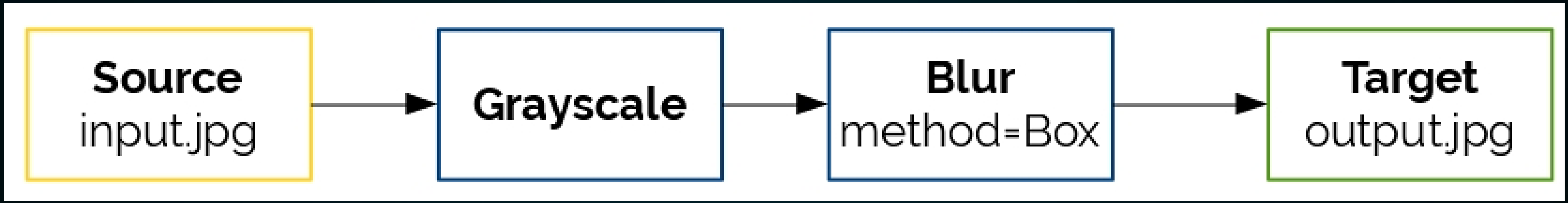
Implementation of a DSL in Papyrus

Dominique Heer | UAntwerpen | Model Driven Engineering | 03.02.2017

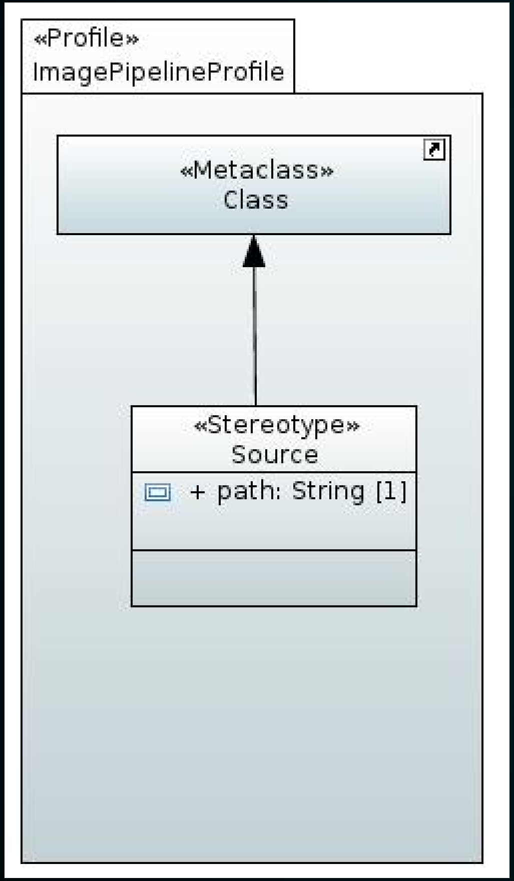
Image Processing Pipeline

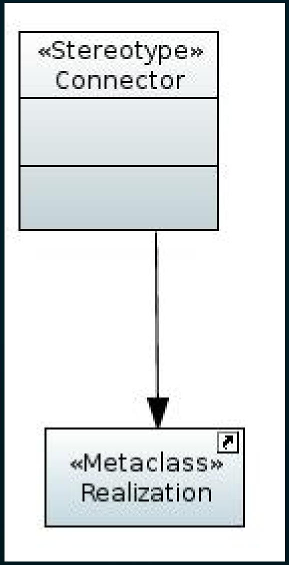


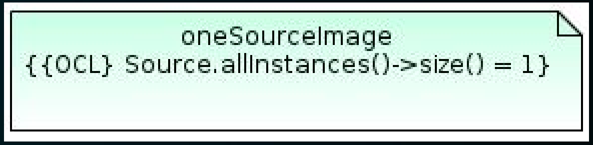
Language Design

- exactly one input and one output image
  - input and output images are either JPEG or PNG
- three processing blocks:
  - grayscale
  - blur
  - resize

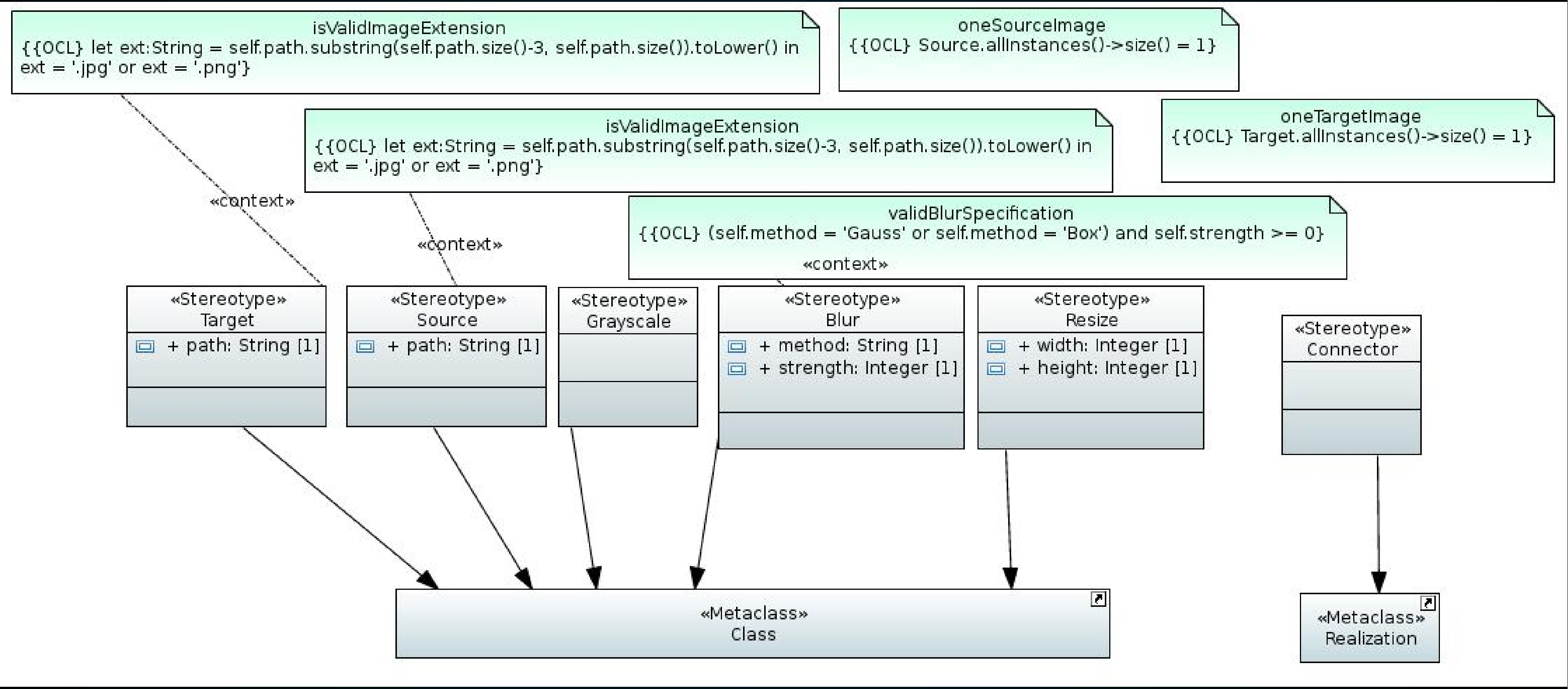
Implementation



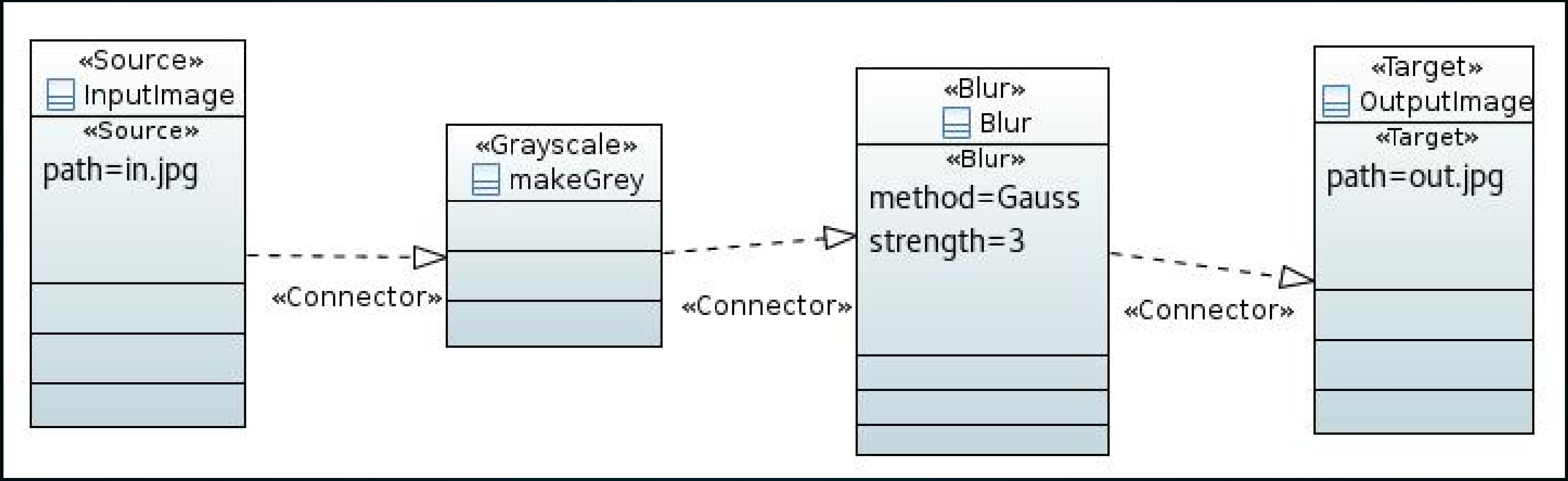




Implementation


isValidImageExtension
{{OCL} let ext:String = self.path.substring(self.path.size()-3, self.path.size()).toLower() in ext = '.jpg' or ext = '.png'}
oneSourceImage
{{OCL} Source.allInstances()->size() = 1}
isValidImageExtension
{{OCL} let ext:String = self.path.substring(self.path.size()-3, self.path.size()).toLower() in ext = '.jpg' or ext = '.png'}
oneTargetImage
{{OCL} Target.allInstances()->size() = 1}
validBlurSpecification
{{OCL} (self.method = 'Gauss' or self.method = 'Box') and self.strength >= 0}
«context»
«context»
«context»
«Stereotype» Target
+ path: String [1]
«Stereotype» Source
+ path: String [1]
«Stereotype» Grayscale
«Stereotype» Blur
+ method: String [1]
+ strength: Integer [1]
«Stereotype» Resize
+ width: Integer [1]
+ height: Integer [1]
«Stereotype» Connector
«Metaclass» Class
«Metaclass» Realization

Usage


«Source»
InputImage
«Source»
path=in.jpg
«Connector»
«Grayscale»
makeGrey
«Connector»
«Blur»
Blur
«Blur»
method=Gauss
strength=3
«Connector»
«Target»
OutputImage
«Target»
path=out.jpg

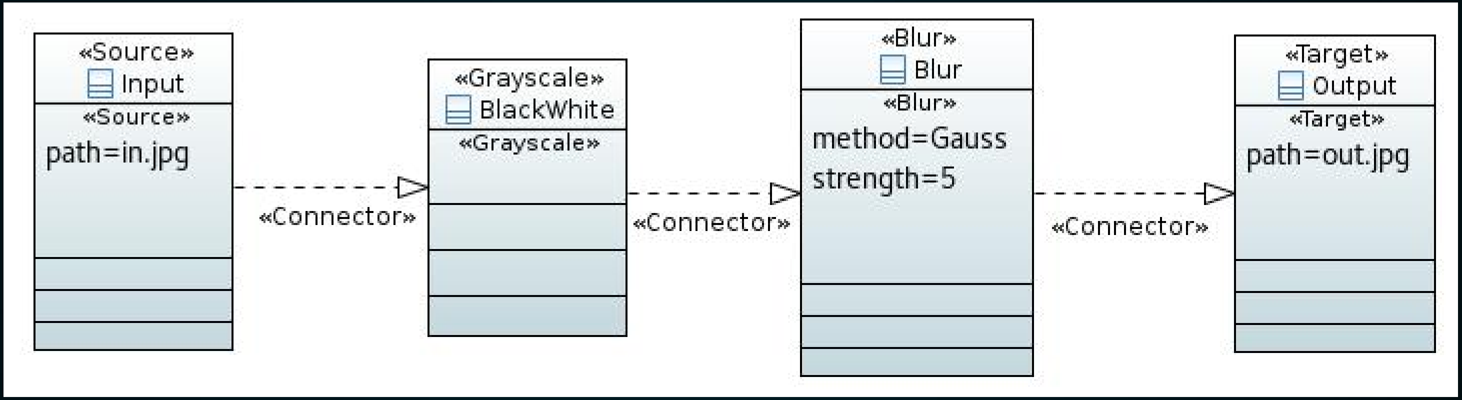
Code Generation

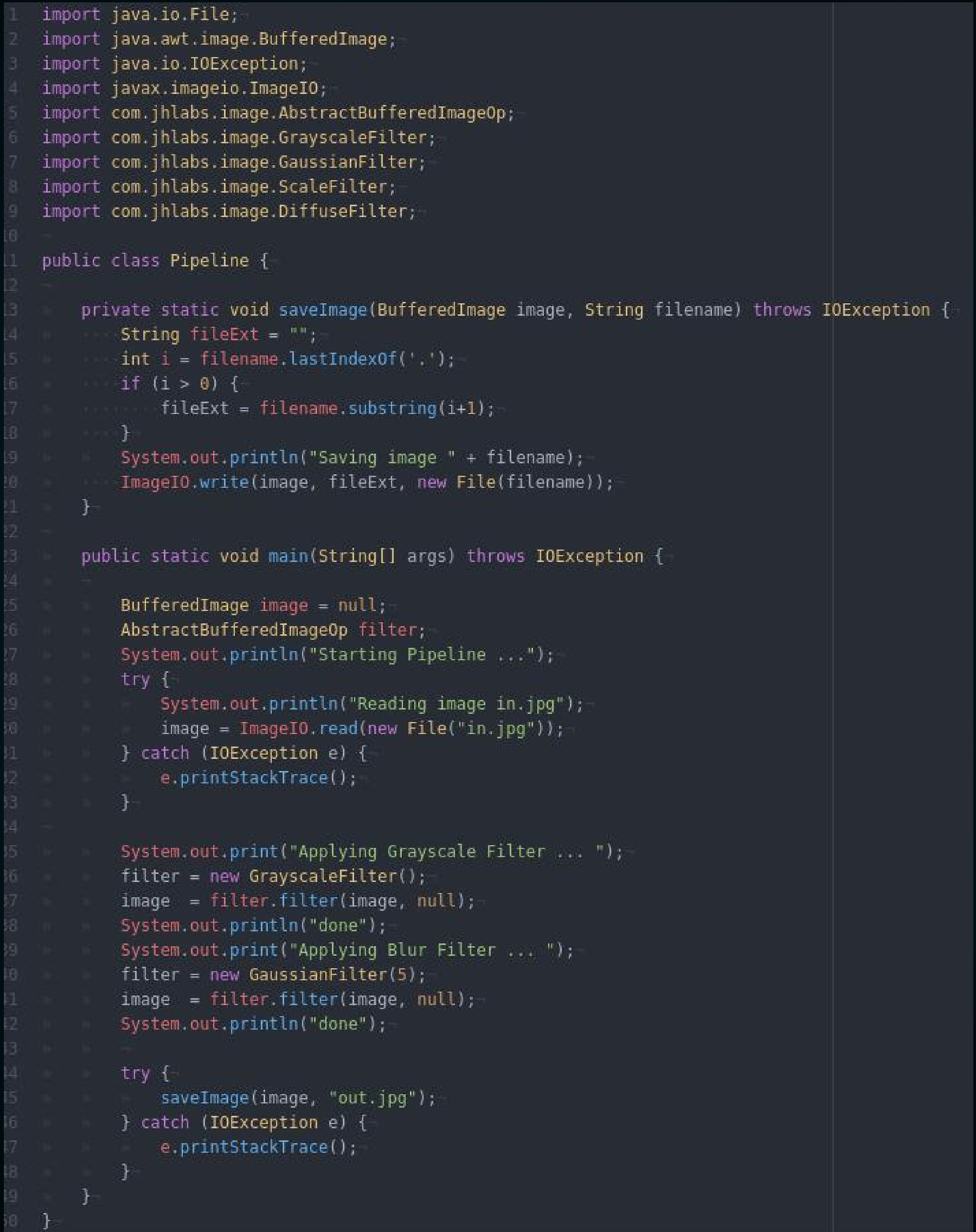
Acceleo

```
[comment encoding = UTF-8 /]
[module generate('http://www.eclipse.org/uml2/5.0.0/UML')]

[template public generateElement(model : Model)]
[comment @main/]

[file ('Pipeline.java', false)]
import java.io.File;
import java.awt.image.BufferedImage;
import java.io.IOException;
import javax.imageio.ImageIO;
import com.jhlabs.image.*;

public class Pipeline {

     private static void saveImage(BufferedImage image, String filename)
          throws IOException {
          // save image 'image' as 'filename'
     }

     public static void main(String[ '[' /][ ']' /] args) throws IOException {

          BufferedImage image = null;
          AbstractBufferedImageOp filter;

          System.out.println("Starting Pipeline ...");

          [for (clazz: Class | model.eAllContents(Class))]
               [for (stereotype : Stereotype | clazz.getAppliedStereotypes())]
                    [if (stereotype.name = 'Source')]
                         [clazz.sourceImage()/]
                    [/if]
               [/for]
          [/for]
```

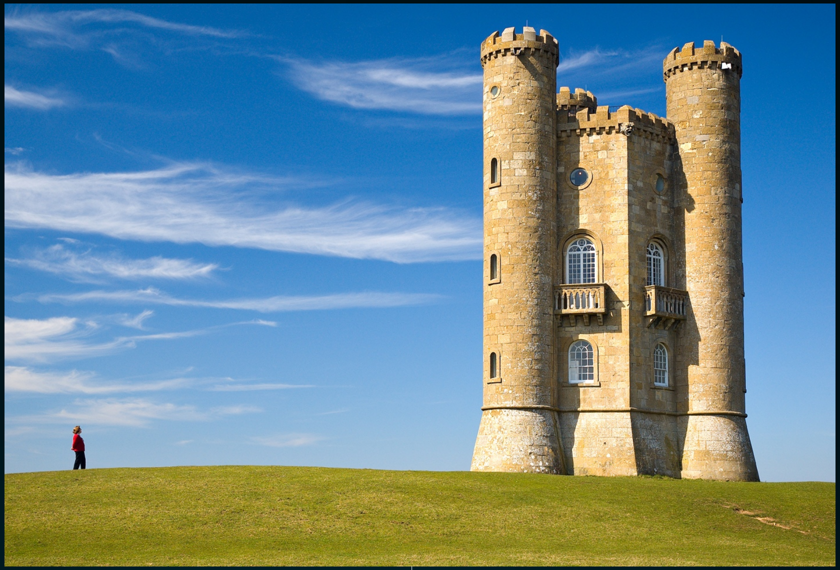
```
[template public sourceImage(clazz : Class)]
  try {
    System.out.println(
      "Reading image [clazz.getValue(clazz.getAppliedStereotype('Source'), 'path')/]");
    image = ImageIO.read(
      new File("[clazz.getValue(clazz.getAppliedStereotype('Source'), 'path')/]"));
  }
  catch (IOException e) {
    e.printStackTrace();
  }
[/template]
```

```
[for (connector: Realization | model.eAllContents(Realization))]
  [for (el : NamedElement | connector.client)]
    [for (clazz : Class | model.eAllContents(Class))]
      [if (el.name = clazz.name)]
       [for (stereotype : Stereotype | clazz.getAppliedStereotypes())]
         [if (stereotype.name = 'Grayscale')]
           System.out.print("Applying Grayscale Filter ... ");
           filter = new GrayscaleFilter();
           image  = filter.filter(image, null);
           System.out.println("done");
         [elseif (stereotype.name = 'Blur')]
           System.out.print("Applying Blur Filter ... ");
          [if clazz.getValue(clazz.getAppliedStereotype('Blur'), 'method') = 'Gauss']
            filter = new GaussianFilter(
               [clazz.getValue(clazz.getAppliedStereotype('Blur'), 'strength')/]);
          [elseif clazz.getValue(clazz.getAppliedStereotype('Blur'), 'method') = 'Box']
            filter = new BoxFilter(
               [clazz.getValue(clazz.getAppliedStereotype('Blur'), 'strength')/]);
          [/if]
            image  = filter.filter(image, null);
            System.out.println("done");
         [elseif (stereotype.name = 'Resize')]
           System.out.print("Applying Resize Filter ... ");
           filter = new ScaleFilter(
               [clazz.getValue(clazz.getAppliedStereotype('Resize'), 'width')/],
               [clazz.getValue(clazz.getAppliedStereotype('Resize'), 'height')/]);
           image  = filter.filter(image, null);
           System.out.println("done");
         [/if]
       [/for]
     [/if]
    [/for]
  [/for]
[/for]
```

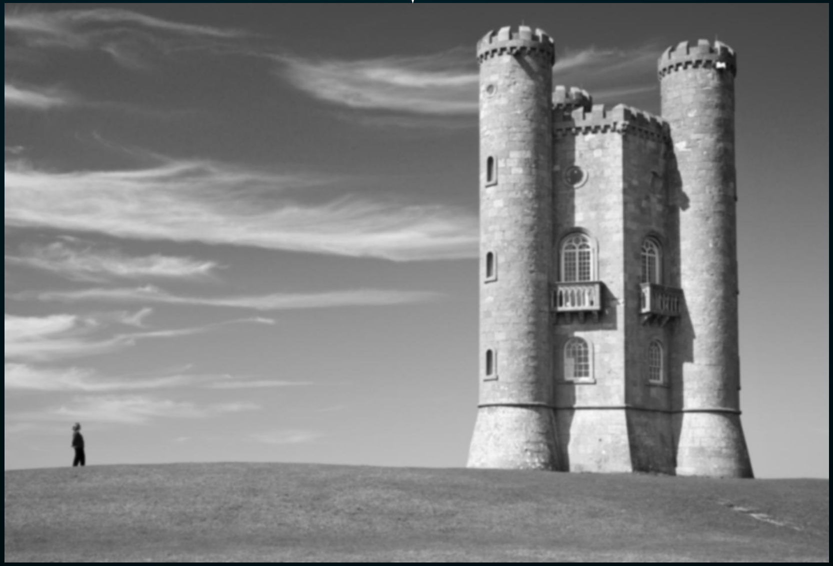

«Source»
Input
«Source»
path=in.jpg
«Connector»
«Grayscale»
BlackWhite
«Grayscale»
«Connector»
«Blur»
Blur
«Blur»
method=Gauss
strength=5
«Connector»
«Target»
Output
«Target»
path=out.jpg



```java
import java.io.File;
import java.awt.image.BufferedImage;
import java.io.IOException;
import javax.imageio.ImageIO;
import com.jhlabs.image.AbstractBufferedImageOp;
import com.jhlabs.image.GrayscaleFilter;
import com.jhlabs.image.GaussianFilter;
import com.jhlabs.image.ScaleFilter;
import com.jhlabs.image.DiffuseFilter;

public class Pipeline {

    private static void saveImage(BufferedImage image, String filename) throws IOException {
        String fileExt = "";
        int i = filename.lastIndexOf('.');
        if (i > 0) {
            fileExt = filename.substring(i+1);
        }
        System.out.println("Saving image " + filename);
        ImageIO.write(image, fileExt, new File(filename));
    }

    public static void main(String[] args) throws IOException {

        BufferedImage image = null;
        AbstractBufferedImageOp filter;
        System.out.println("Starting Pipeline ...");
        try {
            System.out.println("Reading image in.jpg");
            image = ImageIO.read(new File("in.jpg"));
        } catch (IOException e) {
            e.printStackTrace();
        }

        System.out.print("Applying Grayscale Filter ... ");
        filter = new GrayscaleFilter();
        image  = filter.filter(image, null);
        System.out.println("done");
        System.out.print("Applying Blur Filter ... ");
        filter = new GaussianFilter(5);
        image  = filter.filter(image, null);
        System.out.println("done");

        try {
            saveImage(image, "out.jpg");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



```
[nordraak@laptop src-gen]$ java -cp .:Filters.jar:imgscalr-lib-4.2.jar Pipeline
Starting Pipeline ...
Reading image in.jpg
Applying Grayscale Filter ... done
Applying Blur Filter ... done
Saving image out.jpg
```



Papyrus

vs

MPS

Papyrus

VS

MPS

Tool complexity

- Eclipse-based
- Steep learning curve
- Insufficient documentation
- Active community

Papyrus

VS

MPS

Tool scope

- UML2 Modeling
- Exploit UML's extension mechanism to implement custom DSL
- Extensible by using plugins

Papyrus

VS

MPS

Usability of the DSL

- Cumbersome due to missing plugin
- Stereotypes have to be applied manually

Papyrus

VS

MPS

Extensbility of the DSL

- Add new stereotype
- Edit code generator template
- Element is ready to use