey, K.C., Diaz-Ramirez, J. & Moore, A. (2010). Latis: a Spatial Decision Support System to Assess Low-Impact Site Development Strategies." *Advances in Civil Engineering* Volume 2010. Accessed March 18, 2011. Available from
http://www.hindawi.com/journals/ace/2010/810402/
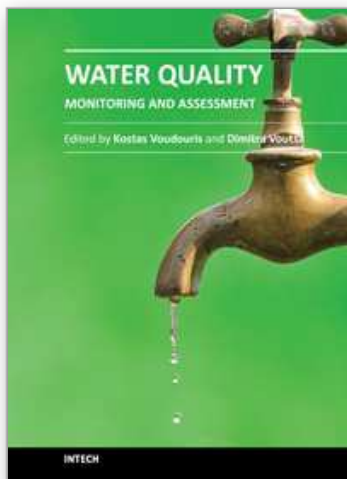
Water Quality Monitoring and Assessment

Edited by Dr. Voudouris

The book attempts to covers the main fields of water quality issues presenting case studies in various countries concerning the physicochemical characteristics of surface and groundwaters and possible pollution sources as well as methods and tools for the evaluation of water quality status. This book is divided into two sections: Statistical Analysis of Water Quality Data;Water Quality Monitoring Studies.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds