LexEVS 6.x Value Set Detailed Design

Contents of this Page

LexEVS Value Set Links

- Value Set Guide Main Page
  - Value Set Design
  - Value Set Service API
  - Pick List Design
  - Pick List Service API
  - Value Set GUI
- Programmer's Guide Main Page
- LexEVS 6.0 Main Page
- LexEVS Current Release

Value Set Definition

Value Set Definition with in the LexGrid logical model defines the contents of Value Set. The contents are concept codes defined in referencing Code System. Value Set can contain concept codes from one or more Code Systems.

Compared to ISO 11179 Model

The LexGrid "value set definition" is analogous to the 11179 enumerated conceptual domain. We support the notion of an intrinsically defined enumerated domain, so the 11179 enumerated conceptual domain is really the output of the resolve value set definition function.

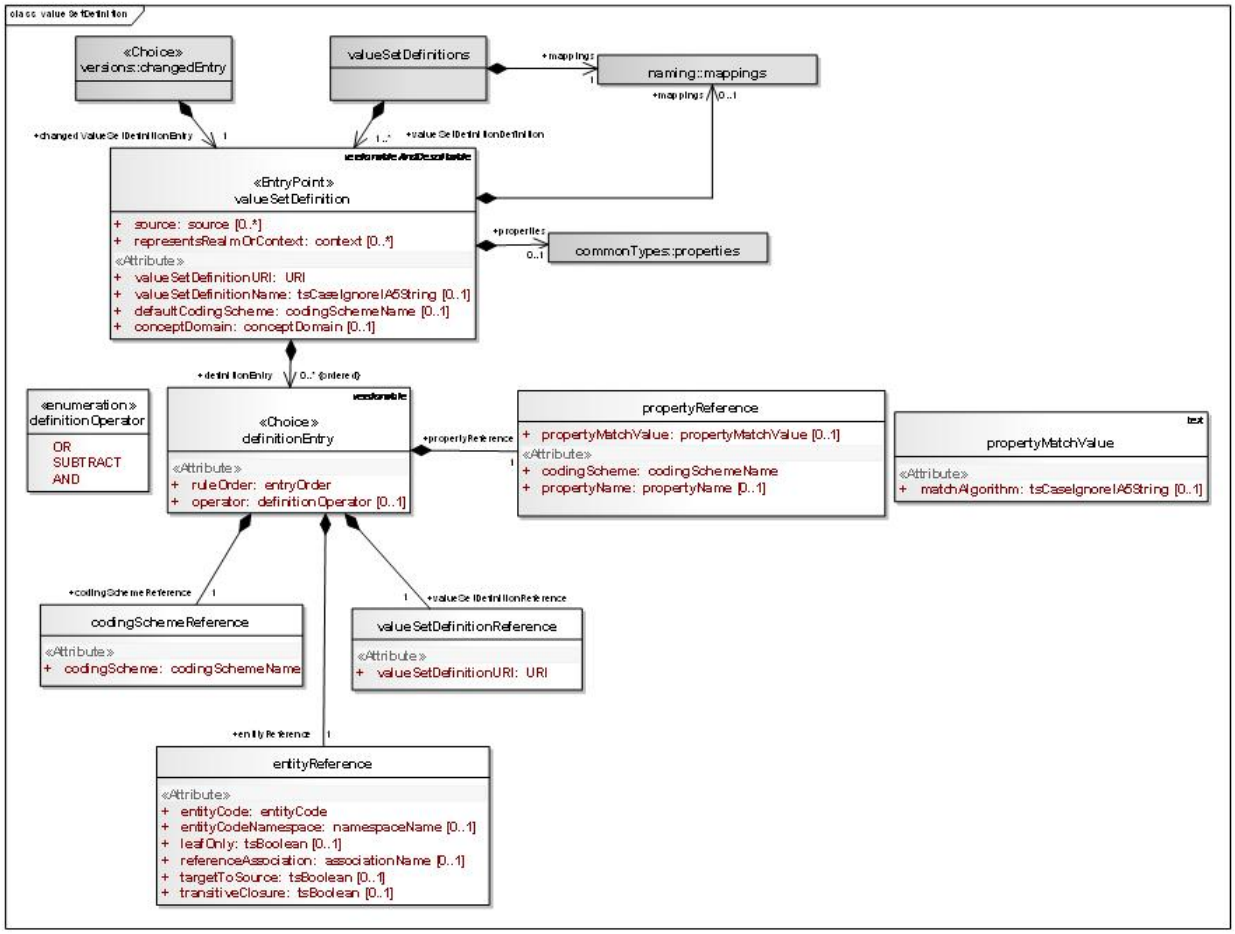
Value meanings are identified by entity codes within a given code system. The mapping between permissible values and value meanings is currently accomplished via a mapping association, where we treat the set of permissible values as a "mini code system". An example of how this might work is that HL7 has a set of permissible values of "M", "F", "U", which are in the administrative gender "code system". In the pure HL7 context, these might well map to the equivalent value meanings identified by "M", "F" and "U". If, however, we were using, say, the UMLS as a source of meaning, the same codes would map to the corresponding UMLS CUIs.

Compared to HL7 Value Set

The term "value set," when discussed in the context of HL7 can be somewhat ambiguous. There are at least three related artifacts that are sometimes called "value sets", including 1) a definition or algorithm that, when interpreted, produces a set of concept codes, 2) the actual set of concept codes that result from the execution of a definition or algorithm, and 3) a subset of this set of concept codes coupled with appropriate designations and identifying information. For this reason, the LexEVS model uses the names "value set definition", "value set resolution" and "pick list definition", "pick list resolution" respectively to represent the three different senses of "value set" described above.

Value Set Definition Logical Model

The following figure shows an overview of the value set definition logical model described in the following section.



Value Set Definition logical Model description

Value Set Definition

A definition of a given value set. A value set definition can be a simple description with no associated value set entries, or it can consist of one or more definitionEntries that resolve to an enumerated list of entityCodes when applied to one or more codingScheme versions.

Attributes of Value Set Definition

The following are attributes of value set definition.

- **Source:** The local identifiers of the source(s) of this property. Must match a local id of a supportedSource in the corresponding mappings section.
- **representsRealmOrContext:** The local identifiers of the context(s) in which this value set applies. Must match a local id of a supportedContext in the corresponding mappings section.
- **valueSetDefinitionURI:** The URI of this value set definition.
- **valueSetDefinitionName:** The name of this value set definition, if any.
- **defaultCodingScheme:** Local name of the primary coding scheme from which the set is drawn. defaultCodingScheme must match a local id of a supportedCodingScheme in the mappings section.
- **conceptDomain:** Local name of the concept domain. When present, the contents of value set are considered to be binded to this specific concept domain. conceptDomain must match a local id of a supportedConceptDomain in the mappings section.

Definition Entry

A Value Set Definition can have 0 to any number of DefinitionEntry. Each DefinitionEntry basically is used to define the rules as to what entityCode(s) to be included or removed from the value set. There are four different kind of rules that can be set :

- **CodingSchemeReference**
- **ValueSetDefinitionReference**
- **EntityReference**
- **PropertyReference**

These references are described in the following sections. Each DefinitionEntry can contain **one and only one** of these references. Each DefinitionEntry will be processed independent of the other DefinitionEntries. They will be processed sequentially according to the **ruleOrder** and will be merged to the final set based on the **operator** specified.

There are three different type of **operator** that can used :

- **ADD** - The processed DefinitionEntry will be merged to the final set using the 'AND'. This will include only the entity codes that are both in the value set and the DefinitionEntry. (logical AND)
- **OR** - The processed DefinitionEntry will be merged to the final set using the 'OR'. This will add the set of entityCodes described by the current DefinitionEntry to the value set. (logical OR)
- **SUBTRACT** - The processed DefinitionEntry will be merged to the final set using the 'SUBTRACT'. This will subtract (remove) the set of entityCodes described by the current DefinitionEntity to the value set. (logical NAND)

Attributes of Definition Entry

The following attributes of definition entry.

- **ruleOrder:** Required unique identifier of the definition entry within the definition as well as the relative order in which this entry should be applied
- **operator:** How this entry is to be applied to the value set. Available operator : *OR*, *AND* and *SUBTRACT*

Coding Scheme Reference

A reference to all of the entity codes in a given coding scheme.

Attributes of Coding Scheme Reference

codingScheme: The local identifier of the coding scheme that the entity codes are drawn from . codingSchemeName must match a local id of a supportedCodingScheme in the mappings section.

Value Set References

A reference to the set of codes defined in another value set definition.

Attributes of Value Set Reference

valueSetDefinitionURI: The URI of the value set definition to apply the operator to. This value set may be contained within the local service or may need to be resolved externally.

Entity Reference

A reference to an entityCode and/or one or more entityCodes that have a relationship to the specified entity code.

Attributes of Entity Reference

The following are attributes of entity reference.

- **entityCode:** The entity code being referenced.
- **entityCodeNamespace:** Local identifier of the namespace of the entityCode. entityCodeNamespace must match a local id of a supportedNamespace in the corresponding mappings section. If omitted, the URI of the defaultCodingScheme will be used as the URI of the entity code.
- **leafOnly:** If true and referenceAssociation is supplied and referenceAssociation is defined as transitive, include all entity codes that are "leaves" in transitive closure of referenceAssociation as applied to entity code. If referenceAssociation is not supplied and transtiveClosure is 'false', this flag will be ignored. Default: false
- **referenceAssociation:** The local identifier of an association that appears in the native relations collection in the default coding scheme. This association is used to describe a set of entity codes. If absent, only the entityCode itself is included in this definition.
- **targetToSource:** If true and referenceAssociation is supplied, navigate from entityCode as the association target to the corresponding sources. If transitiveClosure is true and the referenceAssociation is transitive, include all the ancestors in the list rather than just the direct "parents" (sources).
- **transitiveClosure:** If true and referenceAssociation is supplied and referenceAssociation is defined as transitive, include all entity codes that belong to transitive closure of referenceAssociation as applied to entity code. Default: false

Property Reference

A reference to a propertyName or propertyValue and matchAlgoritm to use.

Attributes of Property Reference

The following are attributes of property reference.

- **codingScheme:** The local identifier of the codingScheme that this propertyReference will be resolved against. codingScheme must match a local id of a supportedCodingscheme in the corresponding mappings section.
- **propertyName:** The local identifier to be used to restrict the entities to have property with this name. Must match a local id of a supportedProperty in the corresponding mappings section.
- **propertyMatchValue:** Value to be used to restrict entity property. matchAlgorithm can be used in conjunction to get matching entity properties.

Property Match Value

Property match value to be used to restrict entity property. matchAlgorithm can be used in conjunction to get matching entity properties.

Attributes of Property Match Value

matchAlgorithm: Algorithm to be used in conjunction with propertyValue.

Definition Operator

The description of how a given definition entry is applied.

Attributes of Definition Operator

The following are attributes of Definition Operator.

- **OR:** Add the set of entityCodes described by the currentEntity to the value set. (logical OR)
- **SUBTRACT:** Subtract (remove) the set of entityCodes described by the currentEntity to the value set. (logical NAND)
- **AND:** Only include the entity codes that are both in the value set and the definition entry. (logical AND)

Possible Forms of Value Set Definitions

The following are possible forms of value set definitions.

- Definition containing just the reference to Code System - includes all the concept codes in the referencing code system.
- Definition containing just the reference to other Value Set Definition -includes all the concept codes defined in the referencing Value Set Definition.
- Definition containing reference to Code System plus concept codes - includes individual concept codes defined in the definition from the referencing code system.
- Definition containing reference to Code System plus concept codes plus relationship plus additional rules (leaf only, immediate children, matching property name/value etc) - includes concept codes from the referencing code system that satisfies the rule set defined in the definition.
- Combination of any of the above with OR/AND/SUBTRACT operations.

Definition of the Value Set could be as simple as specifying just individual concept codes or specifying to include all the children of concept 'Body Structure' from Code System 'SNOMED CT' to complex definition containing multiple rule sets.

Examples of Value Set Definitions

Example 1

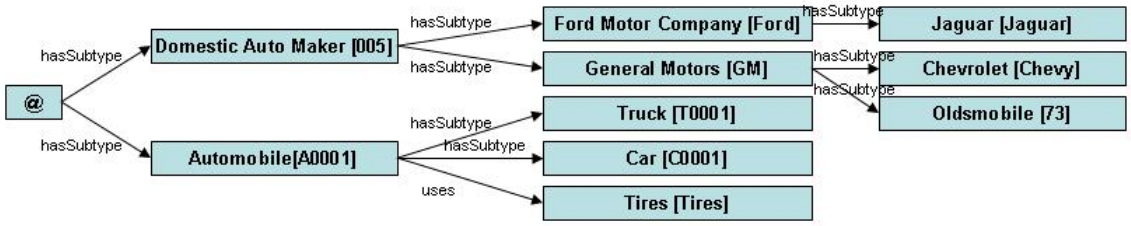
So we have a Code System "Automobiles" that contains concept codes Domestic Automaker and Automobile. Domestic Automaker contains Ford and General Motors. Automobile contains Truck, Car, and Tires.

We define a Value Set Definition to contain all the concept codes to contain Code System 'Automobiles.

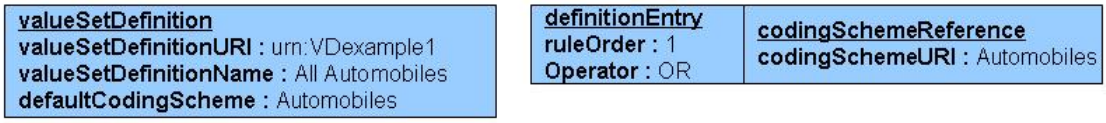
```
valueSetDefinition
ValueSetDefinition URI : urn:VDexample1
ValueSetDefinition : All Automobiles
defaultcodingcheme: Automobiles
```

And when this definition is resolved, we will get back all the preferred designations.

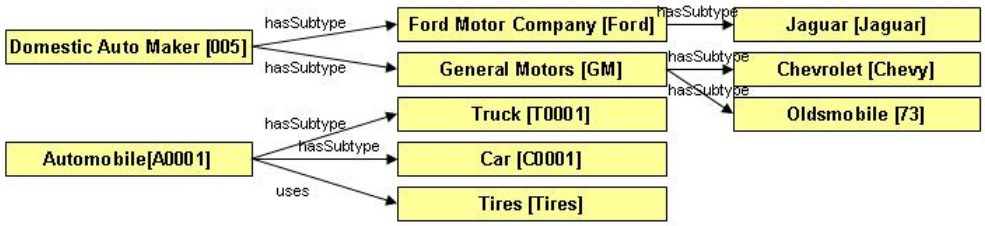
Say we have a Code System 'Automobiles' that contains following concept codes:



So to define Value Set Definition to include all the concept codes in Code System 'Automobiles'. We could define Value Set Definition as simple as:



And when this definition is resolved, we get back following concept codes:



Example 2

Using the same Code System "Automobiles" if you want to define the same value set definition to include all concept codes except "General Motors" and the children, it can be defined as follows:

```
valueSetDefinition
ValueSetDefinition URI : urn:VDexample1
ValueSetDefinition : All Automobiles BUT GM
defaultcodingcheme: Automobiles
```

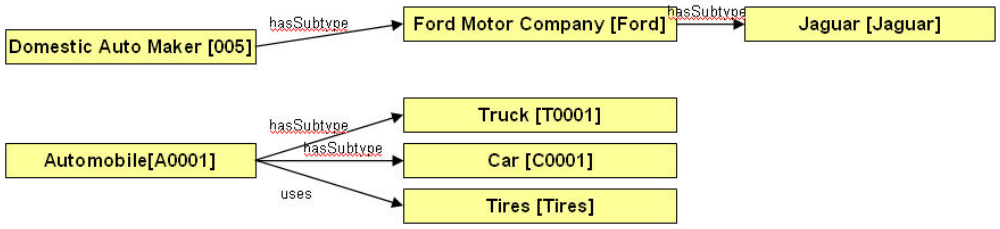
And when this definition is resolved, we will get back all the preferred designations.

Using the same Code System 'Automobiles', if we want to define Value Set Definition to include all concept codes except 'General Motors' and its children, we could define it some thing like the following using two definition entries:

valueSetDefinition
valueSetDefinitionURI : urn:VDexample2
valueSetDefinitionName : All Automobiles BUT GM
defaultCodingScheme : Automobiles

definitionEntry **ruleOrder :** 1 **Operator :** OR	**codingSchemeReference** **codingSchemeURI :** Automobiles
definitionEntry **ruleOrder :** 2 **Operator :** SUBTRACT	**entityReference** **entityCode :** GM **referenceAssociation :** hasSybtype **transitiveClosure :** true

And when this definition is resolved, we get back following concepts:



Value Set Definition Resolution

The following is information about value set definition resolutions.

- A value set definition has to be made against a specific version of a code system.
- But it doesn't have to be resolved against the same version.
- Even a simple list (a, b, c, d) needs to be resolved as, at some future date, "c" might be retired.
- Resolution does not create static artifact.

Value Set Definition Versions

Value Set Definitions are versioned. The version of Value Set Definition changes when ever the definition is changed, it could be adding or removing Code System reference or changes in the rule set. Value Set authoring services are exposed in CTS 2 implementation of LexEVS. <Link here>

Value Set Services

Visit Value Set Services for detailed Value Set Definition Services documentation.

Value Set GUI

Visit LexEVS 6.0 Value Set GUI for detailed functionality and howto's about using Value Sets developer GUI tool