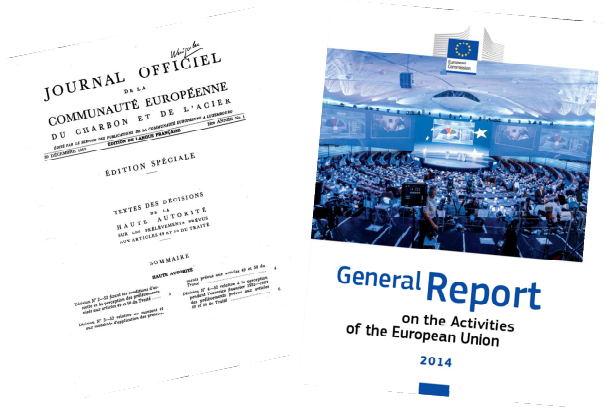
Towards executable Application Profiles for European vocabularies

Smart Descriptions & Smarter Vocabularies (SDSVoc) workshop
Amsterdam 30/11-01/12/2016

Willem van Gemert & Eugeniu Costetchi
Publications Office of the EU

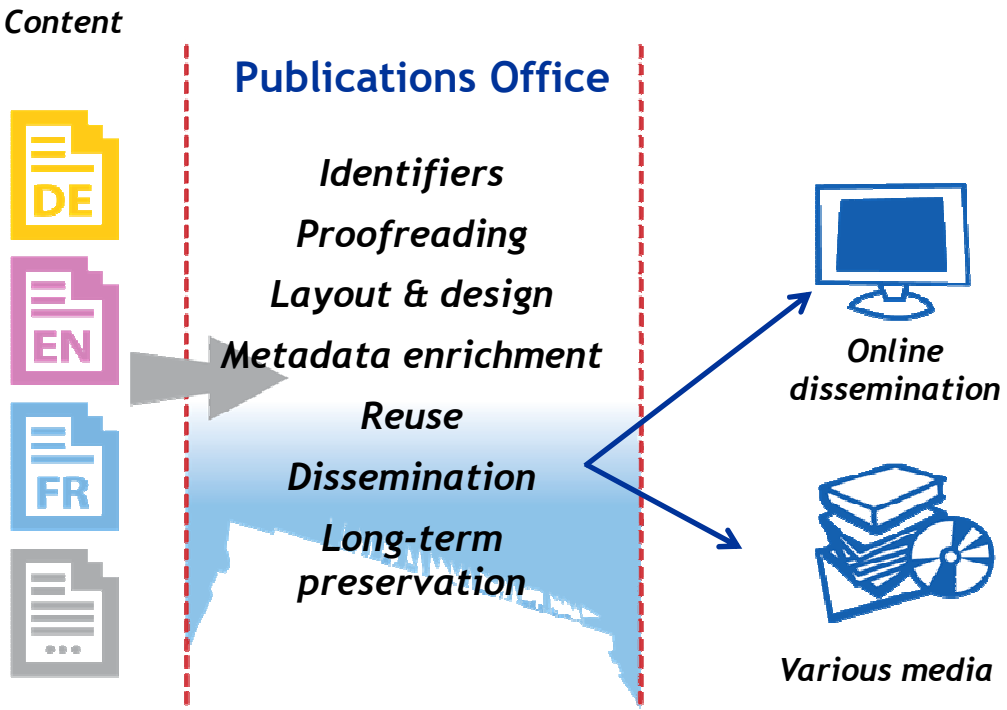
The Publications Office of the EU

- Inter-institutional service provider, evolving from traditional publisher to provider of information management services
- We publish EU law and other information from EU institutions, and we make it available for easy long-term access and reuse
- Three pillars of activities: **production**, **access** and **reuse**, **long-term preservation**
- We work for around 150 'author services' from EU institutions, agencies and bodies



Service provider for the EU institutions

- Commission
- Council
- Parliament
- Court of Auditors
- Court of Justice
- Committee of the Regions
- Economic and Social Committee
- European Central Bank
- EU agencies and bodies



Dissemination - main public online services

The Publications Office and metadata standardisation

- Main vocabularies
  - Metadata Registry (reference data repository)
    - 70 authority tables (dereferencable URIs)
  - EuroVoc (multilingual thesaurus of EU)
- Our services include:
  - **Maintenance**
  - **Governance** (Interinstitutional Metadata Maintenance Committee, EuroVoc maintenance committee)
  - **Alignment** with other controlled vocabularies (e.g. Agrovoc, Gemet, Inspire Themes, …)
  - **Persistent identification** (European Legislation Identifier (ELI), data.europa.eu persistent URIs, DOI for data)
  - **Tooling** (VocBench, IMMC builder, …)

Application profiles

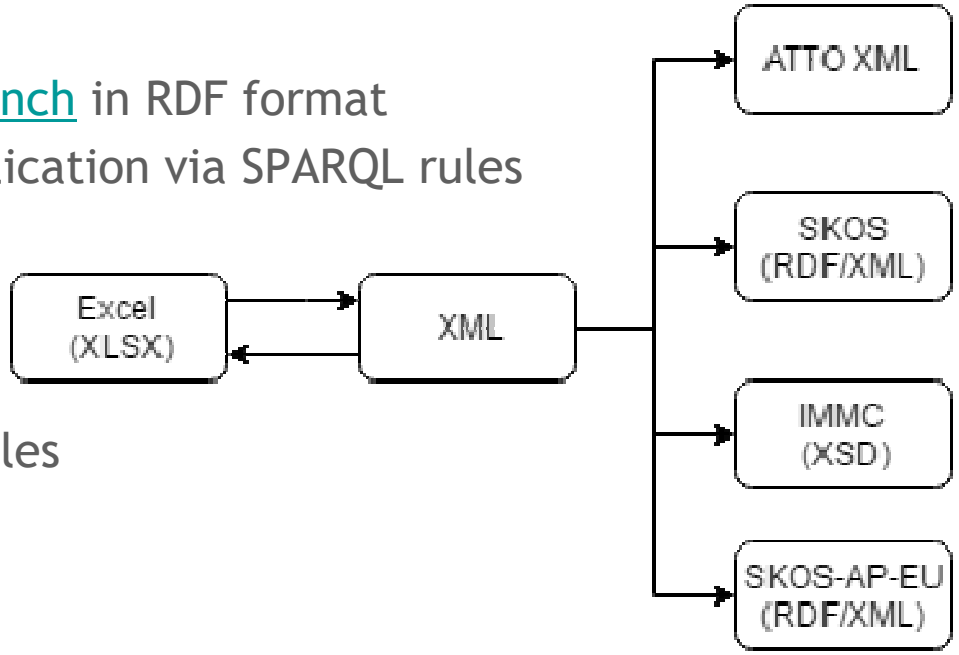
- For EU controlled vocabularies: SKOS-AP-EU
  - SKOS, DCT, Lemon, LexVo, etc.
- For the EU Open Data Portal: DCAT-AP-OP
  - DCAT, ADMS, DCT, etc.
- For the EU directory “Who is Who”: ORG-AP-OP
  - ORG, FOAF, Person, etc.

Context (NALs and EuroVoc)

- SKOS-AP-EU: **common representation** for EuroVoc and NALS

- EuroVoc
  - natively edited in VocBench in RDF format
  - transformed before publication via SPARQL rules

- NALs
  - source is XML
  - Transformed via XSLT rules



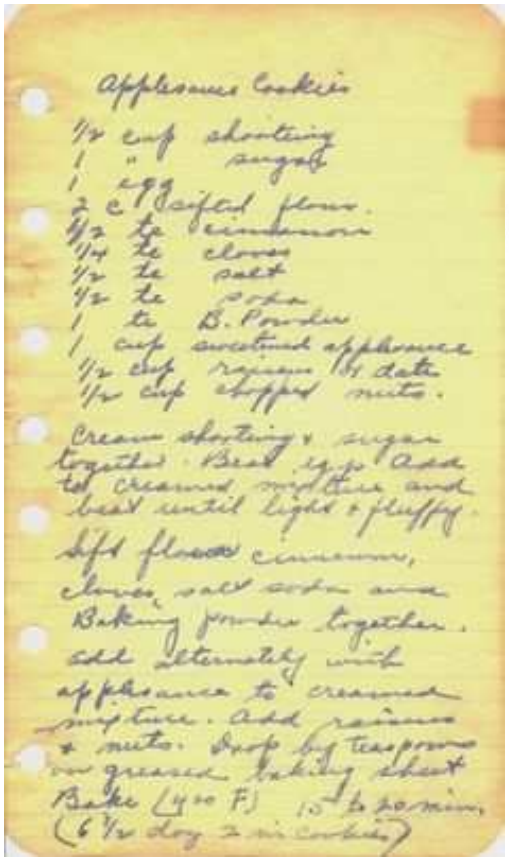
Issues

- Need for RDF **validation**
  - Conversion from RDF/XML and XML to RDF (SKOS-AP-EU)
  - (extra) checking of integrity constraints on source data

- The problem of **segregation** between documentation and implementation
  - Write it once (re)use in many places
  - How can be precisely *described*, *for both humans and machines*, what the data ought to be?

- **Data fingerprinting**
  - What are the patterns that data exhibit and to which level of regularity?

Christmas is coming



Potential approaches to “RDF validation”

- Modeling languages
  - OWL and RDFS
- XSD schema validation
- Query languages
  - SPARQL
- Rule languages
  - SPIN, ShEx, SHACL, SWRL, RIF

SHACL approach (answer to many but not all)

- SHACL shapes
  - Allows expressing AP constraints *and more*
  - Actually allows expressing *RDF graph patterns*
  - Allows shape templating (for reuse)

- Executable by SHACL **validation** engines
  - Available SHACL API (link)
  - We created a command line wrapper (link)

- Translatable into **human readable** documents
  - rdf:label, comment
  - Tabular structure of cardinality constraints
  - Set of properties per class organization

Example Constraints in SKOS-AP

- Property cardinality constraints
  - Exactly one creation date
- Property domain and range constraints
  - Concept status musty be of type euvoc:ConceptStatus
- Conditional constraints (if P then Q)
  - If there is an end date then there must be a start date
  - If C1 replaces C2 then C2 must have a deprecated status
- Complex expressions
  - Preferred label can occur only once per label
  - Cycle detection via SPARQL queries
- *and more*

Data Fingerprinting

- Reconstitution (archaeological approach). Analyzing RDF data is not straight forward task, reconstructing potential shapes that were applied at constructing the data.

- *Show an example report*

- Source Code
  - RDF fingerprinter Python script (link)
  - NodeJs UI wrapper (link, demo)

Conclusions

- We use SHACL
  - for validation
  - for documentation generation
- SHACL allows us to discover inconsistencies
  - in the data
  - in the transformation rules
- For now used for SKOS-AP-EU
  - can be applied to any other AP (e.g. **DCAT-AP, ORG-AP**)

- BUT!
  - is SHACL stable enough?
  - what are we to expect from it in the future?

Questions?

Thank You