

KOTKA Collection Management System

Tapani Lahti

Finnish Museum of Natural History

DINA Workshop 17 Sep 2014

Background

- Need for a new collection management system was expressed in 2010
- Evaluation of existing systems and projects (incl. Specify and DINA): no satisfactory solution found
- A decision was made to develop an in-house system from scratch and make it compatible with existing infrastructure

Architecture

- Completely based on W3C Semantic Web technologies
 - Resource Description Framework (RDF)
 - Resources and statements
 - Classes and properties
- Agile Software Development Methods
 - Rapid deployment: first version out after 6mo
 - Continuous feedback from end users: 2wk cycle

Resources

- All data objects are described as resources having Uniform Resource Identifiers (URIs)
- Domain "http://id.luomus.fi/"
- Namespaces for different types of resources
 - MX for taxa
 - MA for people
 - MY and MZ for specimens and observations
 - ML for locations
 - e.g. "http://id.luomus.fi/MA.2", QName "MA.2"

Statements

- Collection data are stored as statements having subject, predicate and object/literal
- Example (N3 Notation)
<MA.2> <rdf:type> <abcd:Person> .
<MA.2> <MA.fullName> "Tapani Lahti" .
<MA.2> <MA.lastName> "Lahti" .
<MA.2> <MA.firstName> "Tapani" .
- Database structure is very simple (triplestore)
- New data structures can be implemented rapidly

Specimen class hierarchy

- Document: specimen details
- Gathering: locality, time, agents
- Unit: collected individuals
- Identification: species identification of samples
- Each class has properties
- New classes and properties can be added easily

From NoSQL to NoNoSQL

- First implementation running on MongoDB
- Later moved to Oracle DBMS
- Database schema has two tables:
 - RDF_RESOURCES: stores URIs
 - RDF_STATEMENTS: actual triplestore for subject-predicate-object triples
 - additional tables for history management