

# Feature requests

## Feature requests

OpenMI Feature requests (Updated January 31st 2007)		
ID	Request	OpenMI Association Technical Committee evaluation
1	<p><b>Latest Overlapping Date/Time</b> By: wrmdaniels (wrmdaniels) - 2006-12-05 11:17</p> <p><a href="https://sourceforge.net/forum/forum.php?thread_id=1625591&amp;forum_id=481283">https://sourceforge.net/forum/forum.php?thread_id=1625591&amp;forum_id=481283</a></p> <p>In the GUI, the models run to the latest overlapping date/time. If this date is, say, the 7th January 2000, then the models will not actually reach the 7th January, they run to the timestep just before 7th January. I think it would be more logical for the models to run to, and including, the end time. What does anyone else think?</p> <p><i>RE: Latest Overlapping Date/Time</i> By: Rob Knapen (rknapen) - 2006-12-05 14:09</p> <p><i>I think that from the outside (the GUI) you should only see the request to a linkable component for a specific date. It is up to the inner workings of the component how best to answer the request. Perhaps no time steps are used or time steps are very long. And perhaps it is OK for the model to stop one time step short of the requested date. The model specific tracing / logging information however should clearly indicate how the requested data has been calculated.</i></p> <p><i>However yes, if the linkable component uses an engine that is driven by time steps, most likely the component will do a performTimeStep call until the engine's currentTime is the same as or exceeds the requested time.</i></p>	<p>Accepted [A], Rejected [R] or Considered [C] for implementation in version: OpenMI .Net 1.3 [ A ] OpenMI Java 1.3 [ ] OpenMI .Net 2.0 [ ] OpenMI Java 2.0 [ ]</p> <p>Comments: The may be a bug in the Utilities.Wrapper implementation. Could also be something that relates to how GetInputTime is implemented for the individual model. We will construct unit tests in Utilities.Wrapper.Test that demonstrates this behavior, the fix the problem and make sure that the new tests behaves as expected. (should be implemented in OpenMI version1.3). Task: Stef Hummel. Stef will check it this also has implications for the Java version also.</p> <p>Implementation status: Implementation expected to be completed by January 22 2007.</p>
2	<p><b>Add symbols to DimensionBase</b> By: Rob Knapen (rknapen) - 2006-12-05 10:27</p> <p><a href="http://sourceforge.net/forum/forum.php?thread_id=1625571&amp;forum_id=481283">http://sourceforge.net/forum/forum.php?thread_id=1625571&amp;forum_id=481283</a></p> <p>To represent dimensions as text it would convenient to add the symbols to the DimensionBase enumeration.</p>	<p>Accepted [A], Rejected [R] or Considered [C] for implementation in version: OpenMI .Net 1.3 [ R ] OpenMI Java 1.3 [ R ] OpenMI .Net 2.0 [ C ] OpenMI Java 2.0 [ C ]</p> <p>Comments: Such implementation can easily be done by anyone that e.g. wants such feature in a user interface. All information is already available. We see little need for this and do not want to complicate the official OpenMI implementations with this. However, when the overall architecture for version 2 is in place we will add a few things that belong to the <i>nice to have</i> category, such as this issue.</p> <p>Implementation status:</p>

3	<b>IGridElementSet</b> By: Rob Knapen (rknapien) - 2006-12-05 10:23  <a href="http://sourceforge.net/forum/forum.php?thread_id=1625570&amp;forum_id=481283">http://sourceforge.net/forum/forum.php?thread_id=1625570&amp;forum_id=481283</a>  In some domains where we want to use OpenMI it is very common to work with grid based (spatial) data. It is rather awkward to represent this in the current available ElementSet. In general the information that the data represents a grid is lost and there is no way to optimize processing anymore. Preferably ElementTypes for grid cells are added together with methods for handling grid data. This could be done in a IGridElementSet interface that inherits from IElementSet.	Accepted [A], Rejected [R] or Considered [C] for implementation in version: OpenMI .Net 1.3 [ ] OpenMI Java 1.3 [ ] OpenMI .Net 2.0 [ ] OpenMI Java 2.0 [ ] Comments: We do not like to add specific subtypes of the elementset to the OpenMI standard or implementation. However, anyone are allowed to make his own implementation of the IElementSet and when this is used in e.g. a user interface to check if a ElementSet is of this type and then utilize the specific features of this implementation. Implementation status:
4	<b>LinkableComponentGroup</b> By: Rob Knapen (rknapien) - 2006-12-05 01:18  <a href="http://sourceforge.net/forum/forum.php?thread_id=1625564&amp;forum_id=481283">http://sourceforge.net/forum/forum.php?thread_id=1625564&amp;forum_id=481283</a>  To group linkable components introduce a LinkableComponentGroup class (in the backbone). This class implements the ILinkableComponent interface, but internally can contain a number of ILinkableComponents and ILinks. Further nesting is possible. A Composition then can be implemented as a specific form of a LinkableComponentGroup. Groups are handy to re-use (parts) of compositions and for user interface representation. The Java org.openmi.alterra.configuration package contains an implementation of this.	Accepted [A], Rejected [R] or Considered [C] for implementation in version: OpenMI .Net 1.3 [ C ] OpenMI Java 1.3 [ C ] OpenMI .Net 2.0 [ C ] OpenMI Java 2.0 [ C ] Comments: Considered for implementation in the OpenMI.Tools GUI package. This does not require changes to the standard. We think this a useful utility and if someone needs it will support such implementation. One way this could work is that the GUI when saving the xml configuration file also saves a corresponding omi file. The such configuration can be loaded into another configuration by selecting this omi file. Implementation status:
5	<b>Persist Argument in String</b> By: Rob Knapen (rknapien) - 2006-12-05 01:12  <a href="http://sourceforge.net/forum/forum.php?thread_id=1625561&amp;forum_id=481283">http://sourceforge.net/forum/forum.php?thread_id=1625561&amp;forum_id=481283</a>  For persistency a specification should be added for representing the state of an Argument in a string, and to restore it from a string. For example it can be: a key-value list with "=" as separator between key and value, and ";" to separate the pairs. Together with a list of standard key names.	Accepted [A], Rejected [R] or Considered [C] for implementation in version: OpenMI .Net 1.3 [R] OpenMI Java 1.3 [R ] OpenMI .Net 2.0 [R] OpenMI Java 2.0 [R] Comments: The current .net implementation of the configuration user interface (OmiED) can not support such changes because the method initialize is invoked immediately when a model is added to the configuration. However, this could be changed. Then the user interface could load the LinkableComponents and show these on the canvas without invoking the initialize. After that the user may be allowed to change the arguments and finally the GUI will invoke the Initialize methods. With respect to making the arguments persistent we think that this should be handled by the user interface and such information should be saved in the corresponding configuration file (e.g. opr file) Implementation status:
6	<b>Add properties to the Argument</b> By: Rob Knapen (rknapien) - 2006-12-05 01:08  <a href="http://sourceforge.net/forum/forum.php?thread_id=1625557&amp;forum_id=481283">http://sourceforge.net/forum/forum.php?thread_id=1625557&amp;forum_id=481283</a>  The Argument class should be extended with some additional properties (required, possible values, argument type) to allow for automatic processing and constructing (input) GUI widgets for them. Currently the contents of an argument is to unstructured to do this.	Accepted [A], Rejected [R] or Considered [C] for implementation in version: OpenMI .Net 1.3 [R] OpenMI Java 1.3 [R] OpenMI .Net 2.0 [C] OpenMI Java 2.0 [C] Comments: We think this can be handle by making your own interface which inherits the IArgument interface. Your user interface can then check if an argument implement such interface and utilize the additional features / methods of this interface. However, for OpenMI version two we may make some improvements such as proposed here. Implementation status:
7	<b>Add method to retrieve init arguments</b> By: Rob Knapen (rknapien) - 2006-12-05 01:06  <a href="http://sourceforge.net/forum/forum.php?thread_id=1625556&amp;forum_id=481283">http://sourceforge.net/forum/forum.php?thread_id=1625556&amp;forum_id=481283</a>  A method added to the LinkableComponent to request the list of initialization arguments (e.g. getInitialisationArguments()) would be convenient. Currently there is no way for an application to find out what arguments a component needs. It can only be provided by documentation, or by an additional (non OpenMI) interface.	Accepted [A], Rejected [R] or Considered [C] for implementation in version: OpenMI .Net 1.3 [R] OpenMI Java 1.3 [R ] OpenMI .Net 2.0 [C] OpenMI Java 2.0 [C] Comments: See comment from request 5. Implementation status:

8	<p><b>Add support for qualitative information</b> By: Rob Knapen (rknapen) - 2006-12-05 00:51</p> <p><a href="http://sourceforge.net/forum/forum.php?thread_id=1625550&amp;forum_id=481283">http://sourceforge.net/forum/forum.php?thread_id=1625550&amp;forum_id=481283</a></p> <p>To be able to use OpenMI in other domains there is a large request to support qualitative data (besides the current quantitative data). To implement this a parent interface (IDataType) could be introduced, which specifies into the IQuantity and IQuality interfaces. I have a Java sample implementation of this, for those interested. If possible it should be included into the (next) specification.</p>	<p>Accepted [A], Rejected [R] or Considered [C] for implementation in version: OpenMI .Net 1.3 [R] OpenMI Java 1.3 [R] OpenMI .Net 2.0 [A] OpenMI Java 2.0 [A] Comments: We think that this is necessary. We will continue the discussions of how specifically to define this in the OpenMI standard. Implementation status:</p>
9	<p><b>To display an image on the model</b> By: wmdaniels (wmdaniels) - 2005-09-07 07:20</p> <p><a href="http://sourceforge.net/forum/forum.php?thread_id=1347872&amp;forum_id=481283">http://sourceforge.net/forum/forum.php?thread_id=1347872&amp;forum_id=481283</a></p> <p>Jeremy from WRC requested that an image be displayed on the models when they load up in the GUI to make them more identifiable. <i>RE: To display an image on the model</i> By: Jan Gregersen (jan_gregersen) - 2006-08-07 04:48 <i>I think this would be a nice feature, since the visual appearance of a OpenMI configuration can pretty messy in the graphical view of the OpenMI configuration editor (OmiED). However, including such feature would require changes to the OpenMI standard (the OmiED needs a standardized way to request the image to show). Therefore I think we should concede this feature for OpenMI version 2.0</i></p>	<p>Accepted [A], Rejected [R] or Considered [C] for implementation in version: OpenMI .Net 1.3 [R] OpenMI Java 1.3 [R] OpenMI .Net 2.0 [R] OpenMI Java 2.0 [R] Comments: These should be no changes to the standard. Anyone can make user interfaces that features import of images to symbolize the individual linkable component. Implementation status:</p>
10	<p><b>Copy Paste functionality for the DataMonitor</b> By: wmdaniels (wmdaniels) - 2006-11-30 03:14</p> <p><a href="http://sourceforge.net/forum/forum.php?thread_id=1622968&amp;forum_id=481283">http://sourceforge.net/forum/forum.php?thread_id=1622968&amp;forum_id=481283</a></p> <p>There is no copy and paste functionality in the datamonitor at the moment. It would be useful for copying the data from the DataMonitor and pasting it into Excel for data analysis</p>	<p>Accepted [A], Rejected [R] or Considered [C] for implementation in version: OpenMI .Net 1.3 [A] OpenMI Java 1.3 [A] OpenMI .Net 2.0 [A] OpenMI Java 2.0 [A] Comments: RIZA has developed at data monitor that data exchange with Excel. <b>Flip</b> will upload this to source forge. (the version 1.3 branch) Implementation status:</p>
11	<p><b>Latest Overlapping Date/Time Control in GUI</b> By: wmdaniels (wmdaniels) - 2006-12-01 07:19</p> <p><a href="http://sourceforge.net/forum/forum.php?thread_id=1623983&amp;forum_id=481283">http://sourceforge.net/forum/forum.php?thread_id=1623983&amp;forum_id=481283</a></p> <p>Please may I request that the latest overlapping time control in the GUI show the date/time format depending on the user-culture, for instance, in the UK we use dd/mm/yyyy but in the US they use mm/dd/yyyy. At the moment the GUI always shows the date/time the US way which can be a bit confusing for us!</p>	<p>Accepted [A], Rejected [R] or Considered [C] for implementation in version: OpenMI .Net 1.3 [A] OpenMI Java 1.3 [A] OpenMI .Net 2.0 [A] OpenMI Java 2.0 [A] Comments: Yes, <b>Wiki</b> will implement this change Implementation status:</p>
12	<p><b>Include the DataMonitor in the installation</b> By: wmdaniels (wmdaniels) - 2006-11-30 03:11</p> <p><a href="http://sourceforge.net/forum/forum.php?thread_id=1622965&amp;forum_id=481283">http://sourceforge.net/forum/forum.php?thread_id=1622965&amp;forum_id=481283</a></p> <p>Can the datamonitor be part of the standard install - at the moment you could get hold of it by doing a build of the source code but this is not really practical for non-programmers. Two files would need to be installed - org.OpenMI.Tools.DataMonitor.dll and datamonitor.omi</p>	<p>Accepted [A], Rejected [R] or Considered [C] for implementation in version: OpenMI .Net 1.3 [A] OpenMI Java 1.3 [A] OpenMI .Net 2.0 [A] OpenMI Java 2.0 [A] Comments: <b>Jan</b> will make sure that the data monitor is included in the installation for the next release. Implementation status:</p>

13	<p><b>Possibilities of hardware environment</b>  By: Ronse Yves (ronse_yves) - 2006-11-27 05:13</p> <p><a href="http://sourceforge.net/forum/forum.php?thread_id=1620711&amp;forum_id=462113">http://sourceforge.net/forum/forum.php?thread_id=1620711&amp;forum_id=462113</a></p> <p>Dear members of the OpenMI Association</p> <p>In the project proposal of the OpenMI LIFE project, there is stated in form T2a for task B at paragraph 2 Trial integrated modelling that the (linked) models (shall be) running on different platforms and on different machines.</p> <p>The Scheldt partners are interpreting this as each model running on its own platform, thus on its own location ; the link shall be created by Internet data or communication exchange on line.</p> <p>As the runs of linked models shall spend many hours, even more than one day time, the Scheldt partners believe this way of working shall not be successful, and thus not reliable, because of a too high risk of communication failure.</p> <p>The proposal of the Scheldt partners is to organise a hardware environment with linked multiprocessors on one location instead of to organise the adaption of existing hardware environments (for each model) on different locations.</p> <p>Which is the point of view of the OpenMI Association about this matter ?</p> <p>RE: Possibilities of hardware environment  By: Jan Gregersen (jan_gregersen) - 2006-11-29 01:47</p> <p>The current version of OpenMI (version 1.*) requires that all LinkableComponents are running in one thread and on one machine. However, you can implement your LinkableComponent in such a way that it internally communicates with your calculating engine that is located on a different machine. The advantages of not including such communication protocols into the OpenMI standard is that choice of protocol for communication between processes or machines can be made based on what is optimal for your particular model. Practically, seen from the users perspective, all LinkableComponents that are part of a OpenMI configuration are installed on the local machine, and the user does not have to worry about that one of more of these components internally exchange data with their calculation engines e.g. over the internet. These special types of LinkableComponents can be very small in size, since they only act as proxy's for the real models. People that are migrating models to comply with the OpenMI standard are typically using the OpenMI wrapper utilities package, which makes the migration much easier. For the proxy kind of components there are no utilities available, but the OpenMI Association will discuss this issue at the next technical committee meeting, and may decide to create such utilities.</p> <p>You also ask about running all models on one multiprocessor machine. Basically, you can do this with "normal" kind of LinkableComponents. However, since OpenMI is single threaded I would not expect that you will improve the performance due to multiple processors. When one model is calculating all other models in the configuration are just waiting for this model to return the data. There may be some cases where some models may be allowed to proceed with their calculations after they have returned the requested data. Such a feature could be implemented in the wrapper utilities.</p>	<p>Accepted [A], Rejected [R] or Considered [C]  for implementation in version:  OpenMI .Net 1.3 [ ]  OpenMI Java 1.3 [ ]  OpenMI .Net 2.0 [ ]  OpenMI Java 2.0 [ ]  Comments: Such implementations are tasks for the model providers. However, we will provide a description for how to do such implementations (design pattern and description). <b>Stef</b> will make a first draft and <b>Jan</b> will complete it.  Implementation status:</p>
14	<p><b>Links should be owned by the delivering linkable component</b>  By Rob Brinkman</p> <p>Importance: high Impact: high Target: org.OpenMI.Standard</p> <p>Beneficial to: trainers, SmartWrapper and any developer who wants to overrule the link implementation, connection managers (e.g. distribution connector and ensemble manager)</p> <p>As a result of the interface approach, there is freedom to choose which class implements the interface. A linkable component can choose the class which is most appropriate for implementing the required functionality. For example, classes which implement the IQuantity, IElementSet, IExchangeItem and IValueSet are the responsibility of the linkable component. In Delft we use several classes which implement IElementSet.</p> <p>There is one exception to this methodology, being the ILink. In practice the link is created by a surrounding application (OmiEd.exe), which cannot do something else than creating a default link, being the backbone implementation. However, we have experienced that we would like to add additional functionality to the link (Actually, the smart wrapper had the same problem and therefore introduced the smart link; the smart wrapper would have been simpler if it could have defined its own links) To accomplish this, we propose the following:</p> <ol style="list-style-type: none"> <li>When a link is needed, ask the delivering component</li> <li>Regard the link as property of the delivering component. Note that all knowledge is implemented still by the delivering linkable component (actually a port or socket would be a better name, but for historical reasons we should keep the name link)</li> </ol> <p>If we regard the link as a socket or port for data, a logical consequence is that we ask the link for data. In other words, we move the GetValues call from the linkable component to the link. Also for this reason the need for a target component and a source component is gone.</p> <p>This results in the following changes in the interfaces. The quantity and element set in the link are the quantity and element set of the accepting component.</p> <p><a href="#">public interface IlinkableComponent</a></p> <pre> public interface IlinkableComponent {     ILink GetLink( string linkID );     ILink GetLink( IQuantity quantity,     IElementSet elementSet, IDataOperation[] dataOperations ); } </pre> <p><a href="#">public interface Ilink</a></p> <pre> public interface Ilink {     ILinkableComponentSourceComponent     {         Unknown macro: {get;}     }     IValueSet GetValues( ITime time ); } </pre>	<p>Accepted [A], Rejected [R] or Considered [C]  for implementation in version:  OpenMI .Net 1.3 [ ]  OpenMI Java 1.3 [ ]  OpenMI .Net 2.0 [C]  OpenMI Java 2.0 [C]  Comments:  Implementation status:</p>
15		<p>Accepted [A], Rejected [R] or Considered [C]  for implementation in version:  OpenMI .Net 1.3 [ ]  OpenMI Java 1.3 [ ]  OpenMI .Net 2.0 [C]  OpenMI Java 2.0 [C]  Comments:  Implementation status:</p>

## Caller doesn't have to be a linkable component

By: Rob Brinkman

*Importance:* medium *Impact:* medium *Target:* org.OpenMI.Standard

*Beneficial to:* trainers, visualization tools, result saving tools, possibly connection managers

Currently it is necessary that the caller of a linkable component is a linkable component itself. There is no functional need for this, because the callee should answer to the GetValues call, no matter who called it. It is also inconvenient to implement the complete ILinkableComponent as it is not used in reality by a variety of applications (especially visualization tools).

In the previous issue we removed the source and target component from the link. This makes it easier to allow for callers which are not linkable components. However, there is some functionality which is invoked on the caller's side, being the earliest input time. This method has become unreachable, because we removed the target component from the link.

As a linkable component doesn't know its caller, while the caller does know the link (from its internal administration), it seems logical to move this functionality to the link and change the calling logic. From now on the caller should set an appropriate value for the earliest needed time. Its implementation is now in the link, which is a better place because most often it will clear a buffer in the link.

In principle a linkable component caller can be any kind of object. Custom applications can be built which create any kind of linking. Only in the case that a general user interface is used or that a general composition reader is used, we need to perform some actions on the caller. This involves management of links and supporting the user in making right choices (just like it is done now).

Note that on composition level there is still a need for knowing the source and target of a link. Only then a generic application will be capable of instantiating a link based on an xml file. The xml file structure of a composition therefore will not change: To create a link it will invoke the GetLink method of the source linkable component and use the AddLink method in the caller to add the link to the target linkable component (Such an application will need its own administration of links, because otherwise it is not possible to find out which links have been established).

`public interface ILinkableComponentCaller`

```
{
    int InputExchangeItemCount {{color:blue}get;}
    IExchangeItem GetInputExchangeItem(int index);
    void AddLink(ILink link);
    void RemoveLink(string linkID);
}
```

`public interface ILinkableComponent`

```
{
    int InputExchangeItemCount

    Unknown macro: {get;}

    IInputExchangeItem GetInputExchangeItem(int index);
    void AddLink(ILink link);
    void RemoveLink(string linkID);
    ITimeStamp EarliestInputTime
```

```
}
public interface ILink
{
    ITimeStamp EarliestInputTime {{color:blue}set;}
} | Accepted [A], Rejected [R] or Considered [C]
```

for implementation in version:

OpenMI .Net 1.3 [ ]

OpenMI Java 1.3 [ ]

OpenMI .Net 2.0 [C]

OpenMI Java 2.0 [C]

Comments:

Implementation status: |

| 16 | **Stay close to the concept "Ask for values as needed by the requesting linkable component"**

By Rob Brinkman

*Importance:* high *Impact:* high *\_Target:* org.OpenMI.Standard

*Beneficial to:* trainers, SmartWrapper, any mapping functionality, future developments

The most important basic rule of OpenMI is that you ask a component for values of a quantity, define their location and specify how they are calculated. In OpenMI terminology you ask for values at a ITime for a IQuantity on an IElementSet using several IDataOperations.

During the development of OpenMI we drifted away from this principle, which is regarded as one of the strengths of OpenMI. This was done for many good reasons, for example in the beginning a typical usage of OpenMI was to link two ID based element sets. In order to perform appropriate mapping, the source element set was specified. But as we use geographical mappings nowadays, the source element set is not of any importance and confuses a lot of people. Actually, if you want to deliver data on an ID based element set, the source element set is part of *how* you map two ID based element sets and therefore should be part of a data operation

As links define the necessary meta data for the GetValues call, links should only hold data for doing this. Therefore links don't need to have a property for the target component, since this should not have any influence on handling the GetValues call.

Actually, all information needed for defining how data are to be calculated should be stored in data operations. This is the case for element set mappings, but also for quantity mappings. This removes the need for having a source quantity as property of the link. The link is changed as follows (quantity and element set are the former target quantity and element set):

`public interface ILink`

```
{
    IQuantity TargetQuantity

    Unknown macro: {get;}

    IElementSet TargetElementSet

    IQuantity SourceQuantity
```

Unknown macro: {get;}

ICollectionSourceElementSet

ICollection Quantity {{color:blue}get;}

ICollectionElementSet {{color:blue}get;}

}

A common data operation will be the quantity mapper. This data operation is needed almost always, but we think it is good to realize that it is a data operation. Unit conversion should be performed by this data operation.

As a consequence, we doubt the functionality of the IOutputExchangeItem. Its properties reflect very much the approach that a source element set is needed. For a geographical mapping this is not needed.

However, in user interfaces you need something to apply the right data operations. To support a user interface, you need the following:

- The available data operations for a certain input exchange item.
- The available options for a certain data operation (e.g. the source quantities for a quantity mapping data operation)

The first requirement is achieved by additional methods in the linkable component. The second requirement is realized by using the arguments of a data operation. An argument is extended with choices, i.e. the possible values from which the user can choose an appropriate value. So for quantity mapping, there is one argument named "Source Quantity" with all source quantities as possible values. This leads to the following changes in interfaces:

[public interface](#) ILinkableComponent

{

intOutputExchangeItemCount

Unknown macro: {get;}

IOutputExchangeItemGetOutputExchangeItem(intindex);

[int](#) GetDataOperationCount (IexchangeItem

exchangeItem);

IDataOperation GetDataOperation ( [int](#) dataOperationIndex,

IExchangeItemexchangeItem);

}

[publicinterface](#) IArgument

{

[string](#)[] Choices {{color:blue}get;}

}

[public interface](#)IOutputExchangeItem

[public interface](#)IInputExchangeItem

Note that we don't need to distinguish between an IInputExchangeItem and an IExchangeItem any more.

We will use IExchangeItem only.

| Accepted [A], Rejected [R] or Considered [C]

for implementation in version:

OpenMI .Net 1.3 [ ]

OpenMI Java 1.3 [ ]

OpenMI .Net 2.0 [C]

OpenMI Java 2.0 [C]

Comments:

Implementation status: |

| 17 | **Data operation hardly used**

By: Rob Brinkman

*Importance:* high *Impact:* low *Target:* org.OpenMI.Standard

*Beneficial to:* any mapping functionality

The data operation is an important issue in OpenMI, but nevertheless we hardly use it. An important reason is that the commonly used transformations are not defined in a data operation, but in the link itself (ID based element set mapping and quantity mapping). But given the change above that also these transformations are stored in a data operation, they should be used more often.

Another reason is that data operations are definitions only. When you want to use them, you should find the appropriate routine for each data operation ID. This should be repeated in each linkable component. It would be easier if the data operation was capable of doing a data transformation itself. In other words, there should be an additional method which takes care of this:

[publicinterface](#) IDataOperation

Unknown macro: {\ IValueSetGetConvertedValues (IValueSet source);\ }

A quantity mapper would perform unit conversion in this method. An element set mapper between ID based element sets would consider the order of elements in both element sets and fill in the resulting value set. A geographical mapper would invoke the right interpolation routine in the spatial utility (actually, the class ElementMapper would implement the IDataOperation interface).

The quantity mapper class (implemented in the backbone) would look as follows:

```
public class QuantityMapper : IDataOperation
{
    IValueSetGetConvertedValues (IValueSet source) {}
    IQuantitySourceQuantity {{color:blue}get; set;}
    IQuantityTargetQuantity {{color:blue}get; set;}
}
```

How would a user interface work with these new concepts? If we look at OmiEd.exe, there are two lists of exchange items. The list on the right side, representing the input exchange items, remains unchanged. However, on the delivering side we don't have exchange items any more, so this list will be removed. Instead of this list there will be a number of data operations (retrieved by the call GetDataOperation on the source linkable component).

These data operations are presented to the user. Often a data operation has a property which should be set by the user (e.g. the source quantity for the quantity mapper). The user interface can detect this from its arguments: there will be an argument called "Source Quantity" with choices all possible source quantities. The user interface will present this to the user in a list or combo box.

In this way the source quantity can be filled. But what about the target quantity, which should be filled usually with the quantity of the accepting component? This value is filled in automatically in the GetDataOperation call (which has an argument exchange item, which is the input exchange item of the accepting side). Since the linkable component knows its own data operations, it can fill in the right property. [ Accepted [A], Rejected [R] or Considered [C]

for implementation in version:

OpenMI .Net 1.3 [ ]

OpenMI Java 1.3 [ ]

OpenMI .Net 2.0 [C]

OpenMI Java 2.0 [C]

Comments:

Implementation status: |

# **IUnit and IDimension**

By: Rob Brinkman

Importance: medium Impact: low

Target: org.OpenMI.Standard

Beneficial to: trainers

Now there is no way to find out the dimension of a unit, if you start from a unit. This isn't logical in real life; a unit always refers to a dimension.

We think it is simpler to combine the two interfaces. There will be a new method comparing dimensions, to be used for validation. The resulting interface changes as follows:

```
public interface Iunit
```

```
    double GetPower(DimensionBase baseQuantity);\n    bool DimensionEquals (IUnit\n    otherUnit);\n}
```

```
public interface Iquantity
```

```
{\n    IDimensionDimension
```

```
}
```

```
public interface IDimension
```

19	<p><b>Combined quantities and ValueSets</b>  By: Rob Brinkman  Importance: high Impact: medium Target: org.OpenMI.Standard  Beneficial to: data assimilation, water quality modelling, water allocation, vector fields  Often there are quantities which are related closely to each other. For example, on a source the concentration of the added water amount is defined, but you can only use this value if you know the amount of water (discharge) too. They must be defined on the same element set and if you link to the concentration in this example, you must link to the discharge too. Although you can enforce this by validation, it would be convenient if you were able to link them in one go. This is convenient both for validation as for user interfaces.  Another issue is the usage of the IValueSet. In practice most often a scalar set is used. But sometimes we use a vector set too and discover that a lot of already implemented software is not capable of handling this (for example the wrapper). A solution is to check what kind of value set is in focus and then perform an appropriate task (this is the way the ElementMapper works). If we define a new class implementing IValueSet, we have to do this again. In fact we want obtain the array of doubles on a high level. This array of doubles represent the actual values for further processing, either in a data operation for interpolation or in the computational core.  These issues are combined, because the same improvement can be applied here. The closely related quantities are regarded as a combined quantity. A vector quantity can be seen as a combined quantity of three scalar quantities. In fact a quantity set is a quantity with several parts (X, Y, Z for vector quantities). This leads to the following changes:</p> <ol style="list-style-type: none"> <li>1. A quantity contains several parts (a part is defined by a string)</li> <li>2. The IValueSet contains an array of doubles (instead of the IScalarSet and IVectorSet)</li> <li>3. The number of values in the IValueSet array is the number of elements in the element set times the number of components in the quantity and interfaces:</li> </ol> <pre>public interface IQuantity</pre> <div>Unknown macro: {\      ValueType ValueType</div> <div>\      string GetQuantityPart(int quantityPartIndex);\      IUnit GetUnit(int quantityPartIndex);\ }</div> <pre>public interface IValueSet</pre> <div>Unknown macro: {\      int Count;\      int QuantityPartCount</div> <div>\      bool[] Valid</div> <div>GetValues(int quantityPartIndex);\      double GetValue(int quantityPartIndex, int elementIndex);\ }</div> <pre>public enum ValueType public interface IScalarSet public interface IVectorSet</pre> <p>The latter two methods in the IValueSet are convenience methods; they return a slice or single value of the value set. The main method is the Values property. There must be a rule how to interpret the array, since you can choose between (x,y,z,x,y,z,x,y,z,x,y,z) and (x,x,x,x,y,y,y,z,z,z,z) for vector quantities. We prefer the latter one, since that enables easier subarray extraction.  Note that there is no difference for the simple quantity (i.e. not combined, delivering a scalar set). The resulting value set will be very similar to a scalar set.</p>	<p>Accepted [A], Rejected [R] or Considered [C]  for implementation in version:  OpenMI .Net 1.3 [ ]  OpenMI Java 1.3 [ ]  OpenMI .Net 2.0 [C]  OpenMI Java 2.0 [C]  Comments:  Implementation status:</p>
20	<p><b>Time series</b>  By: Rob Brinkman  Importance: medium Impact: medium Target: org.OpenMI.Standard  Beneficial to: data source components  In some situations it is necessary to obtain time series from a delivering component. Typically such a component is an input file reader or a database component, but this is not necessarily the case.  To obtain a time series at the moment, one should do a GetValues for each ITime in the time series. Although possible, one can imagine that this is an inconvenient and performance consuming task. It would be more convenient if it were possible to obtain the time series in one go.  To do this, we can extend the IValueSet again (like we did for the combined quantities). The array of doubles returned by the IValueSet contains values for all ITimes requested in the GetValues call. The parameter ITime in the GetValues call is replaced by ITime[] (the keyword params allows that on the calling side a single ITime is passed instead of an array, just like now; on the implementation side one should use the array of ITimes always).</p> <pre>public interface Ilink</pre> <div>Unknown macro: {\      IValueSet GetValues(params ITime[] time);\ }</div> <p>The size of the array in the IValueSet is the number of ITimes in the GetVales call times the number of elements in the element set times the number of quantity parts in the quantity. See the following examples:  Asking for two times for a simple quantity on an element set of three elements would return an array like : <math>x_{t=1,e=1}, x_{t=1,e=2}, x_{t=1,e=3}, x_{t=2,e=1}, x_{t=2,e=2}, x_{t=2,e=3}</math>  Asking for multiple times (2) for a combined quantity (x,y) on an element set (3) would return an array like : <math>x_{t=1,e=1}, x_{t=1,e=2}, x_{t=1,e=3}, y_{t=1,e=1}, y_{t=1,e=2}, y_{t=1,e=3}, x_{t=2,e=1}, x_{t=2,e=2}, x_{t=2,e=3}, y_{t=2,e=1}, y_{t=2,e=2}, y_{t=2,e=3}</math></p>	<p>Accepted [A], Rejected [R] or Considered [C]  for implementation in version:  OpenMI .Net 1.3 [ ]  OpenMI Java 1.3 [ ]  OpenMI .Net 2.0 [C]  OpenMI Java 2.0 [C]  Comments:  Implementation status:</p>



21	<p><b>Hot starts</b>  By: Rob Brinkman  Importance: high Impact: low Target: org.OpenMI.Standard  Beneficial to: any restart of a model combination which needs time for stabilization, flood forecasting, filtering /data assimilation...</p> <p>Models need initial conditions to start with. As shown by the HarmoniLake case, inappropriate initial conditions can create stability problems which are difficult to overcome as models influence each other. A facility that allows a model combination to store its joint equilibrium state persistently and use it for other simulations (e.g. scenario runs) would drastically improve applicability.</p> <p>The current IManageState interface is very much oriented towards run-time state management e.g. for iteration purposes. It does not guarantee that a state is stored in a persistent way. Therefore a separate (additional) interface is suggested. This interface could be as follows:</p> <pre>public interface IPersistentState {     void SaveStateToDisk(string stateReference);\     bool LoadStateFromDisk (string stateReference);\ }</pre> <p>This interface is used just before the run. The deployer (or another piece of software triggering the run) checks if this interface is implemented by a linkable component and if so, loads the state. The state reference is retrieved from the omi file by a new, optional xml attribute.</p>	<p>Accepted [A], Rejected [R] or Considered [C]  for implementation in version:  OpenMI .Net 1.3 [ ]  OpenMI Java 1.3 [ ]  OpenMI .Net 2.0 [C]  OpenMI Java 2.0 [C]  Comments:  Implementation status:</p>
22	<p><b>Integers for the QuantityID and ElementSetID</b>  By: wmdaniels (wmdaniels)  My specific suggestion is that SetValues and GetValues are made as efficient as possible as they are called many times. I would suggest using integers for the QuantityID and ElementSetID instead of strings. If any other functions are called many times then these should be looked at as well.</p>	<p>Accepted [A], Rejected [R] or Considered [C]  for implementation in version:  OpenMI .Net 1.3 [ ]  OpenMI Java 1.3 [ ]  OpenMI .Net 2.0 [C]  OpenMI Java 2.0 [C]  Comments:  Implementation status:</p>
23	<p><b>Equals methods in backbone</b>  By: Rob Brinkman  Importance: medium  Impact: low  Target: org.OpenMI.Backbone  Beneficial to: simultaneous access to same exchange items in various runs  A lot of classes in the backbone implement the Equals method. They have been implemented in the right way as long as there is one run involved.</p> <p>We have a user interface where results of several runs can be compared. An essential part in this is retrieving the output exchange items and offering them to the user. But implementation has become difficult, because same quantity element set combinations from different linkable components are considered to be equal.</p> <p>Especially for the exchange item we would like the Equals method to be removed.</p>	<p>Accepted [A], Rejected [R] or Considered [C]  for implementation in version:  OpenMI .Net 1.3 [ ]  OpenMI Java 1.3 [ ]  OpenMI .Net 2.0 [C]  OpenMI Java 2.0 [C]  Comments:  Implementation status:</p>
24	<p><b>Backbone.Quantity constructor should use IUnit</b> Currently the constructor of the Quantity-object (org.OpenMI.Backbone) uses at the same time a Unit-clas and an IDimension-interface. The constructor should adopt the interfaces (and thus IUnit) instead of the classes.</p>	<p>Accepted [A], Rejected [R] or Considered [C]  for implementation in version:  OpenMI .Net 1.3 [ ]  OpenMI Java 1.3 [ ]  OpenMI .Net 2.0 [C]  OpenMI Java 2.0 [C]  Comments:  Implementation status:</p>