Breakout session Information Domain

DELOS DL Reference Model

Resource Manager

- Resource = anything that can have an ID: Information objects, users, functions, etc.
- Hierarchy of resource types modeled as an object hierarchy with inheritance
- Relationships between resources Hierarchy of relationship types (themselves resources)
- Advantages
 - Unified handling of relationships and functions that apply to all resource types
 - Unified view in the user interface for operating on different resource types
 - Unified search

Types of resources (examples)

- Information objects (incl. annotations and metadata packets, also streams, databases)
- Queries and their results (virtual collections, virtual information objects, such as database reports)
- Users / agents that can search for, create, and manage resources
- Functions and services
- Software modules
- Policies
- Languages
- Each resource type still has its own domain where the semantics are defined. But they are all managed here

Resource type hierarchy

- Conceptual root: Resource Properties that hold for the root class resource
 - Resource has unique identifier
 - Resource < hasDescription> Information object
 - Resource <hasAnnotation> Information object
 - Resource has metadata
 - Other relationships: <isa>, <hasPart>, <has member>
 <isa> creates a hierarchy of resources within a type
 - Resources can be <u>combined</u> to form a new, compound resource or <u>collected</u> to form a collection (also a resource).
 Operators to create new resources
- Special resource types have their own properties / relationship types, for example
 - Image A <isColorHistogramOf> Image B

Information objects

- Annotations and metadata records are their own information objects A, B, C, ...
 A <annotates> B
 A <describes> B
- Information objects can be related to uses, for example,
 - intended use: external display vs merely for internal system use
- Information objects can be related in many ways (typed links) to form a hypermedia base
- Compound information objects, many types, including paths
- Special information objects: Streams, databases, virtual
- Need make model parallel to FRBR as much as possible, articulate differences

Relationship types (link types)

- Include basic relationship types and some examples in the reference model. DL designers may add their own
- Include notion of a "global" relationship type registry with syntax for defining properties of relationship
- Local relationship registry for a specific DL, managed by the resource manager
- Look to the Knowledge Organization Systems (KOS) community on thoughts how to do this

Role of databases

- There was discussion but not total agreement. Some considerations
- Databases (perhaps with the exception of straight transaction databases) can be part of a DL
- Search functionality being developed in the DL context can be used to improve retrieval in databases

Digital rights management (DRM)

- A pervasive problem, especially important in the information domain, but also for software objects and others
- Right = the ability of an agent to execute a function on a resource (User, functionality, resource, in a context)
- Resource manager needs to store data on rights, often relying on hierarchical inheritance (for example, any member of a group of users may have the right to see the information objects in a collection)
- Need a a mechanism to determine the rights of a given user with respect to a given resource, possibly a DRM expert system

Information objects

- Annotations and metadata records are their own information objects A, B, C, ...
 A <annotates> B
 A <describes> B
- Information objects can be related to uses, for example, intended use:
 - external display vs
 - merely for internal system use (such as an automated translation made merely for cross-language retrieval, not for display)