LexEVS 5.x Design and Architecture Guide 4 - LexEVS Information Models

Contents of this Page

Introduction

This document is a section of the Design and Architecture Guide.

Information Models Overview

The information below is provided for introductory purposes. A full description of all available model components is also available in the javadoc distributed with the LexEVS installation package (see file breakdown in the Installation Guide). Since the javadoc is automatically generated and synchronized during the build process, it is recommended as the primary reference for use by LexEVS developers.

LexGrid Model

The LexGrid model is mastered in XML schema. The LexBIG project currently builds on the 2009 version of the LexGrid schema. A formal representation, showing portions of this structure that are of primary interest to the LexEVS project, is presented below. A complete version of the model is available at http ://informatics.mayo.edu?page=lgm .

CodingSchemes

The CodingSchemes branch of the model defines high level containers for concepts and relations. Each CodingScheme represents a unique code system or version in the LexBIG service. Components of interest include:

codingSchemes
A collection of one or more coding schemes.

codingScheme
A resource that makes assertions about a collection of terminological entities.

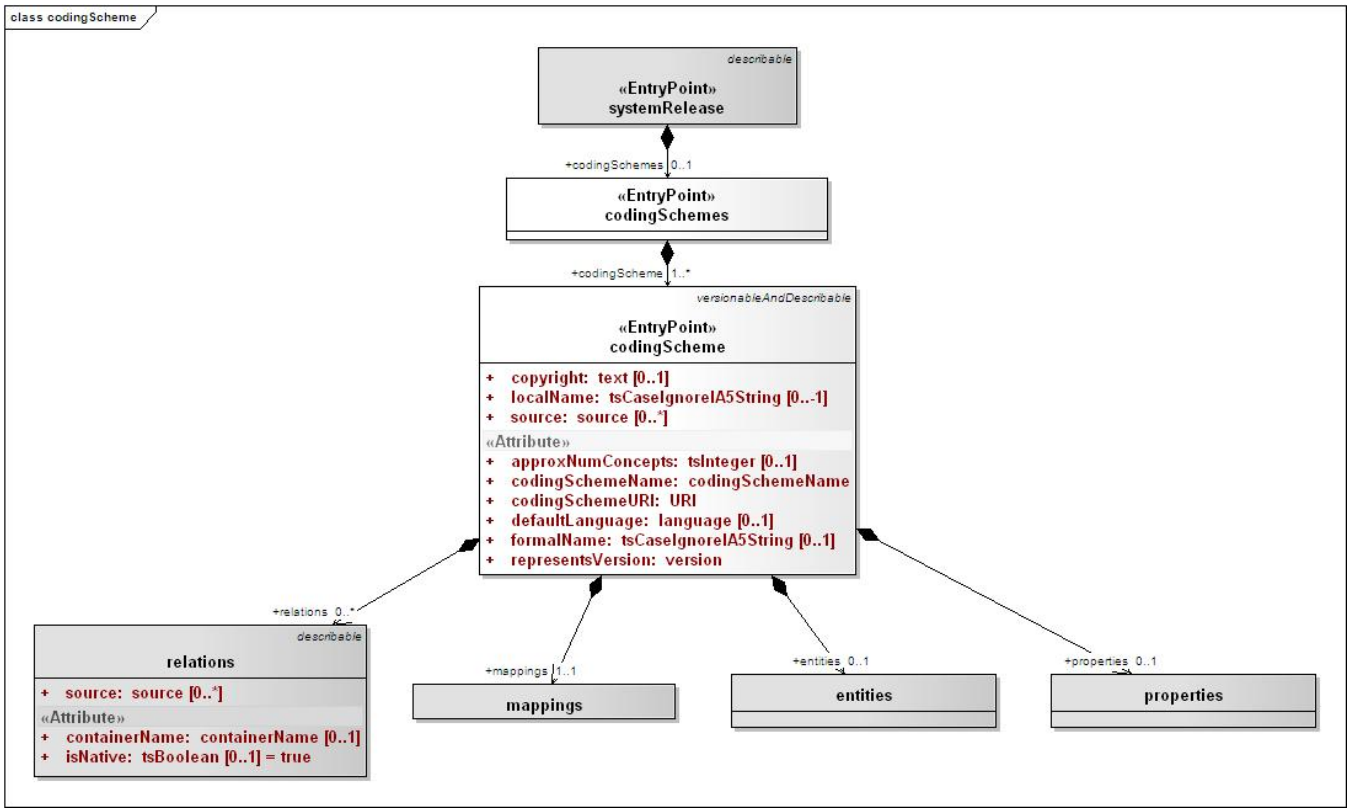
entities
A set of entity codes and their lexical descriptions

relations
A collection of relations that represent a particular point of view or community.

versions
A list of past versions of the coding scheme.

mappings
A list of all of the local identifiers and defining URI's that are used in the associated resource

properties
A collection of properties.



Concepts

Each concept represents a unique entity within the code system, which can be further described by properties and related to other concepts through relations.

conceptsAndInstances

codingScheme
A resource that makes assertions about a collection of terminological entities.

entities
A set of entity codes and their lexical descriptions

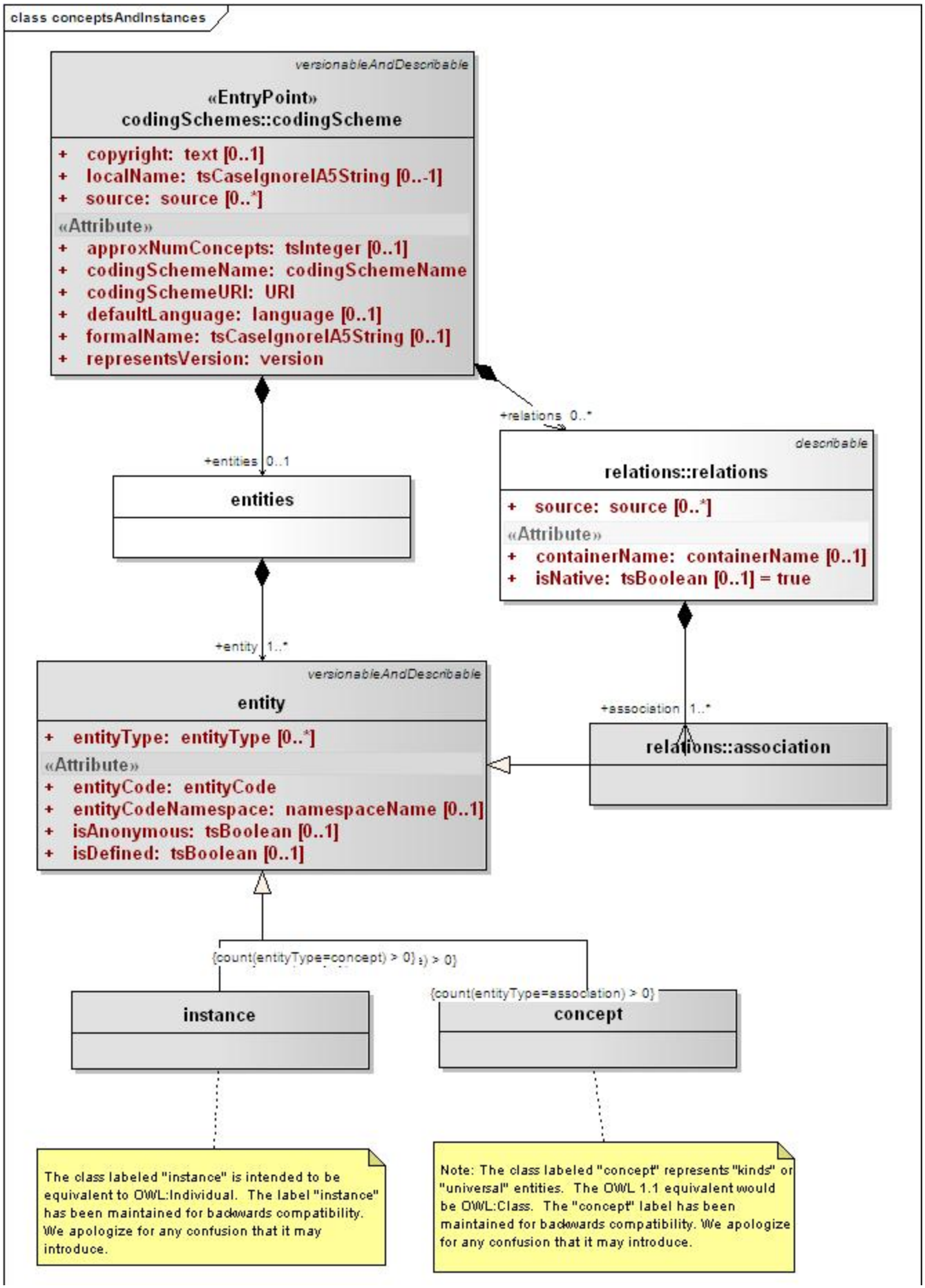
entity
A set of lexical assertions about the intended meaning of a particular entity code.

concept
An entity that represents a class or category. The entityType for the class concept must be "concept".

instance
An entity that represents an instance or an individual. The entityType for the class concept must be "instance".

relations
A collection of relations that represent a particular point of view or community.

association
A binary relation from a set of entities to a set of entities and/or data. The entityType for the class concept must be "association".


class conceptsAndInstances
versionableAndDescribable
«EntryPoint»
codingSchemes::codingScheme
+ copyright: text [0..1]
+ localName: tsCaseIgnoreIA5String [0..1]
+ source: source [0..*]
«Attribute»
+ approxNumConcepts: tsInteger [0..1]
+ codingSchemeName: codingSchemeName
+ codingSchemeURI: URI
+ defaultLanguage: language [0..1]
+ formalName: tsCaseIgnoreIA5String [0..1]
+ representsVersion: version
+relations 0..*
+entities 0..1
describable
relations::relations
+ source: source [0..*]
«Attribute»
+ containerName: containerName [0..1]
+ isNative: tsBoolean [0..1] = true
entities
+entity 1..*
versionableAndDescribable
entity
+ entityType: entityType [0..*]
«Attribute»
+ entityCode: entityCode
+ entityCodeNamespace: namespaceName [0..1]
+ isAnonymous: tsBoolean [0..1]
+ isDefined: tsBoolean [0..1]
+association 1..*
relations::association
{count(entityType=concept) > 0}
{count(entityType=association) > 0}
instance
concept
The class labeled "instance" is intended to be equivalent to OWL:Individual. The label "instance" has been maintained for backwards compatibility. We apologize for any confusion that it may introduce.
Note: The class labeled "concept" represents "kinds" or "universal" entities. The OWL 1.1 equivalent would be OWL:Class. The "concept" label has been maintained for backwards compatibility. We apologize for any confusion that it may introduce.

entities

codingScheme
A resource that makes assertions about a collection of terminological entities.

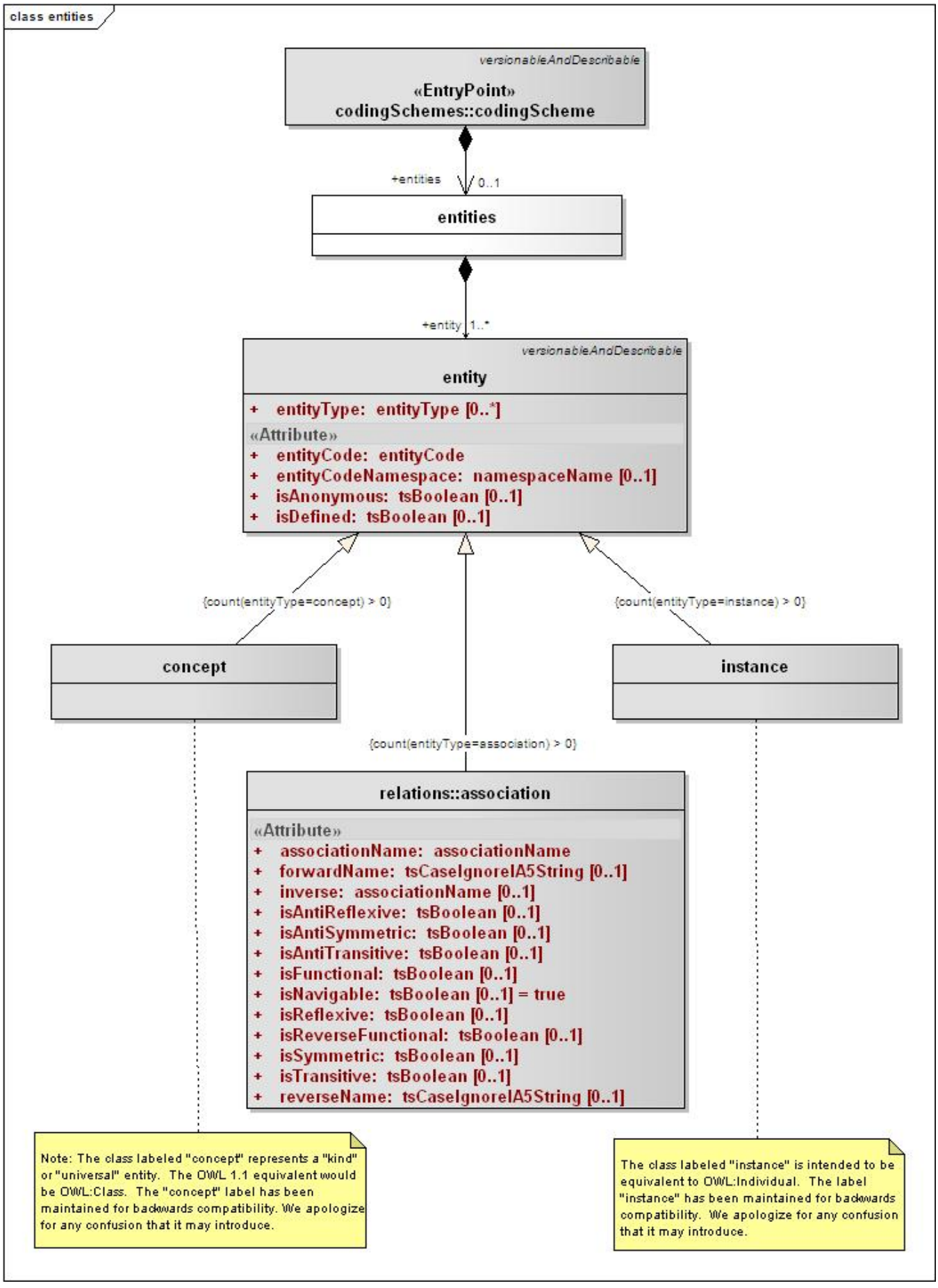
entities
A set of entity codes and their lexical descriptions

entity
A set of lexical assertions about the intended meaning of a particular entity code.

concept
An entity that represents a class or category. The entityType for the class concept must be "concept".

instance
An entity that represents an instance or an individual. The entityType for the class concept must be "instance".

association
A binary relation from a set of entities to a set of entities and/or data. The entityType for the class concept must be "association".


class entities
versionableAndDescribable
«EntryPoint»
codingSchemes::codingScheme
+entities
0..1
entities
+entity
1..*
versionableAndDescribable
entity
+ entityType: entityType [0..*]
«Attribute»
+ entityCode: entityCode
+ entityCodeNamespace: namespaceName [0..1]
+ isAnonymous: tsBoolean [0..1]
+ isDefined: tsBoolean [0..1]
{count(entityType=concept) > 0}
{count(entityType=instance) > 0}
concept
instance
{count(entityType=association) > 0}
relations::association
«Attribute»
+ associationName: associationName
+ forwardName: tsCaseIgnoreIA5String [0..1]
+ inverse: associationName [0..1]
+ isAntiReflexive: tsBoolean [0..1]
+ isAntiSymmetric: tsBoolean [0..1]
+ isAntiTransitive: tsBoolean [0..1]
+ isFunctional: tsBoolean [0..1]
+ isNavigable: tsBoolean [0..1] = true
+ isReflexive: tsBoolean [0..1]
+ isReverseFunctional: tsBoolean [0..1]
+ isSymmetric: tsBoolean [0..1]
+ isTransitive: tsBoolean [0..1]
+ reverseName: tsCaseIgnoreIA5String [0..1]
Note: The class labeled "concept" represents a "kind" or "universal" entity. The OWL 1.1 equivalent would be OWL:Class. The "concept" label has been maintained for backwards compatibility. We apologize for any confusion that it may introduce.
The class labeled "instance" is intended to be equivalent to OWL:Individual. The label "instance" has been maintained for backwards compatibility. We apologize for any confusion that it may introduce.

entity

entity
A set of lexical assertions about the intended meaning of a particular entity code.

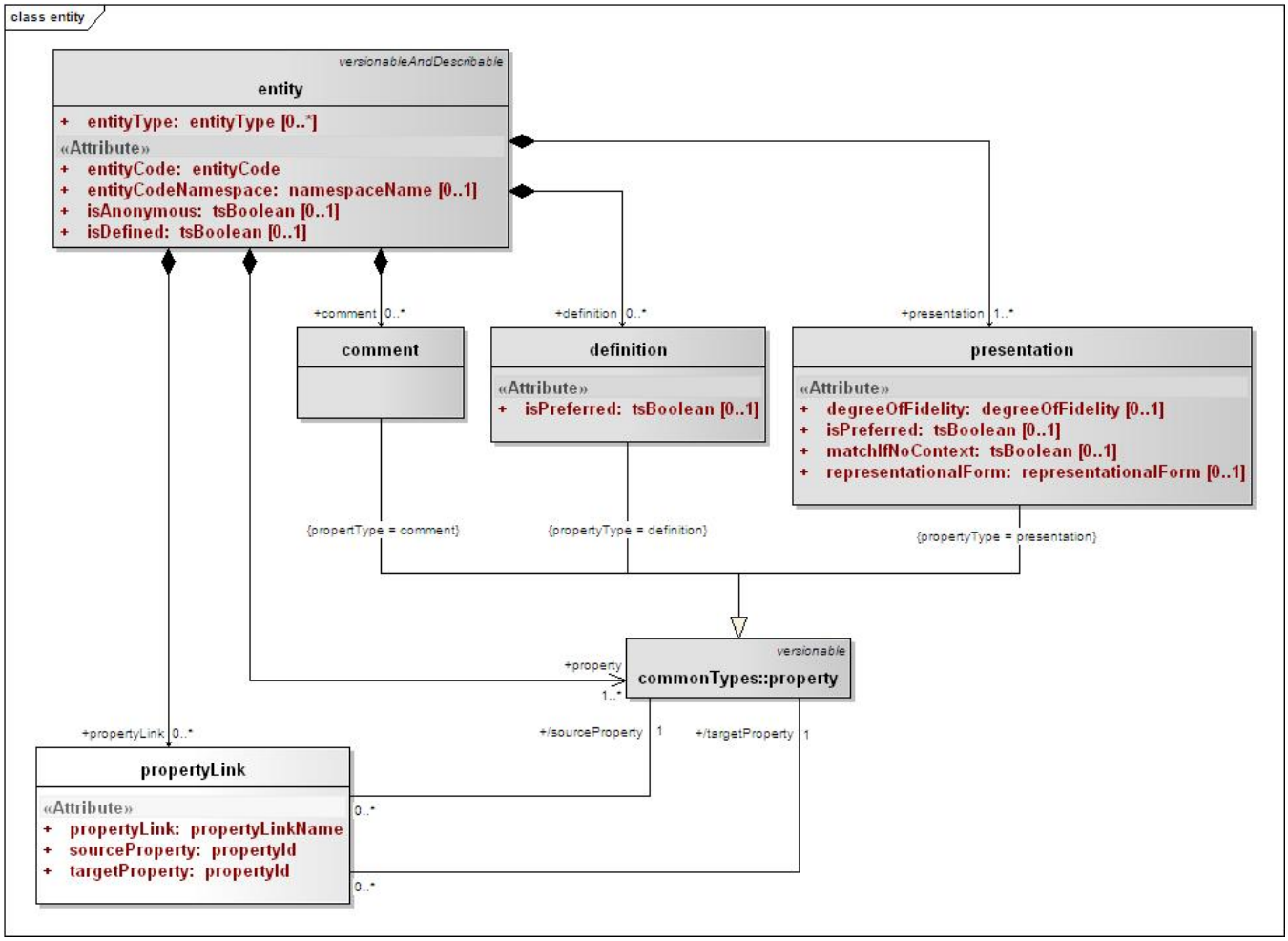
comment
A property that is used as an annotation or other note about the state or usage of the entity. The propertyType of comment must be "comment"

definition
A property that defines the entity in a particular langage or context.. The propertyType of definition must be "definition"

presentation
A property ths represents or designates the meaning of the entityCode. The propertyType of presentation must be "presentation"

property
A description, definition, annotation or other attribute that serves to further define or identify an resource.

propertyLink
A link between two properties for an entity.. Examples include acronymFor, abbreviationOf, spellingVariantOf, etc. Must be in supportedPropertyLink.



Relations

Relations are used to define and qualify associations between concepts.

association

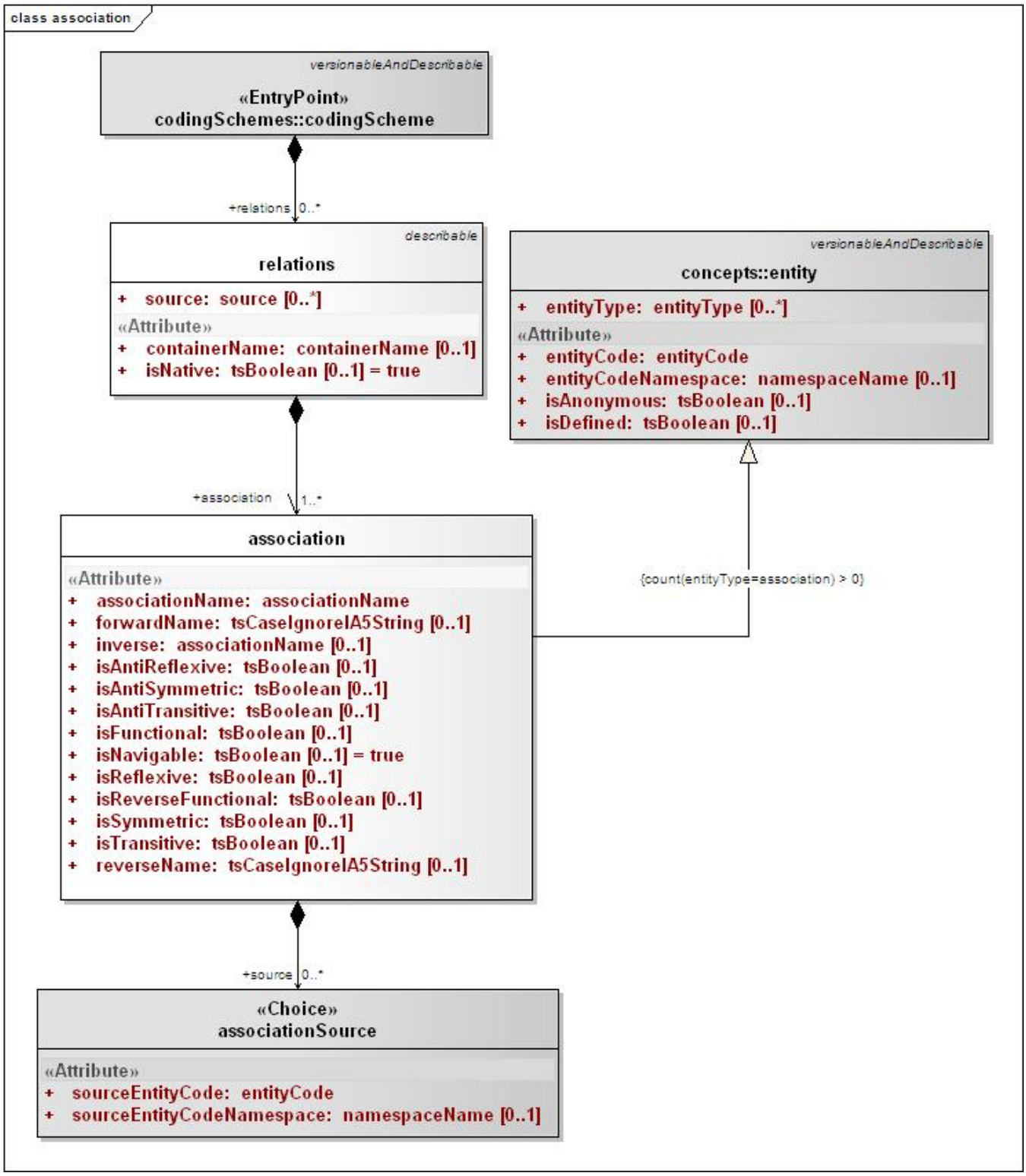
codingScheme
A resource that makes assertions about a collection of terminological entities.

relations
A collection of relations that represent a particular point of view or community.

entity
A set of lexical assertions about the intended meaning of a particular entity code.

association
A binary relation from a set of entities to a set of entities and/or data. The entityType for the class concept must be "association".

associationSource
An entity that occurs in one or more instances of a relation on the "from" (or left hand) side of a particular relation.



associationInstance

association
A binary relation from a set of entities to a set of entities and/or data. The entityType for the class concept must be "association".

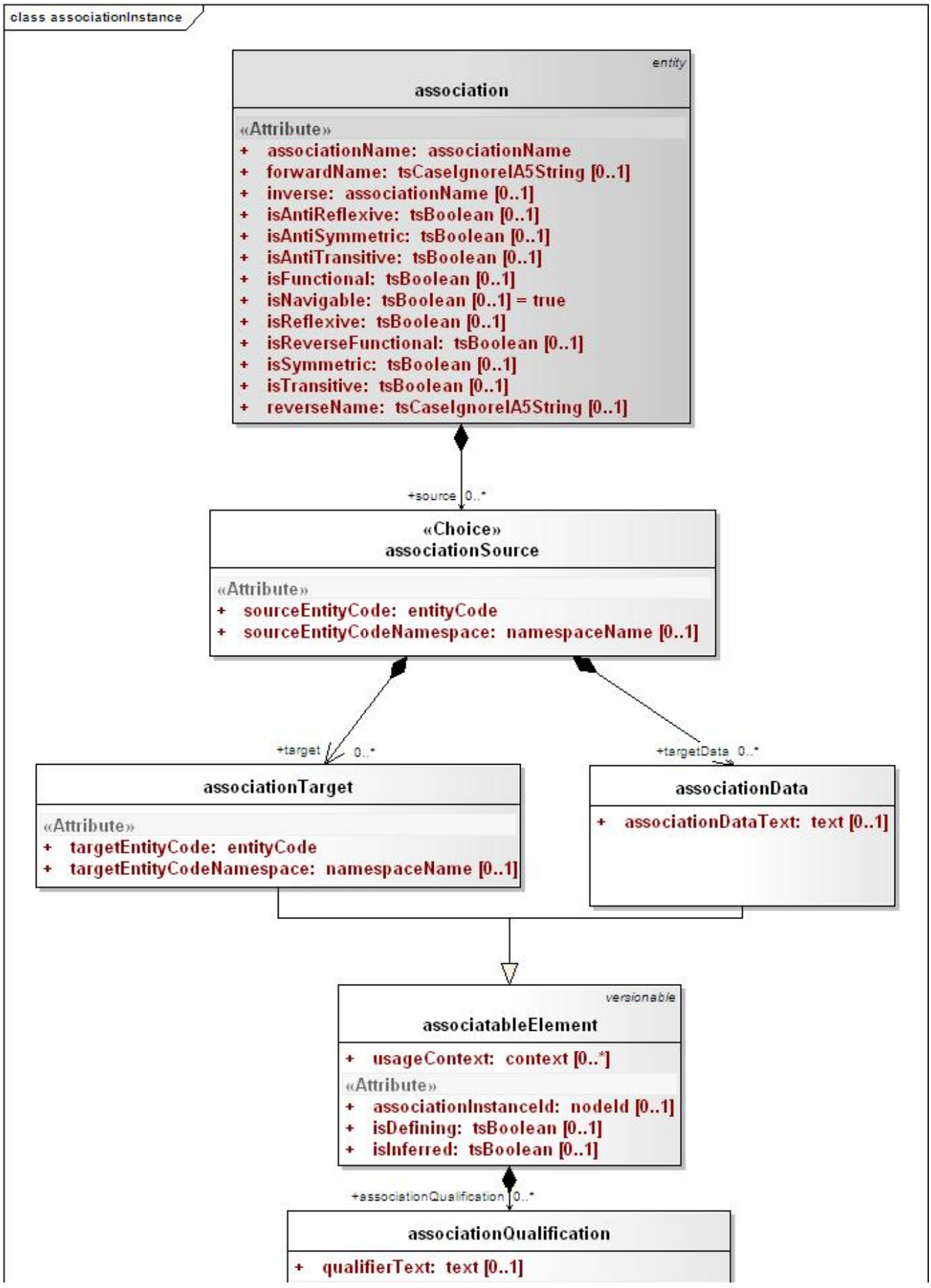
associationSource
An entity that occurs in one or more instances of a relation on the "from" (or left hand) side of a particular relation.

associationTarget
An entity on the "to" (or right hand) side of a relation.

associationData
An instance of a target or RHS data value of an association.

associatableElement
Information common to both the entity and data form of the "to" (or right hand) side of an association.

associationQualification
A modifier that further qualifies the particular association instance.


class associationInstance
entity
association
«Attribute»
+ associationName: associationName
+ forwardName: tsCaseIgnoreIA5String [0..1]
+ inverse: associationName [0..1]
+ isAntiReflexive: tsBoolean [0..1]
+ isAntiSymmetric: tsBoolean [0..1]
+ isAntiTransitive: tsBoolean [0..1]
+ isFunctional: tsBoolean [0..1]
+ isNavigable: tsBoolean [0..1] = true
+ isReflexive: tsBoolean [0..1]
+ isReverseFunctional: tsBoolean [0..1]
+ isSymmetric: tsBoolean [0..1]
+ isTransitive: tsBoolean [0..1]
+ reverseName: tsCaseIgnoreIA5String [0..1]
+source 0..*
«Choice»
associationSource
«Attribute»
+ sourceEntityCode: entityCode
+ sourceEntityCodeNamespace: namespaceName [0..1]
+target 0..*
+targetData 0..*
associationTarget
«Attribute»
+ targetEntityCode: entityCode
+ targetEntityCodeNamespace: namespaceName [0..1]
associationData
+ associationDataText: text [0..1]
versionable
associatableElement
+ usageContext: context [0..*]
«Attribute»
+ associationInstanceId: nodeId [0..1]
+ isDefining: tsBoolean [0..1]
+ isInferred: tsBoolean [0..1]
+associationQualification 0..*
associationQualification
+ qualifierText: text [0..1]

Naming

These elements are primarily used to define metadata for a coding scheme, mapping locally used names to global references.

URIMap
A local identifier that is used in a specific context (e.g. language, property name, data type, etc) and an optional URI that can be used to find the exact definition and meaning of the local id. Note: the string portion of this entry can be used to provide additional documentation or information, especially when a URI is not supplied.

supportedAssociation
An associationName and the URI of the defining resource.

supportedAssociationQualifier
An associationQualifier and the URI of the defining resource

supportedCodingScheme
A codingSchemeName and the URI of the defining resource

supportedStatus
An entryStatus and the URI of the defining resource

supportedEntityType
An entityType and the URI of the defining resource

supportedContext
A context and the URI of the defining resource

supportedContainerName
A containerName and the URI of the defining resource

supportedDegreeOfFidelity
A degreeOfFidelity and the URI of the defining resource

supportedLanguage
A language and the URI of the defining resource

supportedProperty
A propertyName and the URI of the defining resource

supportedSortOrder
The local identifier and the URI of the defining resource

supportedHierarchy
A list of associations that can be browsed hierarchically.

supportedNamespace
A namespaceName and the corresponding URI

supportedPropertyType
A propertyType and the URI of the defining resource

supportedPropertyQualifier
A propertyQualifierName the URI of the defining resource

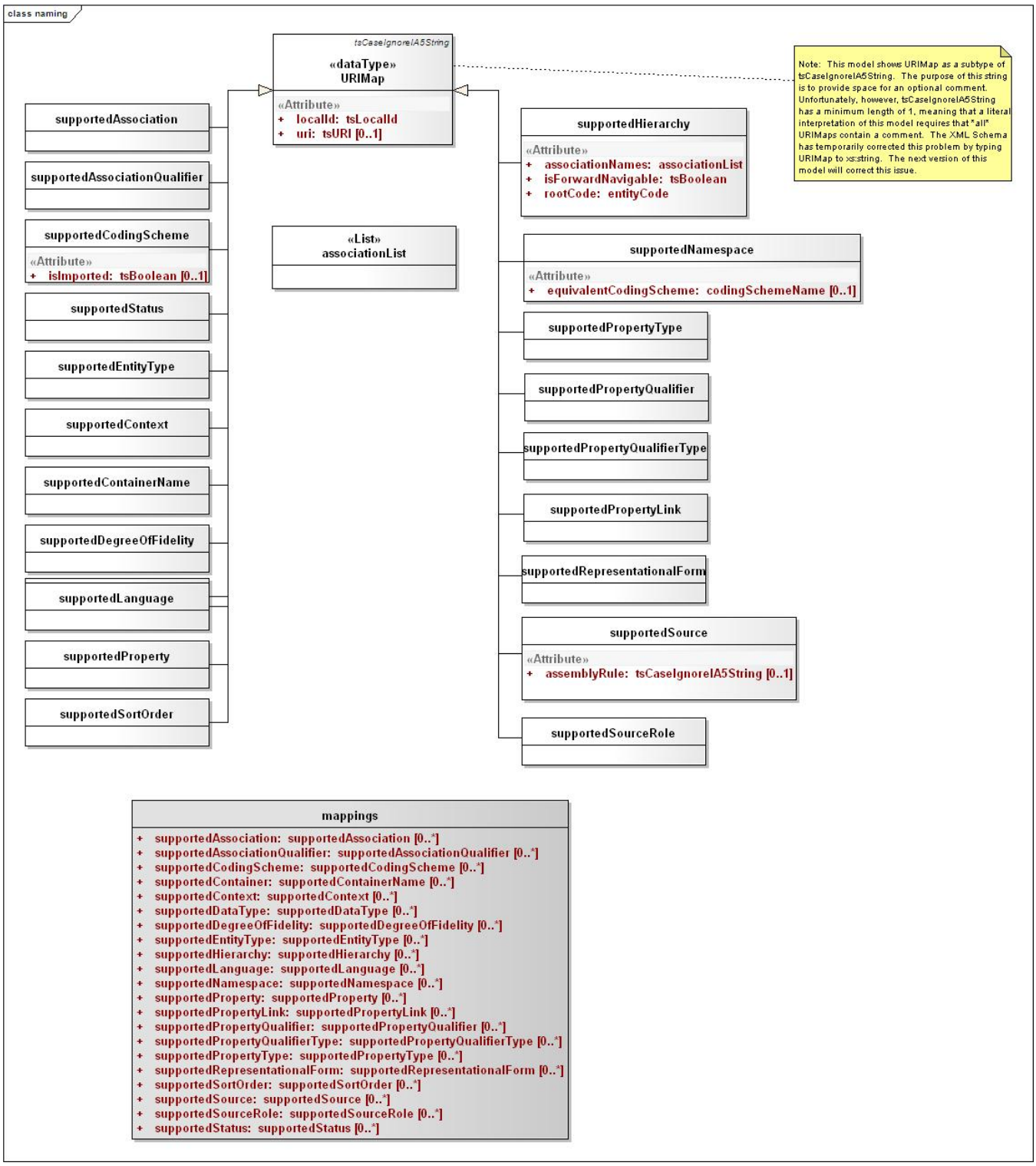
supportedPropertyQualifierType
A propertyQualifierType the URI of the defining resource

supportedPropertyLink
A propertyLinkName and the URI of the defining resource

supportedRepresentationalForm
A representationalForm and the URI of the defining resource

supportedSource
A source and the URI of the defining resource. Source references can also carry an additional compositional rule section that describes how to combine a subpart such as a page number, section name, etc. with the core URI in order to form a meaningful URL. An optional role can also be specified.

supportedSourceRole
A source role and athe URI of the defining resource



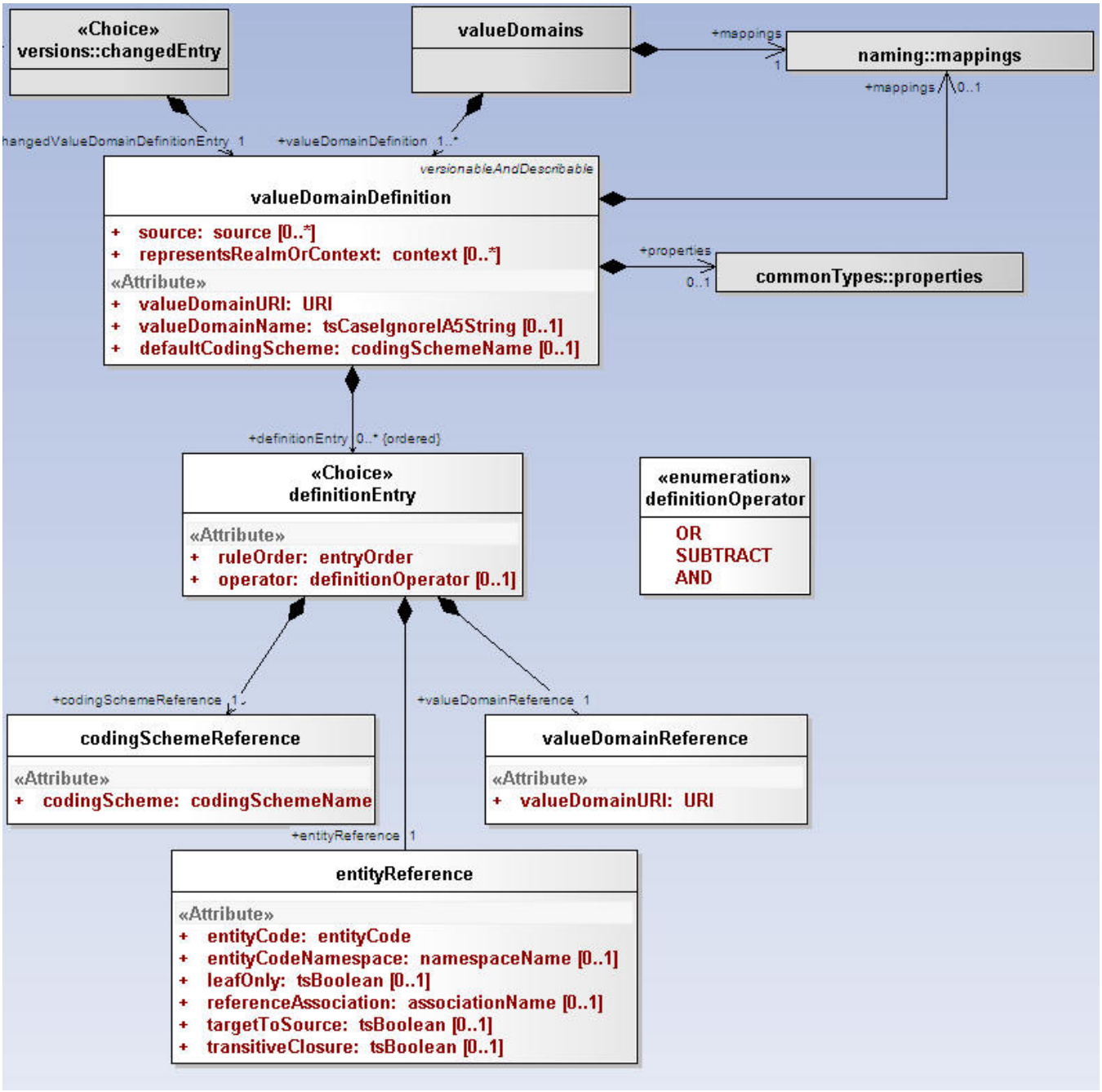
Value Domain Definition

The Value Domain Definition branch of the LexGrid model defines the contents of a value domain.

Value Domain can be defined in following forms:

- **Code System :** *All* concept codes in the referencing code system.
- **Value Domain :** *All* concept codes definied in the referencing Value Domain Definition.
- **Code System + Concept Code :** Individual codes.
- **Code System + Concept Code + relationship + additional rules(leafOnly, targetToSource, transitiveClosure..) :** Only the concept codes that matches all the definited conditions(rules).
- Combination of any of the above with*or/and/difference* operators

The following diagram is a UML representation of Value Domain Definition in LexGrid 200901 model:



Model Components

valueDomains
A collection of value domain definitions.

mappings
A list of all of the local identifiers and defining URI's that are used in the associated value domains.

properties
A collection of value domain properties.

changedEntry
A top level versionable entry.

valueDomainDefinition
A definition of a given value domain. A value domain can be a simple description with no associated value domain 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 Domain 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 domain applies. Must match a local id of a supportedContext in the corresponding mappings section.
- *valueDomainURI:* The URI of this value domain.
- *valueDomainName:* The name of this domain, if any.
- *defaultCodingScheme:* Local name of the primary coding scheme from which the domain is drawn. defaultCodingScheme must match a local id of a supportedCodingScheme in the mappings section.

definitionEntry
A reference to an entry code, a coding scheme or another value domain along with the instructions about how the reference is applied. Definition entrys are applied in entryOrder, with each successive entry either adding to or subtracting from the final set of entity codes.

Attributes of Value Domain Definition Entry:

- *ruleOrder:* The 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 domain.

codingSchemeReference
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 from which the entity codes are drawn. codingSchemeName must match a local ID of a supportedCodingScheme in the mappings section.

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

Attributes of Value Domain Reference:

- *valueDomainURI:* The URI of the value domain to which to apply the operator. This value domain may be contained within the local service or may need to be resolved externally.

entityReference
A reference to an entityCode and/or one or more entityCodes that have a relationship to the specified entity code plus the rules(leafOnly, targetToSource..) to be applied.

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. 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.

definitionOperator
The description of how a given definition entry is applied.

Attributes of Definition Operator:

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

Value Domain Services

This feature is new in LexEVS 5.1. For details on using value domain services, see the Programmer's Guide, Value Domain Services section.

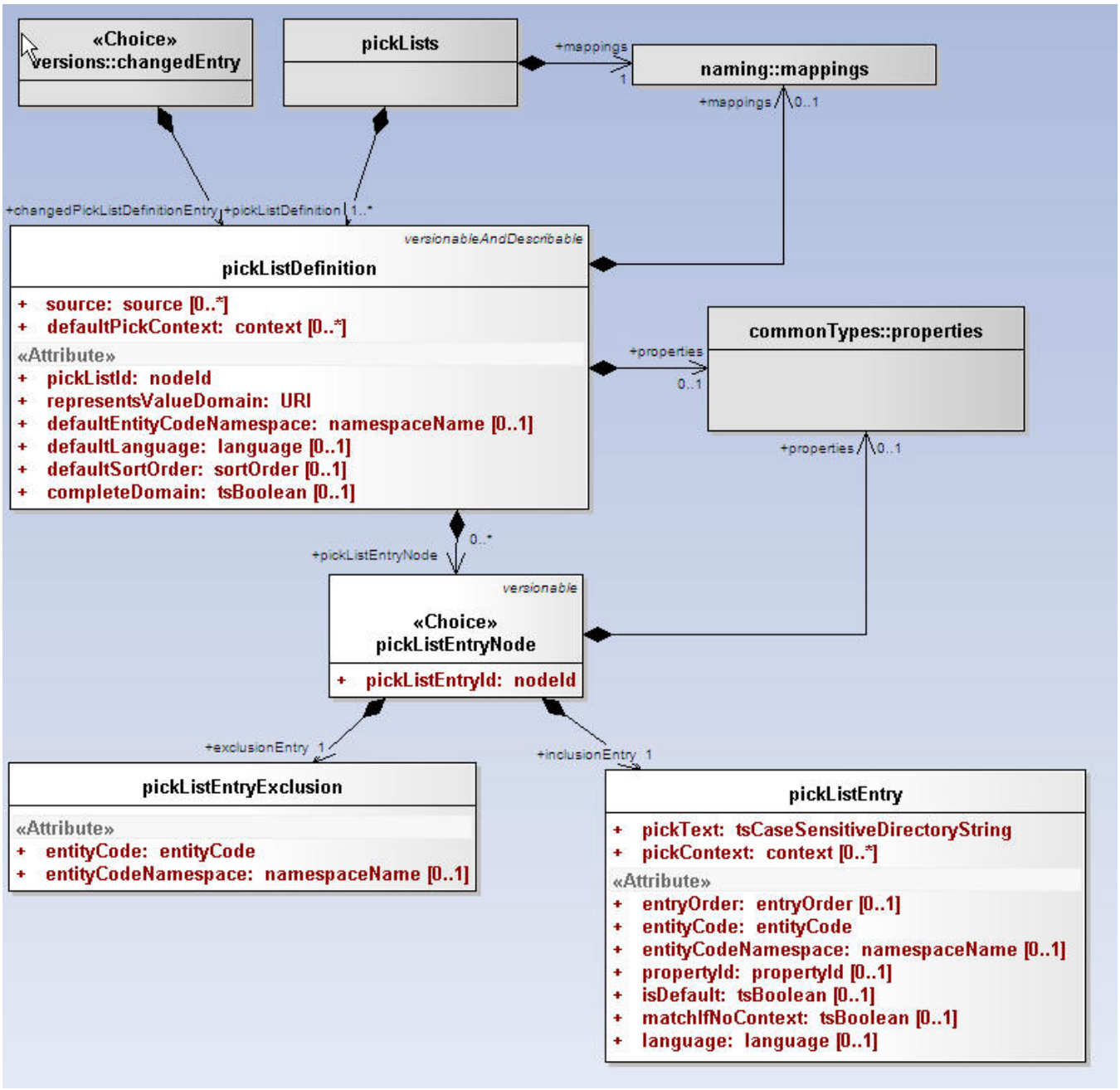
Pick List Definition

Pick List Definition branch of LexGrid model definies the contents of a pick list.

Pick List can be defined in following forms:

- **Value Domain:** *all* concept codes definied in referenced value domain
- **Code System + Concept Code:** individual codes (inclusion and exclusion)

The following diagram is a UML representation of Pick List in LexGrid 200901 model:



Model Components

pickLists
A collection of pick list definitions.

mappings
A list of all of the local identifiers and defining URI's that are used in the associated pick list definitions.

properties
A collection of properties.

changedEntry
A top level versionable entry.

pickListDefinition
An ordered list of entity codes and corresponding presentations drawn from a value domain.

Attributes of Pick List Definition:

- *Source:* The local identifiers of the source(s) of this pick list definition. Must match a local ID of a supportedSource in the corresponding mappings section.

- *pickListId:* An identifier that uniquely names this list within the context of the collection.
- *representsValueDomain:* The URI of the value domain definition that is represented by this pick list.
- *defaultEntityCodeNamespace:* Local name of the namespace to which the entry codes in this list belong. defaultEntityCodeNamespace must match a local ID of a supportedNamespace in the mappings section.
- *defaultLanguage:* The local identifier of the language that is used to generate the text of this pick list if not otherwise specified. Note that this language does NOT necessarily have any coorelation with the language of a pickListEntry itself or the language of the target user. defaultLanguage must match a local ID of a supportedLanguage in the mappings section.
- *defaultSortOrder:* The local identifier of a sort order that is used as the default in the definition of the pick list.
- *defaultPickContext:* The local identifiers of the context used in the definition of the pick list.
- *completeDomain:* True means that this pick list should represent all of the entries in the domain. Any active entity codes that are not in the specific pick list entries are added to the end, using the designations identified by the defaultLanguage, defaultSortOrder, and defaultPickContext; default=false.

pickListEntryNode
An inclusion (pickListEntry) or exclusion (pickListEntryExclusion) in a pick list definition

Attributes of Pick List Entry Node:

- *pickListEntryId:* Unique identifier of this node within the list.

pickListEntry
An entity code and corresponding textual representation.

Attributes of Pick List Entry:

- *pickText:* The text that represents this node in the pick list. Some business rules may require that this string match a presentation associated with the entityCode.
- *pickContext:* The local identifiers of the context(s) in which this entry applies. pickContext must match a local ID of a supportedContext in the mappings section.
- *entryOrder:* Relative order of this entry in the list. pickListEntries without a supplied order follow the all entries with an order, and the order is not defined.
- *entityCode:* Entity code associated with this entry.
- *entityCodeNamespace:* Local identifier of the namespace of the entity code if different than the pickListDefinition defaultEntityCodeNamespace. entityCodeNamespace must match a local ID of a supportedNamespace in the mappings section.
- *propertyId:* The property identifier associated with the entityCode and entityCodeNamespace from which the pickText was derived. If absent, the pick text can be anything. Some terminologies may have business rules requiring this attribute to be present.
- *isDefault:* True means that this is the default entry for the supplied language and context.
- *matchIfNoContext:* True means that this entry can be used if no contexts are supplied, even though pickContext is present.
- *Language:* The local name of the language to be used when the application/user supplies a selection language matches. If absent, this matches all languages. language must match a local ID of a supportedLanguage in the mappings section.

pickListEntryExclusion
An entity code that is explicitly excluded from a pick list.

Attributes of Pick List Entry Exclusion:

- *entityCode:* Entity code associated with this entry.
- *entityCodeNamespace:* Local identifier of the namespace of the entity code if different than the pickListDefinition defaultEntityCodeNamespace. entityCodeNamespace must match a local ID of a supportedNamespace in the mappings section.

Pick List Services

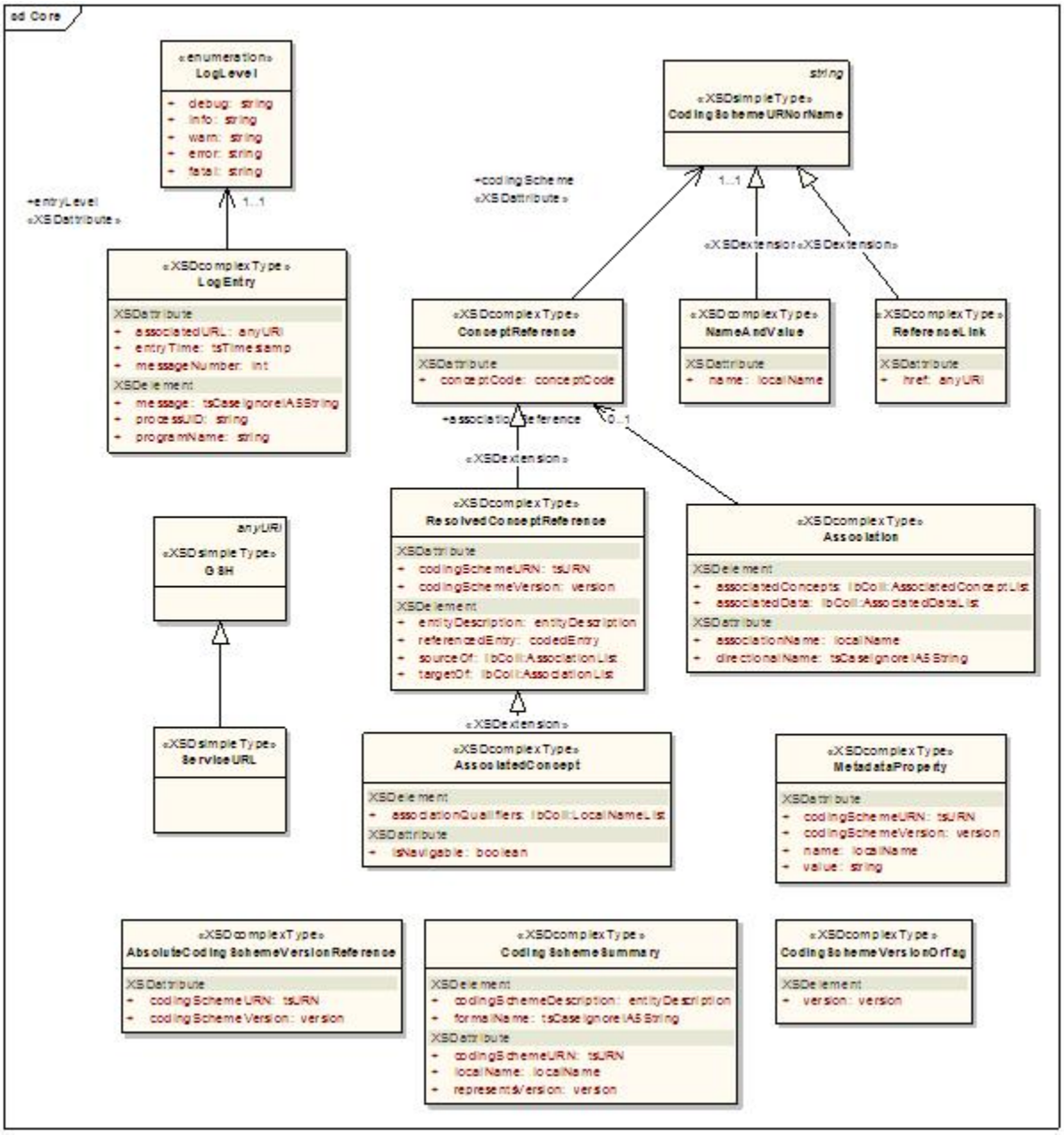
This feature is new in LexEVS 5.1. For details on using pick list services, see the Programmer's Guide, Pick List Services section.

LexBIG Model

The following extensions to the LexGrid model were introduced in support of caBIG requirements. As with the LexGrid model, this document provides a summary of the most significant elements for consideration by LexBIG programmers. The complete and current version of the model is available online at http://informatics.mayo.edu?page=lexex .

Core

LexBIG core elements provide enhanced referencing and controlled resolution of LexGrid model objects.



Core components of interest include:

AbsoluteCodingSchemeVersionReference
An absolute reference to a coding scheme. This form of reference is service independent, as it doesn't depend on local coding schemes names or virtual tags.

AssociatedConcept
A concept reference that is the source or target of an association.

Association
The representation of a particular association as it appears in a CodedNode.

CodingSchemeSummary
Abbreviated list of information about a coding scheme.

CodingSchemeURNorName
Either a local name or the URN of a coding scheme. These two are differentiated syntactically - if the entity includes a colon, :, or a hash "#" it is assumed to be a URN. Otherwise it is assumed to be a local name.

CodingSchemeVersionOrTag
A named coding scheme version or a virtual tag (e.g. latest, production, etc). Note that the tagged form of identifier is only applicable in the context of a given service, as one service may identify the scheme as "production" and another as "staging".

ConceptReference
A reference to a coding scheme and a concept code.

LogEntry
A single recorded log entry.

LogLevel
Indicates severity of the log entry.

MetadataProperty
Reference to a property name and value stored in the coding scheme metadata.

NameAndValue
A simple name/value pair.

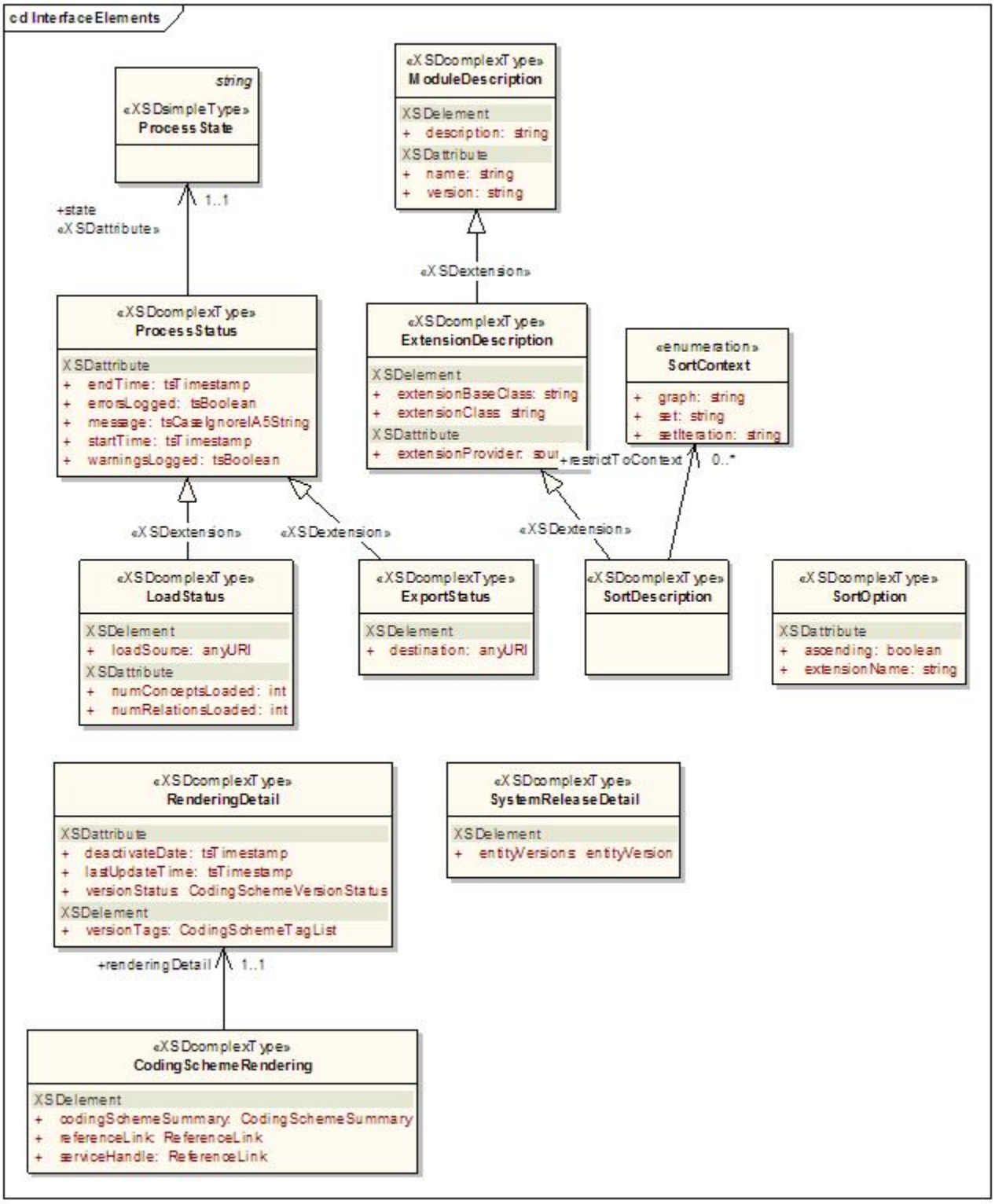
ReferenceLink
Any reference to another document element. Used by the REST architecture to embed links.

ResolvedConceptReference
A resolvable concept reference.

ServiceURL
References a service in the Globus environment, this will be a global service handle (GSH).

InterfaceElements

Defines metadata related to model objects required by the runtime.



InterfaceElements components of interest include:

CodingSchemeRendering
Information about a coding scheme as it appears in a particular service.

ExportStatus
Reports the state of LexBIG export operations.

ExtensionDescription
Describes an add-on module registered to the LexBIG environment.

LoadStatus
Reports the state of LexBIG load operations.

*ModuleDescription *
Describes a LexBIG integrated software module.

ProcessState
Enumerates possible status reported for LexBIG runtime operations.

ProcessStatus
Reports the state of LexBIG runtime operations.

RenderingDetail
The details of how a coding scheme is rendered in a given service.

SortContext
Describes a LexBIG sort module.

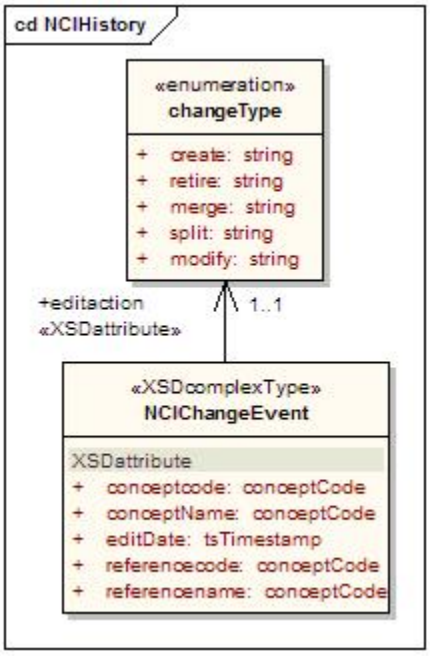
SortDescription
A description of a LexBIG extension module.

SortOption
Represents a pairing of sort algorithm and order.

SystemReleaseDetail
The combination of a system release and all of the entityVersions that accompanied that release.

NCIHistory

Maintains a record of modifications made to a code system.



NCIHistory components of interest include:

changeType
Atomic modification actions. Currently populated from a combination of Concordia, SNOMED-CT list and NCI's action list.

NCIChangeEvent
A change event as documented in:

```
Link provided for historical purposes ftp://ftp1.nci.nih.gov/pub/cacore/EVS/ReadMe_history.txt
```

Note that date and time of the change event is recorded in the containing version. All change events for the same/date and time a recorded in the same version.