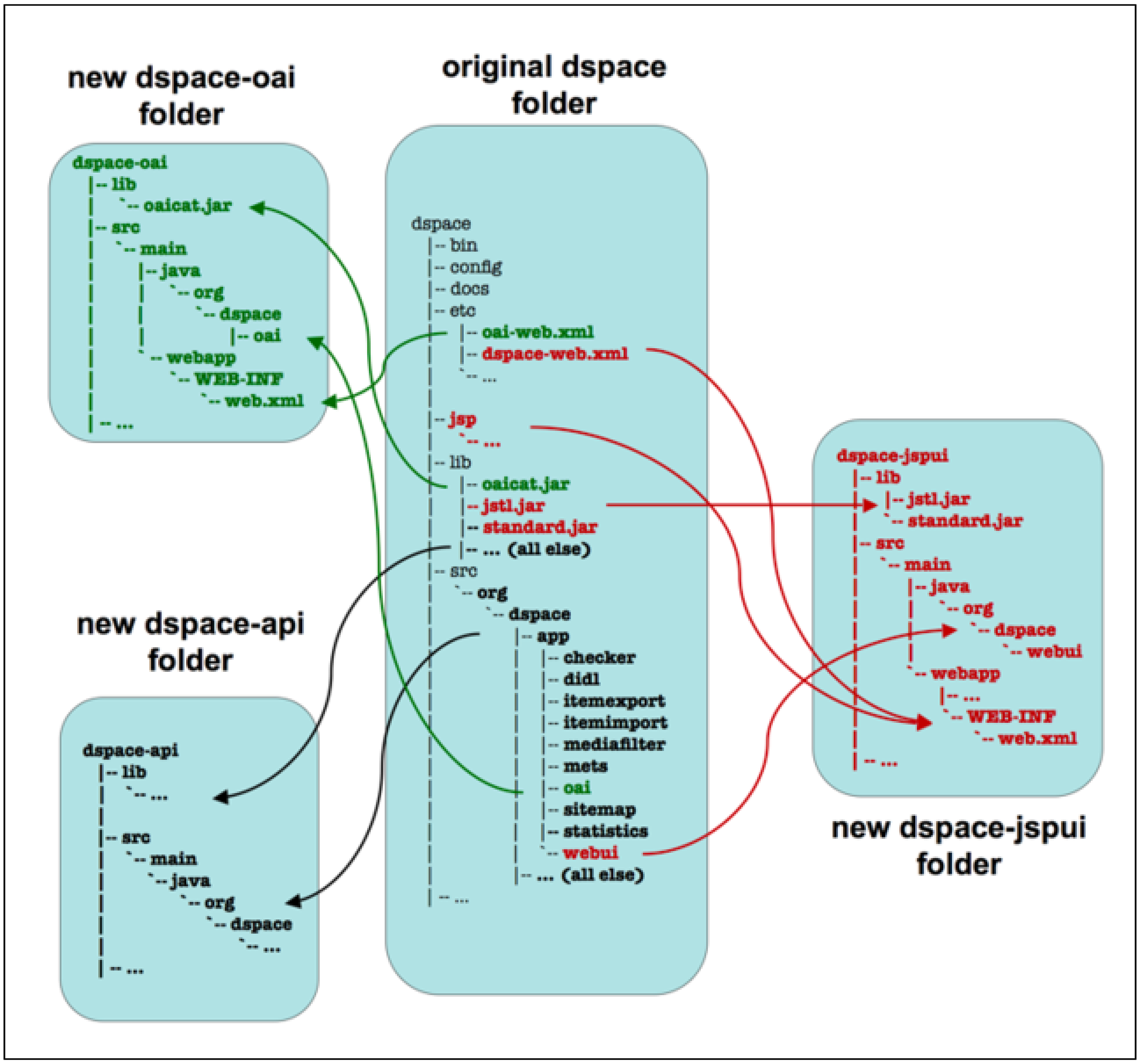
SimpleAddonMechanism CodeReorganization

<?xml version="1.0" encoding="utf-8"?>
<html>
Reorganizing the codebase for 1.5 will open up many doors for dspace modularity. But more simply, in this initial refactoring, we are factoring out the webapplication codebase for the JSPUI and the OAI webapplications into separate projects. These modifications outlined below will complete the following restructuring:

Overview



Itinerary of moves:

<table border="1" width="100%">
<tr><td colspan="2">**dspace-oai**</td></tr>
<tr><td>Move dspace/src/org.dspace.app.oai</td><td>to dspace-oai/src/main/java/org.dspace.app.oai</td></tr>
<tr><td>Move dspace/etc/oai-web.xml</td><td>to dspace-oai/src/webapp/WEB-INF/web.xml</td></tr>
<tr><td>Move dspace/lib/oaicat.jar</td><td>to dspace-oai/lib/oaicat.jar</td></tr>

<tr><td colspan="2">**dspace-jspui**</td></tr>
<tr><td>Move dspace/src/org.dspace.app.webapp</td><td>to dspace-jspui/src/main/java/org.dspace.app.webapp</td></tr>
<tr><td>Move dspace/etc/dspace-web.xml</td><td>to dspace-jspui/src/webapp/WEB-INF/web.xml</td></tr>
<tr><td>Move dspace/lib/jstl.jar</td><td>to dspace-jsp/lib/jstl.jar</td></tr>
<tr><td>Move dspace/lib/standard.jar</td><td>to dspace-jsp/lib/standard.jar</td></tr>

<tr><td colspan="2">**dspace-api**</td></tr>
<tr><td>Move dspace/src/org.dspace</td><td>to dspace-api/src/main/java/org.dspace</td></tr>
<tr><td>Move dspace/lib</td><td>to dspace-api/lib</td></tr>
</table>

These steps describe all the operations necessary to prepare DSpace for 1.5 (dspace-xmlui, addon mechanism, etc). Once the structure has been worked out and the community likes the result this script will be run on the SVN trunk.

Preparation

1. Check out DSpace trunk
2. Apply Uiutil-refactor_patch.mht to release dependencies on webui in mets.MetsExporter

SVN shell script to accomplish important source tree moves

Sometimes it's better just to converse on a subject in code, Here are the following svn moves and commits to accomplish restructuring above (this doesn't include any refactorings of code, build or configuration, just code, resource and library moves).

Note: Incremental Commits occur throughout this script to assure that parent locations exist in svn repo prior to svn mv's, this helps to assure that file history is properly brought along and the mv doesn't turn into a add without history.

Svn-restructure.sh

Post Reorganization additions

Apply Create-addon-project-dirs.htm to create the maven pom.xml build descriptors.

Handling Changes to project organization in the patch queue and working copies.

Patch Queue

Most patches already require "massaging" the level at which they are applied, there are patches in the patch queue which were made outside of svn and cvs which will never really be applicable.

To get patches that are properly created against the current dspace trunk to be applicable against the new reorganized trunk, I've written a little string replacement ant script that should do the trick.

Build.xml

Place this in the same directory as your patch and execute it using the following command:

```
ant -Dpatch=<your-file-here>
```

It will process the paths within the patch so that they point at the appropriate locations. **Caution, it's best to make a backup copy and compare them using diff to verify that the paths have been adjusted properly.**

Working copies

1. If the developer has done minor changes and can save those somewhere, then simply checkout a new copy of trunk and move changes by hand.
2. If the changes are more pervasive.
    - Create a patch against the current working copy revision
    - process it with the above script to move the file locations
    - reapply the patch to a new checkout of the trunk.

</html>