

Ontology based alignment of SNOMED CT Quality Assurance assertions

Jay Kola, Noesis Informatica Ltd, UK

Brian Carlsen, West Coast Informatics LLC, USA

Outline

- Why QA of SNOMED CT matters?
- How SNOMED CT is quality assured
- Motivation of work
- Methodology
- Findings
- Next steps

Why QA of SNOMED CT matters?

- Implementers need valid data
 - E.g. Metadata representations
 - E.g. Structural alignment (like single FN/PT)
 - E.g. Correct language representations (for consistent preferred name display)
- Processing deltas requires consistent history tracking
- Extension maintainers need to meet minimum validation requirements of the core
- QA enables strong statements about what is known about the terminology.
 - E.g. “every concept has a unique FSN for a given language”

How is SNOMED CT quality assured?

- Long topic... Published on IHTSDO website
- Overview
 - Ensure content is structurally valid
 - Ensure content is clinically valid
 - Distributed content follows editorial principles and RF2 specification
- QA performed:
 - During authoring – built into IHTSDO Workbench
 - Publishing time – nightly/on-demand prior to content release
- This presentation is NOT about 'authoring time QA' - Technical QA

What does Technical QA cover?

- Conformance of content to
 - IHTSDO Editorial Principles
 - RF2 Specification (including representation of historical states)
- Examples
- Editorial: All FSNs must have a trailing semantic tag.
- RF2 Spec: All referencedComponentIds in the en-US Language Refset must have a corresponding Description (id).
 - Variants based on RF2 release type – Full, Delta, Snapshot
- Structural: Concept file must have the following columns : id, effectiveTime, active, moduleId, definitionStatusId,
- All fully defined Concepts must have more than one Relationship - Guess type..?

QA Tooling Status

- IHTSDO uses a home grown tool called the 'Release Assertion Toolkit'
- Collection of various SQL scripts and Java code implemented as a Maven project
- Being rewritten as an 'API'
- Different member countries have their own home grown tools.
- Extension QA suites may require additional or different rules (e.g. a component cannot enter a release as inactive)

Motivation

- SNOMED CT facilitates cross border healthcare – advantage for international vendors
- However, different IHTSDO members interpret editorial and technical specifications slightly different
 - So published content is slightly different for different members.
 - E.g. Does ModuleDependencyRefset include transitive closure of all dependencies?
 - This is an issue for vendors operating across different members.
- RF2 Member Subgroup formed to resolve issue
- Review corpus of QA rules used by different members; to identify differences & harmonise if possible
- Create a corpus of common QA assertions that are applicable to all SNOMED CT releases and specifically to extensions.

Motivation (2)

- Corpus of QA assertions collated from members
 - 790 assertions
 - QA “meta model” developed to characterize checks
- Are all of these unique? – **Question # 1**
 - Visual inspection revealed overlap, but how do we identify equivalent assertions even when they are phrased differently?
 - Assertion 1250: Active field should be 0 or 1 - UKTC
 - Assertion 1044: 04 active is boolean value - AU

Motivation (3)

- Is it possible to identify assertions applicable to International Release vs. National Extensions? – **Question #2**
- Is it possible to identify assertions applicable to individual components – e.g. assertions applicable to concepts only! – **Question #3**

Methodology

- Our “problem space” aligns well with “harmonisation/normalization” of “data from heterogenous sources”.
- This is a well recognised use case for ontologies!
- Since we know assertions have overlap and are possibly duplicates, the OWL ‘open-world’ assumption works to our advantage
- We can actually state that two rules, even with different ids and labels are equivalent - **Question # 1**

Methodology (2)

Excel Spreadsheet
with Assertions



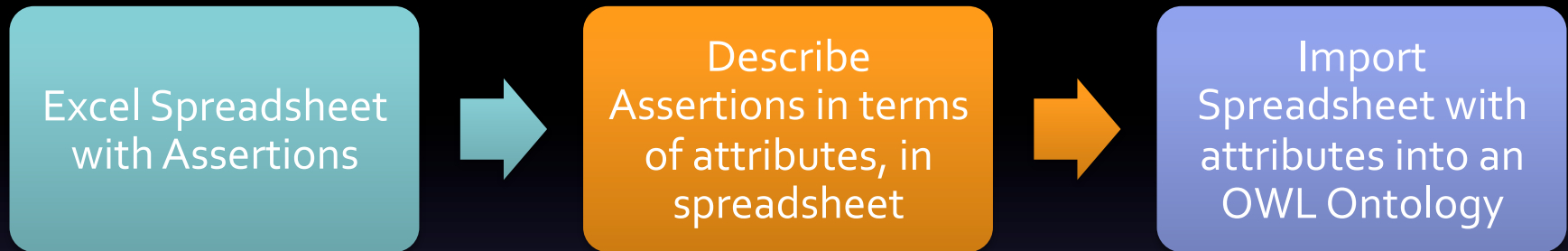
Describe
Assertions in terms
of attributes, in
spreadsheet



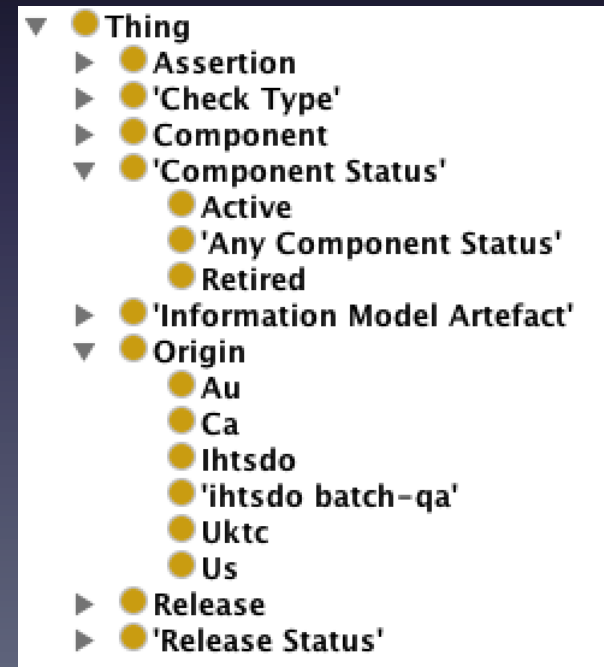
Import
Spreadsheet with
attributes into an
OWL Ontology

A	B	C	D	E	F	G	H	I
Id	Rule	Origin	Domain Objec(s)	Check Type (and ext- specific?)	Field, Line, File, All	Full, Snapshot, Delta	New, Current, Prior	Active, Retired
1001	inactive components have not been reactivated	au	Component	History	file	full	Any Release Statu	Any Component Sta
1002	Any changes to existing components have a new entry with current release date	au	Component	History	file	full	current	Any Component Sta
1003	All components from last full release are present in the current	au	Component	RF2	file	full	Any Release Statu	Any Component Sta
1004	All components from Reference international release full release are present in the current	au	Component	RF2	file	full	Any Release Statu	Any Component Sta
1005	All new components are active	au	Component	RF2	file	full	new	active

Methodology (3)



- Assign unique Ids to all assertions
- Treat all user assigned description as a 'label' – no role in inference
- Model each assertion in terms of attributes that allow inference/query
- All values of a given attribute, become a hierarchy in the OWL ontology



Ontology based alignment

- Describe each assertion based on attributes

Attribute	Possible values
Text/Human readable description	
Origin	IHTSDO, AU, UKTC...
Check Type	Referential Integrity, Valid values...
Release Type	Full, Delta, Snapshot
Component Status	Active, Inactive
Applicable Component	Concept, Description...
Release Status	Current, Prior
Information Model Artefact Type	File, Field Name

Ontology Based Alignment (2)

- Assertion 1173: Concept should have at least one IS_A relationship

Attribute	Value
Origin	IHTSDO
Component	Concept
Check Type	Referential Integrity
Release Type	Snapshot
Release Status	Current
Component Status	Active

Ontology Based Alignment (3)

- Assertion 1173: Concept should have at least one IS_A relationship

● Assertion
● has_associated_component some Concept
● has_origin some 'ihtsdo batch-qa'
● is_check_type some 'Referential Integrity'
● relates_to_artefact some File
● relates_to_component_status some Active
● relates_to_release some Snap
● relates_to_release_status some Current

Detecting equivalent assertions

FSN semantic tag related assertions

● 'component-centric-validation - All active FSNs have a semantic tag'

● 'FSNs of active concepts should terminate with a semantic tag'

● 'FSN descriptions must have a semantic tag'

- Level of definition
- What is the most appropriate level?

● Assertion
● has_associated_component some Description
● has_origin some Ihtsdo
● is_check_type some 'Valid Values'
● relates_to_artefact some Field
● relates_to_component_status some Active
● relates_to_release some <u>All</u>
● relates_to_release_status some Current

● Assertion
● has_associated_component some Description
● has_origin some Ca
● is_check_type some 'Valid Values'
● relates_to_artefact some Field
● relates_to_component_status some Active
● relates_to_release some <u>Snap</u>
● relates_to_release_status some Current

● Assertion
● has_associated_component some Description
● has_origin some Uktc
● is_check_type some 'Valid Values'
● relates_to_artefact some Field
● relates_to_component_status some Active
● relates_to_release some <u>Snap</u>
● relates_to_release_status some Current

Detecting Equivalence (2)

- Two approaches – manual equivalence assertion vs. create new assertion
- Manual:
 - We state $\text{Assertion\#1323} \equiv \text{Assertion\#1154}$
 - Quicker to create but slightly less reusable
 - Some equivalences may not be detected

● 'FSN descriptions must have a semantic tag'

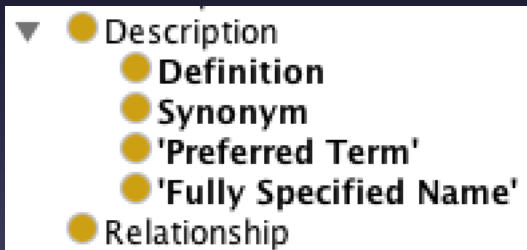
≡

● 'FSNs of active concepts should terminate with a semantic tag'

☰ 'FSN descriptions must have a semantic tag' = 'FSNs of active concepts should terminate with a semantic tag'

Detecting Equivalence (3)

- Create a normalised assertion
- Based on existing assertions, but needs further ontology modeling to fully encapsulate meaning.



```
● Assertion
  and (has_associated_component some 'Fully Specified Name')
  and (is_check_type some 'Valid Values')
  and (relates_to_artefact some Field)
  and (relates_to_component_status some Active)
  and (relates_to_release some All)
  and (relates_to_release_status some Current)
  and ('relates to component characteristic' some 'Semantic Tag')
```

Advantages of Normalised Assertion

- We can reuse the additional classes added to derive more information.
- Use the DL reasoner to infer related/associated assertions

Assertions associated with Fully Specified Name

-
- ▼ ⓘ 'Assertions pertaining to Fully Specified Name'
 - 'file-centric-validation - Active Fully Specified Name is unique in DESCRIPTION'
 - 'FSN cannot start with open parentheses'
 - 'FSN must end in closing parentheses'
 - 'FSN should be unique among all concepts'
 - ▼ ⓘ 'Normalised - All active FSNs have a semantic tag'
 - 'component-centric-validation - All active FSNs have a semantic tag'
 - 'FSN descriptions must have a semantic tag'
 - 'FSNs of active concepts should terminate with a semantic tag'


Assertions associated with semantic tag

Advantages (2)




- Derive usable information from the exercise
 - Given our knowledge of Release Types, can we infer assertions applicable for a Delta release? - **Question**

#3

Description: "Assertions pertaining to Delta Release"

Equivalent To 

- Assertion
and relates_to_release some Delta

- ▼  'Assertions pertaining to Delta Release'
 - 'Effective time should be the date of the latest release'
- ▼  'Assertions pertaining to Fully Specified Name'
 - 'file-centric-validation - Active Fully Specified Name is un
 - 'FSN cannot start with open parentheses'
 - 'FSN must end in closing parentheses'
 - 'FSN should be unique among all concepts'
 - ▶  'Normalised - All active FSNs have a semantic tag'

Advantages (3)


- Derive usable information from the exercise
 - Given our knowledge of Release Types, can we infer assertions applicable for a Delta release? - **Question**

#3

Description: 'Assertions pertaining to Delta Release'

Equivalent To 

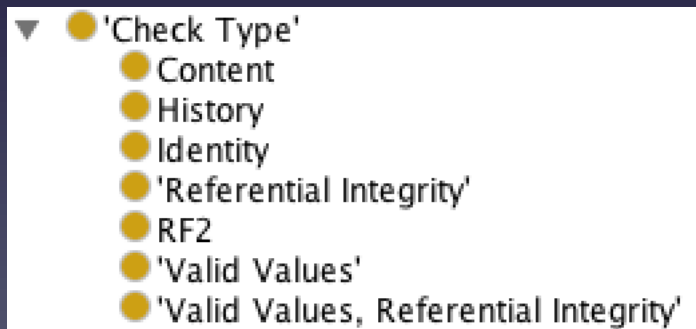
- Assertion
and relates_to_release some (Delta or All)

- ▼  'Assertions pertaining to Delta Release'
 - '001 rf2_cr_relationships_full has been populated'
 - '01 Descriptions conceptId hasn't changed since last release'
 - '01 rf2_cr_cRefset_full has been populated'
 - '01 rf2_cr_descriptions_full has been populated'
 - '01 rf2_cr_identifiers_full has been populated'
 - '01 rf2_cr_refset_full has been populated'
 - '01 rf2_cr_sRefset_full has been populated'
 - '01 Typeld for active relationships are descendent of 4106620

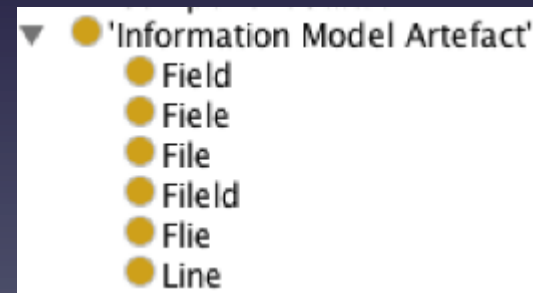


Advantages (4)

- Helps identify data quality issues – typos, minor alterations in labels, etc.
 - Note this is an 'added benefit' of the approach. The objective is not data cleansing, but data alignment...



Grouping of different checks



Errors in categorisation

Status

- Current
 - Still work in progress – ontology still in development
- Future
 - How can an NRC use this ontology? – Implementation!
 - Investigate possible alignment with the Release Validation Framework being developed by IHTSDO
 - Can this be used as an 'ontology' of SNOMED CT components and release artefacts?
 - Should IHTSDO publish and maintain an ontology like this for use by the member community?
 - Additional attributes could help reveal opportunities.

Questions....

- How can I contribute to this work (or contribute something better)? – get in touch with...
- Jay Kola: jay@noesisinformatica.com
- Brian Carlsen: bcarlsen@wcinformatics.com