Dominic Lowe 04/07/06

Creation of CSML:

To create CSML, *somebody has to decide* upon the content of a data-granule, and then create a CSML file by either.

a) Scanning the data with the Scanner.
b) Develop some other automated process – could use the CSML parser to help.
c) Writing it by hand (not really an option, won't explore further).

Issues (BADC Specific):

- Underlying data is not usually CF-Compliant. If we want Standard Names for parameters at the browse level how do we create them from non standard names?
- NASA-Ames data-holdings are proving to be complicated to automate. This is mainly due to a large number of NASA-Ames files being poorly constructed. Many won't even open with NAPPY.
- BADC input needed on specification and identification of Data Granules.
- Also input needed on long-term storage questions. eXist/files, if files, where?
- We should add the access control metadata at the point of creation. Can this be derived from the unix file permissions?

Issues (Non BADC Specific):

- The scanner may not be suitable for use by all data providers
- The easiest alternative way to create CSML is using the CMSL parser. This is written in Python – is this a problem for data providers?

Creation of MOLES (inheriting from CSML):

I don't know how this is going to work. The options as I see it are:

- Write an XSLT to extract relevant bits from CSML doc and create a MOLES document.
- Use python and the CSML parser to extract the relevant bits (easy) and then write out MOLES (would need some code to write MOLES).

And what about the bits of MOLES that aren't in CSML? What is the process for population?

Parsing CSML:

Overall the parser seems fairly robust.
What about parsing CSML from eXist as opposed to files? (if this is necessary).

Discover and Browse Data Entities... leads to services accessing Data Granules (CSML)..

Services interacting with CSML:

An API to CSML (via the parser) is provided so services can:

- access a CSML file
- read the access control metadata
- get a list of features/featuretypes in that file
- get the domain of the feature, and more.
- subset a feature in time or space to create a new feature.
- receive a new CSML file and NetCDF file to describe the new feature.

This has been tested with the Data Extractor and a Web Coverage Service using a GridSeriesFeature. Both services worked and could subset and access data via the CSML api/parser. Following alpha, we need to review details of this interaction. The Data Extractor threw up more issues than the WCS, mainly because the DX has historically interacted with a full NetCDF data model. One of the main questions raised is whether the content of CSML is rich enough for client applications if all the 'browse' metadata is in MOLES.

Overall concerns:

Issues concerning CSML API interactions with client services will be fixable. I am more concerned about the quality of underlying data and the range of data formats and legacy data we agree to support. Especially with 'flexible' formats such as NASA-Ames which are often hand written, and a hindrance to systematic generation of CSML within BADC.