

OpenMI Association Technical Committee meeting no 23

See also: [OATC Wiki Home](#)

Date: April 21.4 - 23.4, 2009

Venue: [DHI](#), Hørsholm, Agern Allé 5, Denmark

Participants:

[Rob Knapen](#), Alterra, Wageningen UR (Rob.Knapen@wur.nl)
[Standa Vanecek](#), DHI (s.vanecek@dhi.cz)
[Adrian Harper](#), Wallingford Software (adrian.harper@wallingfordsoftware.com)
[Stef Hummel](#), Deltares (stef.hummel@deltares.nl)
[Unknown User \(don\)](#), Deltares (gennadii.donchyts@deltares.nl)
[Jesper Grooss](#), DHI (jgr@dhigroup.com)

Apologies:

[Peter Schade](#), BAW (peter.schade@baw.de)
[Andrea Antonello](#) (andrea.antonello@hydrologis.eu)
[Jan Gregersen](#), LicTek (gregersen@lictek.dk)
[Unknown User \(onnooroos\)](#), Alterra (Onno.Roosenschoon@wur.nl)
[Unknown User \(jnh@dhigroup.com\)](#) Johan Hartnack , DHI (jnh@dhigroup.com)

Documents:

<http://www.openmi.org/>
<http://sourceforge.net/projects/openmi>
wiki.openmi.org

Table of contents

- [Participants:](#)
- [Documents:](#)
- [Table of contents](#)
- [Minutes](#)
- [1. Minutes from previous OATC meeting](#)
- [2. Maintenance and support](#)
- [3. OpenMI 2.0 Issues](#)
 - [3.1 Road map to 2.0 alpha version](#)
 - [3.1.1 Changest to the standard](#)
 - [3.1.2 Backbone](#)
 - [3.1.3 Buffer](#)
 - [3.1.3 Time interpolation Decorator](#)
 - [3.1.4 GUI](#)
 - [3.1.5 SimpleRiver Model - running in GUI](#)
 - [3.1.6 OpenMI 2.0 wrapper around OpenMI 1.4 Components](#)
 - [3.2 Pending Remarks on Standard as result of Java review](#)
 - [3.3 Suggestions for the states and possible transitions between the states of the ILinkableComponent](#)
 - [3.4 Limitations of time representation in OpenMI \(by W. de Winter \(\[wim.dewinter@wur.nl\]\(mailto:wim.dewinter@wur.nl\)\)\)](#)
 - [3.5 More suggestions on v2.0 changes](#)
 - [3.6 The ITime Issue](#)
 - [3.6.1 Time Interface\(s\)](#)
 - [3.6.2 Platform specific time or Modified Julian?](#)
- [4. Decorators](#)
- [5. OATC Procedures](#)
- [6. \[www.OpenMI.org\]\(http://www.OpenMI.org\) and \[wiki.OpenMI.org\]\(http://wiki.OpenMI.org\)](#)
- [7. Miscellaneous issues](#)
 - [7.1. OpenMI Goes Linux](#)
 - [7.1.1. The OpenMI Linux version ...](#)
 - [7.1.2. The Linux HowTo pages are on the wiki](#)
- [8. Tasks and unresolved issues](#)
- [9. Any other business](#)

Minutes

1. Minutes from previous OATC meeting
2. Maintenance and support

3. OpenMI 2.0 Issues

3.1 Road map to 2.0 alpha version

3.1.1 Changest to the standard

- Time interfaces, according to suggestion I, see [Section 3.6.1](#)
- Elements - HasZ, HasM
- Decorator - Factory,....
- Keep, Save and Clear state, convert to/from byte[] to make persistent

To be done by: **Gena**

Will be finished by: april 27th

3.1.2 Backbone

To be done by: **Stef** and/or **Rob B.**

Will be finished by: may 8th

3.1.3 Buffer

To be done by: **Stef** and/or **Rob B.**

Will be finished by: may 8th

3.1.3 Time interpolation Decorator

To be done by: **Stef** and/or **Rob B.**

Will be finished by: may 15th

3.1.4 GUI

To be done by: **Adrian**

Will be finished by: try before the next meeting (june 8th).

3.1.5 SimpleRiver Model - running in GUI

To be done by: **Adrian**

Will be finished by: try before the next meeting (june 8th).

3.1.6 OpenMI 2.0 wrapper around OpenMI 1.4 Components

To be done by: **Stef** and **Jesper**

Will be finished by: try before the next meeting (june 8th).

Schedule:

- before may 22nd Stef will finish the already create Upwards compatibility 'kit' [See UpwardsComp in SVN](#)
- after that, Jesper will make the wrapper work [See UpwardsComp/EngineWrapper in SVN](#).

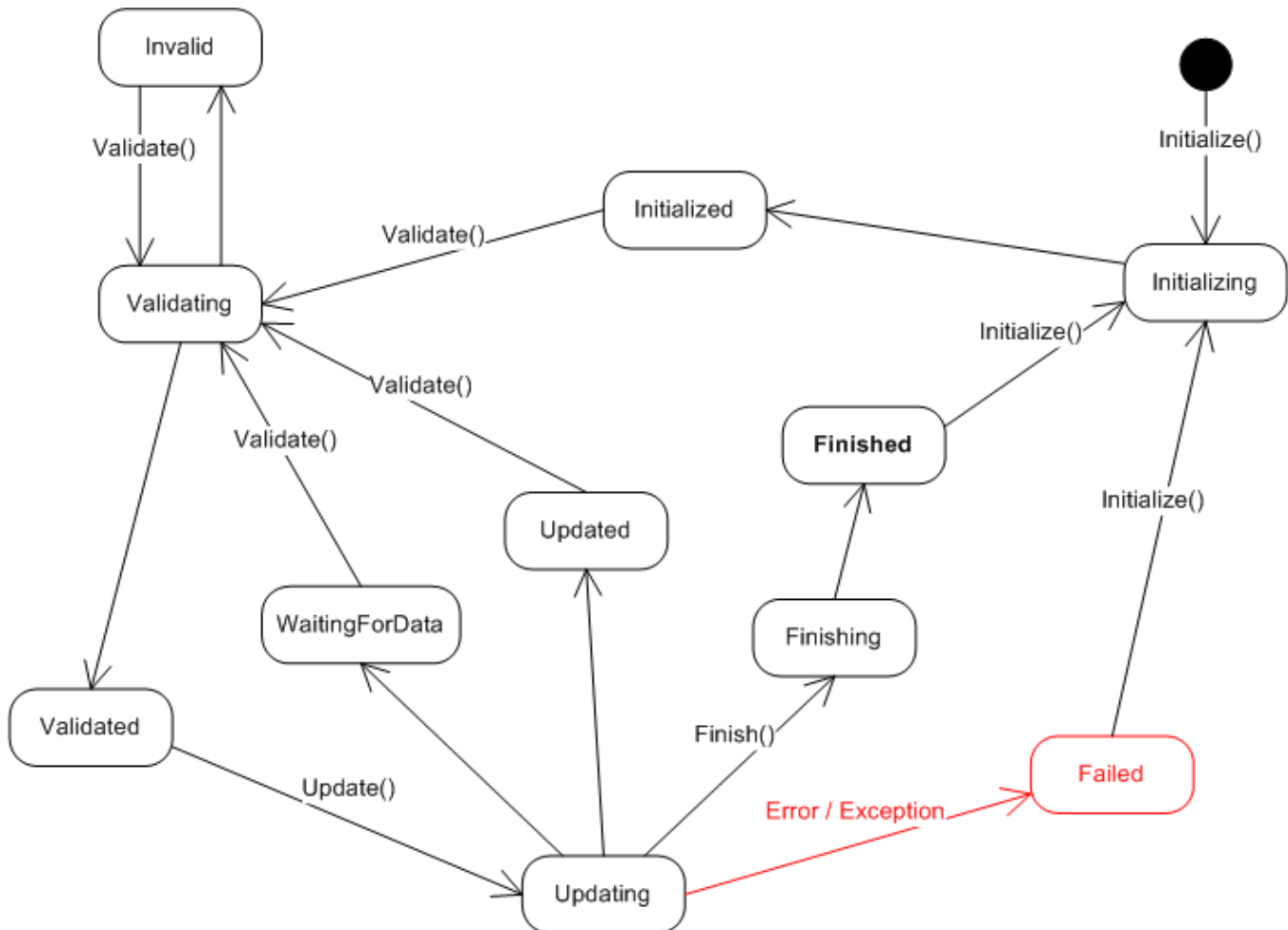
3.2 Pending Remarks on Standard as result of Java review

This led to some remarks, of which a few were "java only", but some of them reflected on the Standard in general. Items to be addressed at the next meeting:

1. Question: what does isAvailable() mean in case of multiple consumers. Do all consumers need to have the same time(s)?
 - a. Leave in for now, review after more implementation, one suggestion of modification bool isAvailable(whosAsking);
 - b. ⚠️ Will be discussed later during the meeting, Stef will show an example of using isAvailable(whosAsking);
2. Question: Collection or Set instead of List for (In|Out)putExchangeItems in LC?
 - a. It will be **ICollection** 👍 .
3. Question: Introduce IState instead of Object? This will emphasize the relation between IManageState and IPersistentState. Additional suggestion: IPersistentState extends IState.
 - (b.t.w: makePersistent / readPersistentState are wrong names, should be something like getAsByteStream)
 - a. Gena has discussed with Rob between last and current meeting. After some discussion the final result 👍 is described on: [Suggestion about changed to state management](#)
4. To be discussed: Introduce generics in IOputItem.Values()? i.e. weakly typed via <T>
 - a. 🙋 We will not introduce generics in the standard. However, they could be used in the SDK.
5. Remark: Working with providers/consumers, we should take care that such a circularity might give garbage collection problems. After a short discussion, we see no real general way so solve this risk, so it has been decided that we should leave this up to the Garbage Collector.
 - Point noted, no specific agreement as to solution

6. 👉 We will introduce something like `OpenMIException : Exception`. However, first we will have a detailed look at the guidelines for introducing custom exceptions in both .Net and Java. Rob K. has added a page for this, [Guidelines for introducing custom exceptions](#). It now includes Java suggestions, ⚠️ Gena will add .Net suggestions.

3.3 Suggestions for the states and possible transitions between the states of the ILinkableComponent



*If component has **Invalid** status and we call `Validate()` - it may even call `Update()` on another components in order to validate itself.*

3.4 Limitations of time representation in OpenMI (by W. de Winter (wim.dewinter@wur.nl))

The [Limitations of time representation](#) was discussed as part of the time issue, see 3.6.2.

3.5 More suggestions on v2.0 changes

- Move `IsAvailable` to `IXchangeItem`. Is it only for a thread-safety? If yes than see also `ICollection.IsSynchronized`

Stef: The question is confusing, the actual issue is that we should distinguish between `IsAvailable` and `IsValid`. We are now realized that there is a difference between `IsAvailable` (only for output item, means it is for consumers) `IsValid` (probably for `IXchangeItem`, meaning if item is valid in the context of current component, maybe updated during `Validate()` call)?

- Move `Values { get; set; }` to `IXchangeItem`
 - ⚠️ `get; set;` to be reviewed at this meeting

When discussing `get/set`, the subject of the discussion gradually moved to the pull versus loop driven approach. This led to an important decision 👉 : We will introduce a boolean argument that indicates whether the component should run in a pull mode (like in OpenMI 1.4) or in loop mode

: [Discuss component operation mode here](#)

- Add **params** keyword to IArguments[] args parameter in Initialize(IArgument[] args).
 - Adrian review, if happy then ... ?👍
- "string Validate()" / "bool IsValid(IExchangeItem item)" / "IsAvailable()", etc.
 - ⚠️ to be reviewed at this meeting
- 👍 Add "params IOutputItem[] items" argument to Update() in order to specify explicitly what should be updated.
 - ⚠️ Maybe generalise to IExchangeItem i.e. allow IInputItem?
- Remove AddConsumer/RemoveConsumer and use BindingList in the implementation instead to listen to IOutputItem.Consumer.Add() / Remove()

We will leave AddConsumer/RemoveConsumer for now. The BindingList mechanism has been implemented, though, so anyone can use it

- When we get loops in the compositions - we have to tell which component will be first! In the new GUI, we will let the user indicate the 'Start up component'.
- Add Durations, TimeZone properties to ITimeSet? (see also changes from Rob in Java version)
- Replace all ITime, ITimeStamp, ITimeSpan by: System.DateTime, System.TimeSpan. Should be straightforward. For components working in Julian only time in FORTRAN CalendarConverter can be used. But for .NET all nice features of the .NET date time, time span will be available, e.g. DateTime.AddDays(int), DateTime.AddHours(int), time1 > time2, time1.AddDays(1) > DateTime.Now, and so on.
 - ⚠️ all time issues will be discussed at this meeting
- Change XYPoint+XYZPoint, ZYPolyline+XYZPolyline => Point, Polyline (make it look more like OGC entities)
 - Reviewed: OGC approach to see how we know when a Z coordinate exists:
 - 👍 IElementSet.SpatialReferenceWkt contains Z (vertical reference) and M systems, see http://en.wikipedia.org/wiki/Well-known_text.
 - To get the M-coordinate, a method should be added:

```
...
double GetVertexMCoordinate(int elementIndex, int vertexIndex);
...

// proposal:
// the four methods could be replaced with this:
public enum SpatialDimension { X, Y, Z, M }
public double getVertexCoordinate(int elementIndex, int vertexIndex, SpatialDimension dim);
```

- IIdentity has been split in IIdentifiable and IDescribable. Some interfaces already have been derived from IIdentifiable and/or IDescribable** ? At some stage, we should check all interfaces on whether to derive from IIdentifiable and/or IDescribable or not.
 - 👍 At this meeting, we have checked that IIdentifiable and/or IDescribable are not used as separate entities.
- IArgument needs a mechanism for identifying the type of argument e.g. File, Path, int double etc Very useful for providing customised GUI functionality
 - ? Adrian's proposal is below

```

namespace OpenMI.Standard
{
    public enum EArgType
    {
        String = 0, // default
        Bool,
        Int,
        Double,
        Path, // presumed to exist and accessible
        FilePath, // presumed to exist and accessible
        PathNew,
        FilePathNew,
        XML, // Rob: can be used together with a schema for complex argument data
        MIME, // Rob: might be useful to pass in images, or text documents, etc.
    }

    /// <summary>
    /// The IArgument interface defines a key - value pair. If the property ReadOnly is
    /// false the value is editable otherwise it is read-only.
    /// </summary>
    public interface IArgument : IDescribable
    {
        /// <summary>
        /// Type that string value represents and can be converted to
        /// </summary>
        EArgType ValueType { get; }

        /// <summary>
        /// The key (string) in key-value pair.
        /// </summary>
        string Key { get; }

        /// <summary>
        /// <para>The value (double) in key-value pair.</para>
        /// <para>If the ReadOnly property is true and the property is attempted to be
changed
        /// from outside an exception must be thrown.</para>
        /// </summary>
        string Value { get; set; }

        /// <summary>
        /// Defines whether the Values property may be edited from outside.
        /// </summary>
        bool ReadOnly { get; }
    }
}

```

- Using empty *Temporal* and *ElementSet* in the Input items.
See <http://public.wldelft.nl/display/OPENMI/Empty+Sets+in+the+Input+Items+-+Element+set+changing+in+time>
- Changing of the ElementSet during the time and different Times (steps) for different Elements
 - For the Time can be probably solved by missing values

3.6 The ITime Issue

3.6.1 Time Interface(s)

After some discussions, there were four proposals. See list below (including the original 1.4 situation).
Note: naming to be done yet.

Suggestion 2.0, I

```
public interface ITime {
    double StampAsModifiedJulianDay { get; }
    double DurationAsModifiedJulianDay { get; }
}
public interface ITimeSet {
    List<ITime> Times { get; }
    bool HasDurations { get; }
}
```

Suggestion 2.0, II

```
public interface ITime {
    double StampAsModifiedJulianDay { get; }
}
public interface ITimeSpan : ITime {
    double DurationAsModifiedJulianDay { get; }
}
public interface ITimeSet {
    List<ITime> Times { get; }
    bool HasDurations { get; }
}
```

Suggestion 2.0, III

```
public interface ITimeSet
{
    IList<double> TimeStamps { get; set; }
    IList<double> Durations { get; set; }

    TimeZoneInfo TimeZone { get; set; }

    bool HasDurations { get; }
}
```

Suggestion 2.0, IV

```
public interface ITimeSet
{
    IList<double> TimeStamps { get; }
    TimeZoneInfo TimeZone { get; }
}
public interface ITimeSpanSet : ITimeSet
{
    IList<double> Durations { get; }
}
```

Original 1.4 situation

```
public interface ITime
{
}
public interface ITimeStamp : ITime
{
    double ModifiedJulianDay {get;}
}
public interface ITimeSpan : ITime
{
    ITimeStamp Start { get; }
    ITimeStamp End { get; }
}
```

After studying the proposal, we have made a decision 👍. We will take version I.

3.6.2 Platform specific time or Modified Julian?

Adrian suggested that we incorporate as less platform specific things in the Standard as possible. This also holds for Lists and events. 🤔 We will be put the issue on the agenda for next meeting.

After some detailed discussion on the .Net.DateTime/Java.Date versus the MJD implementation, we have reached a conclusion 👍 : We will keep using the Modified Julian Day.

We should take into consideration whether we want to support a c++ version of the standard in the future.

4. Decorators

The decorators were discussed thoroughly, and we came up with some extensions to the standard.

See list of questions discussed on the following page: [Exchange item decorators](#).

Current changes to the standard:

```
// add a new interface:
public interface IExchangeItemDecoratorFactory
{
    IIdentifiable[] GetAvailableOutputDecorators(IOutputItem decoratedItem, IInputItem targetItem);
    IOutputItemDecorator CreateOutputItemDecorator(IIdentifiable decoratorId, IOutputItem decoratedItem,
    IInputItem targetItem);

    IDescribable GetDecoratorDescription(IIdentifiable id);
}

// add the following methods to the ILinkableComponent:
public interface ILinkableComponent
{
    ...

    IExchangeItemDecoratorFactory GetDecoratorFactory();

    void AddOutputItemDecorator(IOutputItemDecorator decorator);

    void RemoveOutputItemDecorator(IOutputItemDecorator decorator);
}
```

5. OATC Procedures

We will keep track of a [wish list for version 3.0](#).

6. www.OpenMI.org and wiki.OpenMI.org

Gena will move the content of the current wiki to the new wiki, and make a redirect page to the new wiki on the current one, and will ask the Greek to adjust the name server to let it refer to the new new wiki.

7. Miscellaneous issues

7.1. OpenMI Goes Linux

7.1.1. The OpenMI Linux version ...

Peter Schade achieved quite a bit of result, making nearly everything working. During a Skype session Peter explained the status to the other OATC members, who were quite impressed.

The Linux version has been committed to SourceForge: <http://openmi.svn.sourceforge.net/svnroot/openmi/branches/OpenMI-Version-1-Trunk/>

7.1.2. The Linux HowTo pages are on the wiki

<https://public.deltares.nl/display/OPENMI/How+to+port+the+OpenMI+from+Windows+to+Linux>

How to port the OpenMI environment from Linux to Windows:

- OpenMI standard
- sdk
- ConfigurationEditor
- some LCs of the examples

A set of Mono dlls and exe files was generated and tested on openSUSE 11.0 Linux. Interested persons are invited to test the set, especially the ConfigurationEditor on further (Linux) platforms supported by Mono.

<https://public.deltares.nl/display/OPENMI/How+to+generate+a+LinkableComponent+with+a+Fortran+engine+on+Linux>

How to generate a LC (Fortran 90 engine and C# wrapper) on Linux

Deltares will test the Linux version on Ubuntu and Fedora in about 3 weeks from now (say May 15th). Results will be reported at next meeting, after which Adrian will take care of the distribution.

8. Tasks and unresolved issues

All tasks are handled by sourceForge. GOTO: [OpenMI Tasks on source forge](#)

9. Any other business

- Check ideas mentioned in the poster of Jon Goodall - see meeting 22