OOR in the Classroom
An Experience Report

Ken Baclawski

Northeastern University

OOR Promotion

- Techniques for promoting OOR
  - Personal contacts (bottom-up)
  - Mandates (top-down)
  - User-friendly tools and GUIs (lower the bar)
  - Classroom materials (raise awareness)
- This is a report on developing classroom materials
  - Provides useful feedback as a side-effect

Typical Course Assignments

- Course topic: component software with SOA
- UML model for the OOR as a web service
- XSLT transformations for OOR results
- Develop marshalling and basic semantics
- Develop a rudimentary OOR
  - Initially RESTful, later SOAP based
  - Federate OOR instances
  - Compose with other services

Issue: Storage Model

- Current storage model
  - Defined implicitly by program
  - Language dependent
- Better storage model
  - Formally defined by an ontology
  - Additional benefits
    - Better understanding and documentation of OOR
    - Important for federation and interoperability

Issue: Identifiers

- Currently identifiers are integers
  - Not clear how they are generated
  - Multiple codes/identifiers
- A standard for names/identifiers is needed
  - Essential for federated OOR instances
  - Useful for interoperability

Issue: Basic Semantics

- The most basic level of semantics is equality: when two entities are the same.
- Students are given the task of determining when two ontology labels (metadata only) are the same.
  - No formal definition so task is not well-defined
  - Equality of XML (infosets) is obviously not correct
  - Must separate fundamental attributes from incidental ones
- Formal semantics would resolve this issue

Issue: Marshalling

- Students develop serialization and deserialization using type mappings
  - Current approach
    - Procedural definition
    - Programming language dependent
  - Better approach
    - Formal specification of transformation
    - Unfortunately, ontologies are not designed for this

Issue: Service Specification

- Current model is RESTful
  - Informally defined
  - Request parameters are encoded in a URL
  - Result is encoded in XML
- Better model
  - Formal definition with ontologies
  - Request and result use RDF
  - Compatible with WSDL

Issue: Transformations

- OOR requires multiple representations for ontologies and metadata
- Two-way transformations are necessary
  - Invertibility would be ideal but usually impossible
  - Naïve transformations are usually unbounded
  - An attainable goal is stable transformations but these require annotations and not all languages can be extended to accommodate annotations.
- Students explore two-way transformations but this is a difficult research problem in general.

Future Issues

- Ontology configurations
- Situation semantics
- OOR component architecture