# How to make linking of models with different grids possible.

Spatial Mapping

Hydrological models usually have a schematization or grid that represents the spatial resolution of the model. For groundwater models, regular or non-regular two- or three-dimensional grids may be used, whereas river models typically use a one-dimensional grid. Conceptual catchment models may use a closed polygon to describe the catchment boundary.

When models with different spatial schematizations are linked, the values associated to one schematization in the source model must be transformed to be represented on the schematization of the target model. In order to make such transformations possible for any combination of models a standardized spatial representation was defined. In OpenMI terms this is called the IElementSet interface (see Figure below).
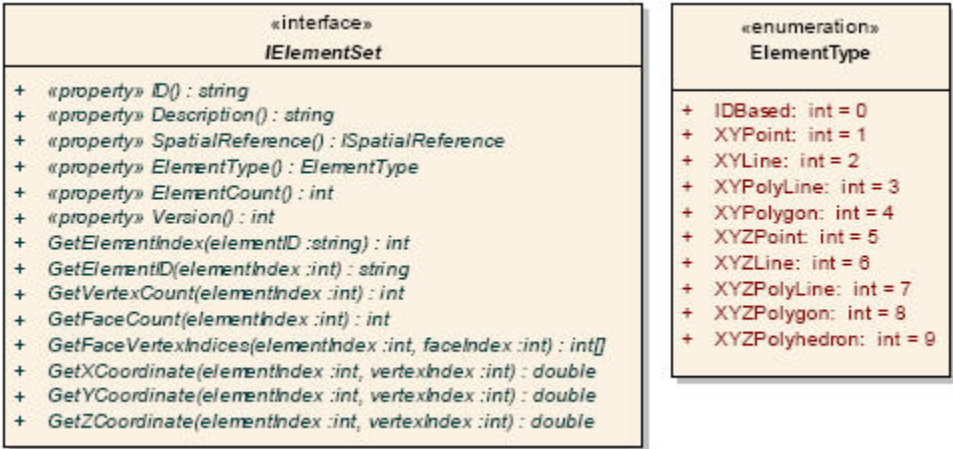
A two-dimensional grid for a groundwater model and a one-dimensional grid for a river model are shown in the figure below.
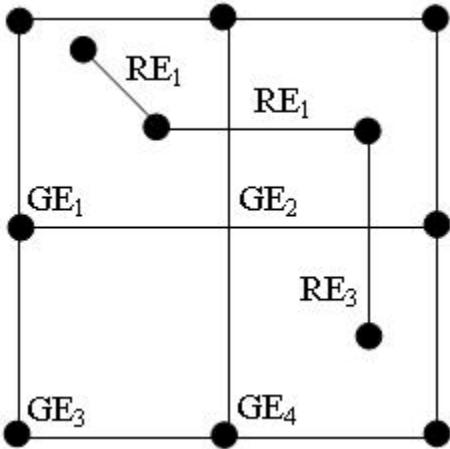
The groundwater model grid consists of four elements GE1, GE2, GE3 and GE4. These elements have the type Polygon. Each corner of these polygons is a Vertex, and each Vertex has Co-ordinates. The type Polygon is defined in the standard in the enumeration ElementType, whereas there are no interface definitions for Vertex and Co-ordinate; these are only part of the OpenMI terminology and are used in the naming of some of the methods in the IElementSet interface.

The grid for the river model can be represented by an implementation of the IElementSet interface, where each branch is an element of type Polyline. Each Polyline element will have two vertices, one at each end of the line. The elements for the river model are shown in figure above as RE1, RE2 and RE3.

Assume that the river model and the groundwater model illustrated in the figure above are linked for the purpose of transferring information about groundwater leakage from the river model to the groundwater model. When the groundwater model invokes the GetValues method in the river model in order to obtain the leakages, the river model must return a ValueSet, where each value represents the leakage that enters each element in the groundwater model for the requested time or timespan. With this in mind it may seem like a huge task to implement the GetValues method in the linkable component. However, since all spatial representations can be accessed generically through the IElementSet interface, a generic element mapper could be developed. Such a mapper is provided in the open source OpenMI SDK (OATC.OpenMI.SDK.Spatial).

The two most essential methods in the ElementMapper class are:

void Initialize(string method, IElementSet fromElements, IElementSet toElements)
and

IValueSet MapValues(IValueSet inputValues)

During calculations, when the GetValues method is invoked, the source component uses this mapping matrix to make the spatial conversion, simply by multiplying the vector of values associated to the grid of the source components with the mapping matrix.
For the example shown in the figure above, the mapping matrix will look as shown below:

$$\underline{\underline{A}} = \begin{pmatrix} 1 & 1/3 & 0 \\ 0 & 2/3 & 1/2 \\ 0 & 0 & 0 \\ 0 & 0 & 1/2 \end{pmatrix} \quad (eq.1)$$

When the groundwater model invokes the GetValues method in the river model, the river model can make the spatial transformation of its internal calculated leakages using the following multiplication:

$$\underline{I} = \underline{L} \times \underline{\underline{A}} \quad (eq.2)$$

where $\underline{I}$ is a vector with four components describing the leakage contribution to each grid cell in the groundwater model and $\underline{L}$ is a vector with three components, each value being the calculated leakage in a river branch.

It is important to note that element mapping as described above has nothing to do with the OpenMI standard. Anyone can implement the transformations as they please; for example, if the linkable component is an analytic model and therefore has no grid element, mapping does not make sense.