

RF2 To OWL Transformation

Presentation Notes
SNOMED International MAG 04-25-2017
Harold Solbrig

RF2 to OWL

Project Scope

Formally document the rules in the “Spackman Perl Transformation”

- Fix errors
- Generalize — current transform only works with International Edition / US language refset
- Minor tweaks

RF2 to OWL

Not in Scope

- Topics not addressed in Spackman Script
 - Non-defining attributes
 - Concept History Attribute
 - Link Assertion
 - Unapproved Attribute
 - Non-defining Relationship File entries
 - Universal quantifiers
- OWL2 EL Logic Profile
- Anything not in Concept / Description / StatedRelationship / TextDefinition / Language tables
- Anything other than Snapshot
- (Many) other good ideas, ...

Minor Tweaks

1. owl:intersectionOf wrapper for rdfs:subClassOf entries removed
2. Labels:
 - sctf:Description.term.en-us.preferred “text”@us —> skos:prefLabel “text”@en-us
 - sctf:Description.term.en-us.synonym “text”@us —> skos:prefLabel “text”@en-us
 - sctf:TextDefinition.term “text”@us —> skos:definition “text”@en-us
3. 410662002 | Concept model attribute |
 - Descendants in Spackman Transform
 - Descendants plus root in proposed spec
4. Multiple language refsets (Spackman coded for exactly one)

Modules

Spackman Transform does *one* module at a time

1. Descendants of 410662002 | Concept model attribute | belong to ALL modules
2. All concepts that are defined in target module are emitted
3. All concepts have a FSN (another topic...)
4. If and only if a concept belongs to target module, all qualifying StatedRelationship entries are emitted
5. Any Description (TextDefinition) belonging to target module emitted iff corresponding Language refset entry exists in target module and language
6. Any Description (TextDefinition) having qualifying Language refset entry in target module and language emitted

Module issues

Approach 1

One OWL ontology for *everything* that is included in the module

- No owl:imports
- Inclusion is on the RF2 file level (or some clever use of ModuleDependencies file)
- Core metadata concept issues - tooling needs certain hierarchies

Module issues

Approach 2

One OWL ontology per module id

- owl:imports for each module dependency
- Modules are nonsensical by themselves
- Access to module content (!)
 - US Edition
 - all dependencies are part of distribution
 - Non-monotonic is possible (likely) (i.e. $f(\text{US Edition, INTL module}) \neq f(\text{INTL Edition, INTL Module})$)
 - CA Edition
 - NO dependencies are part of distribution
 - Where do we get them?
 - For import
 - For metadata (!) — all defining concepts, all descendants of X

State of Affairs

Module Philosophy Options

1. Do what Spackman transform (sort of) does...
 - Not all extensions / editions will be available in OWL
2. Make module decision and proceed

State of Affairs (continued)

<https://confluence.ihtsdotools.org/display/mag/Representation+of+SNOMED+in+OWL.v0.9>

- Needs review / formalization / process
- Q: Turtle — would OWL Functional be preferable?

State of Affairs (continued)

Tooling

- <https://github.com/hsolbrig/SNOMEDToOWL>
- Intended to be *a* reference implementation — the idea being that the SNOMEDToOWL tool serves as the formal description of the spec
- Three modules
 1. RF2Filter — extract parts of RF2 for simplicity
 2. SNOMEDToOWL — RF2 to OWL (many formats)
 3. CompareRDF — diff two RDF files

State of Affairs (continued)

Tooling (continued)

- **Maintained by Mayo**
 - **No guarantees for how long / how stable**
- **Already being extended and used for tasks beyond basic spec**
- **Recommendation:**
 - **Mayo does what is required to “productize”**
 - **Root moves to space owned by SNOMED INTL**
 - **Open source — extensions, etc. can be forked for future mayo work**
- **Spackman Perl Transform**
 - **Update code to make behavior consistent with new specification for the International Edition**
 - **Document clearly that extensions / editions do and do not work**
 - **Open source resulting software**