Call Java methods from XSLT (Manakin)

Sometimes you may want to do a complicated operation in your theme. While it may be possible to create a XSLT template that achieves the desired result, it may be too much code or it may be too slow - XSLT is best used as a templating language, not a programming language. Especially since DSpace currently only supports XSLT 1.0, it lacks many useful functions which are part of XSLT 2.0 and may be too cumbersome to implement in form of templates.

Luckily, the extensibility of XSLT allows us to call functions implemented in other programming languages. In DSpace XMLUI, you call call simple Java functions directly from XSLT.

Example 1: URL as URL parameter

Let's say we want to use a URL shortening service. This hypothetical service ("`http://example.com`") requires us to pass the URL we want to be shortened to the service as a parameter ("`?url=`"): `http://example.com?url=http://url-we-want-to-shorten`. But not all characters that can occur in url-we-want-to-shorten are allowed in a URL parameter value (e.g. the "?", "&" and "=" characters). Thus, we need to escape such characters, which is commonly done by url encoding. Implementing the urlencode function in XSLT would be certainly possible, but quite lengthy. Instead, we can just use a Java method from the java.net.URLEncoder class that already implements this.

```
<xsl:stylesheet
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
        ...
        xmlns:confman="org.dspace.core.ConfigurationManager"
        xmlns:urlenc="java.net.URLEncoder"
        exclude-result-prefixes="xsl ... urlenc">

...

<xsl:template match="dim:dim" name="url-in-url-parameter-example">
    <a href="{concat('http://example.com?url=', urlenc:encode(dim:field[@element='relation' and
@qualifier='uri'], 'UTF-8'))}">example link</a>
</xsl:template>
```

Example 2: accessing DSpace configuration parameters

This way, we can use any class that your servlet container has in its classpath. That includes DSpace classes, so we can call any simple DSpace method that returns simple result type and runs in a reasonable time.

In this use case, we'll explore rendering your theme based on configuration parameters in the DSpace configuration files. To access them, we'll use the ConfigurationManager class. We'll print the configuration parameter located in this file: https://github.com/DSpace/DSpace/blob/dspace-5_x/dspace/config/modules/discovery.cfg#L8

```
<xsl:stylesheet
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
        ...
        xmlns:confman="org.dspace.core.ConfigurationManager"
        exclude-result-prefixes="xsl ... confman">

...

<xsl:template name="configuration-value-example">
    <xsl:value-of select="confman:getProperty('discovery', 'search.server')"/>
</xsl:template>
```