# Integrating Arc Hydro Features with a Schematic Network

(Accepted for publication in *Transactions in GIS*)

Timothy L. Whiteaker[1], David R. Maidment[2], Jonathan L Goodall [3] and Masatsugu

Takamatsu [4]


Center for Research in Water Resources
University of Texas at Austin
Austin, TX 78712

1. (Corresponding Author) Postdoctoral Fellow, Institute for Geophysics, P.O. Box 7456, MC R2200, University of Texas at Austin. Austin, TX 78712. Phone: (512) 471-0306, Email: twhit@mail.utexas.edu.

2. Professor of Civil Engineering, Center for Research in Water Resources, University of Texas at Austin.

3. Graduate Research Assistant, Center for Research in Water Resources, University of Texas at Austin.

4. Graduate Research Assistant, Center for Research in Water Resources, University of Texas at Austin.

## Abstract

A framework for integrating GIS features with processing engines to simulate hydrologic behavior is presented.  The framework is designed for compatibility with the ArcGIS ModelBuilder environment, and utilizes the data structure provided by the SchemaLink and SchemaNode feature classes from the ArcGIS Hydro data model.  SchemaLink and SchemaNode form the links and nodes, respectively, in a schematic network representing the connectivity between hydrologic features pertinent to the movement of surface water in the landscape.  A specific processing engine is associated with a given schematic feature, depending on the type of feature the schematic feature represents.  Processing engines allow features to behave as individual hydrologic processors in the landscape. The framework allows two types of processes for each feature, a Receive process and a Pass process.  Schematic network features operate with four types of values: received values, incremental values, total values, and passed values.  The framework assumes that the schematic network is dendritic, and that no backwater effects occur between schematic features.  A case study is presented for simulating bacterial loading in Galveston Bay in Texas from point and nonpoint sources.  A second case study is presented for simulating rainfall-runoff response and channel routing for the Llano River in Texas.

# 1 Introduction

The ArcGIS Hydro data model, or Arc Hydro, provides a framework for organizing and preprocessing geospatial and temporal data in a geographic information system (GIS) for use in hydrologic and hydraulic simulation models (Maidment 2002). One of the products created by Arc Hydro is a schematic network, which is a graphical representation of connectivity between hydrologic features in the landscape through a dendritic network of nodes and links. Each node represents a given feature, while each link connects two nodes, showing connectivity between the features represented by those nodes. While this connectivity is essential in modeling the movement of water and its constituents through the landscape, Arc Hydro does not provide a mechanism for describing **how** each feature in the schematic network handles the water it receives or determines the water it passes to other features in the network.

This paper describes a methodology for processing schematic networks using ArcGIS 9 geoprocessing technology. The methodology builds on the connectivity provided by the Arc Hydro schematic network, by allowing certain behaviors that define how features in the network process information received from and passed to other schematic features. A GIS tool called the Schematic Processor implements this methodology, and extends Arc Hydro from a preprocessing framework to a system for actually performing hydrologic simulations. Example applications of how the Schematic Processor is used to integrate features with processing engines to simulate hydrologic behavior are presented.

## 2 Previous work

A GIS is a useful tool in supplying simulation models with their spatial data inputs (Hellweger and Maidment 1997, Prisloe Jr. et al. 2000, Yoon 1996). Arc Hydro provides a structure for organizing hydrologic data, analyzing that data, and establishing relationships between hydrologic features. Whiteaker (2001) developed a tool for creating a schematic network from Arc Hydro features based on feature connectivity established through Arc Hydro attributes. ESRI subsequently improved the tool and now distributes it with the Arc Hydro data model.

Features in a schematic network may undergo different hydrologic processes, depending on the type of spatial entity those features represent. These processes can be described by hydrologic simulation models. Charnock et al. (1996) recommend simulating a given hydrologic process with an appropriate model designed for that process, and then chaining a sequence of models together to model the entire situation. Feng (2000) recommends that such components be assembled together in a common environment with rules governing the communication between components. Pullar (2003) develops such a framework for modeling fluxes and flows as continuous fields with Map Algebra in a GIS. Pullar and Springer (2000) cite flexibility as an additional advantage attained by creating individual hydro components that can be assembled to perform work.

Correia et al. (1998) observe that a GIS may be suitable for modeling processes that are not time-dependent, but recommends calling external simulation models when time series of data are used. Charnock et al. (1996) state that information transfer

between models and GIS can be inefficient, especially when the GIS is the storehouse for model information, due to the large size of GIS files.

This paper describes a framework for associating individual modeling components with features, for simulating hydrologic processes relevant to those features. Rather than operate from software outside of the GIS (e.g. Paniconi et al. 1999), the framework operates with Arc Hydro schematic features in the ArcGIS 9 ModelBuilder environment, resulting in a tighter linkage between the modeling components and GIS features. Following Feng's (2000) recommendations, the framework defines the rules governing communication between each feature. Rather than follow the process-oriented approach of Charnock et al. (1996) or the field –oriented approach of Pullar (2003), the framework takes a feature-oriented approach, with each feature executing its own hydrologic process as the flow of water passes from one feature to the next in the landscape. The suitability of the framework for modeling time-varying and time-invariant situations is explored through case studies simulating each type of scenario, with results confirming the findings of Correia et al. (1998) and Charnock et al. (1996) regarding time variant modeling and data storage in a GIS.

## 3 Arc Hydro and the Schematic Network

The Arc Hydro data model defines a structure for storing water resources features in a GIS, such that the attributes and relationships between those features facilitate the use of the GIS data in hydrologic and hydraulic simulation models. A key concept of Arc Hydro is the HydroID, a unique identifier for all hydrologic features in the landscape. Arc Hydro uses the HydroID to establish relationships defining the direction of flow of water between GIS features. For example, a watershed feature is related to the point

feature representing its outlet by storing the HydroID of the point feature as an attribute of the watershed feature (Maidment 2002).

Through the attributes and relationships of Arc Hydro, a schematic network may be created that describes the connectivity between geographic features (Figure 1). Schematic features differ from geographic features in that schematic features describe connectivity, while geographic features describe the true location of a physical entity.

Typically, Watersheds and HydroJunctions are used to create the schematic network. The JunctionID attribute on a Watershed stores the HydroID of the HydroJunction that serves as the outlet for that Watershed. The NextDownID attribute of a HydroJunction stores the HydroID of the next downstream HydroJunction. Thus, the HydroID is a key attribute allowing the linking of schematic features. With these attributes, connectivity is established between Watersheds and HydroJunctions. From this connectivity, a schematic network representing those features may be created.

The schematic network is composed of two feature classes: SchemaNode and SchemaLink, representing schematic nodes and schematic links, respectively. A schematic node represents a hydrologic feature, such as a watershed or a junction in the stream network. Schematic nodes are typically created from polygon or point features, with a node being located at the centroid of its related polygon feature or at the same location as its related point feature. The schematic nodes contain a field called FeatureID, which stores the HydroID of the source feature that is represented by the schematic feature.

Schematic links are straight lines that connect Schematic nodes. A schematic link cannot be connected to more than two schematic nodes, although a schematic node could

be connected to several schematic links.  In other words, a schematic link exists for every connection that a schematic node has with other schematic nodes.  Schematic links contain FromNodeID and ToNodeID, which identify the HydroID of the schematic nodes connected to a given schematic link.  FromNodeID identifies the from node, and ToNodeID identifies the to node.

Arc Hydro schematic features are further subdivided into two types.  Type 1 features are associated with a JunctionID-HydroID relationship from the source features, such as JunctionID on Watershed pointing to the HydroID of that Watershed's outlet HydroJunction.  Type 1 SchemaNodes are shown as squares in Figure 1.  Type 2 features are associated with a NextDownID-HydroID relationship from the source features, such as NextDownID on HydroJunction pointing to the HydroID of that HydroJunction's next downstream HydroJunction.  Type 2 SchemaNodes are shown as circles in Figure 1.

The types are specified in the SrcType attribute of SchemaNode, and the LinkType attribute of SchemaLink.  Additional node or link types may be added to denote additional feature categories.  For this research, a "Type 3" was created to connect the stream network nodes to underlying water bodies.

Once the schematic network has been created, attributes from the source features may be copied to the schematic features using the FeatureID-HydroID association between a schematic node and its related Arc Hydro feature.  These attributes are stored in fields, which are supplemental to the base schematic network attributes defined by Arc Hydro.  Additional attributes to further describe schematic features may also be added.

7

## 4 Methodology

The Schematic Processor is a tool which extends the functionality of the Arc Hydro schematic network to include behaviors that process attributes throughout the network. Each feature in a schematic network has a set of attributes that describe the state of that feature and what information it will pass to the next downstream feature in the network. Also associated with each feature is a set of behaviors that define how the feature responds to information from upstream features, and how the feature determines what information will be passed to the next downstream feature.

This methodology makes the following assumptions:

- The schematic network is dendritic. If branching occurs, information from the node at a branch will be passed to all downstream links at the branch.

- No backwater effects occur. The processing takes place from upstream to downstream features. Once an upstream feature has been processed, its values cannot be changed due to influences from downstream features.

These assumptions can be relaxed by implementing custom processing engines, which are described later in this paper.

### 4.1 Schematic Values

The movement of water (or contaminants in the water) across the landscape can be simulated by passing values down through the schematic network. Schematic network features incorporate four types of values: **received values**, **incremental values**, **total values**, and **passed values**.

**Received values** are those values received by a schematic feature from adjacent upstream schematic features. A node can only receive values from adjacent upstream

8

links, and a link can only receive a value from its adjacent upstream node. As an example, consider the simple schematic network shown in Figure 2. Node 4 represents the most downstream feature in the network. Suppose that bacterial loads are being transmitted through the stream network. Link 1 and Link 2 both flow into Node 3. Therefore, the received values for Node 3 consist of the bacterial loads discharging from Link 1 and Link 2. Once the loads are combined, Node 3 passes those flows to Link 3. Thus, the received value for Link 3 consists of the load from Node 3.

An **incremental value** is the value incorporated into the schematic network at a given schematic feature's location. Continuing the example above, suppose that Node 3 is a representative location for a plot of agricultural land. This land contributes a certain bacterial load into the stream network. Thus in addition to the loads from Link 1 and Link 2, this is an incremental loading at Node 3 from this local area.

The **total value** for a schematic feature is obtained by combining the incremental value and the received values. For example, the total load for Node 3 in the above example is the sum of the loads from Links 1 and 2, plus the incremental load from Node 3. A simple method to combine incremental and received values is by addition, although the total value could be any function of incremental and received values. (Eq. 1).

$$t = f(r,i) \hspace{4cm} \text{(Eq. 1)}$$

Where:

      $t$ = total value

      $r$ = received value

      $i$ = incremental value

A **passed value** is the value that a schematic feature passes to the next

downstream feature in the network.  Mathematically, a passed value could be any

function of the total value (Eq. 2).  In the case of Node 3 in the example above, the

passed value is the same as the total value.  However, as the bacteria travel along the

length of the stream connecting the junctions represented by Nodes 3 and 4, suppose that

the bacteria decay according to a first order decay rate.  The bacterial load at the most

downstream point of Link 3, after the decay rate has been applied, is the load that Link 3

passes to Node 4.  Thus, Link 3 receives a certain value from Node 3, but passes a

smaller value to Node 4 due to decay.

$$p = g(t) \hspace{5cm} \text{(Eq. 2)}$$

Where:

p = passed value

t = total value

### 4.2 Schematic Behaviors

When communicating with other schematic features, a given schematic feature possesses

certain behaviors that control how that feature will accept values from upstream features,

and how it will pass values to downstream features.  These behaviors are called **receiving**

**behavior** and **passing behavior**, respectively.  Each behavior uses the values associated

with a given schematic feature, along with other attributes and algorithms as necessary, to

determine the final value for a given attribute of that feature.  No distinction is made

among the types of schematic features as to which types of schematic behaviors they may

have.  In other words, both schematic links and schematic nodes may possess passing and

receiving behavior, although the nature of that behavior may vary from feature to feature (Figure 3).

**Receiving behavior** defines how a schematic feature processes incoming values along with its incremental value to produce a total value for that feature. The simplest example of receiving behavior is to calculate the sum of all incoming values and the incremental value for a given feature. Using the example above, if Link 1 and Link 2 pass bacterial loads of 7 and 8 lb/day, respectively, and Node 3 has an incremental value of 2 lb/day, then the total value for Node 3 is simply the sum of those values, or 17 lb/day.

For a more complex example of receiving behavior, consider two links that pass a time series of information (rather than a single value) to a node. The node must combine the two time series to produce a total time series. This process involves grouping together and processing values based on each time step, which may require temporal interpolation of one or both time periods.

By default, the Schematic Processor invokes a simple accumulation for a receive process, so that all values received by a feature are added to its incremental value to produce the total value for that feature. If more complex processing is required (such as decaying bacterial loads in a lake), a processing engine may be provided to simulate the receiving behavior.

**Passing behavior** defines how a schematic feature processes its total value to produce a value that will be passed to the next downstream schematic feature. The simplest example of passing behavior is to pass the total value for a given feature. More complex passing behaviors may also be modeled. Using the example above, the passing

11

behavior for Link 3 involves a first order decay process.  Thus, without any new bacterial load inputs, the load passed to Node 4 from Link 3 will be less than the load received by Link 3, due to decay while the flow passes through the link.  Note that the Schematic Processor does not define where the passing behavior takes place on the link (e.g. at the beginning, middle, or end, or along the entire length of the link).  It merely defines how the total value for the link is changed to produce a value that is passed to its downstream node.

By default, the Schematic Processor simply passes the total value for a given feature to the next downstream feature.  If more complex processing is required, a processing engine may be provided to simulate the passing behavior.

**4.3 Processing Order**

This methodology applies to schematic networks in which no backwater effects occur.  In other words, each schematic feature receives no information from or about downstream schematic features.  However, since each feature does receive information from upstream features, all upstream features should be processed (so that they know what value they should pass) before processing a given schematic feature.  The Schematic Processor uses the Hydro, FromNodeID and ToNodeID attributes on schematic features to arrange those features in a processing order from upstream to downstream.

# 5 Procedure of Application

This section describes the procedure used to implement the methodology described above within the ArcGIS 9 geoprocessing environment.  This procedure includes a data model to support schematic navigation and attribute storage, an algorithm for sorting the

schematic features from upstream to downstream, and the actual implementation of the procedure with an ArcGIS geoprocessing tool called the Schematic Processor.

**5.1 Data Model**

The Schematic Processor is designed to work with ArcGIS software. ArcGIS stores geospatial data in a geodatabase, which is a relational database containing both spatial and attribute data. Feature classes are tables in the database for which each record contains a shape which may be displayed in the GIS. Object classes are tables for which each record is not associated with a specific shape to be displayed in the GIS.

The Schematic Processor operates on two geodatabase object classes: a schematic node class and a schematic link class. These classes may be feature classes or object classes, as the geometry of the nodes and links is not explicitly taken into consideration by the Schematic Processor. Rather, the tool uses attributes within the node and link classes to establish connectivity between them. Thus, any two feature classes or tables may be used with the Schematic Processor, as long as they are located in the same geodatabase.

Because the Schematic Processor was designed to operate on an Arc Hydro schematic network, the tool assumes that the Arc Hydro schematic attributes exist in the schematic node and schematic link classes. These attributes are used for identifying schematic features and navigating the schematic network. For Schematic Nodes, the relevant attributes include:

- HydroID - Long integer automatically defined providing a unique identifier for the feature in the geodatabase

13

- SrcType - Short integer, user-defined, which indicates the type of the schematic feature, populated by Node/Link Schema Generation tool in the Arc Hydro tools

For Schematic Links, these attributes include:

- HydroID - Unique identifier for the feature in the geodatabase

- LinkType - Short integer, user-defined, which indicates the type of the schematic feature, populated by Node/Link Schema Generation tool in the Arc Hydro tools

- FromNodeID - The HydroID of the schematic node located at the upstream end of the schematic link

- ToNodeID - The HydroID of the schematic node located at the downstream end of the schematic link

In addition to the Arc Hydro attributes, schematic features may also possess attributes that describe the values that those features operate with.  For both the schematic node and link classes, these attributes include:

- Incremental Value - Stores the incremental value for the given feature

- Total Value - Stores the total value for the given feature

- Passed Value - Stores the value that the feature will pass to the next downstream feature in the network

Depending on what type of process is specified for each feature, other attributes may be required.  For example, when decaying loads along a schematic link, travel time and decay constants may be required.  In these cases, the attributes may be located on the features to which they pertain.  These attributes are not read directly by the Schematic

Processor, but by the specific processing engine designed to work with those attributes. Processing engines are described in later in this chapter.

**5.2 Processing Procedure**

This section describes the procedure used by the Schematic Processor for working with schematic networks.  The procedure may be summarized into two major components: **Data Preparation** and the **Process Loop**.

In **Data Preparation**, the features in the schematic network are sorted from upstream to downstream, and Collections are initialized that will store values during the processing of the network.  A Collection is a standard Visual Basic object which stores a set of values or pointers to other objects.  For each schematic feature, a Topology Collection stores a list of HydroIDs of upstream features, indexed by the HydroID of the current feature.  A Value Collection stores the schematic values that each feature will pass to the next downstream feature, indexed by the HydroID of the current feature. Using Collections improves the performance of the Schematic Processor.

Once the features have been sorted, each feature is processed in the correct order until all features have been processed.  This is called the **Process Loop**.  There are five steps in each interation of the process loop (Figure 4):

1.      Get upstream features

2.      Get upstream values

3.      Process upstream and incremental values

4.      Process current value to pass downstream

5.      Update Value Collection with value to pass

Thus, for a given feature, adjacent upstream features are located. The values from these upstream features are combined with the incremental value for the current feature using Receiving behavior to produce a total value for the current feature. This value is then modified by Passing behavior to produce a value to pass to the next downstream feature. This value is then updated in the Value Collection. The Process Loop then moves to the next downstream feature in the schematic network and repeats the five steps for that feature.

## 5.3 Implementation in ArcGIS 9

ArcGIS 9 is the latest GIS software from ESRI. It features a geoprocessing framework called ModelBuilder that allows users to chain together a series of geoprocessing tasks and data sources in a workflow model. In addition to standard ArcGIS tools, the framework allows users to create custom Script Tools, which may also be inserted into a workflow model. Each Script Tool is attached to an underlying script, written in a scripting language such as VBscript or Python, which performs the actual work. The Schematic Processor is implemented in this framework as a script tool called ProcessSchematic. The associated script is called ProcessSchematic.vbs. This script calls a Visual Basic DLL called MBSchematic.dll, which sorts the schematic features and handles the calling of process DLLs. Thus, ProcessSchematic.vbs is simply used as an intermediary between ArcGIS and MBSchematic.dll. A DLL (Dynamic Linked Library) contains a library of classes and subroutines which may be used by another application. A DLL was used rather than a standalone script because a DLL typically allows for more robust programming.

16

In addition to MBSchematic.dll, zero, one, or more DLLs may be called as processing engines to simulate custom behavior associated with schematic features. The ProcessSchematic Script Tool instructs MBSchematic.dll which processing engines are associated with each type of schematic feature. If no processing engine is specified for a given type of schematic feature, the Process Schematic tool uses the defaults for receiving and passing behavior as described above. This gives the user the ability to create as many processing engines as desired, or to use the default behaviors if no specialized behaviors are required (Figure 5).

## 6 Case Studies

Two case studies are presented, which use the Schematic Processor to model hydrologic behavior with features in a GIS. The first simulates rainfall-runoff response and channel routing using routines from the Hydrologic Engineering Center 's library of hydrologic functions, LibHydro. The second simulates bacterial loading in Galveston Bay from point and nonpoint sources.

### 6.1 Case Study I: LibHydro Application

The Hydrologic Engineering Center (HEC) is an organization within the US Army Corps of Engineers which provides technical expertise, models, methods, research, and training in the areas of hydrology, hydraulics, and water resources. HEC-HMS (Hydrologic Modeling System) is a publicly available standalone rainfall-runoff simulation model provided by the HEC designed for dendritic watershed systems. HEC-HMS has been utilized in many applications throughout the years, and is now considered one of the standard hydrologic simulation models to use in the United States. The HEC recognized that certain functions in its models could be extracted to provide useful routines

17

applicable in many situations outside of the HEC modeling environment.  This led to the

development of a library of functions called LibHydro.  These functions were derived

from the HEC-1 software, which is the predecessor to HEC-HMS.  The routines are

coded in FORTRAN, and can be accessed by calling the LibHydro DLL using a variety

of programming languages, including Visual Basic and C++.  Examples of routines

include unit conversions and Muskingum routing (Hydrologic Engineering Center, 1995).

By referencing LibHydro, a hydrologic application can make use of effective modeling

routines that have already been proven through years of use.

In this example, functions from LibHydro are used to simulate rainfall-runoff and

routing calculations for a portion of the Llano River basin in Texas (Figure 6).  The raw

data for this application includes a Watershed feature class representing the target

subbasins for the analysis, precipitation data stored in the Arc Hydro TimeSeries table,

and a HydroNetwork composed of HydroEdges and HydroJunctions.  The precipitation

data covered a period from November 2nd through November 8th, 2000, at 15-minute

intervals.  A schematic network was built using the Arc Hydro tools (Figure 7).  The

schematic network includes nodes representing Watersheds (Type 1 nodes) and

HydroJunctions (Type 2 nodes), and links connecting those nodes.  Type 1 links connect

Type 1 nodes to Type 2 nodes, while Type 2 links connect Type 2 nodes to Type 2 nodes.

Four LibHydro functions were used in this analysis: Initial Constant Loss, Snyder

Unit Hydrograph, Baseflow, and Modified Puls Routing.  The first three functions were

used to prepare time series of watershed outflows to the stream network, while the last

function was used by the schematic network to route the flows to the outlet of the

subbasin network.  These functions were incorporated into an ArcGIS 9 workflow model

called Rainfall to Routed Flow, and executed in the ModelBuilder environment.  More information about the specifics of each LibHydro function can be found in Takamatsu (2003).

The Rainfall to Routed Flow workflow model consists of four script tools, one associated with each of the LibHydro functions listed above.  The first three tools prepare time series information to be used by the Schematic Processor for routing the flows.  The last tool is the Schematic Processor.  Each tool accesses a DLL called MBlibHydro in order to use the functions in LibHydro.

The first tool, Loss Initial Constant, calls the Loss Initial Constant function from LibHydro to calculate precipitation excess for each Watershed using rainfall data from the TimeSeries table.  Fields in the Watershed feature class provide the values of initial loss, constant loss rate, and impervious area ratio for each Watershed.  The second tool, UnitgraphSnyder, calls the Snyder Unit Hydrograph function from LibHydro to calculate a runoff hydrograph for each watershed, given the precipitation excess time series determined by the Loss Initial Constant Tool.  Fields in the Watershed feature class provide the values of Snyder Cp, Snyder Tp, and Basin Area for each Watershed.  The third tool, Baseflow, calls the Baseflow function from LibHydro to calculate Baseflow for each watershed, given the precipitation excess time series determined by the Loss Initial Constant Tool.  The tool then adds the baseflow to the runoff hydrograph to produce an outflow time series for each watershed.  Fields in the Watershed feature class provide the values of recession ratio and recession threshold for each Watershed.

The final tool in the workflow model is the Schematic Processor.  Once outflow time series have been calculated for each watershed, this tool uses the schematic network

to route the flows through the stream network to the outlet of the target subbasins. This particular application of the Schematic Processor uses the tool in a unique way. The tool is designed to pass single values from feature to feature. However, the Rainfall to Routed Flow model requires time series of flows to be passed between features, rather than a single value. Therefore, instead of passing single values between features, the Schematic Processor is set up to pass time series type IDs, or TSTypeIDs, which categorize values in the Arc Hydro TimeSeries table (such as rainfall, runoff, etc.). SchemaLinkPuls.DLL and SchematicNode.DLL are set as processing engines, whose purpose is to extract time series data for each feature based on the TSTypeID, route the time series, and write the resulting time series of routed flows back to the TimeSeries table. SchemaLinkPuls.DLL uses the Modified Puls Routing function from LibHydro to route flows along a river reach, represented by a schematic link of Type 2. When two reaches converge, their time series of routed flows are added together to produce a combined flow. SchematicNode.DLL performs that task, with the assumption that there are no backwater effects (Table 1).

Once the Rainfall to Routed Flow model has completed execution, a time series of outflow values for the basin comprised of the target Llano subbasins is recorded in the Arc Hydro TimeSeries table.

The Rainfall to Routed Flow model was run for the Llano subbasins covering precipitation events from November 2nd through the 8th in the year 2000. An identical model was setup and run within HEC-HMS for comparison. The results from both models matched almost exactly (Figure 8), with differences of less than one-half of one percent in most cases (Table 2). Rounding within each model may account for the

differences in results between the two models.  Still, with nearly identical results, this experiment verifies that the Rainfall to Routed Flow model is successful in calling the same functions used by HMS.

Another important observation from the execution of these two models is that HMS performed the simulation much quicker than the Rainfall to Routed Flow model.  A complete run of Rainfall to Routed Flow within ArcGIS took approximately 20 minutes to finish.  The HMS model took only 20 seconds to run.  From these results, it is clear that improvements in efficiency of handling time series in GIS are needed in order for time varying modeling within a GIS to be practical.  For example, time series records from HEC-HMS are processed in memory or stored in a special binary format designed for handling time series data, and so manipulation of time series is very fast.  However, in the Rainfall to Routed Flow model, time series data are stored as records in a database, which results in very costly transactions in terms of processing time.

**6.2 Case Study II: Water Quality Modeling in Galveston Bay**

Whiteaker and Goodall (2003) applied the Schematic Processor to water quality modeling in Galveston Bay, Texas, by examining fecal coliform loading from point and nonpoint sources into the bay (Figure 9).  The procedure and equations used to model bacterial loadings were adapted from Zoun (2003), who used the Arc Hydro tools and conventional GIS techniques to determine loadings.  Three processing engines were developed for the Schematic Processor to simulate bacterial load transport, while an ArcGIS workflow model, WQModel, was assembled to perform the analysis.  WQModel includes components to determine watershed loads, as well as the Schematic Processor to accumulate loads into the bay.  WQModel assumes that general data development, such

as watershed delineation and the calculation of travel times on reaches, has already been accomplished and verified. The schematic network must also have already been created with three types of nodes and links to represent the hydrologic features used in the water quality analysis (Table 3).

WQModel uses rainfall data and watershed characteristics such as land use to determine annual watershed loads. The Schematic Processor is then called to move the loads from the watersheds into the bay, decaying the loads as they travel through the stream network. The model contains fifteen tools, including standard ArcGIS tools as well as custom script tools that access DLLs. The reader is referred to Whiteaker and Goodall (2003) for a detailed explanation of each tool in the model.

The three types of links and three types of nodes signify the behavior that features in the schematic network will implement. When behavior more complex than simple accumulation is required, the Schematic Processor calls functions from a WaterQualityProcessors DLL that was written to support water quality modeling. The processes associated with each type of link and node are shown in Tables 2 and 3. The objects in the 'dll' column in the tables are classes that exist in the WaterQualityProcessors DLL., which were constructed to function as processing engines. The classes are associated with the appropriate link or node types when the user specifies inputs to the Schematic Processor.

Three processing engines were constructed for this case study. The first engine, *Decay*, used a first order decay rate and received loads to simulate the decay of bacteria along stream segments. The decay rate represents the loss of mass due to biological decay, sorption, uptake, etc, as material moves downstream. Decay rates for each stream

22

segment were determined by Zoun (2003) and stored as attributes on the SchemaLink feature representing that segment.  The *Decay* processing engine is called when a SchemaLink is passing a load to its next downstream node.  The load passed is reduced according to a first-order reaction shown in Equation 3.

$$load_{passed} = load_{receieved} \times e^{-kt}$$  (Eq. 3)

Where:

loadpassed = downstream bacteria load (cfu/yr)

loadreceived = upstream bacteria load (cfu/yr)

k = first-order decay coefficient (day-1)

t = travel time (day)

The second processing engine, *CFSTR*, calculates a bay's increase in concentration associated due to an import of bacteria mass.  Each bay's volume, estimated yearly flow, and first-order decay term are stored in the SchemaNode feature class.  The load received by the SchemaNode representing the bay is used to estimate the bay concentration, assuming the bay has constant inflow equal to its outflow, and that mass entering the bay is instantaneous and perfectly mixed within the bay (Eq. 4).   These are commonly known as continuous flow, stirred tank rector (CFSTR or CSTR) assumptions.

$$c = \frac{L}{Q + kV}$$  (Eq. 4)

Where:

c = concentration within bay (coliform units/m$^3$)

L = bacteria load entering bay (coliform units/yr)

23

$Q$ = total flow (m$^3$/yr)

$k$ = first-order decay term (day$^{-1}$)

$V$ = volume of bay (m$^3$)

The third processing engine, *LossCoef*, accounts for bacteria from the watersheds upstream of East Bay that flows into Sabine Lake and not East Bay. From Zoun (2003), half of the load from these watersheds is assumed to flow into East Bay, while the other half is delivered to Sabine Lake. The *LossCoef* processing engine uses the LossCoef attribute of a SchemaNode feature to determine the fraction of the load from that node that is passed to its downstream link. This allows one to easily adjust the proportioning of load between East Bay and Sabine Lake. The LossCoef value for the appropriate nodes in the Galveston Bay database were assigned a value of 0.5.

Once the data were prepared and the processing engines defined, the WQModel was run. The results of WQModel execution match up quite well with Zoun's results. However, some discrepancies as high as 2.6% were present. These discrepancies were due to differences in the runoff grid used by Zoun and the one used during ModelBuilder execution. The cells in the runoff from Zoun's work did not exactly overlap the cells in the precipitation grid. However, because the runoff grid used by WQModel in ModelBuilder was generated directly from the input precipitation grid, the cells overlap exactly. Because the cell size for the runoff grid was so coarse, a slight shift in the cell locations resulted in enough variation in load calculations to produce different concentrations in the bay. However, even with this discrepancy, the loads matched up very well, especially given the inexact nature of nonpoint load calculations for this watershed.

WQModel was constructed in 3 weeks, and executed in 20 minutes for Galveston Bay on an average desktop computer. The ease of execution greatly improves the efficiency of the water quality modeling algorithms developed by Zoun (2003), as those algorithms need no longer be carried out manually. The algorithms may also be repeated for different applications by supplying different inputs datasets or parameters to the model.

## 7 Conclusions

One limitation of ModelBuilder is that the standard ArcGIS tools are designed to operate at the dataset level, rather than the feature level. However, feature level operations may be implemented through custom tools, such as the Schematic Network Processor, which may operate in the ModelBuilder environment but utilize DLLs (or other programming constructs) to analyze data one feature at a time.

The Schematic Processor allows links and nodes (which represent hydrologic features) in a schematic network to control how information is received from upstream features, and what information is passed to downstream features, by associating each type of feature with a specific processing engine which simulates hydrologic behavior. The modular design of the Schematic Processor gives the user the ability to develop processing engines geared towards solving a wide variety of water resources problems. However, the Schematic Processor assumes that the schematic network is dendritic, and that no backwater effects occur between schematic features. Violating these assumptions could hamper the performance of the Schematic Processor, or produce incorrect results.The Rainfall to Routed Flow model proves that hydrologic and hydraulic calculations comparable to those possible with simulation models such as HMS can be

carried out within a GIS application by accessing functions from LibHydro.  The work could be extended to include functions from other object libraries.  The modular nature of ModelBuilder tools makes the components in Rainfall to Routed Flow extensible and reusable.

One important finding from the LibHydro application was that the creation and manipulation of large amounts of time series data is very costly in time when writing to an ArcGIS geodatabase.  For time-invariant analysis, such as with the water quality modeling in Galveston Bay example, the computation time was very fast.  However, the LibHydro application executed much slower than an identical model setup in the HMS program.  For these types of applications, the best approach may be to write only the final results of model simulations back to the Arc Hydro TimeSeries table.  The intermediate time series can be calculated through LibHydro and managed in a more efficient medium, or an independent simulation model (e.g. HMS) can be called from ModelBuilder to perform the simulation, returning control to the GIS when the simulation is completed.

Using the Schematic Processor and the ModelBuilder environment provides a good deal of flexibility to the user.  In addition to bacterial loads, WQModel may be applied to other contaminants simply by applying the model to different attributes on the schematic network.  The workflow model could also be applied to different regions, by supplying the model with different input datasets. The modular nature of the Schematic Processor also permits the addition of other processing engines, to appropriately model the behavior of hydrologic features in the region of interest.

The integration of GIS features with processing engines simulating hydrologic behavior provide water resources engineers with valuable analysis tools that take full

advantage of Arc Hydro and the ArcGIS 9 geoprocessing framework. Arc Hydro

changes the role of GIS features from blue lines on a map to data constructs that correlate

to specific features in a hydrologic simulation model. The Schematic Processor goes one

step further, by providing a mechanism to attach simulation models or components of

those models directly with the Arc Hydro features themselves.

## Acknowledgements

# References

Charnock T W, Hedges P D, and Elgy J 1996 Linking Multiple Process Level Models
with GIS. In *Application of Geographic Information Systems in Hydrology and
Water Resources Management*. Netherlands, Krips Repro, Meppel: 29-36

Correia F N, Rego F C, Saraiva M G, and Ramos I 1998 Coupling GIS with Hydrologic
and Hydraulic Flood Modelling. *Water Resources Management* 12: 229-49

Feng C 2000 Open Hydrologic Model for Facilitating GIS and Hydrologic Model
Interoperability. In *Proceedings of the 4th International Conference on
Integrating GIS and Environmental Modeling*. WWW document,
http://www.colorado.edu/Research/cires/banff/pubpapers/6

Hydrologic Engineering Center 1995 *LibHydro Users Manual*. Davis, CA, Hydraulic
Engineering Center

Hellweger F and Maidment D R 1997 *HEC-PREPRO: A GIS Preprocessor for Lumped
Parameter Hydrologic Modeling Programs*. Thesis. The University of Texas at
Austin. WWW document, http://www.crwr.utexas.edu/reports/199
8.shtml

Maidment, D R 2002 *Arc Hydro: GIS for Water Resources*. Redlands, CA, ESRI Press

Paniconi C, Kleinfeldt S, Deckmyn J, and Giacomelli A 1999 Integrating GIS and Data
Visualization Tools for Distributed Hydrologic Modeling. *Transactions in GIS*
3(2): 97-118

Prisloe Jr. M P, Arnold Jr. C L, Civco D L, and Lei Y 2000 A Simple GIS-based Model
to Characterize Water Quality in Connecticut Watersheds. In *Proceedings of the*

*4th International Conference on Integrating GIS and Environmental Modeling*.
WWW document, http://i7777777o636f6c6f7261646fo6564752f...hanos/pub.paper.Research/cires

Pullar D 2003 Simulation Modelling Applied to Runoff Modelling Using MapScript.
*Transactions in GIS* 7(2): 267-283

Pullar D and Springer D 2000 Towards Integrating GIS and Catchment Models.
*Environmental Modelling & Software* 15: 451-9

Takamatsu M 2003 LibHydro.  WWW document,
https://webspace.utexas.edu/mtaka/www/lh_home.htm

Whiteaker T 2001 *A Prototype Toolset for the ArcGIS Hydro Data Model*.  Thesis. The
University of Texas at Austin.  WWW document,
http://www.crwr.utexas.edu/reports/2001/rpt01-6.shtml

Whiteaker T and Goodall J 2003 Water Quality Modeling in Galveston Bay with Model
Builder.  In *Proceedings of ESRI International User Conference*.  San Diego, CA,
ESRI Press

Yoon J 1996 Watershed-Scale Nonpoint Source Pollution Modeling and Decision
Support System Based on a Model-GIS-RDBMS Linkage.  In *Proceedings of the
AWRA Symposium on GIS and Water Resources*.  WWW document,
http://www.awwa.org/proceedings/gis32/jyoon

Zoun R J 2003 *Estimation of Fecal Coliform Loadings to Galveston Bay*.  Thesis.  The
University of Texas at Austin.  WWW document,
http://i777777o63727772o75746578617o656475z.oszar.com/reports/2003/rpt03-5.shtml

# Tables

| Links | | | |
|---|---|---|---|
| **Type** | **Purpose** | **Process** | **DLL** |
| 1 | land to river transport | TSTypeID of outflow time series passed from watershed to outlet | none |
| 2 | along river routing | Flow is routed along a stream segment using the Modified Puls method from libHydro | SchemaLinkPuls |
| | | | |
| **Nodes** | | | |
| **Type** | **Purpose** | **Process** | **DLL** |
| 1 | represent watershed | Store TSTypeiD of outflow time series | none |
| 2 | combine flows | Sum time series of flows at a junction along the stream network | SchematicNode |

**Table 1.** Explanation of processing engines for schema links and nodes in Rainfall to Routed Flow model

| Timestamp | Flow (cms) from HMS | Flow (cms) from Rainfall to Routed Flow Model | Percent Difference (%) |
|---|---|---|---|
| 11/3/00 6:45 PM | 577.980 | 577.785 | 0.034 |
| 11/3/00 7:00 PM | 560.730 | 560.627 | 0.018 |
| 11/3/00 7:15 PM | 542.840 | 542.861 | 0.004 |
| 11/3/00 7:30 PM | 523.750 | 523.928 | 0.034 |

**Table 2.** Values Computed by HMS and Rainfall to Routed Flow Model Varied by Fractions of a Percent

| Links | | | |
|---|---|---|---|
| **Type** | **Purpose** | **Process** | **Class in DLL** |
| 1 | land to river mass transport | Mass is passed to downstream node without decay because incremental value represents mass of bacteria which reaches watershed outlet. | none |
| 2 | along river transport | Mass passed to downstream node is decayed according to time of travel and decay coefficient. | clsDecay |
| 3 | river to bay | Mass is passed from river outlet to bay without decay. | none |
| | | | |
| **Nodes** | | | |
| **Type** | **Purpose** | **Process** | **Class in DLL** |
| 1 | losses or additions of mass for watersheds | Mass is passed to downstream link without decay. | none |
| 2 | losses or additions of mass along river network | Mass is passed to downstream link without decay except for one feature for which the LossCoef field is nonzero.  The 0.5 factor is to account for bacteria that enters Sabine Lake from the upstream Watersheds for East Bay. | clsLossCoef |
| 3 | losses or additions of mass for bays | Mass is received by the bay and used to estimate concentration (cfu/m3) within the bay.  Estimated concentration in bay is based on continuous flow stired tank reactor (CFSTR) assumptions. | clsCFSTR |

**Table 3.** Explanation of processing engines for schema links and nodes in WQModel (Whiteaker and Goodall, 2003)

**Figures**



**Figure 1.** The schematic network describes connectivity between features in the landscape. Type 1 features connect watersheds to the stream network. Type 2 features connect junctions in the stream network.
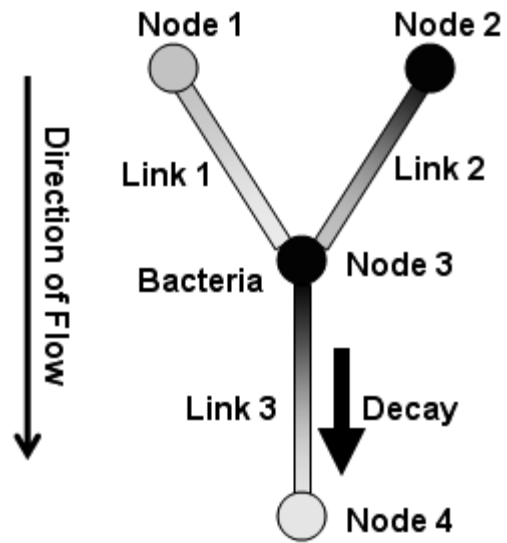
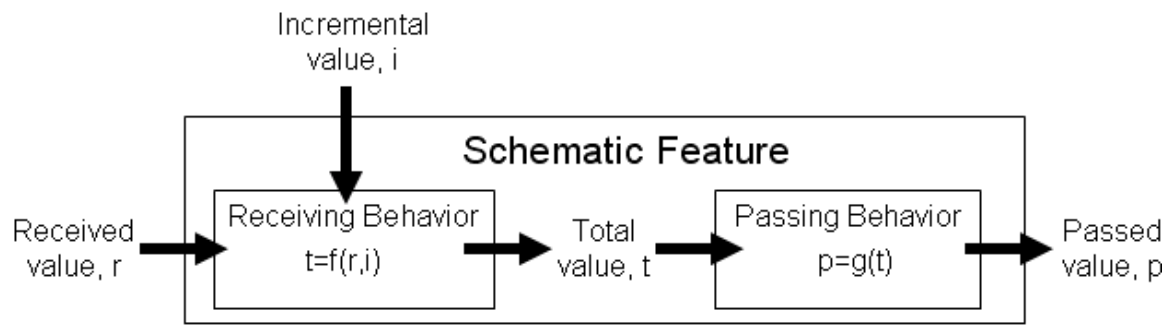**Figure 2.** Example schematic network with bacterial load

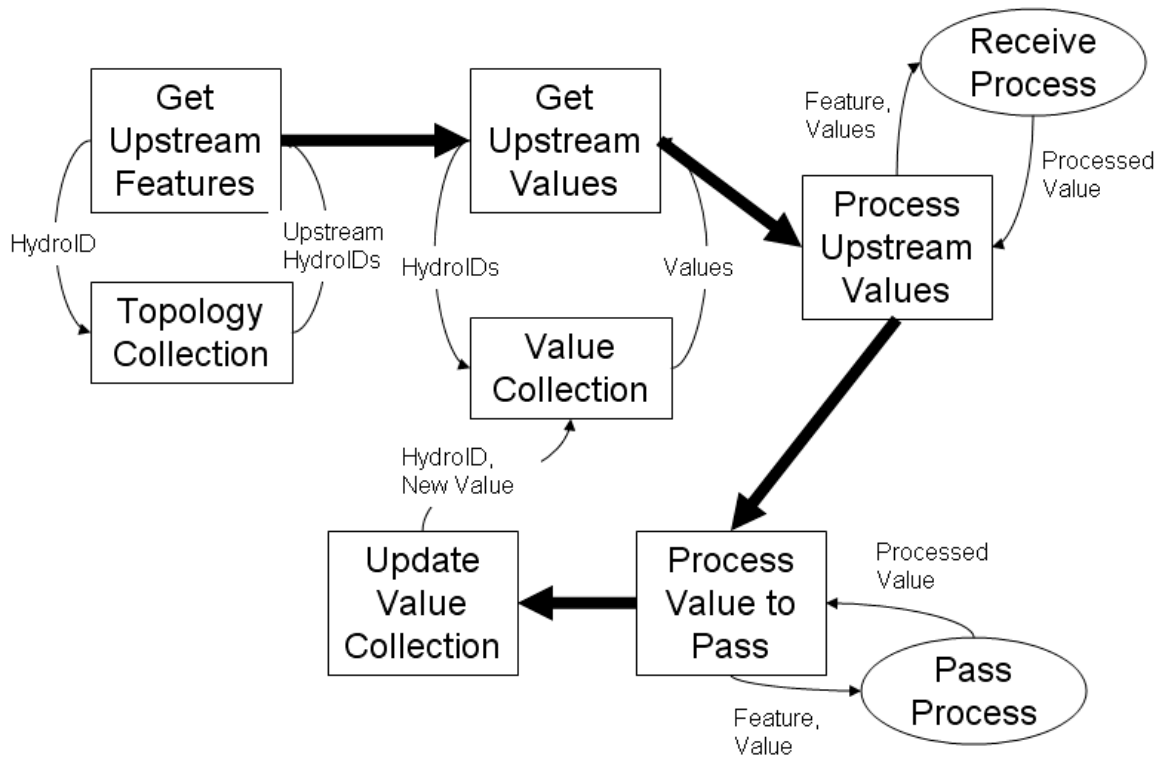**Figure 3.** Behaviors for schematic features

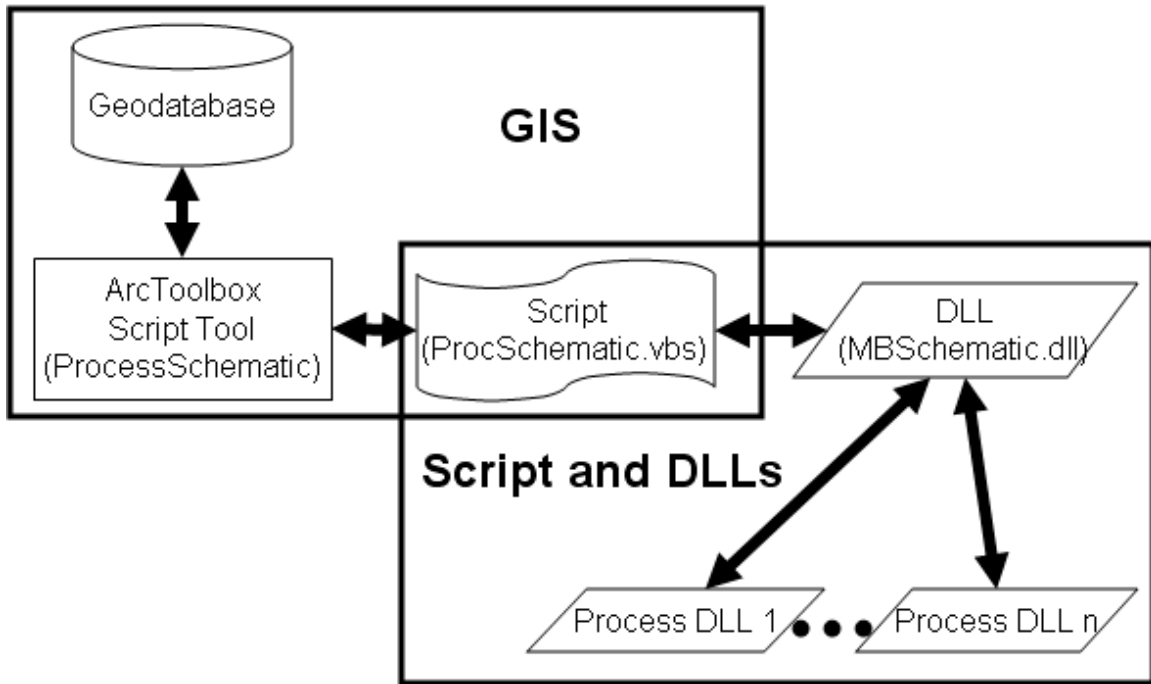**Figure 4.** Summary of single iteration in the process loop

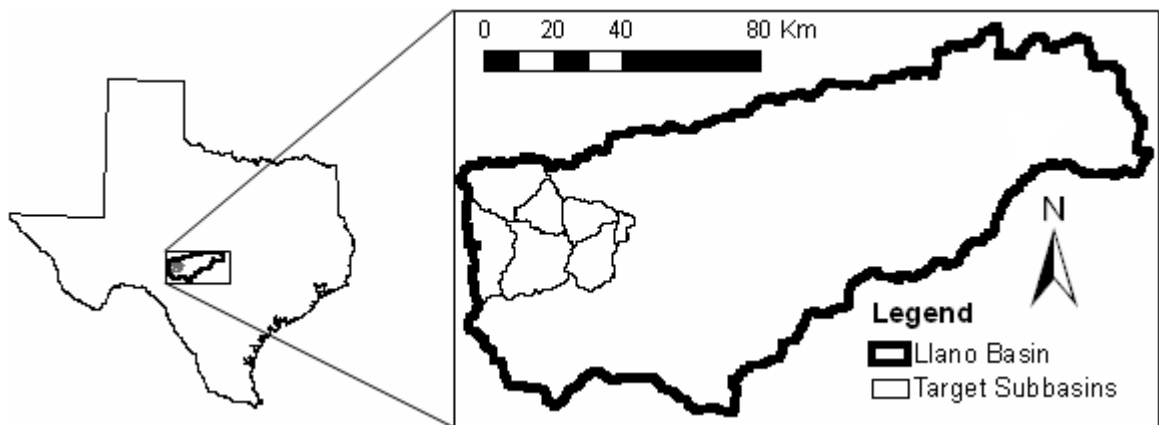**Figure 5.** Implementation of methodology using scripts and DLLs

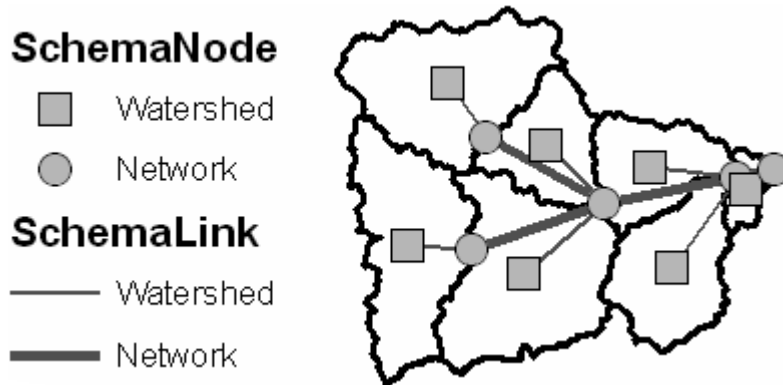**Figure 6.** Target subbasins in Llano River Basin
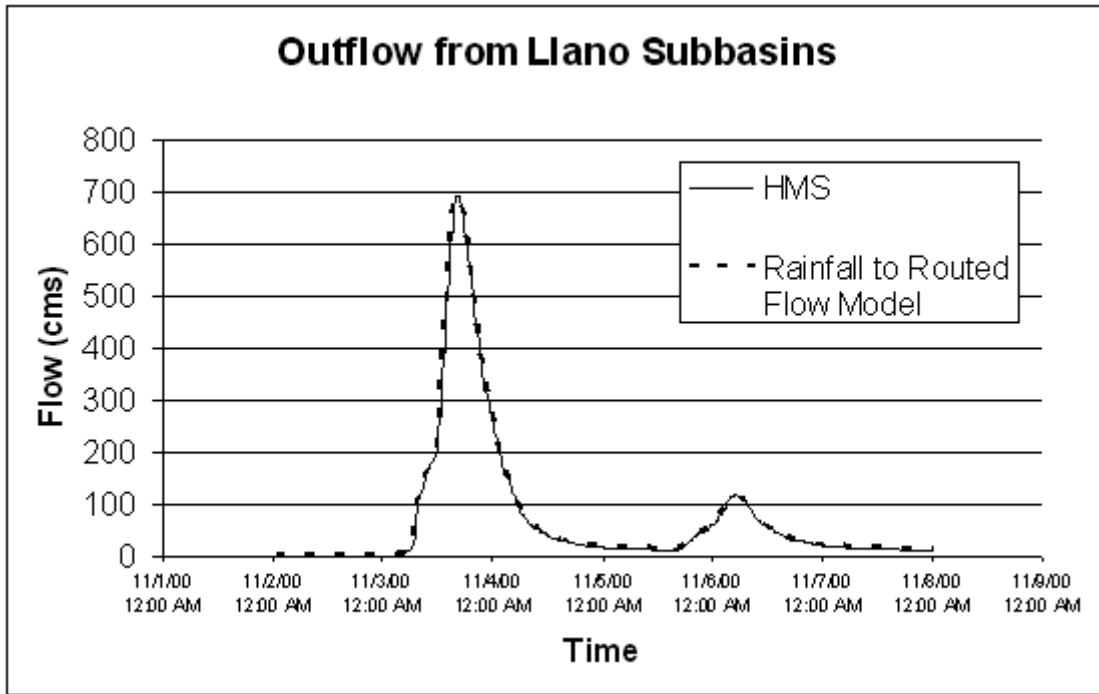
**Figure 7.** Schematic network for Llano target subbasins

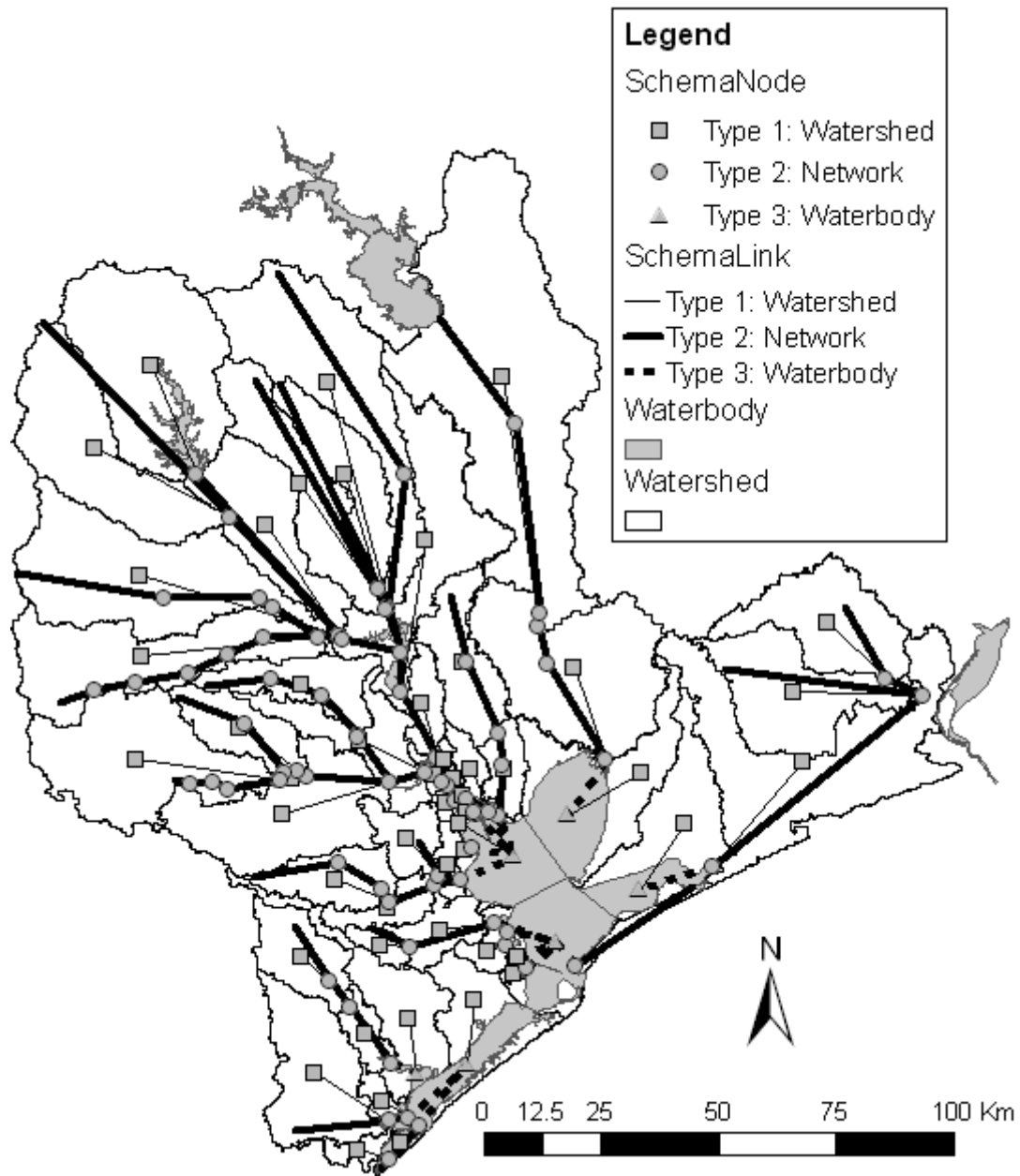**Figure 8.** Outflow hydrographs computed by HMS and Rainfall to Routed Flow Model

**Figure 9.** Gavleston Bay watershed with schematic network

## Figure Captions

**Figure 1.** The schematic network describes connectivity between features in the landscape.  Type 1 features connect watersheds to the stream network.  Type 2 features connect junctions in the stream network.

**Figure 2.** Example schematic network with bacterial load

**Figure 3.** Behaviors for schematic features

**Figure 4.** Summary of single iteration in the process loop

**Figure 5.** Implementation of methodology using scripts and DLLs

**Figure 6.** Target subbasins in Llano River Basin

**Figure 7.** Schematic network for Llano target subbasins

**Figure 8.** Outflow hydrographs computed by HMS and Rainfall to Routed Flow Model

**Figure 9.** Gavleston Bay watershed with schematic network