The Thesis Committee for Stephen Robert Jackson
Certifies that this is the approved version of the following thesis:

RiverML: A Harmonized Transfer Language for River Hydraulic Models

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

David R. Maidment

Ben R. Hodges

RiverML: A Harmonized Transfer Language for River Hydraulic Models

by

Stephen Robert Jackson, B.A.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

August, 2014

Dedication

This thesis is dedicated to my parents, who gave me the tools to succeed, and the memory of my grandfathers, both of whom were respected engineers.

Acknowledgements

This work has benefited from the input from a great number of people around the world. The OGC/WMO Hydrology Domain Working Group, the CUAHSI HydroShare team, the University of Texas CRWR Research Team, as well as an assorted group of industry and agency professionals provided valuable advice and expertise throughout the development process.

In particular I would like to thank David Maidment, Ben Hodges, Paola Passalacqua, David Tarboton, David Arctur, Tim Whiteaker, Fernando Salas, Johnny Sullivan, Alison Wood, Dean Djokic, Peter Taylor, David Valentine, Irina Dornblut, Pete Singhofen, David Blodgett, Sean Hodges, Brent Watson, Venkatesh Merwade, and Dongsu Kim for their contributions and encouragements.

Support for this research was provided by the National Science Foundation under NSF Prime Award OCI-1148453, “Collaborative Research: SI2-SSI: An interactive software infrastructure for sustaining collaborative community innovation in the hydrologic sciences.”

Abstract

RiverML: A Harmonized Transfer Language for River Hydraulic Models

Stephen Robert Jackson, M.S.E.
The University of Texas at Austin, 2014

Supervisor: David R. Maidment

The multitude of data formats for storing river network, geometry, and flow data presents a challenge for the sharing of information both internally between software applications and externally between agencies. An analysis of existing software applications and data models used for one-dimensional hydraulic modelling of river systems was performed. The commonalities and differences between the model inputs were identified in order to determine the necessary characteristics of a common transfer language. A prototype transfer language was developed using Unified Modeling Language (UML) and implemented as an Extensible Markup Language (XML) schema. This prototype is intended to serve as a first step towards developing an international open standard to facilitate the sharing of hydraulic data. This work was performed in cooperation with the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) and the Open Geospatial Consortium/World Meteorological Organisation Hydrology Domain Working Group.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 MOTIVATION

The past few decades have seen an evolution in the way water resource professionals have interacted with information about rivers. Throughout the 20th century, agencies such as the United States Geological Survey (USGS) developed comprehensive paper topographical maps, with rivers represented as blue lines. In the latter portion of the century, large efforts were made to digitize these paper maps, and to organize the various types of information, such as contours, roadways, and rivers, into thematic layers which can be downloaded, analyzed, and viewed either separately or collectively. This allows users to interact with the data in any number of computer environments and develop derivative products such as flood risk maps. As both the amount and resolution of data increases, so do the challenges of finding appropriate data and loading it to an individual user's computer. More recently, there has been a trend toward 'data as a service', where data providers host large datasets in remote servers, and provide standardized methods for users to search and download specific data sets of interest. This allows data acquisition to be automated as part of a larger workflow. Data as a service enhances the ability for government agencies, academic researchers, and industry professionals to share the information required to generate hydraulic models of rivers, as well as the results from those models.

One challenge faced when sharing information is that many different software applications exist for modeling rivers, each of which stores the required model inputs and outputs in a custom format. Converting between the wide range of data formats is non-trivial, and represents a barrier which must be overcome before industry-wide interoperability with regards to hydraulic models is possible. One possible approach to solving this problem would be to develop automated translation tools. Such tools convert between a specific pair of data formats, and thus the number of required tools increases as $\frac{n!}{2\cdot(n-2)!}$, where n is the number of data formats (Kreyszig, 1999). An alternate approach is to create a single standard ‘harmonized’ transfer language specifically designed for cross-application data exchange. Harmonize[1] is a term used within the data community that means “to create the possibility to combine data from heterogeneous sources into integrated, consistent and unambiguous information products, in a way that is of no concern to the end-user (Flanders Marine Institute, 2013).” This research takes the latter approach and focuses on enabling interoperability in one-dimensional hydraulic river modelling through the creation of a standard transfer language for river geometry and flow.

1.2 BACKGROUND

The research presented in this thesis has been performed in cooperation with two organizations. The first is the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI), and the second is the OGC/WMO Hydrology Domain Working Group. These organizations each recognize the need for a standard transfer language for river hydraulic information, and are participating in the development process of such a language. The language currently under development has been named RiverML.

---

[1] The international spelling ‘harmonise’ is often used in the literature.

1.2.1 CUAHSI HydroShare

This work is supported primarily with funding from the National Science Foundation (NSF) as part of a project called HydroShare. HydroShare is an initiative of CUAHSI to provide a collaborative web space for hydrologic scientists to share data and models. The NSF project proposal for HydroShare contains the following:

> As an exemplar for advancing data access, we will establish a national repository within HydroShare for **river channel cross section data**: a new data type not presently supported by CUAHSI HIS. Since 2003, the United States has spent more than $2 billion on digital flood map modernization. A great deal of river channel cross-section, morphology and hydraulic modeling data has been developed to support this mapping and some of that could be repurposed to advance water science. This repository will include a mechanism for voluntary submission of information and it will **provide access to this data in a standard way** such that it is easy to run hydraulic models that use this data on either local or HPC environments. (Tarboton et al., 2011, emphasis added)

A review of existing formats for cross section data revealed no suitable candidates for such a standard. Therefore, in order to fulfil this portion of the proposal, development of a standard river data format is required. This standard format for cross section and profile line data should support the interoperability which is central to a services oriented architecture. It should have a clear logical structure which is applicable to many modeling tasks, and include both data and metadata so that the fitness for use can be readily determined.

1.2.2 OGC/WMO Hydrology Domain Working Group

In 2009, a partnership between the Open Geospatial Consortium (OGC) and the World Meteorological Organisation (WMO) was begun to improve the mechanisms for sharing water information (Lemon and Maidment, 2009). The resulting OGC/WMO Hydrology Domain Working Group (Hydrology DWG) has since been active in the development and promotion of international standards for water data with participation from academic institutions, government agencies, and industry partners from around the world. One such contribution was the development of WaterML 2.0, a standard for the communication of time series data, through a three-pronged approach. First, existing formats were identified and compared in a harmonization study. Second, WaterML 2.0 was created by utilizing existing OGC and ISO standards as building blocks. Third, the candidate standard was tested and refined in a series of Interoperability Experiments where interested parties implemented WaterML 2.0 for particular scenarios and provided feedback (Taylor et al., 2014).

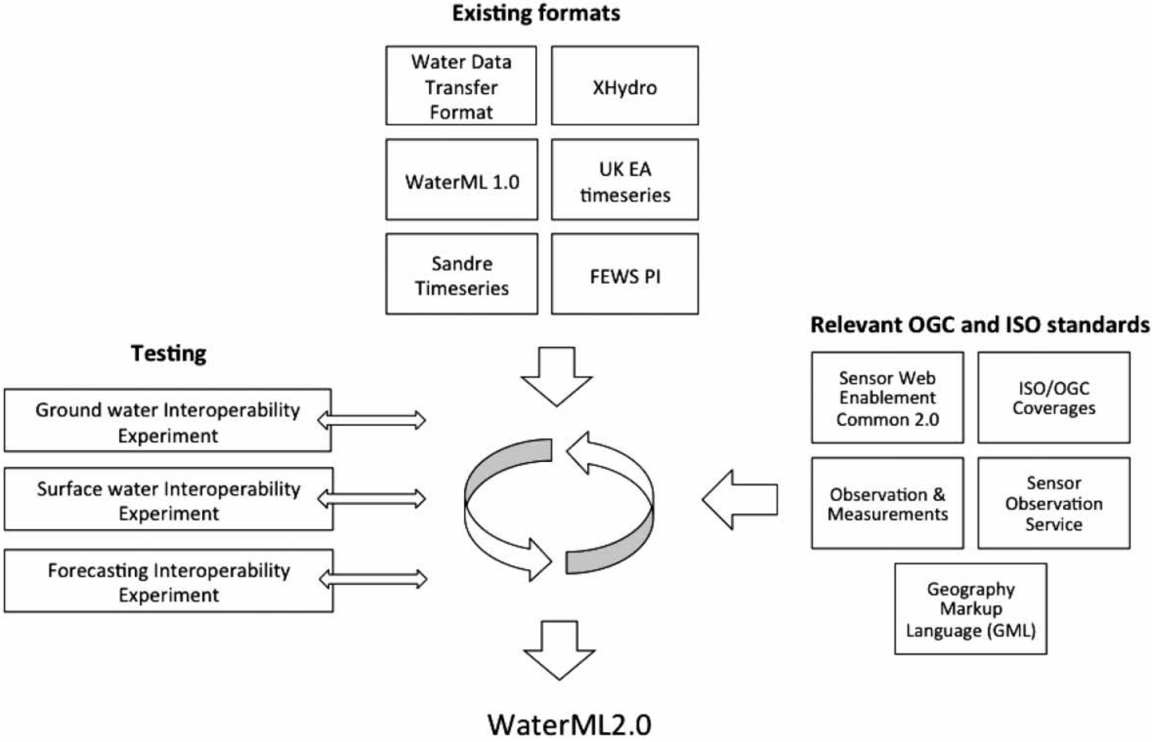


Figure 1.1 WaterML 2.0 Harmonization Process (Taylor et al., 2014)

1.2.3 RiverML

At the June 2013 meeting of the Hydrology DWG in Quebec City, a proposal presented by the author to begin development of a standard language for river geometry and flow was approved. This standard would be designed to meet the needs of the CUAHSI HydroShare project, as well as those of the broader worldwide water resources community represented by the OGC and WMO. The language was named RiverML, which stands for River Markup Language (following the convention of HTML, XML, and WaterML). The proposed development path for RiverML follows the successful precedent of WaterML 2.0, beginning with a harmonization effort and prototype, proceeding to a set of formal Interoperability Experiments, and leading to the adoption of an international open standard which can be implemented by water resource professionals around the world. This paper represents the first stage in the development of RiverML, offering two primary contributions: a harmonization of existing technologies and the description of a prototype schema.

1.3 OBJECTIVES AND SCOPE

1.3.1 Objectives

The first goal of this paper is to harmonize existing technologies for one-dimensional hydraulic models. Similar to the harmonization effort for water observation data overseen by Taylor (2010), this involves analyzing existing hydraulic software applications and data models to determine whether the various information frameworks are similar enough to support a common transfer language. The commonalities and differences between the model inputs are identified in order to determine the necessary characteristics of such a language.

The second goal of this paper is to describe RiverML 0.2, a prototype transfer language for river models based on the findings of the harmonization effort. This language is designed to support interoperability between any combination of terrain processing software, hydrologic calculation software, and hydraulic software (see Figure 1.2). Once reviewed and revised as necessary by the Hydrology DWG and CUAHSI members, the prototype can serve as the basis for a set of Interoperability Experiments leading to an officially adopted international standard.

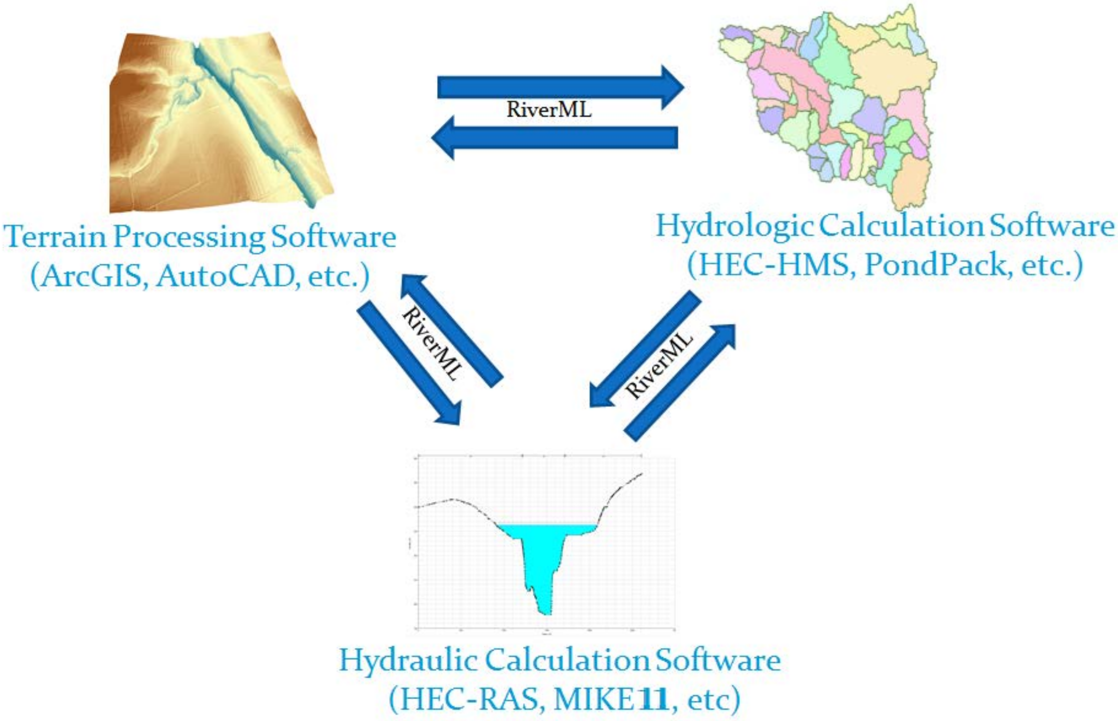


Figure 1.2 Interoperability through a standard transfer language

1.3.2 Scope

This research focuses on the information central to one-dimensional hydraulic modeling, including cross section and profile line geometry, network connectivity, and properties such as roughness coefficients, levees, and obstructions. Purely two-dimensional models are out of scope for this research, although some consideration is given to the integration of one- and two-dimensional models. The geometry and network connectivity of reservoirs has been given preliminary analysis in this research, but would benefit from more detailed study.

Some time-varying properties of rivers such as discharge and water surface elevation, which are central to hydraulic modeling, can be communicated through the existing WaterML 2.0 standard language. The scope of this research includes integrating WaterML 2.0 time series into a framework of river geometry. Methods of communicating other time-varying properties of rivers such as the shape of cross sections and the connectivity of river reaches are within the scope of this research.

The description of hydraulic structures such as bridges, dams, culverts, and pipe networks are out of scope for this research, except for a preliminary treatment of simple weirs used as reservoir inlets and outlets. The interactions between surface water and subsurface water are out of scope. The inputs required for hydrologic models such as drainage area, precipitation, and land use are out of scope for this research. Future versions of RiverML can be expanded to include a more complete representation of the water cycle.

With regards to software, a detailed analysis of terrain processing, such as feature extraction and floodplain interpolation, is out of scope for this paper. ArcGIS is presented as a sample software application which can perform such tasks in order to provide appropriate context. Likewise, a detailed analysis of hydrology is out of scope for this paper. HEC-HMS is presented as a sample software application which can perform hydrologic computations in order to illustrate the use of schematic networks and to identify a common source of the time series inputs required by hydraulic models.

1.4 OUTLINE

A technology review is presented in Chapter 2 in which a set of software applications and data models used for hydraulic modeling are identified and briefly described. Several authoritative data model implementations as well as methods for coupling computational models are included in this review. The input requirements for the hydraulic applications and data models are analyzed in Chapter 3 to determine commonalities and differences. Where differences are found, it is examined whether these are matters of convention or whether they represent fundamentally different information frameworks. From this analysis, a set of key challenges are identified which must be addressed in order to maximize interoperability. Chapter 4 contains a detailed description of a prototype transfer language which is based on the harmonization findings of Chapter 3. An example demonstrating the use of the prototype language is presented in Chapter 5. Conclusions and recommendations for future work can be found in Chapter 6, including a detailed list of suggestions for improving the RiverML prototype and a description of the process toward the adoption of RiverML as an official international open standard.

Chapter 2: Technology Review

2.1 TECHNOLOGY REVIEW OVERVIEW

In this chapter, a representative sample of the technology used for one-dimensional hydraulic models of rivers is presented. Each technology is briefly described as it relates to this research. The goal of this chapter is to familiarize the reader with the heterogeneous sources of data that must be harmonized prior to creating a standard transfer language. Three types of data of particular interest to this research are methods of describing features, methods of describing connectivity between features, and methods of describing changes in features over time.

Hydrologic Information Systems are discussed first as an overarching approach to modeling that may be made up of various specific combinations of technologies. Next, examples of such technologies are identified including software applications which either perform hydraulic computations or support such computations, data models for storing hydraulic information, and methods for coupling multiple computational models. Finally, a sample of authoritative river data sets published by agencies with regional or national jurisdiction are described.

2.2 HYDROLOGIC INFORMATION SYSTEMS

Over the past few decades, significant effort has been put toward developing practical frameworks for organizing and utilizing water-related data, and evolving these frameworks to keep pace with the advances in computer hardware and software. These frameworks have been dubbed Hydrologic Information Systems (HIS) and they combine two functions: data management and simulation (Obenour and Maidment, 2004; Whiteaker, 2004).

Data management involves both the storage of individual files on disk, and defining logical relationships between different classes of data. These aspects are often handled simultaneously using a database. A special type of database that is designed for information with a geospatial component is called a geodatabase. Geographic Information Systems (GIS) are software applications designed to interact with geospatial data. A GIS may incorporate tools for water resources applications, as with ArcGIS (Ackerman, 2009), or be specifically designed for water resources, as with Hydro Desktop (Ames et al., 2012).

Simulation involves using some computational engine to model real-world processes in order to make predictions (Obenour and Maidment, 2004). These models may be developed commercially, by various agencies, or be custom designed by individual users. Simulation models are often linked together in series, such as when using the outputs of a hydrologic model to drive a hydraulic model.

2.2.1 Services-Oriented Architecture

The precise nature of a given HIS will depend on many factors, such as the intended use of the system, the chosen data management and computations software applications, and the preferences of those in charge of administering the system. Providers and users may be operating under different logical frameworks of data organization, as well as different file formats for data storage. The “data as a service” approach addresses this problem by introducing a conceptual separation between transfer and storage. Service architecture works to standardize the data request/data transfer process.

> The *services-oriented architecture* (SOA) relies on a collection of loosely coupled self-contained services that communicate with each other and can be called from

> multiple clients in a standard fashion. Services provide a useful abstraction for functionality accessible over the web, by establishing a standard protocol (e.g. SOAP – Simple Object Access Protocol, or REST – REpresentational State Transfer) for invoking services irrespective of their underlying language, and by establishing a standard "contract" between a service provider and service client that can be used to formulate correct requests against a service (e.g. WSDL – Web Services Description Language). Common benefits associated with services-oriented architecture include: scalability, security, easier monitoring and auditing; standards-reliance; interoperability across a range of resources; plug-and-play interfaces. Internal service complexity is hidden from service clients, and backend processing is decoupled from client applications. (Maidment, 2005)

Provided the selected format can support it, services can be configured to communicate both data and metadata simultaneously. Once a user has a local copy of the data in a standard format, it can be converted to the desired storage format (generally through the use of automated tools), and integrated into the local HIS.

2.3 SOFTWARE APPLICATIONS

Most of the software applications examined here are complex, supporting varied workflows and optional parameters for river modeling. They include data models, computational models and a user interface bundled into a single application.

2.3.1 ArcGIS

ArcGIS is a general purpose GIS application developed by ESRI. It recognizes many different file formats, including TINs, rasters, and shapefiles. The Data Interoperability extension includes the ability to import from and export to GML (described in Section 2.4.2). The native geoprocessing functionality includes a large number of tools which can be applied to flexible workflows, and extended through plug-ins and custom-programmed tools. ArcGIS is often used as a data management, pre-processing, and post-processing platform for water resource modeling, with hydrologic and hydraulic computations performed by external software.

ArcGIS contains many tools for viewing and analyzing the topographic shape of the terrain represented as a Digital Elevation Model (DEM) or a Digital Terrain Model (DTM)[2]. A DEM, also called a raster or grid, specifies a value for elevation at regularly spaced intervals. A DTM, also called a Triangulated Irregular Network (TIN), specifies a value for elevation at irregularly spaced intervals, and can be enhanced using features such as breaklines. For this paper, DEMs and DTMs are assumed to be bare-earth models, where vegetation has been removed. High resolution surfaces can be generated using LiDAR aerial surveys that generate a dense cloud of elevation points which can then be converted to a DEM or DTM coverage (see Figure 2.1).

Many of the types of information used in water resources are derived from topography, such as watershed boundaries, channel centerlines, and channel banks. The methods of feature extraction vary depending on the format of the topographic model, and new methods are being developed which are specifically designed for high resolution datasets such as those obtained from LiDAR surveys. Point, line, and polygon features extracted from a DEM or DTM can be represented in ArcGIS as shapefiles (see Section 2.4.1).

---

[2] The terms DEM and DTM are given various definitions throughout the literature.

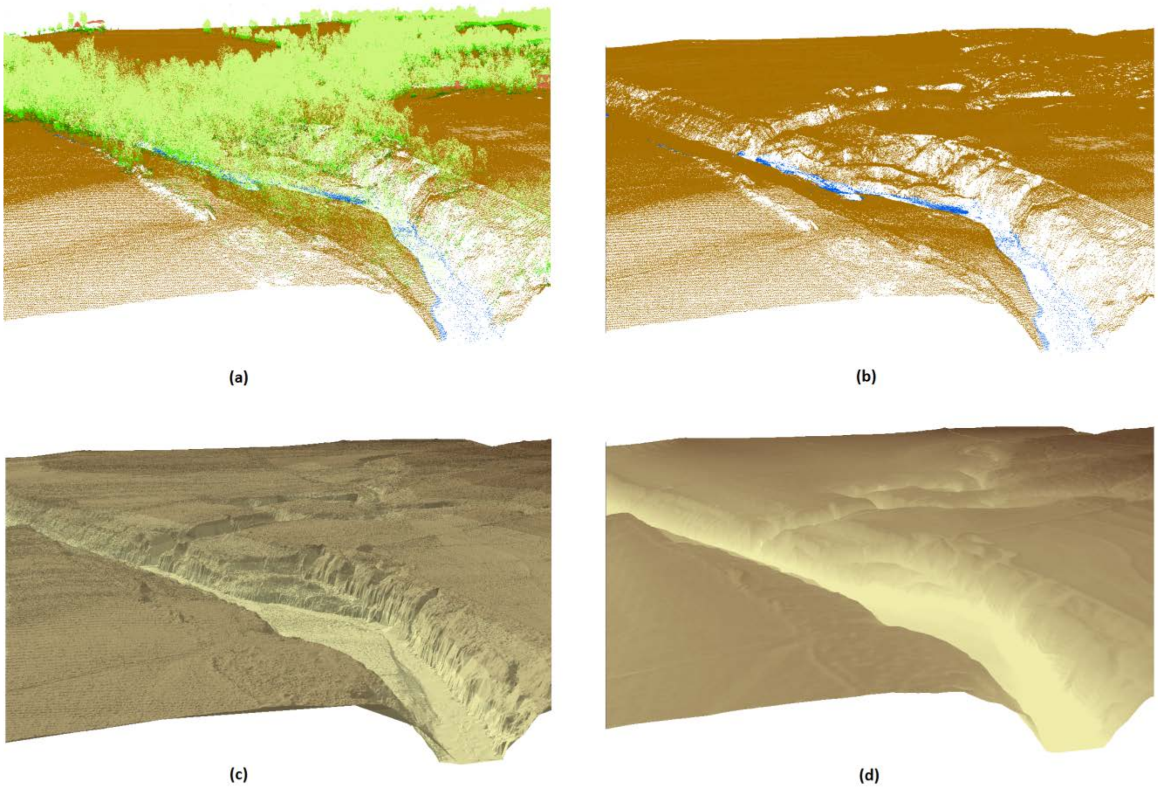


Figure 2.1 Digital elevation formats: (a) LiDAR point cloud, (b) LiDAR point cloud with vegetation and buildings removed, (c) DTM, (d) DEM

2.3.2 HEC-HMS

Hydrologic Engineering Center’s Hydrologic Modeling System (HEC-HMS), created by the US Army Corps of Engineers, is a hydrologic simulation application for infiltration and runoff of precipitation over watersheds (see Figure 2.2). HEC-HMS can be used to generate time series of discharge at selected analysis points that can serve as inputs to a hydraulic model (see Figure 2.3).

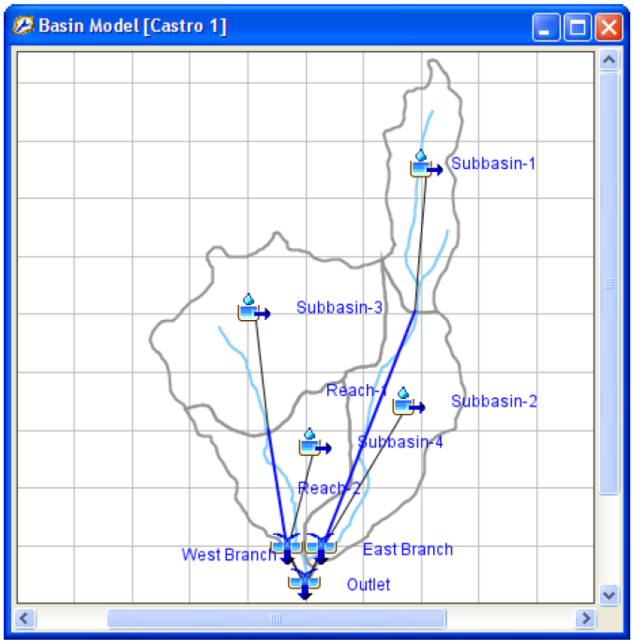


Figure 2.2 HEC-HMS Basin Model (Scharffenberg, 2013)

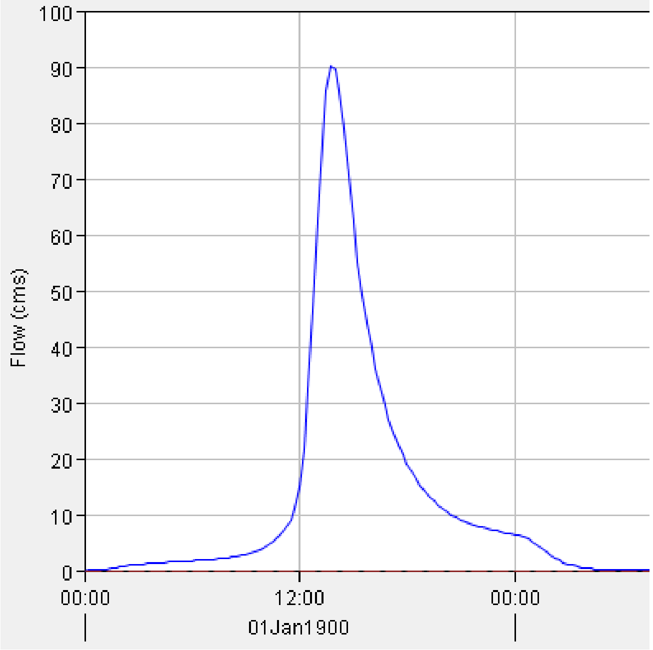


Figure 2.3 HEC-HMS discharge time series

2.3.3 HEC-RAS and HEC-GeoRAS

Hydrologic Engineering Center’s River Analysis System (HEC-RAS), created by the US Army Corps of Engineers, is a 1D hydraulic software application capable of performing steady flow, unsteady flow, sediment transport, and water quality analysis. Steady flow analysis is used primarily for determining water surface elevation for various scenarios, while unsteady flow is used to simulate the operation of structures such as pumps and dams and the failure of structures such as dams and levees (Brunner, 2010 b).

A plug-in for ArcGIS called HEC-GeoRAS has been developed which assists in the pre- and post-processing of data for HEC-RAS. HEC-GeoRAS allows a user to extract cross section and profile information from a DEM or DTM surface, assign network connectivity and roughness values, and export the data to HEC-RAS. The results from HEC-RAS can be imported to HEC-GeoRAS where water surface elevations at cross sections can be interpolated into a continuous floodplain extent polygon along the entire river (Ackerman, 2009). The transfer format used to accomplish this is discussed further in Section 2.4.10.

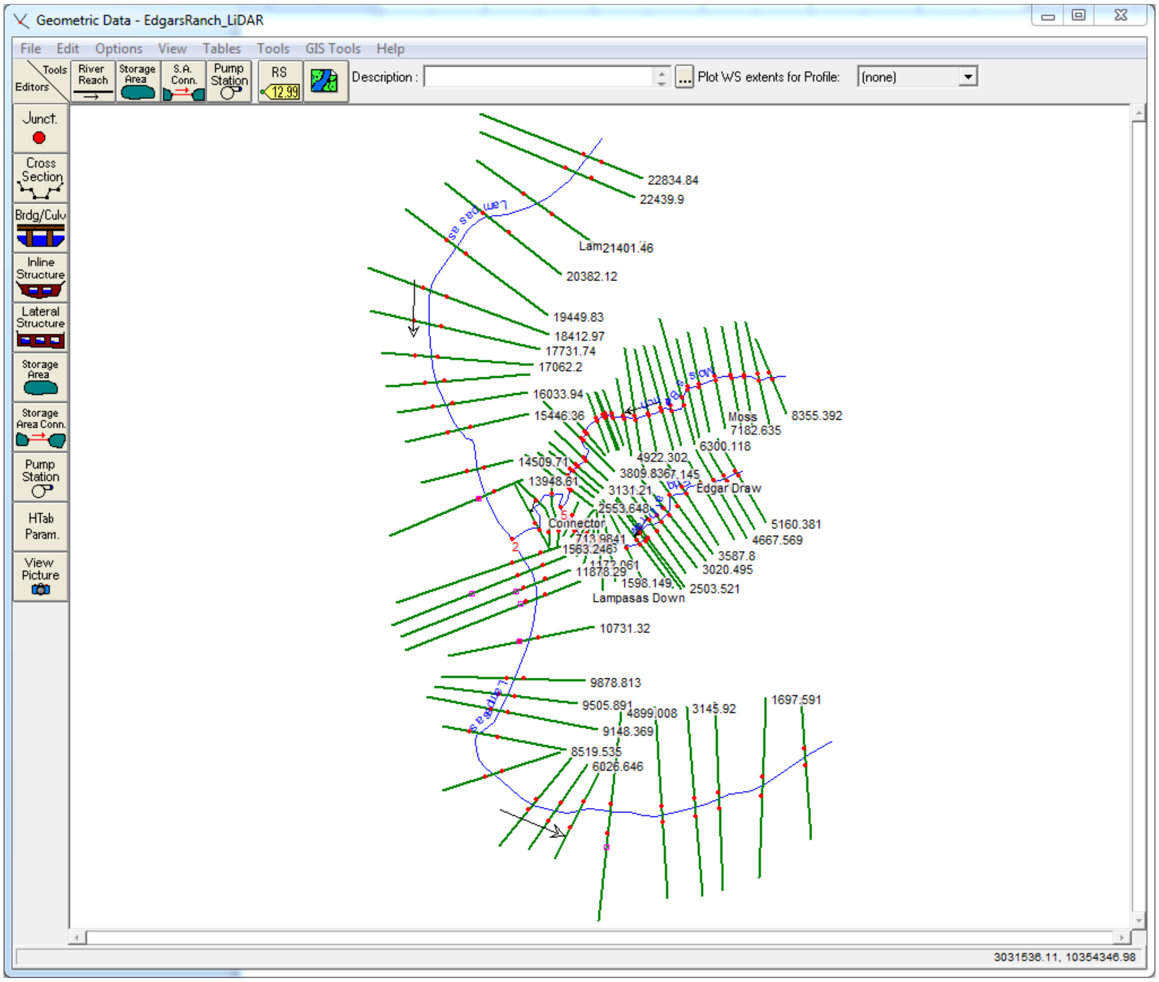


Figure 2.4 HEC-RAS network view

2.3.4 MIKE 11

DHI, a Danish company, has developed an integrated suite of water-related applications called MIKE. MIKE Zero serves as the user interface for a number of computational modules. MIKE 11 is the 1D hydraulic modeling module which is comparable in scope to HEC-RAS. The results of MIKE 11 are based on solutions to the Saint Venant equations (DHI, 2012 b). MIKE 11 has a defined ASCII format which can be used to import river network and topography data, allowing data generated externally from the software to be included in simulations. This format is explored further in Section 2.4.11 and Chapter 3.

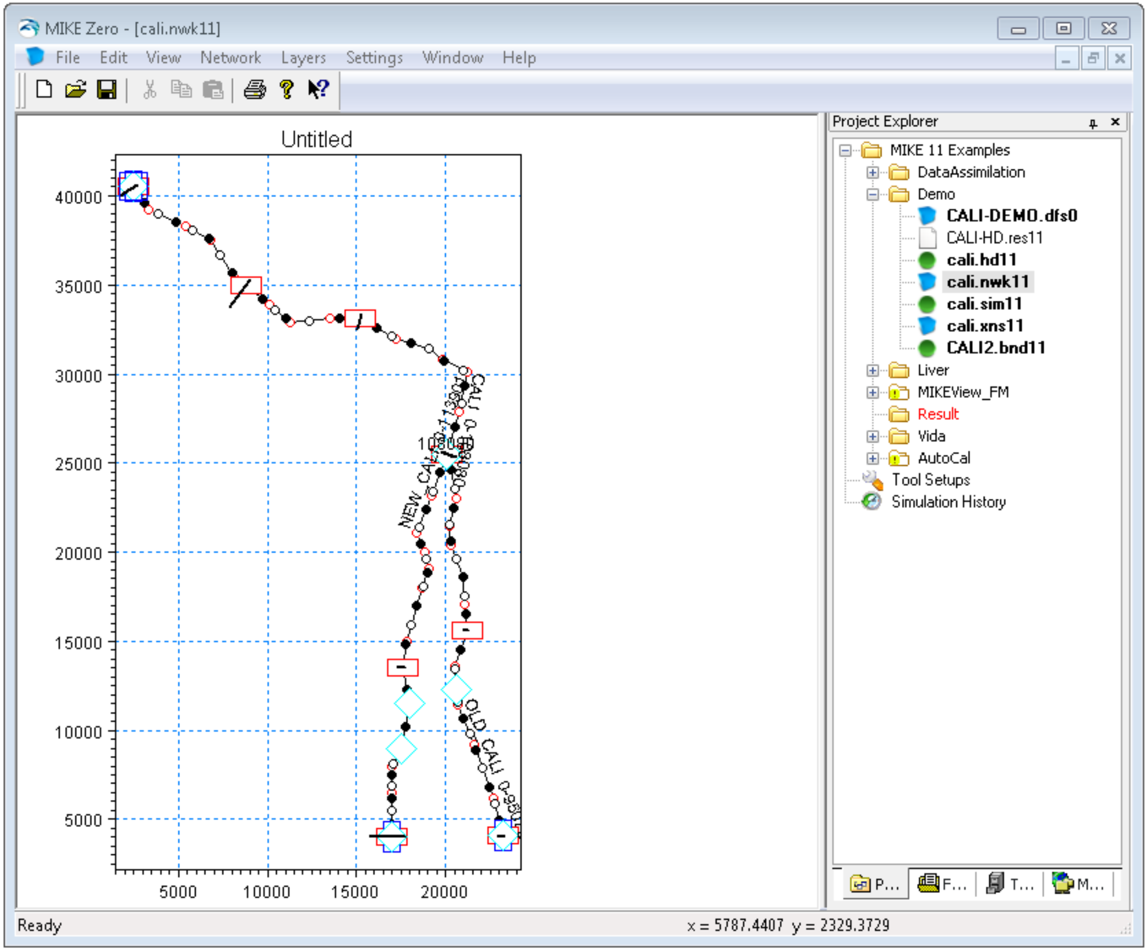


Figure 2.5 MIKE 11 network view

2.3.5 MIKE FLOOD

MIKE FLOOD is another module in the DHI suite, which couples two 1D modules (MIKE 11 and MIKE URBAN) with the 2D MIKE 21 module. This allows the strengths of both approaches to be utilized while mitigating the weaknesses. One-dimensional models provide the ability to handle large river networks efficiently, accurately treat structures, and allow the inclusion of pipe networks whose flow characteristics are not dependent on the surface terrain. Two-dimensional models provide the ability to simulate flow without artificially restricting the flow direction, and thus provide more realistic results for overland flow in urban areas and floodplains for sinuous rivers (DHI, 2012 d). Two applications of MIKE FLOOD are enhancing a 1D river model with a localized 2D model where overland flow is expected to be complicated (Figure 2.6), or enhancing a 2D urban model to capture channel behavior without excessively increasing the data resolution (Figure 2.7).

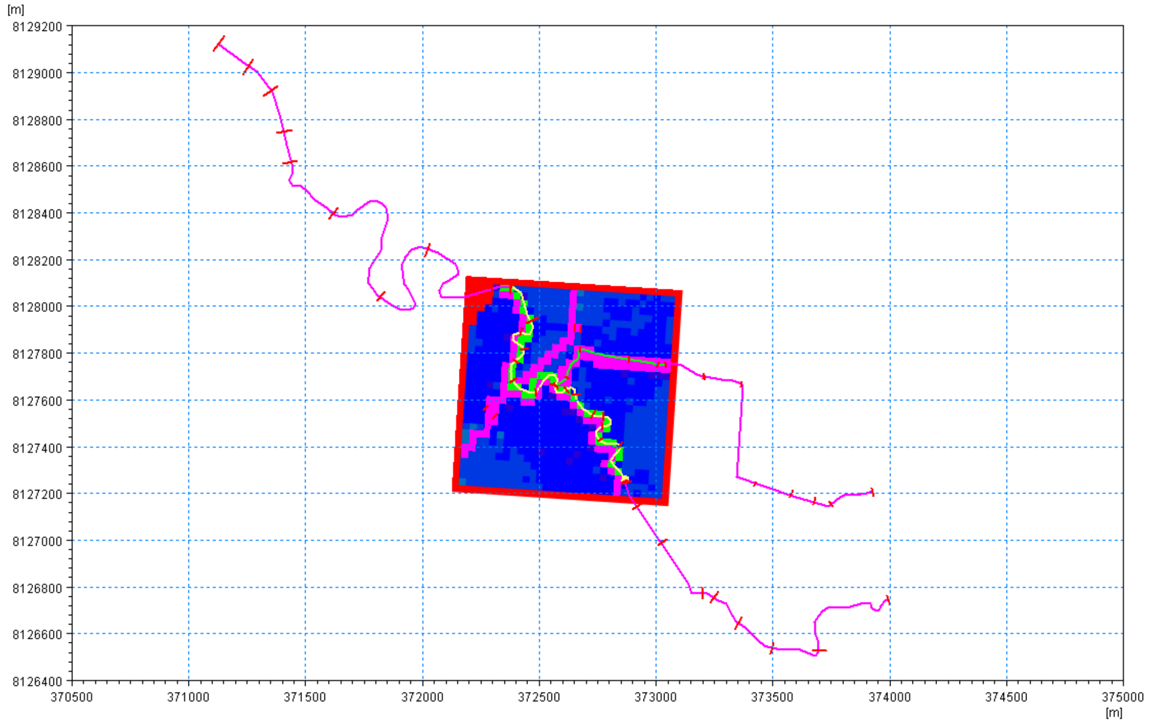


Figure 2.6 Coupled 1D and 2D model in MIKE FLOOD

MIKE FLOOD supports two different types of surfaces for 2D computations. The first is a rectangular grid (DEM), which is computationally simpler and common to many GIS applications. The downside of a grid is that it requires a high resolution in order to capture narrow channels, and provides less accurate results for channels when they are not aligned with the grid orientation. The second type is a flexible mesh, which allows the resolution to be spatially varied based on the complexity and importance of the region (DHI, 2012 d). The flexible mesh is conceptually similar to a TIN, except it supports triangular connections between nodes as well as higher order polygons (DHI, 2012 f). An example of a flexible mesh with a varying density of nodes, coupled with a 1D channel and a 1D pipe network is shown in Figure 2.7.

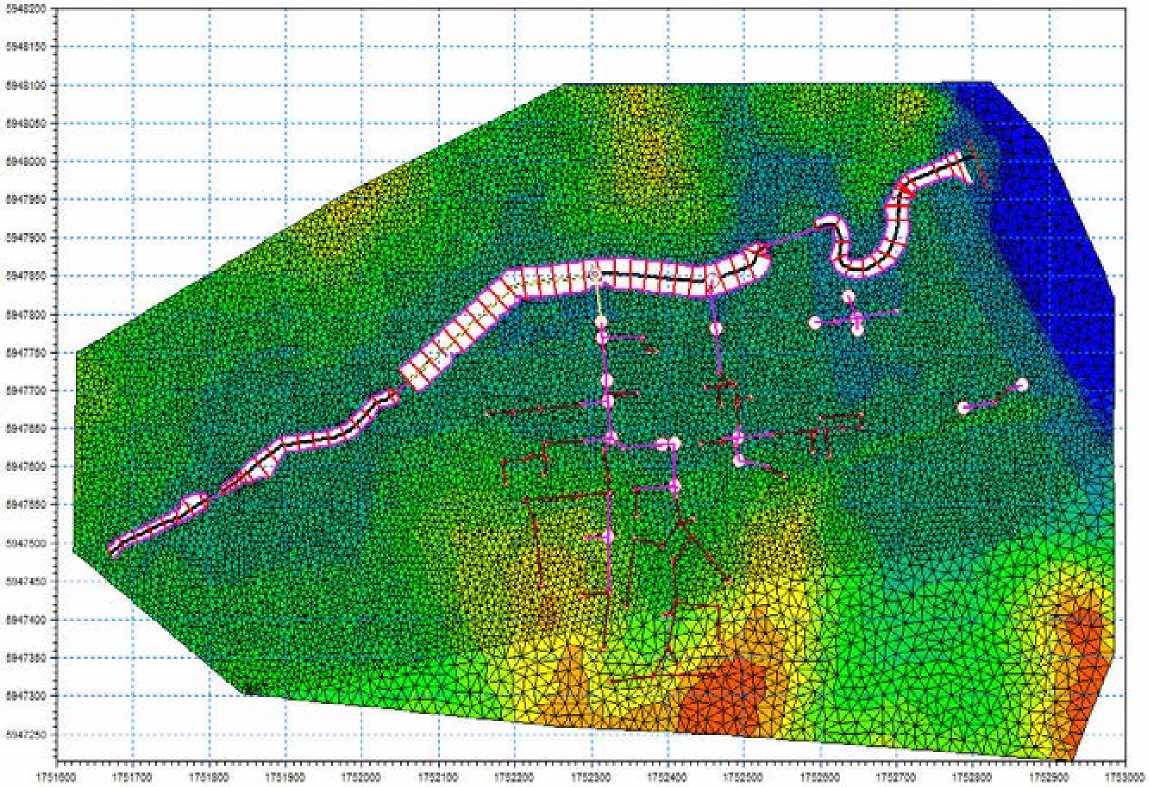


Figure 2.7 Coupled channel, pipe, and overland flow in MIKE FLOOD (DHI, 2012 d)

A number of different options are available in MIKE FLOOD to link the different models together. Of particular interest to this study are the standard link and the lateral link. The standard link (Figure 2.8) connects the end of a 1D river branch to one or more elements in a 2D grid or mesh. The lateral link (Figure 2.9) connects a portion of a 1D reach to the surrounding 2D terrain (DHI, 2012 d).

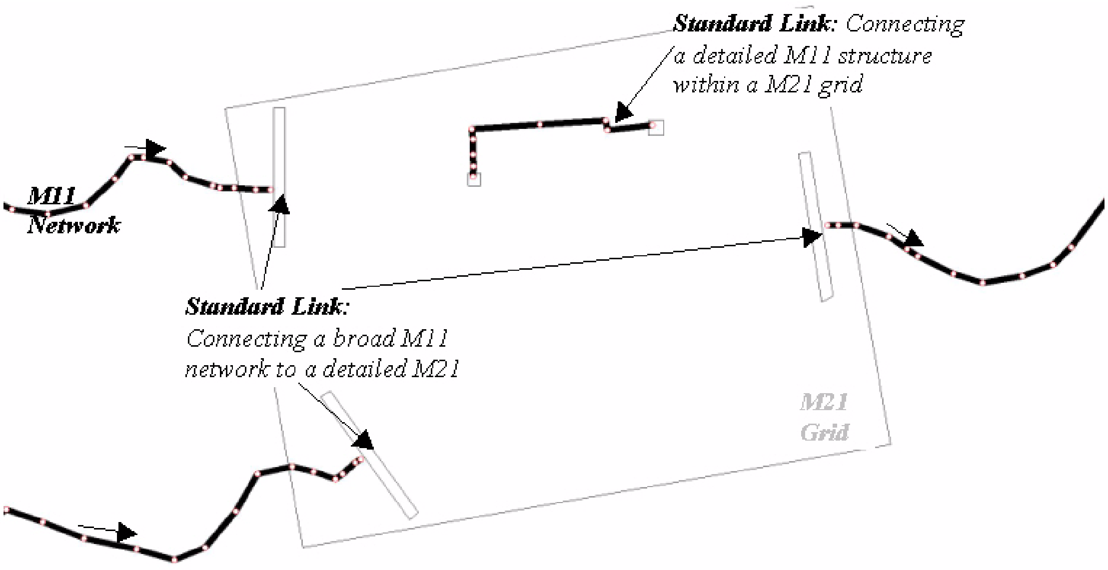


Figure 2.8 Standard link in MIKE FLOOD (DHI, 2012 d)

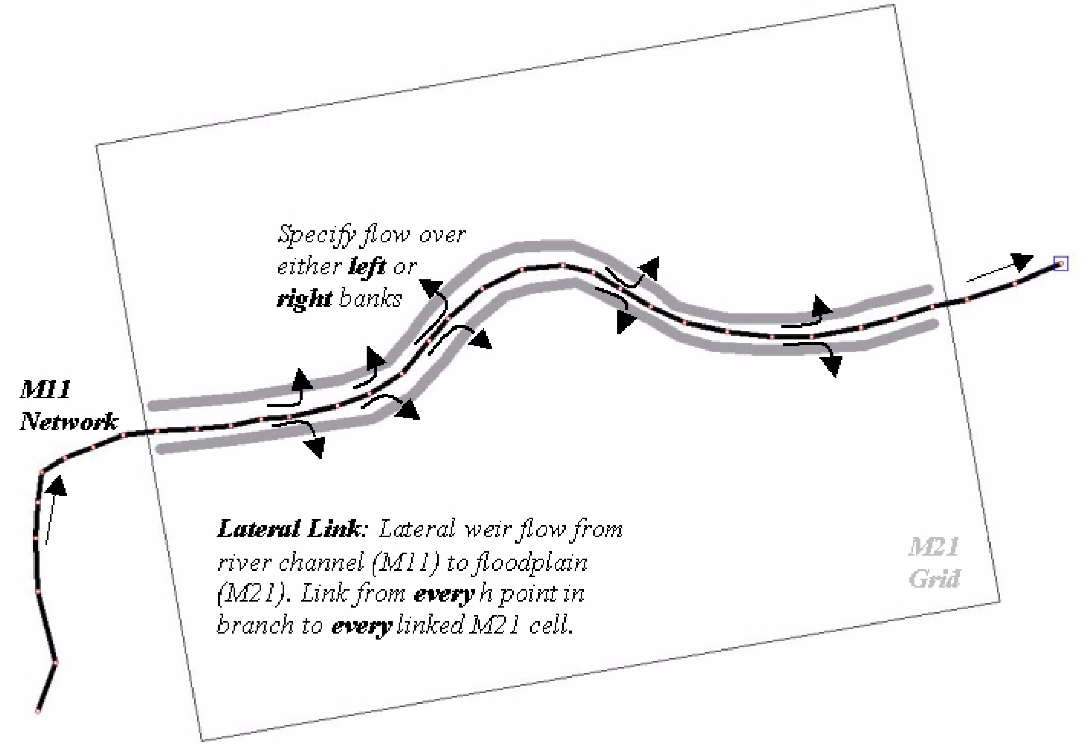


Figure 2.9 Lateral link in MIKE FLOOD (DHI, 2012 d)

2.3.6 ICPR

The Interconnected Channel and Pond Routing Model (ICPR), developed by Streamline Technologies, Inc., is another example of a software application which supports the coupling of one- and two-dimensional models (Figure 2.10). ICPR can combine precipitation, evapotranspiration, 2D surface flow, 1D channel flow, pond storage, and groundwater interactions into a single short-duration or long-duration simulation (Streamline Technologies, 2014 a). As of Version 4, ICPR is fully GIS enabled, with all elements supporting geo-referenced locations.

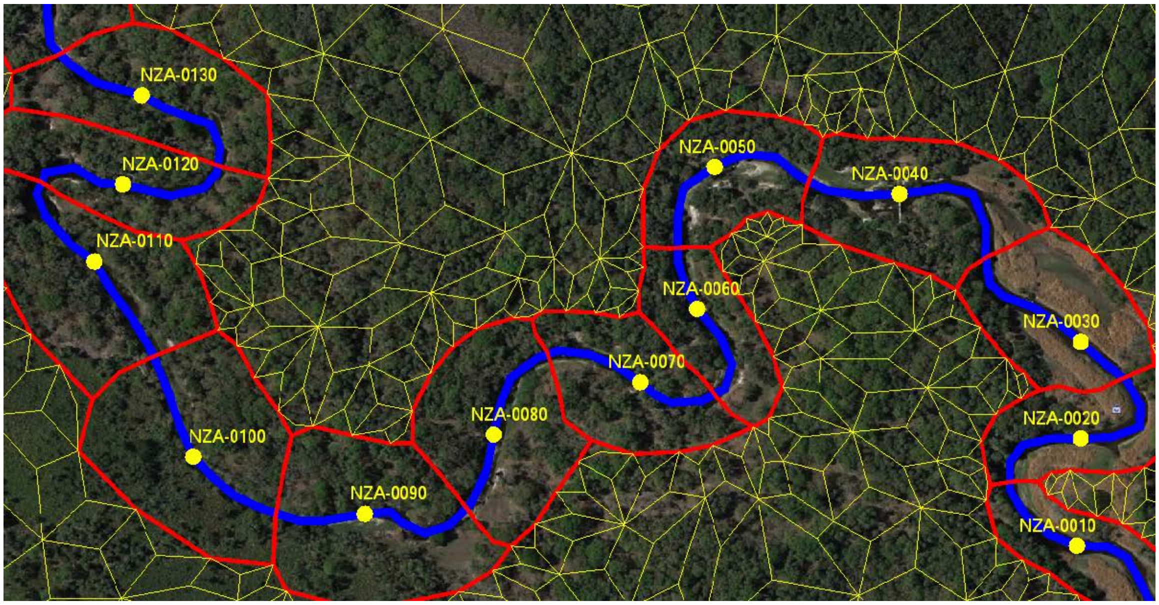


Figure 2.10 Coupled 1D and 2D model in ICPR (Streamline Technologies, 2014 b)

ICPR uses a gridded DEM as a surface input and automatically generates a triangular flexible mesh. This mesh can be refined by the user by adding breaklines and breakpoints, and other features. ICPR derives additional honeycomb and diamond meshes from the triangular mesh to serve as control volumes and roughness zones, respectively (Figure 2.11).

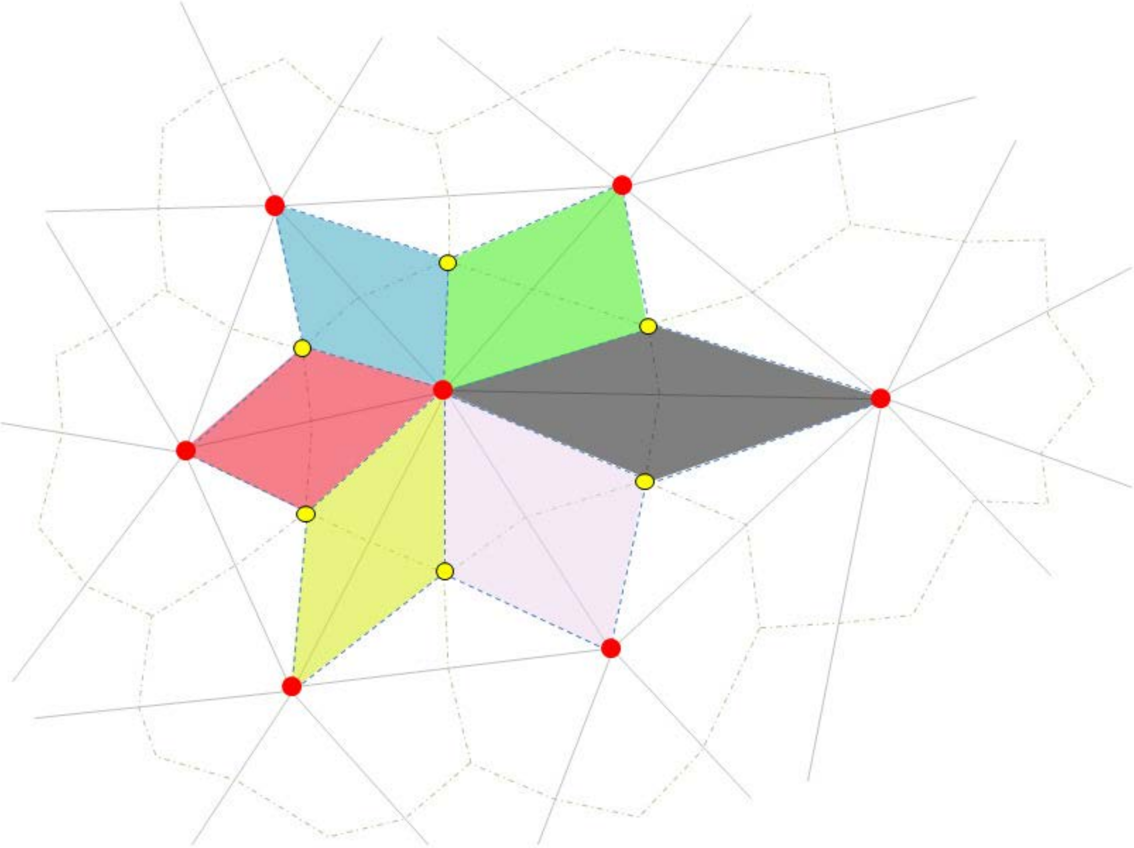


Figure 2.11 ICPR triangular, honeycomb, and diamond mesh (Streamline Technologies, 2014 a)

2.3.7 SPRNT

The Simulation Program for River Networks (SPRNT) is an open source research collaboration between The University of Texas at Austin and IBM (Hodges, 2014). It uses “state-of-the-art nonlinear numerical analysis algorithms developed from computer microchip simulation methods (Liu and Hodges, 2014)” to solve the Saint Venant equations for 1D channel flow. Due to the efficiency of these computations, it is able to perform large-scale simulations that are infeasible under traditional methods, and is also able to calibrate the selection of parameters such as Manning’s n (Liu and Hodges, 2014).

SPRNT serves as an example of a research-level, rather than commercial-level software application that would benefit from a standardized transfer format. As part of the development process for SPRNT, a custom input file format was defined. In order to test and use SPRNT on a large scale, custom tools need to be programmed which extract data from a GIS environment and convert it into the proper format. With a limited number of researchers and a limited budget, the creation and adoption of innovative tools is slowed. Once a standardized format for river geometry and flow exists, researchers can build their tools accordingly and be immediately interoperable with established workflows.

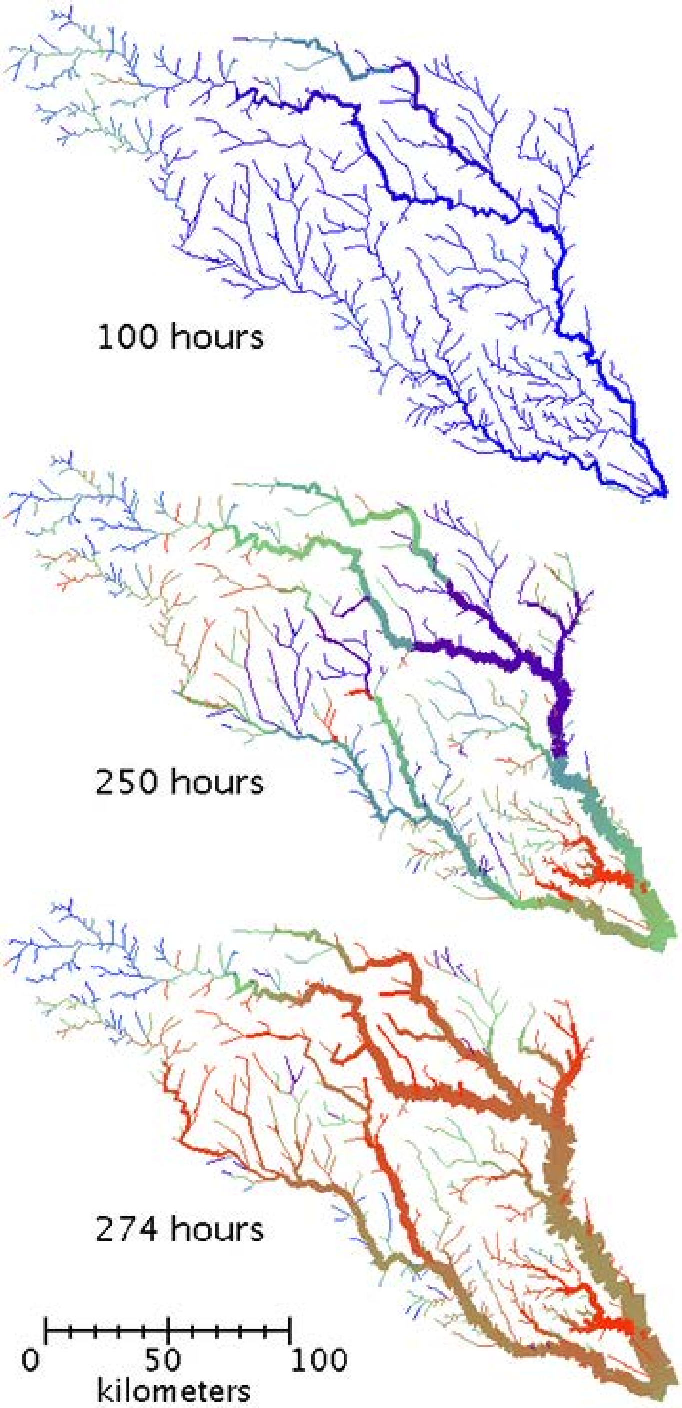
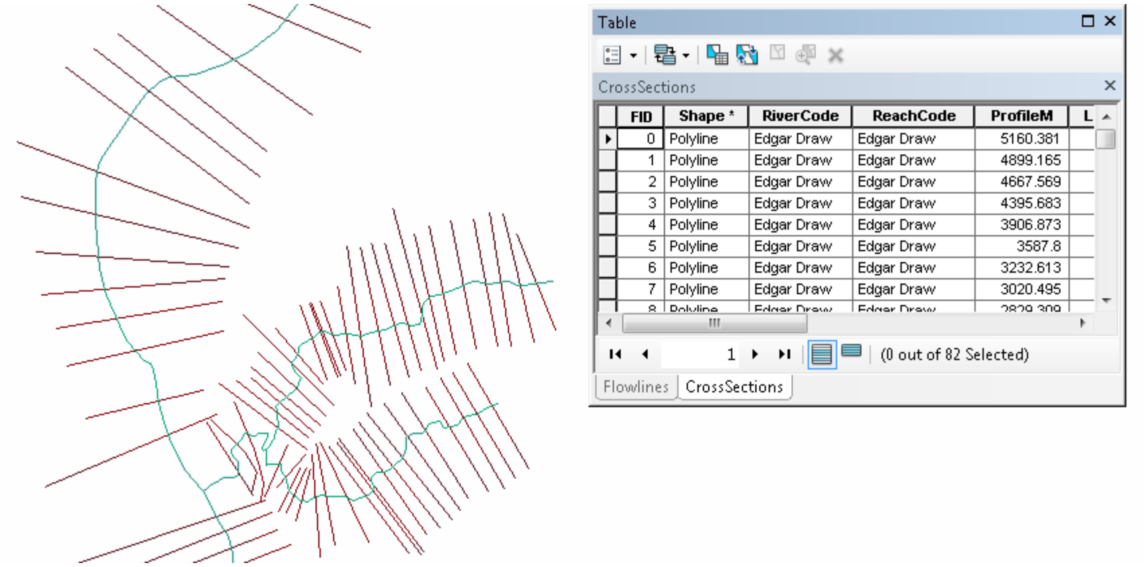


Figure 2.12 Results of SPRNT simulation (Hodges, 2014)

2.4 DATA MODELS

2.4.1 ESRI Shapefile

The shapefile format was created by ESRI for conveying vector geometry in GIS applications. The standard is published so that third party developers can read and write shapefiles. It supports point, line, and polygon geometry and includes a database component which can associate attributes with the individual features (ESRI, 1998). Shapefiles provide a flexible and widely recognized format for conveying river geometry. However, flexibility comes at the cost of variety. Attributes and feature organization vary by agency, project, and user, and thus correct interpretation of shapefiles often requires accompanying documentation. For example, cross sections can be identified as such either in the file name (as in Figure 2.13) or a user-specified field. In the latter case, a single shapefile may contain a mix of cross sections, profile lines, and other features. In order for shapefiles to serve as automated model intermediaries without human interactions, specific metadata schemas must be defined and adhered to.



	FID	Shape *	RiverCode	ReachCode	ProfileM	L
▶	0	Polyline	Edgar Draw	Edgar Draw	5160.381	
	1	Polyline	Edgar Draw	Edgar Draw	4899.165	
	2	Polyline	Edgar Draw	Edgar Draw	4667.569	
	3	Polyline	Edgar Draw	Edgar Draw	4395.683	
	4	Polyline	Edgar Draw	Edgar Draw	3906.873	
	5	Polyline	Edgar Draw	Edgar Draw	3587.8	
	6	Polyline	Edgar Draw	Edgar Draw	3232.613	
	7	Polyline	Edgar Draw	Edgar Draw	3020.495	

Figure 2.13 Cross section and flowline shapefiles

2.4.2 Geographic Markup Language

Geographic Markup Language (GML) is an XML encoding for spatial and temporal features developed by the OGC. It includes geometric primitives such as points, lines, arcs, splines, and surfaces, as well as a mechanism for combining these primitives to form complex features. GML is a general standard, intended to be used in the creation of domain-specific application schemas. GML allows for application schemas to be created either directly as an XML schema, or in UML with a mapping to GML based on a set of encoding rules. UML provides a visual representation of the information model, and supports mapping into various implementations including GML conformant XML (Portele, 2007).

GML is a complex standard designed to cover a wide variety of use cases. “With such a wide scope, interoperability can only be achieved by defining profiles[3] of GML that deal with a restricted subset of GML capabilities (van den Brink et al., 2012).” One

---

[3] In this context, ‘profile’ refers to a specialization of a general standard, as opposed to the river geometry context used elsewhere in this paper.

such profile is the GML Simple Features Profile, which is supported natively by ArcGIS (ESRI, 2014).

2.4.3 HY_Features

HY_Features is a general domain model for hydrologic features which is currently being developed by the Hydrology DWG. At present it is defined in a number of Discussion Papers, and is not yet an official standard of the OGC. HY_Features draws its terminology from the UNESCO/WMO International Glossary of Hydrology, and enhances this glossary by defining relationships between a subset of the terms (Dornblut and Atkinson, 2013).

> The HY_Features model is intended to sufficiently describe hydrologic features referenced in the various data sets in current use and to form a basis for a common and stable referencing of these features to assist the organization of their observation and modeling as well as the aggregation of generated data into integrated suites of datasets on global, regional, or basin scale. (Dornblut and Atkinson, 2013)

Model conformance with HY_Features is defined as the ability to map feature types and feature properties to their equivalent concept (Dornblut and Atkinson, 2013). As RiverML evolves, further work will be required to ensure that RiverML conforms to the HY_Features model. If areas of non-conformance are identified, revisions to either RiverML or HY_Features may be appropriate.

2.4.4 Arc Hydro

Arc Hydro is a data model customizing ArcGIS for water resources. By defining classes and relationships between common hydrologic features in a geodatabase, Arc Hydro can serve as the backbone to an HIS. It can be extended as needed to fit particular use cases, and linked to computational models for simulation. The core Arc Hydro model consists of five components which can be used independently or linked together to form a comprehensive model. These five components are: Network, Drainage, Channel, Hydrograph, and Time series (Maidment, 2002). Three additional components for temporal series have been added to the original Arc Hydro model: Attribute series, Feature series, and Raster series (Arctur and Zeiler, 2004).

2.4.5 Arc River

Arc River is a data model currently under development intended to extend the capabilities of Arc Hydro to support advanced data collection and modeling techniques (Kim, 2008).

> Arc River data model is designed to: (i) represent river data in a curvilinear coordinate system to support river channel oriented spatial analyses; (ii) represent multidimensional river features through points, lines, polygons, and volumes; (iii) represent simulated gridded data for river channels that can be efficiently coupled with observed data; (iv) represent spatio-temporal dynamics of moving river objects (such as bedform) from single or multiple events using Eulerian or Lagrangian observational frameworks, and (v) store tabular and metadata information for river measurements. (Kim et al., 2014)

In addition to data derived from traditional survey methods, Arc River is capable of representing data generated by Acoustic Doppler Current Profilers (ADCP), which can be mounted on a boat to measure three-dimensional hydrodynamic data along any path of the river (Kim, 2008). Arc River supports vector as well as scalar information, and can attach metadata to observations (Figure 2.14).

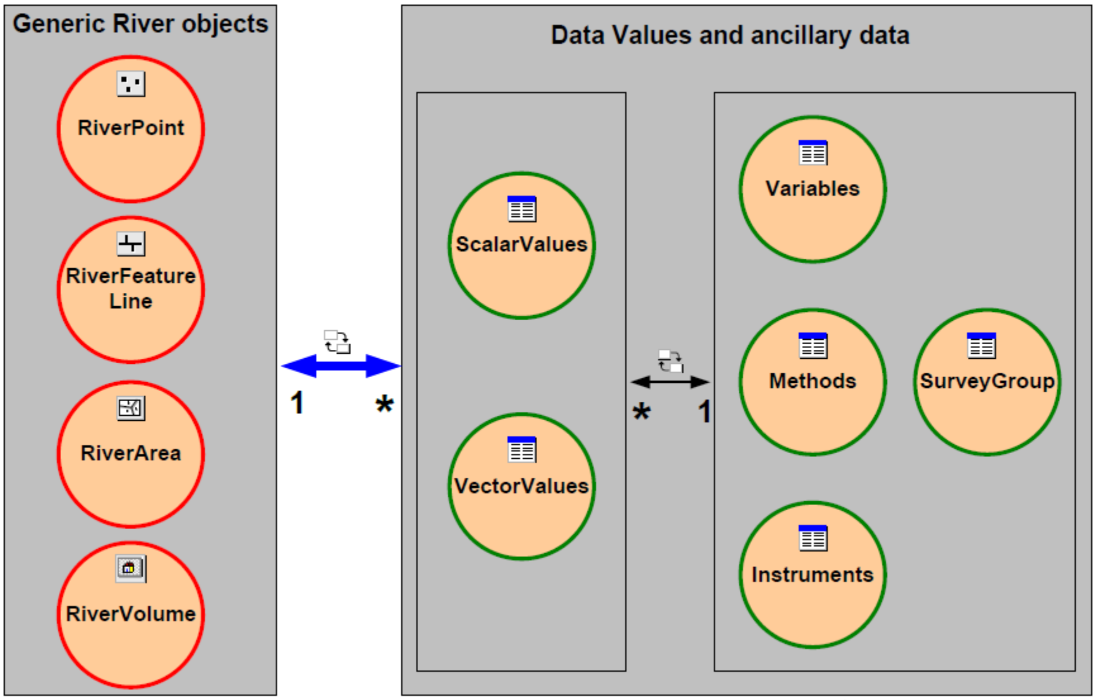


Figure 2.14 Arc River data values and metadata (Kim et al., 2014)

2.4.6 Curvilinear River Channels

One method of transforming river geometry into the curvilinear reference system supported by Arc River was developed by Merwade (2004). The channel boundary and an arbitrary centerline are used to create a coordinate system where locations are identified in the *s*, *n*, *z* coordinate system, where *s* represents the distance along the centerline, *n* represents the distance perpendicular to the centerline, and *z* represents the elevation (Figure 2.15). Bathymetry measurements can be transformed from Cartesian x, y, z coordinates into the curvilinear coordinate system and used to create a DEM and a 3D FishNet mesh. The FishNet can be transformed back into Cartesian coordinates, resulting in a wireframe mesh with the useful properties that all lines are either parallel or perpendicular to the direction of 1D flow. FishNet can be used to create 2D or 3D model curvilinear geometry (Hodges and Imberger, 2001) or create 1D cross sections (Merwade et al., 2005). The FishNet has the added benefit of rendering more quickly than a TIN in 3D visualizations (Merwade et al., 2005).

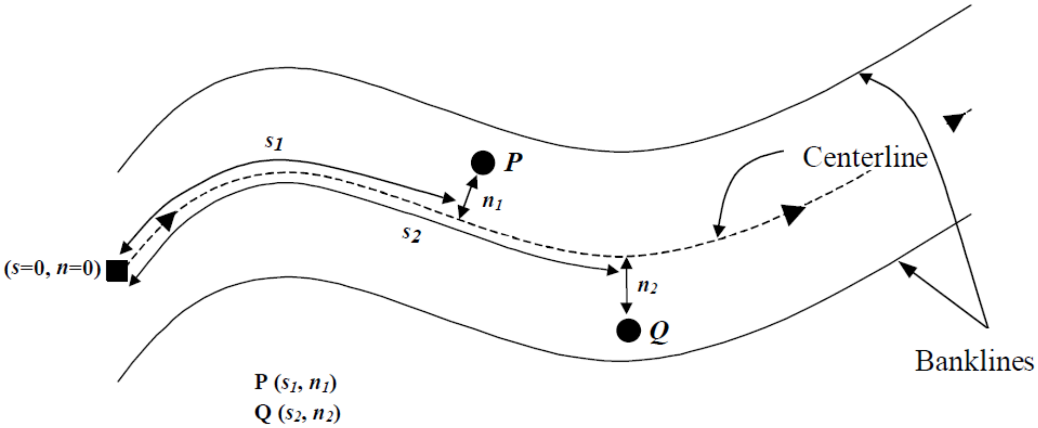


Figure 2.15 Curvilinear coordinate system (Merwade, 2004)

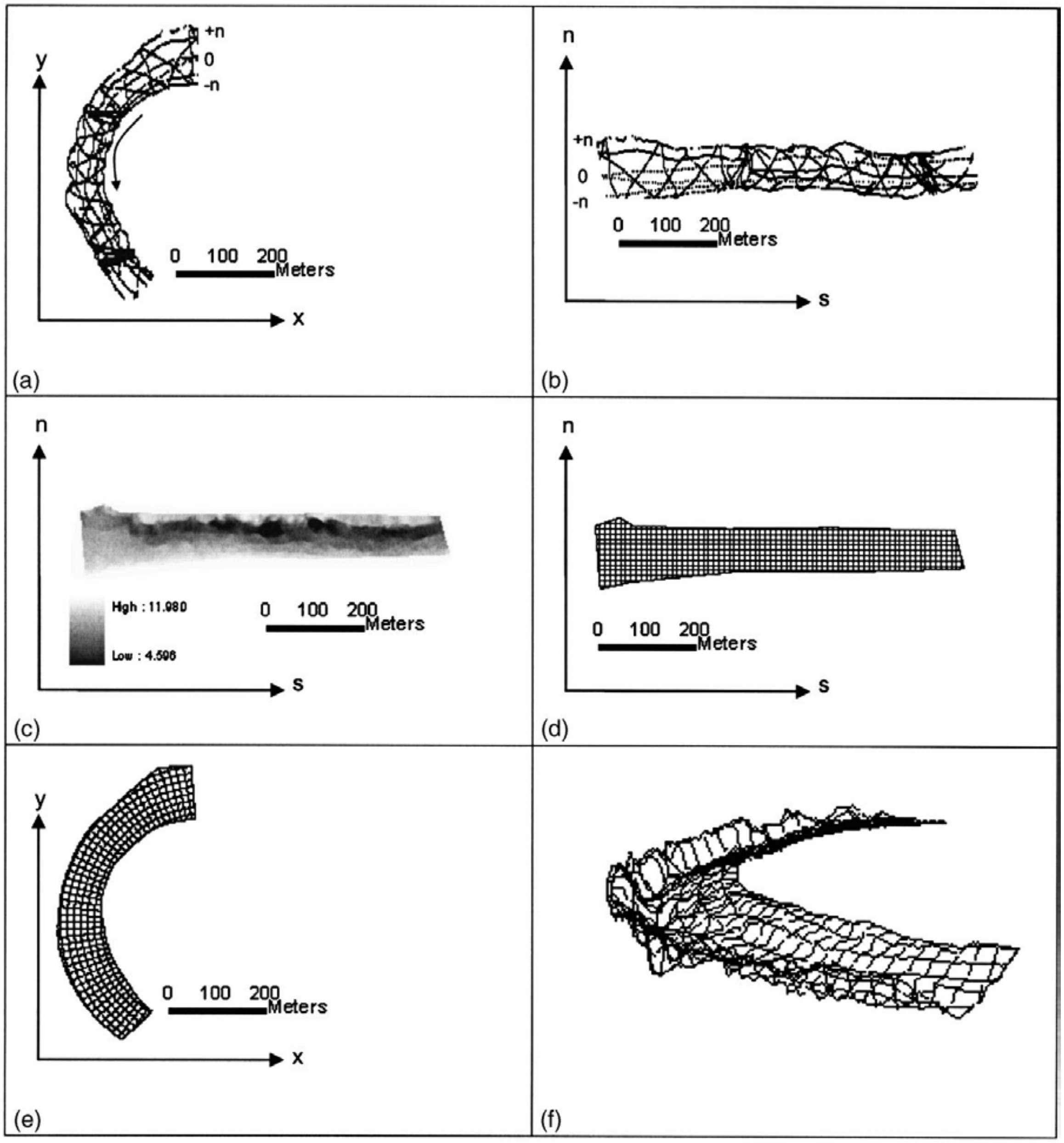


Figure 2.16 Procedure for creating FishNet: (a) bathymetry points in (x, y , z); (b) bathymetry points in (s ,n, z); (c) raster surface for bathymetry points in s ,n, z-coordinates; (d) FishNet in s ,n, z-coordinates; (e) FishNet transferred from s ,n, z-coordinates to x, y , z-coordinates; (f) FishNet in three dimensions (Merwade et al., 2005)

2.4.7 Spatiotemporal Data Model

The Spatiotemporal Data Model (SDM) was developed to promote a tighter coupling between GIS and hydrology models (Goodall and Maidment, 2009). Rather than attempting to adapt existing GIS frameworks to support hydrologic concepts, SDM defines a logical framework and challenges developers to create a new generation of software applications capable of utilizing it. The framework consists of three elements: control volumes, fluxes, and flux couplers. These elements are designed to explicitly support computations based on conservation of mass, energy, and momentum (Goodall and Maidment, 2009).

From a computational perspective, SDM represents an elegant approach to storing data useful for both hydrologic and hydraulic simulations. However, explicit treatments of the data requirements for 1D hydraulic flow have yet to be developed so it is unclear how elements such as cross sections would be represented. It is therefore out of scope for this present harmonization effort to include SDM.

2.4.8 Observations and Measurements

Observations and Measurements (O&M) is a standard originally developed by the OGC and adopted in a revised form by both the OGC and the International Organization for Standardization (ISO). O&M is a conceptual model defined in UML which supports various implementations, and is intended to be restricted and extended as needed to create domain-specific schemas (Technical Committee ISO/TC 211, 2010).

> An observation is an act associated with a discrete time instant or period through which a number, term or other symbol is assigned to a phenomenon. It involves application of a specified procedure, such as a sensor, instrument, algorithm or process chain. The procedure may be applied in-situ, remotely, or ex-situ with respect to the sampling location. The result of an observation is an estimate of the value of a property of some feature. Use of a common model allows observation

> data using different procedures to be combined unambiguously. … **The key idea is that the observation result is an estimate of the value of some property of the feature of interest, and the other observation properties provide context or metadata to support evaluation, interpretation and use of the result.** (Technical Committee ISO/TC 211 2010; emphasis added)

O&M supports simple result data types, such as integers and strings, as well as complex data types such as geometry (Technical Committee ISO/TC 211, 2010).

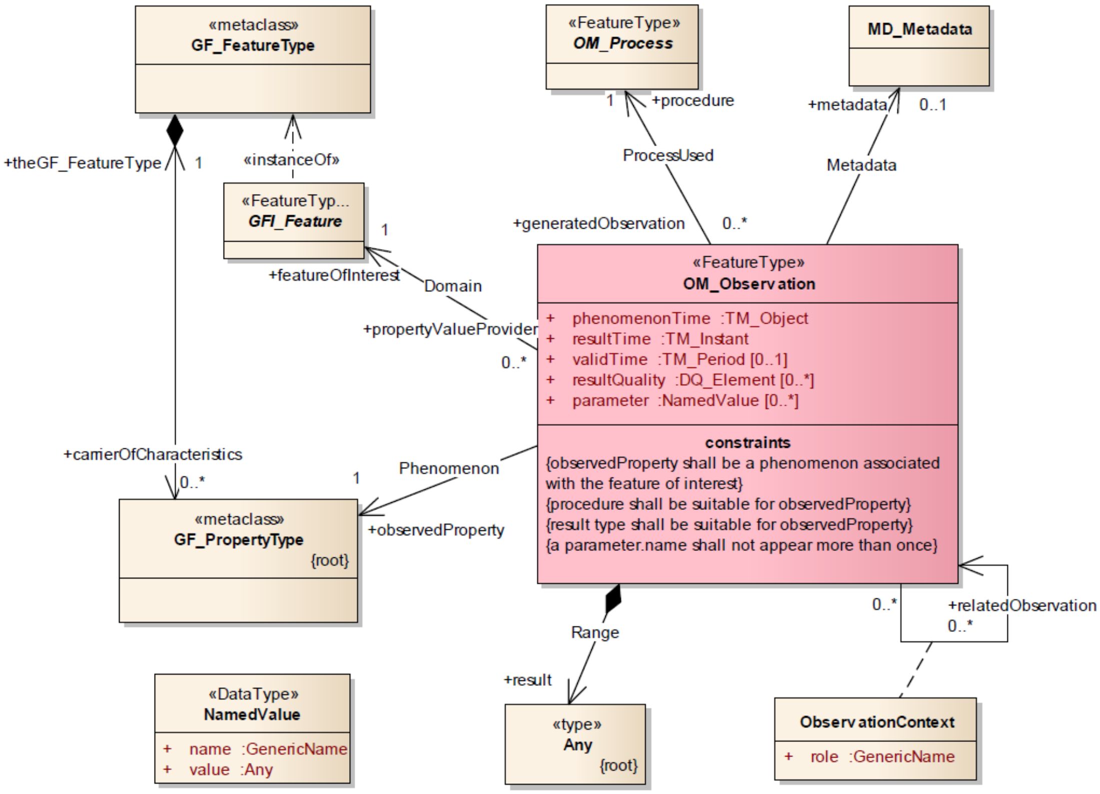


Figure 2.17 Observation as defined by O&M (Taylor, 2012)

2.4.9 WaterML 2.0

WaterML 2.0 is an open standard developed by the OGC for communicating water observations data. It builds on O&M as well as GML to define a time series observation associated with a geospatial location. WaterML 2.0 includes both data and metadata, such as provenance, accuracy, and interpolation methods (Taylor, 2012).

The general characteristics of WaterML2.0:

1. Communicates the semantics of hydrological time series data;
2. An explicit time series model that supports encoding of information crucial to correct interpretation of time series, such as properties describing the nature of individual data values and their relationships;
3. A flexible transfer schema that can be re-used in a number of scenarios. Includes concepts to deal with common complexities in cross-system data exchange, such as multiple identifiers and names;
4. The schema is reusable across different transport technologies, including FTP, and a variety of web services etc.;
5. Ability to extend through use of external schema and soft-typing;
6. Ability to capture information relating to the provenance of a time series (i.e. how the time series was created). Allows for interpretation of ‘data products’ such as statistical summaries; (Taylor, 2012)

WaterML 2.0 Part 2 is currently under development, and will include the ability to communicate ratings and gaugings information. Ratings and gaugings are used to describe the relation between two dependent variables such as river stage and discharge (Taylor, 2013).

2.4.10 HEC-GeoRAS

HEC-GeoRAS uses a pair of custom formats which can be expressed either in XML or ASCII. The Import File conveys pre-processed geometry information from ArcGIS to HEC-RAS, and the Export File conveys geometry and flow results from HEC-RAS back to ArcGIS for post-processing (Ackerman, 2009).

> [The RAS GIS Import File contains] river, reach and station identifiers; cross-sectional cut lines; cross-sectional surface lines; cross-sectional bank stations; downstream reach lengths for the left overbank, main channel, and right overbank; and cross-sectional roughness coefficients. Additional geometric data defining levee alignments, ineffective flow areas, blocked obstructions, and storage areas may be written to the RAS GIS Import File. GeoRAS Version 4 introduced capabilities for exporting hydraulic structure data for bridges, inline structures, and lateral structures. (Ackerman, 2009)

2.4.11 MIKE ASCII

MIKE 11 uses a custom ASCII file format for importing geometry data. Cross section data can either be raw or processed. Raw data specifies the geometry of the cross section as a station-elevation table, while processed data uses the derived quantities of elevation-area-hydraulic radius (DHI, 2012 c).

2.4.12 SPRNT Netlist

The input format for SPRNT is a custom ASCII file called a netlist. Geometry, network connectivity, and time series are all expressed as blocks of text containing keyword/value pairs. Cross section geometry, including slope and roughness coefficients, is specified at computational nodes. Network connectivity is established by defining segments which are pairs of upstream/downstream nodes in upstream/downstream with a specified flow distance. Discharge and boundary condition information is entered as time series linked to specific nodes (Liu, 2014).

2.4.13 LandXML

LandXML is an XML transfer schema for civil engineering and survey data. It is a non-proprietary format developed by an international consortium of industry partners. The focus is data used for land development and transportation, such as roadways and parcels. While it doesn’t have an explicit channel representation, LandXML Version 1.2 includes surfaces and pipe networks. In April of 2014, it was announced that LandXML 2.0 is currently under development and that some level of engagement with the OGC was intended (LandXML.org, 2014).

2.5 MODEL COUPLING

An HIS can contain both data models and simulation models. To support automated simulation, a mechanism for coupling these models is required. This coupling can range on a spectrum from tight, where the models communicate directly, to loose, where the models share data through a bridge. Tight couplings provide simplified user experience and greater assurance of model fitness-for-use at the cost of increased development time and decreased flexibility. Loose couplings can be developed more rapidly and allow simulation models to be chained in a flexible manner, but require greater expertise on the side of the user (Charnock et al., 1996; Whiteaker, 2004). This research is concerned with loose coupling cases where a bridge is required, either between a data provider and data user or by a data user between data models and simulation models.

2.5.1 Information Exchange Points

One method for achieving the coupling is to use a central GIS database linked to simulation models through bi-directional Interface Data Models. Results from simulations are returned to the geodatabase, where they become available for use in subsequent simulations. Interface Data Models are custom-built bridges between specific data models and simulation models (see Figure 2.18). The simulation models are run independently, with information exchange taking place only after a simulation is complete. This exchange happens at user-defined Information Exchange Points, which are locations such as cross sections or the outlets to watersheds where results such as time series can logically be shared (Whiteaker, 2004).

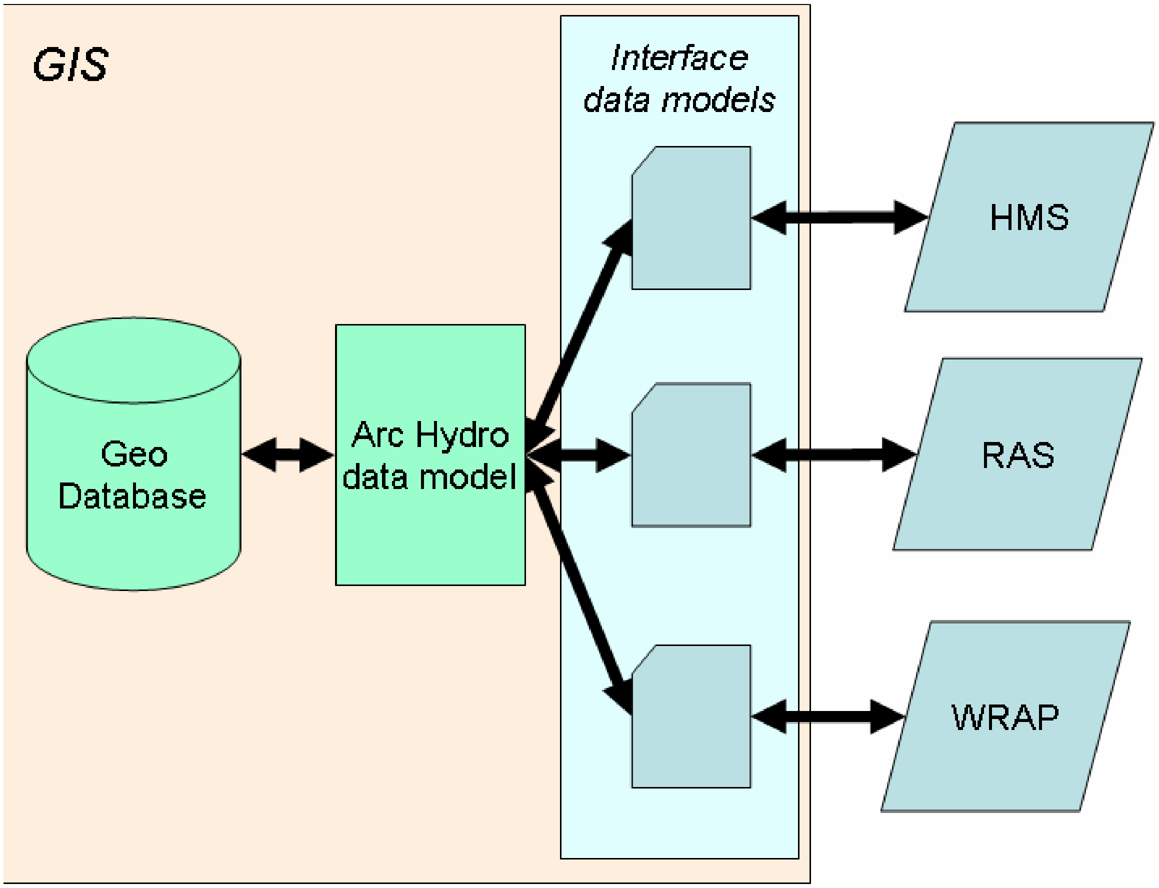


Figure 2.18 Multiple Interface Data Models for HEC-HMS, HEC-RAS, and WRAP (Whiteaker, 2004)

2.5.2 OpenMI

The Open Modelling Interface (OpenMI) is a standard data exchange interface that serves as a bridge between simulation models[4] which has been adopted by the OGC (Vanecek and Moore, 2014). OpenMI differs from the Interface Data Models described in Section 2.5.1 in that OpenMI allows data exchange to occur during execution of a simulation (Moore, 2010). On the spectrum of model coupling, OpenMI represents a tighter integration between simulation models. This method does not require a central

---

[4] The OpenMI literature uses slightly different terminology. What is here referred to as a *simulation model* is termed an *engine* or a *linkable component*, and the term *model* is reserved for a specific instance of an engine with input data.

geodatabase serving as intermediary between simulation models. The present recommended method for integrating multiple simulation models using OpenMI is the ‘Pull driven’ approach:

> A component is defined as the primary driver of the composition; update is repeatedly called on this primary component until it reaches completion. On each update call, before the component performs its computation, it updates all its active targets until they can provide the required input data. If necessary, these targets will in turn call update on their connected source components and so on. This update and compute mechanism then propagates data around all the components in the composition in a sequential manner. (Moore, 2010)

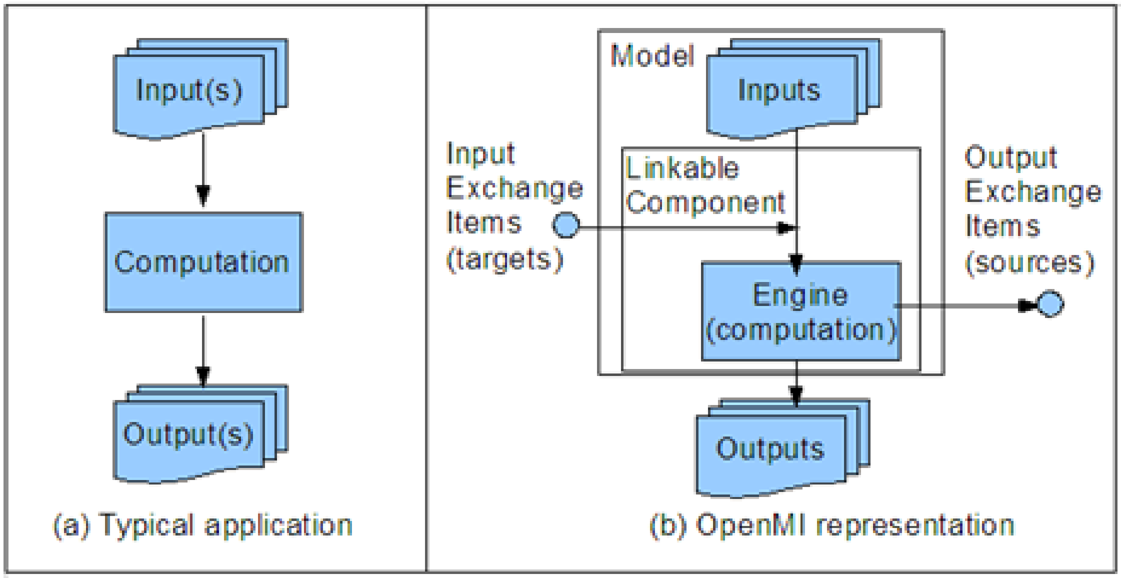


Figure 2.19 OpenMI Model Representation (Moore, 2010)

2.6 AUTHORITATIVE MODEL IMPLEMENTATIONS

2.6.1 NHDPlus

NHDPlus Version 2 (NHDPlus) is the national hydrologic platform for the United States developed by Horizon Systems Corporation in partnership with the US Environmental Protection Agency (EPA) and the US Geological Survey (USGS). It includes a stream network, catchment boundaries, various value-added attributes which aid in network analysis, as well as the source datasets from which the NHDPlus was derived. NHDPlus is distributed in a compressed folder structure containing shapefiles, grids, and database tables. The data model and data content are both versioned and updated periodically, and the data content of individual drainage basins can be updated independently (McKay et al., 2014).

2.6.2 Geofabric

The Geofabric is the national hydrologic platform for Austrailia developed by the Australian Bureau of Meteorology, containing relationships between feature classes, tables, and raster datasets. It consists of six integrated data products: Surface Cartography, Surface Network, Surface Catchments, Groundwater Cartography, Hydrology Reporting Catchments, and Hydrology Reporting Regions. The Geofabric is based on a modular conceptual model which supports multiple implementations (Bureau of Meteorology, 2012).

> The design and development of the Geofabric recognises the inherent problems of function, scale and accuracy in representing spatial data. The Geofabric product suite attempts to distinguish between the functional requirements of both topological and geometric representations of hydrological features. Topologically consistent spatial features are those that show connectivity, e.g. consistent direction, node-link connectivity, schematic networks and feature relationships. Geometric spatial features are those that are represented by points, lines or polygons (in ESRI's ArcGIS environment) and are commonly described as the blue lines for streams and the associated water features (e.g. cartographic representations). (Bureau of Meteorology, 2012)

2.6.3 Hilltop Software

At present, hydrography data in New Zealand is managed at a regional rather than national level. Agencies such as Horizons Regional Council utilize the DataTamer application developed by Hilltop Software (Hilltop) for data management (Jeff Watson, Manager Catchment Data, Horizons Regional Council, pers. comm.). Hilltop consists of several modules including a database manager, a ratings and gaugings module, and a module for surveyed cross sections. It supports import and export in a variety of formats including Excel and MIKE11 (Hilltop Software, 2014).

Chapter 3: Harmonizing Core Concepts

3.1 HARMONIZATION OVERVIEW

Three key types of information required for hydraulic modeling were identified in Chapter 2. The first is the data and metadata for each feature (i.e. cross sections, river centerlines). The second is the relationships between features such that simulation models can route flow through the system (i.e. establish upstream/downstream relationships). The third is to describe changes in features over time, such as natural changes in river geometry or the evaluation of proposed modifications. These latter two tasks were not faced by the WaterML 2.0 harmonization effort, which treated a system of observing stations as independent Monitoring Points.

In order to create a standard transfer language for communicating hydraulic information between the technologies listed in Chapter 2, a harmonized approach to describing these three types of information must be developed. The three tasks will be examined in Sections 3.2, 3.3, and 3.4, respectively. Principles used in the harmonization of time series observations for WaterML 2.0 which are relevant to the present work are identified in Section 3.5. The recommendations for developing a standard transfer language which are described throughout this chapter are summarized in Chapter 6.

3.2 RIVER FEATURES

We can identify the following common features of river data models:

- Surfaces (i.e. DEM, DTM)
- Cross Sections lines (perpendicular to direction of flow)
- Profile lines[5] (parallel to direction of flow)

---

[5] The initial RiverML 0.1 prototype used the term *flowline*. This paper uses *profile line* to conform with Arc Hydro.

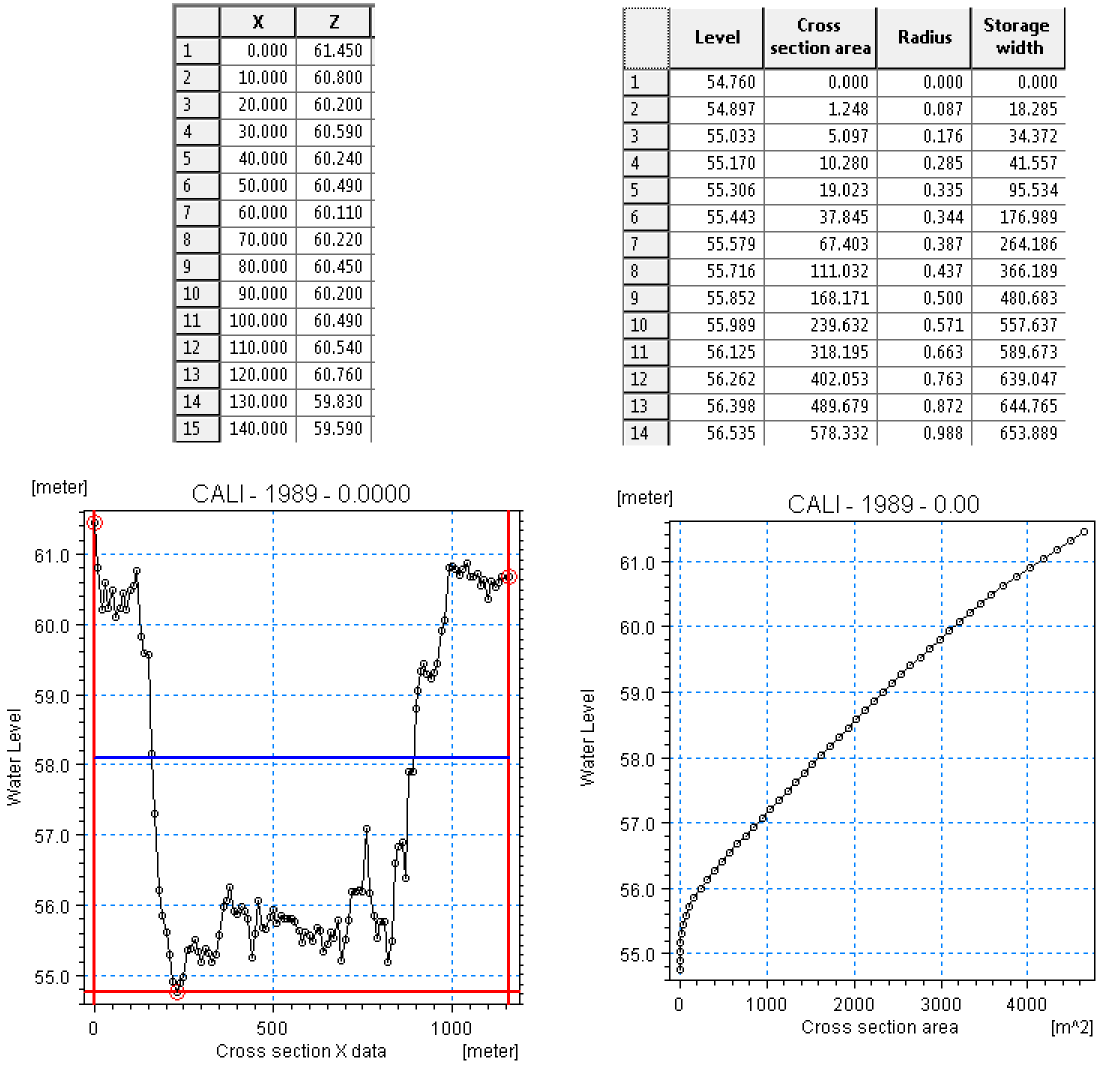
- Storage areas (i.e. reservoirs, lakes)
- Linear attributes (i.e. roughness coefficients, thalweg location)
- Time series (i.e. water surface elevation, flow rate)
- Structures (i.e. culverts, bridges, culverts, pumps)
- Catchments (i.e. drainage area, land use, soil type)

DEMs and time series have established standardized formats, and therefore will not be considered in this chapter. Proposed methods for incorporating these existing standards into RiverML are discussed in Chapter 4. While structures are necessary for the complete representation of rivers, harmonizing the various encodings are out of scope for this present work. It is recommended that structures be included in a later version of RiverML after the basic model for geometry has been tested, refined, and successfully implemented. Likewise, the description of catchments is currently out of scope, and can either be treated by a future extension of RiverML or by a separate but compatible standard. Therefore in this section the four features examined are cross sections, profile lines, storage areas, and linear attributes.

3.2.1 Cross Section Harmonization

3.2.1.1 Raw geometry vs. processed values

The equations solved by 1D hydraulic models typically involve parameters such as wetted area, wetted perimeter, and hydraulic radius (Brunner, 2010 a; DHI, 2012 a; Liu, 2014), which are a function of the geometry and the water surface elevation (depth). For a given cross section geometry, these values can be tabulated by depth (see Figure 3.1). MIKE11 supports importing the processed tabular values directly instead of importing the geometry, which reduces the number of computations required by the simulation model (DHI, 2012 c). However, as the processed values can be derived from geometry, and geometry inputs were supported by all simulation models investigated, it is recommended that RiverML not support the encoding of processed cross section parameters. This reduces the number of assumptions made about the end use of the data, thereby enhancing interoperability.



	X	Z
1	0.000	61.450
2	10.000	60.800
3	20.000	60.200
4	30.000	60.590
5	40.000	60.240
6	50.000	60.490
7	60.000	60.110
8	70.000	60.220
9	80.000	60.450
10	90.000	60.200
11	100.000	60.490
12	110.000	60.540
13	120.000	60.760
14	130.000	59.830
15	140.000	59.590

	Level	Cross section area	Radius	Storage width
1	54.760	0.000	0.000	0.000
2	54.897	1.248	0.087	18.285
3	55.033	5.097	0.176	34.372
4	55.170	10.280	0.285	41.557
5	55.306	19.023	0.335	95.534
6	55.443	37.845	0.344	176.989
7	55.579	67.403	0.387	264.186
8	55.716	111.032	0.437	366.189
9	55.852	168.171	0.500	480.683
10	55.989	239.632	0.571	557.637
11	56.125	318.195	0.663	589.673
12	56.262	402.053	0.763	639.047
13	56.398	489.679	0.872	644.765
14	56.535	578.332	0.988	653.889

(a) (b)

Figure 3.1Cross section input values: (a) raw cross section geometry (b) processed cross section parameters

3.2.1.2 Coordinate systems

There are numerous approaches to defining the coordinates for cross section geometry, including georeferenced, non-georeferenced, curvilinear, and station-elevation. For all approaches, elevations may be defined as having a constant offset to a specified or implied datum.

In georeferenced cross sections, vertices are given as x, y, z coordinates in an explicit geographic coordinate system. These sections can be directly mapped in GIS applications in conjunction with maps and other data products. The HEC-GeoRAS Import and Export Files, as well as Arc Hydro and Arc River, use georeferenced cross sections (Maidment, 2002; Ackerman, 2009; Kim et al., 2014).

Non-georeferenced sections are given as x, y, z coordinates but without an explicit geographic coordinate system. As GIS approaches information sharing becomes more integral to workflows within water resources engineering, non-georeferenced features become less common. Software applications such as ICPR which originally operated in generic Cartesian space have been upgraded to support GIS (Streamline Technologies, 2014 a).

Curvilinear coordinate systems use a reference profile line such as a river centerline and express vertices as s, n, z coordinates, where s is measured along the profile line from a define starting point, n is measured perpendicular to the profile line, and z is the elevation above some datum. In the FishNet methodology described in Section 2.4.6, this is the coordinate system used before the final transformation back into x, y, z coordinates. The curvilinear coordinate system can be seen as a special case of the non-georeferenced system.

Station-elevation sections express vertices as *m*, *z* coordinates, where *m* is measured along the cross section and *z* is the elevation above some datum[6]. In order to be rendered in plan view, a table of station-elevation values must be associated with a defined or assumed *x*, *y* section line. In the absence of a defined section line, the cross section is assumed to be a single line segment oriented perpendicular to the direction of flow. The SPRNT Netlist and ICPR use station-elevation tables, and the MIKE11 ASCII format uses station-elevation tables with an optional 2D section line (see discussion in Section 3.2.1.3) (DHI, 2012 c; Liu, 2014; Streamline Technologies, 2014 a). When entering data manually into HEC-RAS as opposed to using HEC-GeoRAS, station-elevation tables with an optional georeferenced 2D section line are used (Brunner, 2010 b). Arc Hydro supports station-elevation tables using the CrossSectionPoint class (Maidment, 2002). HEC-RAS and MIKE11 each support a defining a skew or correction angle between the direction of flow and the orientation of the section. If such an angle is included in RiverML, the angle frame of reference should be clearly defined to avoid ambiguity. In addition to station-elevation tables, SPRNT allows rectangular or trapezoidal cross sections to be defined. However, as these values can be readily transformed to and from station-elevation data where applicable, it is not necessary for an transfer standard to explicitly handle these regular shapes.

Georeferenced cross sections and profile lines are the most versatile, as the other formats can be derived from them[7]. It is recommended that wherever possible, georeferenced geometry be used for data exchange. However, there is substantial legacy

---

[6] The coordinates in this approach may also be termed *x*, *z*, or *x*, *y*.

[7] In a curvilinear reference system, a smoothed reference line is required in order to properly assign an *n* coordinate, and problems may be encountered in braided or highly sinuous streams. See discussion in Merwade (2004) and Merwade et. al. (2005).

information obtained from nearly a century of field surveys in the form of station-elevation tables (Maidment, 2002). It may therefore be beneficial for RiverML to be able to support this format. If supported, station-elevation tables should have the optional ability to be associated with georeferenced cross sections both for visualization purposes and to support migrating data to GIS environments. Furthermore, if the explicit description of the geographic coordinate system is left as optional within the standard, RiverML would support both non-georeferenced and curvilinear measurements. Additional support for alternate coordinate systems, such as metadata for describing the curvilinear system, can be included if compelling use cases are put forth by the community.

3.2.1.3 Two-dimensional vs. three-dimensional

Georeferenced cross sections can be 2D (x, y) or 3D (x, y, z). 2D sections are not used for computation, but rather for visualization and feature extraction (see Figure 3.2). The HEC-GeoRAS Import File encodes both the 2D line (cut line) and 3D line (surface line) as part of a cross section definition. The Export File encodes only the 2D line and specifies water surface elevations, which can be used by post-processing GIS tools to interpolate the floodplain extents between cross sections (Ackerman, 2009). Arc River contains classes for a 2D section line[8] and a cross section area (which is a 3D section line closed to form a polygon). Each of these feature classes support child point features for detailed measurements (Kim et al., 2014).

From a geometrical perspective, 2D cross sections can be derived by simplifying 3D sections. However, from a workflow perspective, 2D sections are typically defined

---

[8] The Arc River documentation assigns special meaning to the terms *1D*, *2D*, and *3D*, which is not used in this paper.

first. The modeler first draws sections in plan view at locations of interest, which may be single line segments or multiple segments to account for irregular terrain. The 2D section only requires vertices at the end points of each segment. These sections can then be ‘draped’ over a DEM using GIS tools to obtain a 3D section, with vertices added at each vertical slope change. Multiple DEMs can be exchanged to obtain different 3D sections with the same 2D geometry. The 2D section can also be associated with tabular station-elevation data as discussed in Section 3.2.1.2. As 2D and 3D sections serve distinct purposes within hydraulic analysis workflows, it is valuable for a transfer standard to be able to communicate both types for the same location (Dean Djokic, ESRI Professional Services, pers. comm.).

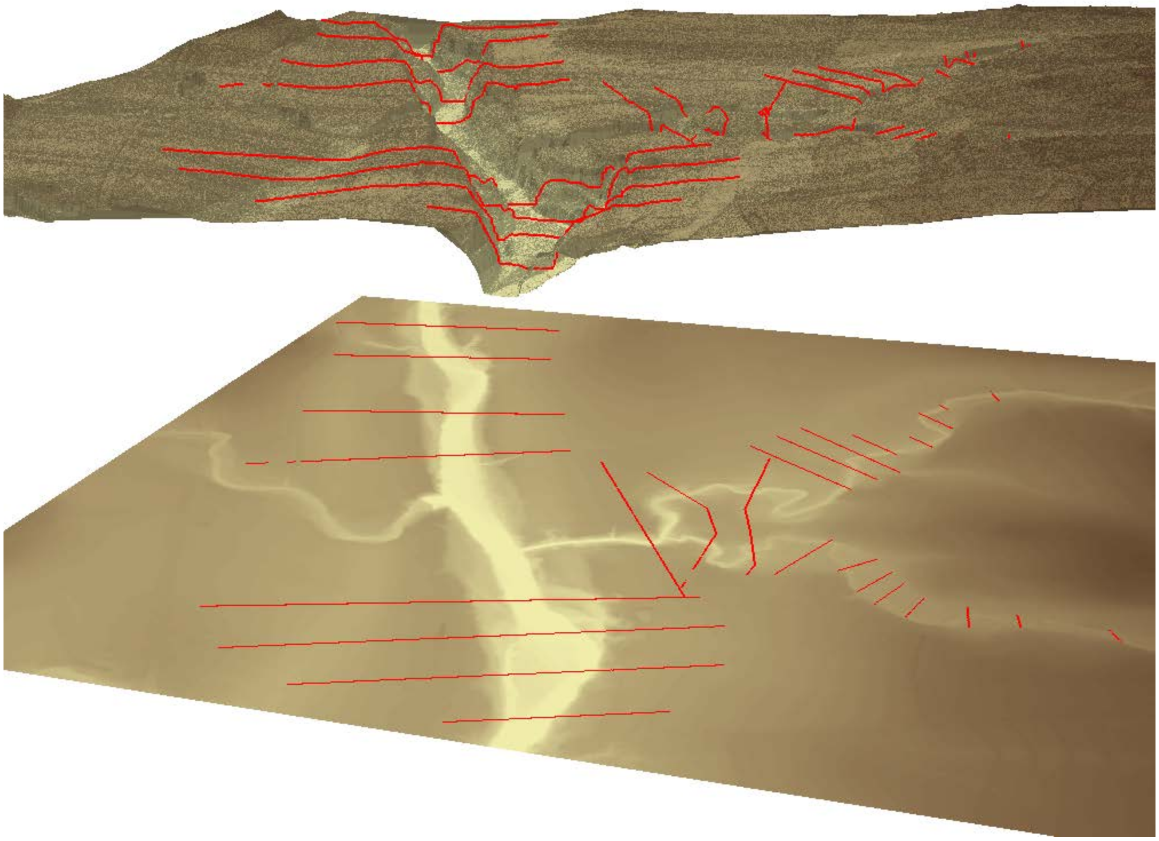


Figure 3.2 Two-dimensional cross section cut lines used to extract three-dimensional cross sections from a digital elevation model

3.2.1.4 *Orientation*

Cross sections, whether georeferenced or not, are typically assumed to be in a consistent orientation: either left-to-right when looking downstream or the opposite. This is important for cases where the simulation divides flow up into discrete regions of the cross section (such as left overbank, channel, and right overbank), in order to ensure that flow quantities are transferred to the appropriate region of the upstream and downstream cross sections. It also aids in locating structures and obstructions, and provides the user with a consistent visual representation. In most existing formats, the inputs are assumed to follow a convention, and no explicit indication of the orientation is provided. It is recommended that RiverML include an orientation metadata property which can either be applied as a default to an entire dataset or specified on a section-by-section basis.

3.2.1.5 Additional considerations

HEC-RAS allows the inclusion of blocked obstructions and ineffective flow areas. These are defined by specifying a set of station-elevation points. The blocked or ineffective region extends vertically from the ground elevation up to the specified elevation, and horizontally between the specified stations. The linear attribute techniques described in Section 3.2.4 could be used by a standard transfer format to communicate the dimensions and description of regions such as these.

MIKE11 supports closed cross sections, where the geometry is a polygon rather than a line (DHI, 2012 c). A simple method of allowing this in a transfer standard would be to have a Boolean property indicating that the first and last vertices should be connected, which defaults to false.

SPRNT uses a smoothed river bottom reference line, and requires that the slope of this line and distance between the reference line and true bottom be specified for each cross section (see Figure 3.3) (Liu, 2014). It is likely that this special use case can be satisfied by a pre-processing algorithm using standard cross section and profile line inputs.

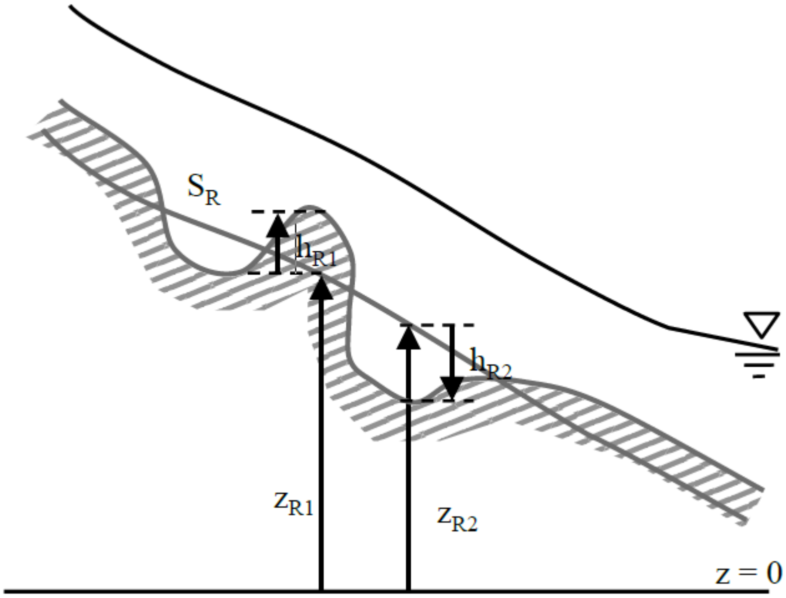


Figure 3.3 SPRNT reference line (Liu, 2014)

3.2.2 Profile Line Harmonization

The discussion in Section 3.2.1.2 and Section 3.2.1.3 concerning coordinate systems and dimensionality generally apply to profile lines as well, with the exception that station-elevation tables are not typically used. While there isn’t a standard list of marker types for profile lines, MIKE11 does support user-defined markers (DHI, 2012 c). For orientation, the important property is whether the vertices are listed in the direction of flow or opposite the direction of flow.

3.2.2.1 Types of Profile Lines

There are many types of profile lines which carry specific information useful to modeling. A few common types are discussed here.

A thalweg line follows the lowest elevation in a channel along the direction of flow. This can be created by connecting the thalweg of each cross section, or through feature extraction tools using a DEM. For the latter approach, the length and sinuosity of the thalweg line is related to the resolution of the DEM. High resolution surfaces derived from LiDAR can have excessively sinuous thalwegs.

A centerline has a more flexible definition. It may be the thalweg, or a smoothed version of the thalweg, or an approximation of the path of the center of mass of the river flow (Brunner, 2010 a). The centerline is typically displayed as the ‘blue line’ representing the general path of the river.

Bank lines follow the slope break which distinguishes the main channel from the surrounding floodplain. In some cases this may be clearly defined, while in others the transition may be gradual and require judgment on the part of the modeler. For large rivers, the area between the bank lines is often displayed as a blue polygon, rather than using the centerline (Maidment, 2002).

Where the surrounding floodplain drains away from the river rather than toward it, levee lines can be used to mark the corresponding slope break.

Flow paths can be defined for the channel and the floodplain on either side. These represent the path of the center of mass of flow for each region, and are used by HEC-RAS to calculate flow distances between cross sections (Brunner, 2010 a). For the channel, this may be the same as the centerline. For the floodplains, this requires judgment on the part of the modeler.

Profile lines obtained from a curvilinear FishNet transformed back into x, y, z coordinates have the special property of being parallel to each other, and perpendicular to the FishNet cross sections (see discussion in Section 2.4.6). These lines create a wireframe model of the river which can be used for visualization and to accurately interpolate additional cross sections, as well as form a mesh grid which can be used for finite element analysis. For this paper these will be termed curvilinear mesh profile lines.

Contour lines are lines of constant elevation, which can be derived from surveys or extracted from DEMs. Contour lines can either be expressed as 3D lines or as 2D lines with an elevation attribute. They are not strictly profile lines, as they may wander in any orientation to the direction of flow and may even form isolated polygons. However, contour lines provide a method for creating a wireframe model similar to a FishNet without the complications of a coordinate transformation.

In general, profile lines require two properties in order for their purpose to be communicated: type and location. The type should draw from a standard code list where possible but allow for user-defined terms for specific situations such as the SPRNT reference line. The location refers to whether it is considered center, left of center, or right of center when viewed along the direction of flow. Including the ability to soft type additional properties would be beneficial. The author recommends that cross sections also be able to carry a type property. This would allow sections belonging to a curvilinear mesh to be identified, and support future extensions.

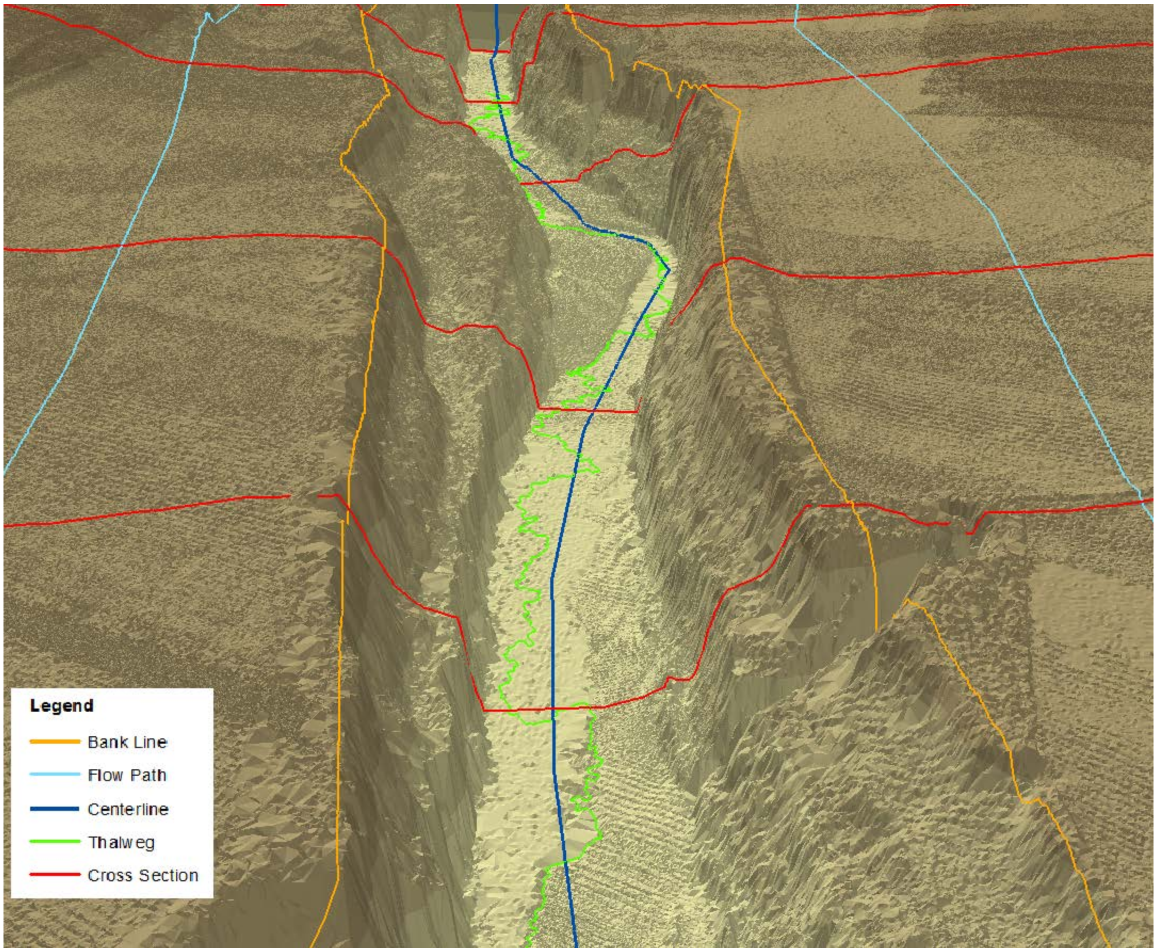


Figure 3.4 Profile line example.

3.2.3 Storage Area Harmonization

3.2.3.1 Raw geometry vs. processed values

As with cross sections, storage areas can be described either using geometry or as processed values. For storage areas, processed values take the form of elevation-area or elevation-volume tables.

3.2.3.2 In-line vs. off-line

HEC-RAS allows storage areas to be defined using elevation-volume tables, and either function in-line as a boundary condition for a reach or off-line. Off-line storage areas are connected to the river using lateral structures, which can be weirs, gates, culverts, or rating curves. Connections can be made between storage areas using a weir (Brunner, 2010 b). HEC-GeoRAS communicates storage areas using an outer 2D polygon and an elevation-volume table calculated based on the DEM. There is also the ability to include an array of *x*, *y*, *z* points from within the storage area, but this functionality is not currently used within HEC-RAS (Ackerman, 2009). ICPR allows storage areas using either elevation-area or elevation-volume tables. These can be in-line or off-line, and connected either using a channel or a structure (Streamline Technologies, 2014 a). MIKE 11 supports off-line reservoirs as Side Structures with Reservoirs, where the reservoir storage is defined by an elevation-area or elevation-volume table (DHI, 2012 c). In-line reservoirs are included in the MIKE HYDRO module using elevation-volume tables and an optional outlet structure (DHI, 2012 e). Reservoirs in Arc Hydro are defined using a Waterbody polygon which is connected to the river at an outlet junction. The specific properties of a Waterbody are soft typed using user-defined tables (Maidment, 2002).

It is recommended that a transfer standard be able to support both the tabular formats in general use, as well as more explicit geometry using contour polygons and an outer boundary polygon. Elevation-area and elevation-volume tables can be readily derived from a set of closed contours. This would allow detailed lake bathymetry to be communicated, and support integration with 2D simulation models. Full support of interfaces between rivers and storage areas requires structures, which are out of scope for this paper. However, a relatively simple approach that would cover many use cases would be to allow reservoirs to have multiple weirs, where a weir has a unique identifier and either 3D georeferenced geometry or a station-elevation table. This approach is used for RiverML 0.2 as described in Chapter 4.

3.2.4 Linear Attributes Harmonization

3.2.4.1 Marked locations

HEC-RAS and MIKE11 allow specific vertices to be marked with attributes (Brunner, 2010 b; DHI, 2012 c). Both applications support left and right levee banks and left and right flow banks. Levee banks are used in cases where the highest points on either side of the section are not the end vertices, which may be due to a structure or simply the natural shape of the landscape. During computation, flow area is restricted to be within the levee boundaries unless the water surface overtops the levee elevation. Flow banks (also called channel banks) are used to divide the cross section into three regions to allow for differences in behavior between the channel and the surrounding floodplain. HEC-RAS allows levees to be assigned both a station and an elevation, for cases where the highest elevation is not captured by the surface geometry. MIKE11 has additional markers for the lowest point of the section (thalweg), left and right coordinate markers (used for interpreting the correction angle), and any number of user-defined markers.

3.2.4.2 Hydraulic Coefficients

There are a number of hydraulic coefficients used in 1D modeling. Expansion and contraction coefficients apply to a cross section as a whole. Roughness coefficients can apply to the entire section or to a portion of it. HEC-RAS, MIKE11, and ICPR each allow Manning’s n roughness coefficients to be specified for discrete segments of a cross section (Brunner, 2010 b; DHI, 2012 c; Streamline Technologies, 2014 a). ICPR allows separate shallow and deep roughness coefficients to be specified, based on the depth of flow. MIKE11 supports additional roughness coefficients: Manning's M (the reciprocal of Manning's n), Chezy number, Darcy-Weisbach, and relative resistance. Relative resistance is relative to a project-defined value; for the purposes of a transfer standard this can be converted to an absolute.

3.2.4.3 Event tables

Arc Hydro and Arc River support linear attributes for cross sections as soft typed 'events'. These are tables linked to a specific cross section which identify the location, and user-defined fields such as attribute type, value, and units. The location can either be a single point on the line, defined using the *m* coordinate system, or a segment of the line defined using a starting and ending *m* coordinate. Event tables support marked locations and hydraulic coefficients, as well as any number of user-defined attributes such as land use or soil type (Maidment, 2002; Kim et al., 2014).

3.2.4.4 Summary

The event formulation provides a flexible mechanism for assigning linear attributes to either cross sections or profile lines. It is recommended that RiverML contain a point event class and a line event class to support either a single *m* location or a range. The recommended properties are: *m* location, type, value, value units, elevation, and elevation units. The value property should support numeric and text values, for cases such as land use. The elevation property would allow features such as levees or even vegetation and buildings to be described. Standard code lists for event types should be created, such as Manning's n, Chezy number, left bank, thalweg, etc. Attributes which apply features as a whole, such as cross section expansion and contraction coefficients, could either be handled using line events specifying the full length of the line, or using a separate soft typing mechanism.

Within the O&M framework, linear attributes could be considered observations with the geometry line being the feature of interest. This would allow metadata such as data source and method of determination to be included, as well as allowing attributes to change over time or season without modifying the underlying geometry. However, none of the formats or applications examined provided mechanisms for storing this metadata, indicating that there is not a strong emphasis within the community for record keeping at this scale. For the prototype schema described in Chapter 4, linear attributes were treated as events rather than as observations in order to keep the information model simpler. This should be re-evaluated for future versions of RiverML.

3.3 RIVER REFERENCING

There are at least four methods of relating features in a system of rivers: topographical, topological, river addressing, and relative addressing. Because each method has advantages and limitations, data models often employ a hybrid approach. The nature of these hybrids varies from model to model. In order to design a standard transfer language, the key elements of each approach need to be extracted such that they can be applied in a clear and consistent manner.

3.3.1 Reference Approach Descriptions

3.3.1.1 Topographical

The topographical approach is based on the principle that water flows downhill, and therefore features pertaining to surface flow can be related based on the shape of the landscape. Given a set of georeferenced locations and a surface DEM, various relationships of interest can be derived. For each location, a flow path to the edge of the DEM or to an internal depression can be determined. Locations which drain to a common outlet belong to the same drainage area; locations which fall on another's flow path are downstream. These principles are used to determine drainage area boundaries, stream centerlines, and network connectivity.

This form of referencing represents a fundamental way in which river networks and hydrologic models are created and utilized in a GIS environment, as well as how 2D hydraulic models are created which allow flow calculations without predetermined flow directions. However, in the context of data exchange, referencing features using a DEM is inefficient. First, high-resolution DEMs require large files which are time-consuming to transfer between users. Second, the burden for determining relationships is placed entirely on the end user, requiring time-consuming and potentially non-trivial analysis. This analysis is required of each end user, which leads to redundant expenditure of effort. Third, compatibility suffers. There is an ever-expanding choice of algorithms for analyzing DEMs, meaning that different results may be achieved. Fourth, the topographical method does not work for pipe networks and other artificial structures which don't follow the natural ground surface. Providing users with the DEM from which features were extracted can be a valuable supplement, but effective data exchange for 1D simulation models requires a more structured format.

3.3.1.2 Topological

The topological approach bypasses geometry and focuses purely on the logical connections. These are derived from knowledge about the topography, but then abstracted into a purely conceptual framework. Locations of interest (herein called nodes) are assigned a unique identifier, and information such as cross section geometry and calculated water surface elevations are linked to each other by referencing these identifiers. This approach fits naturally in a database environment, where separate tables of information are linked through ID keys. Network connectivity can either be established by including a field in each node for identifying the next downstream node, or by maintaining separate tables which list the node order. These nodes function as Information Exchange Points for model coupling, as described in Section 2.5.1. Topological models can be viewed as a schematic diagram of nodes and links. The elements in the schematic diagram might be arranged to approximate the shape of the river network in order to facilitate understanding.

Hydrologic models such as HEC-HMS often use a topological approach (see Figure 3.5). Hydrographs for individual drainage basins are calculated using averaged properties, and these hydrographs interact at river junctions whose precise coordinates are not relevant. The topological approach requires minimal up-front development time; no complicated terrain analysis is required to begin building a model (Scharffenberg, 2013) .

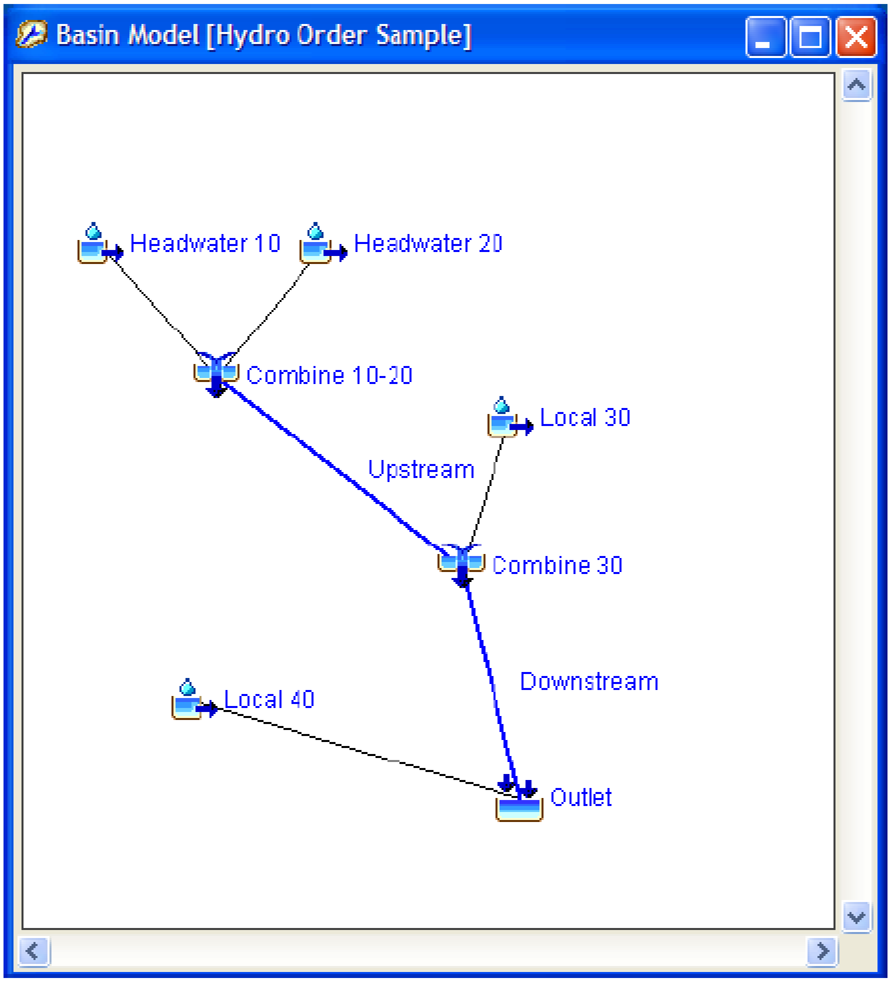


Figure 3.5 Schematic hydrologic network in HEC-HMS (Scharffenberg, 2013)

3.3.1.3 River Addressing

River addressing uses a curvilinear reference system as described in Section 2.4.6 to locate features within a pre-defined river network. In the context of 1D modeling, the *s* coordinate is reported (also called *m*, station or chainage), and the *n* and *z* coordinates are typically omitted. River addresses can be stored as text descriptions such as "on the Colorado River 2 miles upstream of the Highway 35 crossing," or in any formalized structure where the reference line identifier, starting location, distance, and direction are provided.

The river addressing approach requires an up-front effort in determining a network (the ‘blue lines’ on a map), and assigning unique identifiers to each section of river or feature of interest. Once established, however, it provides a compact method of communicating information in a manner that is naturally conducive to river modeling. When defining a river addressing network, the station value of each vertex can either be automatically calculated using the distance from the previous vertex, or be manually assigned. The former case can be communicated as a set of *x*, *y* coordinates, while the latter requires *x*, *y*, *m* coordinates for each vertex.

If a single network is used by many data providers and clients, such as NHDPlus and the Geofabric, interoperability in addressing between clients is achieved without requiring redundant network derivation efforts. Topographical knowledge is abstracted to topological relationships while maintaining a georeferenced framework, which means the key elements of the hydraulic information content of a DEM can be conveyed at a fraction of the file size. An advantage of the river addressing approach over the purely topological approach is that features can easily be assigned locations independently. A schematic diagram does not support automatic interpolation; the node connectivity must be redefined as each new node is added to the system, and the modeler must use judgment to determine when a new node is required. One disadvantage of the river addressing approach is that the network is not necessarily stable. As higher resolution data becomes available, it is often desirable to add additional branches to the network, which may subdivide previous lines and change unique identifiers. The shape of the reference lines is also a function of the data resolution and time, which affects the distance measurements.

3.3.1.4 Relative Addressing

Relative addressing identifies the location of a feature by establishing the bearing and distance from a known point location. This approach provides a geo-referenced location, but does not encode any topological or topographical information, and is therefore not well suited for model coupling. There may be benefit in including the capacity for relative addressing in a standard transfer format; however, this is out of scope for the present harmonization effort.

3.3.2 Reference Approach Examples

3.3.2.1 Arc Hydro

Arc Hydro uses a hybrid approach. The network model component serves as the reference system. Arc Hydro supports two different types of networks: geometric and schematic. Geometric networks are made up of georeferenced lines and points (called HydroEdges and HydroJunctions, respectively) which support river addressing in an *m* coordinate system. Schematic networks are also built of lines and points (called SchematicNodes and SchematicLinks, respectively) which may be georeferenced, but whose links simply connect an upstream node to a downstream node without following the flow path, and thus do not support river addressing and are primarily topological. More generally, topological connections are supported between all features within an Arc Hydro databases through a unique HydroID (Maidment, 2002).

> All hydro features can be associated with any other hydro feature by storing the HydroID of the first feature as an attribute of the second. By this process, drainage areas may be associated with the junctions on the network to which these areas drain, thus defining the correct path of raindrop movement between the land surface and the discharge point on the water flow network. […] The concept that all features in the database are uniquely labeled hydro features is a powerful idea for supporting behavioral modeling, because it means that the database can be

> considered as an integrated whole rather than as a set of separate layers. (Maidment, 2002)

This hybrid approach supports numerous workflows. In the Information Exchange Points approach illustrated in Figure 3.6, flow information calculated in a schematic HEC-HMS environment is communicated to cross sections in a river addressed HEC-RAS environment through a mediating geometric network that stores topological connections using the appropriate IDs (Cesur, 2007).

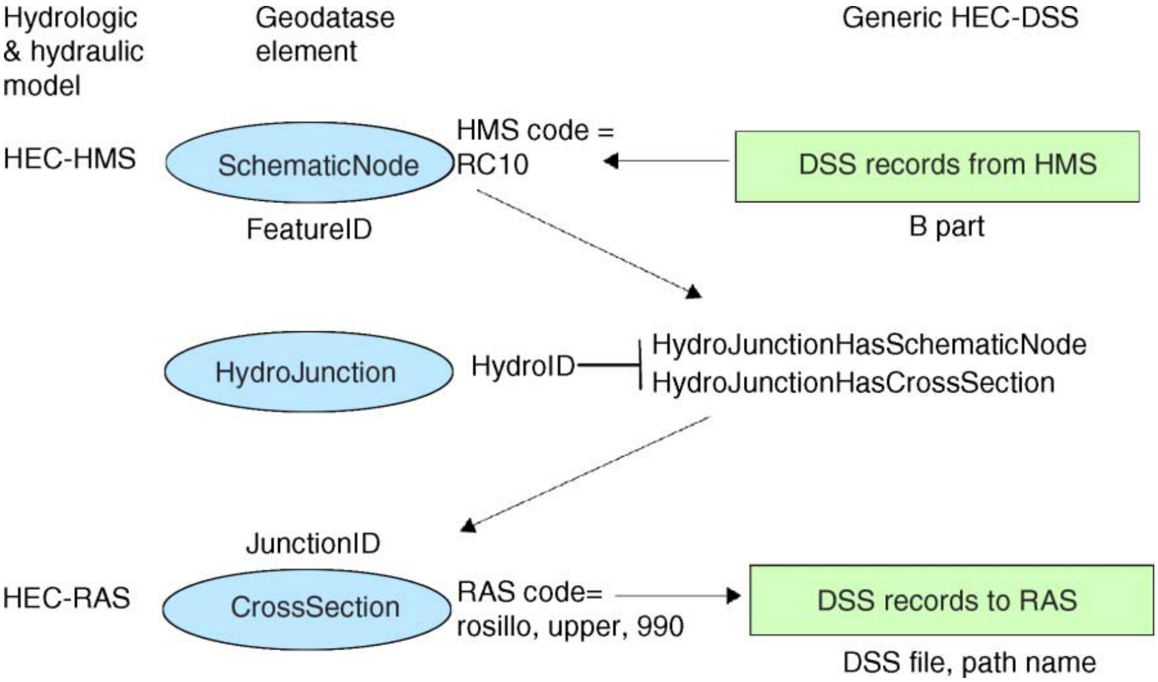


Figure 3.6 Model and GIS Linkage through Information Exchange Points (Cesur, 2007)

3.3.2.2 Arc River

Arc River builds off of the one-dimensional river network in Arc Hydro to provide a three-dimensional framework. Geometry features in Arc River are assigned a dimensionality based on the type of quantities associated with them, and the network connectivity of higher dimensionality features is determined by their associated lower dimensionality features[9]. 1D features support an *s* coordinate, 2D features support *s*, *n* coordinates, and 3D features support *s*, *n*, and *z* coordinates (Kim, 2008; Kim et al., 2014).

> A CrossSectionLine [see Figure 3.7], named after a polyline connecting point measurements across a cross-section, is classified as a one-dimensional object because it contains various cross-section averaged one-dimensional quantities such as discharge. The CrossSectionLine object contains multiple two-dimensional points (having a one to many relationship), which are called as CrossSection2DPoint. So by exchanging key identifiers between them, CrossSection2DPoints are connected to a CrossSectionLine. Similarly, many three-dimensional objects, such as CrossSection3DPoint along the vertical direction of the cross-section can be related to a two-dimensional object, CrossSection2DPoint. (Kim, 2008)

In this hybrid referencing approach, features are defined using a river addressing method, and those features are given IDs which function as topological identifiers. For example, a time series observation would not be directly assigned *s*, *n*, *z* coordinates in Arc River, but rather be associated with a CrossSection3DPoint, and the coordinates inferred from that point (and its associated lower dimensional features).

---

[9] In some cases, 3D features link to 2D features which link to 1D features with network connectivity. In other cases, there is no associated 1D feature and the 2D feature carries the network connectivity.

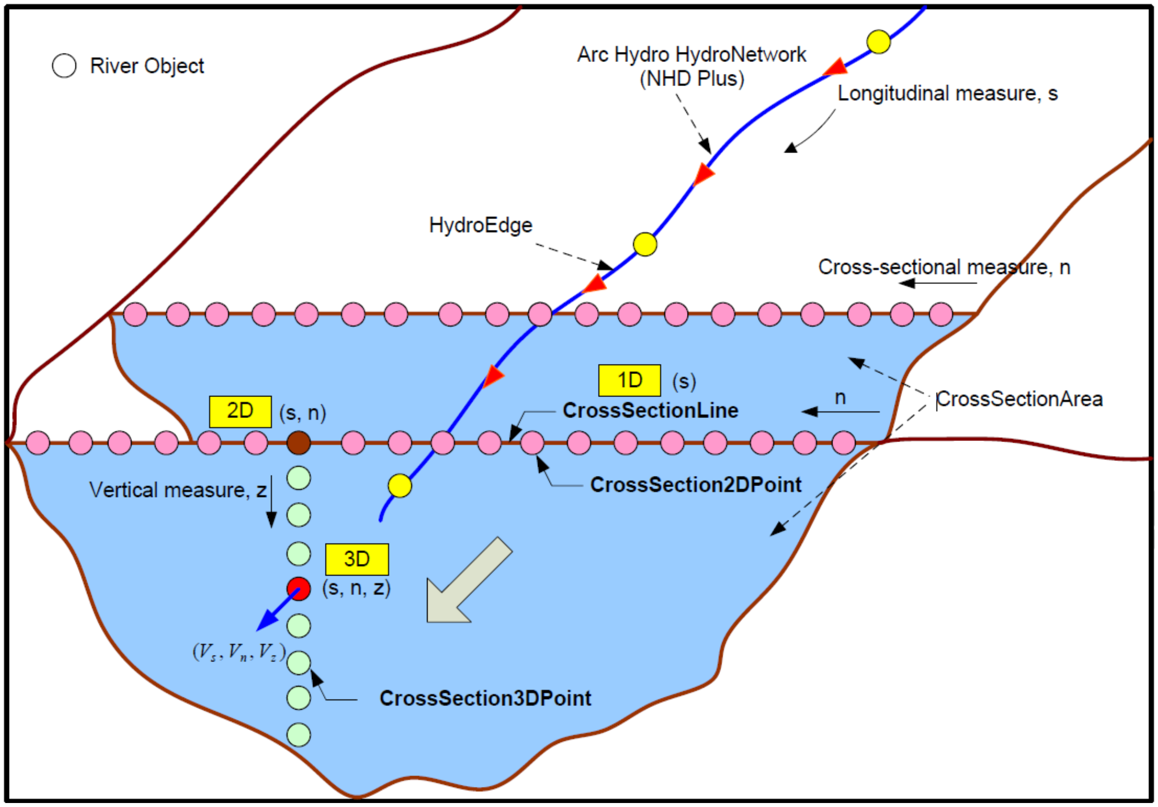


Figure 3.7 The connectivity of a river network in Arc River (Kim et al., 2014)

3.3.2.3 HEC-RAS

HEC-RAS uses a hybrid approach to referencing. A network of rivers, reaches and junctions must be defined prior to adding cross sections. A reach must end at each junction, while a river is made up of one or more consecutive reaches. Cross sections are then addressed based on their station on a particular reach. HEC-RAS distinguishes between station and the distance to the next downstream cross section. Station is a river address for locating and ordering cross sections where values increase with distance downstream. The distance to the next cross section is used in the hydraulic computations, and is specified separately for the main channel, left overbank, and right overbank. HEC-RAS supports both georeferenced and non-georeferenced geometry. In either case, the geometry is for visual purposes only. The spacing and even order of the geometry features drawn by the user may not correlate to the assigned distances and stations. Creating features using georeferencing techniques mitigates confusion from inaccurate drawings. Thus the HEC-RAS reference system consists of three parts: a geometric network used for visualization, a topological connection of rivers, reaches and junctions, and a river address using an arbitrary scale which may or may not correlate with the geometry (Brunner, 2010 b).

3.3.2.4 MIKE11

MIKE11 uses a fairly pure river addressing approach. The ASCII input format for a network specifies a series of points with x, y coordinates, a reach (branch) name, and a station (chainage), which is conceptually the same as providing a set of *x*, *y*, *m* vertices for each reach. Station values can either be automatically calculated, in which case they start at zero for each reach and increase by the Cartesian distance between points, or be user-defined to start at any value. Unlike HEC-RAS, the order of points is defined by the reach geometry, and user-defined station values are restricted to be in numerical order from point to point. Each branch has a flow direction property which specifies whether station values increase or decrease along the direction of flow. Cross sections are assigned a branch and a station. When displayed on the network, cross section locations are interpolated based on the station values for the nearest upstream and downstream points. Thus MIKE11 has a tighter coupling between river addressing and geometry than HEC-RAS does, while still allowing user-defined stationing (DHI, 2012 c).

3.3.2.5 ICPR

ICPR uses a topological approach with optional geometry for visualization. There are three fundamental features in an ICPR model: nodes, links, and basins. These are topological concepts, with basins draining to nodes, and nodes exchanging flow across links. These features can be created independent of geometry, or associated with georeferenced or non-georeferenced coordinates. Links have a ‘from node’ and a ‘to node’ representing the assumed flow direction, although backwater conditions can cause flow to travel opposite of the assumed direction. Cross section geometry is specified as a property of a link; changes in cross sections require intermediate nodes (Streamline Technologies, 2014 a).

3.3.2.6 SPRNT

The SPRNT Netlist is entirely topological. Computational nodes are defined with a unique ID and hydraulic parameters such as cross section geometry and slope, and these nodes are connected in a list specifying the distance between each pair of upstream and downstream nodes. The nodes can optionally be given x, y coordinates, but this is purely for viewing purposes and plays no role in computations or connectivity (Liu, 2014).

3.3.2.7 NHDPlus

NHDPlus represents an authoritative river addressing approach. Using feature extraction techniques, a network has been defined for the continental United States. This is provided to users as shapefiles where each polyline has a unique identifier and supports m addressing. As revisions to the network are required, new versions are released with an explicit versioning scheme that covers both the data model and the data content (McKay et al., 2014). Data users from various agencies and organizations can develop a wide range of catchment and river related datasets using the NHDPlus for addressing, and these datasets can be integrated as long as the versions align.

3.3.2.8 Geofabric

The Geofabric is an authoritative blend of topological and river addressing approaches. As with NHDPlus, a geometric network of streams and nodes extracted from a digital elevation model is provided, which supports river addressing. The Geofabric includes a provision to mitigate disruption due to versioning which functions as a topological approach. A subset of features which are expected to appear in future versions of the network have been assigned a ‘contracted’ ID, which will remain stable. A subset of these contracted nodes have topological links which form stable, simplified network independent of geometry (Bureau of Meteorology, 2012).

> The resulting contracted catchments form a stable, logical, dendritic hierarchy that can be reliably reproduced when moving to a higher resolution or larger scale data. They also provide a stable set of catchments that, among other things, can be aggregated to a number of types of water reporting areas depending on the use case. (Bureau of Meteorology, 2012)

Points representing monitoring stations which do not fall directly on the geometrical network have been associated with a ‘ghost node’ on the network that can be used for addressing and catchment delineation, without losing the original real-world coordinates (Bureau of Meteorology, 2012). This method is also supported by the Arc Hydro database structure (Maidment, 2002), and should be included in RiverML.

3.3.2.9 HY_Features

HY_Features is a general model which supports both topological and network addressing approaches. The Outfall feature class which serves to connect drainage basins with rivers is a topological concept without specific geometry. River addressing is accomplished through the IndirectPosition which uses a linear distance along a defined river network. IndirectPosition supports absolute and relative measures, as well as verbal descriptions.

3.3.2.10 OpenMI

OpenMI allows models to define input and output exchange items which are then connected by the user. These exchange items can be spatial features such as points, lines, or polygons, or non-spatial with a unique identifier. The geometry can be plotted in a GIS environment to aid the user in connecting input and output items. However, the OpenMI framework does not establish an explicit addressing scheme. “The onus is on the user who links models together to ensure that the linkages are physically and numerically meaningful. Clearly in many instances the sources made visible on one component will not be compatible with the targets on another (Moore, 2010).” Therefore, OpenMI can be considered a topological approach even when geometry is present.

3.3.3 Reference Harmonization

The river addressing approach is a geometry-based solution, and requires support from geospatial algorithms, whereas the topological approach is a database-like solution, and requires support from database algorithms.

As each method of river referencing has advantages and disadvantages, a transfer standard should support a variety of approaches. It is recommended that these four methods (topographical, topological, river addressing, and relative addressing) be viewed as analogous to orthogonal unit vectors, such that information stored in an information model using any hybrid referencing format can be split into distinct components for transfer, and recombined as needed for the destination information model. This requires that the four fundamental methods be clearly differentiated, and mechanisms provided to establish unambiguous connections between them. For example, an Arc Hydro georeferenced schematic network contains a blend of topology and geometry. When converted to a standard format, it would be split into two components: a geometric component consisting of the SchematicNode points, and a schematic component consisting of node IDs and connectivity tables. The points could be integrated into a full geometric network if one has been defined, and the node IDs would be associated with their respective points. The arbitrary stationing allowed by HEC-RAS and MIKE11 networks requires that a transfer schema support user-defined *m* values on network lines.

The RiverML 0.2 prototype described in Chapter 4 is designed to satisfy these conditions, and draws heavily from the Arc Hydro model in which networks (either schematic or geometric) serve as the referencing system for drainage, channel, and hydrography features. The model allows schematic reference features to be related to geometric reference features, which provides a simple mechanism for coupling schematic hydrologic models to geometric hydraulic models. Of the four methods of river referencing, RiverML 0.2 focuses on topological and river addressing, with limited support for topographical and no support for relative addressing.

3.4 TIME-VARYING GEOMETRY

There are two different approaches to representing changing river geometry. The first is to define scenarios, which are snapshots of river geometry under a particular set of circumstances. Under this approach, a modeler may set up one scenario based on a USGS topographic map from 1970, a second scenario based on a LiDAR survey from 2012, and a third scenario based on proposed future modifications to a channel. In general, hydraulic modeling software allows the user to set up a project, which is a top level organization framework for a particular investigation, and each project can contain any number of scenarios, which represent a consistent set of features which can be used for a particular simulation run. For MIKE11, scenarios are defined by assigning a matching TopoID to all cross sections belonging to a scenario (DHI, 2012 c). HEC-RAS allows the user to set up multiple geometry files within a particular project. Each geometry file can be given a description to aid in interpretation (Brunner, 2010 b). ICPR defines a project as a set of scenarios which describe the geometry, connectivity, and simulation settings, and a set of ancillary data which is available to all scenarios (see Figure 3.8) (Streamline Technologies, 2014 b).

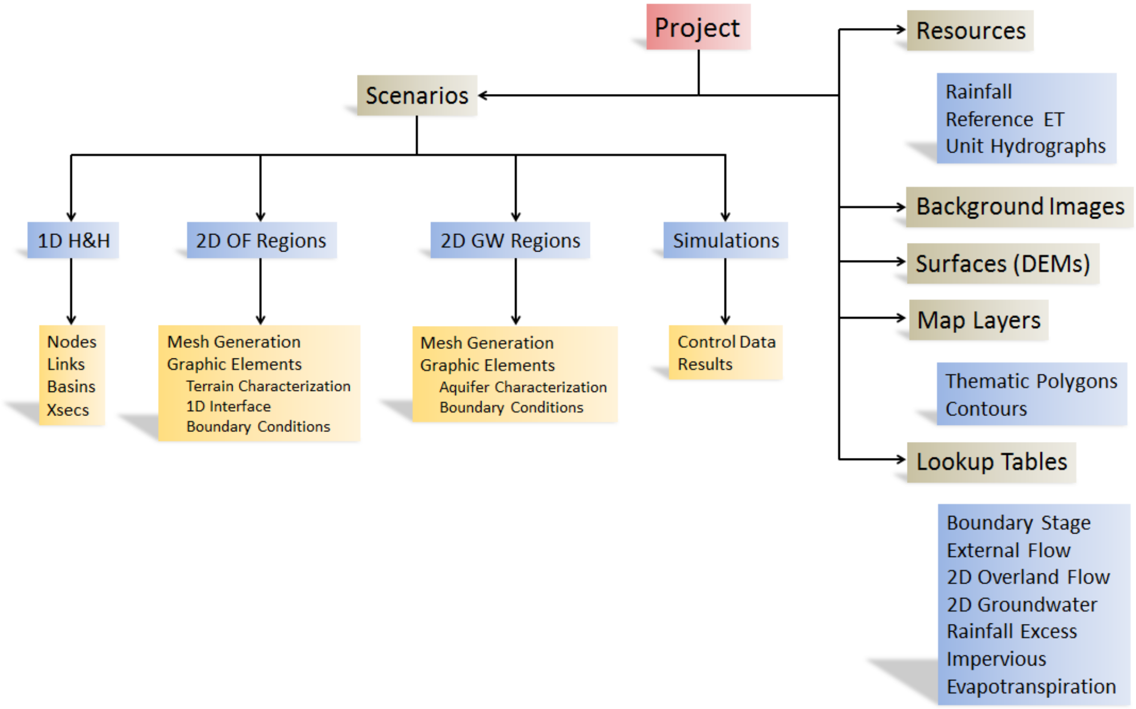


Figure 3.8 ICPR project structure (Streamline Technologies, 2014 b)

The second approach is to define feature series, which tracks the evolution of a particular geometry feature over space and time. This is accomplished in Arc River by creating a set of features, each with its own geometry and time stamp, then assigning a common seriesID to each. A feature series may be used for situations such as tracking the location of the bank lines during a highly erosive storm, or the position of a fish or cloud of pollutant during an environmental study (Kim, 2008).

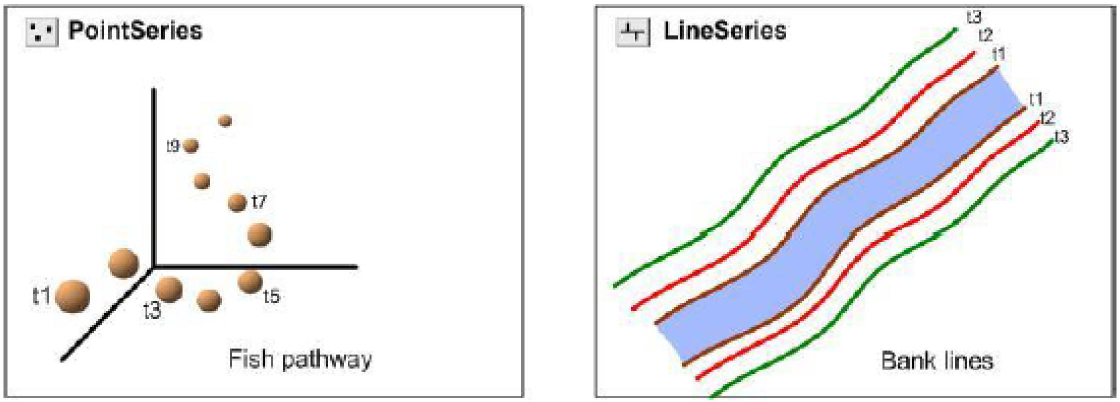


Figure 3.9 Feature series (Kim et al., 2014)

Traditional one-dimensional hydraulic models use a scenario approach, and thus it is recommended that RiverML initially be focused in this direction. Advanced environmental modelling of rivers may benefit from a feature series approach such as that supported by ArcRiver, and thus it may be advantageous for later versions of RiverML to incorporate this capability. The two methods of describing time-varying features are not mutually exclusive; a set of feature series could be included in a scenario.

3.5 PRINCIPLES FROM WATERML 2.0 HARMONIZATION

During the harmonization and development stages of WaterML 2.0, a number of challenges were encountered which can be considered general to the development of any domain-specific standard. The principles adopted by the WaterML 2.0 team to meet these challenges set a successful precedent; it is therefore recommended that RiverML adopt a similar approach, as described below.

3.5.1 Optimization

It was found during the WaterML 2.0 Interoperability Experiments that the XML files were “more complex and larger than most of the existing formats, and thus slower to produce and parse using standard XML tools (Taylor et al., 2014).” This was due to the

GML rules which limit the use of certain XML features, and the level of metadata which WaterML 2.0 was capable of conveying.

> The decision was made not to prematurely optimise the WaterML2.0 XML encoding in order to save a few XML elements. The [Interoperability Experiments] demonstrated that use of XML tools such as FastInfoset (a binary encoding of XML) could significantly reduce issues relating to XML generation and parsing. It was also shown that simple compression of XML removed issues relating to file size. (Taylor et al., 2014)

3.5.2 Hard Typing vs. Soft Typing

Elements of a data models can be constructed in a hard-typed fashion, where the element name and value type are pre-defined by the standard, or soft-typed, where the element name and value type are defined by the user. WaterML 2.0 uses a combination of hard- and soft-typing.

> Balancing between hard-typing and soft-typing in descriptions of concepts and types in a conceptual model is important. Soft typing allows flexibility but reduces the specificity of the model, which creates ambiguity, reduces interoperability and affects the validation process of encoded documents; hard-typing tightly defines concepts making semantics clear and validation using existing tools easier, but reduces the ability to extend definitions without revising the schema. The general approach is that if a concept is core to the domain and can be harmonised to provide a common definition, then it is a candidate to be hard-typed. Concepts that are more specific to particular organisations or contexts should be made available through the use of soft-typed definitions. (Taylor, 2010)

3.5.3 Vocabularies

Concepts within hydrology often have terms that vary based on region and organization. For example, “station” and “chainage” both refer to a measured distance along the length of a river. In other cases the same term may be assigned different meaning. WaterML 2.0 did not attempt to harmonize vocabularies for all information types that might be conveyed by the standard, but rather only those which were of importance to the structure of the standard (Taylor et al., 2014). For non-structural components where standard vocabularies are seen as valuable, WaterML 2.0 can reference URIs where definitions are recorded (Taylor, 2012).

3.5.4 OGC Framework

The OGC has created a number of standards for geospatial information. These standards can be linked together and extended as needed to provide an integrated suite of standards for a particular domain. By developing new standards using the existing standards as building blocks, duplication of effort is reduced and cross-domain interoperability is made easier. Three existing standards which are particularly relevant to river modeling are GML, O&M, and WaterML 2.0. As GML is built on well-defined relationships between features and lies at the heart of the OGC suite of standards including WaterML 2.0, it is an ideal candidate for building an application schema for the domain of river modeling. O&M provides a mechanism to assign metadata to data on a feature-by-feature basis. WaterML 2.0 allows time series for any variable such as water surface elevation and discharge to be described. Together these standards lay the groundwork that a transfer language for river geometry and flow can expand on.

Chapter 4: Prototype Schema – RiverML 0.2

4.1 OVERVIEW

In this chapter, a conceptual outline for an open standard for encoding river surface and water observations data is presented. The model, called RiverML 0.2, is a prototype intended to demonstrate a feasible approach to implementing the findings of the river data harmonization effort presented in Chapter 3. RiverML 0.2 should be evaluated by the water resources community for clarity, functionality, and conformance to existing standards. Where unresolved questions have been identified by the author, they have been noted throughout the documentation as requiring further work, and summarized in Chapter 6.

RiverML 0.2 is based on the information models of Observations and Measurements version 2.0 (O&M) and WaterML2.0, and implemented as an application schema according to the rules of Geography Markup Language version 3.2 (GML). O&M is a conceptual model for describing observations. WaterML2.0 is a standard for encoding water observations data. GML is an extensible international standard for the exchange of spatial data.

RiverML is designed as an extensible schema to allow encoding of data to be used in a variety of exchange scenarios. Example areas of usage are: exchange of data for operational flood modelling programs; dissemination of national river morphology data; cross-border exchange of observational data; supporting operation of infrastructure (e.g. dams, supply systems); enhancing disaster management through data exchange; facilitating the protection of aquatic ecology; and exchange in support of national reporting.

The core aspect of the model is a consistent framework for relating observational data (e.g. cross section geometry and water surface elevation) to clearly defined reference features. These reference features can be purely topological nodes and links, a geometrically defined network, or a combination of the two. The distinction in RiverML between reference features and geometry observations encourages network stability. As higher resolution data becomes available and desirable for both cartographic and analytic purposes, a stable FlowlineEdge reference feature can be described by progressive ProfileLines. The ReferenceNetwork need only be modified when there are topological changes or when the level of detail is too imprecise for the desired river addressing accuracy.

RiverML 0.2 consists of two parts: a conceptual UML model for observational data as a profile of ISO19156 – Observations & Measurements, and an implementation of the model in XML Schema, specifically a GML 3.2 conformant XML schema. This separation allows capturing the information model in an implementation-agnostic fashion, using UML, to allow multiple implementations to occur. In addition to GML, other implementations in future work may include JSON, NetCDF, non-GML conformant XML, etc.

The general characteristics of RiverML 0.2:

1. Communicates the semantics of hydraulic data used in one-dimensional flow models;
2. Allows observational data regarding geometry and time series to be unambiguously related via association to stable reference features.
3. Allows changes in river geometry as a function of time or survey techniques to be expressed.

4. Enables the distinction between actual and hypothetical surface data, such as the existing conditions and those proposed by a flood management project.
5. Allows definition of scenarios that specify a subset of surface and water observations which represent a coherent unit, such as the inputs and outputs for a specific hydraulic model run.
6. Provides a flexible transfer schema which can be re-used to meet a number of exchange objectives;
7. Enables the encoding of metadata relating to the provenance of surface data (i.e. how the geometry was created).

4.2 UML CONCEPTUAL MODEL

The RiverML 0.2 conceptual model presented uses the Unified Modeling Language (UML), and was designed in Enterprise Architect by Sparx Systems. More information about UML, including tutorials, can be found at http://www.uml.org/.

In order to align with existing standards while preserving readability, this chapter contains both paraphrased and verbatim quotes from related standards without using quotation marks or block quote formatting. Citations to the source material are provided where significant intellectual content is included. In the UML diagrams, properties within a feature class are hidden except in the definition diagram for that class. In some diagrams, connections between classes have been hidden where they are not important for understanding the relationships being demonstrated.

4.2.1 HydroFeature

4.2.1.1 HydroFeature

This is an abstract class which serves as the generic template for most RiverML feature classes, comparable to HydroFeature within the Arc Hydro data model. See Figure 4.1.

Properties:

id [1]: A mandatory GML property which provides a unique identifier to the object within the scope of the document (Portele, 2007). This identifier is comparable to HydroID within Arc Hydro, except it accepts a string rather than integer. As this is inherited from GML, when implementing RiverML in a non-GML environment an *id* attribute should be added to HydroFeature.

name [0..]:* An optional GML property which provides a label or identifier for the object, commonly a descriptive name. An object may have several names, typically assigned by different authorities. *gml:name* uses the *gml:CodeType* content model. The authority for a name is indicated by the value of its (optional) *codeSpace* attribute. The name may or may not be unique, as determined by the rules of the organization responsible for the *codeSpace*. In common usage there will be one name per authority, so a processing application may select the name from the *codeSpace* that it prefers. (Portele, 2007). This is comparable to HydroCode within Arc Hydro, with the added flexibility of allowing multiple names. As this is inherited from GML, when implementing RiverML in a non-GML environment a *name* property should be added to HydroFeature.

description [0..1]: An optional GML property that provides a text description of the object (Portele, 2007). As this is inherited from GML, when implementing

RiverML in a non-GML environment a *description* property should be added to HydroFeature.

parameter [0..]*: A soft-typed field for arbitrary name-value pairs, using the O&M NamedValue type[10]. This may be used to extend the available metadata properties (Taylor, 2012).

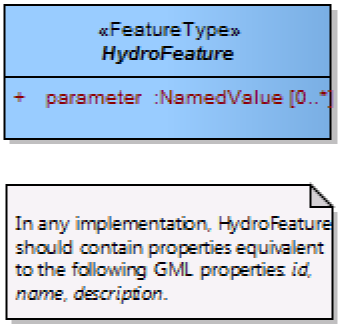


Figure 4.1 UML Diagram: HydroFeature

4.2.2 Collection

4.2.2.1 RiverCollection

RiverML defines a generic collection feature type, *RiverCollection*, to allow the grouping of observations and/or sampling features with metadata to describe the nature of the collection. Such collections are required in a number of data exchange scenarios; whether the underlying transport technology is web services, FTP or other technologies. The grouping may indicate a relationship between the contained entities, however the relationship will depend on the individual use of the collection class. The collection class may be replaced by services that already define such collections - such as in the Sensor

---

[10] In the UML model created for RiverML 0.2, difficulty was encountered referencing the O&M version of NamedValue, so a local copy was made. This should be corrected in future versions.

Observation Service - but the model may be used as a guide to the content of collections (Taylor, 2012). RiverCollection inherits from HydroFeature. See Figure 4.2.

Properties:

metadata [0..1]: Describes the metadata associated with the document. See Section 4.2.2.2 for the definition of DocumentMetadata.

scenario [0..]*: This property allows for multiple Scenario members to be included in the collection document. See Section 4.2.3.1 for the definition of Scenario.

referenceNetwork [0..]*: This property allows for multiple ReferenceNetwork members to be included in the collection document. See Section 4.2.4.1 for the definition of ReferenceNetwork.

surfaceObservation [0..]*: This property allows for multiple SurfaceObservation members to be included in the collection document. See Section 4.2.7.1 for the definition of SurfaceObservation.

crossSectionObservation [0..]*: This property allows for multiple CrossSectionObservation members to be included in the collection document. See Section 4.2.11.14.2.3.1 for the definition of CrossSectionObservation.

profileLineObservation [0..]*: This property allows for multiple ProfileLineObservation members to be included in the collection document. See Section 4.2.12.1 for the definition of ProfileLineObservation.

shorelineObservation [0..]*: This property allows for multiple ShorelineObservation members to be included in the collection document. See Section 4.2.13.1 for the definition of ShorelineObservation.

reservoirObservation [0..]*: This property allows for multiple ReservoirObservation members to be included in the collection document. See Section 4.2.14.1 for the definition of ReservoirObservation.

structureObservation [0..]*: This property allows for multiple StructureObservation members to be included in the collection document. See Section 4.2.15.1 for the definition of StructureObservation.

timeseriesObservation [0..]*: This property allows for multiple WaterML 2.0 TimeseriesObservation members to be included in the collection document. See Section 4.2.16 for a discussion of TimeseriesObservation.

4.2.2.2 DocumentMetadata

Describes the metadata associated with the document. DocumentMetadata inherits from nothing. See Figure 4.2.

Properties:

defaultReferenceSystem [0..1]: Specifies the default coordinate reference system for objects in the document. Value shall be of type SC_CRS as defined in Lott (2010). In future versions of RiverML it may be advantageous to allow separate vertical and horizontal reference systems.

generationDate [1]: Specifies the date the document was generated (Taylor, 2012).

generationSystem [0..1]: textual description of the system that generated the document (Taylor, 2012).

version [0..1]: This version property is distinct from the schema version. It indicates the package version that is being used where package is the combination of schema, vocabularies and any profiles used. This allows versions to be more specific based on their implemented usage of the schema (Taylor, 2012).

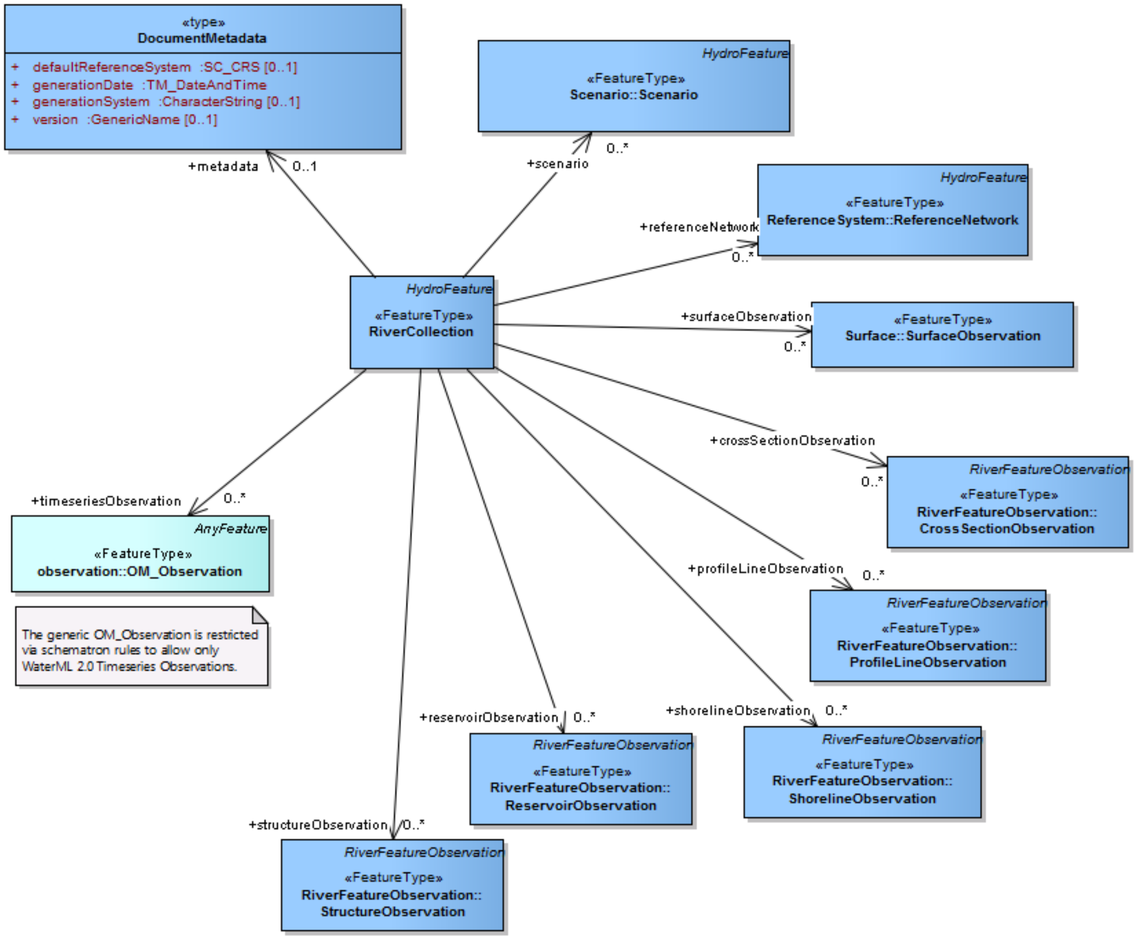


Figure 4.2 UML Diagram: Collection

4.2.3 Scenario

4.2.3.1 Scenario

A Scenario is a grouping of observations which are valid within a particular context. Observations are defined elsewhere in the document and referenced in a list using their unique GML ids. The Scenario class provides the ability to easily filter a complex document to allow visualization and analysis on relevant coherent portions. Scenario inherits from HydroFeature. See Figure 4.3.

Scenarios can be used to compare changes in geometry over time, differences in geometry due to survey and extraction techniques, differences between existing and proposed conditions, or time series results of various model runs. Each observation can be referenced as valid for any number of scenarios, which allows file size to be minimized. For example, if evaluating Existing and Proposed scenarios where only one cross section is modified, only one additional CrossSectionObservation is required. The two Scenarios would have identical lists of valid CrossSectionObservation members except where the modification applies. Scenario may be omitted when all observations belong together and no additional metadata is needed.

Properties:

validSurfaceObservation [0..]*: This property allows for multiple SurfaceObservation members to be included in a Scenario, by reference to their unique *id*.

validRiverFeatureObservation [0..]*: This property allows for multiple RiverFeatureObservation members to be included in a Scenario, by reference to their unique *id*.

validTimeseriesObservation [0..]*: This property allows for multiple TimeseriesObservation members to be included in a Scenario, by reference to their unique *id*.

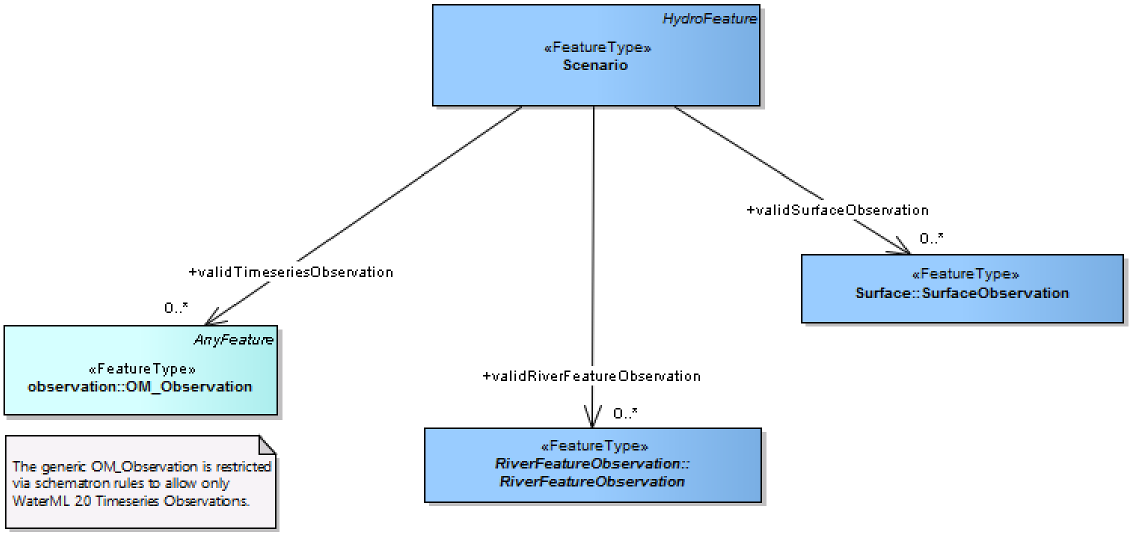


Figure 4.3 UML Diagram: Scenario

4.2.4 Reference Network

4.2.4.1 ReferenceNetwork

A ReferenceNetwork is a consistent collection of ReferenceFeature members. It is not required that all members of a ReferenceNetwork be topologically connected; isolated members or groups are allowed. A ReferenceNetwork may be established based on an authoritative source such as the NHD or Geofabric and used by multiple data users to provide consistency across RiverML documents, or be created for a specific project and used with a limited scope. In the case of an authoritative data source, each version of the source dataset which modifies the topology requires a separate ReferenceNetwork. The *name* and *description* property values should be of sufficient detail to identify the source and version of the network. ReferenceNetwork inherits from HydroFeature. See Figure 4.4.

A ReferenceNetwork can contain GeometricReferenceFeature members, SchematicNetworkFeature members, or any mix of both. Where both types are used, SchematicReferenceFeature members can be associated with GeometricReferenceFeature members using the *relatedReferenceFeature* property. This allows a network which was originally developed schematically to be gradually upgraded to a more detailed geometric representation.

Because the ReferenceFeature members are essential for interpreting the contents of a RiverML document, as a general rule a ReferenceNetwork should be fully described within the document. However, if a standard ReferenceNetwork is established based on an authoritative source and published with a stable URI, it could be acceptable to link to this via external references.

Properties:

referenceFeature: [0..]*: This property allows for multiple ReferenceFeature members to be included in a ReferenceNetwork.

4.2.4.2 ReferenceFeature

This is an abstract class which serves as a template for specific feature classes which extend the O&M SamplingFeature class. ReferenceFeature members serve as the *featureOfInterest* for GeometryObservation and TimeseriesObservation members. ReferenceFeature inherits from HydroFeature. See Figure 4.4.

If future versions of RiverML are expanded to include description of drainage areas for hydrologic model integration, additional schematic and geometric ReferenceFeature classes can be defined to support area-based observations.

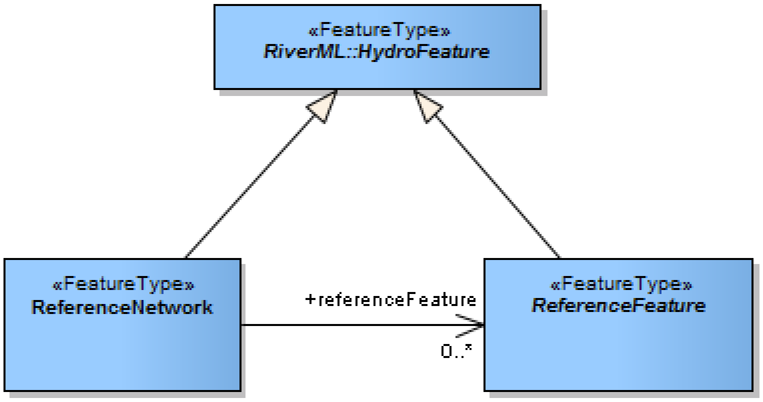


Figure 4.4 UML Diagram: ReferenceNetwork

4.2.5 Schematic Reference Features

4.2.5.1 SchematicReferenceFeature

This is an abstract class which serves as a template for reference features with no explicit geometry. SchematicReferenceFeature inherits from ReferenceFeature and O&M SF_SamplingFeature. See Figure 4.5.

Properties:

relatedReferenceFeature [0..1]: Specifies an equivalent GeometricReferenceFeature by reference to its unique *id*. Observations associated with such ReferenceFeature members should be interpreted as applicable to both the GeometricReferenceFeature and the SchematicReferenceFeature. For example, if results from a schematic hydrologic catchment model are merged with a geometric hydraulic river model, the catchment outfall Node members can be linked to catchment Junction members using *relatedReferenceFeature*. While RiverML 0.2 does not explicitly enforce this, the value for *relatedReferenceFeature* should be restricted such that a Node can only associate with a Point, and a Link can only associate with an Edge. Care should be taken that the topological relationships among SchematicReferenceFeature members do not conflict with those among their associated GeometricReferenceFeature members, though differences in resolution are acceptable.

4.2.5.2 Node

Represents a location of interest such as the outfall of a drainage area, the intersection of two rivers, a bridge, or a monitoring station. A Node has no explicit geometry and consists fundamentally of a unique *id*. Node members are not required to be topologically connected; isolated features such as an atmospheric monitoring station may be defined. Node inherits from SchematicReferenceFeature. See Figure 4.5.

Properties:

relatedReferenceFeature [0..1]: The value of this property inherited from SchematicReferenceFeature shall be restricted to type Point.

4.2.5.3 Link

Provides the topological relationships between nodes. A Link has no explicit geometry and consists fundamentally of a unique *id* and a pair of connected Node members. A *fromNode* and *toNode* are defined according to the assumed direction of flow. However, certain conditions such as backwater effects may cause flow to travel against the assumed direction. This should be indicated with negative values in the appropriate TimeseriesObservation. See Figure 4.5.

Multiple Link members can share the same *fromNode* and/or *toNode* values to describe situations such as split flow or braided streams. RiverML 0.2 does not enforce a dendritic network; topological loops are allowed. It may be useful for future versions of

RiverML to have an optional conformance class which requires that networks be dendritic.

Properties:

relatedReferenceFeature [0..1]: The value of this property inherited from SchematicReferenceFeature shall be restricted to type Edge.

fromNode [1]: Specifies the upstream Node in the assumed flow direction by reference to its unique *id*.

toNode [1]: Specifies the downstream Node in the assumed flow direction by reference to its unique *id*.

reachCode [0..1]: A label or identifier for a set of Edge members linearly connected to form a single river reach, usually defined between stream confluences (Maidment, 2002). If provided, this shall be unique within the scope of Edge members that share a common *riverCode* and ReferenceNetwork. This is analogous to reachCode within the Arc Hydro data model, and the Reach name in HEC-RAS[11].

riverCode [0..1]: A label or identifier for a set of Edge members linearly connected to form a single river, usually defined from a headwater point to the confluence with a larger water body. If provided, this shall be unique within the scope of a ReferenceNetwork. This is analogous to the River name within in HEC-RAS.

---

[11] In HEC-RAS, Reach names must be unique. If multiple line segments share a Reach name, they must either be merged to form a single line, or be given unique names such as by adding a suffix.

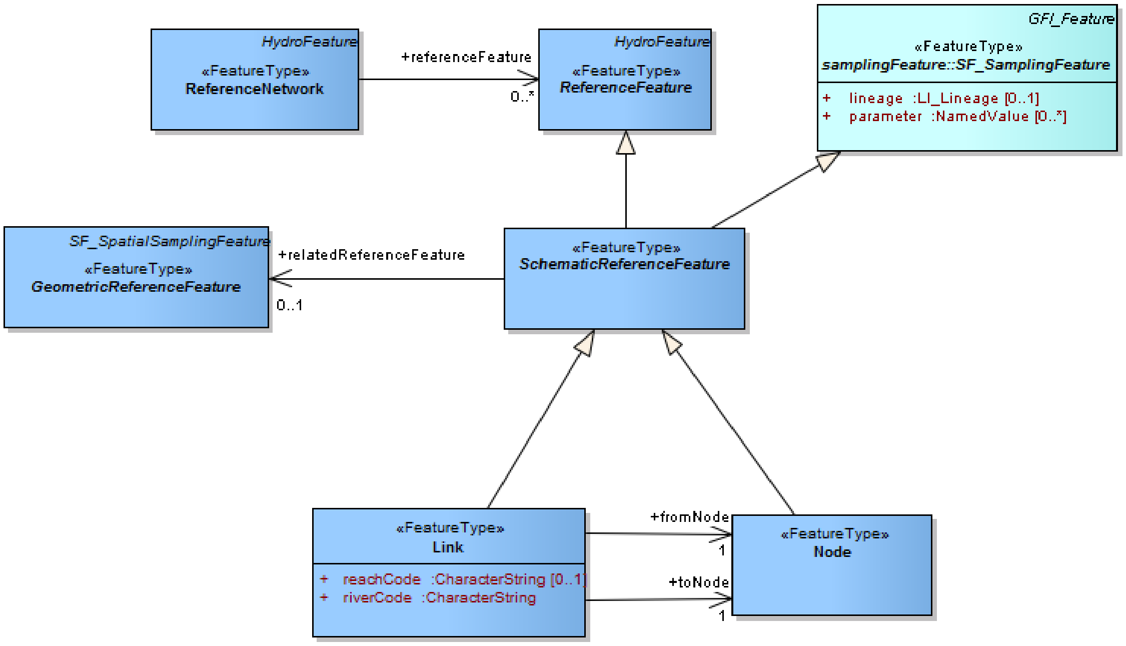


Figure 4.5 UML Diagram: SchematicReferenceFeature

4.2.6 Geometric Reference Features

A typical geometric network will consist of Junction members and FlowlineEdge members, which are comparable to Node and Link members respectively. Additional feature classes are provided to handle special cases.

4.2.6.1 GeometricReferenceFeature

This is an abstract class which serves as a template for reference features that have explicit geometry. GeometricReferenceFeature inherits from ReferenceFeature and O&M SF_SpatialSamplingFeature. See Figure 4.6.

Properties:

referenceSystem [0..1]: Specifies the coordinate reference system. Value shall be of type SC_CRS as defined in Lott (2010). This overrides *defaultReferenceSystem* in

DocumentMetadata. In future versions of RiverML it may be advantageous to allow separate vertical and horizontal reference systems.

4.2.6.2 Point

This is an abstract class which serves as a template for reference point features. It represents a location of interest such as the outfall of a drainage area, the intersection of two rivers, a bridge, or a monitoring station. A Point has explicit geometry and consists fundamentally of a unique *id* and the coordinates of a single vertex. Point members are not required to be topologically connected; isolated features such as an atmospheric monitoring station may be defined. Point inherits from GeometricReferenceFeature. See Figure 4.6.

Properties:

shape [1]: The value shall be of type GM_Point as defined in ISO 19107. The point can have either *x*, *y* or *x*, *y*, *z* coordinates.

4.2.6.3 Junction

A Junction is a location which serves as a topological endpoint to one or more OnNetworkEdge members. Junction inherits from Point. See Figure 4.7.

4.2.6.4 EdgePoint

An EdgePoint is a location which lies between the end vertices of a specific OnNetworkEdge member. The EdgePoint coordinates shall either match an interior vertex of the OnNetworkEdge or be on the interpolated path between two vertices. EdgePoint inherits from Junction. See Figure 4.7.

There are two primary use cases for the EdgePoint class. The first is to identify locations of interest along the flow path which do not warrant splitting an Edge into two smaller sections, such as when identifying the nearest stream location for an OffNetworkPoint. The second is to function as a junction allowing a new OnNetworkEdge to be added to an existing ReferenceNetwork without disrupting existing identifiers. This could be used when high resolution features are modeled in conjunction with a lower resolution authoritative ReferenceNetwork. At present, most software applications reviewed do not support this type of non-disruptive junction. The import procedures from a RiverML file would include subdividing Edges as needed, and when subsequently exported back to RiverML an updated ReferenceNetwork would be required. While the use of EdgePoints as non-disruptive junctions may be limited in the near future, the concept has potential to enhance the stability and compatibility of ReferenceNetworks given appropriate software support.

Properties:

onEdge [1]: Specifies an OnNetworkEdge by reference to its unique *id*.

edgeMeasure [1]: An absolute or relative measure for the position along the associated OnNetworkEdge. Future versions of RiverML should include an explicit method for distinguishing between absolute and relative measures. For RiverML 0.2, values between 0 and 1 are assumed to be relative.

4.2.6.5 OffsetEdgePoint

The OffsetEdgePoint class provides a mechanism for communicating the 2D and 3D river observations supported by the Arc River data model. OffsetEdgePoint members are assigned a location along an OnNetworkEdge as well as a perpendicular and vertical distance corresponding to *s, n, z* coordinates. OffsetEdgePoint is analogous to the Arc River class RiverPoint. A cross section line is not explicitly provided as a reference feature, but can be interpolated by connecting 2D OffsetEdgePoint members whose onEdge and edgeMeasure values match. Further testing of the OffsetEdgePoint class is recommended prior to inclusion in future versions of RiverML. OffsetEdgePoint inherits from Point. See Figure 4.7.

Properties:

onEdge [1]: Specifies an OnNetworkEdge by reference to its unique *id*.

edgeMeasure [1]: An absolute or relative measure for the position along the associated OnNetworkEdge. Future versions of RiverML should include an explicit method for distinguishing between absolute and relative measures. For RiverML 0.2, values between 0 and 1 are assumed to be relative.

crossMeasure [1]: An absolute or relative measure for the position perpendicular to the associated OnNetworkEdge. The measure should correspond to the implied cross section line obtained by connecting 2D OffsetEdgePoint members whose onEdge and edgeMeasure values match, starting at zero for the farthest left point when looking along the flow direction.

hasZ [1]: A Boolean value indicating whether an elevation value is provided. When the value is false, the OffsetEdgePoint should be considered analogous to a River2DPoint in Arc River even if a *z* coordinate is included in the *shape* property.

elevation [0..1]: The elevation of the feature in the applicable coordinate reference system. Elevation can either be provided as a *z* coordinate in the *shape* property or as a separate *elevation* property.

4.2.6.6 OffNetworkPoint

An OffNetworkPoint is a location which does not participate directly in a topological network. It can be used to specify features such as buildings or monitoring stations. OffNetworkPoint inherits from Point. See Figure 4.7.

Properties:

nearestJunction [0..1]: Specifies a Junction by reference to its unique *id*. This may be an EdgePoint.

4.2.6.7 Edge

This is an abstract class which serves as a template for reference edge features. An Edge has explicit geometry. Edge members are not required to be topologically connected; isolated features such as cartographic boundaries may be defined. Edge inherits from GeometricReferenceFeature. See Figure 4.6.

The Edge class supports custom stationing using m coordinates. Where m coordinates are specified, they shall be used by all features with an *edgeMeasure* property referencing the Edge. Where no m coordinates are specified, *edgeMeasure* values shall assume linear interpolation starting with 0 at the first vertex.

Properties:

shape [1]: The value shall be of type GM_LineString as defined in ISO 19107. The point can have either (x, y), (x, y, z), or (x, y, z, m) coordinates. Support for additional curve geometries should be evaluated in future versions of RiverML.

4.2.6.8 OnNetworkEdge

This is an abstract class which serves as a template for topologically connected reference edge features. An OnNetworkEdge provides the topological relationships between Junctions, as well as a geometric path which can be used for river addressing. A *fromPoint* and *toPoint* are defined according to the assumed direction of flow. However, certain conditions such as backwater effects may cause flow to travel against the assumed direction. This should be indicated with negative values in the appropriate TimeseriesObservation. OnNetworkEdge inherits from Edge. See Figure 4.8.

Multiple OnNetworkEdge members can share the same *fromPoint* and/or *toPoint* values to describe situations such as split flow or braided streams. RiverML 0.2 does not enforce a dendritic network; topological loops are allowed. It may be useful for future versions of RiverML to have an optional conformance class which requires that networks be dendritic.

In general, the first and last vertices of an OnNetworkEdge should match the coordinates of the *toPoint* and *fromPoint*, respectively. However, there may be specific use cases where this is not feasible, so it is not enforced in RiverML 0.2.

Properties:

fromPoint [1]: Specifies the upstream Junction in the assumed flow direction by reference to its unique *id*.

toPoint [1]: Specifies the downstream Junction in the assumed flow direction by reference to its unique *id*.

reachCode [0..1]: A label or identifier for a set of Edge members linearly connected to form a single river reach, usually defined between stream confluences (Maidment, 2002). If provided, this shall be unique within the scope of Edge members that share a common *riverCode* and ReferenceNetwork. This is analogous to reachCode within the Arc Hydro data model, and the Reach name in HEC-RAS[12].

---

[12] In HEC-RAS, Reach names must be unique. If multiple line segments share a Reach name, they must either be merged to form a single line, or be given unique names such as by adding a suffix.

riverCode [0..1]: A label or identifier for a set of Edge members linearly connected to form a single river, usually defined from a headwater point to the confluence with a larger water body. If provided, this shall be unique within the scope of a ReferenceNetwork. This is analogous to the River name within in HEC-RAS.

flowDirection [1]: An integer value indicating the direction of flow. There are four values: 0 (uninitialized), 1 (with digitized), 2 (against digitized), and 3 (indeterminate). "With digitized" and "against digitized" compare the flow direction to the direction in which the line was digitized, as indicated by the order of the vertices defining each segment of the line (Maidment, 2002). In general, values of 1 or 2 should be used, with 1 being preferable. Values of 0 and 3 are provided to support the Arc Hydro data model. In any case, the phrase 'assumed flow direction' should be interpreted according to the associated *fromPoint* and *toPoint* (i.e. an assumed flow direction exists even where the *flowDirection* is uninitialized or indeterminate).

enabled [0..1]: A Boolean value indicating whether flow through this Edge is allowed (Maidment, 2002). If omitted, the value is assumed to be true. For future versions of RiverML it may be beneficial to provide a Scenario-scoped method for enabling and disabling Edge features.

4.2.6.9 FlowlineEdge

A FlowlineEdge is an OnNetworkEdge that traces water movement through streams, rivers, and water bodies (Maidment, 2002). FlowlineEdge inherits from OnNetworkEdge. See Figure 4.8.

4.2.6.10 ShorelineEdge

A ShorelineEdge is an OnNetworkEdge that forms the interface between land and water for water bodies. Shorelines include those of lakes and reservoirs, coastlines to the sea or ocean, and bank lines for wide streams or rivers that are considered areal or water body features (Maidment, 2002). ShorelineEdge inherits from OnNetworkEdge.

The Shoreline class requires further evaluation. The functionality for closed boundaries such is at least partially covered by the Reservoir class, and the concept of flow direction is ambiguous.

4.2.6.11 OffNetworkEdge

An OffNetworkEdge is a path which does not participate directly in a topological network. It can be used to specify features such as roadways, building footprints, or political boundaries. OffNetworkPoint inherits from Point. See Figure 4.8.

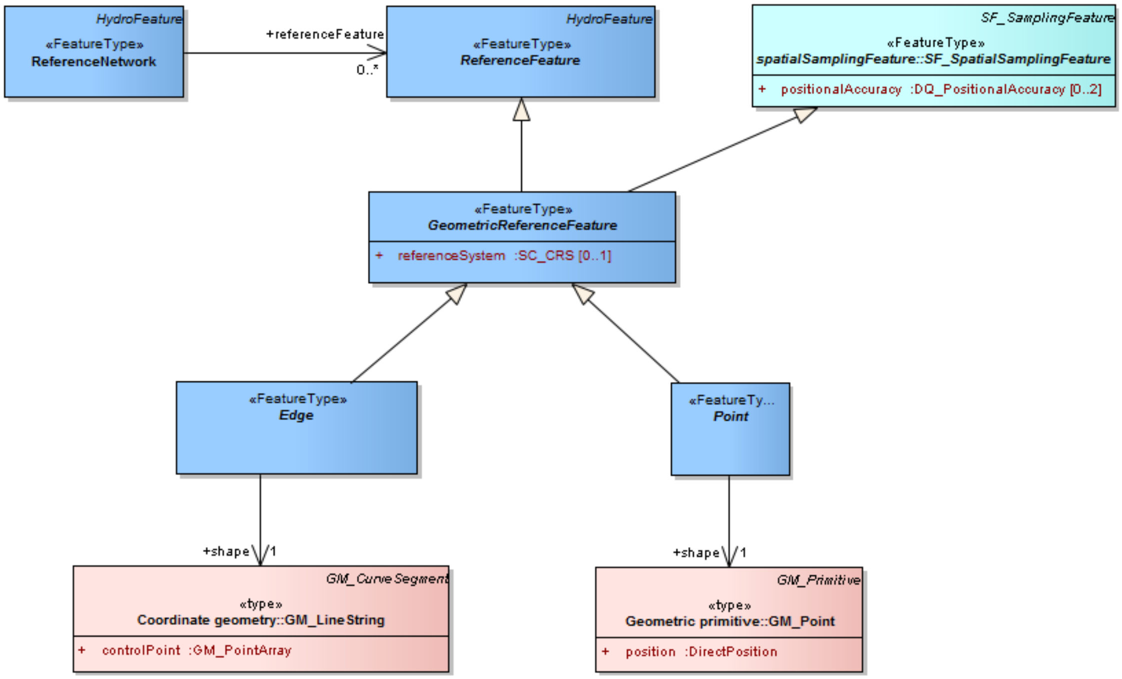


Figure 4.6 UML Diagram: GeometricReferenceFeature

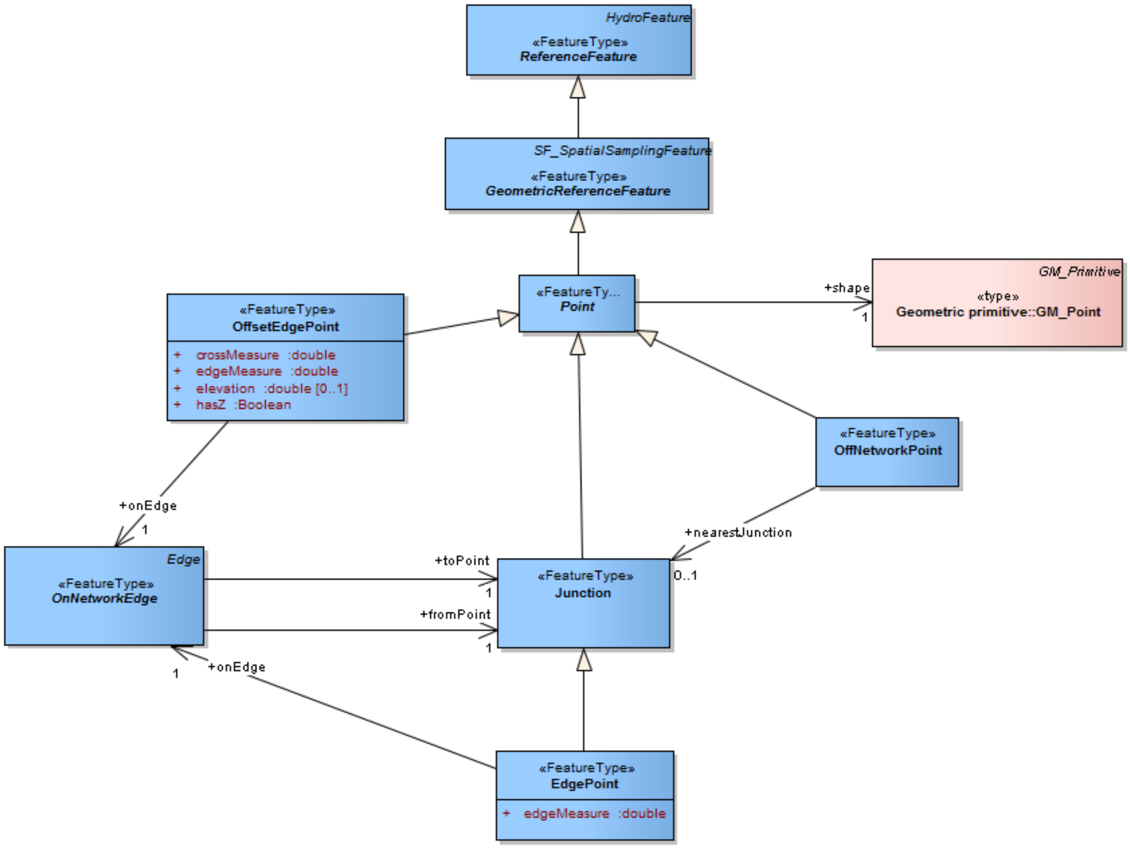


Figure 4.7 UML Diagram: Point

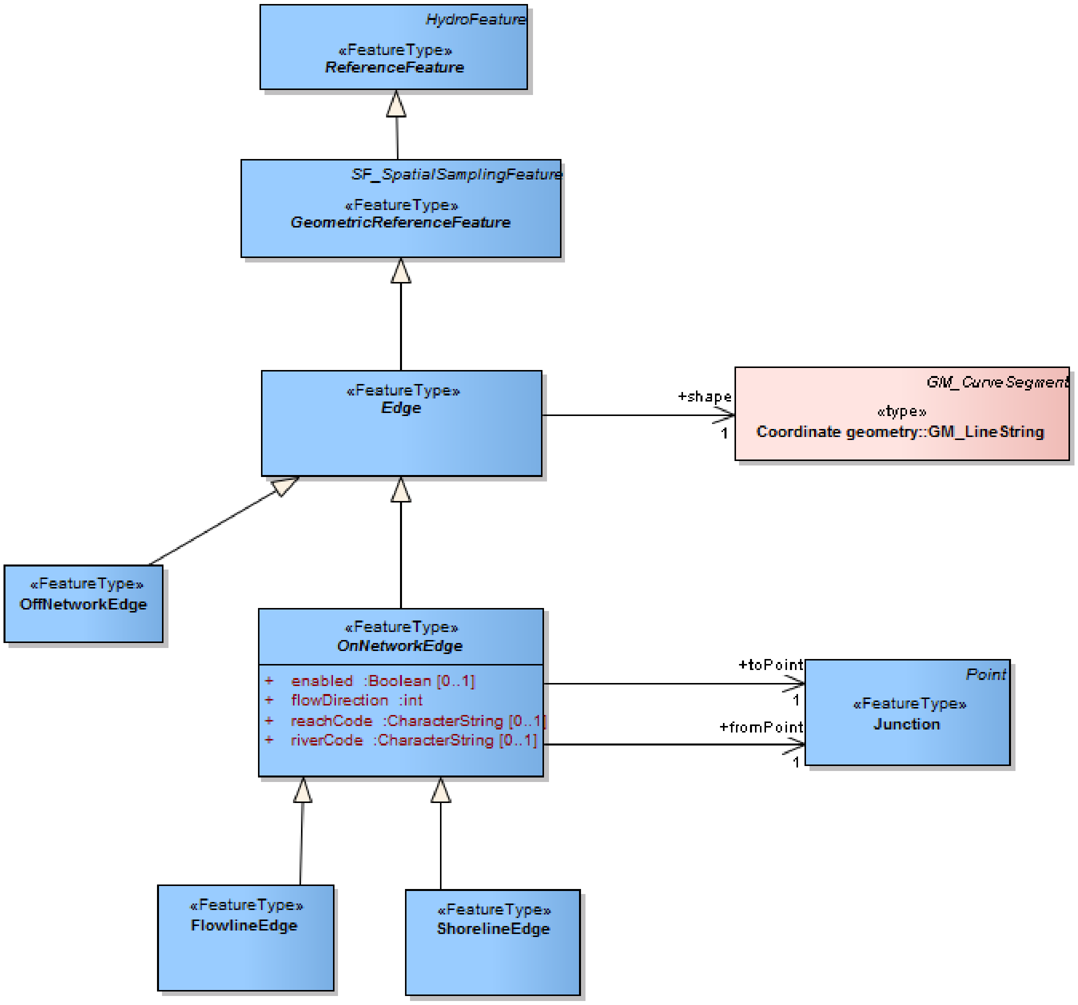


Figure 4.8 UML Diagram: Edge

4.2.7 Surface Observation

4.2.7.1 SurfaceObservation

O&M groups observations into two types based on the nature of the result: observations whose result is static (e.g. a single measurement) and observations where the result varies as some function. SurfaceObservation takes the second view and represents a set of elevation measurements as a coverage that varies in *x*, *y* space. The result of a SurfaceObservation may be a continuous function such as a DEM or DTM, or it may be a discrete function such as a LiDAR point cloud (LAS) or isolated survey points. SurfaceObservation inherits from OM_Observation. See Figure 4.9.

The SurfaceObservation has two purposes within RiverML. First, it allows the communication of detailed metadata regarding the provenance of the source DEM used to extract features such as cross sections and profile lines. Second, it allows the communication of the source DEM itself (via the Surface *shape* property). It is envisioned that in most cases of 1D models, RiverML documents will either not include the source DEMs, or include them by external reference. Where RiverML is used for coupled 1D/2D models, the increased file size resulting from including the full DEM may be considered worthwhile. The use of Surfaces and the supported file types (TIN, raster, LAS, etc.) should be evaluated for future versions of RiverML.

SurfaceObservation and related classes within RiverML 0.2 are in a preliminary development form. Additional input from the water resources community is required to identify the appropriate properties and code lists. A few of the open questions are listed here:

1. What is the appropriate *featureOfInterest* for a SurfaceObservation?
2. How should the following cases be distinguished: raw measurement data, data which has been adjusted to better represent the conditions at the time of measurements (e.g. by adding breaklines to a TIN), data which has been adjusted to represent a future or hypothetical situation? This is currently handled using an *observationType* property.
3. For future or hypothetical situations, what metadata is required, and does it belong with the SurfaceObservation or the Scenario?

4. What surface types and encoding formats should RiverML be able to directly support?
5. How should unsupported surface types and encoding formats be described?
6. What additional metadata is required, such as horizontal and vertical accuracy?
7. Are conformance classes required that would further restrict the use of Surfaces?

Properties:

metadata [0..1]: The O&M *metadata* property is restricted to type of SurfaceMetadata.

procedure [0..1]: The O&M *procedure* property is restricted to type of SurfaceObservationProcess.

result [0..1]: The O&M *result* property is restricted to type of Surface.

featureOfInterest [0..1]: In RiverML 0.2 the *featureOfInterest* is ambiguous and has no specific target type. The implied *featureOfInterest* is a portion of the surface of the earth defined by the limits of the coverage. Future versions of RiverML should refine this property.

observationType [0..1]: This takes a value of type SurfaceObservationTypeCode.

4.2.7.2 SurfaceMetadata

This should be expanded in future versions of RiverML. SurfaceMetadata inherits from MD_Metadata as defined in ISO 19115. See Figure 4.10.

4.2.7.3 SurfaceObservationProcess

This should be expanded in future versions of RiverML. SurfaceObservationProcess inherits from OM_Process as defined in O&M. See Figure 4.9.

Properties:

processType [0..1]: In future versions of RiverML, the *processType* property should be restricted to type of SurfaceObservationProcessCode. As this code list has yet to be defined, for RiverML 0.2 a generic CharacterString is used.

4.2.7.4 SurfaceObservationProcessCode

This is a placeholder class for future versions of RiverML. See Figure 4.9.

4.2.7.5 SurfaceTypeCode

A list of surface types. This should be revised and expanded in future versions of RiverML. See Figure 4.9.

4.2.7.6 SurfaceObservationTypeCode

Used to distinguish between the following cases: raw measurement data (raw), data which has been adjusted to better represent the conditions at the time of measurement such as by adding breaklines to a TIN (corrected), and data which has been adjusted to represent a future or hypothetical situation (modified). See Figure 4.9.

4.2.7.7 Surface

A Surface may be a continuous function such as a DEM or DTM, or it may be a discrete function such as a LiDAR point cloud (LAS) or isolated survey points. For RiverML 0.2, the detailed data for a Surface is provided as an optional GM_Surface through the *shape* property. It is intended that SurfaceObseravations and Surfaces be included in RiverML files to provide appropriate metadata even in cases where the *shape* property is omitted. The acceptable values for this property should be evaluated for future versions of RiverML in order to support an appropriate balance of flexibility and interoperability to fit the needs of the water resources community. Consideration should be given to the types of surface data used by 1D models for feature extraction and by 2D models for computation. Surface inherits from HydroFeature. See Figure 4.9.

Properties:

referenceSystem [0..1]: Specifies the coordinate reference system. Value shall be of type SC_CRS as defined in Lott (2010). This overrides *defaultReferenceSystem* in DocumentMetadata. In future versions of RiverML it may be advantageous to allow separate vertical and horizontal reference systems.

surfaceType [0..1]: This takes a value of type SurfaceTypeCode.

shape [0..1]: The value shall be of type GM_Surface as defined in ISO 19107. In general for 1D models the *shape* property should either be omitted or provided by external reference in order to minimize document file size.

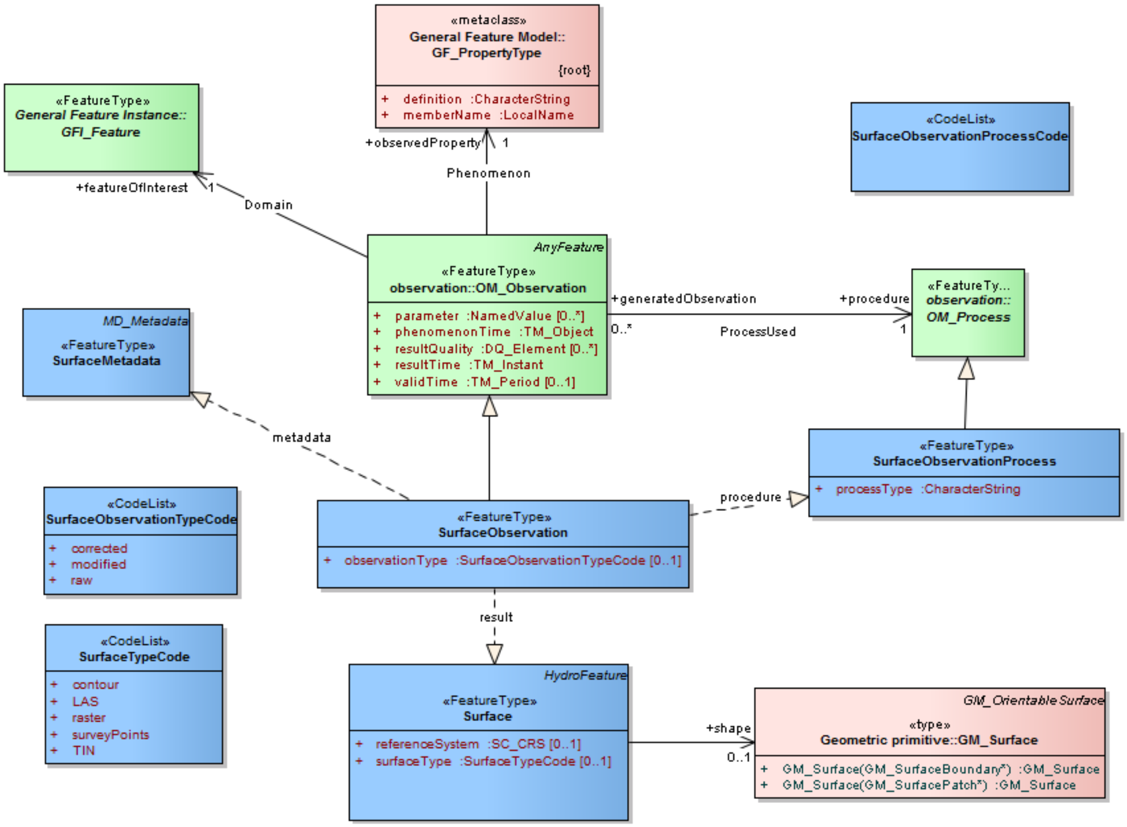


Figure 4.9 UML Diagram: SurfaceObservation

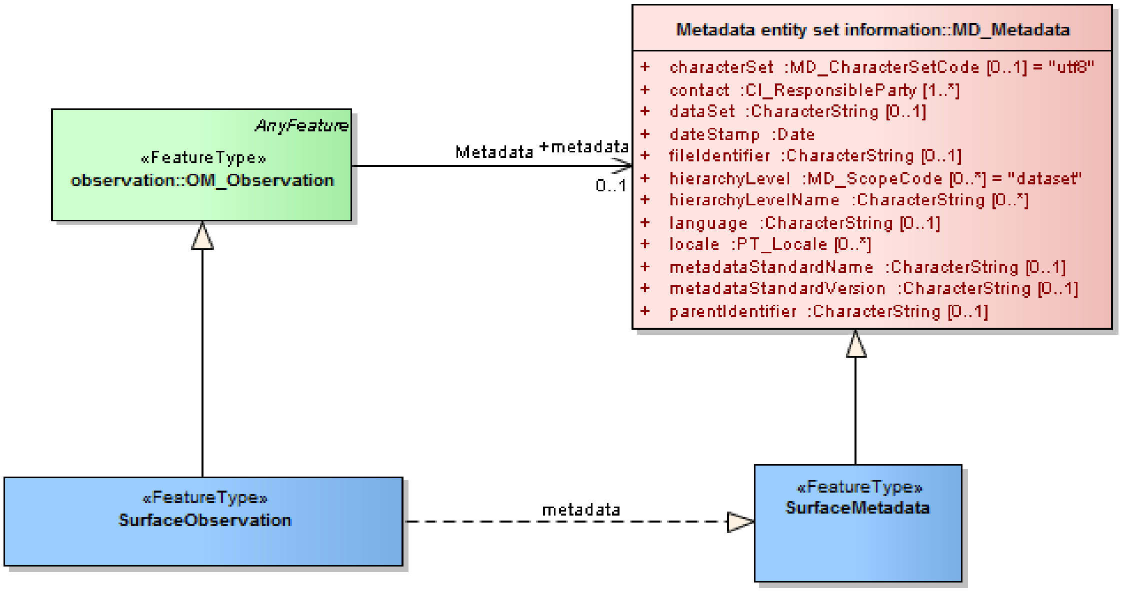


Figure 4.10 UML Diagram: SurfaceMetadata

4.2.8 Feature Observation

4.2.8.1 RiverFeatureObservation

This is an abstract class which serves as a template for observations about specific river features. The elevation of river features is typically extracted from a DEM, and thus can be thought of as an observation about an observation. In RiverML, the *featureOfInterest* for a RiverFeatureObservation is a ReferenceFeature. This allows RiverFeature members to be linked topologically and compared across different Scenarios. The SurfaceObservation describing the source DEM can optionally be referenced using the *relatedObservation* property. RiverFeatureObservation inherits from OM_Observation. See Figure 4.11.

RiverFeatureObservation and related classes within RiverML 0.2 are in a preliminary development form. Additional input from the water resources community is required to identify the appropriate properties and code lists. Further work is also required to fully integrate the RiverFeatureObservation with the OM_Observation framework. In RiverML 0.2, the result of a RiverFeatureObservation is a RiverFeature, which is a complex entity that may contain multiple geometric and tabular representations, and that combines elevation measurements with roughness measurements and other descriptive attributes. This level of complexity may not be compatible with O&M, and may not support a GML Simple Feature implementation.

Properties:

metadata [0..1]: The O&M *metadata* property is restricted to type of RiverFeatureMetadata.

procedure [0..1]: The O&M *procedure* property is restricted to type of RiverFeatureObservationProcess.

result [0..1]: The O&M *result* property is restricted to type of RiverFeature.

featureOfInterest [0..1]: The O&M *featureOfInterest* property is restricted to type of ReferenceFeature by reference to its unique *id*.

relatedObservation [0..1]: The O&M *relatedObservation* property is restricted to type of SurfaceObservation by reference to its unique *id*.

observedProperty [1]: The value of this property inherited from OM_Observation is not important for RiverML 0.2, as the *observedProperty* is implied by the specific type of RiverFeatureObservation.

4.2.8.2 RiverFeatureMetadata

This should be expanded in future versions of RiverML. RiverFeatureMetadata inherits from MD_Metadata as defined in ISO 19115. See Figure 4.12.

4.2.8.3 RiverFeatureObservationProcess

This should be expanded in future versions of RiverML. RiverFeatureObservationProcess inherits from OM_Process as defined in O&M. See Figure 4.11.

Properties:

processType [0..1]: In future versions of RiverML, the *processType* property should be restricted to type of RiverFeatureObservationProcessCode. As this code list has yet to be defined, for RiverML 0.2 a generic CharacterString is used.

4.2.8.4 RiverFeatureObservationProcessCode

This is a placeholder class for future versions of RiverML. See Figure 4.11.

4.2.8.5 RiverFeature

This is an abstract class which serves as a template for specific river features such as cross sections, profile lines, reservoirs, shorelines, and structures. In general, a RiverFeature supports both a 2D and a 3D representation. The 2D representation is a line or polygon typically used for visualization and feature extraction. The 3D representation is either a line, polygon, or table. The 2D representation will often be consistent across multiple 3D representations of different underlying DEMs, each of which requires a separate RiverFeatureObservation. RiverFeature inherits from HydroFeature. See Figure 4.11.

Properties:

hasGeometry [0..]:* This takes a value of type GeometryFeature. In RiverML 0.2 this is a conceptual association which is renamed and actualized by specific classes which inherit from RiverFeature.

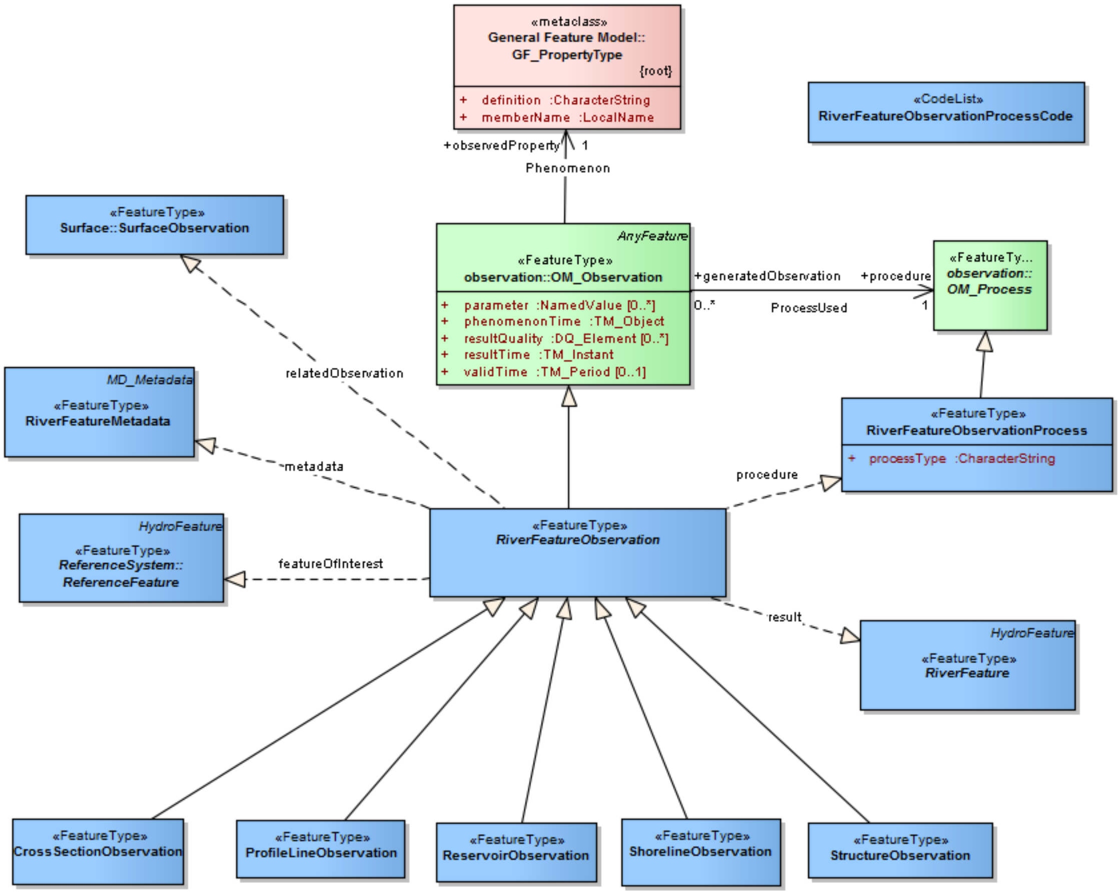


Figure 4.11 UML Diagram: RiverFeatureObservation

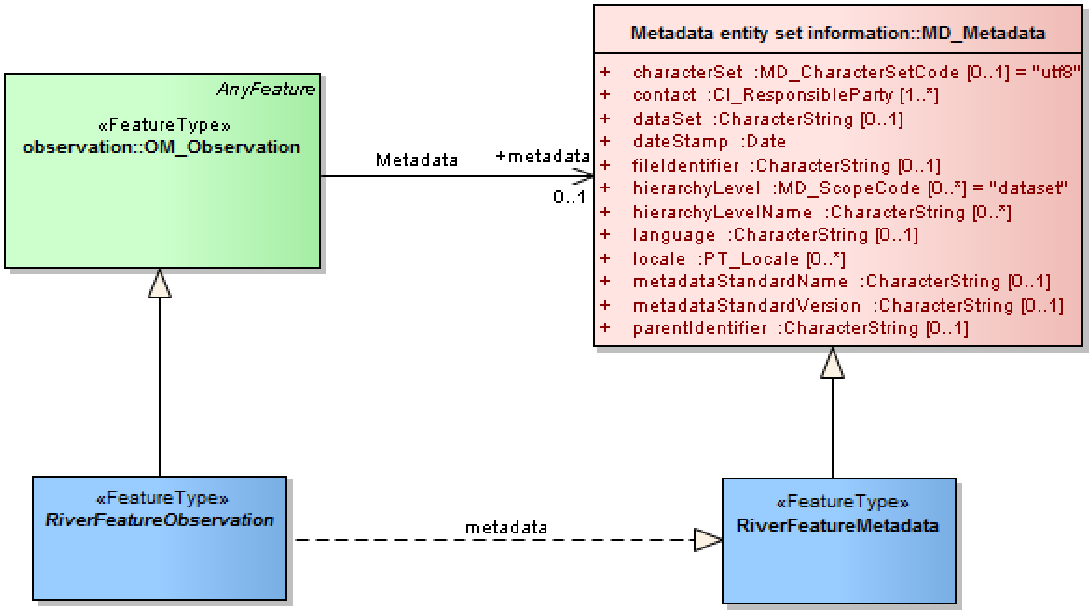


Figure 4.12 UML Diagram: RiverFeatureMetadata

4.2.9 GeometryFeature

4.2.9.1 GeometryFeature

This is an abstract class which serves as a template for geometry features. A GeometryFeature consists of a property which defines the coordinates a property which allows description of attributes which vary as a function of that geometry. The type of coordinate geometry and the descriptive attributes vary based on the type of feature, as as defined in classes which inherit from GeometryFeature. GeometryFeature inherits from HydroFeature. See Figure 4.13.

Properties:

hasProperty [0..]:* This takes a value of type GeometryProperty. In RiverML 0.2 this is a conceptual association which is renamed and actualized by specific classes which inherit from GeometryFeature.

referenceSystem [0..1]: Specifies the coordinate reference system. Value shall be of type SC_CRS as defined in Lott (2010). This overrides *defaultReferenceSystem* in DocumentMetadata. In future versions of RiverML it may be advantageous to allow separate vertical and horizontal reference systems.

4.2.9.2 LineFeature

This is an abstract class which serves as a template for features with open linear geometry in *x*, *y* geospatial coordinates. LineFeature inherits from GeometryFeature. See Figure 4.13.

Properties:

shape [1]: The value shall be of type GM_LineString as defined in ISO 19107.

event [0..]*: The value shall be of type HydroEvent. This is an actualization of the *hasProperty* property inherited from GeometryFeature.

4.2.9.3 TwoDLine

A line feature with two dimensions. TwoDLine inherits from LineFeature. See Figure 4.14.

Properties:

shape [1]: The value shall be of type GM_LineString as defined in ISO 19107, restricted to *x*, *y* coordinates.

4.2.9.4 ThreeDLine

A line feature with three dimensions. ThreeDLine inherits from LineFeature. See Figure 4.14.

Properties:

shape [1]: The value shall be of type GM_LineString as defined in ISO 19107, and allows *x*, *y*, *z* coordinates.

4.2.9.5 OpenContour

This is a special case for a ThreeDLine in which all vertices have the same elevation, and can thus be expressed in a more compact form which is useful for developing topographic maps. OpenContour inherits from ThreeDLine. See Figure 4.14.

Properties:

shape [1]: The value shall be of type GM_LineString as defined in ISO 19107, restricted to *x*, *y* coordinates.

elevation [1]: The value shall be of type double, and represents a constant *z* coordinate for all vertices listed in the *shape* property.

4.2.9.6 PolygonFeature

This is an abstract class which serves as a template for features with closed linear geometry in *x*, *y* geospatial coordinates. PolygonFeature inherits from GeometryFeature.

There are no GeometryProperties supported for PolygonFeatures in RiverML 0.2, and thus no actualization of the *hasProperty* property. Future versions of RiverML can expand on this to enable description of properties such as drainage area characteristics. See Figure 4.13.

Properties:

shape [1]: The value shall be of type GM_Ring as defined in ISO 19107.

4.2.9.7 TwoDPolygon

A closed line feature with two dimensions. TwoDPolygon inherits from PolygonFeature. See Figure 4.15.

Properties:

shape [1]: The value shall be of type GM_Ring as defined in ISO 19107, restricted to *x*, *y* coordinates.

4.2.9.8 ThreeDPolygon

A closed line feature with three dimensions. This is an abstract class in RiverML 0.2, though it may be made concrete in future versions. ThreeDPolygon inherits from PolygonFeature. See Figure 4.15.

Properties:

shape [1]: The value shall be of type GM_Ring as defined in ISO 19107, and allows *x*, *y*, *z* coordinates.

4.2.9.9 ClosedContour

This is a special case for a ThreeDPolygon in which all vertices have the same elevation, and can thus be expressed in a more compact form which is useful for developing topographic maps. ClosedContour inherits from ThreeDPolygon. See Figure 4.15.

Properties:

shape [1]: The value shall be of type GM_Ring as defined in ISO 19107, restricted to *x*, *y* coordinates.

elevation [1]: The value shall be of type double, and represents a constant *z* coordinate for all vertices listed in the *shape* property.

4.2.9.10 TableFeature

This is an abstract class which serves as a template for features with geometry in a system other than *x*, *y* geospatial coordinates. TableFeature inherits from GeometryFeature. See Figure 4.13.

The TableFeature class is conceptually similar to the ConversionTable class proposed for Part 2 of WaterML 2.0 (Taylor, 2013). The appropriate relation between these classes should be investigated for future versions of RiverML.

Properties:

point [2..]*: The points that make up the table.

coord1Uom [1]: The unit of measure for the first coordinate in the *point* property. Future versions of RiverML should use a standard code list for units of measure.

coord2Uom [1]: The unit of measure for the second coordinate in the *point* property. Future versions of RiverML should use a standard code list for units of measure.

4.2.9.11 TableTuple

A tuple represents the relationship between two values: a value of the parameter being converted from (the independent variable) and the value of parameter being converted to (the dependent variable). The TableTuple is conceptually similar to the TableTuple class in Part 2 of WaterML 2.0 (Taylor, 2013). See Figure 4.16

coord1Value [1]: The unit of measure for the first coordinate in the *pointArray* property. Future versions of RiverML should use a standard code list for units of measure.

coord2Value [1]: The unit of measure for the second coordinate in the *pointArray* property. Future versions of RiverML should use a standard code list for units of measure.

4.2.9.12 StationElevationTable

This is a TableFeature in which the coordinates represent station (distance along a 2D path) and elevation, respectively. The 2D path can either be implied, or explicitly identified. StationElevationTable inherits from TableFeature. See Figure 4.16.

Properties:

hasCutLine [0..1]: Specifies a TwoDLine by reference to its unique *id*. If provided, station measures should begin at zero for the first vertex and proceed by linear interpolation through subsequent vertices. Where a TwoDLine is specified, Station values in the StationElevationTable should not be less than zero or greater than the length of the TwoDLine.

event [0..]*: The value shall be of type HydroEvent. This is an actualization of the *hasProperty* property inherited from GeometryFeature.

elevationOffset [0..1]: The value shall be of type double. If provided, this represents a constant value to be added to all elevation values to align with the vertical datum in the applicable coordinate reference system.

4.2.9.13 ElevationVolumeTable

This is a TableFeature in which the coordinates represent elevation and volume, respectively. This is typically used to specify the volume of a reservoir as a function of elevation. ElevationVolumeTable inherits from TableFeature. See Figure 4.16.

There are no GeometryProperties supported for ElevationVolumeTables in RiverML 0.2, and thus no actualization of the *hasProperty* property.

Properties:

elevationOffset [0..1]: The value shall be of type double. If provided, this represents a constant value to be added to all elevation values to align with the vertical datum in the applicable coordinate reference system.

4.2.9.14 ElevationAreaTable

This is a TableFeature in which the coordinates represent elevation and area, respectively. This is typically used to calculate the volume of a reservoir as a function of elevation. ElevationAreaTable inherits from TableFeature. See Figure 4.16.

There are no GeometryProperties supported for ElevationAreaTable in RiverML 0.2, and thus no actualization of the *hasProperty* property.

Properties:

elevationOffset [0..1]: The value shall be of type double. If provided, this represents a constant value to be added to all elevation values to align with the vertical datum in the applicable coordinate reference system.

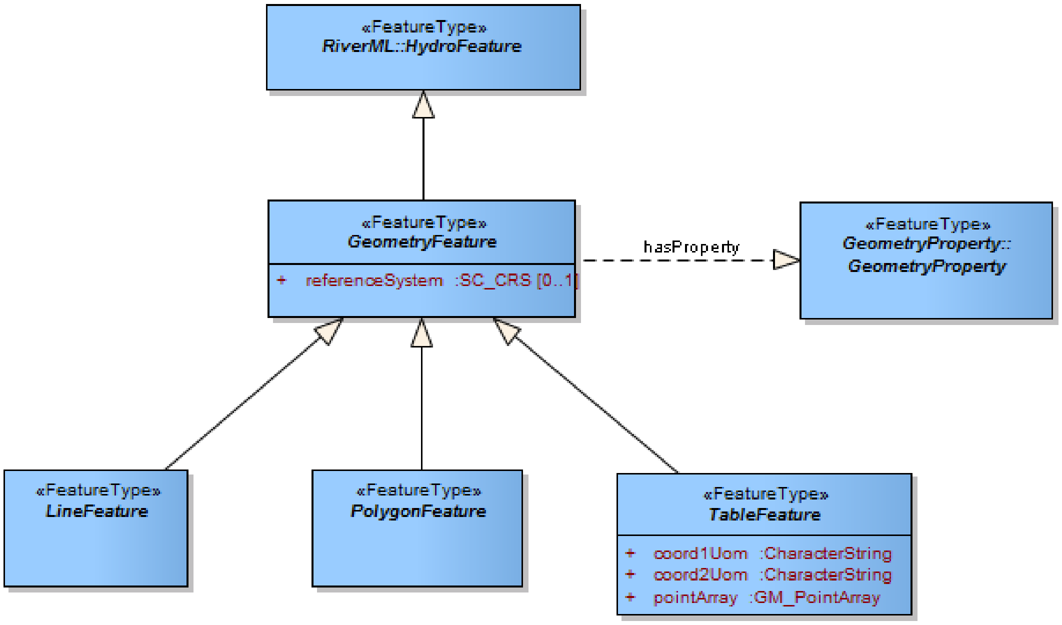


Figure 4.13 UML Diagram: GeometryFeature

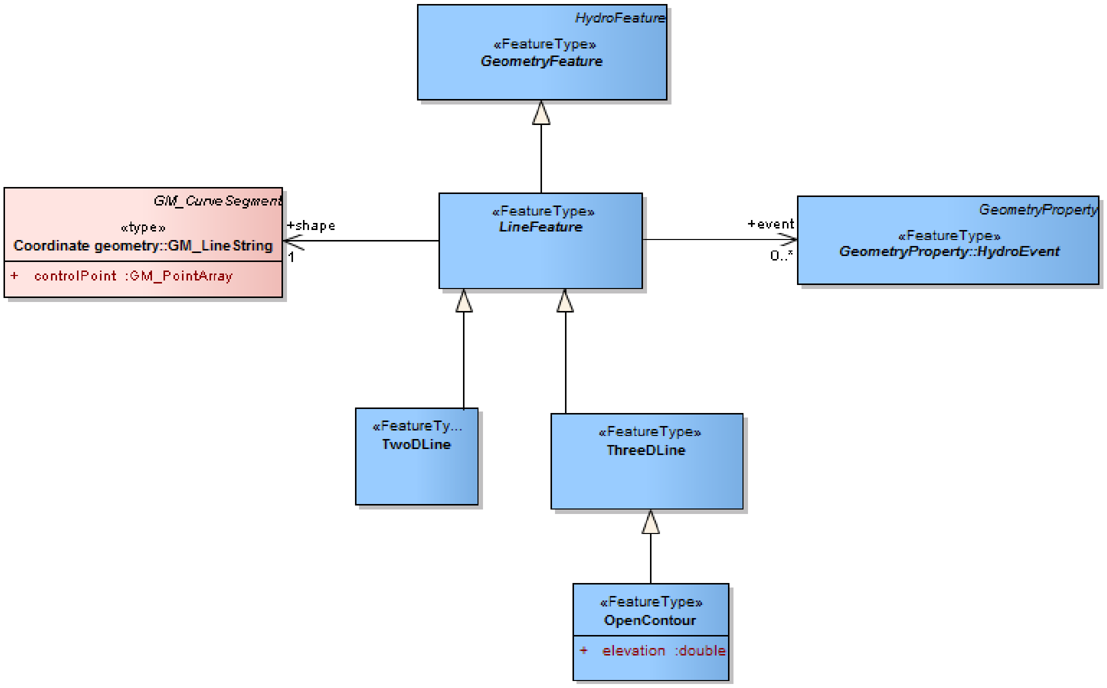


Figure 4.14 UML Diagram: LineFeature

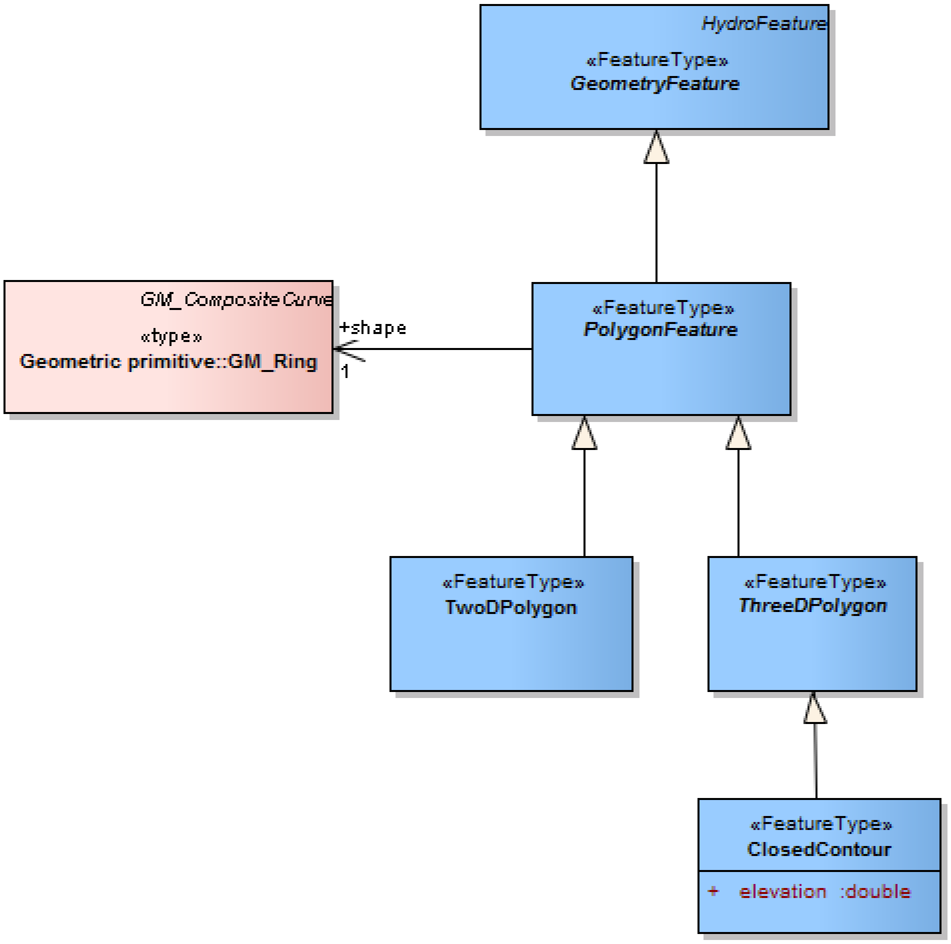


Figure 4.15 UML Diagram: PolygonFeature

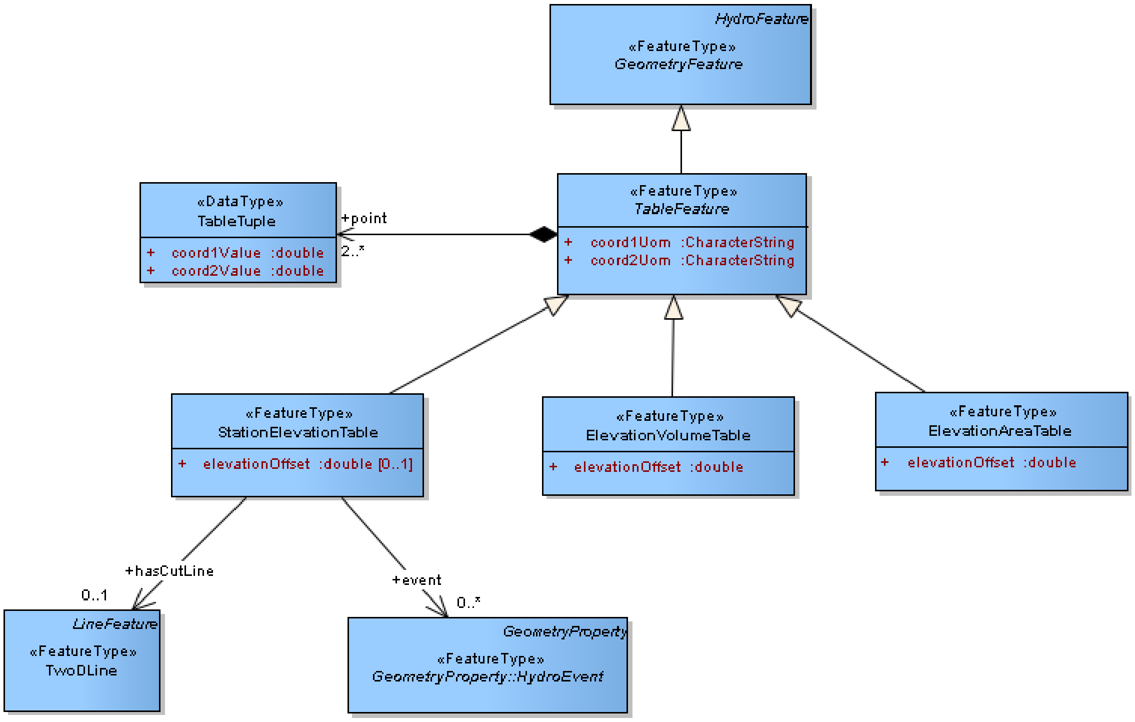


Figure 4.16 UML Diagram: TableFeature

4.2.10 GeometryProperty

4.2.10.1 GeometryProperty

This is an abstract class which serves as a template for specific geometry properties. For RiverML 0.2, all GeometryProperty classes follow the ‘event’ concept of Arc Hydro (Maidment, 2002). Future versions of RiverML may include additional methods suitable for describing properties of polygons or volumes. GeometryProperty inherits from HydroFeature. See Figure 4.17.

4.2.10.2 HydroEvent

This is an abstract class which serves as a template for geometry properties in an event style. Levees, obstructions, and the top of a road can be given a vertical offset value to indicate their height above ground level. This allows basic structures to be modeled and visualized. HydroEvent inherits from GeometryProperty. See Figure 4.17.

RiverML 0.2 uses a code list with common geometry properties used in 1D hydraulic modeling. Future versions of RiverML should allow for user-defined event descriptions, and values that can be numbers or text.

Properties:

eventType [1]: The value shall be of type EventTypeCode.

value [0..1]: The value shall be of type double. This represents the coefficient value for expansion, contraction, or roughness events, and should be omitted for all other event types.

verticalOffsetUom [0..1]: The unit of measure for the vertical offset. Future versions of RiverML should use a standard code list for units of measure.

4.2.10.3 HydroPointEvent

This is a geometry property which has a value at a distinct point. HydroPointEvent inherits from HydroEvent. See Figure 4.17.

Properties:

measure [1]: The measure along the associated GeometryFeature. Station measures should begin at zero for the first vertex and proceed by linear interpolation through subsequent vertices. The *measure* value should not be less than zero or greater than the length of the associated GeometryFeature.

verticalOffset [0..1]: The value shall be of type double. If provided, this represents the height of a geometry property above the GeometryFeature at the specified *measure*. This is applicable to levees, obstructions, and roads.

4.2.10.4 HydroLineEvent

This is a geometry property which has a constant value along a portion of a line. HydroLineEvent inherits from HydroEvent. See Figure 4.17.

Properties:

fromMeasure [1]: The start measure along the associated GeometryFeature. Station measures should begin at zero for the first vertex and proceed by linear interpolation through subsequent vertices. The *fromMeasure* value should not be less than zero or greater than the length of the associated GeometryFeature.

toMeasure [1]: The end measure along the associated GeometryFeature. Station measures should begin at zero for the first vertex and proceed by linear interpolation through subsequent vertices. The *toMeasure* value should not be less than zero or greater than the length of the associated GeometryFeature.

fromVerticalOffset [0..1]: The value shall be of type double. If provided, this represents the height of a geometry property above the GeometryFeature at the specified *fromMeasure*. This is applicable to levees, obstructions, and roads.

toVerticalOffset [0..1]: The value shall be of type double. If provided, this represents the height of a geometry property above the GeometryFeature at the specified *toMeasure*. The absolute height of the property is assumed to vary linearly from the start of the event to the end of the event[13].

---

[13] If a horizontal structure is desired and the elevation at the *fromMeasure* and *toMeasure* differ, the *fromVerticalOffset* and *toVerticalOffset* must be calculated accordingly. Future versions of RiverML should include a provision for absolute elevations in order to simplify this procedure.

4.2.10.5 EventTypeCode

A code list for types of events. For RiverML 0.2 a list of common 1D hydraulic event types is used. This list should be expanded and made customizable in future versions of RiverML. See Figure 4.17.

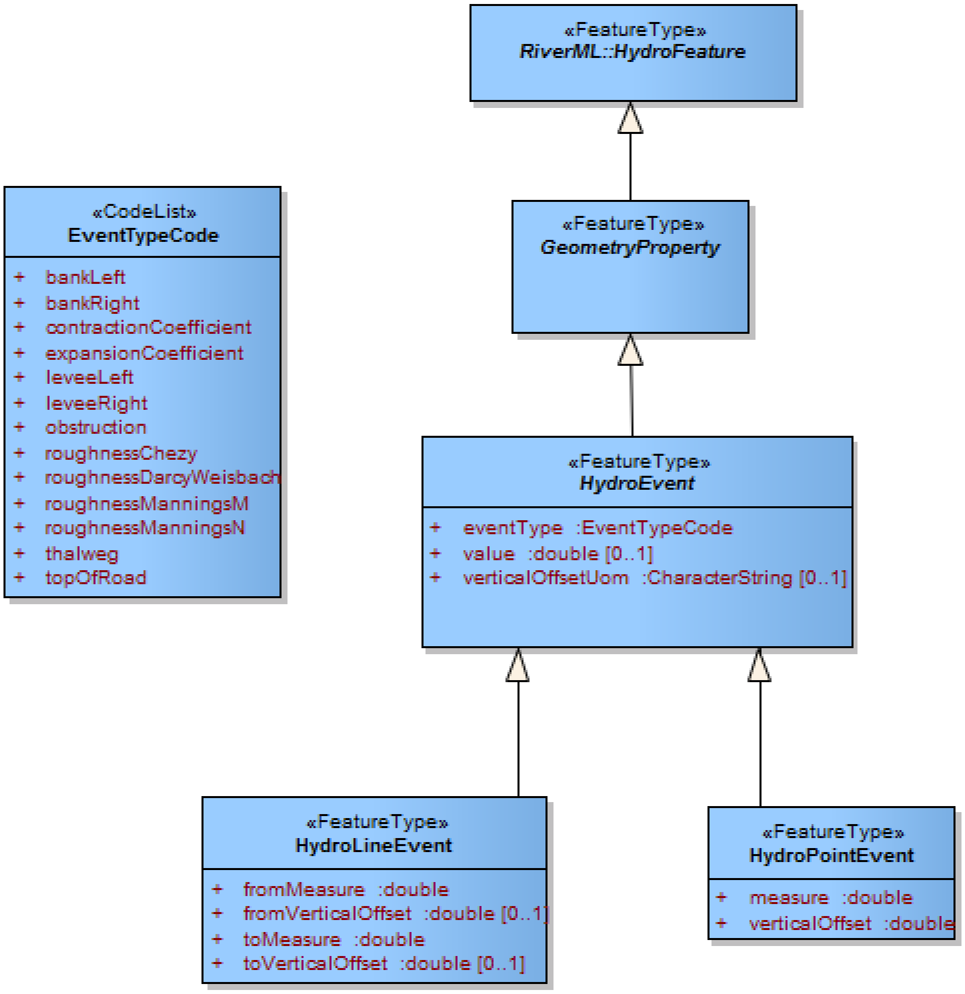


Figure 4.17 UML Diagram: GeometryProperty

4.2.11 Cross Section Observation

4.2.11.1 CrossSectionObservation

A CrossSectionObservation is a RiverFeatureObservation which returns a CrossSection as a result. CrossSectionObservation inherits from RiverFeatureObservation. See Figure 4.18.

Properties:

result [1]: The O&M *result* property is restricted to type of CrossSection.

4.2.11.2 CrossSection

A CrossSection represents the geometry of a river perpendicular to the direction of flow. CrossSection inherits from RiverFeature. See Figure 4.18.

Properties:

edgeMeasure [0..1]: A relative or absolute measure for the location along a ReferenceFeature. If the *featureOfInterest* is a Point or a Node, this property should be omitted. If the *featureOfInterest* is an Edge or a Link, this property should be included. While a Link has no explicit geometry, an *edgeMeasure* value should still be provided using any arbitrary scale to allow proper ordering of observations between two Node members.

cutLine [0..1]: The value shall be of type TwoDLine.

surfaceLine [0..1]: The value shall be of type ThreeDLine. If a value is provided for the *surfaceLine* property, a value shall not be provided for the *table* property. In general, if a value is provided for both the *cutLine* and the *ThreeDLine* properties, the two shapes should be coincident when projected into 2D space. There may be exceptions to this rule, and it is not enforced in RiverML 0.2.

table [0..1]: The value shall be of type StationElevationTable. The StationElevationTable can optionally reference the *cutLine* value. If a value is provided for the *table* property, a value should not be provided for the *surfaceLine* property.

isClosed [0..1]: The value shall be of type Boolean. If omitted, it is assumed to be false. If true, the first and last vertices of the *surfaceLine* value shall be identical. This can be used as a rudimentary method for describing pipe geometry. Future versions of RiverML should expand to a more detailed description of pipe geometry.

orientation [0..1]: The value shall be of type integer. A value of 0 indicates the vertices of the related GeometryFeatures are ordered from left to right when facing along the direction of assumed flow. A value of 1 indicates the vertices are ordered from right to left. If omitted, it is assumed to be 0.

crossSectionType [0..1]: The value shall be of type CrossSectionTypeCode. If omitted, it is assumed to be regular.

4.2.11.3 CrossSectionTypeCode

A code list for types of cross sections. In RiverML 0.2, the only types are regular and curvilinear. Curvilinear indicates a special case where a set of CrossSections and ProfileLines obtained from a curvilinear FishNet are transformed back into x, y, z coordinates. These lines create a wireframe model of the river which can be used for visualization and to accurately interpolate additional cross sections, as well as form a mesh grid which can be used for finite element analysis. See Figure 4.18.

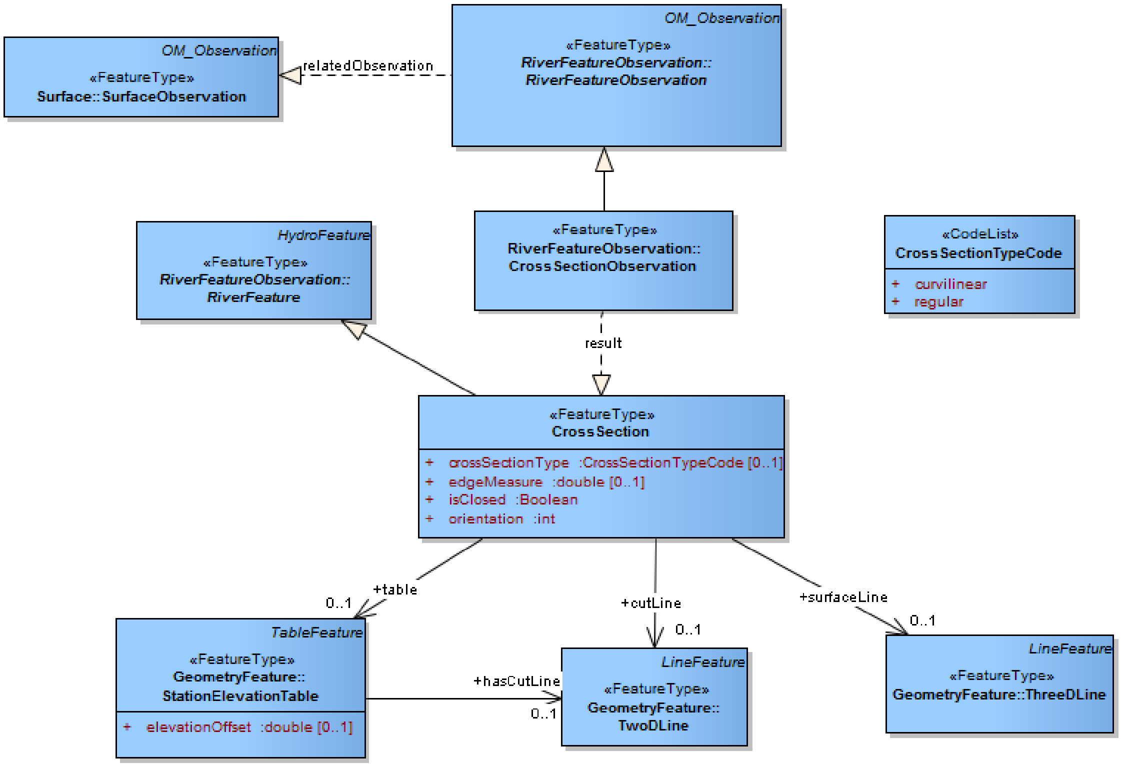


Figure 4.18 UML Diagram: CrossSectionObservation

4.2.12 Profile Line Observation

4.2.12.1 ProfileLineObservation

A ProfileLineObservation is a RiverFeatureObservation which returns a ProfileLine as a result. In general, the *featureOfInterest* for a ProfileLineObservation should either be an Edge or a Link. This is not enforced in RiverML 0.2. ProfileLineObservation inherits from RiverFeatureObservation. See Figure 4.19.

Properties:

result [1]: The O&M *result* property is restricted to type of ProfileLine.

4.2.12.2 ProfileLine

A ProfileLine represents the geometry of a river parallel to the direction of flow. ProfileLine inherits from RiverFeature. See Figure 4.19.

Properties:

profileLineType [1]: The value shall be of type ProfileLineTypeCode.

profileLineLocation [1]: The value shall be of type ProfileLineLocationCode.

4.2.12.3 ProfileLineTypeCode

A code list for types of profile lines. For RiverML 0.2 a list of common 1D types is used. This list should be expanded and made customizable in future versions of RiverML. Curvilinear indicates a special case where a set of CrossSections and ProfileLines obtained from a curvilinear FishNet are transformed back into x, y, z coordinates. These lines create a wireframe model of the river which can be used for visualization and to accurately interpolate additional cross sections, as well as form a mesh grid which can be used for finite element analysis. The value *curvilinearReference* should be used for the ProfileLine used to define a curvilinear FishNet coordinate system. See Figure 4.19.

4.2.12.4 ProfileLineLocationCode

A code list indicating the location of the profile line. Thalwegs and center lines should be assigned a value of center. All other lines should be assigned a value of left or right, based on their location when facing along the assumed direction of flow. See Figure 4.19.

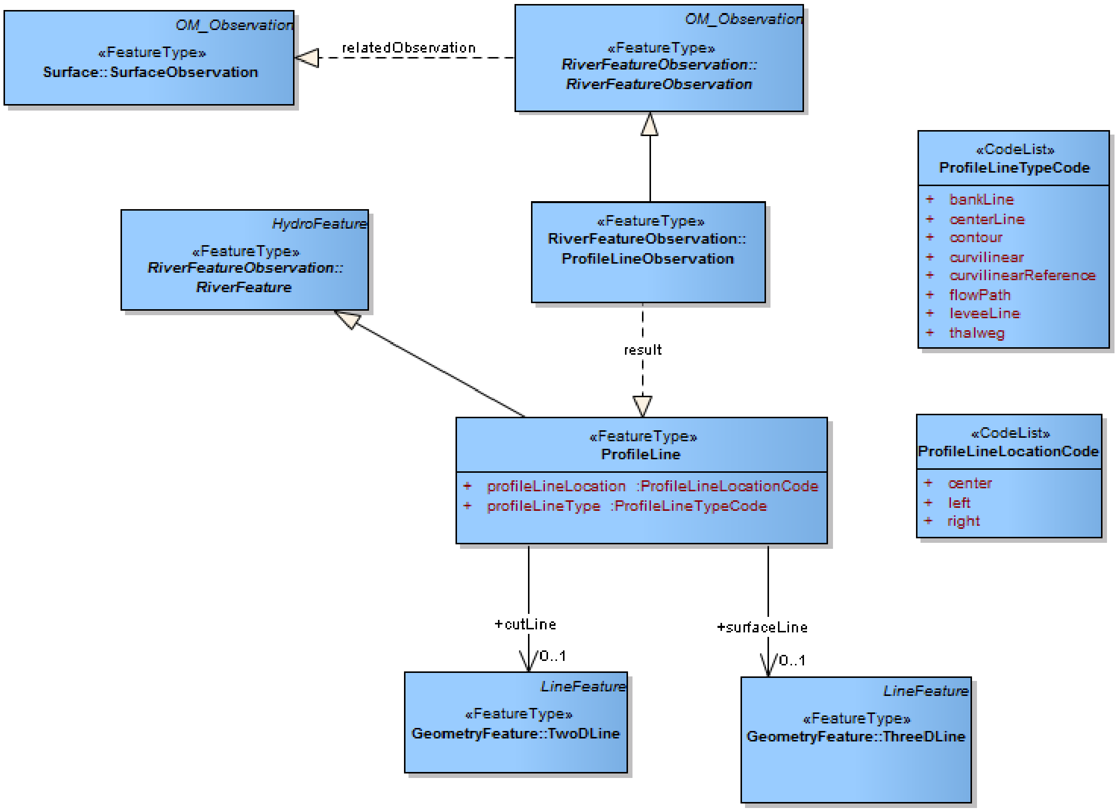


Figure 4.19 UML Diagram: ProfileLineObservation

4.2.13 Shoreline Observation

4.2.13.1 ShorelineObservation

A ShorelineObservation is a RiverFeatureObservation which returns a Shoreline as a result. ShorelineObservation inherits from RiverFeatureObservation. See Figure 4.20.

Properties:

result [1]: The O&M *result* property is restricted to type of Shoreline.

4.2.13.2 Shoreline

A Shoreline represents the geometry along an open water body. This feature class is intended primarily for coasts; closed water bodies such as lakes can be described using the Reservoir class. In addition to a cutLine and surfaceLine representing the boundary of the land-water interface, any number of OpenContour lines can be included on either side of the boundary to delineate the bathymetry. Shoreline inherits from RiverFeature. See Figure 4.20.

Properties:

cutLine [0..1]: The value shall be of type TwoDLine and represents the general cartographic boundary of the Shoreline.

surfaceLine [0..1]: The value shall be of type ThreeDLine. In general, if a value is provided for both the *cutLine* and the *surfaceLine* properties, the two shapes should be coincident when projected into 2D space. There may be exceptions to this rule, and it is not enforced in RiverML 0.2.

contour [0..]*: The value shall be of type OpenContour.

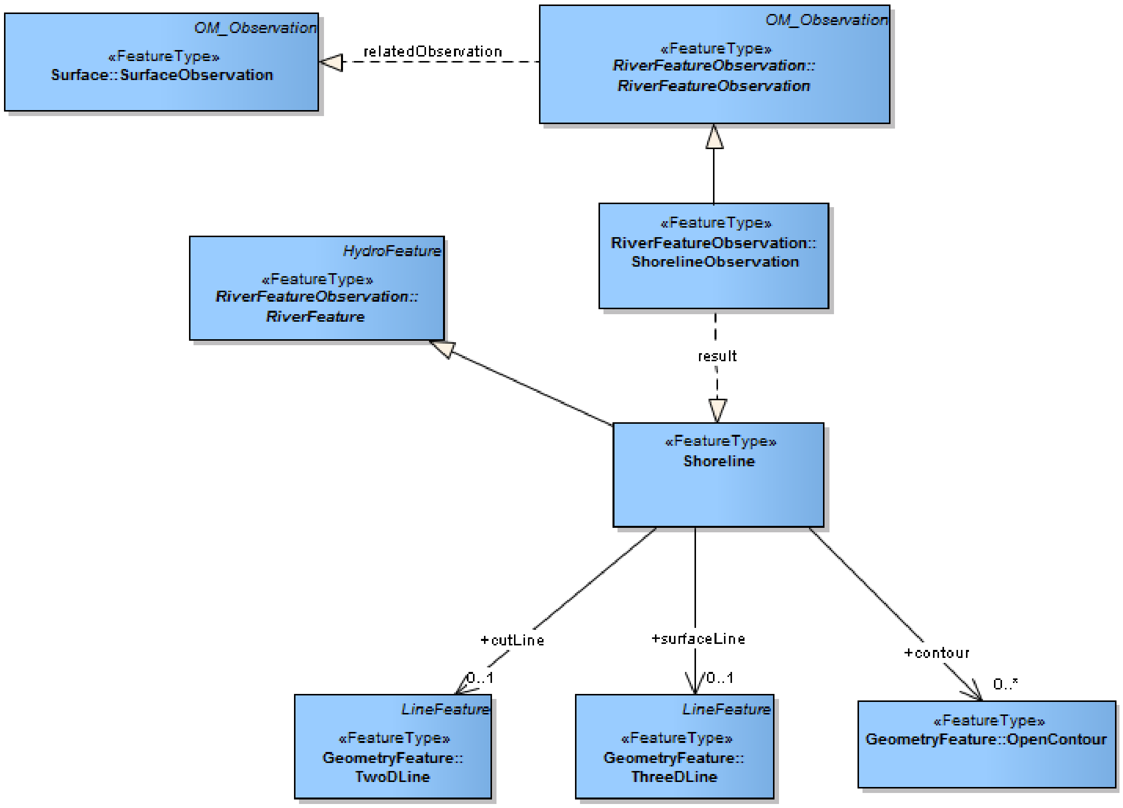


Figure 4.20 UML Diagram: ShorelineObservation

4.2.14 Reservoir Observation

4.2.14.1 ReservoirObservation

A ReservoirObservation is a RiverFeatureObservation which returns a Shoreline as a result. ReservoirObservation inherits from RiverFeatureObservation. See Figure 4.23.

Properties:

result [1]: The O&M *result* property is restricted to type of Reservoir.

4.2.14.2 Reservoir

A Reservoir is a generic class for any closed water body such as a pond, lake, or sea. The capacity of a Reservoir can either be established by a table or by a set of closed contours. If closed contours are provided, an elevation-area table can be derived by summing the polygon area of all contours with a given elevation (see Figure 4.21). Reservoir inherits from RiverFeature. See Figure 4.23.

As Reservoirs may be localized around a single ReferenceFeature or extend across multiple features, and may have multiple inlets and outlets, further work is required to determine the best practices for identifying the appropriate *featureOfInterest*. For RiverML 0.2, the following approach is used. The *featureOfInterest* for a Reservoir does not play a role in establishing network connectivity, and should be assigned to the ReferenceFeature most centrally located within the Reservoir. Any number of inlet and outlets can be assigned to establish network connectivity using the ReservoirInterface class (see Section 4.2.14.3). This approach should be tested and revised as needed for future versions of RiverML, especially in regards to interconnected Reservoirs.

Properties:

areaTable [0..1]: The value shall be of type ElevationAreaTable. If a value is provided for the *areaTable* property, a value should not be provided for the *volumeTable* or *contour* properties.

volumeTable [0..1]: The value shall be of type ElevationVolumeTable. If a value is provided for the *volumeTable* property, a value should not be provided for the *areaTable* or *contour* properties.

cutLine [0..1]: The value shall be of type TwoDLine and represents the general cartographic boundary of the Reservoir.

contour [0..]*: The value shall be of type ClosedContour. The *contour* shapes may extend beyond the boundary of the *cutLine*. If a value is provided for the *contour* property, a value should not be provided for the *areaTable* or *volumeTable* properties.

interface [0..]*: The value shall be of type ReservoirInterface.

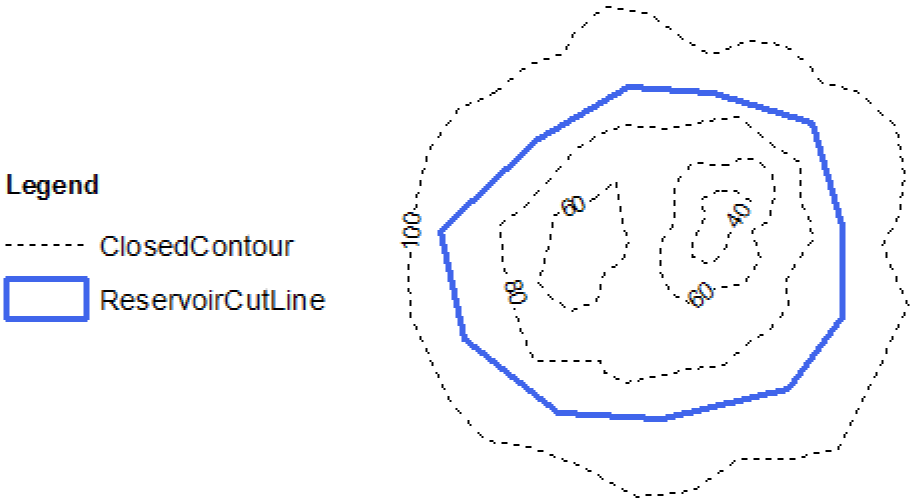


Figure 4.21 Reservoir contour example

4.2.14.3 ReservoirInterface

The ReservoirInterface class is used to establish the network connectivity of a Reservoir. Each ReservoirInterface is assigned a ReferenceFeature and defined as either an inlet or an outlet according to the direction of assumed flow. ReservoirInterface inherits from HydroFeature..

In cases where a Reservoir is centered around a single Junction or Node, the appropriate inlet and outlet features will be the connected Edges or Links. An *edgeMeasure* can be assigned to provide a river location for the interface. In cases where a Reservoir is centered around an Edge or Link with Junctions or Nodes at the boundaries, appropriate inlet and outlet features will be the Junctions or Nodes.

Figure 4.24 illustrates the use of Reservoirs. In Case 1, the *featureOfInterest* for the Reservoir is Junction J3. Three ReservoirInterfaces are defined, with *reference* to Edges E1, E2, and E3, respectively. These interfaces can optionally be assigned an *edgeMeasure*. In Case 2, the *featureOfInterest* for the Reservoir is Edge E6 which represents the path of the main river through the Reservoir. Junctions are defined at appropriate points along the boundary of the Reservoir, and serve as the *reference* for the ReservoirInterfaces. There are two inlet interfaces (J6 and J8), and two outlet interfaces (J10 and J12). The inlet at Junction J8 is directly integrated to the geometric network using Edge E7, which connects Junction J6 to EdgePoint J9 (defined as a station along Edge E6). The outlet at Junction J12 is not directly integrated with the geometric network, and thus J12, E9, and J13 are only connected to the larger river network in Scenarios where the relevant ReservoirObservation is listed as a *validObservation*. This flexibility allows the description of river networks whose connectivity varies based on the situation, such as whether a bypass spillway is in use. It is recommended that these connectivity approaches be further evaluated for future versions of RiverML.

Properties:

reservoirInterfaceType [1]: The value shall be of type ReservoirInterfaceTypeCode.

allowBackflow [0..1]: The value shall be of type Boolean. If true, flow can travel in either direction across the ReservoirInterface. If false, flow is restricted to only flow in the direction indicated by *reservoirInterfaceType*. If omitted, the value is assumed to be true.

edgeMeasure [0..1]: A relative or absolute measure for the location along a ReferenceFeature. If the *reference* is a Point or a Node, this property should be omitted. If the *reference* is an Edge or a Link, this property may be included or omitted. While a Link has no explicit geometry, an *edgeMeasure* value may still be provided using any arbitrary scale to allow proper ordering of observations.

structure [0..1]: The value shall be of type Structure by reference to its unique *id*. For RiverML 0.2, the only available Structure is a simple weir. If omitted, flow will be assumed to pass through the interface unimpeded.

reference [1]: Specifies the associated ReferenceFeature by reference to its unique *id*. If the Reservoir *featureOfInterest* is a Junction or Node, the *reference* should be an Edge or Link, and vice versa.

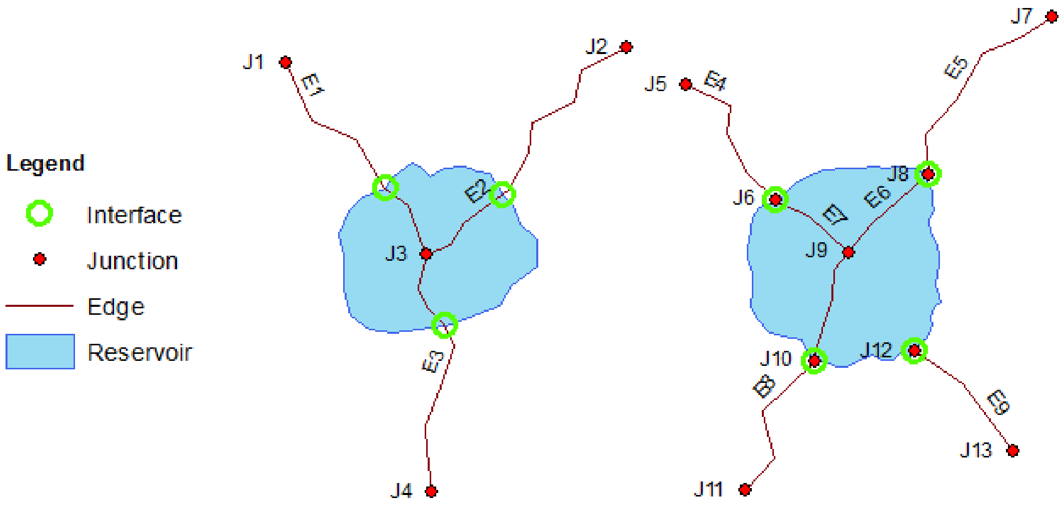


Figure 4.22 ReservoirInterface example

4.2.14.4 ReservoirInterfaceTypeCode

A code list indicating the type of interface. For RiverML 0.2 the values are inlet and outlet. See Figure 4.24.

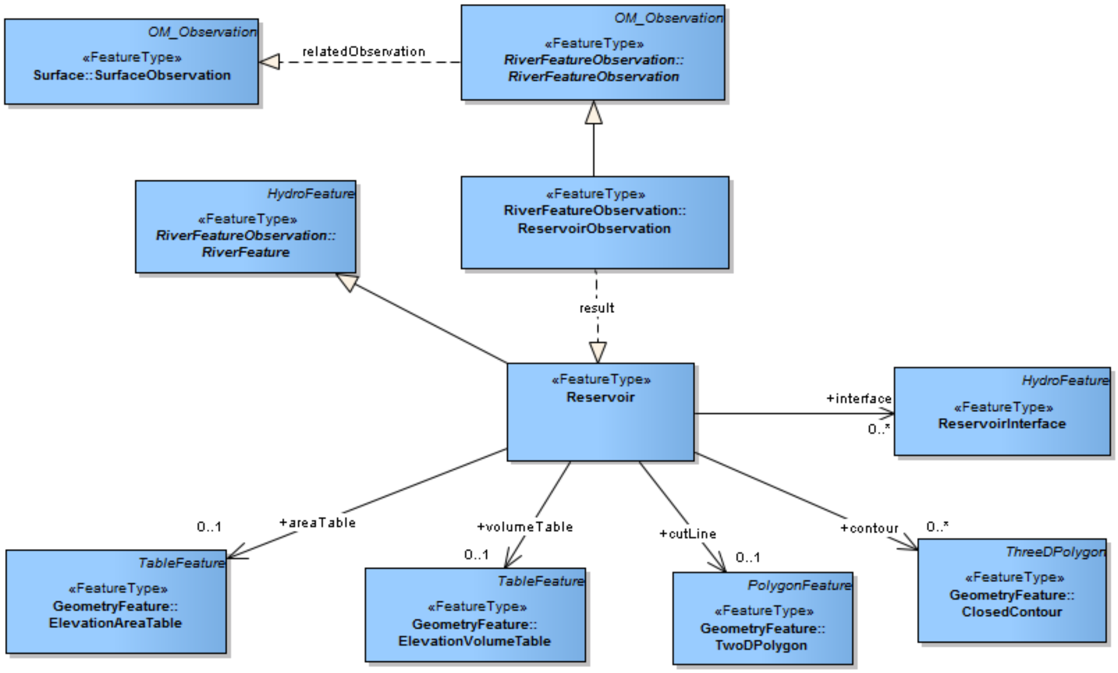


Figure 4.23 UML Diagram: ReservoirObservation

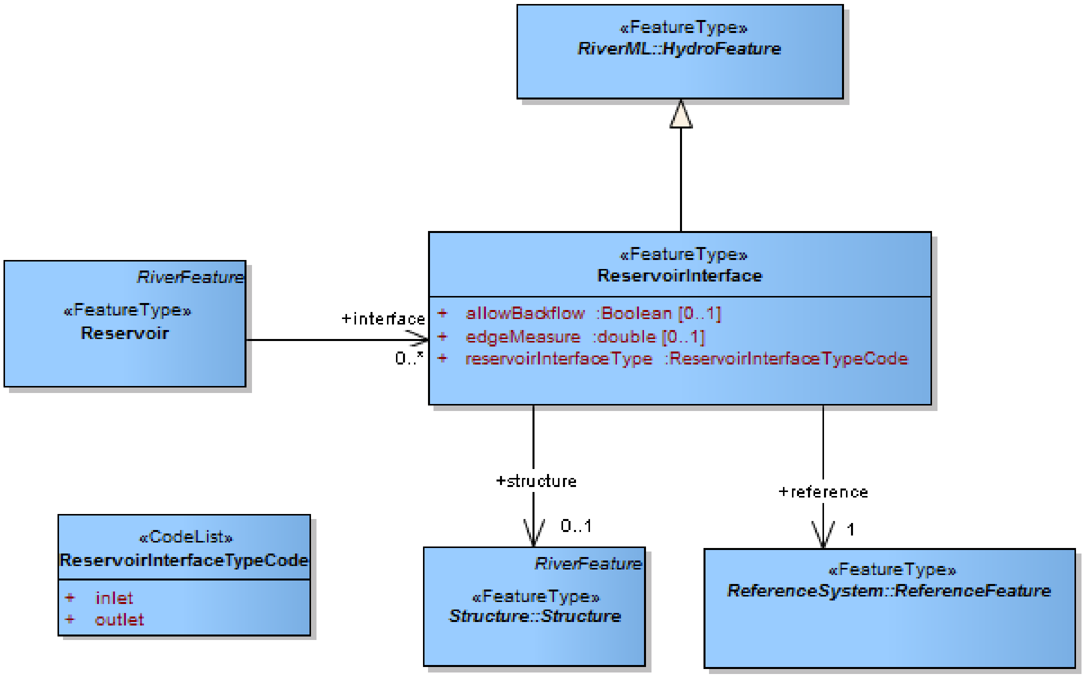


Figure 4.24 UML Diagram: ReservoirInterface

4.2.15 Structure Observation

4.2.15.1 StructureObservation

A StructureObservation is a RiverFeatureObservation which returns a Structure as a result. For RiverML 0.2, the only Structure is a simple weir. Additional Structures such as dams, bridges, and pumps should be added in future versions of RiverML. StructureObservation inherits from RiverFeatureObservation. See Figure 4.25.

Properties:

result [1]: The O&M *result* property is restricted to type of Structure.

4.2.15.2 Structure

This is an abstract class which serves as a template for specific structures. Structure inherits from RiverFeature. See Figure 4.25.

4.2.15.3 Weir

The Weir class defines a simple weir with cross sectional geometry and no width. For RiverML 0.2, weirs are only intended to be used to describe the geometry of ReservoirInterface features. Weir inherits from Structure. See Figure 4.25.

Properties:

cutLine [0..1]: The value shall be of type TwoDLine.

surfaceLine [0..1]: The value shall be of type ThreeDLine. If a value is provided for the *surfaceLine* property, a value shall not be provided for the *table* property. In general, if a value is provided for both the *cutLine* and the *ThreeDLine* properties, the two shapes should be coincident when projected into 2D space. There may be exceptions to this rule, and it is not enforced in RiverML 0.2.

table [0..1]: The value shall be of type StationElevationTable. The StationElevationTable can optionally reference the *cutLine* value. If a value is provided for the *table* property, a value should not be provided for the *surfaceLine* property.

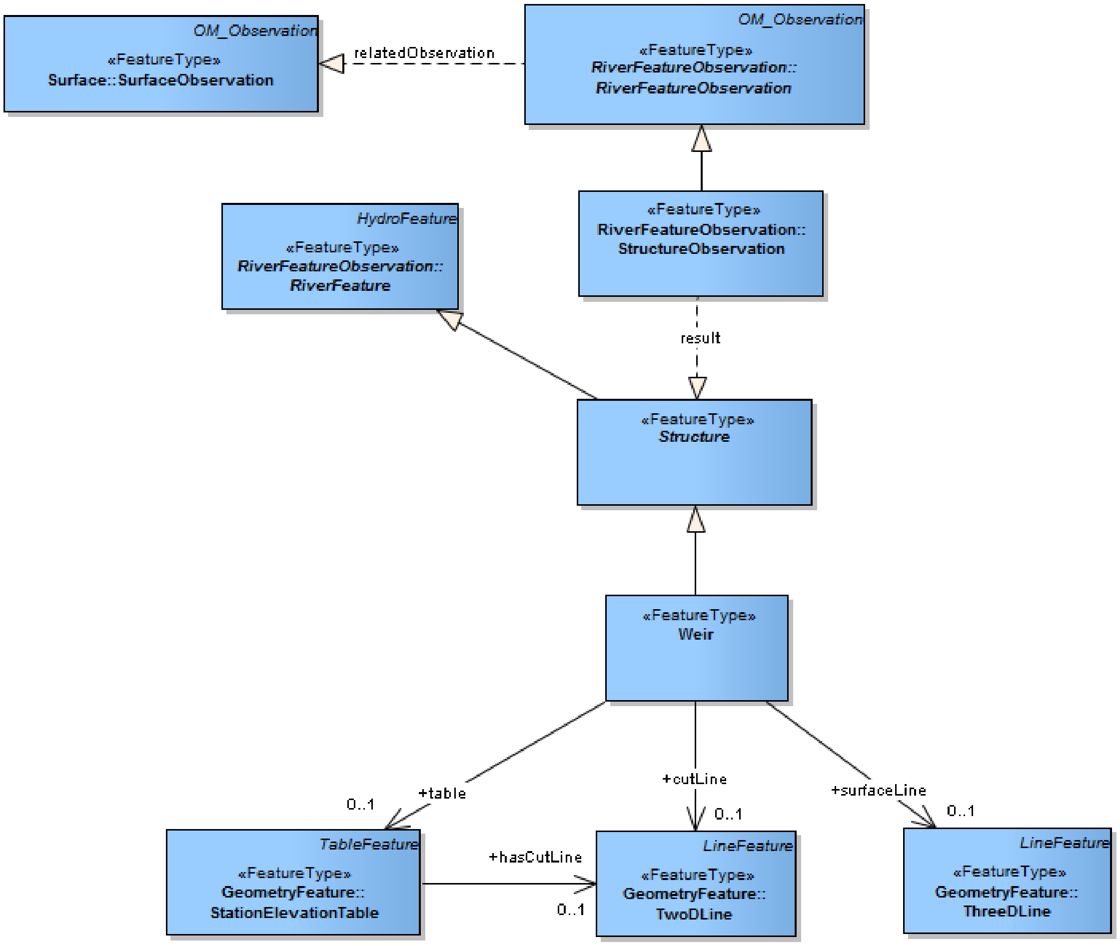


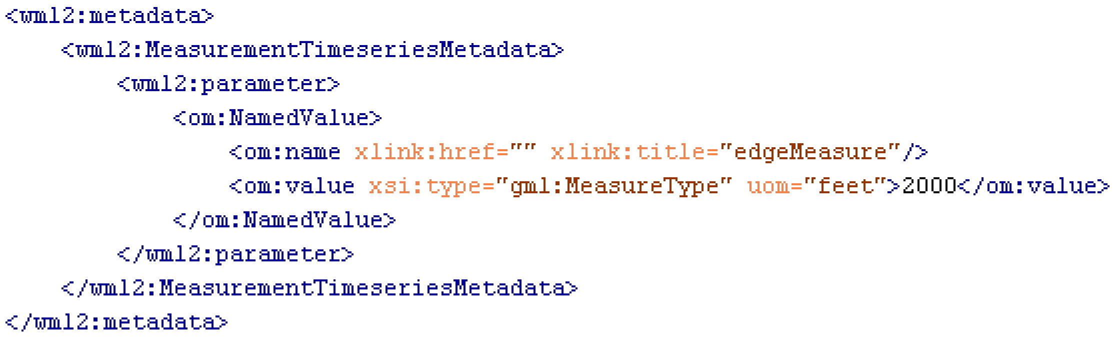
Figure 4.25 UML Diagram: StructureObservation

4.2.16 Timeseries Observation

RiverML 0.2 uses the WaterML 2.0 TimeseriesObservation for any non-geometric time varying property such as water surface elevation or flow rate. This may require a minor modification to the TimeseriesObservation definition to allow

ReferenceFeature as a valid *featureOfInterest*. Existing WaterML 2.0 services can be easily integrated into RiverML by converting MonitoringPoint members into either Point or Node members depending on whether a GM_Point is provided.

For TimeseriesObservations whose *featureOfInterest* is a Link or an Edge, a station value is required in order to associate the time series with a cross section. As WaterML 2.0 does not have a stationing property, this requires use of a soft-typed parameter (see Figure 4.26). Where a TimeseriesObservation applies to a specific station along an Edge or Link, a NamedValue parameter shall be included in the MeasurementTimeseriesMetadata. The name shall be *edgeMeasure*, and the value represents a relative or absolute measure for the location along a ReferenceFeature. While a Link has no explicit geometry, an *edgeMeasure* value should still be provided using any arbitrary scale to allow proper ordering of observations between two Node members.



```
<wml2:metadata>
    <wml2:MeasurementTimeseriesMetadata>
        <wml2:parameter>
            <om:NamedValue>
                <om:name xlink:href="" xlink:title="edgeMeasure"/>
                <om:value xsi:type="gml:MeasureType" uom="feet">2000</om:value>
            </om:NamedValue>
        </wml2:parameter>
    </wml2:MeasurementTimeseriesMetadata>
</wml2:metadata>
```

Figure 4.26 Soft-typed station values in WaterML 2.0

For 1D unsteady hydraulic modeling, time series are well suited to use as input boundary conditions and output results. For steady hydraulic modeling, boundary conditions and results are typically a single value for each model run. Model runs may be grouped by geometry (e.g. ‘Existing’ vs. ‘Proposed’), by risk factor (e.g. ’25-year, ’50-

year', '100-year'), or by some other means. While WaterML 2.0 does not explicitly support this form of non-temporal aggregation, these cases can be communicated by taking advantage of soft-typed parameters (Peter Taylor, Research Engineer, CSIRO, pers. comm.). For RiverML 0.2, the recommended procedure is to use the MeasurementTimeseriesMetadata extension to aggregate risk-based values which are valid for a particular Scenario (see Figure 4.27), and to use separate TimeseriesObservations in all cases where the geometry varies. Further work is required to ensure that steady flow values can be interpreted in a consistent fashion.

```
<wml2:metadata>
    <wml2:MeasurementTimeseriesMetadata>
        <wml2:parameter>
            <om:NamedValue>
                <om:name xlink:href="" xlink:title="25-year flow"/>
                <om:value xsi:type="gml:MeasureType" uom="cfs">75</om:value>
            </om:NamedValue>
        </wml2:parameter>
        <wml2:parameter>
            <om:NamedValue>
                <om:name xlink:href="" xlink:title="50-year flow"/>
                <om:value xsi:type="gml:MeasureType" uom="cfs">100</om:value>
            </om:NamedValue>
        </wml2:parameter>
        <wml2:parameter>
            <om:NamedValue>
                <om:name xlink:href="" xlink:title="100-year flow"/>
                <om:value xsi:type="gml:MeasureType" uom="cfs">150</om:value>
            </om:NamedValue>
        </wml2:parameter>
    </wml2:MeasurementTimeseriesMetadata>
</wml2:metadata>
```

Figure 4.27 Soft-typed steady flow values in WaterML 2.0

4.3 XML SCHEMA

The UML model described in Section 4.2 was converted into an XML Schema using the Enterprise Architect GML Extension. Certain aspects of the conceptual model are best enforced using Schematron rules rather than being encoded directly in the schema. Document validation is achieved by comparing an XML document to both the referenced schemas and the appropriate Schematron files. For RiverML 0.2, a partial set of ancillary validation files has been completed. Additional work is required to fully capture the restrictions described in the conceptual model. The schema files can be found at http://tools.crwr.utexas.edu/riverml/.

Chapter 5: Prototype Example – RiverML 0.2

5.1 SAMPLE PROJECT DESCRIPTION

In this chapter, excerpts from the file for a sample project are presented in order to clarify the use of the information model described in Chapter 4. This example file uses the XML schema described in Section 4.3. The complete example can be found along with the schema files at http://tools.crwr.utexas.edu/riverml/.

The sample project location is the upper region of the Rebecca Creek watershed near Canyon Lake in Comal County, Texas (see Figure 5.1). The region consists of three reaches in the NHDPlus as shown in Table 5.1. The goal of this project is to communicate the model inputs required to evaluate the effects of a (hypothetical) proposed cross section modification on the surrounding floodplain.

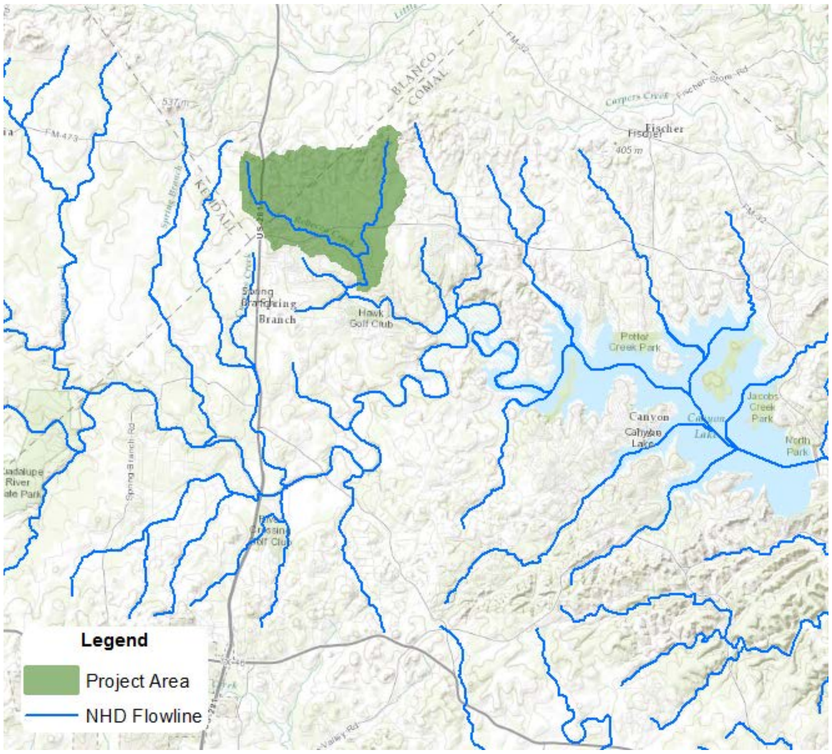


Figure 5.1 Sample Project Location

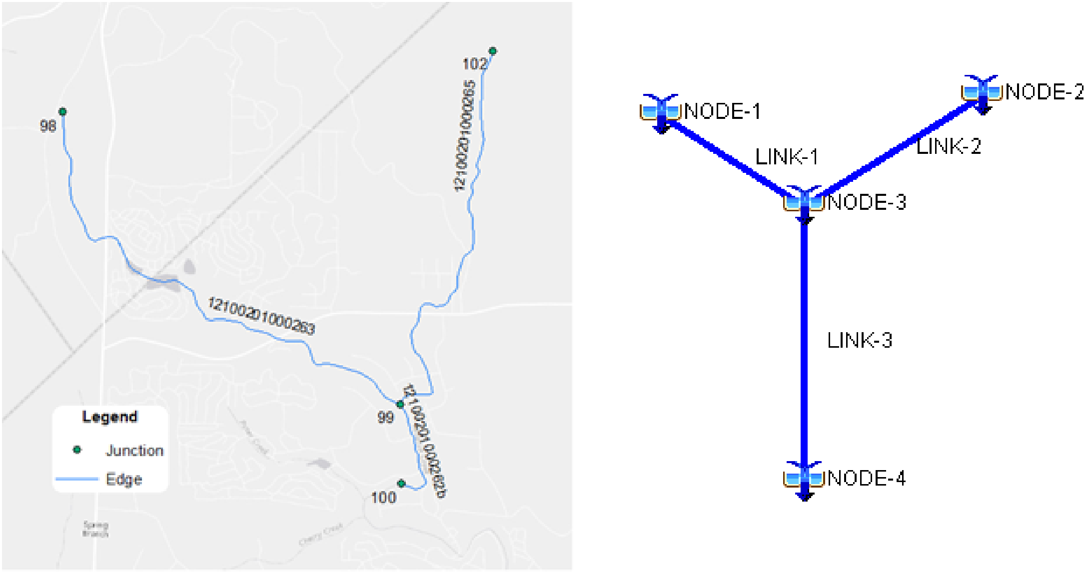
River Name	Reach Name	From Junction	To Junction
Rebecca Creek	12100201000263	98	99
Rebecca Creek	12100201000262b*	99	100
Unnamed_40*	12100201000265	102	99

**Suffix added to differentiate between multiple identically named features in NHDPlus*

Table 5.1 Rebecca Creek Reaches

5.2 SAMPLE PROJECT DATA

For this project, hydraulic river features were extracted from the 30m resolution National Elevation Dataset (NED) using the HEC-GeoRAS tools in ArcGIS. The discharge values for each reach were calculated using a hydrologic model in HEC-HMS. The hydraulic data is georeferenced, while the hydrologic data is schematic. The reference network therefore consists of a combination of geometric reference features and schematic reference features (see Figure 5.2). The geometric reference features were created by using the NHDPlus flowlines for the FlowlineEdges, and the endpoints of those flowlines for the Junctions. Schematic reference features were drawn manually in HEC-HMS. Each schematic reference feature is associated with the corresponding geometric reference feature, as shown in Table 5.2.



(a) (b)

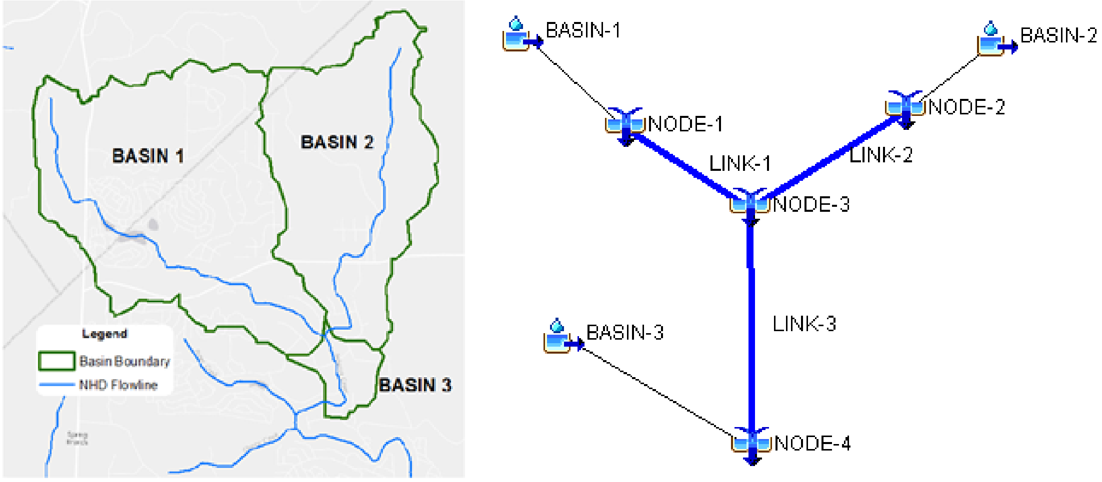
Figure 5.2 (a) Geometric Reference Features (b) Schematic Reference Features

Reference Type	Reference Name	Related Reference Feature
FlowlineEdge	12100201000263	N/A
FlowlineEdge	12100201000262b	N/A
FlowlineEdge	12100201000265	N/A
Junction	98	N/A
Junction	99	N/A
Junction	100	N/A
Junction	102	N/A
Link	LINK-1	12100201000263
Link	LINK-2	12100201000265
Link	LINK-3	12100201000262b
Node	NODE-1	98
Node	NODE-2	102

Node	NODE-3	99
Node	NODE-4	100

Table 5.2 Reference Feature Details

Within HEC-HMS, basins were attached to their appropriate nodes (see Figure 5.3). Peak discharge values at each node were computed for the 25-year, 50-year, and 100-year storm events[14]. For this project, the peak discharge at the outlet of each basin is assumed to apply to all cross sections within that basin. This is a simplifying assumption which overestimates the discharge at the upstream cross sections. If necessary, the inputs could be refined by specifying interpolated discharge values at stations along either the Links or FlowlineEdges.



(a) (b)

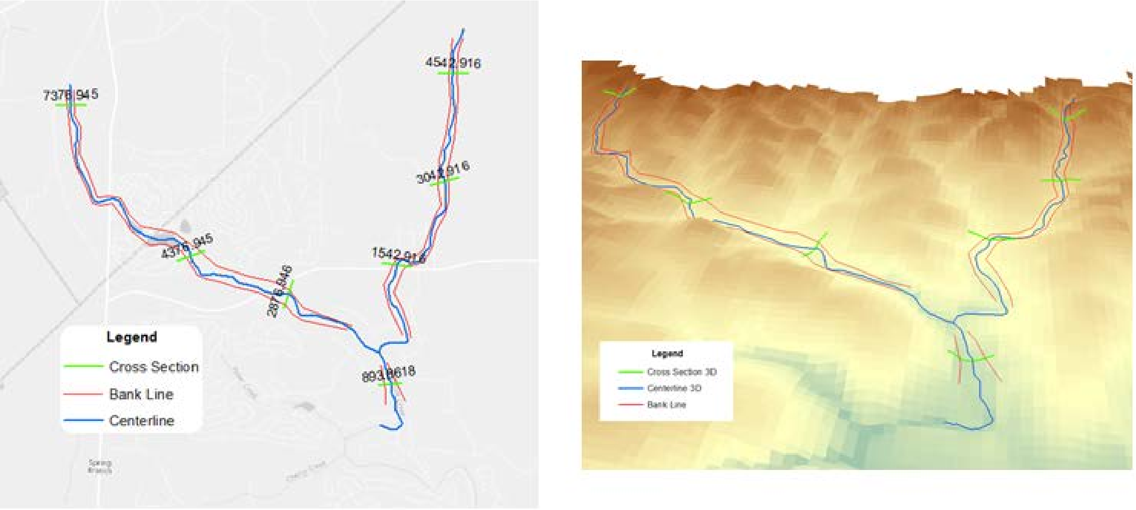
Figure 5.3 (a) Drainage Basins (b) HEC-HMS Schematic Model

Cross sections were automatically generated and assigned station values using HEC-GeoRAS with a width of 400 meters and a spacing of 1200 meters. Two different types of profile lines were created: center lines and bank lines. The NHDPlus flowlines

---

[14] The computed discharge values are estimates for demonstration purposes only.

were used as center lines. Bank lines were drawn manually using aerial imagery to estimate the extents of the channels. Using HEC-GeoRAS, elevations were extracted from the 30 m NED to create 3D lines for the cross sections and center lines (see Figure 5.4).



(a) (b)

Figure 5.4 (a) River Features Plan View (b) River Features Isometric View

The project involves two scenarios. The river features with elevations extracted from the NED, along with the discharge results of the HEC-HMS model, represent the Existing conditions. The Proposed conditions are identical, except the elevations for the cross section at Station 3042.916 on Reach 12100201000265 have been manually adjusted. This adjustment represents a hypothetical proposed excavation in order to increase the stream capacity and decrease the extents of flooding. The difference between Existing and Proposed scenarios is shown in Figure 5.5.

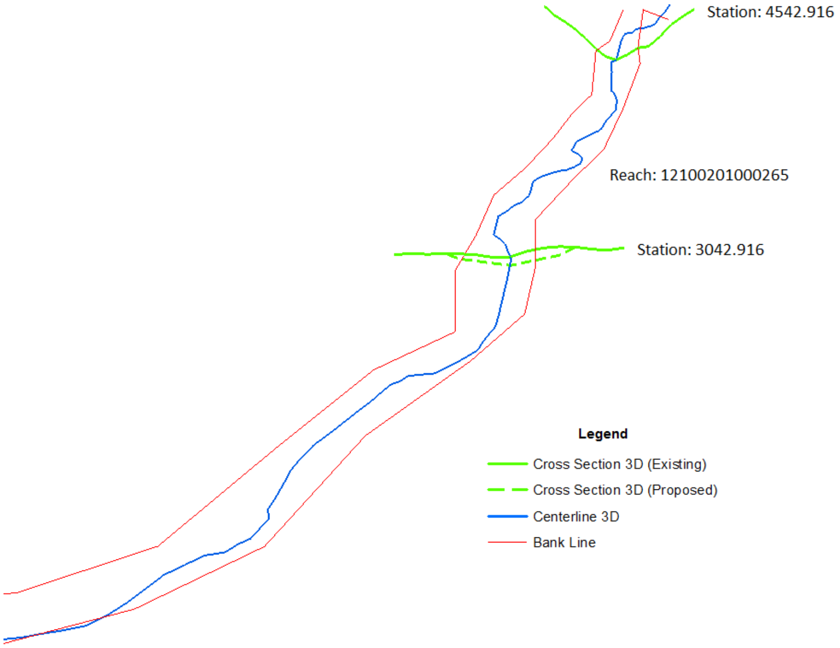


Figure 5.5 Existing Scenario vs. Proposed Scenario 3D Cross Sections

5.3 SAMPLE PROJECT RIVERML ENCODING

Excerpts from the RiverML 0.2 encoding of the Rebecca Creek project are shown below in order to illustrate how each feature class is represented. RiverML files are given a ".xml" file extension and can be created or viewed in any text editing software. OxygenXML was used for this project, which is a software application has tools for viewing and editing XML documents, as well as the ability to validate them against a specified set of schema files and Schematron rules.

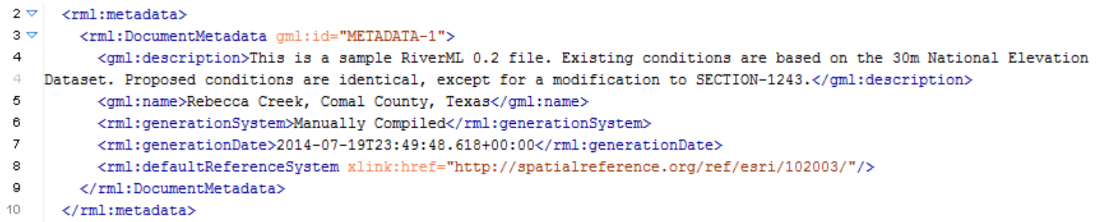
Figure 5.6 is an overview of the sample document. Line numbers (determined by line breaks in the file) are shown on the left. If an element has been collapsed, hiding the contents of the element, the number of hidden lines is shown on the right in brackets. The RiverCollection element at the root of the document identifies the URL for all schema documents required to validate the document. This project document consists of

DocumentMetadata, a set of Scenarios, a ReferenceNetwork, a set of CrossSectionObservations, a set of ProfileLineObservations, and a set of TimeseriesObservations. The file size is 86 KB, which can be compressed to 18 KB using the standard Windows zip tool.

```
<rml:RiverCollection xmlns:om="http://www.opengis.net/om/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:wml2="http://www.opengis.net/waterml/2.0" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:rml="http://tools.crwr.utexas.edu/riverml/schemas/riverml_0.2"
xsi:schemaLocation="http://tools.crwr.utexas.edu/riverml/schemas/riverml_0.2
http://tools.crwr.utexas.edu/riverml/schemas/riverml_0.2/collection.xsd" gml:id="RIVERCOLLECTION-1">
  <rml:metadata>                     [8 lines]
  <rml:scenario>                     [23 lines]
  <rml:scenario>                     [24 lines]
  <rml:referenceNetwork>             [140 lines]
  <rml:surfaceObservation>           [20 lines]
  <rml:surfaceObservation>           [13 lines]
  <rml:crossSectionObservation>      [61 lines]
  <rml:crossSectionObservation>      [61 lines]
  <rml:crossSectionObservation>      [61 lines]
  <rml:crossSectionObservation>      [61 lines]
  <rml:crossSectionObservation>      [61 lines]
  <rml:crossSectionObservation>      [61 lines]
  <rml:crossSectionObservation>      [61 lines]
  <rml:crossSectionObservation>      [61 lines]
  <rml:profileLineObservation>       [38 lines]
  <rml:profileLineObservation>       [38 lines]
  <rml:profileLineObservation>       [38 lines]
  <rml:profileLineObservation>       [23 lines]
  <rml:profileLineObservation>       [23 lines]
  <rml:profileLineObservation>       [23 lines]
  <rml:profileLineObservation>       [23 lines]
  <rml:profileLineObservation>       [23 lines]
  <rml:profileLineObservation>       [23 lines]
  <rml:timeseriesObservation>        [28 lines]
  <rml:timeseriesObservation>        [28 lines]
  <rml:timeseriesObservation>        [28 lines]
  <rml:timeseriesObservation>        [28 lines]
</rml:RiverCollection>
```

Figure 5.6 Sample XML Overview

The DocumentMetadata consists of a description, a name, the method of document generation, the date of document generation, and the default coordinate reference system (see Figure 5.7). The name and description are user-defined text values. As this file was created manually rather than with an automated export tool, the *generationSystem* value was set to “Manually Compiled.” The spatial reference used for this project was “USA Contiguous Albers Equal Area Conic,” which is described by reference using a URL.



```xml
<rml:metadata>
  <rml:DocumentMetadata gml:id="METADATA-1">
    <gml:description>This is a sample RiverML 0.2 file. Existing conditions are based on the 30m National Elevation
Dataset. Proposed conditions are identical, except for a modification to SECTION-1243.</gml:description>
    <gml:name>Rebecca Creek, Comal County, Texas</gml:name>
    <rml:generationSystem>Manually Compiled</rml:generationSystem>
    <rml:generationDate>2014-07-19T23:49:48.618+00:00</rml:generationDate>
    <rml:defaultReferenceSystem xlink:href="http://spatialreference.org/ref/esri/102003/"/>
  </rml:DocumentMetadata>
</rml:metadata>
```

Figure 5.7 Sample XML Metadata

The Scenario element consists of a name and a set of observations which are valid for that Scenario (see Figure 5.8). The observations are identified by reference to their gml:id values. For the Existing Scenario, all elevations were extracted from the NED, so only one *validSurfaceObservation* is needed. For the Proposed Scenario, one cross section was manually modified, so a second *validSurfaceObservation* is given which provides metadata for that process. The only other difference between the two Scenarios is that the Proposed Scenario substitutes “#SECTION-1243-MODIFIED” for “#SECTION-1243.” Two benefits of the Scenario encoding are highlighted here. The first is that multiple Scenarios can use the same observations, which reduces the file size for cases where differences between Scenarios are restricted to a subset of the total observations. The second benefit is that software that is parsing a large RiverML file can easily provide the user with a list of Scenarios and import the data associated with user-selected Scenarios, while ignoring any extraneous observations.

```xml
<rml:scenario>
  <rml:Scenario gml:id="SCENARIO-1">
    <gml:name>Existing Conditions</gml:name>
    <rml:validSurfaceObservation xlink:href="#SURFACE-1"/>
    <rml:validRiverFeatureObservation xlink:href="#SECTION-1242"/>
    <rml:validRiverFeatureObservation xlink:href="#SECTION-1243"/>
    <rml:validRiverFeatureObservation xlink:href="#SECTION-1244"/>
    <rml:validRiverFeatureObservation xlink:href="#SECTION-1768"/>
    <rml:validRiverFeatureObservation xlink:href="#SECTION-1769"/>
    <rml:validRiverFeatureObservation xlink:href="#SECTION-1771"/>
    <rml:validRiverFeatureObservation xlink:href="#SECTION-2500"/>
    <rml:validRiverFeatureObservation xlink:href="#CENTER-2158"/>
    <rml:validRiverFeatureObservation xlink:href="#CENTER-2161"/>
    <rml:validRiverFeatureObservation xlink:href="#CENTER-2166"/>
    <rml:validRiverFeatureObservation xlink:href="#BANK-1"/>
    <rml:validRiverFeatureObservation xlink:href="#BANK-2"/>
    <rml:validRiverFeatureObservation xlink:href="#BANK-3"/>
    <rml:validRiverFeatureObservation xlink:href="#BANK-4"/>
    <rml:validRiverFeatureObservation xlink:href="#BANK-5"/>
    <rml:validRiverFeatureObservation xlink:href="#BANK-6"/>
    <rml:validTimeseriesObservation xlink:href="#TIMESERIES-1"/>
    <rml:validTimeseriesObservation xlink:href="#TIMESERIES-2"/>
  </rml:Scenario>
</rml:scenario>
```

Figure 5.8 Sample XML Scenario

The ReferenceNetwork consists of a description and a set of Nodes, Links, Junctions and FlowlineEdges (see Figure 5.9 and Figure 5.10). For this project, there is one schematic ReferenceFeature for each geometric ReferenceFeature. In other words, the georeferenced geometric network was duplicated in schematic form. All cross sections and profile lines are associated with the geometric reference, and the timeseries are associated with the schematic network. The bridge between these two representations is the *relatedReferenceFeature* attribute of each schematic feature. It is important to note that there is no requirement that both geometric and schematic networks are used; the ideal case is to use a single geometric network for all observations within a project document.

The Nodes simply consist of a name and a related Junction. Links consist of a name, a related FlowlineEdge, the upstream and downstream Nodes, and the name of the river and reach. Junctions consist of a name and a shape, where the shape is a 2D point in the default coordinate reference system. FlowlineEdges consist of an upstream and downstream Junction, the name of the river and reach, the direction of flow relative to the ordering of vertices, and a shape. The shape is a 2D line in the default coordinate reference system.

```
  <rml:referenceNetwork>
    <rml:ReferenceNetwork gml:id="REFNET-1">
      <gml:description>This reference network is based on the NHDPlus flowlines. Where duplicate reach names exist in
NHDPlus, a unique subscript has been added.</gml:description>
      <rml:referenceFeature>
        <rml:Node gml:id="NODE-1">
          <gml:name>NODE-1</gml:name>
          <rml:relatedReferenceFeature xlink:href="#JUNCTION-98"/>
        </rml:Node>
      </rml:referenceFeature>
      <rml:referenceFeature>
      <rml:referenceFeature>
      <rml:referenceFeature>
      <rml:referenceFeature>
        <rml:Link gml:id="LINK-1">
          <gml:name>LINK-1</gml:name>
          <rml:relatedReferenceFeature xlink:href="#FLOWLINEEDGE-12100201000263"/>
          <rml:toNode xlink:href="#NODE-3"/>
          <rml:fromNode xlink:href="#NODE-1"/>
          <rml:riverCode>Rebecca Creek</rml:riverCode>
          <rml:reachCode>12100201000263</rml:reachCode>
        </rml:Link>
      </rml:referenceFeature>
```

Figure 5.9 Sample XML Schematic Reference Features: Link and Node

```xml
      <rml:referenceFeature>
        <rml:Junction gml:id="JUNCTION-102">
          <gml:name>102</gml:name>
          <rml:shape>
            <gml:Point gml:id="JUNCTION-102A">
              <gml:pos srsDimension="2">-227772.1483 -837755.1133</gml:pos>
            </gml:Point>
          </rml:shape>
        </rml:Junction>
      </rml:referenceFeature>
      <rml:referenceFeature>
        <rml:FlowlineEdge gml:id="FLOWLINEEDGE-12100201000263">
          <rml:shape>
            <gml:LineStringSegment>
              <gml:posList srsDimension="2">-232945.2401 -838495.628 -232934.197899999 -838864.8214 -232911.3858
-838967.1638 -232857.276799999 -839037.2242 -232850.2301 -839060.3058 -232849.041999999 -839113.755 -232851.717700001
-839220.5473 -232865.432 -839260.9122 -232866.7696 -839314.2972 -232855.2345 -839462.1488 -232830.262499999 -839579.8175
-232759.8572 -839708.8002 -232687.6856 -839868.3429 -232664.5448 -839957.9708 -232627.818600001 -840012.3289
-232562.609300001 -840044.4856 -232506.0776 -840017.9276 -232414.0803 -839994.7995 -232365.8683 -839996.0001
-232300.2589 -840012.9145 -232275.8309 -840051.679 -232271.3399 -840074.6948 -232263.787799999 -840077.4242 -232232.9033
-840162.1633 -232162.626399999 -840296.2198 -232151.0207 -840339.7645 -232109.037699999 -840386.6021 -231951.396400001
-840479.6037 -231802.8079 -840529.1228 -231726.741900001 -840533.5701 -231606.7619 -840508.5963 -231566.4794
-840522.3343 -231449.3127 -840609.2203 -231386.698999999 -840643.8777 -231372.1132 -840669.6708 -231365.966399999
-840728.3415 -231335.135500001 -840815.6202 -231276.149599999 -840893.4277 -231230.037599999 -840978.5313
-231156.891899999 -840998.18 -231092.934 -840979.4299 -231034.439099999 -840975.8067 -230799.509199999 -841024.9399
-230727.437999999 -841087.808 -230679.675100001 -841106.8136 -230611.407099999 -841118.6988 -230472.792400001
-841160.3378 -230392.799900001 -841210.6752 -230264.2051 -841246.9681 -230088.7685 -841238.6479 -230048.487 -841252.3789
-230001.4242 -841299.3553 -229967.6689 -841371.4366 -229965.639900001 -841391.8285 -229958.1631 -841397.1142
-229929.925799999 -841486.8671 -229902.849300001 -841520.6143 -229875.185799999 -841531.4825 -229775.7039 -841513.6218
-229692.343599999 -841530.9766 -229369.8003 -841630.63 -229232.056700001 -841707.8556 -229189.932399999 -841749.6205
-229155.9976 -841814.0736 -229097.013 -841891.8799 -229072.331499999 -841920.4736 -228947.100199999 -841989.7528
-228881.5668 -842009.2073</gml:posList>
            </gml:LineStringSegment>
          </rml:shape>
          <rml:fromPoint xlink:href="#JUNCTION-98"/>
          <rml:toPoint xlink:href="#JUNCTION-99"/>
          <rml:riverCode>Rebecca Creek</rml:riverCode>
          <rml:reachCode>12100201000263</rml:reachCode>
          <rml:flowDirection>1</rml:flowDirection>
        </rml:FlowlineEdge>
      </rml:referenceFeature>
```

Figure 5.10 Sample XML Geometric Reference Features: Junction and FlowlineEdge

In general, there are two uses for the SurfaceObservation class (see Figure 5.11). The first is to provide metadata describing the source of elevation data found in river features. The second is to digitally encode the surface itself or provide a link to such digital encoding. Both uses are optional, but can enhance the ability of the RiverML recipient to properly interpret and expand on the data provided. For the NED Surface, the *phenomenonTime* is given by the date range of the USGS survey. A textual description of the procedure used is given, though the appropriate use of the procedure element should be refined by the OGC/WMO Hydrology Domain Working Group. The

SurfaceObservation result is a surface which is given a name, type, and a URL to the source raster file.

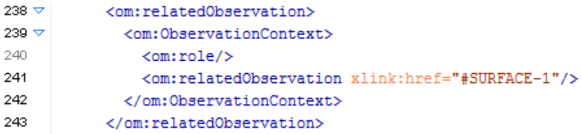
```xml
<rml:surfaceObservation>
  <rml:SurfaceObservation gml:id="SURFACE-1">
    <om:phenomenonTime>
      <gml:TimePeriod gml:id="SURFACE-1A">
        <gml:beginPosition>1962</gml:beginPosition>
        <gml:endPosition>1963</gml:endPosition>
      </gml:TimePeriod>
    </om:phenomenonTime>
    <om:resultTime/>
    <om:procedure xlink:title="Clipped NED Snapshot elev_cm raster to project boundary and converted from centimeters
to meters."/>
    <om:observedProperty xlink:title="Elevation"/>
    <om:featureOfInterest xlink:href=""/>
    <om:result>
      <rml:Surface gml:id="SURFACE-1B">
        <gml:name>National Elevation Dataset (NED) Region 12d [clipped]</gml:name>
        <rml:shape
xlink:href="http://www.horizon-systems.com/NHDPlusData/NHDPlusV21/Data/NHDPlusTX/NHDPlusV21_TX_12_12d_NEDSnapshot_01.7z"
/>
        <rml:surfaceType>raster</rml:surfaceType>
      </rml:Surface>
    </om:result>
  </rml:SurfaceObservation>
</rml:surfaceObservation>
```

Figure 5.11 Sample XML Surface Observation

An overview of a CrossSectionObservation is presented in Figure 5.12. Each CrossSectionObservation can specify a *relatedObservation*, which associates it with the appropriate SurfaceObservation (see Figure 5.13). The *phenomenonTime* can be left empty, as that information is conveyed by the SurfaceObservation. The network connectivity of the cross section is determined by the *featureOfInterest* and the *edgeMeasure* attribute. The *featureOfInterest* specifies a FlowlineEdge from the ReferenceNetwork, while the *edgeMeasure* gives a station along that feature. For this project, absolute measures rather than relative measures are used for stationing. Each cross section line has a 2D *cutLine* and a 3D *surfaceLine*. The *surfaceLine* consists only of the shape (see Figure 5.14). The *cutLine* has a shape as well as a set of events which describe the Mannings N value across the length of the line (see Figure 5.15). Each event consists of the type, the value, and the start and end measures. These measures are given as fractions of the total line length.

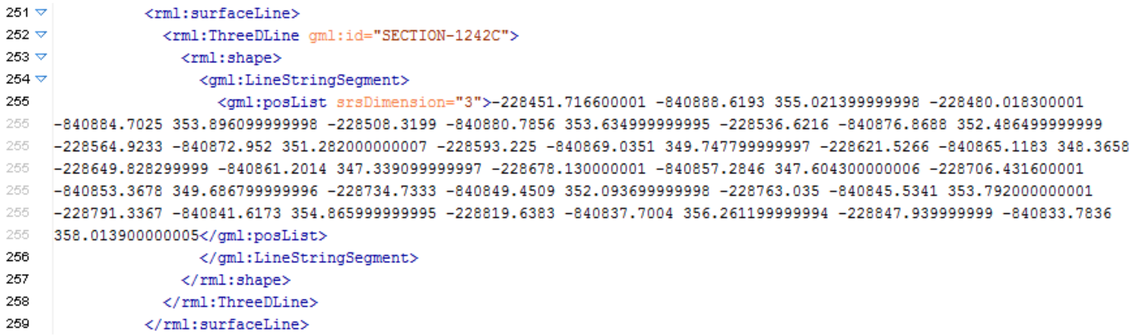
```
  <rml:CrossSectionObservation gml:id="SECTION-1242">
    <om:relatedObservation>
    <om:phenomenonTime/>
    <om:resultTime/>
    <om:procedure xlink:title="Sections automatically generated by HEC-GeoRas. Elevations extracted from NED raster."/>
    <om:observedProperty xlink:title="CrossSection"/>
    <om:featureOfInterest xlink:href="#FLOWLINEEDGE-12100201000265"/>
    <om:result>
      <rml:CrossSection gml:id="SECTION-1242A">
        <rml:surfaceLine>
          <rml:ThreeDLine gml:id="SECTION-1242C">
            <rml:shape>
              <gml:LineStringSegment>
            </rml:shape>
          </rml:ThreeDLine>
        </rml:surfaceLine>
        <rml:cutLine>
          <rml:TwoDLine gml:id="SECTION-1242B">
            <rml:shape>
            <rml:event>
            <rml:event>
            <rml:event>
          </rml:TwoDLine>
        </rml:cutLine>
        <rml:edgeMeasure>1542.92</rml:edgeMeasure>
      </rml:CrossSection>
    </om:result>
  </rml:CrossSectionObservation>
</rml:crossSectionObservation>
```

Figure 5.12 Sample XML Cross Section Observation Overview



```
<om:relatedObservation>
  <om:ObservationContext>
    <om:role/>
    <om:relatedObservation xlink:href="#SURFACE-1"/>
  </om:ObservationContext>
</om:relatedObservation>
```

Figure 5.13 Sample XML Cross Section Related Observation



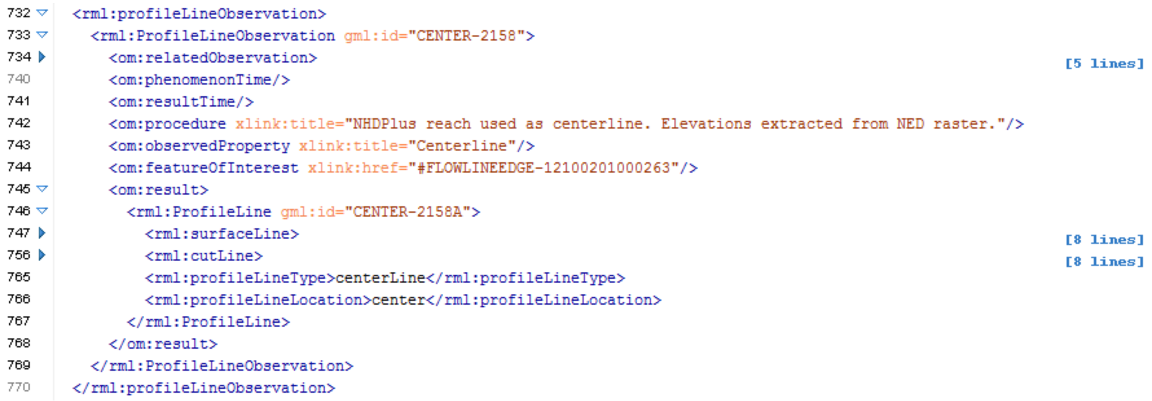
```
        <rml:surfaceLine>
          <rml:ThreeDLine gml:id="SECTION-1242C">
            <rml:shape>
              <gml:LineStringSegment>
                <gml:posList srsDimension="3">-228451.716600001 -840888.6193 355.021399999998 -228480.018300001
-840884.7025 353.896099999998 -228508.3199 -840880.7856 353.634999999995 -228536.6216 -840876.8688 352.486499999999
-228564.9233 -840872.952 351.282000000007 -228593.225 -840869.0351 349.747799999997 -228621.5266 -840865.1183 348.3658
-228649.828299999 -840861.2014 347.339099999997 -228678.130000001 -840857.2846 347.604300000006 -228706.431600001
-840853.3678 349.686799999996 -228734.7333 -840849.4509 352.093699999998 -228763.035 -840845.5341 353.792000000001
-228791.3367 -840841.6173 354.865999999995 -228819.6383 -840837.7004 356.261199999994 -228847.939999999 -840833.7836
358.013900000005</gml:posList>
              </gml:LineStringSegment>
            </rml:shape>
          </rml:ThreeDLine>
        </rml:surfaceLine>
```

Figure 5.14 Sample XML Cross Section Surface Line

```
        <rml:cutLine>
          <rml:TwoDLine gml:id="SECTION-1242B">
            <rml:shape>
              <gml:LineStringSegment>
                <gml:posList srsDimension="2">-228451.716600001 -840888.6193 -228847.939999999
-840833.7836</gml:posList>
              </gml:LineStringSegment>
            </rml:shape>
            <rml:event>
              <rml:HydroLineEvent gml:id="SECTION-1242D">
                <rml:value>0.05</rml:value>
                <rml:eventType>roughnessManningsN</rml:eventType>
                <rml:toMeasure>0.4</rml:toMeasure>
                <rml:fromMeasure>0</rml:fromMeasure>
              </rml:HydroLineEvent>
            </rml:event>
            <rml:event>                                                                            [7 lines]
            <rml:event>                                                                            [7 lines]
          </rml:TwoDLine>
        </rml:cutLine>
```

Figure 5.15 Sample XML Cross Section Cut Line

ProfileLines are similar to CrossSections, except no events were included in the *cutLine* (see Figure 5.16). Each ProfileLine also contains a type and location attribute. For center lines, the location is “center.” For bank lines, the location is either left or right, depending on the side of the river when facing downstream.



```xml
<rml:profileLineObservation>
  <rml:ProfileLineObservation gml:id="CENTER-2158">
    <om:relatedObservation>
    <om:phenomenonTime/>
    <om:resultTime/>
    <om:procedure xlink:title="NHDPlus reach used as centerline. Elevations extracted from NED raster."/>
    <om:observedProperty xlink:title="Centerline"/>
    <om:featureOfInterest xlink:href="#FLOWLINEEDGE-12100201000263"/>
    <om:result>
      <rml:ProfileLine gml:id="CENTER-2158A">
        <rml:surfaceLine>
        <rml:cutLine>
        <rml:profileLineType>centerLine</rml:profileLineType>
        <rml:profileLineLocation>center</rml:profileLineLocation>
      </rml:ProfileLine>
    </om:result>
  </rml:ProfileLineObservation>
</rml:profileLineObservation>
```

Figure 5.16 Sample XML Profile Line Observation

TimeseriesObservations use the WaterML 2.0 encoding (see Figure 5.17). However, for this project the discharge results are static risk-based values rather than a traditional temporal series. Therefore the values are encoded using the soft-typed NamedValue attribute rather than a series of points. The *featureOfInterest* for each TimeseriesObservation in this project is a Node.

```xml
<rml:timeseriesObservation>
  <om:OM_Observation gml:id="TIMESERIES-1">
    <om:phenomenonTime/>
    <om:resultTime/>
    <om:procedure xlink:title="HEC-HMS Model Output"/>
    <om:observedProperty xlink:href="http://kiwis.kisters.de/parameters/Q" xlink:title="Q"/>
    <om:featureOfInterest xlink:href="#NODE-1"/>
    <om:result>
      <wml2:MeasurementTimeseries gml:id="TIMESERIES-1A">
        <wml2:metadata>
         <wml2:MeasurementTimeseriesMetadata>
           <om:NamedValue>
             <om:name xlink:href="" xlink:title="25-year discharge"/>
             <om:value xsi:type="gml:MeasureType" uom="cumec">90</om:value>
           </om:NamedValue>
           <om:NamedValue>
           <om:NamedValue>
         </wml2:MeasurementTimeseriesMetadata>
        </wml2:metadata>
      </wml2:MeasurementTimeseries>
    </om:result>
  </om:OM_Observation>
</rml:timeseriesObservation>
```

Figure 5.17 Sample XML Timeseries Observation

The RiverML file described above merges data from terrain processing software used to extract river geometry information with data from hydrologic processing software used to calculate peak discharge values. This file can serve as the input for hydraulic processing software used to calculate flood depth data. Once calculated, the flood depths can be encoded as a set of TimeseriesObservations in the same fashion as the discharge data, using the same ReferenceNetwork. This sample project illustrates the use of RiverML as a software-independent transfer language for the communication of data for one-dimensional hydraulic river models.

Chapter 6: Conclusions and Future Work

6.1 CONCLUSIONS

An international standard format for river geometry and flow would be beneficial to the water resources community in order to support the growing use of Hydrologic Information Systems, enhance model interoperability, and promote data sharing among agencies responsible for measurement and forecasting. Such a format should capture common data types and concepts used in one-dimensional hydraulic modeling, and be extensible towards two-dimensional hydraulic models and hydrologic models.

The harmonization study performed here demonstrates that the concepts used across various data and simulation models bear strong similarities. This indicates that a standard transfer format is feasible. However, standardization of river geometry and flow faces challenges beyond those overcome by the standardization of time series during the development of WaterML 2.0. These include the need to establish network connectivity, to support multiple feature representations of varying dimensionality (such as 2D and 3D cross sections), to relate time series observations to geometry, and to manage scenarios.

A prototype model was developed based on the findings of the harmonization study. The prototype organizes data into three primary categories. First, a network of reference features is defined, which may or may not have explicit geometry. Second, observations about geometry and time series are listed, using the reference features as a framework to establish connectivity relationships. Finally, scenarios are defined that identify a set of observations which form a cohesive modeling unit. The information model also allows the inclusion of metadata and data describing the Digital Elevation Models used as the source for geometry features. These DEMs can be used either to inform one-dimensional modeling efforts or to couple one- and two-dimensional models together.

The following list presents specific recommendations for a standard river transfer language based on the harmonization effort:

- Develop the standard using the existing framework of OGC standards such as GML, O&M, and WaterML.
- Focus on the clarity of the information model rather than optimization of the XML encoding.
- Use hard-typing for all elements which can be consistently communicated across the domain of hydraulic modeling, and use soft-typing to allow flexibility for context-specific definitions.
- Support community-defined definitions of vocabularies through the use of URI references.
- Use geometry values (x,y,z coordiantes) rather than processed values (wetted area, hydraulic radius).
- Emphasize georeferenced features rather than tabular data (i.e. station-elevation tables). However, tabular data may be useful to include, especially for storage areas.
- Support the concurrent use of both 2D and 3D representations of feature geometry.
- Profile lines require differentiation by type (i.e. bank, thalweg) and location (i.e. left, right, center).
- Linear attributes can be described using the event formulation.
- It may be beneficial to treat linear attributes as O&M observations. However, this is more detailed than any present format and thus may be excessive for a transfer schema.

- The most common reference approaches for relating features in a river hydraulic model are some blend of topological and river addressing. At a minimum a transfer language should be able to clearly differentiate these two methods. Two other reference approaches are topographical and relative addressing; support for these methods can be optionally included.
- The initial focus for describing time-varying geometry should be a scenario approach. Consider extending the model to include feature series in a later version.
- Reevaluate the use of a control volume/flux conception as the Spatiotemporal Data Model matures.
- Structures will require a separate harmonization effort.
- Catchments will require a separate harmonization effort.
- Explore the potential for collaboration between the LandXML 2.0 and the RiverML initiatives.

6.2 FUTURE WORK

6.2.1 Develop Improved Prototype

Following the presentation of this report to the OGC/WMO Hydrology Domain Working Group, it is recommended that the members of the community perform an initial review of the findings and the prototype model. This review should include checking the model for compliance with GML, O&M, and WaterML 2.0. The errors or improvements identified can revised and be incorporated into RiverML 0.3, which will serve as the basis for more detailed testing.

The following list presents specific recommendations for improving RiverML 0.2:

- Confirm that the prototype is consistent with relevant OGC and ISO standards.

- Develop a set of Schematron files for rules which are not explicitly enforced by the XML schema.
- Determine whether a GML Simple Features implantation of RiverML is possible.
- Investigate the need for optional conformance classes for RiverML. Possible conformance classes include cases where the networks are restricted to be dendritic and cases where restrictions are placed on the use of Surfaces.
- Rename feature classes and properties where clearer names are identified.
- Identify areas where the model can be simplified without losing critical functionality.
- Clarify the use of relative and absolute measures for linear referencing.
- Coordinate Reference Systems:
    - Confirm that SC_CRS is the appropriate property type for simply and concisely identifying reference systems common to hydraulic modeling.
    - Determine whether separate horizontal and vertical reference systems are required.
- Reference Network:
    - Test and refine the general approach of using geometric & schematic network elements.
    - Test the clarity of the *relatedReferenceFeature* property of SchematicReferenceFeature.
    - Experiment with establishing an authoritative ReferenceNetwork based on sources such as NHDPlus and Geofabric.
    - Determine whether the number of subclasses for GeometricReferenceFeature should be reduced in order to provide a

simpler information model, or whether the specialized classes are required in order to provide adequate clarity.

    - Investigate the utility and clarity of the EdgePoint class for allowing both extensible networks and the network-snapped locations of off network monitoring stations.
    - Test and refine the use of the OffsetEdgePoint class for communicating the 2D and 3D observations supported by Arc River.
    - Determine whether the start and end vertices for the OnNetworkEdge *shape* property should be restricted to match the respective *fromPoint* and *toPoint* coordinates.
    - Determine whether there should be Scenario-level control over the value of the *enabled* property for OnNetworkEdge features.
    - Evaluate the use of the ShorelineEdge feature class.
- River Features:
    - Refine the RiverFeatureObservation with appropriate properties and code lists.
    - Test and refine the use of CrossSection *edgeMeasure* when the *featureOfInterest* is a Link (which has no explicit linear measurement).
    - Expand the functionality of the ProfileLine *profileLineType* property to include user-defined descriptions.
    - Determine whether RiverML should support tabular, non-georeferenced feature geometry for cross sections and reservoirs. Note that in the case of reservoir volumes, neither HEC-RAS nor MIKE11 support any alternative to tabular data.

    - If tabular data is supported in future versions, align the usage of TableFeature with the ConversionTable class from WaterML 2.0 Part 2.
    - Expand the structure capabilities of RiverML to include dams, bridges, culverts, pumps, pipe networks, and other structures.
- Geometry:
    - Confirm that GM_LineString is the appropriate property for the Edge *shape* property.
    - Investigate the use of OM_Observation to support a feature series approach to time-varying geometry such as is supported by Arc River. This would require each feature to have a stable identifier across a set of time-stamped observations. A specific challenge for this approach is integration with models such as HEC-RAS and MIKE11 which use a scenario rather than feature series approach.
- Geometry Properties:
    - Determine whether RiverML should define geometry properties such as roughness coefficients as O&M Observations where the *featureOfInterest* is a GeometryFeature. This would allow added clarity at the expense of added complexity.
    - Expand the functionality of the HydroEvent *eventType* property to include user-defined descriptions.
    - Allow event elevations to be specified as either relative or absolute values.
- Surface:
    - Test and refine the general approach of using a SurfaceObservation to provide metadata on the source of RiverFeatureObservation geometry.

  - Test the practicality and utility of including the source terrain file using the Surface *shape* property.
  - Determine what surface types and encoding formats RiverML should be able to directly support with the Surface *shape* property. These formats should be appropriate for both 1D feature extraction and 2D hydraulic modeling.
  - Determine how unsupported surface types and encoding formats should be described.
  - Determine the appropriate *featureOfInterest* class for a SurfaceObservation.
  - Determine how the following cases should be distinguished: raw measurement data, data which has been adjusted to better represent the conditions at the time of measurements (e.g. by adding breaklines to a TIN), and data which has been adjusted to represent a future or hypothetical situation. The RiverML 0.2 method of using *observationType* is ambiguous.
  - Determine what metadata is required for future or hypothetical situations, and whether it belongs with the SurfaceObservation or the Scenario.
  - Determine what additional metadata is required, such as horizontal and vertical accuracy.
- Reservoir:
  - Determine the appropriate *featureOfInterest* class for a ReservoirObservation.

    - Test and refine the use of the ReservoirInterface class for both clarity and functionality.
- Time series:
    - Determine whether WaterML 2.0 standard needs to be modified to allow ReferenceFeature as an acceptable *featureOfInterest* for TimeseriesObservation.
    - Determine the appropriate method of conveying station location for a TimeseriesObservation along a Link or an Edge.
    - Determine appropriate method for communicating properties such as water surface elevation and discharge which are aggregated non-temporally, such as by risk factor (e.g. 50-year, 100-year). RiverML 0.2 currently uses the soft-typed capabilities of WaterML 2.0.
- Hydrology:
    - Perform Harmonization effort for hydrologic models.
    - Determine whether RiverML should be expanded to include hydrologic information or whether a separate, related standard should be developed.
    - Add a ReferenceFeature class to support area-based observations.
    - Expand the GeometryProperty class to allow description of properties for polygons, such as land use and soil type.

6.2.2 OGC Adoption

Once a suitable version of RiverML is ready, involvement should be sought by interested agencies and technology partners around the world. A set of formal Interoperability Experiments within the OGC framework for testing specific exchange scenarios should be conducted. The results of these experiments should be used to revise and improve the model in an iterative fashion. When a suitably robust version has been developed, it can be recommended for adoption by the OGC as an official standard.

6.2.3 HydroShare Implementation

In parallel with the official OGC adoption procedures, it is recommended that the CUAHSI HydroShare project continue to serve as a driving force for the development of RiverML. The following efforts are aligned with the general goals of HydroShare, and would assist the OGC community in advancing the functionality of RiverML:

1. Create a RiverML Resource type within Hydroshare that can be used to upload, visualize, store, and share river model data.
2. Develop conceptual mappings between RiverML and related information models such as Arc Hydro, Arc River, HEC-RAS, MIKE11, NHDPlus, Geofabric, and HY_Features.
3. Develop automated tools which can convert RiverML to and from the mapped models.
4. Demonstrate the use of RiverML as an intermediary between various models.

6.2.4 Long Term Implementation

Once RiverML has been adopted as an official OGC standard, it is recommended that software developers such as ESRI, HEC, DHI, and Streamline Technologies begin offering native support for RiverML import and export. Developers of datasets such as the NHDPlus and Geofabric could offer a version of their products in RiverML format, establishing authoritative reference networks to enhance data sharing and model interoperability. Finally, it is recommended that the scope of RiverML gradually be increased either directly or by integration with related standards in order to convey information pertaining to structures, pipe networks, drainage areas, and subsurface water.

Appendix A: Terminology and Abbreviations

A.1 TERMINOLOGY

Coverage

Feature that acts as a function to return values from its range for any direct position within its spatial, temporal, or spatiotemporal domain. [ISO 19123:2005, definition 4.17]

Data model (or **information model**)

Any framework for storing information according to logical classes, typically including a prescribed set of relationships between classes.

Data provider (or simply **provider**)

Any agency or organization that hosts data on a remote server and allows access via the internet.

Data user (or simply **user**)

An individual or organization that accesses data for the purpose of analysis or visualization. Users may access data from multiple providers, pass data between multiple local software applications, and may also share both raw data and analysis results with other users. Thus the relationship between provider and user is that of one-way package delivery, while the relationship between users may be a complex collaboration.

Feature

Abstraction of a real-world phenomenon. [ISO 19156, definition 4.4]

Harmonize

To create the possibility to combine data from heterogeneous sources into integrated, consistent and unambiguous information products, in a way that is of no concern to the end-user. (Flanders Marine Institute, 2013)

Hydraulic models

Computational models which focus on determining the flow characteristics such as depth and velocity of water that has been concentrated into either natural or artificial channels.

Hydrologic models

Computational models which focus on determining the discharge and storage of surface water, generally by combining the effects of precipitation, land cover, and topography.

Property

Facet or attribute of an object referenced by a name. Depending on the implementation context, the terms *property* and *attribute* can have special connotations (Portele, 2007). In this paper, the term *property* is used in a general sense.

Profile line

A line drawn in the primary direction of flow for a river, which may or may not have elevation coordinates. This is generally synonymous with longitudinal lines and flow lines. In this paper the term *profile* is not used to indicate variations in elevation along a cross section line.

A.2 ABBREVIATIONS

CUAHSI	Consortium of Universities for the Advancement of Hydrologic Science Incorporated
DEM	Digital Elevation Model
DTM	Digital Terrain Model
DWG	Domain Working Group
GIS	Geographic Information System
GML	Geography Markup Language
HEC	Hydrologic Engineering Center
HIS	Hydrologic Information System

HMS	Hydrologic Modeling System
ICPR	Interconnected Pond Routing
ISO	International Organization for Standardization
NED	National Elevation Dataset
NHD	National Hydrography Dataset
NSF	National Science Foundation
O&M	Observations and Measurements
OGC	Open Geospatial Consortium
RAS	River Analysis System
RiverML	River Markup Language
SOA	Services-Oriented Architecture
SPRNT	Simulation Program for River Networks
TIN	Triangulated Irregular Network
UML	Unified Modeling Language
USGS	United States Geological Survey
URI	Uniform Resource Identifier
WaterML	Water Markup Language
WMO	World Meteorological Organisation
XML	Extensible Markup Language
1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional

References

Ackerman, C. T. *HEC-GeoRAS GIS Tools for Support of HEC-RAS Using ArcGIS® User's Manual*; US Army Corps of Engineers, Hydrologic Engineering Center, 2009.

Ames, D. P.; Horsburgh, J.; Cao, Y.; Kadlec, J.; Whiteaker, T. L.; Valentine, D. HydroDesktop: Web Services-Based Software for Hydrologic Data Discovery, Download, Visualization, and Analysis. *Environ. Model. Softw.* **2012**, *37*, 146–156.

Arctur, D.; Zeiler, M. *Designing Geodatabases - Case Studies in GIS Data Modeling*; ESRI Press: Redlands, California, 2004.

Brunner, G. *HEC-RAS River Analysis System Hydraulic Reference Manual*; Version 4.1; US Army Corps of Engineers, Hydrologic Engineering Center, 2010 a.

Brunner, G. *HEC-RAS River Analysis System User's Manual*; Version 4.1; US Army Corps of Engineers, Hydrologic Engineering Center, 2010 b.

Bureau of Meteorology. *Australian Hydrological Geospatial Fabric (Geofabric) Product Guide*; Version 2.1; Australian Government Bureau of Meteorology, 2012.

Cesur, D. GIS as an Information Technology Framework for Water Modeling. *J. Hydroinformatics* **2007**, *9*, 123.

Charnock, T. W.; Hedges, P. D.; Elgy, J. Linking Multiple Process Level Models with GIS. In *Application of Geographic Information Systems in Hydrology and Water Resources Management, Proceedings of the HydroGIS '96 Conference*; Kovar, K.; Nachtnebel, H. P., Eds.; International Association of Hydrological Sciences: Vienna, Austria, 1996; pp. 29–36.

DHI. *MIKE 11 Reference Manual*; DHI, 2012 a.

DHI. *MIKE 11 Short Introduction Tutorial*; DHI, 2012 b.

DHI. *MIKE 11 User Guide*; DHI, 2012 c.

DHI. *MIKE FLOOD User Manual*; DHI, 2012 d; pp. 1–160.

DHI. *MIKE HYDRO User Guide*; DHI, 2012 e; Vol. 1.

DHI. *MIKE ZERO Mesh Generator Step-by-Step Training Guide*; DHI, 2012 f.

Dornblut, I.; Atkinson, R. *OGC HY_Features: A Common Hydrologic Feature Model*; 11-039r3; Open Geospatial Consortium, Inc., 2013.

ESRI. *ESRI Shapefile Technical Description*; Environmental Systems Research Institute, Inc, 1998.

ESRI. Working with GML http://resources.arcgis.com/en/help/main/10.2/index.html#//003700000004000000.

Flanders Marine Institute. Data harmonization http://www.coastalwiki.org/wiki/Data_harmonization.

Goodall, J.; Maidment, D. R. A Spatiotemporal Data Model for River Basin-scale Hydrologic Systems. *Int. J. Geogr. Inf. Sci.* **2009**, *23*, 233–247.

Hilltop Software. DataTamer: Hilltop Software http://www.hilltop.co.nz/.

Hodges, B. R. Simulation Program for River Networks (SPRNT) http://www.crwr.utexas.edu/hodges/sprint/.

Hodges, B. R.; Imberger, J. Simple Curvilinear Method for Numerical Methods of Open Channels. *J. Hydraul. Eng.* **2001**, 949–958.

Kim, D. Arc River - Geotemporal River Hydrodynamic Data Model. Ph.D. Dissertation, The University of Iowa, 2008.

Kim, D.; Muste, M.; Merwade, V. M. A Relational Data Model for Multi-Dimensional Representation of River Hydrodynamics and Morphodynamics. **2014**, submitted for publication.

Kreyszig, E. *Advanced Engineering Mathematics*; 8th Editio.; John Wiley & Sons, Inc.: New York, 1999.

LandXML.org. LandXML 2.0 Data Schema Plan Announced http://www.landxml.org/ (accessed Apr 23, 2014).

Lemon, D.; Maidment, D. R. *Hydrology Domain Working Group Charter*; OGC 08-095r54; Open Geospatial Consortium, Inc., 2009.

Liu, F. *SPRNT User’s Manual*; Version 1.2.2; IBM Research Austin, 2014.

Liu, F.; Hodges, B. R. River Dynamics Modeling 2.0. *EWRI Curr.* **2014**, *16*, 8–9.

Lott, R. *The OGC Abstract Specification Topic 2: Spatial Referencing by Coordinates*; OGC 08-015r2; Open Geospatial Consortium, Inc., 2010.

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

Maidment, D. R. *Hydrologic Information System Status Report*; Version 1; The Consortium of Universities for the Advancement of Hydrologic Science, Inc., 2005.

McKay, L.; Bondelid, T.; Dewald, T.; Rea, A.; Johnston, C.; Moor, R. *NHDPlus Version 2 : User Guide*; Version 2.1; United States Environmental Protection Agency, 2014.

Merwade, V. M. Geospatial Description of River Channels in Three Dimensions. Ph.D. Dissertation, The University of Texas at Austin, 2004.

Merwade, V. M.; Maidment, D. R.; Hodges, B. R. Geospatial Representation of River Channels. *J. Hydrol. Eng.* **2005**, *10*, 243–251.

Moore, R. *The OpenMI "in a Nutshell"*; The OpenMI Association Technical Committee, 2010.

Obenour, D. R.; Maidment, D. R. *Arc Hydro Developments for the Lower Colorado River Basin*; Center for Research in Water Resources, 2004.

Portele, C. *OpenGIS® Geography Markup Language (GML) Encoding Standard*; OGC 07-036; Open Geospatial Consortium, Inc., 2007.

Scharffenberg, W. A. *Hydrologic Modeling System (HEC-HMS) User's Manual*; Version 4.0; US Army Corps of Engineers, Hydrologic Engineering Center, 2013.

Streamline Technologies. *ICPR Version 4 User Manual*; Second Printing; Streamline Technologies, Inc., 2014 a.

Streamline Technologies. Migrating from ICPR 3 to ICPR 4, 2014 b.

Tarboton, D.; Ames, D. P.; Arrigo, J.; Goodall, J.; Maidment, D. R.; Idaszak, R.; Band, L. E.; Merwade, V. M.; Song, X. C.; Valentine, D. *Collaborative Research: SI2-SSI: An Interactive Software Infrastructure for Sustaining Collaborative Community Innovation in the Hydrologic Sciences*; Proposal submitted to NSF that served as the basis for Collaborative NSF awards 1148453 and 1148090, 2011.

Taylor, P. *Harmonising Standards for Water Observation Data - Discussion Paper*; OGC 09-124r2; Open Geospatial Consortium, Inc., 2010.

Taylor, P. *OGC® WaterML 2.0: Part 1- Timeseries*; OGC 10-126r3; Open Geospatial Consortium, Inc., 2012.

Taylor, P. *WaterML2.0 - Part 2 : Ratings, Gaugings and Sections Discussion Paper*; 13-021r3; Open Geospatial Consortium, Inc., 2013.

Taylor, P.; Cox, S.; Walker, G.; Valentine, D.; Sheahan, P. WaterML2.0: Development of an Open Standard for Hydrological Time-Series Data Exchange. *J. Hydroinformatics* **2014**, *16*, 425.

Technical Committee ISO/TC 211. *Geographic Information - Observations and Measurements*; ISO/DIS 19156; International Organization for Standardization, 2010.

van den Brink, L.; Portele, C.; Vretanos, P. A. *Geography Markup Language (GML) Simple Features Profile (with Corrigendum)*; OGC 10-100r3; Open Geospatial Consortium, Inc., 2012.

Vanecek, S.; Moore, R. *OGC® Open Modelling Interface Interface Standard*; OGC 11-014r3; Open Geospatial Consortium, Inc., 2014.

Whiteaker, T. L. Geographically Integrated Hydrologic Modeling Systems. Ph.D. Dissertation, The University of Texas at Austin, 2004.