HL7® FHIR® DevDays

JUNE | 2024

VCL, the ValueSet Compose Language, Brings CodeSystems to Life

Michael Lawley, CSIRO

HL7 FHIR DevDays 2024 | Minneapolis, MN | June 10-13, 2024 | @HL7 | @FirelyTeam | #fhirdevdays | www.devdays.com

ORGANIZED BY firely | HL7 International

Who am I?

- Michael Lawley, PhD
- Group Leader
  Semantics & Interoperability
  - CSIRO*
  - Australian e-Health Research Centre
- Lead, Ontoserver
- SNOMED CT, 15+ years
  - ECL, post-coordination, OWL / description logic

* Commonwealth Scientific and Industrial Research Organisation
Australia's National Science Organisation

History

- SNOMED's ECL is very powerful mixed with implicit ValueSets
- FHIR-36651: Define a "term-filter" modifier on type "token"
  - define a URL-friendly syntax for ValueSet.compose
- Introduced VCL at FHIR DevDays 2023
- Feedback
  - Cannot be more expressive than ValueSet.compose
  - Syntax can be simplified

bit.ly/fhir-vcl

SNOMED's Expression Constraint Language (ECL)

- A compact language that allows matching sets of concepts based on code system properties:
  - Hierarchy traversal
  - Attribute (property) matching
  - Attribute navigation
  - ReferenceSet membership

SNOMED's ECL in FHIR

- Exposed as a ValueSet.compose filter ("`constraint`"), and via implicit ValueSet URIs:

  `http://snomed.info/sct?fhir_vs=ecl/`**`[expression]`**

- `[base]/ValueSet/$expand?url=http://snomed.info/sct?fhir_vs=ecl/`**`[expression]`**

- Very powerful for dynamic use-cases such as analytics or dynamic UIs

Scope

Goals

- Easy to write & read
  - e.g., ECL-like
- Compact
- URI-friendly
  - Limit need for whitespace
  - Avoid chars like **#**, **?**, and **&**

Non-goals

- Completeness
  - Parity with ValueSet.compose
  - Parity with ECL
- ECL compatibility

VCL intuition

- Sub-expressions for each `compose.(in/ex)clude.filter`
  - `property :op = value`
- Some special syntax for hierarchy traversal (borrowed from ECL)
  - `< code, << code, > code, >> code`
- Syntax for “and” and “or”
- Quoting rules
- Extras:
  - Path navigation
  - Term filtering
  - ValueSet membership

New since last year

- And/or *
  - “nesting”
- Text (designation) matching *
  - {{ term=“der Satz”, use=display, language=de }}
- Simplified syntax (remove ECL “quirks”?)
  - No more “:” – and semantics already covered using “,”

[*] beyond current ValueSet.compose capability

Using VCL in FHIR

- Exposed as a ValueSet.compose filter ("vcl"), and via implicit ValueSet URIs:

  [CodeSystem_URI]?vs=vcl/**[expression]**

  http://loinc.org?vs=vcl/SCALE_TYP='LP32888-7'

- [base]/ValueSet/$expand?url=**[CodeSystem_URI]**?vs=vcl/**[expression]**

- Very powerful for dynamic use-cases such as analytics or dynamic UIs

VCL Syntax

```
vQuery      : expr EOF ;
expr        : subExpr (conj | disj | excl )? ;
subExpr     : simpleExr filterExpr* ;
disj        : (or subExpr)+ ;
conj        : (and subExpr)+ ;
excl        : minus simpleExpr ;
simpleExpr : ('*' | conceptExpr | valueConstraint) ;

propNav     : (dot name)+ ;
propPath    : (name dot)+ ;
valueConstraint : propPath?
           ( simpleAttr | existsAttr | regexAttr | inAttr ) ;

simpleAttr : name '=' value ;
existsAttr : name ':exists' '=' ('true' | 'false') ;
regexAttr  : name ':regex' '=' STRING_VALUE ;
inAttr     : name (':in'|':not-in') '=' ( codelist | uri ) ;

value       : bool | code | conceptExpr | number | str ;

codelist    : '(' code (',' code)* ')' ;
uri         : URI ;
name        : NAME | NUMBER | '*' ;
bool        : 'true' | 'false' ;
code        : CODE | NAME | NUMBER ;
number      : NUMBER ;
str         : STRING_VALUE ;
```

```
or          : ';' ;
and         : ',' ;
minus       : '-' ;
dot         : '.' ;

conceptExpr
            : ('<<' | '<' | '>' | '>>')?
            ( code | '(' expr ')' )
            propNav? ;

filterExpr : '{{' name '=' value '}}' ;
```

**

Strings are (double) quoted

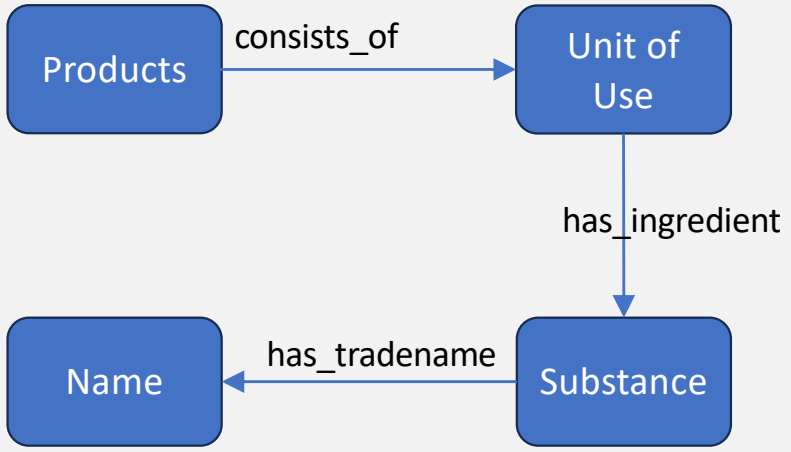
Codes must be (single) quoted if they contain non-alpha characters (in both cases with backslash as an escape char)

URIs are surrounded by < and >

Examples

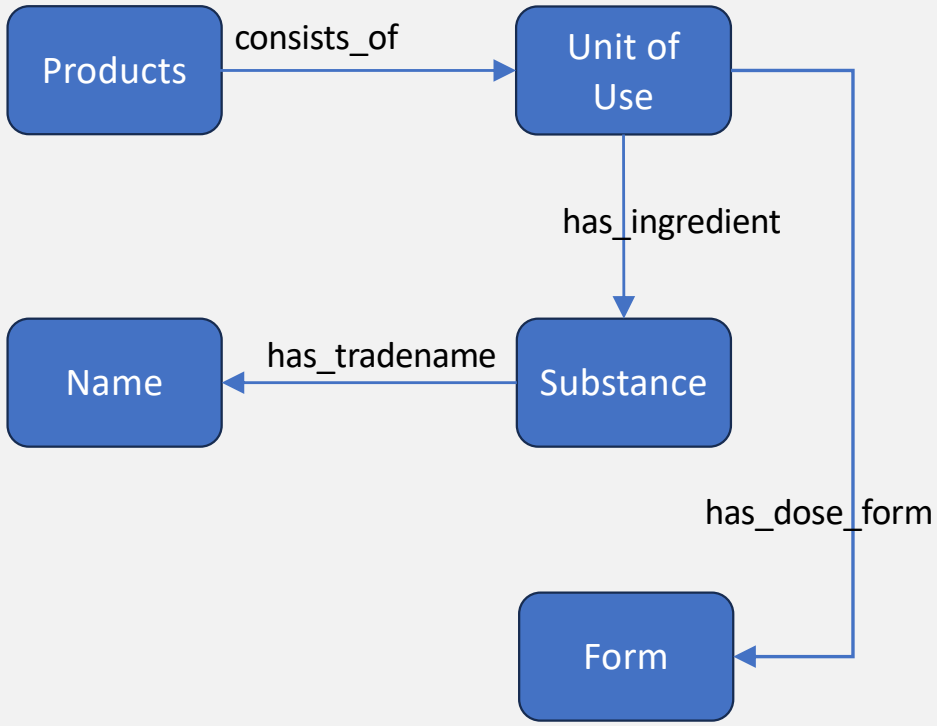
- Caffeine
  `1886`
- contains caffeine
  `has_ingredient=1886`
- Products with caffeine
  `consists_of.has_ingredient=1886`
- Products with same ingredient as NoSnooze

```
consists_of.has_ingredient=
     (has_tradename=2201670)
```



Examples

- contains caffeine in Oral Pill form
  `has_ingredient=1886, has_dose_form=317541`
- contains caffeine in any Pill form
  `has_ingredient=1886, has_dose_form=<<1151133`



Expression	Meaning
B	the code ‘B’
<< B	the codes ‘B’ or any descendant of ‘B’
*	all codes

Expression	Meaning
B	the code ‘B’
<< B	the codes ‘B’ or any descendant of ‘B’
*	all codes
parent = B	the codes that have ‘B’ as a (direct) parent
prop_name = B	the codes that have ‘B’ as the value for their ‘prop_name’ property
prop_name = “string”	Same, but for string-valued properties
inactive = true	All inactive codes
prop1 = B , prop2 = “C”	”and” – intersection of codes matching each sub-expression
prop1 = B ; prop2 = “C”	“or” -- union of codes matching each sub-expression

Expression	Meaning
B.codeprop	the values of the ‘codeprop’ property of ‘B’
<B.codeprop (< B).codeprop	the values of the ‘codeprop’ property of all codes that are descendants of ‘B’
B.codeprop1.codeprop2	the values of the ’codeprop2’ property for all codes that are the value of the ‘codeprop1’ property of ‘B’
codeprop1.codeprop2 = C	Codes that have a ‘codeprop1’ property with a value that has a ‘codeprop2’ property with the value ‘C’

Expression	Meaning
B.codeprop	the values of the ‘codeprop’ property of ‘B’
<B.codeprop (< B).codeprop	the values of the ‘codeprop’ property of all codes that are descendants of ‘B’
B.codeprop1.codeprop2	the values of the ’codeprop2’ property for all codes that are the value of the ‘codeprop1’ property of ‘B’
codeprop1.codeprop2 = C	Codes that have a ‘codeprop1’ property with a value that has a ‘codeprop2’ property with the value ‘C’
ingredient:exists = true	concepts that have a value for the property ‘ingredient’
code:regex = “A[0-9]*\\.9”	concepts where the code matches the regex ‘A[0-9]\.9’
property:in = (123,456)	concepts where the ‘codeprop’ property value is ‘123’ or ‘456’
has_ingredient:in = <http://acme.com/VS/controlled>	concepts with an ingredient in the controlled substance ValueSet

Open issues

- ValueSet.compose limitations:
  - designation filters
  - ValueSet membership (also required for nesting)
  - “reverse” filters (property navigation)
- Filter ops:
  - = | is-a | descendent-of | **is-not-a** | regex | in | not-in | generalizes | child-of | **descendent-leaf** | exists
- Multi-code system ValueSets
- CodeSystem supplements

FHIR-36651: next steps

ONTOSERVER

PoC work-in-progress implementation in Ontoserver sandbox

”vcl” filter available for all* CodeSystems

https://r4.ontoserver.csiro.au/fhir

Feedback please: bit.ly/fhir-vcl
(https://chat.fhir.org/#narrow/stream/179202-terminology/topic/VCL)

Alternate experimental implementation

- Josh Mandel: https://github.com/jmandel/TermSqlite

Changes along the way in making this work:

- *Added optional quoting for identifiers*
- *Replaced ":" in the modifiers to avoid this ambiguity*
- *Added support for ands/ors at the expression level as well as the constraint level*
- *Added reverse paths: ".^" syntax*

Contact

- You can find / reach me here:
  - Email – michael.lawley@csiro.au
  - Twitter – @lawley
  - Zulip

ORGANIZED BY

RxNorm examples

Things consisting of things having ingredient caffeine (1886)

- `consists_of.has_ingredient = 1886`

…and also having Oral Tablet dose form (317541)

- `consists_of.has_ingredient = 1886, has_dose_form = 317541`

…or, what are the available dose forms:

- `(consists_of.has_ingredient = 1886).has_dose_form`

http://hl7.org/fhir/ValueSet/doc-typecodes

```
"compose": {
  "include": [{
  "system": "http://loinc.org",
    "filter": [{
      "property": "SCALE_TYP",
      "op": "=",
      "value": "LP32888-7"
    }]
  }]
}
```

```
SCALE_TYP='LP32888-7'
```

```
"compose": {
  "include": [{
  "system": "http://loinc.org",
    "filter": [{
      "property": "vcl",
      "op": "=",
      "value": "SCALE_TYPE='LP32888-7'"
    }]
  }]
}
```