Code organization

Our subversion repository can be browsed here:

https://open.med.harvard.edu/svn/eagle-i-dev

The two relevant directories are:

- **apps**: the code for the different software components and the artifacts to compile them.
- **datamodel**: artifacts for generating the eagle-i ontology and packaging it into a jar. Note that directories were most *.owl reside are pulled in as externals, so they won't be visible with the above links (you'll need to checkout the tree to see all directories).

apps

- **trunk** - contains the active development of the different software components.
- **branches** - contains the releases of the software.

Directory	Description
common	Miscellaneous Java services and utilities used by search and data tools applications. Includes: Solr index, auto-suggest, authentication and connection, logging, Jira connector for feedback,... |
configuration	Sample configuration files, configuration files used in AWS installer
common/ui	Front-end GWT code shared by search and data tools applications (SWEET and data management); static resources shared by all webapps (used as maven overlay).
datatools	Shared back-end code for read/write operations on an eagle-i repository; ETL code for generating eagle-i resource descriptions from excel and XML; Data management command line tools for performing bulk changes to an eagle-i dataset; SWEET front-end GWT code.
dev-resources	Resources for developers, such as eclipse configuration files.
model	Abstraction for interacting with an eagle-i annotated ontology, optimized for access from client code (API and Jena-based implementation).
reference-implementations	Code that is no longer active in the main codebase but is helpful as reference.
repository	Repository code and administrative tools. This directory is pretty much self-contained (except for its dependency on common/ui/static-resources).
search	Search application back-end.
webapp	User-facing web applications and web services except for repository. The modules in this directory produce .war files.

datamodel

- **trunk** - contains the active development of the ontology.
- **releases** - contains the releases of the ontology.