Advanced visualization and programming with VisNow platform

Module programming

UNIVERSITY OF WARSAW
Interdisciplinary Centre for Mathematical and Computational Modelling
www.icm.edu.pl

VisNow – modules

- **Java**
  - VisNow implementation is written in Java programming language
  - Currently approx. 300,000 code lines (1,500 classes)
  - Creation of VisNow modules – Java SE programming
  - Recommended Java source version <=1.6

- **Modules – basics**
  - Module = Java package
  - Module schema description
  - Main module class

- **Package naming policy**
  - Following recommended Java package naming policy
  - **com.my.company.project**
  - E.g. pl.edu.icm.visnow, **sg.edu.a-star.acrc.visnow**
  - Modules are usually in the subpackage **lib**, e.g.: **sg.edu.a-star.acrc.visnow.lib**

VisNow – modules

- **Module schema definition – module.xml**
  - XML file module.xml inside the package

- **Defines**
  - Main class (required)
  - Name (required)
  - Short description
  - Input ports
  - Output ports

Module name visible in the library tree and in the workspace

Short description is used as a help tooltip in the library tree.

- **File structure**

```
<module name="my module name" class="ModuleMainClass">
    <inputs>
        Input definitions goes here
    </inputs>
    <outputs>
        Output definitions goes here
    </outputs>
    <description value="my module short description"/>
</module>
```

VisNow – modules

- **Input port definition – <input>**
  - One entry for each port

- **Defines**
  - Name (required)
  - Data type (required)
    - pl.edu.icm.visnow.lib.types.VNField
    - pl.edu.icm.visnow.lib.types.VNRegularField
    - pl.edu.icm.visnow.lib.types.VNIrregularField
    - pl.edu.icm.visnow.lib.types.VNGeometryObject
    - java.lang.Float, Double, Byte, Short, Integer, Long, Boolean, Object
  - Modifiers (required)
    - NECESSARY – required port
    - TRIGGERING – triggering port
  - Short description (optional)

Port name visible on the module in workspace and used in module code for accessing data.

```
<input name="input_name" type="data_type_class" modifiers="port_modifiers">
        Acceptor definitions and description goes here
</input>
```

VisNow – modules

- **Output port definition – <output>**
  - One entry for each port

- **Defines**
  - Name (required)
  - Data type (required)
    - pl.edu.icm.visnow.lib.types.VNField
    - pl.edu.icm.visnow.lib.types.VNRegularField
    - pl.edu.icm.visnow.lib.types.VNIrregularField
    - pl.edu.icm.visnow.lib.types.VNGeometryObject
    - java.lang.Float, Double, Byte, Short, Integer, Long, Boolean, Object
  - Short description (optional)

Port name visible on the module in the workspace and used in the code for data access.

```
<output name="input_name" type="data_type_class">
        Schema definitions and description goes here
</output>
```

VisNow – modules

- **Example module.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<module name="isosurface" class="Isosurface">
  <inputs>
    <input name="inField" type="pl.edu.icm.visnow.lib.types.VNField"
                                                          modifiers="NECESSARY:TRIGGERING">
      <description value="Input for data field to extract isosurface"/>
      <acceptor>
          <param name="NSPACE" value="3"/>
          <param name="REGULAR" value="true"/>
          <param name="NDIMS" value="3"/>
          <param name="DATA_VECLEN" value="1"/>
      </acceptor>
      <acceptor>
          <param name="NSPACE" value="3"/>
          <param name="IRREGULAR" value="true"/>
          <param name="DATA_VECLEN" value="1"/>
          <param name="CELLS_3D" value="true"/>
      </acceptor>
    </input>
```

VisNow – modules

- **Example module.xml**

```
    <input name="threshold" type="java.lang.Float" modifiers="TRIGGERING">
      <description value="Input for isolevel value"/>
    </input>
  </inputs>
  <outputs>
    <output name="isosurfaceField" type="pl.edu.icm.visnow.lib.types.VNIrregularField">
      <description value="Output for surface field"/>
      <schema>
        <param name="NSPACE" value="3"/>
        <param name="IRREGULAR" value="true"/>
        <param name="CELLS_TRIANGLE" value="true"/>
      </schema>
    </output>
    <geometryOutput/>
  </outputs>
  <description value="Maps volumetric data creating surface of constant data values."/>
</module>
```

VisNow – modules

- **Main module class**
  - The instance of this class is created when a new module is created in the workspace
  - Generally it extedns **pl.edu.icm.visnow.engine.core.ModuleCore** class
    - Data flow engine communication
    - Port service
    - Events service
    - In practice – too low level for entry point
- **Visualization module**
  - Module that provides most of the required „administration”
  - Provides a default visual presentation of the output data
  - Extends **pl.edu.icm.visnow. … .OutFieldVisualizationModule** class
- **Required elements of the main class**
  - Overriden method **public void onActive() {}**
  - Variable **public static InputEgg[] inputEggs = null;**
  - Variable **public static OutputEgg[] inputEggs = null;**

Main computational method of the module. Executed on module running.

VisNow – modules

- **Module API**
- **Methods for overriding** (in ModuleCore or OutFieldVisualizationModule)
  - **public void onActive()**
    - Executed as module action when the module starts (in own module's thread)
  - **public void onInputAttach(LinkFace link)**
    - Executed on inpurt port attach
  - **public void onInputDetach(LinkFace link)**
    - Executed on input port detach
  - **public void onOutputAttach(LinkFace link)**
    - Executed on outpurt port attach
  - **public void onOutputDetach(LinkFace link)**
    - Executed on outpurt port detach
  - **public void onDelete()**
    - Executed on module deletion
  - **public boolean isGenerator()**
    - Should return true if the module is the network entry point (e.g. data reader or data generator) and usually has no required input ports.
  - **public boolean isViewer()**
    - Should return true if the module is the exit point of the network (e.g. viewer) and does not have output ports.

VisNow – modules

- **Module API**
- **Methods to call in the module**
  - **String getName()**
    - Returns module name.
  - **void startAction()**
    - Non-blocking method used to start the module. It notices the network engine that this module needs execution. As a consequence the **onActive()** method will be executed and the modules in the network below.
      WARNING!! Direct usage of **onActive()** method does not cause starting of the module! The network will not start and the modules in the network below will not start. Beside the special situations this method call should be avoided!
  - **Vector<Object> getInputValues(String name)**
    - Returns a vector of all data from the input port. Parameter **"name"** is required and is a port name that we wish to read the data from. The name has to be the same as declared in module.xml. The type of the returned Objects will be the same as port type, so it can be cast.
  - **Object getInputFirstValue(String name)**
    - Returns first data from the input port. Parameter **"name"** is required and is a port name that we wish to read the data from. The name has to be the same as declared in module.xml. The type of the returned Object will be the same as port type, so it can be cast.
  - **boolean setOutputValue(String name, Object value)**
    - Sets the data to the output port. Parameter **"name"** is required and is a port name that we wish towrite to. The name has to be the same as declared in module.xml. Parameter **"value"** is the object that we want to set to the port – it has to be of the same type as the port type.

VisNow – modules

- **Module API**
- **Methods to call in the module**
  - **boolean setProgress(float progress)**
    - Sets the progres of module execution represented by the progres bar on the module box in the workspace. The value set should be from 0.0f-1.0f range and reflect the calculation progress. ***WARNING!!*** This method should not be called very often (e.g. in each loop iteration) as it strongly decreases the performance.
  - **float getProgress()**
    - Returns the current module execution progres if it was set with setProgress() method. The returned value is from 0.0f-1.0f range and it represents the calculation progress.
  - **void setPanel(Component component)**
    - Sets the Swing/AWT component that represents the module user interface (GUI) that is visible in the module GUI panel in the main VisNow window and is responsible for module parameters steering. ***WARNING!!*** This method should not be used directly in the **OutFieldVisualizationModule**. In that case the **setPanel(ui)** method should be used in the constructor to set the default presentation GUI (the **ui** variable). Module GUI panel can be then set as computation tab with the use of **ui.addComputeGUI(JPanel gui)** method.

VisNow – modules

- **Module API – only in OutFieldVisualizationModule**
- **Variables to use**
  - **Field outField;**
    - This variable should be used as a default output field of the module. It causes that the automated visualization uses this field, accordingly to presentation settings, on the default output geometry port, after execution of **prepareOutputGeometry()** and **show()**.
  - **RegularField outRegularField;**
    - This is the convenience variable for regular field usage. Usually it is used to cast the outField variable.
  - **IrregularField outIrregularField;**
    - This is the convenience variable for irregular field usage. Usually it is used to cast the outField variable.
- **Methods to call in the module**
  - **void prepareOutputGeometry()**
    - This method should be executed after creating the output field and setting the outField variable. It creates the default geometric representation of the output field. Usually it is executed at the very end of **onActive(**. method. Almost always it is directly followed by executing **show()** method.
  - **void show()**
    - This method should be called right afer executing **prepareOutputGeometry()**. It adds the created geometry to the output geometry object within the default geometry output port (<geometryOutput/>). Usually it is executed at the very end of **onActive()** method and almost always right after **prepareOutputGeometry ()**.

VisNow – modules

- **The next step**
  - **Basic module has only module.xml and module class**
  - **Very unusual in practice**
- **Full module structure**
  - Module schema definition – module.xml
  - Module main class
  - Parameters class
  - Graphical user interface class (GUI)
  - Computational core class
    - It is often advisable to move the main calculation out of the main module class for code clarity and reusability (e.g. as a static utility).
  - Additional classes and resources
    - All classes that are used only within the given module should be kept inside module package or subpackages.

VisNow – modules

- **Parameters**
  - **The class that extends pl.edu.icm.visnow.engine.core.Parameters**
  - **It holds internal module parameters that are usually set with GUI and used in calculations (e.g. selected component, threshold value, data dimension).**
  - **There already is a predefined „parameters” variable that should be set in the module constructor code.**
  - **Usually called Params**
- **Definition**
  - Static array:

```
private static ParameterEgg[] eggs = new ParameterEgg[]
{
        Parameter definitions goes here
};
```

VisNow – modules

- **Parameters**

  - **Each parameter is defined by a single element of that array**

```
new
ParameterEgg<param_class>("param_name",param_type,
param_value)
```

  - *param_class* – parameter object class (e.g. Float, Integer, int[])
  - *param_name* – the name of the parameter used in all set and get methods
  - *param_type* – one of three values: ParameterType.independent (parametr does not depend on input data) ParameterType.dependent (parameter depends on input data, e.g. a selected component) or ParameterType.filename (a path to some local file)
  - *param_value* – the default parameter value, a *param_value* class object.
    WARNING!! It is forbidden to set parameters other than simple variables in the static constructor due to the object reference sharing between module instances of the same class. A **null** should be set then and the exact value set in Params class constructor.

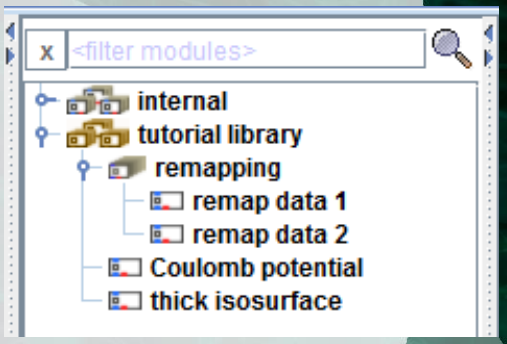
VisNow – modules

- **Parameters**
  - **Methods for parameters manipulation:**
    - **void setValue(String name, Object value)** – to set the value of “name” parameter.
    - **Object getValue(String name)** – to get the current parameter value.
  - **Best practice – write some explicit setters and getters for each parameter**
- **Parameter events**
  - **ParameterChange event**
    - Triggered every time setValue method is executed
    - Uses the **pl.edu.icm.visnow.engine.core.ParameterChangeListener** interface to react on the event with the **parameterChanged(String name)** method.
    - Listeners can be registered with **addParameterChangelistener(ParameterChangeListener listener)** method
    - Listeners can be removed with **removeParameterChangelistener(ParameterChangeListener listener)** method
  - **StateChanged event**
    - Triggered when **fireStateChanged()** method is executed
    - Uses the **javax.swing.event.ChangeListener** interface to react on the event with the **stateChanged(ChangeEvent e)** method.
    - Listeners can be registered with **addChangeListener(ChangeListener listener)** method
    - Listeners can be removed with **removeChangeListener(ChangeListener listener)** method

VisNow – modules

- **GUI**
  - **Swing/AWT component**
  - **Usually a class named GUI**
  - **Usually it extends the javax.swing.JPanel**
  - **Module graphical interface – controls for module parameters steering**
  - **Should be correctly set in the constructor**
  - **It is visible in the main panel GUI area in the main VisNow window**
  - **Recommendation – use the predefined VisNow widgets and panels**
    - **pl.edu.icm.visnow.gui**
    - **pl.edu.icm.visnow.gui.widgets**
    - **pl.edu.icm.visnow.lib.gui**
  - ***WARNING!!!***
    - Synchronization, avoiding of event loops
    - Blocking flags or the *active* flag in parameters

VisNow – modules

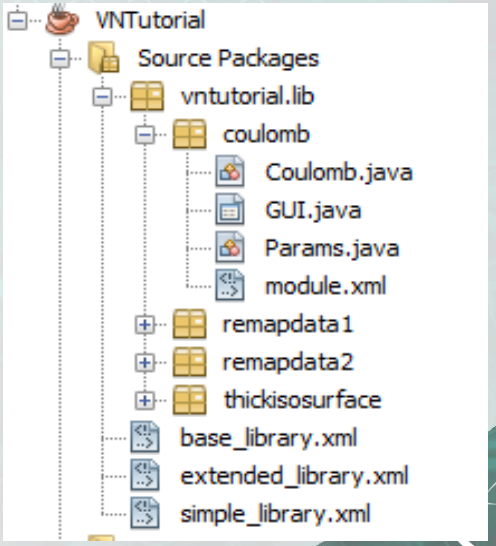


- **Plugins**
  - **External module libraries**
  - **We do not modify VisNow core code**
  - **Read into the application in run-time**
  - **Plugin = JAR file**
    - **VisNow modules in packages**
    - **Library files**
- **Library files**
  - XML files with the library tree structure definition (it is visible in the library panel in the main VisNow window)
  - In the main folder of JAR file
  - Depend on VisNow version
    - **simple_library.xml** – for VisNow Simple
    - **extended_library.xml** – for VisNow Pro

```
<library name="my external library">
    <core package="your.institution.visnow.lib.module1"/>
    <core package="your.institution.visnow.lib.module2"/>
    <folder name="my folder">
        <core package="your.institution.visnow.lib.module3"/>
    </folder>
</library>
```

VisNow – modules

- **Tags in library.xml**
  - **<library>** - root tag defining a library:
    - **name** – parameter being the library name, required
  - **<folder>** - module grouping tag, it groups modules into folders/subfolders/branches of the library tree
    - **name** – parameter, the folder name, required
    - **open** – parameter, sets if the branch is open by default (value set to „yes” or „no”), optional
    - **autosort** – parameter, sets if the folder contents should be alphabetically sorted („yes” or „no” value), optional
  - **<core>** - module definition tag
    - **package** – parameter, module package name, required
- **Typical plugin structure**



visnow.icm.edu.pl

Contact:

visnow@icm.edu.pl

Interdisciplinary Centre for Mathematical and Computational Modelling

University of Warsaw

Bartosz Borucki, Krzysztof Nowiński

UNIVERSITY OF WARSAW
Interdisciplinary Centre for Mathematical and Computational Modelling
www.icm.edu.pl