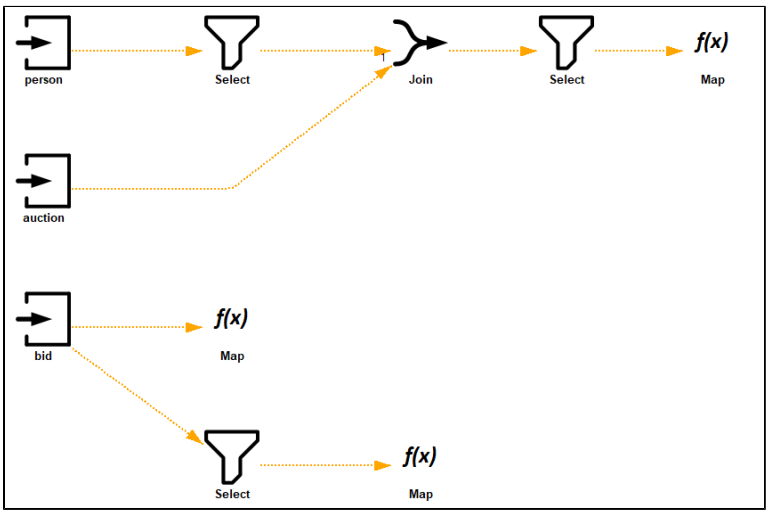
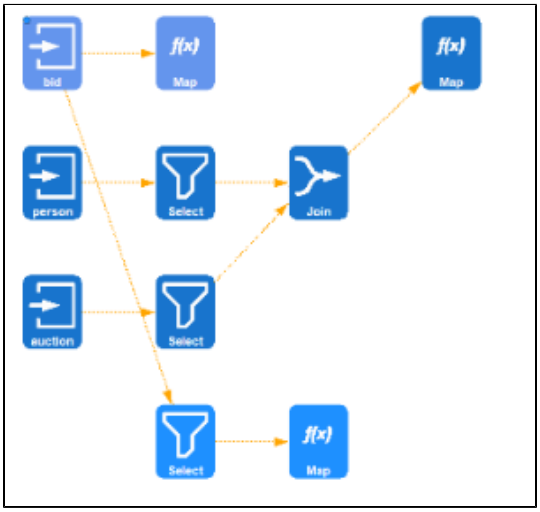
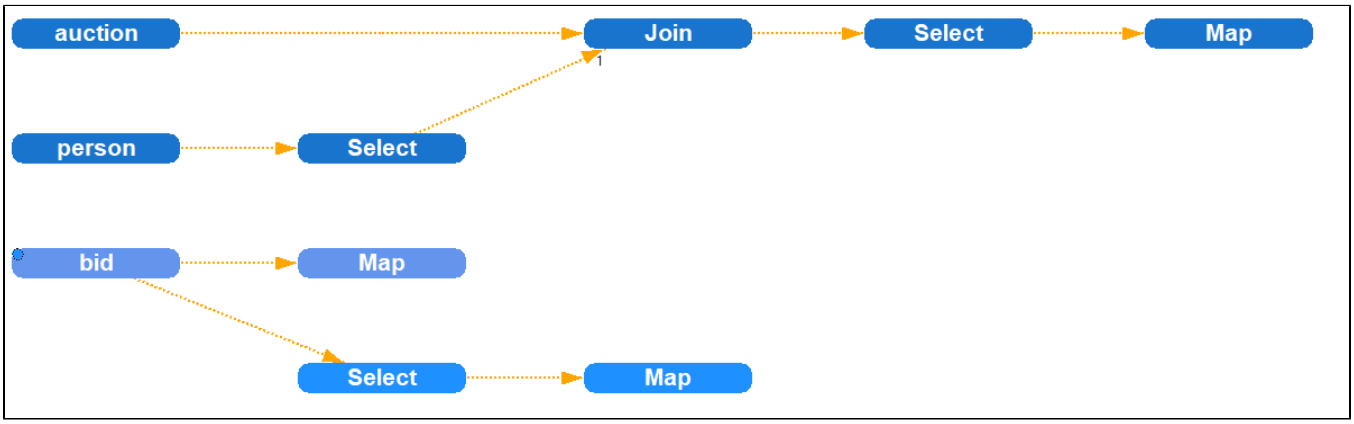
Customizing Operator Graph Visualization (Odysseus Studio)

Background

The visualization of the opertor graph in Odysseus Studio can be customized in order to change its style and appearance. The following pictures show some examples of different visualization styles provided by Odysseus.







The visualization framework provides several ways to control the appearance of the visualization which are described in the following sections.

Viewer configuration files

The way the nodes are drawn is defined in a special configuration file, the viewer configuration. It is an XML file that describes how to compose a node from different symbol elements. The schema of the file is defined in the SymbolSchema.xsd file. The following code snippet shows an example configuration file:

Viewer configuration file: example

```
<Symbols xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="symbolSchema.xsd">

        <Symbol nodeName="ReceiverPO">
        <Size width="150" height="27"/>
        <SymbolElement type="ownerRectangle"/>
        <SymbolElement type="ownerText"/>
    </Symbol>

    <Default>
        <Size width="80" height="100"/>
        <SymbolElement type="ownerRectangle"/>
        <SymbolElement type="operator">
            <Parameter key="iconSetName" value="white"/>
            <Parameter key="marginLeft" value="10"/>
            <Parameter key="marginRight" value="10"/>
            <Parameter key="marginTop" value="10"/>
            <Parameter key="marginBottom" value="26"/>
        </SymbolElement>
        <SymbolElement type="ownerText">
            <Parameter key="offsetY" value="80"/>
        </SymbolElement>
    </Default>

</Symbols>
```

The `Symbols` element contains a list of definitions for node layouts. You can specifiy the layout individually for each type of physical operator by setting the `nodeName` attribute of the `Symbol` element to the respective class name. For each `Symbol`, you must specify its size and one ore more `SymbolElement` elements which describe the visual components to compose the node visualization. Some of these `SymbolElement` elements can additionally be configured using parameters that are specified as simple key value pairs. The following table gives you an overview of the available types of `SymbolElement` and the parameters they support.

Type	Description	Supported parameters
fillCircle	Circle shape with filling	• r • g • b
circle	Circle shape without filling	• r • g • b
fillRectangle	Rectangle shape with filling	• r • g • b
rectangle	Rectangle shaoe without filling	• r • g • b
selector		

image	Image	• resource • marginLeft • marginRight • marginTop • marginBottom
operator	Shows a specific image for the physical operator from the given icon set	• resource • iconSetName • marginLeft • marginRight • marginTop • marginBottom
invisible	Invisible (for layouting only)	
selectivity		
text	Shows the name of the operator	• offsetX • offsetY
ownerText	Shows the name of the operator	• r • g • b • offsetX • offsetY
ownerRectangle	draws a filled rectangle. The filling color is automatically determined to indicate the script/query the oeprator was created from	

The following table gives an overview of the parameters:

Parameter name	Possible values	Description
r, g, b	Integer (0...255)	RGB Color parameters. Important: Each color component must be individually specified as a single `Parameter` element!
marginLeft	Integer (0...100)	left margin (relative in percent!)
marginRight	Integer (0...100)	right margin (relative in percent!)
marginTop	Integer (0...100)	top margin (relative in percent!)
marginBottom	Integer (0...100)	bottom margin (relative in percent!)
offsetX	Integer (0...100)	Relative offset of X position (starting from left) in percent
offsetY	Integer (0...100)	Relative offset of Y position (starting from top) in percent
resource	String	ImageID of the image to show. The image ID must be registered in the ImageManager.
iconSetName	String	Name of the image set that should be used to automatically retrieve a matching image for the underlying physical operator. By default Odysseus provides the following image sets: `white`, `black`, and `default`. It is also possible to use external image sets stored in the Odysseus home directoy.

Besides the definitions for individual physical operators, the viewer configuration must specify the `Default` element which is applied to all operators that are not individually styled by a `Symbol` element. The structure of the `Default` element is similiar to the structure of the `Symbol` element.

How to use your own viewer configuration

In order to use your own viewer configuration, you need to perform the following steps:

1. Create an viewer configuration file as describhed above and store it in the Odysseus home directory.
2. Open the `odysseusRCP.conf` file and set the path of your viewer configuration file for the `viewer.config` key. Note: The path must be relative to the Odysseus home directory.
3. Changes will apply after a restart of Odysseus Studio.