Date: June 2013

CTS2 Map Services

Version 1.1

OMG Document Number: formal/2013-05-11
Standard document URL: http://www.omg.org/spec/cts2/1.1/

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

The CTS2 specification is divided into a number of separably implementable profiles. This section on Map Services describes how sets of `EntityReferences` are mapped from one code system or value set to a second. Map Services includes the ability to map from one source entity to multiple targets, to invoke sets of rules and to map to complex expressions.

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 undescores ('_') 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 Map Catalog Information Model

2.1 Overview

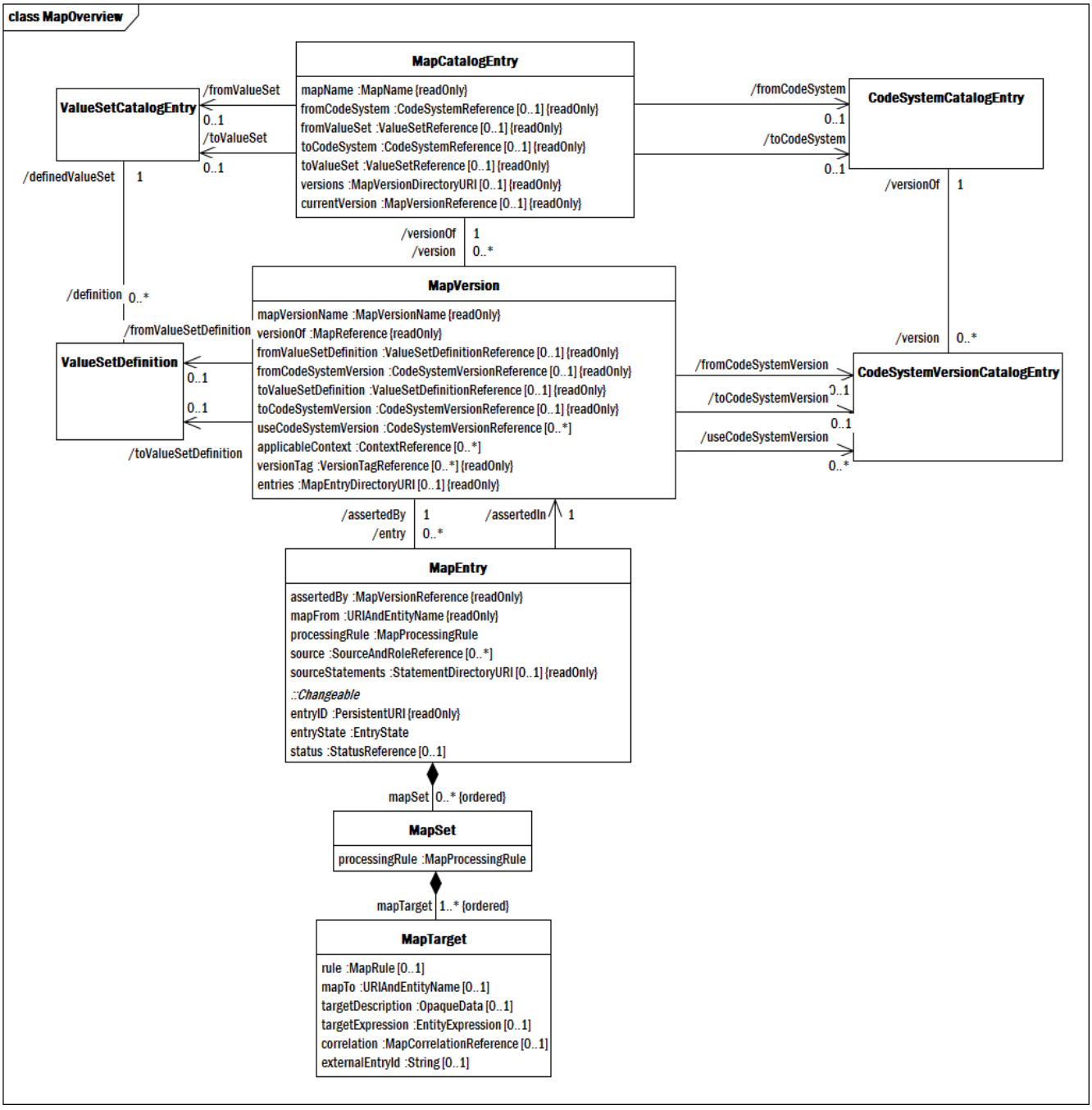

Figure 2.1 - Map Service Overview

This diagram represents an overview of the relationship between the various components in the map package. The MapCatalog and MapVersion packages are separable, meaning that each may exist independently of the other and/or may be implemented on different service environments.

The uppermost element, MapCatalogEntry describes the basic facts about a Map - what it is called, who publishes it, what it is for, etc. The MapVersion package provides a combination of descriptive information about specific versions of maps as well as carrying the actual content. A MapCatalogEntry also names the source of the set of entity references that serve as the "from" portion of the map and a second source for the "to" section. This set of references may either be described by naming the code system from which they were drawn or a value set that, in turn, can reference a complex set of references drawn from subsets of one or more code systems.

Every MapVersion must identify which code system version or which value set definition was used in the creation of the map contents. Exactly one of these must be named for the "from" section and one for the "to" section. Which of these is used depends on the way they are identified in the corresponding MapCatalogEntry.

A MapVersion also carries a DirectoryURI that resolves to the set of defining entries known to the map. Note that this URI is optional and a service implementation may choose to represent the MapEntries within a version or just the version information self.

Each MapEntry carries a unique "from" EntityReference, which must be active and either (a) defined by the fromCodeSystemVersion or (b) a member of the resolution of the from ValueSetDefinition against the supplied code system versions named in useCodesystemVersion.

The MapEntry contains an ordered list of MapSets. If this list is empty, the assertion is that EntityReference does NOT map to anything within this version of the map. If there is not a MapEntry for a given EntityReference, no assumptions may be made - while a map does not exist, it may simply be because the map set is incomplete.

The process flow for individual map entries is shown in a diagram that follows. Basically, MapSets are processed in sequential order until the first matching target (MapEntry. stopOnMatch = STOP) or all matching targets (MapEntry. stopOnMatch = CONTINUE) are found. Each MapTarget is also evaluated sequentially with the same rules applying for MapSet.stopOnMatch.

A MapTarget carries an opaque rule which, when evaluated, returns True - meaning that the target matches or False meaning that it doesn't. If True, the target can return an EntityReference, a textual description of the match and/or a formal EntityExpression that can be interpreted by an outside service to provide the map results.

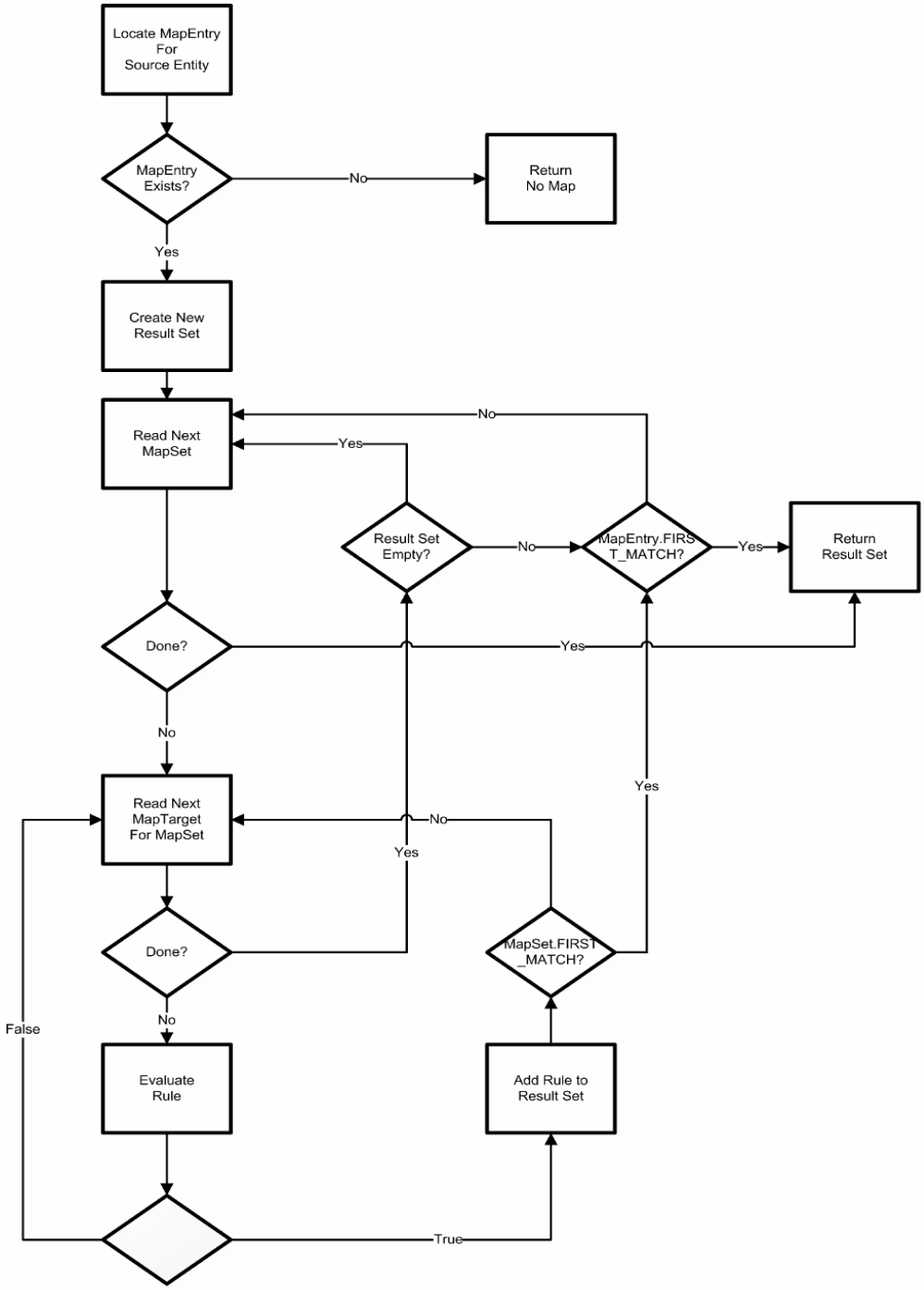

Figure 2.2 - Map Processing Flowchart

2.2 Map Catalog Entry

Figure 2.3 - Map Catalog Entry

`MapCatalogEntry` is an entry in a catalog of maps that are known to a service implementation. They contain basic information about the resource including who created it, what it is for, etc. We anticipate that this particular model will be extended in subsequent versions to include information about the rule languages and interface requirements of the various maps but, at the moment this represents a core set of properties.

2.2.1 Class MapCatalogEntry

An entry in a catalog of maps - organized collections of rules that describe how information encoded using one set of value meanings can be transformed into a second, related set of meanings. A map catalog contains information about who is responsible for creating and distributing the rules, the source code system or value set, the target code system or value set, how often the rules are updated, how they are distributed, etc.

Superclasses

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

Attributes

- **mapName** - the name by which the map is known within the context of the CTS2 service instance. `mapName` must uniquely designate a single, abstract Map within the context of all maps known to the instance of the service. Note that `mapName` is not necessarily globally unique and the `about` URI should be used whenever communication information about the `Map` across time or service instances.
- **fromCodeSystem** - the `CodeSystem` from which the source codes in the Map are drawn. Note that the source codes may be drawn from either a `CodeSystem` or a `ValueSet`, but not both.
- **fromValueSet** - the `ValueSet` from which the source codes in the Map are drawn. Note that the source codes may be drawn from either a `CodeSystem` or a `ValueSet`, but not both.
- **toCodeSystem** - the `CodeSystem` from which the target codes in the Map are drawn. Note that the target codes may be drawn from either a `CodeSystem` or a `ValueSet`, but not both.
- **toValueSet** - the `ValueSet` from which the source codes in the Map are drawn. Note that the target codes may be drawn from either a `CodeSystem` or a `ValueSet`, but not both.
- **versions** - a URI that, when resolved, returns the known versions of the containing `Map`.
- **currentVersion** - The map version associated with this catalog entry that has been assigned the `CURRENT` tag, if any.

Invariants

1. Either `fromCodeSystem` or `fromValueSet` must be specified but not both.
2. Either `toCodeSystem` or `toValueSet` must be specified but not both.

2.3 Map Directory and List

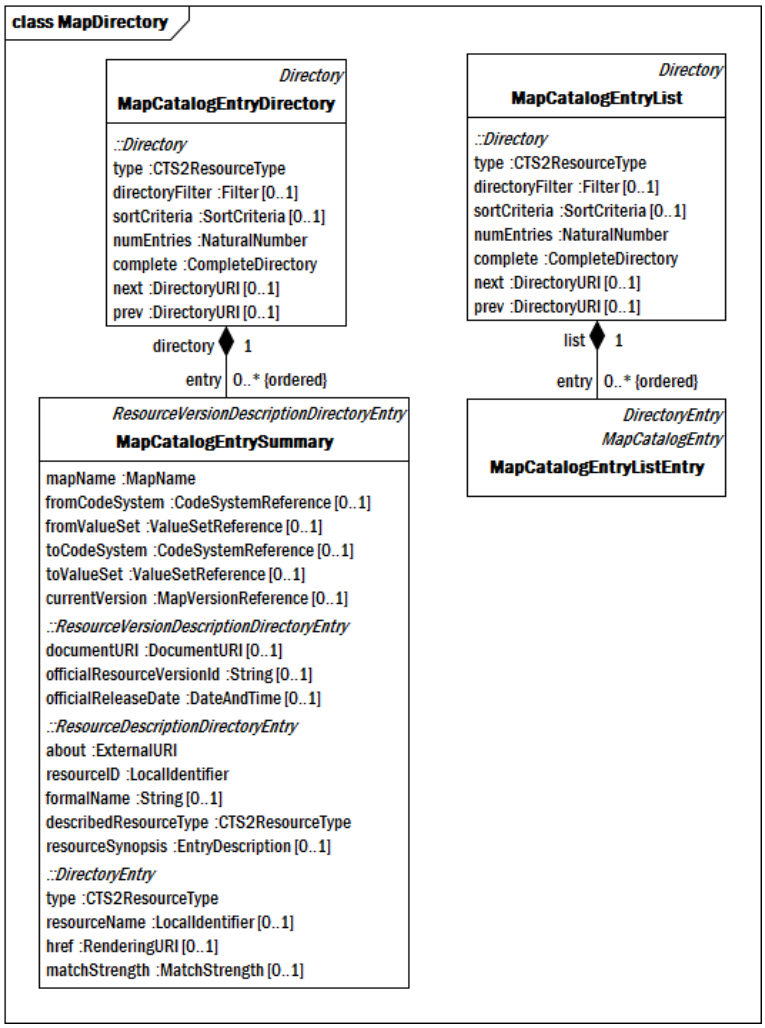

Figure 2.4 - Map Directory

2.3.1 Class Map CatalogEntryDirectory

A directory summarizing all or some of the `MapCatalogEntries` known to the service.

Superclasses

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

Attributes

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

Invariants

1. The directory type must be MAP.
2. All of the directory entries must be of type MAP.

2.3.2 Class MapCatalogEntryList

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

Superclasses

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

Attributes

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

Invariants

1. The type must be MAP.
2. The type of the entries must all be MAP.

2.3.3 Class MapCatalogEntryListEntry

A `MapCatalogEntry` that meets a specified filter criteria.

Superclasses

- Every instance of `MapCatalogEntryListEntry` is also an instance of `MapCatalogEntry`.
- Every instance of `MapCatalogEntryListEntry` is also an instance of `DirectoryEntry`.

Invariants

1. Type must be MAP.

2.3.4 Class Map CatalogEntrySummary

A synopsis of a `MapCatalogEntry`.

Superclasses

- Every instance of `MapCatalogEntrySummary` is also an instance of `ResourceVersionDescriptionDirectoryEntry`.

Attributes

- **mapName** - the name by which the map is known within the context of the CTS2 service instance.
- **fromCodeSystem** - the `CodeSystem` from which the source codes in the Map are drawn.
- **fromValueSet** - the `ValueSet` from which the source codes in the Map are drawn.
- **toCodeSystem** - the `CodeSystem` from which the target codes in the Map are drawn.

- **toValueSet** - the `ValueSet` from which the source codes in the Map are drawn.
- **currentVersion** - a reference to the map version with the `CURRENT` tag, if any.

Invariants

1. Type must be MAP.

3 Map Catalog Services

3.1 Map Catalog Read Service

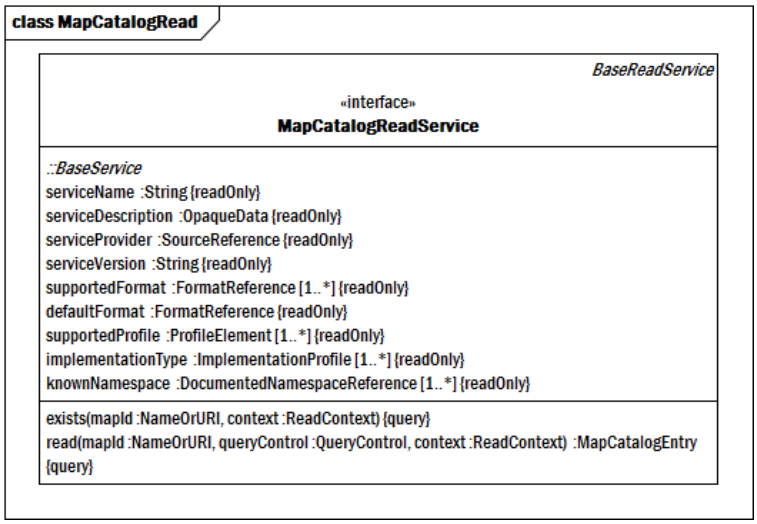

Figure 3.1 - Map Catalog Read Service

The `MapCatalogReadService` provides direct access to the service map catalog. Entries in the catalog can be retrieved either by an identifying URI or by a locally unique name.

3.1.1 Interface MapCatalogReadService

A service that provides direct read access to `MapCatalogEntries`.

Superclasses

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

3.1.1.1 Operation: exists

Determine whether a `MapCatalogEntry` exists for the supplied URI or service specific identifier.

- **mapId** - 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

Retrieve a `MapCatalogEntry` by name or URI.

Input Parameters

- **mapId** - 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: `MapCatalogEntry`

Exceptions

- **UnknownMap** - The named Map 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.
- **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 Map Catalog Query Service

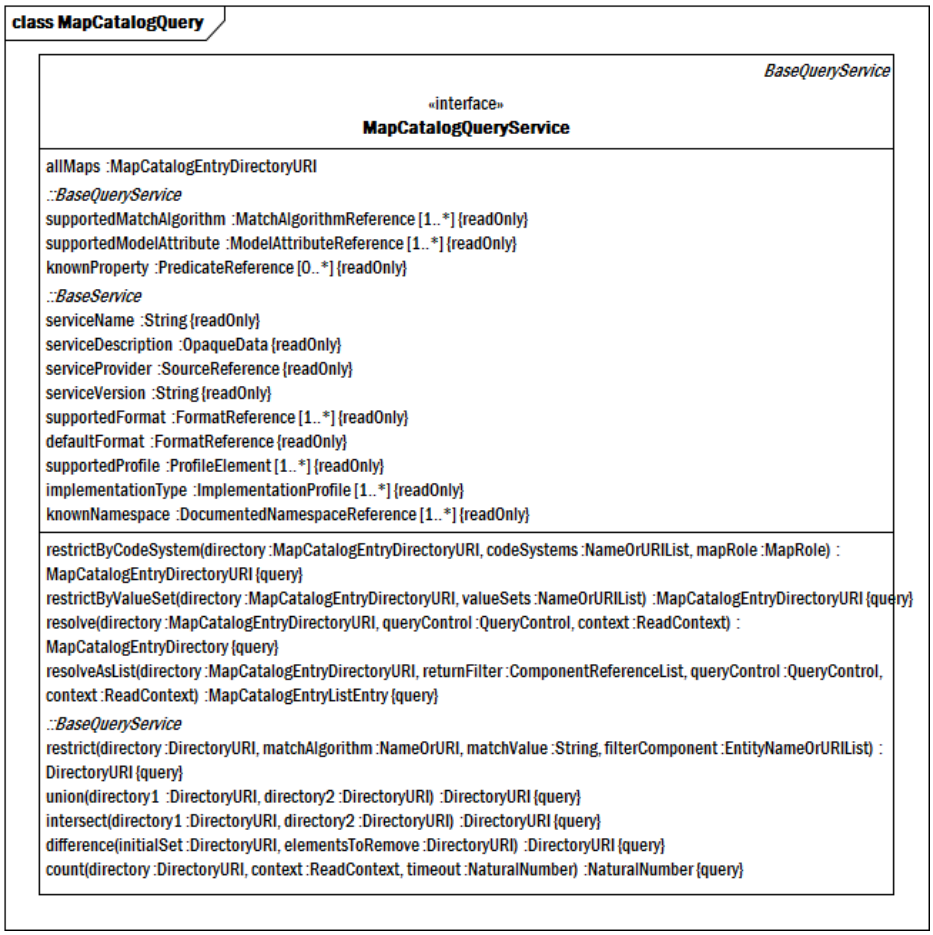

Figure 3.2 - Map Catalog Query Service

3.2.1 Interface MapCatalogQueryService

Superclasses

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

Attributes

- **allMaps** - a `DirectoryURI` that resolves to all map catalog entries known to the service.

3.2.1.1 Operation: restrictByCodeSystem

Restrict the list of catalog entries to those that appear in the `fromCodeSystem`, the `toCodeSystem`, or both depending on `mapRole`.

- **directory** - a `DirectoryURI` that resolves to a set of `MapCatalogEntries` (Type: `MapCatalogEntryDirectoryURI`).
- **codeSystems** - the local identifiers or URIs of one or more code systems (Type: `NameOrURIList`).
- **mapRole** - an indicator that states whether the restriction is on the "from" side, the "to" side, or both sides of a Map (Type: `MapRole`).

Return Type: `MapCatalogEntryDirectoryURI`

Exceptions

- **UnknownCodeSystem** - The referenced `CodeSystem` 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.

3.2.1.2 Operation: restrictByValueSet

Restrict the list of catalog entries to those that appear in the `fromValueSet`, the `toValue`, or both depending on `mapRole`.

Input Parameters

- **directory** - a `DirectoryURI` that resolves to a set of `MapCatalogEntries` (Type: `MapCatalogEntryDirectoryURI`).
- **valueSets** - the local identifiers or URIs of one or more value sets (Type: `NameOrURIList`).

Return Type: `MapCatalogEntryDirectoryURI`

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.

3.2.1.3 Operation: resolve

Resolve a `MapCatalogDirectoryURI` as a set of `MapCatalogSummaryEntries`.

Input Parameters

- **directory** - a `DirectoryURI` that references a set of `MapCatalogEntries` (Type: `MapCatalogEntryDirectoryURI`).
- **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: `MapCatalogEntryDirectory`

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.
- **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.
- **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.

3.2.1.4 Operation: resolveAsList

Resolve a `MapCatalogDirectoryURI` as a list of `MapCatalogEntries`.

Input Parameters

- **directory** - a `DirectoryURI` that references a set of `MapCatalogEntries` (Type: `MapCatalogEntryDirectoryURI`).
- **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: `MapCatalogEntryListEntry`

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.
- **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.
- **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.

3.3 Map Catalog History Service

Figure 3.3 - Map Catalog History Service

The `MapCatalogHistoryService` allows the client to find the first change, last change, and the set of changes that occurred to a `MapCatalogEntry` in a particular range of time.

3.3.1 Interface MapCatalogHistoryService

A service that provides a historical perspective on the changes that occurred in a map catalog over time.

Superclasses

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

3.3.1.1 Operation: getChangeHistoryFor

Return a list of `MapCatalogEntries` that show the states that the named catalog entry went through between `fromDate` and `toDate`.

Input Parameters

- **mapCatalog** - the name or URI of the entry to get the history for (Type: `NameOrURI`).
- **fromDate**$_{OPT}$ - the starting date. If later than `toDate`, the return order is latest to earliest, otherwise it is earliest to latest (Type: `DateAndTime`).
- **toDate**$_{OPT}$ - the end date. If later than `fromDate`, the return order is earliest to earliest, otherwise it is latest to earliest. (Type: `DateAndTime`).

Return Type: `MapCatalogEntryList`

Exceptions

- **UnknownMap** - The named Map is not recognized by the service.

3.3.1.2 Operation: getEarliestChangeFor

Return the earliest known change for the named map catalog entry.

Input Parameters

- **mapCatalog** - the name or URI of the entry to get the history for (Type: `NameOrURI`).

Return Type: `MapCatalogEntry`

Exceptions

- **UnknownMap** - The named Map is not recognized by the service.

3.3.1.3 Operation: getLastChangeFor

Return the last known change for the named map catalog entry.

Input Parameters

- **mapCatalog** - the name or URI of the entry to get the history for (Type: `NameOrURI`).

Return Type: `MapCatalogEntry`

Exceptions

- **UnknownMap** - The named Map is not recognized by the service.

3.4 Map Catalog Maintenance Service

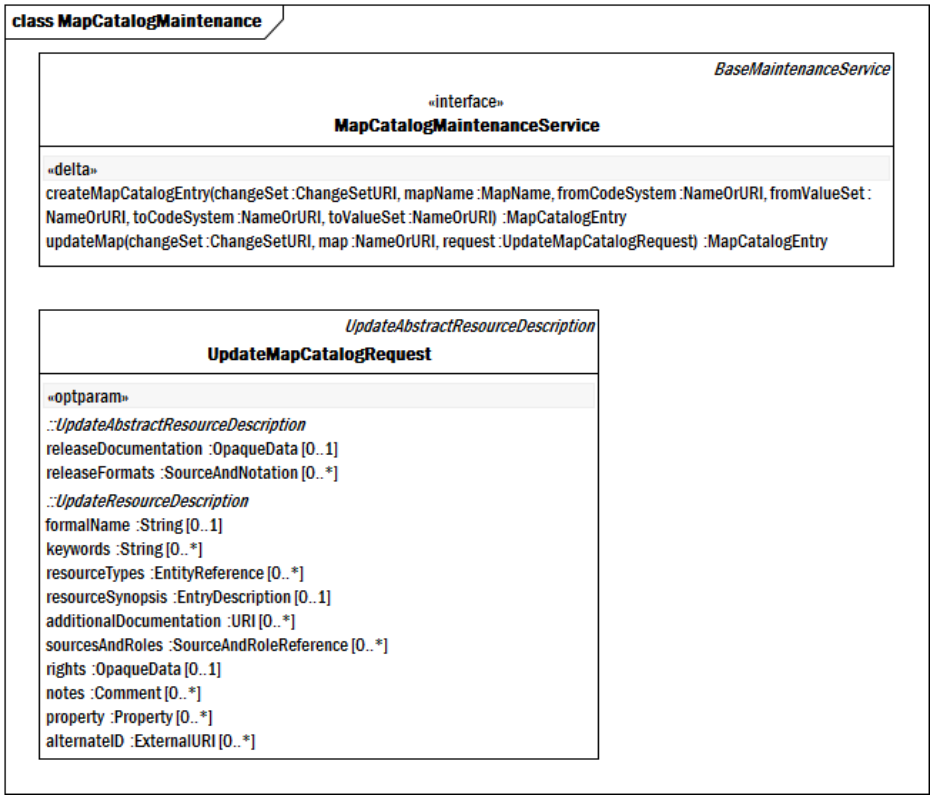

Figure 3.4 - Map Catalog Maintenance Service

3.4.1 Interface MapCatalogMaintenanceService

A service that enables the creation, modification, and removal of map catalog entries. Note that the attributes and methods inherited from `BaseMaintenanceService` have been hidden for the sake of simplicity.

Superclasses

- Every instance of `MapCatalogMaintenanceService` is also an instance of `BaseMaintenanceService`.

3.4.1.1 Operation: createMapCatalogEntry

Create a new entry in the map catalog.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the identifier of an OPEN change set that will record the change (Type: `ChangeSetURI`).

- **mapName** - the name by which the new catalog entry will be known within the context of the CTS2 service instance (Type: `MapName`).
- **fromCodeSystem**$_{OPT}$ - the name or URI of the code system from which the "from" entity references are derived (Type: `NameOrURI`).
- **fromValueSet**$_{OPT}$ - the name or URI of the value set from which the "from" entity references are derived (Type: `NameOrURI`).
- **toCodeSystem**$_{OPT}$ - the name or URI of the code system from which the "to" entity references are derived (Type: `NameOrURI`).
- **toValueSet**$_{OPT}$ - the name or URI of the value set from which the "to" entity references are derived (Type: `NameOrURI`).

Return Type: `MapCatalogEntry`

Exceptions

- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **MapCatalogFromReferenceException** - Exactly one of `fromCodeSystem` and `fromValueSet` must be supplied in a map catalog entry.
- **DuplicateMapName** - The supplied `mapName` already exists in the service.
- **MapCatalogToReferenceException** - Exactly one of `toCodeSystem` and `toValueSet` must be supplied in a map catalog entry.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.

Preconditions

1. Either a `fromCodeSystem` or `fromValueSet` must be supplied but not both.
2. Either a `toCodeSystem` or `toValueSet` must be supplied but not both.

3.4.1.2 Operation: updateMap

Update an existing entry in the map catalog.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the URI of an OPEN change set to record the change (Type: `ChangeSetURI`).
- **map** - the name or URI of the map to be updated (Type: `NameOrURI`).
- **request** - the set of attributes to update (Type: `UpdateMapCatalogRequest`).

Return Type: `MapCatalogEntry`

Exceptions

- **UnknownMap** - The named Map 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`.
- **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 UpdateMapCatalogRequest

Superclasses

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

4 Map Version Information Model

4.1 Map Version

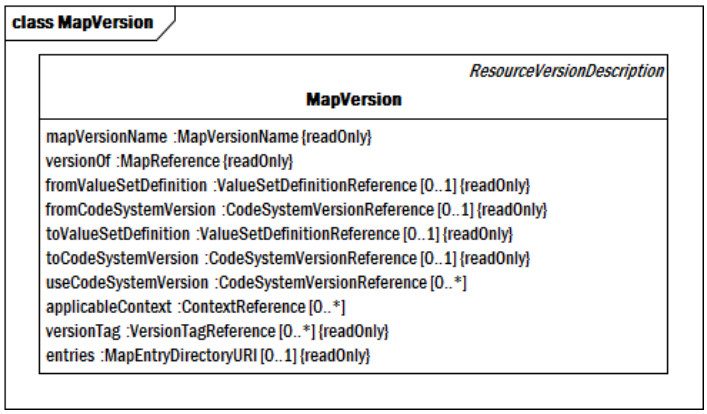

Figure 4.1 - Map Version

`MapVersion` contains information about a specific version of a `Map`, including release information, the particular format of the release being described, etc. It also provides a link to the actual contents of the particular version if the implementation is available.

4.1.1 Class Map Version

A specific version of a `Map`. `MapVersion` is bound to specific code system versions and/or value set versions and references a set of mapping entries (`mapSet`) that, if the resource is FINAL, are fixed with respect to this version.

Superclasses

- Every instance of `MapVersion` is also an instance of `ResourceVersionDescription`.

Attributes

- **mapVersionName** - a local identifier that uniquely names this version within the context of the CTS2 service implementation.
- **versionOf** - a reference to the `Map` resource that this is a version of.
- **fromValueSetDefinition** - the specific value set definition that was used to resolve the from entities in this map.
- **fromCodeSystemVersion** - the specific code system version that provided the from entities in this map.
- **toValueSetDefinition** - the specific value set definition that was used to resolve the to entities in this map.
- **toCodeSystemVersion** - the specific code system version that provided the to entities in this map.

- **useCodeSystemVersion** - a list of code system versions that should be used in the resolution of `fromValueSetDefinition` and/or `toValueSetDefinition`.

  If present, this states that, for these code systems, these specific versions are to be used.
- **applicableContext** - references to the realm, context, or other external factor that determines the applicability of this particular map version.
- **versionTag** - a version tag assigned to this `MapVersion` instance by the implementing service.
- **entries** - a `DirectoryURI` that resolves to the set of `MapEntries` that are contained in this `MapVersion`.

Invariants

1. Either `fromValueSetDefinition` or `fromCodeSystemVersion` must be present.
2. Either `toValueSetDefinition` or `toCodeSystemVersion` must be present.
3. If the `versionOfMap` has a `fromCodeSystem`, the version must have a `fromCodeSystemVersion`.
4. If the `versionOfMap` has a `toCodeSystem`, the version must have a `toCodeSystemVersion`.
5. `useCodeSystemVersion` must include `fromCodeSystemVersion` and `toCodeSystemVersion` if they are present.
6. `useCodeSystemVersion` must include all code system versions that are used by the `fromValueSetDefinition` and the `toValueSetDefinition`.

4.2 Map Version List and Directory

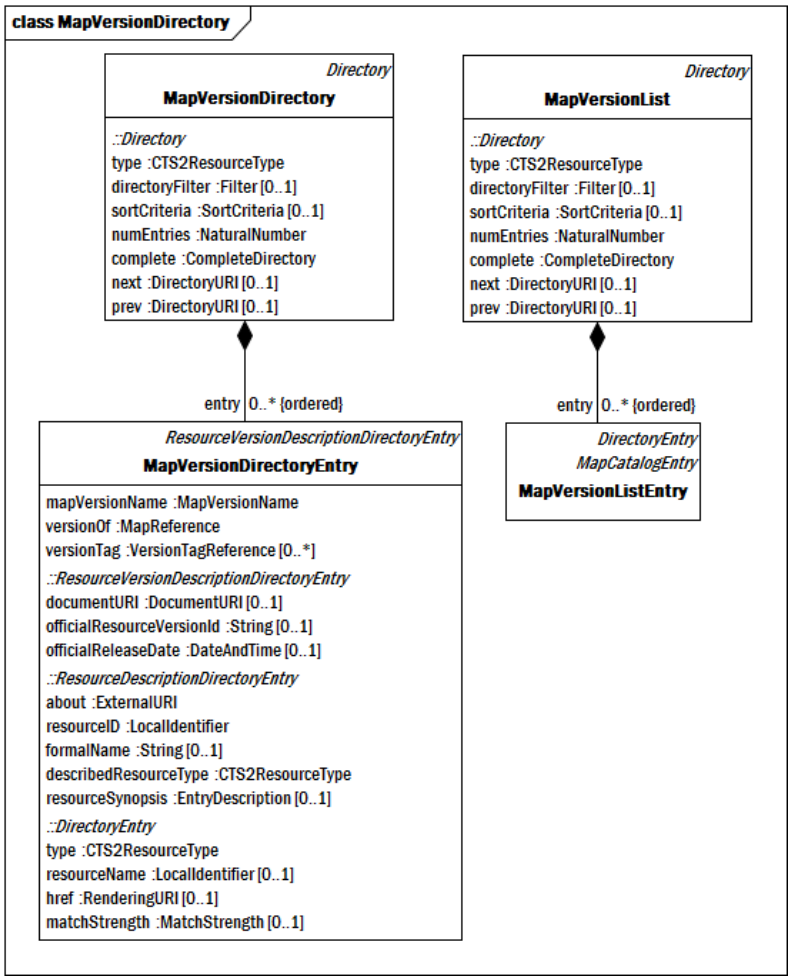

Figure 4.2 - Map Version Directory

4.2.1 Class Map VersionDirectory

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

4.2.1.1 Superclasses

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

Attributes

- entry - an entry in a `MapVersionDirectory`.

4.2.2 Class Map VersionDirectoryEntry

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

Superclasses

- Every instance of `MapVersionDirectoryEntry` is also an instance of `ResourceVersionDescriptionDirectoryEntry`.

Attributes

- **mapVersionName** - a name that uniquely identifies the map version within the context of the service.
- **versionOf** - a reference to the map that this is a version of.
- **versionTag** - a reference to a version tag assigned to this entry by the implementing service.

4.2.3 Class Map VersionList

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

Superclasses

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

Attributes

- entry - an entry in a `MapVersionList`.

4.2.4 Class Map VersionListEntry

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

Superclasses

- Every instance of `MapVersionListEntry` is also an instance of `MapCatalogEntry`.
- Every instance of `MapVersionListEntry` is also an instance of `DirectoryEntry`.

4.3 Map Entry

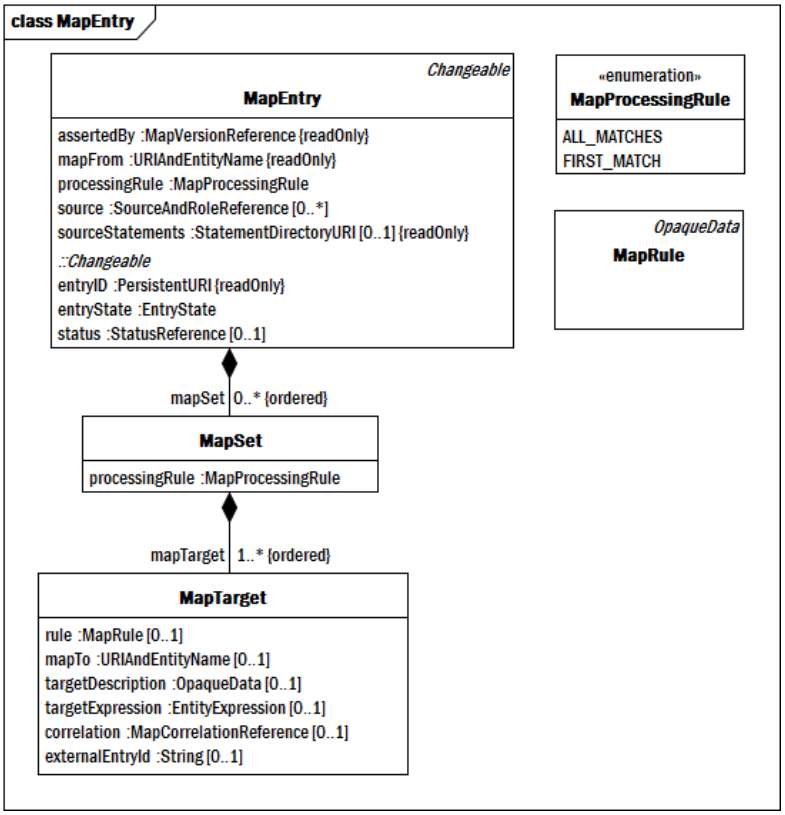

Figure 4.3 - Map Entry

A map entry represents the set of mappings that share the same `mapFrom` entity identifier. There may be at most one `MapEntry` for any unique `URIAndEntityName` within the context of a given `MapVersion`. Note also that is considered unidirectional from the perspective of the CTS2 model. If one wishes that, class A in code system C1 maps to class T in code system C2, and visa versa, one needs to create two map versions - the first from a version of C1 to a version of C2 and the second from a version of C2 to a version of C1.

4.3.1 Class MapEntry

`MapEntry` defines a set of rules that identify how a single `mapFrom` Entity maps to zero or more `mapTo` target Entities. A `MapEntry` instance is uniquely identified by the combination of the `assertedBy` `MapVersionReference` and the `mapFrom` entity identifier.

Superclasses

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

Attributes

- **assertedBy** - the mapping version that asserts this entry.

- **mapFrom** - the name and URI of the "from" or "source" side of the map. There can be at most one `mapFrom` entry for every unique entity in a given `MapVersion`.
- **processingRule** - an indicator that determines how the Map Set entries are evaluated. The first `MapSet` is evaluated and, if no match is found, the second set is evaluated, etc. If a match is found and `processingRule` is set to `FIRST_MATCH`, then processing terminates. Processing continues until all `MatchSets` have been evaluated in all other cases.
- **source** - a list of individuals or organizations and the role(s) they played in this entry.
- **sourceStatements** - a URI that resolves to the set of statements that were used to construct this entry.
- **mapSet** - A set of map rules to be evaluated as part of a `MapEntry`.

4.3.2 Class MapRule

A set of instructions that, when interpreted in the proper context, returns a true/false value, where true means that the context meets the requirements set forth by the rule and false means that it doesn't. Neither the syntax nor the semantics of map rules are included as part of the CTS2 specification.

Superclasses

- Every instance of `MapRule` is also an instance of `OpaqueData`.

4.3.3 Class MapSet

A set of map target rules that, when evaluated, result in zero or qualifying map targets. The setting of `MapSet.stopOnMatch` determines whether at most one qualifying target is returned or whether many qualifying targets are returned. `MapSets` are evaluated in the order that they appear in a `MapEntry`.

If the evaluation results in one or more qualifying `MapTargets` and `MapEntry.processingRule` is `FIRST_MATCH`, the `MapEntry` is considered resolved. If, however, `MapEntry.processingRule` is `ALL_MATCHES`, `allMapSets` in the `MapEntry` are evaluated.

Attributes

- **processingRule** - an indicator that states whether processing is to stop at the first matching `MapTarget` or whether all `MapTargets` in the set are to be evaluated.
- **mapTarget** - A rule/target pair in a `MapSet`.

4.3.4 Class MapTarget

A rule / target entry in a `MapSet`. The entries in a `MapSet` are evaluated in entry order. For each entry, rule is evaluated. If rule returns FALSE, then evaluation moves on to the next `MapTarget` in the set. If rule returns TRUE, the `MapTarget` is returned as a qualifying map in the associated `MapSet` and corresponding `MapEntry`. The setting of `processingRule` in the `MapSet` determines whether subsequent `MapTargets` are evaluated. If `processingRule` is set to `FIRST_MATCH`, the first matching `MapTarget` terminates the evaluation of the `MapSet`. If `processingRule` is set to `ALL_MATCHES`, then processing continues until all `MapTargets` in the set have been evaluated.

Attributes

- **rule** - a predicate of some sort that can be used to determine whether this entry meets the mapping criteria. If absent, the mapping is considered unconditional.
- **mapTo** - the target entity to be used if the rule is satisfied. If not present, the assumption is that the source entity maps to nothing in this set / rule.
- **targetDescription** - a description of the resulting rule. Can be used to carry text to instruct human beings how to do the mapping as well as additional machine readable instructions.
- **targetExpression** - a formal expression that, when interpreted, results in the map target. This expression could carry an RDF fragment, an expression in the SNOMED-CT compositional grammar or something else.
- **correlation** - a reference to how well the mapping correlates. Example correlations might be Exact, not mappable, partial overlap, narrower than, etc.
- **externalEntryId** - an identifier assigned to this particular source/set/target tuple by an outside body.

4.3.5 Enum MapProcessingRule

An indicator that states whether processing is to continue if a condition is satisfied or it is to terminate.

Enumeration Types

- **ALL_MATCHES** - processing continues whether a match is found in the current set or not. This potentially can return a match from every set.
- **FIRST_MATCH** - processing is to stop if a match is found. This mode returns the first match in a set.

4.4 Map Entry List and Directory

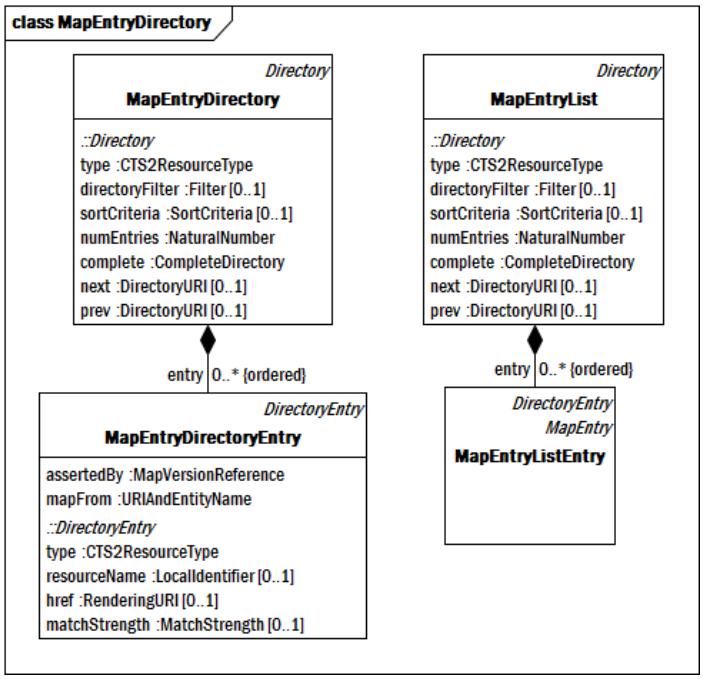

Figure 4.4 - Map Entry Directory

4.4.1 Class MapEntryDirectory

A directory of `MapEntries` that meet a specified criteria.

Superclasses

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

Attributes

- entry - An entry in a `MapEntryDirectory`.

4.4.2 Class MapEntryDirectoryEntry

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

Superclasses

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

Attributes

- **assertedBy** - a reference to the `MapVersion` that contains or asserts this entry.
- **mapFrom** - the URI and name of the source entity for this entry.

4.4.3 Class MapEntryList

A collection of complete `MapEntries` that meet a specified criteria.

Superclasses

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

Attributes

- entry - An entry in a `MapEntryList`.

4.4.4 Class MapEntryListEntry

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

Superclasses

- Every instance of `MapEntryListEntry` is also an instance of `MapEntry`.
- Every instance of `MapEntryListEntry` is also an instance of `DirectoryEntry`.

5 Map Version Services

5.1 Map Version Read Services

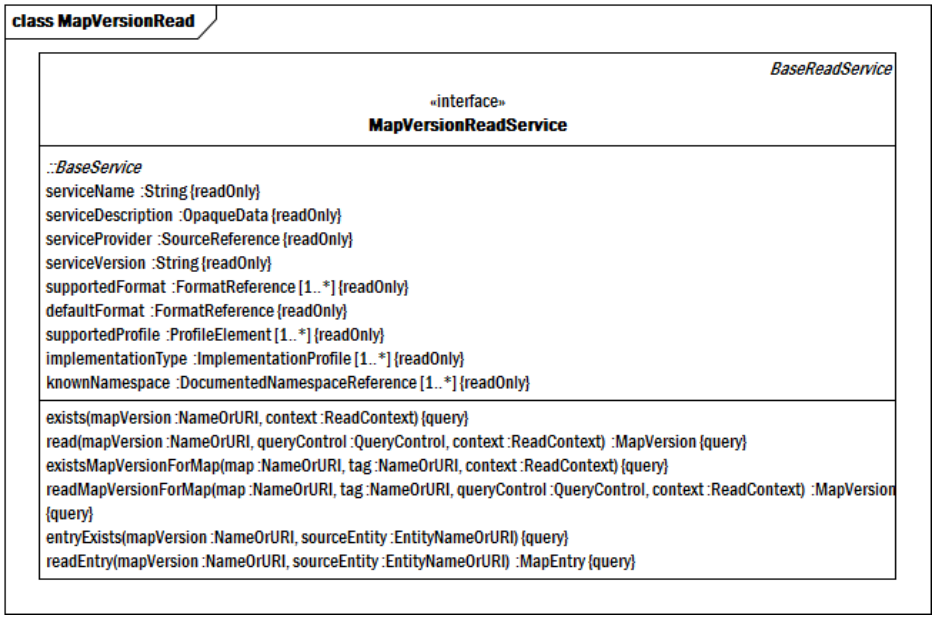

Figure 5.1 - Map Version Read Service

The `MapVersionReadService` provides direct access to the map versions that are known to the service. Entries in the service can be retrieved by URI, local name, or a combination of a map name and version tag.

5.1.1 Interface MapVersionReadService

Superclasses

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

5.1.1.1 Operation: exists

Determine if the specified map version is known to the service.

Input Parameters

- **mapVersion** - 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

- **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.

5.1.1.2 Operation: read

Retrieve a specific version of a map by name or URI.

Input Parameters

- **mapVersion** - 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: `MapVersion`

Exceptions

- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnknownMapVersion** - The named `mapVersion` 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.
- **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: existsMapVersionForMap

Determine whether a map version exists for the map name and version tag.

Input Parameters

- **map** - the local name, about URI, OID or alternateID of the map (Type: `NameOrURI`).
- **tag**$_{OPT}$ - the name or URI of a version tag that references a version of the map. If not supplied, the tag "CURRENT" will be used. (Type: `NameOrURI`)
- **context**$_{OPT}$ - parameters that control the language, date, and other contextual variables (Type: `ReadContext`).

Return Type: `Boolean`

Exceptions

- **UnknownMap** - The named `Map` is not recognized by the service.
- **UnsupportedVersionTag** - The `versionTag` is not recognized by the service.
- **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.

5.1.1.4 Operation: readMapVersionForMap

Retrieve a map version given the name of a map and a version tag.

Input Parameters

- **map** - the local name, about URI, OID or alternateID of the map (Type: `NameOrURI`).
- **tag**$_{OPT}$ - the name or URI of a version tag that references a version of the map. If not supplied, the tag "CURRENT" will be used. (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: `MapVersion`

Exceptions

- **UnsupportedVersionTag** - The `versionTag` is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnknownMap** - The named `Map` 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.

5.1.1.5 Operation: entryExists

Determine whether the supplied entity exists as the source of a map in the supplied version.

Input Parameters

- **mapVersion** - the local name, about URI, OID or alternateID of the containing map version (Type: `NameOrURI`).
- **sourceEntity** - the source entity name or URI (Type: `EntityNameOrURI`).

Return Type: `Boolean`

Exceptions

- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.

5.1.1.6 Operation: readEntry

Read the map entry for the supplied map version and entity.

Input Parameters

- **mapVersion** - the local name, about URI, OID or alternateID of the containing map version (Type: `NameOrURI`).
- **sourceEntity** - the source entity name or URI (Type: `EntityNameOrURI`).

Return Type: `MapEntry`

Exceptions

- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.

- **UnknownEntity** - The `EntityNameOrURI` is not known to the service.

5.2 Map Version Query Services

.

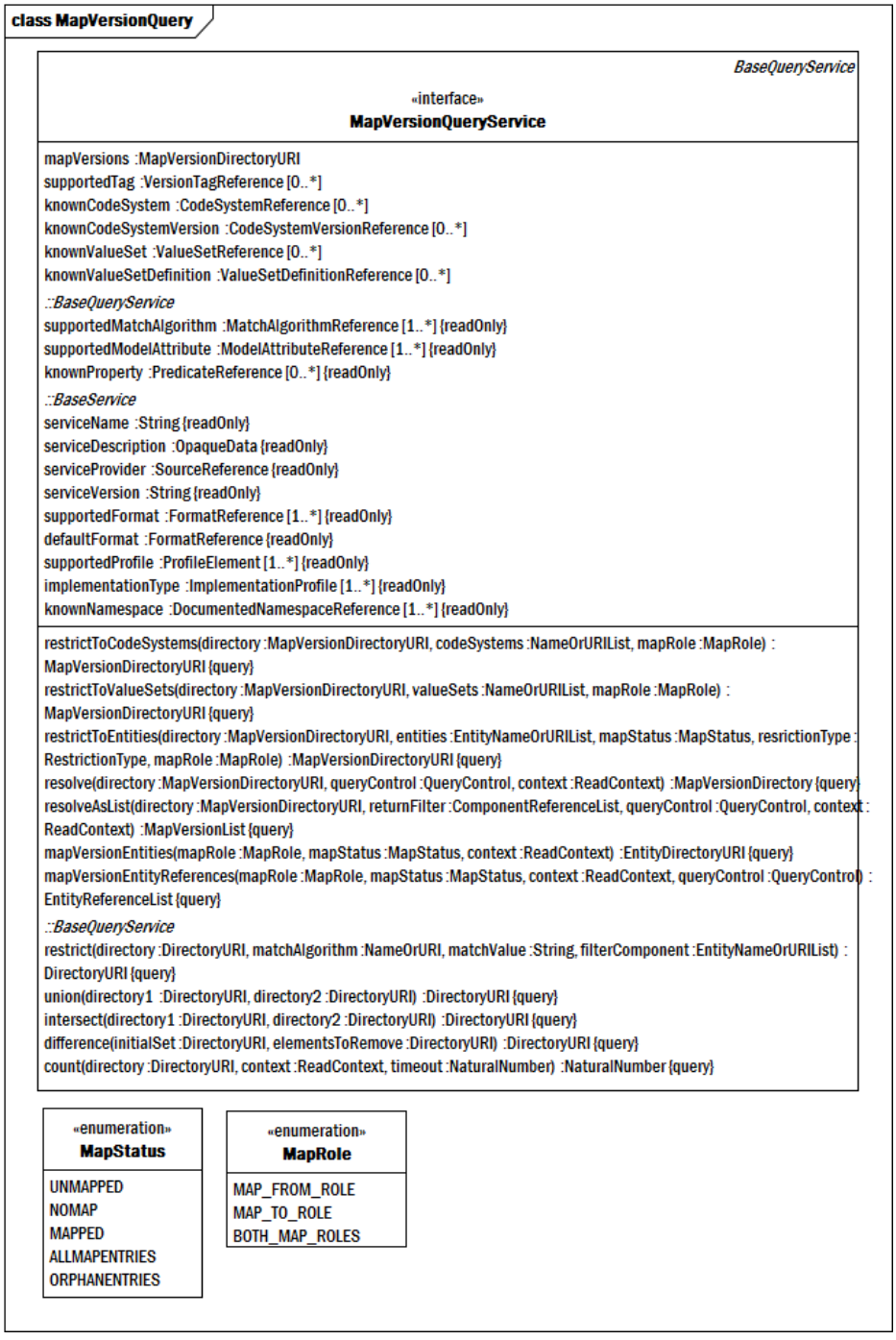

Figure 5.2 - Map Version Query Service

The MapVersionQueryService allows the filtering, query, and resolution of entries in the map catalog.

5.2.1 Interface MapVersionQueryService

Superclasses

- Every instance of MapVersionQueryService is also an instance of BaseQueryService.

Attributes

- **mapVersions** - a DirectoryURI that resolves to all of the map versions known to the service.
- **supportedTag** - a version tag that is recognized by this service for coupling specific map versions with maps at a point in time.
- **knownCodeSystem** - the name and URI of a code system that is known to the map query service.
- **knownCodeSystemVersion** - the name and URI of a code system version that is known to the service.
- **knownValueSet** - the name and URI of a value set that is known to the service.
- **knownValueSetDefinition** - the name and URI of a value set definition known to the service.

5.2.1.1 Operation: restrictToCodeSystems

Return a DirectoryURI that references the subset of the MapVersions referenced in directory that directly (fromCodeSystemVersion, toCodeSystemVersion) or indirectly (fromValueSet, toValueSet) reference entities that are described in the supplied list of code systems.

Input Parameters

- **directory** - a DirectoryURI that resolves to a set of MapVersions (Type: MapVersionDirectoryURI).
- **codeSystems** - a list containing names or URIs of the code systems to be filtered by (Type: NameOrURIList).
- **mapRole** - an indicator that determines whether the filter applies to the source code system, the target code system, or both (Type: MapRole).

Return Type: `MapVersionDirectoryURI`

Exceptions

- **UnknownCodeSystem** - The referenced CodeSystem is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to 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: restrictToValueSets

Return a DirectoryURI that references the subset of the MapVersions referenced in directory that directly (fromValueSetDefinition, toValueSetDefinition) or indirectly via value set containment reference entities that are described in the supplied list of value sets.

Input Parameters

- **directory** - a `DirectoryURI` that resolves to a set of `MapVersions` (Type: `MapVersionDirectoryURI`).
- **valueSets** - a list of value set names or URIs to apply as a filter (Type: `NameOrURIList`).
- **mapRole** - an indicator that determines whether list of catalog entries apply to the "from" and/or the "to" portions of the `MapVersion` (Type: `MapRole`).

Return Type: `MapVersionDirectoryURI`

Exceptions

- **UnknownValueSet** - The value set name or URI is not recognized by the service.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to 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: restrictToEntities

Return a `DirectoryURI` that references the subset of the `MapVersions` referenced in directory that reference at least one (restrictionType =`AT_LEAST_ONE`) or all (`restrictionType` =`ALL`) or the name identities.

Input Parameters

- **directory** - a `DirectoryURI` that resolves to a set of `MapVersions` (Type: `MapVersionDirectoryURI`).
- **entities** - a list of entity namespace/names or URIs to be used in the restriction (Type: `EntityNameOrURIList`).
- **mapStatus** - an indicator that determines whether the query is just for unmapped entries, entries that are declared to have no map or entries that have been mapped (including no maps) or all maps (Type: `MapStatus`).
- **resrictionType** - an indicator that determines whether the set must contain all entities to pass or at least one (Type: `RestrictionType`).
- **mapRole** - an indicator that determines whether the query applies to the map source, the map target, or both (Type: `MapRole`).

Return Type: `MapVersionDirectoryURI`

Exceptions

- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnknownEntity** - The `EntityNameOrURI` is not known to 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

Resolve a `MapVersionDirectoryURI` returning a directory of the map versions that it references.

Input Parameters

- **directory** - a `DirectoryURI` that resolves to a set of `MapVersions` (Type: `MapVersionDirectoryURI`).
- **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: `MapVersionDirectory`

Exceptions

- **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.
- **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.

5.2.1.5 Operation: resolveAsList

Resolve a `MapVersionDirectoryURI` returning a list of the `MapVersions` that it references.

Input Parameters

- **directory** - a `DirectoryURI` that resolves to a set of `MapVersions` (Type: `MapVersionDirectoryURI`).
- **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`).
- **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`).

Return Type: `MapVersionList`

Exceptions

- **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.
- **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.

5.2.1.6 Operation: mapVersionEntities

Return a `DirectoryURI` that resolves to the set of entities that meet the supplied criteria.

Input Parameters

- **mapRole** - an indicator that determines whether the function applies to the “from” side of the map, the “to” side of the map or both (Type: `MapRole`).
- **mapStatus** - an indicator that determines the status of the entity references to be returned. Status may be not mapped (in the code system or value set but not in the map), mapped to nothing, mapped, all entries, or all entries that are mapped that are not in the code system (Type: `MapStatus`).
- **context** - parameters that control the status, date, and time and other contextual variables (Type: `ReadContext`).

Return Type: `EntityDirectoryURI 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.

5.2.1.7 Operation: mapVersionEntityReferences

Return a list of entity references that meet the supplied criteria.

Input Parameters

- **mapRole** - an indicator that determines whether the function applies to the "from" side of the map, the "to" side of the map or both (Type: `MapRole`).
- **mapStatus** - an indicator that determines the status of the entity references to be returned. Status may be not mapped (in the code system or value set but not in the map), mapped to nothing, mapped, all entries, or all entries that are mapped that are not in the code system (Type: `MapStatus`).
- **context** - parameters that control the status, date and time, and other contextual variables (Type: `ReadContext`).
- **queryControl** - parameters that control the time limit (Type: `QueryControl`).

Return Type: `EntityReferenceList`

Exceptions

- **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.

5.2.1.8 Enum MapRole

An indicator that determines whether the "from," the "to," or both components of a `Map` or `MapVersion` are being queried.

Enumeration Types:

- **MAP_FROM_ROLE** - restrict a query to address only the "from" side of a `MapCatalogEntry` or `MapVersion`.
- **MAP_TO_ROLE** - restrict a query to address only the "to" side of a `MapCatalogEntry` or `MapVersion`.

- **BOTH_MAP_ROLES** - the query addresses both the “from” and “to” side of a map.

5.2.1.9 Enum MapStatus

An indicator that determines what class of entities are to be selected in the `mapVersionEntities` and `mapVersionEntityReference` operations.

Enumeration Types:

- **UNMAPPED** - an entity belongs to “from” code system or value set in the supplied context but does not appear in an `MapEntry` or `MapRule`.
- **NOMAP** - an entity is a source in a `MapEntry` but has no target sets. NOMAP references entities that have been explicitly declared to have no mapping.
- **MAPPED** - an entity appears as a source in a `MapEntry` and/or a `mapTo` in a `MapTarget`.
- **ALLMAPENTRIES** - an entity appears in the from value set or code system or the to value set or code system, depending on `MapRole`.
- **ORPHANENTRIES** - an entity appears as a source in a `MapEntry` or a `mapTo` in a `MapTarget` but does not appear in the corresponding code system or value set.

5.3 Map Version History Service

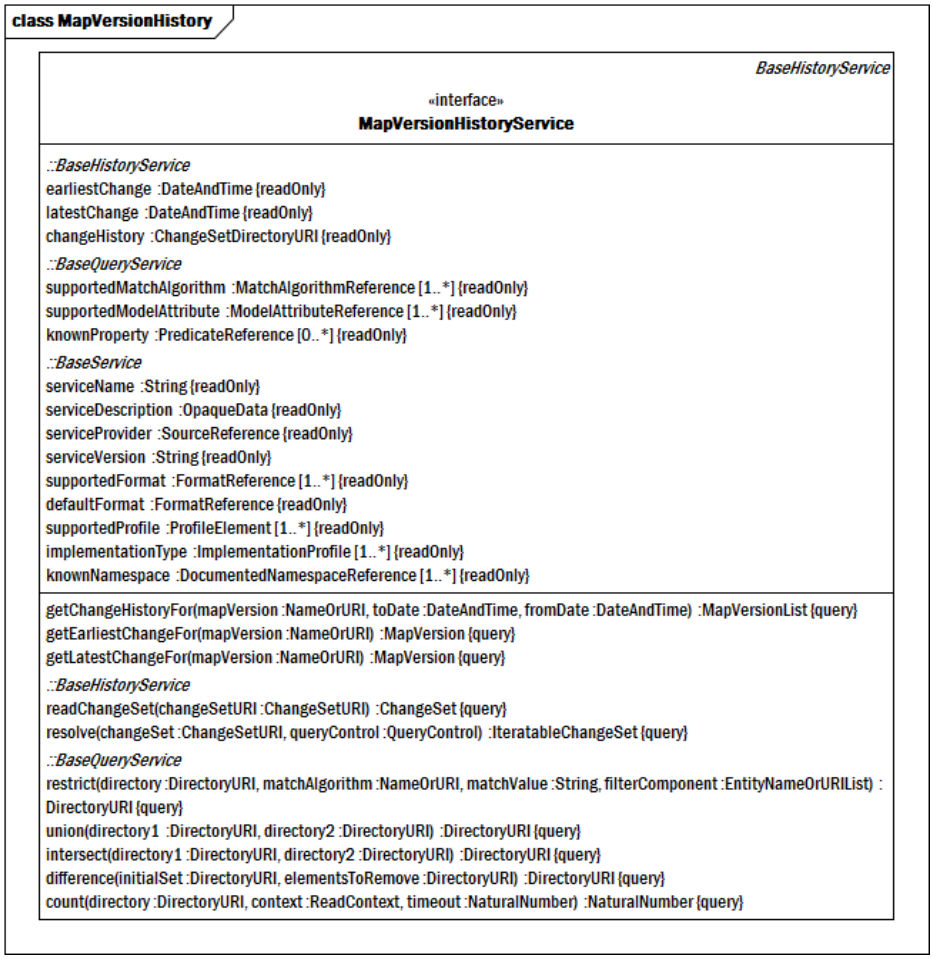

Figure 5.3 - Map Version History Service

The `MapVersionHistoryService` allows the client to find the first change, last change, and the set of changes that occurred to a `MapVersionEntry` over an optional range of time.

5.3.1 Interface MapVersionHistoryService

A service that provides a historical perspective on the changes that occurred in a map catalog over time.

Superclasses

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

5.3.1.1 Operation: getChangeHistoryFor

Return a directory of `MapVersions` that represent the incremental history of the named map version in this service.

Input Parameters

- **mapVersion** - the name or URI of the map version being queried (Type: `NameOrURI`).
- **toDate**$_{OPT}$ - the starting date. If later than `toDate`, the return order is latest to earliest, otherwise it is earliest to latest. (Type: `DateAndTime`)
- **fromDate**$_{OPT}$ - the starting date. If later than `toDate`, the return order is earliest to latest, otherwise it is latest to earliest. (Type: `DateAndTime`)

Return Type: `MapVersionList`

Exceptions

- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.

5.3.1.2 Operation: getEarliestChangeFor

Return the initial state of a map version from the perspective of the service.

Input Parameters

- **mapVersion** - the name or URI of the map version (Type: `NameOrURI`).

Return Type: `MapVersion`

Exceptions

- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.

5.3.1.3 Operation: getLatestChangeFor

Return the current state of a map version from the perspective of the service.

Input Parameters

- **mapVersion** - the name or URI of the map version (Type: `NameOrURI`).

Return Type: `MapVersion`

Exceptions

- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.

5.4 Map Version Maintenance Services

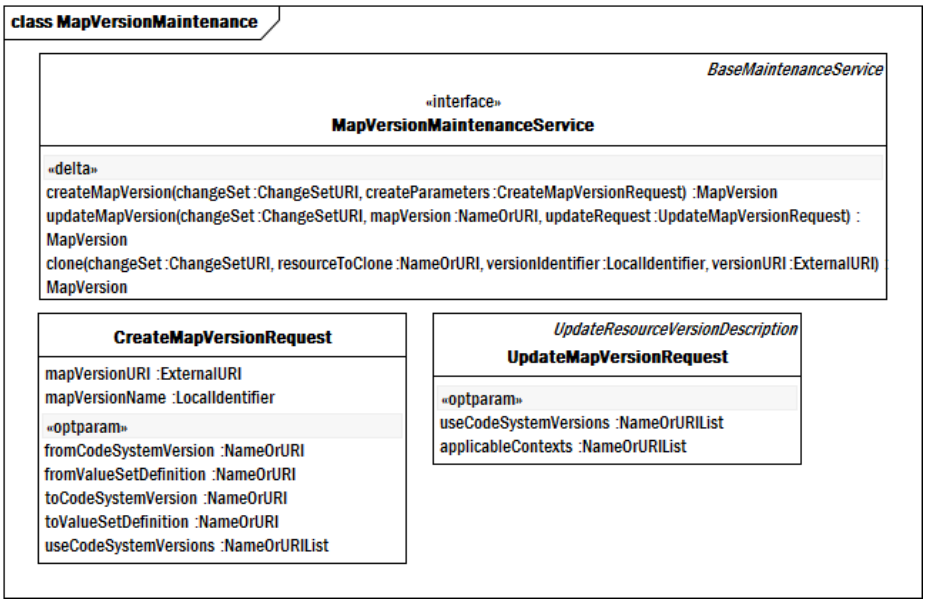

Figure 5.4 - Map Version Maintenance Service

The `MapVersionMaintenanceService` provides the ability to create and modify metadata about map versions.

5.4.1 Class CreateMapVersionRequest

The set of parameters that represent the identity of a `MapVersion`.

Attributes

- **mapVersionURI** - a `DocumentURI` that uniquely identifies the map version.
- **mapVersionName** - an identifier that uniquely names the map version within the context of the service.
- **fromCodeSystemVersion** - the name or URI of the from code system version. Exactly one of `fromCodeSystemVersion` or `fromValueSetDefinition` must be supplied.
- **fromValueSetDefinition** - the name or URI of the from value set definition. Exactly one of `fromCodeSystemVersion` or `fromValueSetDefinition` must be supplied.
- **to CodeSystemVersion** - the name or URI of the from code system version. Exactly one of to `CodeSystemVersion` or `toValueSetDefinition` must be supplied.
- **to ValueSetDefinition** - the name or URI of the “to” value set definition. Exactly one of to `CodeSystemVersion` or `toValueSetDefinition` must be supplied.
- **useCodeSystemVersions** - the names or URIs of the code system versions to be used in the resolution of the value sets.

5.4.2 Interface MapVersionMaintenanceService

Interface for creating and updating the descriptions of specific versions of maps. Note that the attributes and methods that were inherited from `BaseMaintenanceService` have been hidden for the sake of clarity.

5.4.2.1 Superclasses

- Every instance of `MapVersionMaintenanceService` is also an instance of `BaseMaintenanceService`.

5.4.2.2 Operation: createMapVersion

Create a new map version 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`).
- **createParameters** - the creation parameters (Type: `CreateMapVersionRequest`).

Return Type: `MapVersion`

Exceptions

- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **MapVersionToReferenceException** - Exactly one of `toCodeSystemVersion` and `toValueSetDefinition` must be supplied in a map version.
- **DuplicateMapVersionName** - The `mapVersionName` already exists in the catalog.
- **UnknownValueSetDefinition** - The `valueSetDefinition` URI isn't recognized by the service.
- **MapVersionFromReferenceException** - Exactly one of `fromCodeSystemVersion` and `fromValueSetDefinition` must be supplied in a map version.
- **AlternateURINotUnique** - One of the supplied `alternateIDs` is not unique - it appears as an `entryID` or `alternateID` for a different resource of the same type.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.
- **ResourceNameAlreadyExists** - The local identifier for the resource already exists in the service.
- **ResourceURIAlreadyExists** - The URI (`entityID`) of the supplied resource already exists in the service.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **UnknownCodeSystemVersion** - The referenced `CodeSystemVersion` is not recognized by the service.

5.4.2.3 Operation: updateMapVersion

Modify the attributes of an existing map version.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - The identifier of an OPEN change set that will record the change. (Type: `ChangeSetURI`)
- **mapVersion** - The name or URI of the map version to be updated. (Type: `NameOrURI`)
- **updateRequest** - The parameters to update. (Type: `UpdateMapVersionRequest`)

Return Type: `MapVersion`

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.
- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.
- **UnknownCodeSystemVersion** - The referenced `CodeSystemVersion` is not recognized by the service.
- **UnsupportedContext** - One or more `changeSetContext` is not supported by the service.
- **ResourceIsNotOpen** - The target resource version description has been finalized and cannot be updated.
- **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.
- **CycleInPredecessor** - The predecessor resource directly or indirectly precedes the resource being updated.
- **WrongPredecessorType** - The predecessor URI is not of the same type as its successor.

5.4.2.4 Operation: clone

Create an copy of the existing `MapVersion` and all associated `MapEntries` assigning a new version identifier and aboutURI. The cloned resource will be marked as `OPEN` , allowing incremental changes to occur.

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`)
- **resourceToClone** - the name or about URI of the MapVersion to be cloned (Type: `NameOrURI`)
- **versionIdentifier**$_{OPT}$ - the local identifier of the new resource version. If absent, the system is expected to assign the identifier. (Type: `LocalIdentifier`)
- **versionURI**$_{OPT}$ - the about URI of the new cloned resource. If absent, it is assumed that the service will generate the new URI. (Type: `ExternalURI`)

Return Type: `MapVersion`

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.
- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.

5.4.3 Class UpdateMapVersionRequest

The set of properties that can be modified in an existing `MapVersion`.

Superclasses

- Every instance of `UpdateMapVersionRequest` is also an instance of `UpdateResourceVersionDescription`.

Attributes

- **useCodeSystemVersions** - the names or URIs of the code system versions to be used in the resolution of the value sets.
- **applicableContexts** - the contexts to be used in the resolution of the value sets

6 Map Entry Services

6.1 Map Entry Read Services

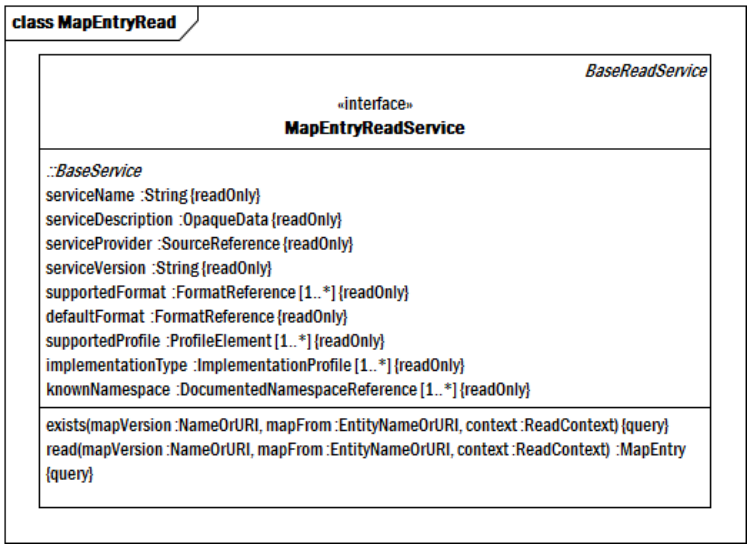

Figure 6.1 - Map Entry Read

The `MapEntryReadService` provides direct access to the map entries in a `MapVersion` via the unique `mapFrom` identifier.

6.1.1 Interface MapEntryReadService

A service providing direct access to entries in map versions.

Superclasses

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

6.1.1.1 Operation: exists

Determine whether a `MapEntry` exists for the supplied entity.

- **mapVersion** - the name or URI of the map version that defines the desired entry (Type: `NameOrURI`).
- **mapFrom** - the namespace/name or URI of the "from" entity in the entry (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.

- **UnknownMapVersion** - The named `mapVersion` 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.

6.1.1.2 Operation: read

Read the `MapEntry` that corresponds to the supplied `sourceEntity`.

Input Parameters

- **mapVersion** - the name or URI of the map version that defines the desired entry (Type: `NameOrURI`).
- **mapFrom** - the namespace/name or URI of the "from" entity in the entry (Type: `EntityNameOrURI`).
- **context**$_{OPT}$ - parameters that control the language, date, and other contextual variables (Type: `ReadContext`).

Return Type: `MapEntry`

Exceptions

- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnknownEntity** - The `EntityNameOrURI` is not known to the service.
- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.
- **FromEntryNotInMap** - The supplied `fromEntry` is not associated with any `MapEntry` in the referenced `MapVersion`.
- **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.

6.2 Map Entry Query Services

.

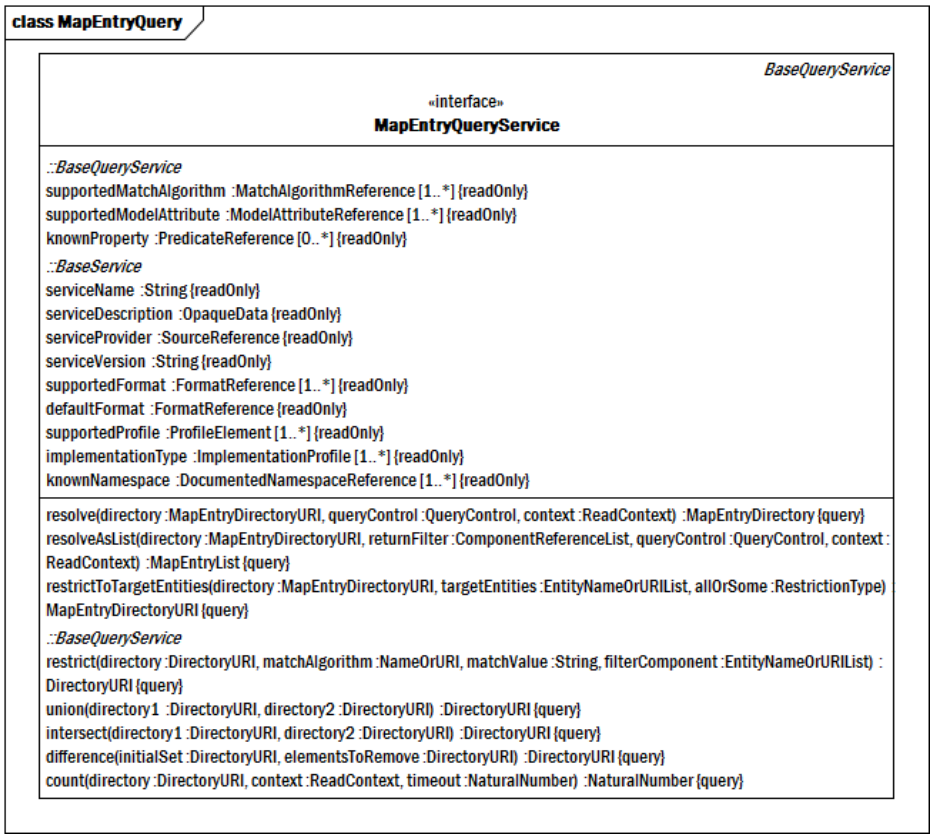

Figure 6.2 - Map Entry Query

The `MapEntryQueryService` allows sets of map entries to be filtered and resolved.

6.2.1 Interface MapEntryQueryService

Services that allow the filtering and query of entries in a `MapVersion`. These services allow the filtering of entries by rule and target content.

Superclasses

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

6.2.1.1 Operation: resolve

Return a directory containing a summary of the map entries referenced by directory that meet any additional requirements from context.

- **directory** - a `DirectoryURI` that resolves to a set of map entries (Type: `MapEntryDirectoryURI`).
- **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: `MapEntryDirectory`

6.2.1.2 Operation: resolveAsList

Return a list of the entries referenced by directory that meet any additional requirements from context.

Input Parameters

- **directory** - a `DirectoryURI` that resolves to a set of map entries (Type: `MapEntryDirectoryURI`).
- **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: `MapEntryList`

6.2.1.3 Operation: restrictToTargetEntities

A `DirectoryURI` that references the subset of directory that references all (`allOrSome` = `ALL`) or at least one (`allOrSome=AT_LEAST_ONE`) as a target.

Input Parameters

- **directory** - a `DirectoryURI` that references a set of map entries from the same `MapVersion` (Type: `MapEntryDirectoryURI`).
- **targetEntities** - a list of the filter target entities (Type: `EntityNameOrURIList`).
- **allOrSome** - an indicator that states whether the `MapEntry` must reference all of the target entities in the list or just one (Type: `RestrictionType`).

Return Type: `MapEntryDirectoryURI`

Exceptions

- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnknownEntity** - The `EntityNameOrURI` is not known to 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.

6.3 Map Entry History Service

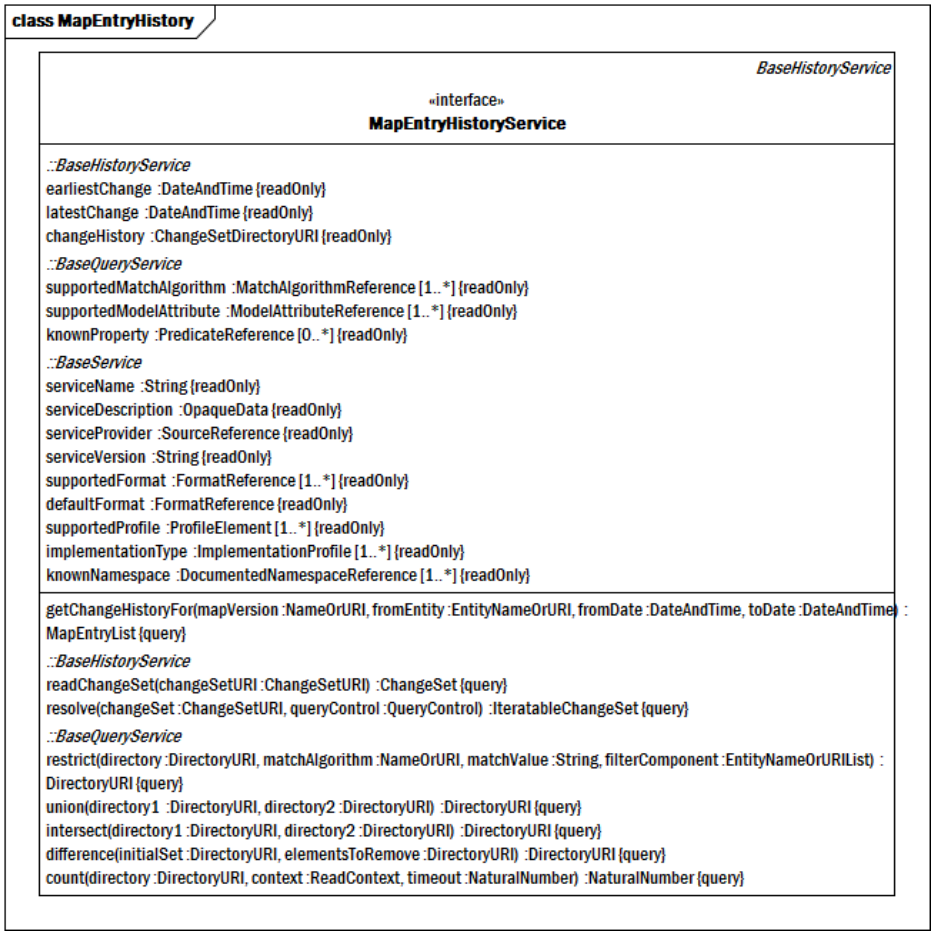

Figure 6.3 - Map Entry History

The `MapEntryHistory` service allows the set of changes that an individual map entry has undergone to be viewed through time.

6.3.1 Interface MapEntryHistoryService

A service that allows the evolution of a map entry to be retrieved and viewed.

Superclasses

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

6.3.1.1 Operation: getChangeHistoryFor

Return an ordered list of map entries that represent the various states that the named entry went through over time.

Input Parameters

- **mapVersion** - the name or URI of the continuing map version (Type: `NameOrURI`).
- **fromEntity** - the target entity to retrieve the history for (Type: `EntityNameOrURI`).
- **fromDate**$_{OPT}$ - the earliest date to select changes for. If not supplied, all known changes up to `toDate` will be returned (Type: `DateAndTime`).
- **toDate**$_{OPT}$ - the latest date to list changes for. If not supplied all changes through the latest will be returned. (Type: `DateAndTime`).

Return Type: `MapEntryList`

Exceptions

- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.
- **UnknownEntity** - The `EntityNameOrURI` is not known to the service.
- **FromEntryNotInMap** - The supplied `fromEntry` is not associated with any `MapEntry` in the referenced `MapVersion`.

6.4 Map Entry Maintenance Services

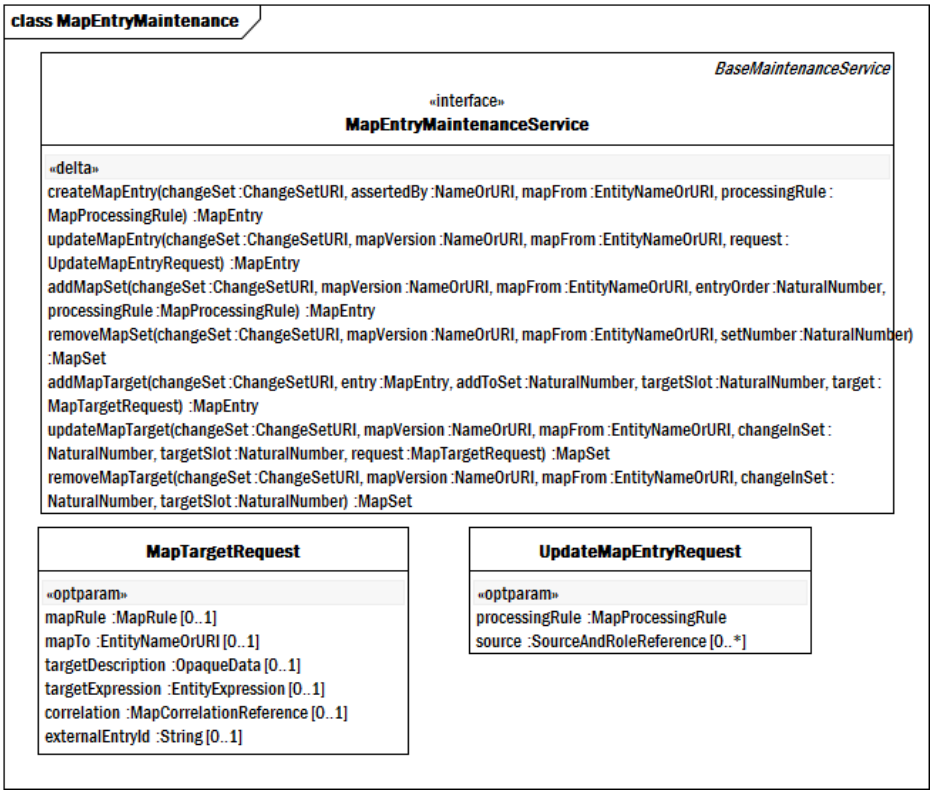

Figure 6.4 - Map Entry Maintenance

6.4.1 Interface MapEntryMaintenanceService

A service that allows the creation, update, and deletion of `MapEntries` within a `MapVersion`.

Superclasses

- Every instance of `MapEntryMaintenanceService` is also an instance of `BaseMaintenanceService`.

6.4.1.1 Operation: createMapEntry

Create a new map 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`).
- **assertedBy** - the name or URI of the `MapVersion` that contains this assertion (Type: `NameOrURI`).
- **mapFrom** - the name or URI of the map entry to map from (Type: `EntityNameOrURI`).
- **processingRule** - an indicator that states whether all map sets will be evaluated or processing will stop on the first hit (Type: `MapProcessingRule`).

Return Type: `MapEntry`

Exceptions

- **DuplicateMapEntry** - The `mapFrom` entity already exists.
- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.
- **FromEntryNotInMap** - The supplied `fromEntry` is not associated with any `MapEntry` in the referenced `MapVersion`.
- **ChangeSetIsNotOpen** - The `changeSetContext` is recognized by the service, but its state is not `OPEN`.
- **UnsupportedNamespaceName** - The supplied namespace name is not one that is known to the service.
- **UnknownEntity** - The `EntityNameOrURI` is not known to the service.
- **UnknownChangeSet** - The change set specified could either not be read or located by the service.

6.4.1.2 Operation: updateMapEntry

Update the attributes of the supplied map entry and return the result of the change.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the URI of an open change set that will record the operation (Type: `ChangeSetURI`).
- **mapVersion** - the name or URI of the containing `MapVersion` (Type: `NameOrURI`).

- **mapFrom** - the name or URI of the entity that uniquely identifies the `MapEntry` (Type: `EntityNameOrURI`).
- **request** - the set of fields to update (Type: `UpdateMapEntryRequest`).

Return Type: `MapEntry`

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.
- **FromEntityNotValid** - The supplied `mapFrom` entry is not a valid entity in the containing map's `fromCodeSystem` or `fromValueSet`.
- **UnsupportedSource** - The supplied source is not recognized by the service.
- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.
- **FromEntryNotInMap** - The supplied `fromEntry` is not associated with any `MapEntry` in the referenced `MapVersion`.
- **UnsupportedRole** - The role name or URI is not recognized by the service.

6.4.1.3 Operation: addMapSet

Add the described `MapSet` to the input entry and return the result of the addition.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the URI of an OPEN change set to that will record this change (Type: `ChangeSetURI`).
- **mapVersion** - the name or URI of the `MapVersion` that contains this entry (Type: `NameOrURI`).
- **mapFrom** - the name or uri that identifies the specific entry (Type: `EntityNameOrURI`).
- **entryOrder**$_{OPT}$ - the relative order of the entry within the existing list of set, where "1" goes into the first slot, "2" into the second, etc. "0" or an omitted entry goes at the end of the list. (Type: `NaturalNumber`)
- **processingRule** - an indicator that states how the targets will be processed. (Type: `MapProcessingRule`)

Return Type: `MapEntry`

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.
- **FromEntityNotValid** - The supplied `mapFrom` entry is not a valid entity in the containing map's `fromCodeSystem` or `fromValueSet`.
- **UnsupportedSource** - The supplied source is not recognized by the service.
- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.

6.4.1.4 Operation: removeMapSet

Remove the referenced `MapSet` and any contained `MapTargets`.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the URI of an OPEN change set to that will record this change (Type: `ChangeSetURI`).
- **mapVersion** - the name or URI of the `MapVersion` that contains the entry (Type: `NameOrURI`).
- **mapFrom** - the entity name or URI that identifies the particular entry (Type: `EntityNameOrURI`).
- **setNumber** - the one based identifier of the map set to change (Type: `NaturalNumber`).

Return Type: `MapSet`

Exceptions

- **IllegalEntryOrder** - The `entryOrder` does not reference an existing `MapEntry`.
- **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.
- **FromEntityNotValid** - The supplied `mapFrom` entry is not a valid entity in the containing map's `fromCodeSystem` or `fromValueSet`.
- **UnsupportedSource** - The supplied source is not recognized by the service.
- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.

6.4.1.5 Operation: addMapTarget

Add the supplied map target in one-based target slot (1 means first entry, 2 second, etc. 0 or missing means at end).

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the URI of an OPEN change set to that will record this change (Type: `ChangeSetURI`).
- **entry** - the existing `MapEntry` to add the target to (Type: `MapEntry`).
- **addToSet** - the one based identifier of the particular `MapSet` to add the target to (Type: `NaturalNumber`).
- **targetSlot**$_{OPT}$ - the one-based slot number to add the new target to (Type: `NaturalNumber`).
- **target** - the parameters of the target to add (Type: `MapTargetRequest`).

Return Type: `MapEntry`

Exceptions

- **InvalidMapRule** - The supplied map rule is not recognized by the supporting tooling.
- **IllegalTargetSlot** - The referenced target slot is not valid in the containing map set.

- **IllegalEntryOrder** - The `entryOrder` does not reference an existing `MapEntry`.
- **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.
- **FromEntityNotValid** - The supplied `mapFrom` entry is not a valid entity in the containing map's `fromCodeSystem` or fromValue Set.
- **UnsupportedSource** - The supplied source is not recognized by the service.
- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.
- **UnsupportedMapCorrelation** - The `mapCorrelation` is not recognized by the service.
- **ToEntityNotValid** - The `mapTo` entity is not part of the to `CodeSystemVersion` or `toValueSet` in the map.

6.4.1.6 Operation: updateMapTarget

Update the referenced map target and return the result of the changes.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the URI of an OPEN change set to that will record this change (Type: `ChangeSetURI`).
- **mapVersion** - the name or URI of the containing MapVersion (Type: `NameOrURI`).
- **mapFrom** - the name/name or URI of the entity that uniquely identifies the entry to be updated (Type: `EntityNameOrURI`).
- **changeInSet** - the one-based identifier of the map set to change (Type: `NaturalNumber`).
- **targetSlot** - the one-based target slot number (Type: `NaturalNumber`).
- **request** - the update request (Type: `MapTargetRequest`).

Return Type: `MapSet`

Exceptions

- **InvalidMapRule** - The supplied map rule is not recognized by the supporting tooling.
- **IllegalTargetSlot** - The referenced target slot is not valid in the containing map set.
- **IllegalEntryOrder** - The `entryOrder` does not reference an existing `MapEntry`.
- **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.
- **FromEntityNotValid** - The supplied `mapFrom` entry is not a valid entity in the containing map's `fromCodeSystem` or `fromValueSet`.
- **UnsupportedSource** - The supplied source is not recognized by the service.
- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.

- **UnsupportedMapCorrelation** - The `mapCorrelation` is not recognized by the service.
- **ToEntityNotValid** - The `mapTo` entity is not part of the `toCodeSystemVersion` or `toValueSet` in the map.

6.4.1.7 Operation: removeMapTarget

Remove the referenced map target.

This operation may alter the state of the service.

Input Parameters

- **changeSet** - the URI of an OPEN change set to that will record this change (Type: `ChangeSetURI`).
- **mapVersion** - the name or URI of the containing `MapVersion` (Type: `NameOrURI`).
- **mapFrom** - the namespace/name or URI of the entity that identifies the `MapSet` (Type: `EntityNameOrURI`).
- **changeInSet** - the one-based identifier of the map set to change (Type: `NaturalNumber`).
- **targetSlot** - the one-based target slot number (Type: `NaturalNumber`).

Return Type: `MapSet`

Exceptions

- **IllegalTargetSlot** - The referenced target slot is not valid in the containing map set.
- **IllegalEntryOrder** - The `entryOrder` does not reference an existing `MapEntry`.
- **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.
- **FromEntityNotValid** - The supplied `mapFrom` entry is not a valid entity in the containing map's `fromCodeSystem` or `fromValueSet`.
- **UnsupportedSource** - The supplied source is not recognized by the service.
- **UnknownMapVersion** - The named `mapVersion` is not recognized by the service.

6.4.2 Class MapTargetRequest

A set of parameters that are used to create or update a map target.

Attributes

- **mapRule** - the map rule for the target. If omitted, the rule always evaluates to TRUE.
- **mapTo** - the entity name or URI to map to.
- **targetDescription** - the description of the rule and/or target.
- **targetExpression** - an expression that, when evaluated, produces a complex target.
- **correlation** - the name or URI of a correlation factor.
- **externalEntryId** - an external identifier assigned to this target entry by an outside body.

6.4.3 Class UpdateMapEntryRequest

The set of parameters that can be updated in a map entry.

Attributes

- **processingRule** - the rule for processing the member sets.
- **source** - a source and the role they played in creating the entry.

7 Map Resolution Service

7.1 Map Resolution Services

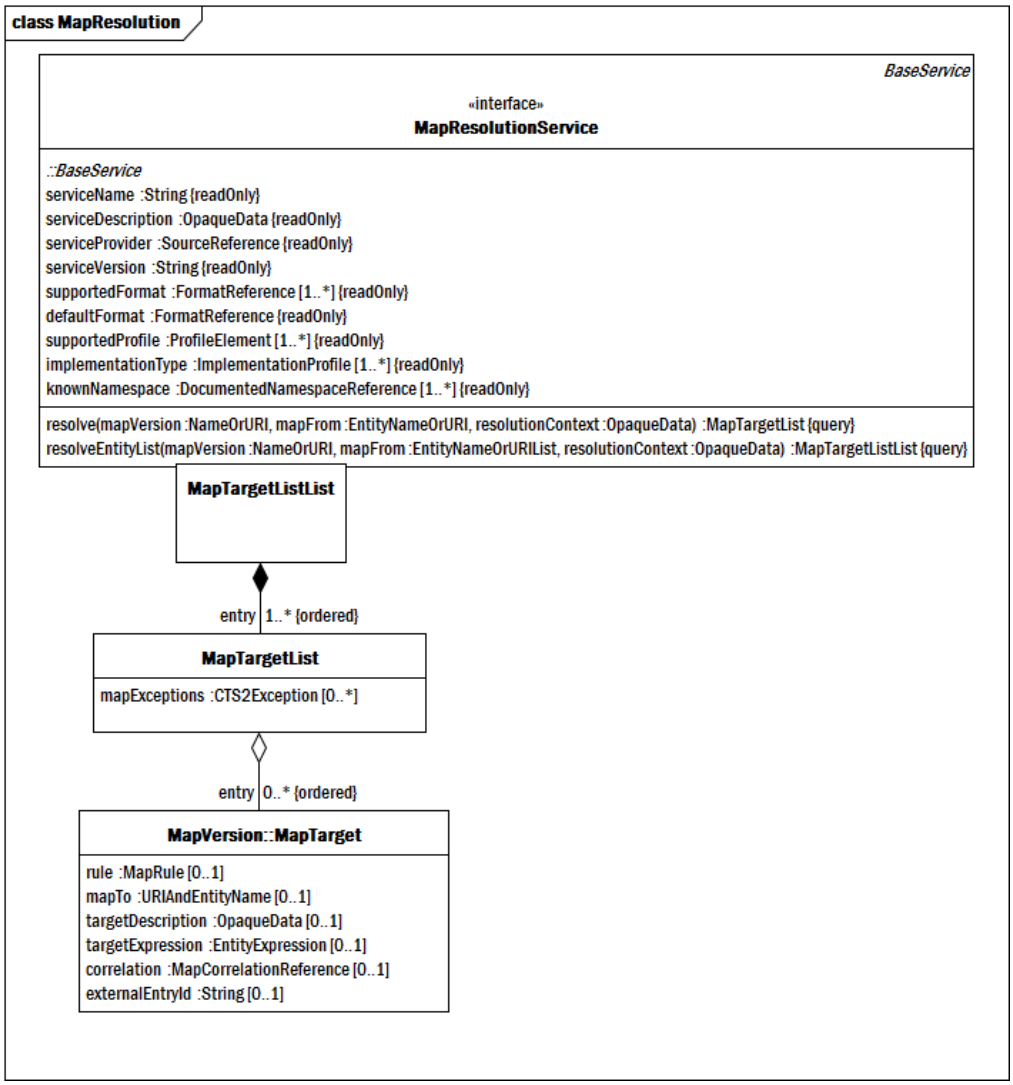

Figure 7.1 - Map Resolution Service

The `MapResolutionService` provides a basic interface for mapping a single entity reference or set of entity references into the corresponding target(s). There are two primary methods - one that takes the name of a map, a single input parameter and an arbitrary block of context and returns an ordered set of map targets and any accompanying exceptions. The second takes a set of entities and performs the same action.

7.1.1 Interface MapResolutionService

A service that interprets and resolves maps.

Superclasses

- Every instance of `MapResolutionService` is also an instance of `BaseService`.

7.1.1.1 Operation: resolve

Evaluate the supplied entity in the context of the map version and return the resulting set of map targets and/or exceptions.

Input Parameters

- **mapVersion** - the name or URI of map version to resolve against (Type: `NameOrURI`).
- **mapFrom** - the namespace/name or URI of the entity to map (Type: `EntityNameOrURI`).
- **resolutionContext** - additional context that is needed for the particular type of resolution (Type: `OpaqueData`).

Return Type: `MapTargetList`

7.1.1.2 Operation: resolveEntityList

Evaluate the supplied list of input entities and return one `MapTargetList` entry per name in the list.

Input Parameters

- **mapVersion** - the name or URI of the map version to resolve against (Type: `NameOrURI`).
- **mapFrom** - the namespace/name or URI of the entity to resolve (Type: `EntityNameOrURIList`).
- **resolutionContext** - additional context that is needed for the particular type of resolution (Type: `OpaqueData`).

Return Type: `MapTargetListList`

7.1.2 Class MapTarget

A rule / target entry in a `MapSet`. The entries in a `MapSet` are evaluated in entry order. For each entry, rule is evaluated. If rule returns FALSE, then evaluation moves on to the next `MapTarget` in the set. If rule returns TRUE, the `MapTarget` is returned as a qualifying map in the associated `MapSet` and corresponding `MapEntry`. The setting of `processingRule` in the `MapSet` determines whether subsequent `MapTargets` are evaluated. If `processingRule` is set to `FIRST_MATCH`, the first matching `MapTarget` terminates the evaluation of the `MapSet`. If `processingRule` is set to `ALL_MATCHES`, then processing continues until all `MapTargets` in the set have been evaluated.

Attributes

- **rule** - a predicate of some sort that can be used to determine whether this entry meets the mapping criteria. If absent, the mapping is considered unconditional.
- **mapTo** - the target entity to be used if the rule is satisfied. If not present, the assumption is that the source entity maps to nothing in this set / rule.

- **targetDescription** - a description of the resulting rule. Can be used to carry text to instruct human beings how to do the mapping as well as additional machine readable instructions.
- **targetExpression** - a formal expression that, when interpreted, results in the map target. This expression could carry an RDF fragment, an expression in the SNOMED-CT compositional grammar or something else.
- **correlation** - a reference to how well the mapping correlates. Example correlations might be Exact, not mappable, partial overlap, narrower than, etc.
- **externalEntryId** - an identifier assigned to this particular source/set/target tuple by an outside body.

7.1.3 Class MapTargetList

An ordered list of map targets.

Attributes

- **mapExceptions** - A list of exceptions that were encountered while processing the rule. This may include issues with the rule format, undefined targets, etc.
- **entry** - an entry in a map target list.

7.1.4 Class MapTargetListList

A list of map target lists, one per input entity.

Attributes

- **entry** - an entry in a list of map target lists.