



Ontolog

Getting OOR Development Going

A Proposal

17 September 2010

Expectations



- **Proposal to Get OOR Development Going (and keep it going)**
- **These are process issues - mostly**

Assumptions



- OOR Supports Evolutionary Development
- Partitioning of Functionality
- OOR does not store instance data apart from that of the OOR infrastructure (**resolved?**)
- OOR Supports arbitrary representation languages
 - Repository architecture (mostly) independent of language
 - Initial support for OWL
- Meta/Provenance information crucial
- Standards based to extent possible

OOB Architecture Principles



- OOB shall support evolutionary development
- OOB shall support distributed instances
- OOB shall be scalable
- OOB shall support federation
- OOB shall provide services decoupled from core repository functionality
- OOB shall have no hierarchical dependencies
- OOB shall support arbitrary ontology representation languages
- OOB shall be platform independent
- OOB shall be ontologically driven

OOR Artifacts Inventory



- Goals & Scope
 - [OntologySummit2008](#) Communiqué: Towards an Open Ontology Repository
- Architecture Principles
- Requirements
- Use Cases
- OOR "sandbox"
- BioPortal code base – The Fork
- Code repository - SemWebCentral

Gap Analysis



- Incomplete Use Cases
- Incomplete list of functionalities
- Incomplete requirements
- No requirements analysis & decomposition
- No (complete) architecture
- No (prioritized) action plan

Proposed Plan - Basis



- **Model Driven Approach**
 - Provides Organizing & Communication Mechanism
 - Eases collaboration (maybe, there are tool issues)
 - Helps in identifying priorities
 - Provides a level of abstraction
 - Eases understanding
 - Facilitates Collaboration
 - Supports parallel efforts
 - Provides basis for coordination

Proposed Plan - Actions



- **Develop up to date UML model**
 - Reverse engineer BioPortal Fork
 - Incorporate existing requirements and use cases
 - Make model part of CM repository
 - Add HTML report to Wiki page – Update regularly
- **Gather proposed functionalities**
 - Vet them
 - Prioritize them
 - Decide
- **Identify parallel/contributing projects**
 - Choose Coordination P.O.C.