

ActiveTriples Ordering

Context:

RDF does not do ordering, therefore we need to develop a method of ordering items in a collection/aggregation. We started the discussion with examining the current ordering ontologies:

1. RDF List / Link List
 - a. Pros: Traction in the community. Cons: Costly. Difficult to manage with large number of items.
2. ORE Model (Nodes and Proxies)
 - a. Pros: Integrates well with ActiveTriples. Cons: Not enough constraints
3. Collection List Ontology / Ordered List Ontology
 - a. Pros: Reduces possibility of error states Cons: Relationship to classes. When making changes you must also change classes.

Decision: We are going to start developing with the ORE Model. Lynnette (from LDL and Cornell) has started an ORE gem that she will share as an integration point with ActiveTriples.

Description of ORE gem and related LD4L work:

Diagrams:

- [High level architecture of the Virtual Collection GEM using various ontology GEMs based on ActiveTriples framework.](#)
- [Open Annotation GEM based on ActiveTriples framework](#) (ORE GEM follows a similar implementation structure)

Code:

- [LD4L](#)
 - [Virtual Collection GEM](#) - currently includes all ontology code embedded in this GEM. Ontology code is being extracted out into separate GEMs that this GEM will be updated to use when they are complete.
 - [ORE GEM](#) - under construction. Not pushed out. Code is in VirtualCollection as LD4L::VirtualCollection::Collection and LD4L::VirtualCollection::Item.
 - [Open Annotation GEM](#)
 - [FOAF GEM](#) - super simple; ready for others to expand

Discussion:

Items to be resolved:

1. What Ordering Identifiers should be used: numeric, previous/next, head
2. What to do about blank nodes?
3. There are some problems with ActiveTriples and composing Resources.
4. How will this map to ruby list, ruby array, ruby set?
5. What does RDF look like? What does API look like?

Action Items:

1. Lynnette will publish her work on the ORE gem to github.
2. Tom Johnson and Trey Terrell will prepare ActiveTriples for this integration.