



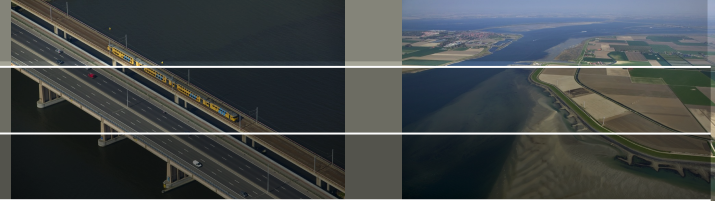
www.openda.org

OpenDA application

Martin Verlaan
Stef Hummel, Julius Sumihar, Albrecht Weerts,
Ghada El Serafy, Herman Gerritsen, ...
Deltares & TU Delft

Martin.Verlaan@deltares.nl

Outline



- Installation of OpenDA for Windows
- Starting the OpenDA application
- The GUI
- Input files
 - Main input file
 - Observations
 - Models
 - Algorithms
 - Results → output files

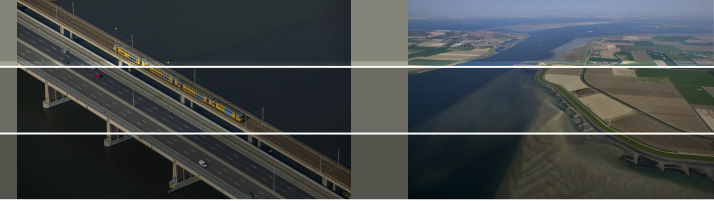
Installation of Windows binaries

1. Locate zip-file on sourceforge (use link on <http://www.openda.org>)
2. Download openda v2.0 windows binaries
3. Unpack zip-file (use a path without spaces)
4. Make a shortcut for run_openda_gui.bat
5. Test by starting the batch script

Known issues:

- A 32bit java environment is included to avoid issues with incompatible java versions.
- Sometimes existing directories in the PATH environment variable cause trouble.

Outline



- **Starting the OpenDA application:**
 - **Windows**
 - > **GUI** : `run_openda_gui.bat`
 - > **Dosbox** : `run_openda_app.bat my_input.oda`
 - > **Dobox MPI** : `run_openda_mpi.bat my_input.oda`
 - **Linux/MAC**
 - > **GUI** : `oda_run.sh -gui`
 - > **Command line** : `oda_run.sh my_input.oda`
 - > **Command line MPI** : `oda_run.sh -p my_input.oda`

GUI Input Tab

The screenshot displays the OpenDA application window titled "Enkf.oda - OpenDaApplication". The interface includes a "File Control" menu with "Open", "Save", "Start", "Stop", and "Pause" buttons. Below this is a tabbed interface with "Input", "Control", "Output", and "Cost function" tabs. The "Input" tab is active, showing a file tree on the left and an XML editor on the right. The XML editor contains the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Sample XML file generated by XMLSpy v2006 rel. 3 sp1 (http://www.altova.com)-->
<openDaApplication xmlns:oda="http://www.openda.org" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.openda.org http://www.openda.org/schemas/openDaApplication.xsd">
  <stochObserver className="org.openda.utils.CsvStochObserver">
    <workingDirectory>./stochobserver</workingDirectory>
    <!-- <configFile>OscillatorSimpleStochObsConfig.xml</configFile> -->
    <configFile>observations_lorenz96_generated.csv</configFile>
  </stochObserver>
  <stochModelFactory className="org.openda.models.lorenz96.Lorenz96StochModelFactory">
    <workingDirectory>./model</workingDirectory>
    <configFile>Lorenz96StochModel.xml</configFile>
  </stochModelFactory>
  <algorithm className="org.openda.algorithms.kalmanFilter.EnKF">
    <workingDirectory>./algorithm</workingDirectory>
    <configString>EnkfAlgorithm.xml</configString>
  </algorithm>
  <resultWriter className="org.openda.resultwriters.MatlabResultWriter">
    <workingDirectory>./</workingDirectory>
    <configFile>enkf_results.m</configFile>
  </resultWriter>
</openDaApplication>
```

Annotations with arrows point to the "Open" button, the "Input" tab, the XML editor, and the "Start", "Stop", and "Pause" buttons.

Open & save input files

Edit input file

Select input file

Start, stop and pause computations

GUI Control Tab

The screenshot displays the 'Control' tab of the OpenDaApplication GUI. The window title is 'Enkf.oda - OpenDaApplication'. The interface includes a menu bar with 'File' and 'Control', and a toolbar with buttons for 'Open', 'Save', 'Start', 'Stop', and 'Pause'. Below the toolbar are tabs for 'Input', 'Control', 'Output', and 'Cost function'. A secondary toolbar contains 'Open restart file', 'Save restart file', and 'Plot selected runs'. The main area is divided into 'Progress' and 'Log and output' sections. The 'Progress' section shows the current analysis time and state vector details. The 'Log and output' section displays a log of iteration results, with an 'Enable Logging' button. Three arrows point from text labels to specific elements: 'Progress of current computation' points to the progress bar, 'Output slows down computations' points to the 'Enable Logging' button, and 'Output of computation' points to the log text.

Enkf.oda - OpenDaApplication

File Control

Open Save Start Stop Pause

Input Control Output Cost function

Open restart file Save restart file Plot selected runs

Progress

analysis at 185811270000UTC (10.0)

length of state vector: 40.
number of observations: 40.
Application starting next step
Application Done

Log and output

xi_f_26 = [-0.726,2.964,4.732,0.577,3.783,7.16,5.374,1.469,-3.419,-2.035,...,2.034,2.176,-0.547,0.54,8.09,6.091,4.659,8.251,-0.206,-4.493]
xi_f_27 (iteration=0):
xi_f_27 = [-0.418,3.26,4.133,-0.027,3.455,7.07,5.565,1.353,-3.615,-1.937,...,2.529,2.467,-0.523,0.631,8.138,5.646,4.215,8.302,1.748,-4.647]
xi_f_28 (iteration=0):
xi_f_28 = [-0.442,3.381,3.809,-0.022,3.469,7.819,5.727,0.209,-3.422,-1.81,...,2.582,2.622,-0.696,0.548,8.052,6.412,4.91,8.297,-0.208,-3.945]
xi_f_29 (iteration=0):
xi_f_29 = [-0.679,2.979,4.467,0.505,3.747,7.53,5.509,0.754,-3.677,-1.709,...,1.771,2.15,-0.521,0.556,8.014,6.233,4.619,8.228,0.142,-4.622]
xi_f_30 (iteration=0):
xi_f_30 = [-0.44,3.004,4.713,0.424,3.732,7.307,5.553,0.931,-3.598,-1.674,...,2.347,2.398,-0.385,0.436,8.187,6.311,4.048,7.985,1.253,-4.829]
xi_f_31 (iteration=0):
xi_f_31 = [-0.656,2.939,4.768,0.584,3.979,7.631,5.274,1.151,-3.395,-1.928,...,1.522,1.991,-0.286,0.53,8.258,5.939,4.155,8.116,1.366,-4.959]
xi_f_32 (iteration=0):
xi_f_32 = [-0.704,3.015,4.603,0.344,3.64,7.432,5.554,0.873,-3.561,-1.788,...,2.26,2.427,-0.465,0.78,8.17,5.68,4.398,8.305,0.852,-4.509]
xi_f_33 (iteration=0):
xi_f_33 = [-0.824,2.843,4.542,0.78,3.854,7.292,5.38,1.062,-3.414,-1.882,...,2.462,2.253,-0.358,1.122,8.216,5.36,4.856,8.675,-0.59,-4.166]
xi_f_34 (iteration=0):
xi_f_34 = [-0.567,3.199,4.351,0.476,3.754,7.372,5.593,1.525,-3.377,-2.098,...,1.818,2.233,-0.328,0.812,8.357,5.793,4.455,8.306,0.526,-4.735]

Progress of current computation

Output slows down computations

Output of computation

GUI Output Tab

Click in a column to plot

Selection of output

Enkf.oda - OpenDaApplication

File Control

Open Save Start Stop Pause

| Input | Control | Output | Cost function | value[0] | value[1] | value[2] | value[3] | value[4] | value[5] | value[6] | value[7] | value[8] | value[9] | value[10] | value[11] | value[12] | value[13] | value[14] | value[15] | value[16] | value[17] | value[18] | value[19] |
|-------|---------|--------|---------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 1 | 0 | 2.943 | 8.279 | 5.614 | 0.905 | -3.838 | -1.952 | -2.315 | 7.808 | 3.146 | -8.412 | 0.998 | -1.902 | -0.14 | 6.383 | 7.104 | 3.854 | 3.467 | 2.542 | | | | |
| 2 | 0 | 1.122 | 5.162 | 2.138 | -5.262 | 1.572 | -2.989 | 0.573 | 12.373 | 1.847 | 1.202 | 4.59 | 1.677 | -2.889 | 0.89 | 0.151 | 6.449 | -0.847 | -2.606 | | | | |
| 3 | 0 | 2.407 | 5.354 | -2.52 | -2.762 | 3.308 | 1.219 | 0.587 | 11.49 | 2.992 | -1.402 | 5.332 | 1.002 | -1.527 | 1.841 | 3.874 | 5.334 | -2.544 | 0.766 | | | | |
| 4 | 0 | 1.873 | 5.723 | -3.077 | -2.228 | 2.53 | 0.6 | | | | | | | | | | | | | | | | |
| 5 | 0 | 3.597 | 2.734 | -3.749 | 0.852 | 3.287 | 3.1 | | | | | | | | | | | | | | | | |
| 6 | 0 | 4.013 | 1.735 | -4.035 | 0.19 | 3.086 | 4.3 | | | | | | | | | | | | | | | | |
| 7 | 0 | 2.086 | -2.597 | -1.324 | -0.479 | 3.729 | 8.7 | | | | | | | | | | | | | | | | |
| 8 | 0 | 1.648 | -2.693 | -1.236 | -0.634 | 3.136 | 8.7 | | | | | | | | | | | | | | | | |
| 9 | 0 | -2.472 | -0.062 | 0.478 | 0.704 | 3.849 | 7.1 | | | | | | | | | | | | | | | | |
| 10 | 0 | -2.497 | 0.045 | 0.485 | 0.714 | 3.772 | 7.0 | | | | | | | | | | | | | | | | |
| 11 | 0 | -2.828 | 2.546 | 3.456 | 2.936 | 5.131 | 1.9 | | | | | | | | | | | | | | | | |
| 12 | 0 | -2.889 | 2.141 | 3.186 | 2.755 | 5.074 | 2.6 | | | | | | | | | | | | | | | | |
| 13 | 0 | -0.322 | 1.495 | 5.796 | 5.642 | 1.904 | -1 | | | | | | | | | | | | | | | | |
| 14 | 0 | -0.286 | 1.46 | 5.8 | 5.618 | 1.435 | -1 | | | | | | | | | | | | | | | | |
| 15 | 0 | 0.311 | 2.902 | 7.642 | 1.425 | -2.855 | 1.1 | | | | | | | | | | | | | | | | |
| 16 | 0 | 0.251 | 3.056 | 7.785 | 1.053 | -2.682 | 1.1 | | | | | | | | | | | | | | | | |
| 17 | 0 | 1.735 | 4.687 | 5.731 | -4.038 | 1.217 | 2.1 | | | | | | | | | | | | | | | | |
| 18 | 0 | 1.687 | 4.733 | 5.609 | -4.127 | 1.331 | 2.2 | | | | | | | | | | | | | | | | |
| 19 | 0 | 5.313 | 5.22 | -1.511 | -3.147 | 1.033 | 6.4 | | | | | | | | | | | | | | | | |
| 20 | 0 | 5.403 | 5.189 | -1.554 | -3.047 | 1.14 | 6.5 | | | | | | | | | | | | | | | | |
| 21 | 0 | 6.524 | -0.134 | -3.681 | 0.419 | 0.592 | 7.3 | | | | | | | | | | | | | | | | |
| 22 | 0 | 6.631 | -0.253 | -3.603 | 0.322 | 0.621 | 7.6 | | | | | | | | | | | | | | | | |
| 23 | 0 | 2.104 | -4.343 | 0.612 | 0.936 | 2.753 | 8.2 | | | | | | | | | | | | | | | | |
| 24 | 0 | 2.138 | -4.338 | 0.663 | 0.972 | 2.826 | 8.2 | | | | | | | | | | | | | | | | |
| 25 | 0 | -4.057 | -0.527 | 0.506 | 3.214 | 5.535 | 1.9 | | | | | | | | | | | | | | | | |
| 26 | 0 | -4.12 | -0.42 | 0.552 | 3.274 | 5.479 | 2.0 | | | | | | | | | | | | | | | | |
| 27 | 0 | -3.628 | 2.241 | 4.247 | 5.071 | 2.598 | -4 | | | | | | | | | | | | | | | | |
| 28 | 0 | -3.54 | 2.202 | 4.019 | 4.902 | 2.871 | -4 | | | | | | | | | | | | | | | | |
| 29 | 0 | -1.982 | 1.058 | 6.918 | 2.638 | -3.364 | -1.712 | 0.032 | -0.047 | 6.67 | 0.134 | -4.007 | 0.305 | 5.831 | 4.227 | -5.144 | -0.279 | -1.44 | 4.937 | | | | |
| 30 | 0 | -2.017 | 0.975 | 7.047 | 2.468 | -3.405 | -1.638 | -0.01 | -0.113 | 6.469 | 0.502 | -4.14 | 0.443 | 6.09 | 3.938 | -5.201 | -0.258 | -1.367 | 4.808 | | | | |
| 31 | 0 | -0.651 | 0.669 | 7.424 | -1.381 | -1.644 | 0.078 | 1.106 | 1.952 | 6.381 | -2.119 | -0.408 | -1.305 | 6.042 | 0.035 | -4.169 | 2.027 | 3.431 | 5.834 | | | | |
| 32 | 0 | -0.551 | 0.568 | 7.451 | -1.288 | -1.649 | 0.177 | 1.141 | 1.874 | 6.371 | -2.006 | -0.597 | -1.315 | 6.055 | -4.8... | -4.143 | 2.03 | 3.472 | 5.802 | | | | |
| 33 | 0 | 1.202 | 2.39 | 7.106 | -1.062 | 1.72 | 1.802 | 2.843 | 4.532 | 4.15 | -1.963 | 2.958 | 2.679 | 5.922 | -1.914 | -1.039 | 0.369 | 6.067 | 6.28 | | | | |

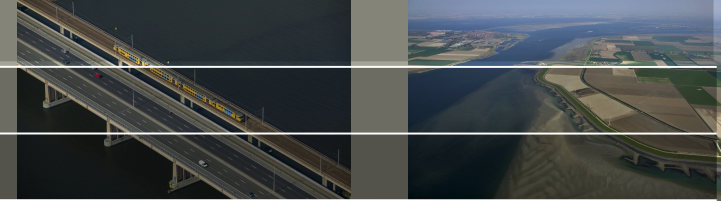
Calibration parameter

value[1]

value[1] 0 4.0 8.0

iteration (index of model run)

Main input file (my_input.oda)



```
<?xml version="1.0" encoding="UTF-8"?>
<openDaApplication xmlns="http://www.openda.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openda.org
  http://www.openda.org/schemas/openda_1.0/openDaApplication.xsd">
  <stochObserver className="org.openda.utils.CsvStochObserver">
    <workingDirectory>./stochobserver</workingDirectory>
    <!-- <configFile>OscillatorSimpleStochObsConfig.xml</configFile> -->
    <configFile>observations_oscillator_generated_for_calibration.csv</configFile>
  </stochObserver>
  <restartOutFilePrefix>restart_</restartOutFilePrefix>
  <stochModelFactory className
    ="org.openda.models.oscillator.OscillatorStochModelFactory">
    <workingDirectory>./model</workingDirectory>
    <configFile>OscillatorStochModel.xml</configFile>
  </stochModelFactory>
  <algorithm className="org.openda.algorithms.Dud">
    <workingDirectory>./algorithm</workingDirectory>
    <configString>dudAlgorithm.xml</configString>
  </algorithm>
  <resultWriter className="org.openda.resultwriters.MatlabResultWriter">
    <workingDirectory>./</workingDirectory>
    <configFile>dud_results.m</configFile>
  </resultWriter>
</openDaApplication>
```

xml-schema

observations

Restart files

Model

Output



XML-Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<openDaApplication xmlns="http://www.openda.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.openda.o
  <stochObserver className="org.openda.utils.CsvStochObserver">
    <workingDirectory>./stochobserver</workingDirectory>
    <!-- <configFile>OscillatorSimpleStochObsConfig.xml</configFile> -->
    <configFile>observations_oscillator_generated_for_calibration.csv</configFile>
  </stochObserver>
  <restartOutFilePrefix>restart_</restartOutFilePrefix>
  <stochModelFactory className="org.openda.models.oscillator.OscillatorStochModelFactory">
    <workingDirectory>./model</workingDirectory>
    <configFile>OscillatorStochModel.xml</configFile>
  </stochModelFactory>
  <algorithm className="org.openda.algorithms.Dud">
    <workingDirectory>./algorithm</workingDirectory>
    <configString>dudAlgorithm.xml</configString>
  </algorithm>
  <resultWriter className="org.openda.resultwriters.MatlabResultWriter">
```

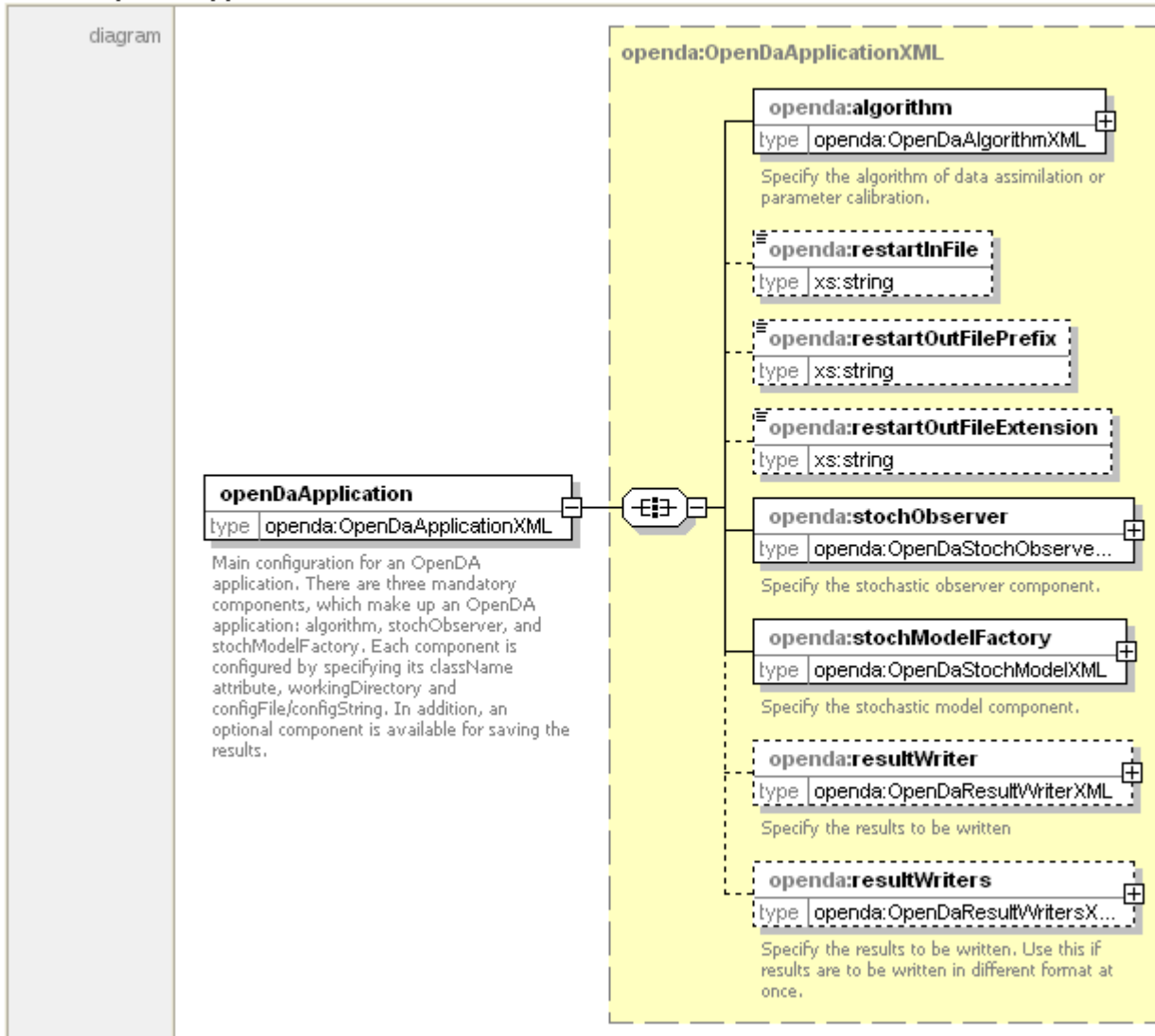
Syntax highlighting

Suggestions and checks

Shown: eclipse indigo

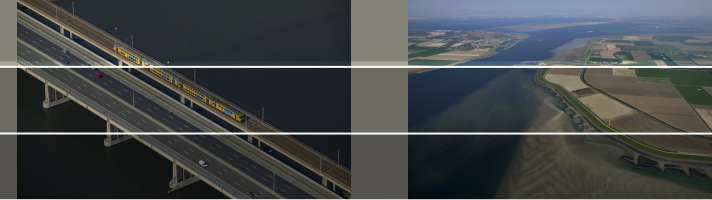
XML-Schema

element `openDaApplication`



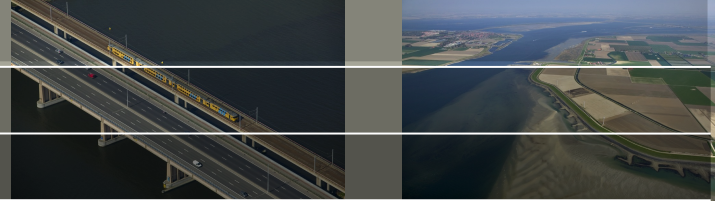
- Docu generated from xml-schemas
- On website and with local release

Observations



- **Options:**
 - **Implement interface yourself**
 - **Use existing an implementation**
 - **CSV**
 - **Noos format**
 - **Use exchange items (see blackbox explanation) eg. For FEWS PI-Timeseries**
- **Missing observations**
- **Uncertainty**

Csv format for observations

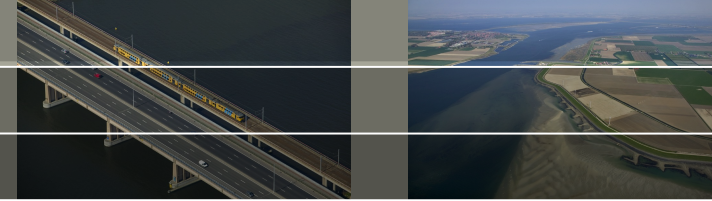


org.openda.utils.CsvStochObserver

```
time,index,value,std
0.0,0.0,0.8,0.1
1.0,0.0,0.05229757378927516,0.1
2.0,0.0,-1.004181583131941,0.1
3.0,0.0,0.10074508247431957,0.1
5.0,0.0,0.1682453171315333,0.1
6.0,0.0,0.047393455366316284,0.1
```

- Mostly for toy problems
- Columns have fixed meaning:
 - Time (in modified Julian days) or unreferenced
 - Index (position in grid)
 - Value
 - Std
- Note missing time 4.0, but no missing value symbol

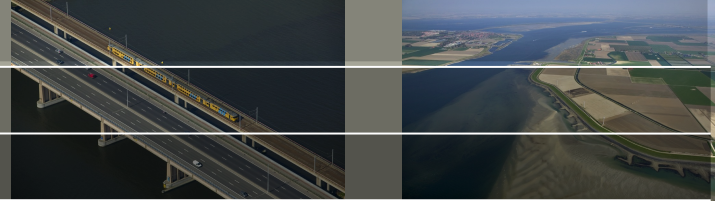
Noos format for TimeSeries



```
<?xml version="1.0" encoding="UTF-8"?>
<noosObserver>
  <timeSeries status="use" standardDeviation="0.05"
    quantity="waterlevel" location="aberdeen">
    waterlevel_aberdeen.noos
  </timeSeries>
  <timeSeries status="use" standardDeviation="0.05"
    minValue="-2.0" maxValue="2.0">
    waterlevel_wick.noos
  </timeSeries>
  <timeSeries status="use" standardDeviation="0.05"
    minDateTime="200801020000" maxDateTime="200801030000">
    waterlevel_sheerness.noos
  </timeSeries>
</noosObserver>
```

```
#=====
# Timeseries retrieved from the MATROOS series database
# Created at Mon Mar 17 10:37:38 CET 2008
#=====
# Location   : aberdeen
# Position   : (-2.045543,57.361939)
# Source     : observed
# Unit       : waterlevel_astro
# Analyse time: most recent
# Timezone   : GMT
#=====
200801010000 -0.8300
200801010010 -0.8800
200801010020 -0.9100
```

- Many observations are timeseries
- Simple to edit or generate
- Create from other exchangeItem using `oda_dumpio`
- Selection for time and values



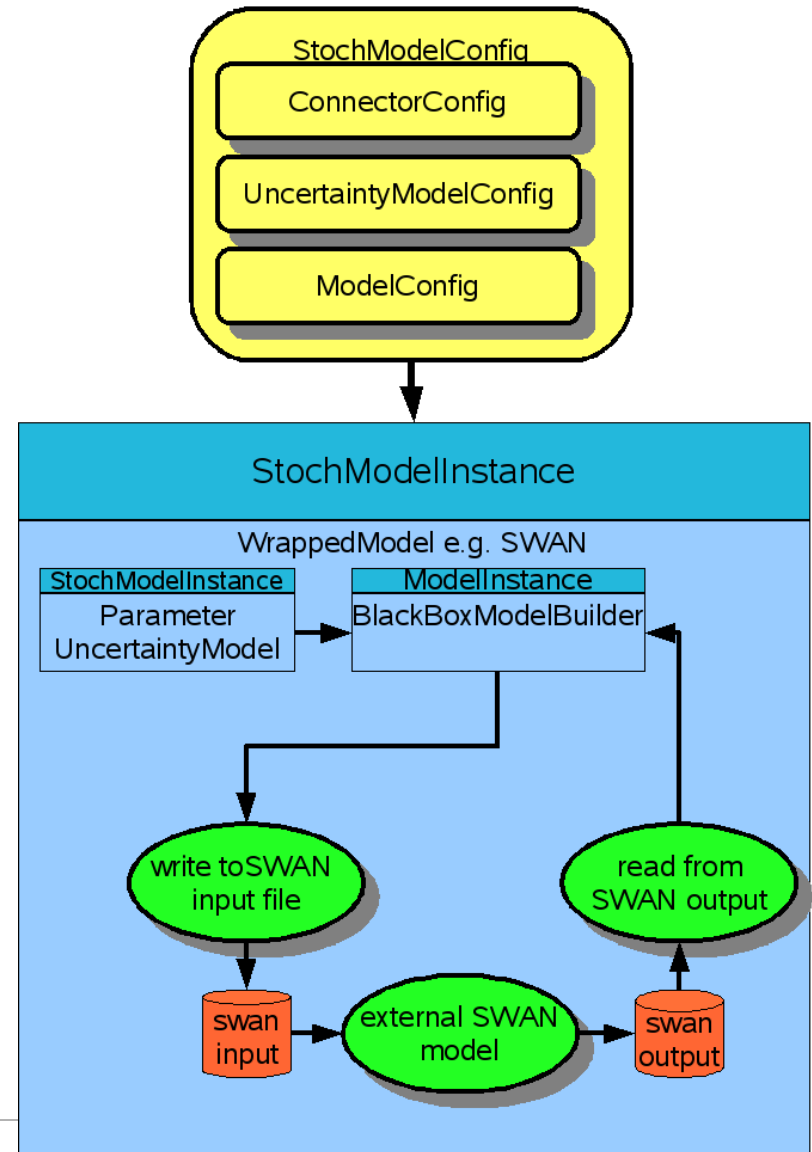
A model wrapper is needed for each model

- 1. Direct implementation of java interface**
- 2. Native toolbox**
- 3. Simple java models for testing and research**
- 4. Blackbox**

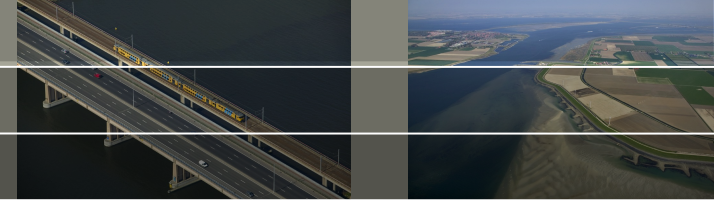
- Stochastic extension

OpenDA black-box wrapper

- Uses input and output files of the model
- No source code of the model is needed
- More info in BlackBox presentation
- `oda_info` and `oda_dumpio` to find the items and classes.



Stochastic model extension



```
<parameters>
  <regularisationConstant>
    <stdDev value=".1" transformation="identity"/>
    <subVector id="depth" sourceVectorId="depth">
      <selector className="nl.deltares.openda.models.d3dflow.D3dField2DMask">
        <arg>6,9:9,11</arg>
      </selector>
    </subVector>
  </regularisationConstant>
</parameters>

<noiseModel>
  <vector id="West Boundary"/>
  <armaModel operation="add">
    <stdDev value="0.032428488" transformation="identity"/>
    <armaConstant>0.94595947</armaConstant>
  </armaModel>
</noiseModel>
<vector id="state"/>
```

Uncertainty for
parameters

Uncertainty for
Time-varying
Model input

ResultWriters

```
// main input file <something>.oda
```

```
<resultWriters>  
  <!-- dump everything to a matlab readable ascii file -->  
  <resultWriter className="org.openda.resultwriters.MatlabResultWriter">  
    <workingDirectory>.</workingDirectory>  
    <configFile>results_dud.m</configFile>  
  </resultWriter>  
  <!-- write calibration summary table as ascii table -->  
  <resultWriter className="org.openda.resultwriters.TextTableWriter">  
    <workingDirectory>.</workingDirectory>  
    <configFile>results_dud.csv</configFile>  
  </resultWriter>  
  <!-- dump everything as netcdf -->  
  <resultWriter className="org.openda.resultwriters.NetcdfResultWriter">  
    <workingDirectory>.</workingDirectory>  
    <configFile>results_dud_.nc</configFile>  
  </resultWriter>  
</resultWriters>
```

- You can use:
 - Matlab
 - Netcdf
 - Csv
 - Your own
- Possible to select parts of data

Multiple writers

```
<resultWriter className="org.openda.resultwriters.MatlabResultWriter">  
  <workingDirectory>.</workingDirectory>  
  <configFile>results_dud_observer.m</configFile>  
  <selection>  
    <!-- only observer items-->  
    <doLog algorithm="false"  
      model="false" observer="true" nonConfiguredItems="false"  
    </doLog>  
  </selection>  
</resultWriter>
```

One writer



OpenDA : an open-source data-assimilation toolbox - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.openda.org/joomla/index.php

OpenDA : an open-source data-assi...

OpenDA

search...

MAIN MENU

- About OpenDA
 - Questions and answers
 - OpenDA applications
 - The OpenDA association
- Downloads
- Documentation
- Forum
- Support
- Getting involved
- Partners & Services

LOGIN FORM

Logging in is only necessary if you want to participate in the discussions on the forum. For all other users of this site or

Integrating models and observations

OpenDA is an open interface standard for (and free implementation of) a set of tools to quickly implement data-assimilation and calibration for arbitrary numerical models. OpenDA wants to stimulate the use of data-assimilation and calibration by lowering the implementation costs and enhancing the exchange of software among researchers and end-users.

A model that conforms to the OpenDA standard can use all the tools that are available in OpenDA. This allows experimentation with data-assimilation/calibration methods without the need for extensive programming. Reversely, developers of data-assimilation/calibration software that make their implementations compatible with the OpenDA interface will make their new methods usable for all OpenDA users (either for free or on a commercial basis).

OpenDA has been designed for high performance. Hence, even large-scale models can use it. Also, OpenDA allows users to optimize the interaction between their model and the data-assimilation/calibration methods. Hence, data-assimilation with OpenDA can be as efficient as with custom-made

Announcements

[Full release now available](#)
The full sources for OpenDA version 1.0 are now available on this OpenDA website. Click [here](#) to download the source, binaries for windows and linux, examples and more.

[OpenDA 1.0 released](#)
OpenDA version 1.0 has been officially released at May 10., 2010 during the JonsMod workshop at Deltares in the Netherlands. Information relating to the release can be found [here](#)

- Download
- Documentation
- Association

Additional wrapper module needed for each model