ProsAndConsOfRdf

This is a page to discuss the strengths and weaknesses of DF as a mechanism for storing metadata in DSpace. It's originally structured as a SWOT analysis, but it might turn out not to be appropriate.

Strengths

Flexibility

DF is excellent for introducing new metadata schemas for existing resources. It would be possible to use DF to drive many aspects of the application, as well as using it for traditional metadata.

Weaknesses

Learning Curve

The learning curve for DF is infamously steep - potentially reducing our developer pool. A part of this comes from the requirement to use a new query language (probably DQL).

Opportunities

Resolution To Support Conflicting Metadata

Resolution could be used to allow the system to sensibly support conflicting metadata information about a resource. This could be used by humans (e.g. an administrator giving a resource a different classification to the one chosen by the author) or machines (e.g. a particular DSpace instance preferring to trust representation information from JHOVE over that from FED).

Convenient Integration With Cocoon

If cocoon is chosen to support the next gen web application DF would be pretty convenient. y being a little careful with our DF (as per http://www.xml.com /pt/a/2002/10/30/rdf-friendly.html Making DF XML friendly) we could use XSLT (or XQuery, potentially) to generate presentation pages directly.

Threats

Volatility

The DF specs are fairly old, but have changed quite a bit between versions. Has the standard settled now? If not - we could end up with a considerable amount of effort to stay current. Will the use of 3rd party tools ameliorate this?

Unknown Scalability

I don't know how well DF tools scale, and I do know that DMS's scale well enough. If anyone has better information on this, clarification would be appreciated. JimDowning