

1-1-2013

## Syntactic and Semantic Interoperability Among Hydrologic Models

Mostafa Mohamed Elag  
*University of South Carolina*

Follow this and additional works at: <https://scholarcommons.sc.edu/etd>



Part of the [Civil and Environmental Engineering Commons](#)

---

### Recommended Citation

Elag, M. M.(2013). *Syntactic and Semantic Interoperability Among Hydrologic Models*. (Doctoral dissertation). Retrieved from <https://scholarcommons.sc.edu/etd/2356>

This Open Access Dissertation is brought to you by Scholar Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact [digres@mailbox.sc.edu](mailto:digres@mailbox.sc.edu).

SYNTACTIC AND SEMANTIC INTEROPERABILITY  
AMONG HYDROLOGIC MODELS

by

Mostafa Mohamed Elag

Bachelor of Science in Civil Engineering  
Cairo University, Egypt, 1999

Master of Science in Irrigation and Hydraulics  
Cairo University, Egypt, 2008

---

Submitted in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy in

Civil Engineering

College of Engineering and Computing

University of South Carolina

2013

Accepted by:

Jonathan L. Goodall, Major Professor

M. Hanif Chaudhry, Committee Member

Michael E. Meadows, Committee Member

Michael N. Huhns, Committee Member

Lacy Ford, Vice Provost and Dean of Graduate Studies

© Copyright by Mostafa Mohamed Elag, 2013  
All Rights Reserved.

## ACKNOWLEDGMENTS

This work is dedicated to my daughters, Miram and Mirnan.

It is a pleasure to thank many people who have helped me to finish this degree. I thank my wife, Nerham, for supporting me through these challenging years. I thank my parents, Adel Elag and Ibteessam El-shanhoury, for instilling within me the value of work and for being my source of inspiration. I sincerely thank my sister, Marwa, for encouraging me to pursue my academic dreams. I thank my friends for their encouragement throughout my years of study.

I am indebted to my advisor, Dr. Jonathan Goodall, for his guidance, encouragement, and dedication throughout my graduate work. I feel fortunate to have had the opportunity to work with him and will be forever grateful for his time, academic standards, and commitment to his students. I would also like to thank Drs. Hanif Chaudhry, Michael Meadows, and Michael Huhns for taking time to serve on my dissertation committee. I would like in particular to extend my gratitude to Dr. Huhns for his helpful guidance and comments on my work. Finally, I wish to thank Dr. Anthony Castronova for his contribution as a coauthor on third chapter of this dissertation

I would like to acknowledge the National Science Foundation (NSF) for supporting this research under the award CBET: 08-46244 “CAREER: Integrated Modeling for Watershed Management.”

## ABSTRACT

Development of integrated hydrologic models requires coupling of multidisciplinary, independent models and collaboration among different scientific communities. Component-based modeling provides an approach for the integration of models from different disciplines. A key advantage of component-based modeling is that it allows components to be created, tested, reused, extended, and maintained by a large group of model developers and end users. One significant challenge that must be addressed in creating an integrated hydrologic model using a component-based approach is enhancing the interoperability of components among different modeling communities and frameworks. The major goal of this work was to advance the integration of water related model components coming from different disciplines using the information underlying these models. This was achieved through addressing three specific research objectives.

The first objective was to investigate the ability of component-based architecture to simulate feedback loops between hydrologic model components that share a boundary condition, and how data is transferred between temporally misaligned model components. The outcome of this work provides evidence that component-based modeling can be used to simulate complicated feedback loop between systems and offers guidance as to how different interpolation schemes minimize mass balance error introduced when components are temporally misaligned.

The second objective was to promote the interoperability of components across water-related disciplinary boundaries and modeling frameworks by establishing an ontology for components' metadata. The purpose of the ontology is to assist in im-

proving communication within and beyond the hydrologic community, to ensure a common semantic understanding of concepts in component-based hydrologic modeling, and to provide a reliable and convenient tool for metadata processing. To address this need, an ontology was proposed for water resources model components that describes core concepts and relationships expressed with the Web Ontology Language (OWL).

The third study objective was to develop a domain-level ontology for defining hydrologic processes. It was motivated by the need to minimize the semantic and syntactic heterogeneity among related hydrologic disciplines. This goal was accomplished by analyzing the domain information about hydrologic processes and structuring this information in a unified formal definition. This work provides a knowledge-based ontology that defines hydrologic domain processes, SPARQL service (endpoint) to retrieve knowledge about a hydrologic process in a semantic-based approach, and a tool for presenting a process knowledge in a declarative modeling approach.

The research described in this dissertation advances the integration of component-based models from various water-related disciplines and enhances the collaboration among different scientific communities, using the information underlying these models. This was accomplished through defining the data exchange paradigm between coupled components that have shared boundary conditions and are temporally misaligned, developing the Water Resources Component (WRC) ontology to define the metadata concepts used in describing a model component, and expanding the WRC ontology by creating a domain-level ontology for describing the hydrologic processes. The results of this research provided a potential solution for component model interoperability across disciplinary boundaries and frameworks that can be applied within and across Earth Science disciplines.

# TABLE OF CONTENTS

ACKNOWLEDGMENTS . . . . .	iii
ABSTRACT . . . . .	iv
LIST OF TABLES . . . . .	viii
LIST OF FIGURES . . . . .	ix
CHAPTER 1 INTRODUCTION . . . . .	1
CHAPTER 2 BACKGROUND . . . . .	5
2.1 Model Integration Approaches . . . . .	5
2.2 Component-Based Architectures . . . . .	6
2.3 Community Modeling in Hydrology . . . . .	11
2.4 Hydrologic Information Representation . . . . .	14
CHAPTER 3 FEEDBACK LOOPS AND TEMPORAL MISALIGNMENT IN COMPONENT-BASED HYDROLOGIC MODELING . . . . .	21
3.1 Abstract . . . . .	21
3.2 Introduction . . . . .	22
3.3 Methods . . . . .	29
3.4 Results and Discussion . . . . .	40
3.5 Summary and Conclusion . . . . .	50
CHAPTER 4 AN ONTOLOGY FOR COMPONENT-BASED MODELS OF WA- TER RESOURCE SYSTEMS . . . . .	55
4.1 Abstract . . . . .	55

4.2	Introduction . . . . .	56
4.3	Methodology . . . . .	61
4.4	Results and Discussion . . . . .	72
4.5	Summary . . . . .	88
CHAPTER 5	TOWARD A HYDROLOGIC PROCESS KNOWLEDGE-BASE: A CASE STUDY FOR THE INFILTRATION PROCESS . . . . .	91
5.1	Abstract . . . . .	91
5.2	Introduction . . . . .	92
5.3	Background . . . . .	95
5.4	Methodology . . . . .	99
5.5	Results and Discussion . . . . .	108
5.6	Summary and Future Work . . . . .	119
CHAPTER 6	SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS . . . . .	122
REFERENCES	. . . . .	127
APPENDIX A	PERMISSION TO REPRINT CHAPTER 3 . . . . .	141



## LIST OF TABLES

Table 3.1	Model parameters for water and sediment media . . . . .	36
Table 3.2	Properties included in the XML configuration files for the water and sediment components. . . . .	36
Table 3.3	The time horizon and time steps used in the simulation. . . . .	38
Table 4.1	Basic concepts and properties used to describe models in Earth System Curator (ESC) (Dunlap et al., 2008) . . . . .	64
Table 4.2	Basic concepts and properties used to describe models in the Community Surface Dynamics Modeling System (CSDMS) (Peckham et al., 2012) . . . . .	65
Table 5.1	Infiltration process analysis using FCA and three resources: Maidment et al. (1992), Hornberger et al. (1998), and Ameri- can Society of Civil Engineers (1996). The ✓ indicates that a resource has a definition for this method. . . . .	103
Table 5.2	Excerpts from the matrix analysis of the local process represen- tation coded in SWAT and RHESSys versus the standardized process representation included in Table 5.1 . . . . .	104

## LIST OF FIGURES

Figure 2.1	Component package adapted from D’Souza and Wills (1998). . .	8
Figure 2.2	Three generations of community modeling projects describing the motivations of each generation adapted from Voinov et al. (2010). . . . .	13
Figure 3.1	Contrasting the more common modular approach for structuring model codes with a component-based approach. . . . .	25
Figure 3.2	Data exchange between component models in a uni-directional composition (top) and a bi-directional composition (bottom) adapted from Tindall (2005) . . . . .	27
Figure 3.3	The water and sediment domains used in the component-based approach. <i>BC1</i> is the water component lower boundary values that act as the sediment component top boundary, <i>BC2</i> is the sediment component top boundary values that acts as the water component lower boundary. . . . .	32
Figure 3.4	Details for the implementation of the component-based model approach. . . . .	35
Figure 3.5	Provided interpolation routine in OpenMI which allows for linear interpolation (when $\alpha = 0$ ), nearest neighbor (when $\alpha = 1$ ), and a weighted approach between these two alternatives (when $0 < \alpha < 1$ ) . . . . .	38
Figure 3.6	Results of the coupled water/sediment system for both the conventional and component-based approach. . . . .	42
Figure 3.7	Communication flow for a bidirectional link within OpenMI between the water and sediment components adapted from Sindling et al. (2005). . . . .	43
Figure 3.8	Communication flow for rescaling data transfers when components inputs and outputs are temporally misaligned. . . . .	46
Figure 3.9	Results of linear, quadratic, and cubic spline schemes in minimizing the mass balance error between temporally mis-	

	aligned coupled components. Comparison is for three locations in the sediment top layer ( $x = 4, 6,$ and $10$ cm) and for three values of internal time step differences ( $2, 4,$ and $10$ s). . . . .	49
Figure 3.10	Total mass error in sediment domain between the aligned component configuration and the misaligned component configurations coupled using a linear, quadratic, and cubic spline scheme for three values of internal time step differences ( $2, 4,$ and $10$ s). . . . .	51
Figure 4.1	Methodology used for development of the Water Resources Component (WRC) ontology, adopted from Uschold and Gruninger (1996) . . . . .	62
Figure 4.2	The hierarchical structure of the Semantic Web for Earth and Environmental Terminology (SWEET) Unit ontology, which integrates with the Water Resources Component (WRC) ontology to describe symbols units. . . . .	71
Figure 4.3	An overview of the Water Resources Component (WRC) ontology, describing the basic four layers and the superclasses of each layer. . . . .	73
Figure 4.4	Resource layer of the Water Resources Component (WRC) ontology, including classes and relationships. . . . .	75
Figure 4.5	Coupling layer of the Water Resources Component (WRC) ontology, including classes and relationships. . . . .	77
Figure 4.6	Scientific layer of the Water Resources Component (WRC) ontology, including classes and relationships. . . . .	79
Figure 4.7	(a) Excerpt from the WRC ontology illustrating the 3-ary relationship between a component, equation, and variable. (b) Description of the Porewater Velocity variable, which is passed between the two coupled model components, using the WRC ontology. . . . .	81
Figure 4.8	Technical layer of the Water Resources Component (WRC) ontology, including classes and relationships. . . . .	82
Figure 4.9	Excerpt from the WRC ontology illustrating the stored information about three evapotranspiration interchangeable components. This information can help users in the process of selecting the appropriate component to represent the evapotranspiration process for a given study. . . . .	85

Figure 4.10	Ontological representation of the relationship between two variables, the Potential Evapotranspiration (PET) rate and the standardized reference Evapotranspiration ( $ET_{sz}$ ), in the Water Resources Component (WRC) ontology, and the conversion factor between them. . . . .	87
Figure 5.1	A schematic representation and examples of knowledge representation paradigms ordered by increasing expressivity and complexity, adopted from Villa et al. (2009). . . . .	98
Figure 5.2	Methodology used for development of the Hydrologic Process (HP) Ontology, adopted from Uschold and Gruninger (1996). . . .	101
Figure 5.3	Hydrologic Process (HP) ontology including classes and relationships.	109
Figure 5.4	Excerpt from the Hydrologic Process (HP) ontology illustrating the stored information for the Green-Ampt method. . . . .	111
Figure 5.5	Excerpt from the Hydrologic Process (HP) ontology illustrating the stored information for the Philip's method. . . . .	113
Figure 5.6	Communication between a client and server via SPARQL endpoint that is hosted in Dydra computational cloud. . . . .	115
Figure 5.7	Pseudo-code for Philip's method. . . . .	118

# CHAPTER 1

## INTRODUCTION

Hydrologic modeling is considered a central process for solving complex problems in many Earth Science disciplines (Hornberger et al., 2012; Maidment et al., 1992). Hydrologic fluxes interconnect water, climate, energy, and ecological cycles (Bloschl and Sivapalan, 1995). As a result, hydrologic models must cross domain boundaries and consider the feedback of processes happening in other domains, as well as introduce an organized system for collaboration among scientists and engineers from different disciplines to face present and future challenges (Hornberger et al., 2012; Syvitski et al., 2011; Voinov et al., 2010). However, most hydrologic models are designed, built, and controlled by a specific scientific or engineering community to answer a specific problem within the hydrologic cycle (Singh and Woolhiser, 2002; Wagener et al., 2010). These models treat a portion of the hydrologic cycle as an independent system without considering its interactions with other systems. This approach is unacceptable because of the growing need to develop comprehensive models that are able to consider the interactions between dynamically connected systems.

Multidisciplinary modeling requires the integration of data, techniques, tools, perspectives, concepts, and theories between related disciplines (Hornberger et al., 2012). Given the multidisciplinary, complex nature of water resource problems, it seems neither achievable nor desirable to aim for a single central computational “super-model” that attempts to represent all the processes in different domains. In fact, no single model will ever be capable of (i) incorporating all of the different processes used in various scientific disciplines (Hornberger et al., 2012; Argent, 2004), (ii) representing

the wide variance in spatial and temporal resolutions of the hydrologic fluxes inter-connecting water, climate, energy, and ecological cycles (Loucks et al., 2005; Blöschl and Sivapalan, 1995; Letcher et al., 2004), and (iii) managing data used by the models through sophisticated information systems and data management approaches (Dozier, 1992; Goodall and Maidment, 2009; Horsburgh et al., 2008).

In the last decade, component-based modeling has been introduced as a paradigm that exploits advances in computer and Internet technologies to decentralize a model’s functionality (Argent, 2004; Goodall et al., 2011; Peckham et al., 2012; Castronova et al., 2012). The component-based approach maintains the balance between a physical process representation and its complexity (Armstrong et al., 2002; Argent, 2004; Moore and Tindall, 2005; Syvitski et al., 2011; Gössler and Sifakis, 2003). A key distinguishing feature of component-based modeling is decentralizing the model functionality into independent component models, which are capable of being integrated with other models in what is described as a “plug-and-play” manner (Peckham et al., 2012). This decentralized approach achieves model integration by satisfying two basic requirements: standardizing the communication interface between modeling components during a simulation run (Szyperski, 2002) and providing a framework whereby a system can be assembled out of a set of independent component models (Argent, 2004). By decentralizing the model functionality into independent components, it (i) gives a relative freedom from the assumptions controlling the more conventional centralized approach, (ii) decreases the development effort of models by avoiding recreating the wheel, (iii) raises the quality of the developed model because developers have more time to focus on enhancing models rather than incorporating new models, and (iv) decreases model maintenance cost.

While there are obvious benefits to using a component-based approach for hydrologic modeling, there are also challenges that must be overcome in order to encourage broad adoption of the approach. Three studies were performed and the results are

presented in this dissertation to advance the understanding of model coupling and increase the interoperability among disciplinary water-related models.

The first study investigated two of the more pressing questions about applying component-based modeling concepts to hydrologic systems: (i) how to simulate fully coupled processes with a shared boundary condition as two distinct components with a bidirectional link and (ii) how to couple two model components that have spatially and/or temporally misaligned inputs and outputs. In this study, first the component-based approach's ability to handle the complicated hydrologic feedbacks within a watershed system was investigated. Second its ability to address coupled components that are operating on independent spatial or temporal discretizations, without violating basic principles of hydrologic modeling (e.g., conservation of mass, energy, momentum) was investigated. To address this research goal, a hypothetical modeling exercise was conducted with the objective of estimating pollution concentration in a multimedia water/sediment system.

The second study aimed to answer another important challenge associated with component integration: model integration is not simply the proper assemblage of components from a technical standpoint, rather it also requires scientific knowledge of the underlying coupling between each component (Athanasiadis et al., 2006; Voinov and Shugart, 2012; Elag et al., 2011). Scientists wanting to couple components across disciplinary boundaries or modeling frameworks face challenges including (i) semantic heterogeneity across disciplines due to the variety of terminology used to describe the equations, variables, parameters, and units within models (Argent, 2004; Voinov and Shugart, 2012; Peckham et al., 2012), (ii) diversity of concepts used to define components' functionality and relationships, which results in overwhelming complexity for linked model compositions (Argent, 2004; Voinov and Shugart, 2012), and (iii) coupling inconsistency resulting from mismatched spatial or temporal data exchanges, or from incompatible semantics used by different models (Voinov and Shugart, 2012;

Elag et al., 2011). To overcome these challenges an ontology for modeling components was created and named Water Resource Component (WRC) ontology. This ontology specifies the concepts and terminologies related to model components used in water-related disciplines.

The third and final study was motivated by the need to structure hydrologic domain information regarding physical processes in a machine understandable format to increase the usability of process definitions across related scientific modeling communities. Because of this, there is a need for a standardized and a robust method to organize and define the domain processes and equations that can be discovered and used across disciplinary boundaries. This work focused on introducing a formal definition for hydrologic domain processes, their interrelationships, and their corresponding equations, by creating a domain-level ontology. This ontology provides a unified understanding among domain users about hydrologic modeling processes, promotes the interoperability of components across the water-related disciplines by integrating with the WRC ontology, facilitates the process of component creation by guiding the user to scientific information and tested codes, and supports the automation of creating a computer code for a hydrologic process.

Following this introduction is a background section that provides an overview of key concepts that underlie this research: model integration approaches, component-based architectures, community modeling in hydrology, and hydrologic information representation. The three studies described earlier are then presented in manuscript form as individual chapters following the conventions outlined by the University of South Carolina Graduate School in the Electronic Thesis and Dissertation Formatting Guide. Because these manuscripts are coauthored, when first person is used in the manuscripts, the plural rather than the singular form is used.



## CHAPTER 2

### BACKGROUND

The following section provides background for four concepts related to the dissertation subject matter. The first section discusses the different model integration approaches and the second section provides a brief overview of community modeling activities in the Earth Sciences. The third section provides an overview of different hydrologic information representation techniques. Finally, the fourth section introduces the concept of the Semantic Web as it relates to this research.

#### 2.1 MODEL INTEGRATION APPROACHES

Computer modeling has become an essential tool in water resources research because it provides a simplified representation of reality that can be used to predict future events and improve understanding of interdisciplinary cause-and-effect relationships (Argent, 2004). Models vary greatly in their complexity and purpose from spreadsheets to Earth-scale prediction models. Most models aim to represent a finite set of physical processes of a single domain accurately (Singh, 2006). One of the coming challenges facing the modeling community will be to simulate complex and multidisciplinary systems. Monodisciplinary models can complement each other through integration, thereby providing a useful tool for solving complex water resource problems (Wagner et al., 2010). A more holistic, accurate, and flexible approach is required to combine these single, well calibrated models into a comprehensive model.

Integration of models from different disciplines to support watershed management has become a high priority (Argent, 2004). Model integration requires two major complementary elements: software resources and communities integration (Voinov et al., 2008). First, tools and technologies from the computer science and software engineering disciplines must be transferred to be applied in water resources modeling. Second, interest in community modeling, which has emerged as a promising paradigm to develop complex modeling systems, must continue to grow. There are different technical approaches for integration of monodisciplinary models. The most straightforward method is to develop an entirely new model that encompasses multiple models, but this is time consuming and does not leverage prior community investments. A better approach is to combine existing models using a model coupling approach.

The term model coupling is widely used, although there is no common agreement upon its definition. Knapen et al. (2012) define five different approaches of model coupling: (i) manually linking of a model by using the output of one model as an input for the other, (ii) automatically linking by writing codes for converting the output of a model into input for another model, (iii) encapsulating, where one model encapsulates the other into a monolithic model (e.g. MODFLOW, <http://www.modflow.com/>), (iv) loosely coupling using a linking mechanism (e.g. service oriented modeling mechanism), and (v) loosely coupling in a modeling framework (e.g. Community Surface Dynamics Modeling System). In the loose coupling approach which is the subject of this dissertation, models are independent and exchange data through a standardized method.

## 2.2 COMPONENT-BASED ARCHITECTURES

Component-based modeling is a key principle in advancing model integration using a loose coupling approach (Syvitski et al., 2004; Leavesley et al., 1996; Argent, 2004).

This is because simpler components are easier to test, debug, update, compare, and verify, and once the components are created, they can be (re)assembled in different ways (Argent, 2004). In this approach, each component has a definite role and the process of its development and maintenance is assigned to a specific group. The next sections provide background about components as a model, components in a modeling framework, and OpenMI as a standard for data exchange in component-based modeling.

### 2.2.1 COMPONENT

Components in software engineering are defined as independent functional units that can communicate with each other (Argent, 2004; Szyperski, 2002). Shaw and Garlan (1996) provided an overview of different component functions and classify components based on their function into five categories: (i) computes pure mathematical operations, (ii) links and transfers information between entities, (iii) controls and governs time sequence operations, (iv) manages states and organizes operations, and (v) shares memory collection of persistent structured data.

D’Souza and Wills (1998) define a component as a package that should include (i) an executable code in a binary format, (ii) a list of the I/O of the component, (iii) an interface for users to communicate with the component and another interface for components to exchange data with each other, (iv) a validation code that ensures a proper connection with other components, and (v) design documentation that includes source code and all of the documents associated with the component (Figure 2.1).

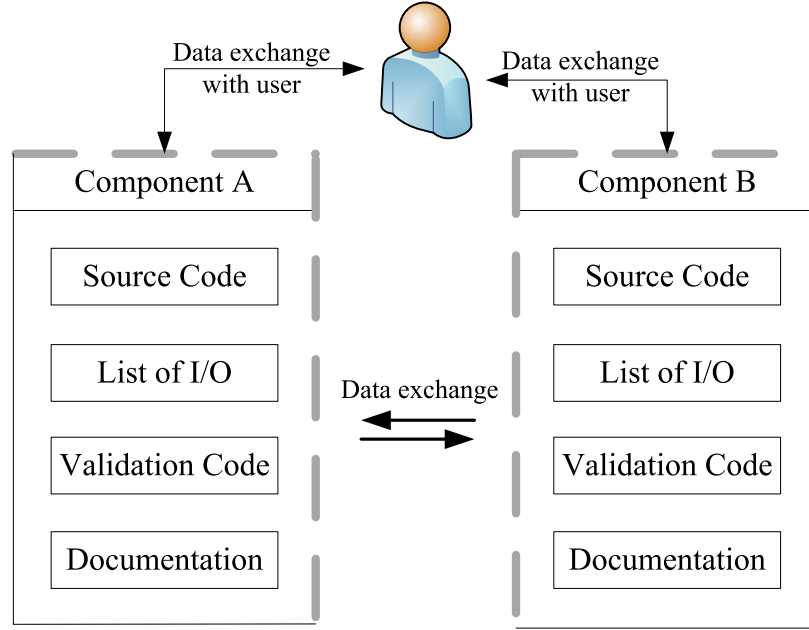


Figure 2.1: Component package adapted from D’Souza and Wills (1998).

### 2.2.2 MODELING FRAMEWORK

A modeling framework is a run-time environment for coupling component models with the intention of standardizing data manipulation and analysis, exchanges between models and datasets, and visualization of model outputs (Argent et al., 2006). The term framework typically refers to the underlying modeling classes and libraries that are used to support module development, model construction, and model execution (Argent et al., 2006). A modeling community that uses the framework defines two methods for components to be connected in the framework. First, frameworks must have standardized argument data types for the component (e.g. Earth System Modeling Framework, <http://www.earthsystemmodeling.org/>). Second, frameworks require put/get calls within the component for data I/O (e.g. Modular Modeling System, <http://pubs.usgs.gov/sir/2006/5041/section3.html>). The Earth System Modeling Framework (ESMF) (Hill et al., 2004), the Community Surface Dynamics Modeling System (CSDMS) (Syvitski et al., 2011), and the HydroModeler (Castronova

and Goodall, 2010) are a few examples of component modeling approaches being developed within the Earth Science modeling community.

The Earth System Modeling Framework (ESMF, <http://www.earthsystemmodeling.org/>) is a system that provides a high-performance yet flexible software infrastructure for code reuse in climate, numerical weather prediction, data assimilation and other Earth Science applications (Hill et al., 2004). ESMF is based on principles of component-based modeling. It enables the integration of models that require high performance computing, such as weather and climate models (Hill et al., 2004). The ESMF has two major components: coupling superstructure and utility infrastructure. The superstructure standardizes the communication between components during a simulation. The infrastructure contains the development libraries used for model component development (Hill et al., 2004).

The Community Surface Dynamics Modeling System (CSDMS, <http://csdms.colorado.edu>) focuses on simulating sediment and material transport over various time and space resolutions (Syvitski et al., 2004). CSDMS has three principal components: Standard Utilities, Modules, and a Toolkit (Syvitski et al., 2004). Standard Utilities handle data structures, graphics rendering, module connectors, and a web interface (Anderson et al., 2004). The CSDMS uses a variation of the Open Modeling Interface (OpenMI) standard to maintain runtime communication, and the Common Component Architecture (CCA) to facilitate high performance computing (Syvitski et al., 2004).

The HydroModeler is a HydroDesktop plug-in application developed under the Consortium of Universities for the Advancement of the Hydrologic Sciences Inc. (CUAHSI, <http://www.cuahsi.org>) and its Hydrologic Information System (HIS, <http://www.his.cuahsi.org>) project that extends the core HydroDesktop application to support model integration using the component-based approach (Castronova and

Goodall, 2010). It uses the Open Modeling Interface (OpenMI) standard and the OpenMI Association Technical Committee (OATC) Software Development Kit (SDK) to provide a “plug-and-play” modeling framework within HydroDesktop.

### 2.2.3 OPEN MODELING INTERFACE

An interface is a formal definition of the functions to be used by models when coupled so that they can interact with the run-time environment (Swayne et al., 2010). The OpenMI (<http://www.openmi.org>) is one such interface standard used for data and model communication in the water resources domain. It enables models to “talk” to each other (Voinov et al., 2008) using a standard language. It was developed to support water resources management under the EU 5<sup>th</sup> framework program project HarmonIT. The main objective of the HarmonIT project was to provide a widely accepted and unified method to link models, both legacy and new models, by providing a Standard Development Kit (SDK) (Moore and Tindall, 2005; Sinding et al., 2005). The SDK enables developers to “wrap” their model into an OpenMI compliant form, which can then be used in any OpenMI modeling framework.

In this dissertation, I used the OpenMI (version 1.4) that provides standardized interfaces to define, describe, and transfer data between software components that run sequentially, based on a pipes and filters architecture (Sinding et al., 2005). The data exchange definition addresses the “five W questions” (why, how, where, when, why). Each component that implements the OpenMI standard is considered to be a linkable component and has metadata describing the exchangeable data. Components are connected using links, which exist as a separate entity. Each link transforms an output of one component into an input of the receivable component. The OpenMI Association (OA) keeps OpenMI under constant development to promote the development, use, management, and maintenance of the standard (Sinding et al., 2005).

A new version of OpenMI (version 2) was released in 2010 to satisfy the new requirements of integration between water resources modeling and related domains, as well as the need to employ the OpenMI in other integration techniques (e.g. Service Oriented Modeling (Goodall et al., 2011)). This version includes some innovations such as (i) improved geographical representations based on GIS concepts for data exchange between models, (ii) new interpolation adapters for spatially and temporally transforming data into a requested form, (iii) removing the restriction to only time step-based models, (iv) setting and varying boundary conditions for individual models for running comparative simulations, and (v) introducing of set of mandatory base interfaces and sets of optional interfaces to make the OpenMI better able to accommodate future requirements of integrated modelling (Knapen et al., 2012).

The OpenMI is currently implemented in C#.Net and Java, and offers ways for wrapping code from other languages. Knapen et al. (2012) recommended the OpenMI as a standard interface for coupling several multidisciplinary models because it (i) requires few modifications to the models in order to implement the standards, (ii) advances rapidly because it is on an open standard, which is developed by a multi-institutional organization and benefits from contributions from the larger research community, (iii) operates over a network and is able to handle both small and large models, (iv) separates interfaces from implementation (increases software maintainability) and has the ability to “wrap” existing models, and (v) is well documented.

## 2.3 COMMUNITY MODELING IN HYDROLOGY

The community modeling approach has been developed to manage the challenges of creating and applying complex models. It is based on an Open Source (OS) philosophy and the need for collaboration among between community members throughout different modeling stages. This enables the community to share and reuse available

data, models, and tools. A primary goal key of community modeling is to increase the involvement of project-related users during the modeling process. This approach results in a comprehensive modeling system. For example, the atmospheric community introduced some of the best examples of this such as Community Earth System Model.

A major motivation for using the community modeling approach in hydrologic modeling is to enable linking of models from different disciplines and research groups. This is expected to result in a decrease in redundancy because it helps developers to evolve new models based on the available models, codes, and algorithms. Also, community modeling approaches increase community engagement, which raises the project potential.

Community modeling has evolved through three different generations (Voinov et al., 2010) (Figure 2.2). The first generation focused on the importance of Open Source (OS) models and how to use them to model scientific problems. The second generation emphasized how to connect the contributors to address the different demands of the community, such as the ability to integrate with new sciences and create clear borders for defining the scientific process. Recently, most attention has been given to the integration of models across scientific disciplinary boundaries using modeling frameworks. The community modeling approach requires a framework to enable the community individuals (developers and users) to build and couple their models (Voinov et al., 2010; Famiglietti et al., 2008). In this context, the component-based modeling approach plays an essential role of introducing a scientific process as a model that can be coupled in a framework.

In the Earth Sciences, there are several ongoing community modeling efforts that have grown in recent years. These communities are seeking to create and organize their own modeling systems and integrate with other modeling systems. For example, the CSDMS has built a structure for community input and involvement



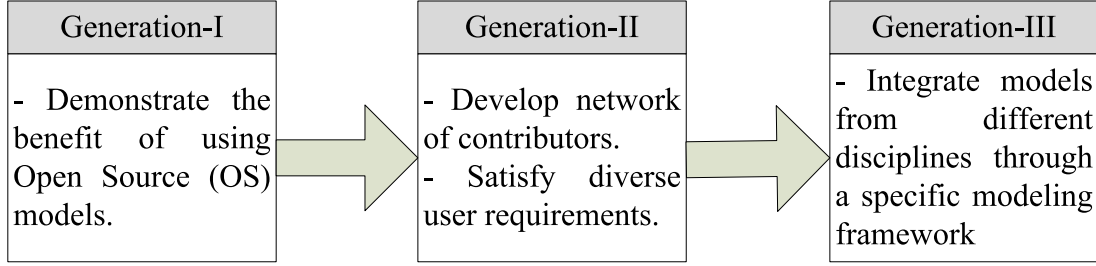


Figure 2.2: Three generations of community modeling projects describing the motivations of each generation adapted from Voinov et al. (2010).

through its main working groups and focus research groups. It identifies standards for model developers and for model coupling to enhance access to the CSDMS by developers and users (Syvitski et al., 2004). The Earth System Modelling Framework (ESMF, <http://www.esmf.ucar.edu>) is a multi-institutional, multidisciplinary effort between Earth Science modeling centers within the United States including the National Weather Service and NASA (Hill et al., 2004). The Community Climate System Model (CCSM, <http://www.cesm.ucar.edu/>) is developed and used by the climate community of scientists and students from universities, national laboratories, and other institutions. CCSM is a general circulation climate model consisting of four components: atmosphere, land, ocean, and sea ice. These components are linked through couplers into a system (Gent et al., 2011). The CCSM provides a state-of-the-art computer simulation of the Earth’s past, present, and future climate states. The latest release of the CCSM includes a new ESMF compliant component interface which allows CCSM model components to be coupled with other systems using ESMF coupling standards (Craig et al., 2012).

In Hydrology, the Consortium of Universities for the Advancement of Hydrologic Science (CUAHSI, <http://www.cuahsi.org>) encourages the development of a Community Hydrologic Modeling Platform (CHyMP) (Famiglietti et al., 2008). One of the goals of CHyMP is to collect all of the model components in hydrology and related fields to enable comprehensive simulation of water anywhere in the North American

continent. This goal requires the interoperability of models across disciplinary boundaries. The role of CHyMP is to provide infrastructure for community-driven model development, model distribution, high performance computing access, and technical support (Famiglietti et al., 2008). This will help to connect the hydrologic modeling community with several ongoing community modeling activities in related disciplines (e.g. ESMF, CSDMS, CCSM).

Creating a hydrologic community modeling system is complicated because hydrologic models are fragmented across different disciplines such as the atmospheric, agricultural, and ecological, sciences. Models are created by different organizations, research groups, and commercial companies, which results in a diversity of model structures, formats, and operation requirements. These result in a lack of connectivity, insufficient understanding, and a lack of motivation for sharing the working environment (Famiglietti et al., 2008; Voinov et al., 2010). These shortcomings are due to (i) semantic heterogeneity across disciplines due to the variety of terminology used to describe the equations, variables, parameters, and units within models (Argent, 2004; Voinov and Shugart, 2012; Peckham et al., 2012) and (ii) the diversity of concepts used to define a model’s functionality and its relationships, which results in overwhelming complexity for linked model compositions (Argent, 2004; Voinov and Shugart, 2012).

## 2.4 HYDROLOGIC INFORMATION REPRESENTATION

Hydrologic science and management often requires both observations and models to solve complex water resource problems. Many scientific communities have realized the importance of managing the information that describes both data and models to increase interoperability between them. This is especially true given the significant increase in observational data produced by the science and engineering communities

(Horsburgh et al., 2009; Cox, 2006). In addition, the growth in models contributed to a modeling library requires organization and standardization of information describing these models (Argent et al., 2006; Goodall et al., 2011; Bermudez and Piasecki, 2006). This prompts the need to develop an information system that unifies the description of objects (data or models) that permits users and computers to better understand model properties (or metadata).

#### 2.4.1 METADATA

The word “meta” is typically used as a prefix to indicate a higher level of description for the suffix term. Therefore the term metadata is used to describe “data about data.” Over the past few decades, the term metadata has emerged as an important concept used to summarize information about an object. Metadata describes an object and depicts its behavior, function, and relationship with other objects (Kumar, 2006). It is needed to facilitate data sharing and enhance the information retrieval process. The use of metadata to describe the content of an object has become one of the focus areas for many research communities (including the hydrologic science community) (Bermudez and Piasecki, 2006; Horsburgh et al., 2008; Syvitski et al., 2011).

Metadata is required to reflect three distinct features about an object: content, context, and structure (Kumar, 2006). First, content defines the intrinsic element of the object and what the object contains. Second, the context element provides the answer for the “five W questions” (who, what, where, why, and how), which are used to define the extrinsic element of the object (Kumar, 2006). In the hydrologic modeling community, for example, these might be: developer, evaporation model, source code location, purpose, and tutorial. Finally, structure defines the metadata syntax (e.g. folder contents list). Scientific communities tend to create their own metadata to satisfy their specific needs, but have not considered interoperability with other communities (Bermudez and Piasecki, 2006).

Integration of the metadata underlying each component model is important to the proper technical assemblage of components to achieve model integration (Athanasiadis et al., 2006; Voinov and Shugart, 2012). In other words, if a set of component models are coupled and working together, this does not guarantee a proper component assemblage from a scientific point of view. Including a rich set of metadata with modeling components is necessary to ensure a proper assemblage of components. Incorporating information with components enables users to overcome the semantic heterogeneity across related disciplines and unifies the concepts used to define components' functionality and relationships (Argent, 2004; Voinov and Shugart, 2012; Peckham et al., 2012). The basic notion of metadata integration is that several metadata come together to make the data describing an object clear for users coming from different disciplines.

Clearly, agreement on techniques to transfer information about models between source and user must consider the Internet as an efficient mean of communication. The Internet provides a dynamic environment for data exchange and retrieval. It is used for information exchange from machine to human and machine to machine (Huhns and Singh, 2005). Recently, advancement in information transfer techniques accelerates the rate of information transfer. The World Wide Web Consortium (W3C, <http://www.w3.org>) develops, accepts, and maintains data interoperable technologies to serve data transfer over the Internet. W3C defines the Semantic Web service as a mechanism to integrate semantic information from different resources. The technology that will be used to describe the component model metadata must have a clear structure that is extendable and easily deployed in different modeling communities.

Various methods have been devised to support information organization and interchange. Zeng (2005) defines three groups of information organization: term lists, classification/categorization, and relationships. Term lists provide a standardized dictionary of terms for use during information indexing (e.g. controlled vocabulary

list). A List of terms does not provide much structure, which makes information retrieve more difficult. Classification/Categorization sorts information into groups with similar attributes (e.g. units). It enhances the information retrieval, however building relationships among resources is necessary as the information becomes more extensive. Relationship groups emphasize the connections between terms and concepts. The presence of relationships among structured information enhances the information retrieval process.

In the coming section I present an overview of ontologies and the Semantic Web, as well as the relationship between them. The emphasis is on the technologies employed in this study to represent the water resources component and hydrologic processes metadata.

#### 2.4.2 ONTOLOGY

The term ontology has roots that lie in philosophy where it is used to represent a systematic account of existence. The term migrated to the realm of artificial intelligence when it was introduced by Gruber (1993). An ontology is an explicit conceptualization of human knowledge that focuses on shared understanding by defining vocabularies that represent and communicate knowledge about a specific domain (Gruber, 1993; Garshol, 2004; Antoniou and Harmelen, 2009). In other words, ontologies express specific vocabularies that are used to describe facts about a domain. Ontology introduces the conceptualization that underlie the knowledge and the vocabularies that represent this knowledge. This effective information representation system offers shared understanding of the domain by users from different disciplines.

Ontologies are often used to facilitate resource representations for the purpose of integrating systems (Uschold and Gruninger, 1996). An ontology usually introduces concepts such as entities and the relationships between them. An Ontology acts as an agreement to use the specified vocabulary based on the concepts and relationships.

Thus, a major motivation for using an ontology in the water resource domain is to integrate the metadata describing models of different modeling communities into a coherent framework. An ontology provides a unified structure for information representation, which minimizes the conceptual and terminological confusion of a specific domain.

Jurisica et al. (1999) classified the ontological representation of knowledge into four categories: static, dynamic, intentional, and social ontologies. A static ontology enumerates the physical objects in a problem definition and identifies their inheritance properties and relationships. A dynamic ontology defines the changing features of the domain (e.g. a modeling algorithm). An intentional ontology codes the domain motivations, goals, beliefs, alternatives, and choices (Jurisica et al., 1999). Lastly, social ontologies cover the organizational structures (e.g. organization chart of research institute) (Jurisica et al., 1999). Typically most of the knowledge representation in the water resource domain fluctuates between the static and dynamic categories. In this way ontologies are used to improve communication among humans because it provides a shared understanding between people with different needs and viewpoints. Also, if it is coded in a formal language that a machine can understand, then it improves the communication between human and machine. In addition, ontologies increase the quality of the software system capabilities (e.g. search, reliability, etc.) Finally, an ontology is considered the backbone of the Semantic Web, as described in the following section.

#### 2.4.3 SEMANTIC WEB

The Semantic Web was introduced by the W3C to develop a method and technologies for information integration, processing, and querying on the Web (Berners-Lee and Fischetti, 2001; Fensel et al., 2011). The W3C tries though the Semantic Web to create something closer to a global database compared to the existing World Wide

Web (Fensel et al., 2011). A key concept of the Semantic Web is to enable transferring of structured well-formed data between users on the Internet. This enables a user to receive information about his search from different data sources relevant to his query (Abadi et al., 2007). In fact, most hydrologic models have metadata that are available on the Web, but these metadata suffer from semantic and syntactic heterogeneity. The Semantic Web overcomes this limitation by introducing a well defined information structure that enables both the human and machine to understand the information meaning.

The Semantic Web community introduces different choices for representing data on the Web. The Semantic Web architecture for information coding consists of different coding blocks. The information coding blocks are organized in the so-called Semantic Web Layer Cake in an ascending order based on the complexity as follow:

- UNICODE is an ISO standard used to define the international character set. It contains a definitions for all letters and signs used in human languages and it has a clearly defined process for adding new characters.
- Unified Resources Identifier (URI) is a standard of the Semantic Web used to define resources on the Web.
- eXtensible Markup Language (XML) was developed by the W3C for data exchange over the Web (Bray et al., 1997). It is a flexible, self-describing markup language format. It provides a hierarchical syntax structure for encoding data and their relationships, however the approach can suffer from syntactic heterogeneity because data can be organized in many different ways.
- Resource Description Framework (RDF) is a simple yet powerful data model and language for describing Web resources. RDF represents data as statements using a graph connecting resource nodes and their property values with labeled arcs representing properties. This graphical relationship can be represented

using the XML syntax (RDF/XML) (Abadi et al., 2007). The major problem of using RDF for representing metadata is that it does not provide a mechanism for defining internal relationships between properties and concepts (Garrido and Requena, 2011).

- RDF Schema (RDFS) extends the RDF approach for representing data by introducing means to model classes, properties, hierarchy of classes and properties (Fensel et al., 2011). RDFS is considered to be a light weight ontology language that can be used for creating taxonomy like ontologies (Fensel et al., 2011). RDFS provides vocabularies to describe the relationships between classes, relationships between properties, and relations between classes and properties. However, it does not provide a mechanism for defining internal relationships between properties and concepts (e.g. explicit cardinalities in properties relationships and union in classes) (Garrido and Requena, 2011).
- Ontology Web Language (OWL) is the standard language for representing knowledge on Web. It is a machine readable language that was designed to process information on the Web instead of just representing information to human users (Fensel et al., 2011). OWL can be used to express more sophisticated relationships between terms compared to RDFS. OWL can explicitly represent the meaning of terms in vocabularies and relationships. OWL is recommended by the W3C as an ontology language. It is XML-based, applies RDF syntax, and is compatible with most querying languages (Antoniou and Harmelen, 2009; Hitzler et al., 2009; Garrido and Requena, 2011).



# CHAPTER 3

## FEEDBACK LOOPS AND TEMPORAL MISALIGNMENT IN COMPONENT-BASED HYDROLOGIC MODELING <sup>1</sup>

### 3.1 ABSTRACT

In component-based modeling, a complex system is represented as a series of loosely-integrated components with defined interfaces and data exchanges that allow the components to be coupled together through shared boundary conditions. Although the component-based paradigm is commonly used in software engineering, it has only recently been applied for modeling hydrologic and earth systems. As a result, research is needed to test and verify the applicability of the approach for modeling hydrologic systems. The objective of this work was therefore to investigate two aspects of using a component-based software architecture for hydrologic modeling: (1) simulation of feedback loops between components that share a boundary condition and (2) data transfers between temporally misaligned model components. We investigated these topics using a simple case study where diffusion of mass is modeled across a water-sediment interface. We simulated the multi-media system using two model components, one for the water and one for the sediment, coupled using the Open Modeling Interface (OpenMI) standard. The results were compared with a more conventional

---

<sup>1</sup>Mostafa M. Elag, Jonathan L. Goodall, Anthony M. Castronova, Feedback loops and temporal misalignment in component-based hydrologic modeling, *Water Resources Research*, 47(12), doi:10.1029/2011WR010792, 2011.

Copyright 2011 by the American Geophysical Union. Reproduced by permission of the American Geophysical Union.

numerical approach for solving the system where the domain is represented by a single multidimensional array. Results showed that the component-based approach was able to produce the same results obtained with the more conventional numerical approach. When the two components were temporally misaligned, we explored the use of different interpolation schemes to minimize mass balance error within the coupled system. The outcome of this work provides evidence that component-based modeling can be used to simulate complicated feedback loops between systems and guidance as to how different interpolation schemes minimize mass balance error introduced when components are temporally misaligned.

## 3.2 INTRODUCTION

Watershed-scale hydrologic models are commonly used tools in water resource management. They are applied to better understand water quality conditions, surface water and groundwater allocations, floodplain management, flood warning, and a range of water resource management activities (Wurbs, 1998). Modeling hydrologic processes at the watershed-scale is challenging for many reasons including the multidisciplinary nature of watersheds where human, ecological, hydrological, and economic factors can influence how water is transported through natural and built systems. Watershed-scale modeling also introduces significant data collection and management challenges that must be addressed using sophisticated information systems and data management approaches (Dozier, 1992; Horsburgh et al., 2008; Goodall and Maidment, 2009). At an even more fundamental level, watershed-scale models must simulate hydrologic processes and flow paths that are difficult to describe and to parameterize within a simulation model (see Beven (2002) for a summary).

Despite these challenges, there are a number of examples of watershed-scale hydrologic models developed by the scientific and engineering community (Singh, 2006).

Each model has been designed to address specific requirements, yet these models include significant overlap in terms of repetitive process representations and duplication of effort. As the demands and stresses placed on water resources continues to increase, the hydrologic science community would benefit from consolidating model development effort so that a community of modelers can work collaboratively to build models capable of addressing a variety of complex water resource management challenges. We argue, along with others (Voinov et al., 2010), that a key step along this path is to understand how software engineering and information technology can be leveraged in order to allow for multi-developer modeling efforts. That is, for future watershed models to be able to keep pace with policy and decision makers' needs, there is a need to encourage community-modeling practices; and to encourage community-modeling practices, there is a need to advance the architectural approaches that underlie watershed model codes. Therefore, this work is primarily focused on the challenge of designing watershed-scale hydrologic models from a software architecture perspective.

Many examples exist of past efforts to infuse information technology into watershed modeling and management. Most of these efforts, however, have resulted in tools that target the time-intensive process of creating model input files and analyzing model output files (Srinivasan and Arnold, 1994), often through the integration of Geographic Information Systems (GIS) with simulation models (Pullar and Springer, 2000; Miller et al., 2004). While this work has resulted in important and necessary tools, it does not fully capture the potential of modern information technologies to advance how we design core architectures of watershed modeling systems. More recent efforts in the hydrologic and Earth Sciences have resulted in promising approaches for integrating both the growing set of hydrologic data and models (Goodall et al., 2008; Horsburgh et al., 2008; Maidment, 2008; Voinov et al., 2010). The focus of this work is on one of the more recent paradigms for structuring earth system models:

component-based software architectures (Armstrong et al., 2002; Gössler and Sifakis, 2003; Heineman and Councill, 2001).

Component-based software development is a method of software construction whereby a system is assembled from a set of prefabricated, reusable, and independently evolving units of code (Clements, 1995). Developing using components allows for the construction of reconfigurable software systems in a timely and more manageable manner, where components of the system can be more easily tested and validated in isolation of and in combination with other system components (Clements, 1995; Garlan et al., 1995; Szyperski, 2002). When the approach is applied for constructing modeling systems, simulations of a particular system can be constructed by coupling reusable components into a composition tailored for a specific modeling objective (Voinov et al., 2010). By adhering to component interface standards, multiple developers can contribute components to the modeling system that are able to be coupled for the simulation.

An important distinction of this work compared to previous approaches in watershed-scale hydrologic modeling frameworks is between modular and component-based approaches as the underlying architecture (Figure 3.1). In a purely modular approach, model integration is achieved by placing restrictions on data structures and conventions used within modular modeling routines. In contrast, component-based approaches achieve integration by standardizing the communication interface between modeling components during a simulation run (Syvitski et al., 2004). The result is a more flexible and extensible system because it is less restrictive on the internal implementation of components. While there are many examples of technologies that emphasize a modular modeling approach for water systems (e.g., Rahman (2003) and Leavesley et al. (1996)), there are fewer examples that use a component-based approach, primarily because it is a newer approach for model integration. The Open Modeling Interface (OpenMI) (Tindall, 2005), however, is one example of this

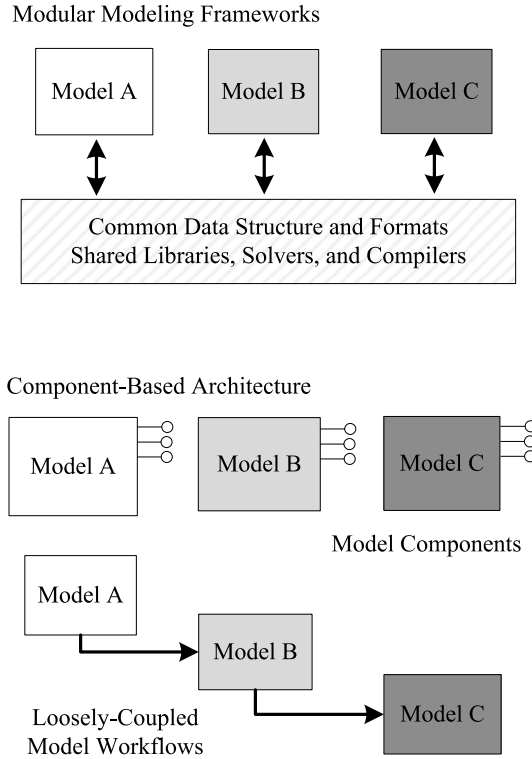


Figure 3.1: Contrasting the more common modular approach for structuring model codes with a component-based approach.

approach developed and used within the water resource community, and the Community Surface Dynamics Modeling System (CSDMS) (Syvitski et al., 2011) is a second example of this approach developed and used in the Earth surface dynamics modeling community. Here we focus on the OpenMI because it is meant primarily as a standard for component-based modeling in the water resources domain.

The Open Modeling interface (OpenMI) is a component interface standard developed through the European Union Water Framework Directive (Tindall, 2005). Its primary purpose is to facilitate interoperability between environmental models (Gregersen et al., 2007). The basic approach used by OpenMI to couple models is to allow access to input and output values of the model directly at run-time. A model that implements an OpenMI standard interface becomes a linkable component and then can be coupled to other components through input and output exchange items

(Tindall, 2005). OpenMI is a pull-based pipe and filter architecture that consists of Linkable components (source components and target components) which exchange memory-based data in a pre-defined way and in a pre-defined format. The OpenMI defines the component interfaces as well as how the data is being exchanged. OpenMI version 1.4 communication protocol consists of three fundamental concepts: a linkable component, an exchange item, and a link (Sinding et al., 2005). A link connects the source and target component, and define the actual data exchange quantity between linkable components. Data exchange in an OpenMI 1.4 component composition is initiated by a trigger component that begins the data communication process. Once the composition has been triggered, components exchange data autonomously without the need for a controller to supervise the interactions. OpenMI follows a single-threaded architecture meaning that a component can handle just one request at a time. Figure 3.2 illustrates how models can be linked by either a uni-directional or bi-directional link in an OpenMI component composition. Uni-directional links are used to integrate sequential models where model A is dependent on the output from model B, which itself is dependent on the output from model C. In bi-directional links, two components are linked such that each requires input from the other in order to run. This type of feedback loop communication is more complicated to implement in component-based modeling and therefore is a focus of this paper.

OpenMI version 2.0, which was released after the initiation of this study, removes the concept of a link being a distinct object and instead components directly provide their output exchange item(s) as input exchange item(s) to one or more components. In both the 1.4 and 2.0 versions of OpenMI, the target component pulls the data when needed using the pull-driven mechanism. We used the OpenMI 1.4 since the Software Development Kit (SDK) for OpenMI 2.0 is still incomplete. Our findings will remain applicable after migrating to OpenMI 2.0 because the pull mechanism with request and return of values is unchanged and so there will still be a need to

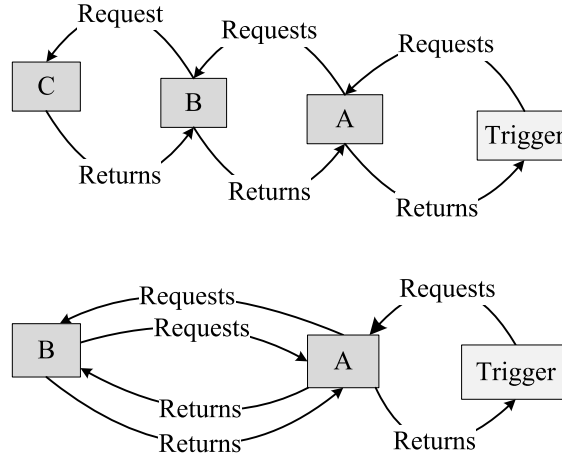


Figure 3.2: Data exchange between component models in a uni-directional composition (top) and a bi-directional composition (bottom) adapted from Tindall (2005).

model systems that have feedback loops and may require rescaling of temporally or spatially misaligned data transfers.

Past work using OpenMI for hydrologic and water resource modeling has primarily focused on applying the standard for modeling integrated water resource systems. For example, OpenMI was used to couple separate models that simulate groundwater hydrogeology, econometric farm level crop choices, and irrigated water use (Steward et al., 2009; Bulatewicz et al., 2010). Steward et al. (2009) emphasized the advantage of component-based modeling to assembly three different components from three different disciplines. In a related application, Fotopoulos et al. (2010) used OpenMI to build a flood model finding that OpenMI eased the integration across various software components and increased the flexibility and extensibility of the overall modeling system. Ewert et al. (2009) presented integrated assessments for policy support in agriculture that used a modeling framework built on OpenMI. The authors found that, while OpenMI provided an improved means for linking models to each other, to a GIS, or to a Decision Support Systems (DSS), the scientific basis for linking models across disciplinary boundaries and spatiotemporal scales required additional research. OpenMI has also been applied to urban hydrology to couple sewer system models with

river hydraulic models, illustrating the benefit of component-based modeling which allows existing models to be coupled through the runtime exchange of data (Reussner et al., 2009). These past efforts all point to the benefits of OpenMI and component-based modeling in general for coupling disparate modeling and information systems, however, as noted by Ewert et al. (2009), there are important scientific questions regarding model coupling that have not been well addressed.

We suggest that two of the most pressing questions in applying component-based modeling concepts to hydrologic systems are (1) how to simulate fully coupled processes with a shared boundary condition as two distinct components with a bidirectional link and (2) how to couple two model components that have spatially and/or temporally misaligned inputs and outputs. This is not to say these are the only two questions raised in regard to component-based modeling (Voinov et al., 2010), but given that hydrology is complicated with many examples of feedbacks between processes operating within watershed system, a successful integrated modeling approach must be capable of simulating such interactions. Likewise, it is well known that hydrologic processes have characteristic time or length scales and that these scales can vary for different fluxes within the hydrological cycle (Bloschl and Sivapalan, 1995). One of the challenges that component-based modeling attempts to address is that different components of the system can operate on independent spatial or temporal discretizations. In such cases, data transfers between spatially and temporally misaligned components are required so that they do not violate basic principles of hydrologic modeling (e.g., conservation of mass, energy, momentum). The goal of this research is therefore to study model coupling for cases where there are feedback loops between model components and those model components may run of different time steps requiring temporal interpolation of data exchanges.

To address this research goal, we conducted a hypothetical modeling exercise with the objective of estimating pollution concentration in a multimedia water/sediment



system. In the first part of the study we simulated this system using two approaches: the first was a tightly-coupled numerical model and the second was as two loosely-coupled model components with one representing the water medium and the other representing the sediment medium. We used the first modeling approach to provide a point of comparison for the component-based modeling approach. The two components were coupled so that they exchanged concentrations across the water/sediment boundary through a bi-directional link during simulation runtime. In the second part of the study, we made the two components run on different time steps in order to study how interpolation schemes can be used to rescale data transfers between components during simulation runtime. Details of the experiments are provided in the following methods section of this paper. Following the methods section, we present results and discussion of the experiments including details of how the bidirectional link and data transformations work in OpenMI. Finally, we present conclusions from our work drawn from the investigation into these two aspects of using component-based modeling to simulate hydrologic systems.

### 3.3 METHODS

#### 3.3.1 MODEL DEVELOPMENT

To better understand the application of component-based modeling for simulating fully coupled systems we considered a hypothetical case where water is above a sediment column, a constant source of a pollutant is injected into the water, and the pollutant is transported through the system by advection and diffusion. A two dimensional representation of the system can be described by the advection-diffusion equation

$$\frac{\partial C}{\partial t} = D_x \frac{\partial^2 C}{\partial x^2} + D_z \frac{\partial^2 C}{\partial z^2} - u \frac{\partial C}{\partial x} - v \frac{\partial C}{\partial z} \quad (3.1)$$

where  $C$  is the concentration of the pollutant (ppm) at a location  $(x, z)$  within the system at a time  $t$ ,  $u$  is the velocity in  $x$ -direction,  $v$  is the velocity in  $z$ -direction ( $\text{cm s}^{-1}$ ), and  $D_x$  and  $D_z$  are diffusion coefficients in the  $x$  and  $z$  directions ( $\text{cm}^2 \text{s}^{-1}$ ), respectively. For simplicity, in the water medium we assumed only advective transport in the  $x$ -direction and diffusive transport in the  $z$ -direction, and in the sediment medium we assumed only diffusive transport in the  $z$ -direction. With these simplifications to Equation 3.1, the governing equation for the water medium becomes

$$\frac{\partial C_w}{\partial t} = D_{w,z} \frac{\partial^2 C_w}{\partial z^2} - u_{w,x} \frac{\partial C_w}{\partial x} \quad (3.2)$$

where  $C_w$  is the concentration of the pollutant in water (ppm),  $D_{w,z}$  is the diffusion coefficient of the pollutant in water ( $\text{cm}^2 \text{s}^{-1}$ ) in the  $z$ -direction,  $u_{w,x}$  is the velocity of the pollutant in water ( $\text{cm s}^{-1}$ ) in the  $x$ -direction. Similarly, the governing equation for the sediment medium after applying the simplifying assumptions becomes

$$\frac{\partial C_s}{\partial t} = D_{s,z} \frac{\partial^2 C_s}{\partial z^2} \quad (3.3)$$

where  $C_s$  is the concentration of the pollutant in sediment (ppm),  $D_{s,z}$  is the diffusion coefficient of the pollutant in sediment ( $\text{cm}^2 \text{s}^{-1}$ ) in the  $z$ -direction. As described earlier, we solved the system using two approaches where in the first approach the water and sediment was modeled as a single system and in the second approach the water domain was modeled separately from the sediment domain but coupled through the shared boundary condition. Boundary conditions for the first approach for modeling the system were defined at the top of the water domain as  $C_w(x, 0, t) = 1$  ppm for  $0 \leq x \leq p$  where  $p$  is the length of the water domain (cm) that has a constant rate of pollutant injection (Figure 3.3). A transmissive boundary condition (Toro, 2009) was used to describe the right, left, and bottom edges of the domain (details for the transmissive boundary condition are discussed after discretization of each domain governing equation). In the second approach for modeling the system, a boundary condition was introduced at the water/sediment interface that required

$C_w(x, h, t) = C_s(x, h, t)$  to be satisfied, where  $h$  is the depth of the water column (cm) as shown in Figure 3.3. The initial condition of both approaches was  $C(x, z, 0) = 0$ .

Equations 3.2 and 3.3 were solved numerically for a two-dimensional grid in the  $x$ - $z$  dimensions as a discrete representation of the water-sediment domain (Figure 3.3). We considered a simple domain consisting of ten rows (indexed by  $i$ 's) and ten columns (indexed by  $j$ 's) to conduct the case study experiment. The water medium was represented by the top row in the domain ( $i = 0$ ), while the other nine rows represented the sediment medium. We approximated (Equation 3.2) using a finite difference approach with a forward explicit scheme

$$\frac{\Delta C_{0,j}}{\Delta t} = D_{w,z} \frac{C_{0,j} - 2C_{1,j} + C_{2,j}}{\Delta z^2} - u_{w,x} \frac{C_{0,j} - C_{0,j-1}}{\Delta x} \quad (3.4)$$

where  $C_{i,j}$  is the pollutant concentration at location  $(i, j)$ . Note that the  $w$  and  $s$  subscripts were not included for  $C$  in Equation 3.4 because the boundary condition will be controlled by the sediment domain. The transmissive boundary condition was used at the right edge of the water domain for both approaches and was approximated using a backward difference scheme such that  $C_w(b, z, t) = C_w(b - \Delta x, z, t)$  where  $b$  is the domain width (cm) and  $\Delta x$  is the cell width (cm) in the computational grid (Figure 3.3). Concentration within the sediment domain was approximated using a finite difference leap frog (time marching) scheme as shown in Equation 3.5.

$$\frac{\Delta C_{i,j}}{\Delta t} = D_{s,z} \frac{C_{i+1,j} + C_{i-1,j} + C_{i,j+1} + C_{i,j-1} - 4C_{i,j}}{\Delta z^2} \quad (3.5)$$

Again the  $w$  and  $s$  subscripts were not included for  $C$  in Equation 3.5 because the boundary condition will be controlled by the water domain. This approach for approximating the governing equation was chosen because it has high stability for PDEs with oscillatory solutions (Shampine, 2009). The transmissive boundary conditions used for the sediment domain for both approaches was approximated using a forward difference scheme for the left edge,  $C_s(0, z, t) = C_s(\Delta x, z, t)$  for  $h \leq z \leq d$  where

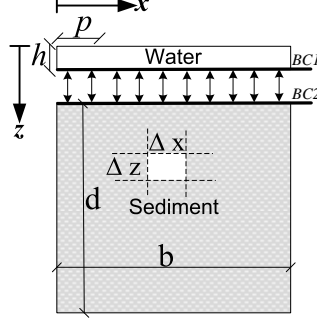


Figure 3.3: The water and sediment domains used in the component-based approach.  $BC1$  is the water component lower boundary values that act as the sediment component top boundary,  $BC2$  is the sediment component top boundary values that acts as the water component lower boundary.

$d$  is the sediment domain height (cm), and a backward difference scheme for the right edge,  $C_s(b, z, t) = C_s(b - \Delta x, z, t)$  for  $h \leq z \leq d$ . The sediment concentration at the bottom boundary was set to the concentration of the interior cell such that  $C_s(x, d, t) = C_s(x, d - \Delta z, t)$  where  $\Delta z$  is the cell height (cm) in the computational grid. In solving the Equations 3.2 and 3.3, we assumed that the diffusion coefficients were constant over space and time. It is well known that a space explicit scheme does not oscillate when the Péclet Number ( $Pe = v\Delta x/D$ ) is less than or equal to 2 and the Courant Number ( $Cr = v\Delta t/\Delta x$ ) is less than or equal to 1. Therefore we selected the parameter values used in the simulation (Table 3.1) to satisfy these conditions.

### 3.3.2 MODEL IMPLEMENTATION

We solved for concentration within the system using two approaches for structuring the code. In the first approach, which we will refer to as the *conventional approach*, the multimedia system was simulated using a tightly-integrated paradigm. By this we mean that the complete system, both the water and sediment medium, was represented within one code unit using a single multi-dimensional array. The numerical methods were programmed to operate on that array to solve Equations 3.4 and 3.5. The solution using this approach was implemented in Matlab. In the second

approach, which we will refer to as the *component-based approach*, the multimedia system was simulated as two model components: one representing the water medium and the second representing the sediment medium. Each component was developed independently without knowledge of how the other component was implemented so that they remain autonomous units. The components were then linked together into a composition so that boundary conditions could be passed between the components during the simulation run. The solution using this approach was implemented in C# .Net using the version 1.4 of the OpenMI Software Development Kit (SDK) and the approach for simplifying the creation of new OpenMI-compliant models proposed by Castronova and Goodall (2010). In the following paragraphs we provide further detail for the component-based modeling approach because it is the primary focus of this paper.

There were two steps in developing the component-based model: the first involved component development and the second involved linking components into a composition in order to define how the components exchange data during a simulation run. Component-based software architectures rely on the standardization of component interfaces to provide the “plug-and-play” functionality where model components can be reused in multiple compositions without the need to change or recompile the model source code. The standard component interface, acts as a contract between the model component and the controlling application that runs the composition (Szyper-ski, 2002). As stated earlier, we used the OpenMI as the component standard because it was designed specifically for water resource modeling. OpenMI provides the technical means for controlling a model component on a time-step basis. Our aim in this work was not to develop a component standard or composition orchestration application, but instead to use the existing tools provided through the OpenMI to model a hypothetical case study. We therefore built the components used in this

study using the OpenMI standard in order to understand how this paradigm can be used to simulate a fully-coupled system.

To create the OpenMI components for this study, we used an approach for creating process-level OpenMI components described by Castronova and Goodall (2010) and named the Simple Model Wrapper (SMW). The approach automates many of the details involved in implementing the OpenMI standard so that the model developer can more easily and quickly create basic process-level hydrologic model components that adhere to the OpenMI standard. Using the SWM approach, the water and sediment components were defined by a configuration file, a geospatial dataset file, and a model engine file (Figure 3.4). The configuration file is an XML file that defines the metadata for each model component (see Castronova and Goodall (2010) for details). The input geospatial dataset file defines model elements, system parameters, initial state variables, and other attributes specific for each model component. Finally, the model engine is a Dynamic Link Library (DLL) file that defines the behavior of the component and must implement three core methods: *Initialize*, *PerformTimeStep*, and *Finish*. The *Initialize* method reads the exchange items from the configuration file and the input parameters of each component from the geospatial dataset file and stores these properties in memory. The *PerformTimeStep* method advances the model component in time and performs the numerical calculations discussed earlier in the paper. Finally, the *Finish* method writes the output data for each component and releases the memory used by the components. The source codes for both components and tutorials describing how to create and use the components are available as part of the HydroModeler plug-in to the Consortium of Universities for the Advancement of Hydrologic Sciences, Inc. (CUAHSI) HydroDesktop software system ([www.hydrodesktop.org](http://www.hydrodesktop.org)).

The water and sediment components were linked into a composition by creating a bi-directional link between the two components. The bi-directional link specified that

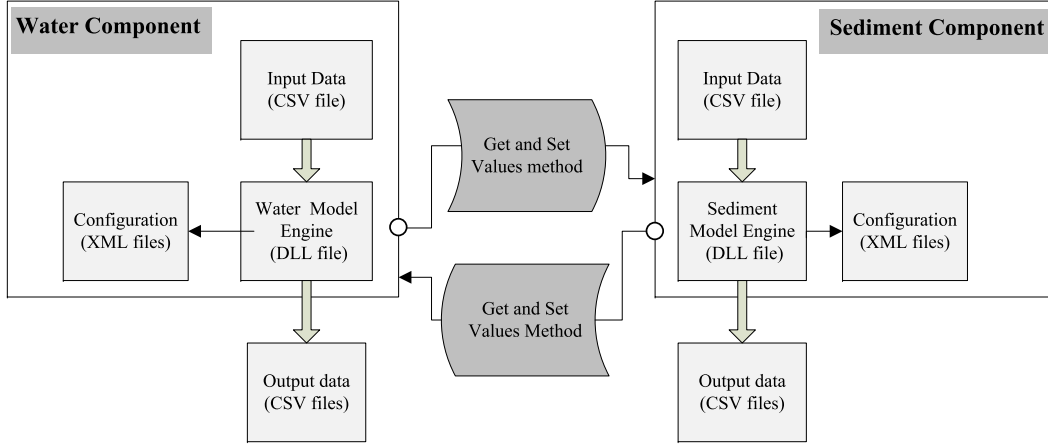


Figure 3.4: Details for the implementation of the component-based model approach.

the sediment component required concentration values from the water component, and that the water component required concentration values for the elements in the top layer of the sediment component. Because OpenMI 1.4 follows a pull-driven communication paradigm, both the water and sediment components request values from the other component within the configuration on each time step. To start the simulation, we linked the sediment component to a trigger component which initiates component communication, as described earlier in this paper. When one component requests values from a linked component, that component is required to reply with the exact information requested. A request is made for values associated with a set of elements, a quantity, and a specific time step or time span. The component to which the request is made must supply values for the requested elements and at the requested time step. If the requested data is not directly calculated by the component, as was the case in the second experiment we conducted where one component operated on a time step greater than the component that it was linked to, there must be an interpolation process to rescale data transfers.

For both the conventional modeling approach and the two experiments using the component-based modeling approach, which are described in the following subsections, we solved for  $C(t)$  assuming a constant pollution point-source located in the

Table 3.1: Model parameters for water and sediment media

	Water	Sediment
Velocity ( $\text{cm s}^{-1}$ )	0.80	0.00
Diffusion Factor ( $\text{cm}^2 \text{s}^{-1}$ )	$2.5 \times 10^{-4}$	$2.5 \times 10^{-4}$
No. of Rows	1	9
No. of Columns	10	10

Table 3.2: Properties included in the XML configuration files for the water and sediment components.

Component Name	Exchange Item Type	Element Set Name	Quantity Name	Units
Water Component	Output Item	ConcentrationBC1	Water Concentration	ppm
	Input Item	ConcentrationBC2	Sediment Concentration	ppm
Sediment Component	Input Item	ConcentrationBC1	Water Concentration	ppm
	Output Item	ConcentrationBC2	Sediment Concentration	ppm

first three nodes of water domain. The parameters for this study are given in Table 3.1 and the component configuration file attributes are given in Table 3.2. A constant velocity in the  $x$ -direction,  $u$ , and a constant rate of change with respect to time were given to avoid unsteady transport in the two-dimensional field. Interaction between the water medium and the sediment medium is through diffusive transport across the water/sediment boundary and therefore requires that the sediment component be able to obtain values calculated by the water component, and the water component be able to obtain boundary conditions from the sediment component during the model simulation run-time, as described earlier.

### 3.3.3 EXPERIMENT 1: BIDIRECTIONAL LINK

In the first experiment we used the component-based modeling approach to simulate the coupled system with a bi-directional link handling the feedback loop between the two components. The goal was to verify that the component-based model correctly simulated the system and to investigate how components coupled in a bi-directional



link communicate with one another. The conventional modeling approach, where the system is solved for the entire domain with the same boundary and initial conditions, was used as a point of comparison to the component-based modeling approach. We compared the results of the models at the first time step to ensure that the models were initialized with correct initial and boundary conditions, the tenth time step to check that the time marching computation and data exchange were occurring correctly, the 100<sup>th</sup> time step to check the stability of the numerical computations in both components, and the 2000<sup>th</sup> time step to judge the stability of the solution after a long time period. This comparison was done for time steps of 0.1 second, 0.5 second, 1 second and 10 seconds to ensure that both models were handling varying time steps correctly.

### 3.3.4 EXPERIMENT 2: TEMPORAL RESCALING

In the second experiment we modified the component-based solution used in the first experiment so that the two components operated on different time steps. While components are required to have the same simulation time period in OpenMI, they are not required to have the same time step. To explore this feature of coupling temporally misaligned components, we varied the internal time step for the water component  $t_w$  from 1, 2, ..., 10 s, while holding the internal time step of sediment component  $t_s$  constant at one second. This change was implemented by changing the time step of the components as specified in the time horizon element of each component's XML configuration file (Table 3.3).

OpenMI compositions that have linked components operating on different space or time discretizations require interpolation in order to transform data exchanges between components. It is common for temporal interpolation algorithms to require values from one or more previous time steps in the simulation. To handle this case, the OpenMI includes a *SmartBuffer* class that is used to store values from previous

Table 3.3: The time horizon and time steps used in the simulation.

Start Date Time:	08/20/2009 00:00:00 AM
End Date Time:	08/20/2009 01:00:00 AM
Time Steps:	0.1, 0.5, 1 and 10 s

time steps for a particular link in the computer's memory so that they can be accessed later when interpolated values are required (Figure 3.5). The functionality of a smart buffer is exposed through three methods: the first is used to fill the buffer with values, the second to empty values from the buffer, and the third for obtaining interpolated values generated using a particular interpolation algorithm from the buffer.

The OpenMI Software Development Kit (SDK) provides a technique to interpolate between misaligned components that can be described by the following equation

$$S_{r,i} = \frac{S_{b,i}^{n+1} - S_{b,i}^n}{t_b^{n+1} - t_b^n} (t_b^r - t_b^n)(1 - \alpha) + (S_{b,i}^n) \quad (3.6)$$

where  $t_b^n$  is the  $n^{th}$  entry in the buffered list,  $S_b^n$  is the  $n^{th}$  scalar set in the buffered list,  $\alpha$  is the relaxation factor, and  $t_r$  is the requested time. If the relaxation factor ( $\alpha$ ) is zero, a linear interpolation is preformed. In the case where the relaxation factor is one, the return value is the nearest available value. For relaxation factors between one and zero, a weighted average between these two cases is returned (Sinding et al., 2005).

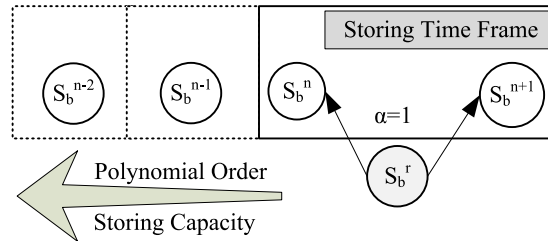


Figure 3.5: Provided interpolation routine in OpenMI which allows for linear interpolation (when  $\alpha = 0$ ), nearest neighbor (when  $\alpha = 1$ ), and a weighted approach between these two alternatives (when  $0 < \alpha < 1$ )

We extended the OpenMI by introducing two other interpolation techniques and applied both new implementations to our study of the impact of interpolation on overall system mass balance. The first interpolation method that we implemented was the quadratic formula (Equation 3.8).

$$\begin{aligned}
S_{r,i} = & \left[ \frac{(t_b^r - t_b^n)(t_b^r - t_b^{n-1})}{(t_b^{n+1} - t_b^n)(t_b^{n+1} - t_b^{n-1})} \right] (S_{b,i}^{n+1}) + \\
& \left[ \frac{(t_b^r - t_b^{n+1})(t_b^r - t_b^{n-1})}{(t_b^n - t_b^{n+1})(t_b^n - t_b^{n-1})} \right] (S_{b,i}^n) + \\
& \left[ \frac{(t_b^r - t_b^{n+1})(t_b^r - t_b^n)}{(t_b^{n-1} - t_b^n)(t_b^{n-1} - t_b^{n+1})} \right] (S_{b,i}^{n-1})
\end{aligned} \tag{3.7}$$

The three  $S_{b,i}$  values in Equation 3.8 can be applied for either uniform or nonuniform time steps. Because the water component was set to have a different time step compared to the sediment component, the storing capacity of the *SmartBuffer* was extended in order to perform this interpolation method.

The second interpolation method implemented was a cubic spline scheme whereby data exchanges are approximated by a curve in a piecewise manner by a third-order polynomial over each interval  $t_b^n < t_b^r < t_b^{n+1}$ . This is done in such a way that both the first and second derivatives of the curve at the end of the interval match those of the approximation of the immediate left at  $t_n$  and those to the approximation to the right at  $t_{n+1}$ . This can be expressed as

$$\begin{aligned}
S_{r,i} = (t_b^r - t_b^n) & \left[ \frac{S_{b,i}^n + S_{b,i}^{n+1}}{(t_b^n - t_b^{n+1})} - (A + B)(t_b^{n+1} - t_b^n) \right] + \\
& C \frac{(t_b^r - t_b^n)^3}{6(t_b^{n+1} - t_b^n)} + D \frac{(t_b^r - t_b^{n+1})^3}{6(t_b^{n+1} - t_b^n)}
\end{aligned} \tag{3.8}$$

where  $A$ ,  $B$ ,  $C$ , and  $D$  are given as follows.

$$\begin{aligned} A &= \frac{(S_{b,i}^{n-2} - 2S_{b,i}^{n-1} + S_{b,i}^n)}{(t_b^{n-2} - t_b^{n-1})^2} \\ B &= \frac{(S_{b,i}^{n-1} - 2S_{b,i}^n + S_{b,i}^{n+1})}{(t_b^{n-1} - t_b^n)^2} \\ C &= \frac{(S_{b,i}^{n-1} - 2S_{b,i}^n + S_{b,i}^{n+1})}{(t_b^{n-1} - t_b^n)^2} \\ D &= \frac{(S_{b,i}^{n-2} - 2S_{b,i}^{n-1} + S_{b,i}^n)}{(t_b^{n-2} - t_b^{n-1})^2} \end{aligned}$$

We used these two new interpolation methods implemented in OpenMI through this work (quadratic and cubic spline) along with the linear interpolation method provided through the OpenMI SDK to rescale data transfers between the sediment and water components in the composition. The goal of this second experiment was both to understand the data rescaling process and how it was handled by OpenMI, and also to quantify the impact of different interpolation routines on minimizing overall system mass balance in this specific case study for varying time step differences between the two model components.

### 3.4 RESULTS AND DISCUSSION

The two experiments conducted through this research focus on two aspects of component-based modeling at its application to hydrologic systems. In the first experiment we explored how data is transferred between two model components linked with a bi-directional link using OpenMI. Bi-directional component communication represents a complicated case for component-based modeling, and we were interested in learning how this case is handled when modeling feedback loops. In the second experiment, we investigated the case where two components are linked but each component runs on a different time step. The OpenMI includes the concept of data transformations to couple spatially or temporally misaligned components. Our goal in this second experiment was to understand how different interpolation algorithms

could be inserted into the OpenMI to rescale data and minimize overall system mass balance. For both experiments, we compared the component-based model to a second model of the same system that was implemented using a more conventional numerical scheme. The conventional model was used to ensure that the component-based models produced the same results and to quantify mass balance errors in the temporally misaligned model configuration.

#### 3.4.1 RESULTS FROM EXPERIMENT 1: BIDIRECTIONAL LINK

Results from the first experiment showed that the component-based model implementation and the conventional model implementation produced identical results for the simulation (Figure 3.6). The results were as expected. In the first time step, the concentration increased across the water medium due to advective transport. Then, as time progressed, the mass was transported by diffusion into the sediment due to the concentration gradient across the water/sediment interface. After ten seconds the concentration gradient was parallel in the water medium and had reached a state where diffusion dominates the propagation of the pollutant in both media. The mass continued to be transferred into the sediment until the sediment reached steady-state so that the entire medium is saturated with the pollutant. The coupled component simulation was run for four different time steps (0.1, 0.5, 1, and 10 s) to investigate the sensitivity of the solution to time step size and also to ensure the numerical stability of the schemes used. When we used a time step that was smaller than one second, we had to change the *TimeEpsilon* attribute in the OpenMI Linkable Engine class of each component to be compatible with the corresponding time step of the model.

The protocol for communicating data between the two components was closely monitored in the experiment and is summarized in Figure 3.7. After the two components are initialized, the data exchange began with the trigger component requesting

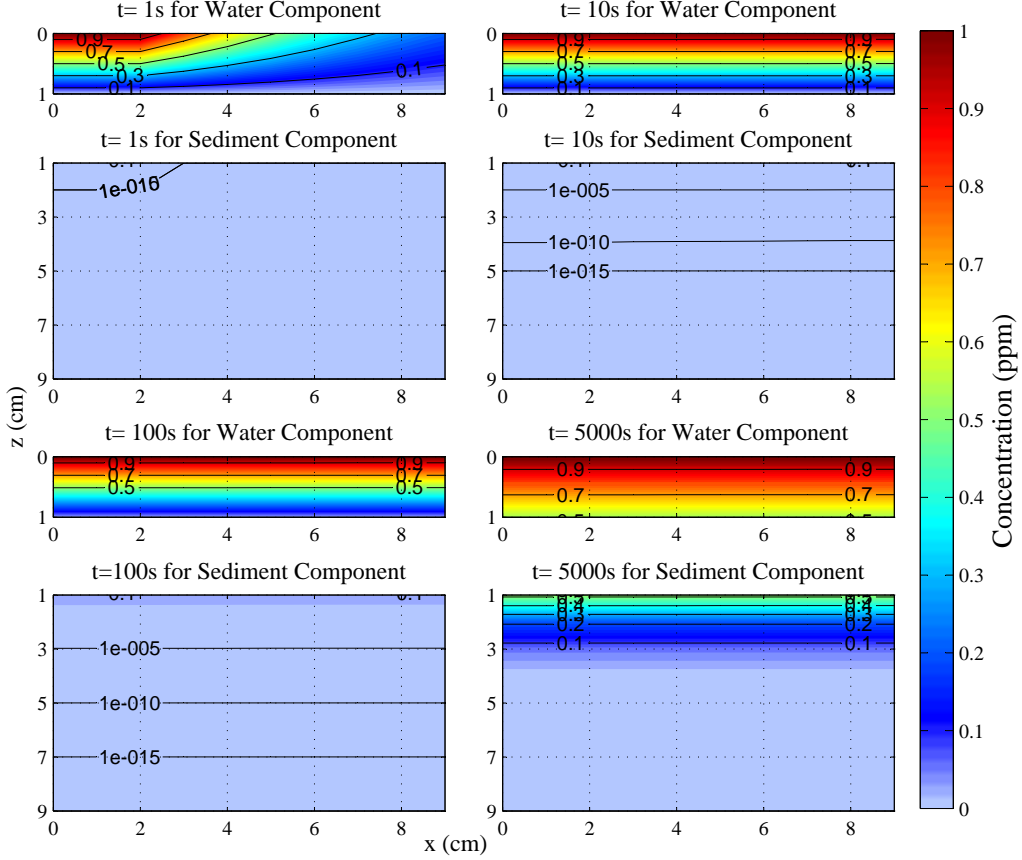


Figure 3.6: Results of the coupled water/sediment system for both the conventional and component-based approach.

an exchange item from the first component in the chain: the water component in this case (Arrow 1 in Figure 3.7). The water component then requested concentration values for the boundary nodes of the sediment component for the initial time step in the simulation (Arrow 2 in Figure 3.7). The sediment component required boundary conditions from the water component, and therefore requested these values from the water component on the same time step (Arrow 3 in Figure 3.7). Time cannot advance until each component's data request has been answered, so at this point there was a deadlock where the sediment and water components were co-dependent on a shared boundary condition. Both of the components have not received their data requests, so they do not have sufficient information to compute the data requested and are therefore unable to advance in simulation time. This is the challenge with a

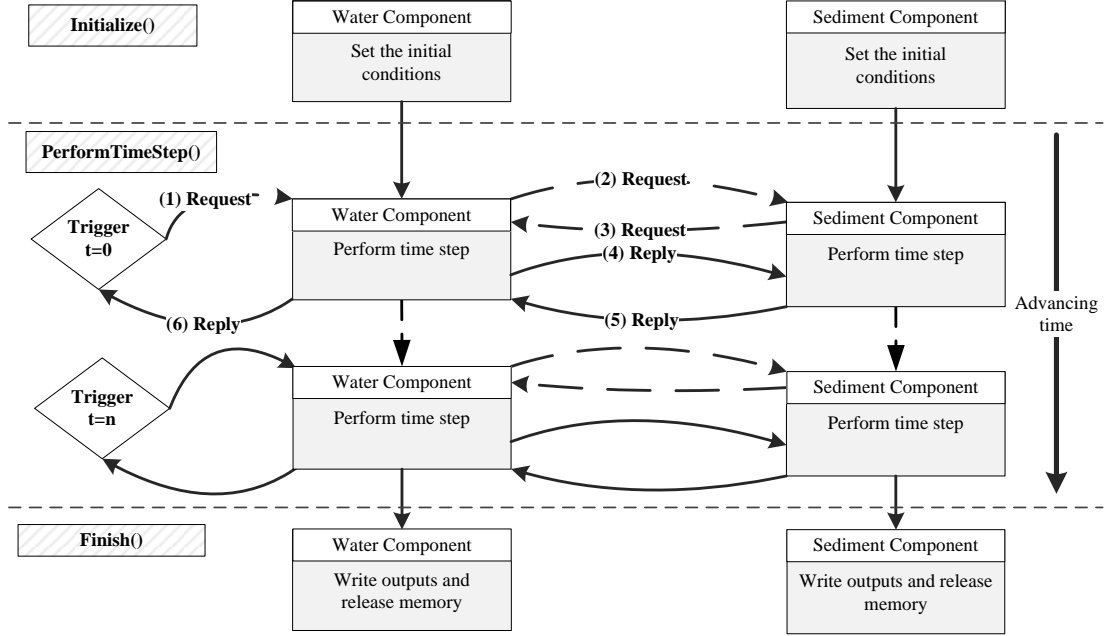


Figure 3.7: Communication flow for a bidirectional link within OpenMI between the water and sediment components adapted from Sinding et al. (2005).

bi-directional link: each component requires data from the other component in order to step in time.

OpenMI handles deadlocks in bi-directional component linkages like the one described in the previous paragraph in the following way. First, the OpenMI standard requires that when a component has an unanswered request for a value, it is not allowed to issue any additional data requests. Second, OpenMI requires that a component always return values when a request for data is issued. Therefore in this experiment, when the water component requested data from the sediment component, the sediment component then requested data from the water component, and the water component must at that point answer that data request because it has an outstanding data request. The water component answers the data request issued by the sediment component with its best estimate for the concentration values at the water/sediment boundary at the current time step. In this case, the best estimate for values are the concentration values on the previous time step, as it is assumed that

these values will be approximately equal to the concentration value on the current time step (Arrow 4 in Figure 3.7). At this point in the model run, the sediment component can answer the water component (Arrow 5 in Figure 3.7), and the water component is able to respond to the request for values issued by the configuration trigger (Arrow 6 in Figure 3.7). This concludes the first time step of the model configuration. The trigger then invokes the water component for the next time step, requesting data for that time step. The same interaction between the sediment and water component is repeated for this time step, and the model continues until all time steps have been completed.

This experiment shows that the OpenMI version 1.4 SDK handles deadlocks in bidirectional links by having the component estimate a value on the current time step based on the values of that same variable calculated for the previous time step. We were able to reproduce the component-based model solution using a more conventional numerical solution by making this same assumption in the conventional numerical algorithm. For some cases, the assumption that current conditions can be approximated by past conditions may not be sufficient, in which case modification of the OpenMI SDK to produce a more sophisticated means for handling component bi-directional links may be necessary. For example, while we used an explicit scheme for the two model components in the experiment, an alternative approach would have been to implement the components using an implicit scheme. An implementation using an implicit scheme would have required iteration within a time step to obtain the correct values for the boundary conditions. The OpenMI Technical Association has created a utility package designed to support advanced data control of model compositions, such as iteration within a time step, that is available in a prototype form but not as part of the official OpenMI SDK. The utility package includes three data control options: iteration, calibration, and logic switch. The iteration controller is a linkable component that acts as a mediator between components and requests



the models to step back one time step to adjust the values in the implicit scheme. The input exchange item of the receiving component and the output exchange item of the providing component should be connected to the same iteration controller component. Because we choose to use an explicit scheme when implementing the components, it was not necessary to use this prototype utility package for advanced control of OpenMI compositions.

### 3.4.2 RESULTS FROM EXPERIMENT 2: TEMPORAL RESCALING

In the second experiment performed as a part of this work, we studied how temporal misalignments of model components, and more specifically interpolation of component data transfers, are handled by OpenMI. Introducing different interpolation techniques to the OpenMI and using existing interpolation techniques provided within the OpenMI SDK, we were able to quantify the impact of interpolation on system mass balance error. To explore this topic, we relaxed the assumption used in the first experiment that the water and sediment components operate on the same time step. This introduces the need to rescale data exchanges between the two components using interpolation schemes. In this second experiment we were interested in better understanding how the case of temporal misaligned of component data exchanges is handled within the OpenMI paradigm. We were also interested in learning how to include new interpolation algorithms into the OpenMI system, and how these different interpolation methods influence the overall mass balance error within the coupled system of components.

Figure 3.8 depicts the protocol OpenMI uses to rescale data transfers by demonstrating the steps involved in one cycle of sediment request for interpolated data from water component. At the point in time which the figure depicts, both components have been initialized and the water component has completed its first time step of the simulation  $t_w^1$ . First, the sediment component attempts to compute values for the

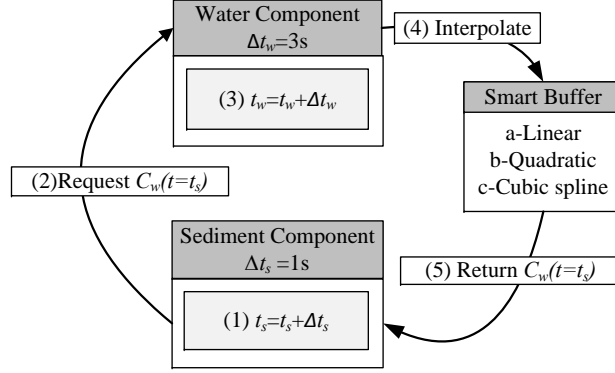


Figure 3.8: Communication flow for rescaling data transfers when components inputs and outputs are temporally misaligned.

current time step of the simulation run ( $t_s = t_s + \Delta t_s$ ). In order to do this calculation, the sediment component requires values from the water component and so it issues a data request to the water component for  $t_s$ . The water component is then required to return values of the water concentration for time equal to  $t_s^1$ . The water component will evaluate if the requested time is before, after, or equal to its own internal time. The water component determines in this case that the requested time is less than its internal time ( $t_s < t_w$ ), therefore the water component will request the *SmartBuffer* to interpolate values at time step  $t_s$ . Finally the *SmartBuffer* returns the data values back to the sediment component. Thus, the *SmartBuffer* object is central in the interpolation process required to couple spatially or temporally misaligned model components.

As stated earlier, the OpenMI SDK provides an interpolation algorithm that can be used to implement linear interpolation, nearest neighbor interpolation, or a weighted combination of these two interpolation algorithms. For some cases, these options may not be optimal for rescaling data between spatially or temporally misaligned model components. Many hydrological flux and state variables follow a polynomial behavior, for example, and therefore it would be necessary to have such interpolation methods available as a part of the OpenMI toolkit. OpenMI is developed as an open source project, so we were able to extend the set of interpolation schemes available

through OpenMI to include both the quadratic polynomial and cubic spline schemes as described in the Methods section.

After adding these new interpolation algorithms to OpenMI, we applied them to handle the case where the water and sediment components had different time steps. In linear interpolation, the *SmartBuffer* was required to store only two values to perform the interpolation (Figure 3.8). Therefore, as time advances, the *SmartBuffer* was updated with the new value as it became available. This means that the *SmartBuffer* was acting as a moving frame only storing two values at any given time. More complicated interpolation methods require the *SmartBuffer* to store more than two data values in the buffer at any given time. This could have performance costs in terms of increased search time on the smart buffer and increased storage required for the buffer, however this aspect of the work was not investigated in detail as part of this study. We did not notice any measurable performance costs in this work, however this was a purposely small modeling case study and so we cannot rule out the possibility of performance costs for a larger modeling exercise.

To understand the benefit of each interpolation scheme in minimizing the mass balance error between the temporally misaligned coupled components, the internal time step of the sediment component ( $\Delta t_s$ ) was fixed at one second and the water component internal time step ( $\Delta t_w$ ) was varied from two through ten seconds with a step of one second. Three interpolation schemes – linear, quadratic, and natural cubic spline – were used to interpolate between the previous stored values and the current value of the water component value at an intermediate time equal to  $t_s$  before the values are passed back to the sediment component so that it can proceed in its own calculations. Figure 3.9 represents the impact of each interpolation scheme on the computed sediment boundary concentration. The comparison was done using the concentration values of the sediment boundary layer calculated when both the water and sediment components were operating on the same internal time step and

the concentration values calculated using interpolation schemes separately when both components were temporally misaligned. The outputs obtained from the temporally aligned component configuration are shown on the horizontal axis and the predicted (interpolated) outputs from the misaligned component configuration are shown on the vertical axis. The results are presented at three different locations in the sediment top layer ( $x = 4, 6$ , and  $10$  cm), and for three different  $\Delta t_w$  values ( $2, 4$ , and  $10$  s). The figure shows that the interpolation error increases as the distance from the source pollution source increases so that the error recorded at point ( $x = 10$  cm) is greater than the error recorded at point ( $x = 2$  cm). The cubic spline scheme best matches the actual values for small differences in time steps between the water and sediment components for all three locations. As the difference between the operating times steps of the components increase, the deviation of the interpolated values using the three interpolation schemes also increases. The linear scheme shows the best results for large differences in temporal time step sizes between components.

Figure 3.10 shows the total mass error in the sediment media as the time advances during the simulation run. Total mass error represents the difference between total mass of the sediment media when both components are temporally aligned and the predicted mass when the components are misaligned with the three differences in time steps ( $2, 4$ , and  $10$  s). The results show that the cubic scheme minimizes the error in mass transfer between coupled components the best for the first two cases ( $\Delta t_w = 2$  s and  $\Delta t_w = 4$  s), but does the worst for the third case ( $\Delta t_w = 10$  s). For this third case, a linear interpolation is best at minimizing mass balance error in the misaligned component configuration. In general we found that for smaller time steps, linear interpolation has the largest error while cubic spline interpolation results in the smallest error. We also found that the opposite is true for larger time steps with spline interpolation producing the largest error and linear interpolation producing the smallest error. The reason for this finding is that an interpolated concentration value

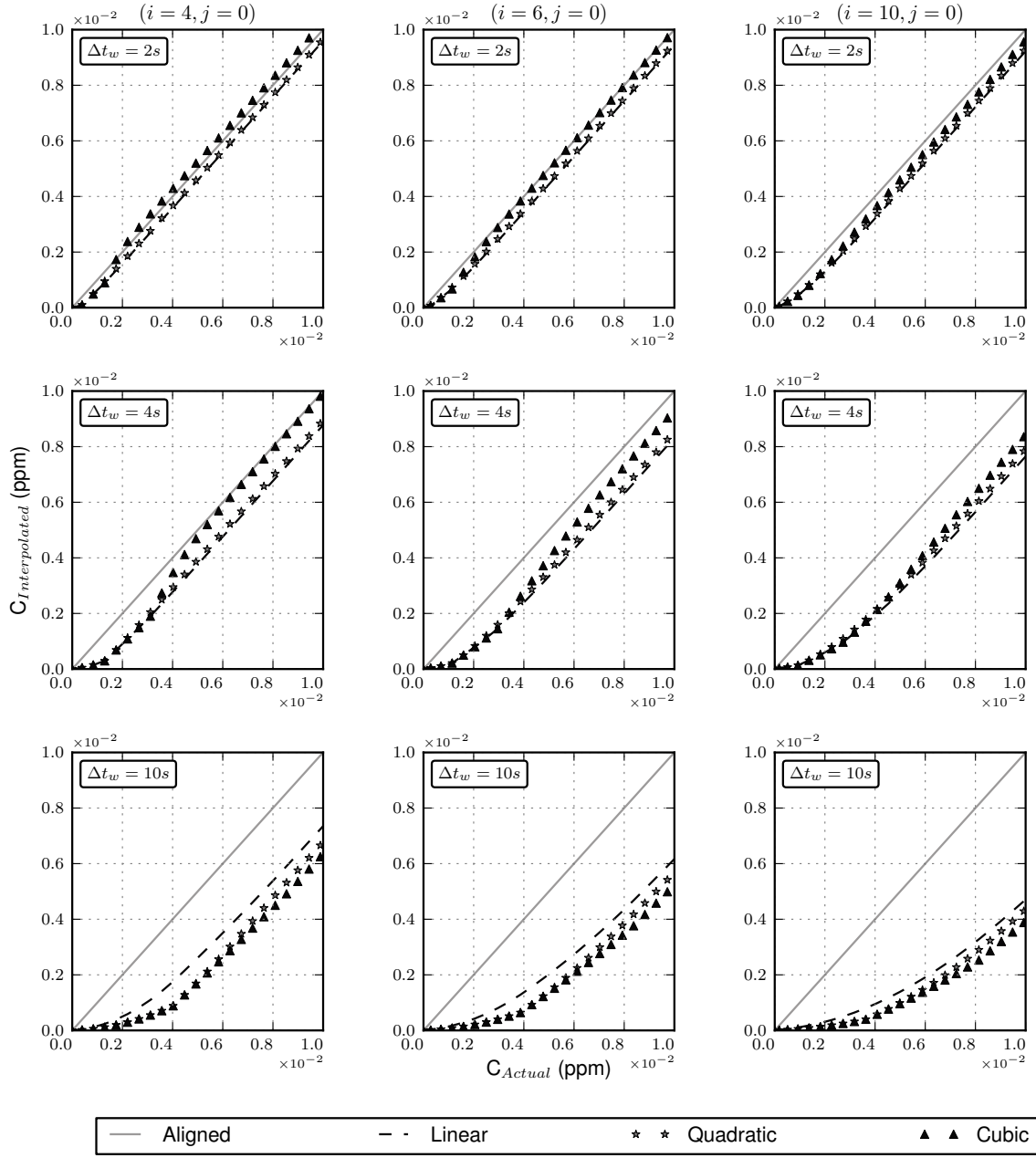


Figure 3.9: Results of linear, quadratic, and cubic spline schemes in minimizing the mass balance error between temporally misaligned coupled components. Comparison is for three locations in the sediment top layer ( $x = 4, 6$ , and  $10$  cm) and for three values of internal time step differences ( $2, 4$ , and  $10$  s).

is estimated from the previous time step value of the node, and the previous time step value of the spatially adjacent nodes. For the case of the linear interpolation scheme, only the previous value is used in the interpolation, while the quadratic scheme uses the two previous values. The cubic spline interpolation scheme is more complicated in that it uses a piece-wise interpolation that employs a third order polynomial between each of the subintervals  $[t_i, \dots, t_{i-3}]$  in order to satisfy continuity at junction points between curve segments and continuity in its first and second derivatives. This high dependence on neighboring values in space and time explains the difference in the interpolated values calculated using cubic spline scheme when compared to the values calculated using linear scheme for large time steps. Furthermore, for the first few time steps, the cubic spline interpolation is actually performing a linear interpolation because there are not a sufficient number of previous values to perform a cubic spline interpolation. Then, once a sufficient number of previous values are available to perform the cubic spline interpolation, the interpolation scheme is able to minimize mass balance error between the two coupled but temporally misaligned model components. This fact explains the sudden decrease in the mass balance error after the third time step when using the cubic spline interpolation for ( $\Delta t_w = 2$  s and  $\Delta t_w = 4$  s).

### 3.5 SUMMARY AND CONCLUSION

While previous work has demonstrated the benefits of using a component-based modeling design for simulating hydrologic systems, there remain important research questions about the applicability of the approach for modeling complicated system dynamics. This work provides a detailed view of two aspects of component-based modeling relevant for simulating hydrologic systems: feedback loops and misalignment of data exchanges. We explored these aspects specifically for the OpenMI component-based modeling protocol because it was designed and development for the water resource

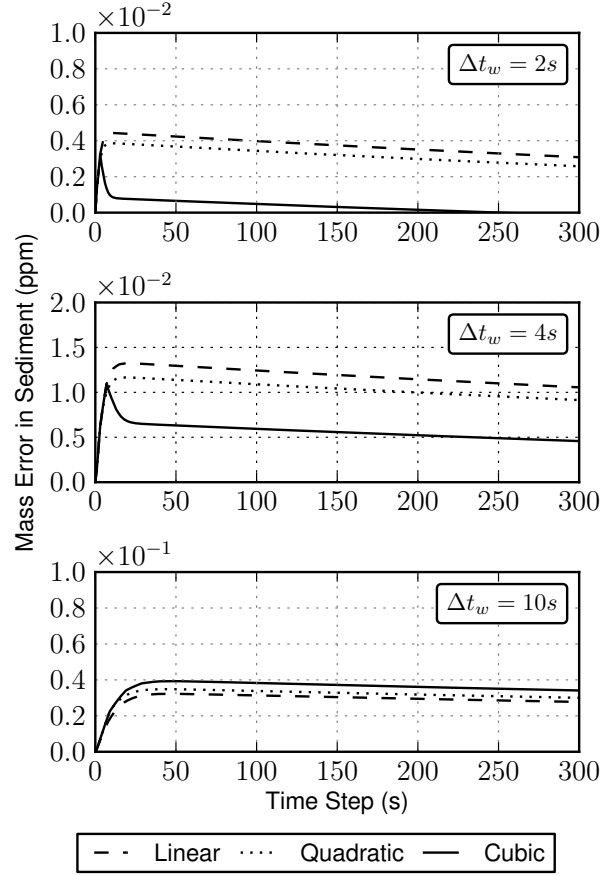


Figure 3.10: Total mass error in sediment domain between the aligned component configuration and the misaligned component configurations coupled using a linear, quadratic, and cubic spline scheme for three values of internal time step differences (2, 4, and 10 s).

modeling community. The topics were explored through a simplified case study of mass transport within a mixed media system with water over a sediment column. A component-based implementation of this system was compared with a more conventional numerical solution to the same system. This comparison provided a means for understanding how the component-based composition was solved, and for quantifying mass balance errors in the case where component output needs to be temporally rescaled to accommodate the input needs of another component.

In the first experiment conducted where we examined how feedback loops (or what OpenMI terms bi-directional links) between components are handled, we found

that OpenMI handles such cases in the following way. First, when a component has an unanswered data request, it is not allowed to issue any additional data requests. Second, a component is required to always return values when a request for data is issued. These two requirements result in components having to estimate values based on previous values calculated by the component when two components are coupled with a bidirectional link. The design of OpenMI makes it possible to enhance the logic for handling feedback loops by adopting a scheme that allows for iteration between components to converge on a shared boundary condition. One finding of this research is that such a solution may be required for cases where model components have large time steps, for example, where it is insufficient for the component to reply to a request for values on the current time step with the values from the previously time step as its best estimate. Future research is needed to more fully explore possibilities for more sophisticated solutions like those available in the prototype advanced controller utilities package to handle iteration between components coupled through a bi-directional feedback loop.

In the second experiment where we examined how misaligned component interactions are handled within OpenMI, we found that the framework provided an open architecture whereby new interpolation algorithms could be easily added in order to diversify the schemes available to users for rescaling transfers between components. For the particular advection-diffusion case study, we found that a cubic spline interpolation was best suited for minimizing system mass balance error for cases where there is a small time step difference between model components, whereas a linear interpolation performed best for large time step differences between model components. An important point is that multiple rescaling techniques can be made available through the OpenMI standard and so the user can select the most appropriate approach for a specific application, or add a new approach to the system if necessary. An alternative approach if the users cannot define the appropriate scheme for their



application is testing the performance of each scheme following the methodology in our second experiment. Open source software and open modeling architectures also allow modelers to share not only model components, but also interpolation routines and other useful tools that are necessary to modeling hydrologic systems. While this is possible in OpenMI version 1.4, OpenMI version 2.0 provides a more flexible approach for adapting output data to fit the input requirements of another model by introducing the AdaptedOutput construct for rescaling data in space or time, or performing on-the-fly unit conversions between linked models. Users could create a time interpolation adapted output construct and easily share it with others for reuse in component-based modeling applications.

In conclusion, component-based modeling, where a complex system is decomposed into a set of simpler components that act as separate but linked units, presents many benefits for modeling hydrologic systems. Each component can be designed, developed, and maintained by different groups, but still be used by a large community of modelers in their own simulations. OpenMI provides an implementation of component-based modeling specifically for the water resources community. This work provides a detailed look at how component-based modeling in general and OpenMI in particular handle what we consider to be two of the more complicated challenges in representing hydrologic systems as coupled components. Feedback loops are handled in a fairly simplistic way in the current implementation of the OpenMI SDK, although more complicated schemes that allow for iteration between components that are coupled through a feedback loop dependency. The OpenMI includes a sophisticated means for handling data transfers between components that do not have the same spatial or temporal discretization. While the framework only includes a few interpolation methods, it is possible to add new interpolation algorithms, as we have done in this work, that plug into the system. Nonetheless, users of component-based architectures must understand potential errors introduced when coupling spatially or

temporally misaligned models. This work is an attempt to quantify such errors for a specific case study and to understand how different interpolation routines can be used to minimize errors.

# CHAPTER 4

## AN ONTOLOGY FOR COMPONENT-BASED MODELS OF WATER RESOURCE SYSTEMS <sup>1</sup>

### 4.1 ABSTRACT

Component-based modeling is an approach for simulating water resource systems where a model is composed of a set of components, each with a defined modeling objective, interlinked through data exchanges. Component-based modeling frameworks are used within the hydrologic, atmospheric, and earth surface dynamics modeling communities. While these efforts have been advancing, it has become clear that the water resources modeling community in particular, and arguably the larger Earth Science modeling community as well, faces a challenge of fully and precisely defining the metadata for model components. The lack of a unified framework for model component metadata limits interoperability between modeling communities and the reuse of models across modeling frameworks due to ambiguity about the model and its capabilities. To address this need, we propose an ontology for water resources model components that describes core concepts and relationships using the Web Ontology Language (OWL). The ontology that we present, which is termed the Water Resource Component (WRC) ontology, is meant to serve as a starting point that can be refined over time through engagement by the larger community until a robust knowledge framework for water resource model components is achieved. This paper

---

<sup>1</sup>Mostafa M. Elag and Jonathan L. Goodall, Accepted for publication in *Water Resources Research*, 2013.

presents the methodology used to arrive at the WRC ontology, the WRC ontology itself, and examples of how the ontology can aid in component-based water resources modeling by (i) assisting in identifying relevant models, (ii) encouraging proper model coupling, and (iii) facilitating interoperability across Earth Science modeling frameworks.

## 4.2 INTRODUCTION

In the hydrologic community, solving complex water resources problems has migrated from a monodisciplinary to multidisciplinary approach (Hornberger et al., 2012; Wagener et al., 2010; Scholten et al., 2007). Earth Science models have also grown in recent years from activities undertaken by individuals or small groups, to larger and more collaborative activities (Syvitski et al., 2011; Voinov et al., 2010; Famiglietti et al., 2008; Blackmon et al., 2001). Together, these trends have increased the sophistication of models, not only in terms of the mathematical representation of physical processes, but also in terms of the software required for constructing state-of-the-art simulation models.

In building modern modeling systems capable of supporting multidisciplinary science within a community of developers and users, we see two basic paradigms that can be adopted. In the first paradigm, the model is designed, built, and controlled by a small group of developers. The model code is under the control of this group, thus simplifying the development and maintenance of the code, but at the same time limiting the size of the community contributing to the model development process. In the second paradigm, a modeling framework is designed, built, and controlled by a small group of developers, but it is possible for a larger community to contribute the models used within the framework. Model developers create their models as components that adhere to standards required for making their model interoperable with

other models in the framework. This component-based approach for building models is newer and less well established, however it has been gaining attention in recent years (Moore and Tindall, 2005; Syvitski et al., 2011; Goodall et al., 2011; Peckham et al., 2012).

A key distinguishing feature of component-based modeling is that each model within the system is independent yet able to be integrated with other models in what has been described as a “plug-and-play” manner (Peckham et al., 2012). Component-based modeling is a key principle in advancing modeling frameworks, providing the flexibility and extendability whereby a system can be assembled out of a set of independent functional units (Argent, 2004). By decentralizing the model functionality into independent components, there can be a relative freedom from the assumptions controlling the more conventional centralized approach for constructing models. It allows specialists to focus on implementation details for individual components within a system, and it allows stakeholders a way to view holistic systems that are built from these more detailed components. These properties also make component-based modeling an attractive approach for building community-based modeling systems (Voinov et al., 2010).

While there are obvious benefits to using a component-based approach for water resources modeling, there are also challenges that must be overcome in order to encourage broad adoption of the approach. One important challenge associated with the component-based modeling approach is that model integration is not simply the proper assemblage of components from a technical standpoint, rather it also requires scientific knowledge of the underlying coupling between each component (Athanasiadis et al., 2006, 2011; Voinov and Shugart, 2012; Elag et al., 2011; Castonvoa et al., 2012). Coupled model components often exchange values based on a message-passing scheme where one component requests a particular variable from a second component. Because components may be built and maintained by different

groups, the variables passed between components must be well described to ensure basic characteristics such as variable units are consistent between coupled components. This requires both establishing core metadata for concepts such as “variable” that are shared between models, and creating software tools that are able to analyze and properly align messages between components (Athanasiadis et al., 2011).

Scientists wanting to couple components across disciplinary boundaries or modeling frameworks face challenges that extend beyond simply ensuring variable units are consistent across coupled models. They must also consider (i) semantic heterogeneity across disciplines due to the variety of terminology used to describe the equations, variables, parameters, and units within models, (ii) diversity of concepts used to define components’ functionality and relationships, which results in overwhelming complexity for linked model compositions, (iii) syntactic heterogeneity in metadata structure used to describe a component across modeling frameworks, which hinders a component’s reusability, and (iv) coupling inconsistency resulting from mismatched spatial or temporal data exchanges, or from incompatible semantics used by different models (Argent, 2004; Voinov and Shugart, 2012; Peckham et al., 2012; Rizzoli et al., 2008; Elag et al., 2011; Janssen et al., 2011; Argent et al., 2006). These issues collectively result in a lack of shared understanding and poor communication within and between users of the component-based modeling framework. They are an important reason for why scientists have argued that working in communities to develop models may result in more focus on the process of creating a model rather than the final product of the model itself (Voinov and Shugart, 2012). Given these challenges, it is clear that if component-based modeling is to be broadly adopted by the community, these issues must be overcome.

We believe that an important step in overcoming these challenges is for the community to agree on an ontology that specifies and organizes the concepts and terminologies related to model components used in water-related disciplines. An ontology

is an explicit conceptualization of human knowledge that focuses on shared understanding by defining vocabularies that represent and communicate knowledge about a specific domain (Gruber, 1993). Establishing a shared understanding of concepts aids in eliminating conceptual and terminological confusion (Beran and Piasecki, 2009; Uschold and Gruninger, 1996). Furthermore, an ontology is the backbone of the Semantic Web, which was introduced by the World Wide Web Consortium (W3C) as a method and technologies for information integration, processing, and querying on the Web (Berners-Lee and Fischetti, 2001; Fensel et al., 2011). Gruber (1993) specifies the primary characteristics of an ontology to be (i) a clear structure, (ii) an easily inferred relationship between concepts, (iii) the flexibility to merge with other ontologies, (iv) the extensibility to accommodate any required future modifications, and (v) the ability to overcome semantic mismatches between the information provider and the user. This paper outlines our effort to build an ontology that satisfies these five characteristics for use in component-based modeling of water resource systems.

Our work relates to recent work in other related scientific disciplines to use ontologies to describe their domains. For example, Zhong et al. (2009) introduced an ontology in the geology domain to organize the concepts of fractures in order to facilitate communication among the highly diverse professional and academic communities related to the domain. In the agricultural domain, the System for Environmental and Agricultural Modeling; Linking European Science and Society (SEAMLESS) project developed a modeling framework to integrate approaches from economic, environmental, and social sciences. In this project different ontologies have been created to ensure the semantic and conceptual integration of models and future scenarios. For example, Rizzoli et al. (2008) created an ontology to enrich the semantics of model exchange items including parameters, I/O variables, and state, and Janssen et al. (2011) developed ontologies to ensure the semantic and conceptual integration between coupled models from different domains to assess agricultural land use changes.

Finally, Janssen et al. (2009) created the assessment project ontology to unify the concepts used among stakeholders, scientists, and modelers in implemented scenarios across models, policy problems, and scales. These ontologies were created for specific use cases, and none of the ontologies address the objective of this work: to define a component model used within the water resources domain.

The major contribution of this research is, therefore, an ontology that provides a unified and structured view of water resources model components. We acknowledge that creating a complete and robust ontology for water resource model components is beyond the scope of a single paper because doing so is an iterative process that requires input and refinement from a larger community (Janssen et al., 2009). Given this, our goal is for the ontology to serve as the starting point for a community agreed upon ontology for water resource model components. If such an ontology can be established, it will enhance component-based modeling activities both within the water resources community and across disciplinary boundaries by establishing an agreed upon understanding of the knowledge underlying model components. This process cannot proceed without a beginning ontology that establishes the core knowledge framework where experts from different domains can contribute their own conceptualizations and metadata needs. Our work is meant to provide this beginning ontology that builds on related efforts to create ontologies within the Earth Science community, but will likely evolve as more developers and users engage in the design process.

The remaining sections of the paper are organized as follows. Section 2 discusses the methodology used to create the ontology along with background theory on ontologies to orient the reader. Section 3 presents the proposed ontology, which we have named the Water Resources Component (WRC) ontology, and provides examples of how the proposed ontology can be applied to support modeling activities. Finally, we summarize our work and discuss possible directions for future research in Section 4.



### 4.3 METHODOLOGY

In designing the water resources component ontology, we used the widely accepted skeletal methodology described by Uschold and Gruninger (1996), which has been successfully applied for building many ontologies (e.g. Kim, 2005; Brillhante and Robertson, 2001; Patil et al., 2005; Biletskiy et al., 2004; Bermudez and Piasecki, 2006). The approach (summarized in Figure 5.1) begins by first defining the purpose of the ontology and its design requirements. Second, building the ontology is accomplished in three phases: (i) concept capture, (ii) coding, and (iii) integration with complementary ontologies. These steps are necessary to define the concepts used within the community, determine the method of presenting these concepts, and benefit from prior efforts in building related ontologies, respectively. Third, the ontology is evaluated before sharing with the community to ensure consistency of concepts and their relationships. Fourth, documentation of all important assumptions and key concepts is included within the ontology in the form of natural language. The fifth and final step is to describe the guidelines used in building the ontology. The following sections elaborate on these steps, especially on the ontology building phases because these are arguably the most complex and challenging steps in creating an ontology.

#### 4.3.1 PURPOSE AND DESIGN REQUIREMENTS

Defining the ontology purpose, use, and its target audience provides a clear focus in the subsequent building stages (Scholten et al., 2007; Uschold and Gruninger, 1996; Beran and Piasecki, 2009; Athanasiadis et al., 2011). The purpose of the Water Resources Component (WRC) ontology is to promote the interoperability of components across water-related disciplinary boundaries and modeling frameworks. We aim to provide modelers using a component-based modeling approach with a tool that helps them in selecting the correct components to be coupled and aids in minimizing cou-

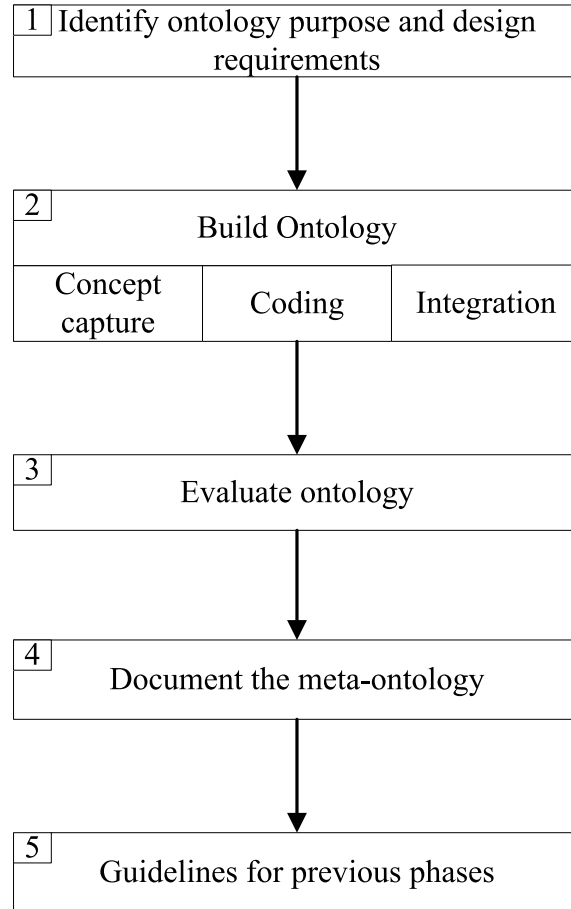


Figure 4.1: Methodology used for development of the Water Resources Component (WRC) ontology, adopted from Uschold and Gruninger (1996)

pling conceptual error. Lastly, we intend for the ontology to be a tool adopted by researchers, educators, and practitioners in water-related disciplines.

## 4.3.2 BUILDING THE ONTOLOGY

### 4.3.2.1 CONCEPT CAPTURE

Uschold and Gruninger (1996) describe the concept capture phase as identifying and defining the basic ideas, relationships, and terms corresponding to a domain. In the WRC ontology, concepts and terminologies were identified from the analysis of domain metadata initiatives and other ontologies developed in related domains. Specifically,

two component-based modeling frameworks, one knowledge management system for water quality modeling, and one web-based hydrodynamic simulation system were analyzed to capture the coupling process concepts. In selecting these examples, the aim was to capture key initiatives in the Earth Science community that directly relate to the purpose and design requirements identified in the prior step of the methodology. Below we summarize each example.

The Earth System Curator (ESC) is a project to develop metadata describing the digital resources used in climate simulations (Dunlap et al., 2008). The aim of the work is to develop a metadata schema that describes numerical climate simulation software as well as their output datasets. Dunlap et al. (2008) defined three tasks of the metadata to describe numerical climate models. First, the metadata has to relate the software and its associated output datasets. Second, it should provide sufficient detail about the degree of technical compatibility between two simulation models. Third, it should describe the climate modeling software interface where the models can be coupled. ESC recognizes six metadata keys to define these three features. These keys are represented as packages using the Unified Modeling Language (UML) to group data elements organized in a hierarchical structure. There are four packages used to describe a model: (i) the configuration package that contains data describing the model arrangement for a simulation, which is required for reproducing simulation runs, (ii) the modeling package for describing the technical, numerical, and scientific model details, (iii) the grid package for defining the spatial metadata used by a model, and (iv) the coupling package for defining technical aspect of model coupling (Table 4.1). The concepts used in the modeling package are heavily based on the Numerical Model Metadata (NMM) schemata and Earth System Modeling Framework (ESMF) metadata. The fifth package describes output datasets, and the sixth package is used for describing general metadata about any resource.

Table 4.1: Basic concepts and properties used to describe models in Earth System Curator (ESC) (Dunlap et al., 2008)

Concept	Properties
Author	Name, organization name, position, telephone, address.
Institution/Agency	Address, telephone, e-mail, fax, hours of service.
Project	Purpose, resource provider, owner, principle investigator.
References	Title, ISBN, edition, publisher, author.
Technical	Platform, programming language, supported compilers.
Scientific	Parameters (name, values used in simulations), model initial conditions.
Numerical	Spatial and temporal discretization of the component, numerical method information.
Interface	Model I/O file (format and name), Model I/O dataset.
Coupling	Exchange data (quantity type, physical units, min/max Value, data relation with the model, field dependency).
Grid	Projection, vertical/horizontal coordination system, geometry, discretization, etc.

The Community Surface Dynamics Modeling System (CSDMS) is a community-based modeling environment established on open source software modules that focus on simulating a wide variety of Earth-surface processes that interact over various time and space resolutions (Syvitski et al., 2004). CSDMS has three principal components: Standard Utilities, Modules, and a Toolkit (Syvitski et al., 2004). Standard Utilities handle data structure, graphics rendering, module connectors, and a web interface (Anderson et al., 2004). We used two sources for capturing concepts from the CSDMS. First, we used a questionnaire that the CSDMS team asks model developers to complete when submitting their model to CSDMS. The information collected from the questionnaire is used primarily to build help documents along with a reference key related to the model that are made available through the CSDMS web site. Second, we used the recently developed Model Metadata File (MMF) that CSDMS requires when models have been componentized to follow their Initialize, Run, Finalize (IRF) standard. Table 4.2 depicts the fields in the questionnaire and MMF that are used to describe a model. Within the MMF file, CSDMS uses a scheme for standardizing names that follows an object + quantity pattern

Table 4.2: Basic concepts and properties used to describe models in the Community Surface Dynamics Modeling System (CSDMS) (Peckham et al., 2012)

Concept	Properties
Modeler	First/last name, type of contact, institute/organization, address, email address, phone, fax.
Summery	Name, synonym, type, future plan.
Identity	Domain, description.
Documentation	Key papers, manual, module forum, comments.
Technical	Platform, programming language, memory requirement, typical runtime, license type, optimal processor, module availability type, source code location.
Softwares	Pre/postprocessing, visualization.
Data File	Type(I/O or calibration), file description, file physical location.
I/O parameter	Parameter description, parameter format.
Process	Description, equation, parameter, time/space constrains, limitations.
Coupling	Interface type, architecture type, framework availability.
Spatial resolution	Spatial dimensions, spatial extent.
Temporal resolution	
Development history and status	

([http://csdms.colorado.edu/wiki/CSDMS\\_Standard\\_Names](http://csdms.colorado.edu/wiki/CSDMS_Standard_Names)). This work to standardize names was not used in the knowledge capture because standardizing names can be viewed as a related but parallel effort to establishing a knowledge framework (Villa et al., 2009).

Chau (2007) introduced an ontology-based knowledge management system for water flow and quality models. It adapts a three level architecture for intelligent decision support, namely the object, application, and description levels. The object level stores knowledge sources and information about models. Users interact with the object level through the application level, which is commonly a Graphical User Interface (GUI). The description level identifies two ontologies: (i) the information ontology, which includes generic concepts and attributes of knowledge sources and information stored in the object level, and (ii) the domain ontology, which includes key concepts and attributes of water quality models and the output flow. Chau (2007)

defined features and conditions to describe water quality models in the domain ontology: (i) features describing the numerical model (e.g. numerical method, scheme, time stepping algorithm, initial conditions and boundary conditions, etc.), (ii) conditions for the model parameters (e.g. discretization method, dimension of influence, and the spatial and temporal conditions of the model parameter). Each feature and condition has one or more possible values, and these values are indexed with a unique identifier. Therefore, a numerical model can be described by a combination of one or more indices.

Islam and Piasecki (2008) introduced an ontology for the data associated with running a hydrodynamic Web Based Simulation (WBS) system. The WBS system depends on communication between the user and the central simulation server via a web browser. Its environment consists of a (i) simulation domain ontology, (ii) code and coding language, (iii) Graphical User Interface (GUI), and (iv) data storage system (Islam and Piasecki, 2008). The simulation ontology focuses on describing the model, as well as the geospatial and hydrodynamic data of the modeling environment. The ontology has three basic concepts: MetadataModel, MetadataModelData, and MetadataGeospatialData. First, the MetadataModel stores information about the model itself (e.g. name, description, start time, end time, etc.). Second, the MetadataModelData includes information about the variables used by the model, including metadata related to the grid, the organization of I/O data, and the flow of the data stream. Third, the MetadataGeospatialData contains the grid, vectors, and rasters. The Grid concept describes the numerical grid contents (nodes and elements) and the relationships of the grid contents with the model data.

From these examples, we extracted a collection of unstructured knowledge information that needed to be organized in an intensional semantic structure, meaning that the information is structured independent of any specific interpretation or situation. There are three approaches for accomplishing this knowledge organization and

restructuring task: top-down, middle-out, or bottom-up (Uschold and Gruninger, 1996). Top-down starts with the highest concept and moves down for details while, logically, bottom-up begins with details and tries to generalize them. The middle-out approach conserves a balance in terms of the level of details and has the advantage over the other approaches that it allows higher level classes to arise naturally (Uschold and Gruninger, 1996; Fernández-López et al., 1997). This middle-out approach starts with the important concepts and then gradually abstracts the higher level concepts; therefore details arise depending on their necessity. For this reason, and based on the recommendation from prior studies (Uschold and Gruninger, 1996; Beran and Piasecki, 2009; Scholten et al., 2007), we elected to use the middle-out approach in this study.

#### 4.3.2.2 CODING THE ONTOLOGY

The process of coding the ontology is where the domain meta-ontology (i.e. information about ontology components) gathered in the concept capture phase is represented in a formal specification using an ontology coding language (Uschold and Gruninger, 1996). The advantage of using a formal language is to establish the information about the concepts and their relationships in the form of axioms. Many ontology coding languages are based on the eXtensible Markup Language (XML) including the Ontology eXchange Language (XOL) and Ontology Markup Language (OML). XML is a flexible, self-describing markup language format designed for data exchange over the World Wide Web (WWW) (Bray et al., 1997). It provides a hierarchical structure for encoding data and its relationships, however the approach can suffer from syntactic heterogeneity because data can be organized in many different ways. To address this shortcoming, ontologies based on Resource Description Framework (RDF) have become popular, for example the Ontology Interface Language (OIL) and Web Ontology Language (OWL). RDF is often described as a simple yet powerful data

model and language for describing Web resources where a “resource” is defined as an object that is uniquely identified by a Uniform Resource Identifier (URI) (Antoniou and Harmelen, 2009; Fensel et al., 2011; Uschold and Gruninger, 1996). RDF is used to represent data as statements about resources where a graph is used to connect resource nodes to their property values with arcs labeled with properties (Abadi et al., 2007). However, RDF by itself does not provide a mechanism for defining internal relationships between properties and concepts (Garrido and Requena, 2011). OWL provides a means for extending RDF with larger vocabularies and stronger syntax for describing properties and concepts in a human and machine-understandable format (Antoniou and Harmelen, 2009; Hitzler et al., 2009). Thus, OWL enables a machine to infer relationships between concepts and make decisions like an expert user.

OWL has been accepted and recommended as an ontology language by the World Wide Web Consortium (W3C). It is XML-based and applies RDF syntax (Antoniou and Harmelen, 2009). We elected to use OWL as the formal ontology coding language for these reasons and because it is currently the most prominent ontology language for the semantic web, and it is compatible with most querying languages (Antoniou and Harmelen, 2009; Hitzler et al., 2009; Garrido and Requena, 2011). OWL provides sophisticated modeling constraints such as explicit cardinalities, universally and existentially quantified property constraints, and class definitions based on the union, intersection, or complement of other classes. These constraints provide a semantically-rich conceptual model with advanced inferencing capabilities (Dunlap et al., 2008).

There are three versions of OWL, each with increasing complexity: OWL-Lite, OWL-Description Logic (OWL-DL), and OWL Full (McGuinness et al., 2004). OWL Full uses the same semantics and syntactics as RDF, however users cannot verify how the machine understands the coded relationships between concepts (Antoniou and Harmelen, 2009). OWL-DL is sublanguage of OWL Full and is compatible with



the Description Logic, which allows the ontology to be changed into a mathematical relationship (e.g. “concept A is a subconcept of B” is interpreted as  $A \subset B$ ). This allows a reasoner to interpret the implicitly coded relationships between concepts into explicit relationships based on the machine understanding, which helps the user to revise the ontology relationships. OWL-Lite is a subset of OWL-DL that excludes some language constructors used in OWL-DL. We chose to use OWL-DL because it provides the needed expressiveness for concepts and relationships, and its consistency can be checked using efficient reasoning in a short time compared to OWL full (Antoniou and Harmelen, 2009; McGuinness et al., 2004).

Some of the available tools for creating OWL documents are OntoEdit, OilEd, and Protégé. The latter is developed in Java and is an open source, user friendly, and extensible software system. Protégé also enables an ontology structure to be defined by automatically generating forms that facilitate knowledge acquisition. While the other tools considered for this work, OntoEdit and OilEd, offer powerful ontological formats, we concluded that they lack the comprehensiveness and user-friendly features of Protégé. Therefore we elected to use the Protégé (version 4.2 alpha) which supports OWL (version 1) for our work.

An OWL ontology consists of namespaces, classes, properties, and individuals. Namespaces are the first component declared in the ontology document and are used to identify the resources of the standard used to define the ontology and its elements. All namespaces used by the ontology are declared at the beginning of the ontology document. Ontology elements are prefixed with a tag referring to a namespace to declare that the element resides in this namespace. Classes define the concept that groups individuals (instances) that have common properties. Classes can be organized in a specific hierarchy using the “Is-A” relationship. For example, the Independent Variable class “Is-A” sub-class of the Variable class. In this relationship, the Variable class is called a superclass. Properties can be one of two types: an object property,

which describes the relationship between classes, or a data property, which defines a data type and value of an instance. Individuals, as stated before, are instances of a class. Property restrictions are used to subset a group of individuals from one class into a new sub-class in which its members participate in a specific relationship.

#### 4.3.2.3 INTEGRATION WITH EXISTING ONTOLOGIES

Ontologies are built to be reused in different applications (Fernández-López et al., 1997). One common way ontologies are reused is through integration with other ontologies. Not only does this help in speeding up ontology construction, but it also allows reuse of definitions and properties already built and tested, providing consistency and robustness to new ontologies. However, it is necessary to check that the meta-ontology of the imported ontology corresponds to the design needs and that the imported ontology provides similar semantics before implementing the concepts within the new ontology (Fernández-López et al., 1997).

An example of ontology reuse is describing measurement units. Units, of course, are a fundamental concept in the water resources modeling. They are used to define variables, parameters, and universal constants, and are therefore necessary metadata for exchanging quantities between components. The Semantic Web for Earth and Environmental Terminology (SWEET) is a project developed to create a knowledge base for improving the shared semantic understanding of Earth Science data (Raskin and Pan, 2005). It provides a higher level representation of keywords used in Earth Sciences through a collection of formal ontologies coded in OWL. One of the ontologies encoded within SWEET is a units ontology (SWEET, <http://sweet.jpl.nasa.gov/1.1/units.owl>) that describes the hierarchical structure of units as depicted in Figure 5.2. Concepts, structure, and relationships are derived from the Unidata's library created by the University Corporation for Atmospheric Research (UCAR) to support manipulation of physical quantity units. The SWEET unit

ontology satisfies our concepts about units because it includes an extensive database of units, definitions for prefixed units, and conversion factors between compatible units (Steward et al., 2009). In addition, it is extendable to reflect units used in other scientific domains. Therefore the SWEET unit ontology was integrated into the WRC ontology for describing units.

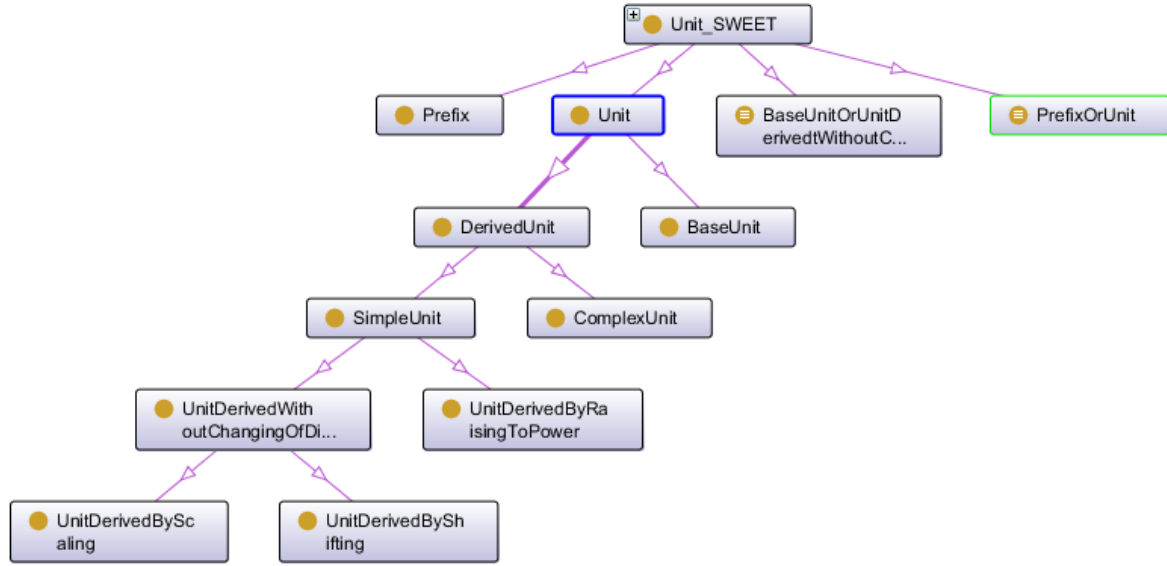


Figure 4.2: The hierarchical structure of the Semantic Web for Earth and Environmental Terminology (SWEET) Unit ontology, which integrates with the Water Resources Component (WRC) ontology to describe symbols units.

#### 4.3.3 EVALUATION, DOCUMENTATION, AND GUIDELINES

Evaluation of the ontology includes both validation and verification (Gómez-Pérez et al., 1995). Validation guarantees that the ontology definitions correspond to the system that they are representing. The validation also provides information about whether the ontology definitions are sufficient and necessary (Gómez-Pérez et al., 1995). Verification ensures that the structure of the ontology is built correctly. It includes individual definitions and axioms, imported definitions from other ontologies, and the inferred axioms from the collection of definition and explicitly mentioned

axioms. We used Pellet as a reasoner for ontology evaluation because it is the most common reasoning engine used with Protégé OWL and because it is compatible with the DL system. Sirin et al. (2007) and Konstantinou et al. (2008) summarized the advantages of using the Pellet reasoner including its ability to (i) check the standard reasoning services such as cardinality restrictions, complex subproperty axioms, and user-defined datatypes, (ii) ensure consistency of an ontology axioms, (iii) compute for each named class the expected sub-class relationship to create the complete class hierarchy, (iv) infer the most specific classes that an individual belongs to, and (v) determine the possibility of a class to accept instances. Documentation of the meta-ontology facilitates the ontology handling among community members and it guides users in the updating process. Finally, documentation of the WRC ontology is straight forward and uses natural languages through the facilities provided by Protégé. Attached with each class in the WRC is a comment defining the underlying concepts, assumptions, and relationships, along with a reference to the concept source if the source is available.

## 4.4 RESULTS AND DISCUSSION

### 4.4.1 WRC ONTOLOGY

An overview of the WRC ontology developed using the methodology described in the previous section is presented in Figure 4.3 and is available at <http://grg.engr.sc.edu/wrc/0.1/WRC.owl>. The analysis accomplished in the concept capture phase results in identifying eighteen superclasses to describe a model component. The superclasses used to describe a component are grouped into four ontological layers: resources, scientific, coupling, and technical. In this section we describe these four layers of the WRC ontology including their class hierarchy, relationships, axioms, and restrictions. We organized the concepts into these four layers

because we wanted to center the ontology around the component concept. Other layer groupings, for example one that focused on model engines and model instances, could also be used to describe the ontology, and using a different layer grouping would not result in any major changes to the ontology concepts themselves.

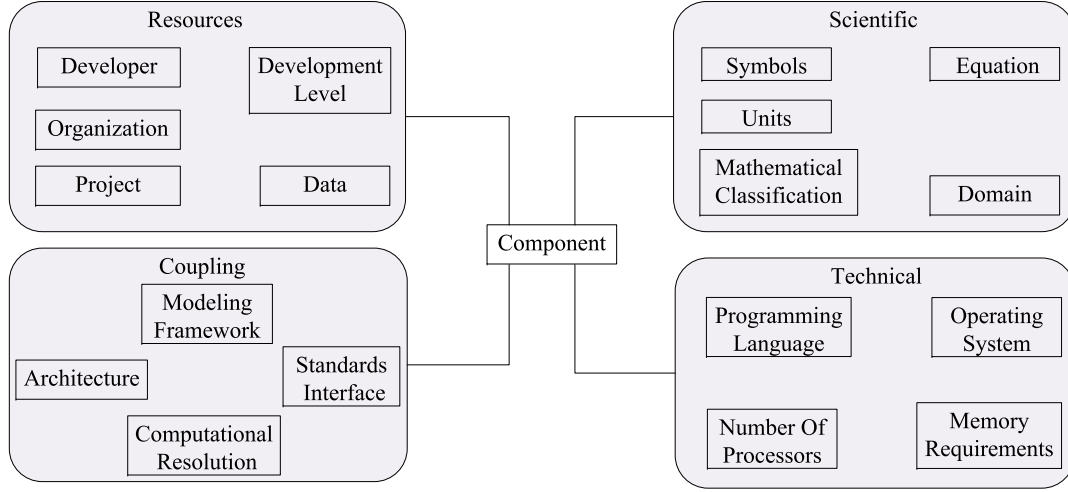


Figure 4.3: An overview of the Water Resources Component (WRC) ontology, describing the basic four layers and the superclasses of each layer.

#### 4.4.1.1 RESOURCES LAYER

The Resources layer has five superclasses that are collectively used to describe the component’s digital resources (Figure 5.3). The *Development Level* class defines the component’s development stage based on a four level development scheme proposed by Argent (2004). These development levels range from Level I, signifying a model that is developed for research purposes, to Level IV, signifying a model used in planning and policy analysis. The *Organization* class identifies the agency or institute where the component is developed. Currently it consists of two sub-classes, *University* and *Company*, but can easily be expanded to include other subclasses. The *Organization* class is related to the *Developer* class that stores information about the component’s development team. Although not shown in Figure 5.3 due to space restrictions,

both the *Organization* and the *Developer* classes include properties adopted from the Observation Data Model (ODM) (Horsburgh et al., 2008).

The *Data* class is divided into two sub-classes: *Data File* and *Data Value*. The *Data File* has four sub-classes: Geospatial, Tabular, TimeSeries, and Extensible Markup Language data. The *Data Value* class stores the numerical or categorial values used by the component. The relationship between the *Component* and *Data* classes can be one of three types: input, output, or associated data. Examples of associated data include model parameters or source code files. Identifying these existing data resources, and describing the exact format of the data document, could also enable components to utilize remote data sources in an automated manner. The *Project* class is used to define information about projects in which components are coupled to form a workflow. When a component is a part of a modeling workflow, it is also necessary to know where and how it is used, within that project including any specific project requirements.

In addition to the class definitions themselves, another important feature of the WRC ontology expressed in Figure 5.3 are the relationships between classes. For example, the many-to-many relationship between individuals of the *Component* class and those of the *Developer* class means that every component must have one or more developers, and that each developer must develop at least one component. Also, because the component's development level classification will change during the component's life cycle, we use Sufficient and Necessary conditions to classify components based on their development level. Sufficient and Necessary conditions mean that, if an individual is a member of a class, then it must satisfy specific conditions, and likewise if any individual satisfies these conditions, then it must be a class member (Motik et al., 2009). A data property is assigned to each individual of the component class to define its current development status. Each sub-class of the *Development Level* class uses a Sufficient and Necessary condition to capture

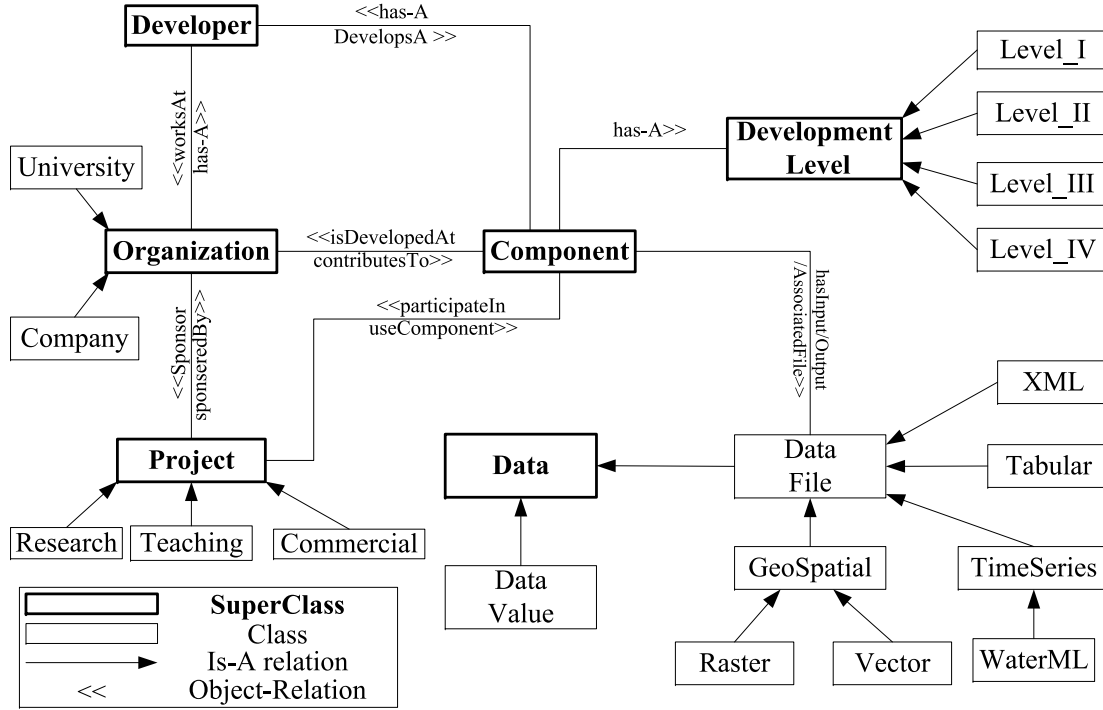


Figure 4.4: Resource layer of the Water Resources Component (WRC) ontology, including classes and relationships.

components with corresponding development status. For example, the condition for class Level-I states that each component that has a development status value equal to Level-I is considered a member of *Development Level-I* class. The Pellet reasoner will capture the new classification of components after any updates are made in the component properties.

#### 4.4.1.2 COUPLING LAYER

The Coupling Layer is designed to answer three questions about component coupling: What are the coupling standards used by the component? In which frameworks can the components be coupled? What is the computational resolution of the component? The Coupling Layer uses four classes to address these questions: (i) *Modeling Framework*, (ii) *Standards Interface*, (iii) *Architecture* and (iv) *Computational Resolution* (Figure 5.4). The role of a *Modeling Framework* is to provide an environment

for components to be coupled. In component-based modeling, a *Modeling Framework* couples components that adopt a specific *Standards Interface* and *Architecture*. A component can be used within a *Modeling Framework* if the *Standards Interface* and *Architecture* used are consistent with those supported by the *Modeling Framework*. A *Modeling Framework* is classified into two sub-classes based on the allowed level of interaction between components: (i) *Concurrent* is where the framework allows components to communicate during the time horizon of the simulation and (ii) *Sequential* is where the framework allows components to communicate after the conclusion of the simulation time horizon.

The *Computational Resolution* class covers both the temporal and spatial resolution of the component model. The *Temporal Resolution* class introduces the order of permissible operating time steps of the component to keep it numerical stability. The *Spatial Resolution* class is based on the CSDMS definition of model space resolution and includes two sub-classes: (1) the *Spatial Extent* defines the smallest/largest grid size that the component can operate on while maintaining its numerical stability and (2) the *Spatial Dimension* specifies the model's ability to be solved in different dimensions (e.g. 1D, 2D). We expanded this to also include the Chau (2007) definition of the plane of dimension for the computational models (e.g. 2D vertical). We use these definitions as individuals in the *Spatial Dimension* class. The computation resolution (temporal and spatial) min/max values are defined using a data type property in the OWL document attached to each individual component.

#### 4.4.1.3 SCIENTIFIC LAYER

The objective of the Scientific Layer is to describe the component's equations, I/O variables, parameters, purpose, and mathematical classification. The *Equation* class stores all the equations used by the component using the Math Markup Language version 3.0 (MathML, <http://www.w3.org/TR/MathML3/>) as a means for describ-



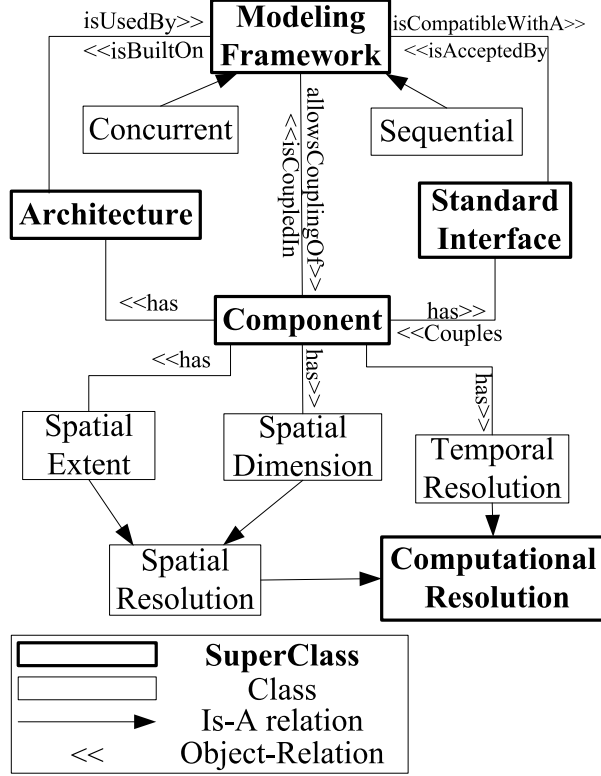


Figure 4.5: Coupling layer of the Water Resources Component (WRC) ontology, including classes and relationships.

ing mathematical notation and capturing both its structure and content (Carlisle et al., 2009). The information required to apply the equation is represented in the *Assumption*, *Initial Condition*, *Boundary Condition*, *Equation Type*, and *Numerical Simulation* classes, which are sub-classes of the *Equation* (Figure 5.6). The first three sub-classes define, as their names imply, the assumptions, initial conditions, and boundary conditions applied to the equation for use in the model. Individuals of these sub-classes are stored as mathematical equations expressed in MathML that a machine can interpret and understand.

As an example of how model assumptions can be expressed as mathematical relationships within the ontology, consider the case of a model that assumes nonaccelerating flow. This assumption can be expressed mathematically as  $\delta v / \delta t = 0$  and is linked to the component model through the *hasAssumptions* relationship established

by the WRC ontology. Using this relationship, a model coupling framework could be designed to ensure that model assumptions are not violated within the framework. In the case of this example, the modeling framework would check that flow velocity within the model component does not change over the simulation time period. By expressing model assumptions within the ontology, model developers are able to focus on the model code itself rather than implementing assumptions and error checking routines that can be handled by a modeling framework and shared across modeling components.

The *Equation Type* is divided into three sub-classes covering all equation types, *Algebraic*, *Differential*, and *Integral*. The *Differential* class has sub-classes describing the types and orders of differential equations (*Ordinary* or *Partial*). The *Numerical Simulation* class contains (i) the *Numerical Technique* describing the method used to discretize the differential equation in space (e.g. backward, forward, etc.) and (ii) the *Time Difference Scheme* defining how the differential numerical equation is discretized with respect to time (e.g. explicit, implicit, semi-implicit).

The symbols used by the equations and components are grouped into the *Symbol* class, which classifies symbols into *Variable*, *Parameter*, or *Constant* classes. Each symbol must have a unique and unambiguous name and we anticipate that experts of each subdiscipline will use well known sources for standard names to define a symbol. For example, the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI, <http://www.cuahsi.org>) and its Hydrologic Information System (HIS, <http://www.his.cuahsi.org>) have developed an ontology to support the discovery of time-series hydrologic observations including physical, chemical, and biological measurements (Beran and Piasecki, 2009) that could be used within the WRC ontology. The standard names in development within the CSDMS community could be used to describe I/O variables and assumptions related to a component model. Finally, the standard names defined by the NetCDF Climate and Forecast (CF) Meta-



both the equation and variable are linked and point to the same component. One of the advantages of this approach is that it minimizes the semantic heterogeneity between components by providing a more complete description of variables (Figure 5.5b). For example, in the figure the two equations, Darcy’s equation and a advective contaminant transport equation, have a common variable that would allow for coupling the two models. However, the variable has a different symbol within the two model components so it may not be obvious to the framework that the two models can be coupled. If both symbols are related to the same variable concept, it allows the framework to understand that the two components can be coupled.

The design purpose of the component model is represented by the *Water Resources Domain*, which contains classes describing the water resources sub-systems such as the *Hydrology* and *Ground Water* classes (Figure 5.6). Each sub-system class has sub-classes representing the basic processes of the system. For example, the *Evapotranspiration* class “Is-A” *Hydrology* class which is “Is-A” *Water Resources Domain* class. A complete hierarchical structure of the classes representing the water resources sub-systems is not defined in this ontology because it would need to be provided through a collaborative effort that included experts across a variety of water resources domains. Finally, the *Mathematical Classification* class is used to define how the variables are treated in space and time, and if they are deterministic or stochastic as described by Chow et al. (1988) (Figure 5.6). Therefore the *Mathematical Classification* class is divided into two sub-classes: *Deterministic* and *Stochastic* (Figure 5.6).

#### 4.4.1.4 TECHNICAL LAYER

The Technical Layer answers questions about the required computer architecture that enables a user to (i) run a component simulation, (ii) edit or update the component code, (iii) determine the computational resources required by the component, and

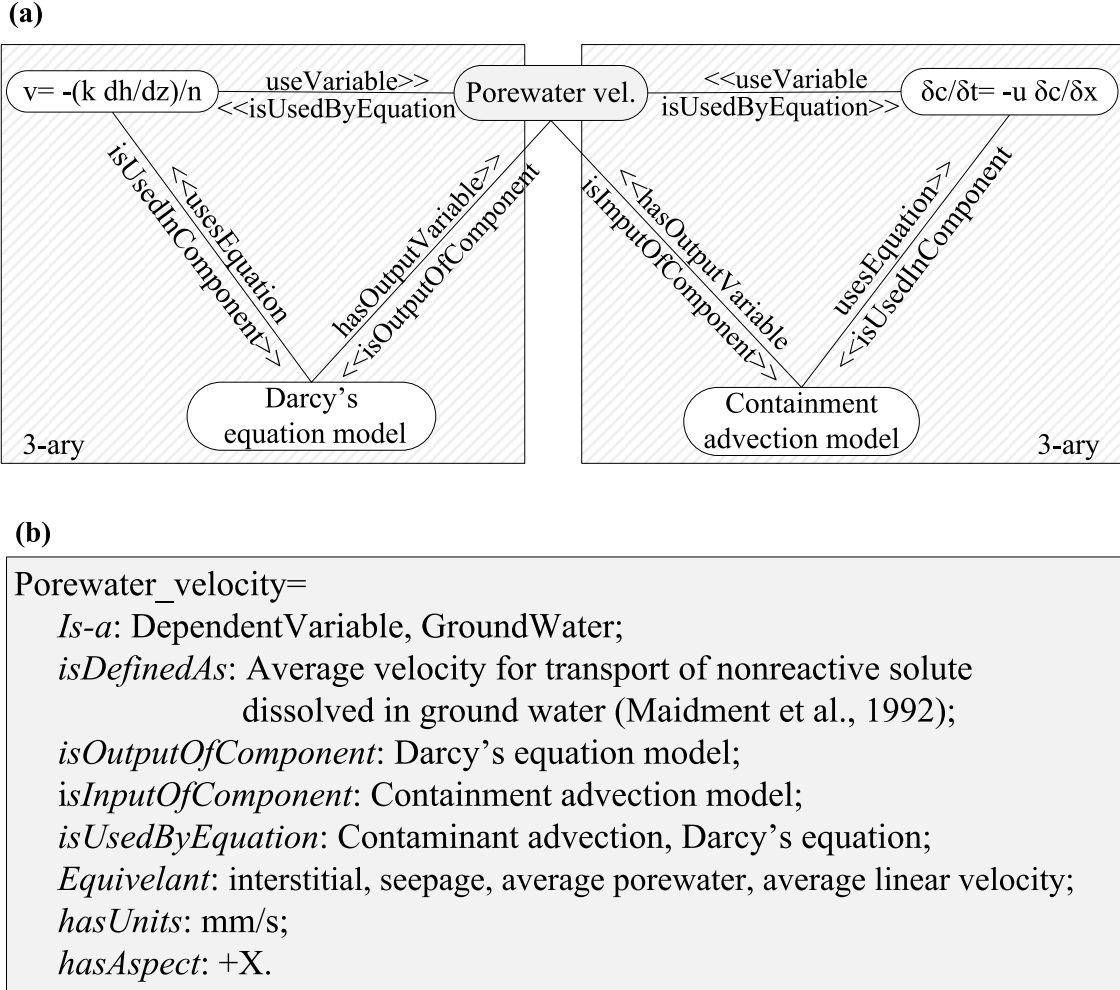


Figure 4.7: (a) Excerpt from the WRC ontology illustrating the 3-ary relationship between a component, equation, and variable. (b) Description of the Porewater Velocity variable, which is passed between the two coupled model components, using the WRC ontology.

(iv) optimize the simulation time given available computational resources. This layer is based on the metadata used by the CSDMS to describe the technical requirements of a model and includes four classes (Figure 5.7). The *Operating System* class defines the different operating systems that are compatible with the component. The *Programming Language* class determines the language used in writing the component. This helps the user to know how to edit or update the component code. The *Memory Required* class is used to describe the required memory capacity for supporting a sin-

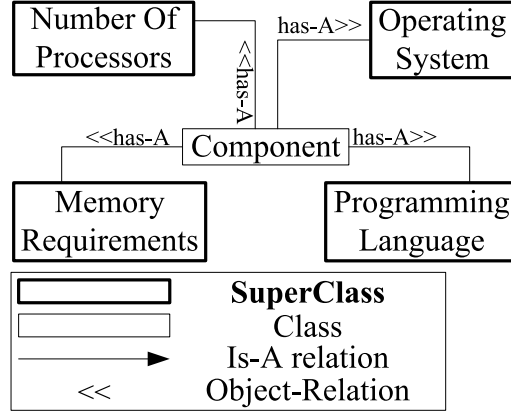


Figure 4.8: Technical layer of the Water Resources Component (WRC) ontology, including classes and relationships.

gle component simulation. Finally, the *Number Of Processors* class includes elements representing the number of processors that the component is able to leverage.

#### 4.4.2 EXAMPLE APPLICATIONS

Three simple example applications are presented to illustrate the capabilities of the WRC ontology for aiding in component-based water resources modeling. The examples assume a hypothetical case where a user wants to estimate stream discharge through the outlet of a given study watershed. We further assume that the user has access to hourly temperature and rainfall observations for the study region, as well as typical watershed geospatial data (e.g., DEM, soil properties, and land cover). Lastly, we assume that the user wishes to only model evapotranspiration and runoff processes for the purposes of the study. Given these assumptions, the following example applications demonstrate how the ontology can be used to guide the user in (i) identifying appropriate model components, (ii) correctly linking model components into a workflow, and (iii) utilizing model components from a discipline other than the modeler’s primary discipline.

#### 4.4.2.1 MODEL COMPONENT SELECTION

Searching for an appropriate model can be a difficult task, especially when using a component-based modeling framework that may contain models which are not in the core area of expertise of the model user. The WRC ontology can assist in guiding users to the appropriate model component by relating needs expressed by the user with concepts expressed in the ontology. In the hypothetical scenario explained earlier, the user wishes to identify two model components: one that computes evapotranspiration and a second that computes runoff. In relation to the ontology, we can say that the evapotranspiration component must meet the following criteria: (i) the *hasPurpose* property points to the *Evapotranspiration* class, (ii) the *hasInputVariable* property links the component to the temperature variable only, and (iii) the *hasOutputVariable* property points to variables that have *isUsedAsInput* property with runoff components.

Figure 4.9 shows the coded information and relationships for three components that meet the first criterion (*hasPurpose=Evapotranspiration*): Penman-Monteith, Priestley-Taylor, and Hargreaves-Sarnani. While all three of these components can be used to calculate evapotranspiration, they use different inputs as shown in Figure 4.9. Taking the second criterion into account (*hasInputVariable = temperature*), the WRC ontology is used to further filter the list of possible components to just one component: Hargreaves-Sarnani. Finally, the third criterion is used to check against the ontology that the Hargreaves-Sarnani can be coupled with a runoff model, based on the compatibility of the data exchanges between the two components (i.e. that the input of each runoff component is the output of the evapotranspiration component) and their coupling attributes (i.e. components use the same standard interface).

The second component required by the user, the runoff component, has to satisfy three criteria: (i) the *hasInputVariable* property points to two variables only, evapotranspiration and precipitation rates, (ii) this evapotranspiration rate is the *ha-*

*sOutputVariable* of one of the previously named evapotranspiration components, and (iii) the *hasInputData* property is linked to the typical watershed data (DEM, soil properties, and land use). In this example, the two components used to calculate runoff are (i) a component that leverages the Green-Ampt Model for infiltration and (ii) a component that implements TOPMODEL (Beven and Beven, 2001). The WRC ontology filters the runoff component list and selects the Green-Ampt component because it meets the two criterion and uses Potential Evapotranspiration (PET) rate as the input variable, which can be supplied by the Hargreaves-Sarnani component. The TOPMODEL component, on the other hand, only accepts standardized reference Evapotranspiration ( $ET_{sz}$ ) as input (Figure 4.9).

Filtering of components based on a set of criteria like this can be easily automated by designing a software application that translates the user requirements into ontology querying statements, such as the questionnaire provided by Chau (2007) for guiding users to a recommended water quality model. Also, this example can be extended to highlight the advantages of the WRC ontology in overcoming search challenges such as inconsistent semantics used by different modeling communities. For example, suppose users from two different disciplines use different terms – “stream outflow” and “stream discharge” – to search for a component that computes the same variable. In the WRC ontology, semantic mediation can occur by having components that calculate either “stream discharge” or “stream outflow” grouped together because they both represent the same concept within the ontology. Therefore the problem of differences in semantics between disciplines that may limit more basic text matching mechanisms for search can be overcoming using the ontology (Grossman and Frieder, 2004). Ideally, the definition of variables in WRC ontology will leverage and extend when necessary other initiatives for defining hydrologic variables including work under the Consortium of Universities for the Advancement in Hydrologic



Science, Inc. (CUAHSI) Hydrologic Information System (HIS) project (e.g. Piasecki and Bermudez, 2003; Beran and Piasecki, 2009; Horsburgh et al., 2009).

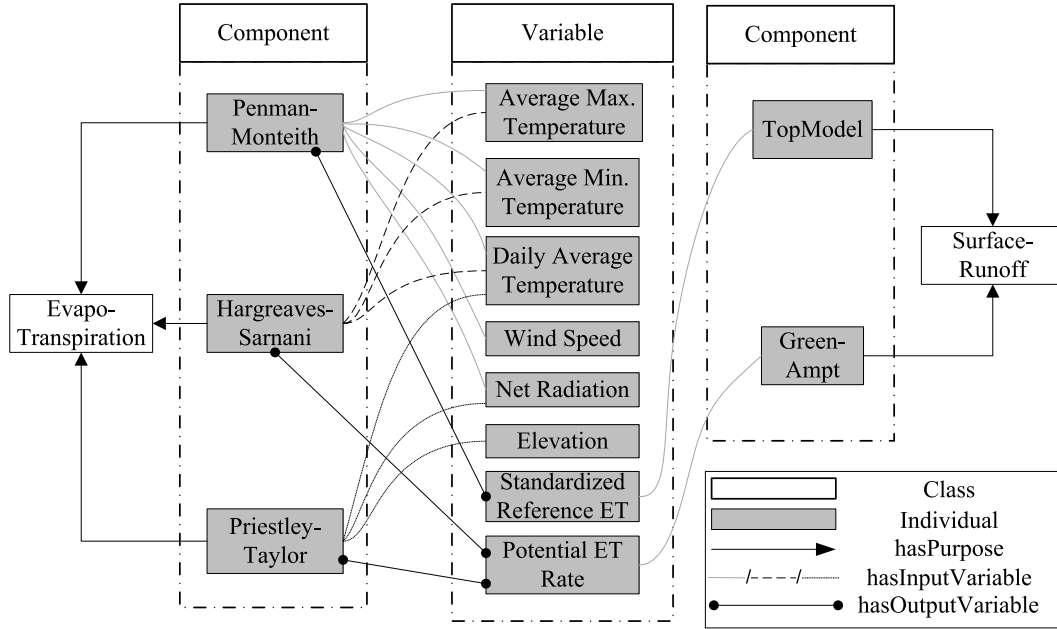


Figure 4.9: Excerpt from the WRC ontology illustrating the stored information about three evapotranspiration interchangeable components. This information can help users in the process of selecting the appropriate component to represent the evapotranspiration process for a given study.

#### 4.4.2.2 COMPONENT COUPLING CONSISTENCY

One of the challenging aspects of component-based modeling is that, while coupling two components may be technically feasible, it may not be conceptually correct. Examples of such situations are presented by Voinov and Shugart (2012) and include the potential of temporal mismatched scales between coupled components. The axioms in the WRC ontology can be used to automate the compatibility check between components proposed by the user to be coupled. This facilitates the coupling process, minimizes conceptual errors, and encourages proper use of both models and data.

To extend our scenario, suppose that the user wishes to couple the Hargreaves-Sarnani component with the TopModel component due to the topographic features

of the watershed. As we discussed in the prior scenario, this coupling would not be correct because of inconsistencies in the data exchanges between the two components, namely that the Hargreaves-Sarnani component supplies PET and the TopModel component accepts  $ET_{sz}$ . This is a subtle difference that may not be obvious to a nonexpert. The WRC ontology can be used to avoid such errors by checking the consistency of technical, coupling, and scientific features of these components inform the user about inconsistencies, and even suggest approaches for overcoming inconsistencies.

To do this in a general sense, the software would perform two checks. The first check would be at a broad level to test the technical and coupling feasibility of the two components based on the information stored in the ontological layers. In the case of incompatibility, the check would raise an error and the ontology would be used to provide the user with the reason for the coupling error. The second check would be at a fine level to test the scientific properties of the linked components in order to verify that both components are compatible in terms of the space and time context, as well as the exchange of variables. In the case of inconsistency, a recommendation could be offered to the user such as changing the time step of one component or providing the required unit conversion factor between the two variables.

In the example scenario where the Hargreaves-Sarnani cannot be coupled with the TopModel, either the Hargreaves-Sarnani component must be modified so that it outputs  $ET_{sz}$  or the TopModel component must be modified so that it inputs PET. A third option for overcoming the inconsistency would be to create a translator between the two components that simply multiplies the  $ET_{sz}$  by a crop coefficient ( $K_c$ ) to estimate the PET, thereby adapting the output of the Hargreaves-Sarnani component to make it consistent with the required input of the TopModel component. The relationship between  $ET_{sz}$  and ( $K_c$ ) to estimate the PET can be expressed in the WRC ontology itself as depicted in Figure 4.10.

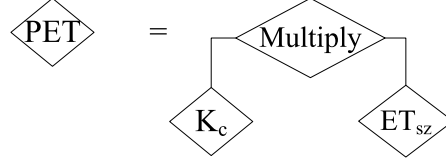


Figure 4.10: Ontological representation of the relationship between two variables, the Potential Evapotranspiration (PET) rate and the standardized reference Evapotranspiration ( $ET_{sz}$ ), in the Water Resources Component (WRC) ontology, and the conversion factor between them.

#### 4.4.2.3 MULTIDISCIPLINARY MODELING

One of the objectives of the WRC ontology is to enhance the component metadata interoperability between modeling frameworks. As an example of this, the WRC ontology could be used to overcome semantic and syntactic heterogeneity between metadata schemas used by different frameworks. Metadata interoperability aids the user in correctly applying a component from a modeling framework used by a different scientific community because the metadata of the component for the unfamiliar framework can be mapped to the ontology used by the familiar modeling framework.

To illustrate this concept, we focused on two modeling frameworks in the following example. The first, HydroModeler, is a plug-in application of the CUAHSI-HIS that extends the core HydroDesktop application to support model integration using the component-based approach (Castronova and Goodall, 2010). It uses the Open Modeling Interface (OpenMI) standard and OpenMI Association Technical Committee (OATC) Software Development Kit (SDK) to provide a “plug-and-play” modeling framework within HydroDesktop. The second, Community Surface Dynamics Modeling System (CSDMS), uses the Common Component Architecture (CCA) to provide a plug-and-play environment for components (Peckham et al., 2012). Both frameworks use an XML schema that describes the component’s exchange variables metadata (Peckham et al., 2012; Castronova and Goodall, 2010), but with different syntactic structure.

As an extension to our scenario, suppose that the user finds a component model in the CSDMS framework that calculates runoff based on infiltration model that uses Richard’s Equation. The user is interested in including this component in a HydroModeler application, but the user is not familiar with the CSDMS and wishes to implement the same component in the HydroModeler framework. Using the metadata available in CSDMS framework, the user is able to populate the WRC ontology for the component. This allows the user to visualize the model metadata (e.g. equations, assumptions, initial conditions, etc.) in a way that is consistent with how native HydroModeler components are visualized. Eventually an ontology could be used to express complete models as abstract entities away from the technical means for implementation of the model as a software tool. Scientists would then express models as an ontology that could be implemented across modeling frameworks using software to automate the process of translating models expressed as ontologies into software components for specific modeling frameworks.

## 4.5 SUMMARY

A well designed and broadly accepted ontology is needed to support current trends toward multidisciplinary, community-focused modeling of water resource systems. If such an ontology can be established, it has the potential to elevate component-based water resources modeling to a level of abstraction where modeling becomes a knowledge representation processes. It will be possible then to use reasoners to automate classification, comparison, and search of models and their related elements. Such functionality is needed to support proper use of models both within and across disciplinary boundaries.

We have taken steps along this path by creating the WRC ontology presented in this paper. The ontology attempts to serve as a knowledge-level specification for the

joint conceptualization used in defining model components across disciplinary boundaries and frameworks. In creating the WRC ontology, we have seen that experts view components from different perspectives. The WRC ontology is an attempt to provide the environment for representing these perspectives as layers defining component resources, coupling, scientific, and technical information using a formal ontology language (OWL). The ontology defines concepts associated with the component in an explicit format that is readable for both the user and the system.

In establishing the WRC ontology, our goal was to satisfy the five primary characteristics of a robust ontology defined by Gruber (1993): clarity, coherence, extendability, flexibility to merge, ability to overcome semantic heterogeneity. We described in this paper our approach for achieving these goals including (i) introducing examples to illustrate the classes and relationships definitions, (ii) checking the consistency of concepts and relationships using the pellet reasoner, (iii) increasing the WRC ontology extendability by minimizing the ontology commitment and encoding bias as recommended by Gruber (1993), (iv) merging with existing ontologies (e.g. SWEET Unit ontology), (v) emphasizing through selected example applications how WRC ontology can help users in overcoming the semantic heterogeneity between water resources related modeling communities. However, we acknowledge that our work is only a first step in building a robust model ontology to support component-based modeling within the water resources communities. It will not be possible to conclude that the WRC ontology has satisfied all five of these characteristics (e.g. “clear structure”) until the ontology has gone through additional testing from a wider set of users.

While the ontology will no doubt benefit heavily from additional use case testing and from seeking community input from a wide set of potential users, this input will not result in a static and perfect ontology. Ontologies are like models in that, even when an ontology is agreed upon, revisions will be required to compensate

for unforeseen conditions and new knowledge. As the underlying conceptual model evolves, new versions of the ontology can be released, and presumably the ontology will grow to become an increasingly robust means for representing component-based water resources models. Therefore, while we are certain that this proposed ontology will be revised in the future, we believe that this revision process must begin with a beginning ontology, which we have provided here.

Future work should focus on conducting further tests of the ontology including a broad set of users and through incorporation into multiple component-based modeling systems. Ways that the ontology can be incorporated into such software systems is one area of future work and potential paths forward include: (i) developing software to guide users' in the model building process by interactive questionnaires, especially when using components outside of the user's core domain of expertise, (ii) automating the consistency check process of coupling components in different modeling frameworks, and (iii) identifying existing sources of data, understanding their composition, and automating data processing steps. In general, we see great potential in combining data and model ontologies in order to improve the connection between data resources and required model input datasets. Also, using ontologies can provide intelligence to modeling frameworks by providing a description of information in a way that enables the framework to search, discover, and couple components. Work in this area will benefit hydrology by aiding in the tasks required for multidisciplinary, community-based studies.

## CHAPTER 5

# TOWARD A HYDROLOGIC PROCESS KNOWLEDGE-BASE: A CASE STUDY FOR THE INFILTRATION PROCESS <sup>1</sup>

### 5.1 ABSTRACT

Hydrologic modeling often requires the re-use and integration of models from different disciplines to simulate complex environmental systems. Component-based modeling introduces a flexible approach for integrating physical-based processes across disciplinary boundaries. Several hydrologic-related modeling communities have adopted the component-based approach for simulating complex physical systems by integrating model components across disciplinary boundaries in a workflow. However, it is not always straightforward to create these interdisciplinary models due to the lack of sufficient knowledge about a hydrologic process. This shortcoming is a result of using informal methods for organizing and sharing information about a hydrologic process. A knowledge-based ontology provides such standards and is considered the ideal approach for overcoming this challenge. The aims of this paper are to present the methodology used in analyzing the basic hydrologic domain in order to identify hydrologic processes, the ontology itself, and how the proposed ontology is integrated with the Water Resources Component (WRC) ontology. The proposed ontology standardizes the definitions of a hydrologic process, the relationships between hydrologic processes, and their associated scientific equations. The objective of the proposed

---

<sup>1</sup>Mostafa M. Elag, Jonathan L. Goodall. In Preparation for submission to *Environmental Modelling & Software*.

Hydrologic Process (HP) Ontology is to advance the idea of creating a unified knowledge framework for components' metadata by introducing a domain-level ontology for hydrologic processes. The HP ontology is a step toward an explicit and robust domain knowledge framework that can be evolved through the contribution of domain users. Analysis of the hydrologic domain is accomplished using the Formal Concept Approach (FCA), in which the infiltration process, an important hydrologic process, is examined. Two infiltration methods, the Green-Ampt and Philip's methods, were used to demonstrate the implementation of information in the HP ontology. Furthermore, a SPARQL service is provided for semantic-based querying of the ontology. Finally, we demonstrate how the coded knowledge about a process can be leveraged and used to generate computer code for a hydrologic process.

## 5.2 INTRODUCTION

Multidisciplinary modeling approaches have become more common in simulating hydrologic systems to overcome the complexities that exist due to interlinked physical subsystems interacting over various spatial and temporal scales (Rizzoli and Young, 1997; Wagener et al., 2010; Argent, 2004). A component-based approach is a modeling paradigm that enables integration of systems' sub-models and maintains their extendibility, flexibility, and reusability across community boundaries (Argent, 2004). This paradigm encourages multidisciplinary collaboration among scientific communities, and requires developing of modeling frameworks that enable models to be coupled (Syvitski et al., 2011; Goodall et al., 2011; Castronova and Goodall, 2010). The component-based modeling approach supports the increased sophistication of models. For example, Elag et al. (2011) used this paradigm for modeling feedback loops in hydrologic domains, and Peckham et al. (2012) described the paradigm for modeling of coupled Earth-surface processes. A component-based paradigm supports



process-based modeling, where a component may execute a single equation or a series of equations to simulate a physical process. These model components can be coupled together in a workflow to create a comprehensive model.

Earth Science modeling communities have largely focused on the conceptual and algorithmic design of components (Leavesley et al., 1996; Argent, 2004; Peckham et al., 2012; Goodall et al., 2011), as well as using metadata to increase the semantic interoperability of models across modeling framework boundaries (Argent, 2004; Voinov and Shugart, 2012). In the context of information integration, ontologies specify and organize the concepts and terminologies used to describe a specific domain (Gruber, 1993). Establishing a shared understanding of concepts aids in eliminating conceptual and terminological confusion for both humans and machines (Beran and Piasecki, 2009; Uschold and Gruninger, 1996).

In a prior work the Water Resources Component (WRC) ontology was created to promote the interoperability of components across water-related disciplinary boundaries and modeling frameworks (Elag and Goodall, In Revision). The WRC ontology is an upper-level ontology that focuses on unifying the concepts used in describing water-resources component models' metadata. It minimizes the semantic heterogeneity across water related disciplines as well as syntactic heterogeneity in the metadata structure used to describe a component across modeling frameworks. While there are benefits to using the WRC ontology to increase the interoperability among water related domains, there are also challenges that must be overcome in order to encourage the broad adoption of this approach. One of these challenges is the absence of an agreement on the description of physical processes in the water resources domain (Argent et al., 2006; Jakeman and Letcher, 2003). Specifically, there is no formal definition of equations that should be coded in every component. There has been an increasing level of duplication in process representations among different modeling frameworks, especially for those parts that include widely-used equations

(Argent et al., 2006). Furthermore, these problems become more complicated when researchers use models from various disciplines, or when the user is not familiar with the scientific basis underlying a given model. As a result, users spend time and effort answering these questions instead of creating state-of-the-art models and executing scientific analysis. Thus, it is essential to standardize the knowledge regarding hydrologic processes.

To address these challenges, we are proposing a domain-level ontology that explicitly describes hydrologic processes and their attributes. The definition of a process must include its limits, input and output, equations, and the numerical representation of each equation. In addition, the process definition should present the available numerical representation of the process within a well-tested comprehensive model that can be used to create a robust component. This will lead to a shared understanding regarding semantically consistent communication within and among users of component-based modeling when applied to hydrologic systems.

This work seeks to introduce a formal definition of the hydrologic domain processes and their interrelationships, corresponding equations and attributes, by creating domain-level ontology. In addition, the proposed HP ontology is integrated with an upper-level ontology, WRC ontology, as a step to promote the interoperability of components across the water-related disciplines. The HP ontology created through this work will (i) expand on the Water Resource Component ontology that was introduced in a previous study (Elag and Goodall, In Revision), (ii) ensure a common understanding among domain users about hydrologic modeling processes and their equations represented within components, (iii) facilitate the process of component creation by guiding the user to the scientific information and tested codes, and (iv) support the automation of creating model source code in various modeling frameworks.

The remaining sections of the paper are organized as follows. Section two discusses the background theory on hydrologic process based modeling and ontologies. Section three describes the design concept used for creating this ontology. Section four gives an overview of the ontology building process and describes some of its features. Section five discusses the implementation of the ontology and gives examples of the processes' information implementation in the ontology. Section six describes the role of ontology in enhancing the hydrologic processes' interoperability and discusses how to overcome the shortcomings of the existing approach toward managing the knowledge of hydrologic processes. Finally, section seven summarizes this work and discuss possible directions for future research.

### 5.3 BACKGROUND

This section presents an overview of the hydrologic processes modeling and the legacy models used in this study, along with background of ontologies and how they can elevate information to knowledge.

#### 5.3.1 PROCESS-BASED HYDROLOGIC MODELING

Hydrologic processes begin with climate inputs and are influenced by landscape characteristics that vary over a range of space and time scales (Loucks et al., 2005; Sivakumar, 2010). Information about hydrologic processes is typically either embedded in a comprehensive software or expressed in publications. Hydrologic models vary from statistical (fully data oriented) to physical (fully process oriented) models. In between, there are hybrid models that are a mixture of both approaches. A statistical model is based on a simple mathematical relation between input and output variables of a basin (e.g. often regression equations). It reduces the model to a pure stochastic process without any cause-effect relationships. In contrast, a physical model is based

on describing the dynamics of hydrologic system using the equations governing the systems' behavior. In this study, the focus will be on two physically based hydrologic models to extract different mathematical equations used to represent a hydrologic process and its numerical representations. These models were chosen because they are open source models, well known, well tested, and well documented.

The Soil and Water Assessment Tool (SWAT) is a hydrologic and water quality model developed by the US Department of Agriculture (USDA) Agricultural Research Service (ARS) (Gassman et al., 2007). SWAT has been used to quantify the impact of different land management practices on the watershed ecology based on daily time step (Srinivasan and Arnold, 1994). It represents physical processes from different domains: weather, hydrology, soil temperature, crop growth, nutrients, bacteria and land management (Srinivasan and Arnold, 1994).

The Regional Hydro-Ecologic Simulation System (RHESSys) is another process-based hydroecological model that is designed to represent the feedback between hydrologic, vegetation, and nutrients cycle processes (Tague and Band, 2004). RHESSys represents hydrologic processes with an intermediate level of complexity, compared to lumped parameter hydrologic models and empirical models, which are less complete (Tague and Band, 2004). The architecture of RHESSys combines a set of physically based process over different spatial representations that are organized in a hierarchical structure. RHESSys uses the TOPMODEL that is developed to represent the hydrologic processes (Beven and Kirkby, 1979). TOPMODEL is a lumped parameter model that depends on the channel network topography and the dynamic responses of different contributing areas around the stream network.

### 5.3.2 ONTOLOGIES

An ontology expresses concise and precise vocabularies to describe facts about a domain and share these vocabularies and facts with both human and machine users

(Gruber, 1993; Antoniou and Harmelen, 2009). It introduces the conceptualization that underlies the knowledge and the vocabularies that represent this knowledge (Gruber, 1993; Garshol, 2004; Antoniou and Harmelen, 2009). This effective information representation system offers shared understanding of a domain for users from different disciplines. Significantly, an ontology provides a unified structure for information representation, which minimizes the conceptual and terminological confusion about a specific domain.

Ontologies are classified into two categories according to their functionality: upper-level ontology and domain-level ontology (Russell et al., 1995). An upper-level ontology describes general concepts shared among various disciplines. The Semantic Web for Earth and Environmental Terminology (SWEET) is an example of an upper-level ontology in Earth system science that describes concepts such as space, time, and physical quantities (Raskin and Pan, 2005). The Water Resource Component (WRC) ontology is another example that is created to promote the interoperability of components across water-related disciplinary boundaries and modeling frameworks (Elag and Goodall, In Revision). In contrast, a domain-level ontology focuses on describing a specific terminology of a particular domain in order to represent a specific meaning. For example, in the water resource domain, Chau (2007) introduced an ontology-based knowledge management system for water flow and quality models.

An ontology represents a specific domain by grouping entities with similar properties into a concept or class. These classes are typically arranged in a hierarchical structure (superclass-class-subclass). Features of each class are described by object and data properties (slots). Object properties connect individuals of two classes, while data properties define the attributes of individuals. Both types of properties may have restrictions on them to describe new concepts.

Ontologies have different patterns for knowledge representation (Zeng, 2005; Villa et al., 2009). They vary from control vocabularies, without any specification for the

used terms, to formalized logical relationships among concepts (Figure 5.1). The degree of complexity increases as the knowledge structure moves toward logical formalization. The amount of meaning added to the ontology (expressivity) increases with knowledge structure complexity. As the ontology expressivity increases, the ambiguity of terms decreases and this enables a machine to interpret the information.

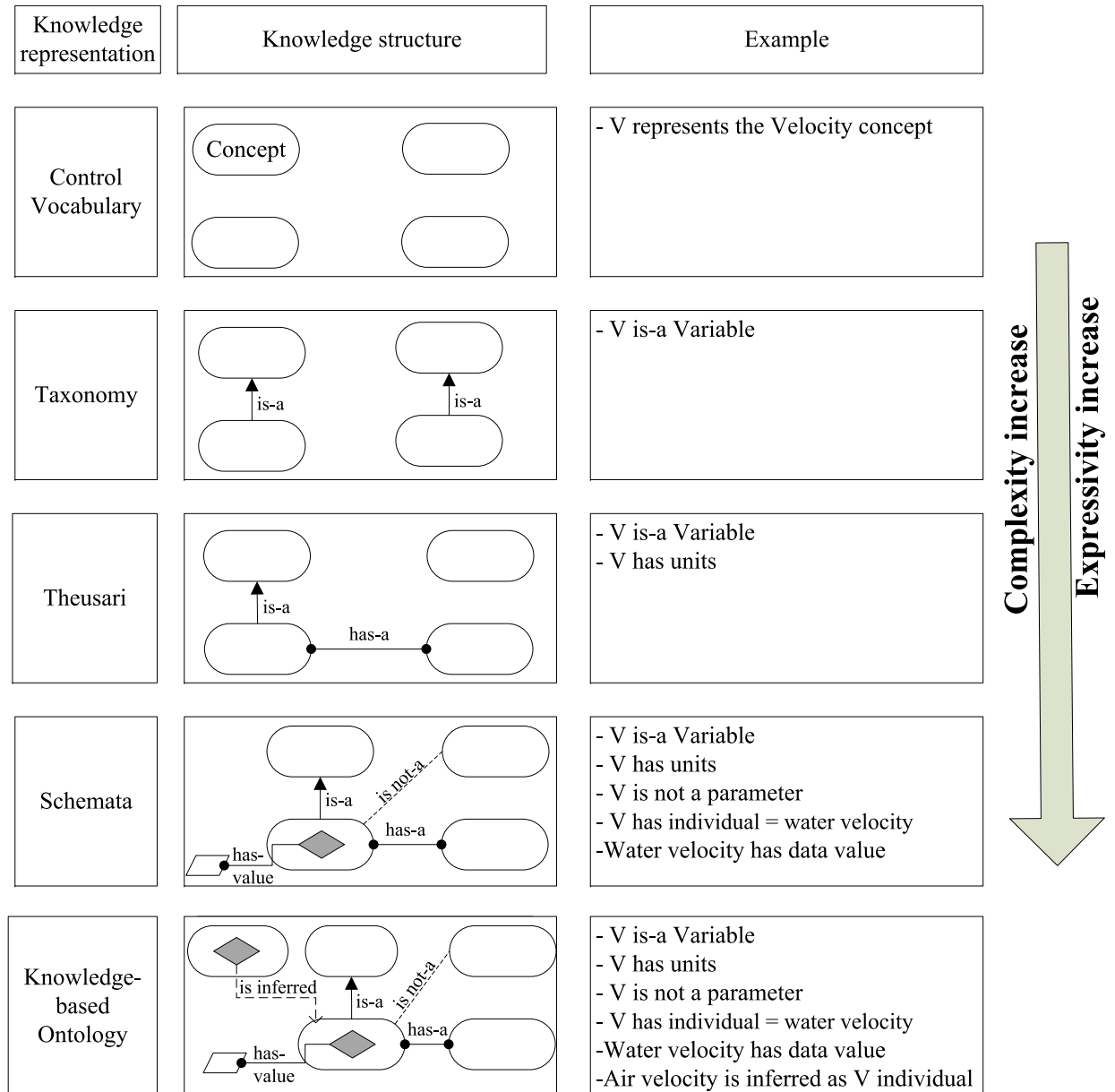


Figure 5.1: A schematic representation and examples of knowledge representation paradigms ordered by increasing expressivity and complexity, adopted from Villa et al. (2009).

## 5.4 METHODOLOGY

### 5.4.1 CONCEPTUAL DESIGN

The Hydrologic Processes (HP) Ontology design relies on the well established skeletal approach described by Uschold and Gruninger (1996), which has been successfully applied for building many ontologies in various scientific domains (e.g. Kim, 2005; Brilhante and Robertson, 2001; Patil et al., 2005; Biletskiy et al., 2004; Bermudez and Piasecki, 2006). The methodology consists of five stages: purpose definition, ontology building, evaluation, documentation, and providing guidelines.

The purpose of the HP ontology is to introduce a formal definition of the hydrologic domain processes and their mathematical representations. Thus, any hydrologic-related discipline can select the suitable hydrologic process and find sufficient information to simulate a process using a component-based modeling approach. The proposed ontology must answer the following questions: What is the ideal level of process granularity? What is the equation that represents the process? Does the equation have any alternative forms? Which equation is the best? What is the numerical representation of a specific equation? Did any of the well-known open source legacy models include a numerical representation of this process?

Building an ontology is a complex and iterative process like any development process of a prototype (Fernández-López et al., 1997). It consists of several activities, some are applied during ontology establishment and other occurs along its life cycle. During the establishment stage, ontology building consists of three phases: conceptualization, formalization, and integration. The conceptualization phase aims to acquire concepts from a given domain. It analyzes a domain to extract the basic ideas, relationships, and terms that are required to describe the domain. The formalization phase aims to serialize the previously captured information into a knowledge-based ontology using a formal ontology coding language. The integration phase aims to

reuse already predefined terminologies and relationships, to speed up the ontology building process, and to provide robustness to new ontologies. During the ontology life cycle, knowledge acquisition, documentation, and evaluation are considered to be continuous processes. The subsequent paragraphs elaborates on the approach used for conceptualization and formalization.

The Formal Concept Analysis (FCA) is used to extract, capture, and organize concepts that are used in describing a hydrologic physical process. It is a method for data analysis, knowledge representation, and information management with applications across various domains (Priss, 2006; Wormuth and Becker, 2004). FCA analyzes data that relates objects or entities to a particular set of attributes. It is widely accepted for deriving concepts from fundamental principles of a specific domain (e.g. Cole and Eklund, 1996; Groh et al., 1998; Stumme et al., 2002; Hu and Bian, 2009). It has two outputs: the first is a concept lattice, which represents the hierarchical relationships of domain concepts, and the second output is attribute implications, which represents relationships between concepts and attributes (Priss, 2006; Wormuth and Becker, 2004).

The concepts extracted by FCA are organized and structured using a top-down approach that starts with the general concept and moves down searching for details. This approach offers better control on the level of detail, however it can result in creating arbitrary high-level classes (Uschold and Gruninger, 1996). The top-down approach is considered appropriate for organizing the HP ontology's concepts because the hydrologic community can easily agree on general concepts but varies in their specialization. The concepts representing the hydrologic processes are serialized under the predefined superclasses.

Ontology Web Language (OWL) is used as a formal language for the ontology coding. OWL has been accepted and recommended as an ontology language by the World Wide Web Consortium (W3C). It is currently the most prominent ontology



language for the Semantic Web, and it is compatible with most querying languages (Antoniou and Harmelen, 2009; Hitzler et al., 2009; Garrido and Requena, 2011). Some of the HP ontology superclasses are extracted from the WRC Ontology because a model and a process are sharing the same concepts.

Building an ontology requires continuous information acquisition. Thus, the evaluation process is a repetitive stage during the ontology building. The evaluation process will be done through the application of a set of rules coded in a reasoner such that these rules check for incompleteness, inconsistencies, and redundancies in concepts and relationships (Gómez-Pérez et al., 1995). Documentation and guidelines are needed to provide the important assumptions, key concepts, and define the rules included within the ontology in a natural language that can be easily understood by users. The following section illustrates the three phases of the ontology building stage.

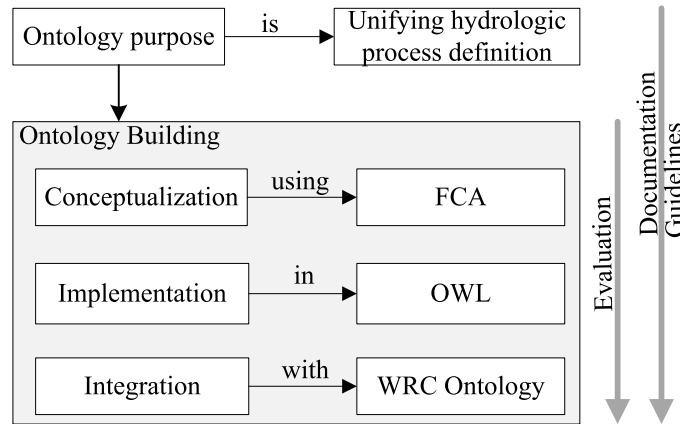


Figure 5.2: Methodology used for development of the Hydrologic Process (HP) Ontology, adopted from Uschold and Gruninger (1996).

#### 5.4.2 ONTOLOGY DEVELOPMENT

The first phase illustrates the hydrologic domain analysis using the FCA to identify the basic processes and extract their associated equations. The second phase depicts

how the concepts are organized and serialized using OWL as a formal language. Finally, the integration between the Water Resources Component (WRC) Ontology and Hydrologic Process (HP) ontology is described.

#### 5.4.2.1 INFORMATION FORMALIZATION

Defining a formal structure of hydrologic domain processes includes (i) identifying the domain basic processes from resources, (ii) organizing the extracted processes in a standard hierarchical structure, (iii) extracting their associated equations from resources, and (iv) decomposing the comprehensive hydrologic models to extract different numerical representation of the processes. In this study we use three resources: the *Handbook of Hydrology* (Maidment et al., 1992), *Elements of Physical Hydrology* (Hornberger et al., 1998), and *Hydrology Handbook* (American Society of Civil Engineers, 1996). These resources are used to define hydrologic processes and their relationships because they are widely cited by the hydrologic modeling community. Extracting the mathematical representation of the hydrologic processes is completed via the analysis of two of the well established and cited models: SWAT and RHESSys.

The three resources use the hydrologic budget equation to describe the mass balance of the hydrologic domain. The super concepts of the hydrologic domain processes and their relationships are defined using the general hydrologic budget equation (Equation 5.1). It defines four major processes in the hydrologic cycle: precipitation, evapotranspiration, runoff, and infiltration. The precipitation process was excluded because it is not typically simulated by hydrologists..

$$P = R + ET + I \quad (5.1)$$

Where  $P$  is the precipitation volume,  $R$  is the runoff volume,  $ET$  is the Evapotranspiration volume, and  $I$  is the infiltration volume. The subconcepts are extracted

Table 5.1: Infiltration process analysis using FCA and three resources: Maidment et al. (1992), Hornberger et al. (1998), and American Society of Civil Engineers (1996). The ✓ indicates that a resource has a definition for this method.

Method	<i>Handbook of Hydrology</i>	<i>Elements of Physical Hydrology</i>	<i>Hydrology Handbook</i>
Richards	✓		✓
Horton	✓	✓	✓
Philip	✓	✓	✓
Holton	✓		✓
Kostiakov	✓		✓
Modified Kostiakov			✓
Green-Ampt	✓	✓	✓
Moral Seytoux & Kanji			✓
Smith Parlange			✓
Beven's Exponential-k		✓	
SCS	✓	✓	✓

from the comparison of the three book resources. The hierarchical structure of the hydrologic processes and their relationships is described in Figure 5.3. The infiltration process was chosen as an example to demonstrate the subsequent FCA analysis because it is a basic process that cannot be further subdivided, and has several methods and equations to estimate the infiltrated water volume. In addition, it connects different domains, and influences various processes such as surface runoff, ground water recharge, evapotranspiration, soil erosion, and chemicals transport in surface and subsurface water (Maidment et al., 1992). FCA is applied to extract formal definition of the infiltration process' equations from the three resources and compare them with the local definition extracted from the SWAT and RHESSys models.

First, we extract different infiltration methods from the formal resources and consider them as the formal methods. The equations are considered as the attributes of the defined formal methods. The equations are gathered from the three book resources without simplification or approximation. Based on the analysis, the infiltration process can be simulated using 11 methods Table 5.1.

Table 5.2: Excerpts from the matrix analysis of the local process representation coded in SWAT and RHESSys versus the standardized process representation included in Table 5.1

Formal method	SWAT	RHESSys
Richards		
Horton	✓	✓
Philip	✓	
Holton		
Kostiakov		
Modified Kostiakov		
Green-Ampt	✓	
Moral Seytoux & Kanji		
Smith Parlange		
Beven's Exponential-k		✓
SCS	✓	✓

The next step is examining RHESSys and SWAT to extract the available mathematical representation of the previously described infiltration methods. The two models are decomposed based on this local definition of the process structure that is provided in their documentations. The decomposition is continued until we reach the finest level of detail in each model. The model process is defined as a local process. Then each local process of a model is examined against the formal hydrologic processes hierarchy that is previously established. The objective of this comparison is to determine how each model addresses the theoretical definition of the hydrologic processes. The comparison is used to determine whether the model includes any of the formal processes and how it mathematically represents a formal process. Relationships are established between the formal equations defined in Table 5.1 and the numerical representations extracted from SWAT and RHESSys models (Table 5.2). Links between the formal methods and their corresponding mathematical representation in models are defined in the ontology.

Finally, a key for selecting an infiltration method to represent a process is its suitability based on the initial and boundary conditions of the simulated domain.

Ravi et al. (1998) specifies seven criteria for selecting the proper model to represent an infiltration process. These criteria are classified as initial and boundary condition concepts. The initial condition concept includes surface ponding, surface runoff, rainfall and irrigation, and evapotranspiration rate. The boundary conditions includes: number of soil layers, soil profile homogeneity, and vegetation cover. These classification will help users in selecting the proper equation for given conditions.

#### 5.4.2.2 KNOWLEDGE CODING

The process of coding the ontology is where the information gathered in the previous step is turned into a knowledge base. It includes defining a domain meta-ontology (i.e. information about ontology components) in a machine understandable format using an ontology coding language. Based on the concepts and relationships defined in the previous step, it is required to define six classes: *Process*, *Method*, *Equation*, *Comprehensive Models*, *Symbol*, and *Units*. We elaborate on these six classes in the subsequent section.

The *Process* class contains the hierarchical structure of the hydrologic process. The *Method* class groups the different methods used in simulating a process. For example, the *Infiltration Method* is a subclass of the *Method* class and contains three subclasses to classify the 11 methods used in simulating an infiltration process. These subclasses are *Empirical* (e.g. Horton), *Approximate* (e.g. Green-Ampt), and *Physical* (e.g. Richard). This classification is extracted from the *Handbook of Hydrology* (Maidment et al., 1992). A method may have an equation or set of equations, as well as initial and boundary conditions. An equation uses symbols, and these symbols may be variable, parameter, or universal constants. A symbol has units to describe it. The three super classes: *Equation*, *Symbol*, and *Units* are previously defined in the WRC ontology. In the HP ontology we reuse the WRC ontology's definition, relationships, and hierarchial structure of these three concepts. Finally, the *Compre-*

*hensive Models* class represents the mathematical representation of each method that is available in both SWAT and RHESSys. This class can be expanded in the future by further analysis of other models.

The hydrologic subprocess classes are related with the “is-A” relationship to the superclasses (e.g. *Stream flow* “is-A” *Surface Runoff*) each *process* class is connected with the corresponding method of calculation using “has-a” relationship. A universal property restriction is coded on a class that represents methods of simulating a process (e.g. *Infiltration Method*) to show that a process can only be represented with one of these methods. These restrictions point out for both user and machine that there is no other method that can be used to simulate a specific process. This approach aids in maintaining the consistency of the ontology and minimizing its maintenance effort.

#### 5.4.2.3 INTEGRATION WITH WRC ONTOLOGY

Reusing an ontology includes two different themes: merging and integration (Fernández-López et al., 1997). Merging is the process where the two different ontologies are bound into one ontology with a unified namespace. The resultant ontology may have a new namespace or one of the merged ontologies namespace. This approach increases the complexity of the ontology and difficulty of tracking changes in the resultant ontology. Integration has many purposes such as extension, specialization, and adaption. The other approach, ontology integration is based on reusing classes while each ontology has its own namespace. This approach enables the reusability of the ontology and decreases the level of its complexity. It also allows reuse of definitions and properties already built and tested, providing consistency and robustness to new ontologies. The HP ontology integrates with the WRC ontology such that each ontology has its own namespace.

Water Resource Component (WRC) Ontology is an upper-level ontology and is available at <http://grg.engr.sc.edu/wrc/0.1/WRC.owl>. It consists of four ontological layers: resources, scientific, coupling, and technical. The meta-ontology of the scientific layer describes three superclasses: equations, symbols, and domain. Details about the classes hierarchal structure and their relationships is available in Elag and Goodall (In Revision). The meta-ontology of the WRC ontology is centralized around defining a component using its metadata while the HP ontology is focusing on defining a hydrologic process and its attributes. The *Water Resources Domain* class contains subclasses that are describing the water resources sub-systems, such as the *Hydrology*. The *Hydrology* class is considered the first integration key between the two ontologies. The other three integration keys are the *Equation*, *Symbols*, and *Units* classes. The integration between the two ontologies is considered a mutual integration. The WRC ontology wants the specification described in hydrologic process definition provided in the HP ontology. In the other side, the HP ontology requires the definition of the three classes described in scientific layer of the WRC ontology.

OWL provides support for ontologies integration by defining the *owl: imports* construct. This construct allows an ontology to reference a namespace of another ontology by using its Unique Resource Identifier (URI) (Grau et al., 2004). This approach provides syntactic independence between the ontologies since both of them stay in different files and are self-contained. The consistency of concepts between integrated ontologies is asserted during the evaluation stage using OWL reasoners. The Pellet reasoner was used to capture the new classification of classes after any updates. Pellet is the most common reasoning engine used with OWL. It is able to ensure the consistency of an ontology structure because it is based on the Description Logic system. The Description Logic translates the relationships between concepts into logical statements. This allows a reasoner to interpret the implicitly coded relationships between concepts into explicit relationships that can be verified.

## 5.5 RESULTS AND DISCUSSION

### 5.5.1 HYDROLOGIC PROCESSES ONTOLOGY

The Hydrologic Process (HP) ontology aims to encapsulate knowledge about hydrologic processes in both a human and machine understandable language. HP ontology has six superclasses that are oriented to describe a hydrologic process (Figure 5.3), and it is available at <http://grg.engr.sc.edu/wrc/0.1/HPO.owl>. The concepts introduced in this study allow developers from hydrology-related disciplines to select a proper function for their model. The *Process* class defines the level of process granularity and identifies the relationships between different processes. The combination between the two classes *Method* and *Equation* describes the available methods to simulate a process and its associated equations. The *Symbol* class identifies all the variables, parameters, and constants that are used by equations. Finally, the relationship between *Method* and *Comprehensive Models* identifies the availability of a numerical representation of a process in two legacy models SWAT and RHESSys.

### 5.5.2 KNOWLEDGE REPRESENTATION FOR THE INFILTRATION PROCESS

This section focuses on demonstrating the conceptualization of a hydrologic process information in the HP ontology and illustrating how a process information is represented in this knowledge-based ontology. Two infiltration methods, Green-Ampt and Philip's methods, were chosen for estimating the infiltrated water depth. These methods were used to demonstrate the knowledge representation in the HP ontology.



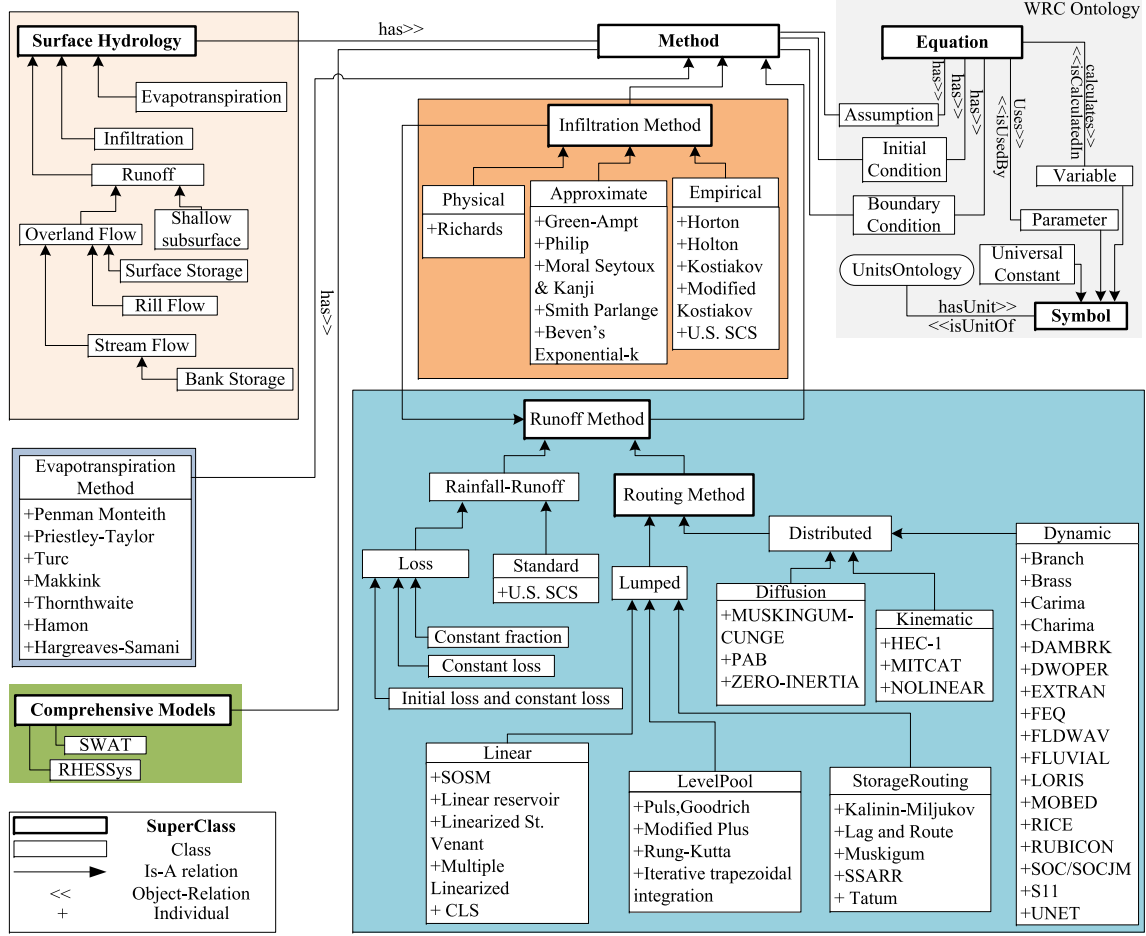


Figure 5.3: Hydrologic Process (HP) ontology including classes and relationships.

#### 5.5.2.1 GREEN-AMPT METHOD

Green-Ampt is an approximate method that is based on Darcy's law to estimate the infiltrated water depth through a uniform single soil layer (Maidment et al., 1992). It assumes a uniform initial water content that is ponded on the top of a homogenous deep soil. The Green-Ampt method has been extensively studied and adapted for modeling infiltration process with different situations such as water infiltrating into a non-uniform soil or under a non-ponding condition. The Green-Ampt method that was applied in this study assumed a constant water depth was ponding on the soil top boundary surface and a constant soil infiltrability. Equation 5.2 represents the cumulative infiltrated water depth.

$$F(t) = Kt + \psi\Delta\theta \ln(1 + \frac{F(t)}{\psi\Delta\theta}) \quad (5.2)$$

where  $F$  is the cumulative infiltration depth (m),  $K$  is the effective hydraulic conductivity (m/s),  $\psi$  is the effective suction at the wetting front (m), and  $\Delta\theta$  is the difference between the soil porosity and its initial water content (unitless). This equation is an implicit mathematical relationship between the cumulative infiltration depth and the soil parameters ( $K$ ,  $\psi$ , and  $\Delta\theta$ ). Given this, an iteration method is required to calculate the value of  $F$ . Once the  $F$  is estimated, the  $f$  infiltrability (m/sec) can be determined from Equation 5.3.

$$f(t) = K(\frac{\psi\Delta\theta}{F(t)} + 1) \quad (5.3)$$

Figure 5.4 illustrates how the Green-Ampt method and its associated equations with their attributes can be stored in the HP ontology. The Green-Ampt method is represented as an individual of the *Approximate* class that is a subclass of the *Infiltration Methods* class. The Green-Ampt individual is linked with an EquationID individual in the *Equation* class. The EquationID individual has a data type property coding the cumulative infiltration depth. This equation is coded using Math Markup Language version 3.0 (MathML) (MathML, <http://www.w3.org/TR/MathML3/>). It is a mean for describing mathematical notation and capturing both its structure and content (Carlisle et al., 2009). The parameters are defined as an input for the equationID based on the type of the relationship between them (e.g. Green-Ampt EquationID *hasInputParameter*  $K$ ).

The top boundary condition of any domain that will apply the simple Green-Ampt method should have a constant water content. This condition is expressed in the ontology using a mathematical statement ( $\delta(P + M)/\delta t = 0$ ), where  $P$  is the precipitation rate (mm/sec) and  $M$  is the snowmelt rate (mm/sec). This statement is machine interpretable, so a modeling framework can ensure that a process boundary

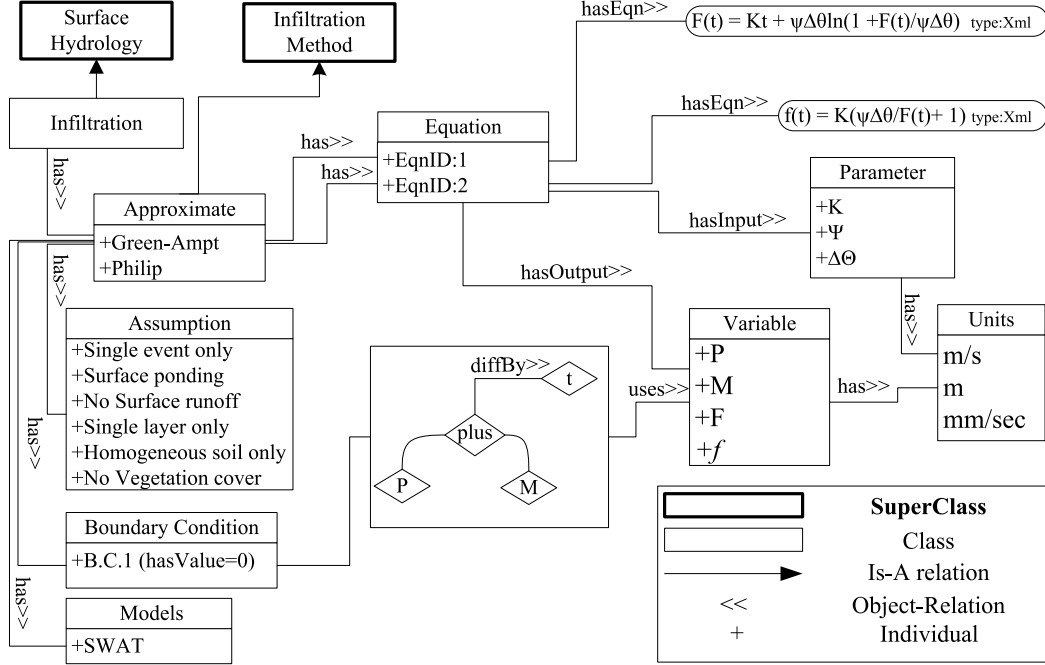


Figure 5.4: Excerpt from the Hydrologic Process (HP) ontology illustrating the stored information for the Green-Ampt method.

condition is not violated within the simulation. The BoundaryConditionID is connected with both the Green-Ampt individual and EquationID in a 3-ary relationship. This type of relationship ensures that the Green-Ampt method and its associated equation are using the same boundary condition and provides a complete description for the process that minimizes the semantic heterogeneity between processes. Some assumptions cannot be expressed through mathematical representation, so they are included as a string statement (e.g. Green-Ampt method is applied for a single layer).

#### 5.5.2.2 PHILIP'S METHOD

Philip's method is another approximate method used to estimate the infiltrated water depth based on Richard's equation (Maidment et al., 1992). It converted Richard's equation into an ordinary differential equation. Philip's method is used for a ho-

homogeneous single soil layer and is valid for short-term infiltration compared to the Green-Ampt method (American Society of Civil Engineers, 1996). Similar to the Green-Ampt method, it assumes a constant water content at the top boundaries of the soil. Philip's equation introduced an approximate mathematical formula (Equation 5.4) to estimate the cumulative infiltration depth  $F$  (m)

$$F(t) = St^{1/2} + Kt \quad (5.4)$$

where  $S$  is the sorptivity parameter and  $K$  is the effective hydraulic conductivity of the soil (m/s). Philip's method is an explicit mathematical relationship between  $F$  and two of the soil parameter. The infiltrability  $f$  (m/sec) can be determined from Equation 5.5.

$$f(t) = \frac{1}{2}St^{-1/2} + K \quad (5.5)$$

Figure 5.5 depicts the coded information about Philip's method in the HP ontology. The namespace of WRC and HP ontologies are declared using their Unique Resource Identifier (URI) because describing Philip's method requires concepts from both ontologies. Two concepts are required to describe Philip's method as an approximate method for simulating the infiltration process. This method is recommended for short term infiltration. This limitation is defined using a string because there is no agreed upon definition for the short term infiltration. In the future, the hydrologic modeling community should define limits for short term infiltration. The mathematical modeling knowledge of the method is ellipticity defined through the OWL instance and relationships. The *hasEquation* is a data property that is populated with the MathML representation of Equation 5.4. The method parameters are connected with the placeholder of their definition through the WRC ontology's namespace, thus ensuring the consistency among concepts of the integrated ontologies.

PREFIX HPO: <http://grg.engr.sc.edu/wrc/0.1/HPO.owl#>	
PREFIX WRC: <http://grg.engr.sc.edu/wrc/0.1/WRC.owl#>	
Philip Method=	
HPO: Is-a <i>Approximate, InfiltrationMethod</i> ;	
HPO: represents <i>Infiltration</i> ;	
HPO: hasSWATRepresentation <i>RepID</i> ;	type: XML literal
HPO: hasLimitation “ <i>short term infiltration</i> ”	type: String
HPO: hasBoundaryCondition “ $\Delta(P+M)/\Delta t$ ”	type: XML literal
WRC: hasEquation “ $St^{1/2} + Kt$ ”;	type: XML literal
WRC: hasInputParameter <i>K, S</i> .	type: Float
WRC: hasInputVariable <i>t</i> .	type: Float

Figure 5.5: Excerpt from the Hydrologic Process (HP) ontology illustrating the stored information for Philip’s method.

### 5.5.3 HP ONTOLOGY IMPLEMENTATION

The previous sections focused on the ontology from the provider perspective (e.g. HP ontology can answer questions by describing a relationship among concepts). On the other hand, a consumer perspective is interested in information retrieval and its utilization. The objective of the HP ontology is to provide disparate users with a tool that supports proper selection and coding of hydrologic processes within and across disciplinary boundaries. The following sections focus on describing the Web architecture used for querying the HP ontology on the Web and illustrating the procedures required to translate a knowledge model to a computer model.

#### 5.5.3.1 HP ONTOLOGY QUERYING

The reliance of the HP ontology on Ontology Web language (OWL) expands the querying process beyond the semantic level to include the facts that are coded in the ontology. OWL is currently the most prominent ontology language for the Semantic Web, and it is compatible with most querying languages (Antoniou and Harmelen, 2009; Hitzler et al., 2009; Garrido and Requena, 2011). Simple Protocol and RDF Query Language (SPARQL) is recommended by W3C as a query language for the Semantic Web. It is a data access protocol for querying remote databases over HTTP.

The query language is used to retrieve RDF graph patterns that matches a pre-defined graph pattern. The query language relies on various SQL languages used for querying relational databases, but have been adapted to query RDF triples (Prud'Hommeaux et al., 2008). The complexity of the graph pattern can be increased by combining simple patterns using various methods.

The SPARQL protocol requires a web service (data endpoint) to transfer SPARQL queries and results between a client and a server. The Dydra computational cloud was used to provide a SPARQL endpoint for the HP ontology (<http://dydra.com/WRC/hydrologicprocessontology/sparql>). Dydra is a cloud-based RDF graph store that relies on the Amazon Web Services (AWS) (Bugiotti et al., 2012). It uses open standards that are recommended by W3C and enable a client to query data using any programming language, platform, or device. Figure(5.6) depicts the client-service interaction. First, the client (human or machine) issues a SPARQL query request to the endpoint defined by Dydra cloud. This request is delegated to the Dydra cloud to query the namespaces described in the query statement. Finally, the obtained RDF datasets (triples) are returned back to the client application and rendered in a various forms (e.g. XML, JSON, and HTML).

To illustrate the benefit of using SPARQL as a query language for the HP ontology from a client's prospective, a SPARQL query is issued to the HP ontology endpoint. It requests all the infiltration methods that are individuals (objects) of the *Approximate method* class (subclass of the *infiltration method*) and has SWAT representation as follow:

PREFIX HPO: <http://grg.engr.sc.edu/wrc/0.1/HPO.owl#>

SELECT ?Individual

WHERE {?Individual a HPO: InfiltrationMethod; a HPO: Approximatemethod;

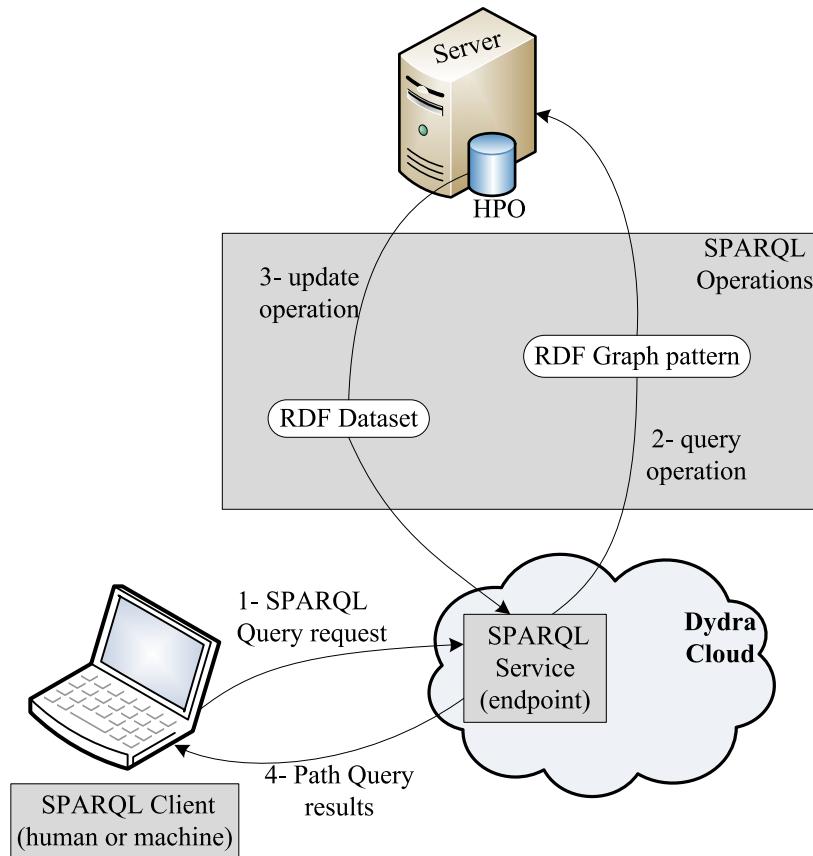


Figure 5.6: Communication between a client and server via SPARQL endpoint that is hosted in Dydra computational cloud.

HPO: hasRep ?SWAT}

The query returns the set of terms shown below.

[http://grg.engr.sc.edu/wrc/0.1/HPO.owl# Green\\_Ampt](http://grg.engr.sc.edu/wrc/0.1/HPO.owl# Green_Ampt)

[http://grg.engr.sc.edu/wrc/0.1/HPO.owl# Philips\\_Equation](http://grg.engr.sc.edu/wrc/0.1/HPO.owl# Philips_Equation)

The return RDF set demonstrates an explicit joint between a query that retrieves approximate infiltration methods and those methods that have a mathematical representation in SWAT. Therefore, SPARQL allows client applications to extract required information in a standard method. Clients are allowed to query an ontology service using SPARQL for concepts, relationships, individuals, and subontologies.

This minimizes the effort required to parse the data model. SPARQL is built on various standards including RDF, HTTP, and XML. This allows interoperability with other software systems (Prud'Hommeaux et al., 2008). Thus, it is easy to issue SPARQL queries from software systems such as Perl, PHP, or Python. The next section demonstrates the capability of the HP ontology to represent a hydrologic process as a knowledge model.

#### 5.5.3.2 FROM A KNOWLEDGE REPRESENTATION TO PSEUDO-CODE

Knowledge extraction from ontology is about discovering the ontological concepts that conceptualize information stored in the ontology and retrieving its attributes. The output of this process is human and machine-readable information. The HP ontology stores information about mathematical methods used to describe a hydrologic process. It includes the methods' mathematical equations and the relationships between an equation and its arguments including variables, parameters, and constants. This standardized information and its attributes have the potential to build a computer model and assert the conceptual consistency of the created model.

Yang et al. (2004) defined two successive phases for constructing a computer model from a knowledge-based ontology: conceptual modeling and model generation. The conceptual modeling phase is responsible for declaring the information required to generate a mathematical model, it includes the declaration of equations and their arguments. The model generation phase is the procedural phase, where a set of instructions that explicitly describe equation control flow (execution procedure) is coded into machine-readable expressions. The hydrologic processes information stored in the HP ontology can be extracted and declared according to different modeling paradigm requirements. The declared information can be used to generate a source code for different component modeling frameworks. For example, the information coded about Philip's method can be translated into imperative instructions in a programming lan-



guage such as Python or Java. A key step to translate the information coded in the HP ontology to a source code is to automate the process of information extraction from the ontology and its declaration.

A Python class is created to abstract information about a hydrologic process from the HP ontology based on the meta-ontology and convert it to a pseudo-code. This class implements the SPARQLWrapper and rdflib packages to query the HP ontology using the SPARQL endpoint (<http://dydra.com/WRC/hydrologicprocessontology/sparql>) and handle the RDF triples extracted from the ontology. Thus, experts can focus on adding knowledge to the HP ontology and users can implement a process mathematical model into any programming language. A simple case study is introduced that queries the HP ontology for approximate infiltration methods that estimate infiltrated water depth. Then, it has been assumed that a user selects Philip's equation for simulation. To get all the attributes of Philip's equation a Python method was defined that considers the selected object as a subject and gets all its predicates and objects. Furthermore, it extracts the data attributes related to the objects, including numerical values, type, and ranges. Another Python method was written to convert the *hasEquation* instance from its MathML representation to a mathematical expression that can be evaluated by the Python engine. Using the Python class allows a user to extract information about a hydrologic process and minimize coding errors.

The Philip's method is used to examine the workability of the algorithm described in the Python class. An example from *Applied Hydrology book* (Chow et al., 1988) was used to verify that the developed model correctly simulates Philip's method. It has two input parameters that must be defined by a user. The code implementation produces identical results for the simulation.

It is acknowledged that creating a pseudo-code from a knowledge-based ontology is beyond the scope of a single paper because doing so requires that the modeling

tools be more generic and extendable. Furthermore, it requires integration of ontologies from different disciplines that are able to implement different simulation algorithms. For example, creating a pseudo-code for the Green-Ampt method requires iteration algorithms (e.g. bisection and Newton-Raphson) to solve its implicit equation. Creating an ontology that describes iteration algorithms requires collaboration with mathematical scientist.

```
//Method comments
Purpose: Approximate method used in estimating infiltration.
Limitation: short term infiltration
B.C. constant water content at the top boundary surface
//Method Inputs
Declare S, K, t As float
S is sorptivity parameter & parameter
S unit unitless
K is effective hydraulic conductivity of the soil & parameter
K unit m/s
//Method Outputs
F is cumulative infiltration depth & Variable
F unit m
f is infiltrability & Variable
f unit m/s
//Function
Philip Func (K, S, t)
     $F = St^{1/2} + Kt$ 
     $f = 0.5 St^{1/2} + Kt$ 
Display F, f
End Philip
```

Figure 5.7: Pseudo-code for Philip's method.

#### 5.5.4 DISCUSSION

The HP ontology is an expressive knowledge-based ontology that contains quantitative values of individuals and explicitly defines all relationships among them. This approach helps to minimize the semantic and syntactic heterogeneity among hydrology-related disciplines. An ontology adds semantics to its individuals by relating standard names used by multiple communities via equivalency relationships. Thus, it removes

redundancy and ambiguity of an individual’s terminology and increases the interoperability of a process definition across related disciplines. A process rich knowledge definition available in the HP ontology can inform any modeling framework about the process modeling workflow. For example, Philip’s equation can be seen as a statement with specific dependency on soil initial moisture content.

The HP ontology acts as a repository to organize and articulate information about hydrologic processes. It combines a set of statements about hydrologic processes from multiple sources to provide an unambiguous identification of entities and enable accurate interpretation of data coded from multiple resources. The HP ontology addresses the syntactic heterogeneity in coding information about a hydrologic process by introducing a formal structure for storing processes’ information. The coded information can be transferred among users and decoded correctly by different machines. This enables the communication and knowledge reuse and allows modeling frameworks to extract the information required for creating a computer model.

## 5.6 SUMMARY AND FUTURE WORK

Increasing the interoperability of process-based hydrologic models among hydrologic-related disciplines is required to support the integration of models across disciplinary boundaries. A key step toward this objective is standardizing the information used in describing hydrologic processes. This study aimed to introduce a formal definition for a hydrologic process. It used the infiltration process as a case study to illustrate the procedures of identifying methods and equations available to represent a process. In this study, the FCA was used to compare, extract, and define the hierarchical structure of hydrologic process and identify the associated equation representation. The resulting information is coded using OWL to introduce a semantic knowledge base for the hydrologic process.

The outcomes of this research are an ontology describing the scientific information about hydrologic processes. The use of the Hydrologic Process Ontology will increase interoperability among related disciplines by (i) minimizing the semantic heterogeneity in the terms used to describe the processes, (ii) decreasing the syntactic heterogeneity in modeling processes by assigning to the process all the expected mathematical representations, and (iii) minimizing the structural heterogeneity in the process relationships by introducing a constant hierarchical structure of the process. Integrating this ontology with the Water Resource Component ontology elevates the hydrologic modeling process using the component-based approach into the level of abstraction where the modeling process will become a knowledge representation process.

Establishing a robust and fully explicit ontology is an iterative work and requires collaboration among scientists from different disciplines. In establishing the HP ontology, it is important to have a parallel process of gathering and identifying standard names from different domains in order to unify the diversity of concepts used by various modeling communities for defining a hydrologic process. Thus, it is important to create a knowledge-based framework to gather information from different resources, elaborate on this information, and turn them into useful knowledge. A key step to establish this knowledge-based framework is to create an environment for users and experts to share their information (Villa et al., 2009). For this purpose, the knowledge-based framework needs to satisfy specific criteria in order to encourage the participation of users from different domains. It should be flexible, user friendly, have a quick learning curve, and be Web accessible.

Future work will focus on combining the Semantic Wiki technology with HP ontology to create a knowledge-based framework. Semantic Wiki systems aim to combine the characteristics of wikis with Semantic Web technologies (Völkel et al., 2006). Semantic Wiki emerged aiming to address problems due to the unstructured accumu-

lated information from different resources in wiki systems. The combination between the semantic wiki system and the HP ontology will provide an easy way to use interface for adding semantic annotations for hydrologic processes, support collaboration between different levels of users, and hide unnecessary complexity.

## CHAPTER 6

### SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

This research was motivated by the fact that solving complex water resources problems requires integration of models across disciplinary boundaries, which necessitates standardization of information describing these models. The work described in this dissertation aimed to address the growing need within the water resources modeling community for increasing the interoperability of independent component models. This work supports the integration of water related model components coming from various disciplines and enhances the collaboration among related scientific communities. First, it was shown that a component-based architecture can simulate feedback loops between hydrologic model components that share a boundary condition, and transfer data between temporally misaligned model components. Next, a well-designed ontology named Water Resources Component (WRC) ontology was developed to serve as a knowledge-level specification for the joint conceptualization used in defining model components across disciplinary boundaries and frameworks. Finally, the WRC ontology was expanded by creating a domain-level ontology for describing the hydrologic processes and increasing their interoperability among related disciplines.

The first study aimed to answer an important research question about the suitability of a component-based modeling approach to simulate complicated system dynamics including feedback loops and misalignment of data exchanges between coupled models. These questions were investigated by developing a simple case study of mass transport within a mixed media system of water over sediment column. Component

models workflow was implemented using the OpenMI component-based modeling protocol to investigate how feedback loops (or what OpenMI terms “bi-directional links”) between components are handled and how misaligned component interactions are handled within OpenMI. The results of the component-based workflow were compared with a more conventional numerical solution to the same system. This comparison provided a means for understanding how the component-based composition was solved, and for quantifying mass balance errors in the case where component output needs to be temporally rescaled to accommodate the input needs of another component.

It was shown that OpenMI has a communication paradigm for bidirectional linked models that have a shared boundary. First, when a component has an unanswered data request, it is not allowed to issue any additional data requests. Second, a component is required to always return values when a request for data is issued. These two requirements result in components estimating values based on their previously calculated values. The design of OpenMI makes it possible to enhance the logic for handling feedback loops between temporal misaligned coupled models by adopting a scheme that allows for iteration between components to converge on a shared boundary condition. The architecture of OpenMI accepts adding new interpolation algorithms in order to diversify the schemes available to users for rescaling data transfers between components. For the particular advection-diffusion case study, it was found that a cubic spline interpolation was best suited for minimizing system mass balance error for cases in which there is a small time step difference between model components, whereas a linear interpolation performed best for large time step.

Motivated by the task of promoting the interoperability of components across water-related disciplinary boundaries and modeling frameworks using metadata, a well-designed ontology termed the Water Resources Component (WRC) Ontology was developed to unify the metadata concepts used for describing a component. This

ontology is considered as a knowledge-level specification for the joint conceptualization used in defining model components across disciplinary boundaries and modeling frameworks. It has the potential to elevate component-based water resources modeling to a level of abstraction where modeling becomes a knowledge representation processes. WRC is coded using Ontology Web Language (OWL) and defines concepts associated with the component in an explicit format that is readable by both the user and the system. The WRC ontology grouped these concepts in four ontological layers: resources, scientific, coupling, and technical to represent a component metadata. It provides modelers using a component-based modeling approach with a tool that helps them in selecting the correct components to be coupled and aids in minimizing coupling conceptual error.

The WRC ontology is an upper level ontology that focuses on unifying the concepts used in describing water resources component models' metadata. It minimizes the semantic heterogeneity across water related disciplines and syntactic heterogeneity in the metadata structure used to describe a component across modeling frameworks. WRC ontology facilitates the interoperability of models across Earth Science modeling frameworks and scientific communities. It is considered to be a sustainable solution to ensure the correct conceptual coupling between components coming from different disciplines based on the scientific knowledge underlying each component. Ontologies are like models in that the development of both is an iterative process that requires input and refinement from a larger community to reach an agreed upon version. Revisions will be required to compensate for unforeseen conditions and new knowledge. Given this, the goal for the ontology was to serve as the starting point for a community agreed upon ontology for water resource model components. This work was meant to provide this beginning ontology that builds on related efforts to create ontologies within the Earth Science community, but will likely evolve as more developers and users engage in the design process.



The final study expands the WRC ontology by creating a domain-level ontology that introduces formal definition of the hydrologic domain processes and their interrelationships, corresponding equations and attributes. This ontology ensures a common understanding among domain users and supports the process of creating components within different modeling frameworks by guiding users to the scientific information and tested codes. The aim of this study was to present the methodology used in analyzing the basic hydrologic domain in order to identify the hydrologic processes, the ontology itself, and how it integrates with the Water Resources Component (WRC) ontology. In this study the infiltration process is used as a case study to illustrate the procedures of identifying methods and equations available to represent a process. In this study, the FCA was used to compare, extract, and define the hydrologic processes hierarchical structure and identify the associated equation representation.

The significance of this work lies not only in the development of a knowledge-based ontology that defines the hydrologic domain processes, but also in the fact that the coded knowledge can be converted into other forms including a source code of a computer program that simulates a hydrologic mathematical model. The outcomes of this work minimizes the semantic heterogeneity in the terms used to describe the processes and decreases the syntactic heterogeneity process modeling by assigning all the expected mathematical representations to the process. Furthermore, to advance the knowledge usability among users (human and machine), a tool was provided that is able to retrieve and present a process knowledge in a declarative approach. This tool advances the usability of coded knowledge and is considered as a step toward a generic tool that can manage and convert coded knowledge into a source code independent of any programming languages.

More research is needed to provide a software environment for users and experts to share and exchange their information. This environment should be flexible, user friendly, have a quick learning curve, and be Web accessible. The Semantic Wiki

system may be a suitable environment for this purpose. Future research should focus on connecting the Wiki system with the WRC knowledge framework. A user can describe his or her information about hydrologic models in a normal wiki page and an underlying subroutine can store this knowledge in the knowledge-based framework and vice-versa. Future research should also focus on using ontologies to provide intelligence to modeling frameworks in order to automate the consistency checks when coupling components and enable the framework to search, discover, and couple components automatically. Finally, there is great potential in combining observation datasets and model ontologies based on their underlying metadata. Using the ontological approach provides a standardized definition for a component model input observation data, this would enable modeling frameworks or Web search engines to retrieve the required data across the Web, based on the metadata standards used in describing the observation data.

## REFERENCES

- Abadi, D., A. Marcus, S. Madden, and K. Hollenbach (2007). Scalable semantic web data management using vertical partitioning. In *Proceedings of the 33rd International Conference on Very Large Data Bases*, pp. 411–422. VLDB Endowment.
- American Society of Civil Engineers (1996). *Hydrology handbook* (2nd ed ed.). Number no. 28 in ASCE manuals and reports on engineering practice. New York: ASCE.
- Anderson, R., W. Dietrich, D. Furbish, D. Hanes, A. Howard, C. Paola, J. Pelletier, R. Slingerland, R. Stallard, J. Syvitski, et al. (2004). Community Surface Dynamics Modeling System (CSDMS) science plan. *A Report to the National Science Foundation, NCED (the National Center for Earth-surface Dynamics), University Minnesota, Minneapolis*, 47.
- Antoniou, G. and F. Harmelen (2009). Web Ontology Language: OWL. *Handbook on ontologies*, 91–110.
- Argent, R. (2004). An overview of model integration for environmental applications—components, frameworks and semantics. *Environmental Modelling & Software* 19(3), 219–234.
- Argent, R., A. Voinov, T. Maxwell, S. Cuddy, J. Rahman, S. Seaton, R. Vertessy, and R. Braddock (2006). Comparing modelling frameworks—a workshop approach. *Environmental Modelling & Software* 21(7), 895–910.
- Armstrong, R., D. Gannon, A. Geist, K. Keahey, S. Kohn, L. McInnes, S. Parker, and B. Smolinski (2002). Toward a common component architecture for high-

- performance scientific computing. In *High Performance Distributed Computing, 1999. Proceedings. The Eighth International Symposium on*, pp. 115–124. IEEE.
- Athanasiadis, I., A. Rizzoli, M. Donatelli, and L. Carlini (2006). Enriching software model interfaces using ontology-based tools. In *Proceedings of the iEMSs Third Biennial Meeting: “Summit on Environmental Modelling and Software”. International Environmental Modelling and Software Society, Burlington, USA*.
- Athanasiadis, I., A. Rizzoli, M. Donatelli, and L. Carlini (2011). Enriching environmental software model interfaces through ontology-based tools. *International Journal of Applied Systemic Studies* 4(1), 94–105.
- Beran, B. and M. Piasecki (2009). Engineering new paths to water data. *Computers & Geosciences* 35(4), 753–760.
- Bermudez, L. and M. Piasecki (2006). Metadata community profiles for the semantic web. *Geoinformatica* 10(2), 159–176.
- Berners-Lee, T. and M. Fischetti (2001). *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. DIANE Publishing Company.
- Beven, K. (2002). Towards an alternative blueprint for a physically based digitally simulated hydrologic response modelling system. *Hydrological processes* 16(2), 189–206.
- Beven, K. and K. Beven (2001). *Rainfall-runoff modelling*. Wiley Online Library.
- Beven, K. and M. Kirkby (1979). A physically based, variable contributing area model of basin hydrology/un modèle à base physique de zone d’appel variable de l’hydrologie du bassin versant. *Hydrological Sciences Journal* 24(1), 43–69.

- Biletskiy, Y., O. Vorochek, and A. Medovoy (2004). Building ontologies for interoperability among learning objects and learners. *Innovations in Applied Artificial Intelligence*, 977–986.
- Blackmon, M., B. Boville, F. Bryan, R. Dickinson, P. Gent, J. Kiehl, R. Moritz, D. Randall, J. Shukla, S. Solomon, et al. (2001). The community climate system model. *Bulletin-American Meterological Society* 82(11), 2357–2376.
- Bloschl, G. and M. Sivapalan (1995). Scale issues in hydrological modelling: A review. *Hydrological Processes* 9(3-4), 251–290.
- Bray, T., J. Paoli, C. Sperberg-McQueen, E. Maler, and F. Yergeau (1997). Extensible Markup Language (XML). *World Wide Web Journal* 2(4), 27–66.
- Brilhante, V. and D. Robertson (2001). Metadata-supported automated ecological modelling. *Environmental Information Systems In Industry and Public Administration*, 313–332.
- Bugiotti, F., F. Goasdoué, Z. Kaoudi, and I. Manolescu (2012). Rdf data management in the amazon cloud. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, pp. 61–72. ACM.
- Bulatewicz, T., X. Yang, J. Peterson, S. Staggenborg, S. Welch, and D. Steward (2010). Accessible integration of agriculture, groundwater, and economic models using the Open Modeling Interface(OpenMI): methodology and initial results. *Hydrology and Earth System Sciences* 14(3), 521–534.
- Carlisle, D., R. Miner, and P. Ion (2009). Mathematical Markup Language (MathML) version 3.0.
- Castronova, A. M. and J. L. Goodall (2010). A generic approach for developing process-level hydrologic modeling components. *Environmental Modelling & Software* 25(7), 819–825.

- Castronvoa, A., J. L. Goodall, and M. Elag (2012). Models as web services using the Open Geospatial Consortium (OGC) Web Processing Service (WPS) standard. *Environmental Modelling & Software In Revision*.
- Chau, K. (2007). An ontology-based knowledge management system for flow and water quality modeling. *Advances in Engineering Software* 38(3), 172–181.
- Chow, V., D. Maidment, L. Mays, et al. (1988). *Applied Hydrology*.
- Clements, P. C. (1995). From subroutines to subsystems: Component-based software development. In *Proceedings of the 17th International Conference on Software Engineering - ICSE '95*, Pittsburgh, Pennsylvania, USA, pp. 179–185.
- Cole, R. and P. Eklund (1996). Application of formal concept analysis to information retrieval using a hierarchically structured thesaurus. In *Proc. Fourth International Conference on Conceptual Structures (to appear)*, Sydney, Australia. Citeseer.
- Cox, S. (2006). Observations and measurements. *Open Geospatial Consortium Best Practices Document*. Open Geospatial Consortium.
- Craig, A., M. Vertenstein, and R. Jacob (2012). A new flexible coupler for earth system modeling developed for ccsm4 and cesm1. *International Journal of High Performance Computing Applications* 26(1), 31–42.
- Dozier, J. (1992). Opportunities to improve hydrologic data. *Reviews of Geophysics* 30(4), 315–331.
- D’Souza, D. and A. Wills (1998). *Objects, Components, and Frameworks with UML-The Catalysis Approach*. Addison-Wesley.
- Dunlap, R., L. Mark, S. Rugaber, V. Balaji, J. Chastang, L. Cinquini, C. DeLuca, D. Middleton, and S. Murphy (2008, November). Earth system curator: metadata infrastructure for climate modeling. *Earth Science Informatics* 1, 131–149.

- Elag, M. M., J. L. Goodall, and A. M. Castronova (2011). Feedback loops and temporal misalignment in component-based hydrologic modeling. *Water Resources Research* 47(12).
- Ewert, F., M. van Ittersum, I. Bezlepkina, O. Therond, E. Andersen, H. Belhouchette, C. Bockstaller, F. Brouwer, T. Heckelei, S. Janssen, et al. (2009). A methodology for enhanced flexibility of integrated assessment in agriculture. *Environmental Science & Policy* 12(5), 546–561.
- Famiglietti, J., L. Murdoch, V. Lakshmi, and R. Hooper (2008). Community modeling in hydrologic science: Scoping workshop on a Community Hydrologic Modeling Platform (CHyMP); washington, dc, 26–27 march 2008. *Eos, Transactions American Geophysical Union* 89(32), 292.
- Fensel, D., F. Facca, E. Simperl, and I. Toma (2011). *Semantic Web Services*. Springer.
- Fernández-López, M., A. Gómez-Pérez, and N. Juristo (1997). Methontology: from ontological art towards ontological engineering.
- Fotopoulos, F., C. Makropoulos, and M. Mimikou (2010). Flood forecasting in trans-boundary catchments using the Open Modeling Interface. *Environmental Modelling & Software*.
- Garlan, D., R. Allen, and J. Ockerbloom (1995). Architectural mismatch or why it’s hard to build systems out of existing parts. In *Proceedings of the 17th International Conference on Software Engineering - ICSE ’95*, Seattle, Washington, United States, pp. 179–185.
- Garrido, J. and I. Requena (2011). Proposal of ontology for environmental impact assessment: An application with knowledge mobilization. *Expert Systems with Applications* 38(3), 2462–2472.

- Garshol, L. (2004). Metadata? thesauri? taxonomies? topic maps! making sense of it all. *Journal of Information Science* 30(4), 378–391.
- Gassman, P., M. Reyes, C. Green, and J. Arnold (2007). *The soil and water assessment tool: historical development, applications, and future research directions*. Center for Agricultural and Rural Development, Iowa State University.
- Gent, P., G. Danabasoglu, L. Donner, M. Holland, E. Hunke, S. Jayne, D. Lawrence, R. Neale, P. Rasch, M. Vertenstein, et al. (2011). The community climate system model version 4. *Journal of Climate* 24(19), 4973–4991.
- Gómez-Pérez, A., N. Juristo, and J. Pazos (1995). Evaluation and assessment of the knowledge sharing technology. *Toward Very Large Knowledge Bases*, 289–296.
- Goodall, J., J. Horsburgh, T. Whiteaker, D. Maidment, and I. Zaslavsky (2008). A first approach to web services for the National Water Information System. *Environmental Modelling & Software* 23(4), 404–411.
- Goodall, J. and D. Maidment (2009). A spatiotemporal data model for river basin-scale hydrologic systems. *International Journal of Geographical Information Science* 23(2), 233–247.
- Goodall, J. L., B. F. Robinson, and A. M. Castronova (2011). Modeling water resource systems using a service-oriented computing paradigm. *Environmental Modelling & Software* 26(5), 573–582.
- Gössler, G. and J. Sifakis (2003). Composition for component-based modeling. In *Formal Methods for Components and Objects*, pp. 443–466. Springer.
- Grau, B. C., B. Parsia, and E. Sirin (2004). Working with multiple ontologies on the semantic web. In *The Semantic Web–ISWC 2004*, pp. 620–634. Springer.



- Gregersen, J. B., P. J. A. Gijsbers, and S. J. P. Westen (2007). OpenMI: open modelling interface. *Journal of Hydroinformatics* 9(3), 175.
- Groh, B., S. Strahringer, and R. Wille (1998). Toscana-systems based on thesauri. *Conceptual Structures: Theory, Tools and Applications*, 127–138.
- Grossman, D. and O. Frieder (2004). *Information retrieval: Algorithms and heuristics*, Volume 15. Kluwer Academic Pub.
- Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2), 199–220.
- Heineman, G. and W. Councill (2001). *Component-based software engineering: putting the pieces together*, Volume 17. Addison-Wesley USA.
- Hill, C., C. DeLuca, M. Suarez, A. Da Silva, et al. (2004). The architecture of the earth system modeling framework. *Computing in Science & Engineering* 6(1), 18–28.
- Hitzler, P., M. Krötzsch, and S. Rudolph (2009). *Foundations of semantic web technologies*. Chapman & Hall/CRC.
- Hornberger, G., J. Raffensperger, P. Wiberg, and K. Eshleman (1998). *Elements of Physical Hydrology*. Johns Hopkins University Press.
- Hornberger, G. M., E. Bernhardt, W. E. Dietrich, E. Dara, G. E. Fogg, E. F. Georgiou, W. J. Gutowski, W. B. Lyons, K. W. Potter, T. S. W., H. J. Vaux, C. J. Vorosmarty, C. Welty, C. A. Woodhouse, and C. Zheng (2012). *Challenges and Opportunities in the Hydrologic Sciences*. The National Academies Press.
- Horsburgh, J., D. Tarboton, D. Maidment, and I. Zaslavsky (2008). A relational model for environmental and water resources data. *Water Resources Research* 44(5), W05406.

- Horsburgh, J., D. Tarboton, M. Piasecki, D. Maidment, I. Zaslavsky, D. Valentine, and T. Whitenack (2009). An integrated system for publishing environmental observations data. *Environmental Modelling & Software* 24(8), 879–888.
- Hu, S. and L. Bian (2009). Interoperability of functions in environmental models—a case study in hydrological modeling. *International Journal of Geographical Information Science* 23(5), 657–681.
- Huhns, M. and M. Singh (2005). Service-oriented computing: Key concepts and principles. *Internet Computing, IEEE* 9(1), 75–81.
- Islam, A. and M. Piasecki (2008). Ontology based web simulation system for hydrodynamic modeling. *Simulation Modelling Practice and Theory* 16(7), 754–767.
- Jakeman, A. and R. Letcher (2003). Integrated assessment and modelling: features, principles and examples for catchment management. *Environmental Modelling & Software* 18(6), 491–501.
- Janssen, S., I. Athanasiadis, I. Bezlepina, R. Knapen, H. Li, I. Domínguez, A. Rizzoli, and M. van Ittersum (2011). Linking models for assessing agricultural land use change. *Computers and Electronics in Agriculture* 76(2), 148–160.
- Janssen, S., F. Ewert, H. Li, I. Athanasiadis, J. Wien, O. Thérond, M. Knapen, I. Bezlepina, J. Alkan-Olsson, A. Rizzoli, et al. (2009). Defining assessment projects and scenarios for policy support: Use of ontology in integrated assessment and modelling. *Environmental Modelling & Software* 24(12), 1491–1500.
- Jurisica, I., J. Mylopoulos, and E. Yu (1999). Using ontologies for knowledge management: An information systems perspective. In *Proceedings of the Annual Meeting-American Society For Information Science*, Volume 36, pp. 482–496. Information Today; 1998.

- Kim, H. (2005). Ontoweb: Implementing an ontology-based web retrieval system. *Journal of the American Society for Information Science and Technology* 56(11), 1167–1176.
- Knapen, R., S. Janssen, O. Roosenschoon, P. Verweij, W. de Winter, M. Uiterwijk, and J. Wien (2012). Evaluating openmi as a model integration platform across disciplines. *Environmental Modelling & Software*.
- Koivunen, M. (2001). W3c semantic web activity. *Semantic Web KickOff in Finland*, 27–41.
- Konstantinou, N., D. Spanos, and N. Mitrou (2008). Ontology and database mapping: a survey of current implementations and future directions. *Journal of Web Engineering* 7(1), 1–24.
- Kumar, P. (2006). *Hydroinformatics : data integrative approaches in computation, analysis, and modeling*. Boca Raton: Taylor & Francis.
- Leavesley, G. H., S. L. Markstrom, M. S. Brewer, and R. J. Viger (1996). The modular modeling system (MMS) -The physical process modeling component of a database-centered decision support system for water and power management. *Water, Air, & Soil Pollution* 90(1-2), 303–311.
- Letcher, R., A. Jakeman, and B. Croke (2004). Model development for integrated assessment of water allocation options. *Water Resources Research* 40(5), W05502.
- Loucks, D. P., E. v. Beek, J. R. Stedinger, J. P. M. Dijkman, and M. T. Villars (2005). *Water resources systems planning and management : an introduction to methods, models and applications*. Paris: UNESCO.
- Maidment, D. (2008). Bringing water data together. *Journal of Water Resources Planning and Management* 134, 95.

- Maidment, D. et al. (1992). *Handbook of Hydrology*. McGraw-Hill Inc.
- McGuinness, D., F. Van Harmelen, et al. (2004). OWL Web Ontology Language overview. *W3C Recommendation 10*, 2004–03.
- Miller, R. C., D. P. Guertin, and P. Heilman (2004). Information technology in watershed management decision making. *Journal of the American Water Resources Association* 40(2), 347–357.
- Moore, R. and C. Tindall (2005). An overview of the Open Modelling Interface and environment (OpenMI). *Environmental Science & Policy* 8(3), 279–286.
- Motik, B., P. Patel-Schneider, B. Parsia, C. Bock, A. Fokoue, P. Haase, R. Hoekstra, I. Horrocks, A. Ruttenberg, U. Sattler, et al. (2009). OWL 2 web ontology language: Structural specification and functional-style syntax. *W3C Recommendation 27*.
- Patil, L., D. Dutta, and R. Sriram (2005). Ontology formalization of product semantics for product lifecycle management. In *Proc. ASME/IDETC CIE Conf.*
- Peckham, S., E. Hutton, and B. Norris (2012). A component-based approach to integrated modeling in the geosciences: The design of csdms. *Computers & Geosciences*.
- Piasecki, M. and L. Bermudez (2003). Hydroml: Conceptual development of a hydrologic markup language. In *XXX IAHR Congress*, pp. 24–29.
- Priss, U. (2006). Formal concept analysis in information science. *Annual Review of Information Science and Technology* 40, 521.
- Prud’Hommeaux, E., A. Seaborne, et al. (2008). Sparql query language for rdf. *W3C recommendation 15*.
- Pullar, D. and D. Springer (2000). Towards integrating GIS and catchment models. *Environmental Modelling and Software* 15(5), 451–459.

- Rahman, J. a. (2003). It's time for a new environmental modelling framework. *Journal of the American Water Resources Association* 4(5723), 1727–1732.
- Raskin, R. G. and M. J. Pan (2005, November). Knowledge representation in the Semantic Web for Earth and Environmental Terminology (SWEET). *Computers & Geosciences* 31(9), 1119–1125.
- Ravi, V., J. R. Williams, Y. Ouyang, et al. (1998). Estimation of infiltration rate in the vadose zone: compilation of simple mathematical models.
- Reussner, F., J. Alex, M. Bach, M. Sch  
"utze, and D. Muschalla (2009). Basin-wide integrated modelling via OpenMI considering multiple urban catchments. *Water science and technology* 60(5), 1241–1248.
- Rizzoli, A., M. Donatelli, I. Athanasiadis, F. Villa, and D. Huber (2008). Semantic links in integrated modelling frameworks. *Mathematics and Computers in Simulation* 78(2), 412–423.
- Rizzoli, A. and W. Young (1997). Delivering environmental decision support systems: software tools and techniques. *Environmental Modelling & Software* 12(2-3), 237–249.
- Russell, S. J., P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards (1995). *Artificial intelligence: a modern approach*, Volume 2. Prentice hall Englewood Cliffs, NJ.
- Scholten, H., A. Kassahun, J. C. Refsgaard, T. Kargas, C. Gavardinas, and A. J. Beulens (2007, May). A methodology to support multidisciplinary model-based water management. *Environmental Modelling & Software* 22, 743–759.
- Shampine, L. (2009). Stability of the leapfrog/midpoint method. *Applied Mathematics and Computation* 208(1), 293–298.

- Shaw, M. and D. Garlan (1996). *Software architecture: perspectives on an emerging discipline*. 1996. Prentice-Hall.
- Sinding, J. B. Gregersen, P. J. A. Gijssbers, R. Brinkman, and S. J. P. Westen (2005). the open document series: part f org.openmi.utilities technical documentation (version 1.0 - google search).
- Singh, V. (2006). *Watershed models*. Boca Raton: Taylor & Francis.
- Singh, V. and D. Woolhiser (2002). Mathematical modeling of watershed hydrology. *Journal of Hydrologic Engineering* 7(4), 270–292.
- Sirin, E., B. Parsia, B. Grau, A. Kalyanpur, and Y. Katz (2007). Pellet: A practical OWL-DL reasoner. *Web Semantics: science, services and agents on the World Wide Web* 5(2), 51–53.
- Sivakumar, B. (2010). Chaos theory for hydrologic modeling and forecasting: Progress and challenges.
- Srinivasan, R. and J. G. Arnold (1994, June). Integration of a Basin-Scale Water Quality Model With GIS. *Journal of the Astronomical Society of Western Australia* 30, 453–462.
- Steward, D. R., J. M. Peterson, X. Yang, T. Bulatewicz, M. Herrera-Rodriguez, D. Mao, and N. Hendricks (2009). Groundwater economics: An object-oriented foundation for integrated studies of irrigated agricultural systems. *Water Resources Research* 45(5).
- Stumme, G., R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal (2002). Computing iceberg concept lattices with titanic. *Data & knowledge engineering* 42(2), 189–222.
- Swayne, D., W. Yang, A. Voinov, A. Rizzoli, and T. Filatova (2010). Linking data, models and tools: An overview.

- Syvitski, J., E. Hutton, S. Peckham, and R. Slingerland (2011). CSDMS – A Modeling System to Aid Sedimentary Research. *The Sedimentary Record* 9(1), 4–9.
- Syvitski, J., R. Paola, R. Slingerland, D. Furbish, P. Wiberg, and G. Tucker (2004). Building a Community Surface Dynamics Modeling System (CSDMS): rationale and strategy. *A Report to the National Science Foundation, Penn State University, State College*, 41.
- Szyperski, C. (2002). *Component software : beyond object-oriented programming* (2nd ed. ed.). New York ,London ,Boston: ACM Press ,Addison-Wesley.
- Tague, C. and L. Band (2004). Rhessys: regional hydro-ecologic simulation system-an object-oriented approach to spatially distributed modeling of carbon, water, and nutrient cycling. *Earth Interactions* 8(19), 1–42.
- Tindall, C. (2005). An overview of the open modelling interface and environment (the OpenMI). *Environmental Science & Policy* 8(3), 279–286.
- Toro, E. (2009). *Riemann solvers and numerical methods for fluid dynamics* (3rd ed. ed.). New York: Springer.
- Uschold, M. and M. Gruninger (1996). Ontologies: Principles, methods and applications. *Knowledge Engineering Review* 11(2), 93–136.
- Villa, F., I. Athanasiadis, and A. Rizzoli (2009). Modelling with knowledge: A review of emerging semantic approaches to environmental modelling. *Environmental Modelling & Software* 24(5), 577–587.
- Voinov, A., C. DeLuca, R. Hood, S. Peckham, and C. Sherwood (2010). A Community Approach to Earth Systems Modeling. *Eos* 91, 13.
- Voinov, A. and H. Shugart (2012). ‘Integronsters’, integral and integrated modeling. *Environmental Modelling & Software In Press*.

- Voinov, A., I. Zaslavskiy, D. Arctur, C. Duffy, and R. Seppelt (2008). Community modelling, and data-model interoperability. In *International Congress on Environmental Modelling and Software (IEMSs)*, pp. 2035e2050.
- Völkel, M., M. Krötzsch, D. Vrandečić, H. Haller, and R. Studer (2006). Semantic wikipedia. In *Proceedings of the 15th international conference on World Wide Web*, pp. 585–594. ACM.
- Wagener, T., M. Sivapalan, P. Troch, B. McGlynn, C. Harman, H. Gupta, P. Kumar, P. Rao, N. Basu, and J. Wilson (2010). The future of hydrology: An evolving science for a changing world. *Water Resource Research* 46(5), 1–10.
- Wormuth, B. and P. Becker (2004). Introduction to formal concept analysis. In *2nd International Conference of Formal Concept Analysis February*, Volume 23.
- Wurbs, R. (1998). Dissemination of generalized water resources models in the united states. *Water International* 23(3), 190–198.
- Yang, A., W. Marquardt, and J. Morbach (2004). *From conceptualization to model generation: the roles of ontologies in process modeling*. LPT, RWTH.
- Zeng, M. (2005). Using software to teach thesaurus development and indexing in graduate programs of lis and iakm. *Bulletin of the American Society for Information Science and Technology* 31(6), 11–13.
- Zhong, J., A. Aydin, and D. McGuinness (2009). Ontology of fractures. *Journal of Structural Geology* 31(3), 251–259.



## APPENDIX A

### PERMISSION TO REPRINT CHAPTER 3

This is a license agreement between the author and John Wiley and Sons (“John Wiley and Sons”) provided by Copyright Clearance Center (“CCC”). The license gives the permission to Mostafa M. Elag to use and reprint the article introduced in chapter 3 in his dissertation.

## JOHN WILEY AND SONS LICENSE TERMS AND CONDITIONS

Jun 20, 2013

---

This is a License Agreement between Mostafa M Elag ("You") and John Wiley and Sons ("John Wiley and Sons") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by John Wiley and Sons, and the payment terms and conditions.

**All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.**

License Number	3173080095585
License date	Jun 20, 2013
Licensed content publisher	John Wiley and Sons
Licensed content publication	Water Resources Research
Licensed content title	Feedback loops and temporal misalignment in component-based hydrologic modeling
Licensed copyright line	Copyright 2011 by the American Geophysical Union
Licensed content author	Mostafa M. Elag, Jonathan L. Goodall, Anthony M. Castronova
Licensed content date	Dec 16, 2011
Start page	n/a
End page	n/a
Type of use	Dissertation/Thesis
Requestor type	Author of this Wiley article
Format	Print and electronic
Portion	Full article
Will you be translating?	No
Total	0.00 USD
Terms and Conditions	

### TERMS AND CONDITIONS

This copyrighted material is owned by or exclusively licensed to John Wiley & Sons, Inc. or one of its group companies (each a "Wiley Company") or a society for whom a Wiley Company has exclusive publishing rights in relation to a particular journal (collectively "WILEY"). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the billing and payment terms and conditions established by the Copyright Clearance Center Inc., ("CCC's Billing and Payment terms and conditions"), at the time that you opened your RightsLink account (these are available at any time at <http://myaccount.copyright.com>).