

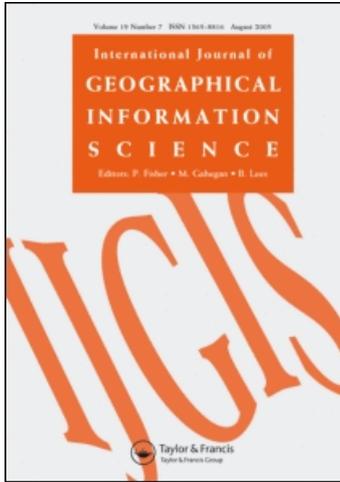
This article was downloaded by: [Duke University]

On: 8 February 2011

Access details: Access Details: [subscription number 930879418]

Publisher Taylor & Francis

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



International Journal of Geographical Information Science

Publication details, including instructions for authors and subscription information:

<http://www.informaworld.com/smpp/title~content=t713599799>

An object-oriented shared data model for GIS and distributed hydrologic models

Mukesh Kumar^a; Gopal Bhatt^a; Christopher J. Duffy^a

^a Department of Civil and Environmental Engineering 212 Sackett Bldg, Penn State University, University Park, PA, USA

Online publication date: 10 June 2010

To cite this Article Kumar, Mukesh , Bhatt, Gopal and Duffy, Christopher J.(2010) 'An object-oriented shared data model for GIS and distributed hydrologic models', International Journal of Geographical Information Science, 24: 7, 1061 – 1079

To link to this Article: DOI: 10.1080/13658810903289460

URL: <http://dx.doi.org/10.1080/13658810903289460>

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: <http://www.informaworld.com/terms-and-conditions-of-access.pdf>

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

An object-oriented shared data model for GIS and distributed hydrologic models

Mukesh Kumar, Gopal Bhatt and Christopher J. Duffy*

Department of Civil and Environmental Engineering, 212 Sackett Bldg, Penn State University,
University Park, PA 16802, USA

(Received 24 May 2008; final version received 26 August 2009)

Distributed physical models for the space–time distribution of water, energy, vegetation, and mass flow require new strategies for data representation, model domain decomposition, *a priori* parameterization, and visualization. The geographic information system (GIS) has been traditionally used to accomplish these data management functionalities in hydrologic applications. However, the interaction between the data management tools and the physical model are often loosely integrated and nondynamic. This is because (a) the data types, semantics, resolutions, and formats for the physical model system and the distributed data or parameters may be different, with significant data preprocessing required before they can be shared; (b) the management tools may not be accessible or shared by the GIS and physical model; and (c) the individual systems may be operating system dependent or are driven by proprietary data structures. The impediment to seamless data flow between the two software components has the effect of increasing the model setup time and analysis time of model output results, and also makes it restrictive to perform sophisticated numerical modeling procedures (real-time forecasting, sensitivity analysis, etc.) that utilize extensive GIS data. These limitations can be offset to a large degree by developing an integrated software component that shares data between the (hydrologic) model and the GIS modules. We contend that the prerequisite for the development of such an integrated software component is a ‘shared data model’, which is designed using an object-oriented strategy. Here we present the design of such a shared data model taking into consideration the data type descriptions, identification of data classes, relationships, and constraints. The developed data model has been used as a method base for developing a coupled GIS interface to Penn State Integrated Hydrologic Model (PIHM), called PIHMgis.

Keywords: PIHM; PIHMgis; model GIS coupling; UML; hydrologic modeling

1. Introduction

Physics-based distributed hydrologic models (DHMs) simulate hydrologic state variables in space and time while using information regarding heterogeneity in climate, land use, topography, and hydrogeology (Freeze and Harlan 1969, Kollet and Maxwell 2006). Because of the large number of physical parameters incorporated in such a model, intensive data development and assignment are needed for accurate and efficient model simulations. A geographic information system (GIS) has the ability to handle both spatial and nonspatial data, and perform data management and analysis. However, it lacks sophisticated analytical

*Corresponding author. Email: cxd11@psu.edu

and modeling capabilities (Maidment 1993, Abel and Kilby 1994, Kopp 1996, Wilson *et al.* 1996). On the other hand, the physical models generally lack the data organization and development functionalities. Moreover, the data structure they are based upon does not facilitate close linkage to GIS and decision support system (DSS) (National Research Council 1999). This increases the model setup time, hinders the analysis of model output results, compounds data isolation, reduces data integrity, and limits concurrent access of data because of broken data flow between the data, physical model, and DSSs. The problem is acute when dynamic interaction is required during the model simulation. A need to seamlessly link individual GIS and physical modeling systems provides the motivation for this paper.

Important efforts in bridging the gap between a hydrologic model and a GIS include the development of Hydrologic Data Development System (HDDS; Smith and Maidment 1995) based on ARC/INFO, water and erosion prediction project (WEPP) interface on GRASS (Engel *et al.* 1993), an interface between ArcInfo and HEC modeling system (Hellweger and Maidment 1999), BASINS by EPA (Lahlou *et al.* 1998), SWAT by Di Luzio *et al.* (2002), inland waterway contaminant spills modeling interface (Martin 2002 and Martin *et al.* 2004), and Watershed Modeling System (WMS; Nelson 1997). A detailed overview of attempts to develop hydrologic models inside GIS is reviewed by Wilson (1999). We note that all the above approaches were basically trying to 'couple' a GIS and a process-based hydrologic model for efficient processing, storing, manipulating, and displaying of hydrogeological data. WMS was a major development and different from other attempts in that it was a stand-alone GIS system totally dedicated to hydrologic application. The development of Arc Hydro (Maidment 2002) was another important step in defining an exhaustive data model for a hydrologic system and providing a framework for storing and preprocessing the geospatial and temporal data in GIS. The developed data model provided rules for the structure, relationships, and operations on data types often used in hydrologic modeling. McKinney and Cai (2002) went a step further in reducing the gap between GIS and models by outlining an object-oriented methodology to link GIS and water management models. In the process, they identified the methods and objects of the water management models that can be represented as the spatial and thematic characteristics in GIS. One criticism of object orientation-based integration has been its susceptibility to produce monolithic systems that need recompilation and linking to create new versions, resulting in slow model development, evaluation, and testing by independent users (He *et al.* 2002). However, for cases where (high-frequency) dynamic interactions between data and the physical model are desired, such as in a fully coupled hydrologic model that uses temporally adaptive mesh refinement, alternative system integration implementations based on service orientation (Zhu 2009) and modeling frameworks (Blind and Gregersen 2004) are slower. Object-oriented integration based on shared methods and data structure are relatively fast and robust (integrity preserving) in such situations.

In this paper we propose a robust integration methodology that facilitates seamless data flow between the data and model functionalities, thus making interactions between them fluid and dynamic. The objective of this work is to lay the foundation for a fully integrated and extensible GIS-DHM system through a shared data model that can support both of them. The shared data model provides (a) flexibility of modification and customization, (b) ease of access of GIS data structure by the hydrologic model, (c) richness for representing complex user-defined spatial relations and data types, and (d) standardization easily applicable to new model settings and modeling goals. The data model has been developed using state-of-the-art computer programming concepts of object-oriented programming (OOP). We also discuss in detail the intermediate steps of designing the shared data model from a GIS data model. The emphasis in this exercise is elucidating program design, not the coding details. The resulting

data model supports an open-source coupled framework that serves as a GIS interface to Penn State Integrated Hydrologic Model (PIHM) and is called PIHMgis (<http://sourceforge.net/projects/pihmgis/>). The strategy presented here shows that the concepts and capabilities unique to the coupling approach can easily be implemented in other GISs and DHMs.

2. Integration methodology

Efforts to couple GIS with hydrologic models generally follow either a loose, tight, or embedded coupling (Goodchild 1992, Nyerges 1993) strategy (see Table 1). Watkins *et al.* (1996) and Paniconi *et al.* (1999) have discussed in detail the relative advantages and disadvantages of coupling in terms of watershed decomposition, sensitivity and uncertainty analysis, parameter estimation, and representation of the watershed. Loose coupling is prone to data inconsistency, information loss, and redundancy, leading to increased model setup time. At the same time, loosely coupled approaches are much simpler to design and program. At the other extreme, embedded coupling can leave the code inertial to change because of its large and complex structure (Goodchild 1992, Fedra 1996). Nonetheless, embedded coupling provides the dynamic ability to visualize and suspend ongoing simulations, query intermediate results, investigate key spatial/temporal relations, and even modify the underlying hydrologic model parameters (Bennett 1997).

From our point of view both tight and embedded coupling strategies offer the necessary degree of sharing between GIS and hydrologic model for efficient data query, storage, transfer, and retrieval. We also note (from Table 1) that both coupling strategies underscore the existence of a shared data model in their implementation. Clearly, the integration of GIS tools and simulation models should first address the conceptual need of a shared data model that is implemented on top of a common data and method base. In order to design such a

Table 1. Characteristics of different levels of integration between a GIS and a hydrologic model.

Characteristics	Coupling level		
	Loose	Tight integration	Embedding
Shared user interface	×	✓	✓
Shared data and method base	×	✓	✓
Intra-simulation model modification	×	×	✓
Intra-simulation query and control	×	×	✓
Above translates to →	<ul style="list-style-type: none"> ● Distinct GIS and hydrologic modeling packages with individual interfaces ● Information sharing through file exchange which can be tedious and error prone ● Underlying advantage is: different packages facilitate independent development 	<ul style="list-style-type: none"> ● Data exchange is automatic ● Merges different tools in a single powerful system ● Avoids inconsistency and data loss originating from redundancy and heterogeneity of method base 	<ul style="list-style-type: none"> ● Steerable numerical simulation in terms possibility of changes in parameter or processes while running ● Significantly complex programming and data management ● Changes to the code are not easy because of its monolithic structure

shared data model, we follow a four-step approach. First we carry out identification and classification of the various data types that form the hydrologic system (Section 3). Then we design the object-oriented data model for the data types identified in the previous step (Section 4). In the third step, we study the hydrologic model structure in terms of its data needs and adjacency relationships (Section 5). Finally, re-representation of the GIS data model classes to conform to the DHM data structure is carried out (Section 6). Next we discuss in detail the design steps of the shared data model.

3. Conceptual classification of raw hydrologic data

A hydrologic model domain encompasses a wide range of hydraulic, hydrologic, climatic, and geologic data, including topography, rivers, soil, geology, vegetation, land use, weather, observation wells, and fractures. A conceptual classification of the raw hydrologic data needs to incorporate data of different origins, representation types, and scales.

Figure 1 illustrates a hierarchical categorization of real data typically required in hydrologic models. The design is intended to incorporate spatially heterogeneous thematic data types along with associated time series data, derived data, and attributes. The data types can be defined as field based and object based (Couclelis 1992, Goodchild 1992). Field-based data define a spatial (or temporal) framework consisting of a set of locations related to each other by (temporal) distance, direction, and contiguity (Galton 2001). Object-based data are collection of individual entities that are characterized by geometry, topology, and nonspatial attribute values (Heuvelink 1998). Spatial information to these entities is explicitly defined either as attributes or as a function of location that is inherent in a point, line, or polygon. We note that this kind of distinction in GIS features has been traditionally associated with raster and vector data only. However, here we extend the concept of field data by considering it as a ‘continuous concept’ whose unitary element exists either in space or time with respective entity information attached to it. For example, a unit element of any tessellation, like a grid or a TIN (triangular regular network), has an associated value that defines a property/characteristic

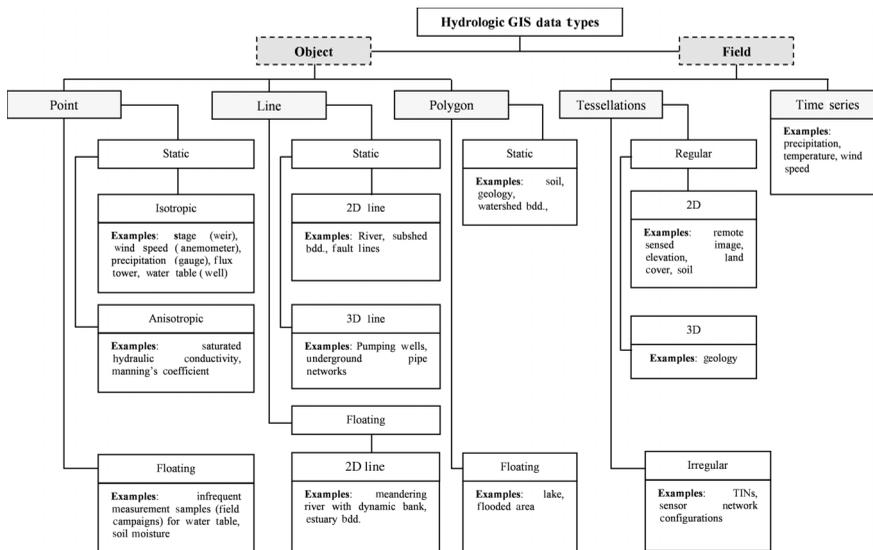


Figure 1. Conceptual classification of existing GIS data types relevant to hydrologic modeling.

magnitude/value anywhere within the field boundary. Similarly for a time series, there is a value attached to any instant in the time series.

Figure 1 shows further subclassification of 'field' and 'object' data types that are relevant to hydrologic modeling. An object consists of points, line, and polygons. The fundamental scope of the object subdata types has been extended in order to incorporate complex features (made up of multiple simple features) and the dynamic nature of observer and observables. We classify points as static and floating depending on their primary existence in space or time. For example, a static point can be identified by a location at which time series data such as wind speeds are being observed. On the other hand, an example of a floating point can be a volunteer in a soil moisture measurement field campaign who goes around the field taking soil moisture samples at different locations. In the former case, the observer is fixed in space and is observing state in time while in the latter case a continuous time clock is fixed to the observer while he/she moves around and takes sporadic samples at different locations. Static points have been further subcategorized into isotropic and anisotropic points. Anisotropic points are locations whose entity attributes need information regarding direction and magnitude and possibly magnitude changing with direction (e.g., a second-rank tensor). An example of an anisotropic property representation at a point is hydraulic conductivity (Freeze and Cherry 1979). Line objects have been subcategorized into standard two-dimensional (2D) and 3D lines. The 3D polylines are made up of line segments that exist in three dimensions, for example, an underground pipe network for drainage/waste removals, etc., which can change directions/planes in three dimensions at junctions. Polygon objects have been subcategorized into static and floating polygons. Floating polygons are bounded regions whose areas changes in time, such as a flooded region or a lake. Field objects have been subclassified into tessellations (spatial) and time series (temporal) components. The unitary elements of tessellations define units of spaces with entity information attached to it.

The conceptual representation discussed above is generic and acts as a template that can be populated by new data. Next we try to formally represent the data types in classes and identify their attributes and their relationships with other classes.

4. Hydrologic data model design

A hydrologic data model is a formal representation of the real world that provides a standard structure for storage, sharing, and exchange of data independent of the software environment and programming languages. It provides a simplified abstraction of reality by (a) isolating real-world hydrologic objects into independent classes, (b) removing redundant class objects, (c) defining relationships between independent classes, and (d) defining integrity constraints on them.

The design of a hydrologic data model is performed keeping in mind the range of required data types (see Figure 1) and their relationships among themselves (Wright *et al.* 2007). Some data, such as elevation and soil properties, vary continuously in space while others like observed streamflow vary continuously in time. The representation of data also changes depending on the scale of interest. On a coarse scale the stream channel can be represented as a 1D curvilinear object, while on a finer scale it can be considered as a 3D topographic section with width, depth, and length. For longer timescales such as climate change or landscape evolution studies, the stream channel representation will also need a time identifier in addition to width, depth, and length attributes. These are necessary in order to track the changes in shape over time due to erosion/deposition on the riverbed or banks. This means that the designed data model (a) must have the *flexibility* to incorporate different

representations of the same object at different scales, (b) should be *extensible* with a potential to incrementally enrich it with new data types and construct complex objects, and (c) should be *robust* and *adaptable* to changing hydrologic conditions by using different instances of a single object (reusability). Maximum information, minimum data redundancy, reduction of storage capacity, and optimum retrievability of data for analysis are the desired objectives in the design process. All these characteristics are sufficed by designing the data model using object-oriented concepts of inheritance, polymorphism, and encapsulation.

The object-oriented data modeling strategy provides a formal definition of objects, their attributes and behaviors, and the operations that can be performed on them (Milne *et al.* 1993, Alonso and El Abbadi 1994, Raper and Livingston 1995). Features sharing a set of attributes and methods are clustered into a single class. Each instance of a class is called a data model object. An example of a class is a line feature and one of its instances is a river. Attribute fields of the river line are an integer identifier, number of line segments, and start and end points of each segment. Methods are the functions that define the interaction of objects to the outside world. For example, calculation of total flow volume by using the river dimension attributes is a method associated with the river object. While every object in a class shares some of the same set of attributes and methods, they may have additional properties attached to them. In addition to descriptions about objects, its attributes, and behaviors, the data model also explains the relationship between classes. For example, in order to account for flow and interactions between each river segment and the watershed, and also to streamline query and storage, a definition of (topological) relationships between classes is needed. Generalization, association, and aggregation are the three main relationships that have been implemented in the data model. The generalization relationship connects a child class to a base class using object-oriented ‘inheritance’ mechanism. The subclasses of a base class share many properties between themselves while separating from each other on the basis of new ‘identity’ properties. Association shows the relationship between instances of classes that exist either in time or in space. These linkages are either bidirectional, which means that both of the connecting classes are aware of the relationship with each other, or unidirectional, where only one of the classes knows about the relationship. This relationship markedly simplifies and clarifies the data model, and minimizes redundancy in definitions, access, and storage. The developed data model also uses another type of linkage, called reflexive association. This linkage relates different instances of the same class. Aggregation relationships have been implemented to explain the interaction of individual parts/components (simple objects) to a complex object.

The formal static structural representation of data model classes and their attributes and relationship is done using three-compartment Unified Modeling Language (UML) class diagrams (shown in Figure 2a). UML class diagrams provide a programming language-independent view of the static structure and behavior of classes. We note that

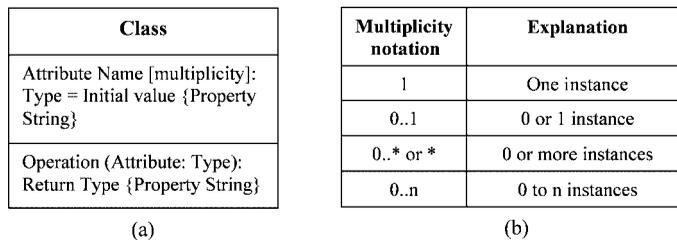


Figure 2. (a) Three-compartment structure of class icons. Options listed inside curly or large brackets are optional. (b) Cardinality/multiplicity notation of relationships in a class diagram.

since the relationship between classes can be one-to-one or one-to-many, each relationship representation appropriately indicates the multiplicity of instances (examples shown in Figure 2b). In order to define the directionality in relationships between classes, UML class diagrams follow standardized notations. Generalization relationships are represented by line drawn from a child class to a base class with a white, solid arrow at the end. Unidirectional associations are represented by single-ended arrowheads where the class from which the arrow initiates is the class which has knowledge of the relationship. The aggregation relationship is denoted by a white diamond (for the aggregate class) on one end of the link and arrow (for the 'part' class) on the other.

Using the UML structure, we represent six primary classes: feature point, feature line, feature polygon, feature volume, grid, and time series (see Figure 3, Appendix). The instance objects of each of these classes can be seen in the conceptual diagram of the data model in Figure 1. A point class is completely defined by its location and attributes. Anisotropic and floating points are child classes of the feature point, which means that though they inherit the properties of point class, they have additional properties such as direction and time, respectively, that uniquely identify them. Line class is basically a collection of line segments that join nodes (points). The multiplicity/cardinality of the aggregation relationship of points to a line class varies from 2 to NumPts. Similarly, line classes aggregate to form polygons. A polygon must have at least three lines. Polygons aggregate to form a feature volume. Three-dimensional feature volumes are an aggregation of two polygons. Figure 4 explains the design of first four feature objects. We note that all the features have an existence in three dimensions. This is particularly important for the accurate characterization of hydrologic data such as watershed boundaries, subsurface properties, or even measurement stations at or above ground which have existence in three dimensions (e.g., met towers). The aggregation

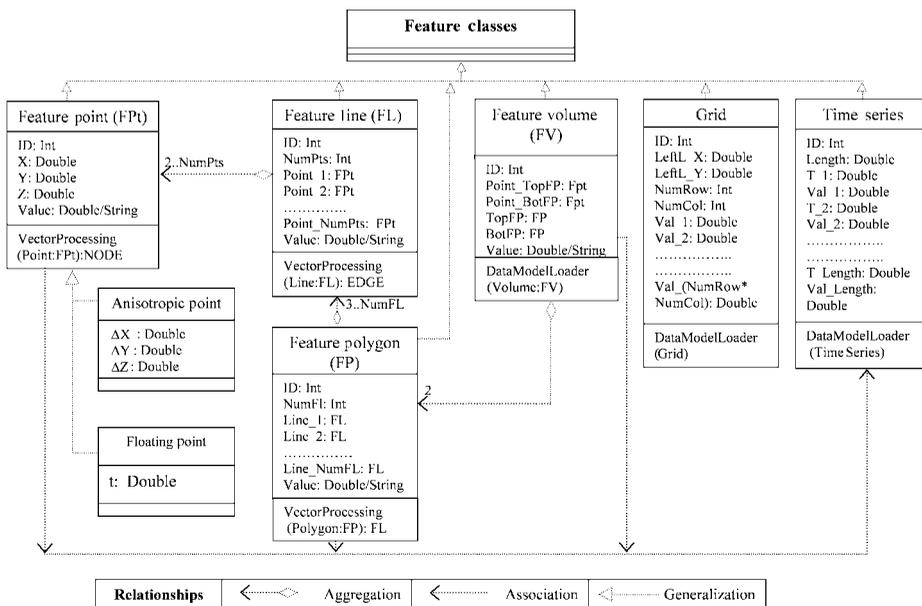


Figure 3. GIS data model class diagram design for hydrologic data in UML 2.0. Note the type and cardinality of relationships between various classes (details in Section 4). The operators in the bottom compartment for each individual class are used in transformation of GIS data model into a shared data model structure that is valid on hydrologic model grids.

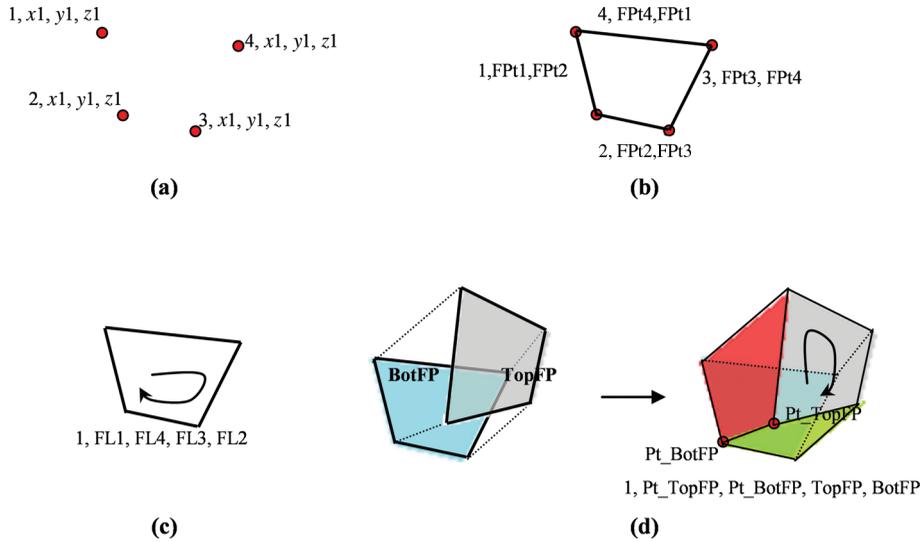


Figure 4. Feature object designs for (a) point, (b) polyline, (c) polygon, and (d) volume. Note the implicitness of the ‘sequence of constructs’ in feature polygon and feature volume design. For example, (c) shows that edge polylines of the polygon are always listed in clockwise direction. Similarly, definition of a 3D feature necessitates identification of a pivot point and boundary polygons in a particular sequence. Note that the identification of one point from both top and bottom polygon in design of feature volume is done in order to pivot the connection sequence of the nodes of the two polygons which results in a 3D feature.

relationships show how the traditional 2D simple objects like points and lines are used to make a composite higher-dimension complex feature. One such example is the description of an underground water pipe network, which is basically a collection of straight pipes that zigzag through the subsurface in various planes. We note that the directionality (clockwise or counterclockwise) of feature line sequence or of connections between polygons is inherently defined by the definition of a feature polygon or feature volume, respectively. Figure 3 also shows details of a time series data class, which is related to the feature objects through unidirectional association.

The developed hydrologic data model acts as a transitional formal representation that bridges the gap between the raw data types and their seamless assimilation in hydrologic applications. Independently, the data model serves as a template to store and organize raw hydrologic data in GIS. For the data model to be used seamlessly in hydrologic modeling, the data structure and relationships need to be modified such that they support representation of data and relationships on a hydrologic model grid. The eventual goal of course is to have a shared data model that can fully describe the hydrologic GIS data objects (shown in Figure 3) as well as their representational complement in the hydrologic model.

5. Hydrologic model structure: process representation and adjacency relationships

The conceptualization of process interactions and the shape and adjacency property of unit elements in the model grid control the design of the hydrologic model data structure. Here we highlight the data and topological needs of the hydrologic model data structure vis-à-vis a finite volume-based PIHM (Kumar 2009, Qu and Duffy 2007). We reiterate that all the steps taken are generic and can be used as a template in other GIS–hydrologic model coupling

efforts that are based on different mesh decomposition strategies (e.g., structured meshes for finite difference models). Next we highlight how the representation of physical processes and discretization of the model domain influences the hydrologic model data structure.

5.1. Physical process interaction

PIHM is a finite volume–based integrated hydrologic model. It simulates multiple physical states on discretized elements (also called model kernel) of a watershed domain by solving semi-discrete form of ordinary differential equations (ODEs) (Leveque 2002) given by

$$\frac{d\bar{\chi}}{dt} = \sum_{k=1}^2 Q_k - \sum_{i=1}^3 Q_i \quad (1)$$

where $\bar{\chi}$ is interpreted as the volumetric storage (L^3), Q_i is the net volumetric flux through the (three) sides of the control volume, and Q_k is the net volumetric flux through upper and lower boundaries. Details of the individual differential equations [of the form shown in Equation (1)] corresponding to each hydrologic processes, such as channel flow, overland flow, unsaturated zone storage, groundwater flow, interception storage, and snow melt, can be referred to in Qu and Duffy (2007). The critical point to note here is that solutions of ODEs

Table 2. Data requirements for calculation of physical states on a model kernel at any simulation time.

Process	Data support
Channel flow	Head in adjacent triangular elements, head in river segment downstream and upstream, initial head value at the start of simulation, precipitation, evaporation, Manning's coefficient, coefficient of discharge for weir flow across river bank, elevation of end nodes of river segment, leakage coefficient, subsurface flow head in adjacent triangles, boundary conditions Note: Head → overland flow (unless specified otherwise)
Overland flow	Head in neighboring elements, head in river segment (if river is neighbor to the prismatic cell), initial head value, net precipitation, evapotranspiration, elevation of nodes of triangular element, boundary conditions Note: Head → overland flow
Unsaturated flow	Capillary flow, initial head value, subsurface flow head, infiltration, hydraulic conductivity, evapotranspiration, root uptake, soil porosity, van Genuchten soil parameters, boundary conditions Note: Head → unsaturated flow
Groundwater flow	Head in adjacent triangles, initial head value, capillary flow, hydraulic conductivity of the elements and its neighbors, bedrock depth, soil porosity, van Genuchten soil parameters, boundary conditions Note: Head → groundwater flow
Interception	Interception storage capacity, precipitation, LAI, evapotranspiration, initial interception
Snow melt	Initial snow depth, initial snow density, initial snow surface layer temperature, initial average snow cover temperature, average snow liquid water content, net solar radiation, incoming thermal radiation, air temperature, vapor pressure, wind speed, soil temperature, precipitation
Infiltration	Overland flow head, unsaturated soil moisture, hydraulic conductivity, porosity, macropore, precipitation rate, maximum infiltration capacity
Evapotranspiration	Wind speed, humidity, net radiation, soil heat flux, vapor pressure deficit, mean air density, interception storage capacity, LAI, soil saturation, atmospheric resistance, stomatal resistance, vegetation fraction, unsaturated zone saturation

for overland and groundwater flow depths depend on the head in adjacent kernels (Qu and Duffy 2007). Similarly, channel head is dependent on lateral fluxes from upstream and downstream channel sections, and the watershed. This means that a design of the hydrologic model data structure must incorporate the topological relationship between neighboring unit elements. In addition to these relationships, Table 2 also lists the data requirements for calculation of each physical state on every model kernel at any time. An inclusive hydrologic model data structure will account for the data requirements at all times.

The hydrologic model data structure is also influenced by the shape and adjacency of unit elements, which are in turn defined by the choice of domain decomposition (structured and unstructured) and numerical solution strategy (finite element, difference, or volume) employed in modeling.

5.2. Domain decomposition

PIHM uses unstructured meshes to decompose the domain. The individual unit control volume elements are either prismatic (for watershed elements) or trapezoidal/cuboidal (for river elements). The flux exchange in prismatic elements takes place through five boundary faces (Kumar 2009). If a model uses structured grids to decompose the domain, then the number of faces across which flux exchange can potentially take place in three dimensions will be equal to 6. So the shape of the unit element also determines how the relationships between neighboring elements need to be represented in a hydrologic model data structure. We note that the unstructured mesh decomposition poses additional challenges in the design of a hydrologic model data structure, particularly in terms of definition of topological relationships, compared to structured grids, where the neighbors are implicitly characterized by the decomposition itself.

With the object-oriented hydrologic data model in place (Section 4) and the spatial relationships and parameter definitions for the hydrologic model identified (in this section), the last step in shared data model design is to represent the hydrologic GIS data types and the hydrologic model structure using the same feature classes, thus providing an automatic connection between GIS and the hydrologic model. The next section discusses the design of this shared data model.

6. Shared data model design

The shared data model captures the spatial structure of hydrographic features and temporal objects by identifying six classes: node, element, channel, soil, land cover, and time series (shown in Figure 5). These classes are representational complements of the six GIS data model classes (see Figure 3) and can be obtained by applying appropriate transformations or redefinitions. The relevant geometric, spatial, and topological transformations performed on GIS data types are shown in Figure 6. By generating mesh decomposition using points and lines as constraints (more details in Kumar *et al.* 2009), nodes of the triangles automatically act as feature points and edges of the triangles act as feature lines. Properties and attributes of boundaries of the feature polygon are assigned to the element edges after converting the polygons to polylines and then to lines. Attributes of the feature polygons and feature volumes are geographically registered to the triangular elements. We note that all the re-representations of hydrologic GIS data types are 'loss-less' mappings, implying that they are reversible. By aggregating element edges, channels, or elements based on their attribute properties, we can revert back from a shared data model class to the original GIS data objects. The operators used in re-representation of classes are shown over the lines

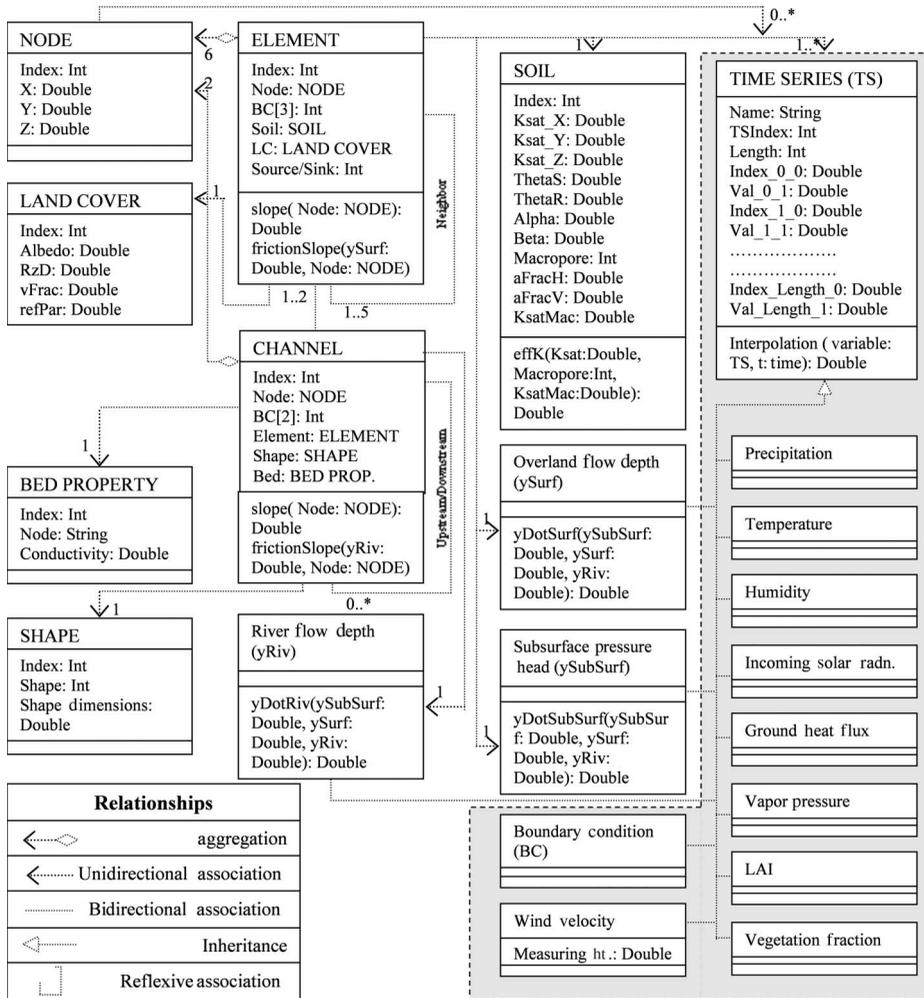


Figure 5. Shared data model class diagram design for GIS–hydrologic model coupling in UML 2.0. Note the type and cardinality of relationships between various classes (details in Section 6).

connecting the source and result classes in Figure 6. These operators are also listed as ‘methods’ (in the bottom-most compartment) in the GIS data model class diagrams (see Figure 3). The name of each of these operators is self-explanatory for their functions. We note that the dotted line in the transformation diagram indicates the intermediate results.

Figure 5 also shows the aggregation, unidirectional association, reflexive association, and generalization relationships supported in the shared data model. An element class represents a discretized triangular element in two dimensions and a prismatic element in three dimensions, and is defined by six nodal locations listed in a clockwise direction at two levels. The prismatic element has five neighbors – three on the sides and one each at the top and the bottom. We note that neighbors of an element also belong to an element class and this recursive relationship is captured by reflexive association. The cardinality of this relationship is 1 to 5, which means that there has to be at least 1 neighboring element to an element object. A maximum cardinality of 5 denotes that a 3D element can have a total of 3 lateral

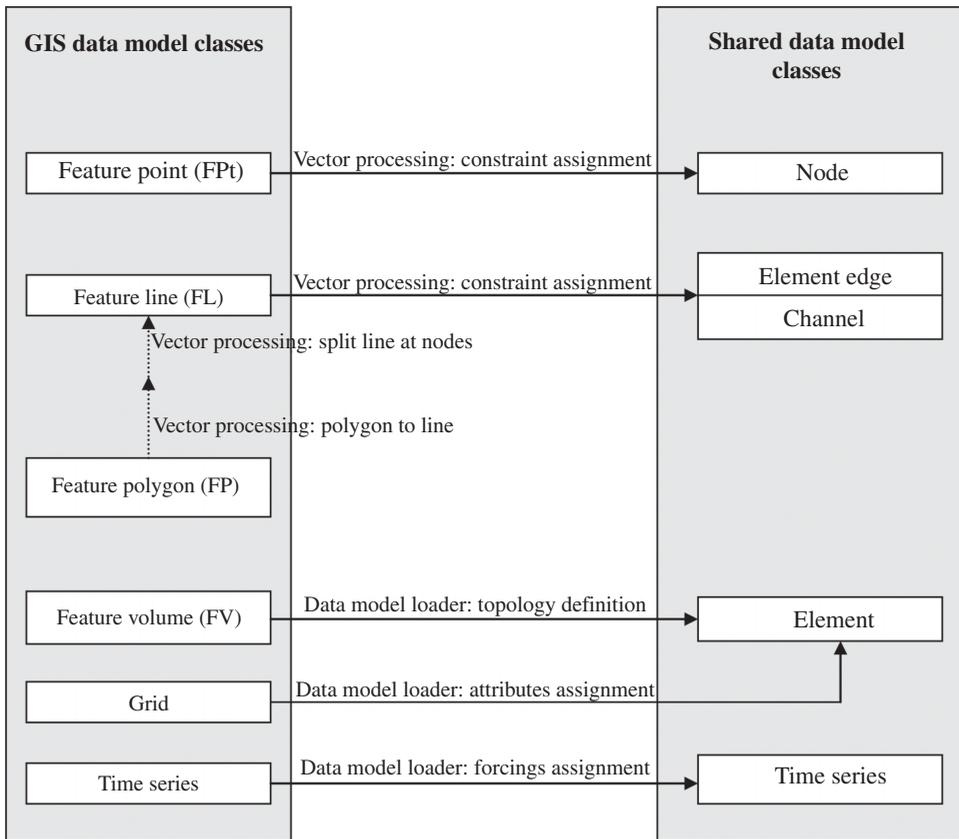


Figure 6. Class re-representation diagram showing the transformation of a GIS-based data model classes into classes identified in shared data model design. The arrows originate from each individual GIS data model class and end in the corresponding complement shared data model class. Operators/functions that perform this transformation are shown along the arrows. Dotted arrows represent intermediate transformation operations.

and 2 vertical neighbors. A channel class is defined by the two end nodes and neighboring elements on the either side of channel. Each channel segment is also composed of an upstream and downstream channel segment, which is captured by a reflexive association. We note that the multiplicity of this relationship varies from 0 to any integer value. This means that a channel segment can stand alone in the watershed with no upstream or downstream channels. A channel is also bidirectionally associated with an element with a multiplicity of 1 to 2. This translates to existence of at least 1 neighboring triangular element to a channel segment. Bidirectionality ensures that both element and channel are aware of this topological relationship. These relations are fundamentally important for spatial integrity of the hydrologic modeling framework. Each element class is also associated with soil, land cover, and time series class. This ensures clean and efficient assignment of properties to each element. Similarly the channel is associated to bed property and shape classes. The soil class contains several attribute fields such as hydraulic conductivities and van Genuchten equation soil retention parameters (van Genuchten 1980). Attributes of the land cover class are root zone depth, albedo, and photosynthetically active radiation from each land cover type. We note that precipitation, temperature, humidity, incoming solar radiation, ground

heat flux, vapor pressure, leaf area index (LAI), vegetation fraction, wind velocity, time-dependent boundary conditions, and the observed and simulated state variables are all instances or child objects to the time series class. Names of the operators shown in Figure 5 are self-explanatory regarding their functions. These operators are concerned with derivation of geometric properties of triangular elements and channels or with the calculation of rate of change of state variables with time. The definitions of various functions are given in the Appendix.

The shared data model design is tested in the development of a coupled GIS–hydrologic modeling system. The integrated software is an open-source, platform-independent, extensible, and ‘tightly coupled’ integrated GIS interface to PIHM and is referred to as PIHMgis.

7. PIHMgis

PIHMgis is an integrated and extensible GIS system with data management, analysis, data modeling, unstructured mesh generation, visualization, and distributed PIHM modeling capabilities. The underlying philosophy of this integrated system is a shared geodata model between GIS and PIHM, which was developed in the previous sections. The shared data model makes it possible to handle the complexity of the representation structures, data types, model simulations, and analysis of results. The graphic interface component of PIHMgis has been written in Qt and C++, which support object-oriented class structures in programming. PIHMgis sits on an open-source Qgis engine (<http://www.qgis.org>) and has been integrated as pluggable software. The interface and the source code can be downloaded from http://www.pihm.psu.edu/pihmgis_downloads.html.

The architectural framework of the interface is shown in Figure 7. The directionality of the arrows indicates the possible flow of output from one method to another. The flow of actions between different class objects in PIHMgis can be shown using an object-oriented UML collaboration diagram (Figure 8). These diagrams represent both the static and the

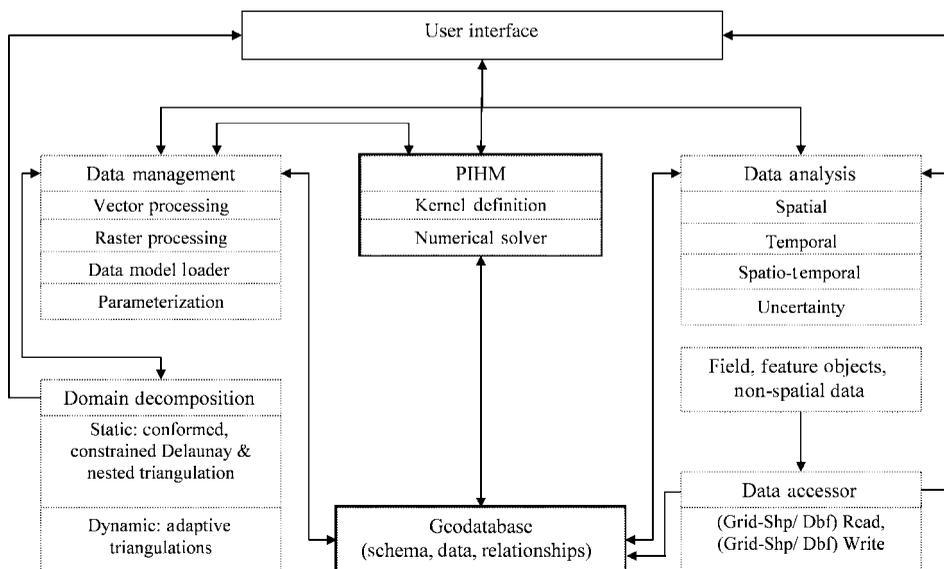


Figure 7. Architectural framework of PIHMgis. Directionality of the arrows indicates the possible flow of output from one module to another.

dynamic behavior of the system by representing collaboration (simple associations) between objects and mapping the sequence of messages they share between them. The rectangles in the diagram enclose the class and its instance (separated by a colon), and the links between rectangles represent the collaborations (communications) between classes. The chronological labeling of the messages between class objects describes the sequence in which actions are executed. The first communication initiated by the integrated system is from the object from where message 1.0 is released. In order to track the messages/actions that are hierarchically associated with a parent object, a nested numbering of messages is performed. Figure 8 shows that a full hydrologic modeling exercise can be carried out in PIHMgis by directly acting upon the raw data types represented in the shared data model, merely by launching a sequence of messages (commands). Starting with digital elevation model raster data, which is an instance of grid class, raster processing operations result in delineation of watersheds, definition of streams, and extraction of very important points (VIPs). A vector processing tool with polyline reconditioning algorithms simplifies and splits the watershed boundaries and channel segments. Thereafter, vector merging of all the available feature layers is performed to create a spatial support for generating constrained domain decomposition. Details about the need of vector processing steps and how they aid flexible domain decompositions are in Kumar *et al.* (2009). Once domain decomposition has been performed, topology definitions and field assignment of properties, and initialization of state variable on each model kernel is performed. A numerical solver module formalizes all the ODEs in each model kernel in the form of $y' = f(y)$ and then solves the system iteratively. Output results in the form of spatial and time series plots are displayed in the visualization toolkit integrated in PIHMgis. Details about all the operator functions in the PIHMgis toolkits are discussed in Bhatt *et al.* (2008).

8. Advantages of shared data model for GIS–hydrologic model coupling

A shared database, and relationships and schemata between GIS and the hydrologic model reduce the model setup time, enhance the data integrity, and streamline the model simulations. As a result, the integrated system simulates the model states accurately and efficiently, steers simulations, and conveniently manages, analyzes, and displays data used and produced by the model. The unique advantages of coupling based on a shared data model development are discussed next.

8.1. Enhanced accuracy and computational efficiency

As mentioned in Section 6, the hydrologic model grids supported by the shared data model are generated by using GIS points, polylines, and polygons as constraints. The unique advantage of using GIS objects as constraints for decomposition is that the resulting model grid can be designed to follow the edges of a single property type (such as soil, land cover, geology, and vegetation). This maintains the data integrity and limits the introduction of additional data uncertainty arising from statistical averaging of multiple class themes within a model grid (Kumar *et al.* 2009). Comparatively, structured grid decomposition will always have a large number of cells with mixed themes. For the same order of accuracy of representation of both raster and vector data, constrained decompositions also use a smaller number of cells (or computational elements) relative to structured meshes (Kumar *et al.* 2009), thus resulting in computational efficiency. Similarly if observation stations (point objects) are used as a constraint in decomposition, hydrologic states can be predicted exactly at the observation stations. The georeferential integrity inherent in the shared data model minimizes any errors during comparison of observed and predicted states, which creep in due to interpolation of prediction variables to the observation locations.

8.2. Storage efficiency

In any watershed model, there are a limited number of parameters and forcing types (e.g., soil, land cover, and precipitation) that are needed to define each hydrologic property over the domain. This translates to storage efficiency at two levels in a shared data model approach. First, the efficiency is gained through storage of (soil or forcing) properties as relational objects, which also ensures that these properties are accessible to both the GIS and the hydrologic model. For example, instead of storing all the nine soil attribute parameters (floating type numbers) as separate grids, we are able to store them as a single layer of soil types (an integer attribute of element class) with associative relations defined for all the nine attributes of soil class. The compression is even more significant in storage of forcing time series, such as those of precipitation, Ppt, and temperature, T . Rather than storing the forcing grid at numerous time steps (e.g., satellite images of time series variables such as temperature), the precipitation-type attribute for each element class is associated with a precipitation magnitude within a time series class. The associative relationships limit the data redundancy by avoiding use of multiple sets of similar data. Significant storage efficiency is also gained due to the description of the data on constrained Delaunay triangulations.

8.3. Model setup, real-time visualization, and decision support

The simple, compact, and procedural structure of PIHMgis (see Figure 8) streamlines the process of organizing the data for model simulations. PIHMgis allows the user to perform semi-automated preliminary model simulations with minimum user input. The ease of use of the coupled system can be judged from the fact that graduate students with no prior knowledge of modeling (in an introductory groundwater modeling class) are able to perform uncalibrated simulations after two training lectures.

The architectural framework of PIHMgis in Figure 7 shows that the outputs from the model simulations continually update the geodatabase of the shared data model. This means that any selected number of state variables or fluxes can be plotted at any location while the simulation proceeds. This is particularly useful in assessing whether the simulation results are physically realistic, and gives an opportunity to adjust the model or make management decisions in real time. Real-time visualization also serves as an 'early warning' system to track errors in simulation arising from wrong/bad data input or numerical 'blowup'. During the simulation the user can search for the appearance of nonphysical states in real time and immediately detect problems in the solution.

8.4. Parameter steering

DHM calibration and sensitivity analysis of parameters require performing multiple model simulations. Since a shared data model stores GIS data in a hydrologic model grid structure, the coupled GIS-model system provides unique flexibility in modifying parameters or forcing values in any selected portion of the watershed. For example, if it is found during calibration that the LAI for a particular land cover type is resulting in under-prediction of interception storage, the shared data model can efficiently query all elements of that particular land cover type and perform the required parameter nudging. For traditional approaches with an isolated data model and data structures, changes in parameters (such as LAI) in a particular region would require GIS processing on the raw data and generation of new input files. In summary, a streamlined data structure and relationship definitions of a shared data model result in an efficient, integrated, and automated steering of parameters.

9. Conclusions

This article presents the design and details of a shared data model that can support coupling of GIS with a hydrologic model. The conceptualization and characterization of this coupling strategy can be used with other physically distributed models and can be extended to management, visualization, and decision support tools (e.g., ecological models). The data model is rich yet flexible in terms of its extensibility and simplicity. The data model incorporates representation of wide range of data types varying from static and floating points to 3D feature line and volume objects. The object-oriented strategy streamlines the design of the data model and clarifies the relationships between classes. UML class and collaboration diagrams have been developed to show the standardized static and dynamic structure of classes, their operations, and activity in the larger software framework. It also provides a clear modular sequencing of operations in the coupled software. The object-oriented data model design leads to seamless assimilation of the classes and their relationships directly in object-oriented software development. The shared data model is successfully used to develop a prototype open-source, platform-independent coupled modeling system referred to as PIHMgis. The shared data model concept creates a process for modeling that improves data flow, model parameter development, parameter steering, and designing an efficient grid and allows real-time visualization and decision support.

References

- Abel, D.J. and Kilby, P.J., 1994. The systems integration problem. *International Journal of Geographical Information Systems*, 8, 1–12.
- Alonso, G. and El Abbadi, A., 1994. GOOSE: Geographic Object Oriented Support Environment. In: *Proceedings of ACM/ISCA workshop on advances in geographic information systems*, Campinas, Brazil, March 1994, 38–43.
- Bennett, D.A., 1997. A framework for the integration of geographical information systems and model base management. *International Journal of Geographical Information Science*, 11 (4), 337–357.
- Bhatt, G., Kumar, M., and Duffy, C.J., 2008. Bridging the gap between geohydrologic data and distributed hydrologic modeling. In: *Proceedings of international congress on environmental modeling and software*, Barcelona, Catalonia, 7–10 July.
- Blind, M. and Gregersen, J., 2004. Towards an open modelling interface (OpenMI): the HarmonIT project. In: *Transactions of the second biennial meeting of the International Environmental Modelling and Software Society (iEMSs 2004)*, Osnabrück, Germany.
- Couclelis, H., 1992. People manipulate objects (but cultivate fields): beyond the raster–vector debate in GIS. In: A.U. Frank and I. Campari, eds. *Theories and methods of spatio-temporal reasoning in geographic space*, Lecture Notes in Computer Science 639, Berlin: Springer-Verlag, 65–77.
- Di Luzio, M., et al., 2002. *Arcview interface for SWAT2000. User's guide*. Temple, TX: US Department of Agriculture, Agriculture Research Service. Available from: <http://www.brc.tamus.edu/swat/download/doc/swatav2000.pdf> [Accessed 18 March 2008].
- Engel, D.M., et al. 1993. Late-summer fire and follow-up herbicide treatments in tallgrass prairie. *Journal of Range Management*, 46, 542–547.
- Fedra, K., 1996. Distributed models and embedded GIS: strategies and case studies of integration. In: M.F. Goodchild, et al., eds. *GIS and environmental modeling: progress and research issues*. Fort Collins, CO: GIS World Books, 413–417.
- Freeze, R.A. and Cherry, J.A., 1979. *Groundwater*. Englewood Cliffs, NJ: Prentice Hall, 29.
- Freeze, R.A. and Harlan, R.L., 1969. Blueprint for a physically-based digitally simulated, hydrologic response model. *Journal of Hydrology*, 9, 237–258.
- Galton, A., 2001. Space, time, and the representation of geographical reality. *Topoi*, 20, 173–187.
- Goodchild, M.F., 1992. Geographical information science. *International Journal of Geographical Information Systems*, 6 (1), 31–45.
- He, H.S., Larsen, D.R., and Mladenoff, D.J., 2002. Exploring component based approaches in forest landscape modeling. *Environmental Modelling & Software*, 17, 519–529.

- Hellweger, F.L. and Maidment, D.R., 1999. Definition and connection of hydrologic elements using geographic data. *Journal of Hydrologic Engineering*, 4 (1), 10–18.
- Heuvelink, G., 1998. *Error propagation in environmental modelling with GIS*. London: Taylor and Francis.
- Kollet, S.J. and Maxwell, R.M., 2006. Integrated surface–groundwater flow modeling: a free-surface overland flow boundary condition in a parallel groundwater flow model. *Advances in Water Resources*, 29 (7), 945–958.
- Kopp, S.M., 1996. Linking GIS and hydrological models: where have we been, where are we going? In: K. Kovar and H.P. Nachtenebel, eds. *HydroGIS 96: application of geographic information systems in hydrology and water resources*. IAHS Publ. no. 235, 13–21.
- Kumar, M., 2009. *Toward a hydrologic modeling system*. Thesis (PhD). Pennsylvania State University.
- Kumar, M., Bhatt, G., and Duffy, C., 2009. Efficient domain decomposition framework for accurate representation of geodata on in distributed hydrologic models. *International Journal of Geographical Information Science*, 23 1569–1596.
- Lahlou, M., et al., 1998. *Better assessment science integrating point and nonpoint sources: BASINS 2.0 user's manual, EPA-823-B-98-006*. Washington, DC: US Environmental Protection Agency, Office of Water.
- Leveque, R.J., 2002. *Finite volume methods for hyperbolic problems*. Cambridge: Cambridge University Press, 64–85.
- Maidment, D., 1993. Developing a spatially distributed unit hydrograph by using GIS. In: K. Kovar and H. Nachtenebel, eds. *Proceedings of applications of GIS in hydrology and water resources*, Vienna. IAHS publ. no. 211. 181–192.
- Maidment, D., 2002. *Arc hydro: GIS for water resources*. Redlands, CA: ESRI Press.
- Martin, R., 2002. *Agile software development, principles, patterns, and practices*. Upper Saddle River, NJ: Prentice Hall.
- Martin, P.H., et al., 2004. Development of a GIS-based spill management information system. *Journal of Hazardous Materials*, B112, 239–252.
- McKinney, D.C. and Cai, X.M., 2002. Linking GIS and water resources management models: an object-oriented method. *Environmental Modeling & Software*, 17, 413–425.
- Milne, P., Milton, S., and Smith, J., 1993. Geographical object-oriented databases: a case study. *International Journal of Geographical Information Systems*, 7, 39–56.
- National Research Council, 1999. *New strategies for America's watersheds*. Washington, DC: National Academy Press.
- Nelson, E.J., 1997. *WMS v5.0 reference manual*. Provo, UT: Environmental Modeling Research Laboratory, Brigham Young University, 462.
- Nyerges, T.L., 1993. Understanding the scope of GIS: it's relation to environmental modeling. In: M.F. Goodchild, B.O. Parks, and L.T. Steyaert, eds. *Environmental modeling with GIS*. New York: Oxford University Press.
- Paniconi, C., et al., 1999. Integrating GIS and data visualization tools for distributed hydrologic modelling. *Transactions in GIS*, 3 (2), 97–118.
- Qu, Y. and Duffy, C.J., 2007. A semidiscrete finite volume formulation for multiprocess watershed simulation. *Water Resources Research*, 43, W08419, doi: 10.1029/2006WR005752.
- Raper, J.F. and Livingstone, D., 1995. Development of a geomorphological spatial model using object-oriented design. *International Journal of Geographical Information Systems*, 9 (4), 359–383.
- Smith, P.N. and Maidment, D.R., 1995. *Hydrologic data development system*. Austin, TX: Center for Research in Water Resource, The University of Texas at Austin, CRWR Online Report 95-1.
- Van Genuchten, M.T., 1980. A closed-form equation for predicting hydraulic conductivity of unsaturated soils. *Soil Sciences Society of America Journal*, 44, 892–898.
- Watkins, D.W., et al., 1996. Use of geographic information systems in ground-water flow modeling. *Journal of Water Resources Planning and Management*, ASCE, 122 (2), 88–96.
- Wilson, J.P., 1999. Current and future trends in the development of integrated methodologies for assessing non-point source pollutants. In: D.L. Corwin, K. Loague, and T.W. Ellsworth, eds. *Assessment of non-point source pollution in the vadose zone*. Washington, DC: American Geophysical Union, 343–361.
- Wilson, J.P., et al., 1996. GIS-based solute transport modeling applications: scale effects of soil and climate data input. *Journal of Environmental Quality*, 25, 445–453.
- Wright, D.J., et al., 2007. *Arc Marine: GIS for a Blue Planet*. Redlands, CA: ESRI press.
- Zhu, D., 2009. A service-oriented city portal framework and collaborative development platform. *Information Sciences*, 179, 2606–2617.

Appendix

List of symbols

aFracH:	aerial fraction of macropore in horizontal soil section
aFracV:	aerial fraction of macropore in vertical soil section
Albedo:	albedo (reflective fraction) of a land cover type
Alpha:	van Genuchten scaling parameter
Beta:	van Genuchten relaxation parameter
BotFP:	bottom feature polygon
FL:	feature line
FPt:	feature point
Ksat_X:	horizontal saturated conductivity in X-direction
Ksat_Y:	horizontal saturated conductivity in Y-direction
Ksat_Z:	vertical saturated conductivity in Z-direction
KsatMac:	saturated macropore conductivity
LC:	land cover
LeftL_X:	lower left x-coordinate location
LeftL_Y:	lower left y-coordinate location
NumCol:	number of columns in grids
NumFl:	number of feature lines in a polygon
NumPts:	number of points in a feature line
NumRow:	number of rows in grids
<i>t</i> :	time
Point <i>i</i> :	<i>i</i> th point in feature line
Pt_TopFP:	pivot point in top polygon boundary of feature volume
Pt_BotFP:	pivot point in bottom polygon boundary of feature volume
Ppt.:	precipitation time series
refPar:	reference incoming solar flux for photosynthetically active canopy
RH:	relative humidity time series
RzD:	rootzone depth
T <i>i</i> :	<i>i</i> th time index
T_Length:	maximum time index
Theta_S:	maximum porosity
Theta_R:	residual porosity
TopFP:	top feature polygon
Val <i>i</i> :	value at <i>i</i> th index
Val_(NumRow × NumCol):	field value at grid location (NumRow, NumCol)
vFrac:	vegetation fraction
VP:	vapor pressure time series
ySurf:	overland flow depth
yRiv:	river stage
ySubSurf:	moisture head

List of functions

areaChannel():	function to calculate cross-section area of the channel element
areaElement():	function to calculate surficial area of the prismatic element
effK():	effective conductivity of the subsurface
frictionSlope():	function to calculate friction slope
Interpolation():	function to interpolate value of a time series at any time using the parsimonious information in time series data structure
yDotRiv():	function to calculate rate of change of river stage
yDotSurf():	function to calculate rate of change of overland flow depth
yDotSubSurf():	function to calculate rate of change of moisture head