AddOnMechanism 2fxmlui

XML UI

The XML UI will be using the sub sitemap feature of cocoon to accomplish a plugin architecture. However this approach could be greatly increased if there was a build time process to ease the administrators burden in configuring these files. I would like to have an XML configuration file that was just as simple as say include this plugin, not this one, etc. Then using that input a build time process could generate the cocoon specific sitemaps.

The main plugin.xmap would reference each individual plugin. In this example there are two plugins each assigned to separate urls, communityList, or the generic plugin2. This file is a bit complicated but it basically performs the current function of the web.xml for servlet inside of cocoon.

```
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:pipelines>
    <map:pipeline>
      <map:match pattern="communityList**">
        <map:mount uri-prefix="test1" check-reload="no" src="plugins/CommunityList.xmap"/>
      </map:match>
      <map:match pattern="plugin2**">
        <map:mount uri-prefix="" check-reload="no" src="plugins/plugin2.xmap"/>
      </map:match>
    </map:pipeline>
  </map:pipelines>
</map:sitemap>
```

Then an individual plugin would look something the following. The basic idea is that a plugin pulls together the various resources of the product. In this case there is one thing that is going on, the community list is put on the page and then a theme is applied.

```
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:components>
    <map:transformers>
      <map:transformer name="CommunityList" src="org.dspace.app.xmlui.CommunityList"/>
    </map:transformers>
  </map:components>
  <map:pipelines>
    <map:pipeline>
      <map:match pattern="**">
        <map:generate src="../protodocument.xml"/>
        <map:transform name="CommunityList"/>
        <map:transform name="applyThemes"/>
        <map:serialize type="xml"/>
      </map:match>
    </map:pipeline>
  </map:pipelines>
</map:sitemap>
```