



CTS2 Concept Domain and Concept Domain Binding Services

Version 1.1

OMG Document Number: formal/2013-05-08
Standard document URL: <http://www.omg.org/spec/cts2/1.1/>

Copyright © 2013, Mayo Clinic
Copyright © 2013, Object Management Group

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT

LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE.

IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 109 Highland Avenue, Needham, MA 02494, U.S.A.

TRADEMARKS

MDA®, Model Driven Architecture®, UML®, UML Cube logo®, OMG Logo®, CORBA® and XMI® are registered trademarks of the Object Management Group, Inc., and Object Management Group™, OMG™, Unified Modeling Language™, Model Driven Architecture Logo™, Model Driven Architecture Diagram™, CORBA logos™, XMI Logo™, CWM™, CWM Logo™, IIOP™, IMM™, MOF™, OMG Interface Definition Language (OMG IDL)™, and OMG Systems Modeling Language (OMG SysML)™ are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

Table of Contents

1. Introduction	1
1.1 Notation	1
1.2 Identifiers	1
1.3 Stereotypes	1
1.4 Read Only Attributes	2
2. Concept Domain Information Model	3
2.1 Concept Domain Catalog Entry	3
2.1.1 Class ConceptDomainCatalogEntry	3
2.2 Concept Domain Directory and List	4
2.2.1 Class ConceptDomainCatalogEntryDirectory	4
2.2.2 Class ConceptDomainCatalogEntryList	5
2.2.3 Class ConceptDomainCatalogEntrySummary	5
2.2.4 Class ConceptDomainCatalogListEntry	5
3. Concept Domain Catalog Services	7
3.1 Concept Domain Catalog Read Service	7
3.1.1 Interface ConceptDomainCatalogReadService	7
3.2 Concept Domain Catalog Query Service	10
3.2.1 Interface ConceptDomainCatalogQueryService	10
3.3 Concept Domain Catalog History Service	13
3.3.1 Interface ConceptDomainHistoryService	13
3.4 Concept Domain Catalog Maintenance Service	15
3.4.1 Interface ConceptDomainCatalogMaintenanceService	15
3.4.2 Class UpdateConceptDomainCatalogEntryRequest	17
4. Concept Domain Binding Information Model	19
4.1 Concept Domain Binding	19
4.1.1 Class ConceptDomainBinding	19
4.2 Concept Domain Binding Directory	20
4.2.1 Class ConceptDomainBindingDirectory	20
4.2.2 Class ConceptDomainBindingDirectoryEntry	20
4.2.3 Class ConceptDomainBindingList	21
4.2.4 Class ConceptDomainBindingListEntry	21
5. Concept Domain Binding Services	23

5.1 Concept Domain Binding Read Services	23
5.1.1 Interface ConceptDomainBindingReadService	23
5.2 Concept Domain Binding Query Services	26
5.2.1 Interface ConceptDomainBindingQueryService	26
5.3 Concept Domain Binding Maintenance Services	30
5.3.1 Interface ConceptDomainBindingMaintenanceService	30
5.3.2 Class UpdateConceptDomainBindingRequest	32

1 Introduction

The CTS2 specification is divided into a number of separably implementable profiles. This document on Concept Domain Services describes the relationship between enumerated data elements as described in models, schemas and databases, and value sets, which represent the set of possible value meanings that can be coded in the data element. This model focuses on sets of meanings and does not directly speak to how a given meaning might be represented in an instance of the described model, although some obvious choices could include the entity name, the entity namespace and name as a tuple or the entity URI.

1.1 Notation

Model elements are referenced using `Typewriter` font.

1.2 Identifiers

- Class names are capitalized camel case (`CodeSystemCatalog`, `String`).
- Attribute and role names are lower camel case (`version`, `sourceAndRole`).
- Enumeration values are all capital letters with underscores ('_') introduced as needed for clarity. (`EMPTY`, `STOP_ON_ERROR`).
- Method names are `lowerCamelCase`.

1.3 Stereotypes

The following stereotypes are used throughout the model:

- `dataType` - UML `dataType` semantics.
- `enumeration` - UML enumeration semantics.
- `interface` - UML interface semantics.
- `mixin` - a class that provides a certain functionality to be inherited by a subclass, while not meant for instantiation.
- `opt` - an optional parameter in a method call. This stereotype should always be accompanied by a 0..1 cardinality and is provided to make optionality visible in the modeling tool.
- `optparam` - an optional attribute in a class instance. This indicates that an implementation must be able to distinguish three alternatives: (1) add or change the attribute to whatever is in the class, (2) remove the attribute if its minimum cardinality is zero, and (3) leave the attribute unchanged.
- `delta` - a method call that changes the state of the service. Note that the semantics of this stereotype is the inverse of the `is query model attribute`.
- `exception` - an exception.
- `exceptionSet` - a set of exceptions, the members of which are represented by aggregation.

1.4 Read Only Attributes

The CTS2 PIM is based on a RESTful architectural style. Resources can be created, updated, read, and/or deleted, and every resource has one or more immutable characteristics - characteristics that, if changed, would change the identity of the resource itself. In the information model we distinguish these characteristics from those that can be modified by declaring the identifying characteristics as *ReadOnly*. The computational model utilizes a pattern based on this information where the *create* operation(s) supply the identifying characteristics plus any of the non-optional mutable characteristics. The *update* operations then supply a unique identifier and a set of one or more mutable characteristics to be changed.

2 Concept Domain Information Model

2.1 Concept Domain Catalog Entry

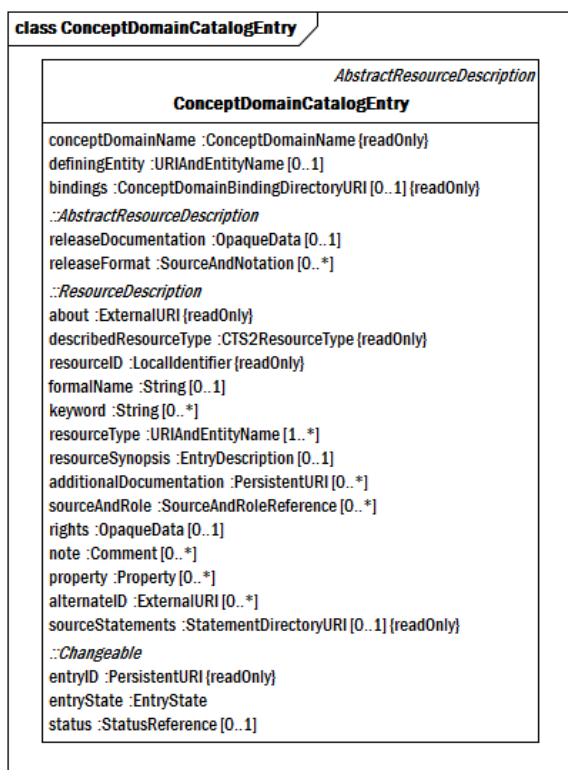


Figure 2.1 - Concept Domain Catalog Entry

2.1.1 Class ConceptDomainCatalogEntry

Metadata about what, in ISO-1 1179, is called a “Data Element Concept” - “a CONCEPT that can be expressed in the form of a DATA ELEMENT, described independently of any particular representation.”

The HL7 SFM defines a `ConceptDomain` as “A named category of like concepts (a semantic type) that will be bound to one or more attributes in a static model whose datatypes are coded. Concept Domains exist to constrain the intent of the coded element while deferring the association of the element to a specific coded terminology until later in the model development process. Thus, Concept Domains are independent of any specific vocabulary or code system. Concept Domains represents an abstract conceptual space such as ‘countries of the world,’ ‘the gender of a person used for administrative purposes,’ ‘languages of the world,’ etc.”

Superclasses

- Every instance of `ConceptDomainCatalogEntry` is also an instance of `AbstractResourceDescription`.

Attributes

- **conceptDomainName** - the name by which the concept domain is known within the context of the *CTS₂* service instance. `conceptDomainName` must uniquely designate a single, abstract `ConceptDomain` within the context of all concept domains known to the instance of the service. Note that `conceptDomainName` is not necessarily globally unique and the `about` URI should be used whenever communication information about the `ConceptDomain` across time or service instances.
- **definingEntity** - the name and URI of an `EntityDescription` that describes the domain and scope of meaning for the `ConceptDomain`. If present, this association is equivalent to the data element concept domain association in section 11.2.3.2 of ISO 11179-3. `entityReference` provides a definition for the `ConceptDomain` and potentially describes its relationship with other `ConceptDomains` as well as providing a semantic link to the possible value meanings available to domain bindings.
- **bindings** - a URI that resolves to the known bindings (aka. Data Elements) of the referenced `ConceptDomain`.

2.2 Concept Domain Directory and List

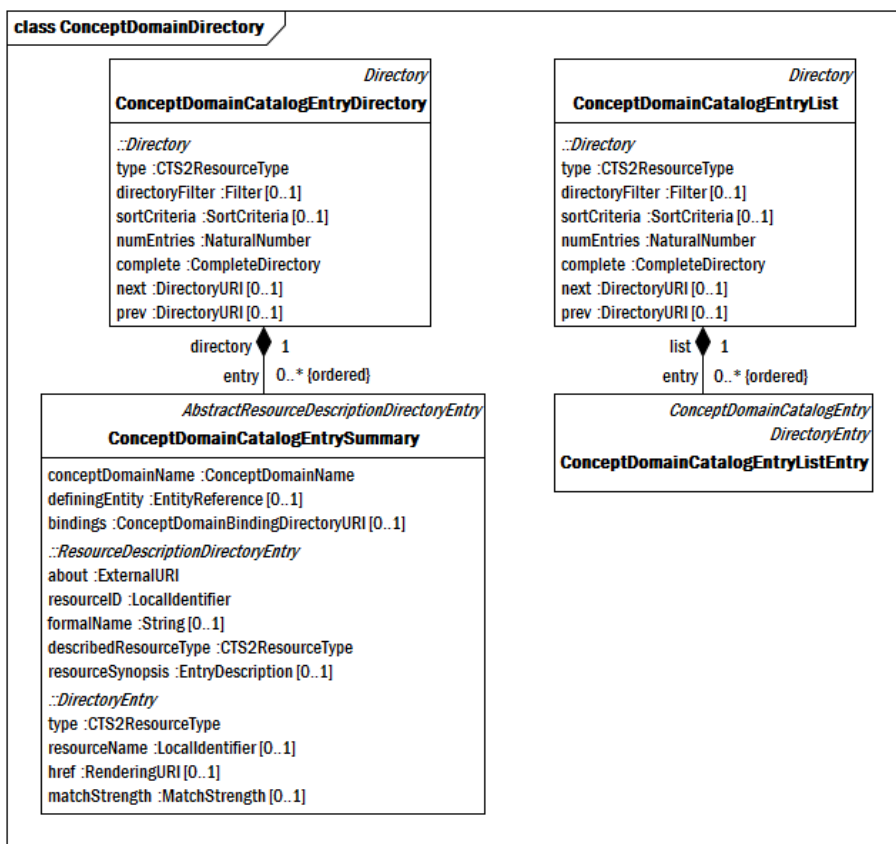


Figure 2.2 - Concept Domain Directory

2.2.1 Class ConceptDomainCatalogEntryDirectory

A directory of `ConceptDomain` resources that meet a specified criteria.

Superclasses

- Every instance of `ConceptDomainCatalogEntryDirectory` is also an instance of `Directory`.

Attributes

- **entry** - an entry in the directory.

2.2.2 Class `ConceptDomainCatalogEntryList`

A collection of complete `ConceptDomain` resources that meet a specified criteria.

Superclasses

- Every instance of `ConceptDomainCatalogEntryList` is also an instance of `Directory`.

Attributes

- **entry** - an entry in the list

2.2.3 Class `ConceptDomainCatalogEntrySummary`

A synopsis of a `ConceptDomain` along with information about how to access the complete resource.

Superclasses

- Every instance of `ConceptDomainCatalogEntrySummary` is also an instance of `AbstractResourceDescriptionDirectoryEntry`.

Attributes

- **conceptDomainName** - the name by which the concept domain is known within the context of the CTS2 service instance. `conceptDomainName` must uniquely designate a single, abstract `ConceptDomain` within the context of all concept domains known to the instance of the service. Note that `conceptDomainName` is not necessarily globally unique and the about URI should be used whenever communication information about the `ConceptDomain` across time or service instances.
- **definingEntity** - a reference to an `EntityDescription` that describes the domain and scope of meaning for the `ConceptDomain`. If present, this association is equivalent to the data element concept domain association in section 11.2.3.2 of ISO 11179-3. `entityReference` provides a definition for the `ConceptDomain` and potentially describes its relationship with other `ConceptDomains` as well as providing a semantic link to the possible value meanings available to domain bindings.
- **bindings** - a URI that resolves to the known bindings (aka. Data Elements) of the referenced `ConceptDomain`.

2.2.4 Class `ConceptDomainCatalogListEntry`

An instance of `ConceptDomain` that meets a specified filter criteria.

Superclasses

- Every instance of `ConceptDomainCatalogListEntry` is also an instance of `ConceptDomainCatalogEntry`.
- Every instance of `ConceptDomainCatalogListEntry` is also an instance of `DirectoryEntry`.

3 Concept Domain Catalog Services

3.1 Concept Domain Catalog Read Service

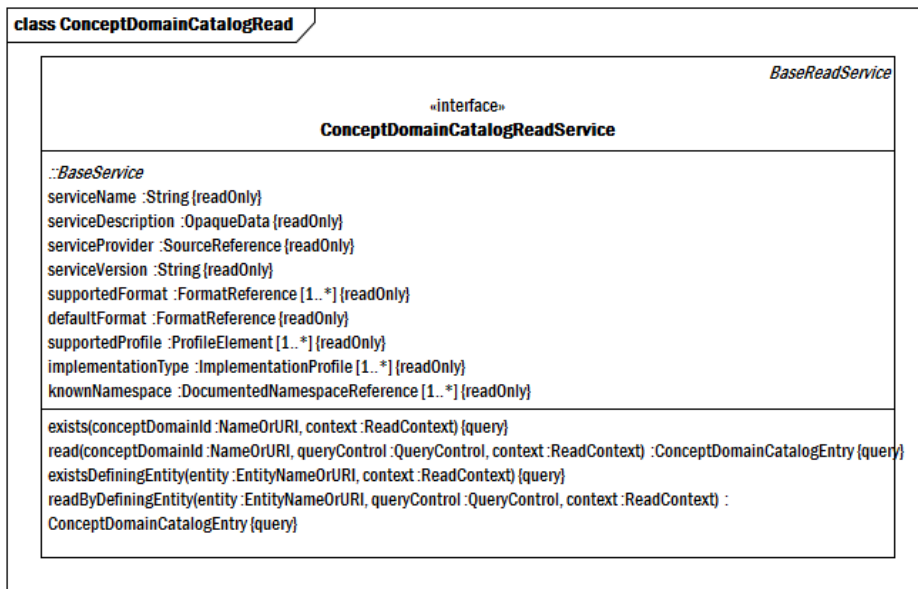


Figure 3.1 - Concept Domain Catalog Read

The `ConceptDomainCatalogReadService` provides direct read access to the contents of a `ConceptDomain` catalog. Catalog entries may be retrieved by name, about URI, alternateID URI, or definingEntity. Context information may be supplied that determines whether only active concept domains will be considered or all will be candidates and, where supported, the temporal context of the retrieval.

3.1.1 Interface ConceptDomainCatalogReadService

A service that provides direct read access to metadata about concept domains that are known to the service implementation.

Superclasses

- Every instance of `ConceptDomainCatalogReadService` is also an instance of `BaseReadService`.

3.1.1.1 Operation: exists

Determine whether the service has a catalog entry for a concept domain with the supplied name or URI in the supplied context.

Input Parameters

- **conceptDomainId** - the local name, about URI, OID or alternateID of the resource to check for existence (Type: `NameOrURI`).
- **context**_{OPT} - parameters that control the language, date, and other contextual variables (Type: `ReadContext`).

Return Type: Boolean

Exceptions

- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.

3.1.1.2 Operation: read

Return the `ConceptDomainCatalogEntry` referenced by the supplied name or URI. Note that the URI may either be the about URI or one of the `alternateIDs`.

Input Parameters

- **conceptDomainId** - the local name, about URI, OID or `alternateID` of the resource to read (Type: `NameOrURI`).
- **queryControl_{OPT}** - parameters that control the ordering, timeout, and query behavior (Type: `QueryControl`).
- **context_{OPT}** - parameters that control the language, date, and other contextual variables (Type: `ReadContext`).

Return Type: ConceptDomainCatalogEntry

Exceptions

- **UnknownConceptDomain** - The named `conceptDomain` is not recognized by the service.
- **UnsupportedMatchAlgorithm** - The `matchAlgorithm` is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The `format` is not supported by the service implementation.
- **QueryTimeout** - The `timeLimit` was exceeded by the service.
- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.

3.1.1.3 Operation: existsDefiningEntity

Determine whether there is a catalog entry that is defined by the referenced entity or “concept.”

Input Parameters

- **entity** - the namespace/name or URI of an entity that is used in the definition of a concept domain (Type: `EntityNameOrURI`).
- **context_{OPT}** - parameters that control the language, date, and other contextual variables (Type: `ReadContext`).

Return Type: `Boolean`

Exceptions

- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.

3.1.1.4 Operation: `readByDefiningEntity`

Return the `ConceptDomainCatalogEntry` whose defining entity is represented by the supplied entity URI or namespace and name.

Input Parameters

- **entity** - the namespace/name or URI of a defining entity (“concept”) (Type: `EntityNameOrURI`).
- **queryControl_{OPT}** - parameters that control the ordering, timeout and query behavior (Type: `QueryControl`).
- **context_{OPT}** - parameters that control the language, date, and other contextual variables (Type: `ReadContext`).

Return Type: `ConceptDomainCatalogEntry`

Exceptions

- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **NoCatalogEntryForDefiningEntity** - There is not a catalog entry that references the supplied entity in the supplied context.
- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.
- **UnsupportedMatchAlgorithm** - The `matchAlgorithm` is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.

- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The `format` is not supported by the service implementation.
- **QueryTimeout** - The `timeLimit` was exceeded by the service.

3.2 Concept Domain Catalog Query Service

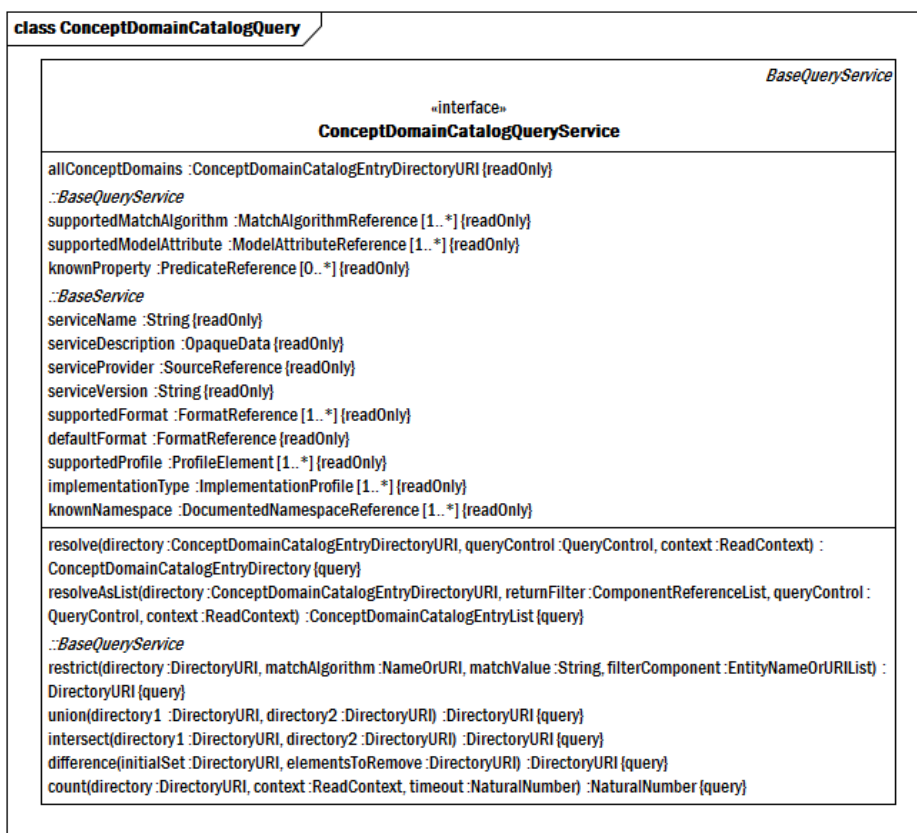


Figure 3.2 - Concept Domain Catalog Query

The `ConceptDomainCatalogQuery` service provides the ability to select and filter `ConceptDomainCatalogEntries` based on property values and other criteria.

3.2.1 Interface `ConceptDomainCatalogQueryService`

Superclasses:

- Every instance of `ConceptDomainCatalogQueryService` is also an instance of `BaseQueryService`.

Attributes:

- **allConceptDomains** - a `DirectoryURI` that references all of the entries known to the `ConceptDomainCatalog`.

3.2.1.1 Operation: resolve

Return a summary of the catalog entries referenced by directory.

Input Parameters:

- **directory** - a `DirectoryURI` that references a set of concept domain catalog entries (Type: `ConceptDomainCatalogEntryDirectoryURI`).
- **queryControl**_{OPT} - parameters that control the ordering, timeout, and query behavior. (Type: `QueryControl`)
- **context**_{OPT} - parameters that control the language, date and other contextual variables (Type: `ReadContext`)

Return Type: ConceptDomainCatalogEntryDirectory**Exceptions:**

- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service
- **InvalidDirectoryURI** - The supplied directory URI is not valid.
- **InvalidDirectoryType** - The type represented by the supplied directory URI is not the one required by the service invocation.
- **UnsupportedMatchAlgorithm** - The `matchAlgorithm` is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The `format` is not supported by the service implementation.
- **QueryTimeout** - The `timeLimit` was exceeded by the service.

3.2.1.2 Operation: resolveAsList

Return a summary of the catalog entries referenced by directory.

Input Parameters:

- **directory** - a `DirectoryURI` that references a set of concept domain catalog entries (Type: `ConceptDomainDirectoryURI`).
- **returnFilter**_{OPT} - a list of zero or more component references. If present, the returned list entries will contain only the required elements and attributes plus any elements or attributes named in the filter. If absent all elements and attributes will be returned. (Type: `ComponentReferenceList`)
- **queryControl**_{OPT} - parameters that control the ordering, timeout, and query behavior. (Type: `QueryControl`)
- **context**_{OPT} - parameters that control the language, date, and other contextual variables (Type: `ReadContext`)

Return Type: `ConceptDomainCatalogEntryList`

Exceptions:

- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service
- **InvalidDirectoryURI** - The supplied directory URI is not valid.
- **InvalidDirectoryType** - The type represented by the supplied directory URI is not the one required by the service invocation.
- **UnsupportedMatchAlgorithm** - The `matchAlgorithm` is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The `format` is not supported by the service implementation.
- **QueryTimeout** - The `timeLimit` was exceeded by the service.

3.3 Concept Domain Catalog History Service

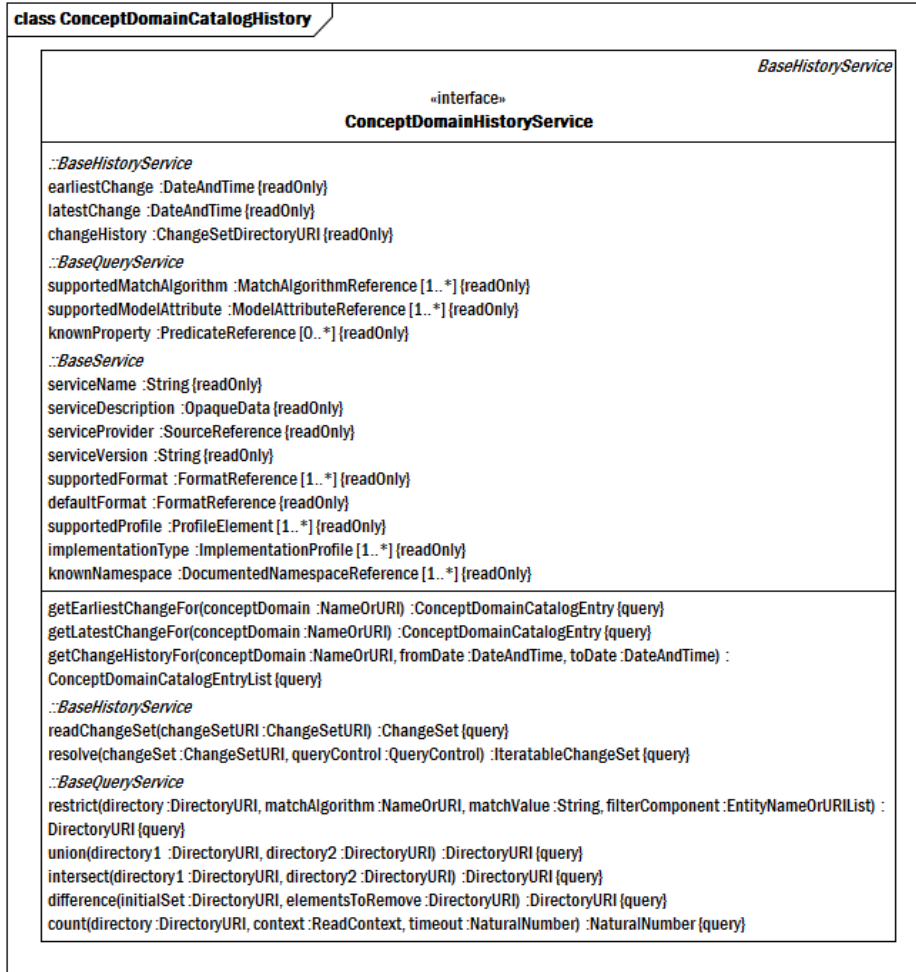


Figure 3.3 - Concept Domain Catalog History

3.3.1 Interface ConceptDomainHistoryService

Superclasses

- Every instance of `ConceptDomainHistoryService` is also an instance of `BaseHistoryService`.

3.3.1.1 Operation: `getEarliestChangeFor`

Retrieve the earliest change made to the catalog entry for the supplied concept domain.

Input Parameters

- **conceptDomain** - the name or URI of a concept domain (Type: `NameOrURI`).

Return Type: ConceptDomainCatalogEntry

Exceptions

- **UnknownConceptDomain** - The named `conceptDomain` is not recognized by the service.

3.3.1.2 Operation: getLatestChangeFor

Return the latest known change to the catalog entry for the supplied concept domain.

Input Parameters

- **conceptDomain** - the name or URI of a concept domain (Type: `NameOrURI`)

Return Type: ConceptDomainCatalogEntry

Exceptions

- **UnknownConceptDomain** - The named `conceptDomain` is not recognized by the service.

3.3.1.3 Operation: getChangeHistoryFor

Return a list of records that reflects the change in the catalog entry over the specified time span. Records will be sorted from earliest to latest if (a) `fromDate` is omitted or (b) `toDate` is omitted, or (c) `fromDate` is earlier than `toDate`. In all other cases, records will be sorted from latest to earliest.

Input Parameters

- **conceptDomain** - the name or URI of a concept domain (Type: `NameOrURI`).
- **fromDate_{OPT}** - the earliest date to return changes for. If omitted, the history begins at the first known catalog entry (Type: `DateAndTime`).
- **toDate_{OPT}** - the last date to return changes for. If omitted, all changes up to the last one made are returned. (Type: `DateAndTime`)

Return Type: ConceptDomainCatalogEntryList

Exceptions

- **UnknownConceptDomain** - The named `conceptDomain` is not recognized by the service.

3.4 Concept Domain Catalog Maintenance Service

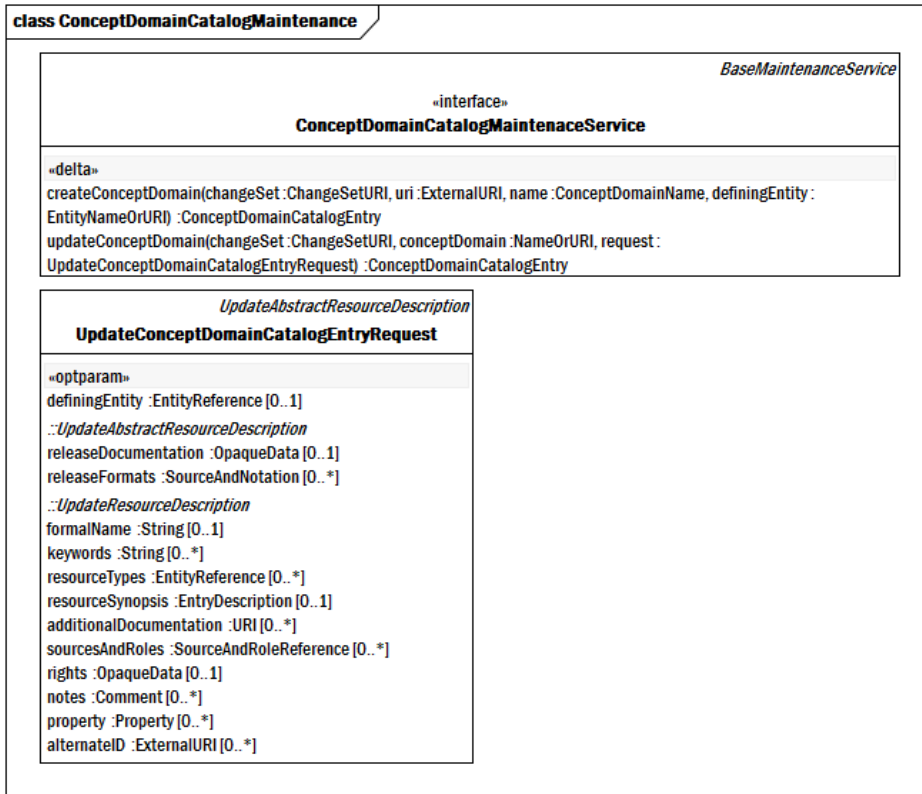


Figure 3.4 - Concept Domain Catalog Maintenance

3.4.1 Interface ConceptDomainCatalogMaintenanceService

A service that allows the creation, update, and deletion of Concept Domains. Note: the attributes and operations of the parent class, *BaseMaintenanceService*, are hidden in this diagram.

Superclasses

- Every instance of *ConceptDomainCatalogMaintenanceService* is also an instance of *BaseMaintenanceService*.

3.4.1.1 Operation: createConceptDomain

Create a new concept domain catalog entry.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the URI of an OPEN change set that will record the change (Type: *ChangeSetURI*).
- **uri** - the URI that of the concept domain referenced by the catalog entry (Type: *ExternalURI*).

- **name** - a label which uniquely identifies the new concept domain within the context of the implementing service (Type: `ConceptDomainName`).
- **definingEntity_{OPT}** - the namespace/name or URI of the entity (or “concept”) that defines the intent of the concept domain (Type: `EntityNameOrURI`).

Return Type: `ConceptDomainCatalogEntry`

Exceptions

- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **DuplicateConceptDomainURI** - The supplied URI is already the about URI or alternate ID for another concept domain in the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **DuplicateConceptDomainName** - The concept domain name is already used for another catalog entry.
- **DuplicateDefiningEntity** - More than one concept domain is defined by the same `definingEntity`.

3.4.1.2 Operation: `updateConceptDomain`

Update a concept domain catalog entry.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the URI of an `OPEN` change set that will record this change (Type: `ChangeSetURI`).
- **conceptDomain** - the name or URI of the concept domain to be updated (Type: `NameOrURI`).
- **request** - the set of parameters to update (Type: `UpdateConceptDomainCatalogEntryRequest`).

Return Type: `ConceptDomainCatalogEntry`

Exceptions

- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **UnknownConceptDomain** - The named `conceptDomain` is not recognized by the service.
- **DuplicateDefiningEntity** - More than one concept domain is defined by the same `definingEntity`.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnsupportedOntologyLanguage** - The supplied ontology language is not supported by the service.
- **UnsupportedStatus** - The name or URI of the `Changeable` status property is not recognized by the service.
- **UnsupportedSource** - The supplied source is not recognized by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.

- **UnsupportedRole** - The role name or URI is not recognized by the service.
- **UnsupportedOntologySyntax** - The supplied ontology syntax is not supported by the service.

3.4.2 Class **UpdateConceptDomainCatalogEntryRequest**

Superclasses

- Every instance of `UpdateConceptDomainCatalogEntryRequest` is also an instance of `UpdateAbstractResourceDescription`.

Attributes

- **definingEntity** - the entity (concept) that defines the concept domain.

4 Concept Domain Binding Information Model

4.1 Concept Domain Binding

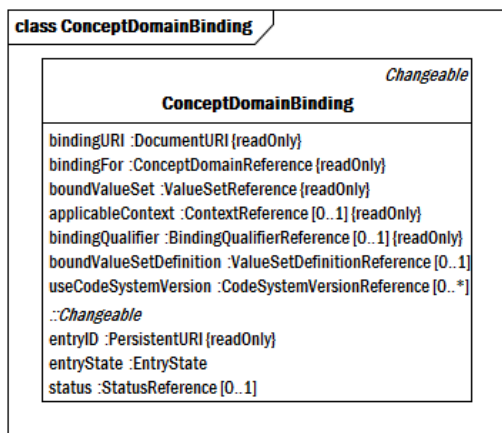


Figure 4.1 - Concept Domain Binding Model

A `ConceptDomainBinding` represents an association between a concept domain and a value set. The identity of a binding is the combination of the concept domain, the value set, the context (if present) and the binding qualifier (if present).

4.1.1 Class `ConceptDomainBinding`

The binding of a `ConceptDomain` and a `Value Set` that supplies the set of permissible value meanings in a given context.

`ConceptDomainBinding` can also bind specific value set definitions and/or specific code system versions where desired.

Superclasses

- Every instance of `ConceptDomainBinding` is also an instance of `Changeable`.

Attributes

- **bindingURI** - the unique identifier of this particular binding instance.
- **bindingFor** - the `ConceptDomain` being bound to a value set.
- **boundValueSet** - the bound value set.
- **applicableContext** - a realm or context in which the particular binding applies. If not present, the binding applies in any context not stated in another binding.
- **bindingQualifier** - additional information that qualifies the intent or purpose of the binding. In the HL7 case, this is intended to indicate “whether the binding is ‘overall,’ ‘minimum,’ or ‘maximum.’” The actual interpretation of the qualifier is outside of the scope of the CTS2 specification but, if present, `bindingQualifier` contributes to the identity of the specific binding.

- **boundValueSetDefinition** - the specific definition of the value set to be bound. If not specified, the definition that is used to determine the permissible values is specified when the concept domain is applied in the context.
- **useCodeSystemVersion** - the specific version(s) of the code systems to be used in the resolution of the value set definition. Note that it is also possible for value set definitions themselves to state which version of a code system is to be used. When this is the case, the code system version(s) specified in the `ConceptDomainBinding` do not override those stated in the value set definition - the value set definition version information takes precedence.

4.2 Concept Domain Binding Directory

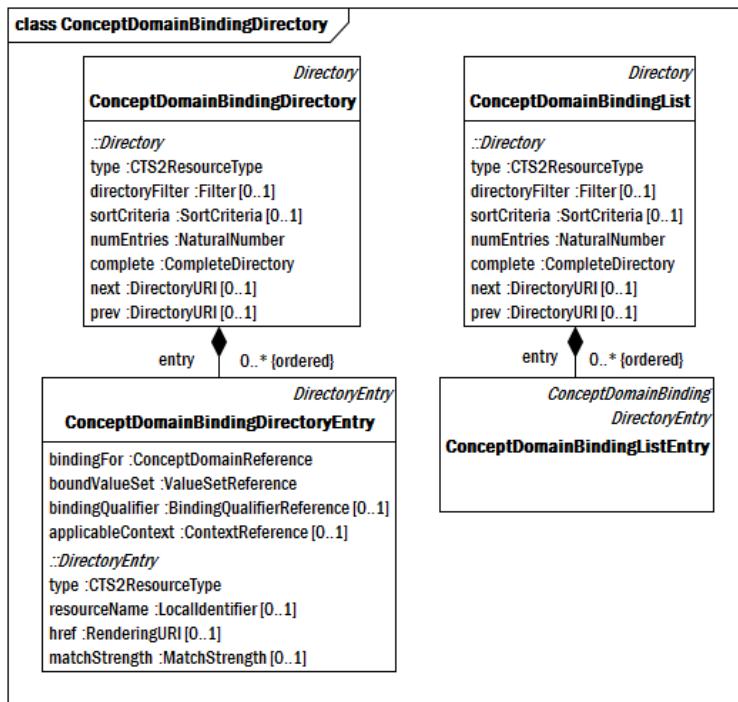


Figure 4.2 - Concept Domain Binding Directory

4.2.1 Class ConceptDomainBindingDirectory

A directory that summarizes a set of concept domain bindings.

Superclasses

- Every instance of `ConceptDomainBindingDirectory` is also an instance of `Directory`.

Attributes

- **entry** - an entry in a binding directory.

4.2.2 Class ConceptDomainBindingDirectoryEntry

A synopsis of a `ConceptDomainBinding` along with information about how to access the complete resource.

Superclasses

- Every instance of `ConceptDomainBindingDirectoryEntry` is also an instance of `DirectoryEntry`.

Attributes

- **bindingFor** - the `ConceptDomain` being bound to a value set.
- **boundValueSet** - the Value Set being bound.
- **bindingQualifier** - additional information that qualifies the intent or purpose of the binding.
- **applicableContext** - the context or realm in which the binding applies.

4.2.3 Class `ConceptDomainBindingList`

A collection of complete `ConceptDomainBinding` resources that meet a specified criteria.

Superclasses

- Every instance of `ConceptDomainBindingList` is also an instance of `Directory`.

Attributes

- **entry** - an entry in a binding list.

4.2.4 Class `ConceptDomainBindingListEntry`

A directory of a set of concept domain bindings.

Superclasses

- Every instance of `ConceptDomainBindingListEntry` is also an instance of `DirectoryEntry`.
- Every instance of `ConceptDomainBindingListEntry` is also an instance of `ConceptDomainBinding`.

5 Concept Domain Binding Services

5.1 Concept Domain Binding Read Services

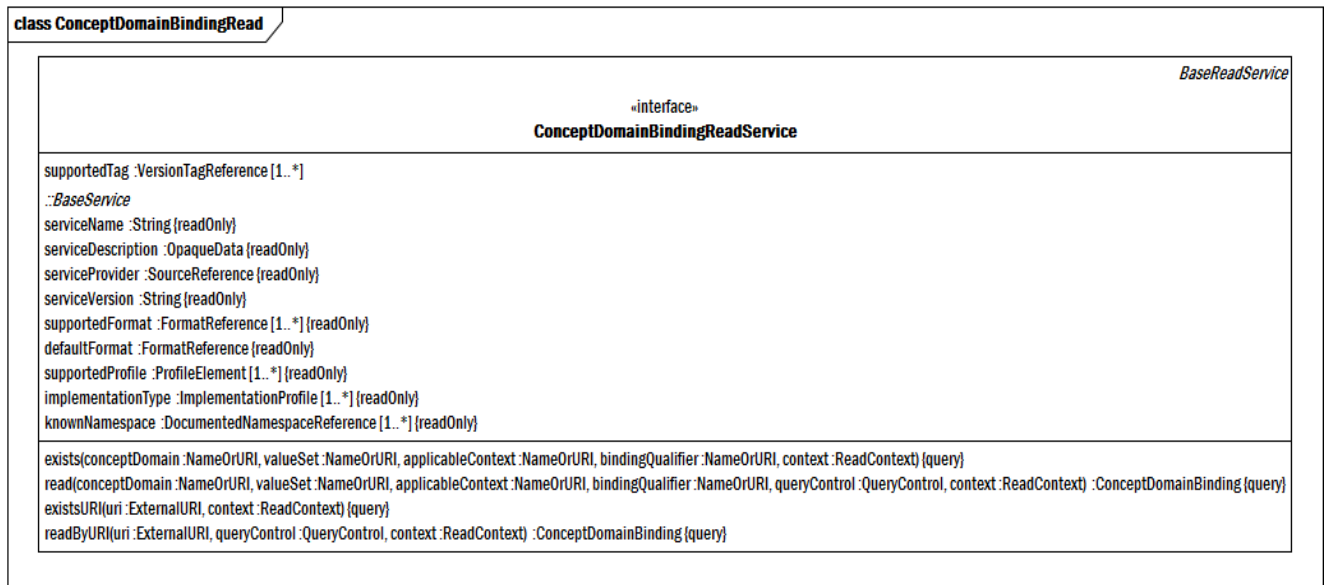


Figure 5.1 - Concept Domain Binding Read

5.1.1 Interface ConceptDomainBindingReadService

A service that provides direct read access to bindings that are asserted between concept domains and value sets.

Superclasses

- Every instance of `ConceptDomainBindingReadService` is also an instance of `BaseReadService`.

Attributes

- **supportedTag** - the set of version tags known to the service. All services must recognize the tag, `CURRENT`, the interpretation of which is service specific.

5.1.1.1 Operation: exists

Determine whether a concept domain binding exists for the specified domain, value set, realm, and context. Note that only the `conceptDomain` name and/or `valueSet` names must be validated. A service implementation may choose whether to validate the URIs or simply assume that, if not found in the service, they reference external elements.

Input Parameters

- **conceptDomain** - the local name, URI, OID or alternateID of the concept domain. (Type: `NameOrURI`)
- **valueSet** - the local name, URI, OID or alternateID of the bound value set. (Type: `NameOrURI`)
- **applicableContext**_{OPT} - the local name, URI, OID or alternateID of a context. If not supplied, only bindings with no

context will be returned. (Type: NameOrURI)

- **bindingQualifier_{OPT}** - the local name, URI, OID or alternateID of binding qualifier. If not supplied, only bindings without qualifiers will be returned. (Type: NameOrURI)
- **context_{OPT}** - parameters that control the language, date and other contextual variables (Type: ReadContext)

Return Type: Boolean

Exceptions

- **UnknownConceptDomain** - The named `conceptDomain` is not recognized by the service.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.
- **UnsupportedBindingQualifier** - The `bindingQualifier` is not recognized by the system.
- **UnknownValueSet** - The value set name or URI is not recognized by the service.
- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not OPEN.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service

5.1.1.2 Operation: read

Return the concept binding for the specified domain, value set, realm, and context. Note that only the `conceptDomain` name and/or `valueSet` names must be validated. A service implementation may choose whether to validate the URIs or simply assume that, if not found in the service, they reference external elements.

Input Parameters

- **conceptDomain** - the name or URI of the bound concept domain (Type: NameOrURI).
- **valueSet** - the name or URI of the bound value set (Type: NameOrURI).
- **applicableContext_{OPT}** - the name or URI of a realm or context. If not supplied, only bindings with no context will be returned (Type: NameOrURI).
- **bindingQualifier_{OPT}** - the name or URI of a binding qualifier. If not supplied, only bindings without qualifiers will be returned (Type: NameOrURI).
- **queryControl_{OPT}** - parameters that control the ordering, timeout, and query behavior (Type: QueryControl).

Return Type: ConceptDomainBinding

Exceptions

- **UnknownConceptDomainBinding** - The `conceptDomainBinding` URI is not recognized by the service.
- **UnknownConceptDomain** - The named `conceptDomain` is not recognized by the service.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.

- **UnsupportedBindingQualifier** - The `bindingQualifier` is not recognized by the system.
- **UnknownValueSet** - The value set name or URI is not recognized by the service.
- **UnsupportedMatchAlgorithm** - The `matchAlgorithm` is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The `format` is not supported by the service implementation.
- **QueryTimeout** - The `timeLimit` was exceeded by the service.

5.1.1.3 Operation: `existsURI`

Determine whether a concept domain binding with the supplied `DocumentURI` exists.

Input Parameters

- **uri** - the concept domain binding `documentURI` or an `alternativeID` (Type: `DocumentURI`).
- **context_{OPT}** - parameters that control the language, date and other contextual variables (Type: `ReadContext`)

Return Type: `Boolean`

Exceptions

- **UnsupportedNameOrURI** - The supplied `NameOrURI` is not recognized by the service.
- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.

5.1.1.4 Operation: `readByURI`

Return the concept domain binding with the supplied `DocumentURI`.

Input Parameters

- **uri** - the concept domain binding `documentURI` or an `alternativeID` (Type: `DocumentURI`).
- **queryControl_{OPT}** - parameters that control the ordering, timeout, and query behavior (Type: `QueryControl`).
- **context_{OPT}** - parameters that control the language, date, and other contextual variables (Type: `ReadContext`)

Return Type: `ConceptDomainBinding`

Exceptions

- **UnknownConceptDomainBinding** - The `conceptDomainBinding` URI is not recognized by the service.
- **UnsupportedMatchAlgorithm** - The `matchAlgorithm` is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **UnknownConceptDomainBinding** - The `conceptDomainBinding` URI is not recognized by the service.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The `format` is not supported by the service implementation.
- **QueryTimeout** - The `timeLimit` was exceeded by the service.

5.2 Concept Domain Binding Query Services

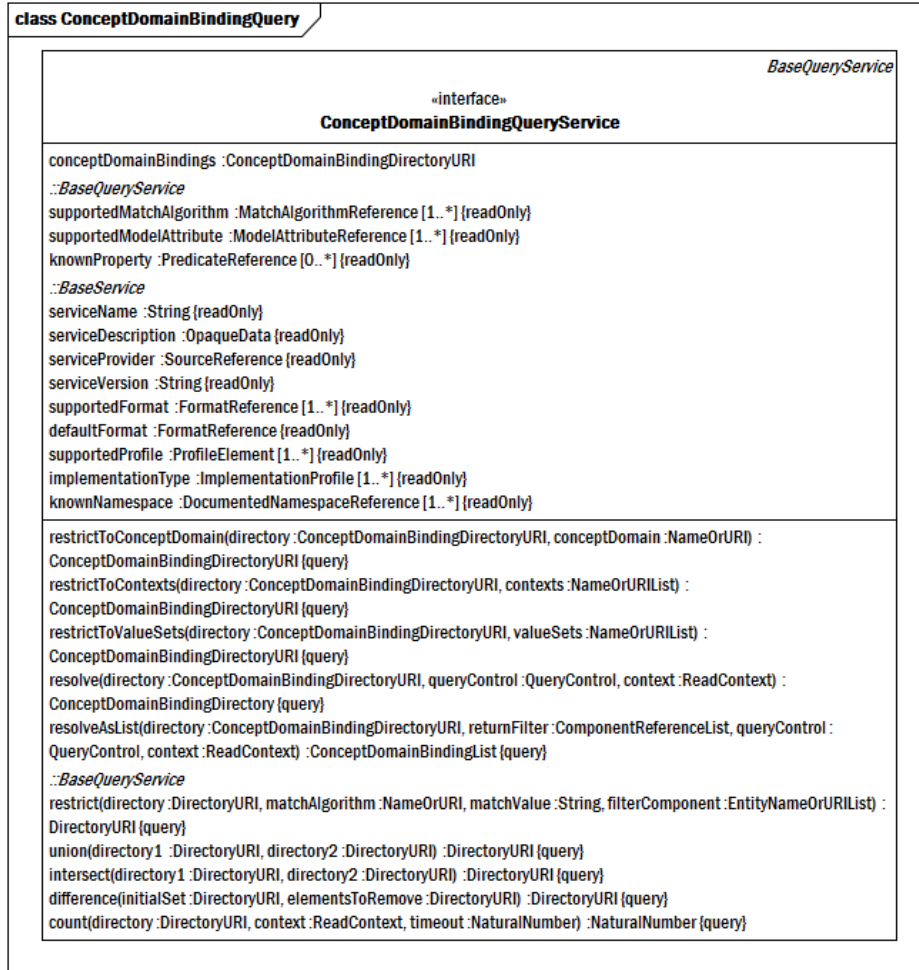


Figure 5.2 - Concept Domain Binding Query

The `ConceptDomainBindingQueryService` allows the query and filtering of the set of concept domain bindings known to the CTS2 service.

5.2.1 Interface ConceptDomainBindingQueryService

A service that allows the enumeration, query and filtering of the concept domain bindings known to the system.

Superclasses

- Every instance of `ConceptDomainBindingQueryService` is also an instance of `BaseQueryService`.

Attributes

- **conceptDomainBindings** - a `DirectoryURI` that resolves to the set of all concept domain bindings known to the service.

5.2.1.1 Operation: restrictToConceptDomain

Restrict the set of bindings referenced by directory to those that are bound to the supplied conceptDomain.

Input Parameters

- **directory** - a `DirectoryURI` that resolves to a set of concept domain bindings (Type: `ConceptDomainBindingDirectoryURI`).
- **conceptDomain** - the local identifier or URI of a specific concept domain (Type: `NameOrURI`).

Return Type: `ConceptDomainBindingDirectoryURI`

Exceptions

- **UnknownConceptDomain** - The named `conceptDomain` is not recognized by the service.
- **InvalidDirectoryURI** - The supplied directory URI is not valid.
- **InvalidDirectoryType** - The type represented by the supplied directory URI is not the one required by the service invocation.

5.2.1.2 Operation: restrictToContexts

Return a directory URI that references all of the concept domain bindings in directory that apply in one or more of the supplied contexts.

Input Parameters

- **directory** - a `DirectoryURI` that resolves to a set of concept domain bindings (Type: `ConceptDomainBindingDirectoryURI`).
- **contexts** - the name(s) or URI(s) of a set of realms or contexts (Type: `NameOrURIList`).

Return Type: `ConceptDomainBindingDirectoryURI`

Exceptions

- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.
- **InvalidDirectoryURI** - The supplied directory URI is not valid.
- **InvalidDirectoryType** - The type represented by the supplied directory URI is not the one required by the service invocation.

5.2.1.3 Operation: restrictToValueSets

Return a directory URI that references all of the concept domain bindings in directory that bind to the supplied value set(s).

Input Parameters

- **directory** - a `DirectoryURI` that resolves to a set of concept domain bindings (Type: `ConceptDomainBindingDirectoryURI`).
- **valueSets** - a list of local identifiers or URIs that name one or more value sets. (Type: `NameOrURI List`).

Return Type: ConceptDomainBindingDirectoryURI

Exceptions

- **UnknownValueSet** - The value set name or URI is not recognized by the service.
- **InvalidDirectoryURI** - The supplied directory URI is not valid.
- **InvalidDirectoryType** - The type represented by the supplied directory URI is not the one required by the service invocation.

5.2.1.4 Operation: resolve

Return a directory that summarizes the set of bindings referenced by the directory URI.

Input Parameters

- **directory** - a `DirectoryURI` that resolves to a set of concept domain bindings (Type: `ConceptDomainBindingDirectoryURI`).
- **queryControl_{OPT}** - parameters that control the ordering, timeout, and query behavior (Type: `QueryControl`).
- **context_{OPT}** - parameters that control the language, date, and other contextual variables (Type: `ReadContext`).

Return Type: ConceptDomainBindingDirectory

Exceptions

- **InvalidDirectoryURI** - The supplied directory URI is not valid.
- **InvalidDirectoryType** - The type represented by the supplied directory URI is not the one required by the service invocation.
- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not OPEN.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.
- **UnsupportedMatchAlgorithm** - The `matchAlgorithm` is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The `format` is not supported by the service implementation.
- **QueryTimeout** - The `timeLimit` was exceeded by the service.

5.2.1.5 Operation: resolveAsList

Return the set of bindings referenced by the directory URI.

Input Parameters

- **directory** - a `DirectoryURI` that resolves to a set of concept domain bindings (Type: `ConceptDomainBindingDirectoryURI`).
- **returnFilter**_{OPT} - a list of zero or more component references. If present, the returned list entries will contain only the required elements and attributes plus any elements or attributes named in the filter. If absent all elements and attributes will be returned. (Type: `ComponentReferenceList`).
- **queryControl**_{OPT} - parameters that control the ordering, timeout, and query behavior (Type: `QueryControl`).
- **context**_{OPT} - parameters that control the language, date, and other contextual variables (Type: `ReadContext`).

Return Type: `ConceptDomainBindingList`

Exceptions

- **InvalidDirectoryURI** - The supplied directory URI is not valid.
- **InvalidDirectoryType** - The type represented by the supplied directory URI is not the one required by the service invocation.
- **UnsupportedLanguage** - The `referenceLanguage` is not supported by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.
- **UnsupportedMatchAlgorithm** - The `matchAlgorithm` is not supported by the service.
- **UnsupportedPredicate** - The predicate name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnsupportedModelAttribute** - The name or URI of a CTS2 model attribute is not recognized and/or supported by the service implementation.
- **MatchValueFormatException** - The match value does not meet the format or syntax required or the supplied match algorithm.
- **UnsupportedFormat** - The `format` is not supported by the service implementation.
- **QueryTimeout** - The `timeLimit` was exceeded by the service.

5.3 Concept Domain Binding Maintenance Services

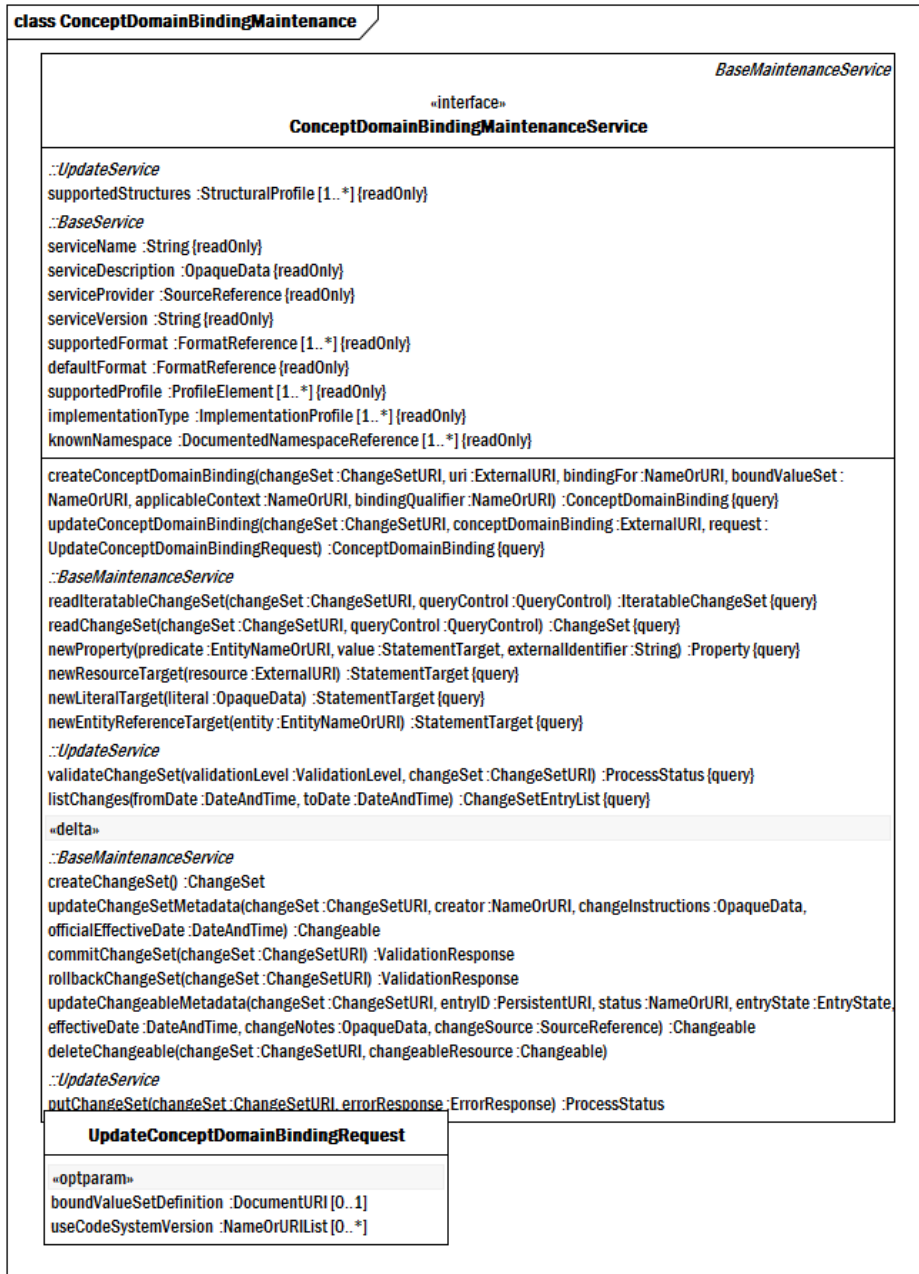


Figure 5.3 - Concept Domain Binding Maintenance

5.3.1 Interface ConceptDomainBindingMaintenanceService

A service that allows the creation, update, and deletion of concept domain bindings Superclasses:

- Every instance of ConceptDomainBindingMaintenanceService is also an instance of

BaseMaintenanceService.

5.3.1.1 Operation: createConceptDomainBinding

Create a new concept domain binding.

Input Parameters

- **changeSet** - the URI of an OPEN change set which will contain this change (Type: ChangeSetURI).
- **uri_{OPT}** - the URI for this specific binding. This parameter is optional and may be returned by the service. (Type: DocumentURI)
- **bindingFor** - the local identifier or URI of the concept domain being bound (Type: NameOrURI).
- **boundValueSet** - the local or identifier of the value set being bound (Type: NameOrURI).
- **applicableContext_{OPT}** - the name or URI of the context in which the binding applies. If not supplied, the binding applies whenever a context is not supplied. (Type: NameOrURI)
- **bindingQualifier_{OPT}** - the local name or URI of a binding qualifier (Type: NameOrURI).

Return Type: ConceptDomainBinding

Exceptions

- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ChangeSetIsNotOpen** - The changeSetContext is recognized by the service, but its state is not OPEN.
- **UnknownValueSet** - The value set name or URI is not recognized by the service.
- **UnknownConceptDomain** - The named conceptDomain is not recognized by the service.
- **UnsupportedBindingQualifier** - The bindingQualifier is not recognized by the system.
- **UnsupportedContext** - One or more changeSetContext is not supported by the service.

5.3.1.2 Operation: updateConceptDomainBinding

Change one or more elements in the named concept domain binding and return the result of the change.

Input Parameters

- **changeSet** - the URI of an OPEN change set that will contain this change (Type: ChangeSetURI).
- **conceptDomainBinding** - the URI of the concept domain binding to be updated (Type: DocumentURI).
- **request** - the set of attributes to update (Type: UpdateConceptDomainBindingRequest).

Return Type: ConceptDomainBinding

Exceptions

- **UnknownConceptDomainBinding** - The conceptDomainBinding URI is not recognized by the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.

- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **UnknownValueSetDefinition** - The `valueSetDefinition` URI isn't recognized by the service.
- **UnknownCodeSystemVersion** - The referenced `CodeSystemVersion` is not recognized by the service.

5.3.2 Class `UpdateConceptDomainBindingRequest`

Attributes

- **boundValueSetDefinition** - the specific value set definition to be used in this binding.
- **useCodeSystemVersion** - the local identifiers for or URIs of a set of code system versions to be used in the binding.

