WaterStrategyMan
EVK1-CT-2001-00098

DELIVERABLE 12
GIS DATABASE

Prepared by the
National Technical University of Athens

February 2003

Contents

Preface

This document presents the concepts, the design and the development of the Water Strategy Man Database. The Deliverable summarizes the research that was carried out in Work Package 6 of the project, "Data and GIS Database", which according to the contract, aimed at "the collection of available information on water resources and demand in the identified paradigms and the development of a GIS Database, to accommodate them in an efficient and updateable mode". The database has been developed under a common analysis framework, in order to provide a consistent methodology for analyzing and organizing the data required to conduct the case study analysis in combination to the developed Decision Support System, described in Deliverable 11 *"Integrated Decision Support System applicable to the paradigms"*.

The design and development of the GIS Database is an integral part of the Analysis Phase of the WSM project, the goal of which was to set the basis for the analysis of Case Studies and the formulation of Strategies in the upcoming work packages of the project.

The Analysis phase (Figure 1) consisted of four work packages:

1. **WP4**, realized during the first year of the project, focused on the development of a consistent methodology for shaping and analysing water resources management scenarios and water management options. The findings of WP4, presented in Deliverable 7 of the WSM project were used as a basis for the definition of required data and structure and the development of the Decision Support System under WP5.
2. **WP5** involved the development of a Decision Support System (DSS) that will be used to support and assess the findings of the Case Studies. Available tools for the assessment of water resources availability and use are reviewed, and the most appropriate of these for the regions identified are selected. These tools are then tested and adapted for the project methodology.
3. **WP6** focused on the design and development of a GIS Database to be used by the Tool, using data collected for the Case Study regions. The outcomes of this work package are discussed in this deliverable.
4. **WP7** involved the organization of a workshop to present the findings of WP 5 and 6. The aims of the workshop were to:
    - Present and discuss results of the previous stages of the Project
    - Validate the developed methodology and selected tools and prepare the development of water management scenarios in the identified Paradigms, and
    - Train participants on the DSS and methodology application issues

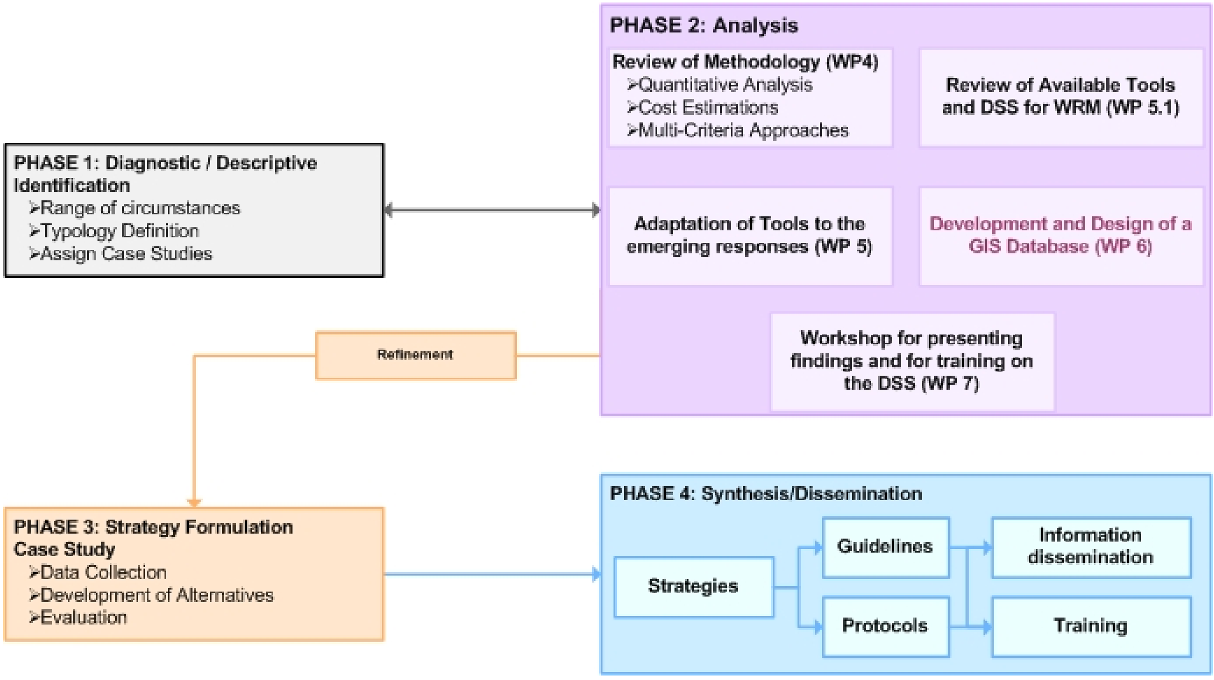


Figure 1. The phases of the WSM Project

The structure of this Deliverable is similar to the one of the WSM Database; the first two introductory chapters present the purpose and the interrelation of the GIS Database with the Decision Support System. The rest of the document presents the data model in terms of spatial and non-spatial attributes and focuses on the main components required during the modelling procedure of a water resource system.

1 Introduction

Data modelling is the first step in database design, representing the blueprint from which the GIS will be built. Through modelling, complexity is reduced so that all parties involved are able to understand the essence of the system. This provides the basis of development of a common understanding of which objects should feature in the geographic database and how they should be represented. The model also aims to facilitate data sharing and encourage consistency in data formats.

1.1 The WSM Data Model

The data model aims to satisfy the requirements set out by the methodology selected and the Decision Support System, to analyze the Case Studies for the selected paradigms and to assist at the formulation and evaluation of strategies. Therefore, the data model has been designed in such a way so as to adequately describe a system in terms of water resources availability, demand, infrastructure and management options and developmental policies to be formulated within the scope of the analysis.

Within the model, logically related features are grouped together. Thus, the model extends the basic distinctions between water resource systems, demands, infrastructure and administrative structures. The core components of the data model are feature datasets as those are described in the following paragraphs. They are used to group spatial data that share the thematic area, in order to facilitate the organization of collected information. For example, the boundaries of administrative regions are all belong in the same feature dataset, allowing for the user of the Decision Support System and the Database to visualize data together. that are

The core components of the data model, which are each analyzed in the respective chapter are:

- Basic Regional Data, organizing general information of the case study area.
- Water Bodies and Monitoring, representing the most important water bodies, as those are classified in the Water Framework Directive, and the monitoring network.
- Water Network data, designed to model the water resource system and infrastructures of the case study region.
- Administrative structures, standing for the administrative organization of the region.
- Time Series, modeling time series data, mostly related to water network objects.

It should be noted that in the approach adopted during the development of the WSM Data Model, some features are represented both as simple features (i.e. points, lines and polygons) and as a complex water network. This allows for both the accurate representation of the physical entity (e.g. a lake) and for the particular modelling requirements set out by the Decision Support System.

1.2 GIS Environment

The GIS is an essential component of the database because it allows the user to make use of the available spatial data. The role of the GIS is not only to depict maps, but also and primarily to create the spatial link among data. Overall, the platform for the development of a database for the modeling of a water resource system should allow for:

- Spreadsheet for problem definition and set-up.
- Catchment delineation.
- Topological unit delineation and mapping.
- Intersection among maps.
- Area averages computation.
- Management and conversion of raster and vector data.
- Visualization of map related results.

The WSM Data Model has been developed under Arc GIS 8.1. The main output of the data model is an Arc Info geodatabase, which stores information on all spatial and non-spatial attributes and classes included within the model.

A geodatabase is a relational database that contains geographic information. Geodatabases contain feature classes and tables. Feature classes can be organized into a feature dataset; they can also exist independently in the geodatabase. Many objects in a geodatabase can be related to each other. Those relationships between geodatabase objects are called relationship class. Relationships allow for attributes stored in a related object to symbolize, label, or query a feature class.

The geodatabase model defines a generic model for geographic information. This generic model can be used to define and work with a wide variety of different user- or application-specific models. By defining and implementing a wide variety of behavior on a generic geographic model, we provide a robust platform for the definition of a variety of user data models. The geodatabase supports a model of topologically integrated feature classes, similar to the coverage model. However, it extends the coverage model with support for complex networks, relationships among feature classes, and other object-oriented features.

1.3 Interrelation of the GIS Database and the WSM Decision Support System

The GIS Database is strictly interrelated with the methodology applied in analyzing and simulating water resource systems and consecutively with the WSM Decision Support System Besides the accommodation of available data and the selection of the appropriate platform for facilitating data collection and entering, the GIS Data Model has been developed keeping in mind the final goal of the WSM DSS, which is the analysis of water management strategies in the project case study area. To this end, the Data model should be able:

- to accommodate all data related to the simulation of different water availability scenarios and demand forecasts, including forecasts of pressures
- to store information on the different water management instruments proposed for the different case studies of the project.

Since the spatial scale of each case study is different ranging from a river basin in Portugal (Ribeiras do Algarve) to a small island in the Greek Aegean Sea, one primary aim of the data model was to allow for the modelling of those very different systems under a unique but flexible framework.

With those objectives in mind, the procedures for updating and entering new information on the case study regions have been fully integrated within the WSM DSS, in order to assist the users of the tool in understanding the essence and importance of each spatial object and to be able to verify data integrity and assumptions which could inhibit the correct simulation of the water system. The WSM DSS actually consists of three layers, all interconnected (Figure 2):

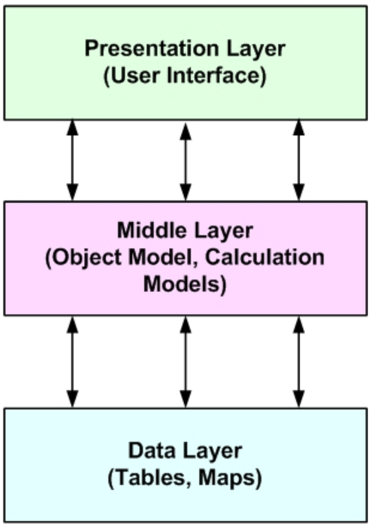


Figure 2. The three layers of the WSM DSS

The Data Layer (database) is strictly interrelated with the most important middle layer which integrates the Object Model of the WSM DSS and the calculation models. In order to facilitate the process of data updating and verification, it has been considered necessary that all data reading/updating procedures are implemented as methods of the respective objects of the WSM Object Model.

2 Terminology and Notation

2.1 About the Unified Modelling Language

The Unified Modelling Language (UML) is a modelling notation that provides tools for modelling every aspect of a software system from requirements to implementation.

UML has become a standard methodology, and is increasingly being applied to the modelling and design of Geographic Information Systems and databases. In this Deliverable, a UML diagrammatic notation, in accordance to the ARC GIS Object Model, is used to present an overview of the logical model of the database.

2.2 Spatial Objects, Feature Classes and Feature Datasets

In general, datasets are defined as a collection of data on a common theme or having similar attributes. In the Arc GIS Object Model for geodatabases, feature datasets are defined as collection of feature classes that share the same spatial reference. Because of this, all classes belonging in a particular feature dataset, they can participate in topological relationships with each other, such as a geometric network.

A feature class is the conceptual representation of a geographic feature, such as a point, a polygon, a polyline/line or annotation. All these classes inherit from the class "*feature*", in that they have geometry and will have a unique internal identifier in the relevant table. It should be noted that feature classes cannot mix geometry types – they must be exclusively points, or lines, or polygons.

2.3 Non-spatial objects

Non-spatial objects can also be included in a GIS Database. They are commonly used to group attributes related to one or more feature classes and are usually connected to them with one-to-many or many-to-many relationships. In the Arc GIS object model, non-spatial objects inherit from the ESRI Object abstract class.

2.4 The concept of inheritance and abstract classes

Inheritance allows for classes to be related to parents through generalization. The more specific class inherits attributes from the general abstract class. In a UML Object Model abstract classes are marked in italic whereas generalization relationships are noted with an arrow directed from the object to the parent. It should be noted that abstract classes are not physically represented in the database; however, their modelling functionality is evident in order to visualize the interrelationships between the objects and establish a conceptual model which is easy to manipulate and interpret. In UML diagrams, abstract classes are marked in italic letters while generalization is symbolised with an arrow originating from the child object and ending at the parent class.

2.5 Relationships and Relationship classes

Objects in a real-world system, often have particular associations with other objects in the database. For example, in a water network, a valve vault contains water valves. In a parcel database, a parcel has one or many owners.

These types of associations between objects in the geodatabase are called relationships. Relationships can exist between spatial objects (features in feature classes), non-spatial objects (rows in a table), or spatial and non-spatial objects. While spatial objects are stored in the geodatabase in feature classes, and non-spatial objects are stored in tables, relationships are stored in relationship classes.

Properties of a relationship

Like any association, relationships have particular characteristics. One obvious characteristic is the notion of **cardinality**. Cardinality describes how many objects of type A are related to an object of type B. In general, relationships can have one-to-one, one-to-many, many-to-one, and many-to-many cardinalities. Certain types of relationships support certain cardinalities, according to the definition of relationship rules.

A relationship between two objects is maintained through attribute values for key fields. Relationship classes can also have attributes. Any relationship class that has attributes must be stored as a table in the database and have a pair of foreign keys, each referencing the origin and destination classes of the relationship class. In this case, each relationship is stored as a row in the relationship classes table. Similarly, any many-to-many relationship classes require a table in the database to store at least the foreign keys.

Relationship classes have path labels. Forward and backward path labels describe the relationship when navigating from one object to another. The forward path label describes the relationship navigated from the origin class to the destination class; the backward path label describes navigating from the destination to the origin class.

Relationship classes can also be used to propagate standard messages between related objects. Messaging is the mechanism that objects related to each other use to notify each other when they change.

Types of relationships

There are two types of relationships: simple and composite. Simple or peer-to-peer relationships are relationships between two or more objects in the database that exist independently of each other. In a relationship between object A and object B, if object A is deleted from the database, object B continues to exist. For example, in a railroad network you may have railroad crossings that have one or more related signal lamps. However, a railroad crossing can exist without a signal lamp, and signal lamps exist on the railroad network where there are no railroad crossings. In composite relationships the lifetime of one object controls the lifetime of its related objects. Therefore, when an object in the primary class is deleted, the related objects in the destination class are deleted as well. Composite relationships are always one-to-many, but can be constrained to be one-to-one using relationship rules.

Attributed relationship classes

One-to-one and one-to-many relationship classes do not require a new table to be created in the geodatabase to store the relationships. However, many-to-many relationship classes do require a new table in the database to store the foreign keys from the origin and destination classes to make the relationship. This table can also have other fields to store attributes of the relationship itself that are not attributed to either the origin or destination class. One-to-one and one-to-many relationship classes may also have attributes; in this case, a table would be created to store the relationships.

Relationship rules

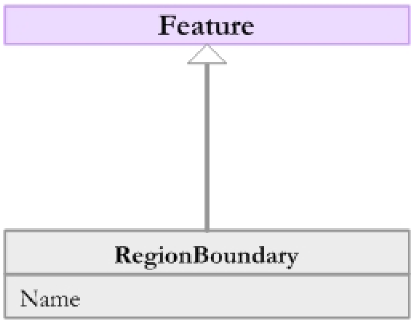
Relationship classes can have an associated set of relationship rules. Relationship rules control which objects subtypes from the origin class can be related to which object subtypes in the destination class. They can also be used to specify a valid cardinality range for all permissible subtype pairs.

3 Basic Regional Data

Some basic information about a case study region is modelled within the *Physical Data* feature dataset, which describes the region in terms of geomorphology, land use patterns and predominant soil phases. Classes in the dataset are used to provide a general overview of the region while some of them can be used as input for the water availability model of the WSM Decision Support System. Basic geographical information include the regional map (boundary of the case study area), elevation, slope, land use patterns, geology patterns, soil types as well as major civil infrastructure such as roads, towns and agglomerations. The structure of the dataset is analyzed in the following paragraphs.

3.1 Geomorphology

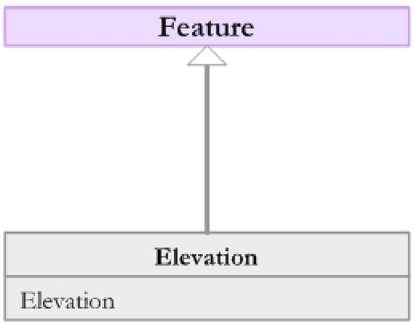
Region Boundary



The *RegionBoundary* feature class inherits from the *Feature* object of the Arc Info Object Model. Spatial objects are defined as polygons and designate the boundary the boundary of the case study area. One attribute is defined:

- **Name** – String. The locally used name of the case study region.

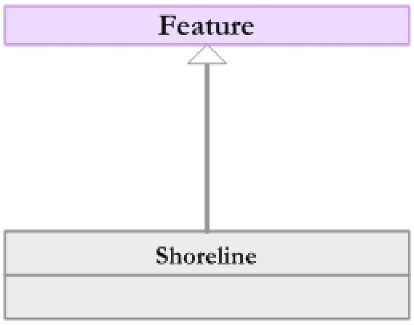
Elevation



The *Elevation* is a polygon feature class describing the altitude ranges encountered in the case study region. Each polygon feature represents an area with a particular altitude range (e.g. from 100 to 200 m), defined by the objects with lower and higher elevation. The *Elevation* class can be used for deriving the Digital Elevation Model for the case study region which is used in the water availability estimations performed by the WSM Decision Support System. The class defines one attribute:

- **Elevation** – Integer. The elevation (altitude range) of the corresponding polygon feature in meters.

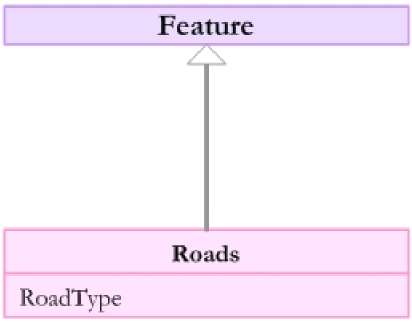
Shoreline



The *Shoreline* feature class is a polyline feature class for the sea-boundary of the case study region. No additional attributes are defined.

3.2 Infrastructure

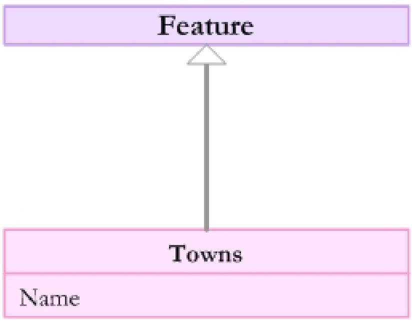
Roads



The *Roads* feature class is a polyline feature class storing information on the road infrastructure of the case study area. One attribute is defined to distinguish between the major road types:

- **RoadType** – Domain attribute (see Appendix for values and descriptions). RoadType is a coded value domain attribute, characterizing the roads of the case study area according to their importance and size (e.g. national road, national country road etc).

Towns

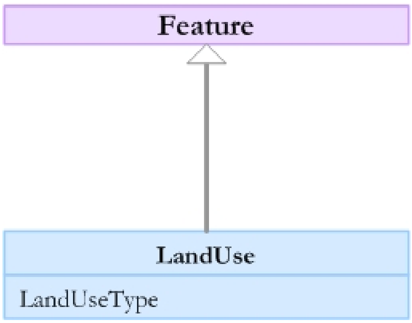


The *Towns* feature class is designed to accommodate the location of important towns and agglomerations. Towns are represented as points. The class defines one attribute:

- **Name** – String. The locally used name of the city.

3.3 Land Use & Geology

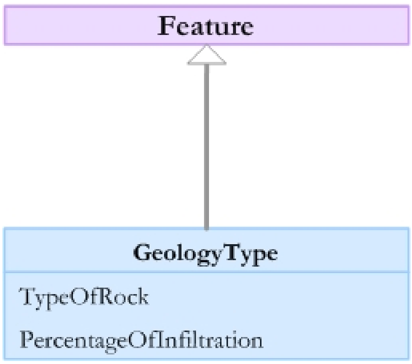
Land Use



The *LandUse* feature class is used to characterize the case study area in terms of land use patterns. Classification is based on the USGS classification system and the different areas are modeled as polygons. The class defines one domain attribute, which is used for the classification:

- **LandUseType** – Domain attribute (see Appendix).

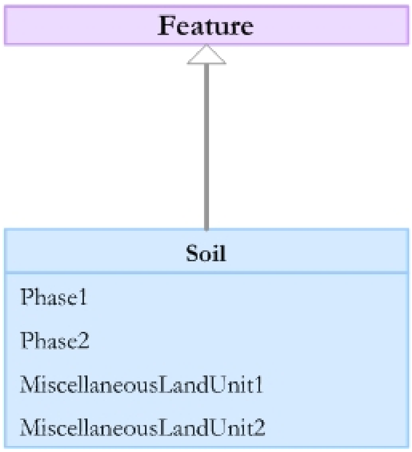
Geology



The *Geology* feature class characterizes the case study region in terms of dominant geology types. Areas with different types of geology are modeled as polygon features. Each object of the class is defined by the following attributes:

- **TypeOfRock** – String. A descriptive text for the geology type of the polygon feature.
- **PercentageOfInfiltration** – String. Share of precipitation that infiltrates the subsurface.

Soil



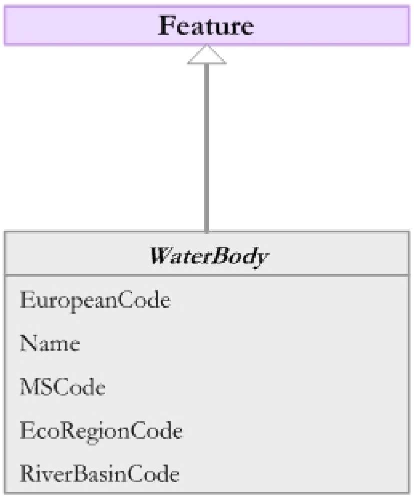
The Soil feature class is used to characterize the case study area in terms of soil type patterns. Spatial objects are defined as polygons on the basis of the first and second soil phases. Phases are subdivisions of soil units based on characteristics which are significant to the use or the management of land but which are not diagnostic for the separation of soil units themselves. The phases are: stony, lithic, petric, petrocalcic, petrogypsic, petroferric, phreatic, fragipan, duripan, saline, sodic and cerrado. The class defines the following coded value domain attributes, which are more explicitly described in the Appendix:

- **Phase1** – Domain attribute
- **Phase2** – Domain attribute.
- **MiscellaneousLandUnit1** – Domain attribute.
- **MiscellaneousLandUnit2** – Domain attribute.

4 Water Bodies & Monitoring

Modelling of water bodies and monitoring stations is performed in accordance to the *"Guidance Document on Implementing the GIS Elements of the Water Framework Directive"*. Spatial objects are included in the *Water Bodies* feature dataset. At this point it should be emphasised that the purpose of the WSM Data Model, as well as that of the Decision Support System, was not to satisfy the reporting requirements of the WFD. However, it has been considered fundamental to incorporate some of the required elements for future use in order to enhance the capabilities of the database and to allow for further improvements of the DSS towards this direction as well. The basic object of the dataset is the *WaterBody* abstract class which is the parent for all surface and groundwater bodies included in the model.

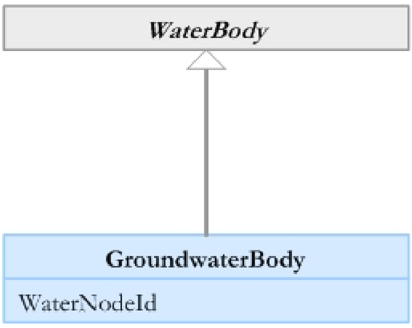
4.1 The Water Body Class



WaterBody is the basic feature class from which all surface and groundwater bodies inherit. The class defines the following attributes:

- **EuropeanCode** – String. European code.
- **MSCode** – String. Member state code.
- **Name** – String. Locally used name.
- **EcoregionCode** – String. The code of the ecoregion to which the water body belongs to.
- **RiverBasinCode** – String. The code of the river basin to which the water body belongs to.

Groundwater Body

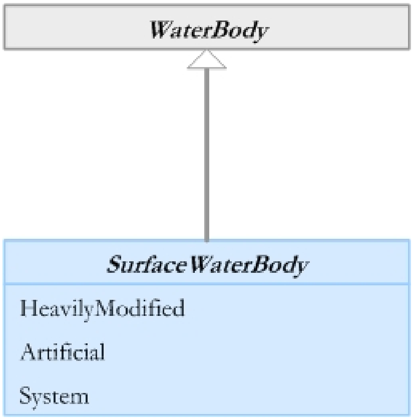


The *GroundwaterBody* object inherits from the *WaterBody* abstract class. Body of groundwater means a distinct volume of groundwater within an aquifer or aquifers. Aquifers are modelled within the *Water Bodies* Dataset as polygons and within the Water Network Dataset as points in order to allow for:

- The representation of individual aquifers as distinct water bodies and the estimation of their water balance;
- The modelling of aquifers as supply sources and a part of a water resource network.

The feature classes are associated with a one-to-one relationship through the *WaterNodeId* field.

Surface Water Body



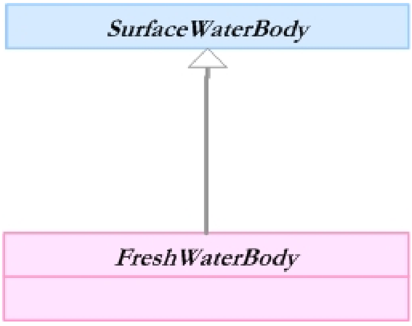
From the Directive definitions *"Surface water means inland waters except groundwater; transitional and coastal waters, except in respect of chemical status for which it shall also include territorial waters"*. Thus, the abstract class *SurfaceWaterBody* is further classified into the abstract classes *FreshWaterBody* and *SalineWaterBody*, both having a different set of attributes.

A *SurfaceWaterBody* abstract feature class defines the following attributes:

- **HeavilyModified** – True/False attribute. Heavily modified water body means a body of surface water which, as a result of physical alterations by human activity, is substantially changed in character, as designated by a Member State in accordance with the provisions of Annex II.

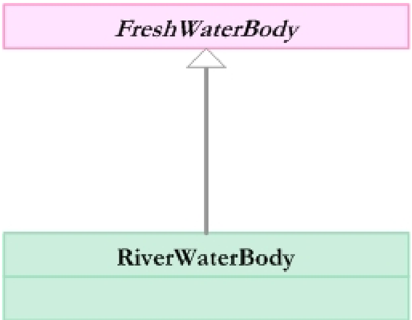
- **Artificial** – True/False attribute. Artificial water body means a body of surface water created by a human activity.
- **System** – A/B Domain attribute. Whether the water body is of Type A or B.

Fresh Water Body



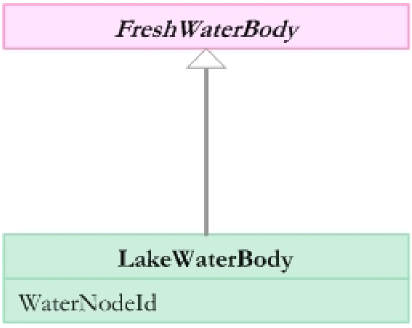
The abstract class *FreshWaterBody* inherits from the *SurfaceWaterBody* class. It has not been considered necessary to add attributes additionally to those inherited by the *SurfaceWaterBody* class. Further on, a fresh water body is classified either as a lake or a river water body.

River Water Body



RiverWaterBody means *"a body of inland water flowing for the most part on the surface of the land but which may flow underground for part of its course"*. A river water body is connected with a one-to-many relationship with the *RiverLink* feature class of the *Water Network* Feature Dataset which models the particular river segments.

Lake Water Body

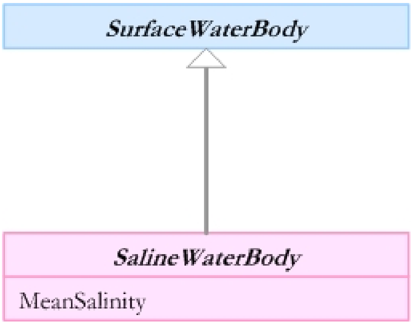


According to the Directive, *"Lake means a body of standing inland surface water"*. Lakes are modelled within the *Water Bodies* Dataset as polygons and at the *Water Network* Dataset as points. Similarly to groundwater bodies, this double representation allows for:

- The representation of individual lakes as distinct bodies and the estimation of their water balance;
- The modelling of lakes as supply sources and a part of a water network.

For this purpose, the polygon *LakeWaterBody* feature class is associated with the point *Lake* feature class of the *Water Network* Dataset with a one-to-one relationship through the *WaterNodeId* attribute (see Chapter 5 on the *Water Network* Dataset).

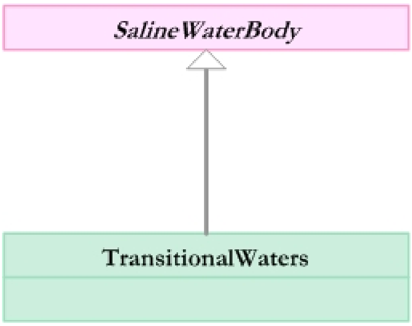
Saline Water Body



The abstract class *SalineWaterBody* inherits from the *SurfaceWaterBody* class and defines the following additional attribute:

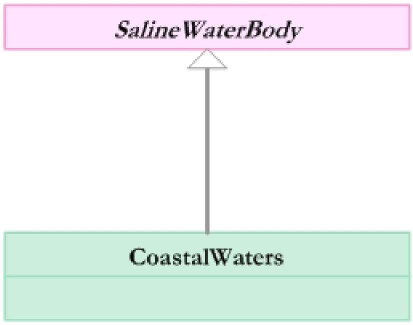
- **MeanSalinity** – Double. The mean annual salinity of the water body, which is used as the basis for its classification.

Transitional Waters



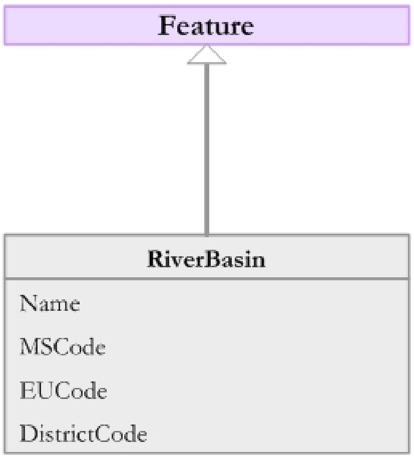
Transitional waters are *"bodies of surface water in the vicinity of river mouths which are partly saline in character as a result of their proximity to coastal waters but which are substantially influenced by freshwater flows"*. Transitional waters are typically estuaries and are modelled as polygon features.

Coastal Waters



Coastal waters means *"surface water on the landward side of a line, every point of which is at a distance of one nautical mile on the seaward side from the nearest point of the baseline from which the breadth of territorial waters is measured, extending where appropriate up to the outer limit of transitional waters"*. The *CoastalWaters* feature class inherits from the *SalineWaterBody* abstract feature class. Although additional attributes are recommended in the GIS Guidance Document, those are optional and therefore not incorporated in the current version of the WSM Data Model.

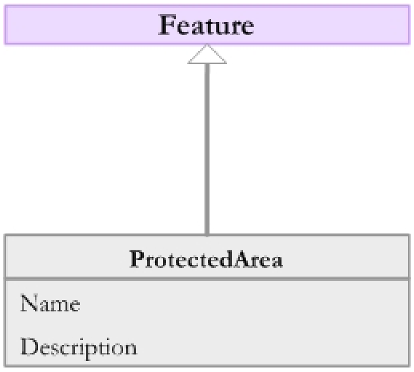
4.2 River Basin



River basin means *"the area of land from which all surface run-off flows through a sequence of streams, rivers and, possibly, lakes into the sea at a single river mouth, estuary or delta"*. The feature type of the class is polygons and in accordance to the Guidance Document on the implementation of the GIS Elements of the WFD the following attributes have been assigned:

- **Name** – String. Locally used name.
- **MSCode** – String. Unique code for a river basin within a Member State.
- **EuropeanCode** – String. Unique code for a river basin at EU Level.
- **DistrictCode** – String. Code of the River Basin District the basin belongs to.

4.3 Protected Areas



Protected areas are referred to in the Annex V of the WFD where it is stated that a river management plan *"shall include maps indicating the location of each protected area and a description of the Community, national or local legislation under which they have been designated".* The GIS Guidance Document on the Water Framework Directive specifies subtypes for the feature class, such as:

a) Areas designated for the abstraction of water intended for human consumption

b) Areas designated for the protection of economically significant aquatic species;

c) Bodies of water designated as recreational waters, including areas designated as bathing waters under Directive 76/160/EEC

d) Nutrient-sensitive areas, including areas designated as vulnerable zones under Directive 91/676/EEC (Nitrates Directive)

e) Areas designated as sensitive under Directive 91/271/EEC (Urban Waste Water Treatment Directive);

f) Areas designated for the protection of habitats or species where the maintenance or improvement of the status of water is an important factor in their protection, including relevant Natura 2000 sites designated under Directive 92/43/EEC (habitats) and Directive79/409/EEC (Birds).

It has been considered however that such a classification is out of scope for the purposes of the WSM Data Model. Therefore, the object is modelled as a single polygon feature class, providing a "*Description*" attribute for a more detailed representation.

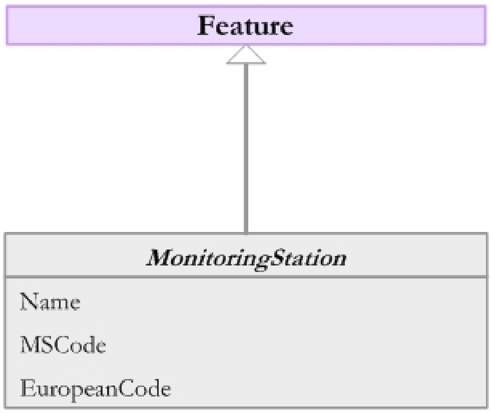
- **Name** – String. Locally used name for the protected area.
- **Description** – String. Description (category) of the protected area.

4.4 Monitoring Stations

Monitoring stations form the basis of the monitoring of water status. The Directive 2000/60 distinguishes between Surface Water Monitoring and Groundwater Monitoring. The monitoring is the basis for the subsequent classification of water bodies; however from a GIS perspective it is not required to access the underlying data used to arrive at these status characterisations. Annex V, Article 1.3 states that *'Member States shall provide a map or maps showing the surface water* *monitoring network in the river basin management plan"*. Similarly, Article 2.2.1 states that *"the groundwater monitoring network should also be provided as a map or maps"*.

Similarly to the WFD GIS Data Model, here monitoring stations are defined as an abstract class, further subdivided into *SurfaceMonitoringStation* and *GroundwaterMonitoringStation*. In order however, to extend the capabilities of the data model, one additional child object is defined for climate stations monitoring only meteorological conditions and not the status of one or more water bodies.

Monitoring Station

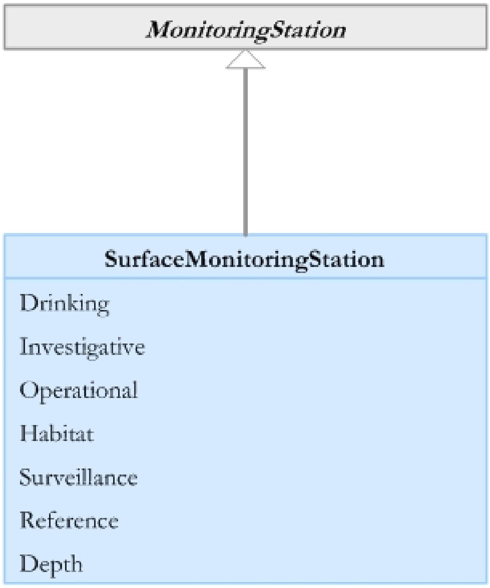


The abstract class *MonitoringStation* defines the following attributes:

- **Name** – String. If appropriate, a name can be provided for the station.
- **EuropeanCode** – String. A unique code, incorporating the ISO Country Code plus the Member State code below.
- **MSCode** – String. A unique code for the monitoring station.

Monitoring stations are point features. As previously stated they are subdivided into *SurfaceMonitoringStations*, *GroundwaterMonitoringStations* and *ClimateStations*. Monitoring stations may have multiple functions and can also monitor multiple water bodies. Therefore they are associated with many-to-many relationships with *WaterBodies* which are analyzed each at the corresponding paragraph.

Surface Monitoring Station

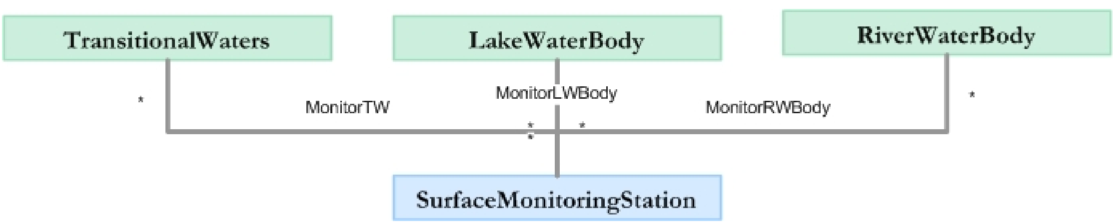


The feature class *SurfaceMonitoringStation* inherits from the *MonitoringStation* abstract class and defines the following additional attributes:

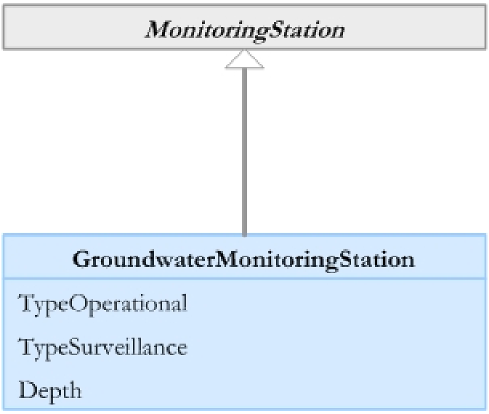
- **Drinking** – True\False attribute. Whether the station is a drinking water abstraction.
- **Investigative** – True\False attribute. Whether the station is an investigative station.
- **Operational** – True\False attribute. Whether the station is an operational station.
- **Habitat** – True\False attribute. Whether the station is a habitat monitoring station.
- **Surveillance** – True\False attribute. Whether the station is a surveillance station.
- **Reference** – True\False attribute. Whether the station is a reference station.
- **Depth** – Integer. Station depth in meters.

The feature class is associated with a many-to-many relationship with the following feature classes:

- *TransitionalWaters* (*MonitorTW* relationship class)
- *RiverWaterBody* (*MonitorRWBody* relationship class)
- *LakeWaterBody* (*MonitorLWBody* relationship class)



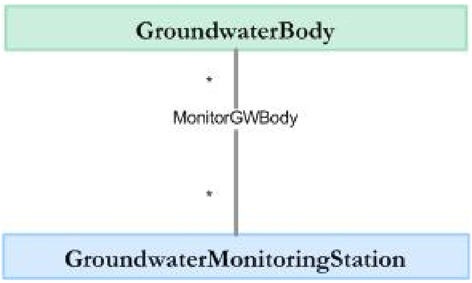
Groundwater Monitoring Station



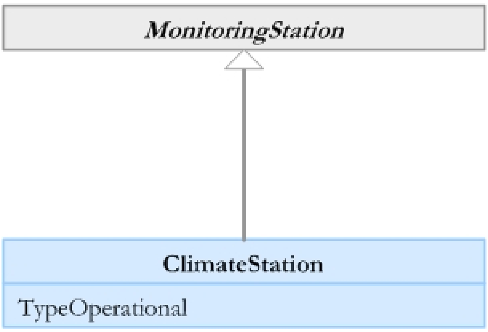
The feature class *GroundwaterMonitoringStation* inherits from the *MonitoringStation* abstract class and defines the following additional attributes which identify the functions it performs:

- **TypeOperational** – True\False attribute. Whether the station is an operational station.
- **TypeSurveillance** – True\False attribute. Whether the station is a surveillance station.
- **Depth** – Integer. Station depth in meters

GroundwaterMonitoringStation is associated through a many-to-many relationship with the *GroundwaterBody* feature class.



Climate Station



The *ClimateStation* feature class is the last class inheriting from the *MonitoringStation* class and is used to model stations that monitor meteorological conditions only. One additional attribute is defined:

- **TypeOperational** – True\False attribute. Whether the station is an operational station.

5 Water Network Data

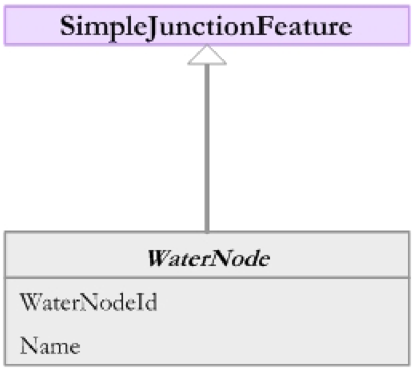
The *Water Network* feature dataset describes the connectivity of water flow through the system. The principal feature class of this dataset is an Arc Info utility geometric network, called the *WaterNetwork*. Generally, a network is described as a set of junctions (points) and edges (polylines) that are topologically connected to each other. A geometric network, besides containing the connectivity information between edges and junctions, also defines rules of behaviour, such as what classes of edges can connect to a particular class of junction or at what type of junction two classes of edges must connect. In utility network applications, knowing the direction of flow along network edges is essential. The flow direction in a network is determined by:

- The topology of the network;
- The locations of sources and sinks in the network, and
- The enabled or disabled state of the features.

In the WSM Data Model junction elements are conceptually represented as *WaterNodes* while the edges that stand for the connections between them are the *WaterLinks*. The *WaterNode* class contains junctions between *WaterLinks* and other points that are vital to the network analysis and the DSS simulation procedure. Those stand for supply (groundwater, storage reservoirs), demand (irrigation, domestic, industrial) and treatment points (drinking and wastewater treatment plants, and desalination). Water flows along *WaterLinks* from supply sources to water demanding activities.

Overall, the data modelled within the *Water Network* dataset serve as the basis for the case study analysis conducted through the WSM Decision Support System. The following paragraphs will analyze the structure of the dataset in terms of spatial objects. However, most of the geographic features that are included in the dataset are related to a variety of non-spatial object classes which organize data vital to the analysis. Therefore, in order to explore all the capabilities of the Data Model it is considered necessary that the presentation of spatial objects is in this case complemented by those of the related non-spatial object classes, in section 5.8. The modelling of time-series data, attributed to most water network objects is presented in Chapter 7.

5.1 The Water Node class



The *WaterNode* abstract class symbolizes any element or point of interest in the water network. It inherits from the *SimpleJunction* of the Arc Info Object Model. As a consequence all water nodes occur at the intersection of two or more edges (see *WaterLink* class) or at the endpoint of an edge, allowing for the transfer of flow between edges. On a first level of conceptualisation, the *WaterNode* class defines the following attributes:

- **WaterNodeId** – Integer**.** The Id attribute acts as the unique identifier of a water node in the network.
- **Name** – String. The name of the water node, standing for the locally used name or a description of the object.

As mentioned, Water Nodes are classified in three broad categories, in order to model their particular behaviour in a water resource system (Figure 3).

- **Supply Nodes**, representing natural and artificial water resources;
- **Demand Nodes**, representing consumptive and non-consumptive uses and activities;
- **Transshipment Nodes**, representing treatment processes or used to maintain network connectivity.

Those categories are then, through generalization, analysed in more child objects according to the specific functionalities and behaviour of each object, and the procedures for data handling and modelling as those have been integrated in the WSM DSS.

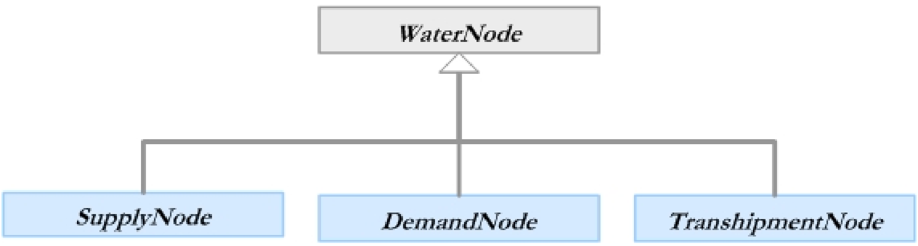
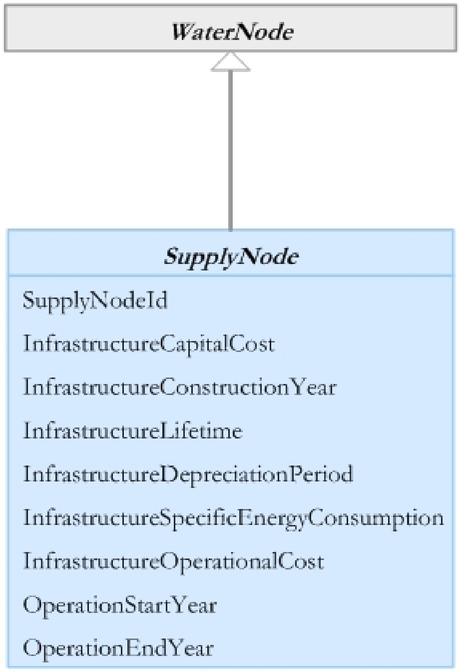


Figure 3. The Water Node class and its child objects

5.2 The Supply Node Class



The *SupplyNode* abstract class inherits from the *WaterNode* class and represents all water nodes acting as sources of supply. The class defines the following attributes:

- **SupplyNodeId** – Integer. The unique identifier of a supply node in the collection.
- **InfrastructureCapitalCost** – Integer. The capital cost for the construction of infrastructure related to abstractions.
- **InfrastructureConstructionYear** – Integer. The year when infrastructure has been constructed.
- **InfrastructureLifetime** – Integer. Infrastructure lifetime in years.
- **InfrastructureDepreciationPeriod** – Integer. The depreciation period for infrastructure related to abstractions in years.
- **InfrastructureSpecificEnergyConsumption** – Double. Specific energy consumption for abstraction in kWh/m^3.
- **InfrastructureOperationalCost** – Double. Specific operation and maintenance costs in €/m^3 abstracted.
- **OperationStartYear** – Integer. Specifies the year when the operation of the node as a supply node starts. Should be greater or equal to the infrastructure construction year.
- **OperationEndYear** – Integer. Specifies the year when the operation of the node as a supply node ends. Should be greater or equal to the infrastructure end year (construction year + lifetime).

To account for the different available water resources, the *SupplyNode* abstract class is subdivided into different child objects conceptualising aquifers, fossil groundwater, and surface water bodies and water transfer from neighbouring areas (Figure 4).

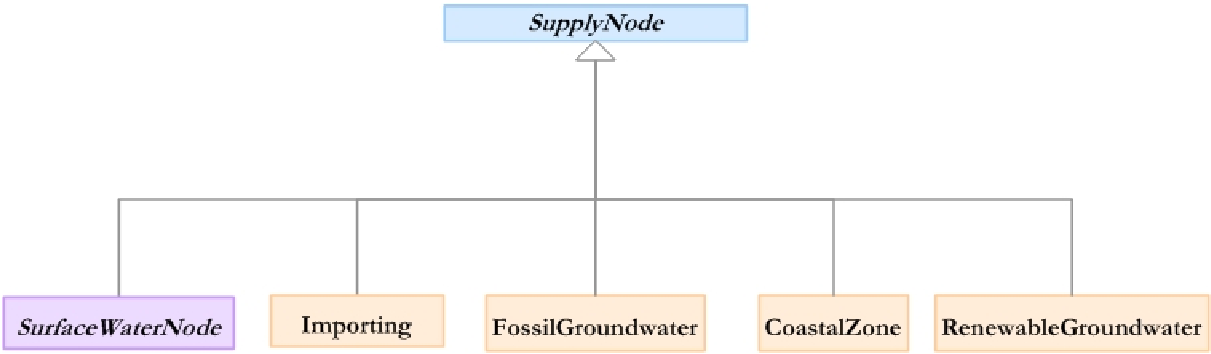
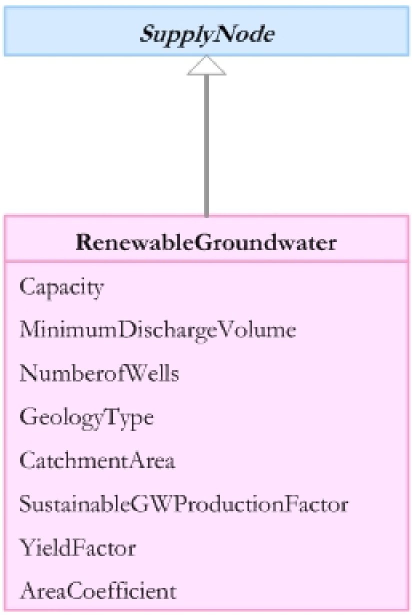


Figure 4. The child objects of the Supply Node class

Some child objects of the *SupplyNode* class are associated through many-to-many attributed relationships with the *QualityVariables* object class (see section 5.8). The aim of this relationship is to model non-spatial information regarding the vulnerability to water pollution of those supply nodes that also act as receptor bodies, namely renewable groundwater, coastal zones and surface water nodes.

Renewable Groundwater

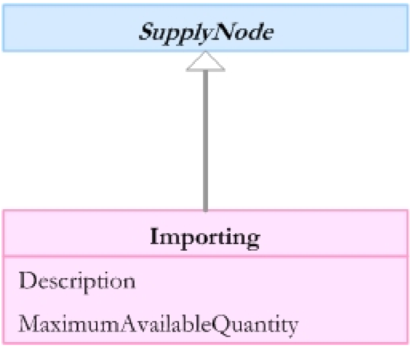


The *RenewableGroundwater* object models shallow, free groundwater that is continuously recharged by the hydrological cycle. The class inherits from the *SupplyNode* abstract class and additionally defines the following attributes:

- **Capacity** – Integer. The capacity (m^3) i.e. maximum volume of water that can be stored in the aquifer.

- **MinimumDischargeVolume** – Integer. The minimum stored volume (in m^3) for which natural discharge from the aquifer occurs.
- **NumberofWells** – Integer. The number of wells for abstracting water from the aquifer.
- **GeologyType** – Domain attribute: calcareous, siliceous, organic, other. The geology type of the aquifer.
- **CatchmentArea** - Double. The catchment area (m^2) i.e. the area from where water infiltrates in the aquifer period for infrastructure related to abstractions in years.
- **SustainableGWProductionFactor** – Double (values between 0 and 1). Factor expressing the percentage of aquifer recharge that can be considered as sustainable for abstraction.
- **YieldFactor** – Double (values between 0 and 1). Factor expressing the fraction of storage that can be abstracted. Accounts for deep percolation losses from the aquifer.
- **AreaCoefficient** – Double. Coefficient expressing the vulnerability of the aquifer to water abstraction.

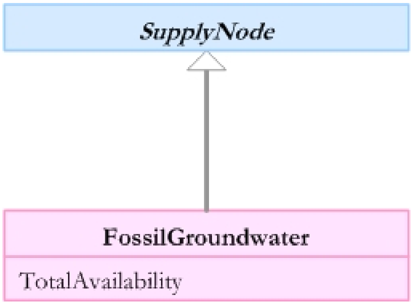
Importing



The Importing object inherits from the *SupplyNode* abstract class and represents the amount of water coming from a neighbouring area. As a supply node it has the role of water source. In addition to attributes inherited by the *SupplyNode* class, the following are defined:

- **Description** – String. A descriptive text for the type of imported water, e.g. inter-basin transfer, water hauling etc.
- **MaximumAvailableQuantity** – Double. The maximum volume of water that can be imported in the region in m^3/month.

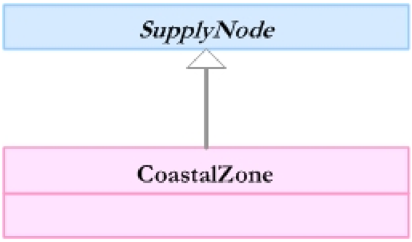
Fossil Groundwater



The *FossilGroundwater* class stands for deep, confined groundwater that is not recharged by the hydrological cycle. One additional attribute is defined:

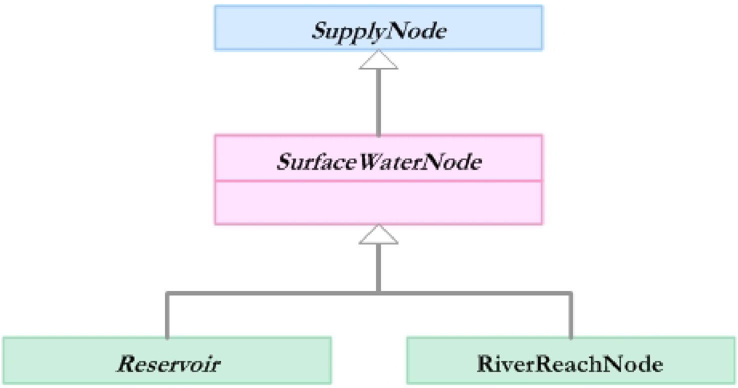
- **TotalAvailability** – Integer. Total volume stored within the aquifer in m^3.

Coastal Zone



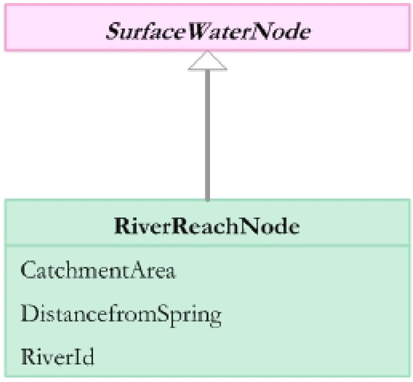
The *CoastalZone* class models a coastal area where seawater is abstracted for desalination, effluents are discharged or quality status, such as eutrophication, is monitored. No additional attributes are defined.

Surface Water Node



The *SurfaceWaterNode* abstract class inherits from the *SupplyNode* class, and groups all nodes related to freshwater surface water bodies. Those are reservoirs, both man-made and artificial and river reach nodes. Each subtype *SurfaceWaterNode* is related with a one-to-one relationship with the *SurfaceNodeParameter* table. No additional attributes are defined at this level.

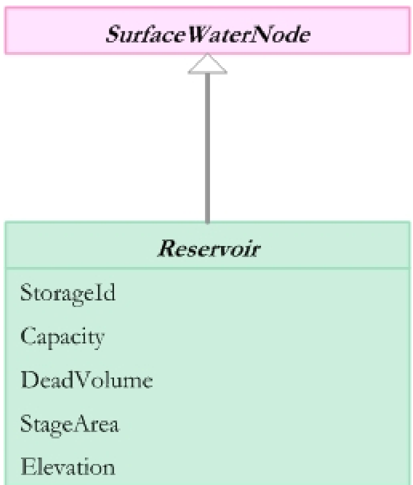
River Reach Node



The *RiverReachNode* feature class inherits from the *SurfaceWaterNode* Class and it represents a physical branch (segment) of a river and its downstream section. The following attributes are defined:

- **CatchmentArea** – Double. The catchment area in m^2 (area of the upstream watershed) for the river reach.
- **DistancefromSpring** – Double. The distance of the RiverReachNode from the spring of the river.
- **RiverId** – String. The ID of the river (MSCode field in the *RiverWaterBody* class) where the river reach node belongs to.

Reservoir



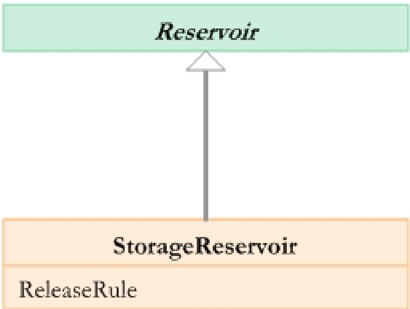
The *Reservoir* class inherits from the *SurfaceWaterNode* Class and represents any surface water storage facility either man-made, such as dams and small reservoirs, or natural lakes. Besides the already inherited attributes, the class defines also the following:

- **StorageID** –The unique identifier of a reservoir in the collection.
- **Capacity** – Integer. The capacity of the reservoir, i.e. maximum volume that can be stored, in m^3.

- **DeadVolume** – Integer. The dead volume of the reservoir in m^3.
- **StageArea** – String. An expression to relate the surface area of the reservoir with its stored volume.
- **Elevation** – String. An expression to relate water height in the reservoir with its stored volume.

Three feature classes inherit from the reservoir class: storage reservoirs, lakes and small reservoirs.

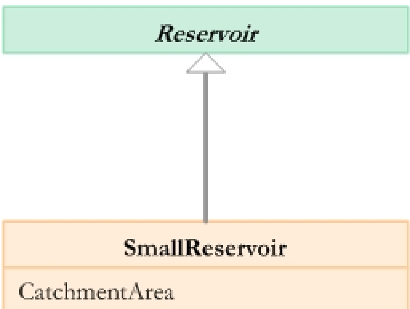
Storage Reservoir



The *StorageReservoir* class a represents man-made reservoir fed by the natural water course of a river water body. The class inherits from the *Reservoir* abstract class and defines one additional attribute:

- **ReleaseRule** – String. An expression which relates downstream releases (not uncontrolled spillage) from the dam to the stored volume.

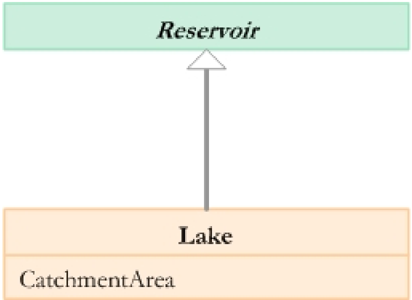
Small Reservoir



The *SmallReservoir* class represents small artificial reservoirs built to collect rainfall or run-off from a catchment area. Modelling requires the definition of one additional attribute to those inherited by *SurfaceWaterNode*, *SupplyNode* and *WaterNode*:

- **CatchmentArea** – Double. The catchment area (in m^2) from which the reservoir is fed.

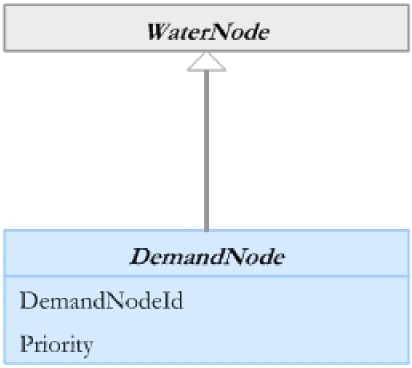
Lake



The *Lake* class stands for natural lakes. Through the *WaterNodeId* attribute, inherited from the *WaterNode* class, lakes, represented as points in the *Water Network* dataset are connected with a one-to-one relationship with the *LakeWaterBody* polygon feature class of the *Water Bodies* dataset. Similarly to the *SmallReservoir* class, one additional attribute is defined:

- **CatchmentArea** – Double. The catchment area (in m^2) from which run-off enters the lake.

5.3 The Demand Node Class



The *DemandNode* abstract class models any type of water use or demanding activity, standing for both consumptive and non-consumptive uses. It inherits from the *WaterNode* abstract class and additionally defines the following attributes:

- **DemandNodeId** – Integer. Unique identifier for a demand node.
- **Priority** – Integer (values from 1 to 99). The attribute describes the rule for allocating water to the particular demand node during the allocation algorithm of the WSM DSS.

The *DemandNode* Class has two child objects: *ConsumptiveDemand* and *Non-ConsumptiveDemand*.

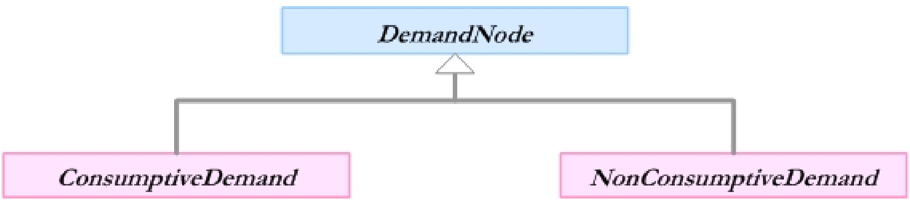
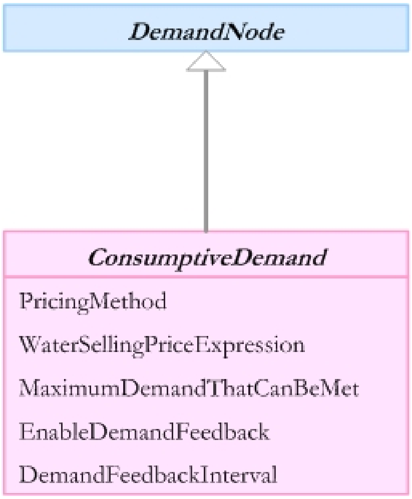


Figure 5. The child objects of the Demand Node class

Consumptive Demand



The *ConsumptiveDemand* abstract class models consumptive water uses such as irrigation, domestic use, animal breeding activities and industrial water use (see the respective subtypes that follow). The object inherits from the *DemandNode* abstract class and defines the following additional attributes:

- **PricingMethod** – Domain attribute: none, volumetric, per area, output pricing, tiered pricing. The field describes the pricing method applied to the particular demand node object.
- **WaterSellingPriceExpression** – String. Expression field describing water selling price as a function of consumption and the elasticity of the related demanding activity to water selling price.
- **MaximumDemandThatCanBeMet** – String. Expression describing a potential limit (quota) to be allocated for a particular water use, as a function of time and/or previous year demands.
- **EnableDemandFeedback** – True/False attribute. The field expresses the behavior (elastic or inelastic) of the consumptive demand to water shortages.
- **DemandFeedbackInterval** – Integer. The field describes the time interval (years) needed for water shortage to have an impact on the pressures and drivers controlling the demand for the particular use (in case that the consumptive demand can be controlled by shortage conditions – i.e. the *EnableDemandFeedback* is set to true).

The child objects of the *ConsumptiveDemand* class are associated with the Activity object class (see section 5.8). The aim of those attributed relationships is to model data regarding the particular water consuming activities encountered within the demand node objects (e.g. washing for domestic use, cooling in industries etc). The activity level and the consumption rate of those activities actually determine the demand of each particular node class, while the pollutant generation per activity unit is used in the estimation of the respective pollution loads.

Finally, the abstract class has 6 child objects, representing each different types of consumptive water use, settlements, tourist sites, irrigation and animal breeding sites, industrial sites and exporting to other regions (Figure 6).

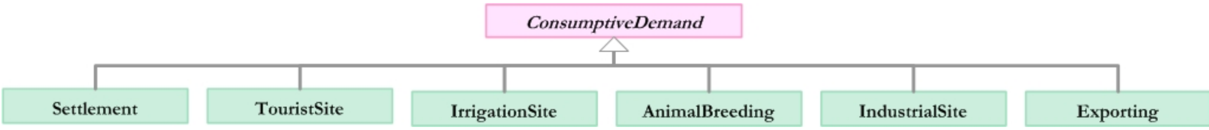
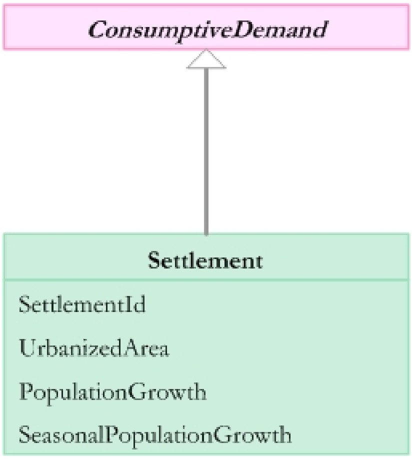


Figure 6. The child objects of the Consumptive Demand abstract class

Settlement

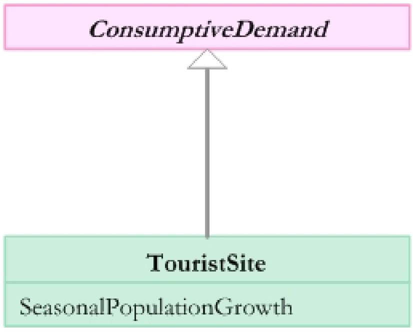


The *Settlement* feature class inherits from the *ConsumptiveDemand* abstract class and describes the civil urban population and infrastructures of a defined area. It can be used to model cities, towns, villages or aggregate urban and tourist population over a larger area e.g. a municipality. The class defines the following attributes:

- **SettlementId** – Integer. The unique identifier of a settlement object.
- **UrbanizedArea** – Double. The area that is covered by the settlements' agglomeration.
- **PopulationGrowth** – String. Expression used for describing the growth of permanent population in the settlement as a function of time.
- **SeasonalPopulationGrowth** – String. Expression used for describing seasonal population growth, in terms of overnight stays, as a function of time.

The feature class is associated with a one-to-many relationship through the *SettlementId* field with the *DistributionNetwork* and *SewageNetwork* object classes.

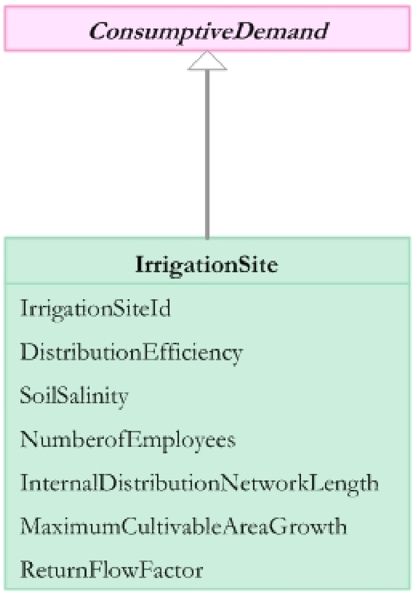
Tourist Site



The *TouristSite* feature class models tourist communities originating a seasonal water demand, and along with Settlement it represents a domestic demand. One additional attribute is defined:

- **SeasonalPopulationGrowth** – String. Expression used for describing seasonal population growth, in terms of overnight stays, as a function of time.

Irrigation Site



The *IrrigationSite* feature class represents the activity of cultivating land either for the survival of land-owners or for commercial purposes. Hereunder, the additional attributes are described:

- **IrrigationSiteId** – Integer. The unique identifier for the irrigation site object.
- **DistributionEfficiency** – Double (values between 0 and 1). The efficiency of water distribution within the irrigation site.
- **SoilSalinity** – Double. Average soil salinity within the irrigation site in mg/l.
- **NumberofEmployees** – Integer. Number of people employed at the irrigation site (land owners and land workers).
- **InternalDistributionNetworkLength** – Integer. The length of the internal distribution network of the irrigation site in meters.

- **MaximumCultivableArea** – String. Expression used for describing the growth of cultivable area for the irrigation site as a function of time. Used to account for land-use change patterns.

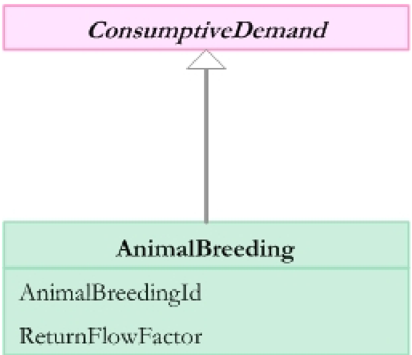
- **ReturnFlowFactor** – Double (values between 0 and 1). Share of water allocated to the irrigation site that is not consumptively used and is consequently rejected to water bodies or infiltrates groundwater bodies.

The *IrrigationSite* class is associated through the *IrrigationSiteId* field with two tables:

- **FieldCrops** and **OrchardCrops** (many-to-many relationships), and

- **AppliedIrrigationMethods** object class (one-to-many relationship).

Animal Breeding



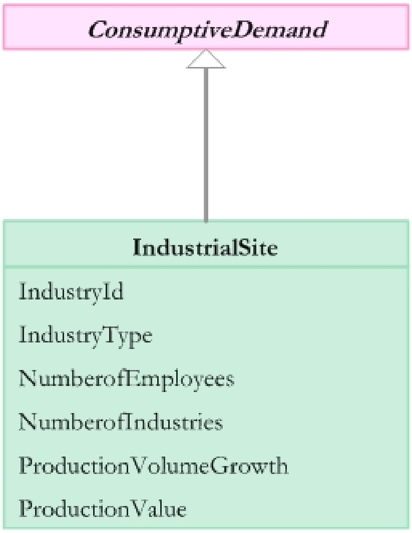
The *AnimalBreeding* feature class inherits from the *ConsumptiveDemand* abstract class and is used to model activities related to animal breeding. The following attributes are defined:

- **AnimalBreedingId** – Integer. The unique identifier for the animal breeding object.

- **ReturnFlowFactor** – Double (values between 0 and 1). Share of water allocated to the animal breeding site that is not consumptively used and is consequently rejected to water bodies.

The *AnimalBreeding* class is associated with the *LivestockType* table through a many-to-many attributed relationship (*AnimalSiteHasLivestock*).

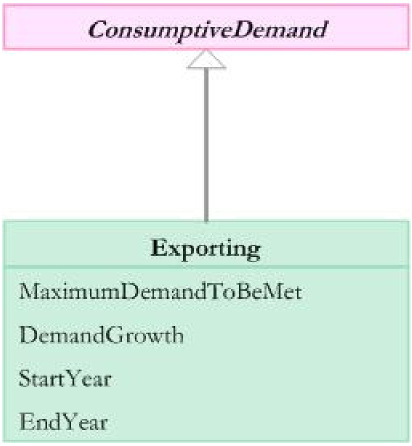
Industrial Site



The *IndustrialSite* feature class describes a unit or a set of units producing or supplying goods, services etc. It is mainly characterised by its field of application: Petrochemical, Electronics, Aerospace, Food and Beverage, Pulp and Paper, Textiles etc and the corresponding production volume. The following attributes are defined:

- **IndustryId** – Integer. The unique identifier for the industrial site object.
- **IndustryType** – String. The type of the industrial unit(s) e.g. Petrochemicals, Textiles, etc.
- **NumberofEmployees** – Integer. Number of people employed at the industrial site.
- **NumberofIndustries** – Integer. Number of industries modeled within the industrial site object (default value of 1).
- **ProductionVolumeGrowth** – String. Expression describing the growth of industrial production over time.
- **ProductionValue** – Double. The value in €/product for the industrial production by the site.

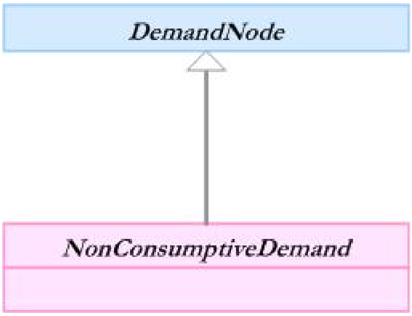
Exporting



The *Exporting* feature class represents the amount of water to be exported to a neighbouring area. The class inherits from the *ConsumptiveDemand* abstract class since return flows are not considered to be re-entering again the case study area. Additional attributes defined by the class are the following:

- **DemandGrowth** – String. Expression to describe the demand growth of the Exporting feature as a function of time.
- **StartYear** – Integer. Year when exporting from the region starts.
- **EndYear** – Integer. Year when exporting from the region ends.

Non Consumptive Demand



NonConsumptiveDemand is an abstract class inheriting from the *DemandNode* abstract class. It is used to model demands which are non-consumptive i.e. where return flows are zero and inflow is equal to the outflow. No additional attributes are defined.

The class has two child objects: *HydroelectricityProduction* and *RiverRequirements.*

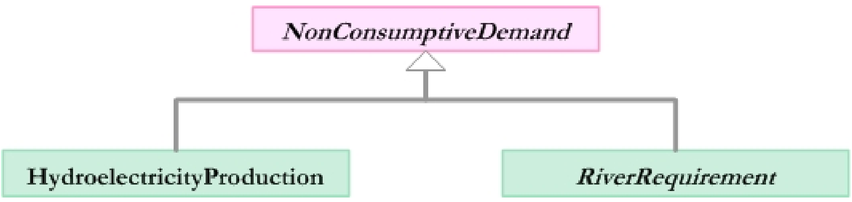
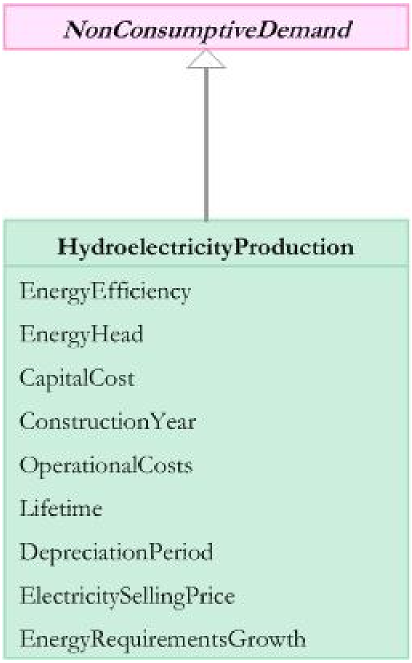


Figure 7. The child objects of the non consumptive demand class

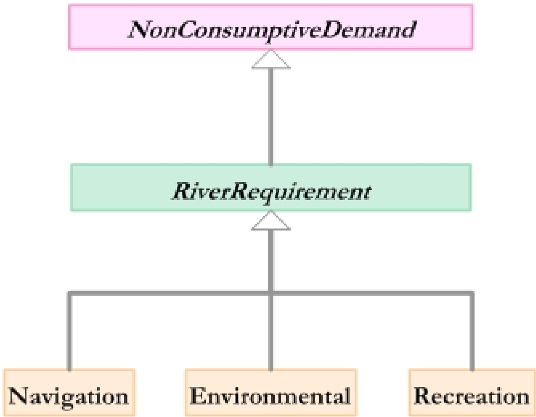
Hydroelectricity Production



The *HydroelectricityProduction* feature class models the amount of water requested by a single plant or a group of plants to generate electricity from falling or fast-flowing water. The class defines the following additional attributes:

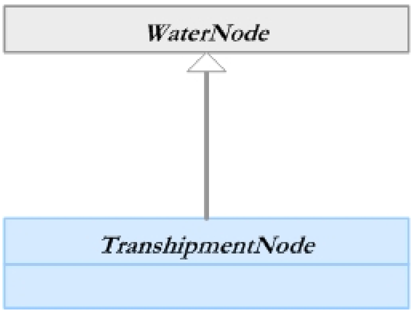
- **EnergyEfficiency** – Double (values between 0 and 1). The overall efficiency of the hydroelectric plant in energy conversion.
- **EnergyHead**– Double. The net energy head available for power generation (in m).
- **ConstructionYear**– Integer. The year when the facility is constructed and starts to operate.
- **CapitalCost** – Integer. Construction cost for the hydroelectric plant.
- **OperationalCosts** – Double. Specific operational costs in €/kWh produced.
- **Lifetime** – Integer. The lifetime of the facility in years.
- **DepreciationPeriod** – Integer. The depreciation period for the capital investment costs.
- **ElectricitySellingPrice** – The price with which energy produced is sold to the local electricity grid or utility.
- **EnergyRequirementsGrowth** – Expression describing the annual energy production required by the facility as a function of time.

River Requirement



RiverRequirement is an abstract class used to model river reach-related water minimum flow requirements. Those consist of minimum monthly flow requirements aiming to guarantee navigation, recreational activities (e.g. fishing), and the preservation of physical and geomorphologic regime of the river, in order to sustain the ecologic value of the aquatic ecosystem. Therefore, three subtypes have been defined, the *Environmental*, *Recreation* and *Navigation* classes.

5.4 The Transhipment Node Class



The abstract *TranshipmentNode* class is used to characterize points in the water system that do not act neither as supply sources nor as water demanding activities. No additional attributes are defined at this stage. One important network class related to water quality, namely *TreatmentNode* is defined as a child object (modelling of treatment plants is analysed in section 5.5). The other class that inherits from the *TranshipmentNode* object is the *NetworkReservoir.*

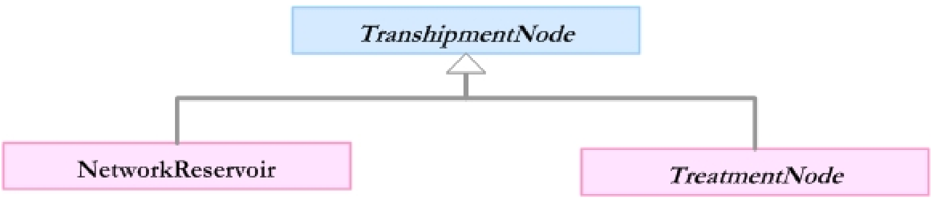
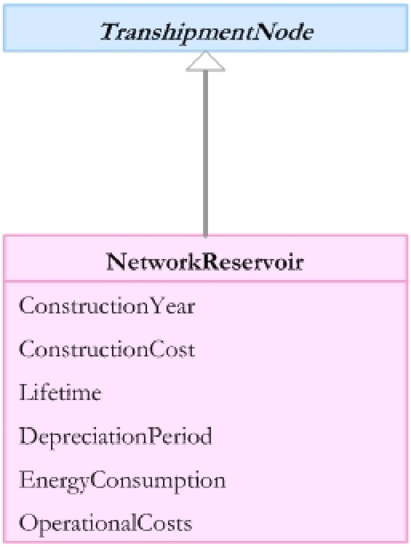


Figure 8. The child objects of the transhipment node class

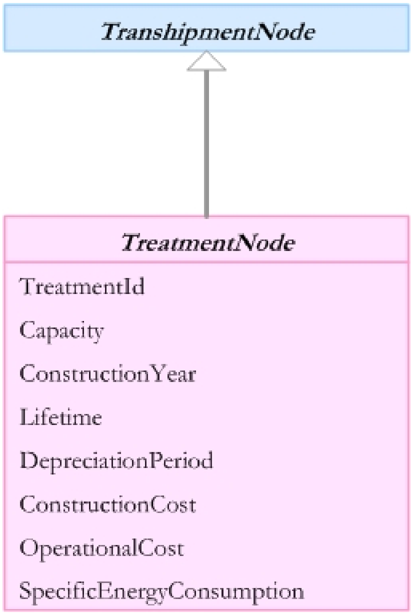
Network Reservoir



The *NetworkReservoir* class inherits from the *TranshipmentNode* abstract class and represents a physical reservoir of very small capacity, around 2000 m^3, which is used to serve the domestic needs of settlements, tourist sites etc. The class defines the following additional attributes:

- **ConstructionYear** – Integer. The construction year of the network reservoir.
- **ConstructionCost** – Integer. The construction cost expressed in €.
- **Lifetime** – Integer. The lifetime of the reservoir in years.
- **DepreciationPeriod** – Integer. The depreciation period for the capital cost for the network reservoir construction.
- **EnergyConsumption** – Double. Required specific energy consumption in kWh/m^3 of inflow.
- **OperationalCosts** – Double. Specific operation and maintenance costs in €/m^3 of network reservoir inflow.

5.5 The Treatment Node Class



The abstract *TreatmentNode* feature class is used to model treatment plants, i.e. points in the network where water quality is modified through a particular treatment process. The class inherits from the more general abstract class for transhipment nodes. Additionally to all inherited attributes from the previous hierarchy levels, the class defines the following attributes:

- **TreatmentId** – Integer. The unique identified of a treatment node in the collection of treatment plant objects.
- **Capacity** – Integer. The capacity of the water treatment plant expressed in m^3 that can be treated on a daily basis. For the Desalination class, capacity is expressed in volume of water produced.
- **ConstructionYear** – Integer. The construction year for the treatment plant.
- **Lifetime** – Integer (default value of 15 years). The lifetime (years) of the treatment plant.
- **DepreciationPeriod** – Integer (default value of 15 years). The depreciation period for the capital costs.
- **ConstructionCost** – Integer. Treatment plant total construction cost.
- **OperationalCost** – Double. The specific operation and maintenance costs expressed in €/m^3 treated (or produced in case of desalination).
- **SpecificEnergyConsumption** – Double. The specific energy consumption of the plant, expressed in kWh\m^3 treated (or produced in case of desalination).

Three subtypes have been so far included in the WSM Data Model: *Desalination*, *DrinkingWaterTreatmentPlant* and *WastewaterTreatmentPlant*.

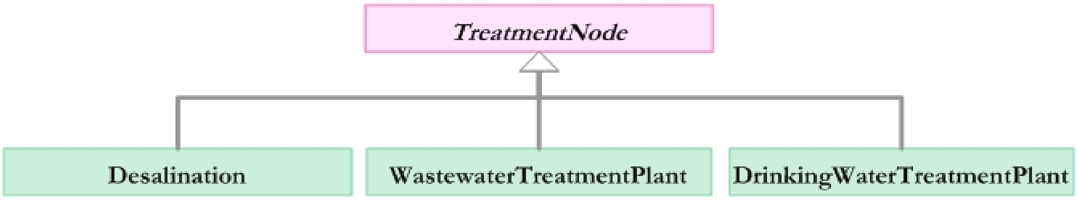
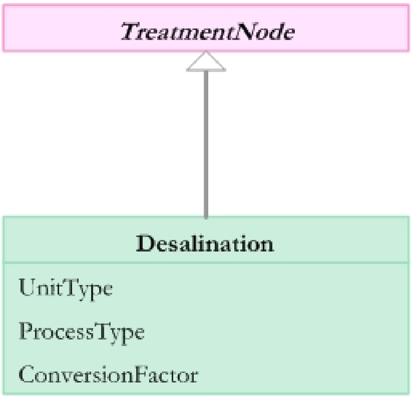


Figure 9. The child objects of the Treatment Node class

The subtypes of the class are associated with a many-to-many attributed relationship with the Quality Variables object class, which is used to describe the particular details of the treatment process, such as pollutant removal rates and required concentrations for drinking water or effluent disposal in case of wastewater treatment plants.

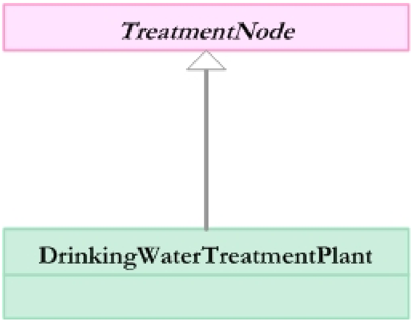
Desalination



The *Desalination* class represents a plant removing dissolved salts from seawater, brackish waters of inland seas or highly mineralized groundwater. Additional attributes defined by the class are the following:

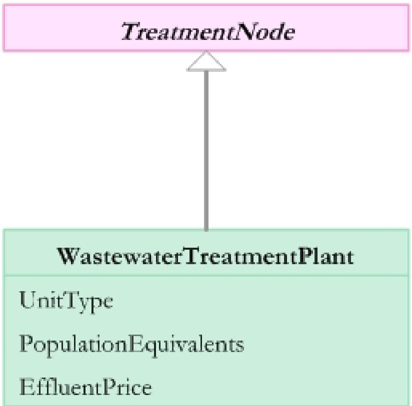
- **UnitType** – Domain Attribute (Brackish or Sea water desalination). The type of the desalination plant according to the type of the inlet.
- **ProcessType** – Domain Attribute (Reverse Osmosis, Electrodialysis, Multi-stage Flash, MultiEffect Distillation, Vapour Compression). The type of the desalting process.
- **ConversionFactor** – Double (values between 0 and 1). The conversion ratio of feed water to potable water by the treatment process.

Drinking Water Treatment Plant



The *DrinkingWaterTreatmentPlant* class inherits from the abstract *TreatmentNode* class and models a plant treating water in order to make it safe and acceptable for human use. No additional attributes are defined.

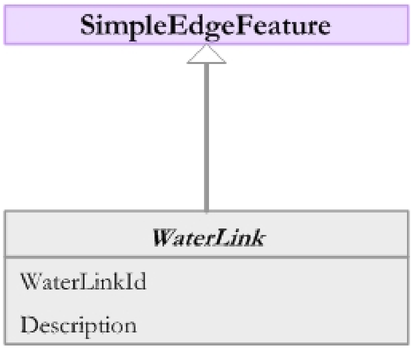
Wastewater Treatment Plant



The *WastewaterTreatmentPlant* feature class node represents a plant treating water in order to remove or at least abate pollutants' concentration before water is re-used or discharged into a body of surface water. The class defines the following additional attributes:

- **UnitType** – Domain attribute (Primary, Secondary, Tertiary). The type of the treatment process, strictly related with the removal of pollutants and the pollution abatement efficiency of the wastewater treatment plant.
- **PopulationEquivalents** – Integer. The capacity of the wastewater treatment plant expressed in maximum population that can be served on a daily basis.
- **EffluentPrice** – Double. The charged price of effluent from the wastewater treatment plant, in case that water treated is reused in irrigation activities.

5.6 The Water Link class



The *WaterLink* abstract class symbolizes any connection in the water network between water nodes. It inherits from the *SimpleEdgeFeature* of the Arc Info Object Model. Edges logically are defined as network features that have a length through which commodity flows. As a consequence all water nodes occur at the intersection of two or more edges. An edge should always start and end at network junctions or water nodes. On a first conceptualisation level, the *WaterLink* class defines the following attributes:

- **WaterLinkId** –Integer**.** The Id attribute acts as the unique identifier of a water link in the network.
- **Description** – String. A descriptive text for the link.

Water link objects are classified in four categories according to the connectivity rules of the network and the particular modelling requirements of the DSS:

- **Supply Links**, conveying water from supply sources to demand nodes.
- **Groundwater Interaction Links**, representing the natural interaction between surface and groundwater bodies.
- **Return Flow Links**, conveying return flows from consumptive demand uses to receptor bodies (surface or groundwater) or wastewater treatment plants.
- **River Links**, (or segments) representing the natural course of a river water body, interconnecting river reach nodes, storage reservoirs and minimum flow requirement objects.

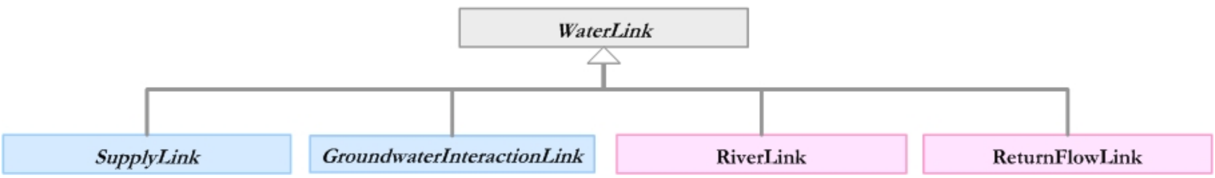
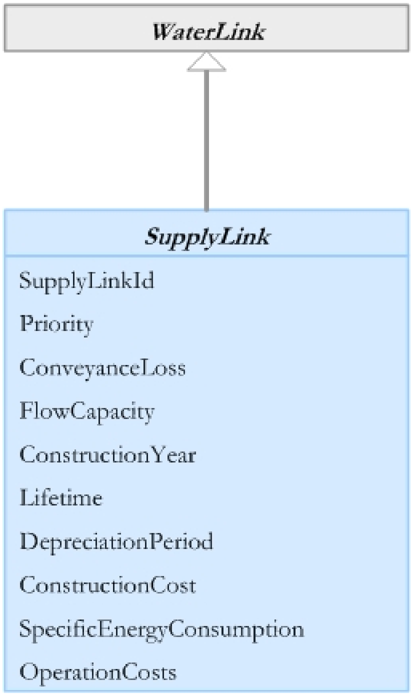


Figure 10. The child objects of the Water Link class

Supply Link



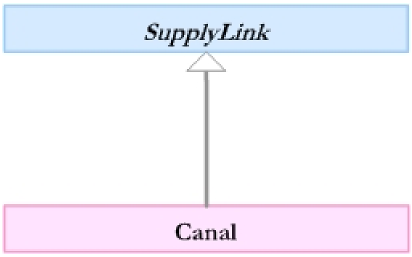
The *SupplyLink* abstract class represents edges transferring water (either fresh or treated wastewater) intended to be used by consumptive demand node objects. One fundamental attribute is the capacity of the link, which represents the maximum monthly flows allowed. In addition to the attributes inherited by the parent *WaterLink* feature class, the class further defines the following fields:

- **SupplyLinkId** – Integer. Unique identifier for a supply link object.
- **Priority** – Integer (values from 1 to 99). The priority with which water is allocated from the start node of the link to the end node.
- **Conveyance Loss** – Double (values between 0 and 1). Share of inflow that is lost due to leakages.
- **FlowCapacity** – Double. The maximum flow in m^3/month that can traverse the supply link.
- **ConstructionYear** – Integer. The year the link is constructed.
- **Lifetime** – Integer. Lifetime of the supply link in years.
- **DepreciationPeriod** – Integer. Depreciation period for the capital cost for link construction.
- **ConstructionCost** – Integer. Link construction cost.
- **SpecificEnergyConsumption** – Double. Specific pumping requirements in €/m^3 of inflow.

- **OperationCosts** – Double. Specific operation and maintenance costs in €/m^3 of link inflow.

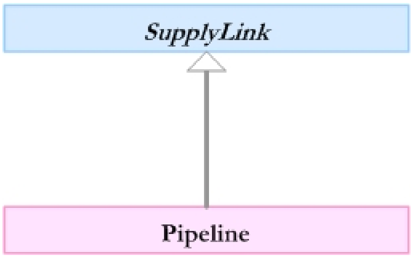
Two child objects are derived from the *SupplyLink* abstract class: *Canals* and *Pipelines*.

Canal



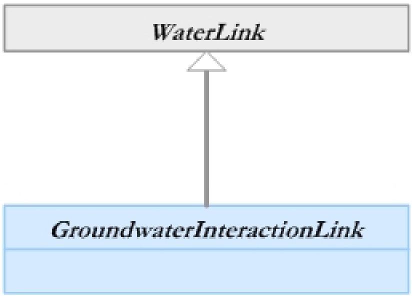
The *Canal* class inherits from the *SupplyLink* abstract class and is used to model supply links which transfer water through an open-air artificial waterway. It can be used to connect river reach nodes to agricultural sites. No additional attributes are defined by the class.

Pipeline



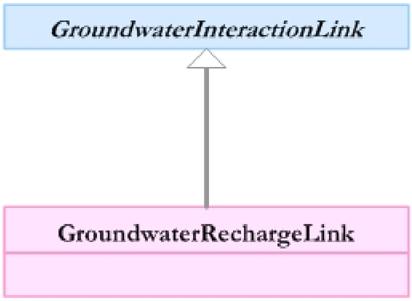
The *Pipeline* class also inherits from the *SupplyLink* abstract class and is used to represent long pressure pipes conveying water from a variety of water sources to demand nodes. Like *Canals*, the class does not define any additional attributes.

Groundwater Interaction Link



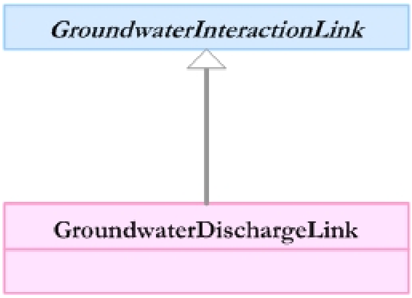
The *GroundwaterInteractionLink* abstract class stands for links `that model the interaction of groundwater with surface water bodies. Two subtypes are defined, each with different connectivity rules: groundwater discharge and groundwater recharge links.

Groundwater Recharge Link



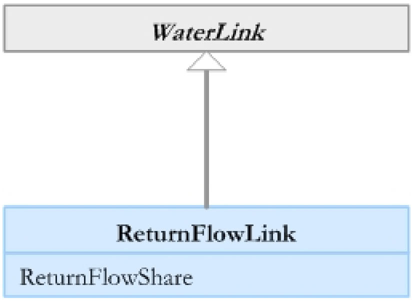
Groundwater recharge links are used to model the natural process of aquifer recharge from surface water bodies, such as lakes, storage reservoirs and river water bodies. No additional attributes are defined.

Groundwater Discharge Link



Groundwater discharge links are used to model the natural process of aquifer discharge to surface water bodies and the sea.

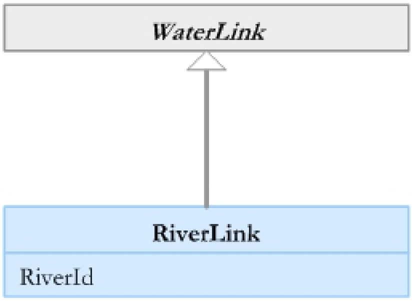
Return Flow Link



The *ReturnFlowLink* class conceptualises edges which transfer return flows from demand sites either at wastewater treatment plants or at water bodies who act as effluent receptors. One additional attribute is defined, for determining the share of return flow that is allocated to the particular return flow link object:

- **ReturnFlowShare** – Double (values between 0 and 1). Share of return flow that is allocated to a return flow link object.

River Link



The last feature class included in the Water Network Dataset is the *RiverLink*. Since the particular river segments are modeled as points, (see *RiverReachNode* at section 5.2), the only functionality of River Link Feature class is to maintain network connectivity along the natural course of a river.

5.7 Connectivity Rules

A fundamental concept in a water resource network, as with every type of utility network, is the maintenance of appropriate connectivity rules which specify what types of junctions (water nodes) can be connected to each other and with which type of edge (water link). In general, network connectivity rules constrain the type of network features that may be connected to one another and the number of features of any particular type that can be connected to features of another type. By establishing these rules along with other rules such as attribute domains, the integrity of the network data in the database can be maintained.

In ArcGIS there are two types of connectivity rules: edge–junction rules and edge–edge rules. An edge–junction rule is a connectivity rule that establishes that an edge of type A may connect to a junction of type B. An edge–edge rule is a connectivity rule that establishes that an edge of type A may connect to an edge of type B through a set of junctions. Edge–edge rules always involve a set of junctions.

The WSM Data Model specifies a number of connectivity rules, in order to ensure the proper modeling of a water resource system and its correct simulation by the WSM Decision Support System. Modeling requires that some types of edges (water links) have a specific type of start or end junction or both. For example, groundwater recharge links can only originate from surface water nodes (reservoirs and river reaches) and should end only at renewable groundwater nodes. Additionally, junctions modeling particular types of water sources such as non-renewable (fossil groundwater) or importing from neighboring regions cannot have incoming edges of any type. To ensure therefore the integrity of network data within the database, network connectivity is modeled within the WSM Decision Support System, with a set of rules that specify which type of junction can be connected to which other junction type and with what type of edge. Those rules are outlined in Tables 1 (for nodes) and 2 (for links).

Table 1. Node connectivity restrictions

Node Type	Restriction	
	In coming Links	**Out coming Links**
Fossil Groundwater	None	Canal

Node Type	Restriction	
	In coming Links	Out coming Links
Importing		Pipeline
Renewable groundwater	No restriction	Canal Pipeline Groundwater discharge link
Reservoir	No restriction	Canal Pipeline Groundwater recharge link
Coastal zone	Return flow link Groundwater discharge link	None
Settlement Tourist Site Industrial Site Irrigation Site Animal Breeding	Canal Pipeline	Return flow link
Wastewater Treatment Plant	Return flow link	Return flow link Canal/pipeline if the end node is an irrigation site object
Drinking water treatment plant Desalination	Canal Pipeline	Canal Pipeline

Table 2. Link connectivity restrictions

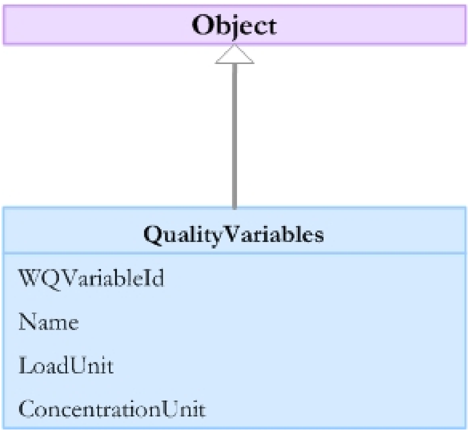
Link Type	Start Node	End Node
Groundwater Recharge Links	Storage Reservoir Small Reservoir Lake	Renewable groundwater node
Groundwater Discharge Link	Renewable groundwater	Storage Reservoir Small Reservoir Lake Renewable groundwater Coastal Zone
River Links	River Reach Node Storage reservoir Hydroelectricity River requirement	River Reach Node, Storage reservoir Hydroelectricity River requirement

5.8 Non-Spatial Objects

The spatial data of the Network Dataset are not by themselves adequate to describe the structure of the water resources system in terms of available resources, demand. Additional data are stored in tabular form in order to accommodate the additional information. Those are related with the spatial objects with one-to-one, one-to-many and many-to-many relationships. In the WSM Data model a distinction is made between data modelled as Time Series, which are presented in Chapter 7 and simple non-spatial attributes. All "simple" data in the Arc Info Data Model inherit from the general abstract class called *Object*.

General

Quality Variables



The *QualityVariables* class hosts all quality variables that are currently monitored and simulated by the WSM Decision Support System. The attributes defined by the class are the following:

- **WQVariableId** – Integer. The unique identifier of a quality variable.
- **Name** – String. The name of the quality variable.
- **LoadUnit** – String. Custom units used for expressing the quality variable load.
- **ConcentrationUnit** – String. Custom units for expressing the quality variable concentration.
- **Charge** – Double. Pollution charge in €/load unit. Pollution charges are used for the estimation of environmental costs in cases of effluent disposal.

The *QualityVariables* class participates in a number of relationships with other object and feature classes. Those are:

- Many-to-many attributed relationship with child objects of the Supply Node feature class (see Ecosystem Vulnerability in section 5.9).
- Many-to-many attributed relationship with the child objects of the Treatment Node feature class (see Drinking Water and Wastewater Treatment Processes in section 5.9)

- Many-to-many relationship with the attributed relationships between the child objects of the ConsumptiveDemand class and the Activity object class (see Pollutant Generation per Activity unit in section 5.9)

Activity

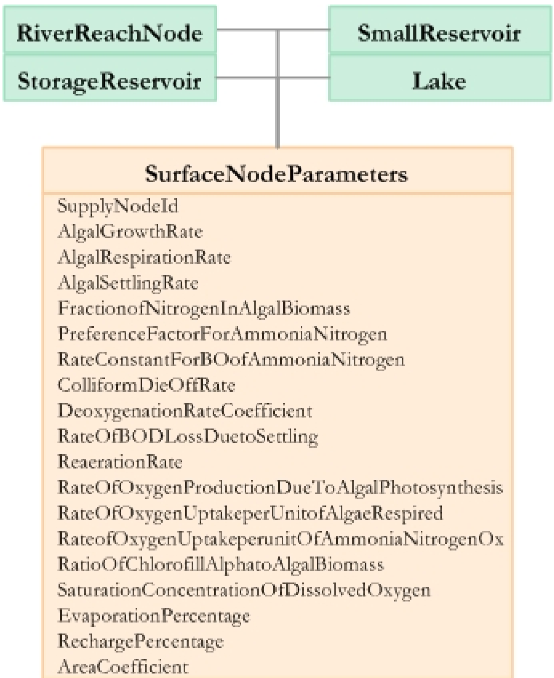
The *Activity* class groups all types of consumptive demand activities. The class by itself contains some default values which can then be modified for each DemandNode of the Water Network dataset: The class defines the following attributes:

- **ActivityId**– Integer. The unique identifier of the activity object.
- **Description** – String.
- **Unit** – String. Custom units used for expressing the activity level.
- **DefaultElasticity** – Double. The default elasticity of the demand activity towards the water selling price.

The *Activity* class is associated through many-to-many attributed relationships with the child objects of the *ConsumptiveDemandNode* abstract class.

Supply Related Data

Surface Node Parameters



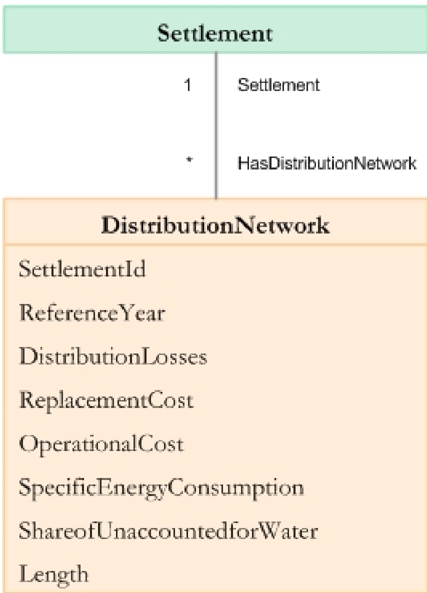
The *SurfaceNodeParameters* object class mainly stores information required for performing quality estimations for surface water bodies. It also includes some attributes for describing seepage losses and evaporation from reservoirs, and groundwater recharge from river reaches, as well as the vulnerability of the surface water body to overabstraction. The object class is connected with a simple, one-to-one relationship with the classes inheriting from the *SurfaceWaterNode* class (storage reservoirs, lakes, small reservoirs and river reach nodes). The attributes defined by the class are outlined below:

- **SupplyNodeId** – Integer. The foreign key of the one-to-one relationship with the *SurfaceWaterNode* Class.
- **AlgalGrowthRate** – Double. Algal growth rate (days^{-1}). Typical values range from 1.0 – 3.0 days^{-1}. The default value is equal to 2 days^{-1}.
- **AlgalRespirationRate** – Double. Algal respiration rate (days^{-1}). Typical values range from 0.05 – 0.5 days^{-1}. The default value is equal to 2.53 days^{-1}.
- **AlgalSettlingRate** – Double. Algal settling rate (day^{-1}). Typical values range from 0.5 – 6.0 days^{-1}. The default value is equal to 3.25 days^{-1}.
- **FractionofNitrogeninAlgalBiomass** – Double. Fraction of algal biomass that is nitrogen (dimensionless). Typical values range from 0.07 – 0.09. The default value is equal to 0.08.
- **PreferenceFactorForAmmoniaNitrogen** – Double. Preference factor for ammonia nitrogen (dimensionless). Typical values range from 0.0-1.0. The default value is equal to 0.5.
- **RateConstantforBOofAmmoniaNitrogen** – Double. Rate constant for the biological oxidation of ammonia nitrogen. Typical values range from 0.1-1.0. The default value is equal to 0.55 days^{-1}.
- **ColiformDieOffRate** – Double. Coliform die-off rate (days^{-1}). Typical values range from 0.05-4 days^{-1}. The default value is set to 2 days^{-1}
- **DeoxygenationRateCoefficient** – Double. Deoxygenation rate coefficient (days^{-1}). Typical values range from 0.02-3.4 days^{-1}. The default value is set to 1.7 days^{-1}.
- **ReaerationRate** – Double. Re-aeration rate (day^{-1}). Typical values range from 0.0-100 days^{-1}. The default value is set to 50 days^{-1}.
- **RateOfOxygenProductionDueToAlgalPhotosynthesis** – Double. Rate of oxygen production per unit of algal photosynthesis (dimensionless). Typical values range from 1.4 – 1.8. The default value is set to 1.6.
- **RateofOxygenProductionperUnitofAlgaeRespired** – Double. Rate of oxygen production per unit of algae respired (day^{-1}). Typical values range from 1.6 – 2.3 day^{-1}. The default value is equal to 2.
- **RateofOxygenUptakeperUnitOfAmmoniaNitrogenOx** – Double. Rate of oxygen uptake per unit of ammonia nitrogen oxidation (dimensionless). Typical values range from 3.0-4.0. The default value is set to 3.5.

- **RatioOfChlorofillAlphainAlgalBiomass** – Double. Ratio of chlorophyll alpha in algal biomass (μg-Chla/mg-A). Typical values range from 10 -100 μg-Chla/mg-A. The default value is set to 55 μg-Chla/mg-A.
- **SaturationConcentrationOfDissolvedOxygen** – Double. Saturation Concentration of Dissolved Oxygen (mg/l).
- **EvaporationPercentage** – Double (values between 0 and 1). Water evaporated from the surface of the reservoir or the river reach as a fraction of reference evapotranspiration over the reservoir (or watershed) area.
- **RechargePercentage** – Double. Share of flow or storage that recharges connected aquifers through groundwater recharge links if such exist.
- **AreaCoefficient** – Double. Coefficient expressing the vulnerability of the surface water body to water abstraction.

Demand Related Data

Distribution Network

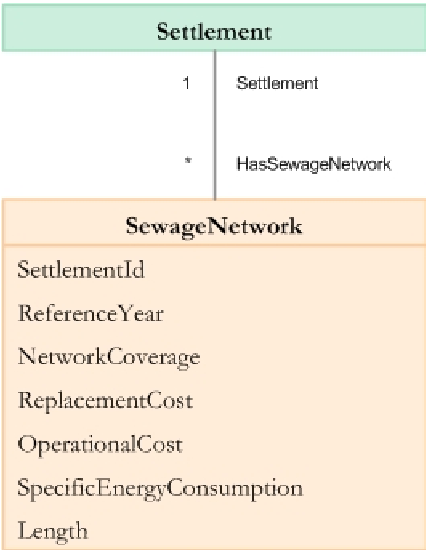


The *DistributionNetwork* object class is used to define the state of the internal water distribution network of settlement objects for the case study analysis period in terms of losses, unaccounted for water and operational costs. The class is therefore associated with the Settlement object through a one-to-many relationship. A distribution network object incorporates the following attributes:

- **SettlementId** – Integer. The foreign key of the origin class (Settlement feature class).
- **ReferenceYear** – Integer. The year that the attributes below refer to.
- **DistributionLosses** – Double (values between 0 and 1). Share of water delivered that is lost due to leakages.

- **ShareofUnaccountedforWater** – Double (values between 0 and 1). Share of water delivered that is supplied (or lost) to unaccounted for uses, i.e. unmetered water.
- **Length** – Integer. The length of the internal distribution network, usually in meters.
- **ReplacementCost** – Integer. Network replacement costs in €.
- **OperationalCost** – Double. Specific network operation and maintenance costs expressed in €/m^3 distributed.
- **SpecificEnergyConsumption** – Double. Specific energy consumption for distributing water allocated to the settlement node in €/m^3 delivered.

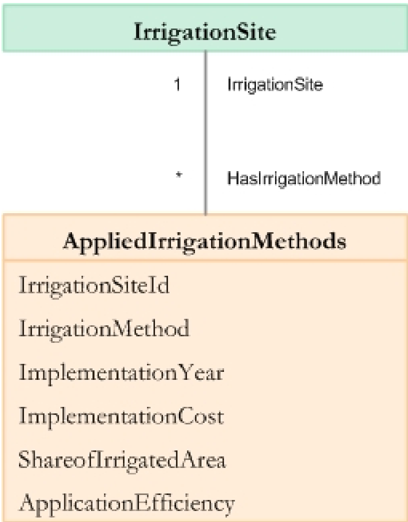
Sewage Network



The *SewageNetwork* object class is used to define the state of the internal sewage network of settlement objects for the case study analysis period in terms people connected, length and operational costs. The class is therefore associated with the Settlement object through a one-to-many relationship. A sewage network object incorporates the following attributes:

- **SettlementId** – Integer. The foreign key of the origin class (Settlement feature class).
- **ReferenceYear** – Integer. The year that attributes refer to.
- **NetworkCoverage** – Double (values between 0 and 1). Population share that is connected to the sewage network at the reference year.
- **ReplacementCost** – Integer. Network replacement costs in €.
- **OperationalCost** – Double. Specific network operation and maintenance costs expressed in €/m^3 of return flow collected.
- **SpecificEnergyConsumption** – Double. Specific energy consumption for collecting sewerage in €/m^3 collected.
- **Length** – Integer. The length of sewage network, usually in meters.

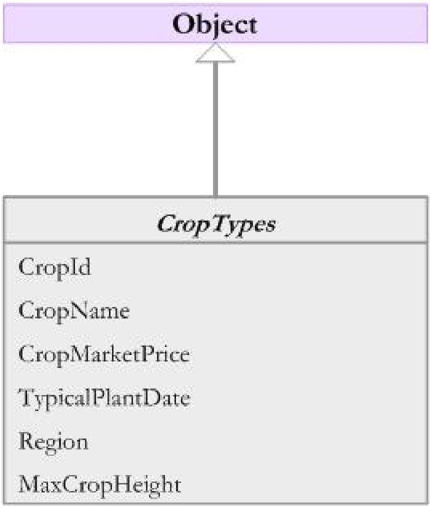
Applied Irrigation Methods



The *AppliedIrrigationMethods* object class is used to define the irrigation methods that are applied in a particular irrigation site and describe them in terms of efficiency and application costs (if the objects of the class refer to proposed irrigation improvement measures). The class is therefore associated with the *IrrigationSite* feature class through a one-to-many relationship. The object class defines the following attributes:

- **IrrigationSiteId** – Integer. The foreign key from the origin class (IrrigationSite feature class).
- **IrrigationMethod** – Domain attribute (Flood, Furrow, Sprinkler, Drip). The type of the irrigation method applied.
- **ImplementationYear** – Integer. The year when the method is implemented at the irrigation site.
- **ImplementationCost**– Double. The cost for applying the irrigation method at its implementation year, in €/m^2.
- **ShareOfIrrigatedArea**– Double (values between 0 and 1). Share of the total irrigated area where the method is applied.
- **ApplicationEfficiency** – Double (values between 0 and 1). The efficiency of the irrigation method (the fraction of water delivered to the field that actually reaches the crop).

Crop Types

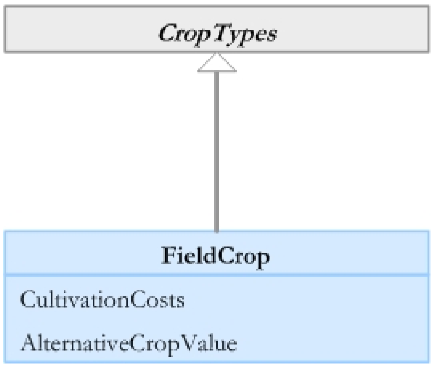


The *CropTypes* abstract class is used to host information on economically important crop types who are irrigated in the case study region. A crop type object incorporates the following attributes:

- **CropID** – Integer. The unique identifier of a crop object in the database.
- **CropName** – String. Crop Name
- **Region** – String. Region where the typical planting date refers to (see below).
- **TypicalPlantingDate** – String. Typical planting date for the crop.
- **MaximumCropHeight** – Double. The maximum height of the crop at the end of the development stage, in meters.
- **CropMarketValue** – Double. The expected market price of the crop net of non-water variable costs (in €/m^2).

Crop types are subdivided into two object classes, *FieldCrops* and *OrchardCrops*. Each of those child classes is associated with two other object classes (one-to-one relationships) which host information about the computation of the computation of irrigation water requirements. Those two object classes contain the relevant data for meeting the modelling requirements of the Complex and Simplified irrigation models of the WSM Decision Support System. Finally, both child objects are associated with the *IrrigationSite* feature class through many-to-many attributed relationship classes, namely *IrrigationSiteHasFieldCrops* and *IrrigationSiteHasOrchardCrops*, analysed in paragraph 5.9.

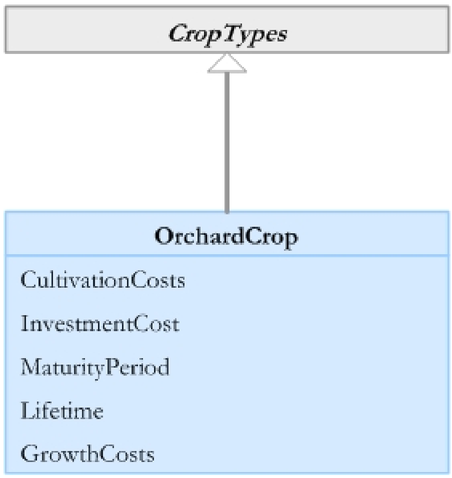
Field Crops



The *FieldCrops* object class inherits from the *CropTypes* abstract class and additionally defines the following attributes:

- **CultivationCosts** – Double. Cultivation costs per m^2 of irrigated area, independent of the level of yield for the field crop.
- **AlternativeCropValue** – Double. The alternative value of land. Commonly measured by the net profits, in €/m^2, of the best alternative non-irrigated crop, like rain-fed wheat).

Orchard Crops

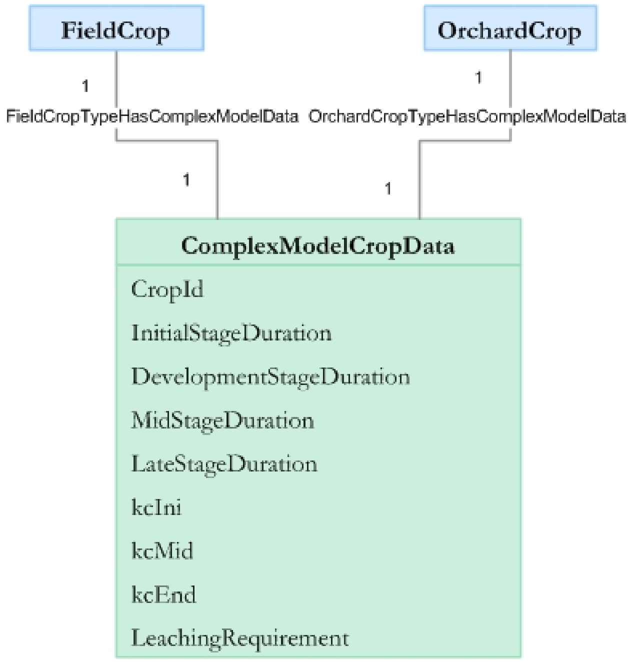


The *OrchardCrop* object class is the other subtype of the *CropTypes* object class and is used to model tree plantations. The following attributes are defined:

- **CultivationCosts** – Double. Cultivation costs per m^2 of irrigated area, independent of the level of yield for the orchard.
- **InvestmentCost** – Double. Investment costs, measured in €/ m^2 (plants, cultivation, new irrigation system, etc) required to grow a new orchard.
- **MaturityPeriod** – Integer. The period (in years) required for a newly planted orchard to mature –duration of the growing stage.
- **Lifetime** – Integer. Total lifetime of an orchard.

- **GrowthCosts** – Double. Costs (excluding water costs) associated with the operation of the premature grove.

Complex Model Crop Data

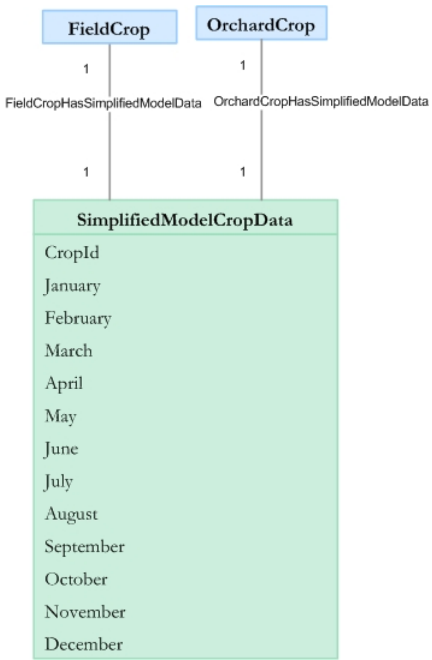


The *ComplexModelCropData* object class hosts data required to compute irrigation demand driven by meteorological factors, such as precipitation and by reference evapotranspiration. Therefore, the object class defines for each crop the length of each growth stage, the related crop coefficient and the leaching requirements. The attributes defined are the following:

- **CropID** – Integer. The foreign key of the origin classes.
- **InitialStageDuration** – Integer. The duration of the initial crop stage running from planting date to approximately 10% ground cover (in days).
- **DevelopmentStageDuration** – Integer. The duration of the crop development stage running from 10% ground cover to effective full cover (in days). Effective full cover for many crops occurs at the initiation of flowering.
- **MidStageDuration** – Integer. The duration of the crop mid-season stage running from effective full cover to the start of maturity (in days). The start of maturity is often indicated by the beginning of the ageing, yellowing or senescence of leaves, leaf drop, or the browning of fruit.
- **LateStageDuration** – Integer. The duration of the crop late season stage running from the start of maturity to harvest or full senescence (in days).
- **kcIni** – Double. The crop coefficient for the initial crop stage.
- **kcMid** – Double. The crop coefficient for the mid season stage.
- **kcEnd** – Double. The crop coefficient for the crop late season stage.

- **LeachingRequirement** – Double. Crop leaching requirements in mm.

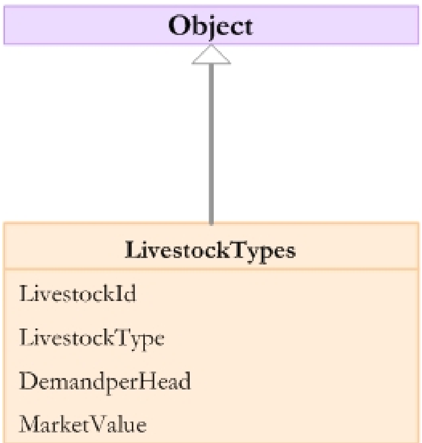
Simplified Model Crop Data



The *SimplifiedModelCropData* object class hosts data required to compute irrigation demand on the basis of simple data on monthly water requirements for each crop. The attributes defined are the following:

- **CropID** – Integer. The foreign key of the origin class (*CropTypes* object class).
- **January-December** – Double. Monthly water requirements for the particular crop in mm for each month respectively.

Livestock Types



The *LivestockTypes* object class is used to host information on animal types whose breeding is water consumptive and economically significant to the case study region. A livestock type object has the following attributes:

- **LivestockId** – Integer. The unique identifier of the animal type.
- **LivestockType** – String. A descriptive text for the animal type e.g. goats, cattle, sheep etc.
- **DemandperHead** – Double. Monthly water requirements for the livestock type in m^3/head/month.
- **MarketValue** – Double. The net value associated with the breeding of the particular animal type, free of water charges. Usually expressed in terms of €/head/yr.

The *LivestockTypes* object class is associated with the *AnimalBreeding* feature class through a many-to-many attributed relationship class, namely *AnimalSiteHasLivestock*.

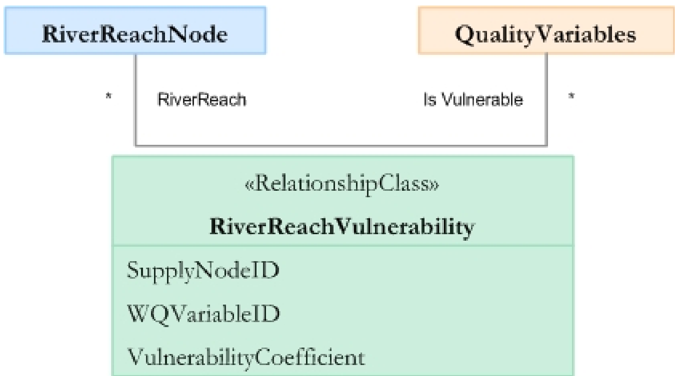
5.9 Attributed Relationship Classes

Many-to-many relationships require a new table in the database to store the foreign keys from the origin and destination classes to make the relationship. In the Data Model this table usually has other fields to store the attributes of the relationship itself which are not attributed to either the origin or destination class. A typical example is the relationship modelling the vulnerability of an ecosystem to discharges of effluents. This relationship, besides an identifier for the quality variable object and the supply node object also requires an additional attribute, expressing the vulnerability of the water body to the particular discharged loads.

Unfortunately, although this is conceptually correct, abstract classes – which eventually do not become an integral part of the database – cannot participate in relationship classes. The following paragraphs however analyse the modelled relationship classes on the abstract class level. The UML Diagram presented is an example of the relationship between one of the subtypes of the abstract class and the related object class.

Supply Node Relationships

Ecosystem Vulnerability



The *EcosystemVulnerability* relationship classes models the vulnerability of an ecosystem (node acting as a receptor body) to the discharge of polluting effluents. The class defines the many-to-many relationship between the Quality Variables object class and objects inheriting from Supply Node abstract class. Those are:

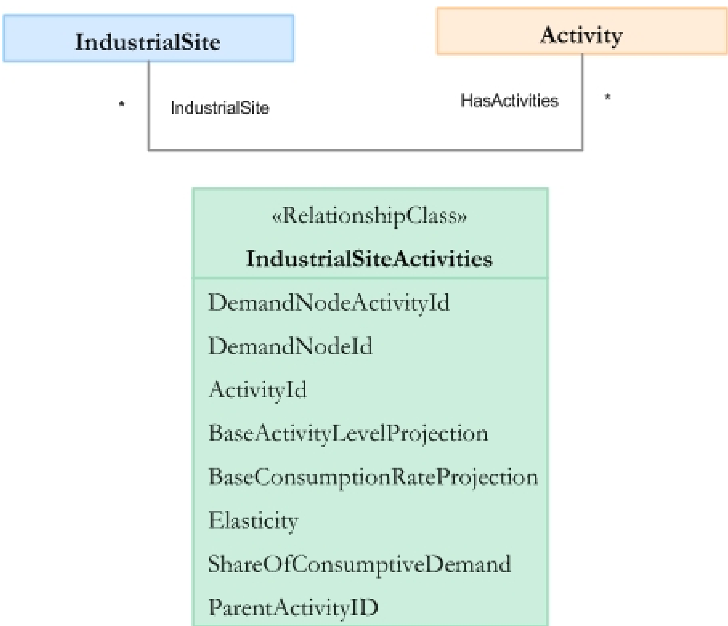
- *CoastalZone* (*CoastalZoneVulnerability* relationship class)
- *RenewableGroundwater* (*RenewableGroundwaterVulnerability* relationship class)
- *RiverReachNode* (*RiverReachVulnerability* relationship class, which is the one presented in the UML Diagram)
- *StorageReservoir* (*StorageReservoirVulnerability* relationship class)
- *Lake* (*LakeVulnerability* relationship class)
- *SmallReservoir* (*SmallResVulnerability* relationship class)

The relationship class defines the following attributes:

- **SupplyNodeId** – Integer. The foreign key of the origin class as defined above.
- **WQVariableId** – Integer. The foreign key of the destination class (QualityVariables)
- **VulnerabilityCoefficient** – Double. The coefficient describing the vulnerability of the water body to the quality variable loads.

Demand Node Relationships

Demand Node Activities



The *DemandNodeActivities* is a relationship class which associates the *Activity* class with the classes inheriting from the *ConsumptiveDemandNode* abstract class, namely:

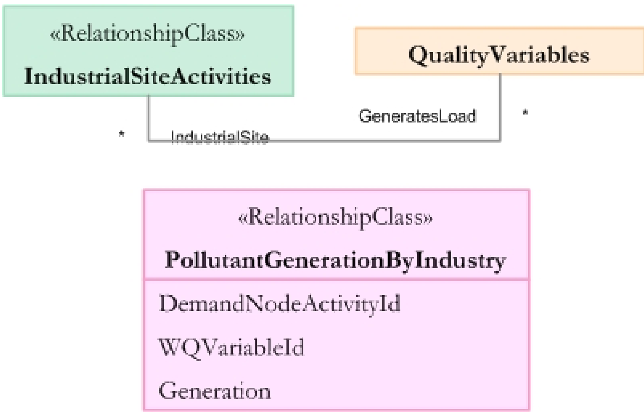
- *Settlement* (*SettlementNodeActivities* relationship class)
- *TouristSite* (*TouristSiteActivities* relationship class)

- *IrrigationSite* (*IrrigationSiteActivities* relationship class)
- *AnimalBreeding* (*AnimalBreedingActivities* relationship class)
- *IndustrialSite* (*IndustrialSiteActivities* relationship class)

The class is used to define the particular characteristics of activities as they appear in the different demand node objects. For this purpose the following attributes are defined:

- **DemandNodeId** – Integer. The foreign key of the origin class as defined above.
- **ActivityId** – Integer. The foreign key of the destination class (Activity object class)
- **DemandNodeActivityId** – Integer. The unique identifier of the particular demand node activity object.
- **BaseActivityLevelProjection** – String. Expression of the demand activity level as a function of time.
- **BaseConsumptionRateProjection** – String. Consumption rate for the activity.
- **Elasticity** – Elasticity of the demand node activity to water selling price.
- **ShareOfConsumptiveDemand** – Double. Share of water that is consumptively used.
- **ParentActivityId** – The code of the parent activity (if any). By default equal to -1.

Pollutant Generation per Activity Unit



Finally, each *DemandNodeActivity* relationship class is associated through a many-to-many relationship with the *QualityVariables* class. The class is used to define the generation of pollutant loads through the different activities of the consumptive demand node objects. The particular relationship classes are:

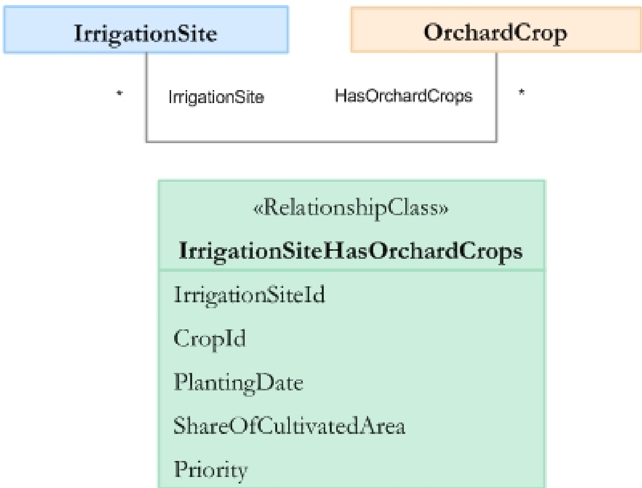
- *PollutantGenerationBySettlement* (Destination class: *SettlementNodeActivities)*
- *PollutantGenerationByTouristSite* (Destination class: *TouristSiteActivities)*
- *PollutantGenerationByIrrigationSite* (Destination class: *IrrigationSiteActivitie)s*

- *PollutantGenerationByAnimalBreedingSite* (Destination class: *AnimalBreedingActivities)*
- *PollutantGenerationByIndustrialSite* (Destination class: *IndustrialSiteActivities)*

PollutantGenerationperActivityUnit is another relationship class, connecting the *DemandNodeActivities* class with the *QualityVariables* object class. The class is used to define the generation of pollutant loads through the different activities of the consumptive demand node objects. The following attributes are defined:

- **DemandNodeActivityId** – Integer. The foreign key of the destination class, as defined above.
- **WQVariableId** – Integer. The foreign key of the origin class (QualityVariables object class)
- **Generation** – Double. Load generation for the particular quality variable by the activity of the consumptive demand node.

Irrigation Site Has Crops

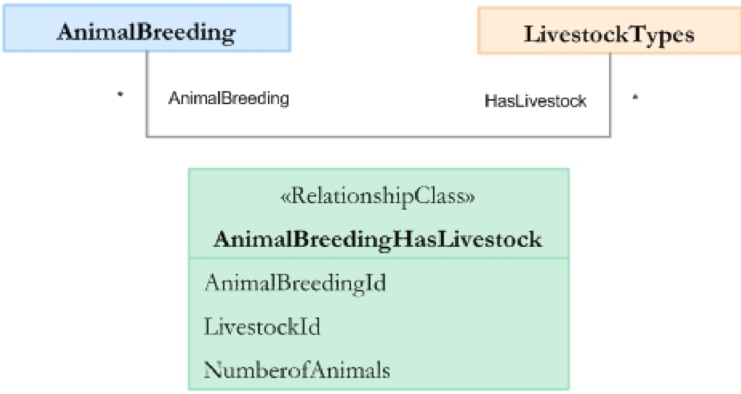


The *IrrigationSiteHasFieldCrops* and *IrrigationSiteHasOrchardCrops* relationship classes are used to relate the child objects of the *CropTypes* class to the irrigation sites they belong to (*IrrigationSite* feature class) and to specify the share of irrigated area they occupy. Thus, each relationship class defines the following attributes:

- **IrrigationSiteId** – Integer. The foreign key of the primary class (*IrrigationSite* feature class).
- **CropId** – Integer. The foreign key from of the destination class (*FieldCrops* or *OrchardCrops*).
- **PlantingDate** – Date. The date when the crop is planted at the irrigation site.
- **ShareOfCultivatedArea** – Double. Share of irrigation site cultivable area that is reserved to the cultivation of the particular crop.

- **Priority** – Integer (values between 1 and 99). The priority with which water distributed at the irrigation site designated by the *IrrigationSiteId* field is allocated to the particular crop.

Animal Site Has Livestock

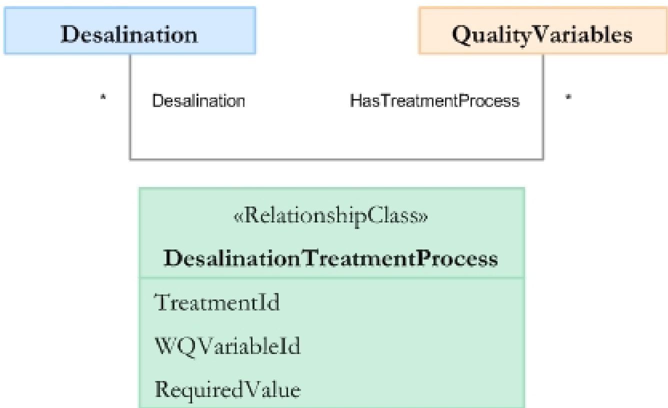


The *AnimalSiteHasLivestock* relationship class is used to relate the animal types of the *LivestockTypes* object class to the animal breeding sites they belong to (*AnimalBreeding* feature class) and to specify their population. According to those requirements, the following attributes are defined by the relationship class:

- **AnimalBreedingId** – Integer. The foreign key of the origin class (*TreatmentNode*)
- **LivestockId**– Integer. The foreign key of the destination class (*QualityVariables*)
- **NumberofAnimals** – Integer. Number of heads of the livestock type specified by the *LivestockID* attribute at the animal breeding site specified by the *AnimalBreedingId* attribute.

Treatment Node Relationships

Drinking Water Treatment Process



The *DrinkingWaterTreatmentProcess* relationship classes model the treatment process of drinking water and desalination plants in terms of required values for drinking water from the legislation of the case study region. The class defines the many-to-many relationship between the

QualityVariables object class and the *Desalination* (*DesalinationTreatmentProcess* relationship class) and *DrinkingWaterTreatmentPlant* (*DWTPTreatmentProcess* relationship class) feature classes.

The relationship class defines the following attributes:

- **TreatmentNodeId** – Integer. The foreign key from the origin class (*TreatmentNode*)
- **WQVariableId** – Integer. The foreign key from the destination class (*QualityVariables*)
- **RequiredValue** – Double. Required concentration for the particular quality variable at the exit of the treatment plant.

Wastewater Treatment Process

The *WastewaterTreatmentProcess* relationship class models the treatment process of a wastewater treatment plant in terms of pollutant removals, required values from existing legislation for effluent disposal and bonus annual coefficients for environmental cost estimations. The class defines the many-to-many relationship between the *QualityVariables* object class and the *WastewaterTreatmentPlant* feature class. The relationship class defines the following attributes:

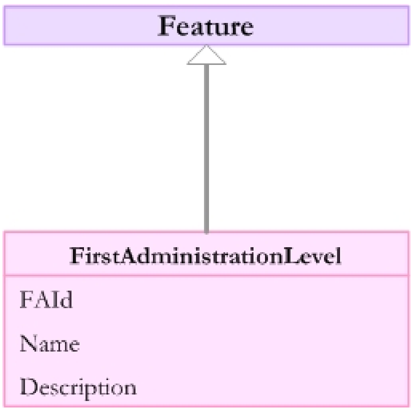
- **TreatmentNodeId** – Integer. The foreign key from the origin class (*TreatmentNode*)
- **WQVariableId** – Integer. The foreign key from the destination class (*QualityVariables*)
- **RemovalRate** – Double (values between 0 and 1). Load removal rate from the treated effluents or water flows.
- **RequiredValue** – Double. Required concentration for the particular quality variable at the exit of the treatment plant.
- **RemovalBonusCoefficient** – Double (values between 0 and 1). Values in this field are used at the estimation of environmental costs incurred by domestic water uses and represent an amount to be subtracted from environmental costs.

6 Administrative Structures

All spatial information concerning the administrative structure of a case study region in terms of governmental and water competent authorities is modelled within the *Administrative Structures* feature dataset. The dataset contains four types of geometric features accounting for first and second level regional administration and water authorities. Attributes included in the water authorities feature class meet also some of the requirements set out by the reporting procedures of the Water Framework Directive. Other attribute data reflect the modelling requirements that have emerged during the data collection of case studies. The feature classes of the dataset are outlined below.

6.1 Regional Administration

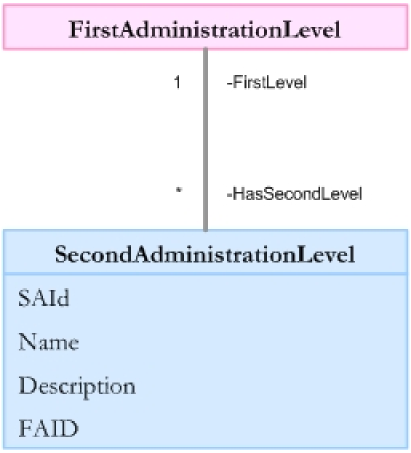
First Administration Level



The *FirstAdministrationLevel* feature class models authorities corresponding to the first (regional) administrative level (e.g. regions or prefectures). The feature class is defined as polygons and incorporates the following attributes:

- **FAID** - String. The unique code for each administrative region object. This may be equal to the codes adopted by the country of the case study.
- **Name** – String. The name of the administrative region object.
- **Description** – String. Other descriptive information.

Second Administration Level

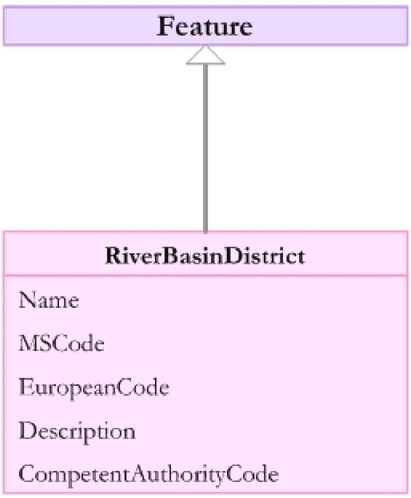


Second administration level means an authority corresponding to the second (regional) administrative level (e.g. municipalities). The feature class is defined as a polygon feature class and is associated with a one-to-many relationship with the *FirstAdministrationLevel* feature class. The class defines the following attributes:

- **SAId** - String. The unique code of the administrative region.
- **Name** – String. The name of the administrative region object.
- **Description** – String. Other descriptive information.
- **FAID**. The foreign key of the origin class (first level administrative region.

6.2 Water Administration

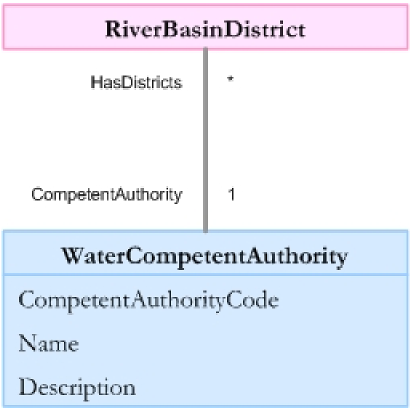
River Basin District



River Basin Districts can be collections of River Basins, Transitional Waters and Coastal Waters areas. Thus, despite the duplication of some geometry from the *Water Features* Dataset, River Basin Districts are defined as a separate polygon feature class. In addition, the following attributes are defined:

- **Name** – String. Locally used name.
- **MSCode** – String Unique code for a river basin district at Member State Level.
- **EuropeanCode** – String. Unique code for a river basin district at EU Level.
- **CompetentAuthorityCode** – String. Code of the competent authority for the river basin district.

Water Competent Authority



Competent Authority means an authority or authorities as those are identified under Article 3(2) of the Water Framework Directive *("Member states shall ensure the appropriate administrative arrangements, including the identification of the appropriate competent authority … within each river basin district within their territory.")*. Because in some cases it is not possible to aggregate River Basin District objects to form the boundary of the Competent Authority, the latter are defined as a separate polygon feature class, with the following attributes:

- **CompetentAuthorityCode** – String. Unique code for the competent authority.
- **Name** – String. Locally used name.
- **Description** – String. Other descriptive information.

7 Time Series

Time series data describe many aspects of a water resource system and are used as an input in the analysis conducted with the WSM Decision Support System, in order to define base demands, available water resources and the initial or reference quality status of water bodies. For example, they can be used to describe the amount of water that is being released from a river system or a series of measurements of BOD concentrations of a river reach.

The two main factors that determine the use of time series are the length of the time series record and the time interval. It is also important to know whether data contain actual recorded values or interpolated values between two recordings or if they represent a result of a calculation procedure, such as those performed within the WSM DSS.

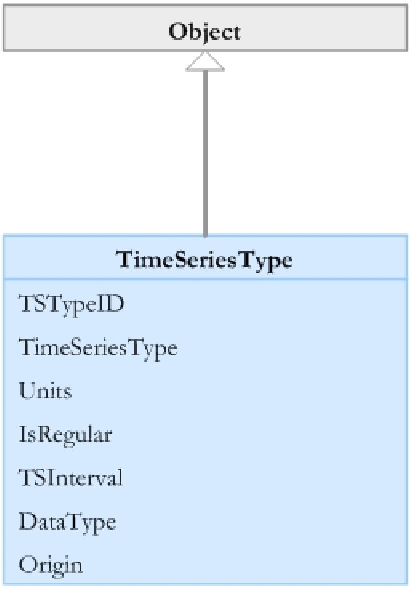
The intention of including Time Series Data in the WSM Data Model is not only to build a data model that will satisfy the requirements of the WSM DSS but also to create a database that perhaps will be applicable to many models that operate independently of the GIS and the DSS.

The modelling and conceptualization of Time Series has been adapted from the Arc Hydro Data Model, developed by the University of Texas and ESRI.

Modelling

Usually, time series data are captured and stored in a variety of formats. Time series data that are directly being stored in the geodatabase can be represented as a standard geodatabase table (inheriting from the Arc Info Object Class) and referred to as a Time Series Object Class. In order to accommodate the main characteristics of Time Series, such as interval, type and origin (recorded or generated) a database object describing those elements that characterize the time series types, namely *TimeSeriesType*, is necessary. Raw time series data are stored in the *SimpleTimeSeries* object class and the *QualityTimeSeries* object class.

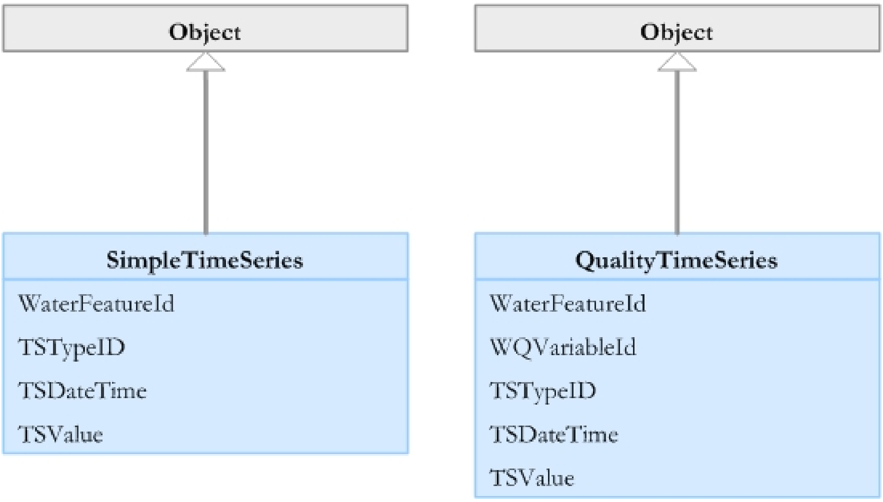
Time Series Type Object Class



The time series type object class contains all information that is essential to characterize the time series object; those contain information about the origin, the time interval, the units and its regularity (e.g. regular daily measurements of BOD Concentrations). The class in the WSM Data Model defines seven attributes.

- **TSTypeID** – Integer. The unique identifier of a time series type object. Used in a one-to-many relationship with the Simple and Quality time series object classes.
- **TimeSeriesType** – Domain attribute. Provides a description of the related time series object e.g. precipitation, population, concentration, load etc.
- **Units** – String. Measurement units for the time series e.g. people for population, mg/l for BOD concentrations, cubic meters for runoff etc.
- **IsRegular** – True/False Domain attribute. Whether the time series is regular or not (all measurements or data have the same time interval).
- **TSInterval** – Domain Attribute. The interval between two measurements of the time series (i.e. hourly measurements, daily measurements etc.).
- **DataType** – Domain attribute. Describes the aggregation procedure of time series data, e.g. cumulative, instantaneous etc.
- **Origin** – Domain attribute (see Appendix). Whether time series data are recorded (measurements) or interpolated/generated by a calculation procedure.

Simple and Quality Time Series Object Classes



Within the geodatabase and the DSS time series source data are treated like any other form of tabular data, except that the time series object class must include *WaterFeatureId*, *TSTypeID*, *TSDateTime* and *TSValue* in its fields. A time series object represents a single row in the data table. The time series object class in the WSM Data Model has four attributes, namely *WaterFeatureId*, *TSType*, *TSDateTime* and *TSValue*.

- **WaterFeatureId** – Integer. The feature ID is an integer identifier, usually set equal either to *WaterNodeId* or *WaterLinkId* of the feature described by the time series. For example, if the *WaterNodeId* of a Settlement feature is 100 then all the time series classes of Settlement will have a *WaterFeatureID* of 100.

- **TSTypeID** – Integer. Each element in a time series class is directly associated with a distinct record of the *TimeSeriesType* table which qualifies the type of data corresponding to each value in the time series class. This attribute is very important as it describes the type of data that are stored in the *TSValue* field.

- **TSDateTime** – Date Time. Each element in a time series has a distinct *TSDateTime*. A *TSDateTime* is a labelled point in time and is a standard Date field in Arc GIS.

- **TSValue** – Double. The value of the time series object.

As previously mentioned, two time series classes are defined, Simple and Quality Time Series. The *QualityTimeSeries* object class is used to store time series on concentrations and loads for the quality variables modelled within the DSS. The structure of the Quality Time Series Table is similar to that of Simple Time Series with the addition of a foreign key relating each record to the relevant quality variable. The UML diagram for the Time Series component of the Data Model is presented in Figure 11.

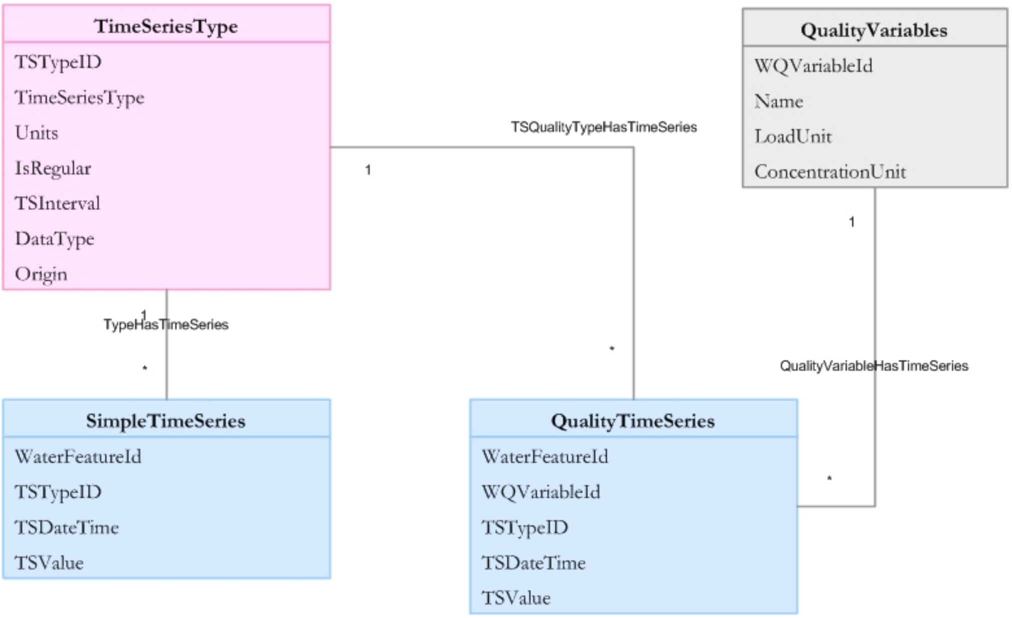


Figure 11. Time Series Object Model

The most important aspect of time series is that all data are stored within one single large file, regardless of the feature type and the data stored in it. Thus, any value which in this case is a *TSValue*, can be represented by a point in the three dimensional space which has its corresponding *WaterFeatureID*, *TSDateTime* and TS type attributes, as shown in Figure 12.

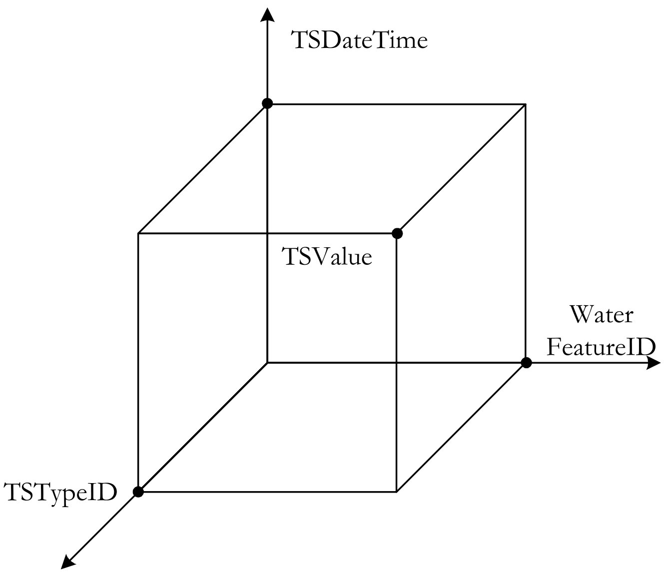


Figure 12. 3D structure of time series in the WSM Data Model

The above three dimensional structure of time series is simple and general, and is formed of a series of vertical and horizontal planes. Therefore, any time series subset (query) that has a single value for *WaterFeatureID* represents a vertical plane, perpendicular to the *WaterFeatureID* axis and it contains several values for *TSType* and *TSDateTime*. A time series subset that has a single value for *TSTypeID* represents another vertical plane that is perpendicular to the *TSType* axis. Similarly, a time series subset that has a single value for *TSDateTime*, represents a horizontal plane that is perpendicular to the *TSDateTime* axis. Therefore, different time series queries can be created from the big object class. When two vertical planes intersect, their line of intersection represents a time series query that corresponds to a single *WaterFeatureID* and therefore a feature, and a single *TSTypeID* for several *TSDateTime* values. On the other hand, the intersection of all the three planes represents a single point that has only one value for *WaterFeatureID*, *TSTypeID* and *TSDateTime* respectively.

Table 3 outlines the most important time series data attributed to water nodes and links.

Table 3. Time Series related to the objects of the Water Network Dataset

Water Network Feature Class	Associated Time Series
Water Link	Inflow Outflow
Water Node	Inflow Outflow
Supply Node	Concentrations for all quality variables
Surface Water Node	Reference Concentrations and Loads for all quality variables
Reservoir	Volume Evaporation Downstream Release Eutrophication Seepage Losses Level Elevation
River Reach Node	Run Off
Renewable Groundwater	Storage Infiltration Discharge Reference Concentrations and Loads for all quality variables
Demand Node	Demand Requirements
Industrial Site	Industrial production volume
Settlement	Population Seasonal Population
Tourist Site	Seasonal Population
Irrigation Site	Irrigated Area Maximum Cultivable Area
Hydroelectricity	Energy Requirements Energy Production

8 References

[1] Water Framework Directive – Common Implementation Strategy (2002), Guidance Document on Implementing the GIS Elements of the WFD.

[2] Official Journal of the European Commission, Directive 2000/60/EC of the European Parliament and of the Council of 23 October 2000.

[3] Maidment D.R. (2002), Arc Hydro GIS for Water Resources, ESRI Publications.

[4] Mc Donald A. (2001), Building a Geodatabase, ESRI Publications.

[5] Zeiler M. (2001), Exploring Arc Objects, Volumes 1 and 2, ESRI Publications.

Appendix - The Domains of the WSM Data Model

A domain of a database represents a named constraint. An attribute domain is used to constrain the values allowed in any particular attribute for a table, feature class, or subtype and can be shared across feature classes and tables in a geodatabase. Types of attribute domains include range and coded value domains. A coded value domain specifies a valid set of values for an attribute. Coded value domains can apply to any type of attribute—text, numeric, date, and so on. On the contrary a range value domain specifies a valid range of values for a numeric attribute.

I.1 Domains Related to Regional Data

Land Use Types

The *LandUseTypes* coded value domain is used to specify valid values for the *LandUseType* field of the *LandUse* feature class (chapter 3.3). The classification is based on the USGS Land Cover classification, with some modification on the codes to make them easily understandable by the user of the DSS.

Table I - 1. Land Use Types coded value domain

Code	Description	Code	Description
1	Urban and Built-up Land	13	Evergreen Broadleaf forest
2	Dry land, Cropland and Pasture	14	Evergreen Needleaf Forest
3	Irrigated Cropland and Pasture	15	Mixed Forest
4	Mixed Dryland-Irrigated Cropland& Pasture	16	Water Bodies
5	Cropland/Grassland Mosaic	17	Herbaceous Wetland
6	Cropland/Woodland Mosaic	18	Wooded Wetland
7	Grassland	19	Barren or sparsely vegetated
8	Shrub land	20	Herbaceous Tundra
9	Mixed Grassland/ Shrub land	21	Wooded Tundra
10	Savanna	22	Mixed Tundra
11	Deciduous Broadleaf Forest	23	Bare Ground Tundra
12	Deciduous Needleaf Forest	24	Snow or Ice

Road Types

The *RoadTypes* coded value domain is used to specify valid values for the *RoadType* field of the *Roads* feature class (chapter 3.2). A distinction is made according to the type (and importance) of the road.

Table I - 2. Road Types coded value domain

Code	Description
1	National Road
2	National Country Road
3	Country Road 1

Code	Description
4	Country Road 2
5	Country Road 3
6	Earthy
7	Unknown

Soil Phases

The *SoilPhases* coded value domain is used to specify valid values for the *Phase1* and *Phase2* of the *Soil* feature class (chapter 3.3). The classification is based on the FAO Soil Phases. FAO Phases are subdivisions of soil units based on characteristics which are significant to the use or the management of land but which are not diagnostic for the separation of soil units themselves.

Table I - 3. Soil Phases coded value domain

Code	Description
1	Stony
2	Lithic
3	Petric
4	Petrocalcic
5	Petrogypsic
6	Petroferric
7	Phreatic
8	Fragipan
9	Duripan
10	Saline
11	Sodic
12	Cerrado

Miscellaneous Land Unit Types

The *MiscellaneousLandUnitTypes* coded value domain gives valid values for the *MiscellaneousLandUnit1* and *MiscellaneousLandUnit2* fields of the *Soil* feature class (chapter 3.3).

Table I - 4. Miscellaneous Land Unit Types coded value domain

Code	Description
0	None
1	Dunes or shifting sands
2	Salt flats
3	Rock debris or desert detritus

I.2 Domains Related to Network Objects

Geology types

The *GeologyTypes* coded value domain specifies valid values for the *GeologyType* field of the *RenewableGroundwater* feature class. Values describe the predominant geology type of each aquifer object of the feature class.

Table I - 5. Geology type coded value domain

Code	Description
1	Calcareous
2	Siliceous
3	Organic
4	Other

Pricing Methods

The *PricingMethods* coded value domain specifies valid values for the *PricingMethod* field of the *ConsumptiveDemand* abstract class. A variety of pricing methods is included, such as volumetric, per area (for irrigation sites), output pricing and tiered pricing.

Table I - 6. Pricing methods coded value domain

Code	Description
1	Volumetric
2	Per Area
3	Output Pricing
4	Tiered Pricing
5	None

Irrigation Methods

The *IrrigationMethods* coded value domain specifies valid values for the *IrrigationMethod* field of the *AppliedIrrigationMethods* object class. The first two types concern surface irrigation which is the application of water by gravity flow to the surface of the field. Either the entire field is flooded (flood irrigation) or the water is fed into small channels (furrows) or strips of land (borders). Sprinkler irrigation is similar to natural rainfall. Water is pumped through a pipe system and then sprayed onto the crops through rotating sprinkler heads. Finally, with drip irrigation, water is conveyed under pressure through a pipe system to the fields, where it drips slowly onto the soil through emitters or drippers, which are located close to the plants.

Table I - 7. Irrigation methods coded value domain

Code	Description
1	Flood
2	Furrow

3	Sprinkler
4	Drip

Wastewater Treatment Process Types

The *WastewaterTreatmentTypes* coded value domain specifies valid values for the *UnitType* field of the *WastewaterTreatmentPlant* feature class, according to the wastewater treatment process of the particular objects.

Table I - 8. Wastewater Treatment Process coded value domain

Code	Description
1	Primary
2	Secondary
3	Tertiary

Desalination unit types

The *DesalinationUnitTypes* coded value domain specifies valid values for the *UnitType* field of the *Desalination* network class according to the type of water treated (sea or brackish) by the desalination process.

Table I - 9. Desalination unit type coded value domain

Code	Description
1	Brackish water
2	Sea water

Desalination Process Types

The *DesalinationProcessTypes* coded value domain specifies valid values for the *ProcessType* field of the Desalination feature class (Water Network dataset), according to the particular desalination technology applied in the unit.

Table I - 10. Desalination process type coded value domain

Code	Description
1	Reverse Osmosis
2	Electrodialysis
3	Multi-stage Flash
4	MultiEffect Distillation
5	Vapour Compression

I.3 Domains Related to Time Series

Time Series Type

The *TimeSeriesType* is a coded value domain which is used to specify valid values for the attribute *TimeSeriesType* of the *TimeSeriesTypes* object class.

Table I - 11. Time Series Type Domain

Code	Description	Code	Description
1	Precipitation	24	Link Inflow
2	Potential Evapotranspiration	25	Link Outflow
3	Minimum Temperature	26	Demand Requirements
4	Maximum Temperature	27	Hydroelectricity Energy Production
5	Mean Diurnal Temperature	28	Water Selling Price
6	Sunlight Hours	29	Reservoir Level Elevation
7	Relative Humidity	30	Population
8	Wind Speed	31	Industrial Production Volume
9	Days with Rainfall	32	Irrigated Area
10	Infiltration	33	Maximum Cultivable Area
11	Moisture Availability Index	34	Reservoir Surface Area
12	Days with Ground Frost	35	Node Inflow Quality
13	RunOff	36	Node Outflow Quality
14	Groundwater recharge	37	Groundwater Storage
15	Groundwater discharge	38	Energy Requirements
16	Reservoir Volume	39	Reservoir Storage
17	Reservoir Downstream Release	40	Concentration
18	Coastal Zone Eutrophication	41	Reference Evapotranspiration
19	Reservoir Eutrophication	42	Reference Concentration
20	Reservoir Evaporation	43	Reference Load
21	Reservoir Seepage Losses	44	Link Concentration
22	Node Inflow	45	Seasonal Population
23	Node Outflow	46	Actual Evapotranspiration

Time Series Origin

The *TSOrigin* coded value specifies valid values for the *Origin* field of the *TimeSeriesType* class. Codes indicate whether the time series values corresponding to the particular time series type object are recorded or generated.

Table I - 12. Time Series Origin coded value domain

Code	Description
1	Recorded
2	Generated

Time Series Data Type

The *TSDataType* coded value domain specifies valid values for the *DataType* attribute of the *TimeSeriesType* class. Codes indicate whether the time series values corresponding to the particular time series type object are instantaneous, cumulative for the time series period, incremental (the value for a particular step is the sum of all previous values), average, or they refer to the maximum or minimum value observed during the time interval.

Table I - 13. Time Series Data Type coded value domain

Code	Description
1	Instantaneous
2	Cumulative
3	Incremental
4	Average
5	Maximum
6	Minimum

Time Series Interval Type

The *TSIntervalType* coded value domain specifies valid values for the *TSInterval* attribute of the *TimeSeriesType* class. Codes specify the time interval between two time series objects. In the WSM Data Model only 3 coded values have been considered necessary.

Table I - 14. Time Series Interval Type coded value domain

Code	Description
1	1 Month
2	1 Year
99	Other