DDI-Views Q2 Review

Release Development Draft Q2

DDI Alliance

Dec 16, 2016

TABLE OF CONTENTS

1 About **3**
- 1.1 Copyright & Licence . . . 3
- 1.2 Contacting Us . . . 3
- 1.3 DDI Versions . . . 4

2 Preface **5**
- 2.1 Development of the Model . . . 5
- 2.2 Funding . . . 5
- 2.3 Further Information . . . 6
- 2.4 Organization of this Document . . . 6
- 2.5 Acknowledgements . . . 6

3 Associated Information and Files **9**

4 The model and its development **11**
- 4.1 Description of the Model . . . 11
- 4.2 Production Framework . . . 17
- 4.3 Design Principles . . . 19
- 4.4 DDI Base Blocks . . . 20

5 User Guides **23**
- 5.1 Using the Collection pattern . . . 23
- 5.2 Using the Process pattern . . . 30
- 5.3 The Variable Cascade . . . 36
- 5.4 Using the Signification pattern . . . 40
- 5.5 A Guide to Binary Relations . . . 41

6 Packages **47**
- 6.1 ComplexDataTypes . . . 47
- 6.2 Conceptual . . . 129
- 6.3 CustomMetadata . . . 159
- 6.4 ProcessPattern . . . 198
- 6.5 MethodologyPattern . . . 204
- 6.6 SegmentLocationSpecifications . . . 214
- 6.7 FormatDescription . . . 230
- 6.8 SignificationPattern . . . 248
- 6.9 Methodologies . . . 252
- 6.10 Representations . . . 259
- 6.11 Workflows . . . 305
- 6.12 LogicalDataDescription . . . 325
- 6.13 Primitives . . . 347

6.14 Discovery . . . 348
6.15 Comparison . . . 362
6.16 CollectionsPattern . . . 362
6.17 BaseObjects . . . 381
6.18 Correspondences . . . 382
6.19 DataCapture . . . 382
6.20 Agents . . . 400
6.21 DDIUtility . . . 418
6.22 DDIDocument . . . 419
6.23 Identification . . . 419
6.24 ComplexProcess . . . 423
6.25 Utility . . . 446

7 Views **455**

7.1 StatisticalClassificationView . . . 455
7.2 InstrumentView . . . 495
7.3 DiscoveryView . . . 544
7.4 DataDictionaryView . . . 562
7.5 AgentRegistryView . . . 618

8 Glossary **645**

DATA DOCUMENTATION INITIATIVE

Warning: this is a development build, not a final product.

CHAPTER

ONE

ABOUT

1.1 Copyright & Licence

1.1.1 Licence

The DDI-Views Specification (Development Draft) us free software you can distribute it and/or modify it under the terms of the Creative Commons Attribution 4.0 International (CC BY 4.0) licence,

This is a human-readable summary of (and not a substitute for) the license

You are free to:

> Share - copy and redistribute the material in any medium or format Adapt - remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

1.1.2 Attribution

You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. No additional restrictions. You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

1.1.3 Notices

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation. No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

1.2 Contacting Us

General information on the review can be found on the Review pages at https://ddi-alliance.atlassian.net/wiki/display/DDI4/How+to+Review+the+2016+Q2+Development+Draft

Comments and questions on the content of this document and the review process more generally should be directed to the Technical Committee list <mailto:ddi-srg@icpsr.umich.edu> or the Technical Committee Chair, Wendy Thomas wlt@umn.edu.

1.3 DDI Versions

Published versions of DDI

Version	Year published	Documentation link
2.1	2003	http://www.ddialliance.org/Specification/DDI-Codebook/2.1/DTD/Documentation/DDI2-1-tree.html
2.5	2012	http://www.ddialliance.org/Specification/DDI-Codebook/2.5/XMLSchema/field_level_documentation.html
3.0	2008	http://www.ddialliance.org/Specification/DDI-Lifecycle/3.0/XMLSchema/Documentation/
3.1	2009	http://www.ddialliance.org/Specification/DDI-Lifecycle/3.1/XMLSchema/FieldLevelDocumentation/
3.2	2012	http://www.ddialliance.org/Specification/DDI-Lifecycle/3.2/XMLSchema/FieldLevelDocumentation/

CHAPTER

TWO

PREFACE

The intent of the DDI-Views documentation is to provide information on the individual classes used in the model; their relationship to each other, and their relationship to DDI Lifecycle 3.2 and other standards such as the General Statistical Information Model.

2.1 Development of the Model

The development of the model has been underway since October 2012 when a first workshop was held at Schloss Dagstuhl. The work has been carried forward subsequently through a series of Sprints, virtual meetings and online discussions.

- Sprint 1: Schloss Dagstuhl, October 28-November 1, 2013
- Sprint 2: Paris, December 6-7, 2013
- Sprint 3: Vancouver, March 24-28, 2014
- Sprint 4: Toronto, May 26-30, 2014
- Sprint 5: Schloss Dagstuhl, October 20-24, 2014
- Sprint 6: London, November 24 - 28, 2014
- Sprint 7: Minneapolis, May 25 - 30 2015
- Sprint 8: Schloss Dagstuhl, October 19 - 23 2015
- Sprint 9: Copenhagen, December 23 - 27 2015
- Sprint 10: Edmonton, April 11 - 15 2016
- Sprint 11: Knutholmen, May 23 - 27 2016

2.2 Funding

Development of the model has been primarily supported by the DDI Alliance, and by in-kind contributions from participants in the effort. DDI Alliance members and their host institutions have generously provided venues and infrastructure for Sprints. The Schloss Dagstuhl sprints are part of a long running series of annual events hosted by the Leibniz Centre for Informatics. The October 2014 Sprint was funded in part by the National Science Foundation under award 1448107.

2.3 Further Information

Further information on the development of DDI-Views is available at the DDI Moving Forward Project site.

2.4 Organization of this Document

The introduction provides a high level view of the structure of the DDI-Views model, its architecture and the way in which the individual classes are organized. The introduction also provides information on the mechanisms by which the classes are utilized, the relationships among them, how the model can be extended and how it will be managed.

Each Library Package has a diagram showing the relationships between the classes within that Library Package. For each class, information is provided on whether it extends any class, and whether it is based on an existing class in DDI 3.2, GSIM or other standards.

- A definition of the class, its properties and relationship to other classes
- Examples of the usage of the class and explanatory notes
- Whether the class is abstract
- Whether the class can be extended
- Mapping to the General Statistical Information Model (GSIM) [http://www1.unece.org/stat/platform/display/gsim/Generic+Statistical+Information+Model]
- Mapping to DDI 3.2 [http://www.ddialliance.org/Specification/DDI-Lifecycle/3.2/]
- Mapping to RDF (pending)
- Mapping to Other standards (pending)

2.5 Acknowledgements

This document was edited by: Jon Johnson and Wendy Thomas

Participants at the Sprints and other contributors to the development of the model and documentation were:

- Alan Li
- Alejandra Gonzalez-Beltran
- Anders Swendsrud
- Anita Rocha
- Arofan Gregory
- Barry Radler
- Brigitte Mathiak
- Bryan Fitzpatrick
- Chris Seymour
- Chuck Humphrey
- Dan Smith
- Daniel Gillman

- Daniella Meeka
- David Barraclough
- Denis Grofils
- Dennis Wegener
- Elizabeth Hostetter
- Eric Prud'hommeaux
- Flavio Rizzolo
- Franck Cotton
- Gary Berg-Cross
- Guillaume Duffes
- Helen Toole
- Henrik Sejersen
- Herve L'Hours
- Hilde Orten
- Ingo Barkow
- Iris Alfredsson
- Jannik Jensen
- Jared Lyle
- Jay Greenfield
- Jean-Pierre Kent
- Jenny Linnerud
- Jeremy Iverson
- Joachim Wackerow
- Johan Fihn
- Johanna Vompras
- John Kunze
- Jon Johnson
- Jon Stiles
- Justin Lynch
- Katherine McNeill
- Kelly Chatain
- Knut Wenzig
- Larry Hoyle
- Marcel Hebing
- Mari Kleemola
- Martin Forsberg

- Martin Jensby
- Mary Vardigan
- Merja Karjalainen
- Michael Witt
- Michel Dumontier
- Michelle Edwards
- Oleg Volguine
- Oliver Hopt
- Olof Olsson
- Ornulf Risnes
- Richard Cyganiak
- Ron Nakao
- Sam Hume
- Sanda Ionescu
- Simon Lloyd
- Simon Wall
- Sophia Kuan
- Steve McEachern
- Stuart Weibel
- Therese Lalor
- Thomas Bosch
- Wendy Thomas
- William Block
- Wolfgang Zenk-Moltgen

CHAPTER THREE

ASSOCIATED INFORMATION AND FILES

In addition to the documentation of the DDI-Views model. There are other products included with the review documents: A fuller description of what they are in can be found in the Production Framework section of *The model and its development*.

- Platform Independent Model * xmi.xmi
- Platform Specific Model (psm folder of the associated files bundle) * ddi4psm.owl.xmi * ddi4psm.xsd.xmi
- Platform Specific Bindings * XML Schemas (XSD folder of the associated files bundle)
  - xxxView.xsd - Functional Views
  - DDI_4-DR0.2.xsd - (the complete model for packages being released)
  - Other files are supporting schemas

CHAPTER

FOUR

THE MODEL AND ITS DEVELOPMENT

4.1 Description of the Model

4.1.1 Introduction

The intent of DDI-View Specification Class Description is to provide information on the individual classes used in the model, their relationship to each other and their relationship to DDI Lifecycle 3.2 and other standards such as General Statistical Information Model (GSIM).

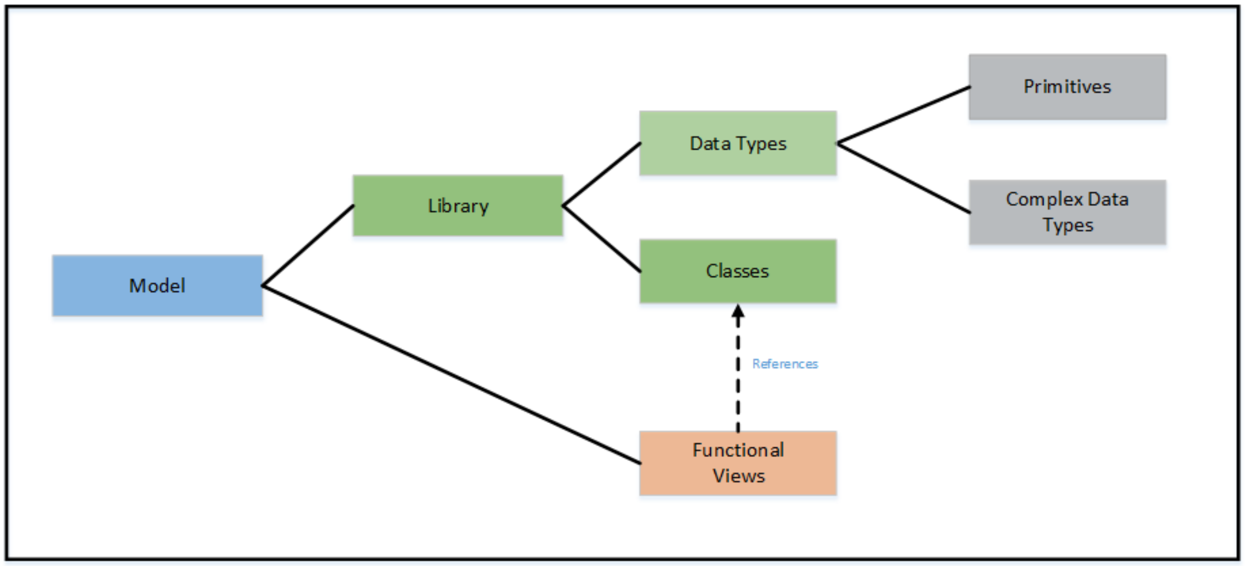
The model based DDI specification consists of two parts – a Library of classes and Functional Views of the model. The Library of classes encompasses the entire DDI-Lifecycle (MD) model. The classes in the Library are the building blocks used to construct the Functional Views. These Functional Views are in essence profiles of the full specification oriented around specific user needs. The model is primarily being developed and surfaced to the user community at http://lion.ddialliance.org/

A Note on Terminology During the development process, the terminology for what is now called classes (to reflect the language used in UML) was previously referred to as 'objects'

4.1.2 Structure of the DDI-Views Model

The model contains a Library and Functional Views. The Library is composed of Library Packages which contain other data types (primitives or complex) or classes. The Functional Views contain references to the classes used by the particular Functional View that are needed to meet the needs of the use case or business application.

Figure 1. DDI-Views Model and its components



4.1.3 Library of Classes

The Library of Classes encompasses the entire DDI-Views model, but without any specific schemas or vocabularies for Functional Views. The classes in the Library contain primitives and complex data types and are the building blocks used to construct the Functional Views. Classes are organized into Library Packages.

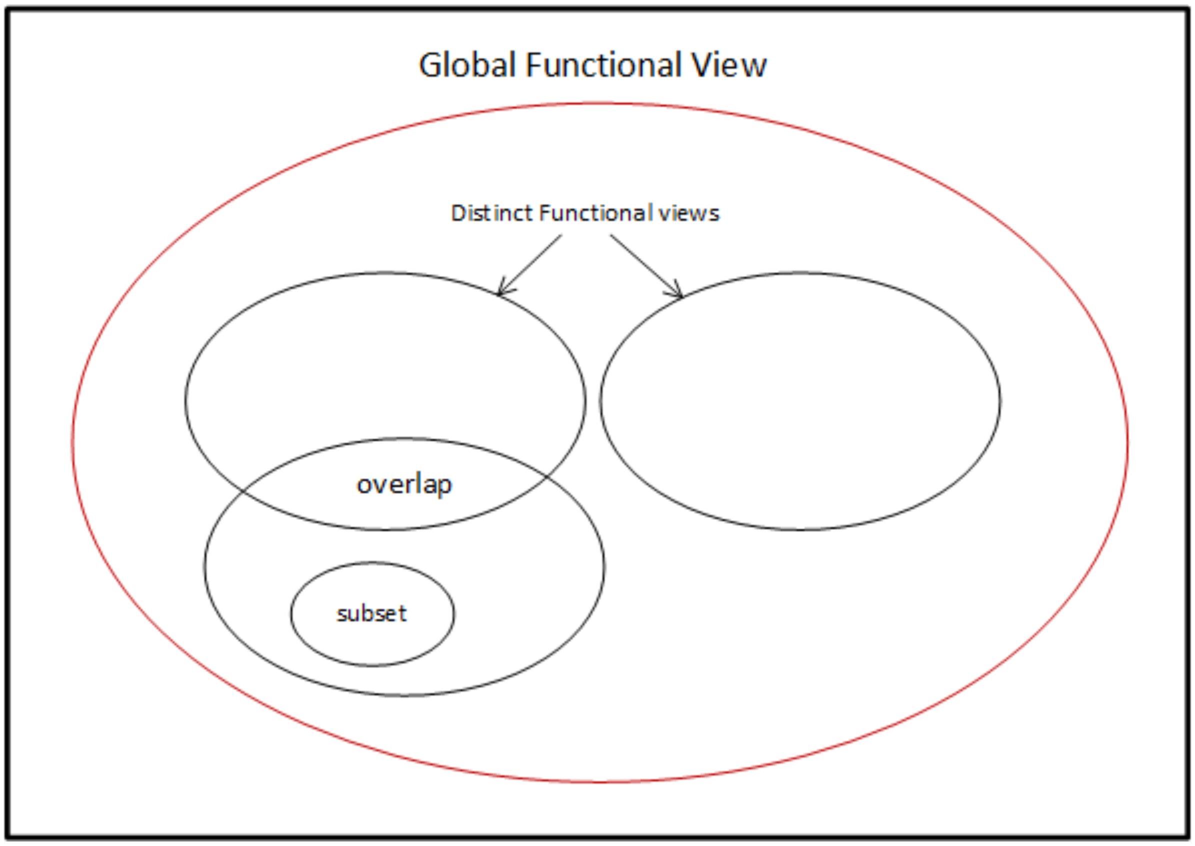
4.1.4 Functional Views

Functional Views are made up of a set of references to the classes in the Library. Functional Views are subsets of the model grouped to support a specific application (for example the description of a questionnaire). The Functional Views are divided into sections. Each section loosely corresponds to a DDI lifecycle business area. Within each business area section there are separate subsections for Functional Views and compositions. Note that Functional Views may include placeholders like an abstract class that need to be substituted before the Functional View can actually be used. Functional Views are always a strict subset of the existing published or (for customization) extended classes. A Functional View identifies a set of classes that are needed to perform a specific task. It primarily consists of a set of references to specific versions of classes. Functional Views are the method used to restrict the portions of the model that are used, and as such they function very much like DDI profiles in DDI 3.*. The subsetting with Functional Views is done on the model and not on the instance level as in DDI Profiles. One may

- restrict the use of non-mandatory properties on a class;
- restrict the cardinality of a class's relationships and properties;
- restrict the use of non-mandatory relationships.

Restrictions may never be made that would violate the mandatory inclusion of a relationship or property. Functional Views may combine classes from any package or set of packages needed. The creation of Functional Views thus has no dependency on the organization of metadata classes within the Library Packaging structure.

Figure 2. Interoperability of Functional Views



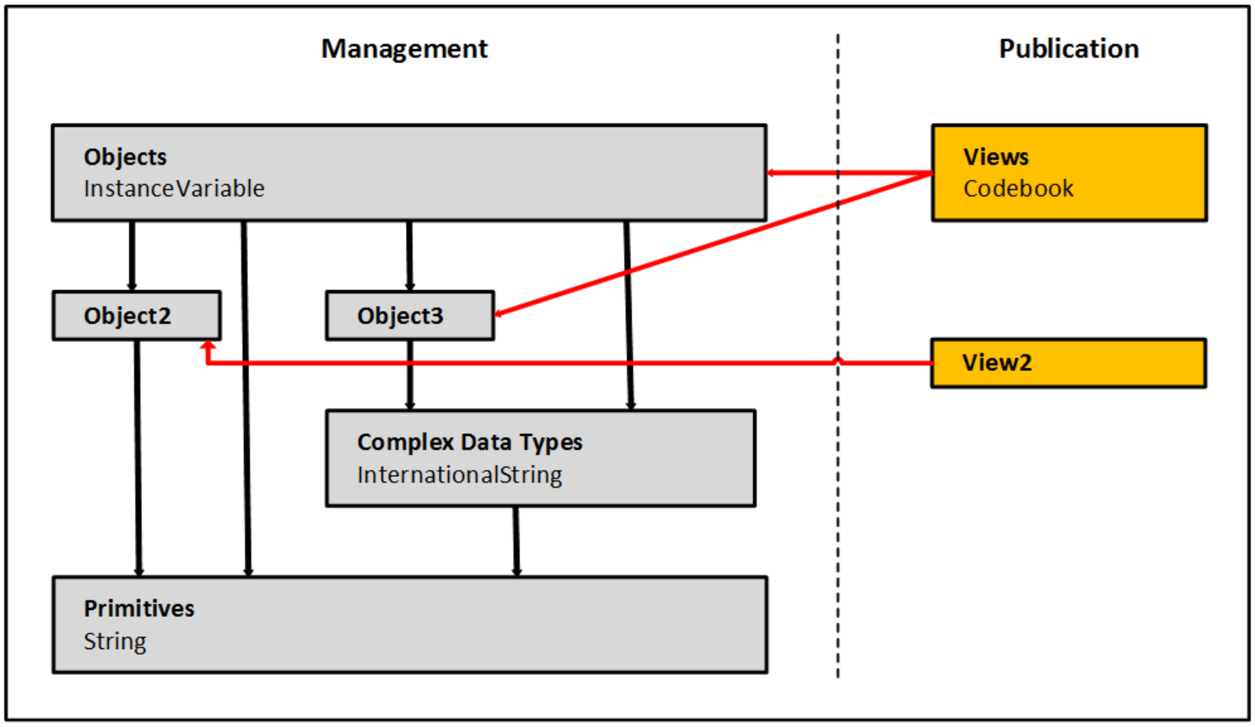
As shown in Figure 2, Each Functional View is a subset of the classes in the Library. Functional Views might be distinct, overlapping in their function or be a subset or superset of another Functional View. Interoperability between two Functional Views is only given for the Library classes which are used in both Functional Views.

A global Functional View could be created which comprehends all classes in the Library and their relationships. It represents all functionality of the class in the Library. Each Functional View would be interoperable to this global Functional View.

4.1.5 Model Constructs and Their Relationships

Figure 3 below shows the basic relationships between the types of constructs in the model. At the lowest level, we have the primitives. These are used directly by classes, and are also used to create complex data types. The complex data types are also used by classes. Classes themselves can relate to other classes, building increasingly complex structures. The classes – along with the primitives and complex data types – form the Class Library. Classes can relate to each other in two ways: a class may have a “direct” relationship (composition, aggregation) with another class, or it may have an inheritance relationship. In this latter case, the DDI model uses additive extension. One class may extend another by inheriting all of its properties and relationships, to which the new class may add additional properties and relationships. This mechanism is used to take more generic classes and alter them for a more specific purpose. Extension is explained more fully below.

Figure 3. DDI-Views Organization of the Model

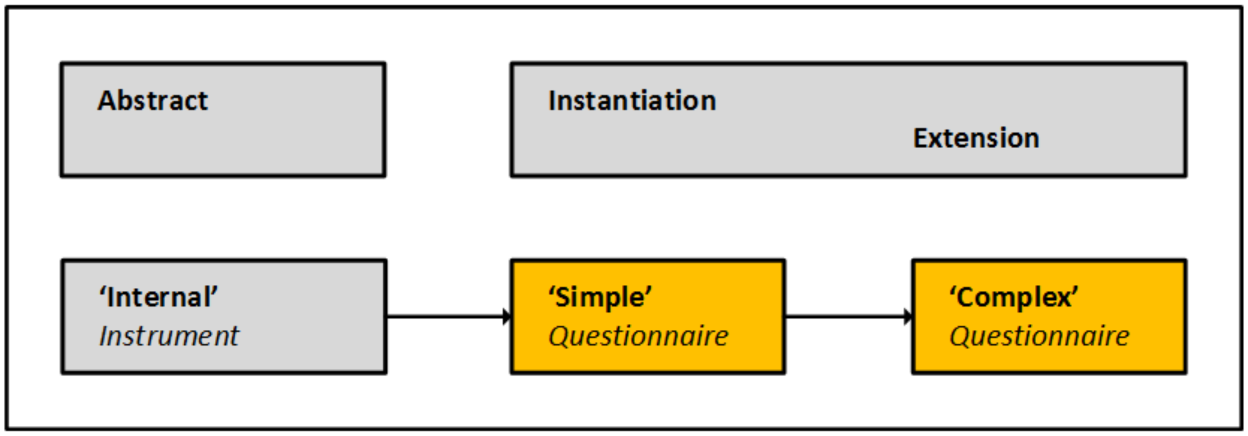


4.1.6 Extension

Extension is the inheritance of one class's properties and relationships from another class. It also has a semantic relationship – an extending class provides a specialized use of the extended class.

Extensions are used within the DDI-published Library Packages to provide relationships between classes as they increase in complexity to meet increasingly complex functionality. Thus, a "simple" version of a questionnaire class might be extended into a more complex class, describing a more complex questionnaire. Some classes exist only for the purpose of extension, and are declared abstract. A Functional View may never include an abstract class. Non-abstract classes may never have direct relationships with abstract classes. Extension is illustrated in Figure 4 below.

Figure 4. Extensions in DDI-Views



Here, an abstract class – Instrument, which is any tool used to collect data – is extended by Simple Questionnaire, which is itself extended by Complex Questionnaire. As we proceed through this chain of extension, the classes have increasingly large numbers of properties and relationships.

For example, if an Instrument class has a name property, a description property, and an ID property, these would be inherited by Simple Questionnaire, which might add a relationship to one or more Question classes. The Complex Questionnaire in turn might add a relationship to a Questionnaire Flow class, to add conditional logic to the questionnaire.

The second use of extension in the DDI model is to allow users to add needed metadata fields for the purposes of customization. Thus, a specific user community may decide to have a standard set of additional properties, classes, and relationships and create their own model Library Package which contains classes extending the classes in the DDI-published Library Packages. The creator of the extensions is the owner and maintainer of the extended classes and Library Packages – this is not the business of the DDI Alliance.

Extension in DDI is strictly defined: you are able to add new properties to existing classes, and add new relationships to existing classes. Extension is always done on a class which is referenced and inherited from: that is, a new class is declared which inherits all the properties and relationships of an existing class. New properties and relationships are then declared for it. Extension is always additive extension. There is no concept of refinement – that is handled using Functional Views.

Those creating their own custom Library Packages based on extensions to the DDI model may also declare entirely new classes which are not extension of DDI classes.

Extensions made by those customizing the DDI model are expressed using the same modeling techniques and information that are used for the development of the model published by the DDI Alliance itself. As a result of this, the same tools for the creation of documentation and syntax artefacts (XML schemas, RDF vocabularies) could potentially be used.

4.1.7 Managing the Library

In order to manage the Library effectively, the classes, together with primitives and complex data types, are grouped into Library Packages. The organization of Library Packages is currently flat. As development continues and the number of Library Packages increases the DDI model may be organized in a hierarchy of Library Packages arranged according to the types of constructs.

Library Packages are mutually exclusive and comprehensive. They are organic entities with a logical organization and are labeled in an accessible way so that developers and modelers can easily understand their content. They are stable and should not be changed often.

Provisional Library Organization

- Core
  - Primitives
  - Complex Data Types
  - Identification and versioning
  - Grouping and Comparison
  - Utility classes
- Conceptual
  - Universe, concept, category unit
  - Representations, code lists, classifications
  - Represented and conceptual variables
- Study

  - Study info
  - Study inception
  - Collection
  - Archiving and preservation
  - Access and discovery
- Data
  - Logical data structures
  - Physical data structures
  - Datasets
  - Instance variables (raw and derived variables)
- Process
- Geography
- Instrument and data source
  - Questionnaires, routing
  - Access to administrative data
  - Questions, items
- Methodology
  - Data transformations e.g. formulas

4.1.8 Versioning the Library

The classes within each Library Package as well as Functional Views are versioned. The whole model has a specific release date that acts as part of its identification. The Library Packages are versioned primarily for maintenance purposes

The versioning rule is that if the contents of a versioned class change, it is versioned. This means that versions "trickle up" – a new class is added to a Library Package, which versions the Library Package; the new version of the Library Package can drive a new release of the Model, and so on.

However, if a class does not change, its version does not change, even if the Library Package within which it lives in is versioned. Once published, a class is always available for use within Functional Views, even if it is not the latest version of the class. (If the old version of a class is good enough, it is still available for use in a new version of a Functional View, etc.) Once published, classes are never removed from the Library. All published classes and Functional Views will be available in the model forever

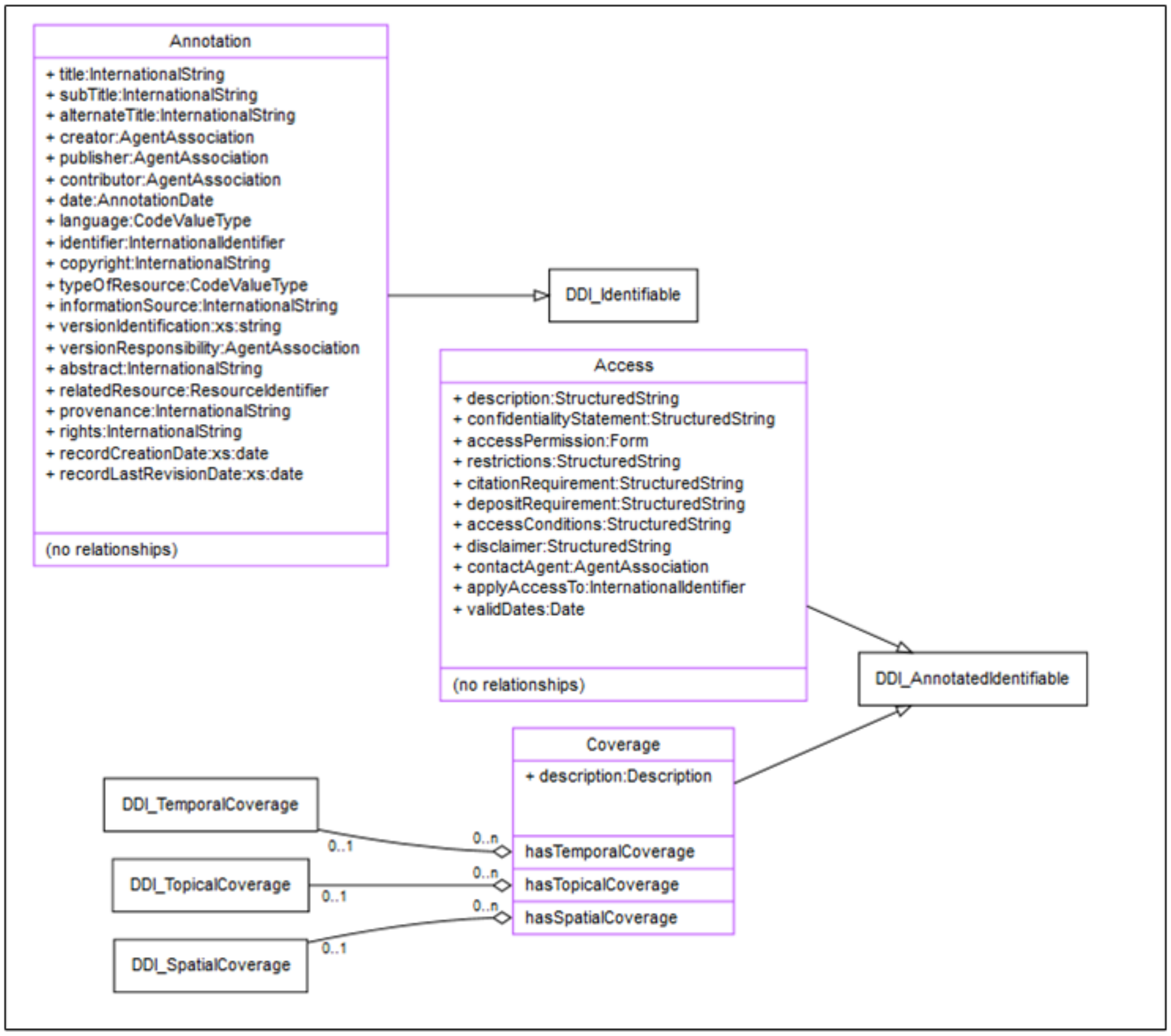
This has the effect of de-coupling the dependencies created by the use of extensions to add new things to the model. Decisions about what release packages consist of are driven by the needs of users and marketing considerations, and not by the chain of dependencies between classes, Library Packages, etc.

It is foreseen that at least initially, the Library will be released alongside sets of useful Functional Views, but incremental releases are possible without causing problems – a new version of the Library is released, but it will always contain all classes already in use.

4.1.9 Example of a Functional View

Figure 5 shows a diagram of the initial Discovery View, which includes the Access, Annotation and Coverage classes. Access and Coverage inherits from the AnnotatedIdentifiable class, while Annotation inherits from the Identifiable class. Coverage has aggregation relationships to TemporalCoverage, TopicalCoverage and SpatialCoverage.

Figure 5. Example Functional View

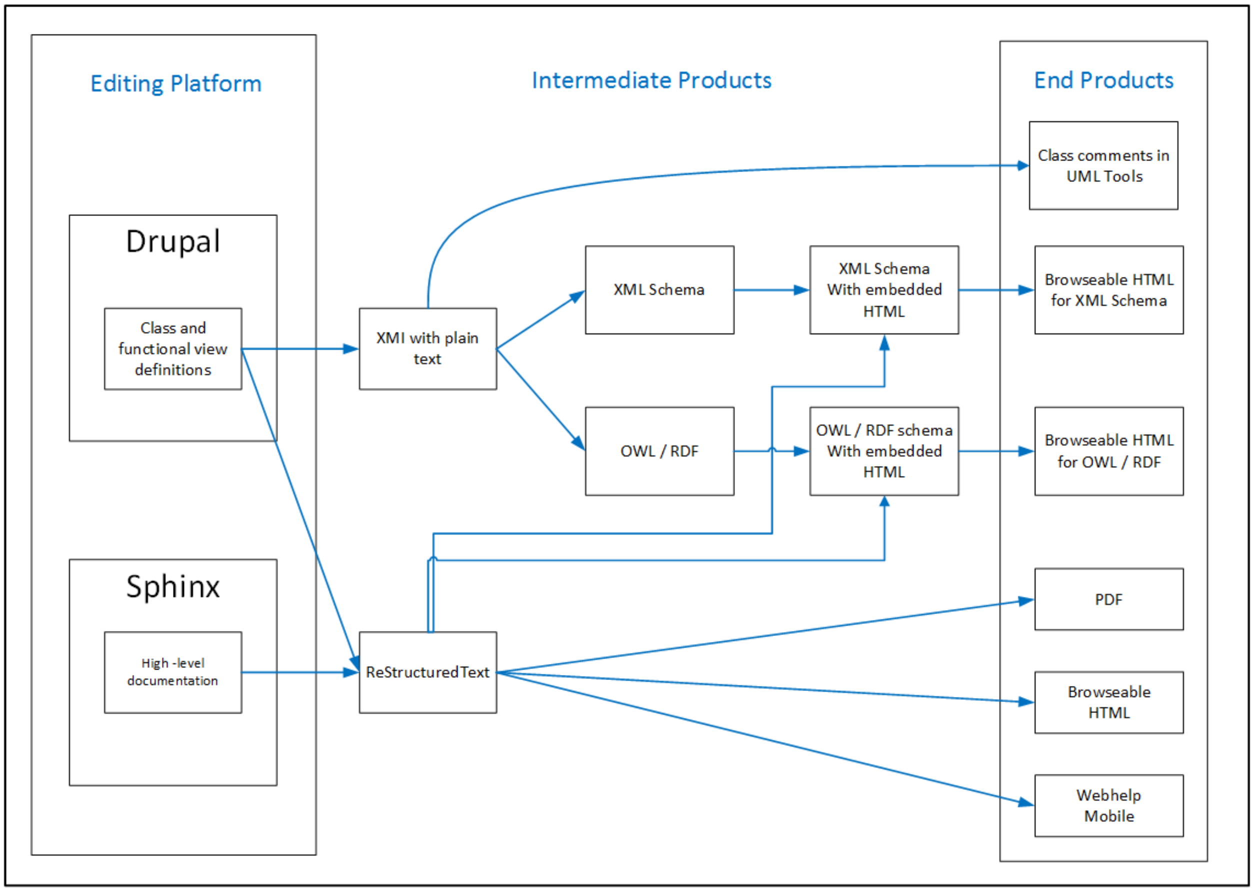

Annotation
+ title:InternationalString
+ subTitle:InternationalString
+ alternateTitle:InternationalString
+ creator:AgentAssociation
+ publisher:AgentAssociation
+ contributor:AgentAssociation
+ date:AnnotationDate
+ language:CodeValueType
+ identifier:InternationalIdentifier
+ copyright:InternationalString
+ typeOfResource:CodeValueType
+ informationSource:InternationalString
+ versionIdentification:xs:string
+ versionResponsibility:AgentAssociation
+ abstract:InternationalString
+ relatedResource:ResourceIdentifier
+ provenance:InternationalString
+ rights:InternationalString
+ recordCreationDate:xs:date
+ recordLastRevisionDate:xs:date
(no relationships)
DDI_Identifiable
Access
+ description:StructuredString
+ confidentialityStatement:StructuredString
+ accessPermission:Form
+ restrictions:StructuredString
+ citationRequirement:StructuredString
+ depositRequirement:StructuredString
+ accessConditions:StructuredString
+ disclaimer:StructuredString
+ contactAgent:AgentAssociation
+ applyAccessTo:InternationalIdentifier
+ validDates:Date
(no relationships)
DDI_AnnotatedIdentifiable
Coverage
+ description:Description
hasTemporalCoverage
hasTopicalCoverage
hasSpatialCoverage
DDI_TemporalCoverage
DDI_TopicalCoverage
DDI_SpatialCoverage
0..1
0..n
0..1
0..n
0..1
0..n

4.2 Production Framework

The model is being developed in Drupal at http://lion.ddialliance.org

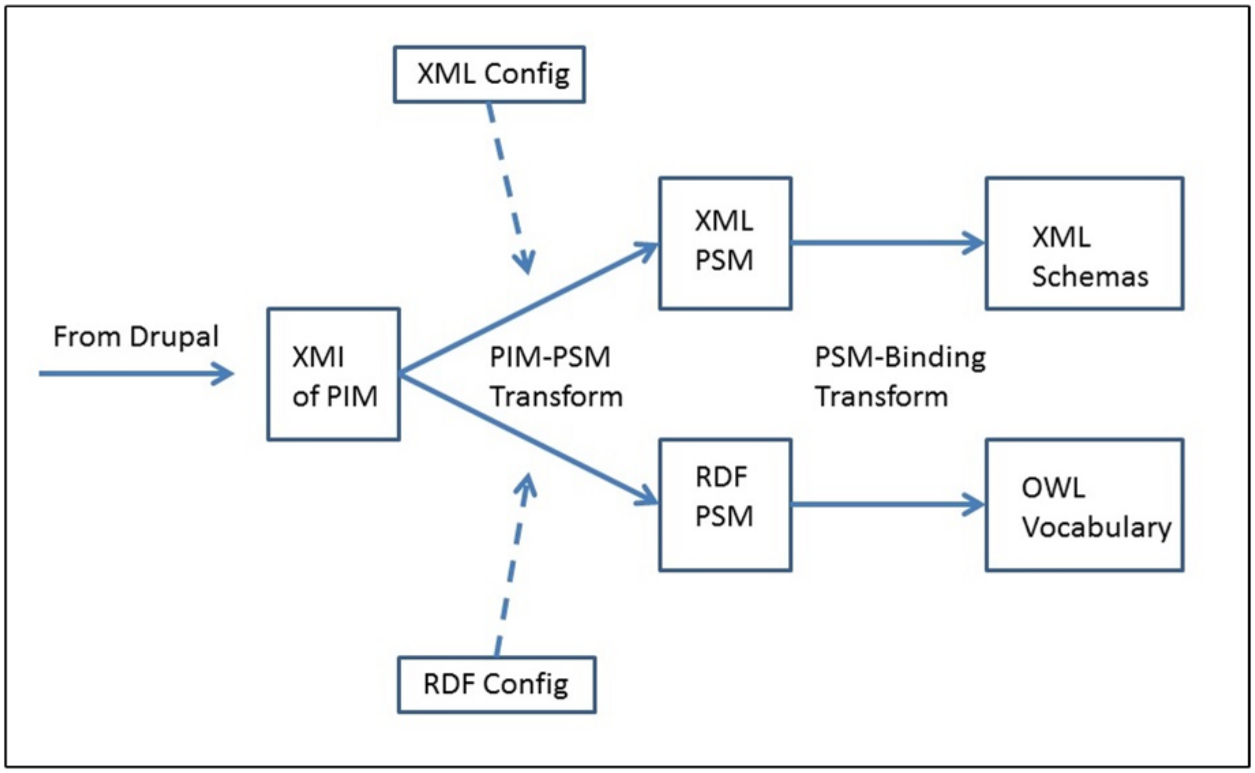
4.2.1 Documentation Flow

Figure 1. Documentation Flow in the Production Process


Editing Platform
Intermediate Products
End Products
Drupal
Class and functional view definitions
Sphinx
High -level documentation
XMI with plain text
ReStructured Text
XML Schema
XML Schema With embedded HTML
OWL / RDF
OWL / RDF schema With embedded HTML
Class comments in UML Tools
Browseable HTML for XML Schema
Browseable HTML for OWL / RDF
PDF
Browseable HTML
Webhelp Mobile

4.2.2 Bindings Production Flow in the Production Framework

Figure 2. Bindings Flow in the Production Process



The XMI for a portion of the model (as configured in Drupal) is exported as XMI for the platform-independent model (PIM). This is then transformed again, for each binding type (XML or RDF) to be produced, creating the platform-specific model (PSM) specific to that binding. This transformation from PIM to a specific PSM is informed by any needed configuration information. The PSM is optimized for the expressive capabilities of the binding to be produced (RDF and XML have different expressive capabilities). A transformation is then run on each PSM to produce the desired bindings. Each view will be expressed as an XML document type and as an RDF Vocabulary expressed in OWL. There will also be an XML and OWL binding for the library as a whole.

4.3 Design Principles

A set of design principles has been identified during the course of the development of DDI-Views, The list is shown below:

4.3.1 Design

The model * is developed in an agile, modular and iterative manner * is responsive to community needs to produce actionable metadata * should balance complexity with functionality and understandability * is extensible and strives towards compatibility between different versions * is maximally interoperable with relevant community standards * supports a plurality of implementations

4.3.2 Documentation

The documentation of the model * is clear, complete, and timely * is concise, comprehensible, accessible, and useable by multiple communities * provides justification for design decisions. * provides reference and functional perspectives

4.3.3 Capability

The model and its documentation * support the discovery, reuse, exchange, and sharing of (meta)data * support the conceptualization, capture, production, management, and analysis of (meta)data * support audit and reproducibility across the (meta)data lifecycle

4.4 DDI Base Blocks

4.4.1 Complex Data Types

These are extensions of base type Primitives. All complex data types (that is, the set of complex structures which are treated within the Drupal modeling platform as primitives, as for the values of properties) are located in the Complex Data Types package. (A set of complex structures which are used as datatypes for properties within a class description).

There is a distinct style of modeling these: each complex data type which has a primary content will have a property named "content" of whatever primitive type is needed. Complex data types will not be extensions of the primitive type of their primary content. This allows standard structures, such as capturing structured language strings, to be expressed in a single way throughout DDI.

4.4.2 Relationships

DDI classes are associated via binary relationships. Relationships have cardinality constraints, e.g. 1..1, 1..n, 0..n, and can be of different types, i.e. aggregation, composition, and neither (simple, untyped). Even though most relationships in the model are undirected, in the Drupal they are always defined within one of the classess participating in the relationship, i.e. the source; the class at the other end of the relationship is called the target. Similarly with names: the predicate represented in a relationship name does not impose a direction since the implicit converse predicate is also true in all undirected relationships. It's important to note that this is just a convention used in the Drupal and by no means imposes an actual conceptual direction in the association. For instance, RepresentedVariable has a relationship with ValueDomain. The relationship is defined in ValueDomain, which is the source, and it is named by a predicate, i.e. takesValuesFrom. This seems to imply a direction from RepresentedVariable to ValueDomain. However, the converse predicate, i.e. providesValuesFor, although not represented in the model, is also valid since such a relationship is conceptually undirected. In general, and unless otherwise indicated, all relationships in the Drupal are undirected.

4.4.3 Identification (Identifiable class)

The Identifiable class is core to the functioning of the standard. The purpose of the DDI Identifiable class is to: * Uniquely identify major objects in a persistent manner to ensure accurate reference and retrieval of the object content * Provide context for classes where an understanding of related class types within a packaging structure is essential to the understanding of the class (i.e., a specific classification within a classification scheme) * Manage metadata object change over time * Support the range of object management used by different organizations * Support early and late binding of references * Support interaction with closely related standards, in particular ISO/IEC 11179 and SDMX

Many organizations may have preexisting URI schemes, or have URI patterns imposed on them other organizations or governments. Unlike DDI3.x DDI4 will not require any specific information or pattern to be contained in the URIs of described resource.

To align with both DDI 3.x and ISO/IEC 11179-6, the Identifiable object will continue to be based on a combination of: * Agency Identifier (a unique identifier for the agency managing the object) * Item Identifier (a unique identifier of the object within the context of the agency) * Item Version (a version number of the object to track change over time)

These parts correspond to the agency, id, and version used in DDI 3.x and to the registration authority identifier (RAI), data identifier (DI), and version identifier (VI) constituting the international registration data identifier (IRDI) in ISO/IEC 11179-6

In DDI 3.x, all items had an agency, item id, and version. However, some types of items could inherit a parent item's agency. Some items would inherit a parent item's version. In DDI 4, all items will have their own Agency, Item Identifier, and Item Version specified. This three part structure is the equivalent of a unique persistent identifier for an object, such as described by DOIs and other similar structures. Note that while use of a version number with a DOI is optional, based on local practice, the Version Number in DDI is required due to the need to manage metadata within a dynamic workflow over time

4.4.4 Character Restrictions

In DDI 3.x, regular expressions restricted the Agency Identifier, Item Identifier, and Item Version. This has been removed for DDI 4. The only restrictions are that it is a string without colons and whitespace. Note that these are restrictions on the specific content not the structure of a DDI URN. The restriction on the use of a colon supports the use of this character as a URN separator. This complies with ISO/IEC 11179-6 as it imposes no limitations on the contents of the IRDI fields. In DDI 3.2, versions are restricted to integers that may be separated by periods. This forces implementers to use a specific versioning system. A more flexible system would use a "based on" reference to determine version history. In addition, a "based on" system adds the ability to branch and merge. A Based On system would be backwards compatible with DDI 3.x versioning systems.

CHAPTER

FIVE

USER GUIDES

DDI-Views introduces a new feature for those developing the model. There are **pattern** packages that consist entirely of abstract classes that can be **realized** by other classes. The pattern classes do not appear in the bindings (e.g. XML or RDF) that users of DDI will see. The realizing classes must mirror all of the properties and relationships of the pattern class although the properties of relationships may be renamed to make the semantics clearer. These pages describe several of the pattern classes and their use.

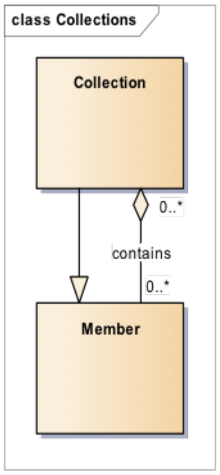
5.1 Using the Collection pattern

DDI-Views introduces a generic Collections pattern that can be used to model different types of groupings, from simple unordered sets to all sorts of hierarchies, nesting and ordered sets/bags.

A collection is a container, which could be either a set (i.e. unique elements) or a bag (i.e. repeated elements), of Members. Collections can also be extended with richer semantics (e.g. generic, partitive, and instance, among others) to support a variety of DDI 3.x and GSIM structures, such as Node Sets, Schemes, Groups, sequences of Process Steps, etc. Collections, together with their related classes in the pattern, provide an abstraction to capture commonalities among a variety of seemingly disparate structures.

A Collection consists of zero, one or more Members (Figure 1). A Member could potentially belong to multiple Collections. A Collection is also a Member, which allows for nesting of Collections in complex structures. Members have to belong to some Collection, except in the case of nested Collections where the top level Collection is a Member that doesn't belong to any Collection.

Figure 1. Collections class

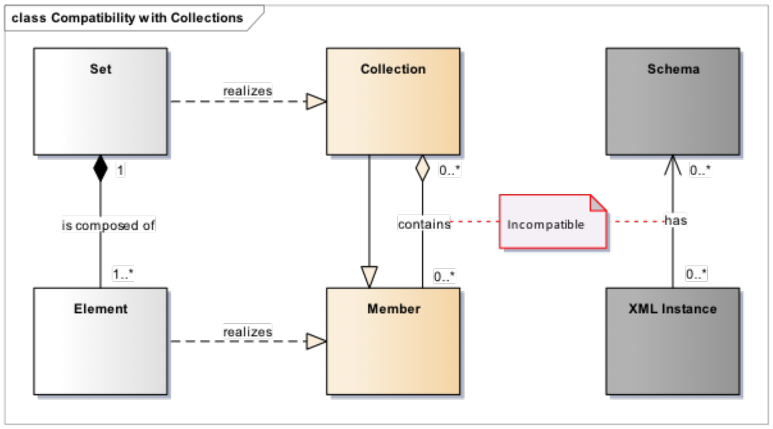


This pattern can be used via a special type of association called *realizes*. DDI-Views uses *realizes* to say that a class 'behaves' like a Collection. For instance, consider a Set that consists of Elements, they implement the Collection pattern as follows: Set *realizes* Collection and Element *realizes* Member.

To realize this pattern all classes involved must be associated in a way that is compatible with the pattern. As a rule of thumb, a more restrictive type of association than the one that appears in the pattern is compatible, a looser one is not. For instance, since the collection pattern has an aggregation association (denoted by the empty diamond), classes realizing the Collection pattern need to be related by either an aggregation or a composition, nothing else. In addition, source and target, when applicable, must also match, e.g. the diamond of the aggregation/composition needs to be on the class realizing Collection, not Member. Similar compatibility rules apply to cardinality. Furthermore, all associations must be realized, with the exception of *IsA* associations, which are usually part of the pattern definition and do not apply to individual realizations in the same way. Renaming associations does not affect compatibility as long as the documentation clearly explains how they map to the association in the pattern.

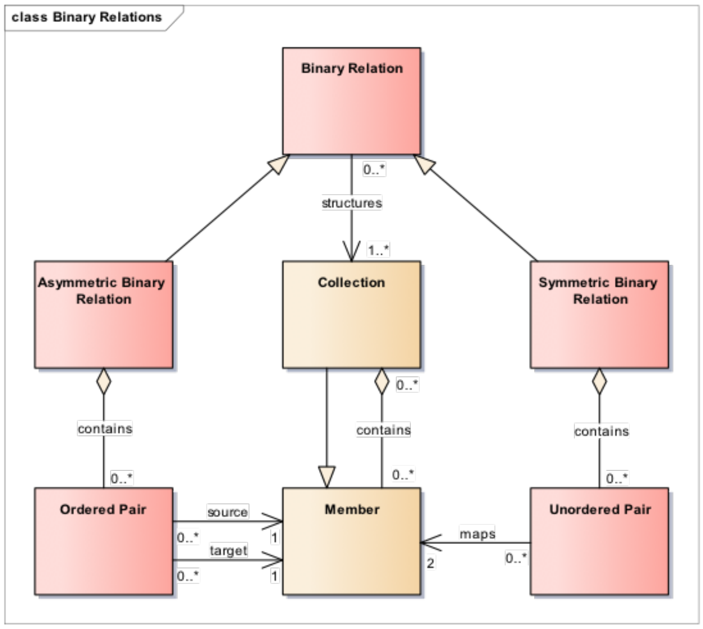
For instance consider Figure 2. In this example, a Set class is defined as being composed of at least one Element, i.e. no empty Sets are allowed. In addition, an Element always belong to one and only one Set, which means that deleting the Set will also delete its Elements. Such an association is compatible with the *contains* association in the Collection pattern and thus Set and Element can realize Collection and Member, respectively. In contrast, Schema and XML Instance cannot realize the pattern: the association is neither an aggregation nor a composition, Schema is not a grouping of XML Instances, and the association points from XML Instance to Schema. None of this is compatible with the Collection pattern, in particular with the semantics of the *contains* association between Collection and Member.

Figure 2. Compatibility with Collections class



Collections can be structured with Binary Relations (Figure 3), which are sets of pairs of Members in a Collection. Binary Relations can have different properties, e.g. *totality*, *reflexivity*, *symmetry*, and *transitivity*, all of which can be useful for reasoning.

Figure 3. Binary Relations



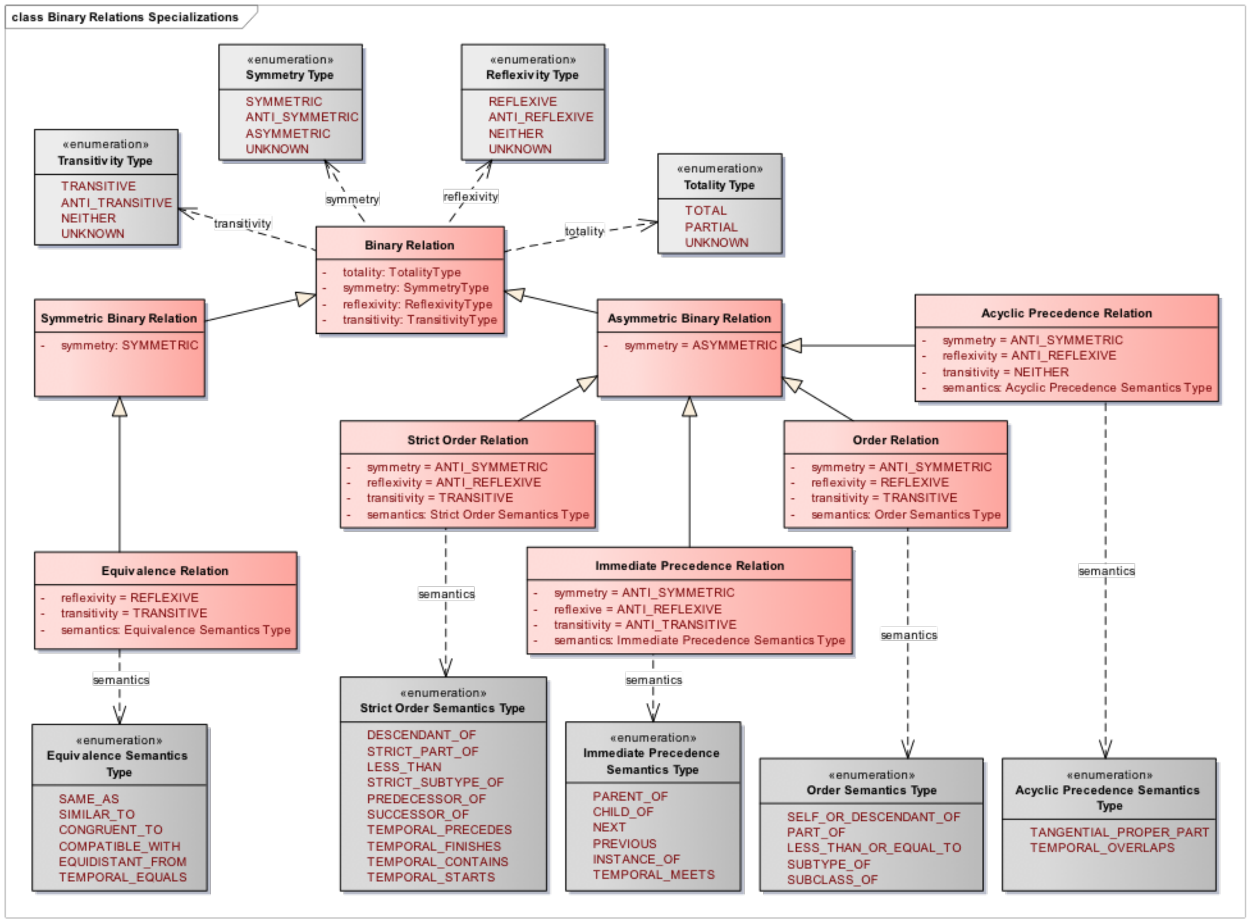
A Binary Relation is said to be symmetric if for any pair of Members *a*, *b* in the associated Collection, whenever *a* is related to *b* then also *b* is related to *a*. Based on this property we define two specializations of Binary Relation: Symmetric Binary Relation, when the property is *true*, and Asymmetric Binary Relation, when the property is *false*. Symmetric Binary Relations can be viewed as collections of Unordered Pairs themselves, whereas Asymmetric Binary Relations can be viewed as collections of Ordered Pairs. However, for simplicity, we do not model Relations themselves with the Collection pattern.

We can further classify Binary Relations based on additional properties. We say that a Binary Relation is *total* if all Members of the associated Collection are related to each other. We call it *reflexive* if all Members of the associated Collection are related to themselves. Finally, we say it is *transitive* if for any Members *a*, *b*, *c8 in the associated Collection, whenever a is related to *b* and *b* is related to *c* then *a* is also related to *c*. Further explanation of this is in A Guide to Binary Relations

These properties can be combined to define subtypes of Binary Relations (Figure 4), e.g. Equivalence Relation, Order Relation, Strict Order Relation, Immediate Precedence Relation, and Acyclic Precedence Relation, among others. Equivalence Relations are useful to define partitions and equivalence classes (e.g. Levels in a Classification). Order Relations can be used to represent lattices (e.g. class hierarchies, partitive relationships), Immediate Precedence Relations can define sequences and trees (e.g. linear orderings, parent-child structures) and Acyclic Precedence Relation can represent directed acyclic graphs (e.g. molecular interactions, geospatial relationships between regions).

These subtypes can also have various semantics, e.g. Part-Of and Subtype-Of for Order Relations, to support a variety of use cases and structures, such as Node Sets, Schemes, Groups, sequences of Process Steps, etc. Note that some of them include temporal semantics, e.g. Strict Order Relation and Acyclic Precedence Relation.

Figure 4. Binary Relations specialization

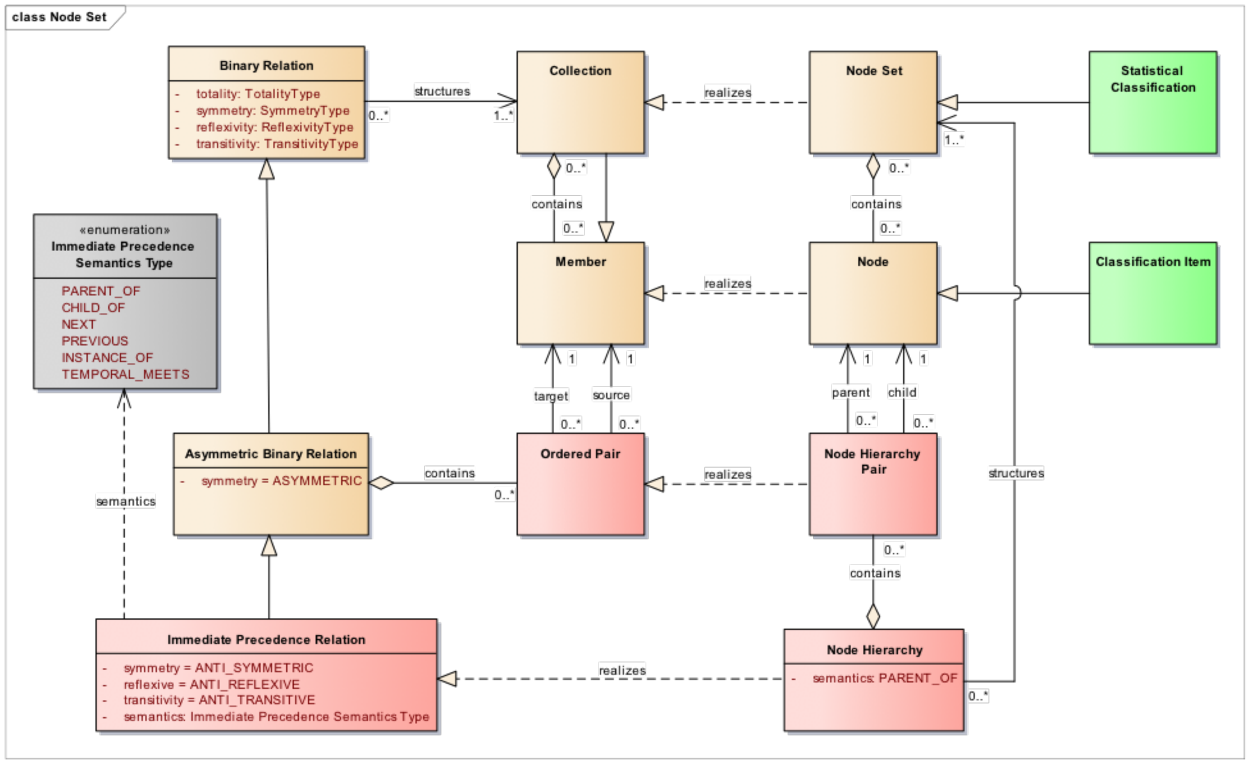

class Binary Relations Specializations
«enumeration» Symmetry Type
SYMMETRIC
ANTI_SYMMETRIC
ASYMMETRIC
UNKNOWN
«enumeration» Reflexivity Type
REFLEXIVE
ANTI_REFLEXIVE
NEITHER
UNKNOWN
«enumeration» Transitivity Type
TRANSITIVE
ANTI_TRANSITIVE
NEITHER
UNKNOWN
«enumeration» Totality Type
TOTAL
PARTIAL
UNKNOWN
symmetry
reflexivity
transitivity
totality
Binary Relation
- totality: TotalityType
- symmetry: SymmetryType
- reflexivity: ReflexivityType
- transitivity: TransitivityType
Symmetric Binary Relation
- symmetry: SYMMETRIC
Asymmetric Binary Relation
- symmetry = ASYMMETRIC
Acyclic Precedence Relation
- symmetry = ANTI_SYMMETRIC
- reflexivity = ANTI_REFLEXIVE
- transitivity = NEITHER
- semantics: Acyclic Precedence Semantics Type
Strict Order Relation
- symmetry = ANTI_SYMMETRIC
- reflexivity = ANTI_REFLEXIVE
- transitivity = TRANSITIVE
- semantics: Strict Order Semantics Type
Order Relation
- symmetry = ANTI_SYMMETRIC
- reflexivity = REFLEXIVE
- transitivity = TRANSITIVE
- semantics: Order Semantics Type
Equivalence Relation
- reflexivity = REFLEXIVE
- transitivity = TRANSITIVE
- semantics: Equivalence Semantics Type
Immediate Precedence Relation
- symmetry = ANTI_SYMMETRIC
- reflexive = ANTI_REFLEXIVE
- transitivity = ANTI_TRANSITIVE
- semantics: Immediate Precedence Semantics Type
semantics
semantics
semantics
semantics
semantics
«enumeration» Equivalence Semantics Type
SAME_AS
SIMILAR_TO
CONGRUENT_TO
COMPATIBLE_WITH
EQUIDISTANT_FROM
TEMPORAL_EQUALS
«enumeration» Strict Order Semantics Type
DESCENDANT_OF
STRICT_PART_OF
LESS_THAN
STRICT_SUBTYPE_OF
PREDECESSOR_OF
SUCCESSOR_OF
TEMPORAL_PRECEDES
TEMPORAL_FINISHES
TEMPORAL_CONTAINS
TEMPORAL_STARTS
«enumeration» Immediate Precedence Semantics Type
PARENT_OF
CHILD_OF
NEXT
PREVIOUS
INSTANCE_OF
TEMPORAL_MEETS
«enumeration» Order Semantics Type
SELF_OR_DESCENDANT_OF
PART_OF
LESS_THAN_OR_EQUAL_TO
SUBTYPE_OF
SUBCLASS_OF
«enumeration» Acyclic Precedence Semantics Type
TANGENTIAL_PROPER_PART
TEMPORAL_OVERLAPS

A modeller can use the different semantics types as a guide when trying to decide what type of Binary Relation to realize. For instance, if the new class to be added to the model is a Node Set containing Nodes that will be organized in a parent-child hierarchy, the modeller can define a Node Hierarchy class with PARENT_OF semantics to structure the Node Set. The type of Binary Relation to realize then is Immediate Precedence Relation because it is the one that has the required semantics in its Semantics Type.

Alternatively, a modeller familiar with the definitions of the Binary Relation properties, i.e. symmetry, reflexivity and transitivity, could make the choice based on what combination represents the type they are looking for. For instance, a parent-child hierarchy requires the Binary Relation to be ANTI_SYMMETRIC (if a Node is the parent of another, the latter is not the parent of the former), ANTI_REFLEXIVE (a Node cannot be a parent of itself) and ANTI_TRANSITIVE (a Node is not the parent of its children's children). It is easy to see that the only one that satisfies that criteria is the Immediate Precedence Relation.

Figure 5 shows an example of the realization of the pattern. We can model Node Hierarchy and Node Hierarchy Pair classes as realizations of Immediate Precedence Relation and Ordered Pair, respectively.

Figure 5. Node Set

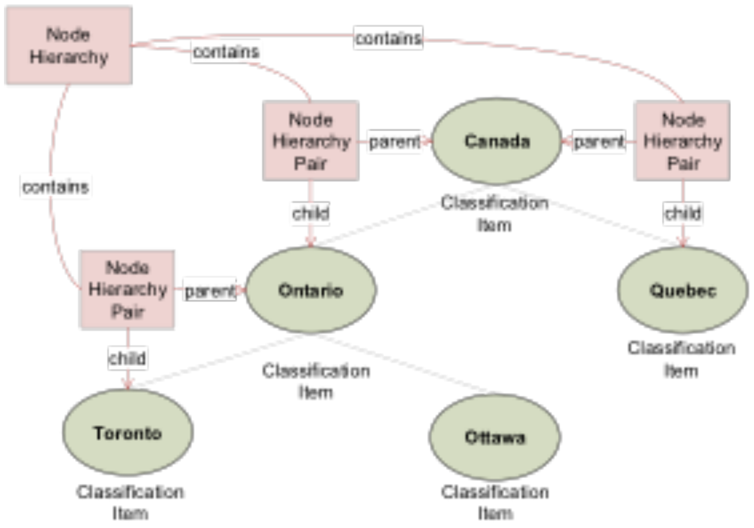


Let us illustrate how this model works with a simple instance. Consider a Geography Statistical Classification with Classification Items (Figure 6) representing Canada, its provinces and cities.

Since Statistical Classifications are Node Sets and Classification Items are Nodes, we can view Classification Items such as Canada, Ontario, Quebec, Toronto, etc. as Members in a Collection structured by a Node Hierarchy Relation. Node Hierarchy Pairs represent the parent-child relationships in the Node Hierarchy Relation. For instance, (Canada, Ontario) is a Node Hierarchy Pair in which Canada is the parent and Ontario is the child. Other Node Hierarchy Pairs are (Canada, Quebec) and (Canada, Toronto).

Note that by maintaining the hierarchy in a separate structure, i.e. the Node Hierarchy, Items can be reused in multiple Classifications. For instance, in another geography Statistical Classification provinces grouped into regions, Ontario can be made the child of the Central Region instead of Canada without changing the definition of the Classification Items involved, i.e. Canada, Ontario and Central Region in this case.

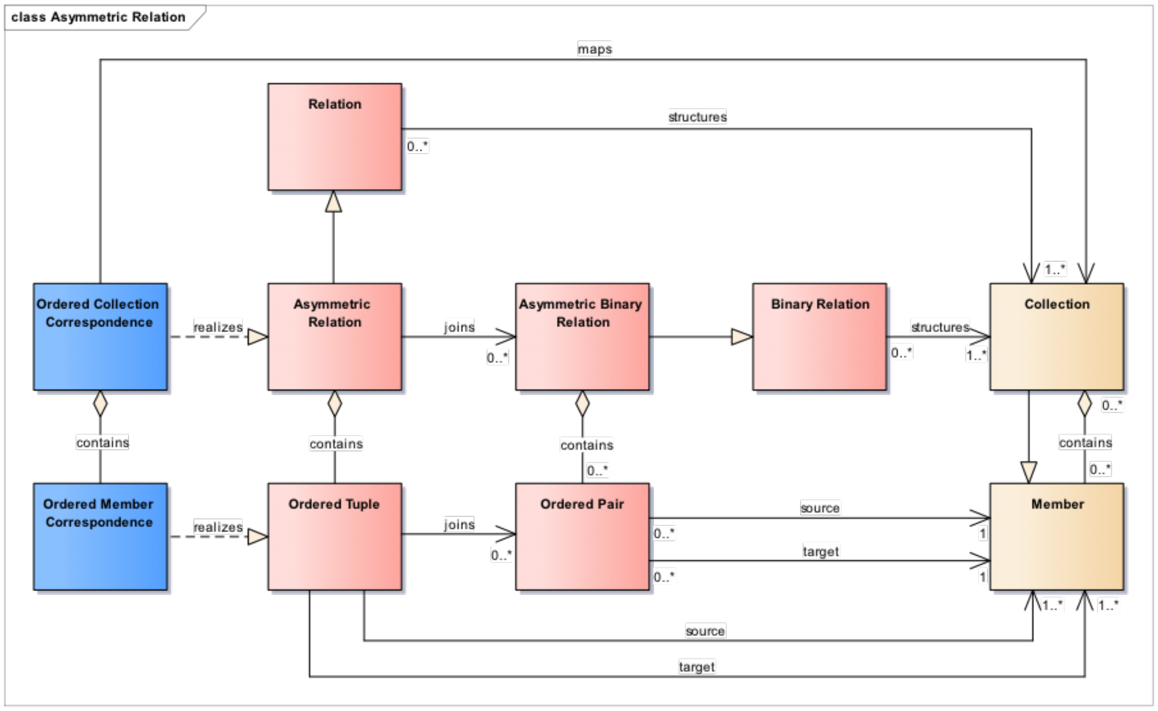
Figure 6. Node Set example



Interestingly, Binary Relations might not be enough for some purposes. First of all, some structures cannot be reduced to binary representations, e.g. hypergraphs, without cumbersome supporting structures. In addition, a Binary Relation could be too verbose in some cases since the same Member in a Collection could appear multiple times in different pairs, e.g. one-to-many relationships like parent-child and ancestor-descendent. An n-ary Relation provides a more compact representation for such cases. Like Binary Relations they come in two flavours: Symmetric Relation and Asymmetric Relation (Figure 7).

Asymmetric Relations (Figure 7) provide an equivalent, yet more compact, n-ary representation for multiple Ordered Pairs that share the same source and/or target Members. In addition, they can be realized by Correspondence Tables to map Node Sets and their Nodes based on some criterion (e.g. similarity, provenance, etc.). Correspondence Table and Map realize Asymmetric Relation and Ordered Tuple, respectively.

Figure 7. Asymmetric relations

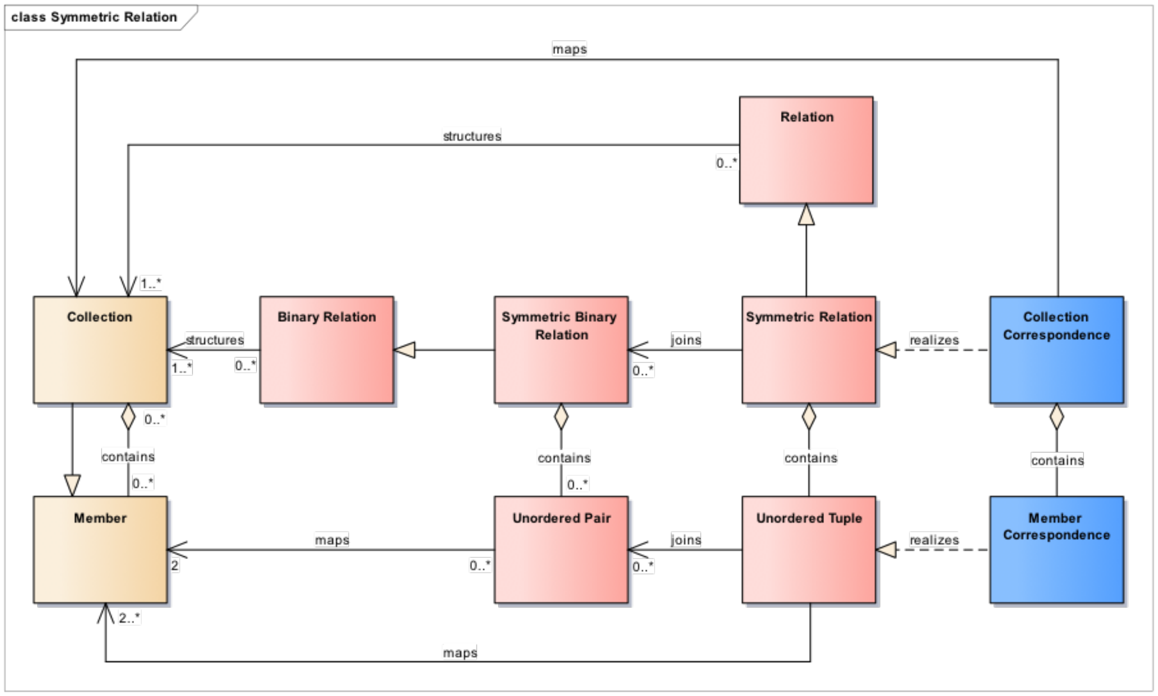


Consider again the geography classification tree of the Canadian example above (Figure 6). All Node Hierarchy Pairs that have the same parent Member could be represented with a single Node Hierarchy Tuple that realizes the Ordered Tuple in the model. The realization will also rename source as parent and target as child.

Although only two cities are shown in the example, Ontario has hundreds of them. Using a Node Hierarchy realizing an Asymmetric Relation, all pairs that have Ontario as parent, e.g. (Ontario, Toronto), (Ontario, Ottawa), (Ontario, Kingston), etc., could be joined into a single n-ary Node Hierarchy Tuple with Ontario as parent and Toronto, Ottawa, Kingston, etc. as children. With this representation we replaced multiple pairs with a single tuple and avoid the repetition of Ontario hundreds of times for each individual pair.

Classifications sometimes need to be mapped to each other. In our geography example we could have two variants with similar structure as shown Figure 8.

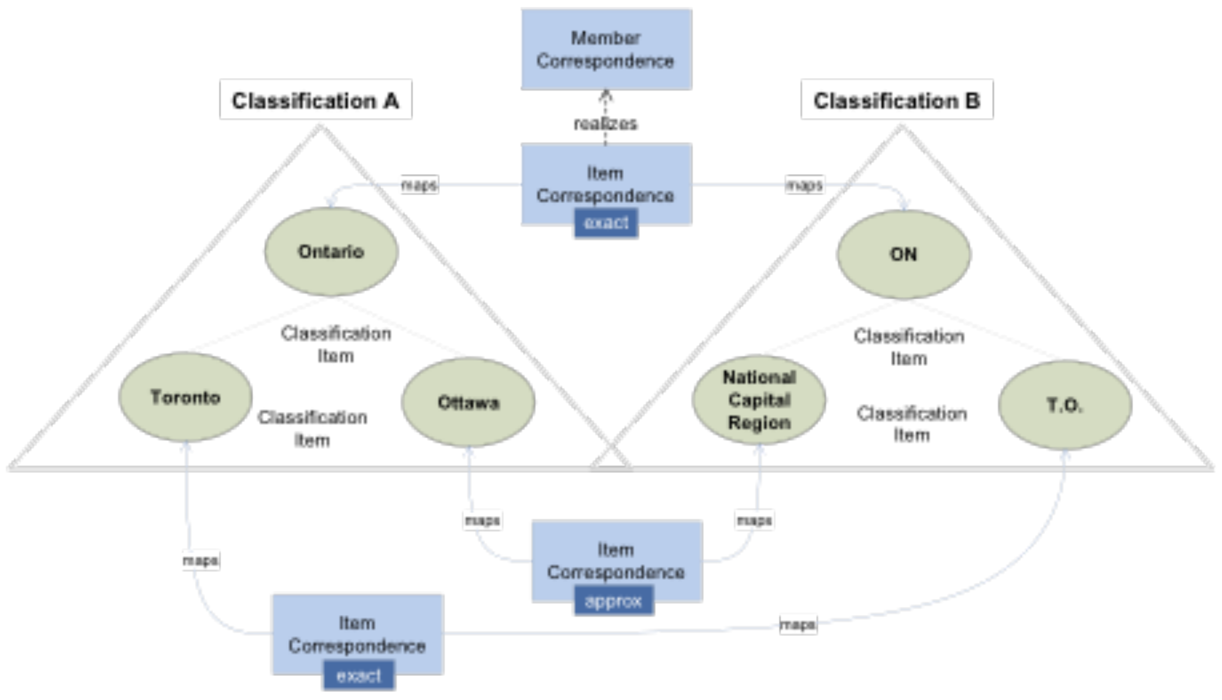
Figure 8. Example showing both Asymmetric and Symmetric relations



Some of the Classification Items in both Classifications are the exactly the same, e.g. Toronto and T.O., they just have different names. Others, e.g. Ottawa and National Capital Region, are only approximate, e.g. Ottawa is part of the National Capital Region (NCR), but the latter is larger and contains other municipalities. For such a case, we use a realization of Ordered Member Correspondence to create a containment mapping between Ottawa and NCR whereas the other mappings are exact.

Symmetric Relations (Figure 9) are similarly structured as Asymmetric Relations. They provide an equivalent, yet more compact, n-ary representation for multiple Unordered Pairs that have some Members in common. In addition, they can be used to model (unordered) Correspondences between Collections and Members.

Figure 9. Symmetric relations

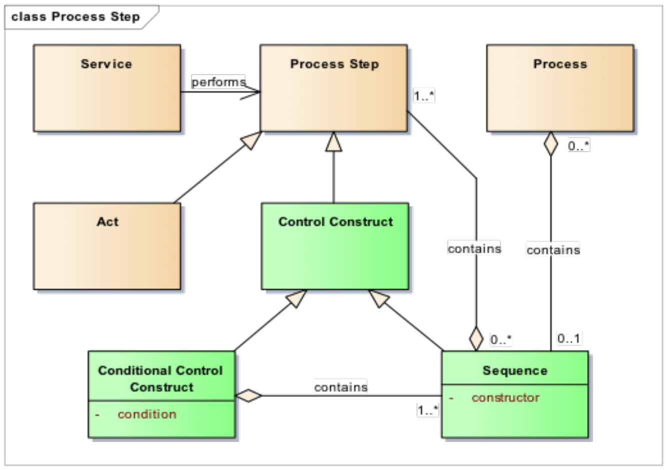


5.2 Using the Process pattern

The Process pattern introduced in DDI-Views consists of a basic set of classes to describe process steps and information flows between them. Some of its classes are extensions of classes in the Collections Pattern. Figure 1 illustrates a high level view of the Process pattern.

A Process Step performs one or more business functions at any granularity. Each Process Step can be performed by a Service. Process Steps and Services can have Interfaces with input, outputs and other interface definitions. Process Steps can be nested and thus describe processes at multiple levels of detail. The Process Control Step handles the execution flow of the steps in its scope. The Information Flow specifies how information objects move between Process Steps by mapping inputs and outputs.

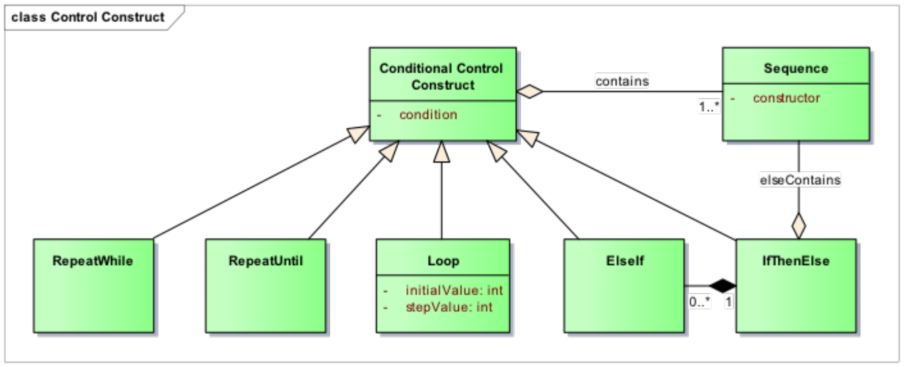
Figure 1. The Process Pattern



The Process Pattern can be realized in multiple ways. DDI-Views include a well-known realization called Workflow (Figure 2), which can be mapped to process execution languages like BPEL or BPMN.

A Workflow is Sequence of Workflow Steps that performs one or more business functions. Each Workflow Step can be performed by a Workflow Service.

Figure 2. Workflow

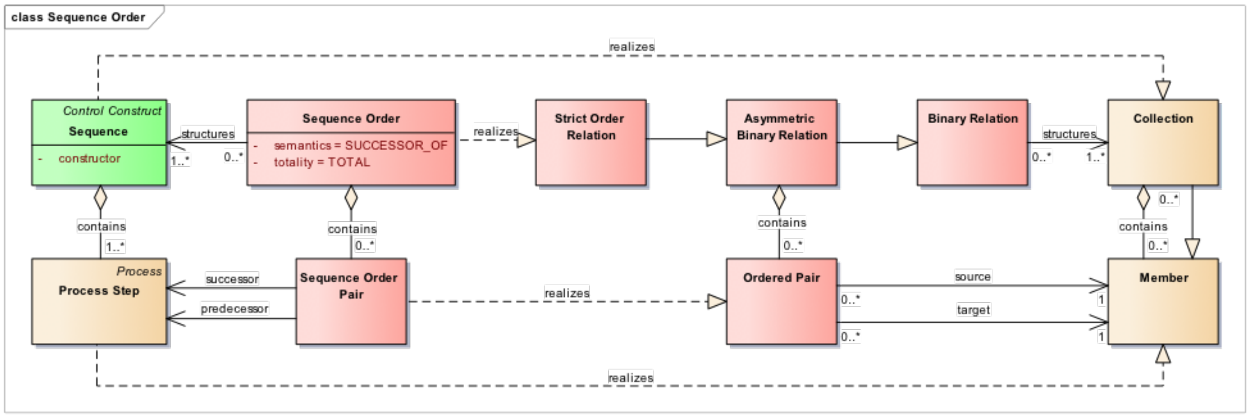


There are two types of Workflow Steps: Acts and Control Constructs.

Acts represent actions and are atomic Process Steps, i.e. they cannot be composed of other Process Steps. An Act is similar to a terminal in the production rules of a formal grammar and an instruction in a programming language.

A Control Construct describes logical execution flows between Process Steps. Control Constructs can be nested via its sub-classes and thus describe workflows at multiple levels of detail. The nesting of Workflow Steps always terminates in an Act. All nesting of Workflow Steps occurs via Workflow Sequences and Conditional Control Constructs, both specializations of Control Constructs. The former models linear execution of Workflow Steps whereas the latter includes three types of iterative constructs: repeatWhile, repeatUntil and Loop. The Workflow Sequence at the end of the *contains* association represents the body of the Conditional Control Construct, which is executed depending on the result of the *condition* evaluation. The specialized sub-classes determine whether the Sequence is executed in each iteration before the condition is evaluated (RepeatUntil), or after (RepeatWhile, Loop). The Loop also provides a counter with *initialValue* and *stepValue* that can be used to specify in the condition how many times the Sequence in the body is executed.

Figure 3. Control Constructs



In addition to the iterative constructs, Conditional Control Constructs includes IfThenElse, which provides a means to specify branching control flows. It contains the *condition* inherited from the parent class and two associations: *contains* (also inherited from the parent class), to the Sequence of Process Steps that is executed when the condition is *true*, and *containsElse*, to an optional Sequence to be executed if the condition is evaluated to *false*. Optionally, IfThenElse can also have an associated ElseIf construct to model switch statements.

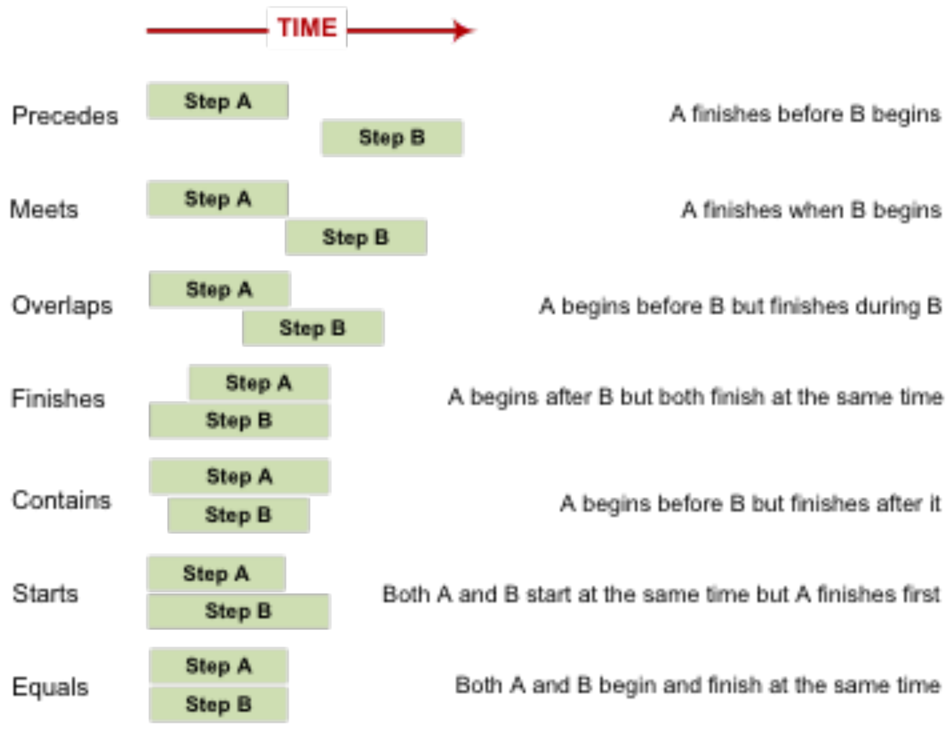
A Workflow Sequence can be viewed as a Collection whose Members are Workflow Steps that can be ordered in two different ways: with one or more Temporal Interval Relations, i.e. a design-time temporal constraint, or with a Rule (constructor) to determine ordering at run-time. Let's begin discussing temporal constraints.

Temporal Interval Relations provide a mechanism for capturing Allen's interval relations, one of the best established formalisms for temporal reasoning. Temporal Interval Relations can be used to define *temporal constraints* between pairs of Workflow Steps, e.g. whether the execution of two Workflow Steps can overlap in time or not, or one has to finish before the other one starts, etc. Note that this also supports parallel processing, which is also implicit in the model: the definition of Conditional Control Constructs can contain multiple Workflow Sequences that could be executed in parallel.

There are thirteen Temporal Interval Relations: twelve asymmetric ones, i.e. *precedes*, *meets*, *overlaps*, *finishes*, *contains*, *starts* and their converses, plus *equals*, which is the only one that has no converse or, rather, it is the same as its converse. Together these relations are distinct (any pair of definite intervals are described by one and only one of the relations), exhaustive (any pair of definite intervals are described by one of the relations), and qualitative (no numeric time spans are considered).

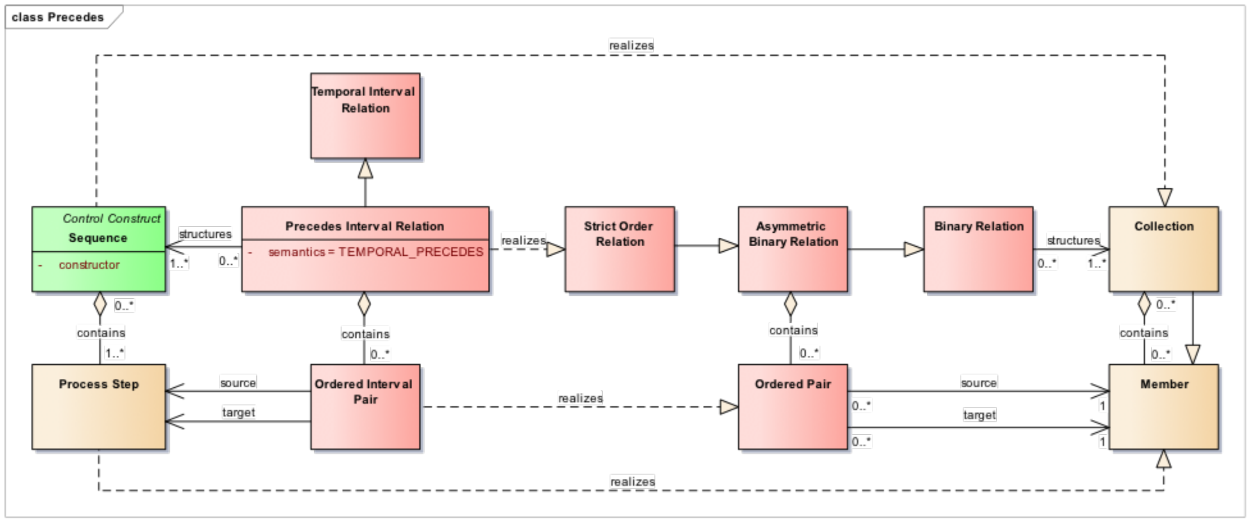
Following Allen's, Temporal Interval Relations are defined as follows.

Figure 4. Allen's Temporal Interval Relations


TIME
Precedes
Step A
Step B
A finishes before B begins
Meets
Step A
Step B
A finishes when B begins
Overlaps
Step A
Step B
A begins before B but finishes during B
Finishes
Step A
Step B
A begins after B but both finish at the same time
Contains
Step A
Step B
A begins before B but finishes after it
Starts
Step A
Step B
Both A and B start at the same time but A finishes first
Equals
Step A
Step B
Both A and B begin and finish at the same time

In DDI-Views, each of the asymmetric Allen's interval relations is a Temporal Interval Relation that realizes different Binary Relations with specific temporal semantics. All asymmetric Temporal Interval Relations contain Ordered Interval Pairs whereas the only symmetric one, i.e. Equals, contains Unordered Interval Pairs. For instance, the Precedes Interval Relation realizes the pattern as shown in the diagram below.

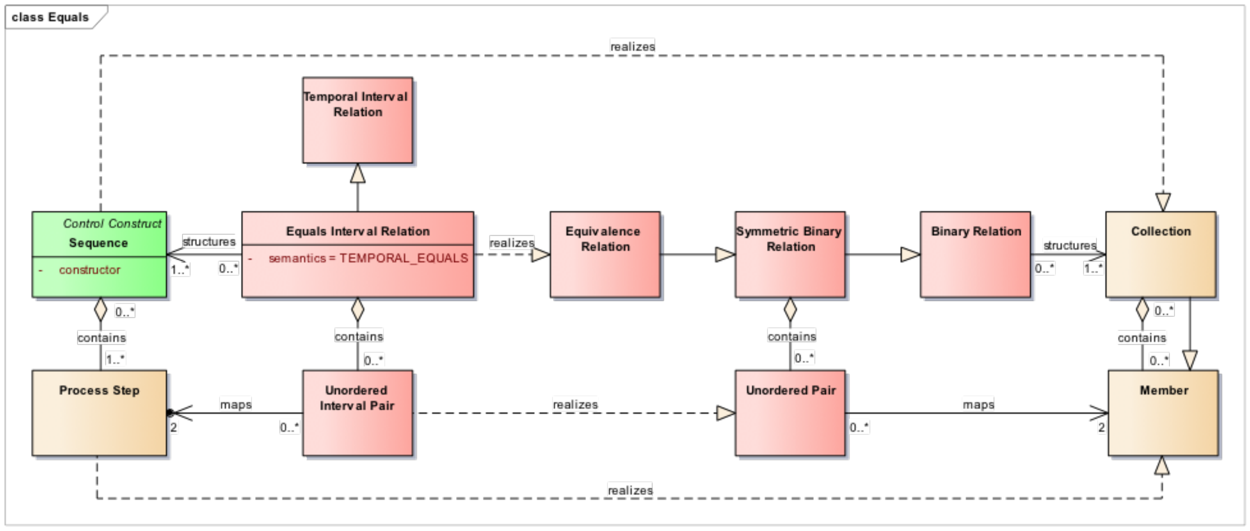
Figure 5. Precedes Interval Relation



Precedes Interval Relation and Ordered Interval Pair realize Strict Order Relation and Ordered Pair, respectively. If we look back to the Binary Relations Specialization diagram (in Using the Collections Pattern - Figure 4) we notice that Strict Order Relation has the TEMPORAL_PRECEDES semantics, among others, which means that the Process Step at the end of the *source* association in the Ordered Interval Pair has to finish before the one at the end of *target* starts.

The Equals Interval Relation is a slightly different case because it is an equivalence relation rather than an asymmetric one and therefore it contains Unordered Interval Pairs.

Figure 6. Equals Interval Relation

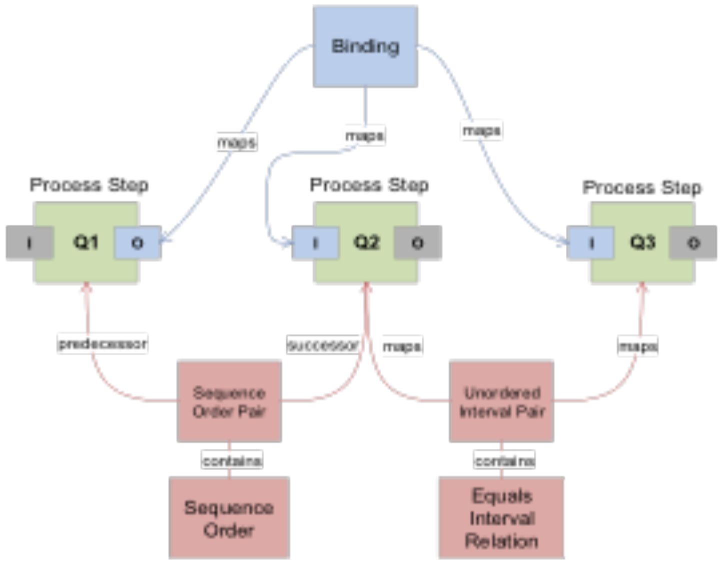


Equals Interval Relation and Unordered Interval Pair realize Equivalence Relation and Unordered Pair, respectively. Equivalence Relation is simply a Symmetric Binary Relation that is REFLEXIVE and TRANSITIVE with some additional semantics, among which we find TEMPORAL_EQUALS, the one required by the Equals Interval Relation. This means that the execution of the two Process Steps at the end of the maps association in Unordered Interval Pair begin and end at the same time. Let's see how Temporal Interval Relations work with an example.

Consider a questionnaire with set of questions and a control flow logic defined between them. The flow between questions can be modeled with a Workflow Sequence where each question is a realization of a Workflow Step. Now, let's assume a couple of conditions: (i) question Q3 requires the answer of both Q1 and Q2, and (ii) question Q4 is triggered by answering question Q2. Note the difference between requiring an answer and being triggered by an answer. The former means that there is no necessary immediate execution, i.e. Q3 can be executed long after both Q1 and Q2 have been answered. In other words, the exact moment for executing Q3 will depend on other parts of the control flow logic of the questionnaire, i.e. other constraints defined in the questionnaire flow. However, when a question triggers another it means that the latter is executed right after the former is answered.

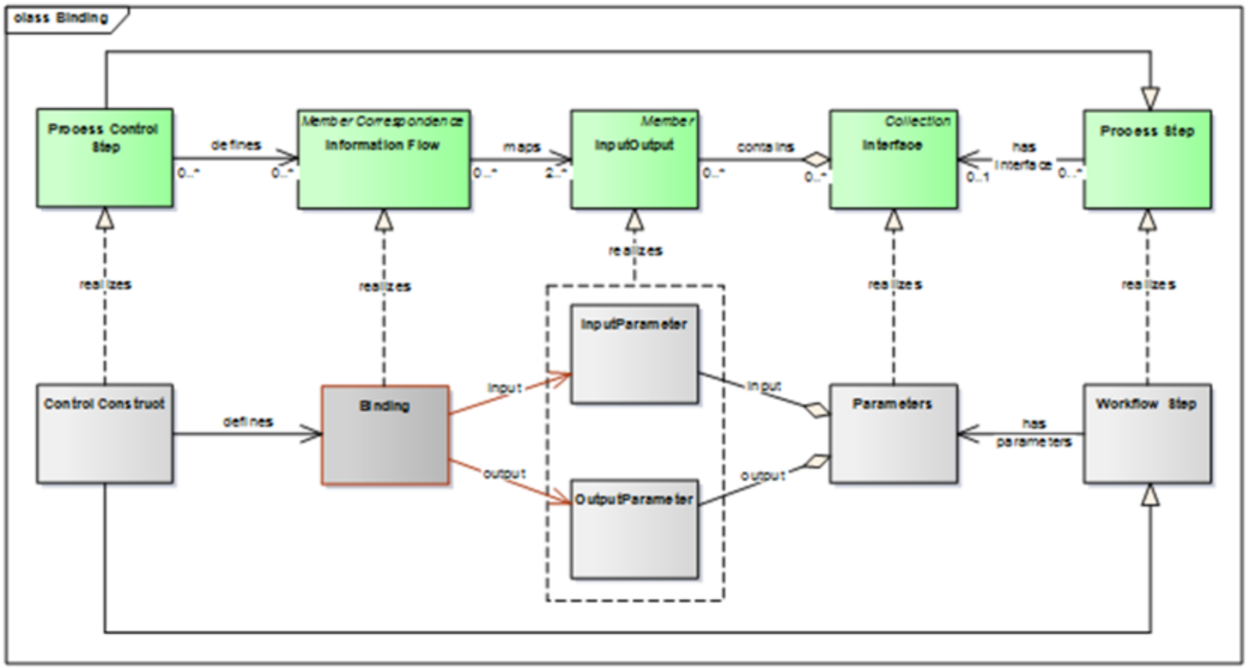
With that in mind, the precedence constraint between Q1, Q2 and Q3 can be modeled with a Precedes Interval Relation whereas the one between Q2 and Q4 can be represented by a Meets Interval Relation. Remember that the fact that Q1 precedes Q3 means that Q1 finishes before Q3 starts whereas Q2 meets Q4 means that Q4 starts exactly when Q2 finishes, which is exactly what the two Temporal Interval Relations mentioned above can express. The following (Figure 7) is the resulting model.

Figure 7. Temporal Interval Relations Example



So far we described the control flow logic that is specified by Control Constructs. How does information flow between the steps in the control flow logic? We mentioned that the pattern has a class for that purpose, i.e. Information Flow. Binding is the realization of Information Flow for Workflows. A Binding is a design-time class that maps Input and Output Parameters of different steps in a Workflow Sequence or between a step and its sub-steps.

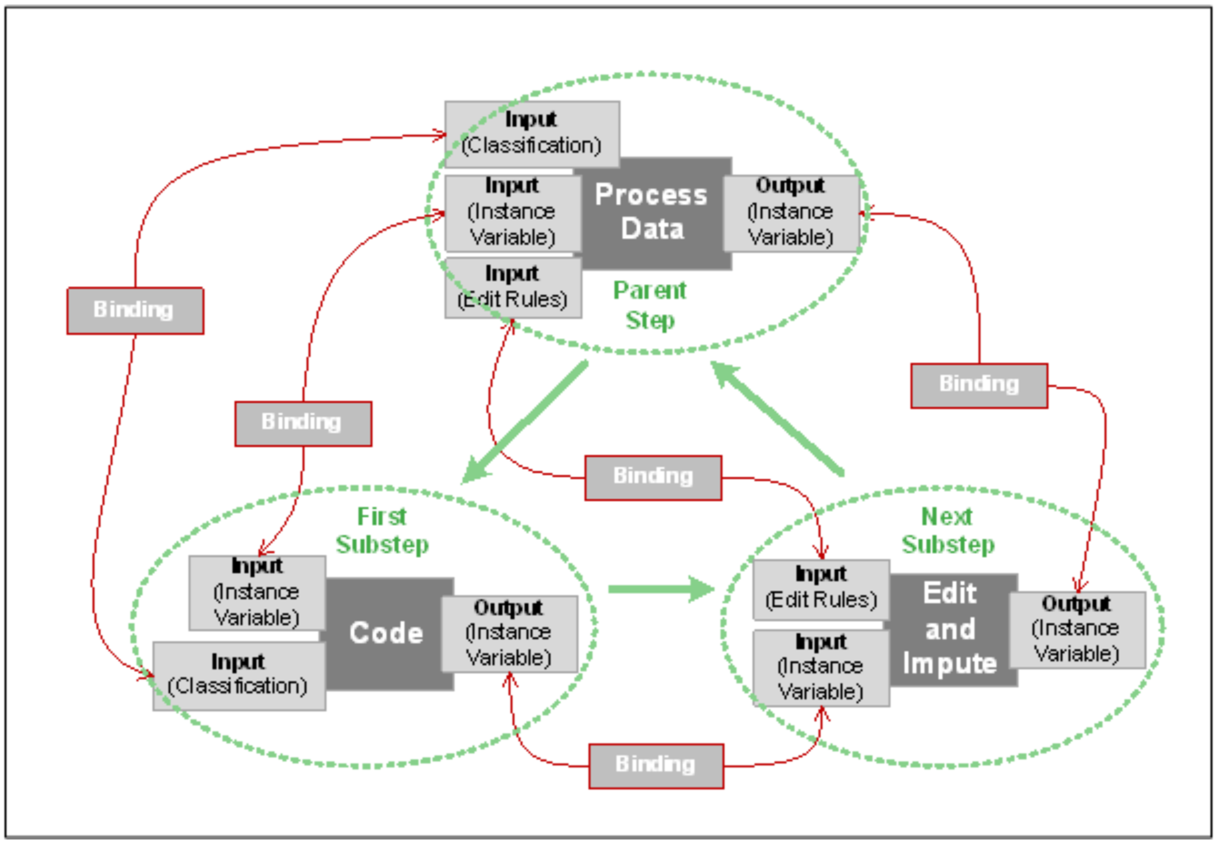
Figure 8. Binding



Let's illustrate the use of Bindings with an example.

Consider a Workflow that implements a fragment of the GSBPM Process phase. This is a specific implementation of GSBPM within the context of an organization in which some sub-steps are not implemented.

Figure 9. Bindings Example



The step at the top, Process Data, has four Parameters, three Input and one Output, each with its type between brackets, i.e. Classification, Instance Variable and Edit Rules for Input and Instance Variable for Output. This means that each parameter can hold at runtime only objects of the specified type.

There are also two sub-steps organized in a Workflow Sequence. The first sub-step, Code, has an Instance Variable and a Classification as Inputs and an Instance Variable as Output. The second sub-step, Edit and Impute, has the coded Instance Variable and Edit Rules as Input and an Instance Variable as Output.

How do we put all the pieces together? The top step gets a collected Instance Variable and produces a coded, edited and imputed Instance Variable based on a Classification and some Edit Rules. This is achieved by invoking the two sub-steps just described in sequence.

In order for this to work, the Input and Output Parameters of each step have to map so that the Instance Variable the top step receives moves thru the two sub-steps and is returned back, transformed, to the top step. This is made possible by the Bindings.

The first Binding on the left maps the Classification Input Parameter of Process Variable to that of the Code sub-step. This is an example of input-to-input Binding between two levels, i.e. a container step and its sub-steps. The bottom Binding maps the Instance Variable Output Parameter of Code to the Instance Variable Input Parameter of Edit and Impute. This is an example of an output-to-input Binding between steps at the same level, i.e. within the same Control Construct container. The other Bindings make sure that all Parameters are mapped.

5.3 The Variable Cascade

The DDI-Lifecycle standard is intended to address the metadata needs for the entire survey lifecycle. This particular document is dedicated to a description of variables as part of the DDI-Lifecycle. It contains a UML (Unified Modeling Language) class diagram of a model for describing variables, and the model is part of the larger development effort called DDI Moving Forward to express DDI-Lifecycle using UML.

Typical models for describing variables take advantage of much reuse, and the model provided here is no exception. It is reuse that makes metadata management such an effective approach. However, effectiveness is due to other factors as well, and an important one is to keep the number of objects described to a minimum. For finding relevant data is much more complicated as the number of objects rises.

For variables, reusable descriptions are brittle in the sense that if one of the related records to a variable changes, then a new variable needs to be defined. This is especially true when considering the allowed values (the Value Domain) for a variable. Many small variations in value domains exist in production databases, yet these differences are often gratuitous (e.g., simple differences in the way some category is described that do not alter the meaning), differences in representation (e.g., simple changes from letter codes to numeric ones), or differences in the way missing (or sentinel) values are represented.

Gratuitous differences in expressions of meaning are reduced or eliminated by encouraging the usage of URIs (Uniform Resource Identifiers) to point to definition entries in taxonomies of terms and concepts. The principle of "write once – use many", very similar to the idea of reuse, is employed. Pointing to an entry rather than writing its value eliminates transcription errors and simple expression differences, promotes comparability, and ensures interoperability.

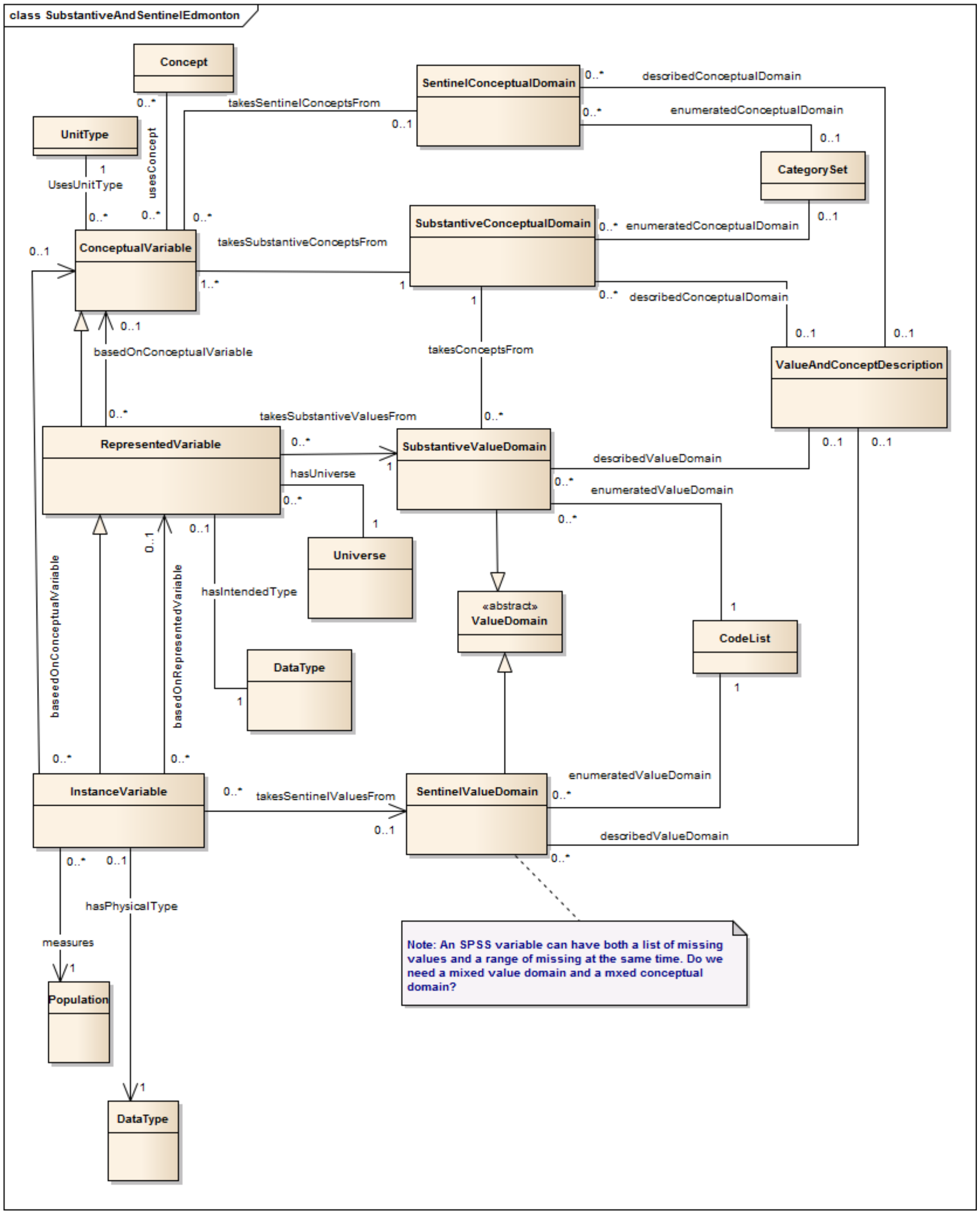
Differences in representations, including codes, are simplified by separating them from the underlying meaning. This is equivalent to the terminological issue of allowing for synonyms and homonyms of terms. Through reuse, all representations with the same meaning are linked to the same concept. This is achieved through the use of Value Domains and Conceptual Domains in the model presented here.

Sentinel values are important for any processing of statistical or experimental data, as there are multiple reasons some data are not obtainable. Typically, these values are added to the value domain for a variable. However, each time in the processing cascade the list of sentinel values changes, the value domain changes, which forces the variable to change as well. Given that each stage of the processing cascade make require a different set of sentinel values due to processing requirements, the number of variables mushrooms. And this variable proliferation is unmanageable and unsustainable.

The model developed here is based on two important standards for the statistical community, GSIM (Generic Statistical Information Model) and ISO/IEC 11179 (Metadata Registries). In fact, the model in the conceptual section of GSIM is based closely on the metamodel defined in ISO/IEC 11179. Both models help to perpetuate the problems described here, if each standard is followed in a conforming way. They reduce redundancy by separating value domains and conceptual domains. However, they do not directly support the use of URIs and they do not separate value domains from sentinel value lists. Also, they do not fully exploit the traceability that inheres in certain relationships between the value and conceptual domains to create a continuum of connected variables that further reduces redundancy.

The purpose of this document is to present a model that significantly reduces the overhead described above. In particular, we separate the sentinel values from the substantive ones. This separation allows us to greatly reduce the number of value domains, and thus variables, that need to be maintained. With this separation, now there are three classes of connected variables in which the represented variable specializes a conceptual variable by adding a value domain, and the instance variable specializes the represented variable by adding a sentinel value domain.

Figure 1. Variable Cascade


class SubstantiveAndSentinelEdmonton
Concept
SentinelConceptualDomain
UnitType
CategorySet
SubstantiveConceptualDomain
ConceptualVariable
ValueAndConceptDescription
RepresentedVariable
SubstantiveValueDomain
Universe
«abstract» ValueDomain
CodeList
DataType
InstanceVariable
SentinelValueDomain
Population
DataType
takesSentinelConceptsFrom
describedConceptualDomain
enumeratedConceptualDomain
usesConcept
UsesUnitType
takesSubstantiveConceptsFrom
basedOnConceptualVariable
takesConceptsFrom
takesSubstantiveValuesFrom
hasUniverse
describedValueDomain
enumeratedValueDomain
hasIntendedType
baseedOnConceptualVariable
basedOnRepresentedVariable
takesSentinelValuesFrom
hasPhysicalType
measures
Note: An SPSS variable can have both a list of missing values and a range of missing at the same time. Do we need a mixed value domain and a mxed conceptual domain?

5.3.1 Example

Detergents

Imagine we are assessing environmental influences at the household level. One question we might ask is "What detergents are used in the home?" In connection with this question we prepared show cards. Each show card lists a series of detergents. There are multiple show cards because products vary by region. Each show card corresponds to a code list. There are overlaps among the code lists but there are differences too.

In our model there is a conceptual variable. It has an enumerated conceptual domain that takes its categories from a category set. Here the category set is an unduplicated list of detergents taken from all the show cards put together. The conceptual domain might be exposure to chemicals in household detergents. The unit type might be households.

In our survey we ask a question corresponding to multiple represented variables, one corresponding to each show card. Each represented variable is measured by an enumerated value domain that takes its values from the show card code list.

All the represented variables here are derived from the same conceptual variable. This is the main point of the example: a conceptual variable under the right conditions can connect multiple represented variables.

Sentinel Values

The represented variable code list in our model excludes sentinel values. Sentinel values were introduced into ISO/IEC 11404 in 2007. ISO/IEC 11404 describes a set of language independent datatypes and defines a sentinel value as follows: a sentinel value is a "signaling" value such as nil, NaN (not a number), +inf and –inf (infinities), and so on. Depending on the study, sentinel values may include missing values. ISO 21090 is a health datatypes standard based on ISO/IEC 11404. ISO 21090 identifies 15 "nullflavors" that correspond to the concept of sentinel values introduced in ISO 11404 .

In our model the instance variable adds a sentinel value domain to the represented variable from which it is derived. In the process it grows the code list it derived from the represented variable to include a set of sentinel values. These sentinel values reference a category set of sentinels called the sentinel conceptual domain. The sentinel values included in any instance variable need not cover all the members of the sentinel conceptual domain. Instead they may refer just to a subset.

During data acquisition, we ask a question that allows don't know and refused, which an interviewer may ask or not, depending on the skip logic. We create an instance variable corresponding to this question based on a represented variable. The instance variable includes sentinel values. Note that the value domain of the represented variable need not be an enumerated value domain. Instead it can be a described value domain. We choose to include three sentinel values corresponding to three ISO 21090 nullflavors:

Nullflavor	Description
UNK	A proper value is applicable, but not known. Corresponds to Refused.
ASKU	Information was sought but not found Corresponds to Don't Know.
NA	No proper value is applicable in this context (e.g., last menstrual period for a male)

Subsequently, we prepare the collected data for processing by SAS. SAS has its own set of sentinel values. For each sentinel category the data acquisition instance variable accounts for, SAS has its own set of sentinel values. As a consequence, the answers that correspond to sentinel values are different, and in the process of constructing a SAS file, a second instance variable is created for the purposes of data processing.

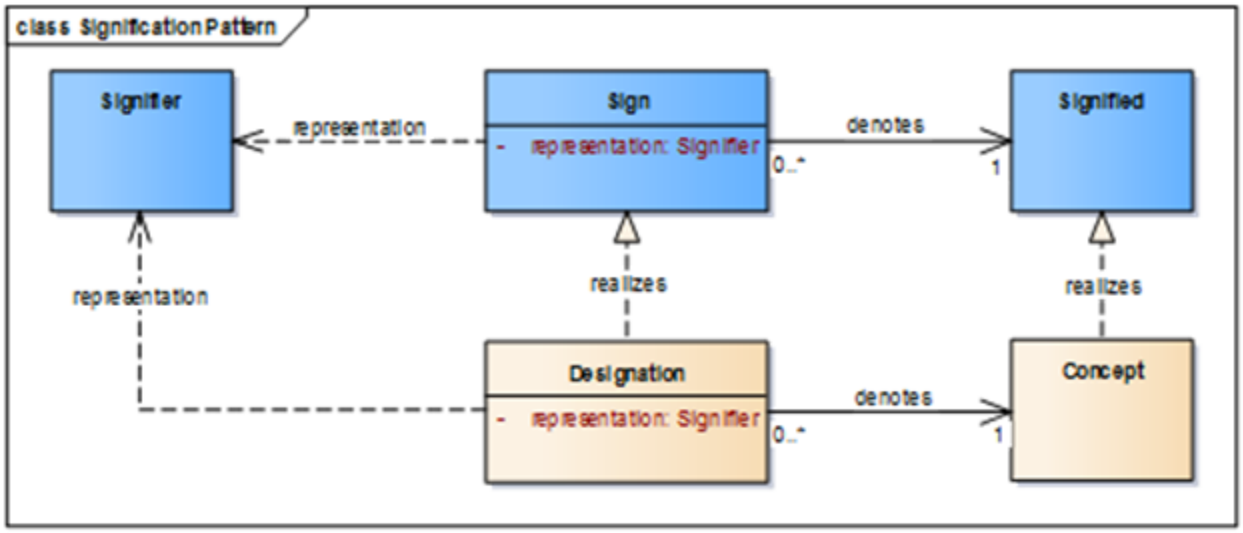
However, both the data acquisition instance variable and the data processing instance variable are derived from the same represented variable. That is the point of this example.

5.4 Using the Signification pattern

A Sign links a Signified with a Signifier that denotes it. A Signifier is a concept whose extension consists of tokens (perceivable objects).

Signifier, Sign and Signified become part of the Signification Pattern.

Figure 1. The Signification Pattern

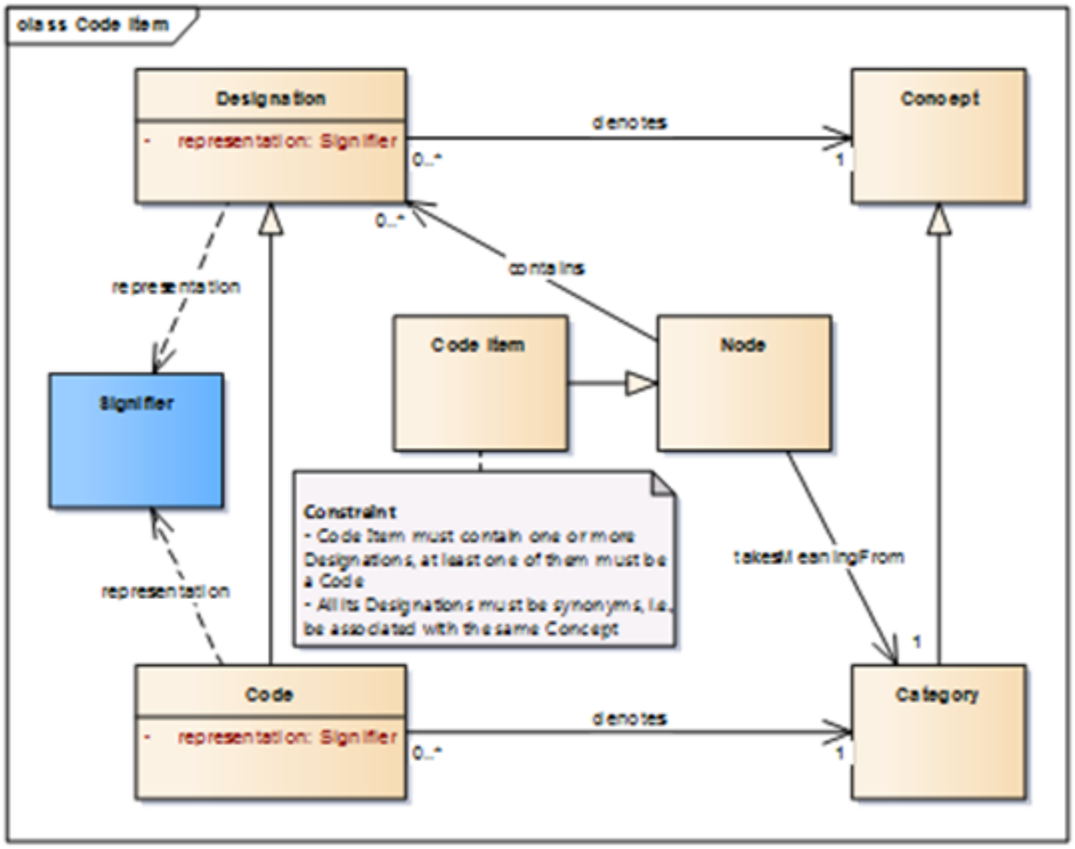


A Designation is simply a Sign where the Signified is a Concept. Therefore Designation and Sign realize Sign and Signified, respectively. Signifier becomes the Data Type of the representation property of Sign.

The reason for making Signifier, Sign and Signified into a pattern to be realized as opposed to classes to be extended is that Concepts are not always Signifieds, which is what a specialization would imply. In fact, a Concept is a Signified only if there is a Designation that denotes it. The realization means that the Concept is going to behave like a Signified only in the context described.

Codes enter into the picture as Designations. A Code then is a type of Designation that has Non-Linguistic Signifiers and where the Signified is a Category (Concept).

Figure 2. Code Item


class Code Item
Designation
- representation: Signifier
denotes
0..*
1
Concept
0..*
contains
representation
Code Item
Node
Signifier
Constraint
- Code Item must contain one or more Designations, at least one of them must be a Code
- All its Designations must be synonyms, i.e. be associated with the same Concept
takesMeaningFrom
1
representation
Code
- representation: Signifier
denotes
0..*
1
Category

A Node takes meaning from a single Category and has an optional set of Designations. In the case of the Code Item subtype, at least one Designation is required, i.e. a Code. It's important to note that a Code Item can have only one Category, thus if there are multiple Codes associated with a Code Item, all of them will have to denote the same Category (see constraint in diagram).

5.5 A Guide to Binary Relations

Object Anything perceivable or conceivable (from ISO 1087-1)

Perceivable Selectable through the senses or an instrument e.g. An apple, A thunder clap, A voltage

Conceivable Abstract or imagined e.g. A law, A standard, A unicorn, A hippogriff

Class Collection of objects sharing specified characteristics and behavior

Tuple Ordered set of objects, one from each of a given set of classes

Relationship Collection of tuples

Relation Meaning of a relationship *Note* We define relation and relationship as above because the same set of tuples may have more than one meaning attached. For example, take the set of two married cousins named John and Mary – call the set P. The elements, or tuples, of PxP (the cross product, or all possible pairs of elements of P) are <John, John>, <Mary, Mary>, <John, Mary>, and <Mary, John>. Now, consider two relations on this set of people: cousins and married. Both relations are anti-reflexive, symmetric, and intransitive (see below). So, we only need to consider the tuple <Mary, John>. This tuple makes up the relationship corresponding to both relations.

Binary relation A binary relation is one that takes 2 arguments. So, more formally, a binary relation applied to (more commonly "on") some set A is a subset of AxA. For example, if A = {1, 2, 3}, then the binary relation "=" (equality) on A is the following subset of AxA: {<1, 1>, <2, 2>, <3, 3>}. For instance, the pair <1, 2> is not in the subset defined by "=".

Total relation A total relation on a set A is the case for which all a, b in A, then either aRb, bRa, or both. This means every pair of elements are comparable. The "less than", or "<", relation on the set of integers is total.

Reflexive relation A reflexive relation on a set A is the case for which all a in A, aRa. For example, "equality", or "=", on the set of integers is reflexive. For example, 5 = 5.

Anti-reflexive An anti-reflexive, or irreflexive, relation on a set A is the case for which all a in A, then it is never the case that aRa. The strict "less than", or "<", relation on the set of integers is anti-reflexive. For example, 17 is not less than 17.

Symmetric relation A symmetric relation on a set A is the case for which all a, b in A, then if aRb implies bRa. The "equality" relation on the set of integers is symmetric.

Anti-symmetric An anti-symmetric relation on a set A is the case for which all a, b in A, then aRb and bRa implies a=b. An anti-symmetric relation is one where symmetry never holds. The strict "less than", or "<", relation on the set of integers is anti-symmetric. For example, since 4 < 5, then it is not the case that 5 < 4.

Transitive relation A transitive relation on a set A is the case for which all a, b, and c in A, then if aRb and bRc, then aRc. The strict "less than", or "<", relation on the set of integers is transitive. For example, since 42 < 69 and 69 < 1024, then 42 < 1024.

Anti-transitive An anti-transitive relation is one which for all a, b, and c in A, if aRb and bRc, then not aRc . Let A = {rock, paper, scissors}. We define a relation where rock beats scissors, scissors beats paper, and paper beats rock. But, given rBs and sBp, it is not the case that rBp. It is easy to see that no other transitivity case is true either. This relation is also anti-reflexive and anti-symmetric.

Intransitive relation An intransitive relation is one which for some but not all a, b, c in A, if aRb and bRc, then aRc. Let R be "distrusts" applied to countries. The US distrusts Iran, and Iran distrusts the UK, yet the US and the UK do not distrust each other. The US distrusts Iran, and Iran distrusts Iraq. Yet, the US distrusts Iraq also.

Partial order A partial order is reflexive, anti-symmetric, and transitive. This applies to the usual "less than or equal" relation on numbers. The strict "less than" relation is anti-reflexive. A partial order that is total is a total order, linear order, or chain. A strict partial order is anti-reflexive.

Graph A graph is a non-empty set of nodes and a set of arcs linking pairs of nodes. It is often useful to think of the nodes of a graph as representing some objects, which are the elements of some set, and the arcs of the graph as the relationships between those elements. We call this a represented graph.

Path A path is a sequence of nodes and arcs connecting two distinct nodes, where no nodes in the sequence are repeated. More precisely, let N.0 and N.n be two distinct nodes in a graph, then a path from N.0 to N.n is a sequence {N.0, a.1, N.1, a.2, . . . , N.n} of nodes and arcs, where nodes N.i &ne; N.j for all i and j, and a.i is an arc connecting N.i-1 to N.i.

Connected graph A connected graph is one for which there is a path between any pair of distinct nodes. A graph is acyclic if for nodes A, B, C and paths P and Q from nodes A to B and from nodes A to C, respectively, then there does not exist a path from B to C.

Tree A tree is a connected, acyclic graph with one node identified as the root. A hierarchy is a represented tree. A relation is hierarchical, a hierarchical relation, if when it is applied to some set of objects it can be represented in a tree.

5.5.1 Explanation of Relations

Less Than

- Hierarchical
- Anti-reflexive
- Anti-symmetric
- Transitive

Meaning: Relation between numbers comparing cardinalities Example – 7 is less than 12 (or 7 < 12)

Less Than or Equal

- Hierarchical
- Reflexive
- Anti-symmetric
- Transitive

Meaning: Relation between numbers comparing cardinalities Example – 7 is less than or equal to 12 (or 7 &le; 12), 10 is less than or equal to 10 (or 10 &le; 10)

Generic

- Hierarchical
- Anti-reflexive
- Anti-symmetric
- Transitive

Meaning: Sub-type; Specialization; Relation between classes whereby the characteristics and behaviors of one (generic) class are a proper subset of those for the other (specialized) class Example – ball (generic) and tennis ball (specialized)

Partitive

- Hierarchical
- Anti-reflexive
- Anti-symmetric
- Transitive

Meaning: Part/Whole, Relation between classes whereby one class constitutes the whole and the other class constitutes a part of that whole Example – car (whole) and engine (part)

Instantiation

- Hierarchical
- Anti-reflexive
- Anti-symmetric
- Intransitive

Meaning: Instance of Relation between classes whereby one class represents a class of objects and the other class represents individual instances of the first class Example – planet (class) and Saturn (instance)

Parent-Child

- Hierarchical
- Anti-reflexive
- Anti-symmetric

- Anti-transitive

Meaning: offspring of Relation between parents and their offspring. Mostly applies to people, but it can apply to other animals as well. Can apply to other situations, such as spawned processes in the UNIX operating system for computers Example – parent (Queen Elizabeth) and child (Prince Charles)

Sequential

- Can be cyclical (consider time as seen on a wall clock)
- Anti-reflexive
- Anti-symmetric
- Intransitive

Meaning: Relation between classes based on temporal or spatial proximity Example – production and consumption Note – A sequence can either be based on precedes and follows or it can be based on previous and next. This description is the general case. Either kind is possible. The next description is the more specific (previous and next) case.

Immediate Sequential

- Hierarchical
- Anti-reflexive
- Anti-symmetric
- Anti-transitive

Meaning: Relation between classes based on immediate temporal or spatial proximity Example – US and Canada

Temporal

- Hierarchical (if date is included, not if time is only measured by a 12 or 24 hour clock)
- Anti-reflexive
- Anti-symmetric
- Intransitive

Meaning: Sequential relation between classes involving events in time Example – spring and summer Note – A temporal sequence can either be based on precedes and follows or it can be based on previous and next. This description is the general case. Either kind is possible. The next description is the more specific (previous and next) case.

Immediate Temporal

- Hierarchical
- Anti-reflexive
- Anti-symmetric
- Anti-transitive

Meaning: Relation between classes based on immediate temporal proximity Example – 1 o'clock and 2 o'clock

Causal

- Can be cyclical
- Anti-reflexive
- Neither symmetric nor anti-symmetric
- Intransitive

Relation between classes involving cause and its effect Example – lightning and thunder; groups of teenaged boys punching each other

CHAPTER

SIX

PACKAGES

Warning: this is a development build, not a final product.

6.1 ComplexDataTypes

A package is a administrative collection of classes in DDI. These are not namespaces.

6.1.1 ArrayBaseCode

Graph

6.1.2 CategoryRelationCode

Graph

6.1.3 CollectionType

Graph

6.1.4 ComputationBaseList

Graph

6.1.5 ItemSequenceType

Graph

6.1.6 OrderRelationshipType

Graph

6.1.7 PointFormat

Graph

6.1.8 ReflexivityType

Graph

6.1.9 SexSpecificationType

Graph

6.1.10 ShapeCoded

Graph

6.1.11 SpatialObject

Graph

6.1.12 SymmetryType

Graph

6.1.13 TotalityType

Graph

6.1.14 TransitivityType

Graph

‘

Properties

Name	Type	Cardinality
externalURLReference	URL	0..n
internalURLReference	anyURI	0..1
mimeType	ExternalControlledVocabularyEntry	0..1
physicalLocation	InternationalString	0..n

externalURLReference

An external URL

internalURLReference

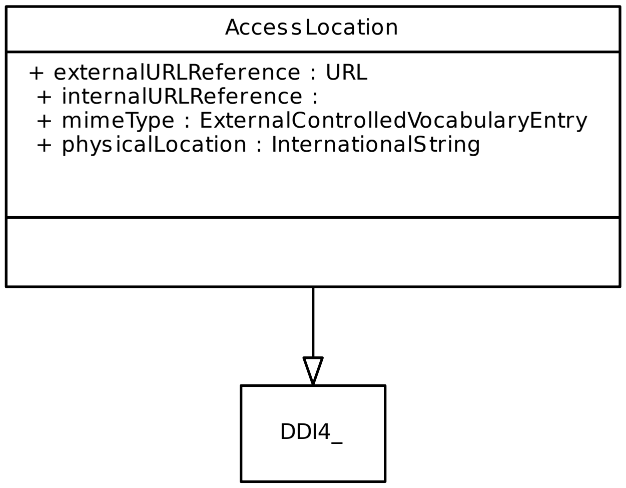
The internal URL.

mimeType

physicalLocation

The physical location of the machine

Graph


AccessLocation
+ externalURLReference : URL
+ internalURLReference :
+ mimeType : ExternalControlledVocabularyEntry
+ physicalLocation : InternationalString
DDI4_

6.1.17 Address

Location address identifying each part of the address as separate elements, identifying the type of address, the level of privacy associated with the release of the address, and a flag to identify the preferred address for contact.

‘

Properties

Name	Type	Cardinality
cityPlaceLocal	String	0..1
countryCode	ExternalControlledVocabularyEntry	0..1
effectivePeriod	UnlimitedNatural	0..1
geographicPoint	Point	0..1
isPreferred	Boolean	0..1
line	String	0..n
locationName	Name	0..1
postalCode	String	0..1
privacy	ExternalControlledVocabularyEntry	0..1
regionalCoverage	ExternalControlledVocabularyEntry	0..1
stateProvince	String	0..1
timeZone	ExternalControlledVocabularyEntry	0..1
typeOfAddress	ExternalControlledVocabularyEntry	0..1
typeOfLocation	ExternalControlledVocabularyEntry	0..1

cityPlaceLocal

City, place, or local area used as part of an address.

countryCode

Country of the location

effectivePeriod

Clarifies when the identification information is accurate.

geographicPoint

Geographic coordinates corresponding to the address.

isPreferred

Set to “true” if this is the preferred location for contacting the organization or individual.

line

Number and street including office or suite number. May use multiple lines.

locationName

Name of the location if applicable.

postalCode

Postal or ZIP Code

privacy

Specify the level privacy for the address as public, restricted, or private. Supports the use of an external controlled vocabulary

regionalCoverage

The region covered by the agent at this address

stateProvince

A major subnational division such as a state or province used to identify a major region within an address.

timeZone

Time zone of the location expressed as code.

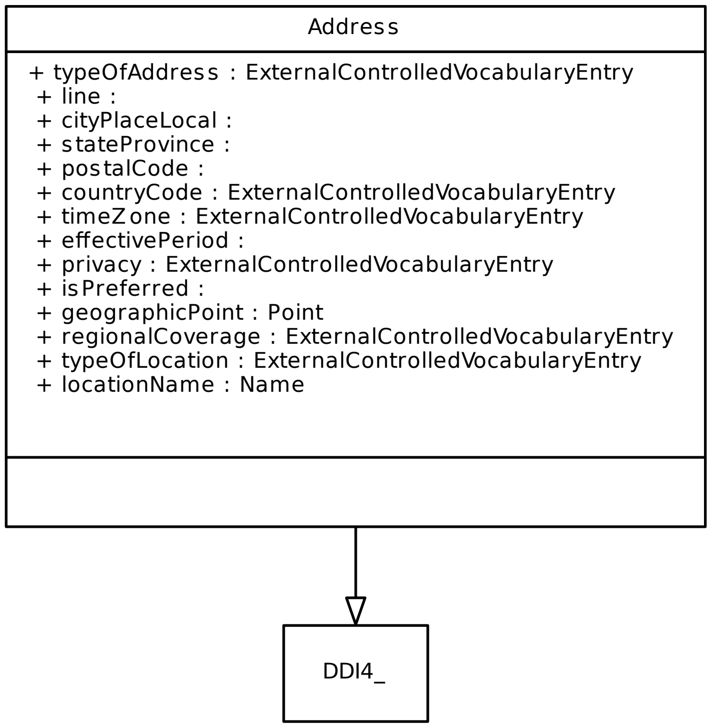
typeOfAddress

Indicates address type (i.e. home, office, mailing, etc.)

typeOfLocation

The type or purpose of the location (i.e. regional office, distribution center, home)

Graph


Address
+ typeOfAddress : ExternalControlledVocabularyEntry
+ line :
+ cityPlaceLocal :
+ stateProvince :
+ postalCode :
+ countryCode : ExternalControlledVocabularyEntry
+ timeZone : ExternalControlledVocabularyEntry
+ effectivePeriod :
+ privacy : ExternalControlledVocabularyEntry
+ isPreferred :
+ geographicPoint : Point
+ regionalCoverage : ExternalControlledVocabularyEntry
+ typeOfLocation : ExternalControlledVocabularyEntry
+ locationName : Name
DDI4_

6.1.18 AgentAssociation

A basic structure for declaring the name of an Agent inline, reference to an Agent, and role specification. This object is used primarily within Annotation.

‘

Properties

Name	Type	Cardinality
agent	BibliographicName	0..1
role	PairedExternalControlledVocabularyEntry	0..n

agent

Full name of the contributor. Language equivalents should be expressed within the International String structure.

role

The role of the of the Agent within the context of the parent property name with information on the extent to which the role applies. Allows for use of external controlled vocabularies. Reference should be made to the vocabulary within the structure of the role. A recommended role for contributors is the CASRAI Contributor Roles Vocabulary (CRediT) http://dictionary.casrai.org/Contributor_Roles

‘

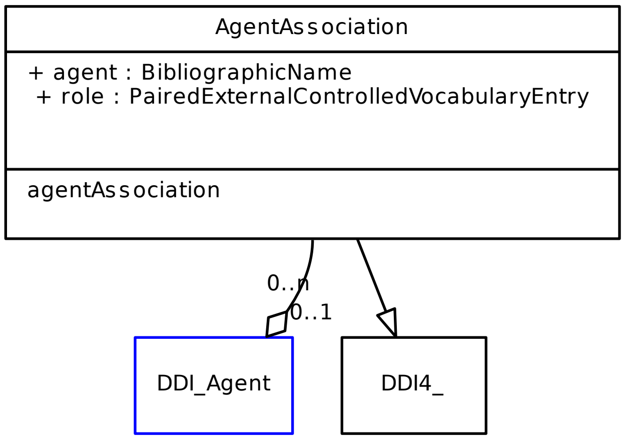
Relationships

Name	Type	Cardinality
agentAssociation	Agent	0..n

agentAssociation

Reference to an agent as described by any object that is a member of the abstract type Agent.

Graph


AgentAssociation
+ agent : BibliographicName
+ role : PairedExternalControlledVocabularyEntry
agentAssociation
0..n
0..1
DDI_Agent
DDI4_

6.1.19 AgentId

Persistent identifier for a researcher using a system like ORCID

‘

Properties

Name	Type	Cardinality
agentIdType	String	1..1
agentIdValue	String	1..1

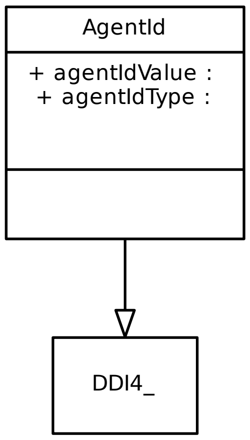
agentIdType

The identifier system in use.

agentIdValue

The identifier for the agent.

Graph



6.1.20 Annotation

Provides annotation information on the object to support citation and crediting of the creator(s) of the object.

‘

Properties

Name	Type	Cardinality
abstract	InternationalString	0..1
alternativeTitle	InternationalString	0..n
contributor	AgentAssociation	0..n
copyright	InternationalString	0..n
creator	AgentAssociation	0..n
date	AnnotationDate	0..n
identifier	InternationalIdentifier	0..n
informationSource	InternationalString	0..n
language		0..n
provenance	InternationalString	0..n
publisher	AgentAssociation	0..n
recordCreationDate	IsoDate	0..1
recordLastRevisionDate	IsoDate	0..1
relatedResource	ResourceIdentifier	0..n
rights	InternationalString	0..n
subTitle	InternationalString	0..n
title	InternationalString	0..1
typeOfResource	ExternalControlledVocabularyEntry	0..n
versionIdentification	String	0..1
versionResponsibility	AgentAssociation	0..n

abstract

An abstract (description) of the annotated object.

alternativeTitle

An alternative title by which a data collection is commonly referred, or an abbreviation for the title.

contributor

The name of a contributing author or creator, who worked in support of the primary creator given above.

copyright

The copyright statement.

creator

Person, corporate body, or agency responsible for the substantive and intellectual content of the described object.

date

A date associated with the annotated object (not the coverage period). Use typeOfDate to specify the type of date such as Version, Publication, Submitted, Copyrighted, Accepted, etc.

identifier

An identifier or locator. Contains identifier and Managing agency (ISBN, ISSN, DOI, local archive). Indicates if it is a URI.

informationSource

The name or identifier of source information for the annotated object.

language

Language of the intellectual content of the described object. Strongly recommend the use of language codes supported by xs:language which include the 2 and 3 character and extended structures defined by RFC4646 or its successors.

provenance

A statement of any changes in ownership and custody of the resource since its creation that are significant for its authenticity, integrity, and interpretation.

publisher

Person or organization responsible for making the resource available in its present form.

recordCreationDate

Date the record was created

recordLastRevisionDate

Date the record was last revised

relatedResource

Provide the identifier, managing agency, and type of resource related to this object.

rights

Information about rights held in and over the resource. Typically, rights information includes a statement about various property rights associated with the resource, including intellectual property rights.

subTitle

Secondary or explanatory title.

title

Full authoritative title. List any additional titles for this item as AlternativeTitle.

typeOfResource

Provide the type of the resource. This supports the use of a controlled vocabulary. It should be appropriate to the level of the annotation.

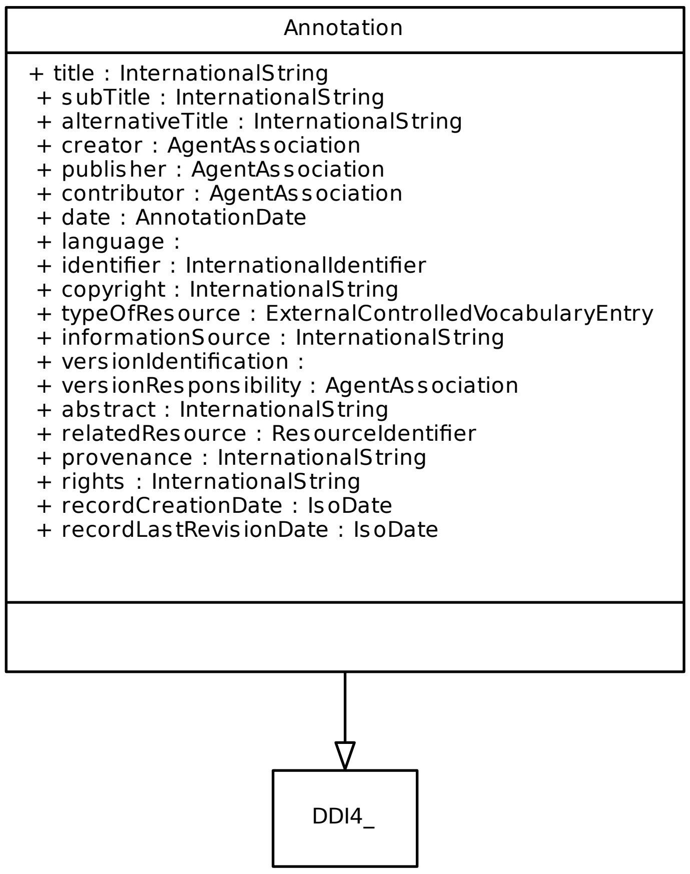
versionIdentification

Means of identifying the current version of the annotated object.

versionResponsibility

The agent responsible for the version. May have an associated role.

Graph


Annotation
+ title : InternationalString
+ subTitle : InternationalString
+ alternativeTitle : InternationalString
+ creator : AgentAssociation
+ publisher : AgentAssociation
+ contributor : AgentAssociation
+ date : AnnotationDate
+ language :
+ identifier : InternationalIdentifier
+ copyright : InternationalString
+ typeOfResource : ExternalControlledVocabularyEntry
+ informationSource : InternationalString
+ versionIdentification :
+ versionResponsibility : AgentAssociation
+ abstract : InternationalString
+ relatedResource : ResourceIdentifier
+ provenance : InternationalString
+ rights : InternationalString
+ recordCreationDate : IsoDate
+ recordLastRevisionDate : IsoDate
DDI4_

6.1.21 AnnotationDate

A generic date type for use in Annotation which provides that standard date structure plus a property to define the date type (Publication date, Accepted date, Copyrighted date, Submitted date, etc.). Equivalent of http://purl.org/dc/elements/1.1/date where the type of date may identify the Dublin Core refinement term.

Extends

Date

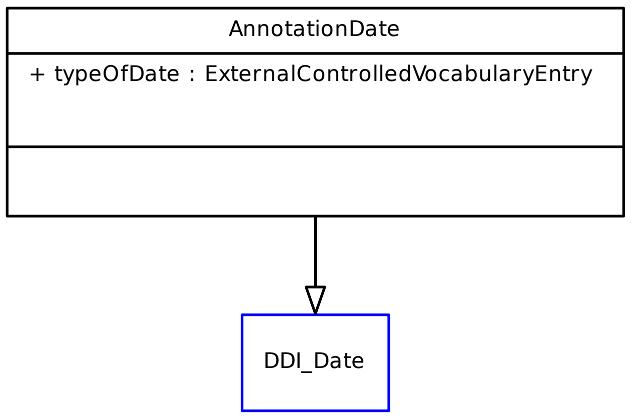
Properties

Name	Type	Cardinality
typeOfDate	ExternalControlledVocabularyEntry	0..n

typeOfDate

Use to specify the type of date. This may reflect the refinements of dc:date such as dateAccepted, dateCopyrighted, dateSubmitted, etc.

Graph



6.1.22 AudioSegment

Describes the type and length of the audio segment.

‘

Properties

Name	Type	Cardinality
audioClipBegin	String	0..1
audioClipEnd	String	0..1
typeOfAudioClip	ExternalControlledVocabularyEntry	0..1

audioClipBegin

The point to begin the audio clip. If no point is provided the assumption is that the start point is the beginning of the clip provided.

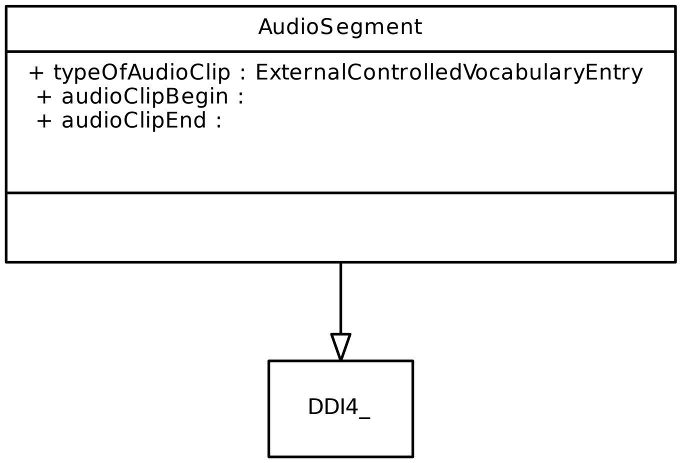
audioClipEnd

The point to end the audio clip. If no point is provided the assumption is that the end point is the end of the clip provided.

typeOfAudioClip

The type of audio clip provided. Supports the use of a controlled vocabulary.

Graph



6.1.23 BasedOnObject

Use when creating an object that is based on an existing object or objects that are managed by a different agency or when the new object is NOT simply a version change but you wish to maintain a reference to the object that served as a basis for the new object. BasedOnObject may contain references to any number of objects which serve as a basis for this object, a BasedOnRationaleDescription of how the content of the referenced object was incorporated or altered, and a BasedOnRationaleCode to allow for specific typing of the BasedOnReference according to an external controlled vocabulary.

‘

Properties

Name	Type	Cardinality
basedOnRationaleCode	ExternalControlledVocabularyEntry	0..1
basedOnRationaleDescription	InternationalString	0..1

basedOnRationaleCode

RationaleCode is primarily for internal processing flags within an organization or system. Supports the use of an external controlled vocabulary.

basedOnRationaleDescription

Textual description of the rationale/purpose for the based on reference to inform users as to the extent and implication of the version change. May be expressed in multiple languages.

‘

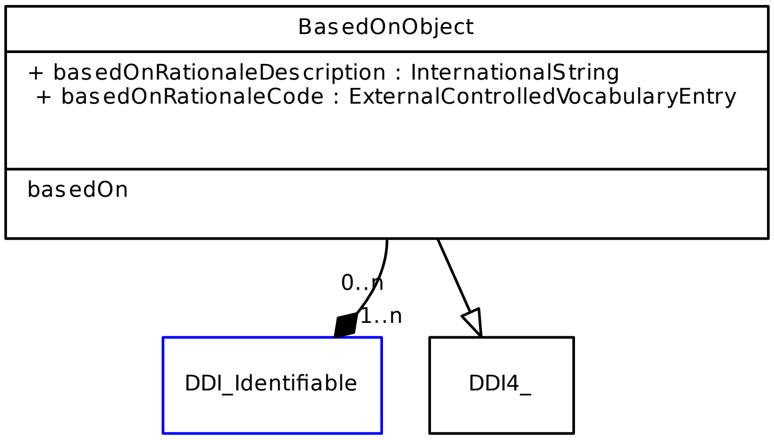
Relationships

Name	Type	Cardinality
basedOn	Identifiable	0..n

basedOn

The identification for the object upon which the current object is based.

Graph


BasedOnObject
+ basedOnRationaleDescription : InternationalString
+ basedOnRationaleCode : ExternalControlledVocabularyEntry
basedOn
0..n
1..n
DDI_Identifiable
DDI4_

6.1.24 BibliographicName

Personal names should be listed surname or family name first, followed by forename or given name. When in doubt, give the name as it appears, and do not invert. In the case of organizations where there is clearly a hierarchy present, list the parts of the hierarchy from largest to smallest, separated by full stops and a space. If it is not clear whether there is a hierarchy present, or unclear which is the larger or smaller portion of the body, give the name as it appears in the item. The name may be provided in one or more languages.

Extends

InternationalString

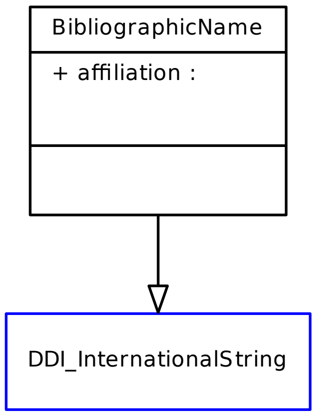
Properties

Name	Type	Cardinality
affiliation	String	0..1

affiliation

The affiliation of this person to an organization. This is generally an organization or sub-organization name and should be related to the specific role within which the individual is being listed.

Graph



6.1.25 CharacterOffset

Specification of the character offset for the beginning and end of the segment, or beginning and length.

‘

Properties

Name	Type	Cardinality
characterLength	Integer	0..1
endCharOffset	Integer	0..1
startCharOffset	Integer	0..1

characterLength

can be used to describe a text segment as start and length

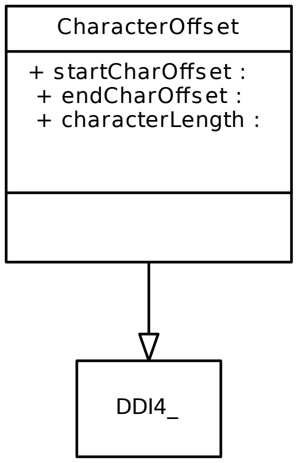
endCharOffset

Number of characters from the beginning of the document, indicating the inclusive end of the text segment.

startCharOffset

Number of characters from beginning of the document, indicating the inclusive start of the text range.

Graph



6.1.26 Command

Provides the following information on the command The content of the command, the programming language used, the pieces of information (InParameters) used by the command, the pieces of information created by the command (OutParameters) and the source of the information used by the InParameters (Binding).

‘

Properties

Name	Type	Cardinality
commandContent	String	0..1
programLanguage	ExternalControlledVocabularyEntry	0..1

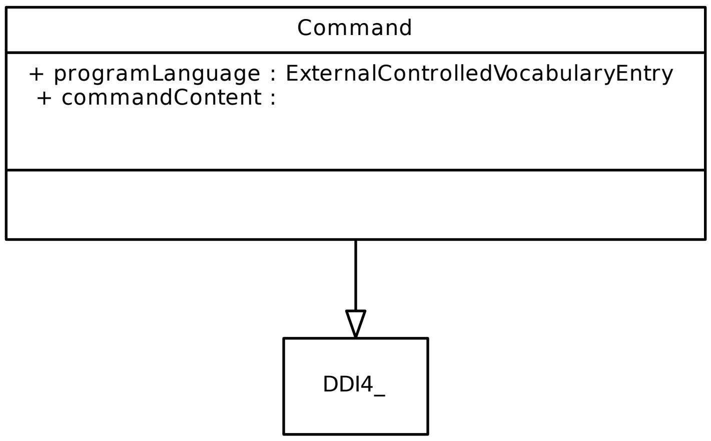
commandContent

Content of the command itself expressed in the language designated in Programming Language.

programLanguage

Designates the programming language used for the command. Supports the use of a controlled vocabulary.

Graph



6.1.27 CommandCode

Contains information on the command used for processing data. Contains a description of the command which should clarify for the user the purpose and process of the command, an in-line provision of the command itself, a reference to an external version of the command such as a coding script, and the option for attaching an extension to DDI to permit insertion of a command code in a foreign namespace. The definition of the InParameter, OutParameter, and Binding declared within CommandCode are available for use by all formats of the command.

‘

Properties

Name	Type	Cardinality
command	Command	0..n
commandFile	CommandFile	0..n
description	StructuredString	0..1
structuredCommand	StructuredCommand	0..1

command

This is an in-line provision of the command itself.

commandFile

Identifies and provides a link to an external copy of the command, for example, a SAS Command Code script. Designates the programming language of the command file, designates input and output parameters, binding information between input and output parameters, a description of the location of the file , and a URN or URL for the command file.

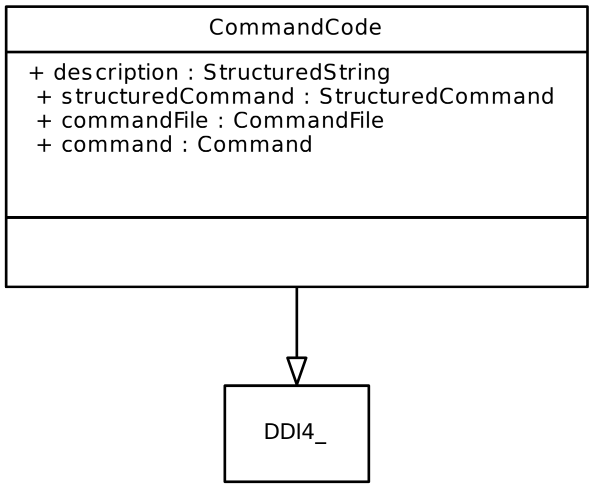
description

A description of the purpose and use of the command code provided. Supports multiple languages.

structuredCommand

The is an extension stub to allow for the insertion of command code using an external namespace.

Graph


CommandCode
+ description : StructuredString
+ structuredCommand : StructuredCommand
+ commandFile : CommandFile
+ command : Command
DDI4_

6.1.28 CommandFile

Identifies and provides a link to an external copy of the command, for example, a SAS Command Code script. Designates the programming language of the command file, designates input and output parameters, binding information between input and output parameters, a description of the location of the file , and a URN or URL for the command file.

‘

Properties

Name	Type	Cardinality
location	InternationalString	0..1
programLanguage	ExternalControlledVocabularyEntry	0..1
uri	anyURI	0..1

location

A description of the location of the file. This may not be machine actionable. It supports a description expressed in multiple languages.

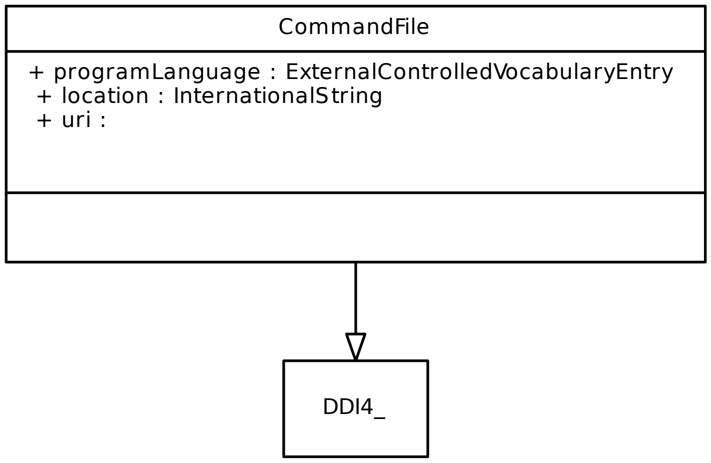
programLanguage

Designates the programming language used for the command. Supports the use of a controlled vocabulary.

uri

The URL or URN of the command file.

Graph


CommandFile
+ programLanguage : ExternalControlledVocabularyEntry
+ location : InternationalString
+ uri :
DDI4_

6.1.29 ConditionalText

Text which has a changeable value depending on a stated condition, response to earlier questions, or as input from a set of metrics (pre-supplied data).

Extends

TextContent

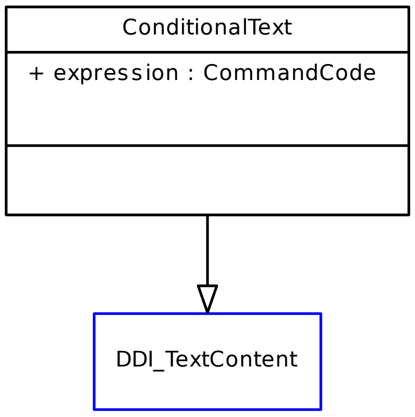
Properties

Name	Type	Cardinality
expression	CommandCode	0..1

expression

The condition on which the associated text varies expressed by a command code. For example, a command that inserts an age by calculating the difference between today's date and a previously defined date of birth.

Graph



6.1.30 ContactInformation

Contact information for the individual or organization including location specification, address, URL, phone numbers, and other means of communication access. Address, location, telephone, and other means of communication can be repeated to express multiple means of a single type or change over time. Each major piece of contact information (with the exception of URL) contains the element EffectiveDates in order to date stamp the period for which the information is valid.

'

Properties

Name	Type	Cardinality
address	Address	0..n
electronicMessaging	ElectronicMessageSystem	0..n
email	Email	0..n
telephone	Telephone	0..n
website	URL	0..n

address

The address for contact.

electronicMessaging

Electronic messaging other than email

email

Email contact information

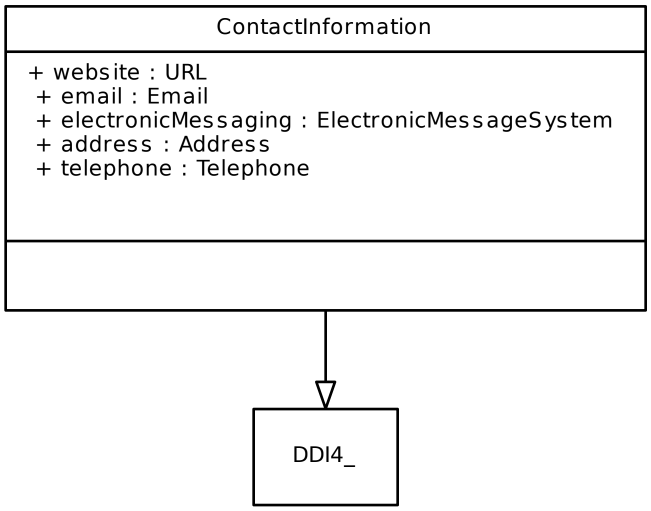
telephone

Telephone for contact

website

The URL of the Agent's website

Graph



6.1.31 Content

Supports the optional use of XHTML formatting tags within the string structure. XHTML tag content is controlled by the schema, see http://www.w3.org/1999/xhtml/ for a detailed list of available tags. Language of the string is defined by the attribute language. The content can be identified as translated (isTranslated), subject to translation (isTranslatable), the result of translation from one or more languages (translationSourceLanguages), and carry an indication whether or not it should be treated as plain text (isPlain).

‘

Properties

Name	Type	Cardinality
content	xhtml:BlkNoForm.mix	1..n
isPlainText	Boolean	0..1
isTranslatable	Boolean	0..1
isTranslated	Boolean	0..1
language		0..1
translationDate	UnlimitedNatural	0..1
translationSourceLanguage		0..n

content

The following xhtml tags are available for use in Content: address, blockquote, pre, h1, h2, h3, h4, h5, h6, hr, div, p, a, abbr, acronym, cite, code, dfn, em, kbd, q, samp, strong, var, b, big, i, small, sub, sup, tt, br, span, dl, dt, dd, ol, ul, li, table, caption, thead, tfoot, tbody, colgroup, col, tr, th, and td. They should be used with the xhtml namespace prefix, i.e., xhtmdiv. See DDI Technical Manual Part I for additional details.

isPlainText

Indicates that the content is to be treated as plain text (no formatting). You may use DDIProfile to fix the value of this attribute to 'true' in cases where you wish to indicate that your system treats all content should be treated as plain text.

isTranslatable

Indicates whether content is translatable (true) or not (false).

isTranslated

Indicates whether content is a translation (true) or an original (false).

language

Indicates the language of content. Note that language allows for a simple 2 or 3 character language code or a language code extended by a country code , for example en-au for English as used in Australia.

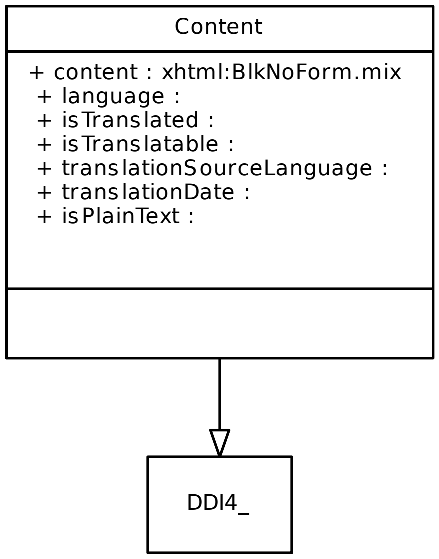
translationDate

The date the content was translated. Provision of translation date allows user to verify if translation was available during data collection or other time linked activity.

translationSourceLanguage

List the language or language codes in a space delimited array. The language original may or may not be provided in this bundle of language specific strings.

Graph



6.1.32 ContentDateOffset

Identifies the difference between the date applied to the data as a whole and this specific item such as previous year's income or residence 5 years ago. A value of true for the attribute isNegativeOffset indicates that the offset is the specified number of declared units prior to the date of the data as a whole and false indicates information regarding a future state.

Extends

ExternalControlledVocabularyEntry

Properties

Name	Type	Cardinality
isNegativeOffset	Boolean	0..1
numberOfUnits	Real	0..1

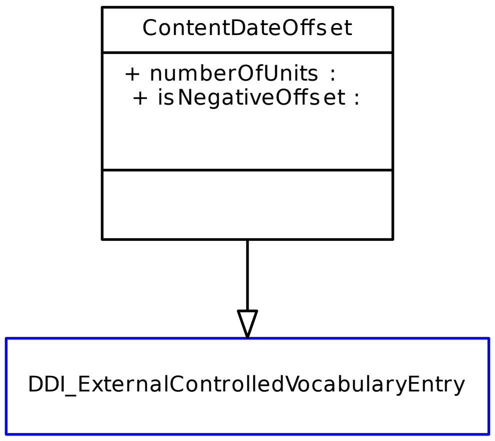
isNegativeOffset

If set to "true" the date is offset the number of units specified PRIOR to the default date of the data. A setting of "false" indicates a date the specified number of units in the FUTURE from the default date of the data.

numberOfUnits

The number of units to off-set the date for this item expressed as a decimal.

Graph



6.1.33 CorrespondenceType

Describes the commonalities and differences between two members using a textual description of both commonalities and differences plus an optional coding of the type of commonality.

‘

Properties

Name	Type	Cardinality
commonality	StructuredString	0..1
commonalityTypeCode	ExternalControlledVocabularyEntry	0..n
difference	StructuredString	0..1

commonality

A description of the common features of the two items using a StructuredString to support multiple language versions of the same content as well as optional formatting of the content.

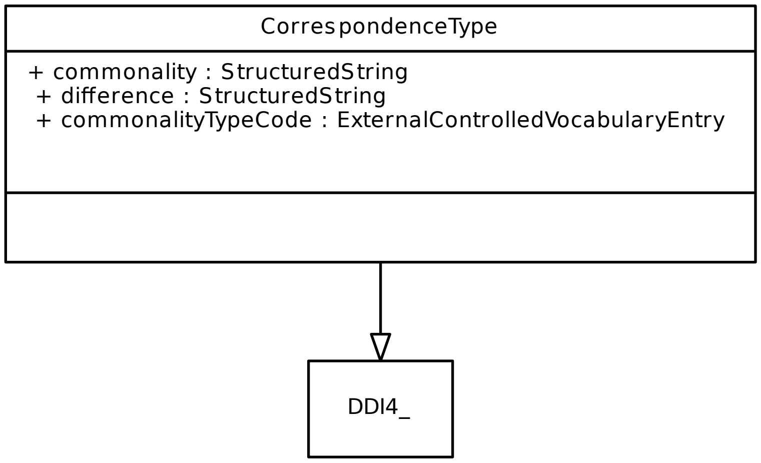
commonalityTypeCode

Commonality expressed as a term or code. Supports the use of an external controlled vocabulary. If repeated, clarify each external controlled vocabulary used.

difference

A description of the differences between the two items using a StructuredString to support multiple language versions of the same content as well as optional formatting of the content.

Graph


CorrespondenceType
+ commonality : StructuredString
+ difference : StructuredString
+ commonalityTypeCode : ExternalControlledVocabularyEntry
DDI4_

6.1.34 Date

Provides the structure of a single Date expressed in an ISO date structure along with equivalent expression in any number of non-ISO formats. While it supports the use of the ISO time interval structure this should only be used when the exact date is unclear (i.e. occurring at some point in time between the two specified dates) or in specified applications. Ranges with specified start and end dates should use the DateRange as a datatype. Commonly uses property names include: eventDate, issueDate, and releaseDate.

‘

Properties

Name	Type	Cardinality
isoDate	IsoDate	0..1
nonIsoDate	NonIsoDateType	0..n

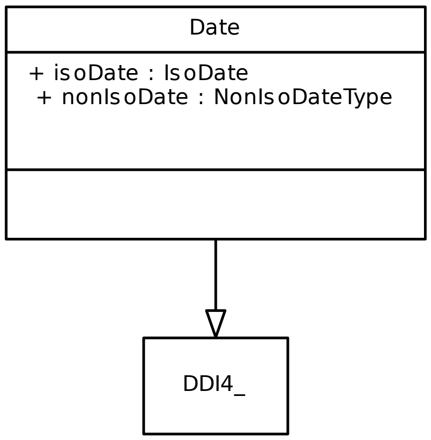
isoDate

Strongly recommend that ALL dates be expressed in an ISO format at a minimum. A single point in time expressed in an ISO standard structure. Note that while it supports an ISO date range structure this should be used in Date only when the single date is unclear i.e. occurring at some time between two dates.

nonIsoDate

A simple date expressed in a non-ISO date format, including a specification of the date format and calendar used.

Graph



6.1.35 DateRange

Expresses a date/time range using a start date and end date (both with the structure of Date and supporting the use of ISO and non-ISO date structures). Use in all locations where a range of dates is required, i.e. validFor, embargoPeriod, collectionPeriod, etc.

‘

Properties

Name	Type	Cardinality
endDate	Date	0..1
startDate	Date	0..1

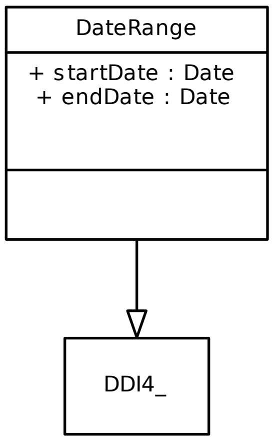
endDate

The date (time) designating the end of the period or range.

startDate

The date (time) designating the beginning of the period or range.

Graph



6.1.36 DescribedRelationship

Relationship specification between this item and the item to which it is related. Provides a reference to any identifiable object and a description of the relationship.

‘

Properties

Name	Type	Cardinality
rationale	StructuredString	0..1

rationale

Explanation of the reasons for relating the external material to the identified object. Supports the use of multiple languages and structured text.

‘

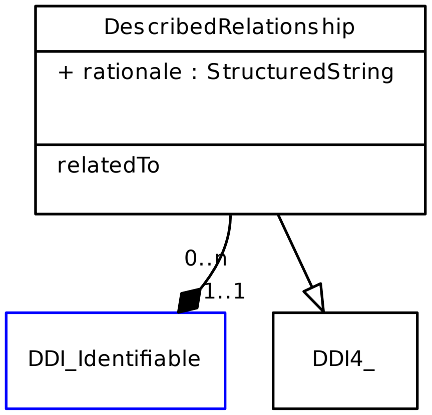
Relationships

Name	Type	Cardinality
relatedTo	Identifiable	0..n

relatedTo

Reference to the item within the DDI Instance to which this item is related.

Graph



6.1.37 DisplayLabel

A structured display label. Label provides display content of a fully human readable display for the identification of the object.

Extends

StructuredString

Properties

Name	Type	Cardinality
locationVariant	ExternalControlledVocabularyEntry	0..1
maxLength	Integer	0..1
validDates	DateRange	0..1

locationVariant

Indicate the locality specification for content that is specific to a geographic area. May be a country code, sub-country code, or area name.

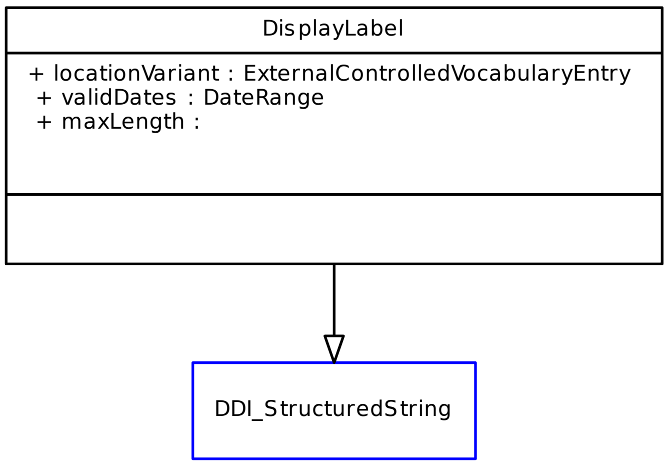
maxLength

A positive integer indicating the maximum number of characters in the label.

validDates

Allows for the specification of a starting date and ending date for the period that this label is valid.

Graph



6.1.38 DynamicText

Structure supporting the use of dynamic text, where portions of the textual content change depending on external information (pre-loaded data, response to an earlier query, environmental situations, etc.).

‘

Properties

Name	Type	Cardinality
audienceLanguage		0..1
content	TextContent	1..n
isStructureRequired	Boolean	0..1

audienceLanguage

Specifies the language of the intended audience. This is particularly important for clarifying the primary language of a mixed language textual string, for example when language testing and using a foreign word withing the question text.

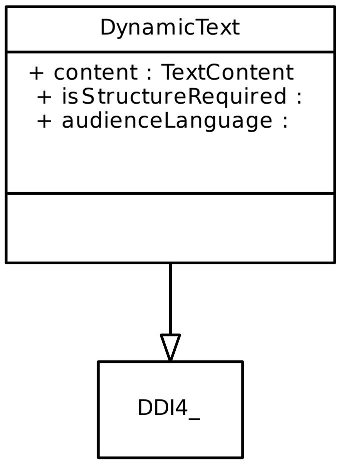
content

This is the head of a substitution group and is never used directly as an element name. Instead it is replaced with either LiteralText or ConditionalText.

isStructureRequired

If textual structure (e.g. size, color, font, etc.) is required to understand the meaning of the content change value to "true".

Graph



6.1.39 ElectronicMessageSystem

Any non-email means of relaying a message electronically. This would include text messaging, Skype, Twitter, ICQ, or other emerging means of electronic message conveyance.

'

Properties

Name	Type	Cardinality
contactAddress	String	0..1
effectiveDates	DateRange	0..1
isPreferred	Boolean	0..1
privacy	ExternalControlledVocabularyEntry	0..1
typeOfService	ExternalControlledVocabularyEntry	0..1

contactAddress

Account identification for contacting

effectiveDates

Time period during which the account is valid.

isPreferred

Set to “true” if this is the preferred address.

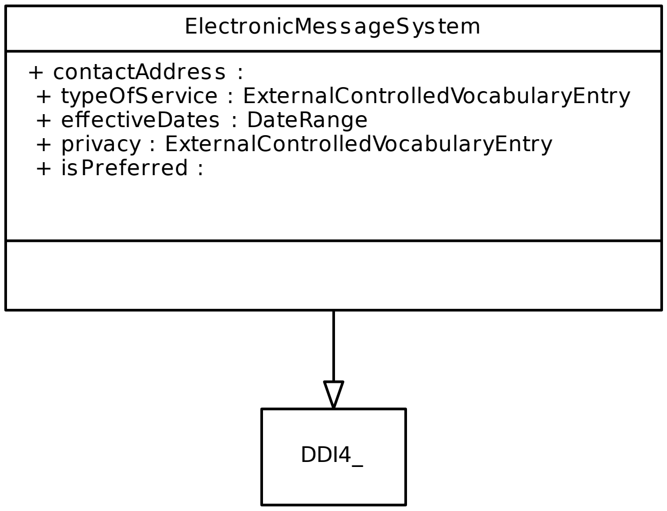
privacy

Specify the level privacy for the address as public, restricted, or private. Supports the use of an external controlled vocabulary.

typeOfService

Indicates the type of service used. Supports the use of a controlled vocabulary.

Graph


ElectronicMessageSystem
+ contactAddress :
+ typeOfService : ExternalControlledVocabularyEntry
+ effectiveDates : DateRange
+ privacy : ExternalControlledVocabularyEntry
+ isPreferred :
DDI4_

6.1.40 Email

An e-mail address which conforms to the internet format (RFC 822) including its type and time period for which it is valid.

‘

Properties

Name	Type	Cardinality
effectiveDates	DateRange	0..1
internetEmail	String	0..1
isPreferred	Boolean	0..1
privacy	ExternalControlledVocabularyEntry	0..1
typeOfEmail	ExternalControlledVocabularyEntry	0..1

effectiveDates

Time period for which the e-mail address is valid.

internetEmail

The email address expressed as a string (should follow the Internet format specification - RFC 5322) e.g. user@server.ext, more complex and flexible examples are also supported by the format.

isPreferred

Set to true if this is the preferred email

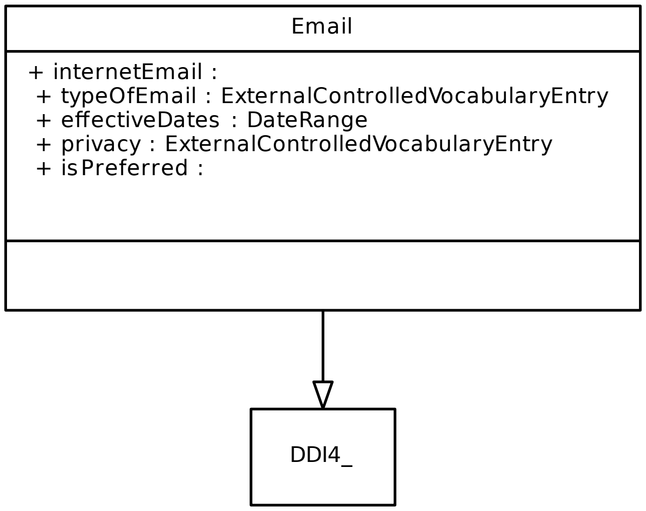
privacy

Indicates the level of privacy

typeOfEmail

Code indicating the type of e-mail address. Supports the use of an external controlled vocabulary. (e.g. home, office)

Graph


Email
+ internetEmail :
+ typeOfEmail : ExternalControlledVocabularyEntry
+ effectiveDates : DateRange
+ privacy : ExternalControlledVocabularyEntry
+ isPreferred :
DDI4_

6.1.41 ExternalControlledVocabularyEntry

Allows for unstructured content which may be taken from an externally maintained as an entry in a controlled vocabulary.If the content is from a controlled vocabulary provide the code value of the entry, as well as a reference to the controlled vocabulary from which the value is taken. Provide as many of the identifying attributes as needed to adequately identify the controlled vocabulary. Note that DDI has published a number of controlled vocabularies applicable to several locations using the ExternalControlledVocabularyEntry structure. If the code portion of the controlled vocabulary entry is language specific (i.e. a list of keywords or subject headings) use language to specify that language. In most cases the code portion of an entry is not language specific although the description and usage may be managed in one or more languages. Use of shared controlled vocabularies helps support interoperability and machine actionability.

‘

Properties

Name	Type	Cardinality
content	String	0..1
controlledVocabularyAgencyName	String	0..1
controlledVocabularyID	String	0..1
controlledVocabularyName	String	0..1
controlledVocabularySchemeURN	String	0..1
controlledVocabularyURN	String	0..1
controlledVocabularyVersionID	String	0..1
language		0..1
otherValue	String	0..1

content

The value of the entry of the controlled vocabulary. If no controlled vocabulary is used the term is entered here and none of the properties defining the controlled vocabulary location are used.

controlledVocabularyAgencyName

The name of the agency maintaining the code list.

controlledVocabularyID

The ID of the code list (controlled vocabulary) that the content was taken from.

controlledVocabularyName

The name of the code list.

controlledVocabularySchemeURN

If maintained within a scheme, the URN of the scheme containing the codelist.

controlledVocabularyURN

The URN of the codelist.

controlledVocabularyVersionID

The version number of the code list (default is 1.0).

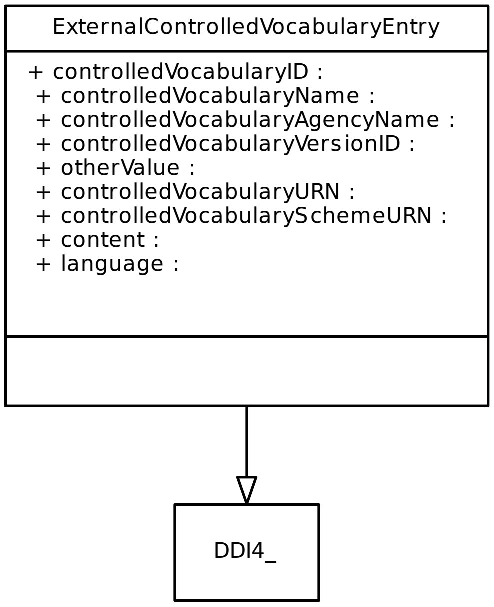
language

Language of the content value if applicable

otherValue

If the value of the string is “Other” or the equivalent from the codelist, this attribute can provide a more specific value not found in the codelist.

Graph


ExternalControlledVocabularyEntry
+ controlledVocabularyID :
+ controlledVocabularyName :
+ controlledVocabularyAgencyName :
+ controlledVocabularyVersionID :
+ otherValue :
+ controlledVocabularyURN :
+ controlledVocabularySchemeURN :
+ content :
+ language :
DDI4_

6.1.42 Form

A link to a form used by the metadata containing the form number, a statement regarding the contents of the form, a statement as to the mandatory nature of the form and a privacy level designation.

‘

Properties

Name	Type	Cardinality
formNumber	String	0..1
isRequired	Boolean	0..1
statement	InternationalString	0..1
uri	anyURI	0..1

formNumber

The number or other means of identifying the form.

isRequired

Set to “true” if the form is required. Set to “false” if the form is optional.

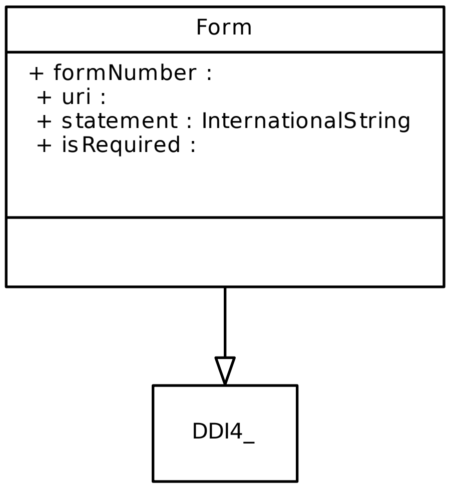
statement

A statement regarding the use, coverage, and purpose of the form.

uri

The URN or URL of the form.

Graph



6.1.43 Image

A reference to an image, with a description of its properties and type.

‘

Properties

Name	Type	Cardinality
dpi	Integer	0..1
imageLocation	anyURI	0..1
languageOfImage		0..1
typeOfImage	ExternalControlledVocabularyEntry	0..1

dpi

Provides the resolution of the image in dots per inch to assist in selecting the appropriate image for various uses.

imageLocation

A reference to the location of the image using a URI.

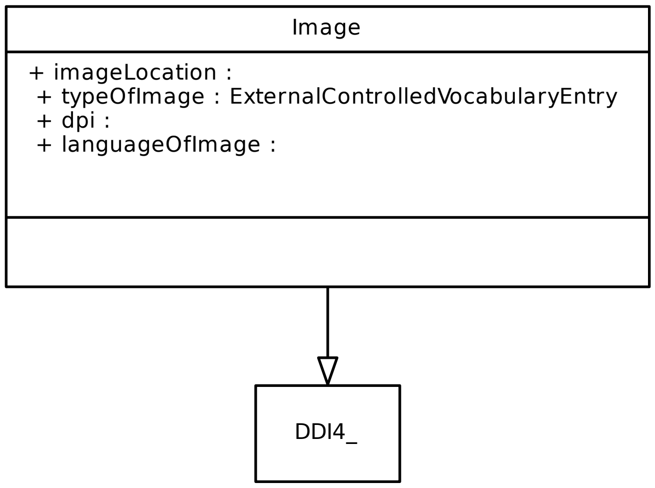
languageOfImage

Language of image.

typeOfImage

Brief description of the image type. Supports the use of an external controlled vocabulary.

Graph



6.1.44 ImageArea

Defines the shape and area of an image used as part of a location representation. The shape is defined as a Rectangle, Circle, or Polygon and Coordinates provides the information required to define it.

‘

Properties

Name	Type	Cardinality
coordinates	String	0..1
shape	ShapeCoded	0..1

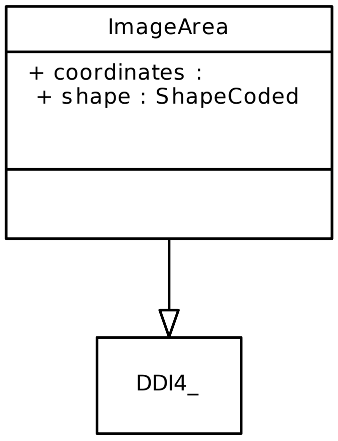
coordinates

A comma-delimited list of x,y coordinates, listed as a set of adjacent points for rectangles and polygons, and as a center-point and a radius for circles (x,y,r).

shape

A fixed set of valid responses includes Rectangle, Circle, and Polygon.

Graph



6.1.45 IndividualName

The name of an individual broken out into its component parts of prefix, first/given name, middle name, last/family/surname, and suffix. The preferred compilation of the name parts may also be provided. The legal or formal name of the individual should have the isFormal attribute set to true. The preferred name should be noted with the isPreferred attribute. The attribute sex provides information to assist in the appropriate use of pronouns.

‘

Properties

Name	Type	Cardinality
abbreviation	InternationalString	0..1
context	String	0..1
effectiveDates	DateRange	0..1
firstGiven	String	0..1
fullName	InternationalString	0..1
isFormal	Boolean	0..1
isPreferred	Boolean	0..1
lastFamily	String	0..1
middle	String	0..n
prefix	String	0..1
sex	SexSpecificationType	0..1
suffix	String	0..1
typeOfIndividualName	ExternalControlledVocabularyEntry	0..1

abbreviation

An abbreviation or acronym for the name. This may be expressed in multiple languages. It is assumed that if only a single language is provided that it may be used in any of the other languages within which the name itself is expressed.

context

A name may be specific to a particular context, i.e. common usage, business, social, etc.. Identify the context related to the specified name.

effectiveDates

Clarifies when the name information is accurate.

firstGiven

First (given) name of the individual

fullName

This provides a means of providing a full name as a single object for display or print such as identification badges etc. For example a person with the name of William Grace for official use may prefer a display name of Bill Grace on a name tag or other informal publication.

isFormal

The legal or formal name of the individual should have the isFormal attribute set to true. To avoid confusion only one individual name should have the isFormal attribute set to true. Use the TypeOfIndividualName to further differentiate the type and applied usage when multiple names are provided.

isPreferred

If more than one name for the object is provided, use the isPreferred attribute to indicate which is the preferred name content. All other names should be set to isPreferred=”false”.

lastFamily

Last (family) name /surname of the individual

middle

Middle name or initial of the individual

prefix

Title that precedes the name of the individual, such as Ms., or Dr.

sex

Sex allows for the specification of male, female or neutral. The purpose of providing this information is to assist others in the appropriate use of pronouns when addressing the individual. Note that many countries/languages may offer a neutral pronoun form.

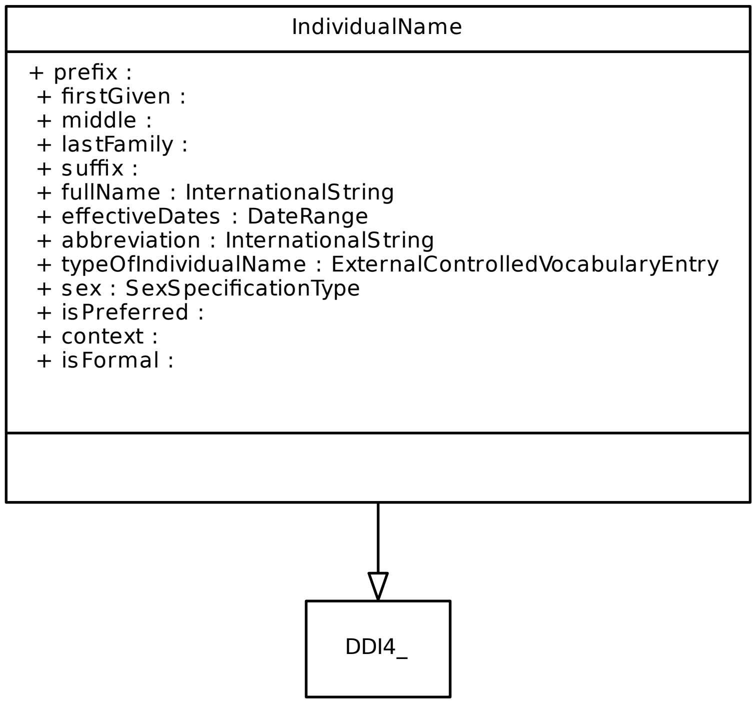
suffix

Title that follows the name of the individual, such as Esq.

typeOfIndividualName

The type of individual name provided. the use of a controlled vocabulary is strongly recommended. At minimum his should include, e.g. PreviousFormalName, Nickname (or CommonName), Other.

Graph


IndividualName
+ prefix :
+ firstGiven :
+ middle :
+ lastFamily :
+ suffix :
+ fullName : InternationalString
+ effectiveDates : DateRange
+ abbreviation : InternationalString
+ typeOfIndividualName : ExternalControlledVocabularyEntry
+ sex : SexSpecificationType
+ isPreferred :
+ context :
+ isFormal :
DDI4_

6.1.46 InternationalIdentifier

An identifier whose scope of uniqueness is broader than the local archive. Common forms of an international identifier are ISBN, ISSN, DOI or similar designator. Provides both the value of the identifier and the agency who manages it.

‘

Properties

Name	Type	Cardinality
identifierContent	String	0..1
isURI	Boolean	0..1
managingAgency	ExternalControlledVocabularyEntry	0..1

identifierContent

An identifier as it should be listed for identification purposes.

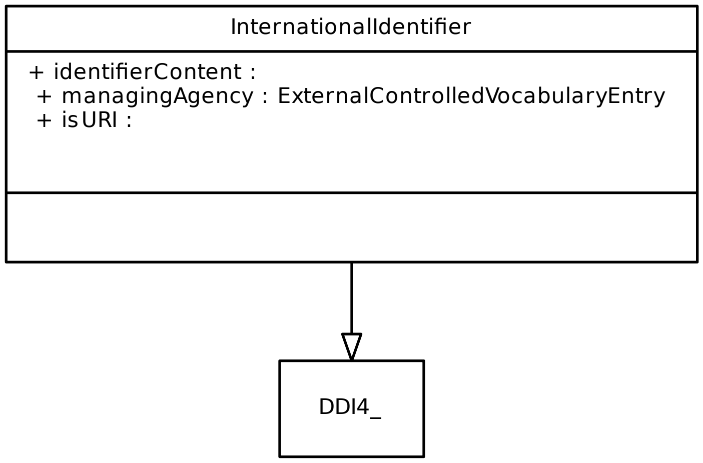
isURI

Set to “true” if Identifier is a URI

managingAgency

The identification of the Agency which assigns and manages the identifier, i.e., ISBN, ISSN, DOI, etc.

Graph



6.1.47 InternationalString

Packaging structure for multiple language versions of the same string content. Where an element of this type is repeatable, the expectation is that each repetition contains different content, each of which can be expressed in multiple languages. The language designation goes on the individual String.

‘

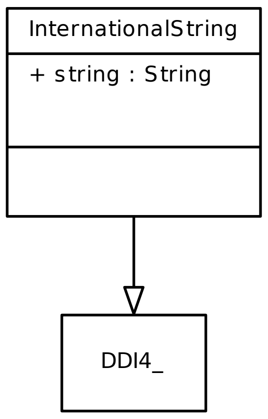
Properties

Name	Type	Cardinality
string	String	0..n

string

A non-formatted string of text with an attribute that designates the language of the text. Repeat this object to express the same content in another language.

Graph



6.1.48 LineParameter

Specification of the line and offset for the beginning and end of the segment.

‘

Properties

Name	Type	Cardinality
endLine	Integer	0..1
endOffset	Integer	0..1
startLine	Integer	0..1
startOffset	Integer	0..1

endLine

Number of lines from beginning of the document.

endOffset

Number of characters from the start of the line specified in EndLine.

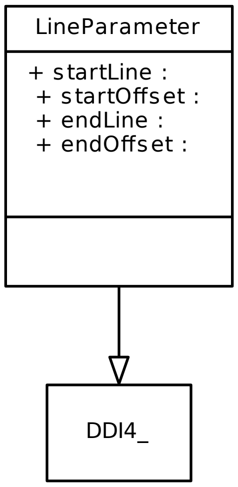
startLine

Number of lines from beginning of the document.

startOffset

Number of characters from start of the line specified in StartLine.

Graph



6.1.49 LiteralText

Literal (static) text to be used in the instrument using the StructuredString structure plus an attribute allowing for the specification of white space to be preserved.

Extends

TextContent

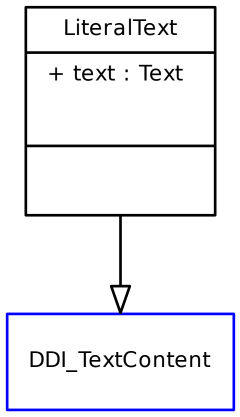
Properties

Name	Type	Cardinality
text	Text	0..1

text

The value of the static text string. Supports the optional use of XHTML formatting tags within the string structure. If the content of a literal text contains more than one language, i.e. "What is your understanding of the German word 'Gesundheit'?", the foreign language element should be placed in a separate LiteralText component with the appropriate xmlang value and, in this case, isTranslatable set to "false". If the existence of white space is critical to the understanding of the content (such as inclusion of a leading or trailing white space), set the attribute of Text xml:space to "preserve".

Graph



6.1.50 LocalId

This is an identifier in a given local context that uniquely references an object, as opposed to the full ddi identifier which has an agency plus the id.

'

Properties

Name	Type	Cardinality
localIdType	String	1..1
localIdValue	String	1..1
localIdVersion	String	0..1

localIdType

Type of identifier, specifying the context of the identifier.

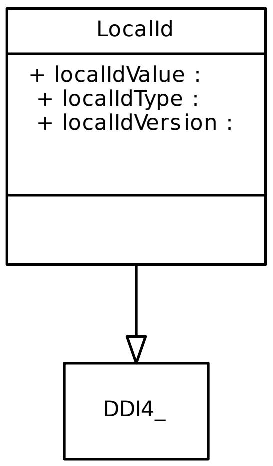
localIdValue

Value of the local ID.

localIdVersion

Version of the Local ID.

Graph



6.1.51 LocationName

Name of the location using the DDI Name structure and the ability to add an effective date.

Extends

Name

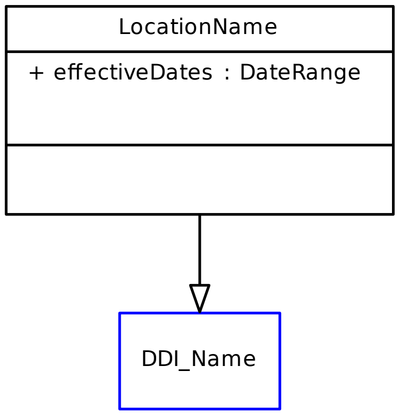
Properties

Name	Type	Cardinality
effectiveDates	DateRange	0..1

effectiveDates

The time period for which this name is accurate and in use.

Graph



6.1.52 Name

A standard means of expressing a Name for a class object. A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. In general the property name should be "name" as it is the name of the class object which contains it. Use a specific name (i.e. xxxName) only when naming something other than the class object which contains it.

'

Properties

Name	Type	Cardinality
content	String	0..1
context	ExternalControlledVocabularyEntry	0..1

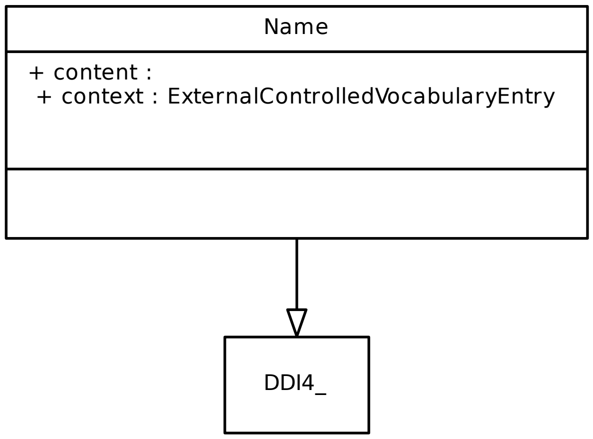
content

The expressed name of the object.

context

A name may be specific to a particular context, i.e., a type of software, or a section of a registry. Identify the context related to the specified name.

Graph



6.1.53 NonIsoDateType

Used to preserve an historical date, formatted in a non-ISO fashion.

‘

Properties

Name	Type	Cardinality
calendar	ExternalControlledVocabularyEntry	0..1
dateContent	String	1..1
nonIsoDateFormat	ExternalControlledVocabularyEntry	0..1

calendar

Specifies the type of calendar used (e.g., Gregorian, Julian, Jewish).

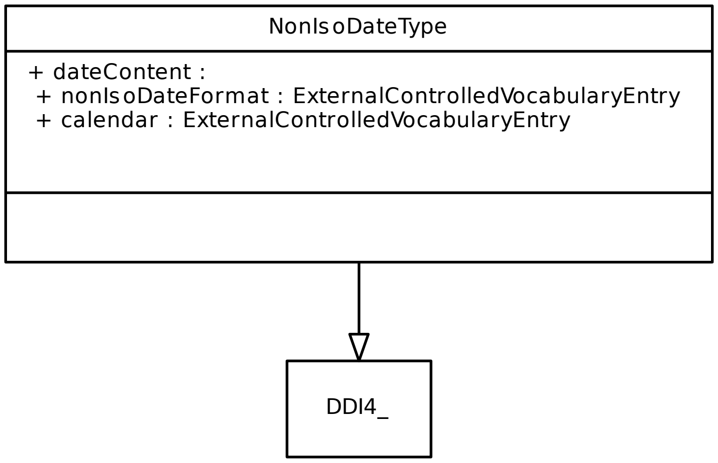
dateContent

This is the date expressed in a non-ISO compliant structure. Primarily used to retain legacy content or to express non-Gregorian calender dates.

nonIsoDateFormat

Indicate the structure of the date provided in NonISODate. For example if the NonISODate contained 4/1/2000 the Historical Date Format would be mm/dd/yyyy. The use of a controlled vocabulary is strongly recommended to support interoperability.

Graph


NonIsoDateType
+ dateContent :
+ nonIsoDateFormat : ExternalControlledVocabularyEntry
+ calendar : ExternalControlledVocabularyEntry
DDI4_

6.1.54 OrganizationName

Names by which the organization is known. Use the attribute isFormal=”true” to designate the legal or formal name of the Organization. The preferred name should be noted with the isPreferred attribute. Names may be typed with TypeOfOrganizationName to indicate their appropriate usage.

Extends

Name

Properties

Name	Type	Cardinality
abbreviation	InternationalString	0..1
effectiveDates	DateRange	0..1
isFormal	Boolean	0..1
typeOfOrganizationName	ExternalControlledVocabularyEntry	0..1

abbreviation

An abbreviation or acronym for the name. This may be expressed in multiple languages. It is assumed that if only a single language is provided that it may be used in any of the other languages within which the name itself is expressed.

effectiveDates

The time period for which this name is accurate and in use.

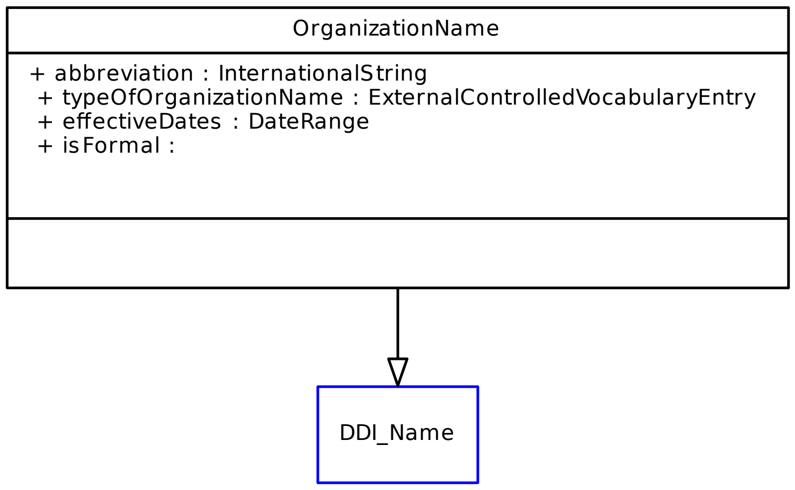
isFormal

The legal or formal name of the organization should have the isFormal attribute set to true. To avoid confusion only one organization name should have the isFormal attribute set to true. Use the TypeOfOrganizationName to further differentiate the type and applied usage when multiple names are provided.

typeOfOrganizationName

The type of organization name provided. the use of a controlled vocabulary is strongly recommended. At minimum this should include, e.g. PreviousFormalName, Nickname (or CommonName), Other.

Graph


OrganizationName
+ abbreviation : InternationalString
+ typeOfOrganizationName : ExternalControlledVocabularyEntry
+ effectiveDates : DateRange
+ isFormal :
DDI_Name

6.1.55 PairedExternalControlledVocabularyEntry

A tightly bound pair of items from an external controlled vocabulary. The extent property describes the extent to which the parent term applies for the specific case.

Extends

ExternalControlledVocabularyEntry

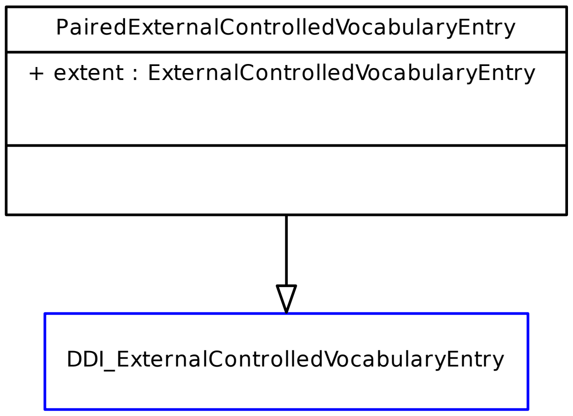
Properties

Name	Type	Cardinality
extent	ExternalControlledVocabularyEntry	0..1

extent

Describes the extent to which the parent term applies for the specific case using an external controlled vocabulary. When associated with a role from the CASRAI Contributor Roles Taxonomy an appropriate vocabulary should be specified as either 'lead', 'equal', or 'supporting'.

Graph


PairedExternalControlledVocabularyEntry
+ extent : ExternalControlledVocabularyEntry
DDI_ExternalControlledVocabularyEntry

6.1.56 Point

A geographic point consisting of an X and Y coordinate. Each coordinate value is expressed separately providing its value and format.

'

Properties

Name	Type	Cardinality
xCoordinate	SpatialCoordinate	0..1
yCoordinate	SpatialCoordinate	0..1

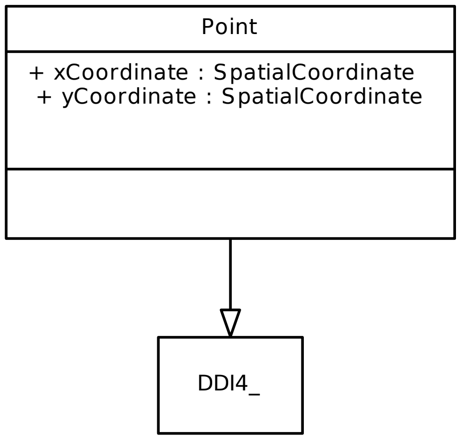
xCoordinate

An X coordinate (latitudinal equivalent) value and format expressed using the Spatial Coordinate structure.

yCoordinate

A Y coordinate (longitudinal equivalent) value and format expressed using the Spatial Coordinate structure.

Graph


Point
+ xCoordinate : SpatialCoordinate
+ yCoordinate : SpatialCoordinate
DDI4_

6.1.57 Polygon

A closed plane figure bounded by three or more line segments, representing a geographic area. Contains either the URI of the file containing the polygon, a specific link code for the shape within the file, and a file format, or a minimum of 4 points to describe the polygon in-line. Note that the first and last point must be identical in order to close the polygon. A triangle has 4 points. A geographic time designating the time period that the shape is valid should be included. If the date range is unknown use a SingleDate indicating a date that the shape was known to be valid.

‘

Properties

Name	Type	Cardinality
externalURI	anyURI	0..1
point	Point	4..n
polygonLinkCode	String	0..1
shapeFileFormat	ExternalControlledVocabularyEntry	0..1

externalURI

Note that ExternalURI points to the boundary file location.

point

A geographic point defined by a latitude and longitude. A minimum of 4 points is required as the first and last point should be identical in order to close the polygon. Note that a triangle has three sides and requires 3 unique points plus a fourth point replicating the first point in order to close the polygon.

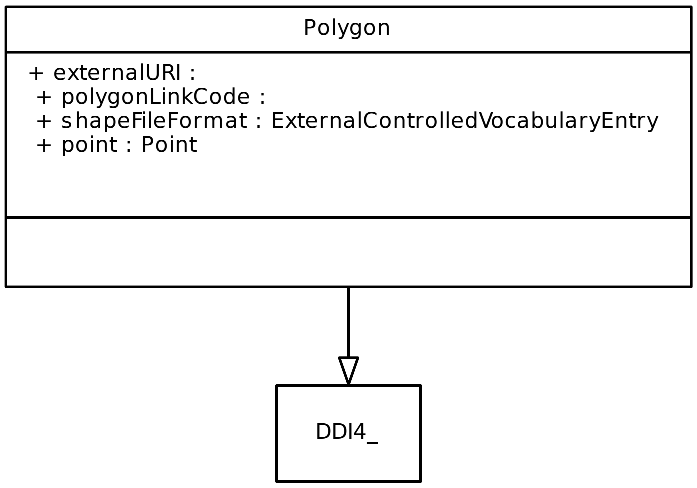
polygonLinkCode

The PolygonLinkCode is the identifier of the specific polygon within the file. For example in an NHGIS file the LinkCodeForPolygon for Tract 101.01 in Hennepin County in Minnesota is 2700530010101.

shapeFileFormat

The format of the shape file existing at the location indicated by the sibling ExternalURI element.

Graph



6.1.58 PrivateImage

References an image using the standard Image description. In addition to the standard attributes provides an effective date (period), the type of image, and a privacy ranking.

Extends

Image

Properties

Name	Type	Cardinality
effectiveDates	DateRange	0..1
privacy	ExternalControlledVocabularyEntry	0..1

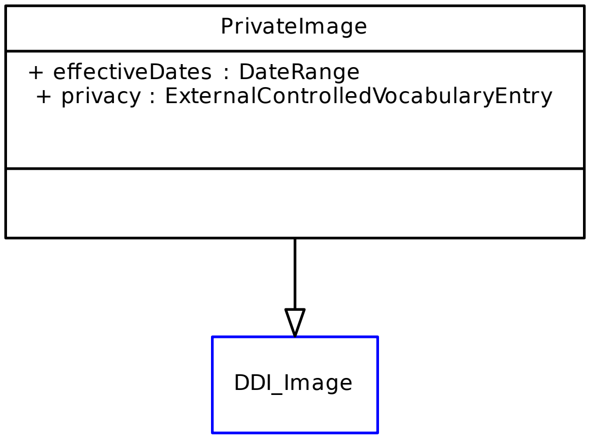
effectiveDates

The period for which this image is effective/valid.

privacy

Specify the level privacy for the image as public, restricted, or private. Supports the use of an external controlled vocabulary.

Graph


PrivateImage
+ effectiveDates : DateRange
+ privacy : ExternalControlledVocabularyEntry
DDI_Image

6.1.59 Range

Indicates the range of items expressed as a string, such as an alphabetic range.

‘

Properties

Name	Type	Cardinality
maximumValue	RangeValue	0..1
minimumValue	RangeValue	0..1
rangeUnit	String	0..1

maximumValue

Maximum value in the range.

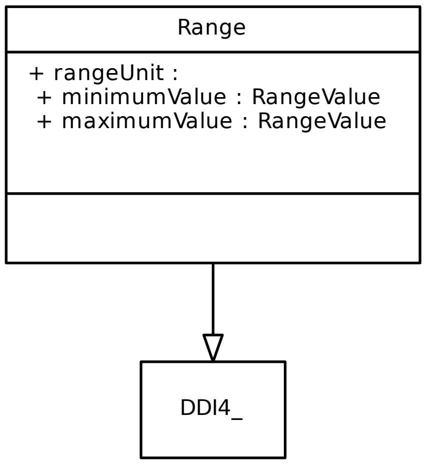
minimumValue

Minimum value in the range.

rangeUnit

Specifies the units in the range.

Graph



6.1.60 RangeValue

Describes a bounding value of a string.

Extends

Value

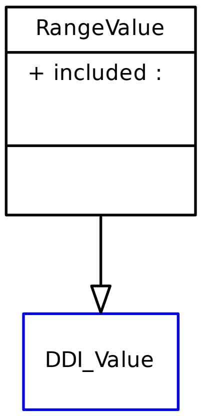
Properties

Name	Type	Cardinality
included	Boolean	0..1

included

Set to “true” if the value is included in the range.

Graph



6.1.61 ReferenceDate

The date covered by the annotated object. In addition to specifying a type of date (e.g. collection period, census year, etc.) the date or time span may be associated with a particular subject or keyword. This allows for the expression of a referent date associated with specific subjects or keywords. For example, a set of date items on income and labor force status may have a referent date for the year prior to the collection date. To express a duration the preference is for Start and End dates expressing two time points separated by a “/”. Note that if needed you may use Start and Duration or Duration and End. These options may not be recognizable by all systems.

Extends

AnnotationDate

Properties

Name	Type	Cardinality
keyword	ExternalControlledVocabularyEntry	0..n
subject	ExternalControlledVocabularyEntry	0..n

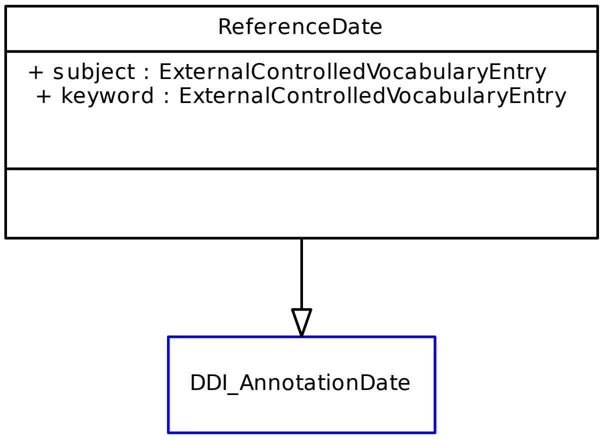
keyword

If the date is for a subset of data only such as a referent date for residence 5 years ago, use keyword to specify the coverage of the data this date applies to. May be repeated to reflect multiple keywords.

subject

If the date is for a subset of data only such as a referent date for residence 5 years ago, use Subject to specify the coverage of the data this date applies to. May be repeated to reflect multiple subjects.

Graph


ReferenceDate
+ subject : ExternalControlledVocabularyEntry
+ keyword : ExternalControlledVocabularyEntry
DDI_AnnotationDate

6.1.62 ResourceIdentifier

Provides a means of identifying a related resource and provides the typeOfRelationship. Makes use of a controlled vocabulary for typing the relationship. Standard usage may include: describesDate, isDescribedBy, isFormatOf, isPartOf, isReferencedBy, isReplacedBy, isRequiredBy, isVersionOf, references, replaces, requires, etc.

Extends

InternationalIdentifier

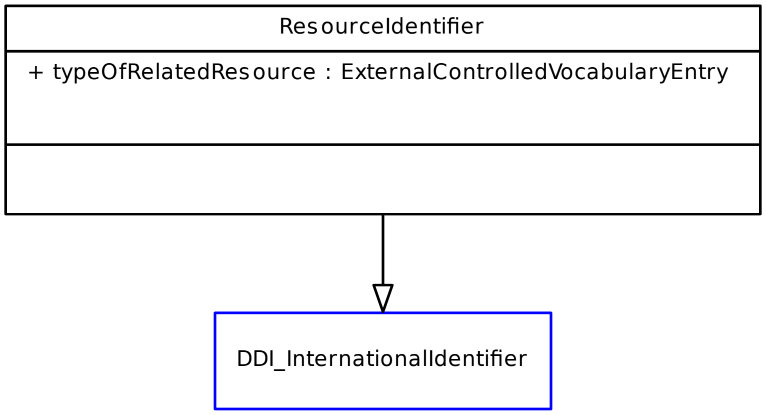
Properties

Name	Type	Cardinality
typeOfRelatedResource	ExternalControlledVocabularyEntry	0..n

typeOfRelatedResource

The type of relationship between the annotated object and the related resource. Standard usage may include: describesDate, isDescribedBy, isFormatOf, isPartOf, isReferencedBy, isReplacedBy, isRequiredBy, isVersionOf, references, replaces, requires, etc.

Graph


ResourceIdentifier
+ typeOfRelatedResource : ExternalControlledVocabularyEntry
DDI_InternationalIdentifier

6.1.63 Segment

A structure used to express explicit segments or regions within different types of external materials (Textual, Audio, Video, XML, and Image). Provides the appropriate start, stop, or region definitions for each type.

‘

Properties

Name	Type	Cardinality
audioSegment	AudioSegment	0..n
imageArea	ImageArea	0..n
textualSegment	TextualSegment	0..n
videoSegment	VideoSegment	0..n
xml	String	0..n

audioSegment

Describes the type and length of the audio segment.

imageArea

Defines the shape and area of an image used as part of a location representation. The shape is defined as a Rectangle, Circle, or Polygon and Coordinates provides the information required to define it.

textualSegment

Defines the segment of textual content used by the parent object. Can identify a set of lines and or characters used to define the segment

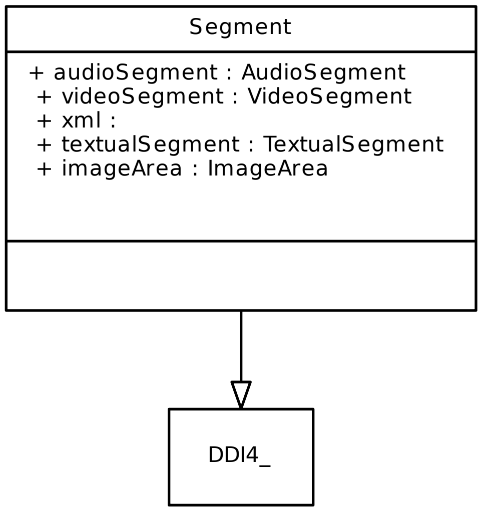
videoSegment

Describes the type and length of the video segment.

xml

An X-Pointer expression identifying a node in the XML document.

Graph


Segment
+ audioSegment : AudioSegment
+ videoSegment : VideoSegment
+ xml :
+ textualSegment : TextualSegment
+ imageArea : ImageArea
DDI4_

6.1.64 Software

Describes a specific software package, which may be commercially available or custom-made.

‘

Properties

Name	Type	Cardinality
function	ExternalControlledVocabularyEntry	0..n
issueDate	Date	0..1
packageLanguage		0..1
purpose	StructuredString	0..1
softwareName	Name	0..n
softwarePackage	ExternalControlledVocabularyEntry	0..1
softwareVersion	String	0..1
usage	StructuredString	0..1

function

Identifies the functions handled by this software. Repeat for multiple functions. It may be advisable to note only those functions used in the specific usage of the software.

issueDate

Supported date of the software package with, at minimum, a release date if known.

packageLanguage

Language (human language) of the software package. Note that language allows for a simple 2 or 3 character language code or a language code extended by a country code , for example en-au for English as used in Australia.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

softwareName

The name of the software package, including its producer.

softwarePackage

A coded value from a controlled vocabulary, describing the software package.

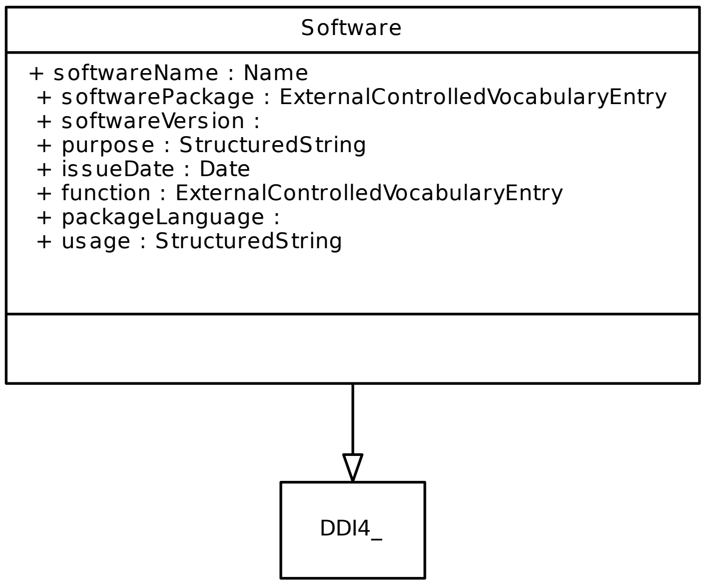
softwareVersion

The version of the software package. Defaults to ‘1.0’.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

Graph


Software
+ softwareName : Name
+ softwarePackage : ExternalControlledVocabularyEntry
+ softwareVersion :
+ purpose : StructuredString
+ issueDate : Date
+ function : ExternalControlledVocabularyEntry
+ packageLanguage :
+ usage : StructuredString
DDI4_

6.1.65 SpatialCoordinate

Lists the value and format type for the coordinate value. Note that this is a single value (X coordinate or Y coordinate) rather than a coordinate pair.

‘

Properties

Name	Type	Cardinality
coordinateType	PointFormat	0..1
coordinateValue	String	0..1

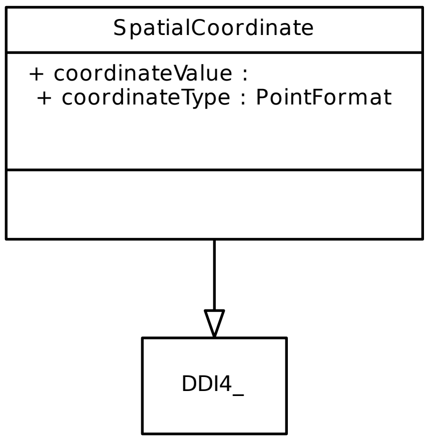
coordinateType

Identifies the type of point coordinate system using a controlled vocabulary. Point formats include decimal degree, degrees minutes seconds, decimal minutes, meters, and feet.

coordinateValue

The value of the coordinate expressed as a string.

Graph



6.1.66 SpecificSequence

Describes the ordering of items when not otherwise indicated. There are a set number of values for ItemSequenceType, but also a provision for describing an alternate ordering using a command language.

'

Properties

Name	Type	Cardinality
alternateSequence	CommandCode	0..1
itemSequence	ItemSequenceType	0..1

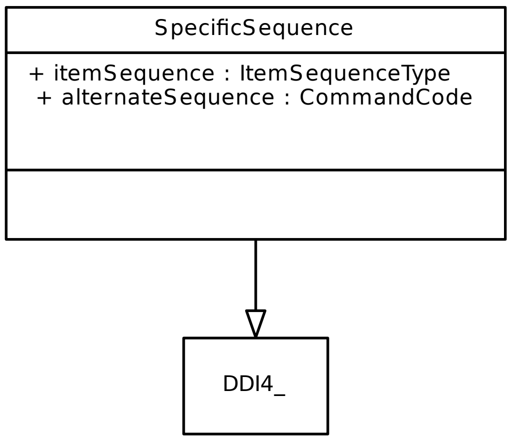
alternateSequence

Information on the command used to generate an alternative means of determining sequence changes. If used, the ItemSequenceType should be "Other".

itemSequence

Identifies the type of sequence to use. Values include InOrderOfAppearance, Random, Rotate, and Other.

Graph


SpecificSequence
+ itemSequence : ItemSequenceType
+ alternateSequence : CommandCode
DDI4_

6.1.67 StandardKeyValuePair

A basic data representation for computing systems and applications expressed as a tuple (attribute key, value). Attribute keys may or may not be unique.

‘

Properties

Name	Type	Cardinality
attributeKey	ExternalControlledVocabularyEntry	0..1
attributeValue	ExternalControlledVocabularyEntry	0..1

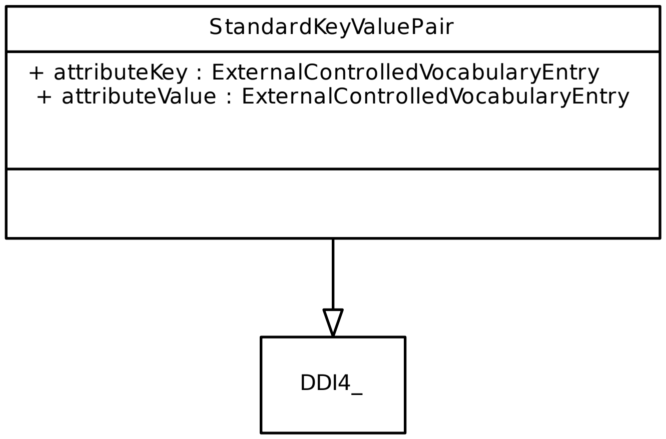
attributeKey

This key (sometimes referred to as a name) expressed as a string. Supports the use of an external controlled vocabulary which is the recommended approach.

attributeValue

The value assigned to the named Key expressed as a string. Supports the use of an external controlled vocabulary.

Graph


StandardKeyValuePair
+ attributeKey : ExternalControlledVocabularyEntry
+ attributeValue : ExternalControlledVocabularyEntry
DDI4_

6.1.68 Statistic

The value of the statistics and whether it is weighted and/or includes missing values.

‘

Properties

Name	Type	Cardinality
computationBase	ComputationBaseList	0..1
isWeighted	Boolean	0..1
value	Real	0..n

computationBase

Defines the cases included in determining the statistic. The options are total=all cases, valid and missing (invalid); validOnly=Only valid values, missing (invalid) are not included in the calculation; missingOnly=Only missing (invalid) cases included in the calculation.

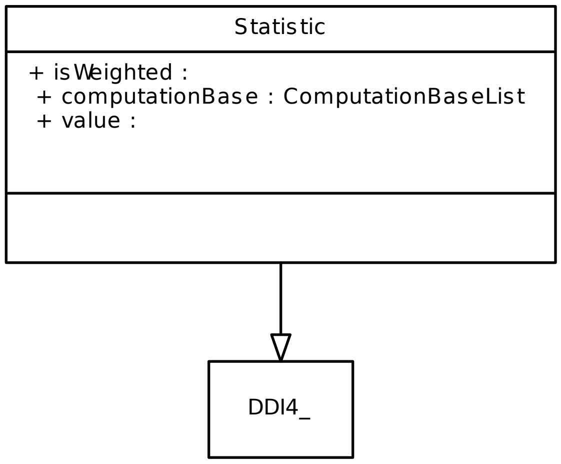
isWeighted

Set to “true” if the statistic is weighted using the weight designated in VariableStatistics.

value

The value of the statistic expressed as a decimal

Graph



6.1.69 String

Allows for non-formatted strings that may be translations from other languages, or that may be translatable into other languages. Only one string per language/location type is allowed. String contains the following attributes, xmlang to designate the language, isTranslated with a default value of false to designate if an object is a translation of another language, isTranslatable with a default value of true to designate if the content can be translated, translationSource-Language to indicate the source languages used in creating this translation, and translationDate.

‘

Properties

Name	Type	Cardinality
content	String	1..1
isTranslatable	Boolean	0..1
isTranslated	Boolean	0..1
language		0..1
translationDate	UnlimitedNatural	0..1
translationSourceLanguage		0..n

content

Value of this string

isTranslatable

Indicates whether content is translatable (true) or not (false). An example of something that is not translatable would be a MNEMONIC of an object or a number.

isTranslated

Indicates whether content is a translation (true) or an original (false).

language

Indicates the language of content. Note that language allows for a simple 2 or 3 character language code or a language code extended by a country code , for example en-au for English as used in Australia.

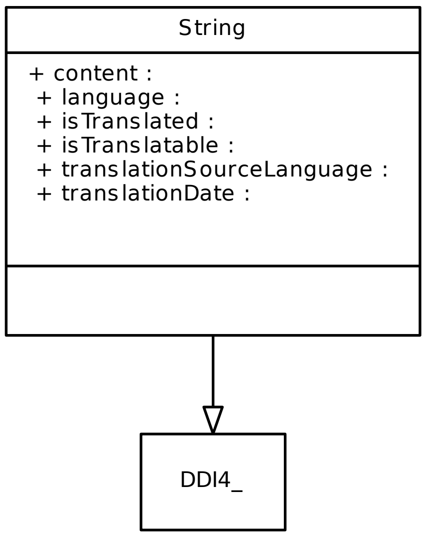
translationDate

The date the content was translated. Provision of translation date allows user to verify if translation was available during data collection or other time linked activity.

translationSourceLanguage

List the language code of the source. Repeat of multiple language sources are used.

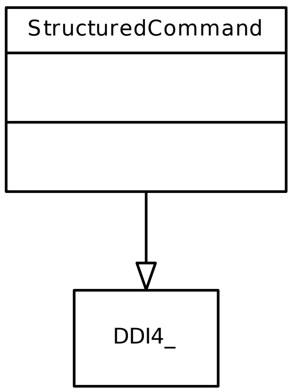
Graph



6.1.70 StructuredCommand

This type structures an empty stub which is used as the basis for extensions added using external namespaces such as MathML. The DDI 3.0 extension methodology is used here - a new module is declared, and the StructuredCommand element is used as the head of a substitution group to insert whatever XML is needed to express the command.

Graph



6.1.71 StructuredString

Packaging structure for multiple language versions of the same string content, for objects that allow for internal formatting using XHTML tags. Where an element of this type is repeatable, the expectation is that each repetition contains different content, each of which can be expressed in multiple languages.

‘

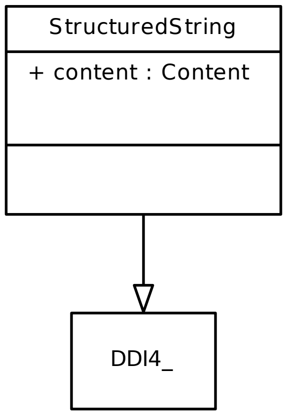
Properties

Name	Type	Cardinality
content	Content	1..n

content

Supports the optional use of XHTML formatting tags within the string structure. In addition to the language designation and information regarding translation, the attribute isPlain can be set to true to indicate that the content should be treated as plain unstructured text, including any XHTML formatting tags. Repeat the content element to provide multiple language versions of the same content.

Graph



6.1.72 TargetSample

Specifies details of target sample size, percentage, type, and universe

‘

Properties

Name	Type	Cardinality
isPrimary	Boolean	1..1
targetPercent	Real	0..1
targetSize	Integer	0..1

isPrimary

Default value is True. The information is for the primary sample unit or universe. Change to False if the information relates to a secondary or sub- sample universe.

targetPercent

The target size of the sample expressed as an integer (70% expressed as .7)

targetSize

The target size of the sample expressed as an integer.

‘

Relationships

Name	Type	Cardinality
fromUniverse	Universe	0..n
hasUnitType	UnitType	0..n

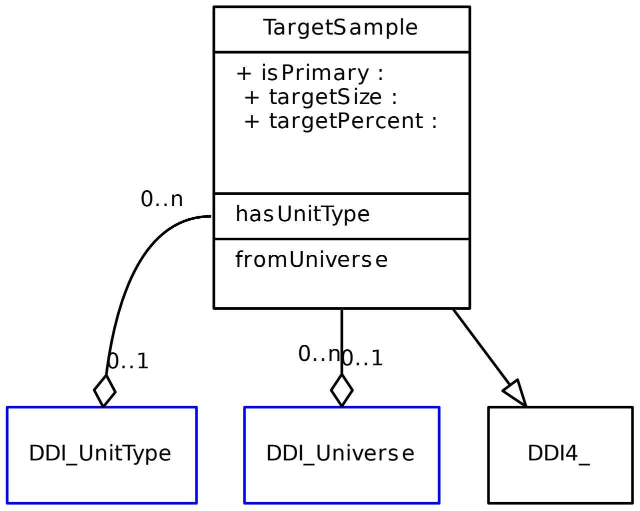
fromUniverse

The universe of the sample. Note that when the Design is implemented the Population should be a Population within this Universe.

hasUnitType

Unit type of the sample

Graph



6.1.73 Telephone

Details of a telephone number including the number, type of number, a privacy setting and an indication of whether this is the preferred contact number.

‘

Properties

Name	Type	Cardinality
effectiveDates	DateRange	0..1
isPreferred	Boolean	0..1
privacy	ExternalControlledVocabularyEntry	0..1
telephoneNumber	String	0..1
typeOfTelephone	ExternalControlledVocabularyEntry	0..1

effectiveDates

Time period during which the telephone number is valid.

isPreferred

Set to "true" if this is the preferred telephone number for contact.

privacy

Specify the level privacy for the telephone number as public, restricted, or private. Supports the use of an external controlled vocabulary.

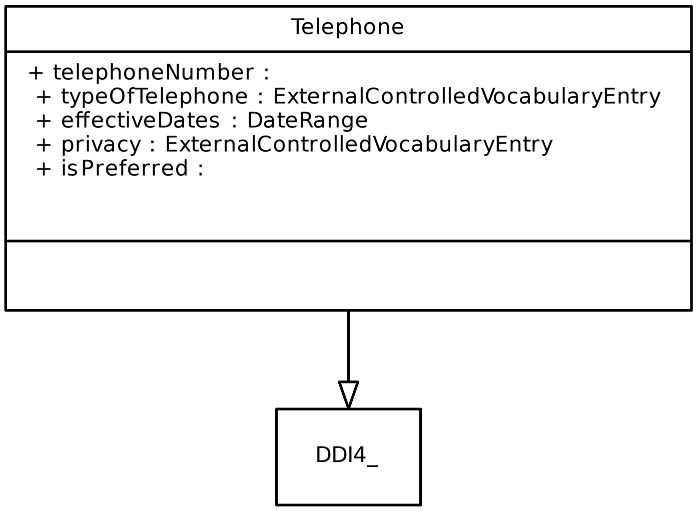
telephoneNumber

The telephone number including country code if appropriate.

typeOfTelephone

Indicates type of telephone number provided (home, fax, office, cell, etc.). Supports the use of a controlled vocabulary.

Graph


Telephone
+ telephoneNumber :
+ typeOfTelephone : ExternalControlledVocabularyEntry
+ effectiveDates : DateRange
+ privacy : ExternalControlledVocabularyEntry
+ isPreferred :
DDI4_

6.1.74 Text

The static portion of the text expressed as a StructuredString with the ability to preserve whitespace if critical to the understanding of the content.

Extends

Content

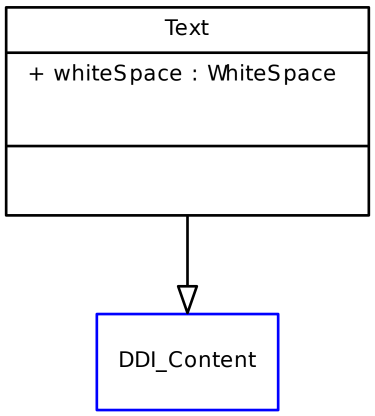
Properties

Name	Type	Cardinality
whiteSpace	WhiteSpace	0..1

whiteSpace

The default setting states that leading and trailing white space will be removed and multiple adjacent white spaces will be treated as a single white space. If the existance of any of these white spaces is critical to the understanding of the content, change the value of this attribute to “preserve”.

Graph



6.1.75 TextContent

Abstract type existing as the head of a substitution group. May be replaced by any valid member of the substitution group TextContent. Provides the common property of purpose to all members using TextContent as an extension base.

‘

Properties

Name	Type	Cardinality
orderPosition	Integer	0..1
purpose	StructuredString	0..1

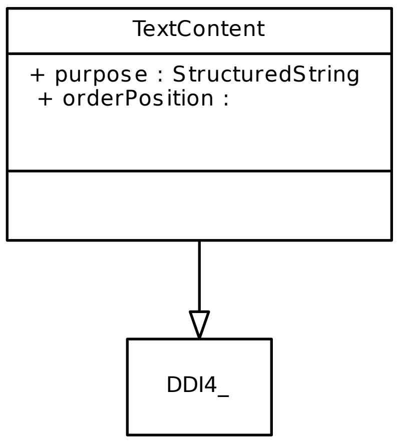
orderPosition

Provides the relative order of TextContent objects in a dynamic text with more than one TextContent object. Uses integer value.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured tex

Graph



6.1.76 TextualSegment

Defines the segment of textual content used by the parent object. Can identify a set of lines and or characters used to define the segment.

‘

Properties

Name	Type	Cardinality
characterParameter	CharacterOffset	0..1
lineParamenter	LineParameter	0..1

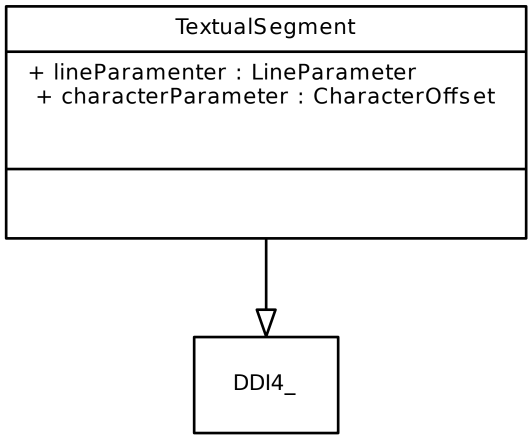
characterParameter

Specification of the character offset for the beginning and end of the segment.

lineParamenter

Specification of the line and offset for the beginning and end of the segment.

Graph


TextualSegment
+ lineParamenter : LineParameter
+ characterParameter : CharacterOffset
DDI4_

6.1.77 TypedDescriptiveText

This Complex Data Type bundles a descriptiveText with an External Controlled Vocabulary Entry allowing structured content and a means of typing that content. For example specifying that the description provides a Table of Contents for a document.

‘

Properties

Name	Type	Cardinality
contentCoverage	ExternalControlledVocabularyEntry	0..1
descriptiveText	StructuredString	0..1

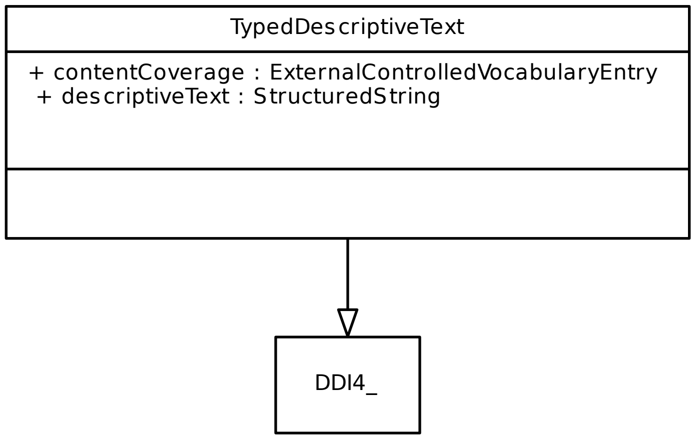
contentCoverage

Uses a controlled vocabulary entry to classify the description provided.

descriptiveText

A short natural language account of the characteristics of the object.

Graph


TypedDescriptiveText
+ contentCoverage : ExternalControlledVocabularyEntry
+ descriptiveText : StructuredString
DDI4_

6.1.78 URI

A URN or URL for a file with a flag to indicate if it is a public copy.

‘

Properties

Name	Type	Cardinality
content	anyURI	0..1
isPublic	Boolean	0..1

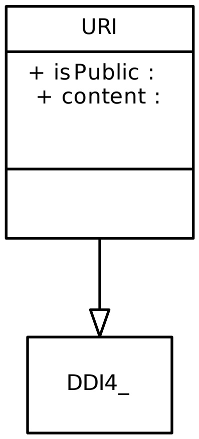
content

The URI in valid format

isPublic

Set to “true” (default value) if this file is publicly available. This does not imply that there are not restrictions to access. Set to “false” if this is not publicly available, such as a backup copy, an internal processing data file, etc.

Graph



6.1.79 URL

A web site URL

‘

Properties

Name	Type	Cardinality
content	anyURI	0..1
effectiveDates	DateRange	0..1
isPreferred	Boolean	0..1
privacy	ExternalControlledVocabularyEntry	0..1
typeOfWebsite	ExternalControlledVocabularyEntry	0..1

content

The content of the URL

effectiveDates

The period for which this URL is valid.

isPreferred

Set to “true” if this is the preferred URL.

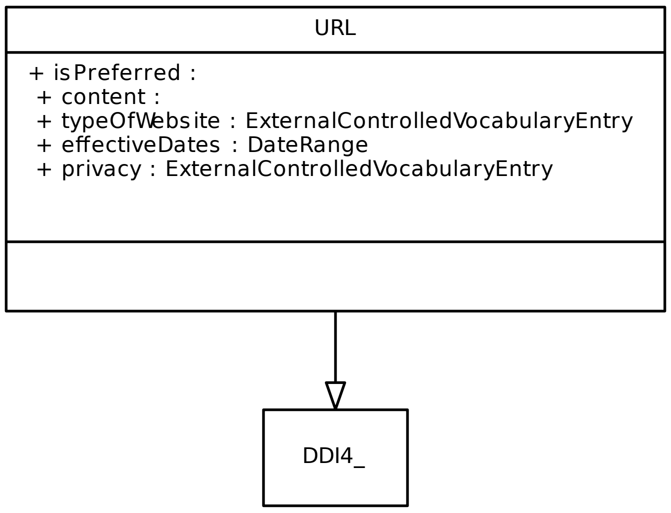
privacy

Indicates the privacy level of this URL

typeOfWebsite

The type of URL for example personal, project, organization, division, etc.

Graph


URL
+ isPreferred :
+ content :
+ typeOfWebsite : ExternalControlledVocabularyEntry
+ effectiveDates : DateRange
+ privacy : ExternalControlledVocabularyEntry
DDI4_

6.1.80 Value

The Value expressed as an xs:string with the ability to preserve whitespace if critical to the understanding of the content.

‘

Properties

Name	Type	Cardinality
content	String	0..1
whiteSpace	WhiteSpace	0..1

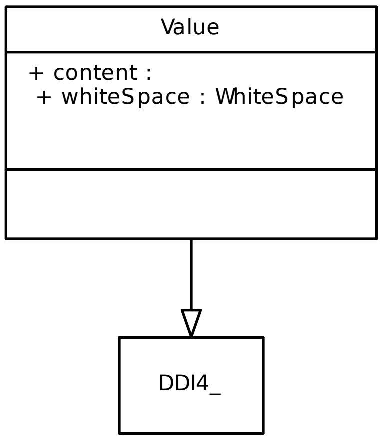
content

The actual content of this value as a string

whiteSpace

The default setting states that leading and trailing white space will be removed and multiple adjacent white spaces will be treated as a single white space. If the existence of any of these white spaces is critical to the understanding of the content, change the value of this attribute to "preserve".

Graph



6.1.81 VideoSegment

Describes the type and length of the video segment.

'

Properties

Name	Type	Cardinality
typeOfVideoClip	ExternalControlledVocabularyEntry	0..1
videoClipBegin	String	0..1
videoClipEnd	String	0..1

typeOfVideoClip

The type of video clip provided. Supports the use of a controlled vocabulary.

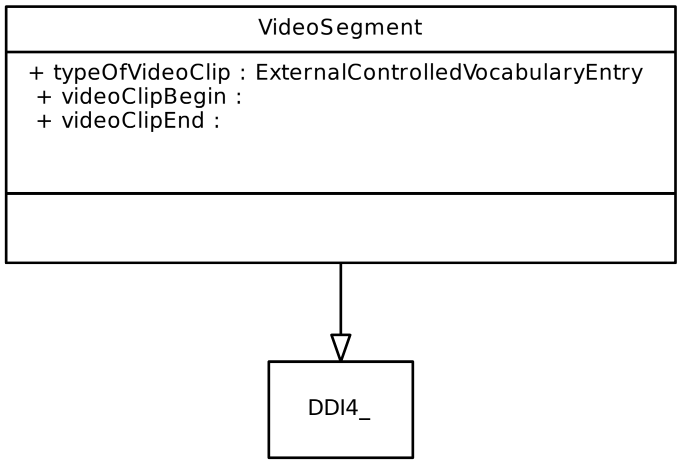
videoClipBegin

The point to begin the video clip. If no point is provided the assumption is that the start point is the beginning of the clip provided.

videoClipEnd

The point to end the video clip. If no point is provided the assumption is that the end point is the end of the clip provided.

Graph



6.1.82 XMLPrefixMap

Maps a specified prefix to a namespace. For each XML namespace used in the profile's XPath expressions, the XML namespaces must have their prefix specified using this element.

‘

Properties

Name	Type	Cardinality
xmlNamespace	String	0..1
xmlPrefix	String	0..1

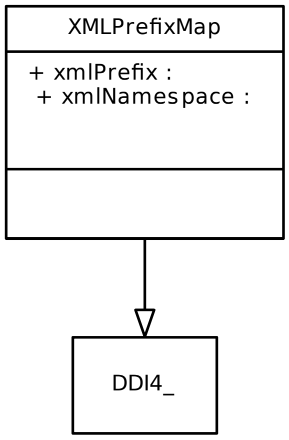
xmlNamespace

Specify the namespace which the prefix represents.

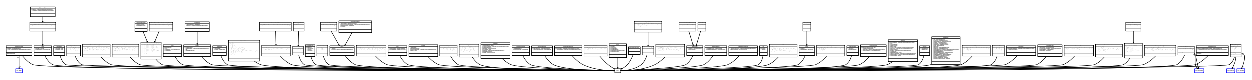
xmlPrefix

Specify the exact prefix used.

Graph



6.1.83 Graph



6.2 Conceptual

A package is a administrative collection of classes in DDI. These are not namespaces.

6.2.1 Category

A Concept whose role is to define and measure a characteristic.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
name	Name	0..n

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

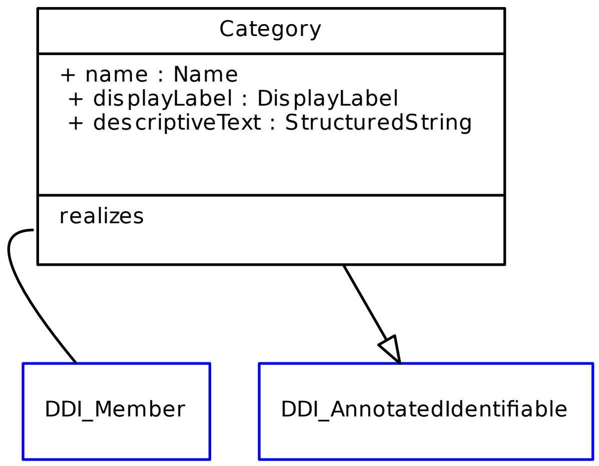
Relationships

Name	Type	Cardinality
realizes	Member	0..n

realizes

Class can play the role of Member within a Collection

Graph


Category
+ name : Name
+ displayLabel : DisplayLabel
+ descriptiveText : StructuredString
realizes
DDI_Member
DDI_AnnotatedIdentifiable

6.2.2 Concept

Unit of thought differentiated by characteristics [GSIM 1.1]

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
definition	StructuredString	0..1
name	Name	0..n

definition

Natural language statement conveying the meaning of a concept, differentiating it from other concepts. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

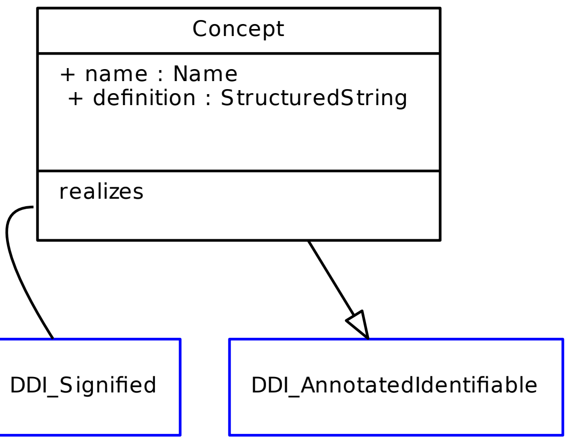
Relationships

Name	Type	Cardinality
realizes	Signified	0..n

realizes

A concept can play the role of a signified when there is a designation that denotes it.

Graph



6.2.3 ConceptParentChild

Parent-child specialization of OrderRelation between Concepts within a ConceptSystem.

Extends

Identifiable

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1

reflexivity

Fixed to Anti_Reflexive

semantics

Controlled vocabulary for the ImmediatePrecedenceRelation semantics. It should contain, at least, the following: Parent_Of, Child_Of, Next, Previous, Instance_Of, Temporal_Meets.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Anti_Transitive

‘

Relationships

Name	Type	Cardinality
contains	ConceptParentChildPair	0..n
realizes	ImmediatePrecedenceRelation	0..n
structures	ConceptSystem	0..n

contains

Ordered sets of Parent Child pairs.

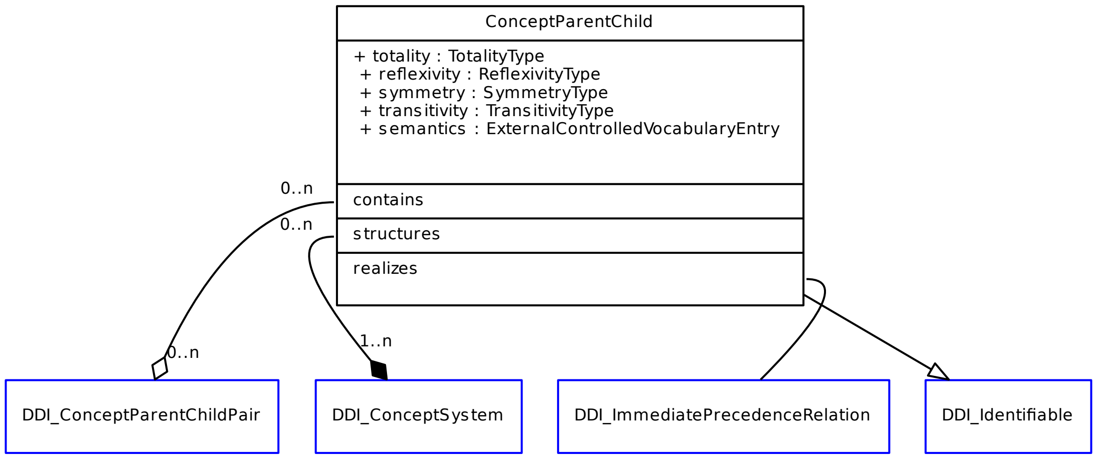
realizes

Uses the pattern ImmediatePrecedenceRelation

structures

Concept System whose members are grouped by pairs to form a variety of structures.

Graph


ConceptParentChild
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
contains
structures
realizes
0..n
0..n
0..n
1..n
DDI_ConceptParentChildPair
DDI_ConceptSystem
DDI_ImmediatePrecedenceRelation
DDI_Identifiable

6.2.4 ConceptParentChildPair

Specifies the Parent Concept and Child Concept in the Concept Parent Child Relationship.

Extends

Identifiable

Relationships

Name	Type	Cardinality
child	Concept	0..n
parent	Concept	0..n
realizes	OrderedPair	0..n

child

Specialization of “target” in OrderedPair

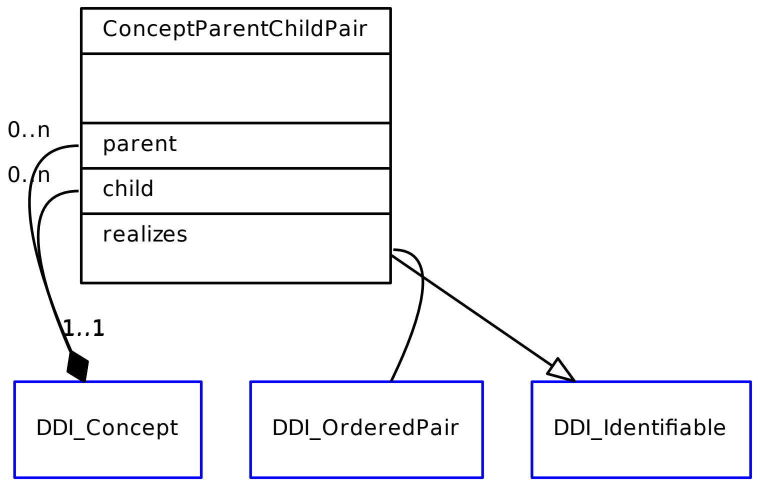
parent

Specialization of “source” in OrderedPair

realizes

realizes the pattern OrderedPair

Graph



6.2.5 ConceptPartWhole

Part-whole specialization of OrderRelation between Concepts within a ConceptSystem.

Extends

Identifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
displayLabel	DisplayLabel	0..n
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

criteria

Intentional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

displayLabel

A display label for the OrderRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

reflexivity

Fixed to Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	ConceptPartWholePair	0..n
realizes	OrderRelation	0..n
structures	ConceptSystem	0..n

contains

Ordered pairs of ConceptPartWholePairs

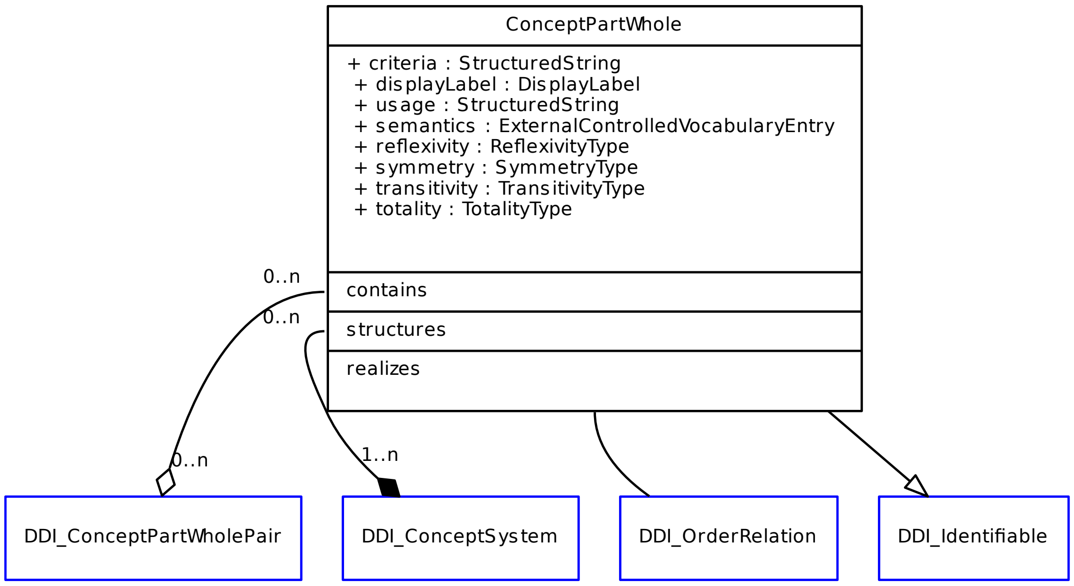
realizes

realizes the pattern of ConceptPartWhole

structures

ConceptSystem whose Members are groups by ConceptPartWholePairs

Graph


ConceptPartWhole
+ criteria : StructuredString
+ displayLabel : DisplayLabel
+ usage : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ totality : TotalityType
contains
structures
realizes
0..n
0..n
0..n
1..n
DDI_ConceptPartWholePair
DDI_ConceptSystem
DDI_OrderRelation
DDI_Identifiable

6.2.6 ConceptPartWholePair

Identifies the concept defining the Whole and the concept defining the Part.

Extends

Identifiable

Relationships

Name	Type	Cardinality
part	Concept	0..n
realizes	OrderedPair	0..n
whole	Concept	0..n

part

Specialization of "target" in OrderedPair

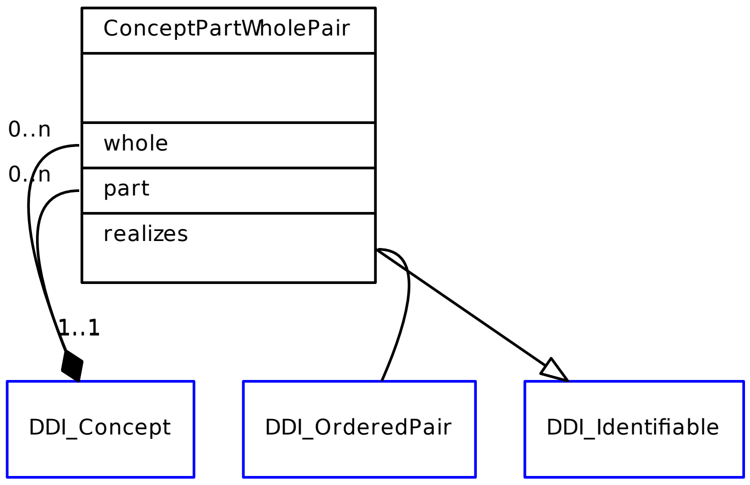
realizes

realizes pattern OrderedPair

whole

Specialization of "source" in OrderedPair.

Graph



6.2.7 ConceptSystem

A set of Concepts structured by the relations among them. [GSIM 1.1]

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
contains	Concept	0..n
hasConceptParentChild	ConceptParentChild	1..n
hasConceptPartWhole	ConceptPartWhole	1..n
realizes	Collection	0..n

contains

The relationship to the concepts contained in the concept system

hasConceptParentChild

Specialization of isOrderedBy in Collection.

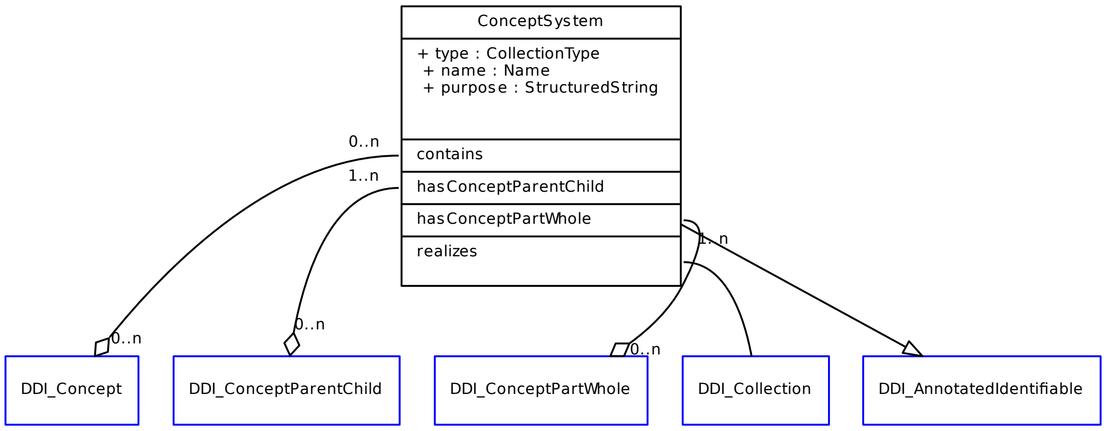
hasConceptPartWhole

Specialization of isOrderedBy in Collection.

realizes

realizes the collection pattern

Graph



6.2.8 ConceptSystemCorrespondence

Relationship between Concept Systems used to group similarity mappings between their Concepts.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n
purpose	StructuredString	0..1
usage	StructuredString	0..1

displayLabel

A display label for the CollectionCorrespondence. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	SimilarConcept	1..n
maps	ConceptSystem	0..n
realizes	SymmetricRelation	0..n

contains

Realization of contains in Symmetric Relation for mapping similar concepts.

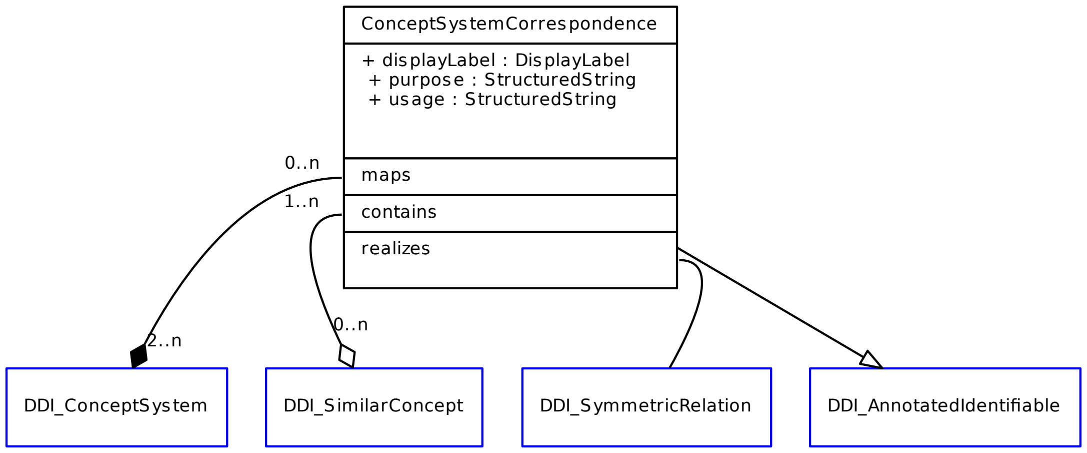
maps

Realization of structures in Symmetric Relation. When Concepts of a single ConceptSystem are mapped, the Concept has to appear twice as target.

realizes

Class in the Collections Pattern realized by this class.

Graph


ConceptSystemCorrespondence
+ displayLabel : DisplayLabel
+ purpose : StructuredString
+ usage : StructuredString
maps
contains
realizes
0..n
1..n
2..n
0..n
DDI_ConceptSystem
DDI_SimilarConcept
DDI_SymmetricRelation
DDI_AnnotatedIdentifiable

6.2.9 ConceptualVariable

The use of a Concept as a characteristic of a Universe intended to be measured [GSIM 1.1]

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
name	Name	0..n

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

Relationships

Name	Type	Cardinality
realizes	Member	0..n
takesSentinelConceptsFrom	SentinelConceptualDomain	0..n
takesSubstantiveConceptsFrom	SubstantiveConceptualDomain	1..n
usesConcept	Concept	0..n
usesUnitType	UnitType	0..n

realizes

Class can play the role of Member within a Collection

takesSentinelConceptsFrom

Identifies the ConceptualDomain containing the set of sentinel concepts used to describe the ConceptualVariable.

takesSubstantiveConceptsFrom

Identifies the ConceptualDomain containing the set of substantive concepts used to describe the ConceptualVariable.

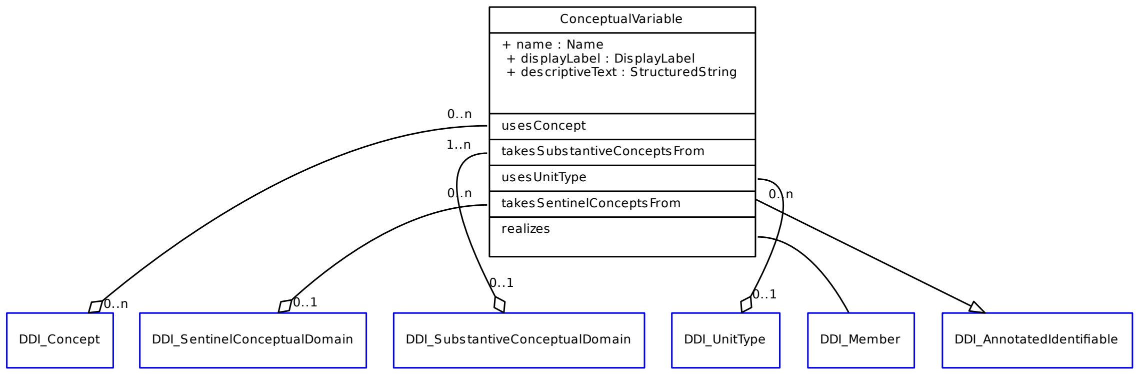
usesConcept

Reference to a Concept that is being used

usesUnitType

Identifies the UnitType associated with the ConceptualVariable

Graph


ConceptualVariable
+ name : Name
+ displayLabel : DisplayLabel
+ descriptiveText : StructuredString
usesConcept
takesSubstantiveConceptsFrom
usesUnitType
takesSentinelConceptsFrom
realizes
0..n
1..n
0..n
0..n
0..n
0..1
0..1
0..1
DDI_Concept
DDI_SentinelConceptualDomain
DDI_SubstantiveConceptualDomain
DDI_UnitType
DDI_Member
DDI_AnnotatedIdentifiable

6.2.10 InstanceVariable

The use of a Represented Variable within a Data Set.

Extends

RepresentedVariable

Properties

Name	Type	Cardinality
physicalDataType	ExternalControlledVocabularyEntry	0..1
variableRole	StructuredString	0..1

physicalDataType

The data type of this variable. Supports the optional use of an external controlled vocabulary.

variableRole

An Instance Variable can take different roles, e.g. Identifier, Measure and Attribute. Note that DataStructure takes care of the ordering of Identifiers.

‘

Relationships

Name	Type	Cardinality
basedOnConceptualVariable	ConceptualVariable	0..n
basedOnRepresentedVariable	RepresentedVariable	0..n
measures	Population	0..n
takesSentinelValuesFrom	SentinelValueDomain	0..n

basedOnConceptualVariable

The ConceptualVariable that can be shared by multiple InstanceVariables. Indicates comparability ConceptualDomain and UnitType. If a relationship to a RepresentedVariable is defined there should be no relationship to a Conceptual-Variable.

basedOnRepresentedVariable

The RepresentedVariable that can be shared by multiple InstanceVariables. Indicates comparability in substantive representation, Universe, and indirectly ConceptualDomain. If a relationship to a RepresentedVariable is defined there should be no relationship to a ConceptualVariable.

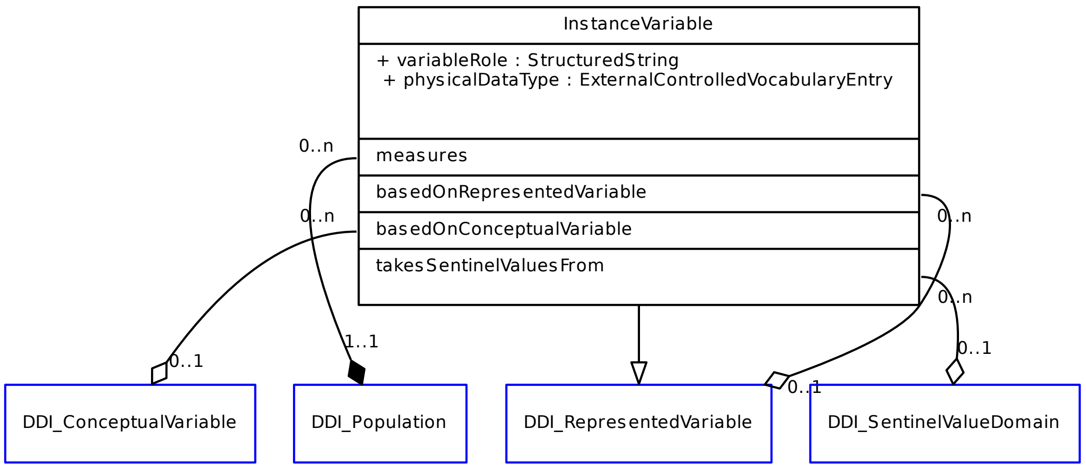
measures

Set of specific units (people, entities, objects, events), usually in a given time and geography, being measured. Can be a specialization of the Universe measured by a related RepresentedVariable.

takesSentinelValuesFrom

The association to the possible sentinel (missing) values.

Graph



6.2.11 Population

Set of specific units (people, entities, objects, events), usually in a given time and geography.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
name	Name	0..n

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

Relationships

Name	Type	Cardinality
contains	Unit	0..n
hasSubpopulation	Population	1..1
narrowsUniverse	Universe	0..n
realizes	Member	0..n
usesConcept	Concept	0..n

contains

Units in the Population

hasSubpopulation

Populations can have sub populations. For example the Sub-Universe Class of Gender for the Universe Population of Canada in 2011 may contain the Universe Canadian Males in 2011 and the Universe Canadian Females in 2011.

narrowsUniverse

Reference to a Universe that the Population narrows

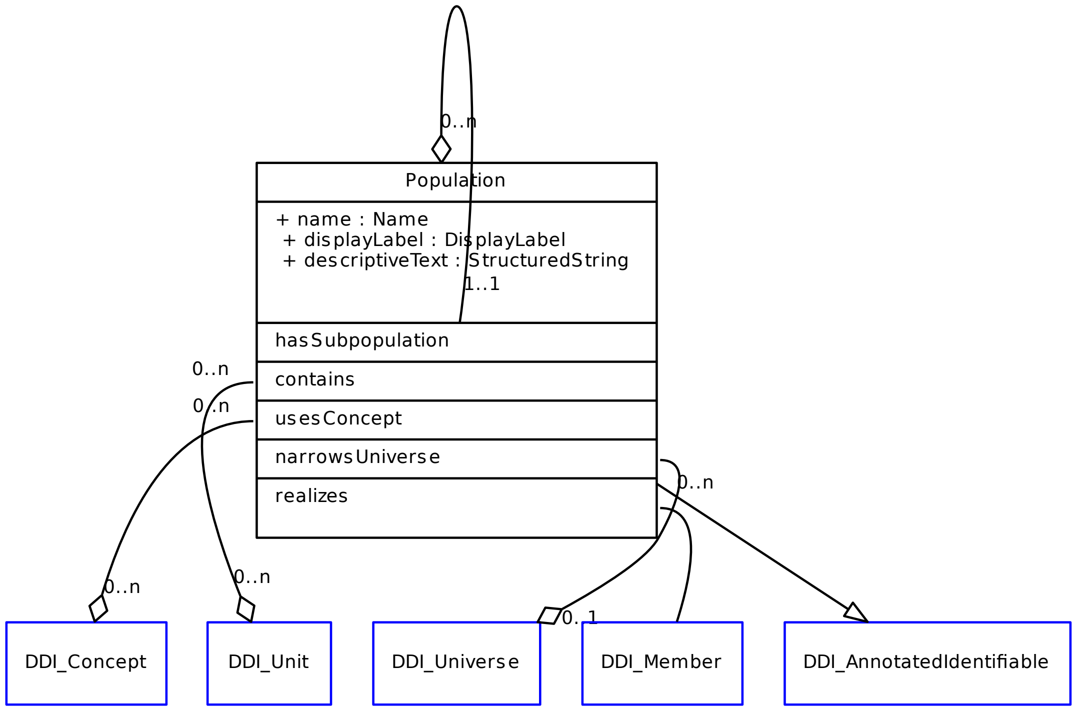
realizes

Class can be used in the role of Member within a Collection

usesConcept

Reference to the Concept that is being used

Graph



6.2.12 RepresentedVariable

A combination of a characteristic of a universe to be measured and how that measure will be represented.

Extends

ConceptualVariable

Properties

Name	Type	Cardinality
hasIntendedDataType	ExternalControlledVocabularyEntry	0..1
unitOfMeasurement	String	0..1

hasIntendedDataType

The data type intended to be used by this variable. Supports the optional use of an external controlled vocabulary.

unitOfMeasurement

The unit in which the data values are measured (kg, pound, euro).

‘

Relationships

Name	Type	Cardinality
basedOnConceptualVariable	ConceptualVariable	0..n
measures	Universe	0..n
takesSubstantiveValuesFrom	SubstantiveValueDomain	0..n

basedOnConceptualVariable

The ConceptualVariable that can be shared by a set of multiple RepresentedVariables. Indicates comparability in ConceptualDomain and UnitType.

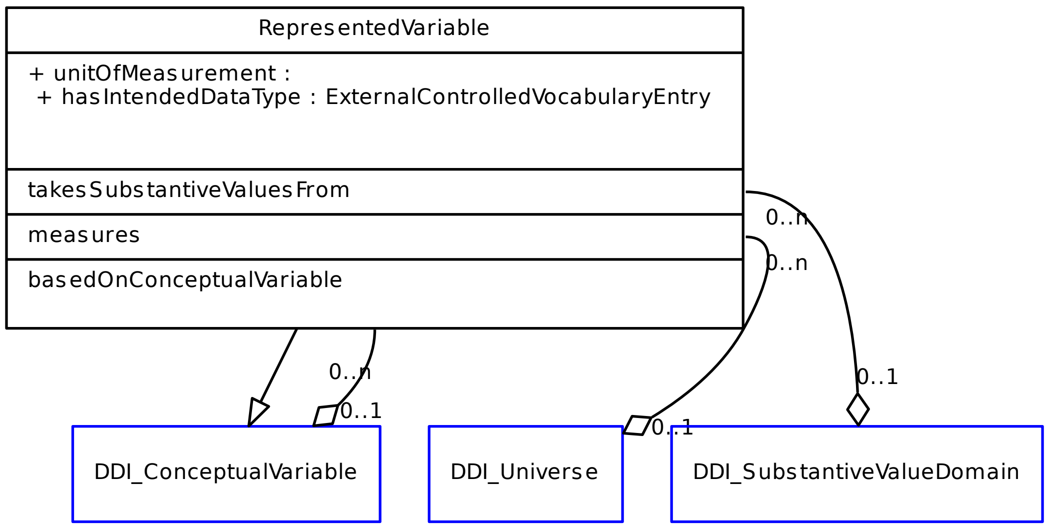
measures

The defined class of people, entities, events, or objects to be measured.

takesSubstantiveValuesFrom

The substantive representation (SubstantiveValueDomain) of the variable. This is equivalent to the relationship “Measures” in GSIM although GSIM makes no distinction between substantive and sentinel values

Graph


RepresentedVariable
+ unitOfMeasurement :
+ hasIntendedDataType : ExternalControlledVocabularyEntry
takesSubstantiveValuesFrom
measures
basedOnConceptualVariable
0..n
0..1
0..n
0..1
0..n
0..1
DDI_ConceptualVariable
DDI_Universe
DDI_SubstantiveValueDomain

6.2.13 SentinelConceptualDomain

Description or list of possible sentinel concepts , e.g. missing values.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
describedConceptualDomain	ValueAndConceptDescription	0..n
enumeratedConceptualDomain	CategorySet	0..n

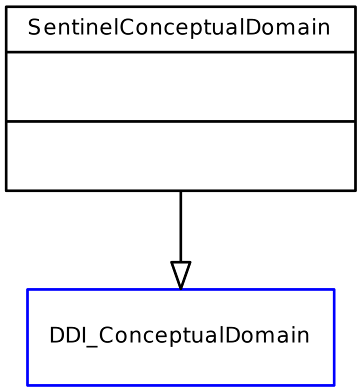
describedConceptualDomain

A description of the concepts in the domain. A numeric domain might use a logical expression to be machine actionable a text domain might use a regular expression to describe strings that describe the concepts.

enumeratedConceptualDomain

The CategorySet containing the concepts in the domain.

Graph



6.2.14 SimilarConcept

A reference to a concept with similar meaning and a description of their differences. The similar concept structure allows specification of similar concepts to address cases where confusion may affect the appropriate use of the concept.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n
type	CorrespondenceType	0..1
usage	StructuredString	0..1

displayLabel

A display label for the MemberCorrespondence. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

type

Type of correspondence in terms of commonalities and differences between two members.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
maps	Concept	0..n
realizes	UnorderedTuple	0..n

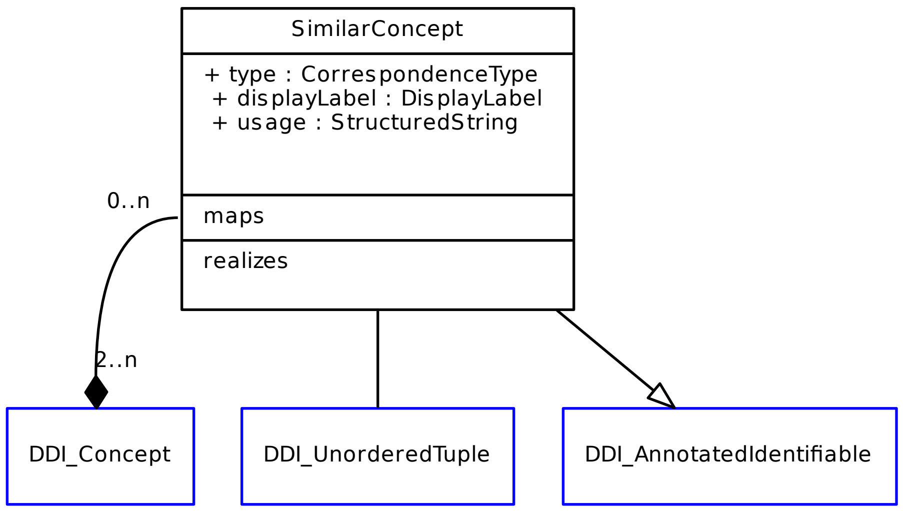
maps

Realization of maps in Unordered Tuple for Concepts that are similar. Used to assist in disambiguation of concepts.

realizes

Class in the Collections Pattern realized by this class.

Graph


SimilarConcept
+ type : CorrespondenceType
+ displayLabel : DisplayLabel
+ usage : StructuredString
maps
realizes
0..n
2..n
DDI_Concept
DDI_UnorderedTuple
DDI_AnnotatedIdentifiable

6.2.15 SubstantiveConceptualDomain

Set of valid Concepts. The Concepts can be described by either enumeration or by an expression.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n

displayLabel

A display label for the Conceptual Domain. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

‘

Relationships

Name	Type	Cardinality
describedConceptualDomain	ValueAndConceptDescription	0..n
enumeratedConceptualDomain	CategorySet	0..n

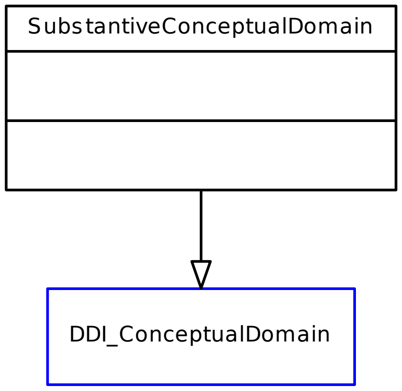
describedConceptualDomain

A description of the concepts in the domain. A numeric domain might use a logical expression to be machine actionable a text domain might use a regular expression to describe strings that describe the concepts.

enumeratedConceptualDomain

The CategorySet containing the concepts in the domain.

Graph



6.2.16 Unit

The object of interest in a process step related to the collection or use of observational data.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n
name	Name	0..n

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

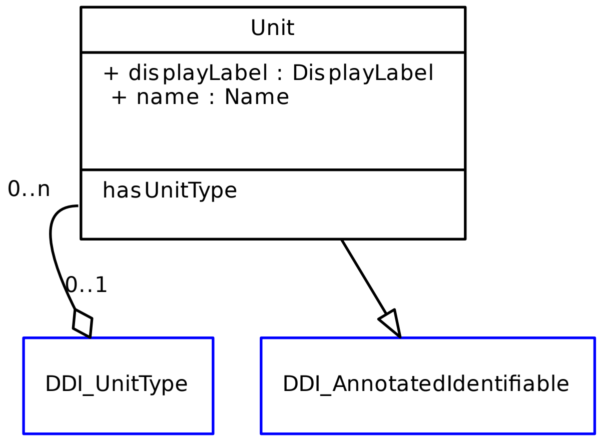
Relationships

Name	Type	Cardinality
hasUnitType	UnitType	0..n

hasUnitType

The UnitType of the Unit

Graph



6.2.17 UnitType

A Unit Type is a class of objects of interest.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
name	Name	0..1

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

Relationships

Name	Type	Cardinality
realizes	Member	0..n
usesConcept	Concept	0..n

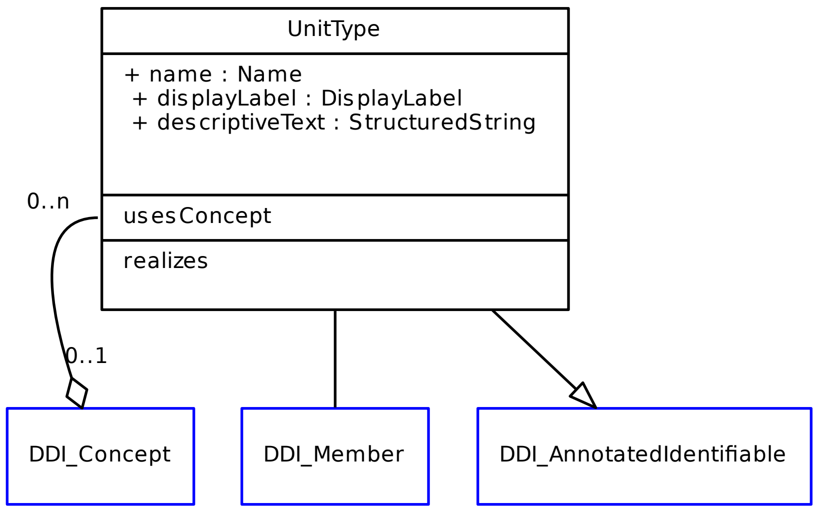
realizes

Class can be used in the role of Member within a Collection

usesConcept

Reference to the Concept that is being used

Graph


UnitType
+ name : Name
+ displayLabel : DisplayLabel
+ descriptiveText : StructuredString
usesConcept
realizes
0..n
0..1
DDI_Concept
DDI_Member
DDI_AnnotatedIdentifiable

6.2.18 Universe

A defined class of people, entities, events, or objects, with no specification of time and geography, contextualizing a Unit Type

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
isInclusive	Boolean	0..1
name	Name	0..n

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

isInclusive

The default value is “true”. The description statement of a universe is generally stated in inclusive terms such as “All persons with university degree”. Occasionally a universe is defined by what it excludes, i.e., “All persons except those with university degree”. In this case the value would be changed to “false”.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

Relationships

Name	Type	Cardinality
hasSubuniverse	Universe	1..1
narrowsUnitType	UnitType	0..n
realizes	Member	0..n
usesConcept	Concept	0..n

hasSubuniverse

Universes can have sub universes. A sub-universe class provides a definition to the universes contained within it. For example the Sub-Universe Class of Gender for the Universe Population may contain the Universe Males and the Universe Females

narrowsUnitType

Reference to the Unit Type that the Universe definition narrows.

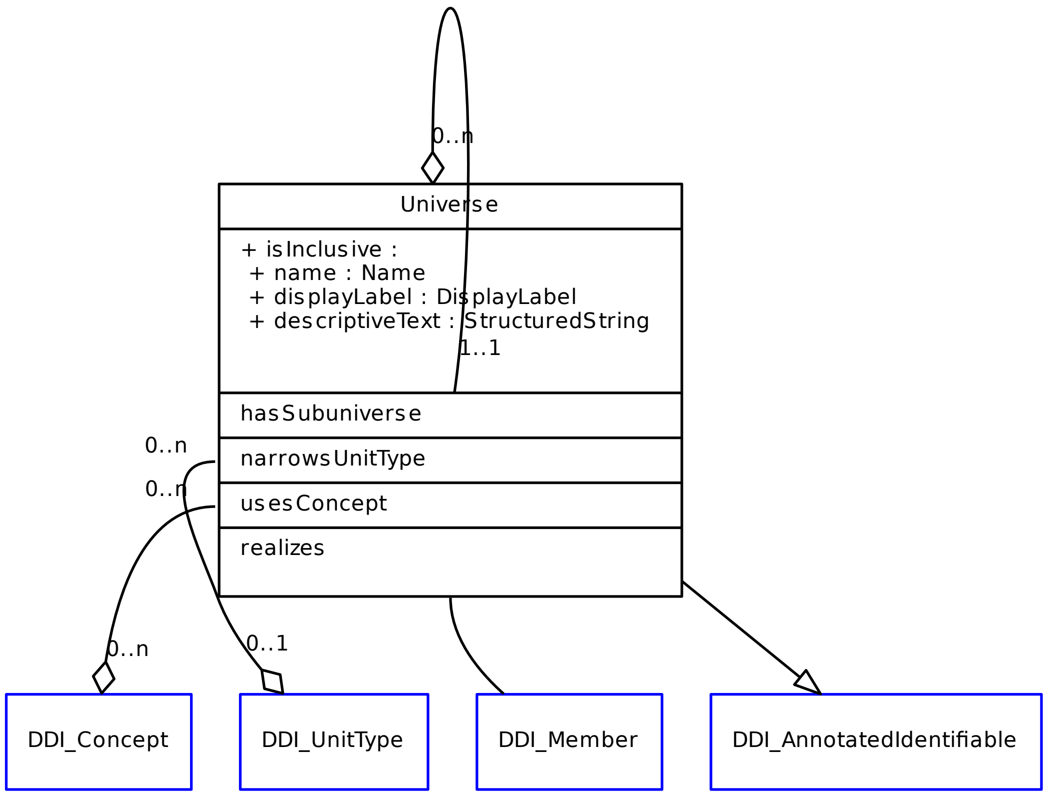
realizes

Class can be used in the role of Member within a Collection

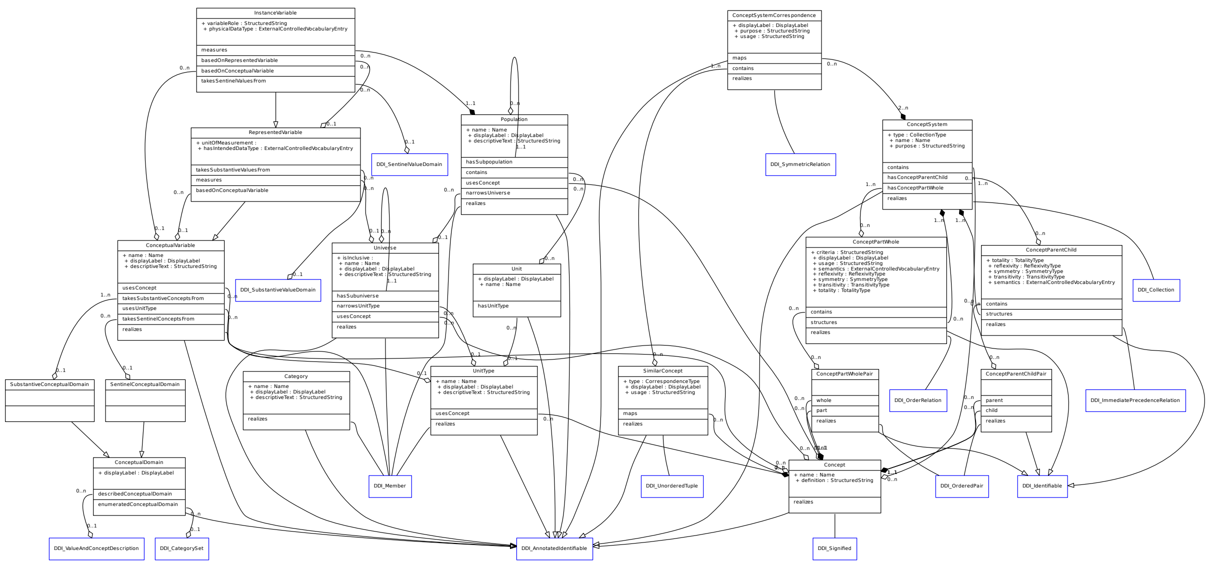
usesConcept

Reference to the Concept that is being used

Graph


0..n
Universe
+ isInclusive :
+ name : Name
+ displayLabel : DisplayLabel
+ descriptiveText : StructuredString
1..1
hasSubuniverse
narrowsUnitType
usesConcept
realizes
0..n
0..n
0..n
0..1
DDI_Concept
DDI_UnitType
DDI_Member
DDI_AnnotatedIdentifiable

6.2.19 Graph



6.3 CustomMetadata

A package is a administrative collection of classes in DDI. These are not namespaces.

6.3.1 ControlledVocabulary

The specification of a controlled vocabulary defines a set of values and their definitions together with the order relationships among those entries.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
contains	VocabularyEntry	1..n
hasVocabularyParentChild	VocabularyParentChild	1..n
hasVocabularySequence	VocabularySequence	1..n
realizes	Collection	0..n

contains

The entries in the vocabulary.

hasVocabularyParentChild

The OrderRelation defining the hierarchy in the vocabulary

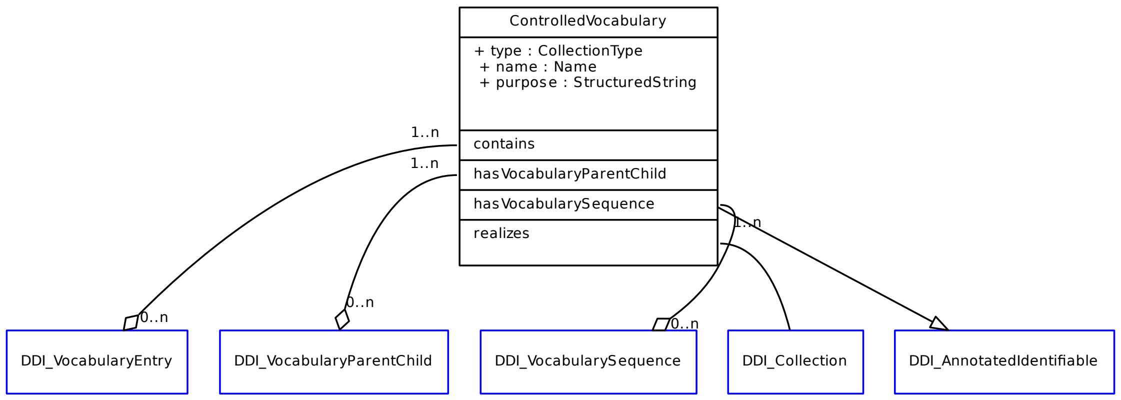
hasVocabularySequence

The OrderRelation defining the sequential order in the vocabulary

realizes

realizes the collection pattern

Graph



6.3.2 CustomInstance

A set of CustomValues to be attached to some object.

Extends

Identifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

see Collection

purpose

see Collection

type

see Collection

‘

Relationships

Name	Type	Cardinality
contains	CustomValue	1..1
correspondsTo	CustomStructure	0..n
hasParentChildOrder	CustomValueParentChild	1..n
hasRelationshipOrder	CustomValueRelationship	1..n
hasSequenceOrder	CustomValueSequence	1..n
realizes	Collection	0..n

contains

the set of CustomItems grouped.

correspondsTo

The Set of CustomItems allowed.

hasParentChildOrder

Associates a specification of a parent-child ordering among the CustomValues in this structure

hasRelationshipOrder

Associates a specification of a source target predicate ordering among the CustomValues in this structure

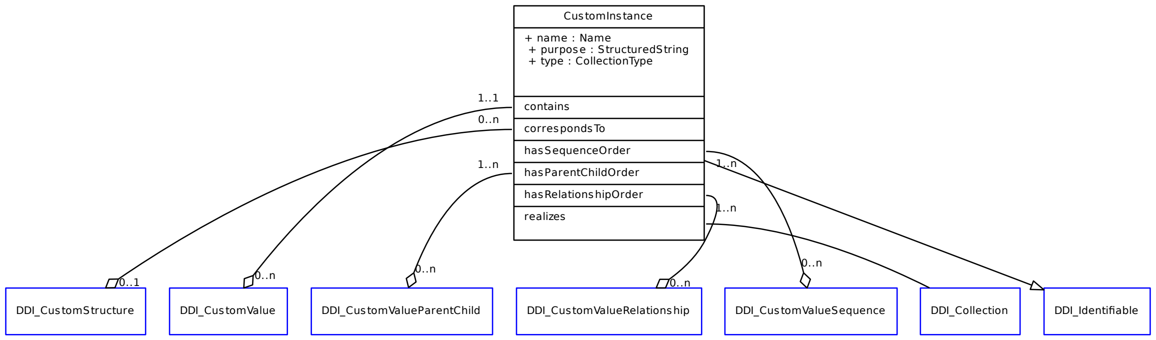
hasSequenceOrder

Associates a specification of a sequential ordering among the CustomValues in this structure

realizes

This is a collection of key-value pairs

Graph



6.3.3 CustomItem

A custom item description. This allows the definition of an item which is a member of a CustomStructure. It defines the minimum and maximum number of occurrences and representation of a CustomValue.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
key	String	0..1
maxOccurs	Integer	0..1
minOccurs	Integer	0..1

key

The key to be used by a key,value pair

maxOccurs

The maximum number of occurrences of an associated CustomValue

minOccurs

The minimum number of occurrences of an associated CustomValue

‘

Relationships

Name	Type	Cardinality
hasConcept	Concept	0..n
realizes	Member	0..n
relatesTo	RepresentedVariable	0..n
uses	ValueDomain	0..n

hasConcept

A Concept associated with the key of a CustomValue

realizes

Class can play the role of a Member in a Collection

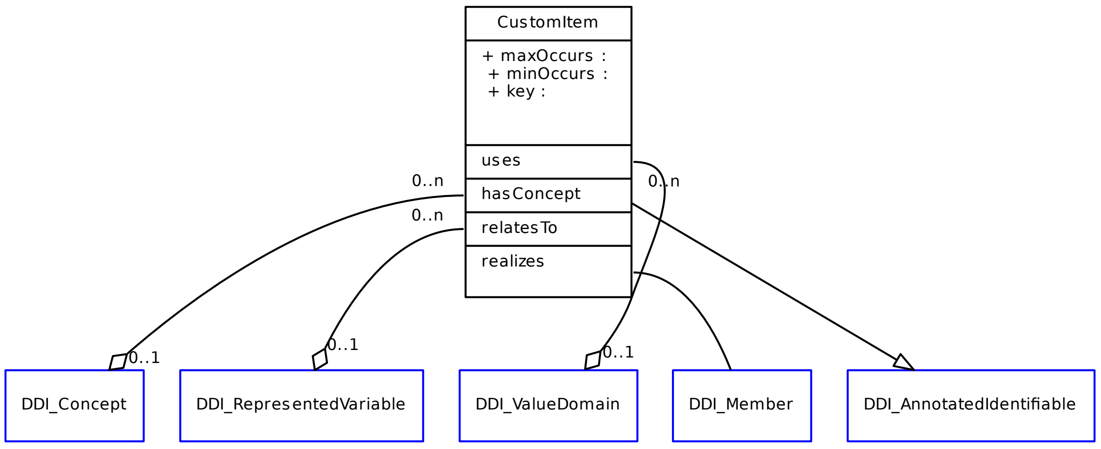
relatesTo

This optional relationship indicates a RepresentedVariable comparable to this key. Custom metadata might be reused as data and data might be used as metadata. This mechanism could enable these transformations.

uses

The type of value for the key.value pair

Graph



6.3.4 CustomItemParentChild

Contains the set of CustomItemParentChildPairs that define the parent child relationships in a CustomStructure.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
displayLabel	DisplayLabel	0..n
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

criteria

Intentional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

displayLabel

A display label for the OrderRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

reflexivity

Fixed to Anti-Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Anti-Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	CustomItemParentChildPair	0..n
realizes	ImmediatePrecedenceRelation	0..n
structures	CustomStructure	0..n

contains

Each CustomItemParentChipdPair defines the ordering between two CustomItems.

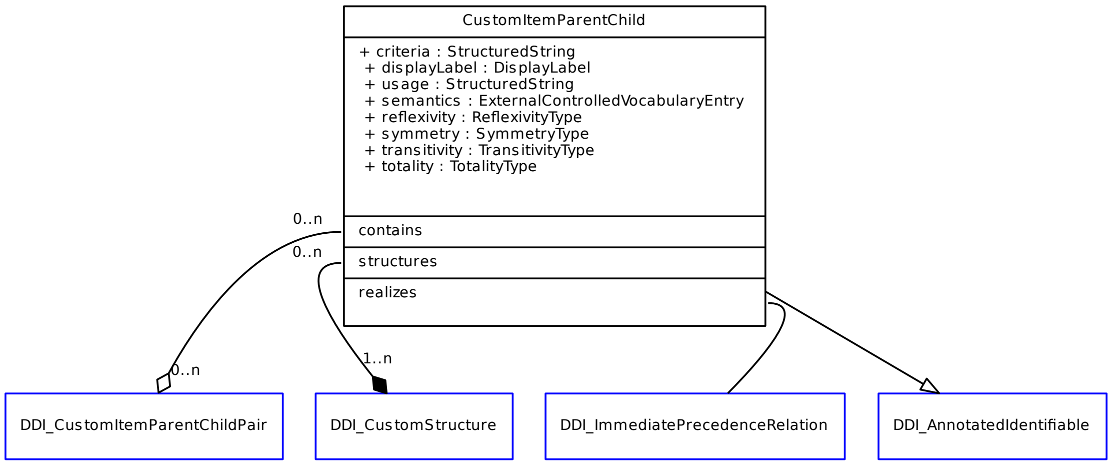
realizes

realizes the pattern ImmediatePrecedenceRelation. (An object is not a parent to a grandchild)

structures

Collection whose Members are grouped by pairs to support a variety of structures.

Graph


CustomItemParentChild
+ criteria : StructuredString
+ displayLabel : DisplayLabel
+ usage : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ totality : TotalityType
contains
structures
realizes
0..n
0..n
0..n
1..n
DDI_CustomItemParentChildPair
DDI_CustomStructure
DDI_ImmediatePrecedenceRelation
DDI_AnnotatedIdentifiable

6.3.5 CustomItemParentChildPair

A parent-child relationship between a pair of CustomItems.

Extends

Identifiable

Relationships

Name	Type	Cardinality
child	CustomItem	0..n
parent	CustomItem	0..n
realizes	OrderedPair	0..n

child

The child in the pair. This item descends from the parent in some sense. Specialization of "target" in OrderedPair

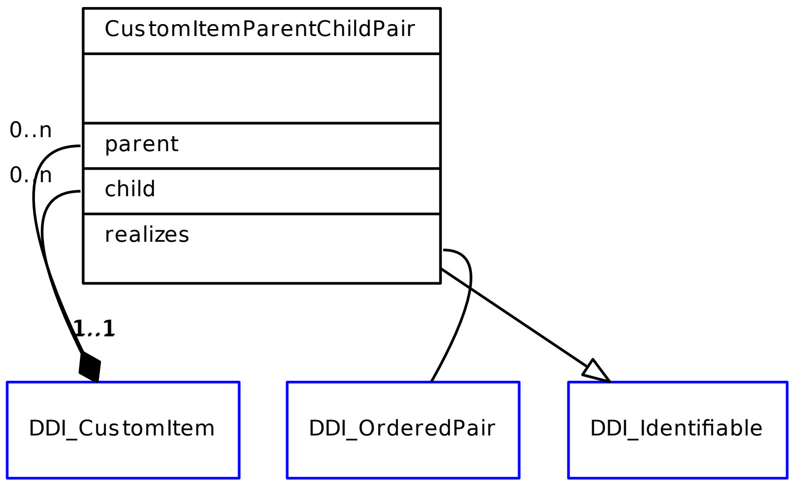
parent

The parent in the pair. Specialization of "source" in OrderedPair

realizes

realizes pattern OrderedPair

Graph



6.3.6 CustomItemRelationship

Contains a set of CustomItemRelationshipPairs which together define the relationships of the type described by a predicate in each pair. A set of triples.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
displayLabel	DisplayLabel	0..n
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

criteria

Intentional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

displayLabel

A display label for the OrderRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

reflexivity

Fixed to Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	CustomItemRelationshipPair	0..n
realizes	OrderRelation	0..n
structures	CustomStructure	0..n

contains

Each pair describes the relationship specified by a predicate between a source and a target.

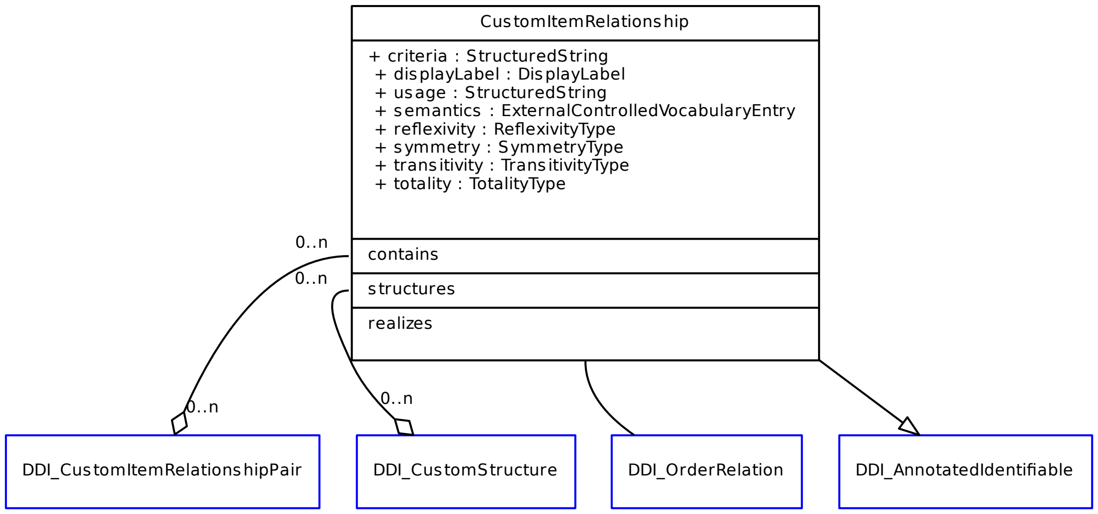
realizes

realizes the pattern OrderRelation

structures

Collection whose Members are grouped by pairs to support a variety of structures.

Graph


CustomItemRelationship
+ criteria : StructuredString
+ displayLabel : DisplayLabel
+ usage : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ totality : TotalityType
contains
structures
realizes
0..n
0..n
0..n
0..n
DDI_CustomItemRelationshipPair
DDI_CustomStructure
DDI_OrderRelation
DDI_AnnotatedIdentifiable

6.3.7 CustomItemRelationshipPair

Defines an 'order' relationship with a predicate between two CustomItems (a triple).

Extends

Identifiable

Properties

Name	Type	Cardinality
predicate	String	0..1

predicate

The type of relationship between the two CustomItems. The verb in the sentence describing the relationship between the source (subject) and the target (object).

‘

Relationships

Name	Type	Cardinality
realizes	OrderedPair	0..n
source	CustomItem	0..n
target	CustomItem	0..n

realizes

realizes pattern OrderedPair

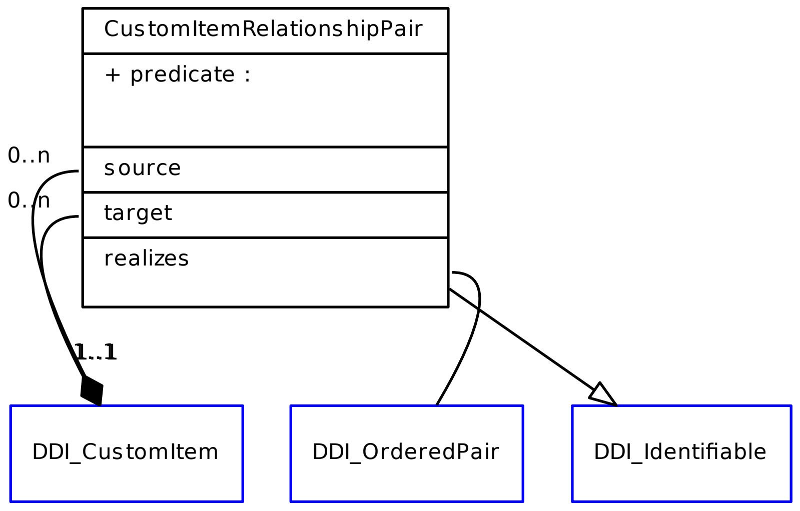
source

The source (subject) of the relationship

target

The target (object) of the relationship defined by the predicate (verb).

Graph



6.3.8 CustomItemSequence

Contains a set of CustomItemSequencePairs which taken together define the sequence relationships among a set of CustomItems.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
displayLabel	DisplayLabel	0..n
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

criteria

Intentional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

displayLabel

A display label for the OrderRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

reflexivity

Fixed to Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	CustomItemSequencePair	1..n
realizes	OrderRelation	0..n
structures	CustomStructure	0..n

contains

Contains a set of CustomItemSequencePairs which together define a sequence among a set of CustomItems.

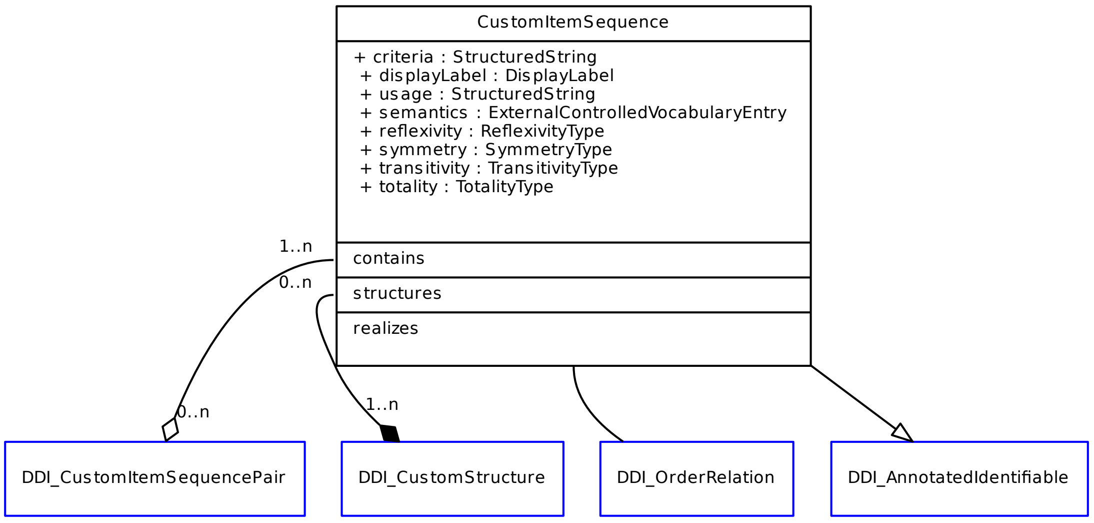
realizes

realizes pattern OrderRelation

structures

Collection whose Members are grouped by pairs to support a variety of structures.

Graph


CustomItemSequence
+ criteria : StructuredString
+ displayLabel : DisplayLabel
+ usage : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ totality : TotalityType
contains
structures
realizes
1..n
0..n
0..n
1..n
DDI_CustomItemSequencePair
DDI_CustomStructure
DDI_OrderRelation
DDI_AnnotatedIdentifiable

6.3.9 CustomItemSequencePair

Defines a simple ordering between two CustomItems - one precedes the other.

Extends

Identifiable

Relationships

Name	Type	Cardinality
predecessor	CustomItem	0..n
realizes	OrderedPair	0..n
successor	CustomItem	0..n

predecessor

The CustomItem coming first in the order. Specializes source in OrderedPair

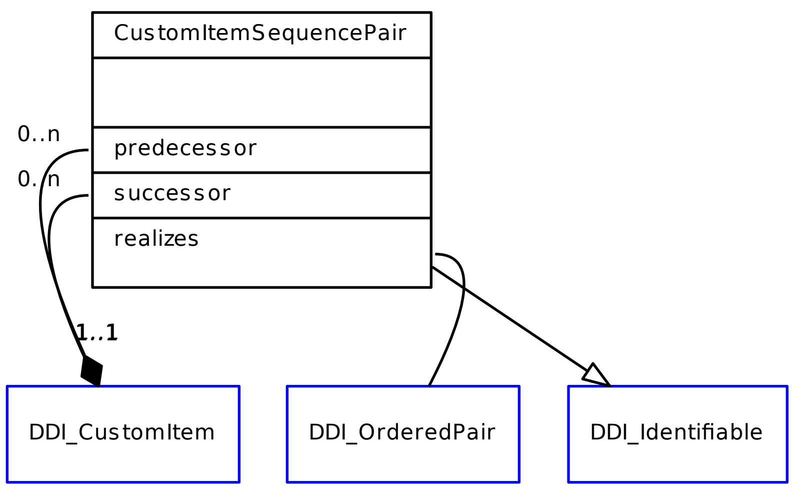
realizes

realizes pattern OrderedPair

successor

The CustomItem coming second in the order. Specializes target in OrderedPair

Graph



6.3.10 CustomStructure

A Collection containing a set of item descriptions defining a structure for a set of key,value pairs

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
contains	CustomItem	0..n
hasParentChildOrder	CustomItemParentChild	1..n
hasRelationshipOrder	CustomItemRelationship	1..n
hasSequenceOrder	CustomItemSequence	1..n
realizes	Collection	0..n

contains

A CustomStructure contains a set of CustomItems. This defines a particular set of key,value pairs that are usable in a particular application.

hasParentChildOrder

Associates a specification of a parent-child ordering among the CustomItems in this structure

hasRelationshipOrder

Associates a specification of a source target predicate ordering among the CustomItems in this structure

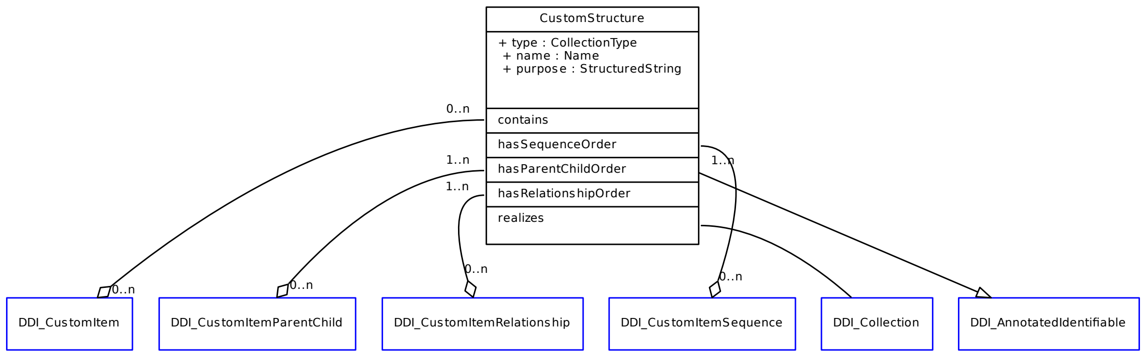
hasSequenceOrder

Associates a specification of a sequential ordering among the CustomItems in this structure

realizes

realizes a collection pattern

Graph



6.3.11 CustomValue

An instance of a key, value pair for a particular key.

Extends

Identifiable

Properties

Name	Type	Cardinality
key	String	0..1
value	Value	0..1

key

The unique String identifying the value

value

the value associated with a particular key expressed as a string. The ultimate value representation is defined by the corresponding CustomValue.

‘

Relationships

Name	Type	Cardinality
correspondsTo	CustomItem	0..n
relatesTo	InstanceVariable	0..n

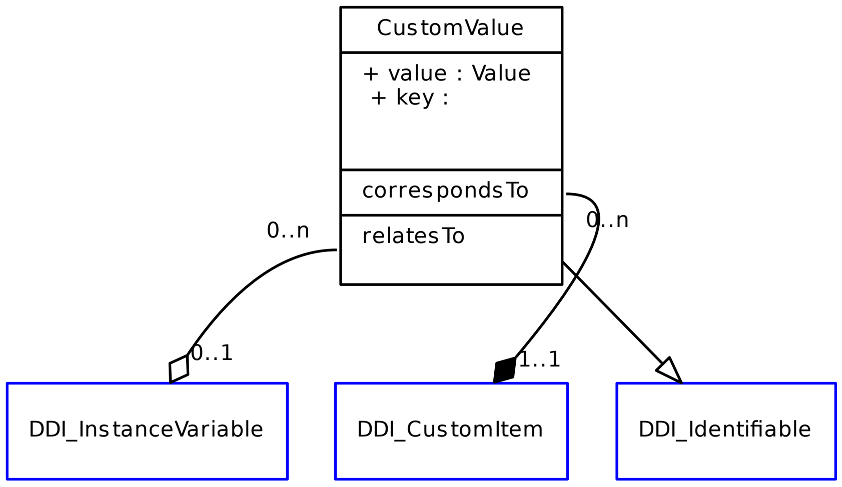
correspondsTo

The CustomItem which defines the key for this key,value pair, It also defines a ValueRepresentation

relatesTo

This optional relationship indicates an InstanceVariable comparable to this key. Custom metadata might be reused as data and data might be used as metadata. This mechanism could enable these transformations. The related Instance-Variable could also carry information about units of measurement, missing values etc.

Graph



6.3.12 CustomValueParentChild

Contains the set of CustomValueParentChildPairs that define the parent child relationships in a CustomInstance.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
displayLabel	DisplayLabel	0..n
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

criteria

Intentional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

displayLabel

A display label for the OrderRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

reflexivity

Fixed to Anti-Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Anti-Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	CustomValueParentChildPair	0..n
realizes	ImmediatePrecedenceRelation	0..n
structures	CustomInstance	0..n

contains

Each CustomValueParentChipdPair defines the ordering between two CustomValues.

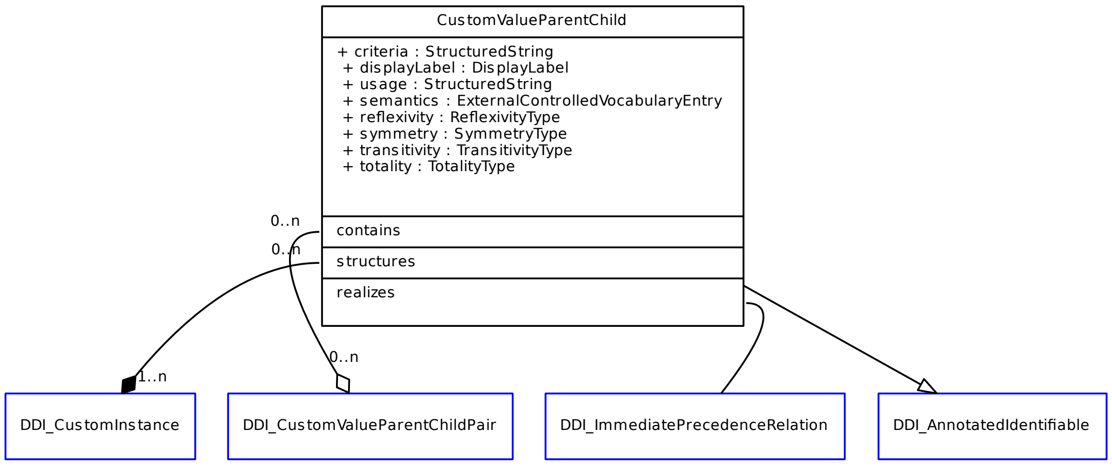
realizes

realizes the pattern ImmediatePrecedenceRelation. (an object is not a parent to a grandchild)

structures

Collection whose Members are grouped by pairs to support a variety of structures.

Graph


CustomValueParentChild
+ criteria : StructuredString
+ displayLabel : DisplayLabel
+ usage : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ totality : TotalityType
contains
structures
realizes
0..n
0..n
1..n
0..n
DDI_CustomInstance
DDI_CustomValueParentChildPair
DDI_ImmediatePrecedenceRelation
DDI_AnnotatedIdentifiable

6.3.13 CustomValueParentChildPair

A parent-child relationship between a pair of CustomValues.

Extends

Identifiable

Relationships

Name	Type	Cardinality
child	CustomValue	0..n
parent	CustomValue	0..n
realizes	OrderedPair	0..n

child

The child in the pair. This item descends from the parent in some sense. Specializes 'target' in OrderedPair

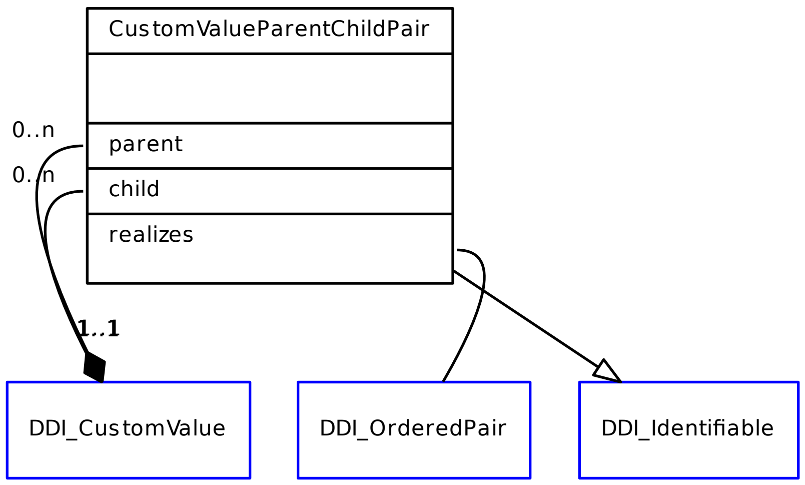
parent

The Parent in the pair. Specializes 'source' in OrderedPair

realizes

realizes pattern OrderedPair

Graph



6.3.14 CustomValueRelationship

Contains a set of CustomValueRelationshipPairs which together define the relationships of the type described by a predicate in each pair. A set of triples.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
displayLabel	DisplayLabel	0..n
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

criteria

Intentional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

displayLabel

A display label for the OrderRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

reflexivity

Fixed to Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	CustomValueRelationshipPair	0..n
realizes	OrderRelation	0..n
structures	CustomInstance	0..n

contains

Each pair describes the relationship specified by a predicate between a source and a target.

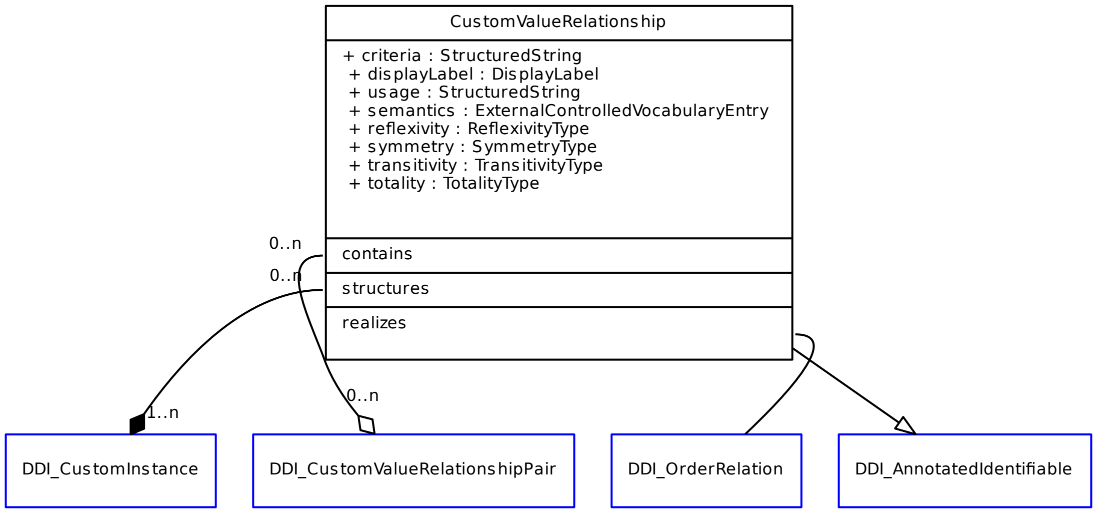
realizes

realizes pattern OrderRelation

structures

Collection whose Members are grouped by pairs to support a variety of structures.

Graph


CustomValueRelationship
+ criteria : StructuredString
+ displayLabel : DisplayLabel
+ usage : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ totality : TotalityType
contains
structures
realizes
0..n
0..n
1..n
0..n
DDI_CustomInstance
DDI_CustomValueRelationshipPair
DDI_OrderRelation
DDI_AnnotatedIdentifiable

6.3.15 CustomValueRelationshipPair

Defines an ‘order’ relationship with a predicate between two CustomValues (a triple)

Extends

Identifiable

Properties

Name	Type	Cardinality
predicate	String	0..1

predicate

The type of relationship between the two CustomValues. The verb in the sentence describing the relationship between the source (subject) and the target (object).

‘

Relationships

Name	Type	Cardinality
realizes	OrderedPair	0..n
source	CustomValue	0..n
target	CustomValue	0..n

realizes

realizes pattern OrderedPair

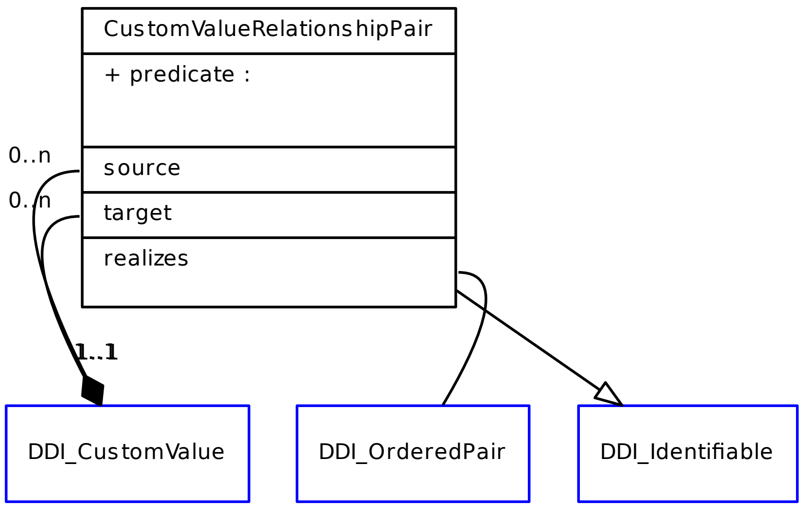
source

The source (subject) of the relationship

target

The target (object) of the relationship defined by the predicate (verb)

Graph



6.3.16 CustomValueSequence

Contains a set of CustomValueSequencePairs which taken together define the sequence relationships among a set of CustomValues.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
displayLabel	DisplayLabel	0..n
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

criteria

Intentional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

displayLabel

A display label for the OrderRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

reflexivity

Fixed to Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	CustomValueSequencePair	0..n
realizes	OrderRelation	0..n
structures	CustomInstance	0..n

contains

Contains a set of CustomValueSequencePairs which together define a sequence among a set of CustomValues.

realizes

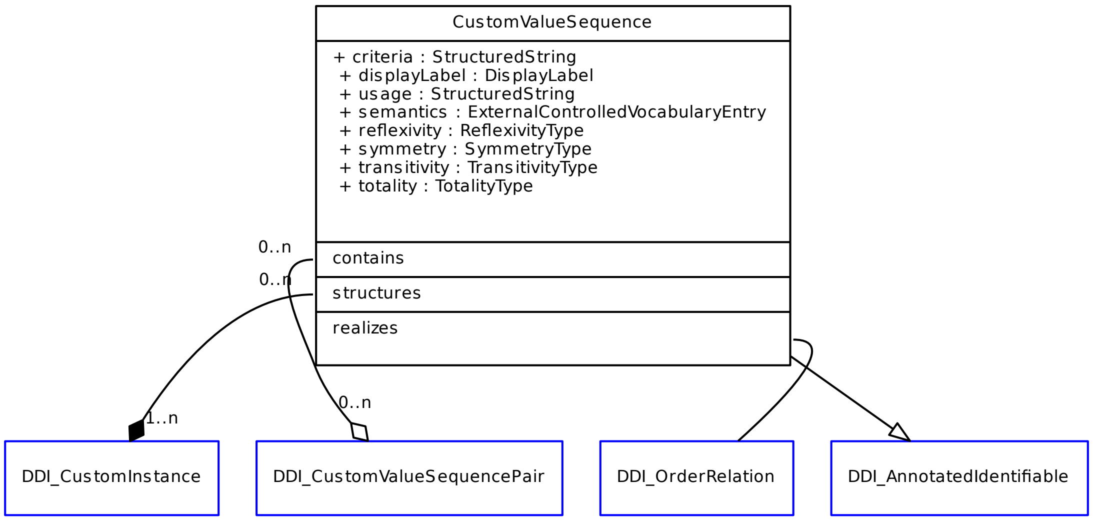
realizes pattern OrderRelation

structures

Provides structure (relationships) for a CustomInstance

Collection whose Members are grouped by pairs to support a variety of structures.

Graph


CustomValueSequence
+ criteria : StructuredString
+ displayLabel : DisplayLabel
+ usage : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ totality : TotalityType
contains
structures
realizes
0..n
0..n
1..n
0..n
DDI_CustomInstance
DDI_CustomValueSequencePair
DDI_OrderRelation
DDI_AnnotatedIdentifiable

6.3.17 CustomValueSequencePair

Defines a simple ordering between two CustomValuess - one precedes the other.

Extends

Identifiable

Relationships

Name	Type	Cardinality
predecessor	CustomValue	0..n
realizes	OrderedPair	0..n
successor	CustomValue	0..n

predecessor

The key,value pair which comes first. Specialization of ‘source’ in OrderedPair

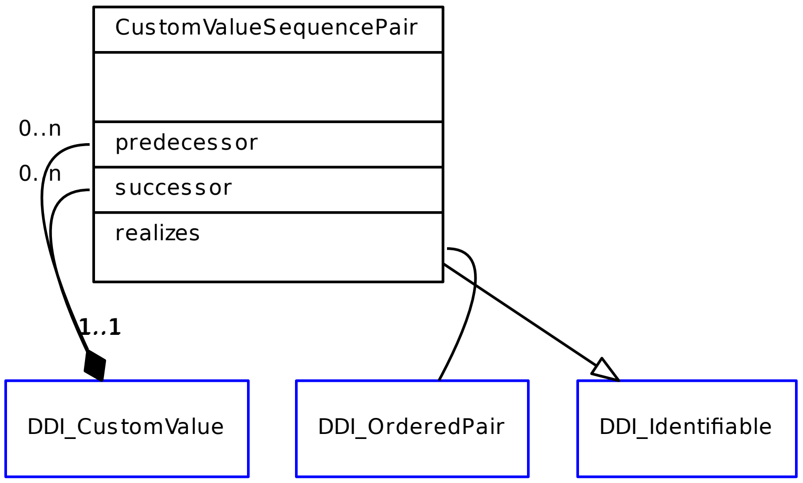
realizes

realizes pattern OrderedPair

successor

The key,value pair which follows. Specialization of ‘target’ in OrderedPair

Graph



6.3.18 VocabularyEntry

One value and its definition in an ordered list comprising a controlled vocabulary.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
definition	StructuredString	0..1
value	StructuredString	0..1

definition

An explanation (definition) of the value

value

the term being defined

‘

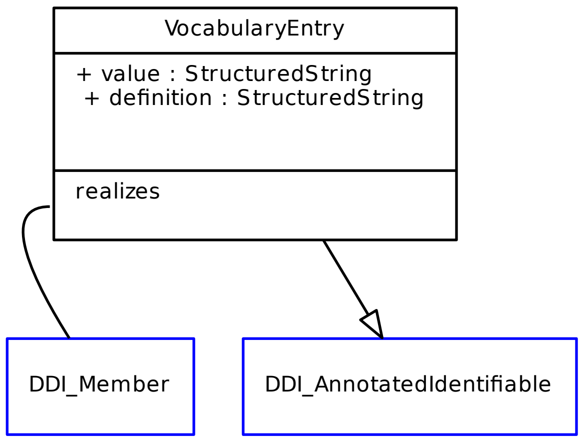
Relationships

Name	Type	Cardinality
realizes	Member	0..n

realizes

Class can play the role of a Member in a Collection

Graph


VocabularyEntry
+ value : StructuredString
+ definition : StructuredString
realizes
DDI_Member
DDI_AnnotatedIdentifiable

6.3.19 VocabularyParentChild

Contains the set of VocabularyParentChildPairs that define the a hierarchical relationship among the vocabulary entries.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
displayLabel	DisplayLabel	0..n
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

criteria

Intentional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

displayLabel

A display label for the OrderRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

reflexivity

Fixed to Anti-Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Anti-Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	VocabularyParentChildPair	1..n
realizes	ImmediatePrecedenceRelation	0..n
structures	ControlledVocabulary	0..n

contains

Enumerates the VocabularyParentChildPairs which taken together defines the hierarchy for the vocabulary

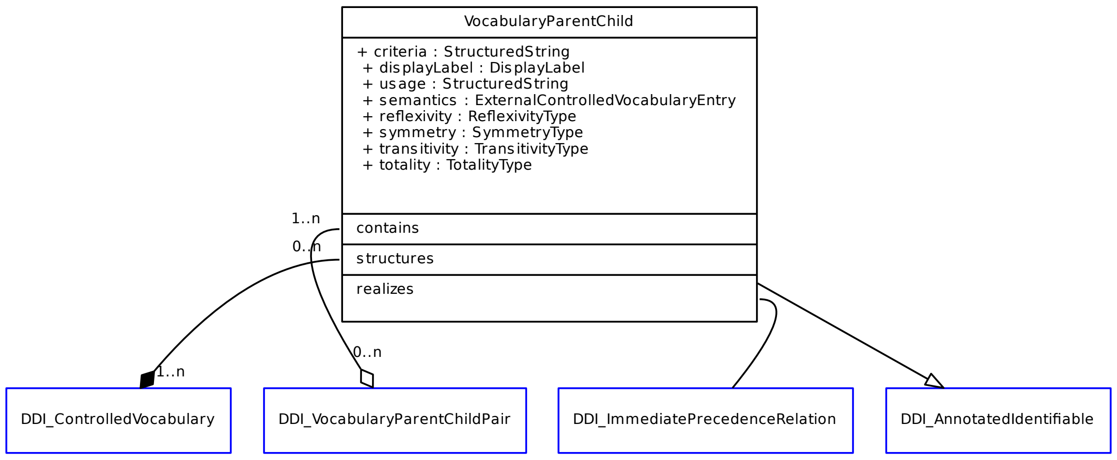
realizes

realizes pattern ImmediatePrecedenceRelation

structures

Collection whose Members are grouped by pairs to support a variety of structures.

Graph


VocabularyParentChild
+ criteria : StructuredString
+ displayLabel : DisplayLabel
+ usage : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ totality : TotalityType
contains
structures
realizes
1..n
0..n
1..n
0..n
DDI_ControlledVocabulary
DDI_VocabularyParentChildPair
DDI_ImmediatePrecedenceRelation
DDI_AnnotatedIdentifiable

6.3.20 VocabularyParentChildPair

a parent-child relationship between two vocabularyEntries

Extends

Identifiable

Relationships

Name	Type	Cardinality
child	VocabularyEntry	0..n
parent	VocabularyEntry	0..n
realizes	OrderedPair	0..n

child

the entry descending from the parent, the child. Specializes ‘target’ in OrderedPair.

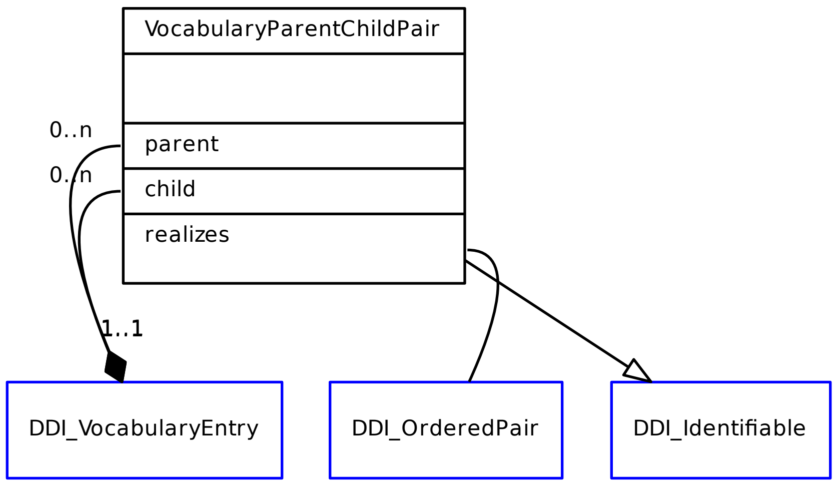
parent

The entry from which the child entry descends, the parent. Specializes ‘source’ in OrderedPair.

realizes

realizes pattern OrderedPair

Graph



6.3.21 VocabularySequence

Contains a set of VocabularySequencePairs which taken together define an ordering for a ControlledVocabulary

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
displayLabel	DisplayLabel	0..n
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

criteria

Intentional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

displayLabel

A display label for the OrderRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

reflexivity

Fixed to Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	VocabularySequencePair	0..n
realizes	OrderRelation	0..n
structures	ControlledVocabulary	0..n

contains

Enumerates the set of VocabularySequencePairs, each of which defines the order within the pair.

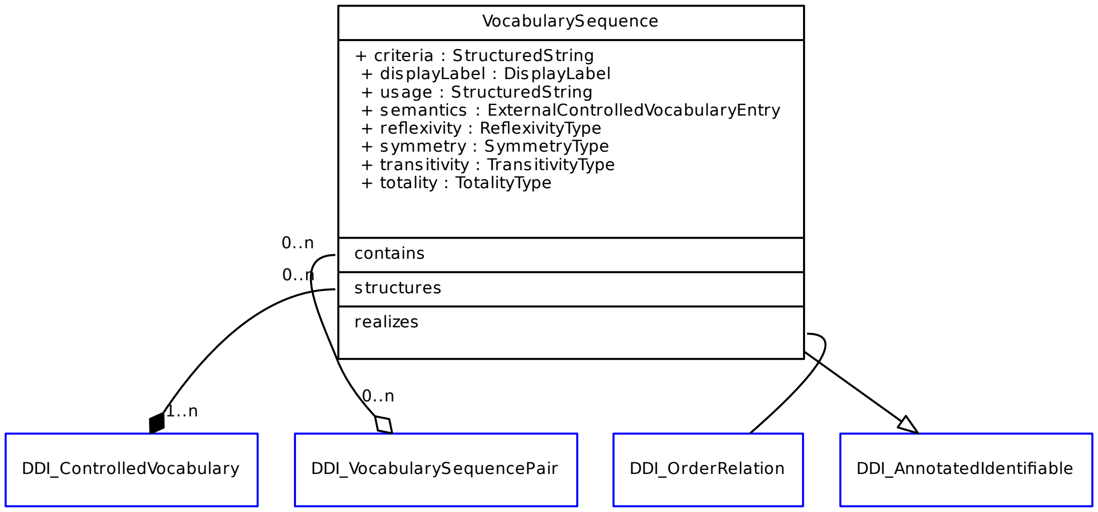
realizes

realizes pattern OrderRelation

structures

Collection whose Members are grouped by pairs to support a variety of structures.

Graph


VocabularySequence
+ criteria : StructuredString
+ displayLabel : DisplayLabel
+ usage : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ totality : TotalityType
contains
structures
realizes
0..n
0..n
1..n
0..n
DDI_ControlledVocabulary
DDI_VocabularySequencePair
DDI_OrderRelation
DDI_AnnotatedIdentifiable

6.3.22 VocabularySequencePair

Defines a simple ordering between two VocabularyEntry(s) one precedes the other.

Extends

Identifiable

Relationships

Name	Type	Cardinality
predecessor	VocabularyEntry	0..n
realizes	OrderedPair	0..n
successor	VocabularyEntry	0..n

predecessor

The vocabulary entry coming first in the order. Specialization of 'source' in OrderedPair

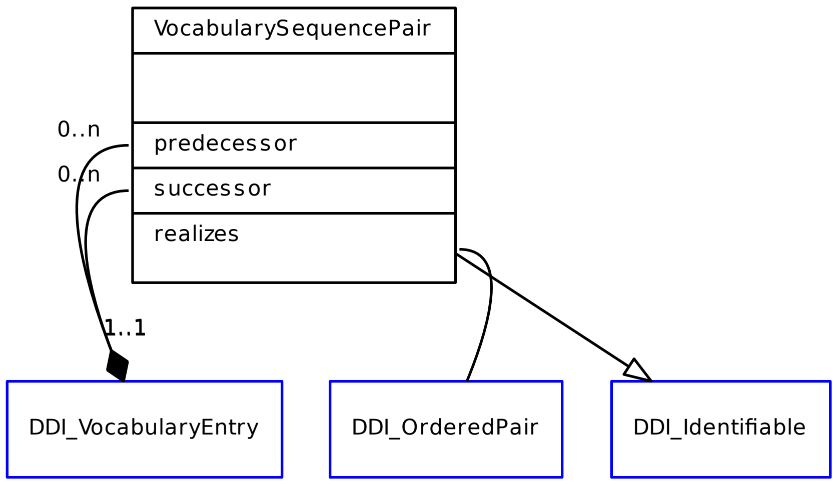
realizes

realizes pattern OrderedPair

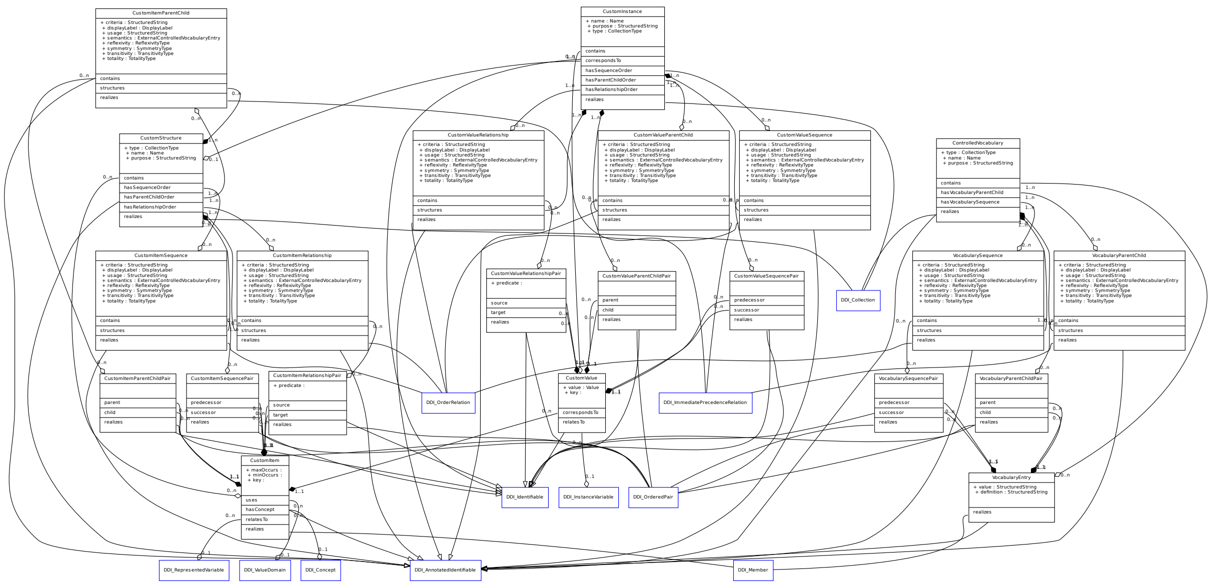
successor

The vocabulary entry coming second in the order. Specialization of 'target' in OrderedPair

Graph



6.3.23 Graph



6.4 ProcessPattern

A package is a administrative collection of classes in DDI. These are not namespaces.

6.4.1 InformationFlow

Mapping between inputs and outputs of Process Steps representing the flow of information to, from and within a Process.

Extends

UnorderedTuple

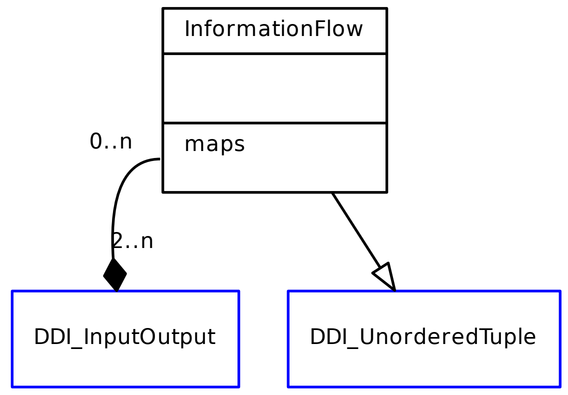
Relationships

Name	Type	Cardinality
maps	InputOutput	0..n

maps

Specialization of maps in Unordered Tuple for Input/Outputs of Process Steps

Graph



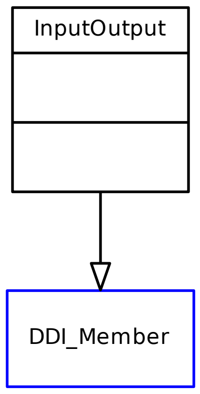
6.4.2 InputOutput

An Input or Output to a Process Step.

Extends

:ref:‘Member

Graph



6.4.3 Interface

Collection of Inputs and Outputs of the Process Step.

Extends

Collection

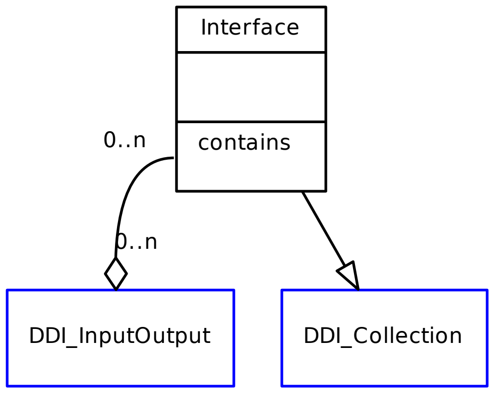
Relationships

Name	Type	Cardinality
contains	InputOutput	0..n

contains

Specialization of contains in Collection for Inputs and Outputs of Process Steps.

Graph



6.4.4 ProcessControlStep

A Process Step that controls the execution flow of the Process by determining the next Process Step in its scope.

Extends

ProcessStep

Relationships

Name	Type	Cardinality
contains	ProcessStep	0..n
defines	InformationFlow	0..n

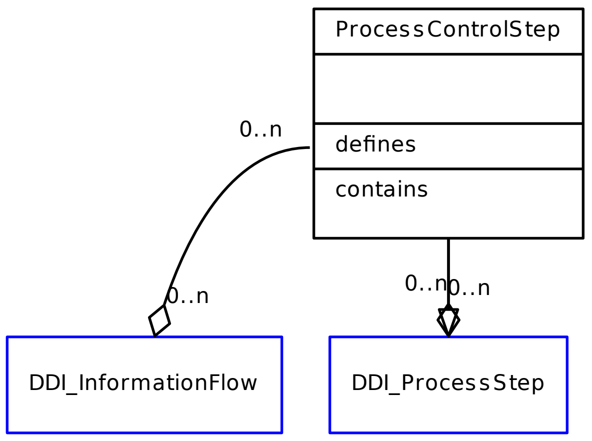
contains

Process Steps in scope of the Process Control Step

defines

Mappings between Inputs and Outputs of Process Steps in scope of the Process Control Step.

Graph



6.4.5 ProcessStep

One of the constituents of a Process. It can be a composition or atomic and might be performed by a Service.

Extends

Identifiable

Relationships

Name	Type	Cardinality
hasInterface	Interface	0..n
isPerformedBy	Service	0..n

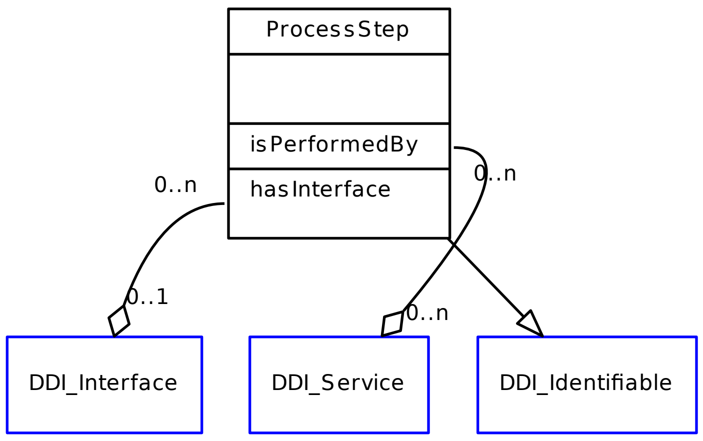
hasInterface

Collection of Inputs and Outputs of the Process Step.

isPerformedBy

Service performing the Process Step.

Graph



6.4.6 Service

A means of performing a Process Step as part of a Business Function (an ability that an organization possesses, typically expressed in general and high level terms and requiring a combination of organization, people, processes and technology to achieve).

Extends

Identifiable

Properties

Name	Type	Cardinality
interface	ExternalControlledVocabularyEntry	0..1
location	ExternalControlledVocabularyEntry	0..1

interface

Specifies how to communicate with the service.

location

Specifies where the service can be accessed.

‘

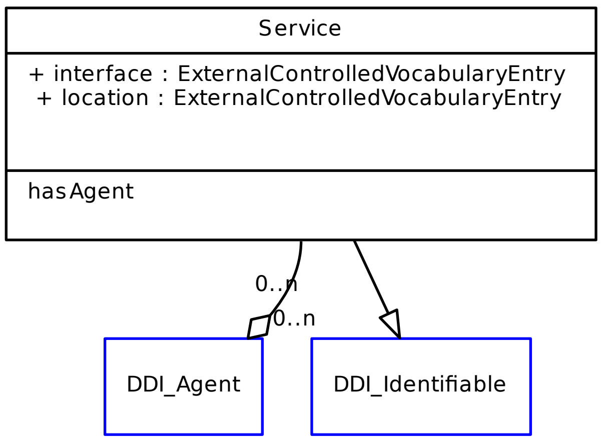
Relationships

Name	Type	Cardinality
hasAgent	Agent	0..n

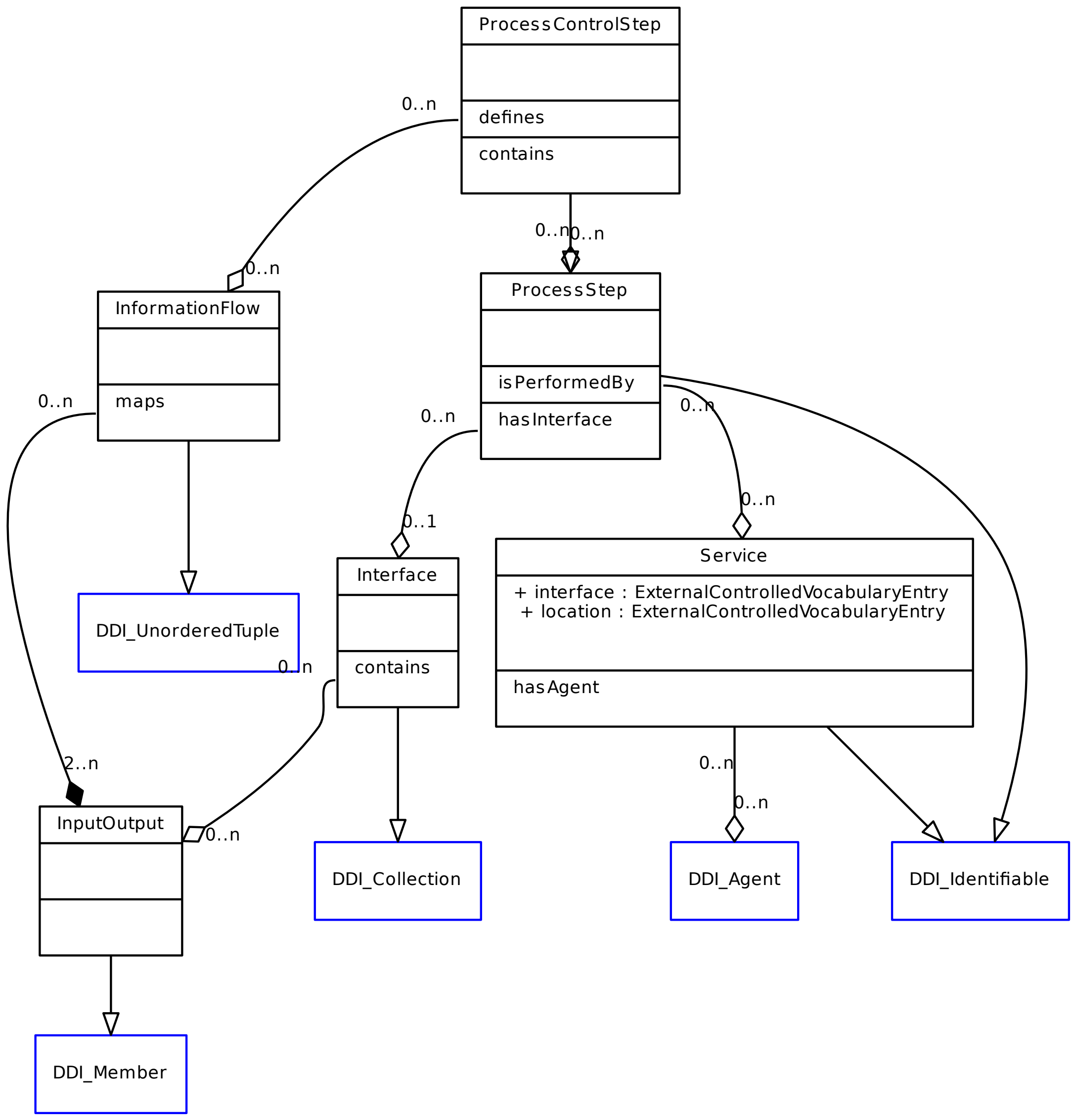
hasAgent

Actor that performs a role in the service,

Graph


Service
+ interface : ExternalControlledVocabularyEntry
+ location : ExternalControlledVocabularyEntry
hasAgent
0..n
0..n
DDI_Agent
DDI_Identifiable

6.4.7 Graph



6.5 MethodologyPattern

A package is a administrative collection of classes in DDI. These are not namespaces.

6.5.1 Algorithm

An algorithm is an effective method that can be expressed within a finite amount of space and time and in a well-defined formal language for calculating a function. Starting from an initial state and initial input (perhaps empty), the instructions describe a computation that, when executed, proceeds through a finite number of well-defined successive states, eventually producing "output" and terminating at a final ending state. The transition from one state to the next is not necessarily deterministic; some algorithms, known as randomized algorithms, incorporate random input. [from www.wikipedia.org 15 April 2016]

The underlying properties of the algorithm or method rather than the specifics of any particular implementation. In short a description of the method in its simplest and most general representation.

Extends

Identifiable

Properties

Name	Type	Cardinality
overview	StructuredString	0..1

overview

Provides a high level overview or summary of the class. Can be used to inform end-users or as part of an executive summary. Supports the use of multiple languages and structured text.

'

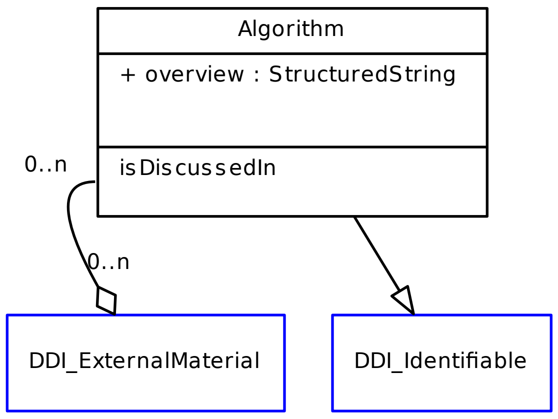
Relationships

Name	Type	Cardinality
isDiscussedIn	ExternalMaterial	0..n

isDiscussedIn

Identifies material discussing the algorithm. The material may be in DDI or other format.

Graph



6.5.2 Design

The design pattern class may be used to specify or, again, defines how a process will be performed in general. The design informs a specific or implemented process as to its general parameters.

Extends

Identifiable

Properties

Name	Type	Cardinality
overview	StructuredString	0..1

overview

Provides a high level overview or summary of the class. Can be used to inform end-users or as part of an executive summary. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
assumesPrecondition	Precondition	0..n
expressesDesign	Algorithm	0..n
isDiscussedIn	ExternalMaterial	0..n
specifiesGoal	Goal	0..n

assumesPrecondition

A precondition that must exist prior to the applied use of this design.

expressesDesign

An algorithm that defines the process used by the design in a generic way.

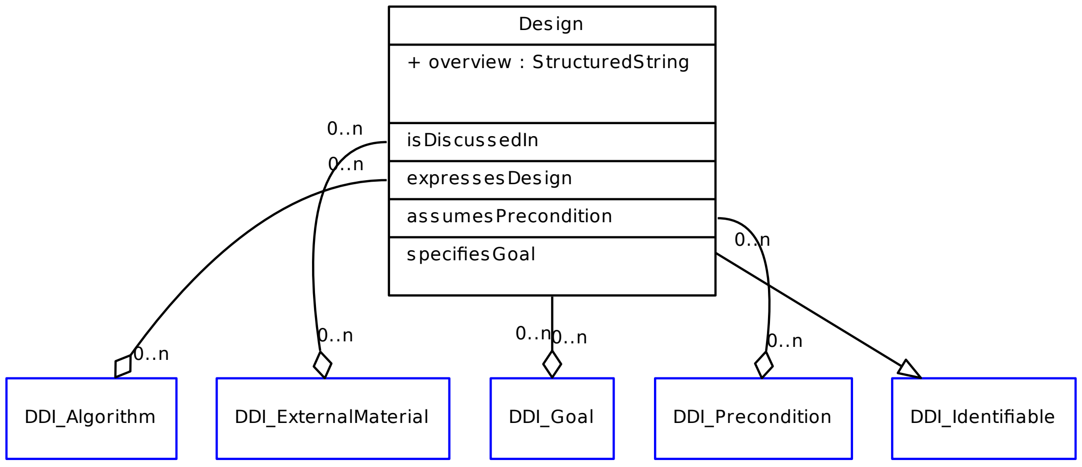
isDiscussedIn

External materials that describe or discuss the design

specifiesGoal

The generic goal of a design.

Graph



6.5.3 Methodology

Methodology may be used to define the design and process used within data collection, management, and use to achieve overall studies or specific activities such as sampling or weighting. A methodology in normally informed by earlier research and clarifies how earlier research methods were incorporated into the current work.

Extends

Identifiable

Properties

Name	Type	Cardinality
name	Name	0..n
overview	StructuredString	0..1
rationale	StructuredString	0..1
usage	StructuredString	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

overview

Short natural language account of the information obtained from the combination of properties and relationships associated with an object. Supports the use of multiple languages and structured text.

rationale

Explanation of the reasons some decision was made or some object exists. Supports the use of multiple languages and structured text.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
componentMethodology	Methodology	0..n
hasDesign	Design	0..n
hasProcess	Process	0..n
isExpressedBy	Algorithm	0..n

componentMethodology

A methodology which is a component part of this Methodology

hasDesign

Design used to form the Methodology

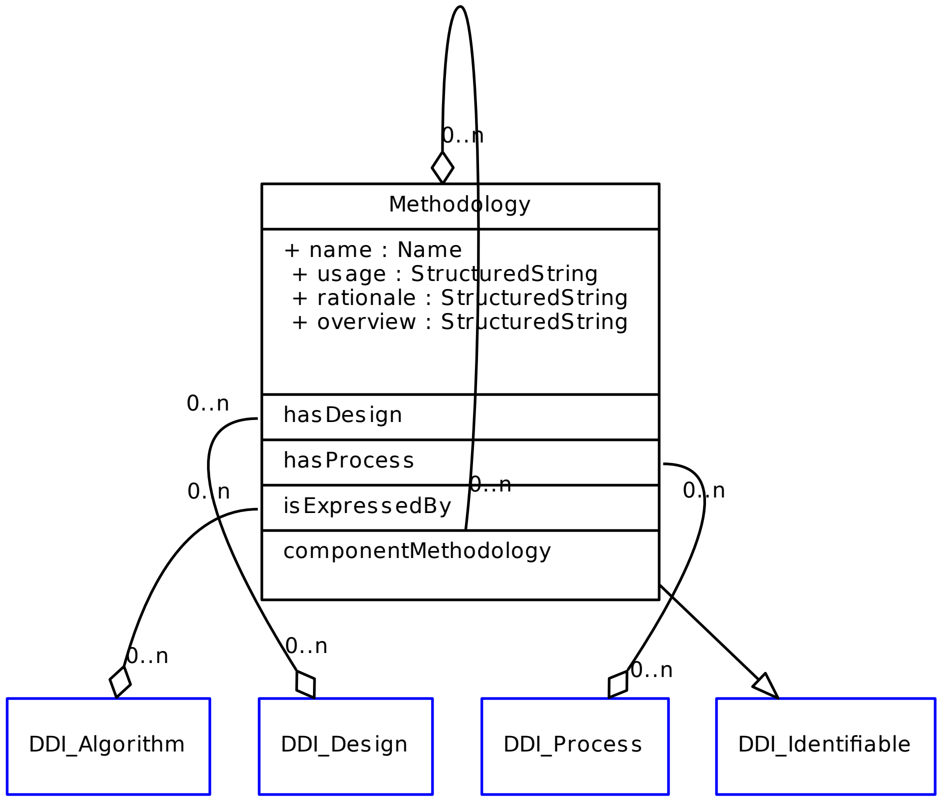
hasProcess

Process used to implement the Methodology

isExpressedBy

An algorithm that defines the Methodology in a generic way.

Graph



6.5.4 Process

Process describes how a design is used to implement a goal. It is the series of steps taken as a whole. It is decomposable into ProcessSteps, but this decomposition is not necessary.

Extends

Identifiable

Properties

Name	Type	Cardinality
overview	StructuredString	0..1

overview

Provides a high level overview or summary of the class. Can be used to inform end-users or as part of an executive summary. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	ProcessStep	0..n
hasDesign	Design	0..1
hasPrecondition	Precondition	0..n
hasResults	Result	0..n
implementsAlgorithm	Algorithm	0..n
isDiscussedIn	ExternalMaterial	0..n

contains

The process step that initiates the detailed process.

hasDesign

The design informs the specific process by providing the underlying rationale, rules and general preconditions for the process. The design is the reusable part of the implemented process.

hasPrecondition

Any condition that must be met prior to the process being implemented.

hasResults

The end result of the process which can contain a binding to specific outcome collections as well as usage instructions for implementing results

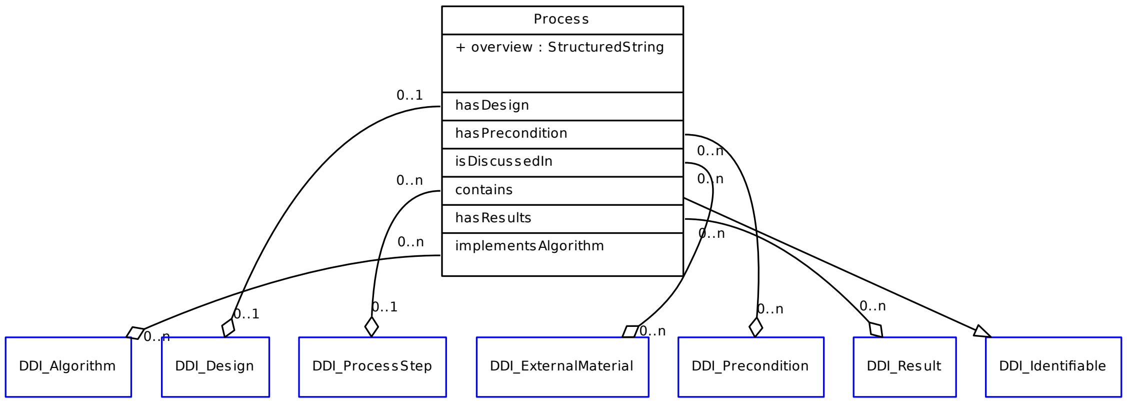
implementsAlgorithm

An implementation and/or execution of an algorithm which is associated with the design of the process.

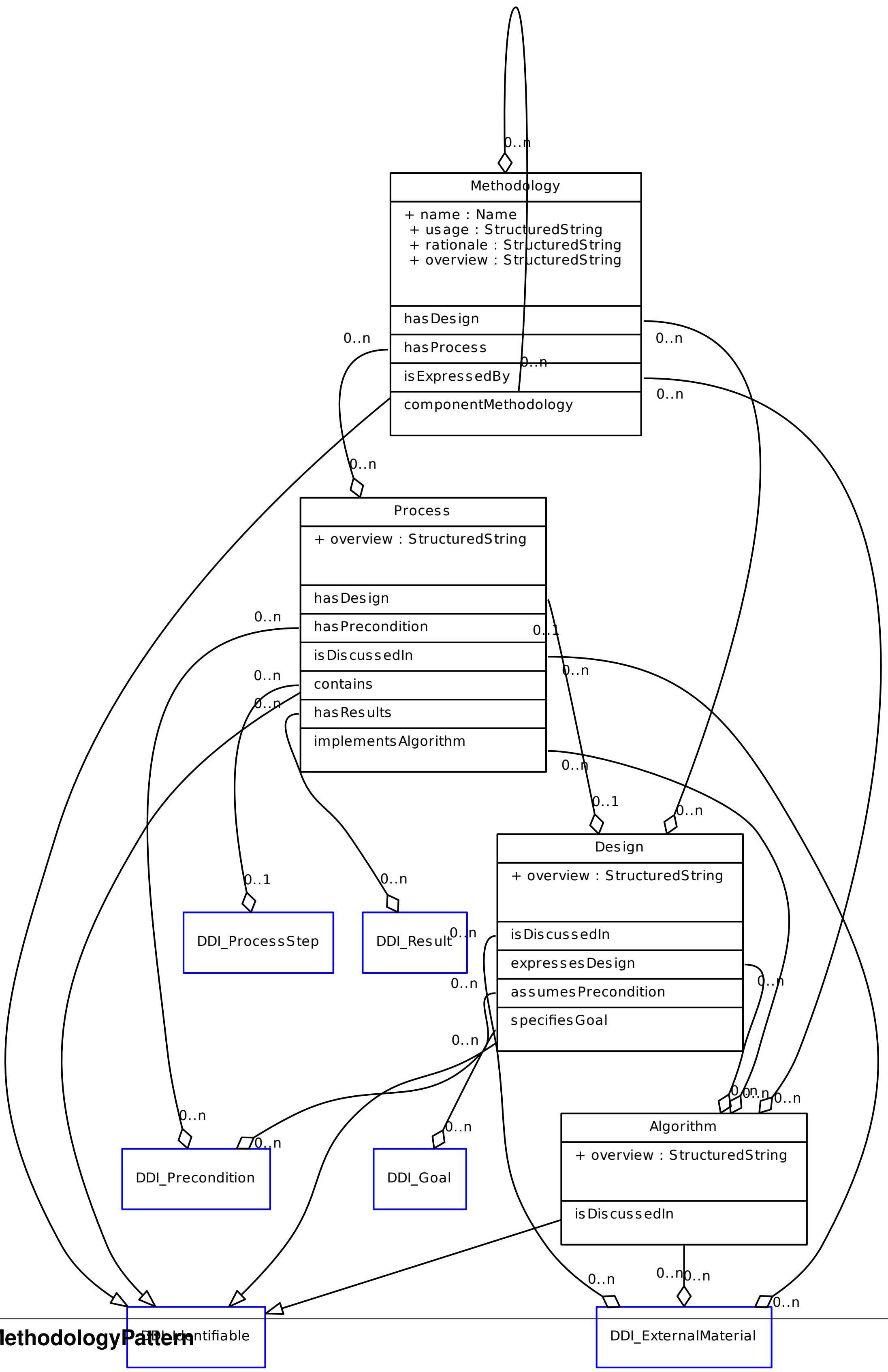
isDiscussedIn

Identifies material discussing the process. The material may be in DDI or other format.

Graph



6.5.5 Graph


Methodology
+ name : Name
+ usage : StructuredString
+ rationale : StructuredString
+ overview : StructuredString
hasDesign
hasProcess
isExpressedBy
componentMethodology
Process
+ overview : StructuredString
hasDesign
hasPrecondition
isDiscussedIn
contains
hasResults
implementsAlgorithm
Design
+ overview : StructuredString
isDiscussedIn
expressesDesign
assumesPrecondition
specifiesGoal
Algorithm
+ overview : StructuredString
isDiscussedIn
DDI_ProcessStep
DDI_Result
DDI_Precondition
DDI_Goal
DDI_Identifiable
DDI_ExternalMaterial
0..n
0..1

6.6 SegmentLocationSpecifications

A package is a administrative collection of classes in DDI. These are not namespaces.

6.6.1 BeginEndClip

Contains parameters for the beginning and ending of a Clip of a type drawn from a controlled vocabulary

Extends

Identifiable

Properties

Name	Type	Cardinality
clipBegin	Real	0..1
clipEnd	Real	0..1
clipType	ExternalControlledVocabularyEntry	0..1
otherClipType	String	0..1

clipBegin

The parameter for the start of the clip

clipEnd

the parameter for the end of the clip

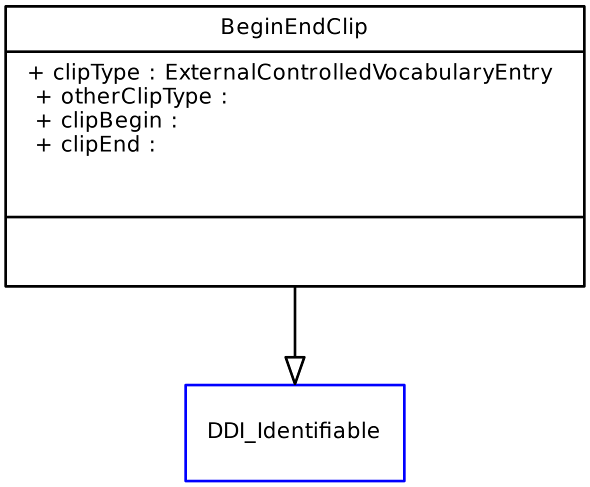
clipType

The type of clip, preferably drawn from a controlled vocabulary.

otherClipType

Text description of a choice of “other”

Graph



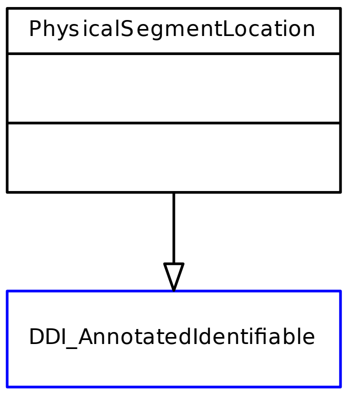
6.6.2 PhysicalSegmentLocation

Defines the location of a segment in a DataStore (e.g. a text file). This is abstract since there are many different ways to describe the location of a segment - character counts, start and end times, etc.

Extends

:ref:‘AnnotatedIdentifiable

Graph



6.6.3 SegmentByAudio

A segment (clip) of an audio file

Extends

PhysicalSegmentLocation

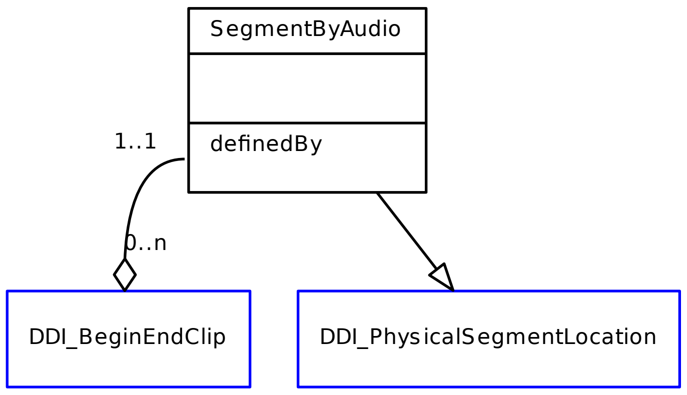
Relationships

Name	Type	Cardinality
definedBy	BeginEndClip	1..1

definedBy

The BeginEnd parameter specification(s) for the clip

Graph



6.6.4 SegmentByCustom

Points to a CustomInstance and a StructureForSegmentByCustom to define segments using a scheme not built in to DDI. The CustomInstance contains a set of key, value pairs defining each segment. The StructureForSegmentByCustom defines the keys.

Extends

PhysicalSegmentLocation

Relationships

Name	Type	Cardinality
definedBy	CustomInstance	1..1
usesStructure	StructureForSegmentOfCustom	0..n

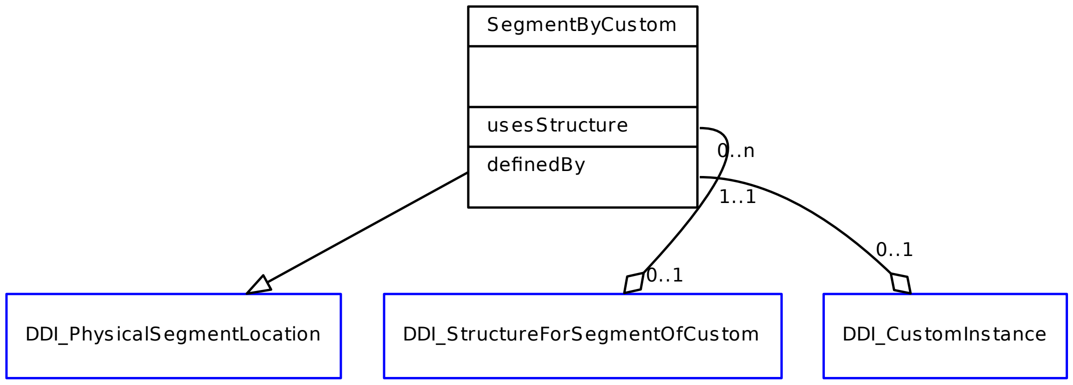
definedBy

A CustomInstance which contains a set of CustomValues (key,value pairs) using the CustomStructure to define the location of a segment

usesStructure

The CustomStructure defined for this method of describing segments

Graph



6.6.5 SegmentByDDI

Points to a set of DDI identifiers that together identify a segment of DDI.

Extends

PhysicalSegmentLocation

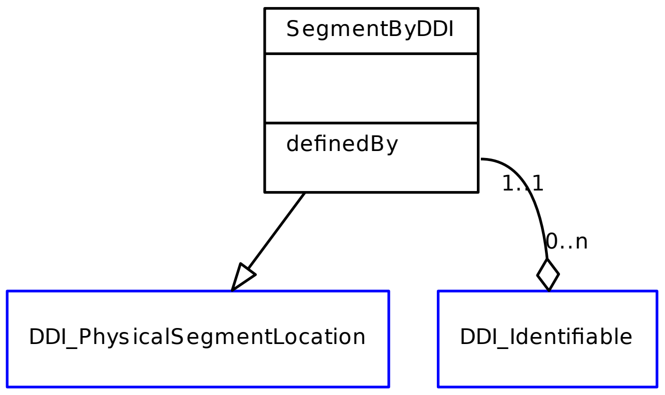
Relationships

Name	Type	Cardinality
definedBy	Identifiable	1..1

definedBy

Pointers to any DDI objects defined by a DDI Identifier (Agent, ID, Version)

Graph



6.6.6 SegmentByImage

Defines an area of an image - e.g. by a rectangle, a polygon

Extends

PhysicalSegmentLocation

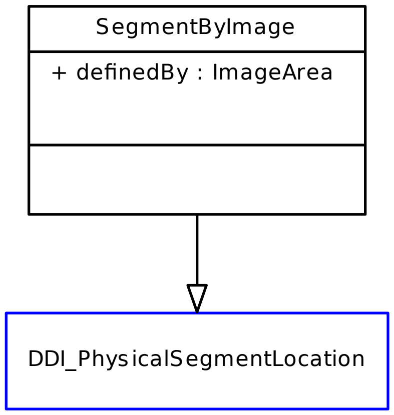
Properties

Name	Type	Cardinality
definedBy	ImageArea	0..1

definedBy

The set of parameters describing the polygon, square, etc

Graph



6.6.7 SegmentByNomenclature

The description of a segment using a descriptive term from a controlled vocabulary.

Extends

PhysicalSegmentLocation

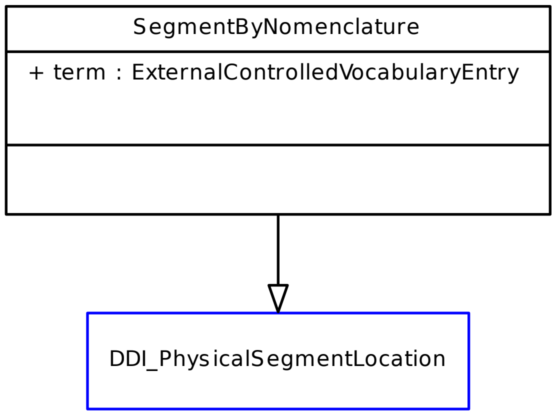
Properties

Name	Type	Cardinality
term	ExternalControlledVocabularyEntry	0..n

term

Points to the descriptive term expressed as a codeValue along with related information

Graph


SegmentByNomenclature
+ term : ExternalControlledVocabularyEntry
DDI_PhysicalSegmentLocation

6.6.8 SegmentBySqlQuery

Points to one or more specifications of an SQL query and related information

Extends

PhysicalSegmentLocation

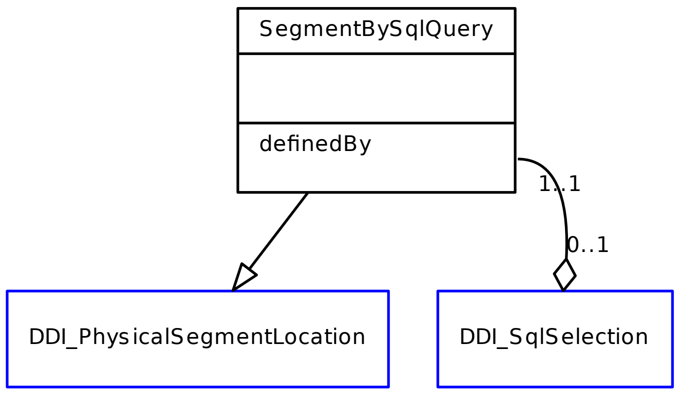
Relationships

Name	Type	Cardinality
definedBy	SqlSelection	1..1

definedBy

Pointer to the SQL statement and related information

Graph



6.6.9 SegmentByText

Defines the location of a segment of text.

Extends

PhysicalSegmentLocation

Properties

Name	Type	Cardinality
definedByCharacterOffset	CharacterOffset	0..n
definedByLineParameters	LineParameter	0..n

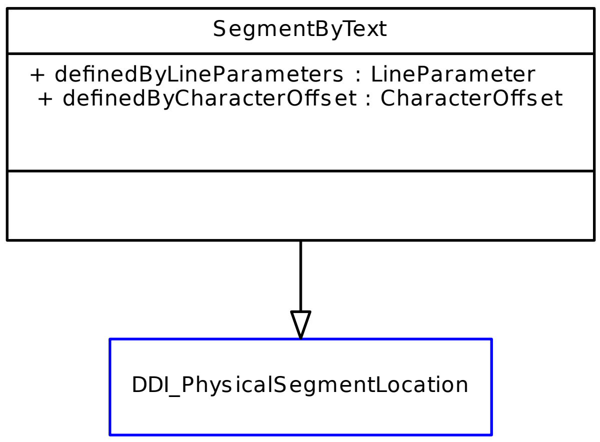
definedByCharacterOffset

Points to parameter pairs using start character (from the beginning of the document) and end character

definedByLineParameters

Points to parameters using start line and character withing line and end line and character within that line

Graph


SegmentByText
+ definedByLineParameters : LineParameter
+ definedByCharacterOffset : CharacterOffset
DDI_PhysicalSegmentLocation

6.6.10 SegmentByTriplestore

Points to a SPARQL query and related information

Extends

PhysicalSegmentLocation

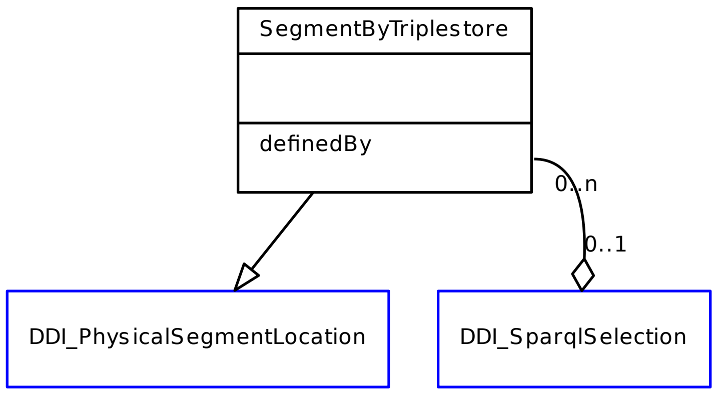
Relationships

Name	Type	Cardinality
definedBy	SparqlSelection	0..n

definedBy

Pointer to the SPARQL query and related information defining a segment of the triplestore

Graph



6.6.11 SegmentByVideo

Points to the BeginEndClip parameter specifications for a Video Clip

Extends

PhysicalSegmentLocation

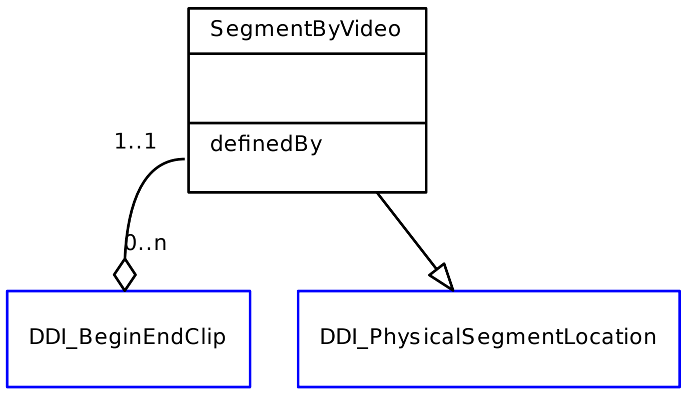
Relationships

Name	Type	Cardinality
definedBy	BeginEndClip	1..1

definedBy

The BeginEndClip parameter specification(s) for the clip

Graph



6.6.12 SegmentByXML

Defines a segment of an XML file (or well formed HTML) using an XPointer.

Extends

PhysicalSegmentLocation

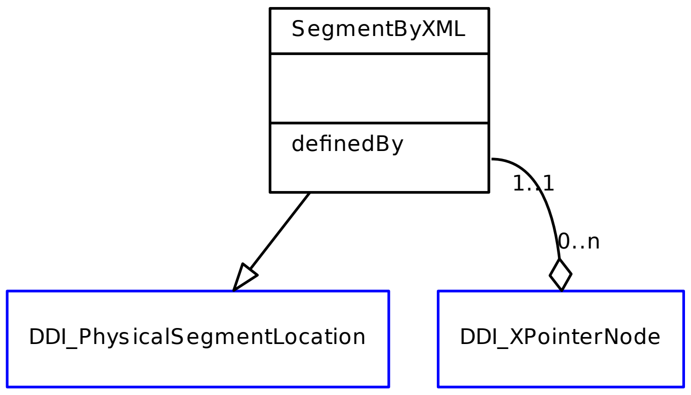
Relationships

Name	Type	Cardinality
definedBy	XPointerNode	1..1

definedBy

The XPointers defining the segment.

Graph



6.6.13 SparqlSelection

Contains a SPARQL query and the necessary related information for retrieving a segment from a triple store.

Extends

Identifiable

Properties

Name	Type	Cardinality
endpoint	String	0..1
sparqlQuery	String	0..1

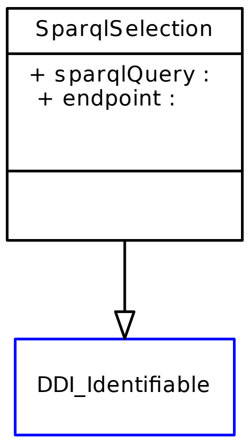
endpoint

A description of the endpoint to be queried by the SPARQL Query

sparqlQuery

The text of the SPARQL query

Graph



6.6.14 SqlSelection

Contains an SQL statement and related information allowing the description of a segment of a relational database.

Extends

Identifiable

Properties

Name	Type	Cardinality
databaseType	ExternalControlledVocabularyEntry	0..1
databaseVersion	String	0..1
schemeName	String	0..1
sqlStatement	String	0..1

databaseType

The type of the database. An example might be MySQL

databaseVersion

The version of the database. For MySQl an example might be 5.7

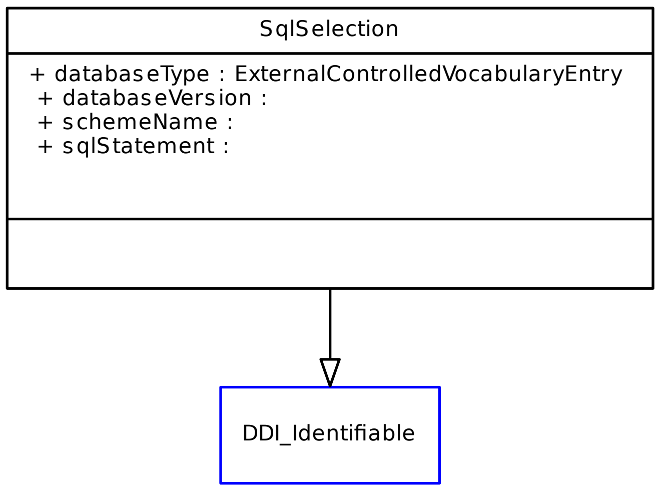
schemeName

The database schema

sqlStatement

The SQL statment (query)

Graph



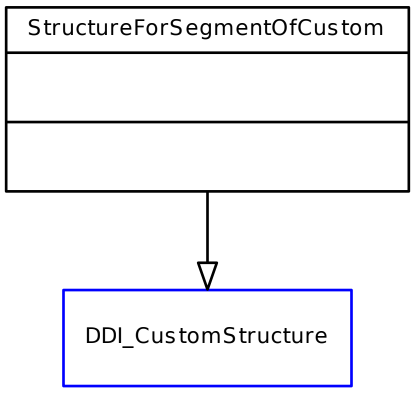
6.6.15 StructureForSegmentOfCustom

Defines a custom set of key,value pairs for the description of the location of segments in some type of PhysicalResource

Extends

:ref:‘CustomStructure

Graph



6.6.16 XPointerNode

Contains an XPointer for identifying part of an XML document

Extends

Identifiable

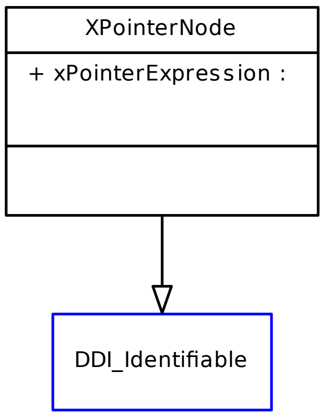
Properties

Name	Type	Cardinality
xPointerExpression	String	0..1

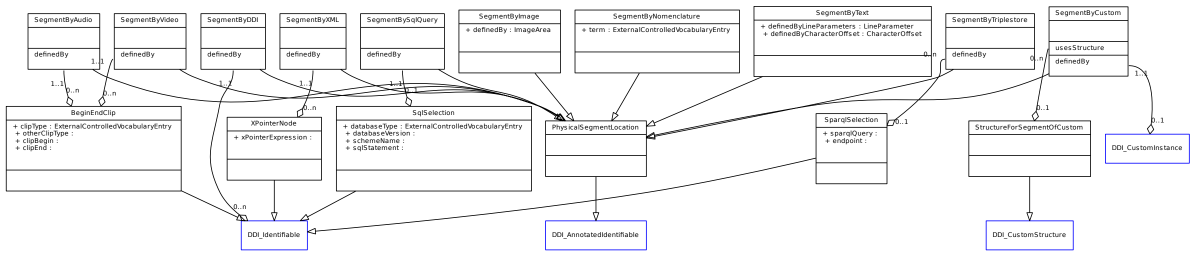
xPointerExpression

an XPointerExpression (http://www.w3.org/TR/xptr-xpointer/)

Graph



6.6.17 Graph



6.7 FormatDescription

A package is a administrative collection of classes in DDI. These are not namespaces.

6.7.1 TableDirectionValues

Graph

6.7.2 TextDirectionValues

Graph

6.7.3 TrimValues

Graph

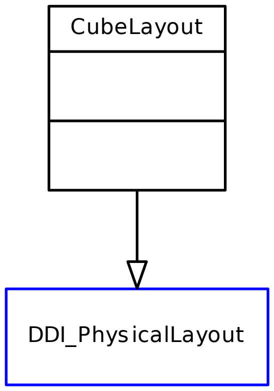
6.7.4 CubeLayout

The CubeLayout supports the description of the structure of aggregate data, and how it may be linked to the microdata of which it is a cross-tabulation.

Extends

:ref:'PhysicalLayout

Graph



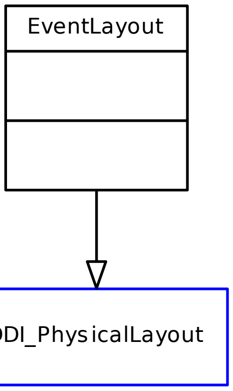
6.7.5 EventLayout

The EventLayout exists to permit the description of data stored in a “tall” dataset format. For event data typically each row in the dataset will contain a single data point’s value, a unit identifier, a reference to a variable, a start time, and end time, and a datatype (that of the referenced variable).

Extends

:ref:‘PhysicalLayout

Graph



6.7.6 PhysicalLayout

The PhysicalLayout is an abstract class used as an extension point in the description of the different layout styles of data structure description. Examples include rectangular layouts, event data layouts, and cube layouts (e.g. summary data).

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
arrayBase	Integer	0..1
commentPrefix	String	0..1
delimiter	String	0..1
encoding	ExternalControlledVocabularyEntry	0..1
escapeCharacter	String	0..1
hasHeader	Boolean	0..1
headerIsCaseSensitive	Boolean	0..1
headerRowCount	Integer	0..1
isDelimited	Boolean	1..1
isFixedWidth	Boolean	1..1
lineTerminator	String	0..n
name	Name	0..n
nullSequence	String	0..1
overview	StructuredString	0..1
purpose	StructuredString	0..1
quoteCharacter	String	0..1
skipBlankRows	Boolean	0..1
skipDataColumns	Integer	0..1
skipInitialSpace	Boolean	0..1
skipRows	Integer	0..1
tableDirection	TableDirectionValues	0..1
textDirection	TextDirectionValues	0..1
treatConsecutiveDelimitersAsOne	Boolean	0..1
trim	TrimValues	0..1
type	CollectionType	0..1

arrayBase

Number cells, rows, columns starting from this value. This is a DDI notion allowing numbering of columns, rows, etc. from either 0 or 1. W3C appears to just use 1. From https://www.w3.org/TR/tabular-data-model/ 4.3 “number — the position of the column amongst the columns for the associated table, starting from 1.”

commentPrefix

A string used to indicate that an input line is a comment, a string which precedes a comment in the data file. From https://www.w3.org/TR/tabular-metadata/ 5.9 Dialect commentPrefix: “An atomic property that sets the comment prefix flag to the single provided value, which MUST be a string. The default is “#”.”

delimiter

The Delimiting character in the data. Must be used if isDelimited is true. “The separator between cells, set by the delimiter property of a dialect description. The default is ,.” see https://www.w3.org/TR/tabular-data-model/#encoding From https://www.w3.org/TR/tabular-metadata/ 5.9 Dialect delimiter: “An atomic property that sets the delimiter flag to the single provided value, which MUST be a string. The default is ”,”.” From https://www.w3.org/TR/tabular-data-model/ 8 “The separator between cells, set by the delimiter property of a dialect description. The default is ,.” From http://specs.frictionlessdata.io/csv-dialect/ delimiter: “specifies a one-character string to use as the field separator. Default = ,”

encoding

The character encoding of the represented data. From https://www.w3.org/TR/tabular-metadata/ 5.9 Dialect: encoding: "An atomic property that sets the encoding flag to the single provided string value, which MUST be a defined in [encoding]. The default is "utf-8"." From https://www.w3.org/TR/tabular-metadata/ 7.2 Encoding: "CSV files should be encoded using UTF-8, and should be in Unicode Normal Form C as defined in [UAX15]. If a CSV file is not encoded using UTF-8, the encoding should be specified through the charset parameter in the Content-Type header:"

escapeCharacter

"The string that is used to escape the quote character within escaped cells, or null" see https://www.w3.org/TR/tabular-data-model/#encoding From https://www.w3.org/TR/tabular-metadata/ 5.9 Dialect " doupleQuote: A boolean atomic property that, if true, sets the escape character flag to ". If false, to . The default is true." From http://specs.frictionlessdata.io/csv-dialect/ doubleQuote: "controls the handling of quotes inside fields. If true, two consecutive quotes should be interpreted as one. Default = true"

hasHeader

True if the file contains a header containing column names. From https://www.w3.org/TR/tabular-metadata/ 5.9 Dialect header: "A boolean atomic property that, if true, sets the header row count flag to 1, and if false to 0, unless headerRowCount is provided, in which case the value provided for the header property is ignored. The default is true." From http://specs.frictionlessdata.io/csv-dialect/ header: "indicates whether the file includes a header row. If true the first row in the file is a header row, not data. Default = true"

headerIsCaseSensitive

If True, the case of the labels in the header is significant. From http://specs.frictionlessdata.io/csv-dialect/ caseSensitiveHeader: "indicates that case in the header is meaningful. For example, columns CAT and Cat should not be equated. Default = false"

headerRowCount

The number of lines in the header From https://www.w3.org/TR/tabular-metadata/ 5.9 Dialect headerRowCount: "A numeric atomic property that sets the header row count flag to the single provided value, which MUST be a non-negative integer. The default is 1."

isDelimited

Indicates that the data are in a delimited format. If not set to true, is FixedWitdh must be set to true.

isFixedWidth

Set to true if the file is fixed-width. If true, isDelimited must be set to false.

lineTerminator

"The strings that can be used at the end of a row, set by the lineTerminators property of a dialect description. The default is [CRLF, LF]." see https://www.w3.org/TR/tabular-data-model/#encoding From https://www.w3.org/TR/tabular-metadata/ 5.9 Dialect lineTerminators: "An atomic property that sets the line terminators flag to either an array containing the single provided string value, or the provided array. The default is ["rn", "n"]." From http://specs.frictionlessdata.io/csv-dialect/ lineTerminator: "specifies the character sequence which should terminate rows. Default = rn"

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

nullSequence

A string indicating a null value. From https://www.w3.org/TR/tabular-metadata/ 4.3 "null — the string or strings which cause the value of cells having string value matching any of these values to be null." From Inherited 5.7 null: "An atomic property giving the string or strings used for null values within the data. If the string value of the cell is equal to any one of these values, the cell value is null. See Parsing Cells in [tabular-data-model] for more details. If not specified, the default for the null property is the empty string "". The value of this property becomes the null annotation for the described column." From http://specs.frictionlessdata.io/csv-dialect/ nullSequence: "specifies the null sequence (for example N). Not set by default"

overview

Short natural language account of the physical layout obtained from the combination of associated properties and relationships. Supports the use of multiple languages and structured text.

purpose

> Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

quoteCharacter

The string that is used around escaped cells, or null, set by the quoteChar property of a dialect description. The default is ". see https://www.w3.org/TR/tabular-data-model/#encoding From https://www.w3.org/TR/tabular-metadata/ 5.9 Dialect quoteChar: "An atomic property that sets the quote character flag to the single provided value, which MUST be a string or null. If the value is null, the escape character flag is also set to null. The default is ""." From http://specs.frictionlessdata.io/csv-dialect/ quoteChar: "specifies a one-character string to use as the quoting character. Default = ""

skipBlankRows

If True, blank rows are ignored. From https://www.w3.org/TR/tabular-metadata/ 5.9 Dialect skipBlankRows: "A boolean atomic property that sets the skip blank rows flag to the single provided boolean value. The default is false."

skipDataColumns

The number of columns to skip at the beginning of the row. From https://www.w3.org/TR/tabular-metadata/ 5.9 Dialect skipColumns: "A numeric atomic property that sets the skip columns flag to the single provided numeric value, which MUST be a non-negative integer. The default is 0." See also https://www.w3.org/TR/tabular-metadata/ 8 skip columns: "The number of columns to skip at the beginning of each row, set by the skipColumns property of a dialect description. The default is 0. A value other than 0 will mean that the source numbers of columns will be different from their numbers."

skipInitialSpace

If true skip spaces at the beginning of a line or following a delimiter. From https://www.w3.org/TR/tabular-metadata/ 5.9 Dialect skipInitialSpace: "A boolean atomic property that, if true, sets the trim flag to "start" and if false, to false. If the trim property is provided, the skipInitialSpace property is ignored. The default is false." From http://specs.frictionlessdata.io/csv-dialect/ skipInitialSpace: "specifies how to interpret whitespace which immediately follows a delimiter; if false, it means that whitespace immediately after a delimiter should be treated as part of the following field. Default = true"

skipRows

Number of input rows to skip preceding the header or data From https://www.w3.org/TR/tabular-metadata/ 5.9 Dialect skipRows: "A numeric atomic property that sets the skip rows flag to the single provided numeric value, which MUST be a non-negative integer. The default is 0." From https://www.w3.org/TR/tabular-metadata/ 8 skip rows: "The number of rows to skip at the beginning of the file, before a header row or tabular data, set by the skipRows property of a dialect description. The default is 0. A value greater than 0 will mean that the source numbers of rows will be different from their numbers."

tableDirection

Indicates the direction in which columns are arranged in each row. From https://www.w3.org/TR/tabular-metadata/ 5.3.2 tableDirection: "An atomic property that MUST have a single string value that is one of "rtl", "ltr", or "auto". Indicates whether the tables in the group should be displayed with the first column on the right, on the left, or based on the first character in the table that has a specific direction. The value of this property becomes the value of the table direction annotation for all the tables in the table group. See Bidirectional Tables in [tabular-data-model] for details. The default value for this property is "auto"."

textDirection

Indicates the reading order of text within cells. From https://www.w3.org/TR/tabular-metadata/ Inherited 5.7 textDirection: "An atomic property that MUST have a single string value that is one of "ltr", "rtl", "auto" or "inherit" (the default). Indicates whether the text within cells should be displayed as left-to-right text (ltr), as right-to-left text (rtl), according to the content of the cell (auto) or in the direction inherited from the table direction annotation of the table. The value of this property determines the text direction annotation for the column, and the text direction annotation for the cells within that column: if the value is inherit then the value of the text direction annotation is the value of the table direction annotation on the table, otherwise it is the value of this property. See Bidirectional Tables in [tabular-data-model] for details."

treatConsecutiveDelimitersAsOne

If True, consecutive delimiters are treated the same way as a single delimiter, if false consiecutive delimiters indicate a missing value. From PHDD consecutiveDelimitersAsOne: "Indicates how consecutive delimiters should be handed by the software. Equivalent element in DDI 3.2: p:Delimiter/@treatConsecutiveDelimiterAsOne DDI 3.2 Documentation: http://www.ddialliance.org/Specification/DDI-Lifecycle/3.2/XMLSchema/FieldLevelDocumentation/schemas/reusable_xsd/complexTy

trim

Specifies which spaces to remove from a data value (start, end, both, neither) From https://www.w3.org/TR/tabular-metadata/ 5.9 Dialect trim: "An atomic property that, if the boolean true, sets the trim flag to true and if the boolean false to false. If the value provided is a string, sets the trim flag to the provided value, which MUST be one of "true", "false", "start", or "end". The default is true."

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

'

Relationships

Name	Type	Cardinality
containsValueMapping	ValueMapping	0..1
formatsLogicalRecordLayout	LogicalRecordLayout	0..n
realizes	Collection	0..n

containsValueMapping

The PhysicalLayout contains a description of the physical representation of each InstanceVariable in a ValueMapping. This relationship fills the role of "Contains" from the Collection pattern.

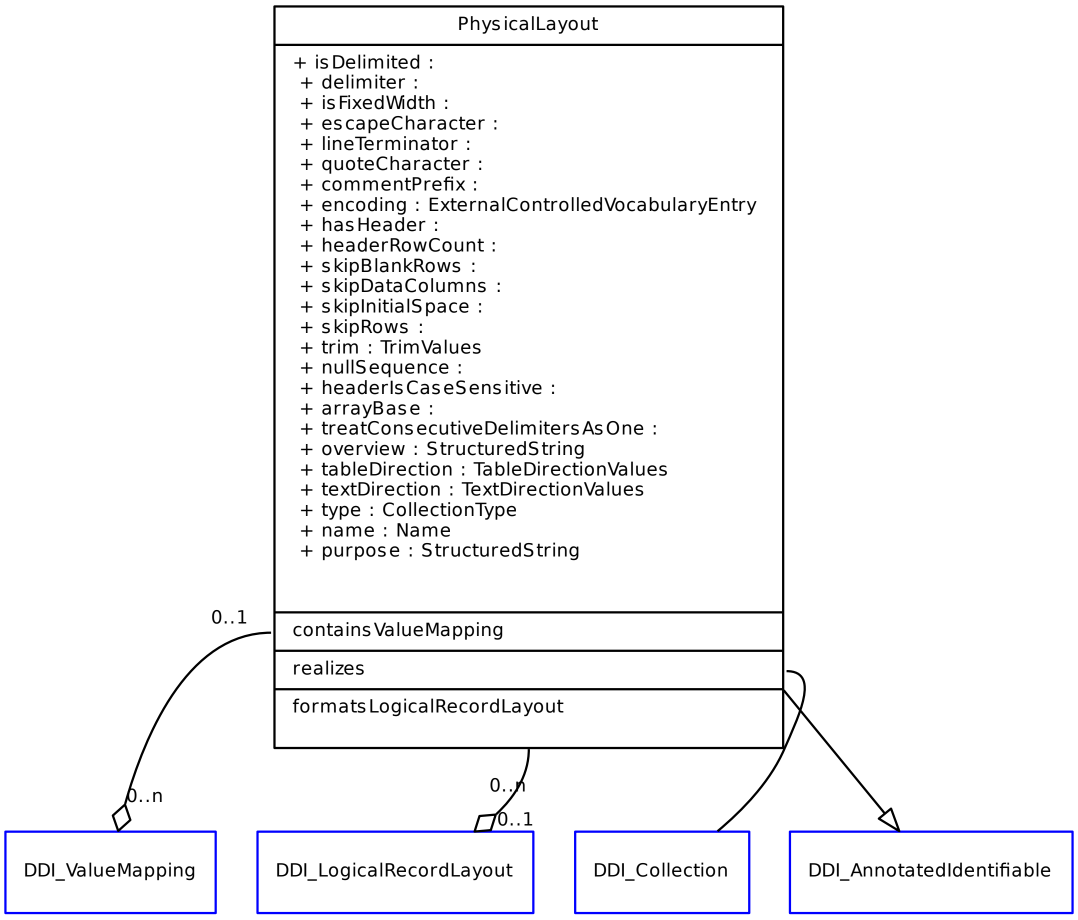
formatsLogicalRecordLayout

Logical Record Layout physically represented by the Physical Layout.

realizes

This relationship allows the PhysicalLayout to play the role of a collection that contains ValueMappings.

Graph



6.7.7 PhysicalLayoutOrder

A realization of OrderRelation that is used to describe the sequence of Value Mappings in a Physical Layout.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
transitivity	TransitivityType	1..1

criteria

Intensional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

reflexivity

Fixed to Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

symmetry

Fixed to Anti_Symmetric

transitivity

Fixed to Transitive

‘

Relationships

Name	Type	Cardinality
contains	PhysicalLayoutOrderedPair	0..n
realizes	OrderRelation	0..n
structures	PhysicalLayout	0..n

contains

Physical layout ordered pairs describing the Value Mappings sequence.

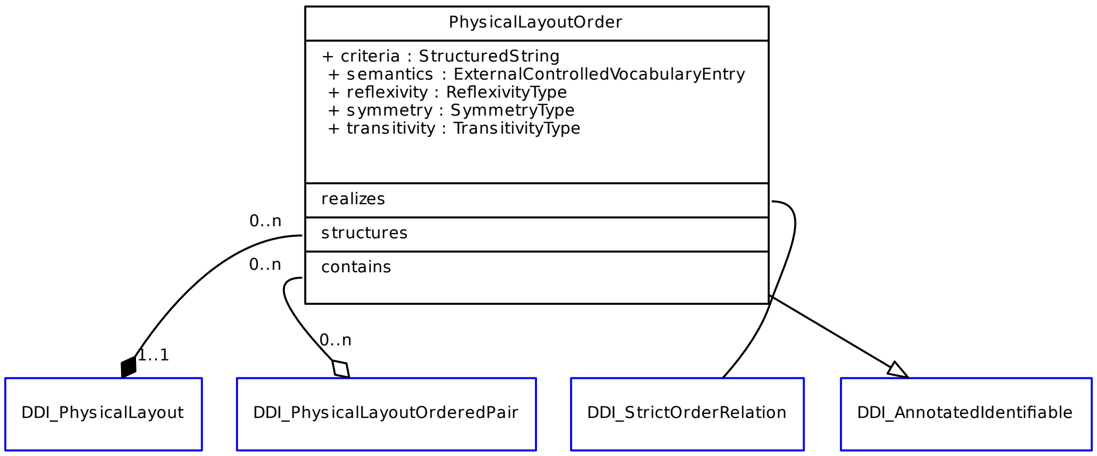
realizes

Class of the Collection pattern realized by this class.

structures

The Value Mappings in a Physical Layout to be sequenced.

Graph


PhysicalLayoutOrder
+ criteria : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
realizes
structures
contains
0..n
0..n
0..n
1..1
DDI_PhysicalLayout
DDI_PhysicalLayoutOrderedPair
DDI_StrictOrderRelation
DDI_AnnotatedIdentifiable

6.7.8 PhysicalLayoutOrderedPair

The pair of Value Mappings in a Physical Layout which are being placed in a sequence.

Extends

Identifiable

Relationships

Name	Type	Cardinality
realizes	OrderedPair	0..n
source	ValueMapping	0..n
target	ValueMapping	0..n

realizes

Class of the Collection pattern realized by this class.

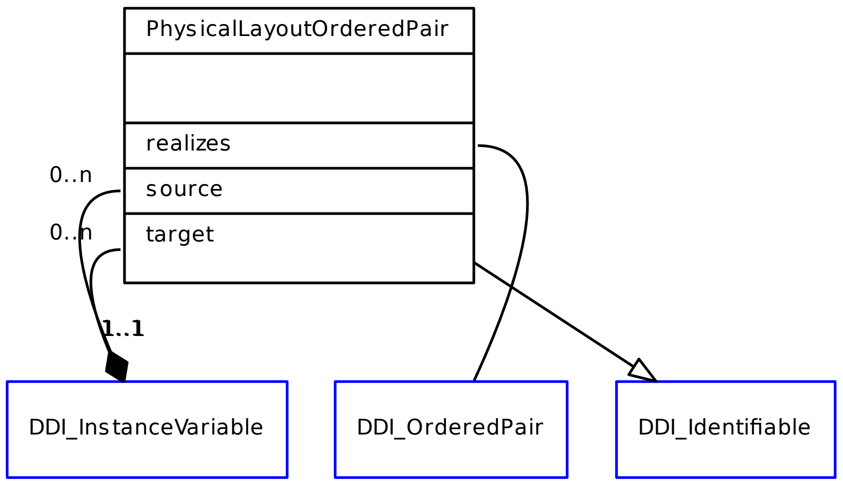
source

Specialization of source in OrderedPair for Value Mappings in a Physical Layout.

target

Specialization of target in OrderedPair for Value Mappings in a Physical Layout.

Graph



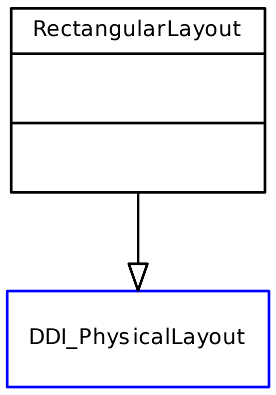
6.7.9 RectangularLayout

RectangularLayout supports the description of unit-record ("wide") data sets, where each row in the data set provides a group of values for variables all relating to a single unit. The columns will contain data relating to the values for a single variable.

Extends

:ref:'PhysicalLayout

Graph



6.7.10 StructureDescription

The information needed for understanding the physical structure of data coming from a file or other source. Multiple styles of structural description are supported: including describing files as unit-record (rectangular) files; describing cubes; and describing event-history (spell) data. A StructureDescription also points to the Data Store it physically represents. StructureDescription is reusable.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
overview	StructuredString	0..1

overview

Short natural language account of the information obtained from the combination of properties and relationships associated with an object. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
containsPhysicalLayout	PhysicalLayout	0..n
formatsDataStore	DataStore	0..n

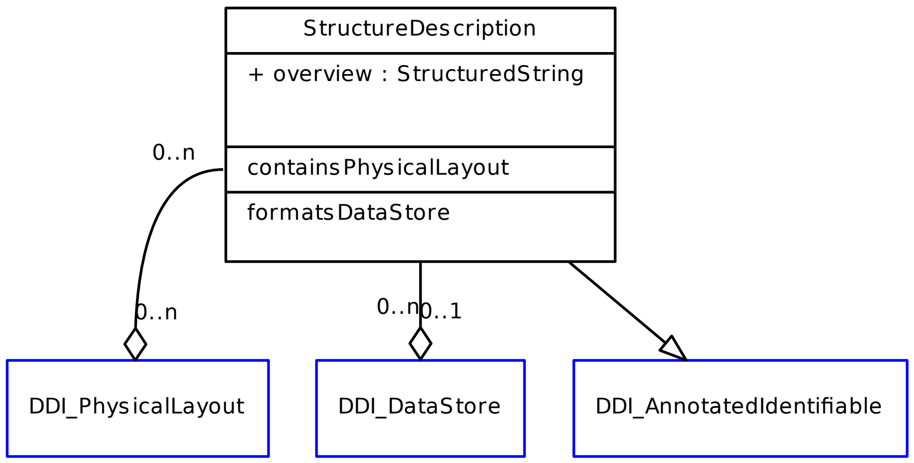
containsPhysicalLayout

Physical description (EventLayout, RectangularLayout, or CubeLayout) of the associated Logical Record Layout consisting of a Collection of Value Mappings describing the physical representation of each related Instance Variable.

formatsDataStore

Data Store physically represented by the Structure Description.

Graph



6.7.11 ValueMapping

Provides physical characteristics for an InstanceVariable as part of a PhysicalLayout

Extends

Identifiable

Properties

Name	Type	Cardinality
decimalPositions	Integer	0..1
defaultDecimalSeparator	String	0..1
defaultDigitGroupSeparator	String	0..1
defaultValue	String	0..1
format	ExternalControlledVocabularyEntry	0..1
length	Integer	0..1
maximumLength	Integer	0..1
minimumLength	Integer	0..1
nullSequence	String	0..1
numberPattern	String	0..1
physicalDatatype	ExternalControlledVocabularyEntry	0..1
required	Boolean	0..1
scale	Integer	0..1

decimalPositions

The number of decimal positions expressed as an integer. Used when the decimal position is implied (no decimal separator is present) See DDI 3.2 ManagedNumericRepresentation@decimalPositions

defaultDecimalSeparator

The string separating the integer part from the fractional part of a decimal or real number. In W3C part of the datatype format From https://www.w3.org/TR/tabular-metadata/ tabular 6.4.2 decimalChar: "A string whose value is used to represent a decimal point within the number. The default value is ".". If the supplied value is not a string, implementations MUST issue a warning and proceed as if the property had not been specified."

defaultDigitGroupSeparator

A string separating groups of digits (for readability). In W3C part of the datatype format From https://www.w3.org/TR/tabular-metadata/ tabular 6.4.2 groupChar: "A string whose value is used to group digits within the number. The default value is null. If the supplied value is not a string, implementations MUST issue a warning and proceed as if the property had not been specified."

defaultValue

A default string indicating the value to substitute for an empty string. From https://www.w3.org/TR/tabular-metadata/ Inherited 5.7 default "An atomic property holding a single string that is used to create a default value for the cell in cases where the original string value is an empty string. See Parsing Cells in [tabular-data-model] for more details. If not specified, the default for the default property is the empty string, "". The value of this property becomes the default annotation for the described column."

format

This defines the format of the physical representation of the value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 format: "An atomic property that contains either a single string or an object that defines the format of a value of this type, used when parsing a string value as described in Parsing Cells in [tabular-data-model]. The value of this property becomes the format annotation for the described datatype." See https://www.w3.org/TR/tabular-metadata/ Tabular 6.4.2 'Formats for numeric datatypes' this may include decimalChar, groupChar, pattern "By default, numeric values must be in the formats defined in [xmlschema11-2]. It is not uncommon for numbers within tabular data to be formatted for human consumption, which may involve using commas for decimal points, grouping digits in the number using commas, or adding percent signs to the number." See https://www.w3.org/TR/tabular-metadata/ Tabular 6.4. Formats for Booleans " Boolean values may be represented in many ways aside from the standard 1 and 0 or true and false." See https://www.w3.org/TR/tabular-metadata/ 6.4.4. Formats for dates and times "By default, dates and times are assumed to be in the format defined in [xmlschema11-2]. However dates and times are commonly represented in tabular data in other formats." See https://www.w3.org/TR/tabular-metadata/ 6.4.5 Formats for durations "Durations MUST be formatted and interpreted as defined in [xmlschema11-2], using the [ISO8601] format -?PnYnMnDTnHnMnS. For example, the duration P1Y1D is used for a year and a day; the duration PT2H30M for 2 hours and 30 minutes." See https://www.w3.org/TR/tabular-metadata/ 6.4.6 Formats for other types "If the datatype base is not numeric, boolean, a date/time type, or a duration type, the datatype format annotation provides a regular expression for the string values, with syntax and processing defined by [ECMASCRIPT]. If the supplied value is not a valid regular expression, implementations MUST issue a warning and proceed as if no format had been provided." From DDI3.2 ManagedNumericRepresentation@format "A format for number expressed as a string." From DDI3.2 ManagedDateTimeRepresentation_DateFieldFormat "Describes the format of the date field, in formats such as YYYY/MM or MM-DD-YY, etc. If this element is omitted, then the format is assumed to be the XML Schema format corresponding to the type attribute value."

length

The length of the physical representation of the value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 length: "A numeric atomic property that contains a single integer that is the exact length of the value. The value of this property becomes the length annotation for the described datatype. See Length Constraints in [tabular-data-model] for details." Corresponds to DDI2.5 var/location/width and DDI 3.2 PhysicalLocation/Width

maximumLength

The largest possible value of the length of the physical representation of the value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 maxLength: "A numeric atomic property that contains a single integer that is the maximum length of the value. The value of this property becomes the maximum length annotation for the described datatype. See Length Constraints in [tabular-data-model] for details."

minimumLength

The smallest possible value for the length of the physical representation of the value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 minLength: "An atomic property that contains a single integer that is the minimum length of the value. The value of this property becomes the minimum length annotation for the described datatype. See Length Constraints in [tabular-data-model] for details."

nullSequence

A string indicating a null value. From https://www.w3.org/TR/tabular-metadata/ 4.3 null — "the string or strings which cause the value of cells having string value matching any of these values to be null." From Inherited 5.7 null: "An atomic property giving the string or strings used for null values within the data. If the string value of the cell is equal to any one of these values, the cell value is null. See Parsing Cells in [tabular-data-model] for more details. If not specified, the default for the null property is the empty string "". The value of this property becomes the null annotation for the described column."

numberPattern

A pattern description of the format of a numeric value. In W3C part of the datatype format From https://www.w3.org/TR/tabular-metadata/ tabular 6.4.2 pattern: "A number format pattern as defined in [UAX35] http://www.unicode.org/reports/tr35/tr35-31/tr35-numbers.html#Number_Format_Patterns . Implementations MUST recognise number format patterns containing the symbols 0, #, the specified decimalChar (or "." if unspecified), the specified groupChar (or "," if unspecified), E, +, % and ‰. Implementations MAY additionally recognise number format patterns containing other special pattern characters defined in [UAX35]. If the supplied value is not a string, or if it contains an invalid number format pattern or uses special pattern characters that the implementation does not recognise, implementations MUST issue a warning and proceed as if the property had not been specified. f the datatype format annotation is a single string, this is interpreted in the same way as if it were an object with a pattern property whose value is that string. If the groupChar is specified, but no pattern is supplied, when parsing the string value of a cell against this format specification, implementations MUST recognise and parse numbers that consist of: an optional + or - sign, … Implementations MAY also recognise numeric values that are in any of the standard-decimal, standard-percent or standard-scientific formats listed in the Unicode Common Locale Data Repository. …"

physicalDatatype

The base datatype of the physical representation. An integer InstanceVariable might, for example, be stored as a floating point number. From https://www.w3.org/TR/tabular-metadata/ Inherited 5.7 datatype: "An atomic property that contains either a single string that is the main datatype of the values of the cell or a datatype description object. If the value of this property is a string, it MUST be the name of one of the built-in datatypes defined in section 5.11.1 Built-in Datatypes and this value is normalized to an object whose base property is the original string value. If it is an object then it describes a more specialized datatype. If a cell contains a sequence (i.e. the separator property is specified and not null) then this property specifies the datatype of each value within that sequence. See 5.11 Datatypes and Parsing Cells in [tabular-data-model] for more details. The normalized value of this property becomes the datatype annotation for the described column. "

required

If True a value is required for this variable. NOTE: this might be better at the InstanceVariable or higher in the variable cascade. From https://www.w3.org/TR/tabular-metadata/ Inherited 5.7 required: "A boolean atomic property taking a single value which indicates whether the cell value can be null. See Parsing Cells in [tabular-data-model] for more details. The default is false, which means cells can have null values. The value of this property becomes the required annotation for the described column."

scale

The scale of the number expressed as an integer (for example a number expressed in 100's, 5 = 500 would have a scale of 100). From DDI 3.2 ManagedNumericRepresentation@scale:

'

Relationships

Name	Type	Cardinality
describesSegmentLocation	PhysicalSegmentLocation	0..n
formatsInstanceVariable	InstanceVariable	0..n
realizes	Member	0..n

describesSegmentLocation

Uses a PhysicalSegmentLocation to describe where in the physical record a segment representing the InstanceVarible is. This could be, for example, described as a start position and end position value for characters in a text record via the SegmentByText extension of PhysicalSegmentLocation.

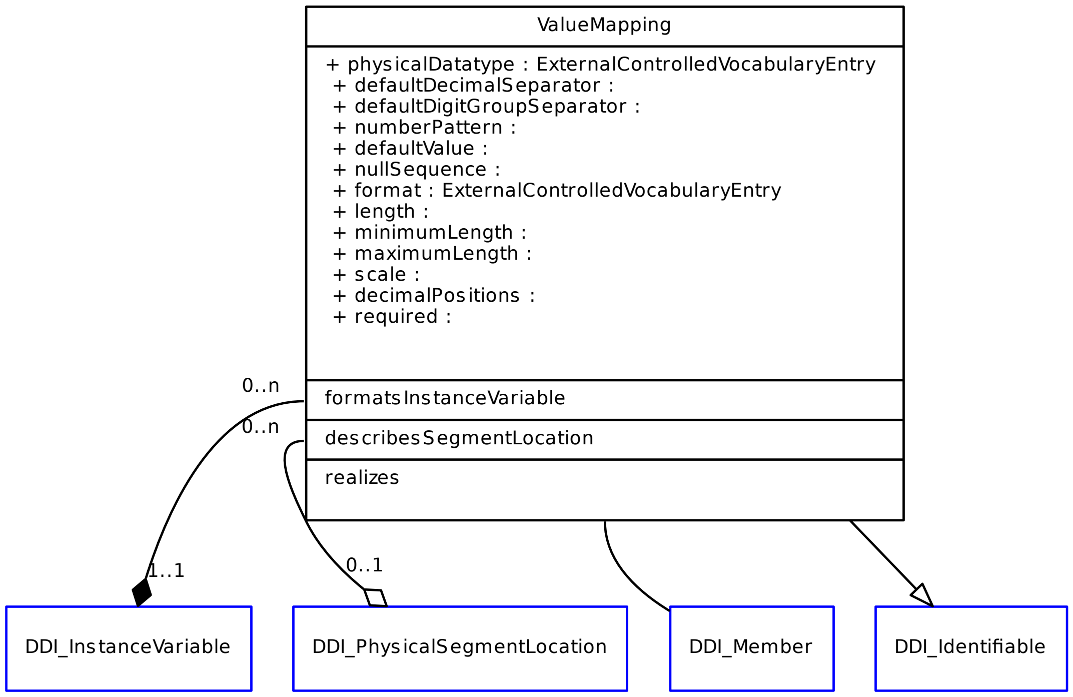
formatsInstanceVariable

Describes the physical representation of the InstanceVariable

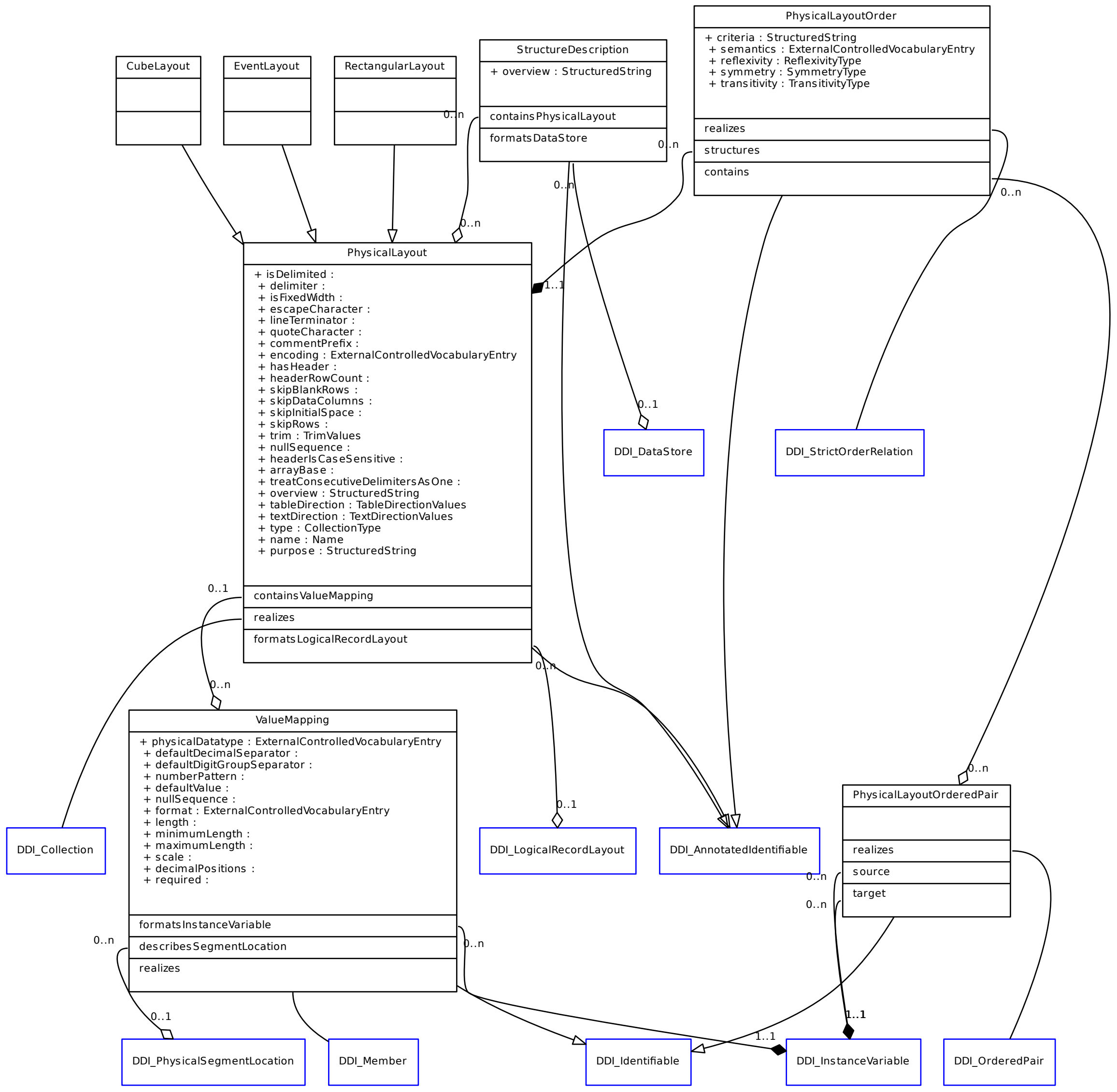
realizes

ValueMappings are members of the PhysicalLayout.

Graph



6.7.12 Graph



6.8 SignificationPattern

A package is a administrative collection of classes in DDI. These are not namespaces.

6.8.1 Sign

Something that suggests the presence or existence of a fact, condition, or quality.

Extends

Identifiable

Properties

Name	Type	Cardinality
representation	Signifier	1..1

representation

A perceivable object used to denote a signified.

‘

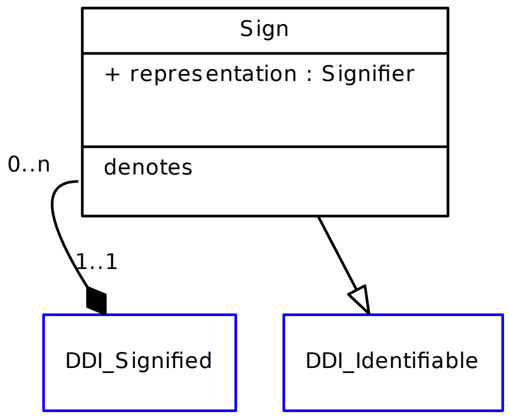
Relationships

Name	Type	Cardinality
denotes	Signified	0..n

denotes

Concept or object denoted by the signifier associated to the sign.

Graph



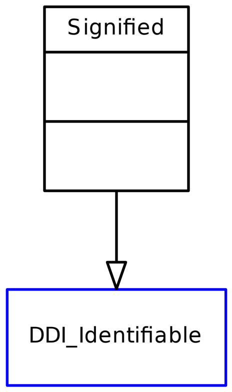
6.8.2 Signified

Concept or object denoted by the signifier associated to a sign.

Extends

:ref:‘Identifiable

Graph



6.8.3 Signifier

Concept whose extension includes perceivable objects.

‘

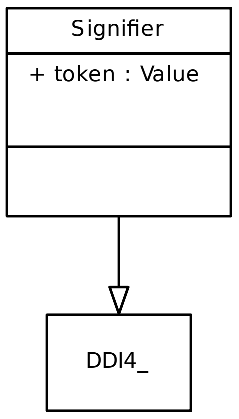
Properties

Name	Type	Cardinality
token	Value	1..1

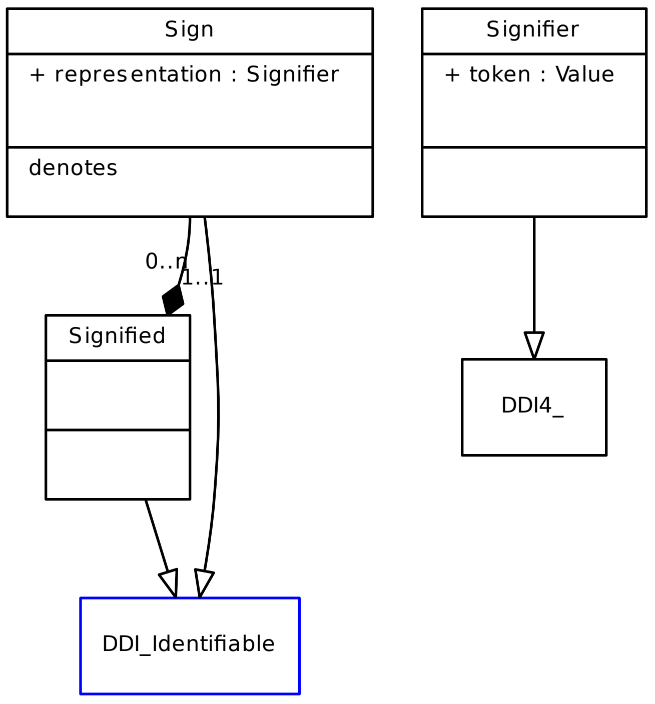
token

Perceivable object in the extension of the signifier.

Graph



6.8.4 Graph



6.9 Methodologies

A package is a administrative collection of classes in DDI. These are not namespaces.

6.9.1 AppliedUse

Links the guidance instructions to specific unit types.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
overview	StructuredString	0..1

overview

Provides a high level overview or summary of the class. Can be used to inform end-users or as part of an executive summary. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
appliesToUnitType	UnitType	0..n
hasGuide	Guide	1..n
isUsedBy	AnnotatedIdentifiable	0..n

appliesToUnitType

The unit type or types for which the guide is appropriate

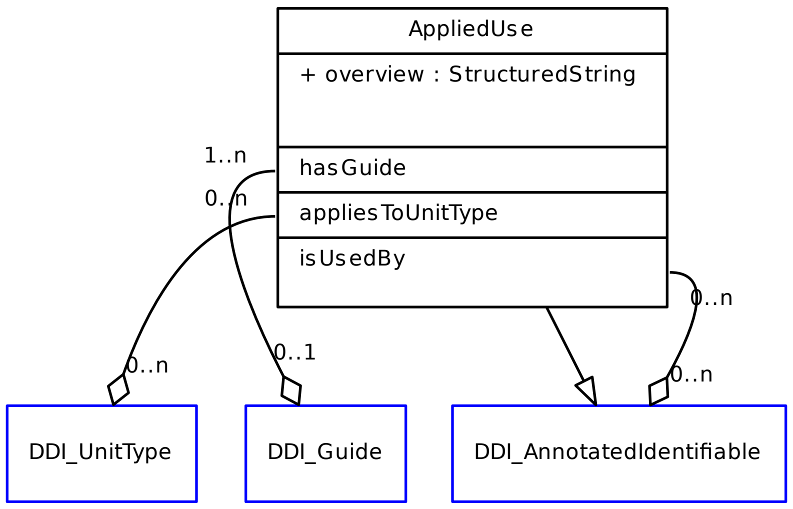
hasGuide

The guide describing the appropriate usage of the result in the context of specified unit types and/or objects.

isUsedBy

Specifies the object for which the guide is appropriate

Graph



6.9.2 BusinessFunction

Something an enterprise does, or needs to do, in order to achieve its objectives.

A Business Function delivers added value from a business point of view. It is delivered by bringing together people, processes and technology (resources), for a specific business purpose.

Business Functions answer in a generic sense "What business purpose does this Business Service or Process Step serve?" Through identifying the Business Function associated with each Business Service or Process Step it increases the documentation of the use of the associated Business Services and Process Steps, to enable future reuse.

Extends

AnnotatedIdentifiable

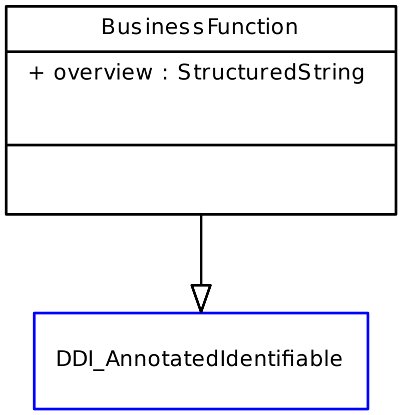
Properties

Name	Type	Cardinality
overview	StructuredString	0..1

overview

Provides a high level overview or summary of the class. Can be used to inform end-users or as part of an executive summary. Supports the use of multiple languages and structured text.

Graph



6.9.3 Goal

Goals are the "things" a method aims to achieve. A goal may be a business function (GSIM) corresponding to a function in a catalog of functions such as GSBPM or GLBMN. However, goals may be specified more broadly. For example, conducting a clinical trial might be the goal of a method. Machine learning might be the goal of a method.

Extends

BusinessFunction

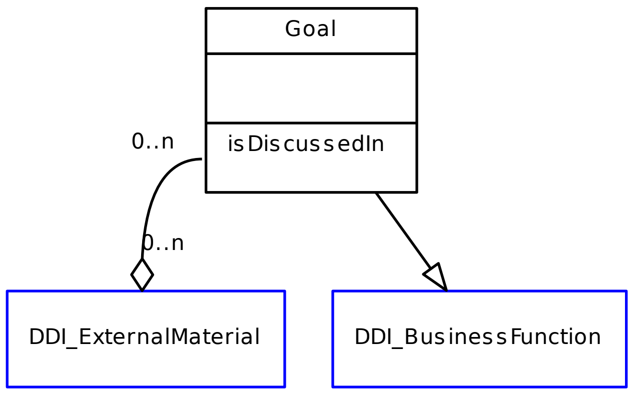
Relationships

Name	Type	Cardinality
isDiscussedIn	ExternalMaterial	0..n

isDiscussedIn

Identifies material discussing the goal. The material may be in DDI or other format.

Graph



6.9.4 Guide

Provides a guide for the usage of a result within a specified application

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
overview	StructuredString	0..1

overview

Provides a high level overview or summary of the class. Can be used to inform end-users or as part of an executive summary. Supports the use of multiple languages and structured text.

‘

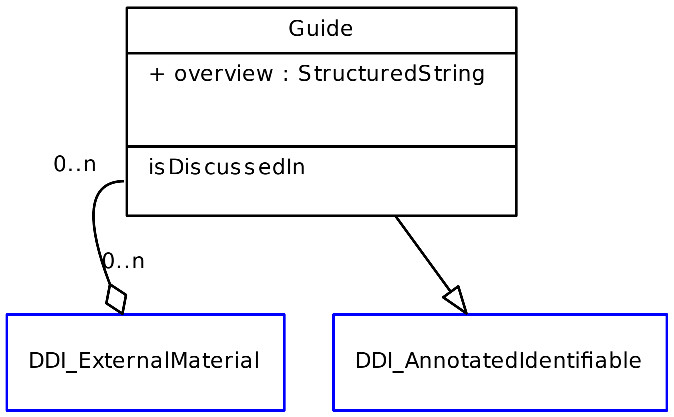
Relationships

Name	Type	Cardinality
isDiscussedIn	ExternalMaterial	0..n

isDiscussedIn

Identifies material discussing the precondition. The material may be in DDI or other format.

Graph



6.9.5 Precondition

A precondition is a state. The state includes one or more goals that were previously achieved. This state is the necessary condition before a process can begin.

Extends

BusinessFunction

Relationships

Name	Type	Cardinality
basedOnPriorResult	Result	0..n
isDiscussedIn	ExternalMaterial	0..n

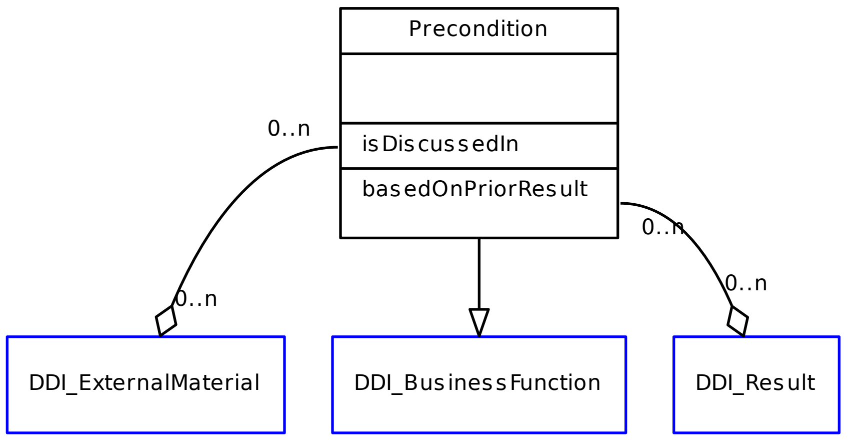
basedOnPriorResult

The Precondition may be based on the Results of earlier Processes, indicated with this relationship.

isDiscussedIn

Identifies material discussing the precondition. The material may be in DDI or other format.

Graph



6.9.6 Result

Describes the results of a process for the purpose of linking these results to guidance for future usage in specified situations. Result is abstract and serves as a substitution base for the specified result of a specific instantiation of a methodology process.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
overview	StructuredString	0..1

overview

Provides a high level overview or summary of the class. Can be used to inform end-users or as part of an executive summary. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
evaluateAgainstGoal	Goal	0..1
hasAppliedUse	AppliedUse	0..n
hasBinding	Binding	0..n
realizes	InputOutput	0..n

evaluateAgainstGoal

Comparison of Result of the Process against the Goal to inform further activities.

hasAppliedUse

The result may have one or more specific usages which provides guidance for the use of the result with a specific unit type.

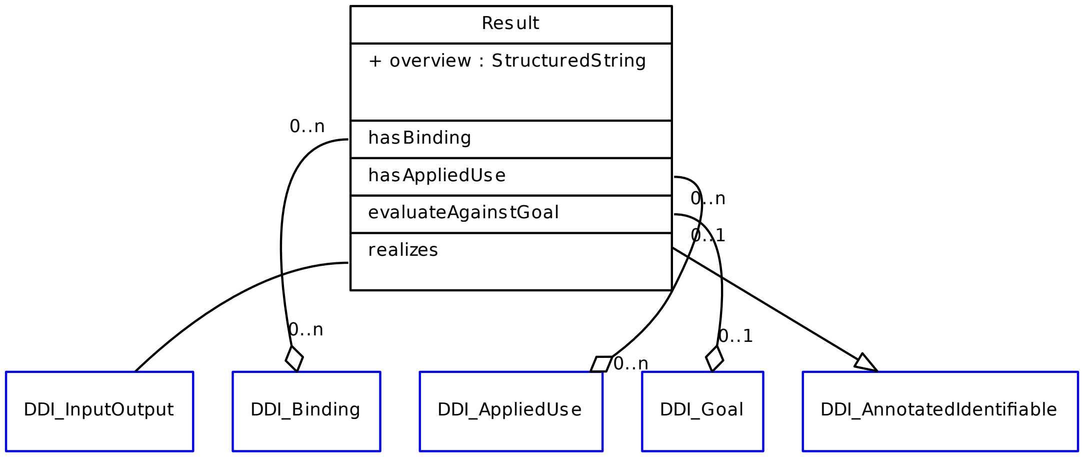
hasBinding

Defines the binding of process outputs to applied usages of these results.

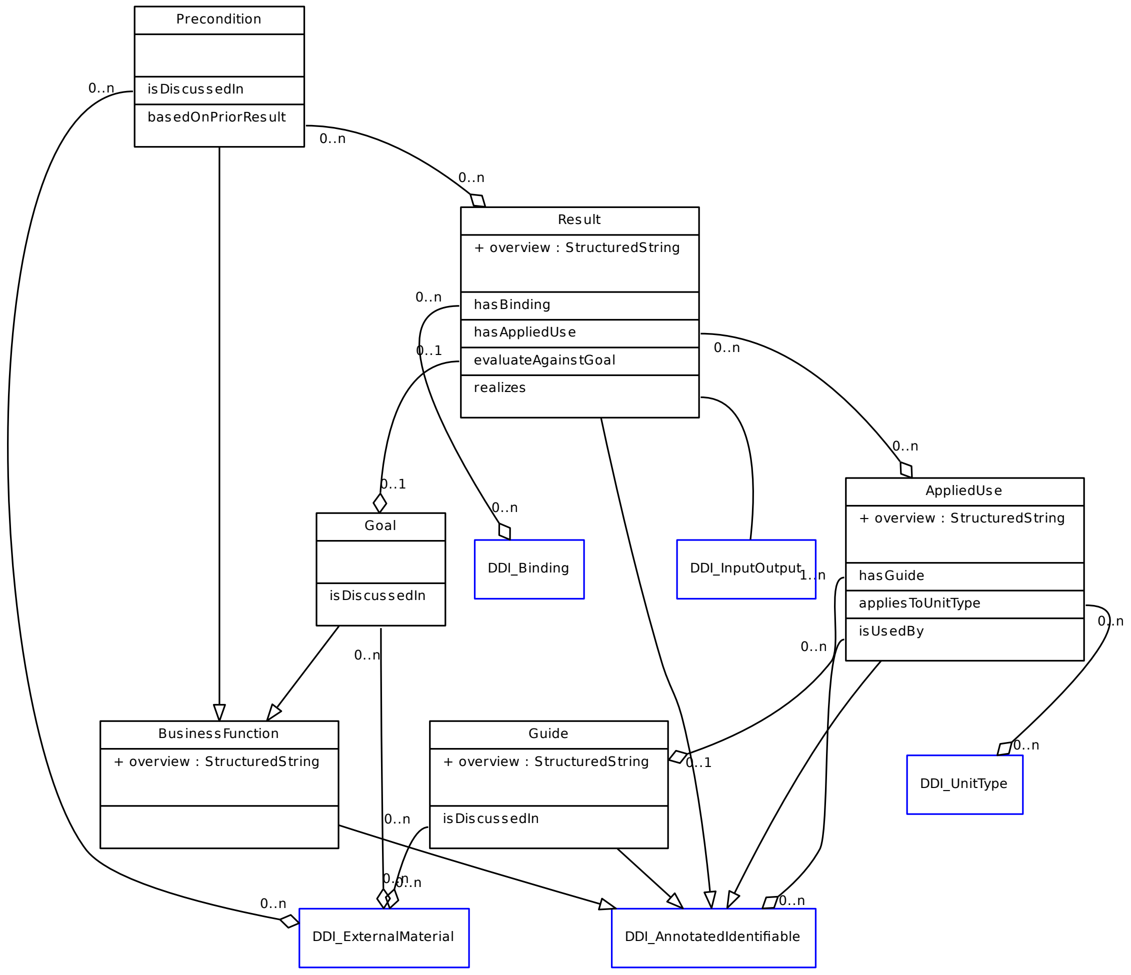
realizes

Result can serve as an input or output of a process

Graph



6.9.7 Graph



6.10 Representations

A package is a administrative collection of classes in DDI. These are not namespaces.

6.10.1 AuthorizationSource

Identifies the authorizing agency and allows for the full text of the authorization (law, regulation, or other form of authorization).

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
authorizationDate	Date	0..1
legalMandate	InternationalString	0..1
purpose	StructuredString	0..1
statementOfAuthorization	StructuredString	0..1

authorizationDate

Identifies the date of Authorization.

legalMandate

Provide a legal citation to a law authorizing the study/data collection. For example, a legal citation for a law authorizing a country's census.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

statementOfAuthorization

Text of the authorization (law, mandate, approved business case).

‘

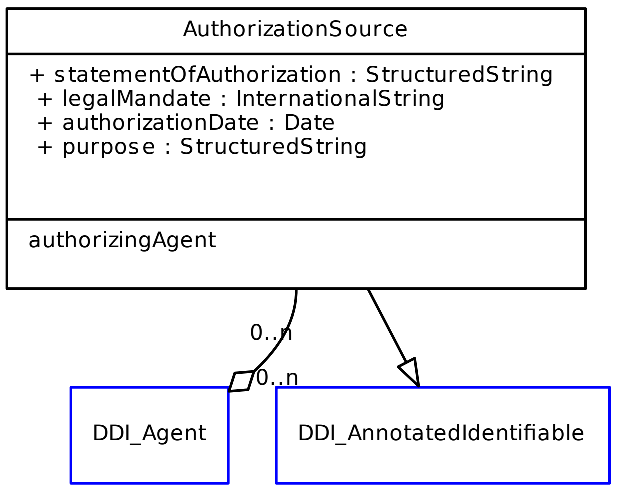
Relationships

Name	Type	Cardinality
authorizingAgent	Agent	0..n

authorizingAgent

References the authorizing agent generally described as an organization or individual

Graph


AuthorizationSource
+ statementOfAuthorization : StructuredString
+ legalMandate : InternationalString
+ authorizationDate : Date
+ purpose : StructuredString
authorizingAgent
0..n
0..n
DDI_Agent
DDI_AnnotatedIdentifiable

6.10.2 CategorySet

A Category Set is a type of Node Set which groups Categories.

Extends

NodeSet

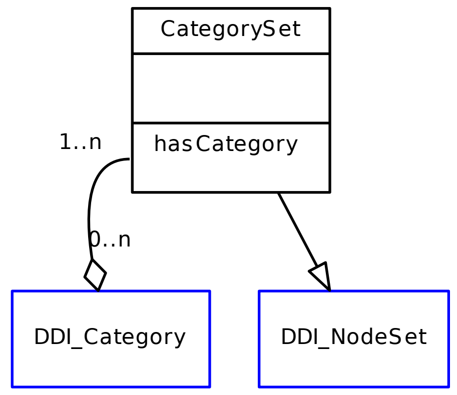
Relationships

Name	Type	Cardinality
hasCategory	Category	1..n

hasCategory

Specialization of contains in NodeSet for Categories.

Graph



6.10.3 ClassificationFamily

A Classification Family is a group of Classification Series related from a particular point of view. The Classification Family is related by being based on a common Concept (e.g. economic activity).[GSIM1.1]

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
groups	ClassificationSeries	1..n
hasClassificationIndex	ClassificationIndex	0..n
realizes	Collection	0..n

groups

Specialization of contains in Collection

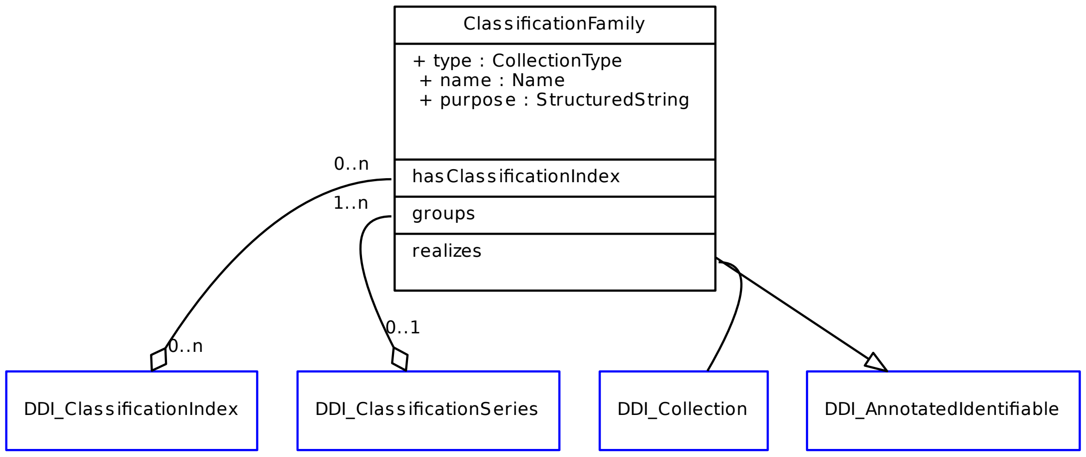
hasClassificationIndex

ClassificationIndexes associated to the ClassificationFamily.

realizes

Class of the Collection pattern realized by this class.

Graph



6.10.4 ClassificationIndex

A Classification Index is an ordered list (alphabetical, in code order etc) of Classification Index Entries. A Classification Index can relate to one particular or to several Statistical Classifications. [GSIM Statistical Classification Model]

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
availableLanguage		0..n
codingInstructions	CommandCode	0..n
contactPersons	AgentAssociation	0..n
corrections	InternationalString	0..n
maintenanceUnit	AgentAssociation	0..1
name	Name	0..n
purpose	StructuredString	0..1
releaseDate	Date	0..1
type	CollectionType	0..1

availableLanguage

A Classification Index can exist in several languages. Indicates the languages available. If a Classification Index exists in several languages, the number of entries in each language may be different, as the number of terms describing the same phenomenon can change from one language to another. However, the same phenomena should be described in each language.

codingInstructions

Additional information which drives the coding process for all entries in a Classification Index.

contactPersons

Person(s) who may be contacted for additional information about the Classification Index.

corrections

Verbal summary description of corrections, which have occurred within the Classification Index. Corrections include changing the item code associated with an Classification Index Entry.

maintenanceUnit

The unit or group of persons within the organisation responsible for the Classification Index, i.e. for adding, changing or deleting Classification Index Entries.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

releaseDate

Date when the current version of the Classification Index was released.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
groups	ClassificationIndexEntry	1..n
hasPublications	ExternalMaterial	0..n
realizes	Collection	0..n

groups

Realization of contains in Collection

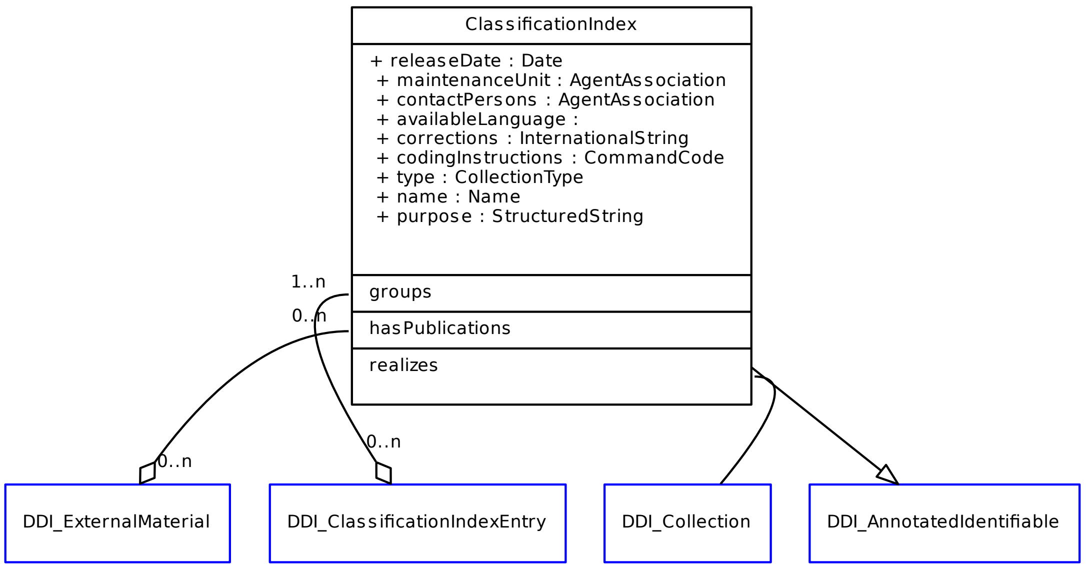
hasPublications

A list of the publications in which the Classification Index has been published.

realizes

Class of the Collection pattern realized by this class.

Graph


ClassificationIndex
+ releaseDate : Date
+ maintenanceUnit : AgentAssociation
+ contactPersons : AgentAssociation
+ availableLanguage :
+ corrections : InternationalString
+ codingInstructions : CommandCode
+ type : CollectionType
+ name : Name
+ purpose : StructuredString
groups
hasPublications
realizes
1..n
0..n
0..n
0..n
DDI_ExternalMaterial
DDI_ClassificationIndexEntry
DDI_Collection
DDI_AnnotatedIdentifiable

6.10.5 ClassificationIndexEntry

A Classification Index Entry is a word or a short text (e.g. the name of a locality, an economic activity or an occupational title) describing a type of object/unit or object property to which a Classification Item applies, together with the code of the corresponding Classification Item. Each Classification Index Entry typically refers to one item of the Statistical Classification. Although a Classification Index Entry may be associated with a Classification Item at any Level of a Statistical Classification, Classification Index Entries are normally associated with items at the lowest Level.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
codingInstructions	CommandCode	0..1
content	InternationalString	0..1
validDates	DateRange	0..1

codingInstructions

Additional information which drives the coding process. Required when coding is dependent upon one or many other factors.

content

Text describing the type of object/unit or object property.

validDates

Date from which the Classification Index Entry became valid (startDate). The date must be defined if the Classification Index Entry belongs to a floating Classification Index. Date at which the Classification Index Entry became invalid (endDate). The date must be defined if the Classification Index Entry belongs to a floating Classification Index and is no longer valid.

‘

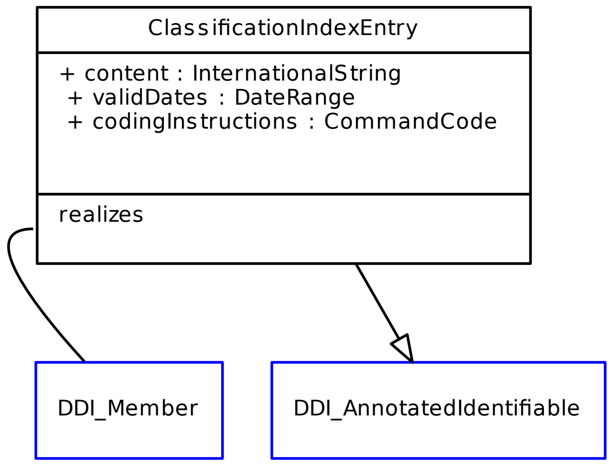
Relationships

Name	Type	Cardinality
realizes	Member	0..n

realizes

Class of the Collection pattern realized by this class.

Graph


ClassificationIndexEntry
+ content : InternationalString
+ validDates : DateRange
+ codingInstructions : CommandCode
realizes
DDI_Member
DDI_AnnotatedIdentifiable

6.10.6 ClassificationItem

A Classification Item represents a Category at a certain Level within a Statistical Classification.

Extends

Node

Properties

Name	Type	Cardinality
changeFromPreviousVersion	InternationalString	0..1
changeLog	InternationalString	0..1
explanatoryNotes	StructuredString	0..n
futureNotes	InternationalString	0..n
isGenerated	Boolean	0..1
isValid	Boolean	0..1
officialName	Name	0..1
validDates	DateRange	0..1

changeFromPreviousVersion

Describes the changes, which the item has been subject to from the previous version to the actual Statistical Classification

changeLog

Describes the changes, which the item has been subject to during the life time of the actual Statistical Classification.

explanatoryNotes

A Classification Item may be associated with explanatory notes, which further describe and clarify the contents of the Category. Explanatory notes consist of: General note: Contains either additional information about the Category, or a general description of the Category, which is not structured according to the "includes", "includes also", "excludes" pattern. Includes: Specifies the contents of the Category. Includes also: A list of borderline cases, which belong to the described Category. Excludes: A list of borderline cases, which do not belong to the described Category. Excluded cases may contain a reference to the Classification Items to which the excluded cases belong.

futureNotes

The future events describe a change (or a number of changes) related to an invalid item. These changes may e.g. have turned the now invalid item into one or several successor items. This allows the possibility to follow successors of the item in the future.

isGenerated

Indicates whether or not the item has been generated to make the level to which it belongs complete

isValid

Indicates whether or not the item is currently valid. If updates are allowed in the Statistical Classification, an item may be restricted in its validity, i.e. it may become valid or invalid after the Statistical Classification has been released.

officialName

A Classification Item has a name as provided by the owner or maintenance unit. The name describes the content of the category. The name is unique within the Statistical Classification to which the item belongs, except for categories that are identical at more than one level in a hierarchical classification

validDates

Dates for which the classification is valid. Date from which the item became valid. The date must be defined if the item belongs to a floating Statistical classification. Date at which the item became invalid. The date must be defined if the item belongs to a floating Statistical classification and is no longer valid

‘

Relationships

Name	Type	Cardinality
caseLaw	AuthorizationSource	0..n
excludes	ClassificationItem	0..n
groups	ClassificationIndexEntry	1..n

caseLaw

Case law rulings related to the Classification Item.

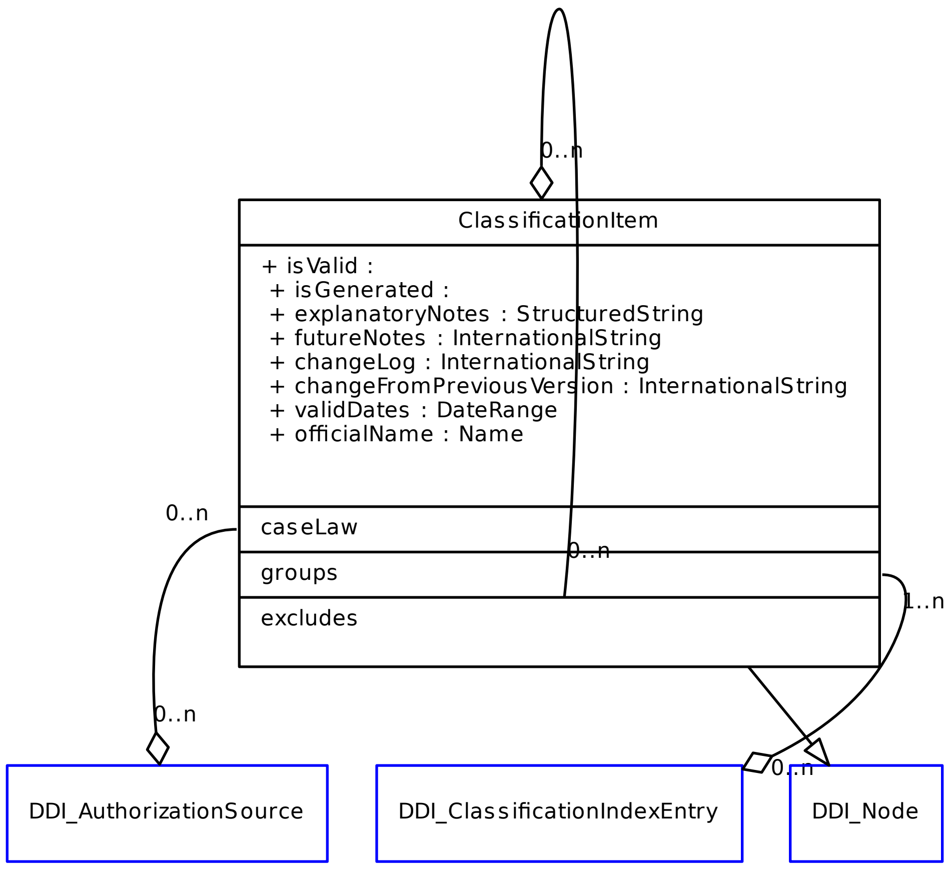
excludes

Classification Items to which the excluded cases belong (as described in explanatoryNotes).

groups

ClassificationIndexEntries related to the ClassificationItem.

Graph



ClassificationItem
+ isValid :
+ isGenerated :
+ explanatoryNotes : StructuredString
+ futureNotes : InternationalString
+ changeLog : InternationalString
+ changeFromPreviousVersion : InternationalString
+ validDates : DateRange
+ officialName : Name
caseLaw
groups
excludes
0..n
0..n
0..n
0..n
1..n
0..n
DDI_AuthorizationSource
DDI_ClassificationIndexEntry
DDI_Node

6.10.7 ClassificationSeries

A Classification Series is an ensemble of one or more Statistical Classifications, based on the same concept, and related to each other as versions or updates. Typically, these Statistical Classifications have the same name (for example, ISIC or ISCO).

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
context	StructuredString	0..1
keywords	ExternalControlledVocabularyEntry	0..n
name	Name	0..n
objectsOrUnitsClassified	ExternalControlledVocabularyEntry	0..1
owners	AgentAssociation	0..n
purpose	StructuredString	0..1
subjectAreas	ExternalControlledVocabularyEntry	0..n
type	CollectionType	0..1

context

ClassificationSeries can be designed in a specific context.

keywords

A ClassificationSeries can be associated with one or a number of keywords.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

objectsOrUnitsClassified

A ClassificationSeries is designed to classify a specific type of object/unit according to a specific attribute.

owners

The statistical office or other authority, which created and maintains the StatisticalClassification (s) related to the ClassificationSeries. A ClassificationSeries may have several owners.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

subjectAreas

Areas of statistics in which the ClassificationSeries is implemented.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
groups	StatisticalClassification	0..n
realizes	Collection	0..n

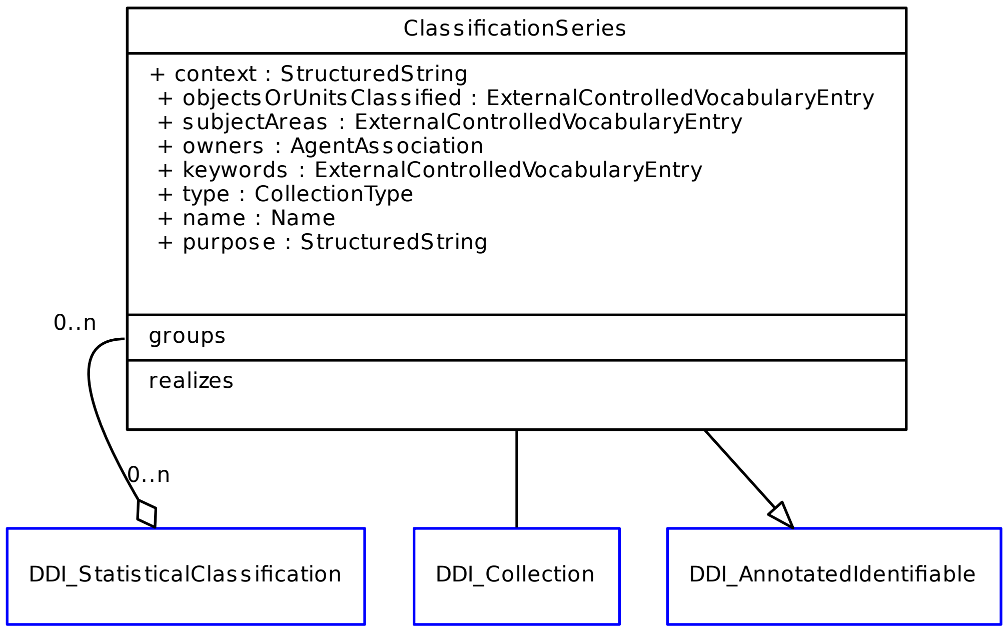
groups

Realization of contains in Collection for StatisticalClassifications.

realizes

Class of the Collection pattern realized by this class.

Graph


ClassificationSeries
+ context : StructuredString
+ objectsOrUnitsClassified : ExternalControlledVocabularyEntry
+ subjectAreas : ExternalControlledVocabularyEntry
+ owners : AgentAssociation
+ keywords : ExternalControlledVocabularyEntry
+ type : CollectionType
+ name : Name
+ purpose : StructuredString
groups
realizes
0..n
0..n
DDI_StatisticalClassification
DDI_Collection
DDI_AnnotatedIdentifiable

6.10.8 Code

A Designation for a Category.

Extends

Designation

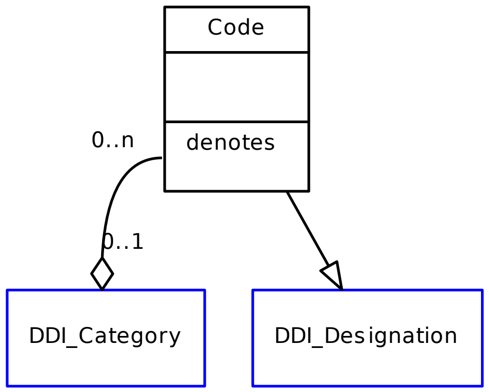
Relationships

Name	Type	Cardinality
denotes	Category	0..n

denotes

A definition for the code. Specialization of denotes for Categories.

Graph



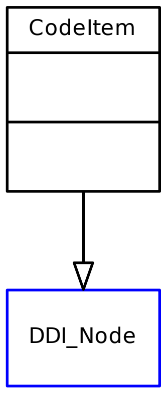
6.10.9 CodeItem

A type of Node contained in a CodeList that has a Code associated to a Category. In addition, a CodeItem can have a number of optional Designations. All Designations associated to the Code Item, including the Code, are synonyms, i.e. are associated with the same Concept.

Extends

:ref:‘Node

Graph



6.10.10 CodeList

A list of Codes and associated Categories. May be flat or hierarchical.

Extends

NodeSet

Relationships

Name	Type	Cardinality
contains	CodeItem	1..n
references	CategorySet	1..n
represents	ValueDomain	1..n

contains

Specialization of contains in NodeSet for CodeItems.

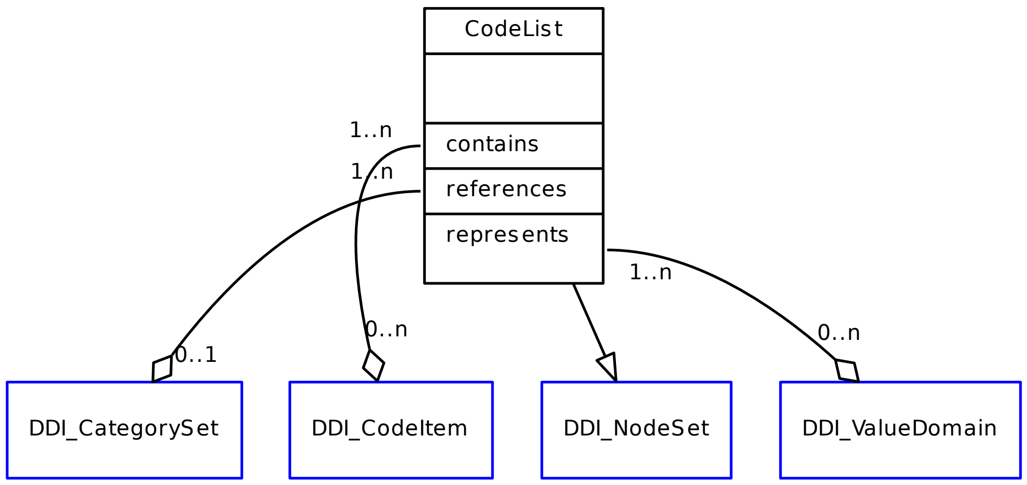
references

CategorySet associated with the CodeList.

represents

Enumerated Value Domain represented by the CodeList.

Graph



6.10.11 CorrespondenceTable

A Correspondence Table expresses a relationship between two NodeSets.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
effectiveDates	DateRange	0..1

effectiveDates

Effective period of validity of the CorrespondenceTable. The correspondence table expresses the relationships between the two NodeSets as they existed on the period specified in the table.

‘

Relationships

Name	Type	Cardinality
contactPersons	Agent	0..n
contains	Map	1..n
hasPublication	ExternalMaterial	0..n
maintenanceUnit	Agent	0..n
maps	NodeSet	0..n
owners	Agent	0..n
realizes	AsymmetricRelation	0..n
sourceLevel	Level	0..n
targetLevel	Level	0..n

contactPersons

The person(s) who may be contacted for additional information about the Correspondence Table. Can be an individual or organization.

contains

Set of mappings between nodes that participate in the correspondence.

hasPublication

A list of the publications in which the Correspondence Table has been published.

maintenanceUnit

The unit or group of persons who are responsible for the Correspondence Table, i.e. for maintaining and updating it.

maps

The NodeSet(s) from which the correspondence is made.

owners

The statistical office, other authority or section that created and maintains the Correspondence Table. A Correspondence Table may have several owners.

realizes

Class of the Collection pattern realized by this class.

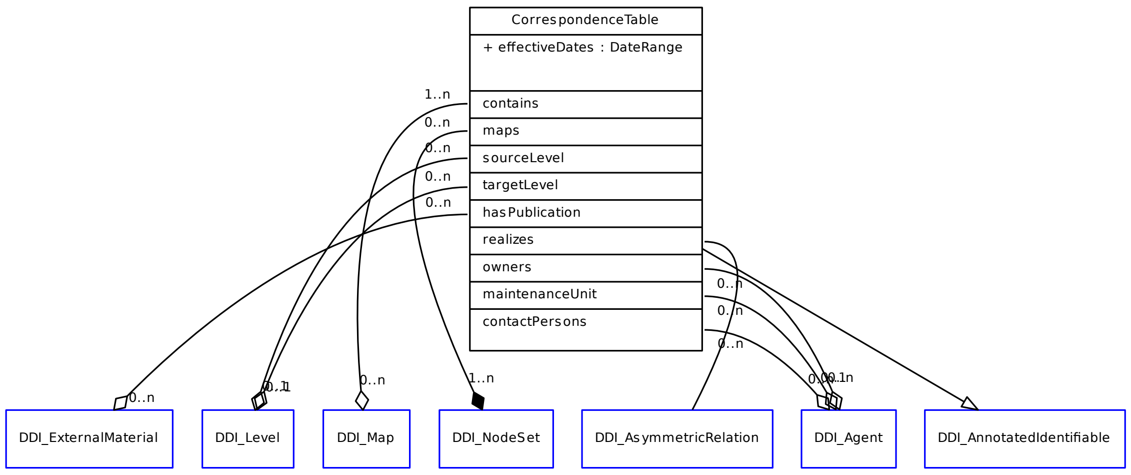
sourceLevel

Level from which the correspondence is made. Correspondences might be restricted to a certain Level in the NodeSet. In this case, target items are assigned only to source items on the given level. If no level is indicated, source items can be assigned to any level of the target NodeSet.

targetLevel

Level to which the correspondence is made. Correspondences might be restricted to a certain Level in the NodeSet. In this case, target items are assigned only to source items on the given level. If no level is indicated, target items can be assigned to any level of the source NodeSet.

Graph



6.10.12 Designation

A sign denoting a concept.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
representation	Signifier	1..1

representation

A perceivable object used to denote a signified, i.e. a concept in this case.

‘

Relationships

Name	Type	Cardinality
denotes	Concept	0..n
realizes	Sign	0..n

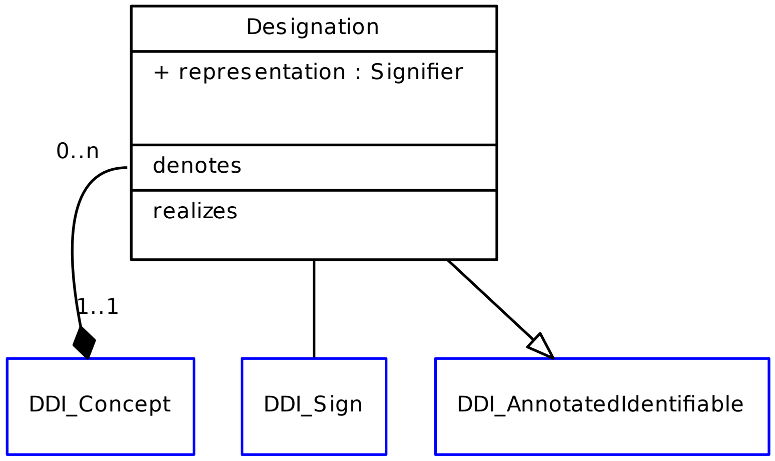
denotes

The concept denoted by the designation.

realizes

Class in the Signification Pattern realized by Designation. A Designation is a type of Sign.

Graph



6.10.13 IndexEntryPair

Indexing order, defined either by predecessor-successor pairs or by a criteria (e.g. alphabetical, in code order, etc.)

Extends

Identifiable

Relationships

Name	Type	Cardinality
followingEntry	ClassificationIndexEntry	0..1
precedingEntry	ClassificationIndexEntry	0..1
realizes	OrderedPair	0..n

followingEntry

Realization of target in OrderedPair for ClassificationEntries.

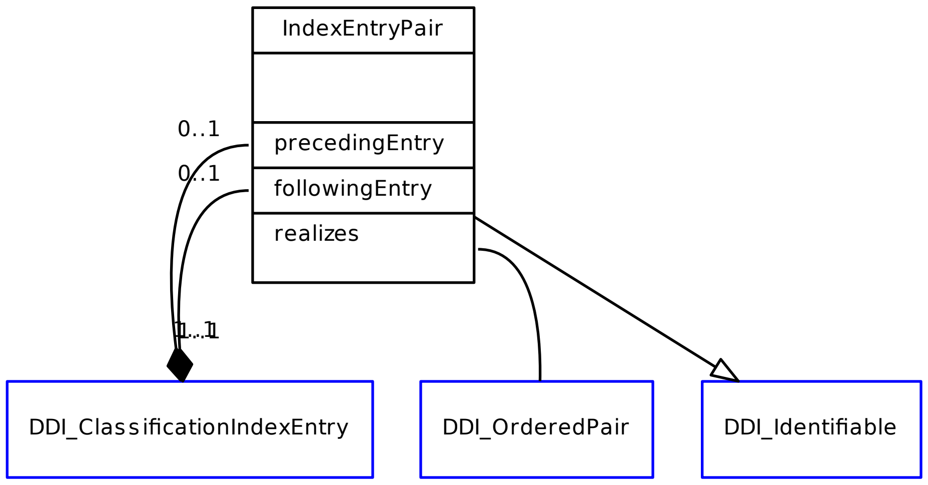
precedingEntry

Realization of source in OrderedPair for ClassificationEntries.

realizes

Class of the Collection pattern realized by this class.

Graph



6.10.14 IndexEntrySequence

Sequence of Classification Index Entries in a Classification Index.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	1..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1

reflexivity

Fixed to Anti-Reflexive

semantics

Fixed to Next

symmetry

Fixed to Anti-Symmetric

totality

Fixed to Total

transitivity

Fixed to Anti-Transitive
‘

Relationships

Name	Type	Cardinality
contains	IndexEntryPair	1..n
realizes	ImmediatePrecedenceRelation	0..n
structures	ClassificationIndex	0..n

contains

Preceding-following pair of entries in the sequence.

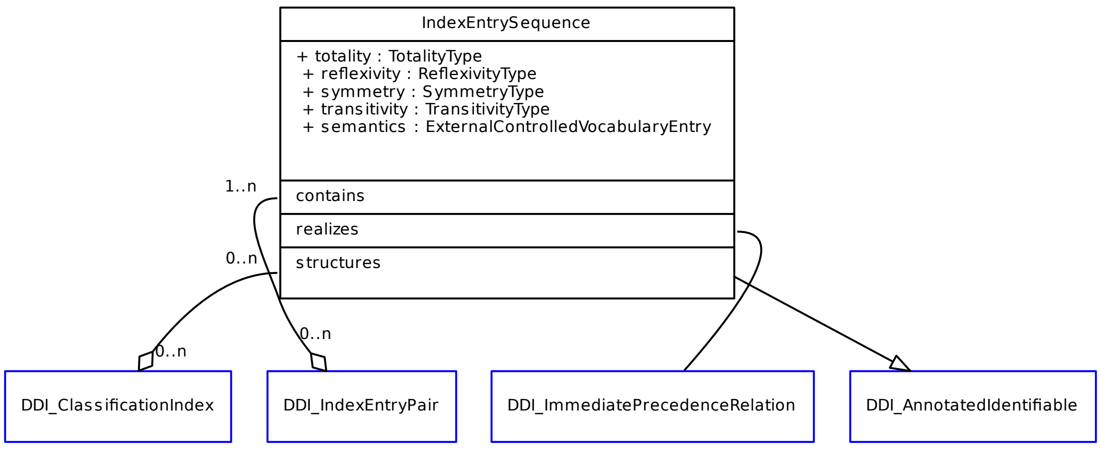
realizes

Class of the Collection pattern realized by this class.

structures

Classification Index ordered by the Index Entry Sequence.

Graph


IndexEntrySequence
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
contains
realizes
structures
1..n
0..n
0..n
0..n
DDI_ClassificationIndex
DDI_IndexEntryPair
DDI_ImmediatePrecedenceRelation
DDI_AnnotatedIdentifiable

6.10.15 Level

The Level describes the nesting structure of a hierarchical collection.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
groups	Node	1..1
isDefinedBy	Concept	0..n
realizes	Collection	0..n

groups

Realization of contains in Collection for Nodes.

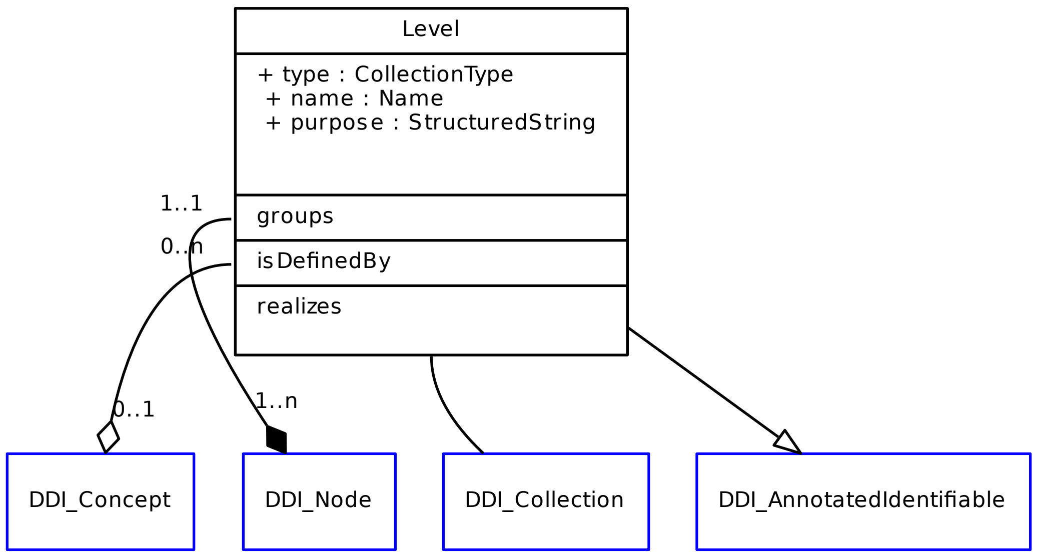
isDefinedBy

Associated concept that provides the conceptual definition of the level.

realizes

Class of the Collection pattern realized by this class.

Graph



6.10.16 Map

Relationship between Nodes in NodeSets.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..1
type	CorrespondenceType	0..1
usage	StructuredString	0..1
validDates	DateRange	0..1

displayLabel

A display label for the OrderedMemberCorrespondence. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

type

Type of correspondence in terms of commonalities and differences between two members.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

validDates

Date from which the Map became valid. The date must be defined if the Map belongs to a floating CorrespondenceTable. Date at which the Map became invalid. The date must be defined if the Map belongs to a floating Correspondence Table and is no longer valid.

‘

Relationships

Name	Type	Cardinality
realizes	OrderedTuple	0..n
source	Node	0..n
target	Node	0..n

realizes

Class of the Collection pattern realized by this class.

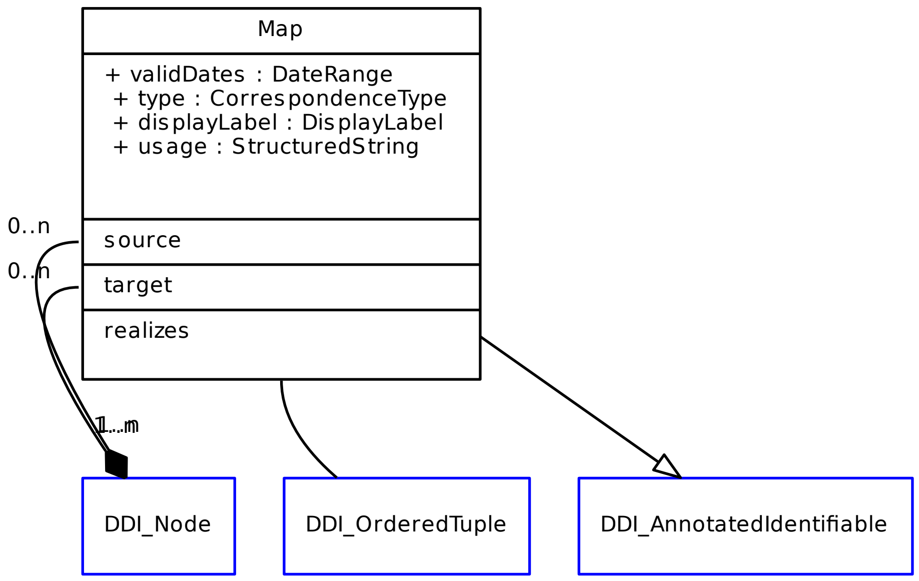
source

Realization of source in Ordered Tuple for Nodes.

target

Realization of target in Ordered Tuple for Nodes.

Graph


Map
+ validDates : DateRange
+ type : CorrespondenceType
+ displayLabel : DisplayLabel
+ usage : StructuredString
source
target
realizes
0..n
0..n
1..m
DDI_Node
DDI_OrderedTuple
DDI_AnnotatedIdentifiable

6.10.17 Node

A combination of a category and its designations.

Extends

AnnotatedIdentifiable

Relationships

Name	Type	Cardinality
contains	Designation	0..n
realizes	Member	0..n
takesMeaningFrom	Category	0..n

contains

Representation(s), i.e. sign(s), of the related Category.

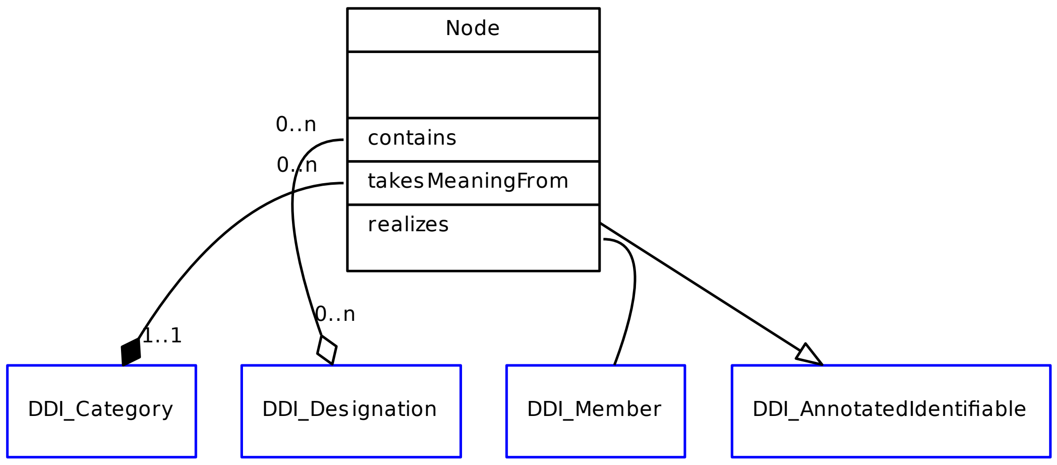
realizes

Class can play the role of a Member in a Collection

takesMeaningFrom

Category denoted by the Code. It providing meaning to the Node.

Graph



6.10.18 NodeHierarchy

Parent-child hierarchy of Nodes in a Node Set.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	1..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1

reflexivity

Fixed to Anti-Reflexive

semantics

Fixed to Parent-Of

symmetry

Fixed to Anti-Symmetric

totality

Fixed to Partial

transitivity

Fixed to Anti-Transitive
‘

Relationships

Name	Type	Cardinality
contains	NodeHierarchyPair	1..n
realizes	ImmediatePrecedenceRelation	0..n
structures	NodeSet	0..n

contains

Parent-child pair of Nodes in the hierarchy.

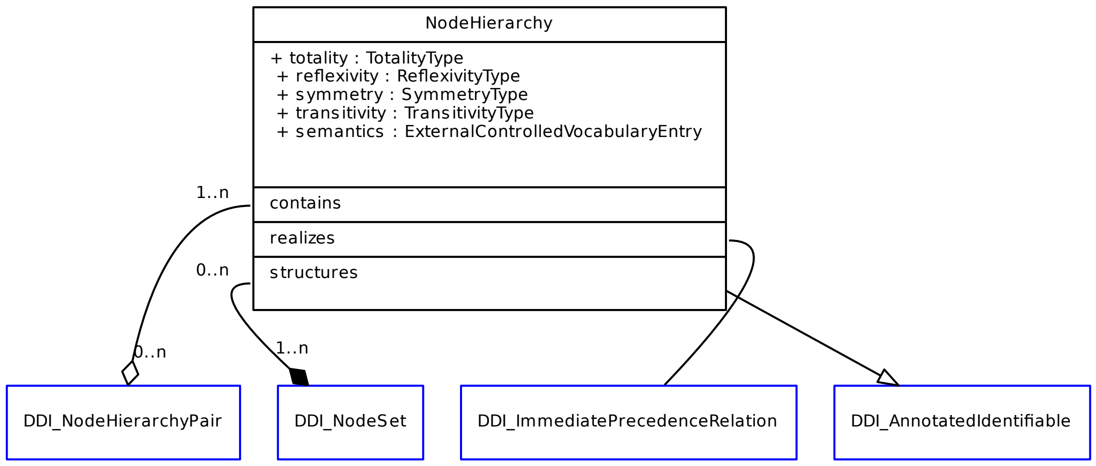
realizes

Class of the Collection pattern realized by this class.

structures

NodeSet structured by the NodeHierachy.

Graph


NodeHierarchy
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
contains
realizes
structures
1..n
0..n
0..n
1..n
DDI_NodeHierarchyPair
DDI_NodeSet
DDI_ImmediatePrecedenceRelation
DDI_AnnotatedIdentifiable

6.10.19 NodeHierarchyPair

Parent-child pair of nodes in a Node Hierarchy.

Extends

Identifiable

Relationships

Name	Type	Cardinality
child	Node	0..1
parent	Node	0..n
realizes	OrderedPair	0..n

child

Specialization of target in OrderedPair for children of Nodes.

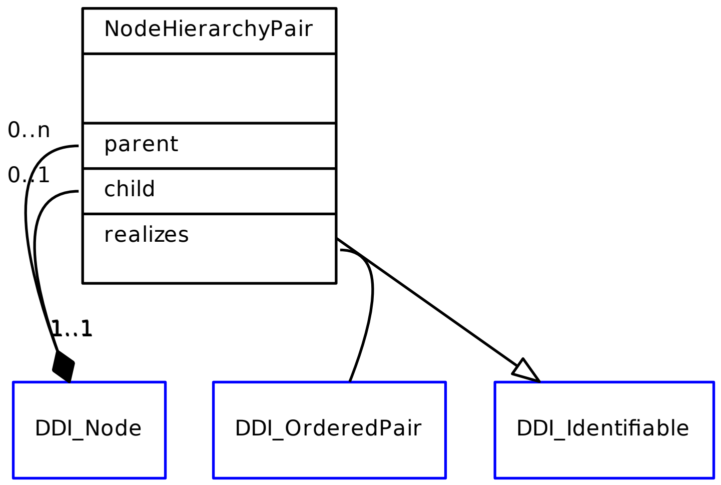
parent

Specialization of source in OrderedPair for parents of Nodes.

realizes

Class of the Collection pattern realized by this class.

Graph



6.10.20 NodePartitivePair

Part-whole pair of nodes in a Node Partitive Relation

Extends

Identifiable

Relationships

Name	Type	Cardinality
part	Node	0..1
realizes	OrderedPair	0..n
whole	Node	0..n

part

Specialization of source in OrderedPair for representing the part in the partitive relationship.

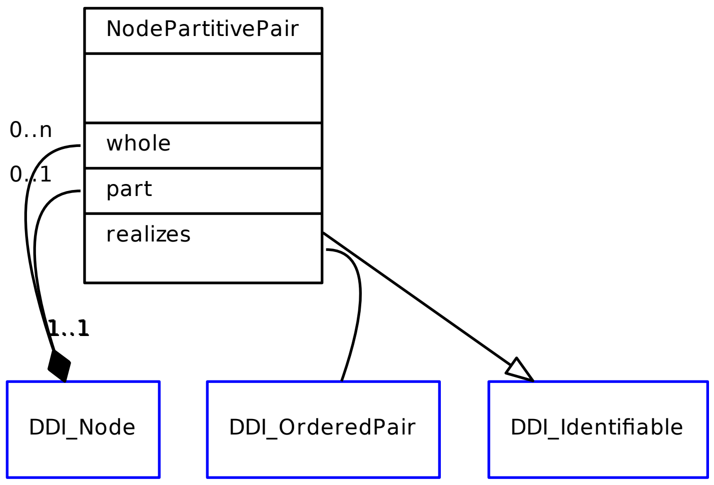
realizes

Class of the Collection pattern realized by this class.

whole

Specialization of target in OrderedPair for representing the whole in the partitive relationship.

Graph



6.10.21 NodePartitiveRelation

Part-whole relation of Nodes in a Node Set.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
displayLabel	DisplayLabel	0..n
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	1..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

criteria

Intentional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

displayLabel

A display label for the OrderRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

reflexivity

Fixed to Reflexive

semantics

Fixed to Part-Of

symmetry

Fixed to Anti-Symmetric

totality

Fixed to Partial

transitivity

Fixed to Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	NodePartitivePair	1..n
realizes	OrderRelation	0..n
structures	NodeSet	0..n

contains

Part-whole pair of nodes in the relation.

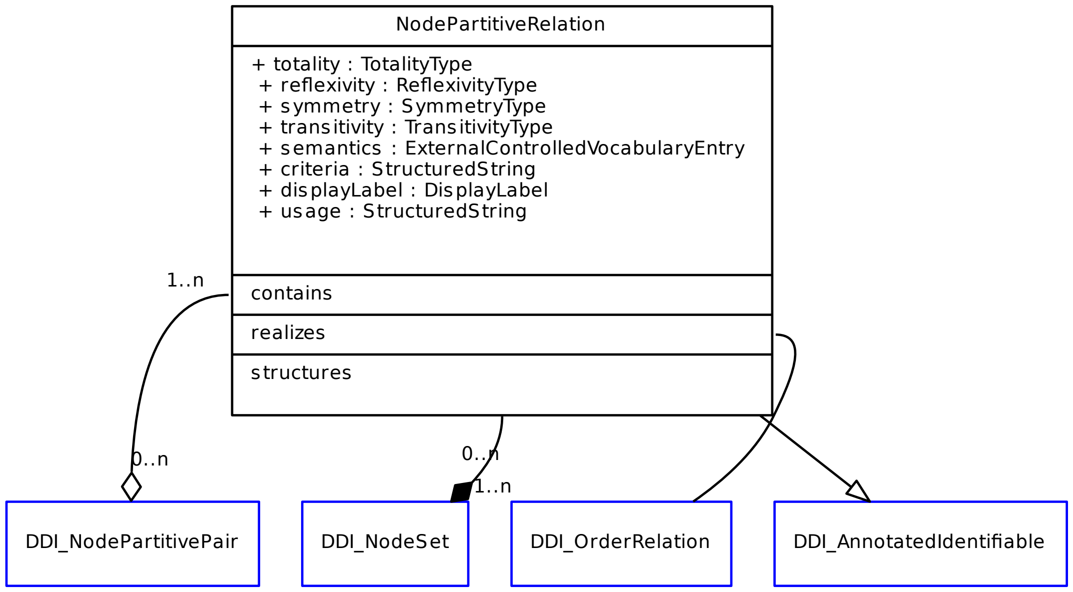
realizes

Class of the Collection pattern realized by this class.

structures

NodeSet structured by the NodePartitiveRelation.

Graph


NodePartitiveRelation
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
+ criteria : StructuredString
+ displayLabel : DisplayLabel
+ usage : StructuredString
contains
realizes
structures
1..n
0..n
0..n
1..n
DDI_NodePartitivePair
DDI_NodeSet
DDI_OrderRelation
DDI_AnnotatedIdentifiable

6.10.22 NodeSet

A NodeSet is a set of Nodes, which could be organized into a hierarchy of Levels.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
contains	Node	1..n
hasLevel	Level	1..n
isBasedOn	Concept	0..n
realizes	Collection	0..n

contains

Specialization of contains in Collection for Nodes.

hasLevel

Specialization of contains in Collection for Levels.

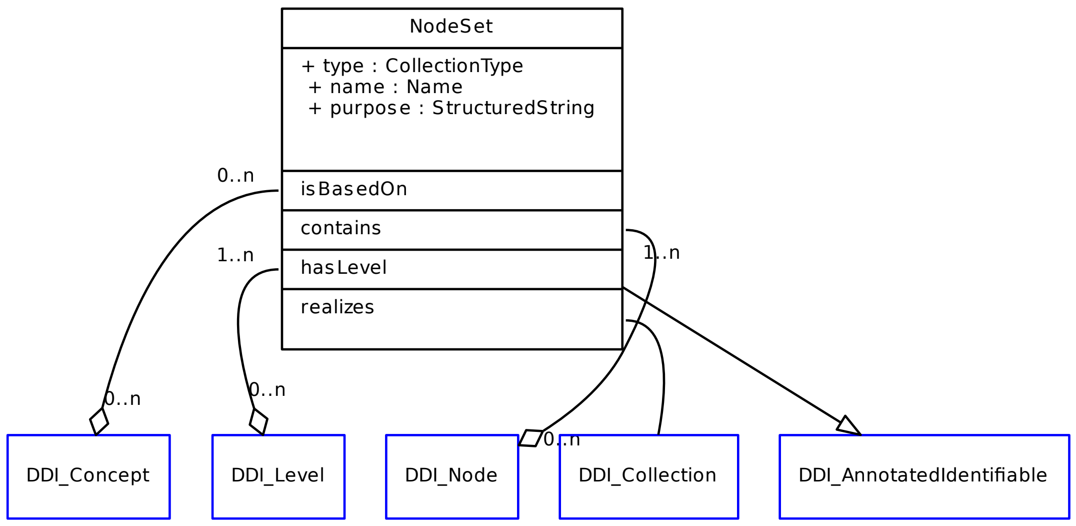
isBasedOn

Associated concept within a system.

realizes

Indicates that this class implements part of the Collection pattern

Graph



6.10.23 SentinelValueDomain

The Value Domain for a sentinel conceptual domain. Sentinel values are defined in ISO 11404 as "element of a value space that is not completely consistent with a datatype's properties and characterizing operations...". A common example would be codes for missing values.

Extends

ValueDomain

Relationships

Name	Type	Cardinality
describedValueDomain	ValueAndConceptDescription	0..n
enumeratedValueDomain	CodeList	0..n
takesConceptsFrom	SentinelConceptualDomain	0..n

describedValueDomain

A formal description of the set of valid values - for described value domains.

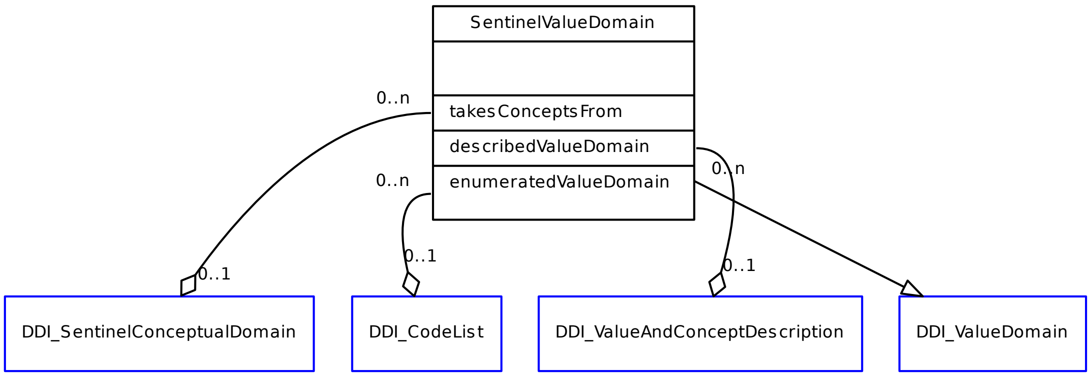
enumeratedValueDomain

A CodeList enumerating the set of valid values.

takesConceptsFrom

Corresponding conceptual definition given by a SentinelConceptualDomain.

Graph



6.10.24 StatisticalClassification

A Statistical Classification is a set of Categories which may be assigned to one or more variables registered in statistical surveys or administrative files, and used in the production and dissemination of statistics. The Categories at each Level of the classification structure must be mutually exclusive and jointly exhaustive of all objects/units in the population of interest. (Source: GSIM StatisticalClassification)

Extends

NodeSet

Properties

Name	Type	Cardinality
availableLanguage		0..n
changeFromBase	StructuredString	0..1
copyright	InternationalString	0..n
displayLabel	DisplayLabel	0..n
isCurrent	Boolean	0..1
isFloating	Boolean	0..1
purposeOfVariant	StructuredString	0..1
rationale	StructuredString	0..1
releaseDate	Date	0..1
updateChanges	StructuredString	0..n
usage	StructuredString	0..1
validDates	DateRange	0..1

availableLanguage

A list of languages in which the Statistical Classification is available. Repeat for each langauge.

changeFromBase

Describes the relationship between the variant and its base Statistical Classification, including regroupings, aggregations added and extensions. (Source: GSIM StatisticalClassification/Changes from base Statistical Classification)

copyright

Copyright of the statistical classification.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

isCurrent

Indicates if the Statistical Classification is currently valid.

isFloating

Indicates if the Statistical Classification is a floating classification. In a floating statistical classification, a validity period should be defined for all Classification Items which will allow the display of the item structure and content at different points of time. (Source: GSIM StatisticalClassification/Floating

purposeOfVariant

If the Statistical Classification is a variant, notes the specific purpose for which it was developed. (Source: GSIM StatisticalClassification/Purpose of variant)

rationale

Explanation of the reasons some decision was made or some object exists. Supports the use of multiple languages and structured text.

releaseDate

Date the Statistical Classification was released

updateChanges

Summary description of changes which have occurred since the most recent classification version or classification update came into force.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

validDates

The date the statistical classification enters production use and the date on which the Statistical Classification was superseded by a successor version or otherwise ceased to be valid. (Source: GSIM Statistical Classification)

‘

Relationships

Name	Type	Cardinality
contains	ClassificationItem	1..n
hasDistribution	ExternalMaterial	0..n
has	ClassificationIndex	1..n
isMaintainedBy	Organization	0..n
predecessor	StatisticalClassification	0..1
successor	StatisticalClassification	0..1
variantOf	StatisticalClassification	0..n

contains

Specialization of contains in NodeSet for ClassificationItems.

hasDistribution

Description and link to a publication, including print, PDF, HTML and other electronic formats, in which the Statistical Classification has been published. This is similar to dcat:Distribution.

has

ClassificationIndex(es) related to the StatisticalClassification.

isMaintainedBy

Organization, agency, or group within an agency responsible for the maintenance and upkeep of the statistical classification.

predecessor

Statistical Classification superseded by the actual Statistical Classification (for those Statistical Classifications that are versions or updates).

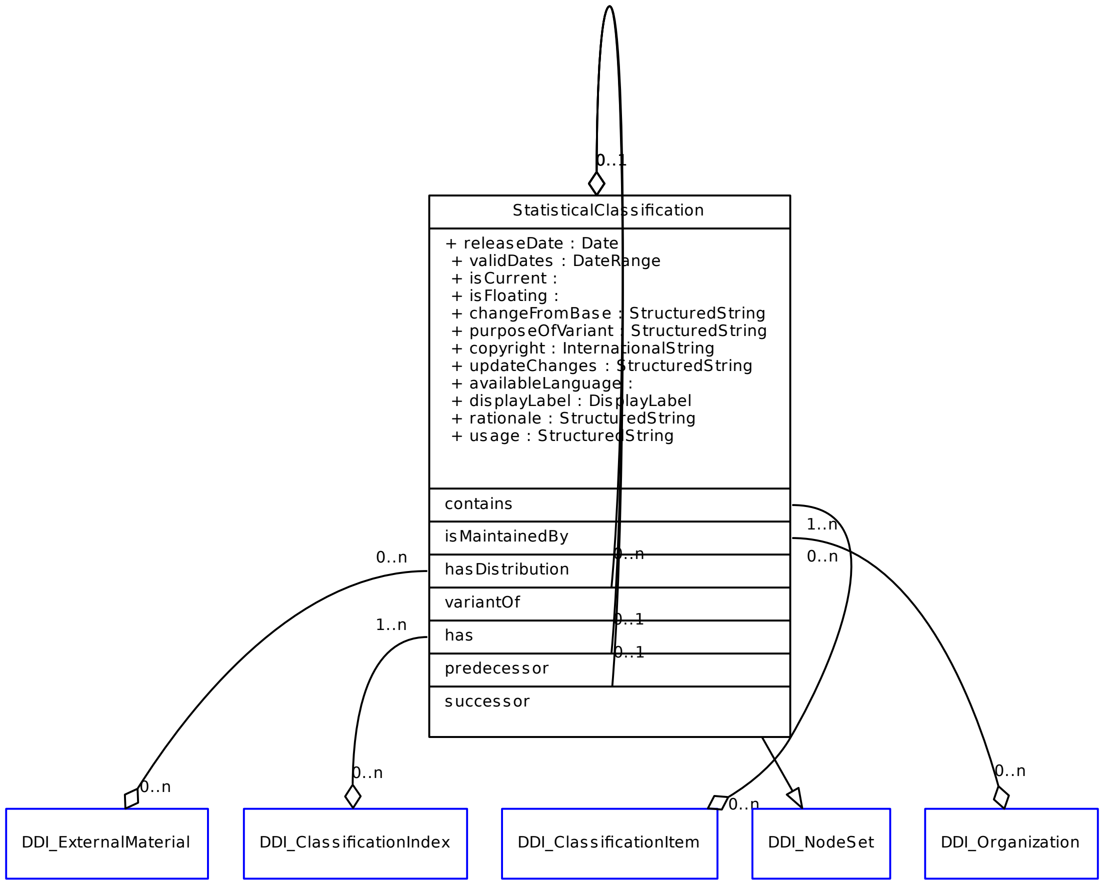
successor

Statistical Classification that supersedes the actual Statistical Classification (for those Statistical Classifications that are versions or updates),

variantOf

Statistical Classification on which the current variant is based, and any subsequent versions of that Statistical Classification to which it is also applicable.

Graph



6.10.25 SubstantiveValueDomain

The Value Domain for a substantive conceptual domain.

Extends

ValueDomain

Relationships

Name	Type	Cardinality
describedValueDomain	ValueAndConceptDescription	0..n
enumeratedValueDomain	CodeList	0..n
takesConceptsFrom	SubstantiveConceptualDomain	0..n

describedValueDomain

A formal description of the set of valid values - for described value domains.

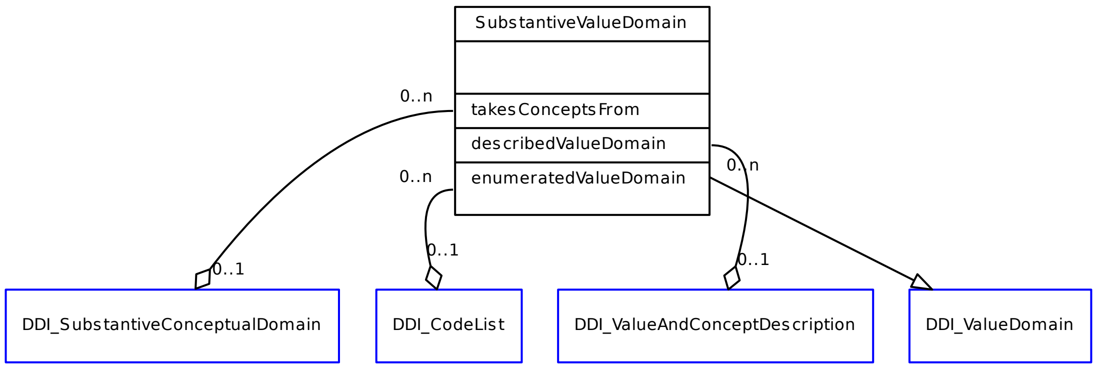
enumeratedValueDomain

A CodeList enumerating the set of valid values.

takesConceptsFrom

Corresponding conceptual definition given by an SubstantiveConceptualDomain.

Graph



6.10.26 ValueAndConceptDescription

A formal description of a set of values.

Extends

Identifier

Properties

Name	Type	Cardinality
classificationLevel	CategoryRelationCode	0..1
description	StructuredString	0..1
logicalExpression	String	0..1
maximumValueExclusive	String	0..1
maximumValueInclusive	String	0..1
minimumValueExclusive	String	0..1
minimumValueInclusive	String	0..1
regularExpression	String	0..1

classificationLevel

Indicates the type of relationship, nominal, ordinal, interval, ratio, or continuous. Use where appropriate for the representation type.

description

A formal description of the set of values.

logicalExpression

A logical expression where the values of "x" making the expression true are the members of the set of valid values. Example: x >0 describes the real numbers greater than 0

maximumValueExclusive

A string denotFrom https://www.w3.org/TR/tabular-metadata/ 5.11.2 maxExclusive: "An atomic property that contains a single number or string that is the maximum valid value (exclusive). The value of this property becomes the maximum exclusive annotation for the described datatype. See Value Constraints in [tabular-data-model] for details." DDI3.2 handles this with a Boolean isInclusive attribute. ing the maximumpossible value (excluding this value)

maximumValueInclusive

A string denoting the maximum possible value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 maximum: "An atomic property that contains a single number or string that is the maximum valid value (inclusive); equivalent to maxInclusive. The value of this property becomes the maximum annotation for the described datatype. See Value Constraints in [tabular-data-model] for details."

minimumValueExclusive

A string denoting the minimum possible value (excluding this value) From https://www.w3.org/TR/tabular-metadata/ 5.11.2 minExclusive: "An atomic property that contains a single number or string that is the minimum valid value (exclusive). The value of this property becomes the minimum exclusive annotation for the described datatype. See Value Constraints in [tabular-data-model] for details." DDI3.2 handles this with a Boolean isInclusive attribute.

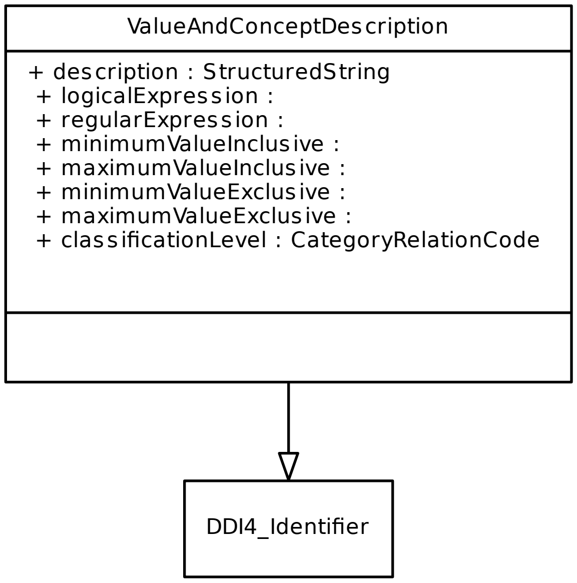
minimumValueInclusive

A string denoting the minimum possible value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 minimum: "An atomic property that contains a single number or string that is the minimum valid value (inclusive); equivalent to minInclusive. The value of this property becomes the minimum annotation for the described datatype. See Value Constraints in [tabular-data-model] for details."

regularExpression

A regular expression where strings matching the expression belong to the set of valid values.

Graph


ValueAndConceptDescription
+ description : StructuredString
+ logicalExpression :
+ regularExpression :
+ minimumValueInclusive :
+ maximumValueInclusive :
+ minimumValueExclusive :
+ maximumValueExclusive :
+ classificationLevel : CategoryRelationCode
DDI4_Identifier

6.10.27 ValueDomain

The permitted range of values for a characteristic of a variable. [GSIM 1.1]

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n
recommendedDataType	ExternalControlledVocabularyEntry	0..n

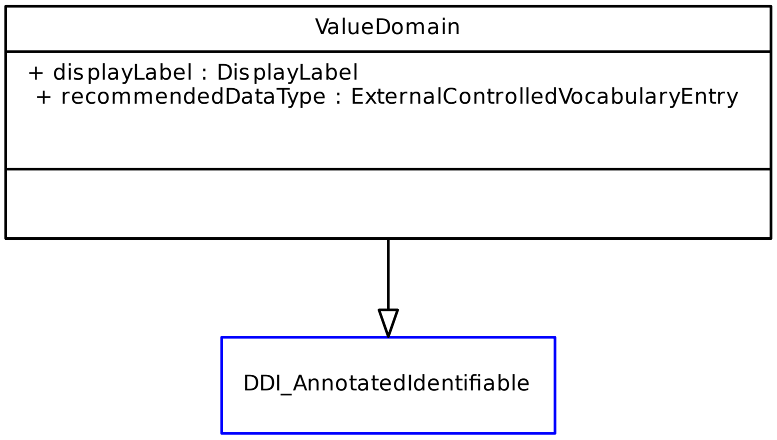
displayLabel

A display label for the object. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

recommendedDataType

The data types that are recommended for use with this domain

Graph


ValueDomain
+ displayLabel : DisplayLabel
+ recommendedDataType : ExternalControlledVocabularyEntry
DDI_AnnotatedIdentifiable

6.10.28 Vocabulary

A vocabulary is an established list of standardized terminology for use in indexing and retrieval of information.

Extends

ConceptSystem

Properties

Name	Type	Cardinality
abbreviation	InternationalString	0..n
comments	StructuredString	0..n
location	URI	0..1

abbreviation

Abbreviation of vocabulary title.

comments

Information for the user regarding the reasons for use of the vocabulary and appropriate usage constraints.

location

Location of external resource providing information about the vocabulary.

‘

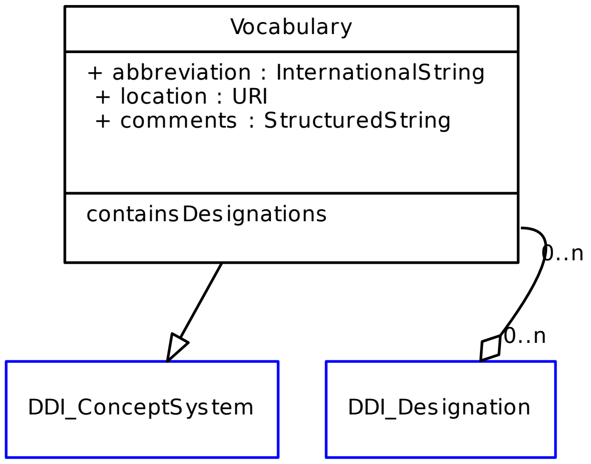
Relationships

Name	Type	Cardinality
containsDesignations	Designation	0..n

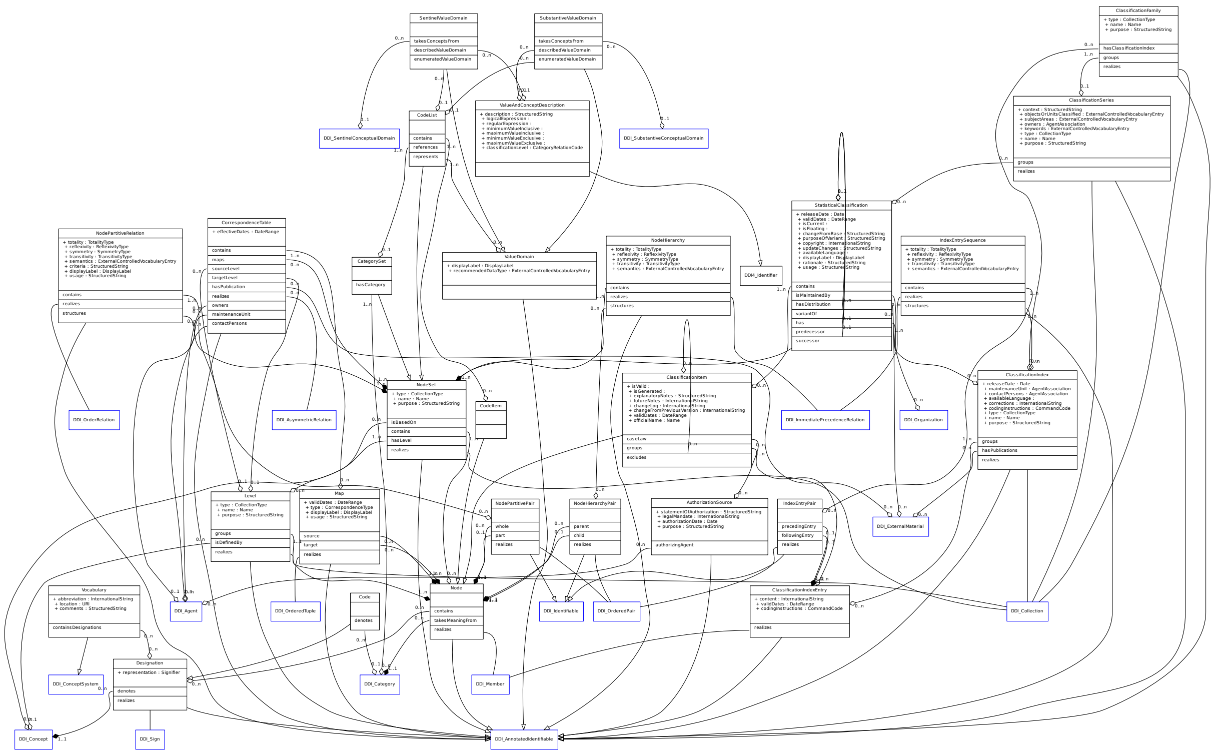
containsDesignations

Vocabularies contain Designations

Graph



6.10.29 Graph



6.11 Workflows

A package is a administrative collection of classes in DDI. These are not namespaces.

6.11.1 Act

An Act is an indivisible, atomic step, i.e. not composed of other steps. An Act can also be viewed as a terminal node in a hierarchy of Workflow Steps.

Extends

WorkflowStep

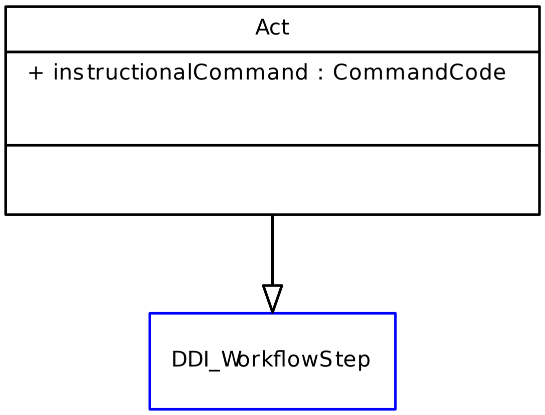
Properties

Name	Type	Cardinality
instructionalCommand	CommandCode	0..1

instructionalCommand

Optionally an act has an instructionalCommand expressed as Command Code. Acts are abstract so, in addition to an instructionalCommand, we can expect other specializations or instructions to be provided by the class which extends it.

Graph



6.11.2 Binding

Mapping between Input and Output Parameters of Workflow Steps representing the flow of information within a Workflow.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n
type	CorrespondenceType	0..1

displayLabel

A display label for the MemberCorrespondence. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

type

Type of correspondence in terms of commonalities and differences between two members.

‘

Relationships

Name	Type	Cardinality
input	InputParameter	0..n
ouput	OutputParameter	0..n
realizes	InformationFlow	0..n

input

Specialization of maps in Information Flow for Input Parameters.

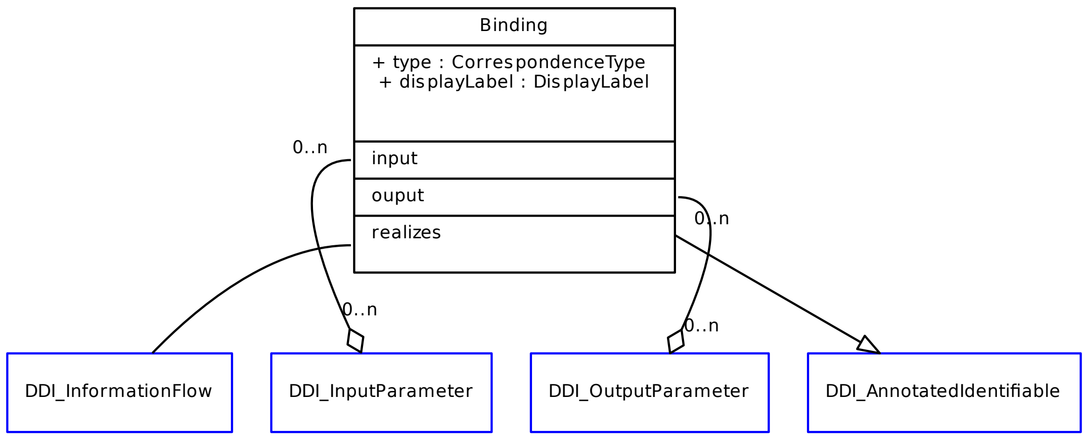
ouput

Specialization of maps in Information Flow for Output Parameters.

realizes

Class in the Process Pattern realized by Binding.

Graph



6.11.3 ConditionalControlConstruct

Type of Control Construct in which the execution flow is determined by one or more conditions.

Extends

ControlConstruct

Properties

Name	Type	Cardinality
condition	CommandCode	0..1

condition

Condition to be evaluated to determine whether or not to execute the Workflow Sequence in contains. It can be an expression and/or programmatic code. The specialized sub-classes determine whether the Sequence is executed when the condition is true or false

‘

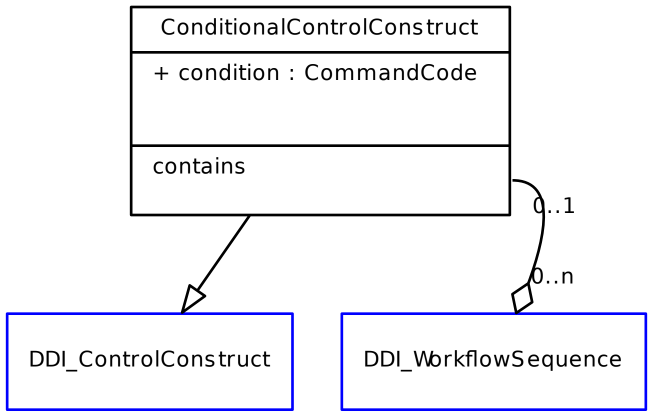
Relationships

Name	Type	Cardinality
contains	WorkflowSequence	0..1

contains

The Workflow Sequence is executed depending on the result of the condition evaluation. The specialized sub-classes determine whether the Sequence is executed when the condition is true or false. Specialization of contains in Control Construct.

Graph



6.11.4 ControlConstruct

A Workflow Step that controls the execution flow of the Workflow by determining the next Workflow Step in its scope.

Extends

WorkflowStep

Relationships

Name	Type	Cardinality
contains	WorkflowStep	0..n
defines	Binding	0..n
realizes	ProcessControlStep	0..n

contains

Workflow Steps in scope of the Control Construct. Realization of contains in Process Control Step.

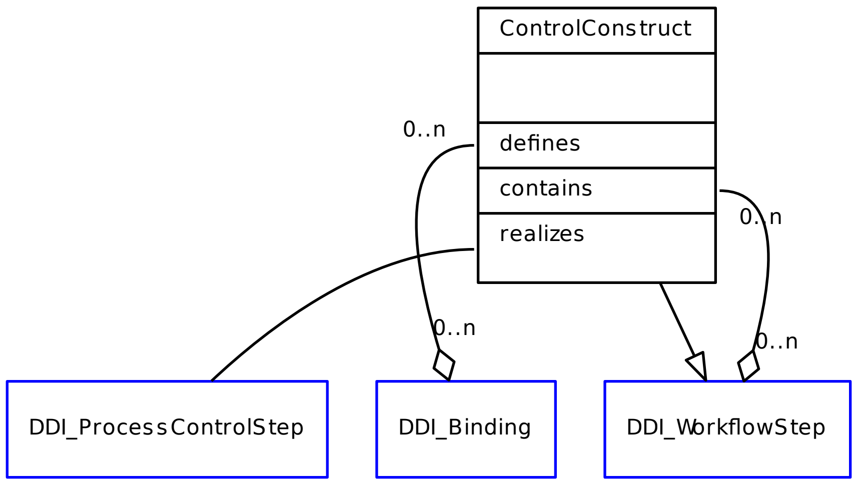
defines

Mappings between Inputs and Outputs of Workflow Steps in scope of the Control Construct. Realization of defines in Process Step.

realizes

Class in the Process Pattern realized by Control Construct.

Graph



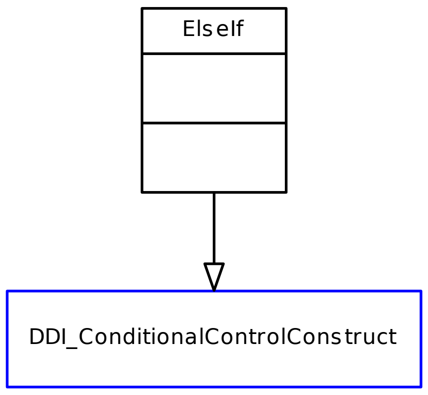
6.11.5 ElseIf

Conditional Control Construct that is evaluated if the condition in the parent IfThenElse is false. It cannot exist without a parent IfThenElse.

Extends

:ref:‘ConditionalControlConstruct

Graph



6.11.6 IfThenElse

IfThenElse describes an if-then-else decision type of control construct. If the stated condition is met, then the associated Workflow Sequence in contains (inherited from Conditional Control Construct) is triggered, otherwise the Workflow Sequence that is triggered is the one associated via elseContains.

Extends

ConditionalControlConstruct

Relationships

Name	Type	Cardinality
elseContains	WorkflowSequence	0..n
hasElseIf	ElseIf	1..1

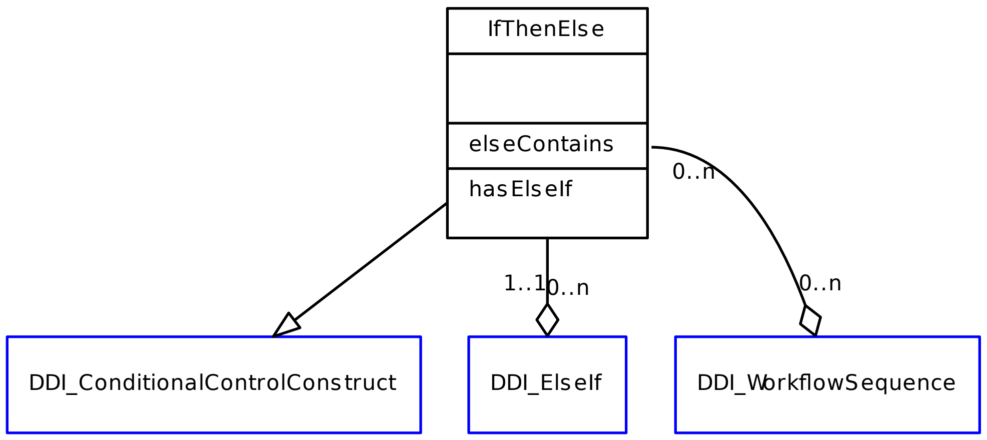
elseContains

Optional sequence of Process Steps that are triggered when the condition is false.

hasElseIf

Set of ElseIf that are evaluated if the condition is false.

Graph



6.11.7 InputParameter

Generic container for an input instance to a Process Step.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
type	ExternalControlledVocabularyEntry	0..n

type

Type of object the parameter accepts (DDI object, Datatype, etc.). If not present the parameter is untyped.

‘

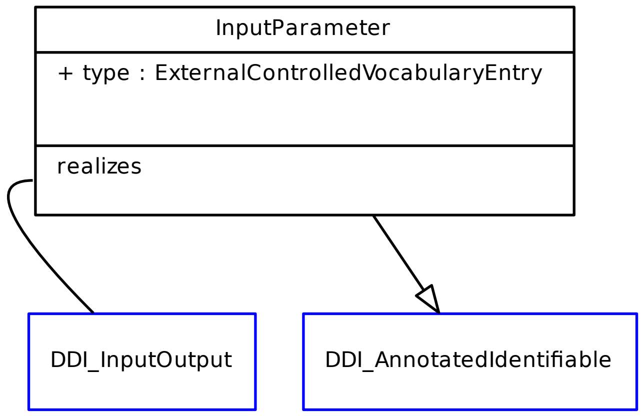
Relationships

Name	Type	Cardinality
realizes	InputOutput	0..n

realizes

Class in the Process Pattern to be realized by Input Parameter.

Graph


InputParameter
+ type : ExternalControlledVocabularyEntry
realizes
DDI_InputOutput
DDI_AnnotatedIdentifiable

6.11.8 Loop

Iterative control structure to be repeated a specified number of times based on one or more conditions. Inside the loop, one or more Workflow Steps are evaluated and processed in the order they appear.

Extends

ConditionalControlConstruct

Properties

Name	Type	Cardinality
initialValue	CommandCode	0..1
stepValue	CommandCode	0..1

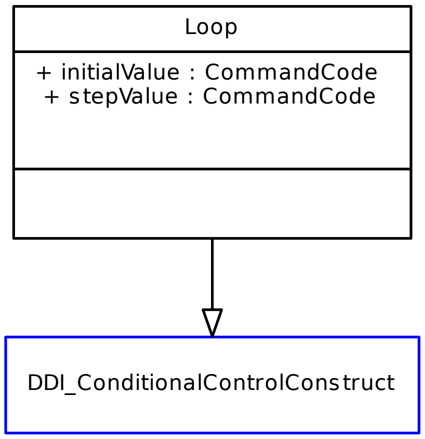
initialValue

The command used to set the initial value for the process. Could be a simple value.

stepValue

The command used to set the incremental or step value for the process. Could be a simple value.

Graph


Loop
+ initialValue : CommandCode
+ stepValue : CommandCode
DDI_ConditionalControlConstruct

6.11.9 OutputParameter

Generic container for an output instance of a Process Step.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
type	ExternalControlledVocabularyEntry	0..n

type

Type of object the parameter accepts (DDI object, Datatype, etc.). If not present the parameter is untyped.

‘

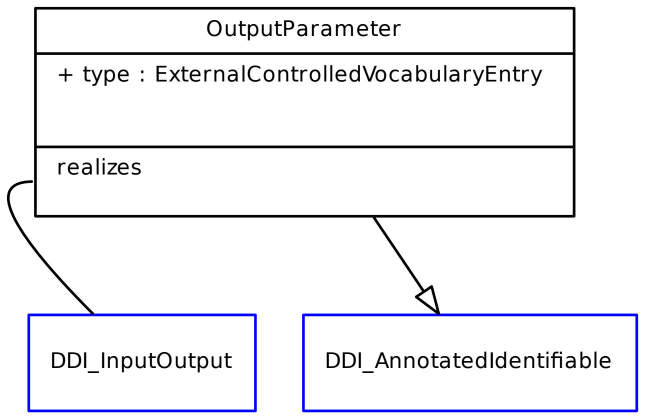
Relationships

Name	Type	Cardinality
realizes	InputOutput	0..n

realizes

Class of the Process Pattern to be realized by Output Parameter.

Graph


OutputParameter
+ type : ExternalControlledVocabularyEntry
realizes
DDI_InputOutput
DDI_AnnotatedIdentifiable

6.11.10 Parameters

Collection of Input and Output Parameters of Workflow Steps.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
type	CollectionType	0..1

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
input	InputParameter	0..n
output	OutputParameter	0..n
realizes	Interface	0..n

input

Realization of contains in Interface for Input Parameters.

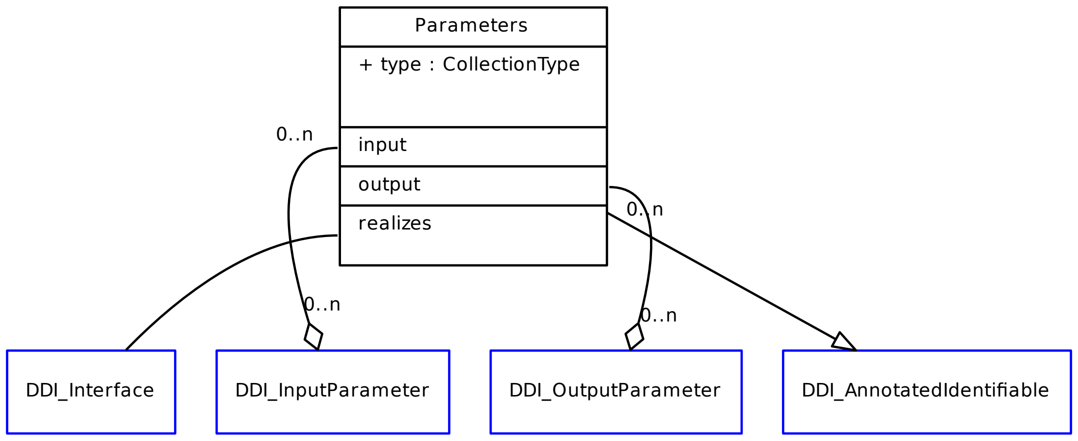
output

Realization of contains in Interface for Output Parameters.

realizes

Class in the Process Pattern realized by Parameters.

Graph



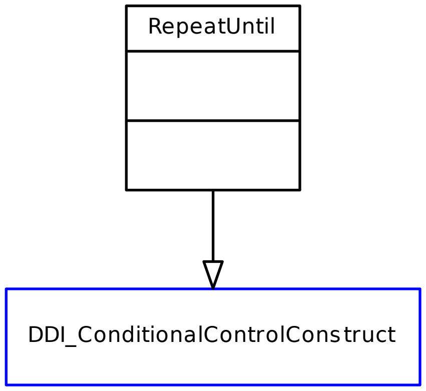
6.11.11 RepeatUntil

Iterative control structure to be repeated until a specified condition is met. Before each iteration the condition is tested. If the condition is not met, the associated Workflow Sequence in contains (inherited from Conditional Control Construct) is triggered. When the condition is met, control passes back to the containing Workflow Step.

Extends

:ref:‘ConditionalControlConstruct

Graph



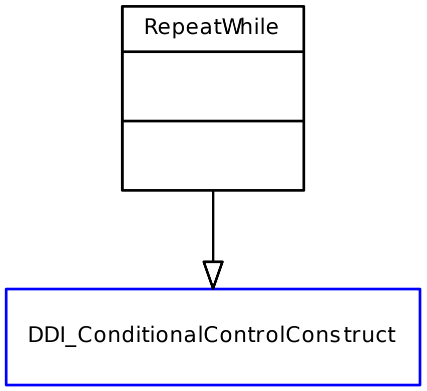
6.11.12 RepeatWhile

Iterative control structure to be repeated while a specified condition is met. Before each iteration the condition is tested. If the condition is met, the associated Workflow Sequence in contains (inherited from Conditional Control Construct) is triggered. When the condition is not met, control passes back to the containing Workflow Step.

Extends

:ref:‘ConditionalControlConstruct

Graph



6.11.13 Workflow

A Workflow is an realized Process which identifies the WorkflowSequence which contains the WorkflowSteps and their order. Describes the design, preconditions, results, algorithm, and workflow sequence of a workflow. This is an abstract class which is instantiated for specific processes.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
overview	StructuredString	0..1

overview

Provides a high level overview or summary of the class. Can be used to inform end-users or as part of an executive summary. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	WorkflowSequence	0..n
hasAlgorithm	Algorithm	0..n
hasDesign	Design	0..1
hasPrecondition	Precondition	0..n
hasResults	Result	0..n
isDiscussedIn	ExternalMaterial	0..n
realizes	Process	0..n

contains

Restriction of contains in Process to identify the Workflow Sequence

hasAlgorithm

An implementation and/or execution of an algorithm which is associated with the design of the process.

hasDesign

The design informs the specific process by providing the underlying rationale, rules and general preconditions for the process. The design is the reusable part of the implemented process.

hasPrecondition

Any condition that must be met prior to the process being implemented.

hasResults

The end result of the process which can contain a binding to specific outcome collections as well as usage instructions for implementing results

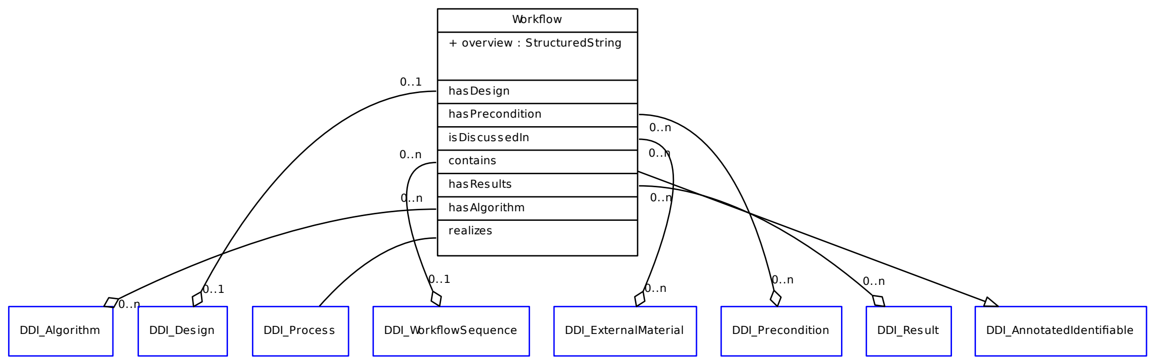
isDiscussedIn

Identifies material discussing the process. The material may be in DDI or other format.

realizes

Can play the role of a Process in a Workflow

Graph



6.11.14 WorkflowSequence

Sequencing of Workflow Steps that can be defined either by Temporal or Order Relations.

The sequence can be typed to support local processing or classification flags and alternate sequencing instructions (such as randomize for each respondent).

Extends

ControlConstruct

Properties

Name	Type	Cardinality
constructSequence	SpecificSequence	0..1
type	CollectionType	0..1
typeOfSequence	ExternalControlledVocabularyEntry	0..n

constructSequence

Describes alternate ordering for different cases using the SpecificSequence structure. If you set the sequence to anything other than order of appearance the only allowable children are QuestionConstruct or Sequence. Contents must be randomizable.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

typeOfSequence

Provides the ability to "type" a sequence for classification or processing purposes. Supports the use of an external controlled vocabulary.

'

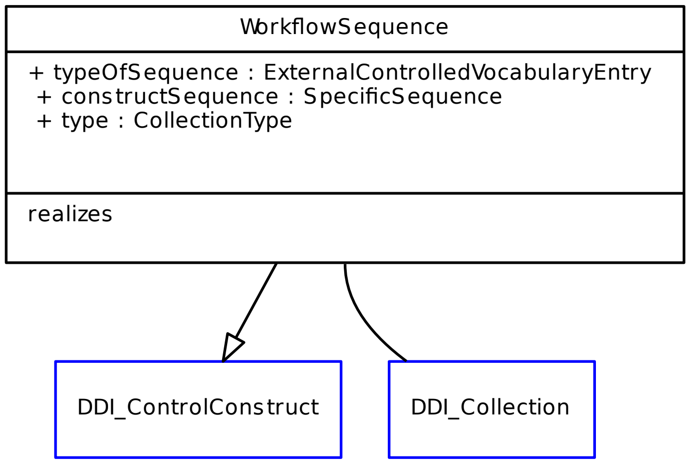
Relationships

Name	Type	Cardinality
realizes	Collection	0..n

realizes

Class in the Collections pattern realized by this class. A Workflow Sequence can be viewed as a Collection of Workflow Steps.

Graph


WorkflowSequence
+ typeOfSequence : ExternalControlledVocabularyEntry
+ constructSequence : SpecificSequence
+ type : CollectionType
realizes
DDI_ControlConstruct
DDI_Collection

6.11.15 WorkflowService

A means of performing a Workflow Step as part of a concrete implementation of a Business Function (an ability that an organization possesses, typically expressed in general and high level terms and requiring a combination of organization, people, processes and technology to achieve).

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
estimatedDuration	Date	0..1
interface	ExternalControlledVocabularyEntry	0..1
location	ExternalControlledVocabularyEntry	0..1

estimatedDuration

The estimated time period associated with the operation of the Service. This may be expressed as a time, date-time, or duration.

interface

Specifies how to communicate with the service.

location

Specifies where the service can be accessed.

‘

Relationships

Name	Type	Cardinality
hasAgent	Agent	0..n
realizes	Service	0..n

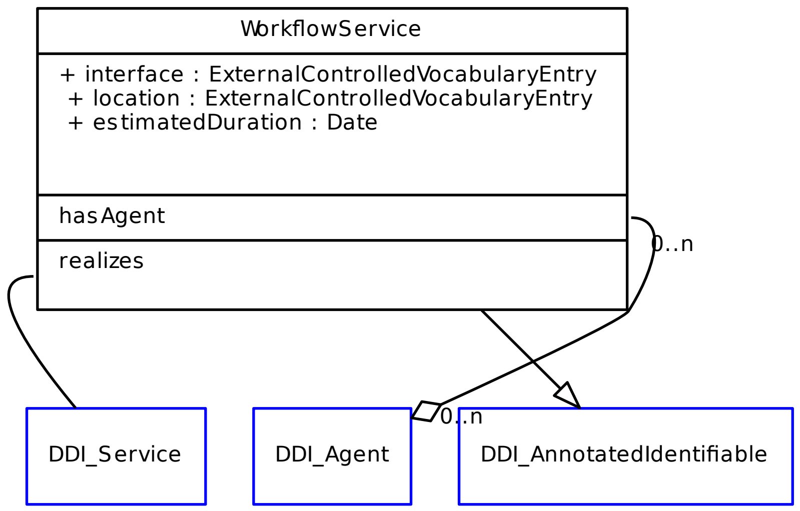
hasAgent

Actor that performs a role in the service,

realizes

Class in the Process Pattern realized by Workflow Service.

Graph


WorkflowService
+ interface : ExternalControlledVocabularyEntry
+ location : ExternalControlledVocabularyEntry
+ estimatedDuration : Date
hasAgent
realizes
0..n
0..n
DDI_Service
DDI_Agent
DDI_AnnotatedIdentifiable

6.11.16 WorkflowStep

One of the constituents of a Workflow. It can be a composition or atomic and might be performed by a Service.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n
name	Name	0..n
overview	StructuredString	0..1
purpose	StructuredString	0..1
usage	StructuredString	0..n

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

overview

Short natural language account of the information obtained from the combination of properties and relationships associated with an object. Supports the use of multiple languages and structured text.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
hasParameters	Parameters	0..n
hasProcessFramework	ExternalMaterial	0..n
isPerformedBy	WorkflowService	0..n
realizes	ProcessStep	0..n

hasParameters

Input and Output Parameters of the Workflow Step. Specialization of hasInterface in Process Step for Parameters of Workflow Steps.

hasProcessFramework

Reference to a standard process framework or step within the process famework such as GSBPM

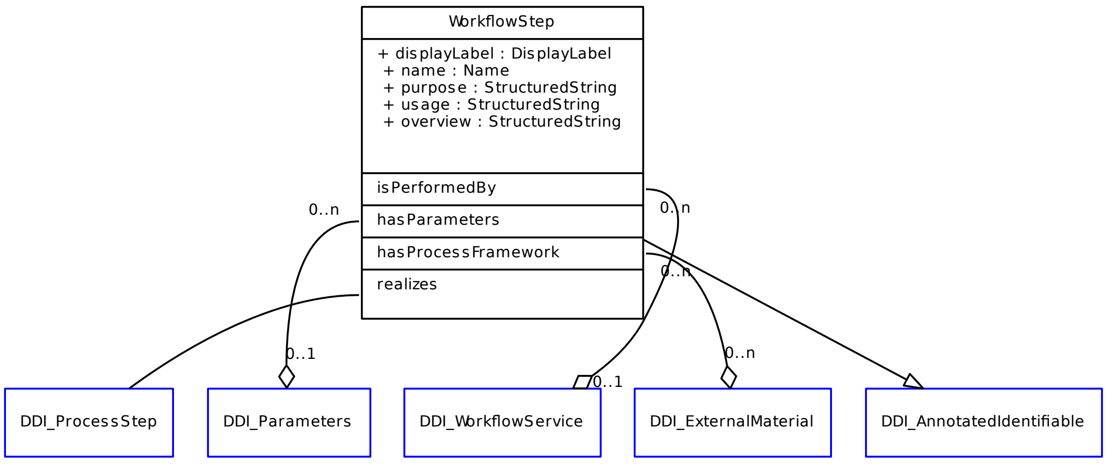
isPerformedBy

Identifies the Service Implementation which performs the Workflow Step. Specialization of isPerformedBy in Process Step for Service Implementations.

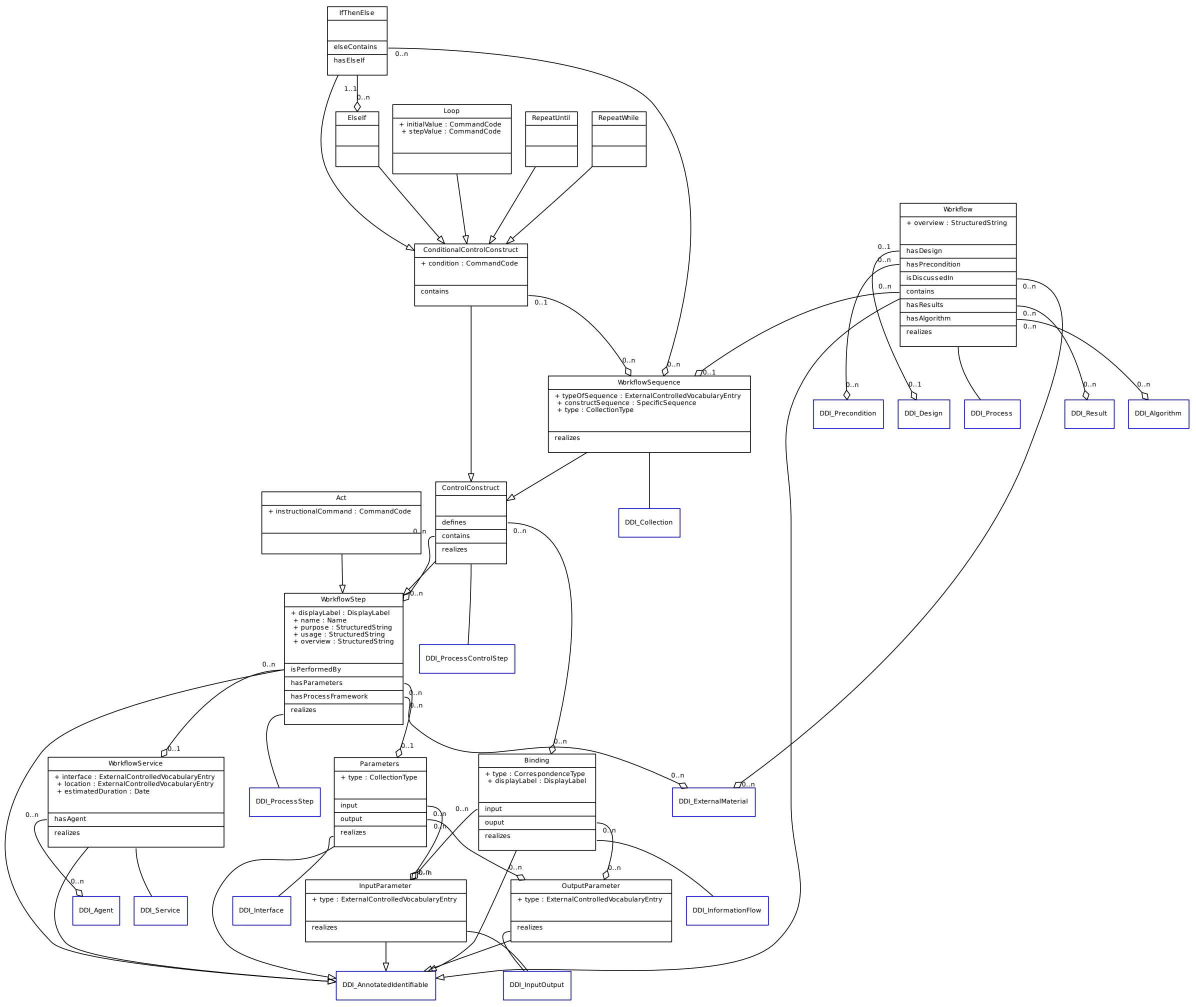
realizes

Class in the ProcessPattern realized by WorkflowStep.

Graph


WorkflowStep
+ displayLabel : DisplayLabel
+ name : Name
+ purpose : StructuredString
+ usage : StructuredString
+ overview : StructuredString
isPerformedBy
hasParameters
hasProcessFramework
realizes
0..n
0..n
0..n
0..1
0..1
0..n
DDI_ProcessStep
DDI_Parameters
DDI_WorkflowService
DDI_ExternalMaterial
DDI_AnnotatedIdentifiable

6.11.17 Graph



6.12 LogicalDataDescription

A package is a administrative collection of classes in DDI. These are not namespaces.

6.12.1 AttributeRole

A MeasureRole identifies an InstanceVariable as being an attribute within a ViewPoint.

Extends

ViewpointRole

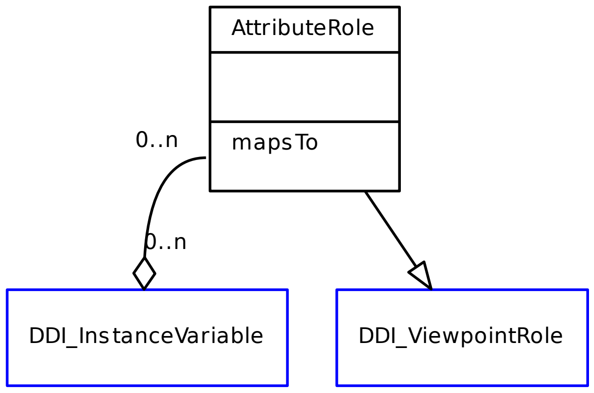
Relationships

Name	Type	Cardinality
mapsTo	InstanceVariable	0..n

mapsTo

The InstanceVariables being assigned the role of attribute

Graph



6.12.2 DataPoint

A DataPoint is a container for a Datum.

Extends

AnnotatedIdentifiable

Relationships

Name	Type	Cardinality
has	Datum	0..n
isDescribedBy	InstanceVariable	0..n
realizes	Member	0..n

has

The Datum populating the DataPoint.

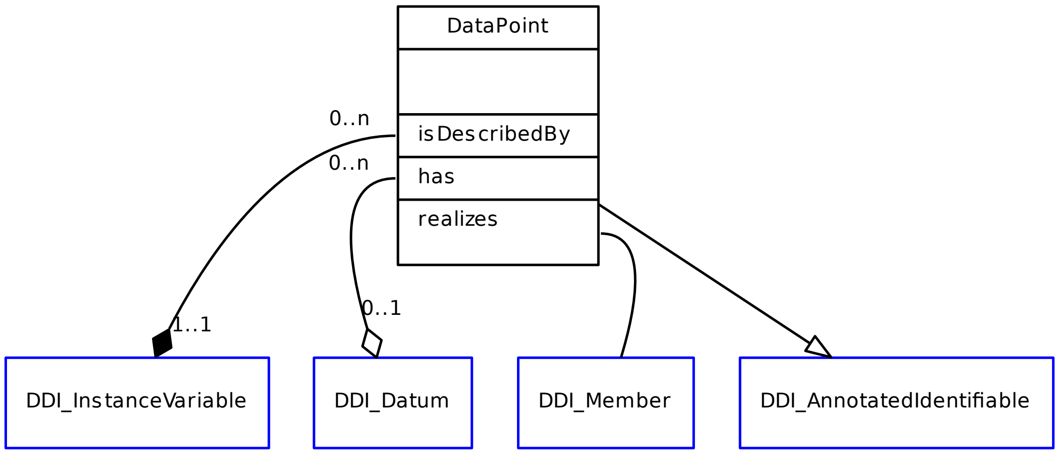
isDescribedBy

The InstanceVariable delimits the values which can populate a DataPoint.

realizes

Classes which realize the class Member fufill the role of Member e.g. they may have relations defined on them, for example one DataPoint follows another in a Record.

Graph



6.12.3 DataRecord

A Record is a Collection of DataPoints with an optional OrderRelation.

[Name of the class needs to be changed to Record].

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	1..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
contains	DataPoint	1..1
hasLogicalLayout	LogicalRecordLayout	0..n
isViewedFrom	Viewpoint	1..n
realizes	Collection	0..n

contains

The set of DataPoints composing the DataRecord

hasLogicalLayout

Structure (type) of the Data Record.

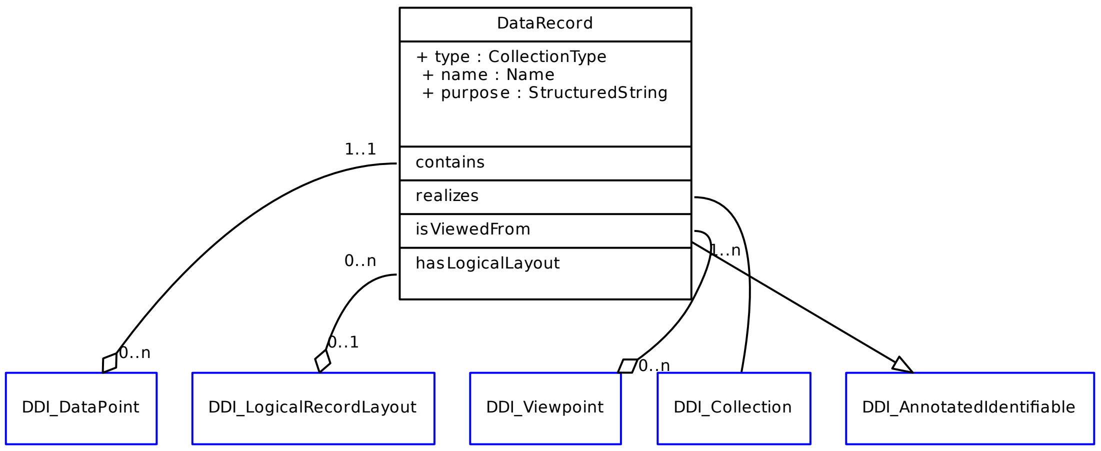
isViewedFrom

A DataRecord may have multiple ViewPoints, each of which assigns among roles identifier, measure, and attribute to the InstanceVariables associated with each DataPoint.

realizes

Allows the DataStucture to function as a Collection, enabling, for example, ordering of its components.

Graph



6.12.4 DataStore

An organized collection of data

Additional information: Using its datapoints and their processing instructions, a datastore can host questionnaire and variable datum wiring them together to form processing pipelines in all shapes and forms. Also, in line with a spreadsheet, a datastore can host multiple data structures. So a datastore can contain unit data as well as a pivot table that represents the unit data dimensionally. Likewise, the same datastore may contain both a normalized recordset in which entities are columns and its denormalized counterpart in which the rows are entities. Given processing instructions, one can losslessly be constructed from the other.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
characterSet	String	0..1
dataStoreType	ExternalControlledVocabularyEntry	0..1
name	Name	0..n
purpose	StructuredString	0..1
recordCount	Integer	0..1
type	CollectionType	0..1

characterSet

Default character set used in the Data Store.

dataStoreType

The type of DataStore. Could be delimited file, fixed record length file, relational database, etc. Points to an external definition which can be part of a controlled vocabulary maintained by the DDI Alliance.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

recordCount

The number of records in the Data Store.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
contains	DataRecord	0..1
realizes	Collection	0..n
references	LogicalRecordLayout	0..1

contains

Data in the form of Data Records contained in the Data Store.

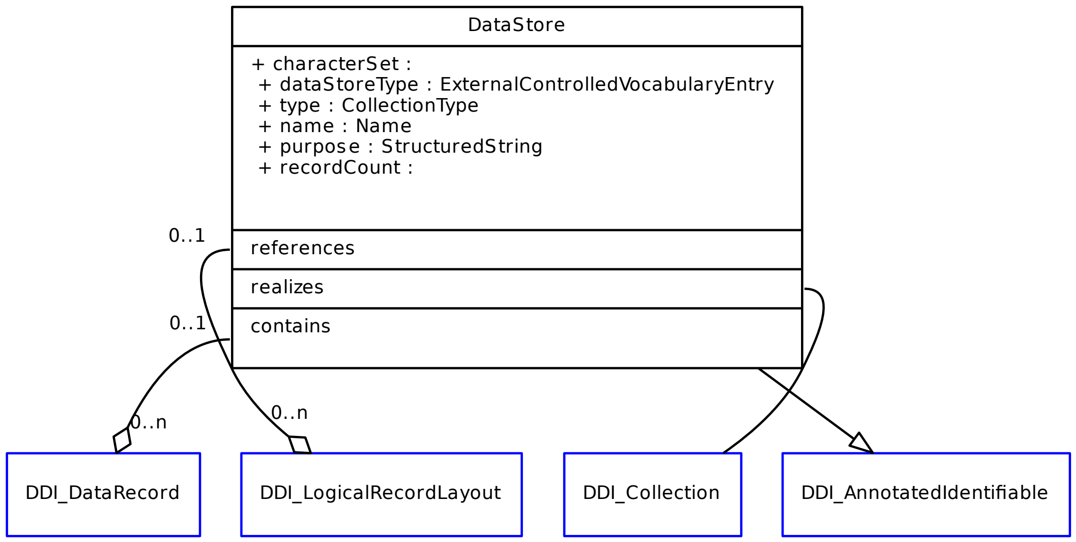
realizes

Allows the DataStore to function as a Collection, enabling, for example, ordering of its components.

references

Can point to one or more descriptions of the structure (type) of the Data Record.

Graph



6.12.5 DatastoreLibrary

A Collection or Library of DataStores

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
contains	DataStore	0..n
realizes	Collection	0..n

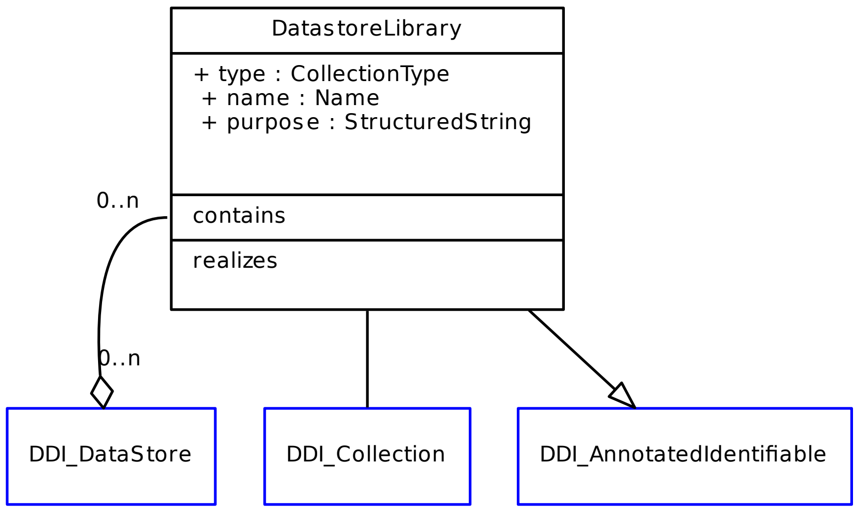
contains

The DataStores in the Catalog

realizes

Realizes the pattern collection

Graph



6.12.6 Datum

A Datum is the designation of a concept with a notion of equality defined.

Extends

Designation

Relationships

Name	Type	Cardinality
isBoundedBy	InstanceVariable	0..n
isConstrainedOf	ValueDomain	0..n

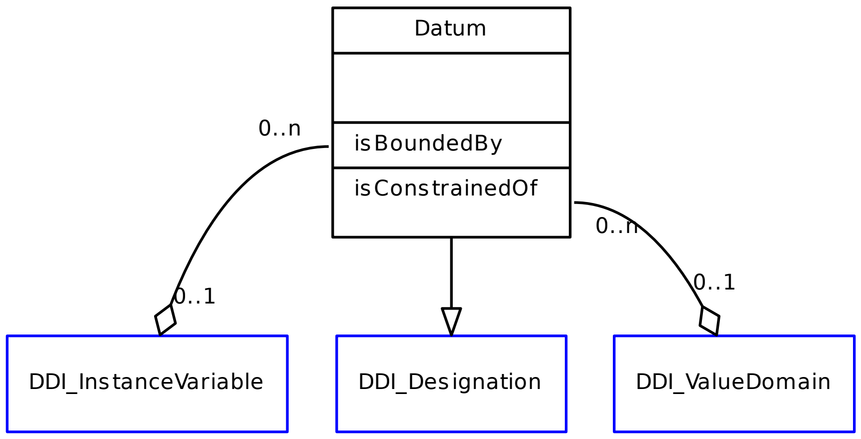
isBoundedBy

A Datum is bounded by an InstanceVariable.

isConstrainedOf

A Datum is drawn from a set of values, either substantive or sentinel

Graph



6.12.7 IdentifierRole

An IdentifierRole identifies an InstanceVariable as being an identifier within a ViewPoint.

Extends

ViewpointRole

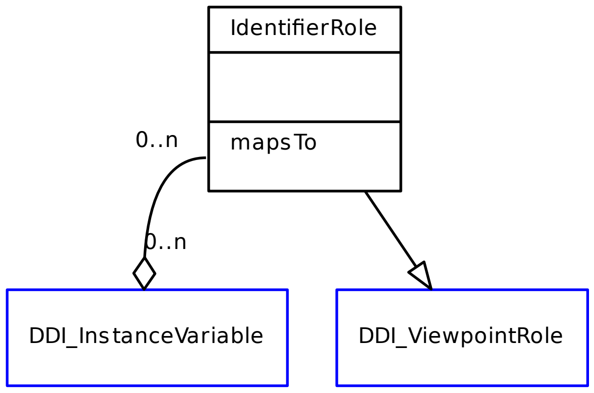
Relationships

Name	Type	Cardinality
mapsTo	InstanceVariable	0..n

mapsTo

The InstanceVariables being assigned the role of identifier

Graph



6.12.8 InstanceVariableMapping

Relation between Instance Variables in different Logical Record Layouts.

Extends

AnnotatedIdentifiable

Relationships

Name	Type	Cardinality
maps	InstanceVariable	0..n
realizes	UnorderedTuple	0..n

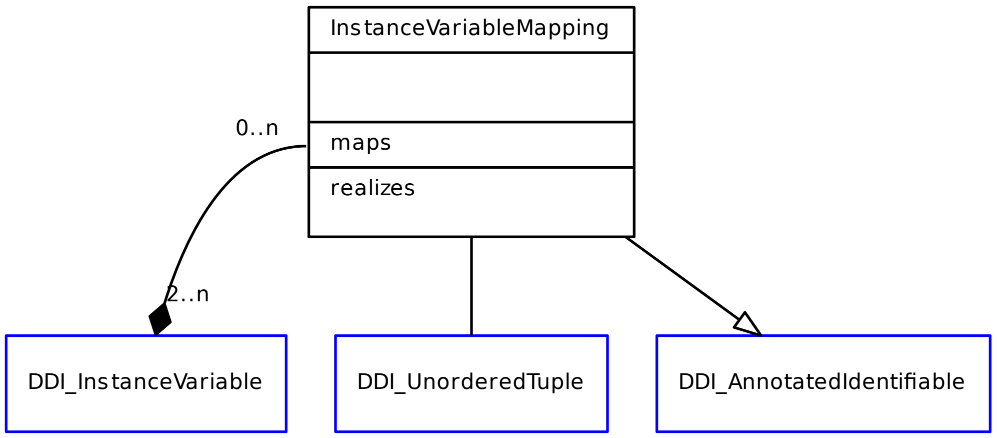
maps

Mapping between Instance Variables in different Logical Record Layouts.

realizes

Class in the Collections Pattern realized by this class.

Graph



6.12.9 LayoutOrderedPair

The pair of Instance Variables in a Record Layout which are being placed in a sequence.

Extends

Identifiable

Relationships

Name	Type	Cardinality
realizes	OrderedPair	0..n
source	InstanceVariable	0..n
target	InstanceVariable	0..n

realizes

Class of the Collection pattern realized by this class.

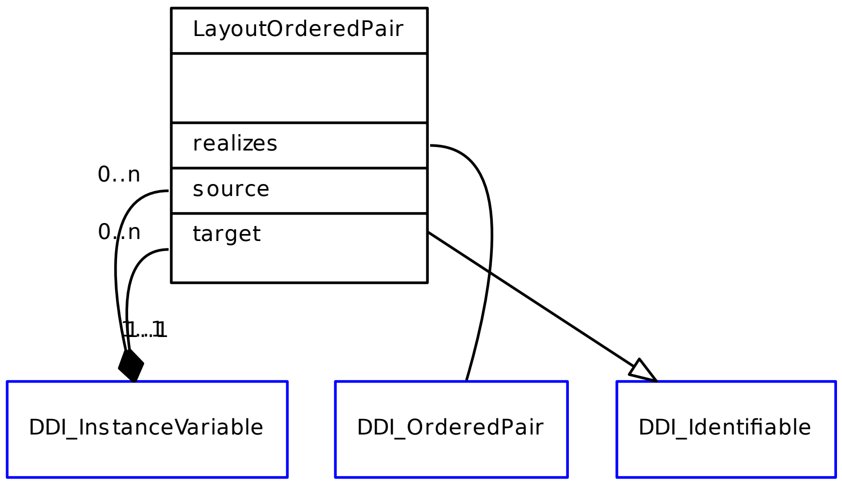
source

Specialization of source in OrderedPair for Instance Variables in a Logical Record Layout.

target

Specialization of target in OrderedPair for Instance Variables in a Logical Record Layout.

Graph



6.12.10 LogicalRecordLayout

Collection of Instance Variables that function as a record type describing the structure of individual records.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
contains	InstanceVariable	0..1
isViewedFrom	Viewpoint	0..n
nests	LogicalRecordLayout	0..n
realizes	Collection	0..n

contains

Instance Variable(s) defining this record layout.

isViewedFrom

Viewpoint associated with the record layout

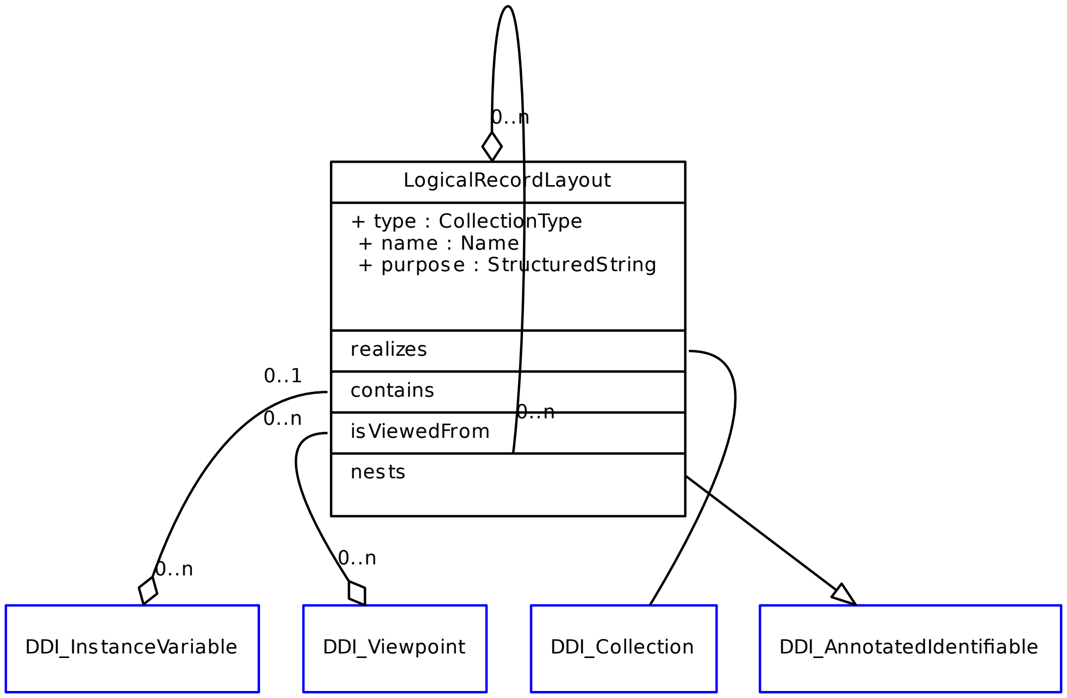
nests

LogicalRecordLayout nested in a LogicalRecordLayout

realizes

Class of the Collection pattern realized by this class.

Graph



6.12.11 LogicalRecordLayoutOrder

A realization of OrderRelation that is used to describe the sequence of Instance Variables in a record layout.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
displayLabel	DisplayLabel	0..n
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

criteria

Intensional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

displayLabel

A display label for the OrderRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

reflexivity

Fixed to Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Transitive

usage

A display label for the OrderRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

‘

Relationships

Name	Type	Cardinality
contains	LayoutOrderedPair	0..n
realizes	OrderRelation	0..n
structures	LogicalRecordLayout	0..n

contains

Layout ordered pairs describing the Instance Variable sequence.

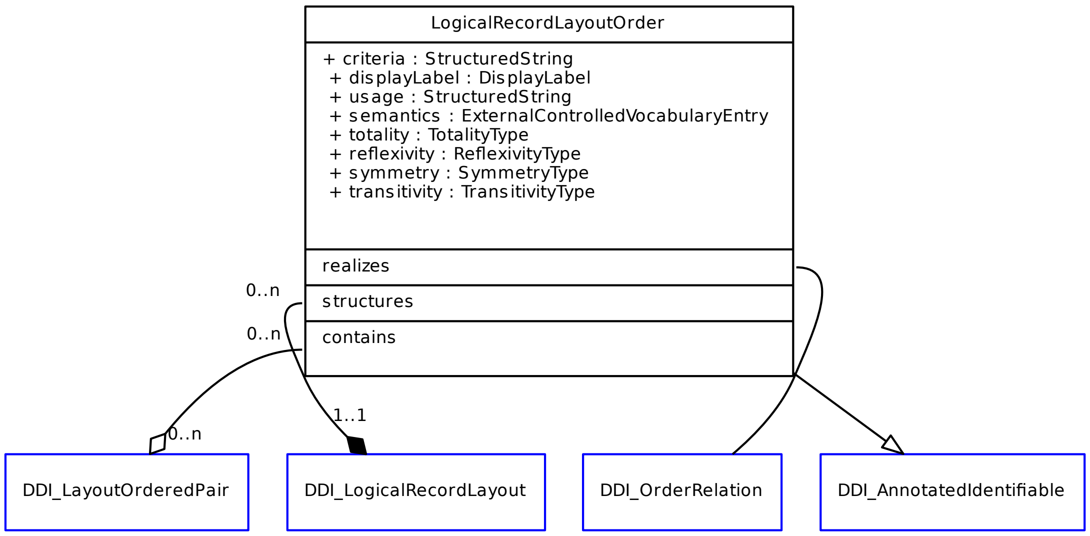
realizes

Class of the Collection pattern realized by this class.

structures

The Instance Variables in a record layout to be sequenced.

Graph


LogicalRecordLayoutOrder
+ criteria : StructuredString
+ displayLabel : DisplayLabel
+ usage : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
realizes
structures
contains
0..n
0..n
0..n
1..1
DDI_LayoutOrderedPair
DDI_LogicalRecordLayout
DDI_OrderRelation
DDI_AnnotatedIdentifiable

6.12.12 MeasureRole

A MeasureRole identifies an InstanceVariable as being a measure within a ViewPoint.

Extends

ViewpointRole

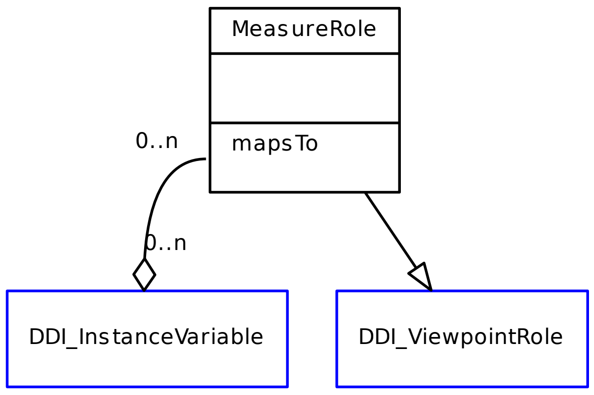
Relationships

Name	Type	Cardinality
mapsTo	InstanceVariable	0..n

mapsTo

The InstanceVariables being assigned the role of measure

Graph



6.12.13 Observation

The result of applying a particular Capture in an Instrument to some Unit.

‘

Relationships

Name	Type	Cardinality
isDesignatedBy	Datum	1..1
isMadeOn	Unit	0..n
producedByInstrumentComponent	InstrumentComponent	1..1

isDesignatedBy

A datum designates an observation

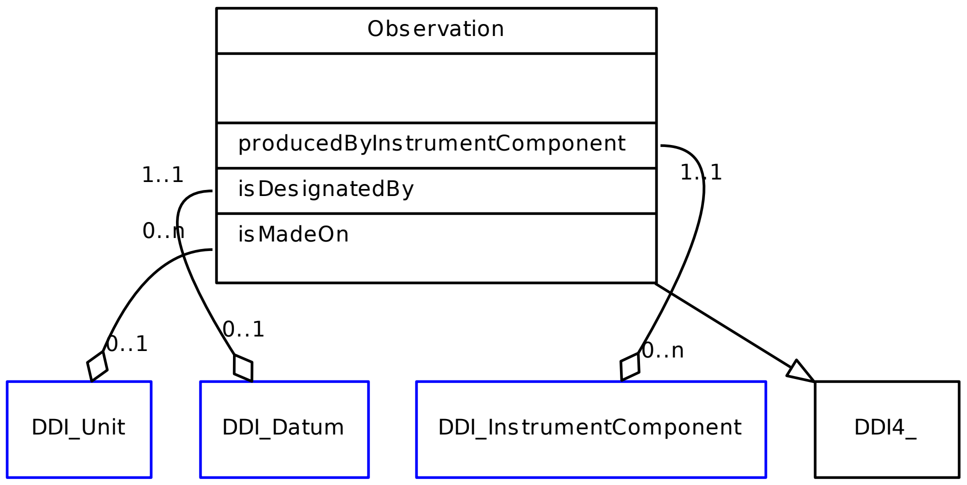
isMadeOn

The Unit of which the Observation is made.

producedByInstrumentComponent

The InstanceMeasure, InstanceQuestion and/or other InstrumentComponents used to create the Observation.

Graph



6.12.14 RecordRelation

The RecordRelation object is used to indicate relationships among record types within and between datasets. For InstanceVariables existing in a data set with multiple record layouts, pairs of InstanceVariables may function as paired keys to permit the expression of hierarchical links between records of different types. These links between keys in different record types could also be used to link records in a relational structure.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n
purpose	StructuredString	0..1
usage	StructuredString	0..1

displayLabel

A display label for the CollectionCorrespondence. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	InstanceVariableMapping	0..n
maps	LogicalRecordLayout	0..n
realizes	SymmetricRelation	0..n

contains

Correspondences between Instance Variables of different Logical Record Layout. Realization of contains in Symmetric Relation.

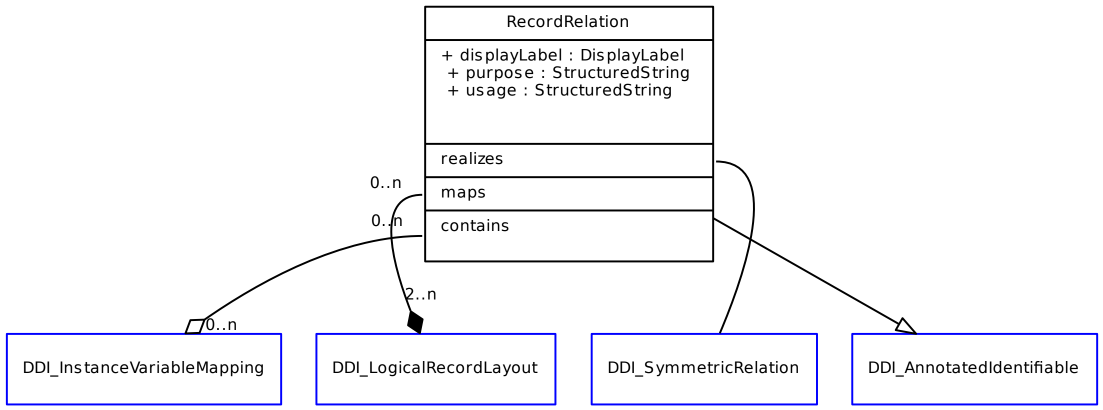
maps

Map related record types, which are Collections of Instance Variables. Realization of structures in Symmetric Relation.

realizes

Class in the Collections Pattern realized by this class.

Graph


RecordRelation
+ displayLabel : DisplayLabel
+ purpose : StructuredString
+ usage : StructuredString
realizes
maps
contains
0..n
0..n
2..n
0..n
DDI_InstanceVariableMapping
DDI_LogicalRecordLayout
DDI_SymmetricRelation
DDI_AnnotatedIdentifiable

6.12.15 Viewpoint

The assignment of measure, identifier and attribute roles to InstanceVariables

Extends

AnnotatedIdentifiable

Relationships

Name	Type	Cardinality
hasAttributeRole	AttributeRole	1..n
hasIdentifierRole	IdentifierRole	1..1
hasMeasureRole	MeasureRole	1..1

hasAttributeRole

The InstanceVaribles functioning as attributes

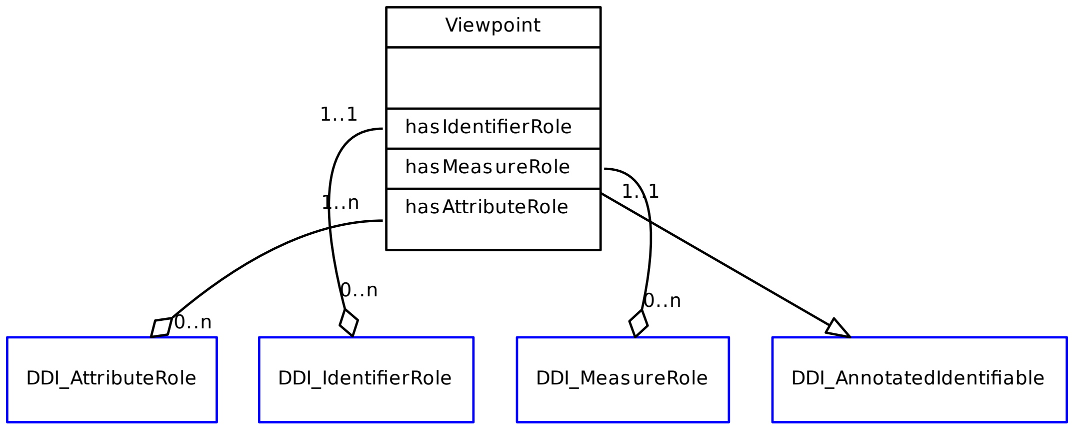
hasIdentifierRole

The InstanceVaribles functioning as identifiers

hasMeasureRole

The InstanceVaribles functioning as measures

Graph



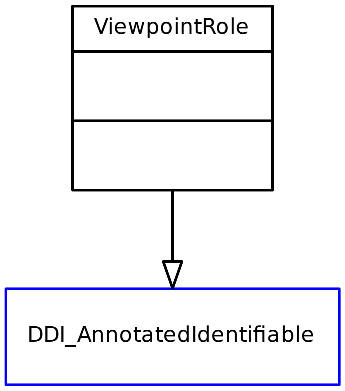
6.12.16 ViewpointRole

A ViewpointRole designates the function an InstanceVariable performs in the context of the ViewPoint. (Identifier-Role, AttributeRole, or MeasureRole of interest).

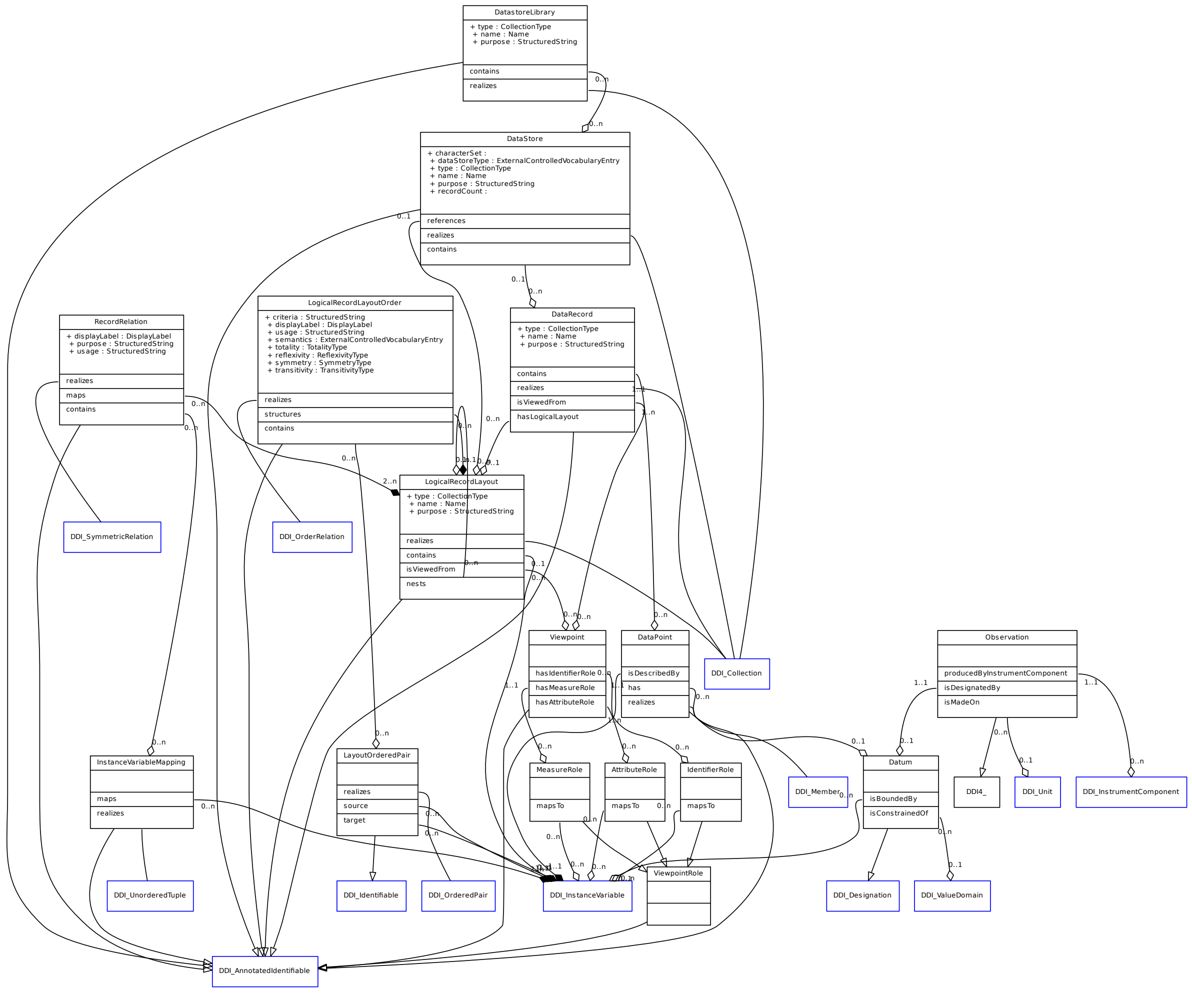
Extends

:ref:‘AnnotatedIdentifiable

Graph



6.12.17 Graph



6.13 Primitives

A package is a administrative collection of classes in DDI. These are not namespaces.

6.13.1 Graph

6.14 Discovery

A package is a administrative collection of classes in DDI. These are not namespaces.

6.14.1 Access

Describes access to the annotated object. This item includes a confidentiality statement, descriptions of the access permissions required, restrictions to access, citation requirements, depositor requirements, conditions for access, a disclaimer, any time limits for access restrictions, and contact information regarding access.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
accessConditions	StructuredString	0..1
accessPermission	Form	0..n
applyAccessTo	InternationalIdentifier	0..n
citationRequirement	StructuredString	0..1
confidentialityStatement	StructuredString	0..1
contactAgent	AgentAssociation	0..n
depositRequirement	StructuredString	0..1
disclaimer	StructuredString	0..1
purpose	StructuredString	0..1
restrictions	StructuredString	0..1
validDates	DateRange	0..1

accessConditions

A statement regarding conditions for access. May be expressed in multiple languages and supports the use of structured content.

accessPermission

A link to a form used to provide access to the data or metadata including a statement of the purpose of the form.

applyAccessTo

Identification for an object covered by the access description. This may be any annotated object (collection, publication, identifiable object).

citationRequirement

A statement regarding the citation requirement. May be expressed in multiple languages and supports the use of structured content.

confidentialityStatement

A statement regarding the confidentiality of the related data or metadata.

contactAgent

The agent to contact regarding access including the role of the agent.

depositRequirement

A statement regarding depositor requirements. May be expressed in multiple languages and supports the use of structured content.

disclaimer

A disclaimer regarding the liability of the data producers or providers. May be expressed in multiple languages and supports the use of structured content.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text

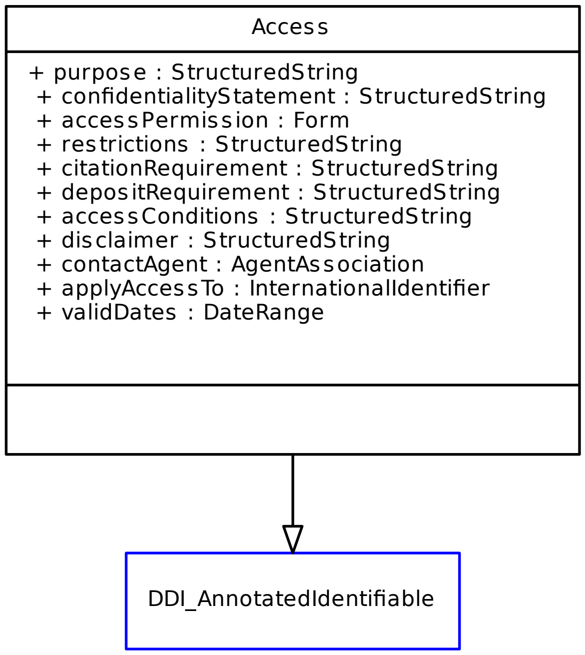
restrictions

A statement regarding restrictions to access. May be expressed in multiple languages and supports the use of structured content.

validDates

The date range or start date of the access description.

Graph


Access
+ purpose : StructuredString
+ confidentialityStatement : StructuredString
+ accessPermission : Form
+ restrictions : StructuredString
+ citationRequirement : StructuredString
+ depositRequirement : StructuredString
+ accessConditions : StructuredString
+ disclaimer : StructuredString
+ contactAgent : AgentAssociation
+ applyAccessTo : InternationalIdentifier
+ validDates : DateRange
DDI_AnnotatedIdentifiable

6.14.2 BoundingBox

A type of Spatial coverage describing a rectangular area within which the actual range of location fits. A BoundingBox can be described by 4 numbers, or two x,y coordinates - the maxima of the north, south, east, and west coordinates found in the area.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
eastLongitude	Real	1..1
northLatitude	Real	1..1
southLatitude	Real	1..1
westLongitude	Real	1..1

eastLongitude

The easternmost coordinate expressed as a decimal between the values of -180 and 180 degrees

northLatitude

The northernmost coordinate expressed as a decimal between the values of -90 and 90 degrees.

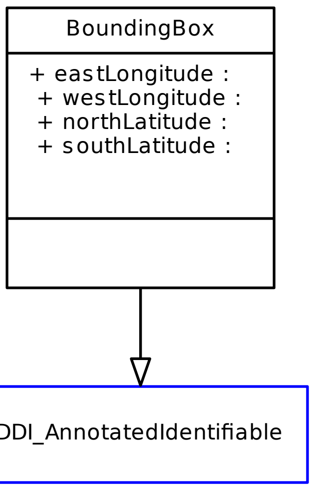
southLatitude

The southermost latitude expressed as a decimal between the values of -90 and 90 degrees

westLongitude

The westernmost coordinate expressed as a decimal between the values of -180 and 180 degrees

Graph



6.14.3 CatalogOfItems

Means of creating a catalog of items in a collection or sub-sets of a larger catalog

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
contains	CollectionItem	0..n
hasSubCollection	CatalogOfItems	0..n
realizes	Collection	0..n

contains

Constrains member types to CollectionItem

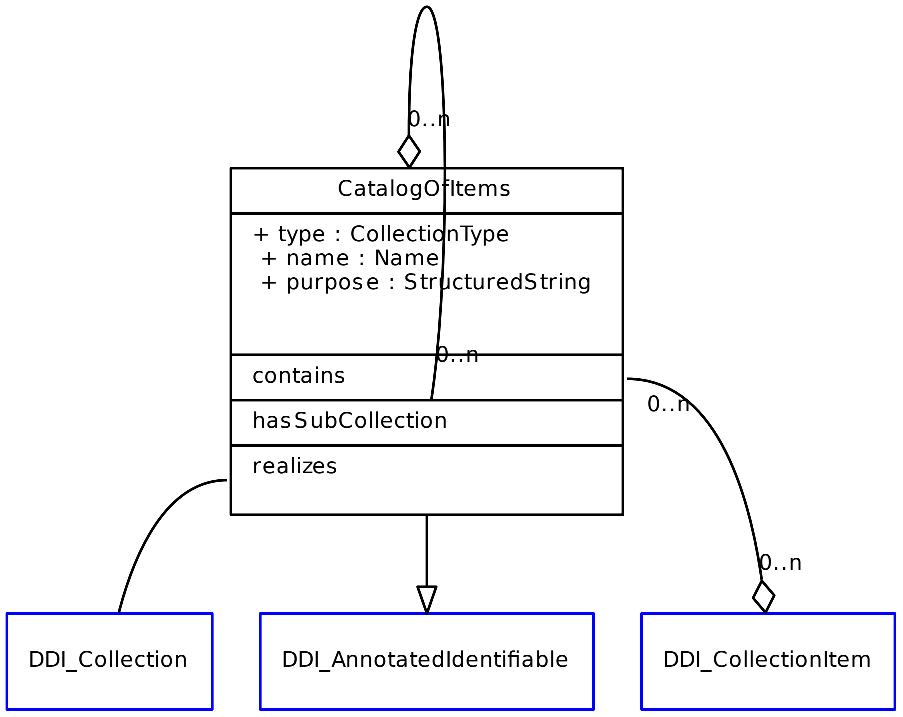
hasSubCollection

May contain sub-collections of catalogs of items

realizes

Realizes the pattern collection

Graph



6.14.4 CollectionItem

A collection item is designed to provide a bibliographic, coverage and related description of an item in a collection.

Extends

AnnotatedIdentifiable

Relationships

Name	Type	Cardinality
belongsToSeries	SeriesStatement	0..n
describesAccess	Access	0..n
hasCoverage	Coverage	0..n
hasFunding	FundingInformation	0..n
realizes	Member	0..n

belongsToSeries

Collection item belongs to this series

describesAccess

Information on how to access the collection item

hasCoverage

Describes the temporal, topical, and spatial coverof the colleciton item

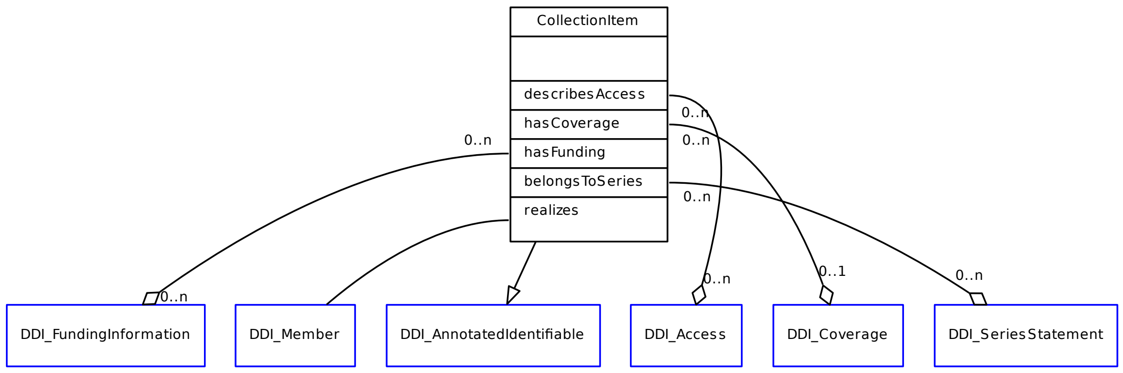
hasFunding

Information concerning the funding source for the colleciton item

realizes

Collection Item can act as the member of a collection

Graph



6.14.5 Coverage

Coverage information for an annotated object. Includes coverage information for temporal, topical, and spatial coverage.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
overview	StructuredString	0..1

overview

A generic description including temporal, topical, and spatial coverage that is the equivalent of dc:coverage (the refinement base of dcterms:spatial and dcterms:temporal. Use specific coverage content for detailed information. Short natural language account of the information obtained from the combination of properties and relationships associated with an object. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
hasSpatialCoverage	SpatialCoverage	0..n
hasTemporalCoverage	TemporalCoverage	0..n
hasTopicalCoverage	TopicalCoverage	0..n

hasSpatialCoverage

Description of the spatial (geographic) coverage of the contents of the annotated object.

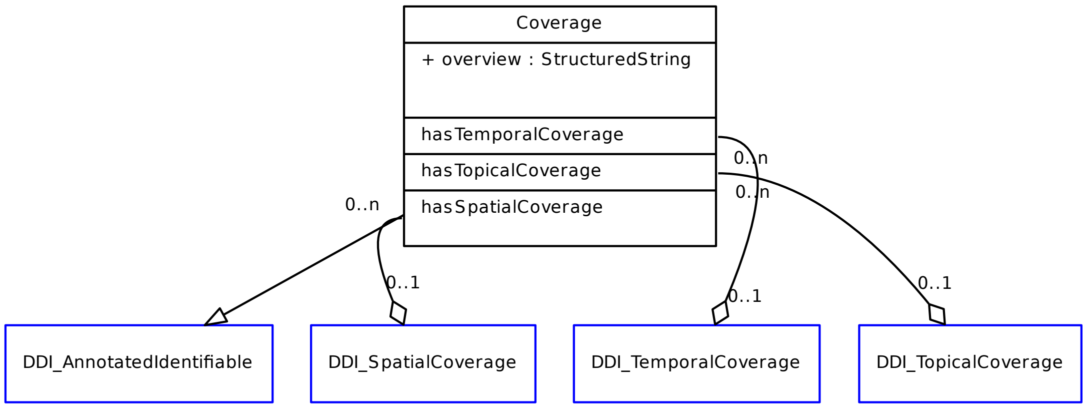
hasTemporalCoverage

The dates and time periods described by the contents of the annotated object.

hasTopicalCoverage

The topics covered by the contents of the annotated object. These may be expressed by subject classification systems and structured or unstructured keywords.

Graph



6.14.6 SeriesStatement

Describes a series by providing citation information and coverage information.

Extends

Identifiable

Properties

Name	Type	Cardinality
citationOfSeries	Annotation	0..n

citationOfSeries

Bibliographic citation of the Series

‘

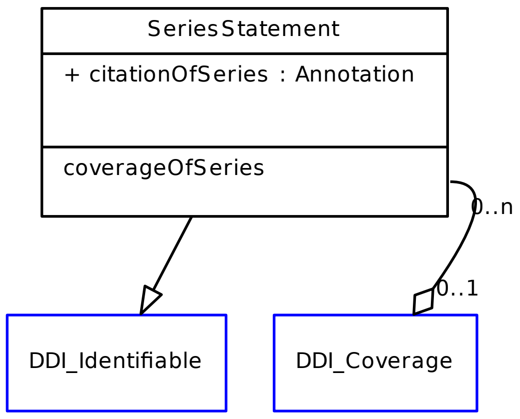
Relationships

Name	Type	Cardinality
coverageOfSeries	Coverage	0..n

coverageOfSeries

Temporal, topical, and spatial coverage of the series

Graph



6.14.7 SpatialCoverage

A description of spatial coverage (geographic coverage) of the annotated object. Spatial coverage is described using a number of objects that support searching by a wide range of systems (geospatial coordinates, geographic classification systems, and general systems using dcterms:spatial.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
description	StructuredString	0..1
spatialAreaCode	ExternalControlledVocabularyEntry	0..n
spatialObject	SpatialObject	0..1

description

A textual description of the spatial coverage to support general searches.

spatialAreaCode

Supports the use of a standardized code such as ISO 3166-1, the Getty Thesaurus of Geographic Names, FIPS-5, etc.

spatialObject

Indicates the most discrete spatial object type identified for a single case. Note that data can be collected at a geographic point (address) and reported as such in a protected file, and then aggregated to a polygon for a public file.

‘

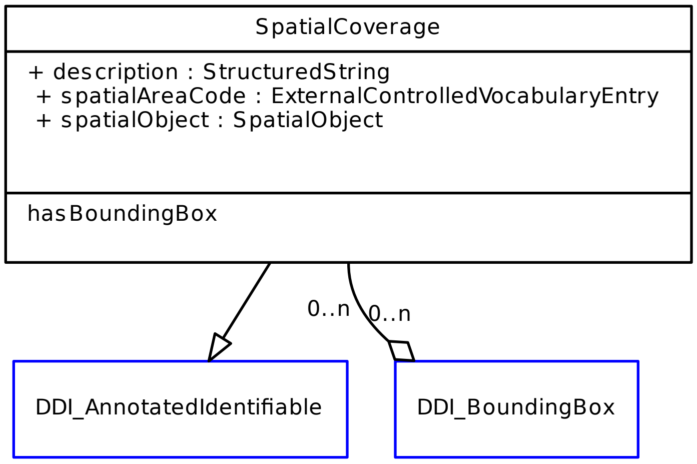
Relationships

Name	Type	Cardinality
hasBoundingBox	BoundingBox	0..n

hasBoundingBox

The north and south latitudes and east and west longitudes that define the spatial coverage area.

Graph


SpatialCoverage
+ description : StructuredString
+ spatialAreaCode : ExternalControlledVocabularyEntry
+ spatialObject : SpatialObject
hasBoundingBox
0..n
0..n
DDI_AnnotatedIdentifiable
DDI_BoundingBox

6.14.8 TemporalCoverage

Describes the date or time period covered by the annotated object. Allows for the use of a specifying the type of coverage date as well as associated subjects or keywords.

Extends

AnnotatedIdentifiable

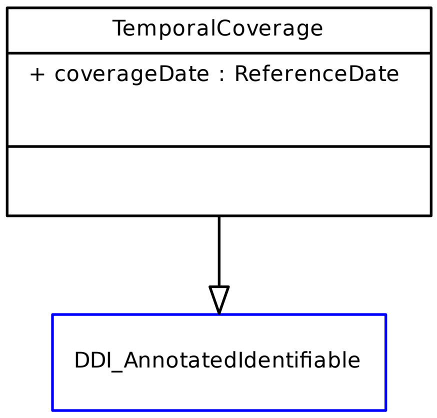
Properties

Name	Type	Cardinality
coverageDate	ReferenceDate	0..n

coverageDate

A date referencing a specific aspect of temporal coverage. The date may be typed to reflect coverage date, collection date, referent date, etc. Subject and Keywords may be associated with the date to specify a specific set of topical information (i.e. Residence associated with a date 5 years prior to the collection date).

Graph



6.14.9 TopicalCoverage

Describes the topical coverage of the module using Subject and Keyword. Note that upper level modules should include all the members of lower level modules. Subjects are members of structured classification systems such as formal subject headings in libraries. Keywords may be structured (e.g. TheSoz thesauri) or unstructured and reflect the terminology found in the document and other related (broader or similar) terms.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
keyword	ExternalControlledVocabularyEntry	0..n
subject	ExternalControlledVocabularyEntry	0..n

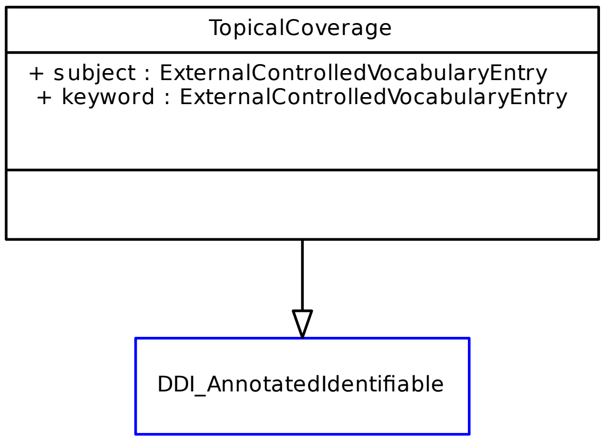
keyword

A keyword that describes the topical coverage of the content of the annotated object. Keywords may be structured (e.g. TheSoz thesauri) or unstructured and reflect the terminology found in the document and other related (broader or similar) terms. Uses and InternationalCodeValue and may indicate the language of the code used.

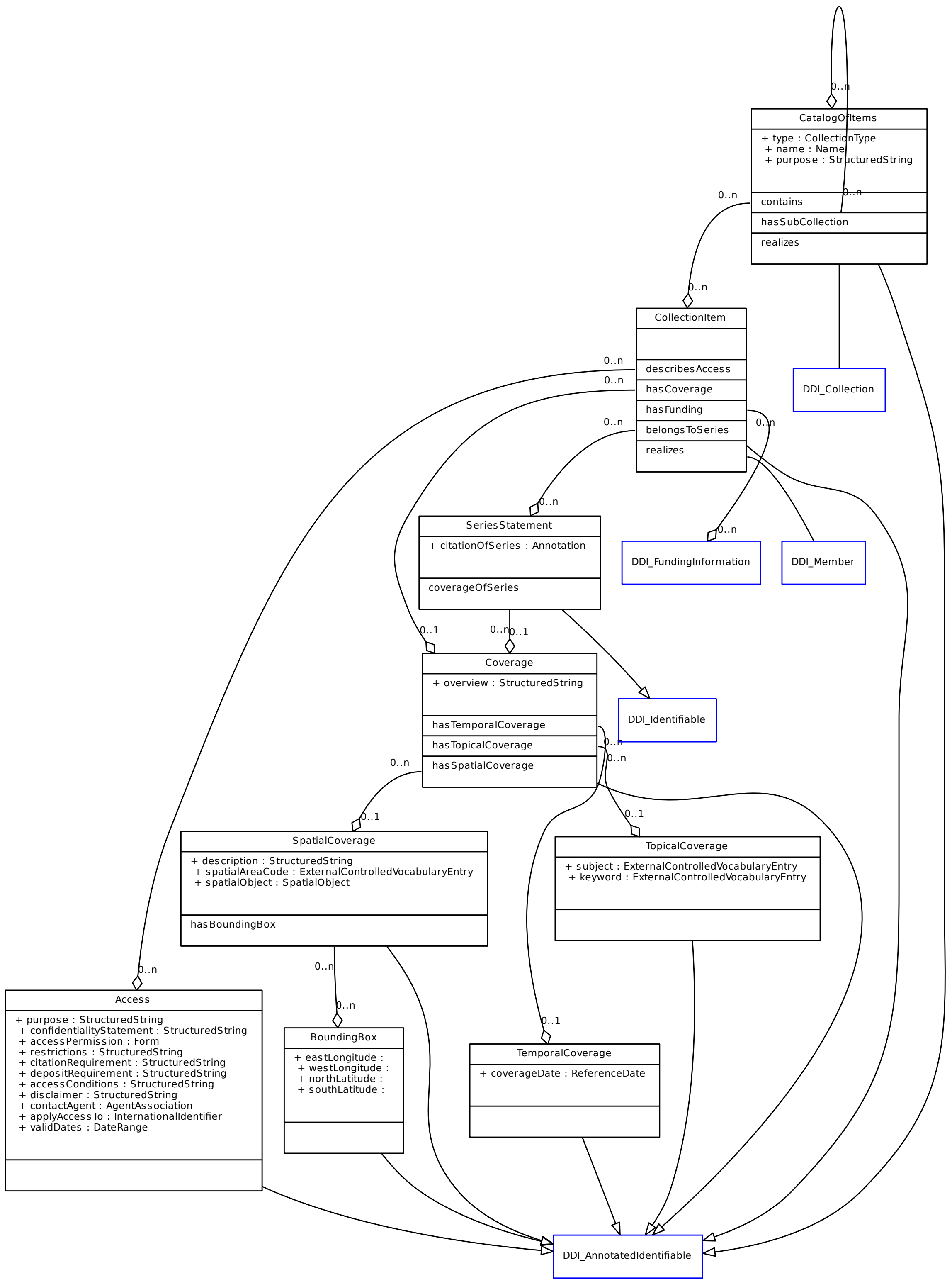
subject

A subject that describes the topical coverage of the content of the annotated object. Subjects are members of structured classification systems such as formal subject headings in libraries. Uses and InternationalCodeValue and may indicate the language of the code used.

Graph


TopicalCoverage
+ subject : ExternalControlledVocabularyEntry
+ keyword : ExternalControlledVocabularyEntry
DDI_AnnotatedIdentifiable

6.14.10 Graph


CatalogOfItems
+ type : CollectionType
+ name : Name
+ purpose : StructuredString
contains
hasSubCollection
realizes
0..n
0..n
0..n
0..n
CollectionItem
describesAccess
hasCoverage
hasFunding
belongsToSeries
realizes
DDI_Collection
0..n
0..n
0..n
0..n
SeriesStatement
+ citationOfSeries : Annotation
coverageOfSeries
DDI_FundingInformation
DDI_Member
0..n
0..n
0..1
0..n
0..1
Coverage
+ overview : StructuredString
hasTemporalCoverage
hasTopicalCoverage
hasSpatialCoverage
DDI_Identifiable
0..n
0..n
0..n
0..1
0..1
SpatialCoverage
+ description : StructuredString
+ spatialAreaCode : ExternalControlledVocabularyEntry
+ spatialObject : SpatialObject
hasBoundingBox
TopicalCoverage
+ subject : ExternalControlledVocabularyEntry
+ keyword : ExternalControlledVocabularyEntry
0..n
0..n
0..n
0..1
Access
+ purpose : StructuredString
+ confidentialityStatement : StructuredString
+ accessPermission : Form
+ restrictions : StructuredString
+ citationRequirement : StructuredString
+ depositRequirement : StructuredString
+ accessConditions : StructuredString
+ disclaimer : StructuredString
+ contactAgent : AgentAssociation
+ applyAccessTo : InternationalIdentifier
+ validDates : DateRange
BoundingBox
+ eastLongitude :
+ westLongitude :
+ northLatitude :
+ southLatitude :
TemporalCoverage
+ coverageDate : ReferenceDate
DDI_AnnotatedIdentifiable

6.15 Comparison

A package is a administrative collection of classes in DDI. These are not namespaces.

6.15.1 Graph

6.16 CollectionsPattern

A package is a administrative collection of classes in DDI. These are not namespaces.

6.16.1 AcyclicPrecedenceRelation

Relation over members in a collection (set or bag) characterized by the following assignments to the types inherited from Asymmetric Binary Relation: - Reflexivity = Anti_Reflexive; - Symmetry = Anti_Symmetric (more restrictive than parent class) ; - Transitivity = Neither.

It must contain like items.

Extends

AsymmetricBinaryRelation

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
transitivity	TransitivityType	1..1

reflexivity

Fixed to Anti_Reflexive

semantics

Controlled vocabulary for the AcyclicPrecedenceRelationsemantics. It should contain, at least, the following: Tangential_Proper_Part and Temporal_Overlaps

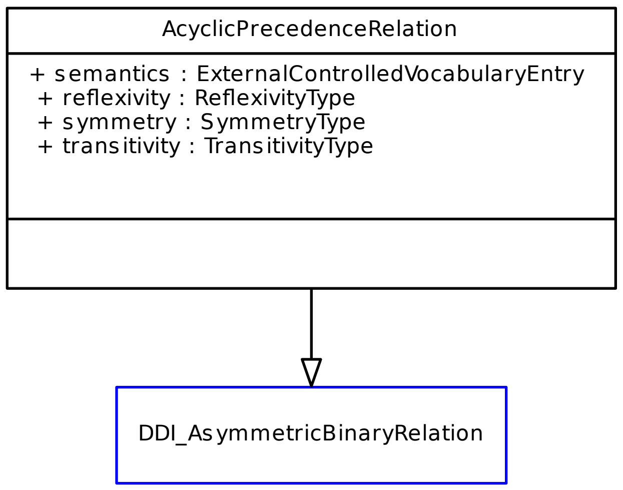
symmetry

Fixed to Anti_Symmetric

transitivity

Fixed to Neither

Graph


AcyclicPrecedenceRelation
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
DDI_AsymmetricBinaryRelation

6.16.2 AsymmetricBinaryRelation

Relation over members in a Collection (set or bag) characterized by the following assignments to the types inherited from Binary Relation: - Symmetry = Asymmetric.

Extends

BinaryRelation

Properties

Name	Type	Cardinality
symmetry	SymmetryType	1..1

symmetry

Fixed to Asymmetric

‘

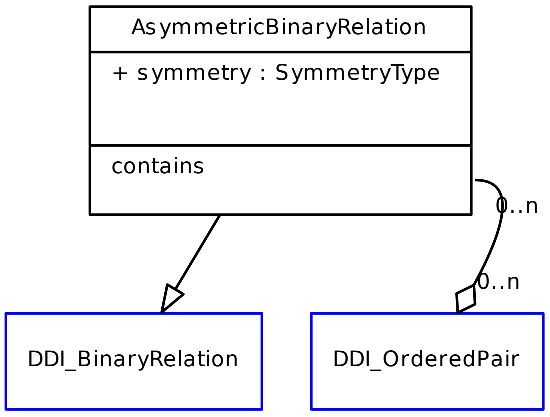
Relationships

Name	Type	Cardinality
contains	OrderedPair	0..n

contains

Ordered pair of Members.

Graph



6.16.3 AsymmetricRelation

Relation whose n-ary tuples are ordered. It could be the result of a join between one or more Binary Relations.

Extends

NaryRelation

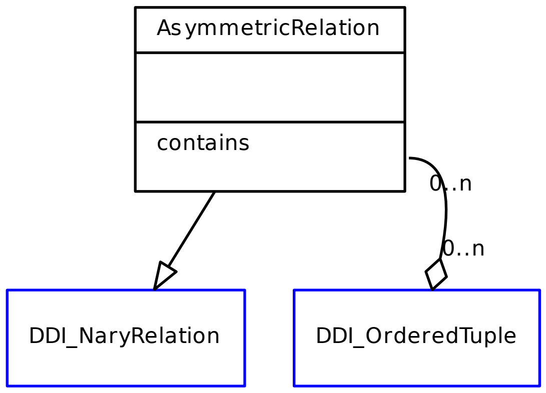
Relationships

Name	Type	Cardinality
contains	OrderedTuple	0..n

contains

Ordered tuples of Members.

Graph



6.16.4 BinaryRelation

Set of ordered pairs of Members of the Collection the Binary Relation structures.

Extends

Identifiable

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1

reflexivity

Controlled Vocabulary to specify whether the relation is reflexive, anti-reflexive, neither or unknown.

symmetry

Controlled Vocabulary to specify whether the relation is symmetric, anti-symmetric, asymmetric or unknown.

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Controlled Vocabulary to specify whether the relation is transitive, anti-transitive, neither or unknown.

‘

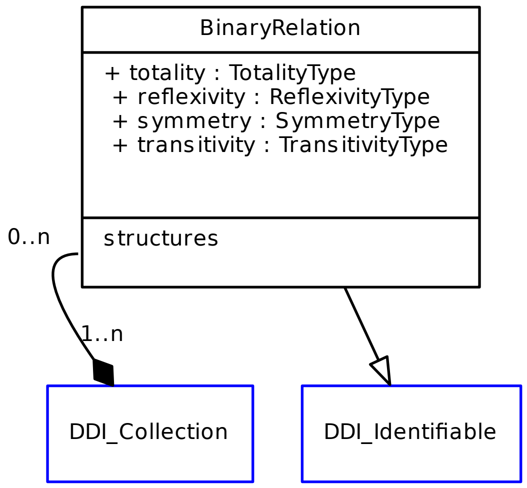
Relationships

Name	Type	Cardinality
structures	Collection	0..n

structures

Collection whose Members are grouped by pairs to support a variety of structures.

Graph


BinaryRelation
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
structures
0..n
1..n
DDI_Collection
DDI_Identifiable

6.16.5 Collection

Collection container (set or bag). It could have an optional order relation (total or partial) associated to it to model linear order, hierarchies and nesting. A Collection is also a subtype of Member to allow for nested collections.

Extends

Member

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

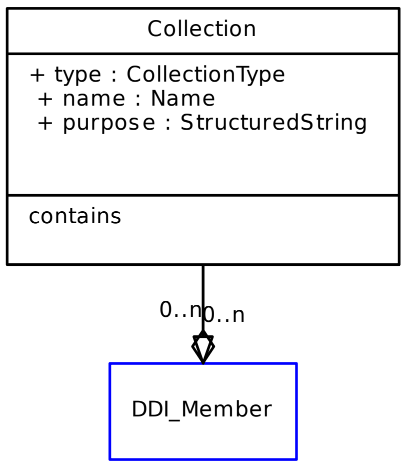
Relationships

Name	Type	Cardinality
contains	Member	0..n

contains

Members of the collection.

Graph



6.16.6 EquivalenceRelation

Relation over members in a collection (set or bag) characterized by the following assignments to the types inherited from Symmetric Binary Relation: - Reflexivity = Reflexive; - Symmetry = Symmetric (same as parent class); - Transitivity = Transitive.

It must contain like items.

Extends

SymmetricBinaryRelation

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
transitivity	TransitivityType	1..1

reflexivity

Fixed to Reflexive

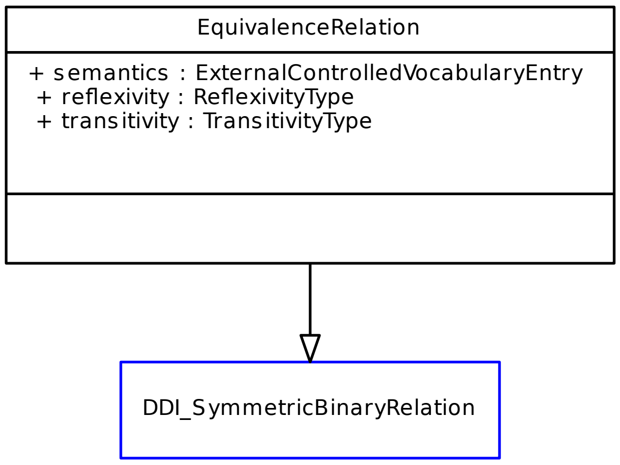
semantics

Controlled vocabulary for the equivalence relation semantics. It should contain, at least, the following: Same_As, Similar_To, Congruent_To, Compatible_With, Equidistant_From, Temporal_Equals.

transitivity

Fixed to Transitive

Graph


EquivalenceRelation
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ transitivity : TransitivityType
DDI_SymmetricBinaryRelation

6.16.7 ImmediatePrecedenceRelation

Relation over members in a collection (set or bag) characterized by the following assignments to the types inherited from Asymmetric Binary Relation: - Reflexivity = Anti_Reflexive; - Symmetry = Anti_Symmetric (more restrictive than parent class) ; - Transitivity = Anti_Transitive.

It must contain like items.

Extends

AsymmetricBinaryRelation

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
trastivity	TransitivityType	1..1

reflexivity

Fixed to Anti_Reflexive

semantics

Controlled vocabulary for the ImmediatePrecedenceRelation semantics. It should contain, at least, the following: Parent_Of, Child_Of, Next, Previous, Instance_Of, Temporal_Meets.

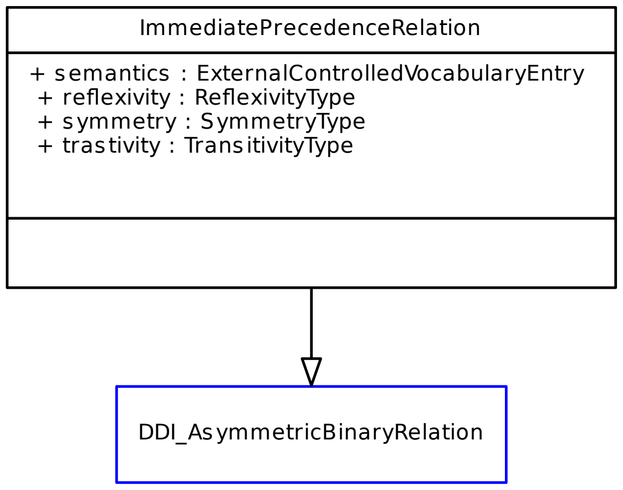
symmetry

Fixed to Anti_Symmetric

trastivity

Fixed to Anti_Transitive

Graph


ImmediatePrecedenceRelation
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ trastivity : TransitivityType
DDI_AsymmetricBinaryRelation

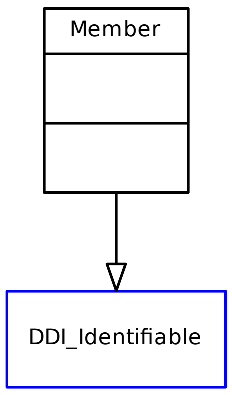
6.16.8 Member

Generic class representing members of a collection.

Extends

:ref:‘Identifiable

Graph



6.16.9 NaryRelation

Set of ordered, n-ary tuples of Members of the Collection the Relation structures.

Extends

Identifiable

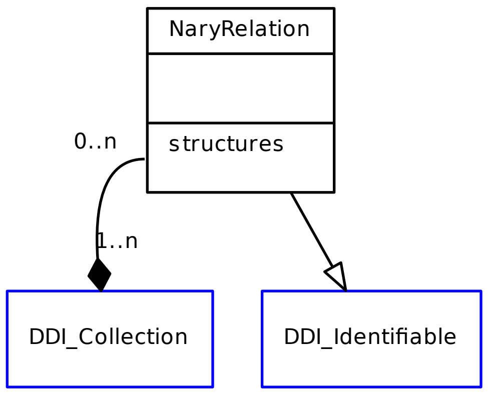
Relationships

Name	Type	Cardinality
structures	Collection	0..n

structures

Collection(s) whose Members are grouped by n-ary tuples to support a variety of structures.

Graph



6.16.10 OrderRelation

Relation over members in a collection (set or bag) characterized by the following assignments to the types inherited from Asymmetric Binary Relation: - Reflexivity = Reflexive; - Symmetry = Anti_Symmetric (more restrictive than parent class) ; - Transitivity = Transitive.

It must contain like items.

Extends

AsymmetricBinaryRelation

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
transitivity	TransitivityType	1..1

criteria

Intentional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

reflexivity

Fixed to Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

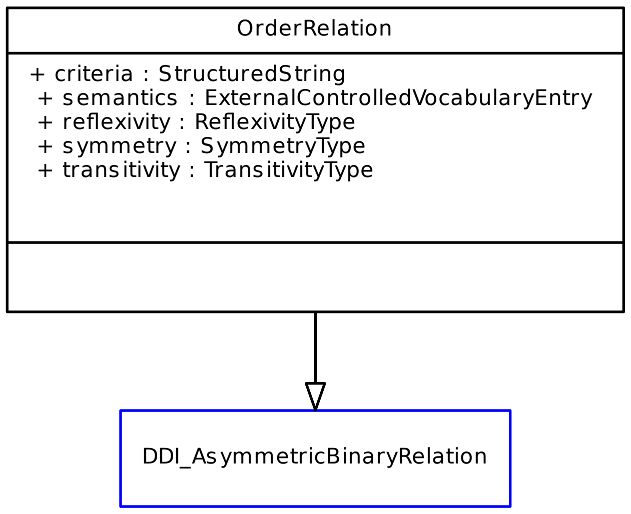
symmetry

Fixed to Anti_Symmetric

transitivity

Fixed to Transitive

Graph


OrderRelation
+ criteria : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
DDI_AsymmetricBinaryRelation

6.16.11 OrderedPair

Ordered pair of Members of the Collection structured by the Asymmetric Binary Relation to which the pair belongs.

Extends

Identifiable

Relationships

Name	Type	Cardinality
source	Member	0..n
target	Member	0..n

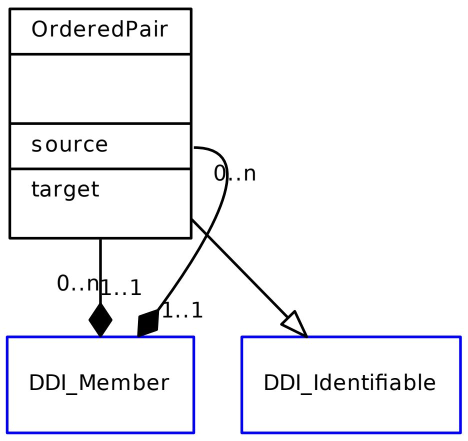
source

First member in the Ordered Pair.

target

Second member in the Ordered Pair.

Graph



6.16.12 OrderedTuple

Ordered n-ary tuple of Members of the Collection structured by the Asymmetric Relation to which the tuple belongs. Members are separated into two groups: source and target.

Extends

Identifiable

Relationships

Name	Type	Cardinality
source	Member	0..n
target	Member	0..n

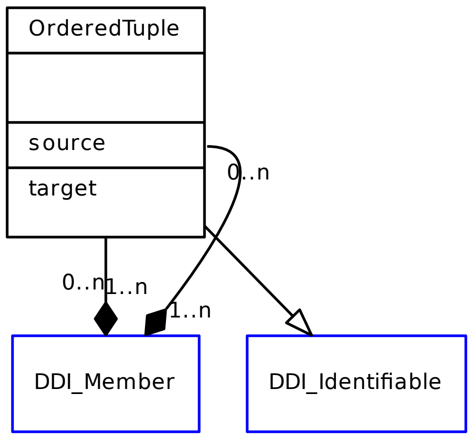
source

Set of Members in the n-ary tuple identified as source. If the Ordered Tuple is the result of joining one or more Ordered Pairs, then it is the union of the source Members in the joined Ordered Pairs.

target

Set of Members in the n-ary tuple identified as target. If the Ordered Tuple is the result of joining one or more Ordered Pairs, then it is the union of the target Members in the joined Ordered Pairs.

Graph



6.16.13 StrictOrderRelation

Relation over members in a collection (set or bag) characterized by the following assignments to the types inherited from Asymmetric Binary Relation: - Reflexivity = Anti_Reflexive; - Symmetry = Anti_Symmetric (more restrictive than parent class) ; - Transitivity = Transitive.

It must contain like items.

Extends

AsymmetricBinaryRelation

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
transitivity	TransitivityType	1..1

criteria

Intensional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

reflexivity

Fixed to Anti_Reflexive

semantics

Controlled vocabulary for the strict order relation semantics. It should contain, at least, the following: Descendant_Of, Strict_Part_Of, Less_Than, Strict_Subtype_Of, Predecessor_Of, Successor_Of, Temporal_Precedes, Temporal_Finishes, Temporal_Contains and Temporal_Starts

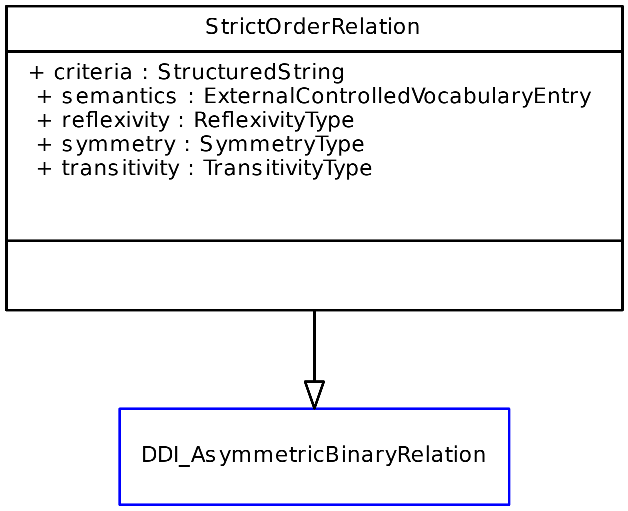
symmetry

Fixed to Anti_Symmetric

transitivity

Fixed to Transitive

Graph


StrictOrderRelation
+ criteria : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
DDI_AsymmetricBinaryRelation

6.16.14 SymmetricBinaryRelation

Relation over members in a Collection (set or bag) characterized by the following assignments to the types inherited from Binary Relation: - Symmetry = Symmetric.

Extends

BinaryRelation

Properties

Name	Type	Cardinality
symmetry	SymmetryType	1..1

symmetry

Fixed to Symmetric

‘

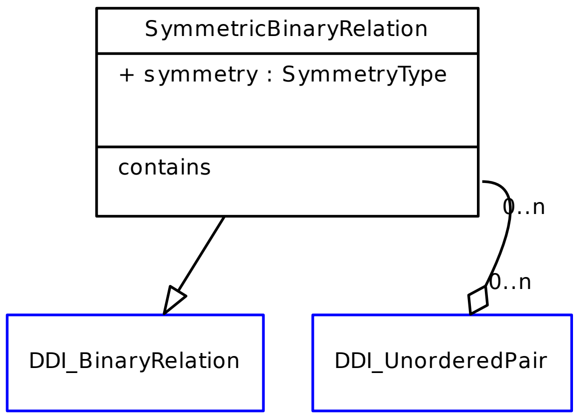
Relationships

Name	Type	Cardinality
contains	UnorderedPair	0..n

contains

Unordered pair of Members

Graph



6.16.15 SymmetricRelation

Relation whose n-ary tuples are not ordered. It could be the result of a join between one or more Binary Relations.

Extends

NaryRelation

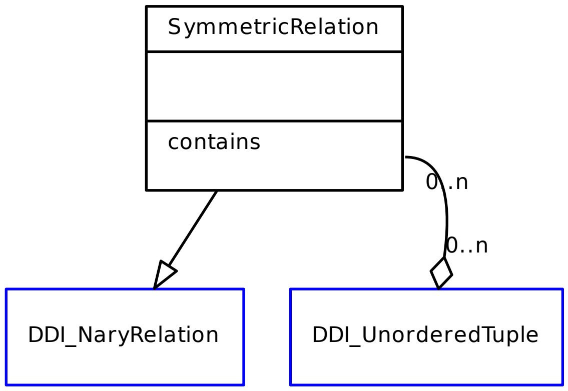
Relationships

Name	Type	Cardinality
contains	UnorderedTuple	0..n

contains

Unordered tuples of Members.

Graph



6.16.16 UnorderedPair

Unordered pair of Members of the Collection structured by the Symmetric Binary Relation to which the pair belongs.

Extends

Identifiable

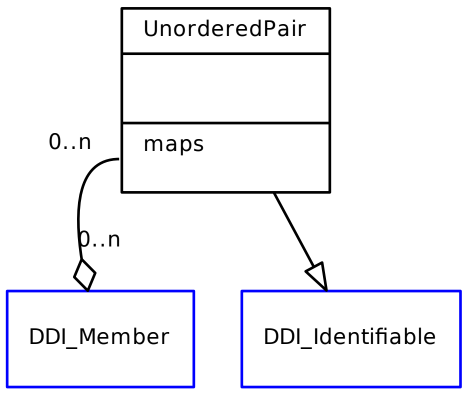
Relationships

Name	Type	Cardinality
maps	Member	0..n

maps

Members in the Unordered Pair

Graph



6.16.17 UnorderedTuple

Unordered n-ary tuple of Members of the Collection structured by the Symmetric Relation to which the tuple belongs.

Extends

Identifiable

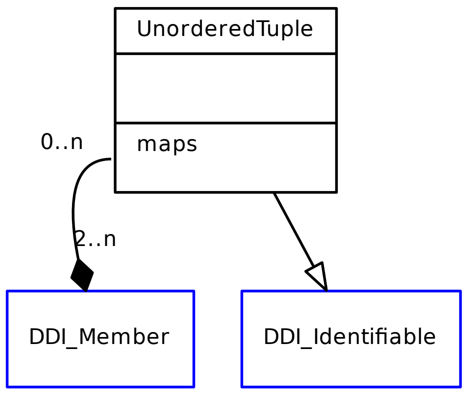
Relationships

Name	Type	Cardinality
maps	Member	0..n

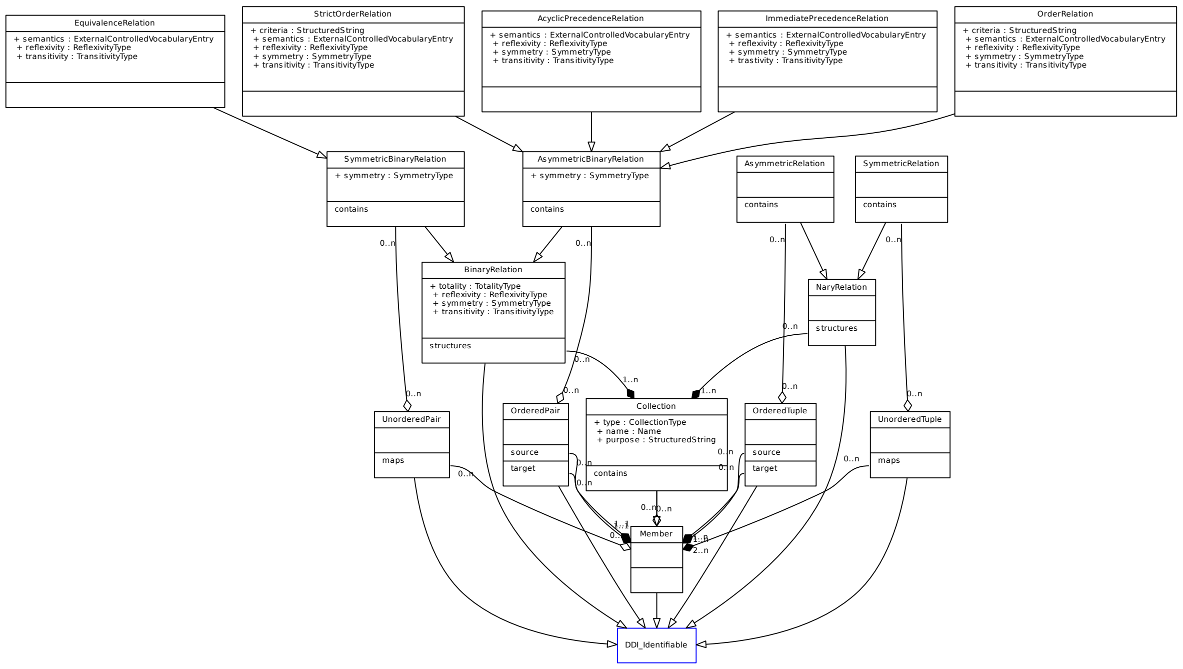
maps

Set of Members in the n-ary tuple. If the Unordered Tuple is the result of joining one or more Unordered Pairs, then it is the union of the Members in the joined Unordered Pairs.

Graph



6.16.18 Graph


EquivalenceRelation
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ transitivity : TransitivityType
StrictOrderRelation
+ criteria : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
AcyclicPrecedenceRelation
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
ImmediatePrecedenceRelation
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ trastivity : TransitivityType
OrderRelation
+ criteria : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
SymmetricBinaryRelation
+ symmetry : SymmetryType
contains
AsymmetricBinaryRelation
+ symmetry : SymmetryType
contains
AsymmetricRelation
contains
SymmetricRelation
contains
BinaryRelation
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
structures
NaryRelation
structures
UnorderedPair
maps
OrderedPair
source
target
Collection
+ type : CollectionType
+ name : Name
+ purpose : StructuredString
contains
OrderedTuple
source
target
UnorderedTuple
maps
Member
DDI_Identifiable
0..n
1..n
2..n
1..1

6.17 BaseObjects

A package is a administrative collection of classes in DDI. These are not namespaces.

6.17.1 Graph

6.18 Correspondences

A package is a administrative collection of classes in DDI. These are not namespaces.

6.18.1 Graph

6.19 DataCapture

A package is a administrative collection of classes in DDI. These are not namespaces.

6.19.1 Capture

A measurement that describes a means of capturing data. This class can be extended to account for different specific means. Use a specific instantiation of a Capture to describe a means of capturing a measurement.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
analysisUnit	ExternalControlledVocabularyEntry	0..n
displayLabel	DisplayLabel	0..n
measurementName	Name	0..n
purpose	StructuredString	0..1
source	ExternalControlledVocabularyEntry	0..1
usage	StructuredString	0..1

analysisUnit

Identifies the unit being analyzed such as a Person, Housing Unit, Enterprise, etc.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

measurementName

A name for the measurement. A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

A description of the purpose or use of the Measurement. May be expressed in multiple languages and supports the use of structured content.

source

The source of a capture structure defined briefly; typically using an external controlled vocabulary

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
hasConcept	Concept	0..n
hasExternalAid	ExternalAid	0..n
hasInstruction	Instruction	0..n
hasResponseDomain	ResponseDomain	0..1
intendedRepresentation	RepresentedVariable	0..1

hasConcept

Capture has a Concept

hasExternalAid

Capture has an External Aid

hasInstruction

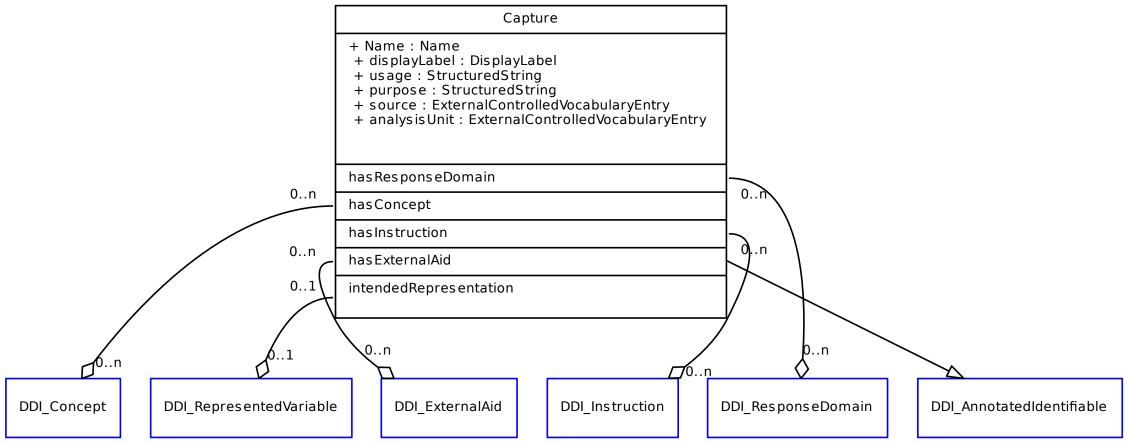
Capture has an Instruction

hasResponseDomain

intendedRepresentation

Associates the InstatiatedMeasure and the InstantiatedQuestion with the RepresentedVariable they are intended to populate.

Graph



6.19.2 ConceptualInstrument

Design plan for creating a data capture tool.

Extends

Workflow

Properties

Name	Type	Cardinality
description	StructuredString	0..1
displayLabel	DisplayLabel	0..n
hasAnnotation	Annotation	0..1
name	Name	0..n
usage	StructuredString	0..1

description

A description of the purpose or use of the Member. May be expressed in multiple languages and supports the use of structured content.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

hasAnnotation

Allows for complete annotation of the Conceptual Instrument

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

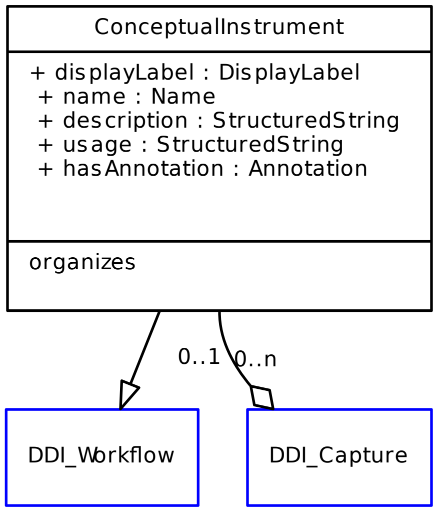
Relationships

Name	Type	Cardinality
organizes	Capture	0..1

organizes

Refers to the RepresentedMeasures and RepresentedQuestions used by the ConceptualInstrument.

Graph


ConceptualInstrument
+ displayLabel : DisplayLabel
+ name : Name
+ description : StructuredString
+ usage : StructuredString
+ hasAnnotation : Annotation
organizes
0..1
0..n
DDI_Workflow
DDI_Capture

6.19.3 ExternalAid

Any external material used in an instrument that aids or facilitates data capture, or that is presented to a respondent and about which measurements are made.

Extends

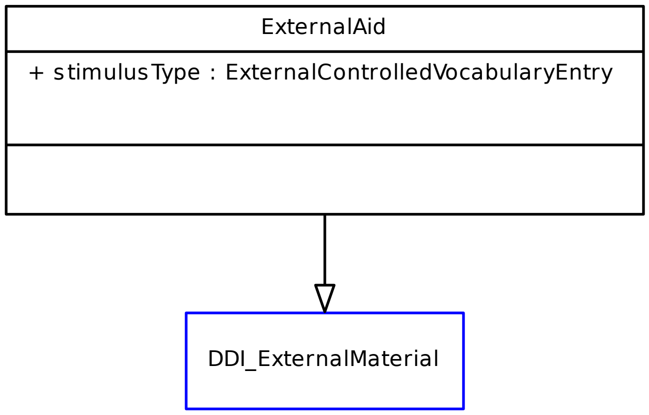
ExternalMaterial

Properties

Name	Type	Cardinality
stimulusType	ExternalControlledVocabularyEntry	0..1

stimulusType

Graph



6.19.4 ImplementedInstrument

ImplementedInstruments are mode and/or unit specific.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n
externalInstrumentLocation	anyURI	0..n
name	Name	0..n
typeOfInstrument	ExternalControlledVocabularyEntry	0..1
usage	StructuredString	0..n

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

externalInstrumentLocation

A reference to an external representation of the the data collection instrument, such as an image of a questionnaire or programming script.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

typeOfInstrument

Specification of the type of instrument according to the classification system of the documentor.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

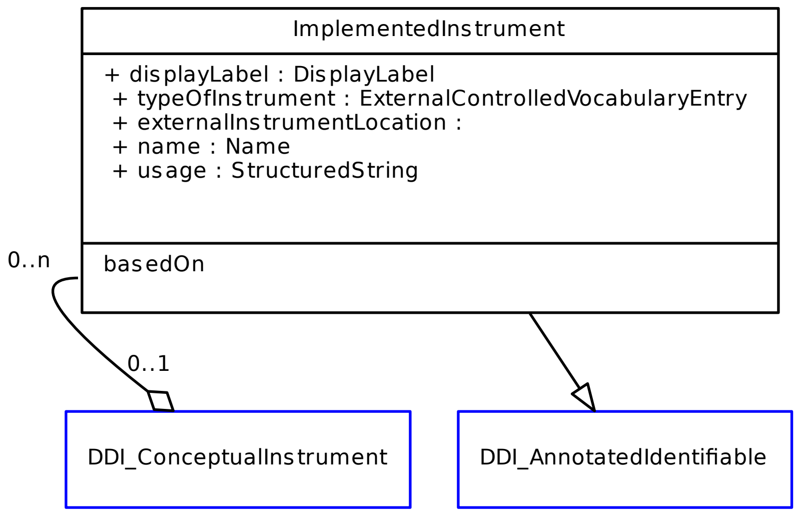
Relationships

Name	Type	Cardinality
basedOn	ConceptualInstrument	0..n

basedOn

The ConceptualInstrument which informs the design of the ImplementedInstrument.

Graph


ImplementedInstrument
+ displayLabel : DisplayLabel
+ typeOfInstrument : ExternalControlledVocabularyEntry
+ externalInstrumentLocation :
+ name : Name
+ usage : StructuredString
basedOn
0..n
0..1
DDI_ConceptualInstrument
DDI_AnnotatedIdentifiable

6.19.5 InstanceMeasurement

An instance measurement instantiates a represented measurement, so that it can be used as an Act in the Process Steps that define a data capture process.

Extends

InstrumentComponent

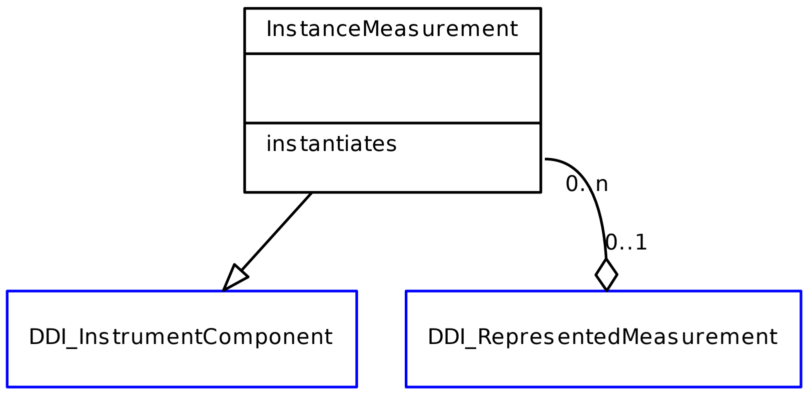
Relationships

Name	Type	Cardinality
instantiates	RepresentedMeasurement	0..n

instantiates

The measurement being instantiated in the data collection process

Graph



6.19.6 InstanceQuestion

An instance question is an instantiation of a represented question,to be used as an Act in the process steps that define a survey questionnaire.

Extends

InstrumentComponent

Properties

Name	Type	Cardinality
name	Name	0..n

name

The name of a question as used in an Instrument. Redefined to provide a more useful description. A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

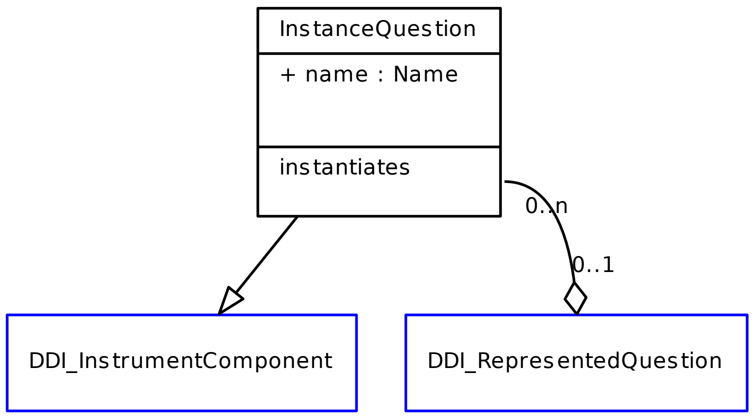
Relationships

Name	Type	Cardinality
instantiates	RepresentedQuestion	0..n

instantiates

The question being used.

Graph



6.19.7 Instruction

Provides the content and description of data capture instructions. Contains the "how to" information for administering an instrument.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n
instructionText	DynamicText	0..n
name	Name	0..n

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

instructionText

The content of the Instruction text provided using DynamicText. Note that when using Dynamic Text, the full InstructionText must be repeated for multi-language versions of the content. The InstructionText may also be repeated to provide a dynamic and plain text version of the instruction. This allows for accurate rendering of the instruction in a non-dynamic environment like print.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

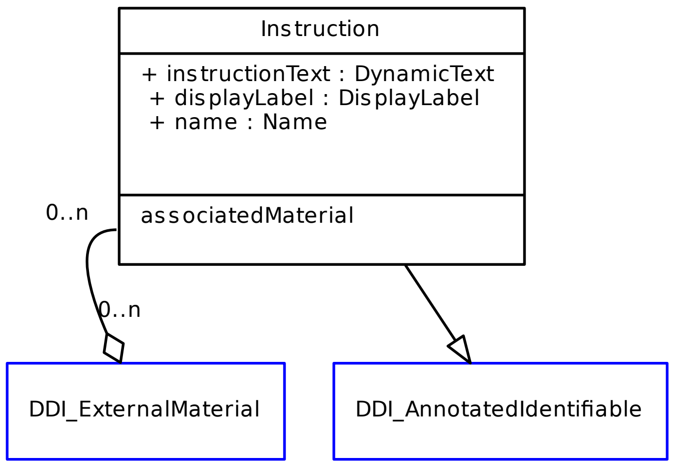
Relationships

Name	Type	Cardinality
associatedMaterial	ExternalMaterial	0..n

associatedMaterial

An image or other external material associated with the Instruction, located at the provided URN or URL.

Graph


Instruction
+ instructionText : DynamicText
+ displayLabel : DisplayLabel
+ name : Name
associatedMaterial
0..n
0..n
DDI_ExternalMaterial
DDI_AnnotatedIdentifiable

6.19.8 InstrumentCode

Extends

InstrumentComponent

Properties

Name	Type	Cardinality
commandCode	CommandCode	0..1
purposeOfCode	ExternalControlledVocabularyEntry	0..1

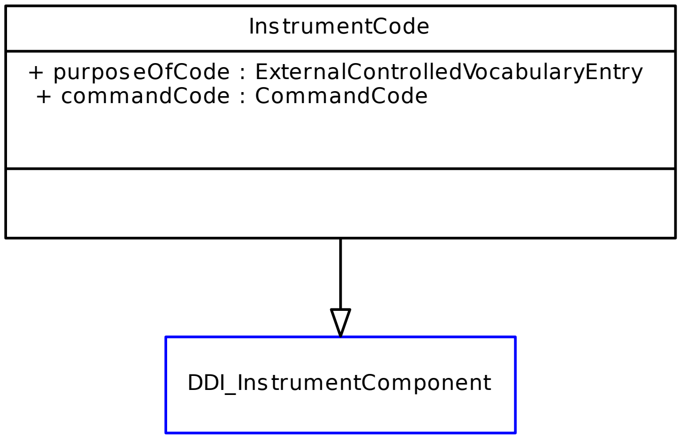
commandCode

Describes the code used to execute the command using the options of inline textual description, inline code, and/or an external file.

purposeOfCode

The purpose of the code (e.g., quality control, edit check, checksums, compute filler text, compute values for use in administering the instrument)

Graph


InstrumentCode
+ purposeOfCode : ExternalControlledVocabularyEntry
+ commandCode : CommandCode
DDI_InstrumentComponent

6.19.9 InstrumentComponent

InstrumentComponent is an abstract object which extends an Act (a type of Process Step). The purpose of Instrument-Component is to provide a common parent for Capture (e.g., Question, Measure), Statement, and Instructions.

Extends

Act

Relationships

Name	Type	Cardinality
hasExternalAids	ExternalAid	0..1
hasInstructions	Instruction	0..1

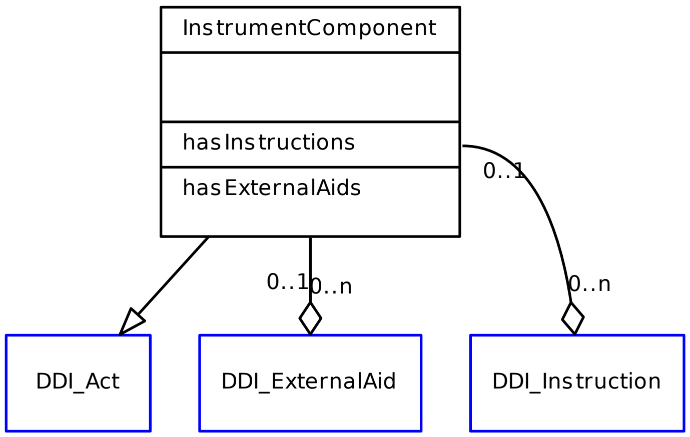
hasExternalAids

Any external object used to clarify, inform, or support the role of the instrument component

hasInstructions

An instrument compoment can have zero to many instructions

Graph



6.19.10 RepresentedMeasurement

The description of a reusable non-question measurement, that can be used as a template that describes the components of a measurement. A type code can be used to describe the type of measure that was used.

Extends

Capture

Properties

Name	Type	Cardinality
measurementType	ExternalControlledVocabularyEntry	0..1

measurementType

The type of measurement performed

‘

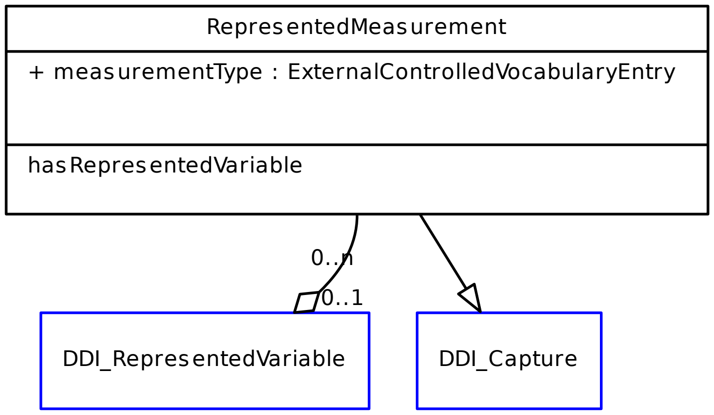
Relationships

Name	Type	Cardinality
hasRepresentedVariable	RepresentedVariable	0..n

hasRepresentedVariable

An optional link to a represented variable which can be used by each instance variable created by a use of this measure.

Graph



6.19.11 RepresentedQuestion

The description of a reusable question, that can be used as a template that describes the components of a question

Extends

Capture

Properties

Name	Type	Cardinality
estimatedResponseTimeInSeconds	Real	0..1
questionIntent	StructuredString	0..1
questionText	DynamicText	0..n

estimatedResponseTimeInSeconds

An estimation of the number of seconds required to respond to the question. Used for estimating overall questionnaire completion time.

questionIntent

The purpose or intent of the question.

questionText

The text of the question which may be literal or dynamic (altered to personalize the question text) in terms of content.
‘

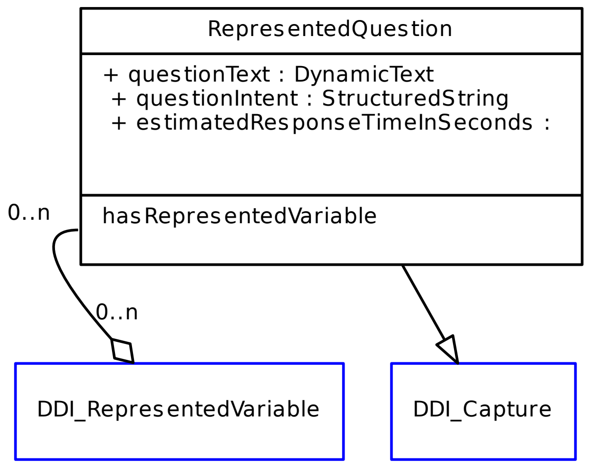
Relationships

Name	Type	Cardinality
hasRepresentedVariable	RepresentedVariable	0..n

hasRepresentedVariable

An optional link to a represented variable which can be used by each instance variable created by a use of this question.

Graph


RepresentedQuestion
+ questionText : DynamicText
+ questionIntent : StructuredString
+ estimatedResponseTimeInSeconds :
hasRepresentedVariable
0..n
0..n
DDI_RepresentedVariable
DDI_Capture

6.19.12 ResponseDomain

The possible list of values that are allowed by a Capture.

Extends

AnnotatedIdentifiable

Relationships

Name	Type	Cardinality
takesSentinelResponsesFrom	SentinelValueDomain	0..n
takesSubstantiveResponsesFrom	SubstantiveValueDomain	0..n

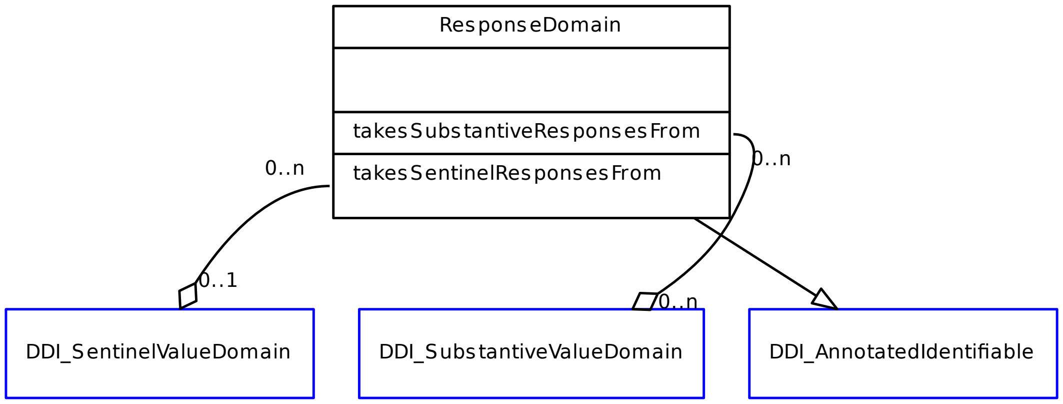
takesSentinelResponsesFrom

The sentinel values (e.g. missing codes) are drawn from this SubstantiveValueDomain

takesSubstantiveResponsesFrom

The values of interest are drawn from these SubstantiveValueDomains

Graph


ResponseDomain
takesSubstantiveResponsesFrom
takesSentinelResponsesFrom
0..n
0..1
0..n
0..n
DDI_SentinelValueDomain
DDI_SubstantiveValueDomain
DDI_AnnotatedIdentifiable

6.19.13 Statement

A Statement is human readable text or referred material.

Extends

InstrumentComponent

Properties

Name	Type	Cardinality
purposeOfStatement	ExternalControlledVocabularyEntry	0..1
statementText	DynamicText	0..n

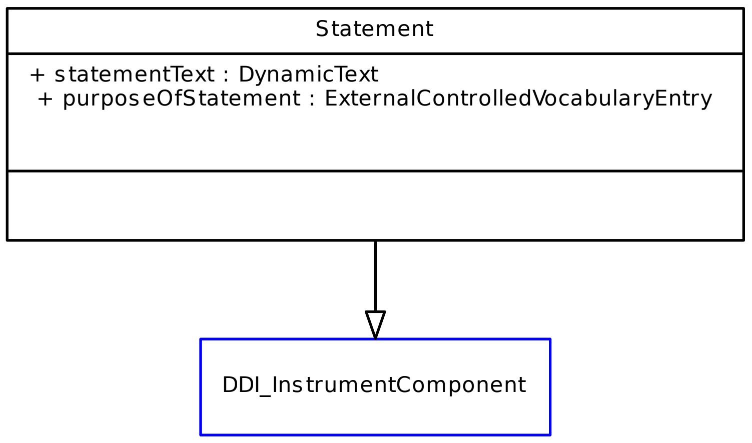
purposeOfStatement

Describes the use of the statement within the instrument. For example, a section divider, welcome statement, or other text.

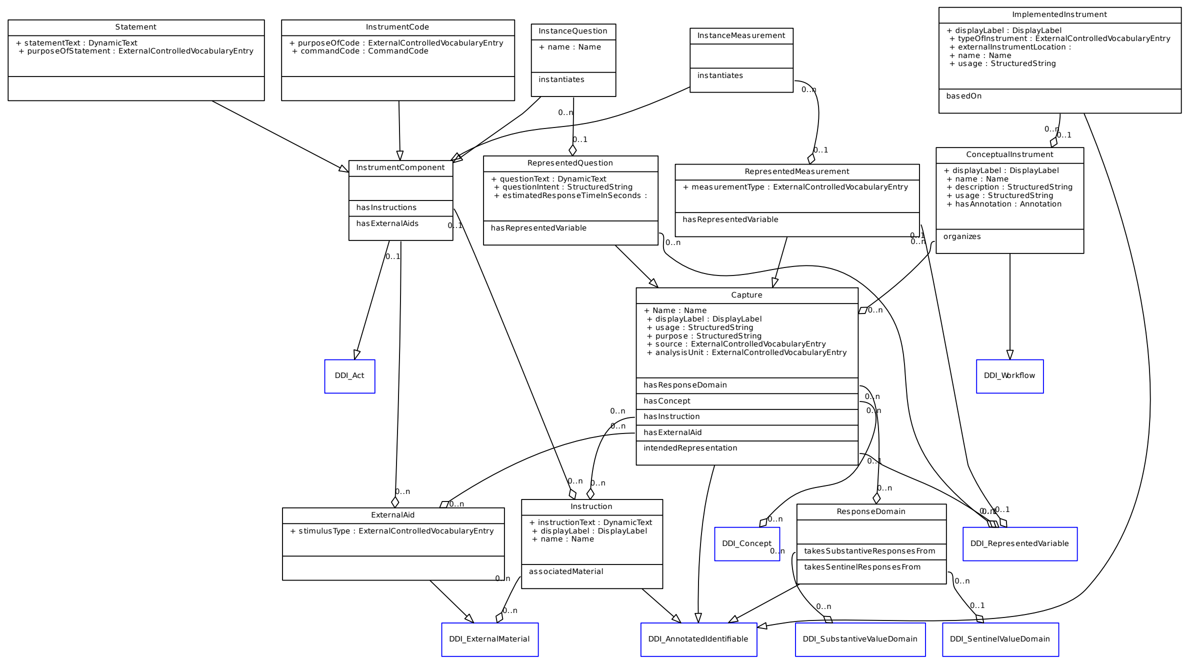
statementText

Structured human-readable text that allows for dynamic content. For example, the insertion of a name or gender specific pronoun. Repeat ONLY for capturing the same content in multiple languages.

Graph


Statement
+ statementText : DynamicText
+ purposeOfStatement : ExternalControlledVocabularyEntry
DDI_InstrumentComponent

6.19.14 Graph


Statement
+ statementText : DynamicText
+ purposeOfStatement : ExternalControlledVocabularyEntry
InstrumentCode
+ purposeOfCode : ExternalControlledVocabularyEntry
+ commandCode : CommandCode
InstanceQuestion
+ name : Name
instantiates
InstanceMeasurement
instantiates
ImplementedInstrument
+ displayLabel : DisplayLabel
+ typeOfInstrument : ExternalControlledVocabularyEntry
+ externalInstrumentLocation :
+ name : Name
+ usage : StructuredString
basedOn
InstrumentComponent
hasInstructions
hasExternalAids
RepresentedQuestion
+ questionText : DynamicText
+ questionIntent : StructuredString
+ estimatedResponseTimeInSeconds :
hasRepresentedVariable
RepresentedMeasurement
+ measurementType : ExternalControlledVocabularyEntry
hasRepresentedVariable
ConceptualInstrument
+ displayLabel : DisplayLabel
+ name : Name
+ description : StructuredString
+ usage : StructuredString
+ hasAnnotation : Annotation
organizes
Capture
+ Name : Name
+ displayLabel : DisplayLabel
+ usage : StructuredString
+ purpose : StructuredString
+ source : ExternalControlledVocabularyEntry
+ analysisUnit : ExternalControlledVocabularyEntry
hasResponseDomain
hasConcept
hasInstruction
hasExternalAid
intendedRepresentation
DDI_Act
DDI_Workflow
ExternalAid
+ stimulusType : ExternalControlledVocabularyEntry
Instruction
+ instructionText : DynamicText
+ displayLabel : DisplayLabel
+ name : Name
associatedMaterial
DDI_Concept
ResponseDomain
takesSubstantiveResponsesFrom
takesSentinelResponsesFrom
DDI_RepresentedVariable
DDI_ExternalMaterial
DDI_AnnotatedIdentifiable
DDI_SubstantiveValueDomain
DDI_SentinelValueDomain
0..n
0..1

6.20 Agents

A package is a administrative collection of classes in DDI. These are not namespaces.

6.20.1 Agent

An actor that performs a role in relation to a process.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
agentId	AgentId	0..n
image	PrivateImage	0..n
purpose	StructuredString	0..1

agentId

An identifier within a specified system for specifying an agent

image

References an image using the standard Image description. In addition to the standard attributes provides an effective date (period), the type of image, and a privacy ranking.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

‘

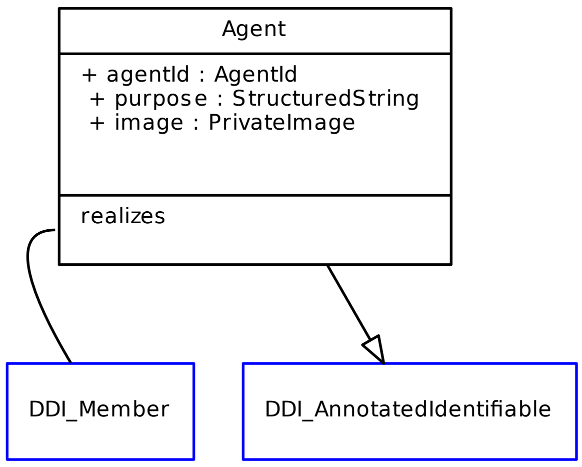
Relationships

Name	Type	Cardinality
realizes	Member	0..n

realizes

An Agent can perform the role of a Member in an Agent Registry

Graph



6.20.2 AgentBinary

This abstract structure is used as the extension base for all Binary Relations between two agents. It allows new Binary Relations to be created without changing the collection Agent Relationships in order to accommodate them. As an extension base it provides a consistent set of properties to all Binary Relations used between Agents.

Extends

Identifiable

Properties

Name	Type	Cardinality
effectiveDates	DateRange	0..1
privacy	ExternalControlledVocabularyEntry	0..1
purpose	StructuredString	0..1
typeOfRelationship	ExternalControlledVocabularyEntry	0..1

effectiveDates

The effective start and end date of the relationship

privacy

Define the level of privacy regarding this relationship. Supports the use of a controlled vocabulary.

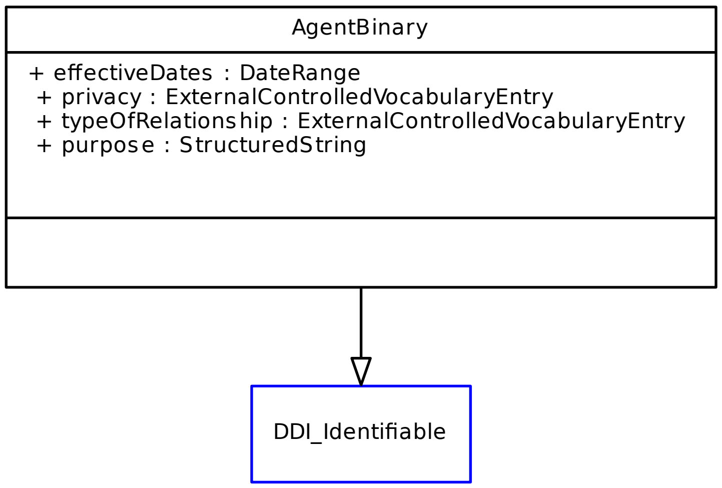
purpose

Explanation of the intent of creating the relation. Supports the use of multiple languages and structured text.

typeOfRelationship

Define the specific nature of the relationship if not captured within the semantics of the specific relation type such as simple similarity or Parent/Child relationship. Supports the use of a controlled vocabulary

Graph


AgentBinary
+ effectiveDates : DateRange
+ privacy : ExternalControlledVocabularyEntry
+ typeOfRelationship : ExternalControlledVocabularyEntry
+ purpose : StructuredString
DDI_Identifiable

6.20.3 AgentHierarchy

A set of relationships between pairs of Agents that is hierarchical (Parent/Child) in nature. Uses the pattern for Immediate Precedence Relation where the relation over members in the Agent Registry is characterized by the following: Reflexivity=Anti_Reflexive; Symmetry=Anti_Symmedtric; Transitivity=Anti_Transitive. It must contain like items (Agents).

Extends

AgentBinary

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1

reflexivity

Fixed to Anti_Reflexive

semantics

Controlled vocabulary for the ImmediatePrecedenceRelation semantics. It should contain, at least, the following: Parent_Of, Child_Of, Next, Previous, Instance_Of, Temporal_Meets.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Anti_Transitive

‘

Relationships

Name	Type	Cardinality
contains	AgentHierarchyPair	0..n
realizes	ImmediatePrecedenceRelation	0..n
structures	AgentListing	0..n

contains

Pair of Agents whose relationship is hierarchical in nature.

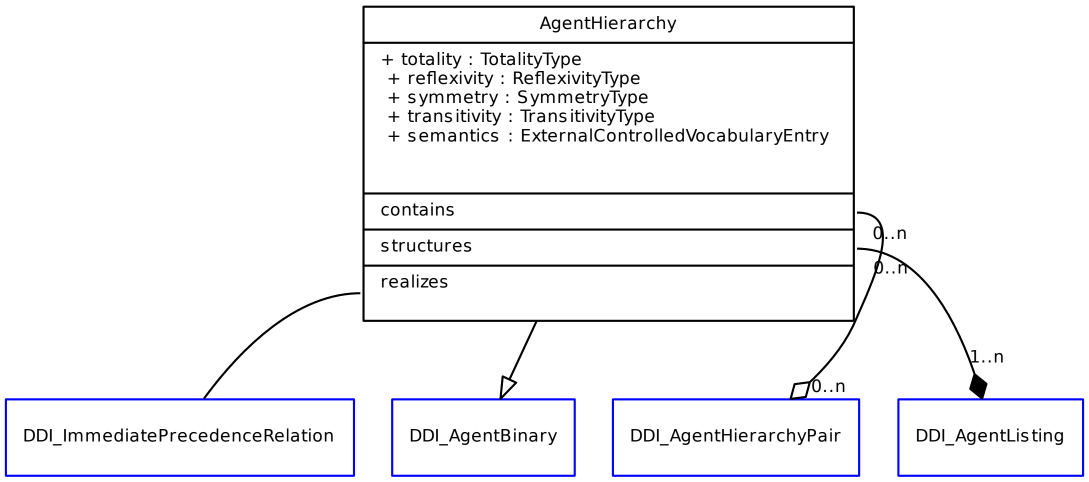
realizes

Uses the pattern of Immediate Precedence Relation

structures

The Agent Listing or Listings whose Agents are grouped by pairs to support a variety of structures

Graph


AgentHierarchy
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
contains
structures
realizes
0..n
0..n
1..n
0..n
DDI_ImmediatePrecedenceRelation
DDI_AgentBinary
DDI_AgentHierarchyPair
DDI_AgentListing

6.20.4 AgentHierarchyPair

Used to define a pair of agents in a hierarchical structure where the source is the parent and the target is the child.

Extends

Identifiable

Properties

Name	Type	Cardinality
effectiveDates	DateRange	0..1

effectiveDates

The effective dates of the relation. A structured DateRange with start and end Date (both with the structure of Date and supporting the use of ISO and non-ISO date structures); Use to relate a period with a start and end date.

‘

Relationships

Name	Type	Cardinality
realizes	OrderedPair	0..n
source	Agent	0..n
target	Agent	0..n

realizes

Uses the pattern of an OrderedPair

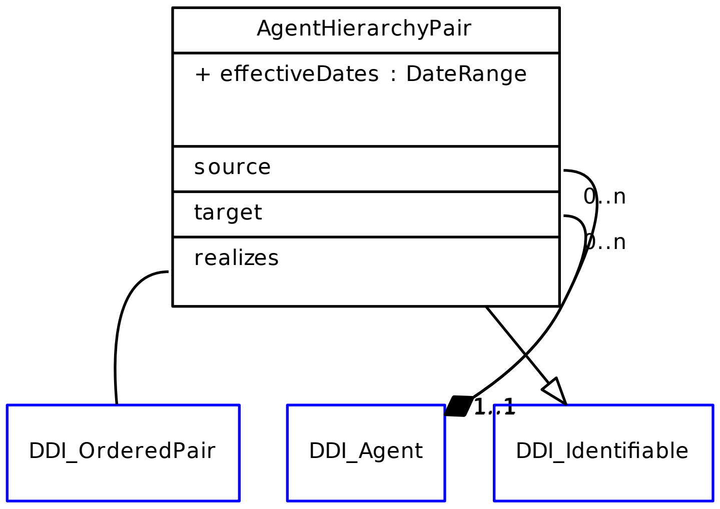
source

The Parent agent in the hierarchical relationship

target

The Child agent in the hierarchical relationship

Graph



6.20.5 AgentListing

A listing of Agents of any type

Extends

Identifiable

Properties

Name	Type	Cardinality
maintainer	AgentAssociation	0..1
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

maintainer

The agent responsible for maintaining the Agent Listing

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of the Agent Listing. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
contains	Agent	0..n
realizes	Collection	0..n

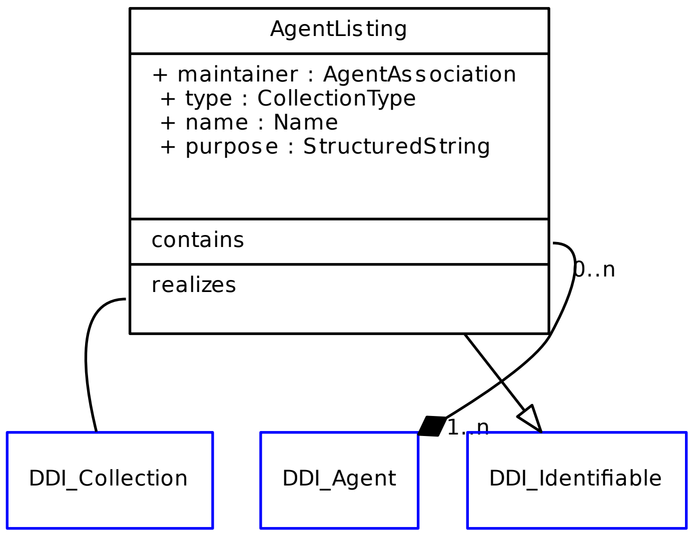
contains

The Agents contained in the Listing

realizes

Uses the pattern Collection

Graph



6.20.6 AgentRegistry

A collection of Agent Listings and/or AgentRelationships describing individual agents and their relationships to each other over time.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
purpose	StructuredString	0..1
usage	StructuredString	0..1
validDates	DateRange	0..1

purpose

Explanation of the intent of creating and maintaining this registry. Supports the use of multiple languages and structured text.

usage

Explanation of the ways in which the contents of the registry, Agent Listing and Agent Relationships can be employed. Supports the use of multiple languages and structured text.

validDates

A structured DateRange with start and end Date (both with the structure of Date and supporting the use of ISO and non-ISO date structures); Use to relate a period with a start and end date.

‘

Relationships

Name	Type	Cardinality
hasListing	AgentListing	0..n
hasRelationships	AgentRelationships	0..n
realizes	Collection	0..n

hasListing

Specialization of the contains relation in Collection. Identifies the specific Agent Listings defined in the Registry.

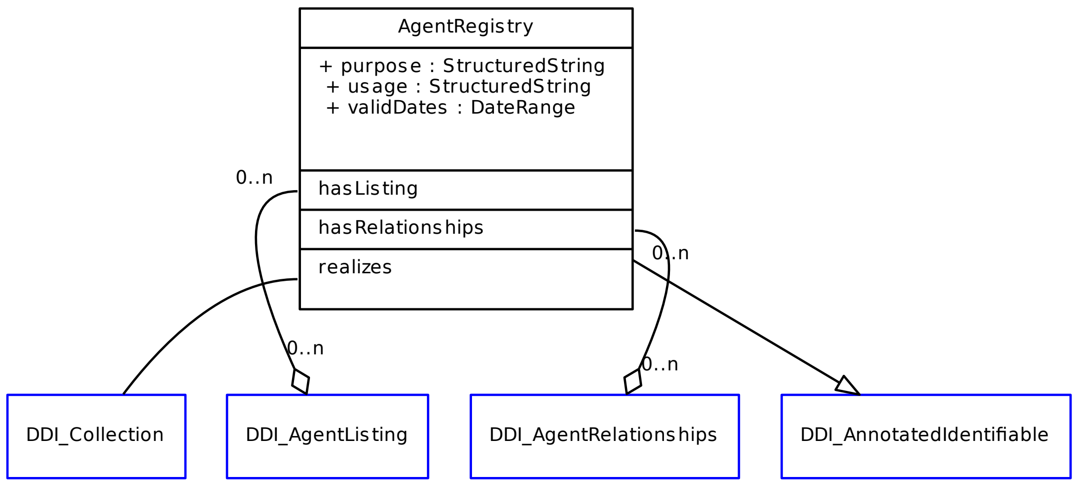
hasRelationships

Specialization of the contains relation in Collection. Identifies the specific Agent Relationships defined in the Registry.

realizes

Uses the pattern of Collection

Graph



6.20.7 AgentRelationships

A collection of relationships between pairs of Agents.

‘

Properties

Name	Type	Cardinality
maintainer	AgentAssociation	0..1
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

maintainer

The agent who maintains the relationship information

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of the collection of Agent Relationships. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality
contains	AgentBinary	0..n
realizes	Collection	0..n

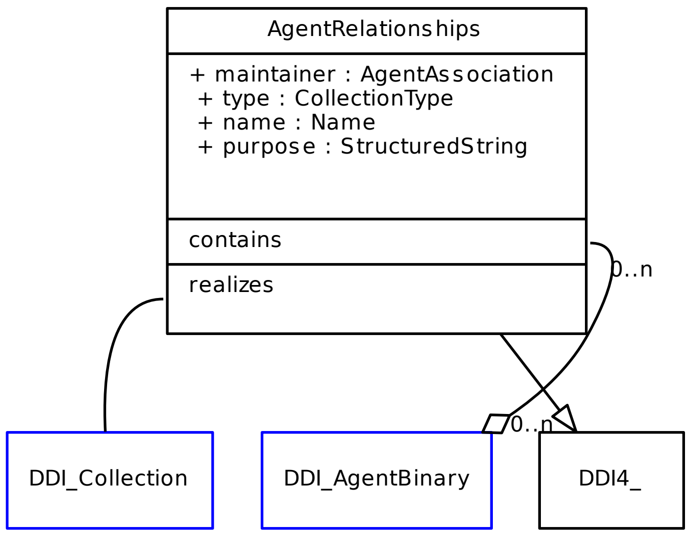
contains

Agent Hierarchy or Agent Similarity sets described in the Agent Relationship

realizes

Uses the pattern for a Collection

Graph



6.20.8 AgentSimilarity

A specific type of Agent Binary relationship that is used when the intent is to express the similarity between a pair of agents. Uses the pattern for EquivalenceRelation. Each pair must contain like items. The relation of the members in the collection is characterized by the following assignments: Reflexivity=Reflexive; Symmetry=Semetric; Transitivity=Transitive.

Extends

AgentBinary

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1

reflexivity

Fixed to Reflexive

semantics

Controlled vocabulary for the equivalence relation semantics. It should contain, at least, the following: Same_As, Similar_To, Congruent_To, Compatible_With, Equidistant_From, Temporal_Equals.

symmetry

Fixed to Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown

transitivity

Fixed to Transitive

‘

Relationships

Name	Type	Cardinality
contains	AgentSimilarityPair	0..n
realizes	EquivalenceRelation	0..n
structures	AgentListing	0..n

contains

Unordered pairs of Agents

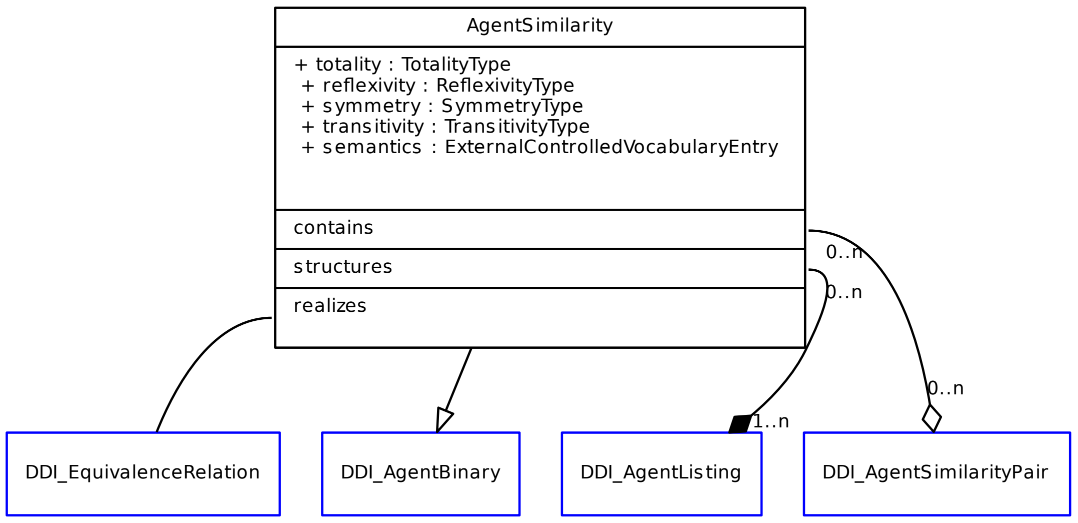
realizes

Reflects the pattern of EquivalenceRelation

structures

AgentListing or Listings whose Agents are grouped by pairs to support a variety of structures.

Graph


AgentSimilarity
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
contains
structures
realizes
0..n
0..n
0..n
1..n
DDI_EquivalenceRelation
DDI_AgentBinary
DDI_AgentListing
DDI_AgentSimilarityPair

6.20.9 AgentSimilarityPair

Describes simple pair of similar agents of any type (i.e. Organization, Individual, Machine, etc.). Uses the pattern for an UnorderedPair.

Extends

Identifiable

Properties

Name	Type	Cardinality
effectiveDates	DateRange	0..1

effectiveDates

The effective dates of the relation. A structured DateRange with start and end Date (both with the structure of Date and supporting the use of ISO and non-ISO date structures); Use to relate a period with a start and end date.

‘

Relationships

Name	Type	Cardinality
maps	Agent	0..n
realizes	UnorderedPair	0..n

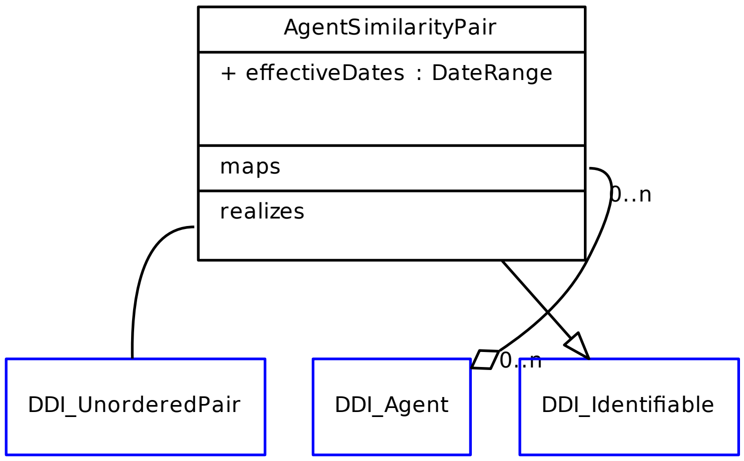
maps

Two Agents that are similar within the context of the type of relationship.

realizes

Reflects the pattern of an UnorderedPair

Graph



6.20.10 Individual

A person who acts, or is designated to act towards a specific purpose.

Extends

Agent

Properties

Name	Type	Cardinality
contactInformation	ContactInformation	0..1
ddiId	String	0..n
individualName	IndividualName	0..n

contactInformation

Contact information for the individual including location specification, address, URL, phone numbers, and other means of communication access. Sets of information can be repeated and date-stamped.

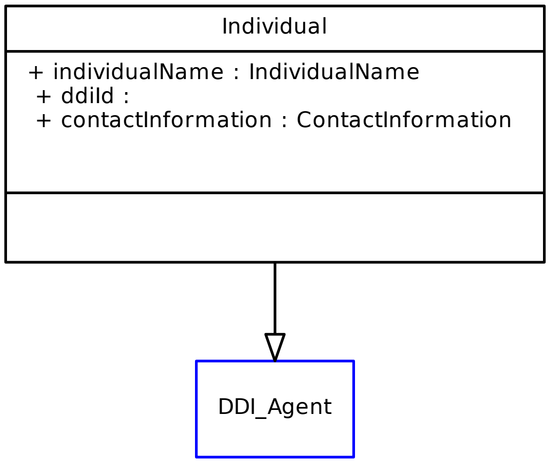
ddiId

The agency identifier of the individual according to the DDI Alliance agent registry.

individualName

The name of an individual broken out into its component parts of prefix, first/given name, middle name, last/family/surname, and suffix.

Graph


Individual
+ individualName : IndividualName
+ ddiId :
+ contactInformation : ContactInformation
DDI_Agent

6.20.11 Machine

Mechanism or computer program used to implement a process.

Extends

Agent

Properties

Name	Type	Cardinality
accessLocation	AccessLocation	0..1
function	ExternalControlledVocabularyEntry	0..n
interface	ExternalControlledVocabularyEntry	0..n
name	Name	0..n
ownerOperatorContact	ContactInformation	0..1
typeOfMachine	ExternalControlledVocabularyEntry	0..1

accessLocation

The locations where the machine can be access

function

The function of the machine

interface

Specified the machine interface. Supports the use of a controlled vocabulary.

name

The name of the machine. A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

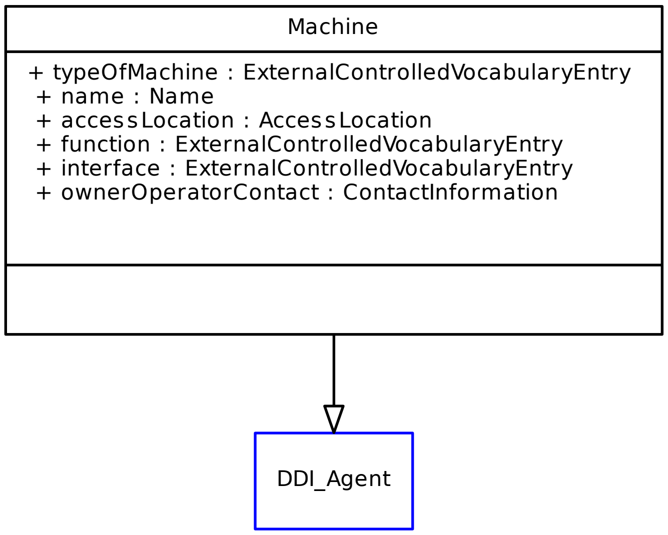
ownerOperatorContact

Contact information for the owner/operator including location specification, address, URL, phone numbers, and other means of communication access. Sets of information can be repeated and date-stamped.

typeOfMachine

The kind of machine used - software, web service, physical machine, from a controlled vocabulary

Graph


Machine
+ typeOfMachine : ExternalControlledVocabularyEntry
+ name : Name
+ accessLocation : AccessLocation
+ function : ExternalControlledVocabularyEntry
+ interface : ExternalControlledVocabularyEntry
+ ownerOperatorContact : ContactInformation
DDI_Agent

6.20.12 Organization

A framework of authority designated to act toward some purpose.

Extends

Agent

Properties

Name	Type	Cardinality
contactInformation	ContactInformation	0..1
ddiId	String	0..n
organizationName	OrganizationName	1..n

contactInformation

Contact information for the organization including location specification, address, URL, phone numbers, and other means of communication access. Sets of information can be repeated and date-stamped.

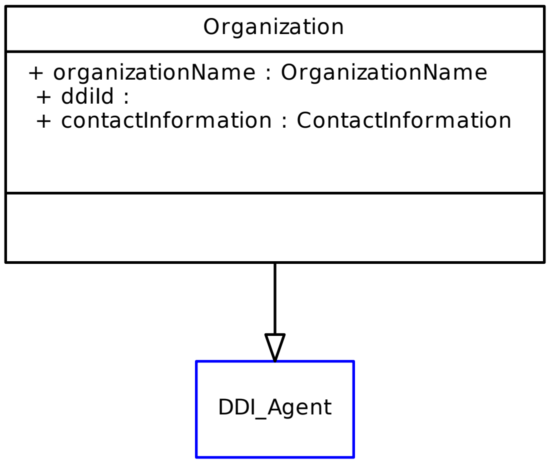
ddiId

The agency identifier of the organization as registered at the DDI Alliance register.

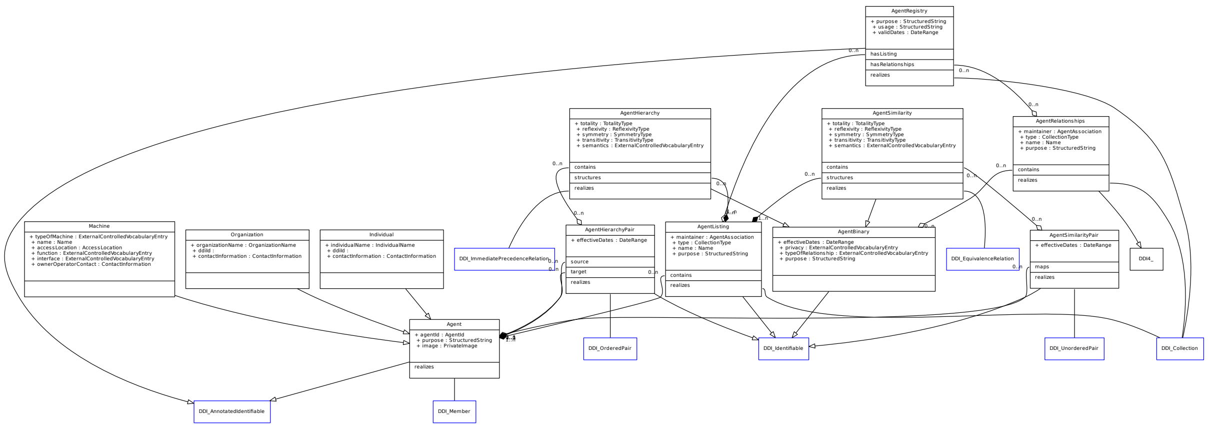
organizationName

Names by which the organization is known.

Graph


Organization
+ organizationName : OrganizationName
+ ddiId :
+ contactInformation : ContactInformation
DDI_Agent

6.20.13 Graph


AgentRegistry
+ purpose : StructuredString
+ usage : StructuredString
+ validDates : DateRange
hasListing
hasRelationships
realizes
AgentHierarchy
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
contains
structures
realizes
AgentSimilarity
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
contains
structures
realizes
AgentRelationships
+ maintainer : AgentAssociation
+ type : CollectionType
+ name : Name
+ purpose : StructuredString
contains
realizes
Machine
+ typeOfMachine : ExternalControlledVocabularyEntry
+ name : Name
+ accessLocation : AccessLocation
+ function : ExternalControlledVocabularyEntry
+ interface : ExternalControlledVocabularyEntry
+ ownerOperatorContact : ContactInformation
Organization
+ organizationName : OrganizationName
+ ddiId :
+ contactInformation : ContactInformation
Individual
+ individualName : IndividualName
+ ddiId :
+ contactInformation : ContactInformation
DDI_ImmediatePrecedenceRelation
AgentHierarchyPair
+ effectiveDates : DateRange
source
target
realizes
AgentListing
+ maintainer : AgentAssociation
+ type : CollectionType
+ name : Name
+ purpose : StructuredString
contains
realizes
AgentBinary
+ effectiveDates : DateRange
+ privacy : ExternalControlledVocabularyEntry
+ typeOfRelationship : ExternalControlledVocabularyEntry
+ purpose : StructuredString
DDI_EquivalenceRelation
AgentSimilarityPair
+ effectiveDates : DateRange
maps
realizes
DDI4_
Agent
+ agentId : AgentId
+ purpose : StructuredString
+ image : PrivateImage
realizes
DDI_OrderedPair
DDI_Identifiable
DDI_UnorderedPair
DDI_Collection
DDI_AnnotatedIdentifiable
DDI_Member
0..n
1..n

6.21 DDIUtility

A package is a administrative collection of classes in DDI. These are not namespaces.

6.21.1 Graph

6.22 DDIDocument

A package is a administrative collection of classes in DDI. These are not namespaces.

6.22.1 Graph

6.23 Identification

A package is a administrative collection of classes in DDI. These are not namespaces.

6.23.1 AnnotatedIdentifiable

Used to identify objects for purposes of internal and/or external referencing. Elements of this type are versioned. Provides identification and administrative metadata about the object. Adds optional annotation. Use this as the extension base for First Order Classes that contain intellectual content that needs to be discoverable in its own right.

Extends

Identifiable

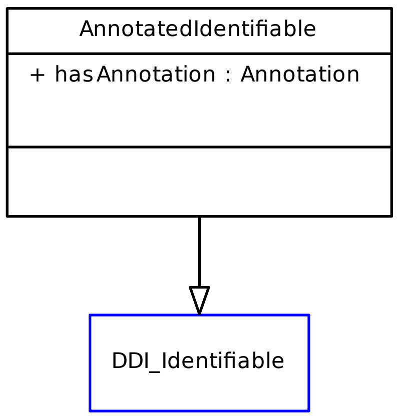
Properties

Name	Type	Cardinality
hasAnnotation	Annotation	0..1

hasAnnotation

Provides annotation information on the object to support citation and crediting of the creator(s) of the object.

Graph



6.23.2 Identifiable

Used to identify objects for purposes of internal and/or external referencing. Elements of this type are versioned and provide administrative metadata properties. Use for First Order Classes whose content does not need to be discoverable in its own right but needs to be related to multiple classes.

‘

Properties

Name	Type	Cardinality
agency	String	1..1
basedOnObject	BasedOnObject	0..1
id	String	1..1
isPersistent	Boolean	1..1
isUniversallyUnique	Boolean	1..1
localId	LocalId	0..n
version	String	1..1
versionDate	IsoDate	0..1
versionRationale	String	0..1
versionResponsibility	String	0..1

agency

This is the registered agency code with optional sub-agencies separated by dots. For example, diw.soep, ucl.qss, abs.essg.

basedOnObject

The object/version that this object version is based on.

id

The ID of the object. This must conform to the allowed structure of the DDI Identifier and must be unique within the Agency.

isPersistent

Default value is false. Usually the content of the current version is allowed to change, for example as the contributor is working on the object contents. However, when isPersistent is true, it indicates the there will be no more changes to the current version.

isUniversallyUnique

Default value is false. Usually the combination of agency and id (ignoring different versions) is unique. If isUniversallyUnique is set to true, it indicates that the id itself is universally unique (unique across systems and/or agencies) and therefore the agency part is not required to ensure uniqueness.

localId

This is an identifier in a given local context that uniquely references an object, as opposed to the full ddi identifier which has an agency plus the id. For example, localId could be a variable name in a dataset.

version

The version number of the object. The version number is incremented whenever the non-administrative metadata contained by the object changes.

versionDate

The date and time the object was changed. Supports standard ISO date and datetime formats.

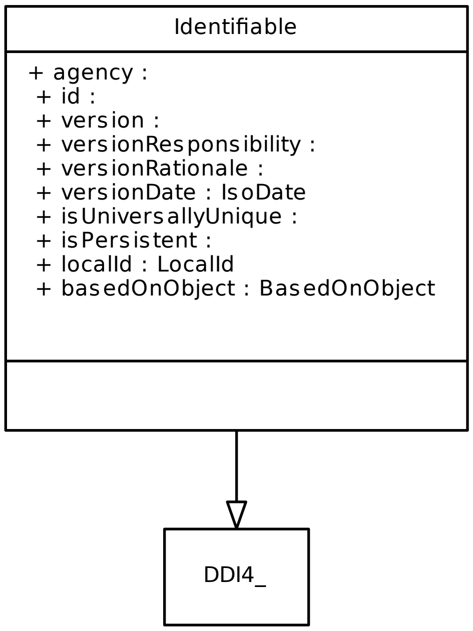
versionRationale

The reason for making this version of the object.

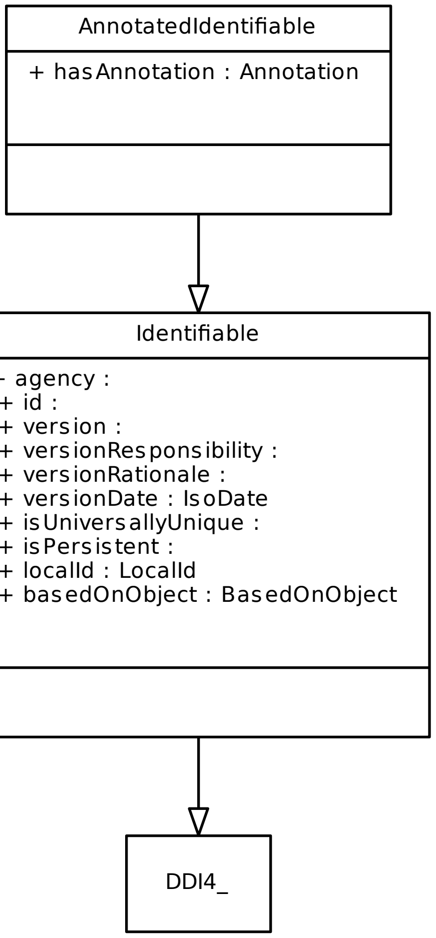
versionResponsibility

Contributor who has the ownership and responsibility for the current version.

Graph


Identifiable
+ agency :
+ id :
+ version :
+ versionResponsibility :
+ versionRationale :
+ versionDate : IsoDate
+ isUniversallyUnique :
+ isPersistent :
+ localId : LocalId
+ basedOnObject : BasedOnObject
DDI4_

6.23.3 Graph



6.24 ComplexProcess

A package is a administrative collection of classes in DDI. These are not namespaces.

6.24.1 ContainsIntervalRelation

Representation of the contains relation in Allen's interval algebra.

We say that an interval A contains another interval B if and only if A begins before B but finishes after it.

More precisely, A.start < B.start < B.end < A.end.

Instead of saying that A contains B we can also say that B is during A (converse).

Extends

TemporalIntervalRelation

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	1..1
symmetry	SymmetryType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

reflexivity

Fixed to Anti-Reflexive

semantics

Fixed to Temporal-Contains

symmetry

Fixed to Anti-Symmetric

transitivity

Fixed to Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	OrderedIntervalPair	0..n
realizes	StrictOrderRelation	0..n
structures	WorkflowSequence	0..n

contains

Pairs of Process Steps in the temporal relation.

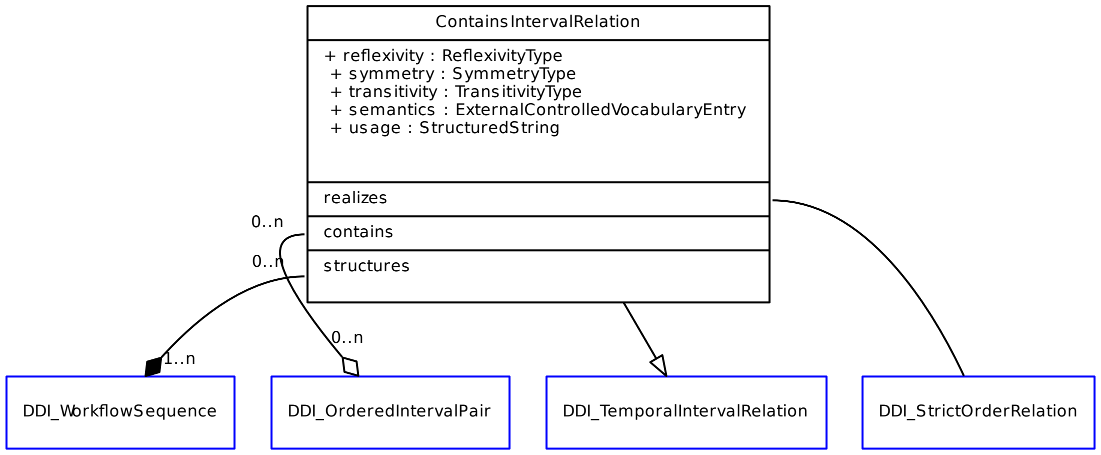
realizes

Class of the Collection pattern realized by this class.

structures

Sequence to which the Temporal Interval Relation is applied

Graph


ContainsIntervalRelation
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
+ usage : StructuredString
realizes
contains
structures
0..n
0..n
0..n
1..n
DDI_WorkflowSequence
DDI_OrderedIntervalPair
DDI_TemporalIntervalRelation
DDI_StrictOrderRelation

6.24.2 EqualsIntervalRelation

Representation of the equals relation in Allen's interval algebra.

We say that an interval A equals another interval B if and only if they both begin and finish at the same time.

More precisely, A.start = B.start < A.end = B.end.

Instead of saying that A equals B we can also say the B equals A (reflexive).

Extends

TemporalIntervalRelation

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	1..1
symmetry	SymmetryType	1..1
transitivity	TransitivityType	1..1

reflexivity

Fixed to Reflexive

semantics

Fixed to Temporal-Equals

symmetry

Fixed to Symmetric

transitivity

Fixed to Transitive

‘

Relationships

Name	Type	Cardinality
contains	UnorderedIntervalPair	0..n
realizes	EquivalenceRelation	0..n
structures	WorkflowSequence	0..n

contains

Pairs of Process Steps in the temporal relation.

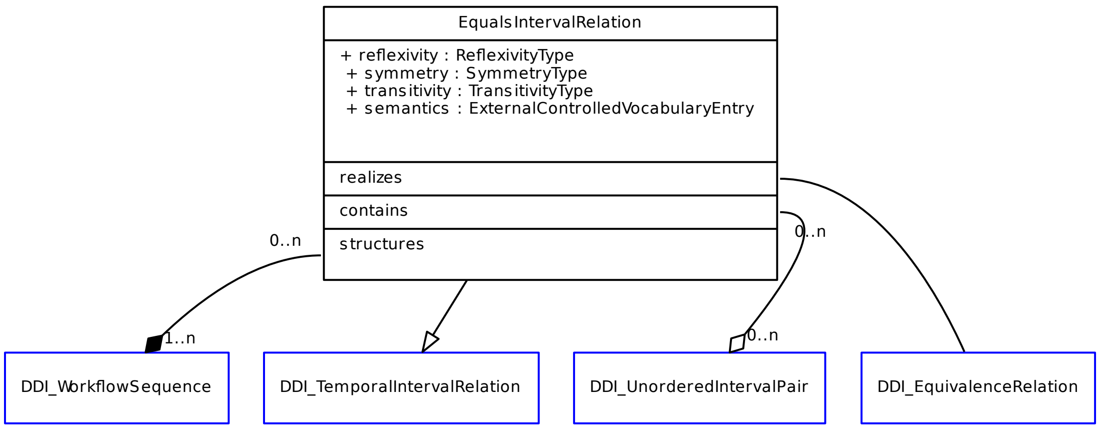
realizes

Class of the Collection pattern realized by this class.

structures

Sequence to which the Temporal Interval Relation is applied

Graph


EqualsIntervalRelation
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
realizes
contains
structures
0..n
1..n
0..n
0..n
DDI_WorkflowSequence
DDI_TemporalIntervalRelation
DDI_UnorderedIntervalPair
DDI_EquivalenceRelation

6.24.3 ExecutionPair

During execution ProcessSteps follow one another over time. This is also true when we parallelize because within each thread work flows. In every thread during execution we advance through a sequence of pairs. Each pair has a “prev” and a “cur”. In the course of execution “cur” becomes “prev” and a new “cur” comes along.

In this context the Execution Pair is the current “prev” and “cur”.

It is between the current “prev” and “cur” that messages flow. The output of prev can be mapped to the input of “cur”. This mapping is commonly referred to as a “binding” or in the terms of a collection an OrderedMemberCorrespondence.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
executionPairPattern	ExternalControlledVocabularyEntry	0..1

executionPairPattern

An execution pair conforms to one of three patterns. In one case the pairs are siblings in a sequence. In a second case a binding might obtain between the control construct that parents the sequence and the sequence. In a third case a binding might obtain between the last element of a sequence and the output of the construct construct that has parented the sequence

‘

Relationships

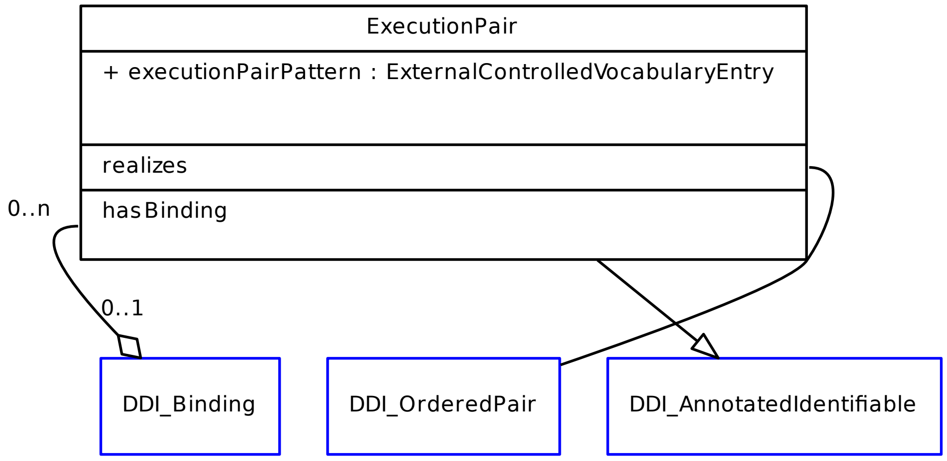
Name	Type	Cardinality
hasBinding	Binding	0..n
realizes	OrderedPair	0..n

hasBinding

realizes

Class of the Collection pattern realized by this class.

Graph


ExecutionPair
+ executionPairPattern : ExternalControlledVocabularyEntry
realizes
hasBinding
0..n
0..1
DDI_Binding
DDI_OrderedPair
DDI_AnnotatedIdentifiable

6.24.4 FinishesIntervalRelation

Representation of the finishes relation in Allen's interval algebra.

We say that an interval A finishes another interval B if and only if A begins after B but both finish at the same time.

More precisely, B.start < A.start < B.end = A.end.

Instead of saying that A finishes B we can also say that B is finished by A (converse)

Extends

TemporalIntervalRelation

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	1..1
symmetry	SymmetryType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

reflexivity

Fixed to Anti-Reflexive

semantics

Fixed to Temporal-Finishes

symmetry

Fixed to Anti-Symmetric

transitivity

Fixed to Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	OrderedIntervalPair	0..n
realizes	StrictOrderRelation	0..n
structures	WorkflowSequence	0..n

contains

Pairs of Process Steps in the temporal relation.

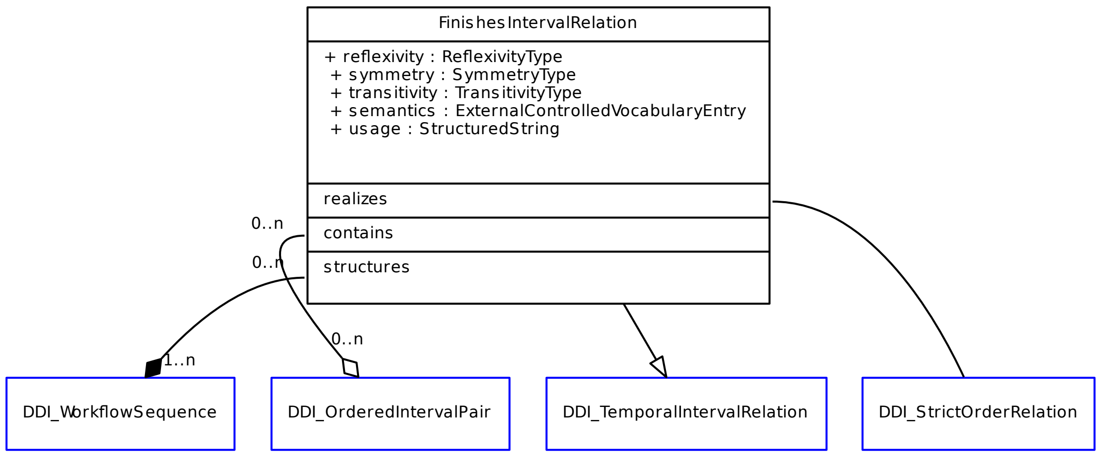
realizes

Class of the Collection pattern realized by this class.

structures

Sequence to which the Temporal Interval Relation is applied

Graph


FinishesIntervalRelation
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
+ usage : StructuredString
realizes
contains
structures
0..n
0..n
0..n
1..n
DDI_WorkflowSequence
DDI_OrderedIntervalPair
DDI_TemporalIntervalRelation
DDI_StrictOrderRelation

6.24.5 MeetsIntervalRelation

Representation of the meets relation in Allen's interval algebra.

We say that an interval A meets another interval B if and only if A finishes when B begins.

More precisely, A.ends = B.start.

Instead of saying that A meets B we can also say that B is met by A (converse).

Extends

TemporalIntervalRelation

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	1..1
symmetry	SymmetryType	1..1
transitivity	TransitivityType	1..1

reflexivity

Fixed to Anti-Reflexive

semantics

Fixed to Temporal-Meets

symmetry

Fixed to Anti-Symmetric

transitivity

Fixed to Anti-Transitive

‘

Relationships

Name	Type	Cardinality
contains	OrderedIntervalPair	0..n
realizes	ImmediatePrecedenceRelation	0..n
structures	WorkflowSequence	0..n

contains

Pairs of Process Steps in the temporal relation.

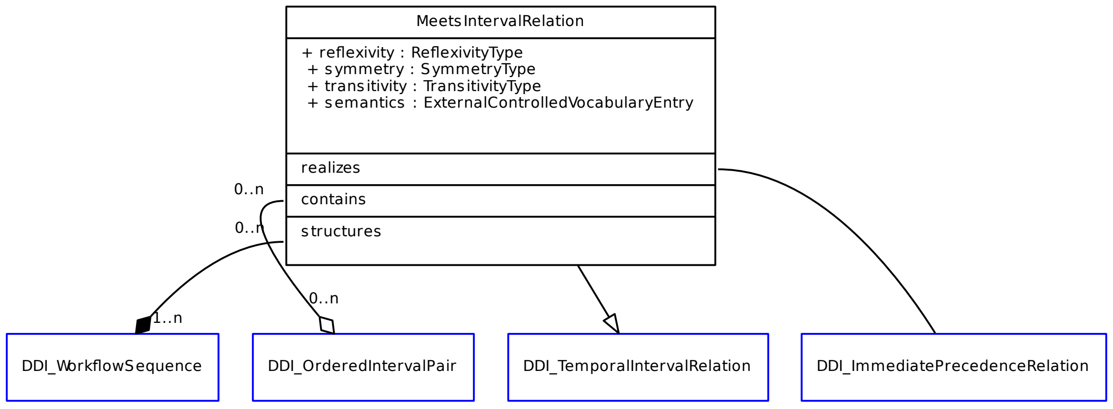
realizes

Class of the Collection pattern realized by this class.

structures

Sequence to which the Temporal Interval Relation is applied

Graph


MeetsIntervalRelation
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
realizes
contains
structures
0..n
0..n
0..n
1..n
DDI_WorkflowSequence
DDI_OrderedIntervalPair
DDI_TemporalIntervalRelation
DDI_ImmediatePrecedenceRelation

6.24.6 OrderedIntervalPair

Ordered pair of Process Steps in a Temporal Interval Relation.

Extends

Identifiable

Relationships

Name	Type	Cardinality
realizes	OrderedPair	0..n
source	WorkflowStep	0..n
target	WorkflowStep	0..n

realizes

Class of the Collection pattern realized by this class.

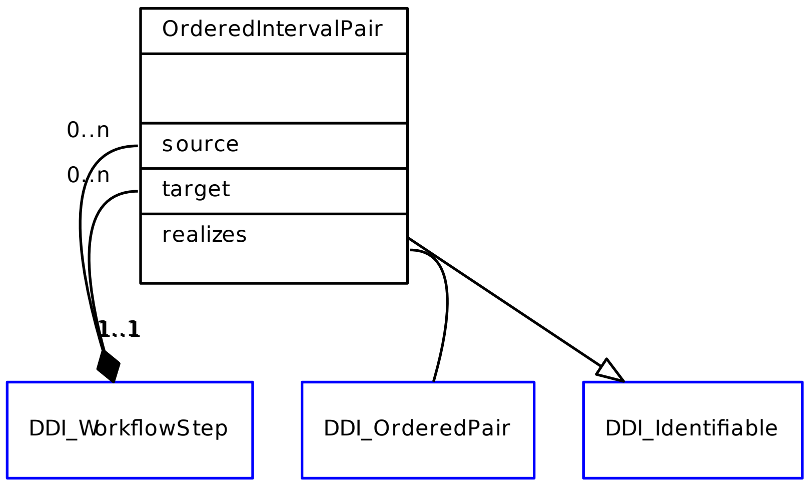
source

First Process Step in the Ordered Pair. Process Step that either starts or ends before the Process Step in the target.

target

Second Process Step in the Ordered Pair. Process Step that either starts or ends after the Process Step in the source.

Graph



6.24.7 OverlapsIntervalRelation

Representation of the overlaps relation in Allen's interval algebra.

We say that an interval A overlaps another interval B if and only if A begins before B but finishes during B.

More precisely, A.start < B.start < A.end < B.end.

Instead of saying that A overlaps B we can also say that B is overlapped by A (converse).

Extends

TemporalIntervalRelation

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..1
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	1..1
symmetry	SymmetryType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

displayLabel

A display label for the OverlapsIntervalRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

reflexivity

Fixed to Anti-Reflexive

semantics

Fixed to Temporal_Overlaps

symmetry

Fixed to Anti-Symmetric

transitivity

Fixed to Neither

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	OrderedIntervalPair	0..n
realizes	AcyclicPrecedenceRelation	0..n
structures	WorkflowSequence	0..n

contains

Pairs of Process Steps in the temporal relation.

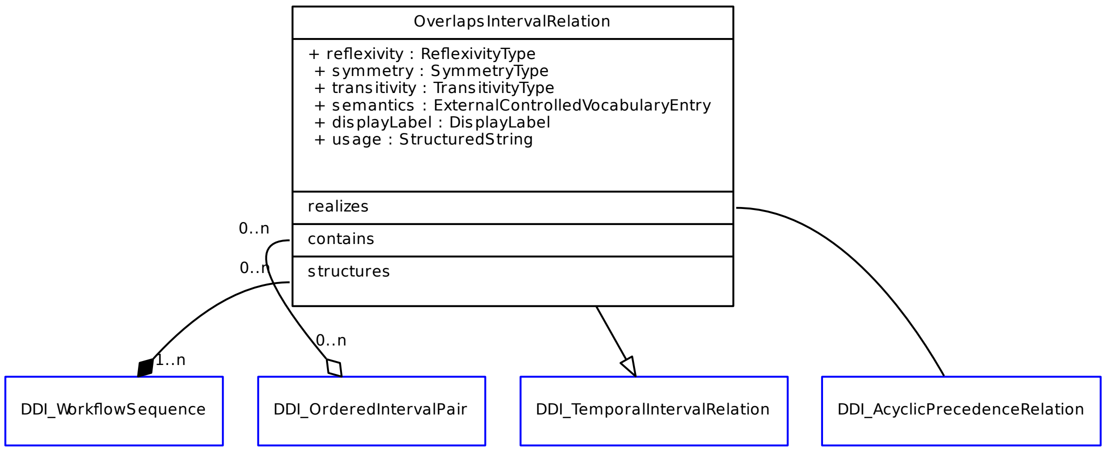
realizes

Class of the Collection pattern realized by this class.

structures

Sequence to which the Temporal Interval Relation is applied

Graph


OverlapsIntervalRelation
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
+ displayLabel : DisplayLabel
+ usage : StructuredString
realizes
contains
structures
0..n
0..n
0..n
1..n
DDI_WorkflowSequence
DDI_OrderedIntervalPair
DDI_TemporalIntervalRelation
DDI_AcyclicPrecedenceRelation

6.24.8 PrecedesIntervalRelation

Representation of the precedes relation in Allen's interval algebra.

We say that an interval A precedes another interval B if and only if A finishes before B begins.

More precisely, A.end < B.start.

Instead of saying that A precedes B we can also say that B is preceded by A (converse).

Extends

TemporalIntervalRelation

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	1..1
symmetry	SymmetryType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

reflexivity

Fixed to Anti-Reflexive

semantics

Fixed to Temporal-Precedes

symmetry

Fixed to Anti-Symmetric

transitivity

Fixed to Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	OrderedIntervalPair	0..n
realizes	StrictOrderRelation	0..n
structures	WorkflowSequence	0..n

contains

Pairs of Process Steps in the temporal relation.

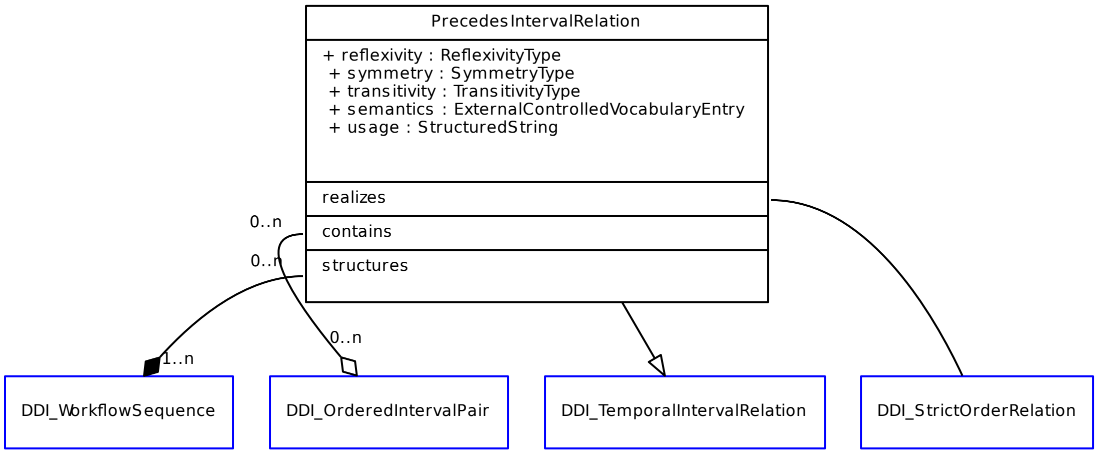
realizes

Class of the Collection pattern realized by this class.

structures

Sequence to which the Temporal Interval Relation is applied.

Graph


PrecedesIntervalRelation
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
+ usage : StructuredString
realizes
contains
structures
0..n
0..n
0..n
1..n
DDI_WorkflowSequence
DDI_OrderedIntervalPair
DDI_TemporalIntervalRelation
DDI_StrictOrderRelation

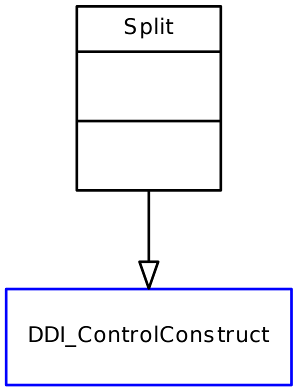
6.24.9 Split

The components of a Split process are a bag of process components to be executed concurrently. Split completes as soon as all of its component processes have been scheduled for execution

Extends

:ref:‘ControlConstruct

Graph



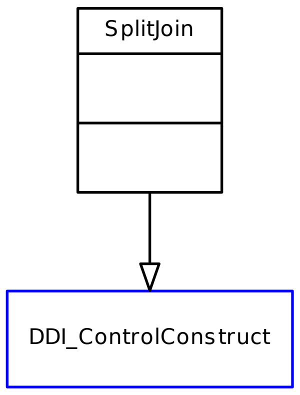
6.24.10 SplitJoin

Here the process consists of concurrent execution of a bunch of process components with barrier synchronization. That is, SplitJoin completes when all of its components processes have completed. With Split and SplitJoin, we can define processes that have partial synchronization (e.g., split all and join some sub-bag)

Extends

:ref:‘ControlConstruct

Graph



6.24.11 StartsIntervalRelation

Representation of the starts relation in Allen's interval algebra.

We say that an interval A starts another interval B if and only if they both start at the same time but A finishes first.

More precisely, A.start = B.start < A.end <B.end

Instead of saying that A starts B we can also say that B is started by A (converse).

Extends

TemporalIntervalRelation

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	1..1
symmetry	SymmetryType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

reflexivity

Fixed to Anti-Reflexive

semantics

Fixed to Temporal-Starts

symmetry

Fixed to Anti-Symmetric

transitivity

Fixed to Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality
contains	OrderedIntervalPair	0..n
realizes	StrictOrderRelation	0..n
structures	WorkflowSequence	0..n

contains

Pairs of Process Steps in the temporal relation.

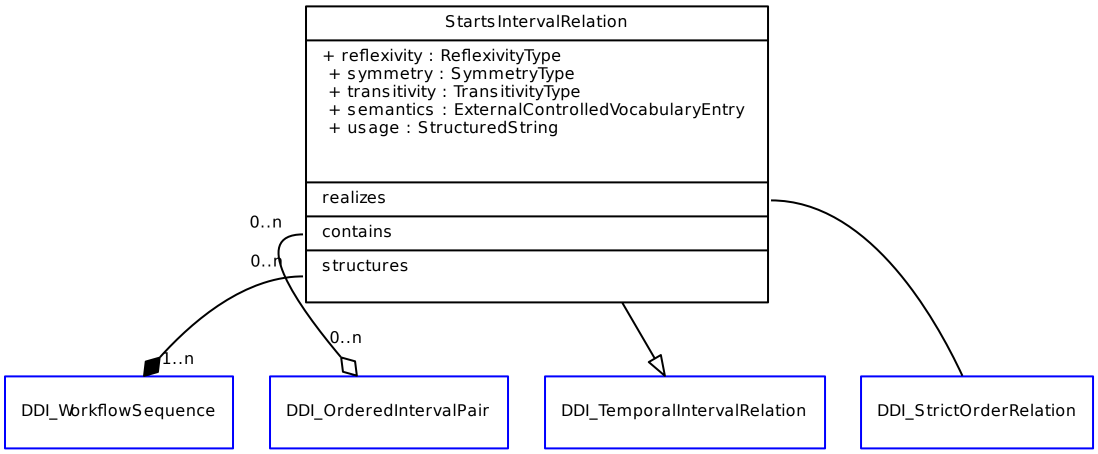
realizes

Class of the Collection pattern realized by this class.

structures

Sequence to which the Temporal Interval Relation is applied

Graph


StartsIntervalRelation
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
+ usage : StructuredString
realizes
contains
structures
0..n
0..n
0..n
1..n
DDI_WorkflowSequence
DDI_OrderedIntervalPair
DDI_TemporalIntervalRelation
DDI_StrictOrderRelation

6.24.12 StudyUnitControl

StudyUnitControl references a StudyUnitType. StudyUnitControl enables the representation of a research protocol at the StudyUnit level. [If the only purpose of the object is to carry a StudyUnit, it should be seriously considered for melting with another object. Modeling rules state that any object needs to have a reality of its own. Carrying another object is not a reality. If I misunderstood, please clarify the description instead]

Extends

Act

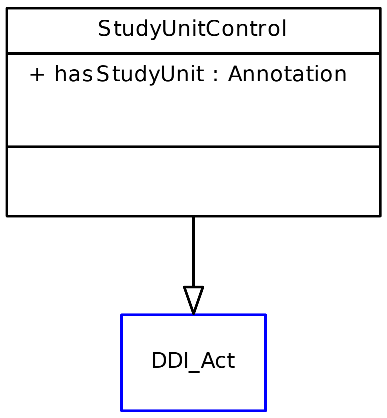
Properties

Name	Type	Cardinality
hasStudyUnit	Annotation	0..1

hasStudyUnit

Provides the Annotation of the Study Unit

Graph



6.24.13 TemporalIntervalRelation

Relation that specifies temporal constraints between pairs of Process Steps in a Sequence following Allen's interval algebra. A Sequence in the Process Model is a Control Construct in which the elements of the sequence (i.e. Process Steps) have positions on a timeline with respect to each other.

In Allen's interval algebra between Process Steps a and b there are seven possible relations. All but one are asymmetric and thus have a converse. The exception is the situation where two intervals are coextensive. Then the relation and its converse are the same, namely equal, which is an equivalence relation.

Extends

AnnotatedIdentifiable

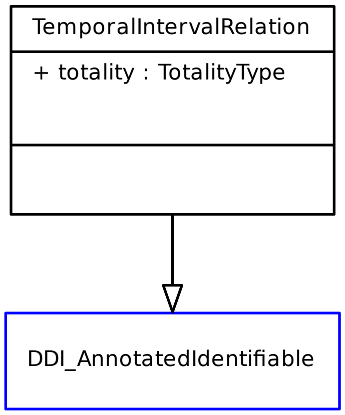
Properties

Name	Type	Cardinality
totality	TotalityType	1..1

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

Graph



6.24.14 TransformationControl

Provides an extensible framework for specific transformation objects.

Extends

Act

Properties

Name	Type	Cardinality
activityDescription	StructuredString	0..1

activityDescription

Describes the transformation activity

‘

Relationships

Name	Type	Cardinality
wasAssociatedWith	Agent	0..n
wasDerivedFrom	WorkflowStep	0..n
wasGeneratedBy	WorkflowStep	0..n

wasAssociatedWith

The agent an activity is associated with

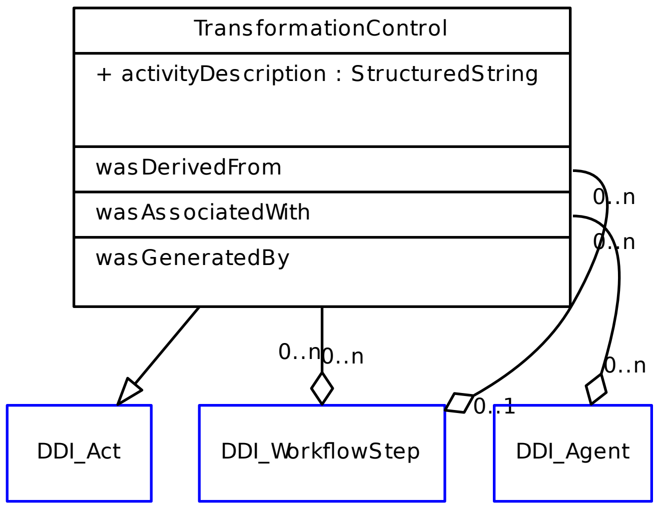
wasDerivedFrom

The entity an activity uses

wasGeneratedBy

The entity an activity generates

Graph



6.24.15 UnorderedIntervalPair

Unordered pair of Process Steps in a Temporal Interval Relation.

Extends

Identifiable

Relationships

Name	Type	Cardinality
maps	WorkflowStep	0..n
realizes	UnorderedPair	0..n

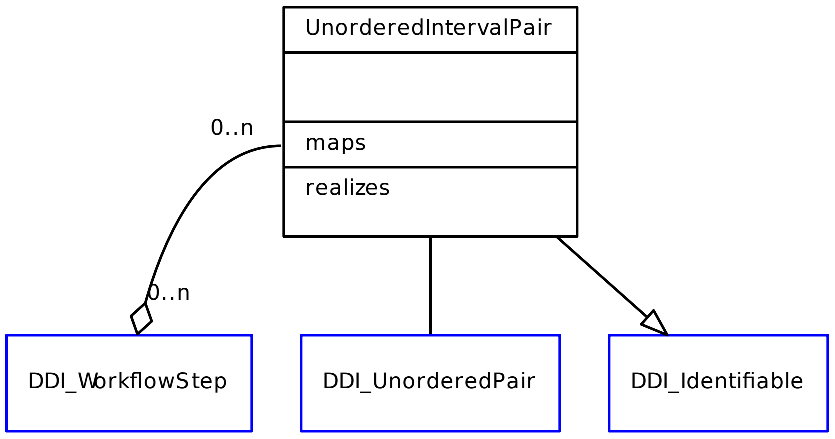
maps

Process Steps in the Unordered Pair. Process Steps that start and end at the same time

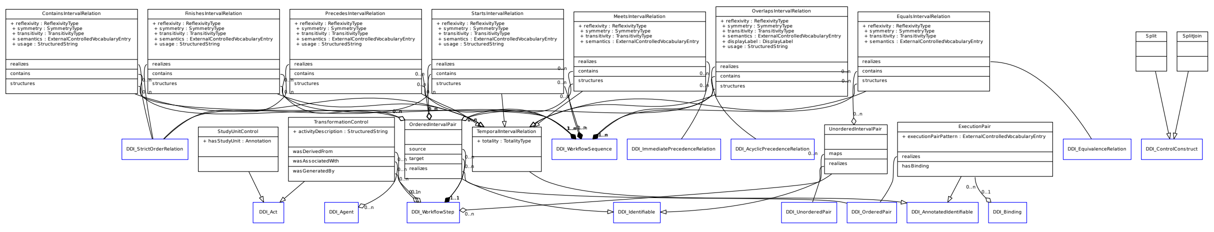
realizes

Class of the Collection pattern realized by this class.

Graph



6.24.16 Graph



6.25 Utility

A package is a administrative collection of classes in DDI. These are not namespaces.

6.25.1 DDI4Version

Graph

6.25.2 DocumentInformation

Provides a consistent path to identifying information regarding the DDI Document and is automatically available for all Functional Views (/DocumentInformation/Annotation etc.). It covers annotation information, coverage (general, plus specific spatial, temporal, and topical coverage), access information from the producer (persistent access control), access information from the local distributor (Local Access Control), information on related series (i.e. a study done as a series of data capture events over time, qualitative material that is part of a larger set, one of a series of funded data capture activities, etc.), funding information, and other document level information used to identify and discover information about a DDI document regardless of its Functional View type. Use the Annotation at this level to provide the full annotation of the DDI Document. Note that related materials can be entered within the Annotation fields.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
contentCoverage	TypedDescriptiveText	0..n
hasPrimaryContent	UnlimitedNatural	0..1
isPublished	Boolean	1..1
ofType	DDI4Version	1..1

contentCoverage

Allows for the addition of a Purpose, Table of Contents, List of Variables, or other forms of information more specific than a general abstract. Provides both the content (using description) and a type identification using an External Controlled Vocabulary Entry.

hasPrimaryContent

A whitespace-delimited list of the DDI URN identifiers of the objects contained in an XML instance or RDF graph which could be considered the primary objects or entry points. In a Codebook View, for example, the top-level Codebook object, fo example. In some views, such as the Agents View, there my be more than one primary object (Individuals, Organizations, and Machines in this case).

isPublished

Default value is False. Indicates that the metadata instance will not be changed without versioning, and is a stable target for referencing.

ofType

Automatically brings in the version number for of DDI 4

‘

Relationships

Name	Type	Cardinality
hasDocumentCoverage	Coverage	0..n
hasExternalMaterial	ExternalMaterial	0..n
hasFundingSource	FundingInformation	0..n
hasLocalAccessConrol	Access	0..n
hasPersistentAccessControl	Access	0..n
partOfSeries	SeriesStatement	0..n

hasDocumentCoverage

Describes the coverage of the document as a whole.

hasExternalMaterial

Allows for listing information on any material external to the DDI metadata instance that is related to it.

hasFundingSource

Describes funding source by description, grant number, and link to funding agent

hasLocalAccessConrol

Access control imposed by the local distributor. For example, distribution only to registered users of a system or members of an association. Provides information on access restrictions as well as the process to gain access.

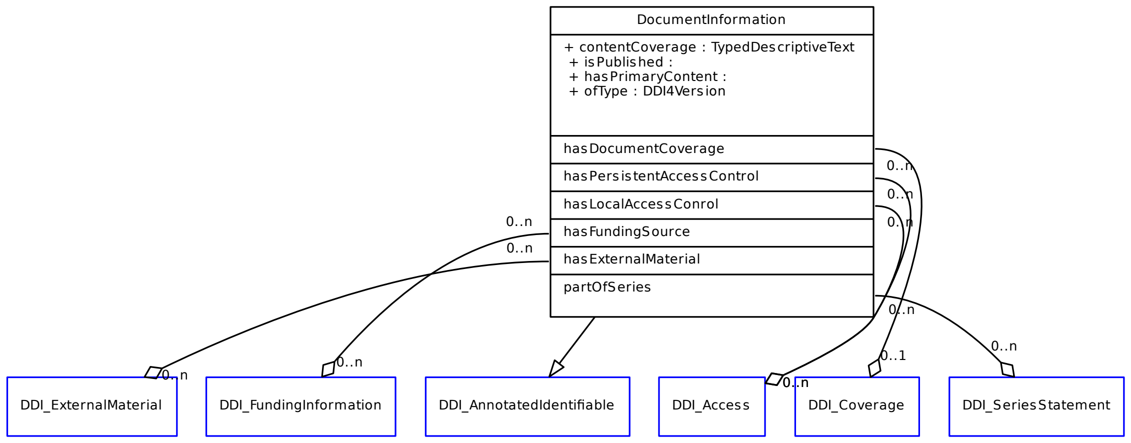
hasPersistentAccessControl

Access control that will persist over time, generally imposed by the producer (i.e. embargo limits, confidentiality constraints). Provides information on access restrictions as well as the process to gain access.

partOfSeries

The series of which the document is a part.

Graph



6.25.3 ExternalMaterial

ExternalMaterial describes the location, structure, and relationship to the DDI metadata instance for any material held external to that instance. This includes citations to such material, an external reference to a URL (or other URI), and a statement about the relationship between the cited ExternalMaterial the contents of the DDI instance. It should be used as follows: As an extension base for specific external materials found within DDI (such as an External Aid); as a target object from a relationship which clarifies its role within a class; or as the target of a relatedResource within an annotation.

Extends

Identifiable

Properties

Name	Type	Cardinality
citationOfExternalMaterial	Annotation	0..1
descriptiveText	StructuredString	0..1
externalURLReference	anyURI	0..n
externalURNReference	anyURI	0..1
mimeType	ExternalControlledVocabularyEntry	0..1
relationshipDescription	StructuredString	0..n
segment	Segment	0..n
typeOfMaterial	ExternalControlledVocabularyEntry	0..1

citationOfExternalMaterial

Bibliographic citation for the external resource.

descriptiveText

A description of the referenced material. This field can map to a Dublin Core abstract. Note that Dublin Core does not support structure within the abstract element. Supports multiple languages and optional structured content.

externalURLReference

Contains a URL which indicates the location of the cited external resource.

externalURNReference

Contains a URN which identifies the cited external resource.

mimeType

Provides a standard Internet MIME type for use by processing applications.

relationshipDescription

Describes the reason the external material is being related to the DDI metadata instance.

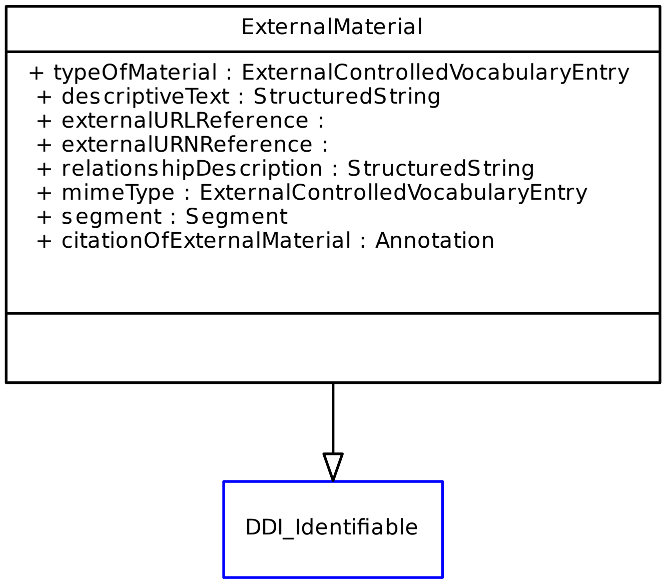
segment

Can describe a segment within a larger object such as a text or video segment.

typeOfMaterial

Designation of the type of material being described. Supports the use of a controlled vocabulary.

Graph


ExternalMaterial
+ typeOfMaterial : ExternalControlledVocabularyEntry
+ descriptiveText : StructuredString
+ externalURLReference :
+ externalURNReference :
+ relationshipDescription : StructuredString
+ mimeType : ExternalControlledVocabularyEntry
+ segment : Segment
+ citationOfExternalMaterial : Annotation
DDI_Identifiable

6.25.4 FundingInformation

Provides information about the individual, agency and/or grant(s) which funded the described entity. Lists a reference to the agency or individual as described by a DDI Agent, the role of the funder, the grant number(s) and a description of the funding activity.

Extends

Identifiable

Properties

Name	Type	Cardinality
funderRole	ExternalControlledVocabularyEntry	0..1
grantNumber	String	0..n
purpose	StructuredString	0..1

funderRole

Role of the funding organization or individual. Supports the use of a controlled vocabulary.

grantNumber

The identification code of the grant or other monetary award which provided funding for the described object.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured tex

'

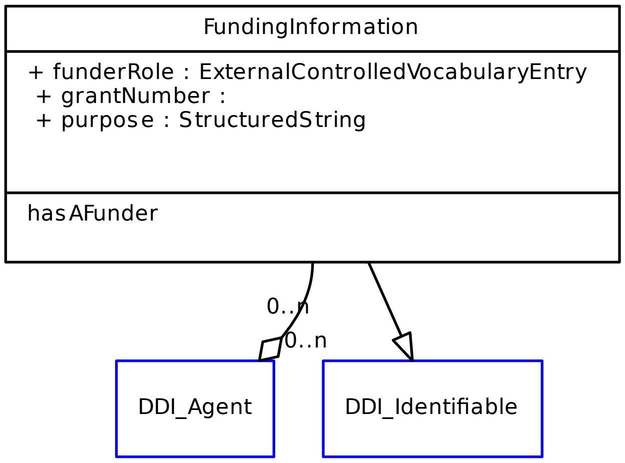
Relationships

Name	Type	Cardinality
hasAFunder	Agent	0..n

hasAFunder

Funding has been provided by

Graph


FundingInformation
+ funderRole : ExternalControlledVocabularyEntry
+ grantNumber :
+ purpose : StructuredString
hasAFunder
0..n
0..n
DDI_Agent
DDI_Identifiable

6.25.5 Note

A note related to one or more identifiable objects. Note is designed to be an inherent part of the DDI. (Unlike XML comments or other types of system-level annotations, which may be removed during processing.) DDI recommends placing the note within the maintainable object containing the objects this note relates to in order to assist tracking of note items within a study. Each note may indicate who is responsible for the note, its type using a controlled vocabulary, the subject of the note, a head and note content, a set of key/value pairs and language specification for the overall note. In addition each note must be related to one or more identifiable objects.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
header	InternationalString	0..1
noteContent	StructuredString	0..1
noteSubject	ExternalControlledVocabularyEntry	0..1
proprietaryInfo	StandardKeyValuePair	0..1
relationship	DescribedRelationship	0..n
responsibility	String	0..1
typeOfNote	ExternalControlledVocabularyEntry	0..1

header

A brief label or heading for the note contents.

noteContent

The content of the note. Note should contain content except when it is a production flag that is fully explained by its "type". If the note provides system specific information in a structured way using XHTML formating, DDI strongly recommends the use of local extensions or the Key/Value pair structure in ProprietaryInfo whenever possible.

noteSubject

The subject of the note.

proprietaryInfo

A set of actions related to the object as described by a set of name-value pairs. This would commonly be used in a case where additional information needs to be recorded regarding the content of a new element or attribute that has not yet been added to the schema, for example when a bug for a missing object has been filed and the user wishes to record the content prior to correction in the schema. Ideally this should be handled by local extensions of the schema as described in Part 2 of the formal documentation. However, the structure in Note allows for an unanticipated need for an extension at run time by providing a means of capturing system specific information in a structured way.

relationship

Reference to one or more identifiable objects which the note is related to.

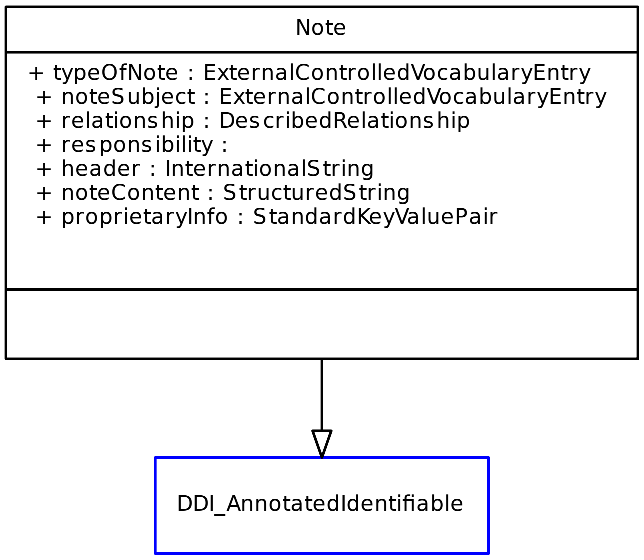
responsibility

The person or agency responsible for adding the note.

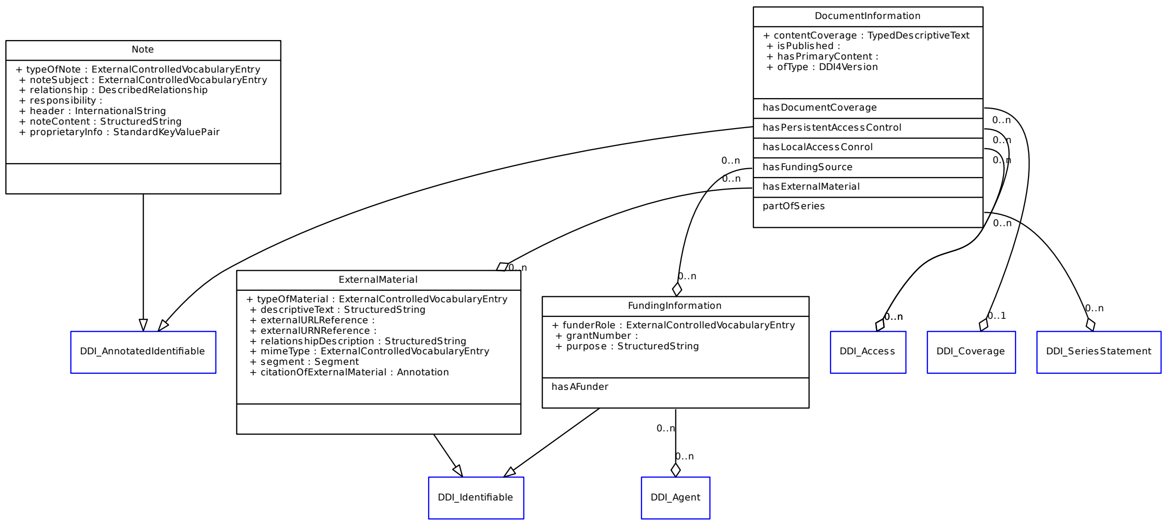
typeOfNote

Specifies the type of note. Supports the use of a controlled vocabulary.

Graph


Note
+ typeOfNote : ExternalControlledVocabularyEntry
+ noteSubject : ExternalControlledVocabularyEntry
+ relationship : DescribedRelationship
+ responsibility :
+ header : InternationalString
+ noteContent : StructuredString
+ proprietaryInfo : StandardKeyValuePair
DDI_AnnotatedIdentifiable

6.25.6 Graph


Note
+ typeOfNote : ExternalControlledVocabularyEntry
+ noteSubject : ExternalControlledVocabularyEntry
+ relationship : DescribedRelationship
+ responsibility :
+ header : InternationalString
+ noteContent : StructuredString
+ proprietaryInfo : StandardKeyValuePair
DocumentInformation
+ contentCoverage : TypedDescriptiveText
+ isPublished :
+ hasPrimaryContent :
+ ofType : DDI4Version
hasDocumentCoverage
hasPersistentAccessControl
hasLocalAccessConrol
hasFundingSource
hasExternalMaterial
partOfSeries
0..n
0..n
0..n
0..n
0..n
0..n
ExternalMaterial
+ typeOfMaterial : ExternalControlledVocabularyEntry
+ descriptiveText : StructuredString
+ externalURLReference :
+ externalURNReference :
+ relationshipDescription : StructuredString
+ mimeType : ExternalControlledVocabularyEntry
+ segment : Segment
+ citationOfExternalMaterial : Annotation
0..n
FundingInformation
+ funderRole : ExternalControlledVocabularyEntry
+ grantNumber :
+ purpose : StructuredString
hasAFunder
0..n
0..n
0..n
DDI_AnnotatedIdentifiable
DDI_Access
0..n
DDI_Coverage
0..1
DDI_SeriesStatement
0..n
DDI_Identifiable
DDI_Agent

CHAPTER

SEVEN

VIEWS

Warning: this is a development build, not a final product.

7.1 StatisticalClassificationView

Purpose: The Statistical Classification View brings together the structures need to record, organize, manage, and index Statistical Classifications. These classifications are often developed cooperatively and managed over time. They may be organized into series and families of classifications and use specified vocabularies. Mapping may be provided to express relationships between two or more Statistical Classifications in order to clarify change over time or the similarities and differences between Statistical Classifications in the same family (i.e. Industry, Occupation, Education).

Use Cases: Industry: US-SIC, NAICS, GICS, ISIC; Occupational: ISCO, US-SOC; Geographic: US-FIPS, NUTS

Target Audience: Organizations maintaining and managing shared Statistical Classifications.

Included Classes: AuthorizationSource, Category, ClassificationFamily, ClassificationIndex, ClassificationIndexEntry, ClassificationItem, ClassificationSeries, Code, Concept, ConceptParentChild, ConceptParentChildPair, ConceptPartWhole, ConceptPartWholePair, ConceptSystem, CorrespondenceTable, ExternalMaterial, Individual, Level, Map, Organization, StatisticalClassification, Vocabulary.

General Documentation: The abstract class Designation serves as a relationship target for the following classes: ClassificationItem and Vocabulary. The only included class belonging to this abstract group is Code. The abstract class Node serves as a relationship target for the following classes: Map and Level. The only included class belonging to this abstract group is ClassificationItem. The abstract class NodeSet serves as a relationship target for the following class: CorrespondenceTable. The only included class belonging to this abstract group is StatisticalClassification. The abstract class Agent serves as a relationship target for the following classes: CorrespondenceTable and AuthorizationSource. The only included classes belonging to this abstract group are Organization and Individual. The abstract pattern class OrderedPair serves as a relationship target for the following class: Map. The only included class realizing this pattern is ConceptParentChildPair

A functional view is a collection of classes in DDI that covers a functional use case. These are not namespaces.

7.1.1 AuthorizationSource

Identifies the authorizing agency and allows for the full text of the authorization (law, regulation, or other form of authorization).

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
authorizationDate	Date	0..1
legalMandate	InternationalString	0..1
purpose	StructuredString	0..1
statementOfAuthorization	StructuredString	0..1

authorizationDate

Identifies the date of Authorization.

legalMandate

Provide a legal citation to a law authorizing the study/data collection. For example, a legal citation for a law authorizing a country's census.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

statementOfAuthorization

Text of the authorization (law, mandate, approved business case).

‘

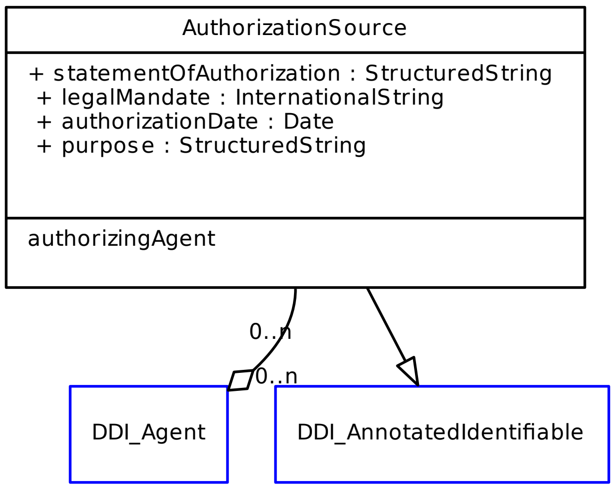
Relationships

Name	Type	Cardinality	allways external
authorizingAgent	Agent	0..n	no

authorizingAgent

References the authorizing agent generally described as an organization or individual

Graph


AuthorizationSource
+ statementOfAuthorization : StructuredString
+ legalMandate : InternationalString
+ authorizationDate : Date
+ purpose : StructuredString
authorizingAgent
0..n
0..n
DDI_Agent
DDI_AnnotatedIdentifiable

7.1.2 Category

A Concept whose role is to define and measure a characteristic.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
name	Name	0..n

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

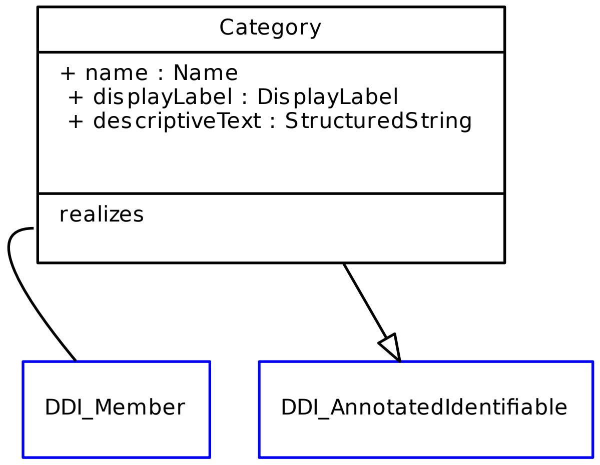
Relationships

Name	Type	Cardinality	allways external
realizes	Member	0..n	yes

realizes

Class can play the role of Member within a Collection

Graph


Category
+ name : Name
+ displayLabel : DisplayLabel
+ descriptiveText : StructuredString
realizes
DDI_Member
DDI_AnnotatedIdentifiable

7.1.3 ClassificationFamily

A Classification Family is a group of Classification Series related from a particular point of view. The Classification Family is related by being based on a common Concept (e.g. economic activity).[GSIM1.1]

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
groups	ClassificationSeries	1..n	no
hasClassificationIndex	ClassificationIndex	0..n	no
realizes	Collection	0..n	yes

groups

Specialization of contains in Collection

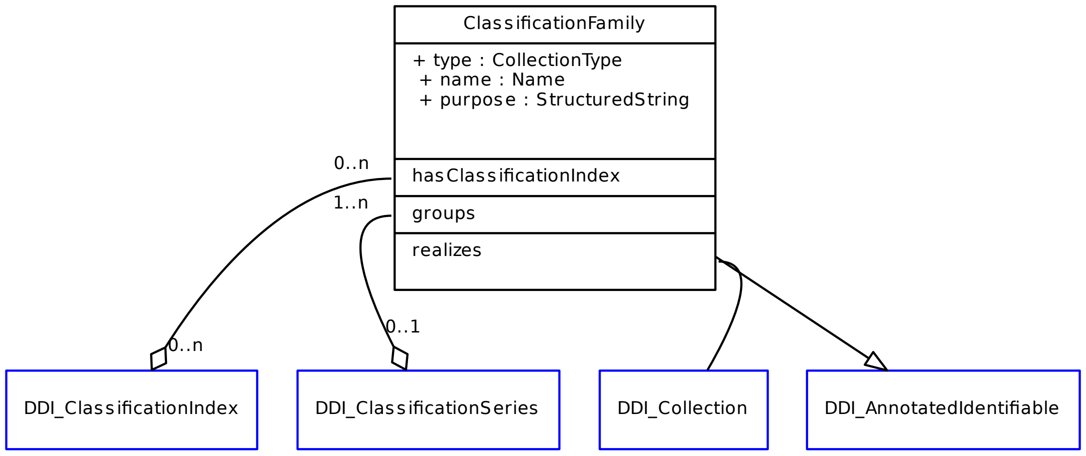
hasClassificationIndex

ClassificationIndexes associated to the ClassificationFamily.

realizes

Class of the Collection pattern realized by this class.

Graph



7.1.4 ClassificationIndex

A Classification Index is an ordered list (alphabetical, in code order etc) of Classification Index Entries. A Classification Index can relate to one particular or to several Statistical Classifications. [GSIM Statistical Classification Model]

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
availableLanguage		0..n
codingInstructions	CommandCode	0..n
contactPersons	AgentAssociation	0..n
corrections	InternationalString	0..n
maintenanceUnit	AgentAssociation	0..1
name	Name	0..n
purpose	StructuredString	0..1
releaseDate	Date	0..1
type	CollectionType	0..1

availableLanguage

A Classification Index can exist in several languages. Indicates the languages available. If a Classification Index exists in several languages, the number of entries in each language may be different, as the number of terms describing the same phenomenon can change from one language to another. However, the same phenomena should be described in each language.

codingInstructions

Additional information which drives the coding process for all entries in a Classification Index.

contactPersons

Person(s) who may be contacted for additional information about the Classification Index.

corrections

Verbal summary description of corrections, which have occurred within the Classification Index. Corrections include changing the item code associated with an Classification Index Entry.

maintenanceUnit

The unit or group of persons within the organisation responsible for the Classification Index, i.e. for adding, changing or deleting Classification Index Entries.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

releaseDate

Date when the current version of the Classification Index was released.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
groups	ClassificationIndexEntry	1..n	no
hasPublications	ExternalMaterial	0..n	no
realizes	Collection	0..n	yes

groups

Realization of contains in Collection

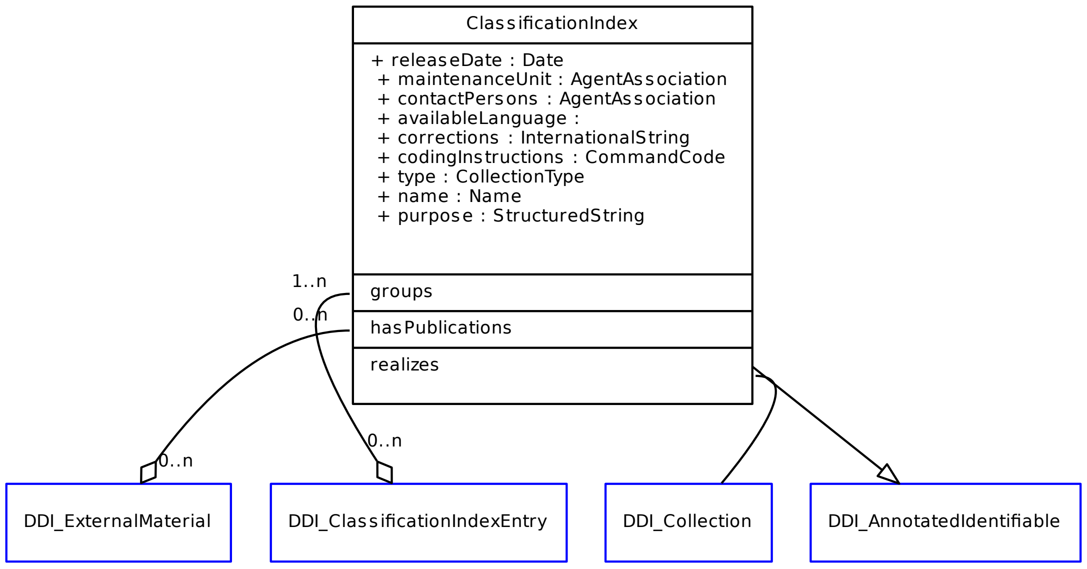
hasPublications

A list of the publications in which the Classification Index has been published.

realizes

Class of the Collection pattern realized by this class.

Graph


ClassificationIndex
+ releaseDate : Date
+ maintenanceUnit : AgentAssociation
+ contactPersons : AgentAssociation
+ availableLanguage :
+ corrections : InternationalString
+ codingInstructions : CommandCode
+ type : CollectionType
+ name : Name
+ purpose : StructuredString
groups
hasPublications
realizes
1..n
0..n
0..n
0..n
DDI_ExternalMaterial
DDI_ClassificationIndexEntry
DDI_Collection
DDI_AnnotatedIdentifiable

7.1.5 ClassificationIndexEntry

A Classification Index Entry is a word or a short text (e.g. the name of a locality, an economic activity or an occupational title) describing a type of object/unit or object property to which a Classification Item applies, together with the code of the corresponding Classification Item. Each Classification Index Entry typically refers to one item of the Statistical Classification. Although a Classification Index Entry may be associated with a Classification Item at any Level of a Statistical Classification, Classification Index Entries are normally associated with items at the lowest Level.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
codingInstructions	CommandCode	0..1
content	InternationalString	0..1
validDates	DateRange	0..1

codingInstructions

Additional information which drives the coding process. Required when coding is dependent upon one or many other factors.

content

Text describing the type of object/unit or object property.

validDates

Date from which the Classification Index Entry became valid (startDate). The date must be defined if the Classification Index Entry belongs to a floating Classification Index. Date at which the Classification Index Entry became invalid (endDate). The date must be defined if the Classification Index Entry belongs to a floating Classification Index and is no longer valid.

‘

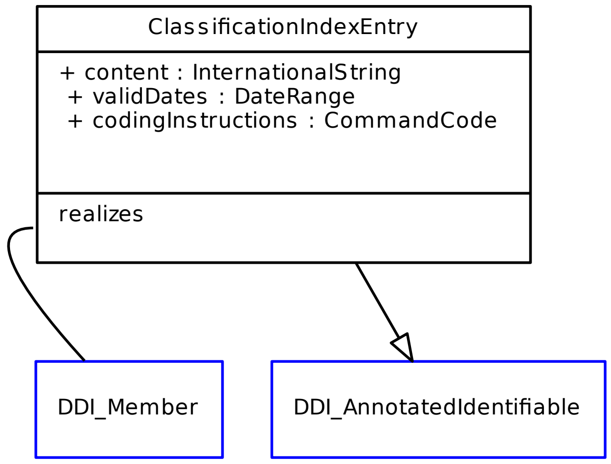
Relationships

Name	Type	Cardinality	allways external
realizes	Member	0..n	yes

realizes

Class of the Collection pattern realized by this class.

Graph


ClassificationIndexEntry
+ content : InternationalString
+ validDates : DateRange
+ codingInstructions : CommandCode
realizes
DDI_Member
DDI_AnnotatedIdentifiable

7.1.6 ClassificationItem

A Classification Item represents a Category at a certain Level within a Statistical Classification.

Extends

Node

Properties

Name	Type	Cardinality
changeFromPreviousVersion	InternationalString	0..1
changeLog	InternationalString	0..1
explanatoryNotes	StructuredString	0..n
futureNotes	InternationalString	0..n
isGenerated	Boolean	0..1
isValid	Boolean	0..1
officialName	Name	0..1
validDates	DateRange	0..1

changeFromPreviousVersion

Describes the changes, which the item has been subject to from the previous version to the actual Statistical Classification

changeLog

Describes the changes, which the item has been subject to during the life time of the actual Statistical Classification.

explanatoryNotes

A Classification Item may be associated with explanatory notes, which further describe and clarify the contents of the Category. Explanatory notes consist of: General note: Contains either additional information about the Category, or a general description of the Category, which is not structured according to the "includes", "includes also", "excludes" pattern. Includes: Specifies the contents of the Category. Includes also: A list of borderline cases, which belong to the described Category. Excludes: A list of borderline cases, which do not belong to the described Category. Excluded cases may contain a reference to the Classification Items to which the excluded cases belong.

futureNotes

The future events describe a change (or a number of changes) related to an invalid item. These changes may e.g. have turned the now invalid item into one or several successor items. This allows the possibility to follow successors of the item in the future.

isGenerated

Indicates whether or not the item has been generated to make the level to which it belongs complete

isValid

Indicates whether or not the item is currently valid. If updates are allowed in the Statistical Classification, an item may be restricted in its validity, i.e. it may become valid or invalid after the Statistical Classification has been released.

officialName

A Classification Item has a name as provided by the owner or maintenance unit. The name describes the content of the category. The name is unique within the Statistical Classification to which the item belongs, except for categories that are identical at more than one level in a hierarchical classification

validDates

Dates for which the classification is valid. Date from which the item became valid. The date must be defined if the item belongs to a floating Statistical classification. Date at which the item became invalid. The date must be defined if the item belongs to a floating Statistical classification and is no longer valid

'

Relationships

Name	Type	Cardinality	allways external
caseLaw	AuthorizationSource	0..n	no
excludes	ClassificationItem	0..n	no
groups	ClassificationIndexEntry	1..n	no

caseLaw

Case law rulings related to the Classification Item.

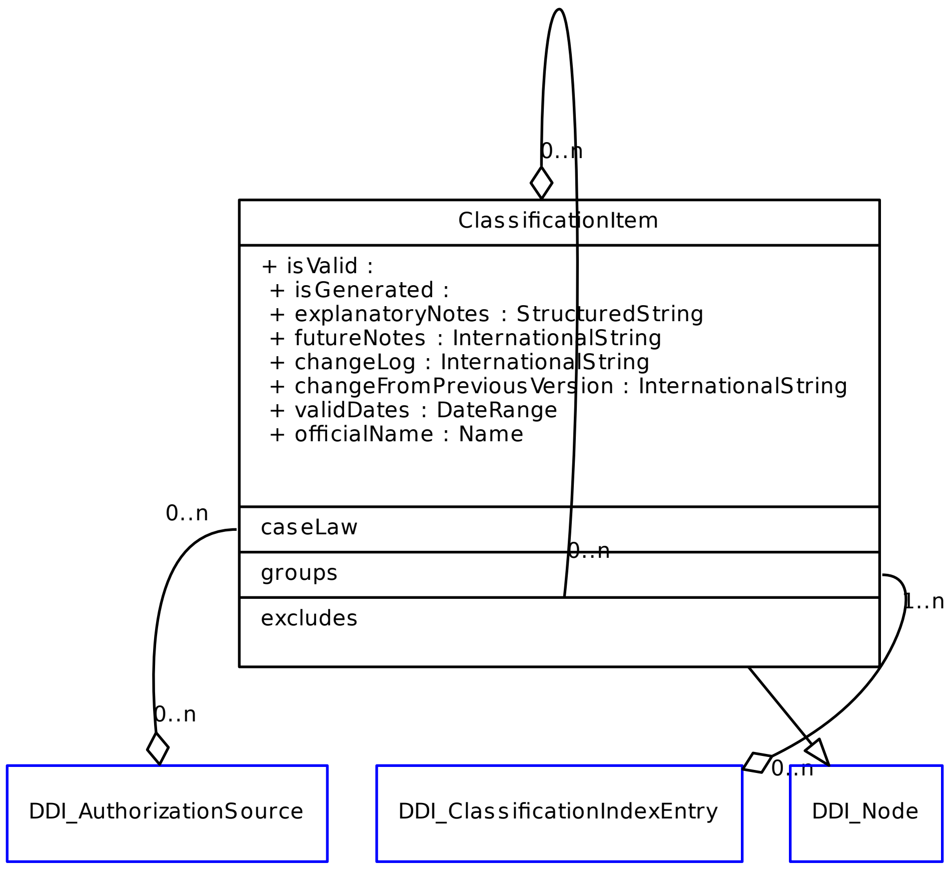
excludes

Classification Items to which the excluded cases belong (as described in explanatoryNotes).

groups

ClassificationIndexEntries related to the ClassificationItem.

Graph


ClassificationItem
+ isValid :
+ isGenerated :
+ explanatoryNotes : StructuredString
+ futureNotes : InternationalString
+ changeLog : InternationalString
+ changeFromPreviousVersion : InternationalString
+ validDates : DateRange
+ officialName : Name
caseLaw
groups
excludes
0..n
0..n
0..n
0..n
1..n
0..n
DDI_AuthorizationSource
DDI_ClassificationIndexEntry
DDI_Node

7.1.7 ClassificationSeries

A Classification Series is an ensemble of one or more Statistical Classifications, based on the same concept, and related to each other as versions or updates. Typically, these Statistical Classifications have the same name (for example, ISIC or ISCO).

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
context	StructuredString	0..1
keywords	ExternalControlledVocabularyEntry	0..n
name	Name	0..n
objectsOrUnitsClassified	ExternalControlledVocabularyEntry	0..1
owners	AgentAssociation	0..n
purpose	StructuredString	0..1
subjectAreas	ExternalControlledVocabularyEntry	0..n
type	CollectionType	0..1

context

ClassificationSeries can be designed in a specific context.

keywords

A ClassificationSeries can be associated with one or a number of keywords.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

objectsOrUnitsClassified

A ClassificationSeries is designed to classify a specific type of object/unit according to a specific attribute.

owners

The statistical office or other authority, which created and maintains the StatisticalClassification (s) related to the ClassificationSeries. A ClassificationSeries may have several owners.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

subjectAreas

Areas of statistics in which the ClassificationSeries is implemented.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
groups	StatisticalClassification	0..n	no
realizes	Collection	0..n	yes

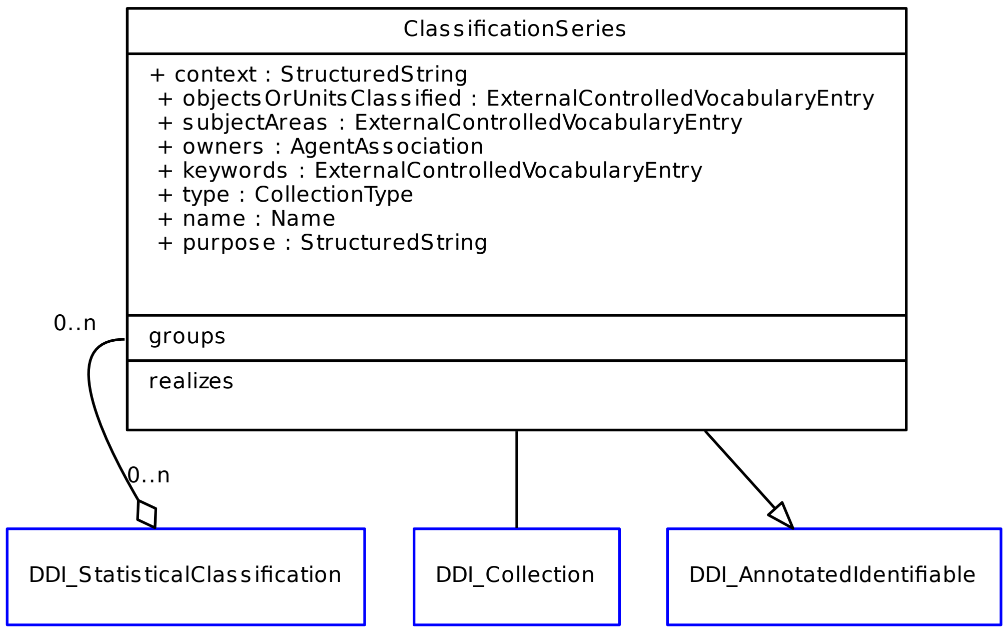
groups

Realization of contains in Collection for StatisticalClassifications.

realizes

Class of the Collection pattern realized by this class.

Graph


ClassificationSeries
+ context : StructuredString
+ objectsOrUnitsClassified : ExternalControlledVocabularyEntry
+ subjectAreas : ExternalControlledVocabularyEntry
+ owners : AgentAssociation
+ keywords : ExternalControlledVocabularyEntry
+ type : CollectionType
+ name : Name
+ purpose : StructuredString
groups
realizes
0..n
0..n
DDI_StatisticalClassification
DDI_Collection
DDI_AnnotatedIdentifiable

7.1.8 Code

A Designation for a Category.

Extends

Designation

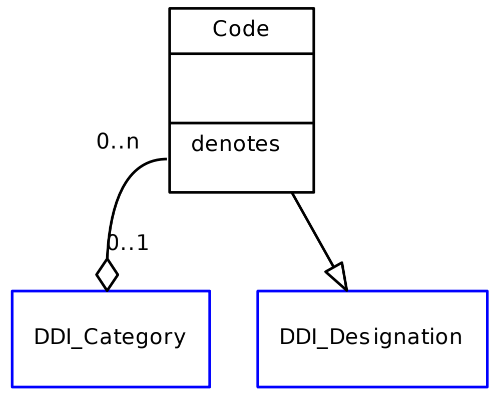
Relationships

Name	Type	Cardinality	allways external
denotes	Category	0..n	no

denotes

A definition for the code. Specialization of denotes for Categories.

Graph



7.1.9 Concept

Unit of thought differentiated by characteristics [GSIM 1.1]

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
definition	StructuredString	0..1
name	Name	0..n

definition

Natural language statement conveying the meaning of a concept, differentiating it from other concepts. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

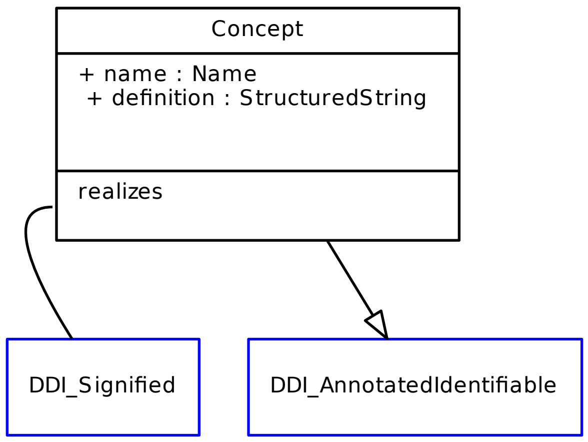
Relationships

Name	Type	Cardinality	allways external
realizes	Signified	0..n	yes

realizes

A concept can play the role of a signified when there is a designation that denotes it.

Graph



7.1.10 ConceptParentChild

Parent-child specialization of OrderRelation between Concepts within a ConceptSystem.

Extends

Identifiable

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1

reflexivity

Fixed to Anti_Reflexive

semantics

Controlled vocabulary for the ImmediatePrecedenceRelation semantics. It should contain, at least, the following: Parent_Of, Child_Of, Next, Previous, Instance_Of, Temporal_Meets.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Anti_Transitive

‘

Relationships

Name	Type	Cardinality	allways external
contains	ConceptParentChildPair	0..n	no
realizes	ImmediatePrecedenceRelation	0..n	yes
structures	ConceptSystem	0..n	no

contains

Ordered sets of Parent Child pairs.

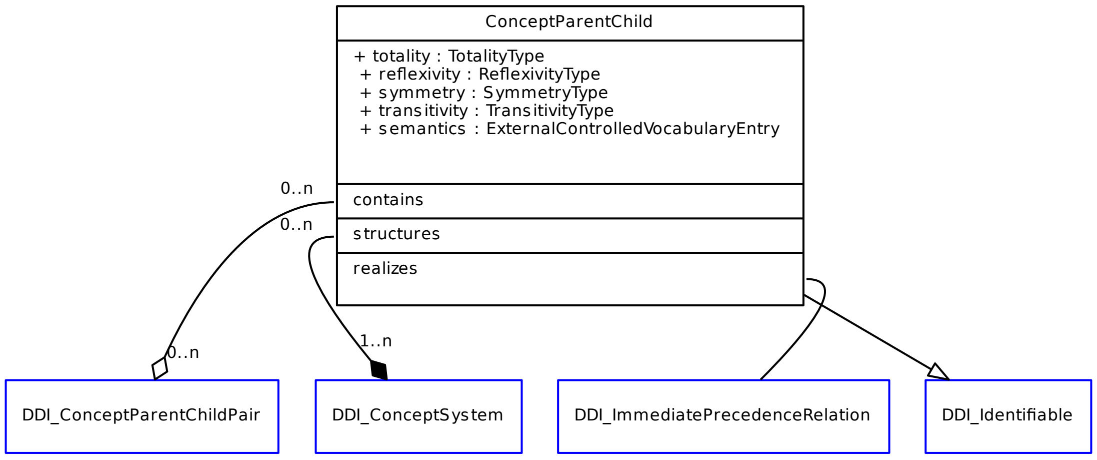
realizes

Uses the pattern ImmediatePrecedenceRelation

structures

Concept System whose members are grouped by pairs to form a variety of structures.

Graph


ConceptParentChild
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
contains
structures
realizes
0..n
0..n
0..n
1..n
DDI_ConceptParentChildPair
DDI_ConceptSystem
DDI_ImmediatePrecedenceRelation
DDI_Identifiable

7.1.11 ConceptParentChildPair

Specifies the Parent Concept and Child Concept in the Concept Parent Child Relationship.

Extends

Identifiable

Relationships

Name	Type	Cardinality	allways external
child	Concept	0..n	no
parent	Concept	0..n	no
realizes	OrderedPair	0..n	yes

child

Specialization of “target” in OrderedPair

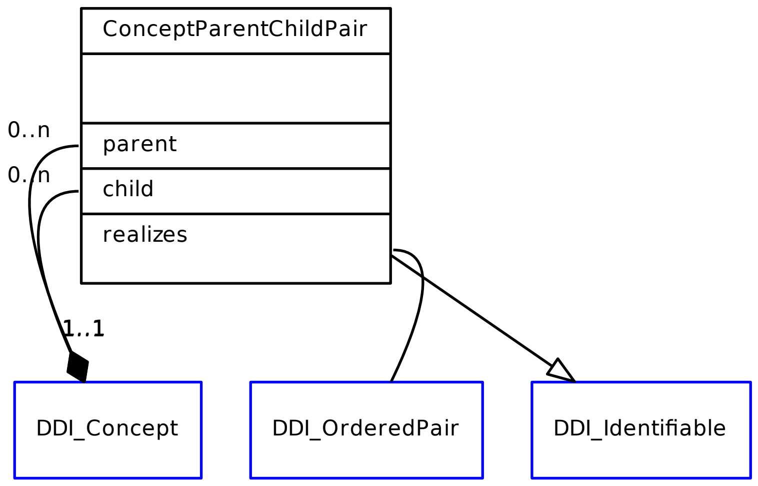
parent

Specialization of “source” in OrderedPair

realizes

realizes the pattern OrderedPair

Graph



7.1.12 ConceptPartWhole

Part-whole specialization of OrderRelation between Concepts within a ConceptSystem.

Extends

Identifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
displayLabel	DisplayLabel	0..n
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

criteria

Intentional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

displayLabel

A display label for the OrderRelation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

reflexivity

Fixed to Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality	allways external
contains	ConceptPartWholePair	0..n	no
realizes	OrderRelation	0..n	yes
structures	ConceptSystem	0..n	no

contains

Ordered pairs of ConceptPartWholePairs

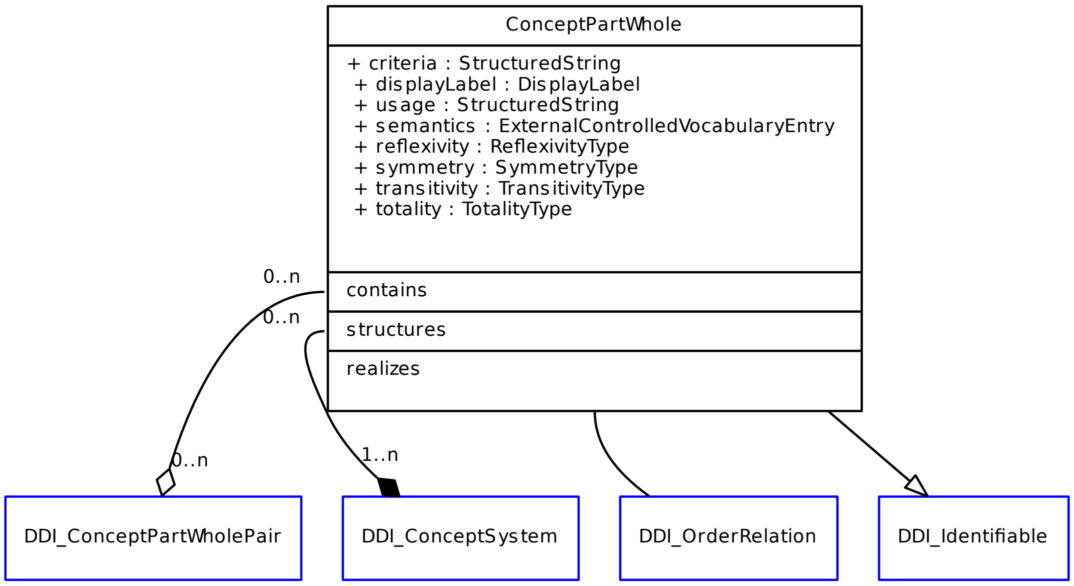
realizes

realizes the pattern of ConceptPartWhole

structures

ConceptSystem whose Members are groups by ConceptPartWholePairs

Graph


ConceptPartWhole
+ criteria : StructuredString
+ displayLabel : DisplayLabel
+ usage : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ totality : TotalityType
contains
structures
realizes
0..n
0..n
0..n
1..n
DDI_ConceptPartWholePair
DDI_ConceptSystem
DDI_OrderRelation
DDI_Identifiable

7.1.13 ConceptPartWholePair

Identifies the concept defining the Whole and the concept defining the Part.

Extends

Identifiable

Relationships

Name	Type	Cardinality	allways external
part	Concept	0..n	no
realizes	OrderedPair	0..n	yes
whole	Concept	0..n	no

part

Specialization of "target" in OrderedPair

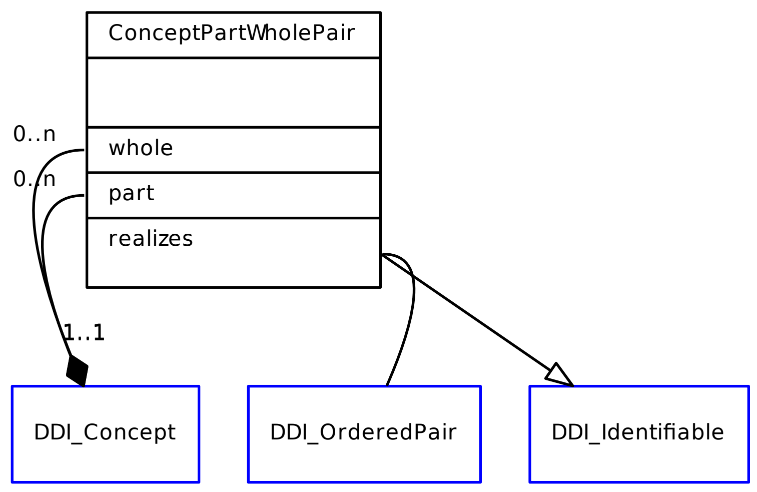
realizes

realizes pattern OrderedPair

whole

Specialization of "source" in OrderedPair.

Graph



7.1.14 ConceptSystem

A set of Concepts structured by the relations among them. [GSIM 1.1]

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
contains	Concept	0..n	no
hasConceptParentChild	ConceptParentChild	1..n	no
hasConceptPartWhole	ConceptPartWhole	1..n	no
realizes	Collection	0..n	yes

contains

The relationship to the concepts contained in the concept system

hasConceptParentChild

Specialization of isOrderedBy in Collection.

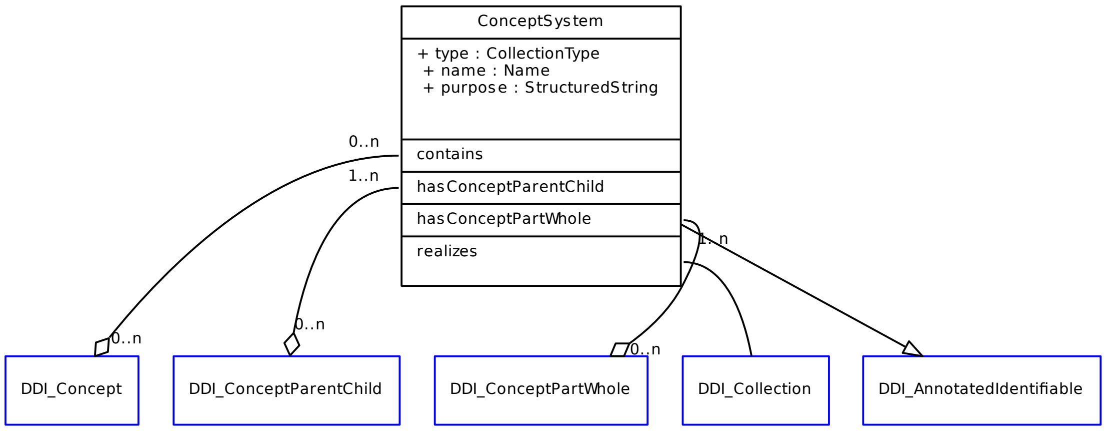
hasConceptPartWhole

Specialization of isOrderedBy in Collection.

realizes

realizes the collection pattern

Graph



7.1.15 CorrespondenceTable

A Correspondence Table expresses a relationship between two NodeSets.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
effectiveDates	DateRange	0..1

effectiveDates

Effective period of validity of the CorrespondenceTable. The correspondence table expresses the relationships between the two NodeSets as they existed on the period specified in the table.

‘

Relationships

Name	Type	Cardinality	allways external
contactPersons	Agent	0..n	no
contains	Map	1..n	no
hasPublication	ExternalMaterial	0..n	no
maintenanceUnit	Agent	0..n	no
maps	NodeSet	0..n	no
owners	Agent	0..n	no
realizes	AsymmetricRelation	0..n	yes
sourceLevel	Level	0..n	no
targetLevel	Level	0..n	no

contactPersons

The person(s) who may be contacted for additional information about the Correspondence Table. Can be an individual or organization.

contains

Set of mappings between nodes that participate in the correspondence.

hasPublication

A list of the publications in which the Correspondence Table has been published.

maintenanceUnit

The unit or group of persons who are responsible for the Correspondence Table, i.e. for maintaining and updating it.

maps

The NodeSet(s) from which the correspondence is made.

owners

The statistical office, other authority or section that created and maintains the Correspondence Table. A Correspondence Table may have several owners.

realizes

Class of the Collection pattern realized by this class.

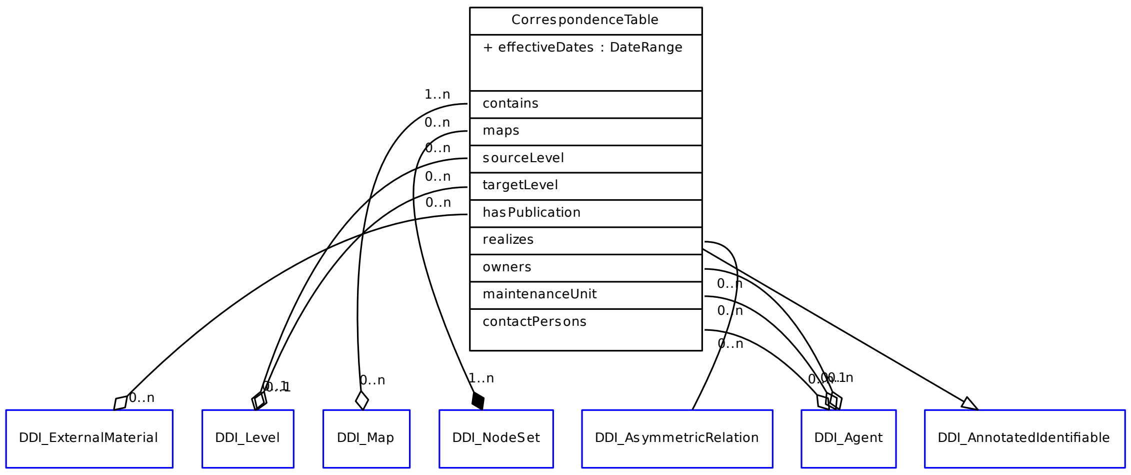
sourceLevel

Level from which the correspondence is made. Correspondences might be restricted to a certain Level in the NodeSet. In this case, target items are assigned only to source items on the given level. If no level is indicated, source items can be assigned to any level of the target NodeSet.

targetLevel

Level to which the correspondence is made. Correspondences might be restricted to a certain Level in the NodeSet. In this case, target items are assigned only to source items on the given level. If no level is indicated, target items can be assigned to any level of the source NodeSet.

Graph



7.1.16 ExternalMaterial

ExternalMaterial describes the location, structure, and relationship to the DDI metadata instance for any material held external to that instance. This includes citations to such material, an external reference to a URL (or other URI), and a statement about the relationship between the cited ExternalMaterial the contents of the DDI instance. It should be used as follows: As an extension base for specific external materials found within DDI (such as an External Aid); as a target object from a relationship which clarifies its role within a class; or as the target of a relatedResource within an annotation.

Extends

Identifiable

Properties

Name	Type	Cardinality
citationOfExternalMaterial	Annotation	0..1
descriptiveText	StructuredString	0..1
externalURLReference	anyURI	0..n
externalURNReference	anyURI	0..1
mimeType	ExternalControlledVocabularyEntry	0..1
relationshipDescription	StructuredString	0..n
segment	Segment	0..n
typeOfMaterial	ExternalControlledVocabularyEntry	0..1

citationOfExternalMaterial

Bibliographic citation for the external resource.

descriptiveText

A description of the referenced material. This field can map to a Dublin Core abstract. Note that Dublin Core does not support structure within the abstract element. Supports multiple languages and optional structured content.

externalURLReference

Contains a URL which indicates the location of the cited external resource.

externalURNReference

Contains a URN which identifies the cited external resource.

mimeType

Provides a standard Internet MIME type for use by processing applications.

relationshipDescription

Describes the reason the external material is being related to the DDI metadata instance.

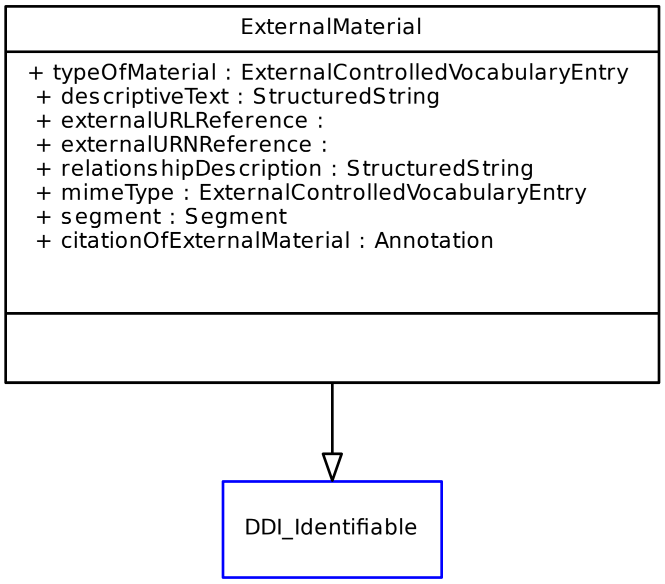
segment

Can describe a segment within a larger object such as a text or video segment.

typeOfMaterial

Designation of the type of material being described. Supports the use of a controlled vocabulary.

Graph


ExternalMaterial
+ typeOfMaterial : ExternalControlledVocabularyEntry
+ descriptiveText : StructuredString
+ externalURLReference :
+ externalURNReference :
+ relationshipDescription : StructuredString
+ mimeType : ExternalControlledVocabularyEntry
+ segment : Segment
+ citationOfExternalMaterial : Annotation
DDI_Identifiable

7.1.17 Individual

A person who acts, or is designated to act towards a specific purpose.

Extends

Agent

Properties

Name	Type	Cardinality
contactInformation	ContactInformation	0..1
ddiId	String	0..n
individualName	IndividualName	0..n

contactInformation

Contact information for the individual including location specification, address, URL, phone numbers, and other means of communication access. Sets of information can be repeated and date-stamped.

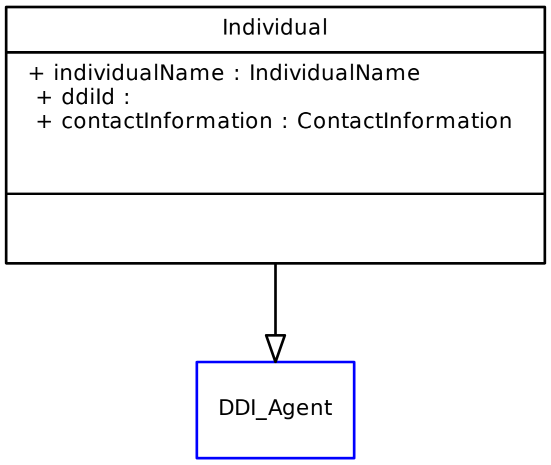
ddiId

The agency identifier of the individual according to the DDI Alliance agent registry.

individualName

The name of an individual broken out into its component parts of prefix, first/given name, middle name, last/family/surname, and suffix.

Graph


Individual
+ individualName : IndividualName
+ ddiId :
+ contactInformation : ContactInformation
DDI_Agent

7.1.18 Level

The Level describes the nesting structure of a hierarchical collection.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
groups	Node	1..1	no
isDefinedBy	Concept	0..n	no
realizes	Collection	0..n	yes

groups

Realization of contains in Collection for Nodes.

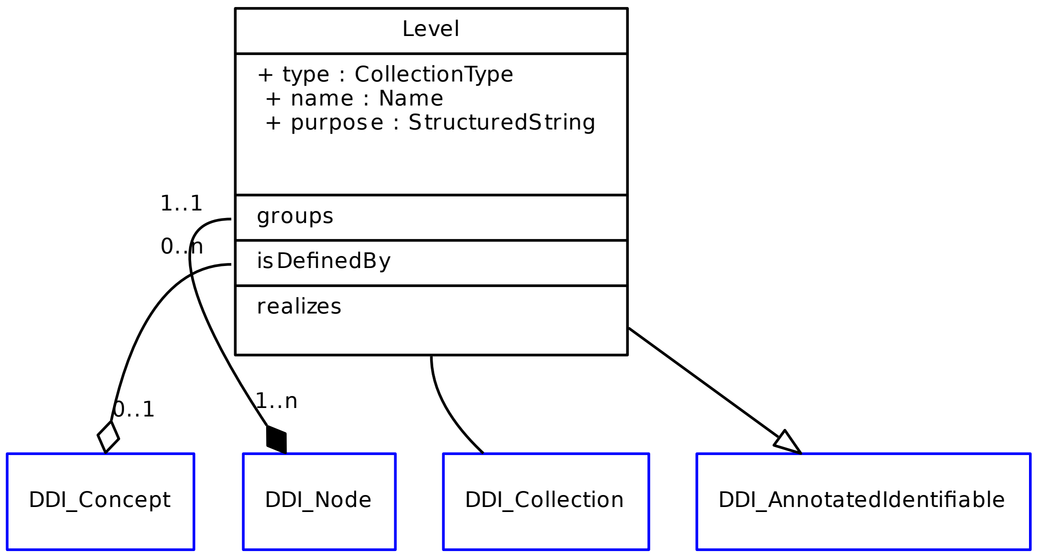
isDefinedBy

Associated concept that provides the conceptual definition of the level.

realizes

Class of the Collection pattern realized by this class.

Graph



7.1.19 Map

Relationship between Nodes in NodeSets.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..1
type	CorrespondenceType	0..1
usage	StructuredString	0..1
validDates	DateRange	0..1

displayLabel

A display label for the OrderedMemberCorrespondence. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

type

Type of correspondence in terms of commonalities and differences between two members.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

validDates

Date from which the Map became valid. The date must be defined if the Map belongs to a floating CorrespondenceTable. Date at which the Map became invalid. The date must be defined if the Map belongs to a floating Correspondence Table and is no longer valid.

‘

Relationships

Name	Type	Cardinality	allways external
realizes	OrderedTuple	0..n	yes
source	Node	0..n	no
target	Node	0..n	no

realizes

Class of the Collection pattern realized by this class.

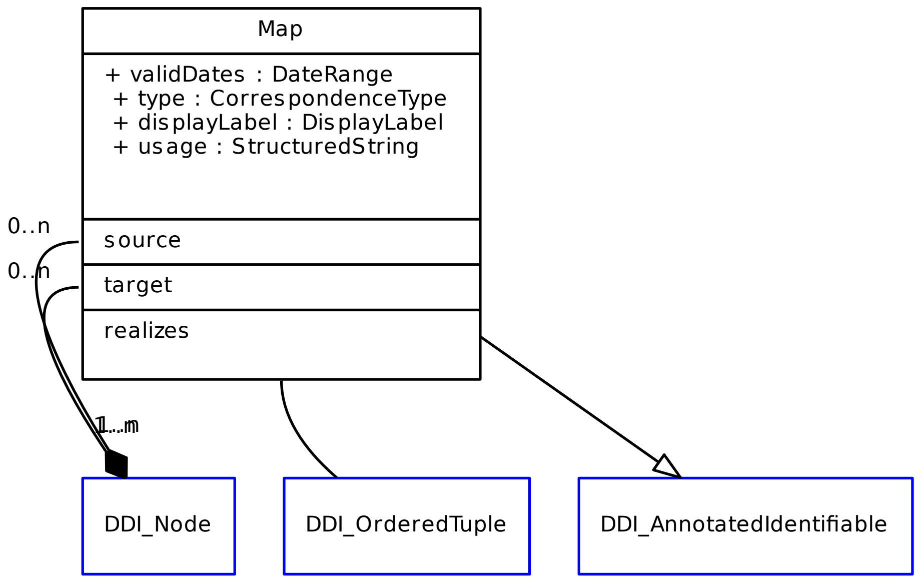
source

Realization of source in Ordered Tuple for Nodes.

target

Realization of target in Ordered Tuple for Nodes.

Graph


Map
+ validDates : DateRange
+ type : CorrespondenceType
+ displayLabel : DisplayLabel
+ usage : StructuredString
source
target
realizes
0..n
0..n
1..m
DDI_Node
DDI_OrderedTuple
DDI_AnnotatedIdentifiable

7.1.20 Organization

A framework of authority designated to act toward some purpose.

Extends

Agent

Properties

Name	Type	Cardinality
contactInformation	ContactInformation	0..1
ddiId	String	0..n
organizationName	OrganizationName	1..n

contactInformation

Contact information for the organization including location specification, address, URL, phone numbers, and other means of communication access. Sets of information can be repeated and date-stamped.

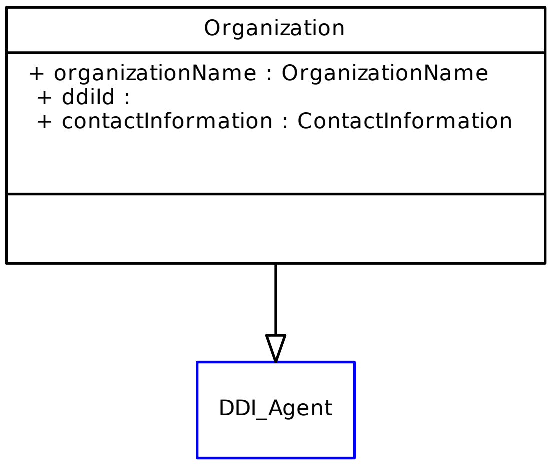
ddiId

The agency identifier of the organization as registered at the DDI Alliance register.

organizationName

Names by which the organization is known.

Graph


Organization
+ organizationName : OrganizationName
+ ddiId :
+ contactInformation : ContactInformation
DDI_Agent

7.1.21 StatisticalClassification

A Statistical Classification is a set of Categories which may be assigned to one or more variables registered in statistical surveys or administrative files, and used in the production and dissemination of statistics. The Categories at each Level of the classification structure must be mutually exclusive and jointly exhaustive of all objects/units in the population of interest. (Source: GSIM StatisticalClassification)

Extends

NodeSet

Properties

Name	Type	Cardinality
availableLanguage		0..n
changeFromBase	StructuredString	0..1
copyright	InternationalString	0..n
displayLabel	DisplayLabel	0..n
isCurrent	Boolean	0..1
isFloating	Boolean	0..1
purposeOfVariant	StructuredString	0..1
rationale	StructuredString	0..1
releaseDate	Date	0..1
updateChanges	StructuredString	0..n
usage	StructuredString	0..1
validDates	DateRange	0..1

availableLanguage

A list of languages in which the Statistical Classification is available. Repeat for each langauge.

changeFromBase

Describes the relationship between the variant and its base Statistical Classification, including regroupings, aggregations added and extensions. (Source: GSIM StatisticalClassification/Changes from base Statistical Classification)

copyright

Copyright of the statistical classification.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

isCurrent

Indicates if the Statistical Classification is currently valid.

isFloating

Indicates if the Statistical Classification is a floating classification. In a floating statistical classification, a validity period should be defined for all Classification Items which will allow the display of the item structure and content at different points of time. (Source: GSIM StatisticalClassification/Floating

purposeOfVariant

If the Statistical Classification is a variant, notes the specific purpose for which it was developed. (Source: GSIM StatisticalClassification/Purpose of variant)

rationale

Explanation of the reasons some decision was made or some object exists. Supports the use of multiple languages and structured text.

releaseDate

Date the Statistical Classification was released

updateChanges

Summary description of changes which have occurred since the most recent classification version or classification update came into force.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

validDates

The date the statistical classification enters production use and the date on which the Statistical Classification was superseded by a successor version or otherwise ceased to be valid. (Source: GSIM Statistical Classification)

‘

Relationships

Name	Type	Cardinality	allways external
contains	ClassificationItem	1..n	no
hasDistribution	ExternalMaterial	0..n	no
has	ClassificationIndex	1..n	no
isMaintainedBy	Organization	0..n	no
predecessor	StatisticalClassification	0..1	no
successor	StatisticalClassification	0..1	no
variantOf	StatisticalClassification	0..n	no

contains

Specialization of contains in NodeSet for ClassificationItems.

hasDistribution

Description and link to a publication, including print, PDF, HTML and other electronic formats, in which the Statistical Classification has been published. This is similar to dcat:Distribution.

has

ClassificationIndex(es) related to the StatisticalClassification.

isMaintainedBy

Organization, agency, or group within an agency responsible for the maintenance and upkeep of the statistical classification.

predecessor

Statistical Classification superseded by the actual Statistical Classification (for those Statistical Classifications that are versions or updates).

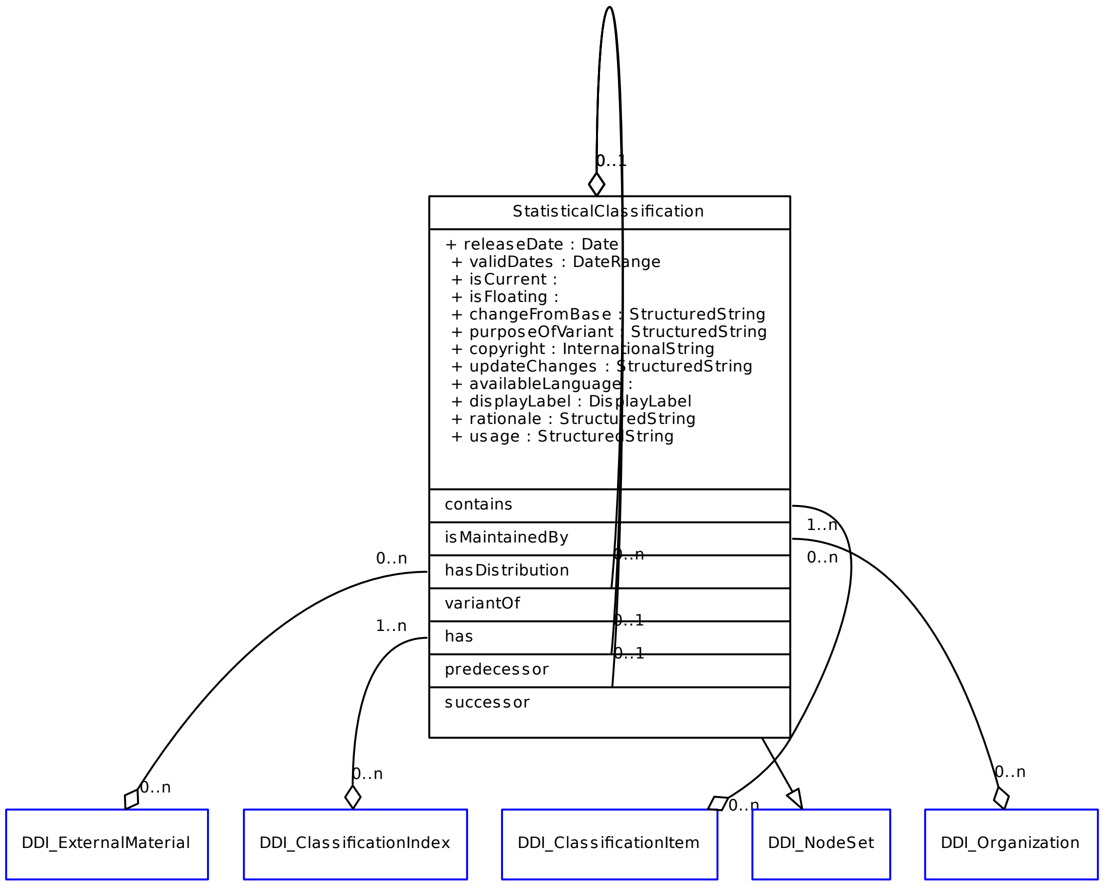
successor

Statistical Classification that supersedes the actual Statistical Classification (for those Statistical Classifications that are versions or updates),

variantOf

Statistical Classification on which the current variant is based, and any subsequent versions of that Statistical Classification to which it is also applicable.

Graph



7.1.22 Vocabulary

A vocabulary is an established list of standardized terminology for use in indexing and retrieval of information.

Extends

ConceptSystem

Properties

Name	Type	Cardinality
abbreviation	InternationalString	0..n
comments	StructuredString	0..n
location	URI	0..1

abbreviation

Abbreviation of vocabulary title.

comments

Information for the user regarding the reasons for use of the vocabulary and appropriate usage constraints.

location

Location of external resource providing information about the vocabulary.

‘

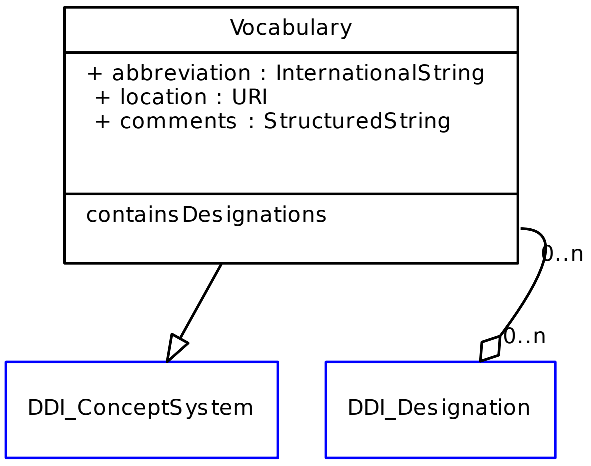
Relationships

Name	Type	Cardinality	allways external
containsDesignations	Designation	0..n	no

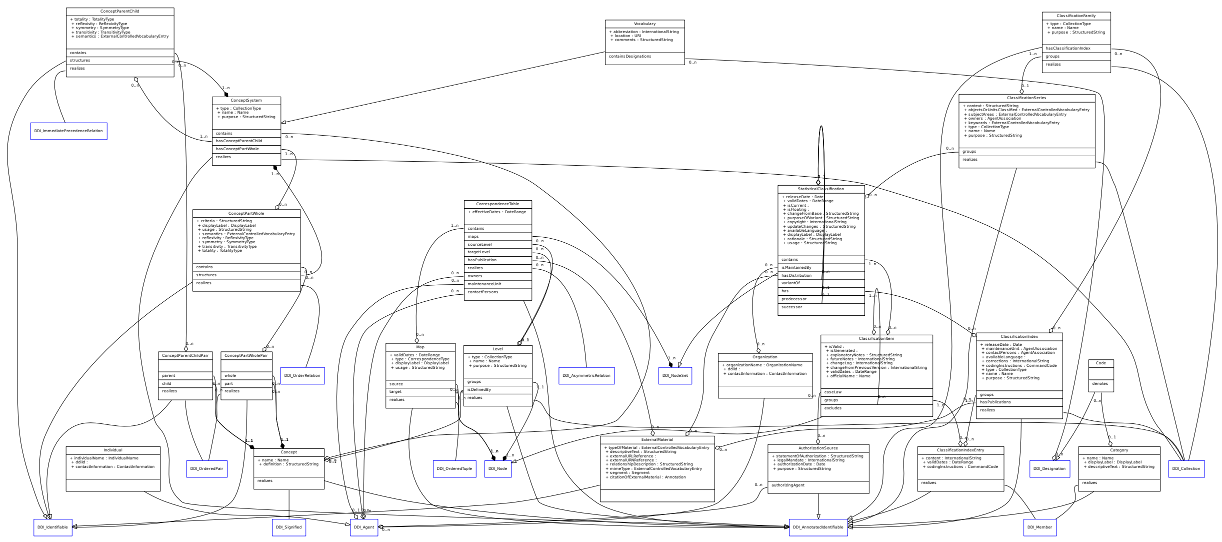
containsDesignations

Vocabularies contain Designations

Graph



7.1.23 Graph



7.2 InstrumentView

Purpose: The purpose of the Instrument Functional View is to describe the data collection instrument and procedures involved in fielding a simple survey questionnaire. It allows only basic Domains (Text, Numeric, and CodeList), but supports the full questionnaire flow (Sequence, IfThenElse, ElseIf, Loop, RepeatUntil, and RepeatWhile). The specific flow of the Instrument is described using PrecedesIntervalRelation. Support has also been provided for binding the output of one capture to the input of another, for example, the number of persons in the household output used as the condition of a Loop.

Target Audience: Creator's of simple questionnaires.

Included Classes: Binding , Category, CategorySet, CodeItem, CodeList, Concept, ConceptualInstrument , ConceptualVariable, ElseIf , ExternalAid , ExternalMaterial, IfThenElse , ImplementedInstrument, InputParameter, InstanceQuestion , Instruction , InstrumentCode , Level, Loop , OrderedIntervalPair, OutputParameter, Parameters , PredesIntervalRelation, RepeatUntil , RepeatWhile , RepresentedQuestion, RepresentedVariable, ResponseDomain , SentinelConceptualDomain, SentinelValueDomain, Statement , SubstantiveConceptualDomain, SubstantiveValueDomain, UnitType, Universe, ValueAndConceptDescription, WorkflowSequence

Restricted Classes: The following classes have been restricted. The relationships with target class "WorkflowService" must be external references, they cannot be included in an implementation of the view. These were restricted as no specific type of WorkflowService relating to an Instrument has been created. Conceptualinstrument can only include Design, Algorithm, Result, and Precondition only by use of an external reference.

General Documentation The general top-level entry points are ImplementedInstrument or ConceptualInstrument for content and PrecedesIntervalRelation for the ordering of Sequences and other control construct types.

A functional view is a collection of classes in DDI that covers a functional use case. These are not namespaces.

7.2.1 Binding

Mapping between Input and Output Parameters of Workflow Steps representing the flow of information within a Workflow.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n
type	CorrespondenceType	0..1

displayLabel

A display label for the MemberCorrespondence. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

type

Type of correspondence in terms of commonalities and differences between two members.

‘

Relationships

Name	Type	Cardinality	allways external
input	InputParameter	0..n	no
ouput	OutputParameter	0..n	no
realizes	InformationFlow	0..n	yes

input

Specialization of maps in Information Flow for Input Parameters.

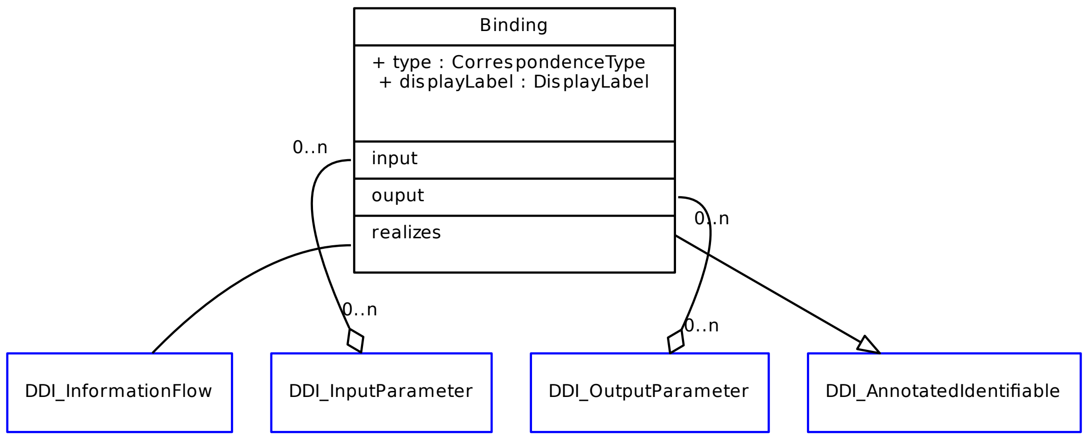
ouput

Specialization of maps in Information Flow for Output Parameters.

realizes

Class in the Process Pattern realized by Binding.

Graph



7.2.2 Category

A Concept whose role is to define and measure a characteristic.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
name	Name	0..n

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

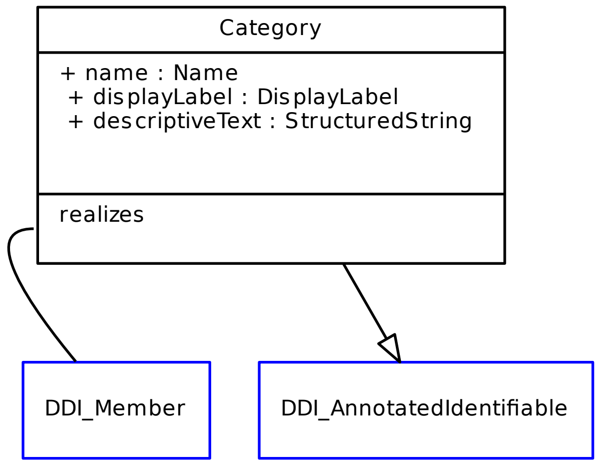
Relationships

Name	Type	Cardinality	allways external
realizes	Member	0..n	yes

realizes

Class can play the role of Member within a Collection

Graph


Category
+ name : Name
+ displayLabel : DisplayLabel
+ descriptiveText : StructuredString
realizes
DDI_Member
DDI_AnnotatedIdentifiable

7.2.3 CategorySet

A Category Set is a type of Node Set which groups Categories.

Extends

NodeSet

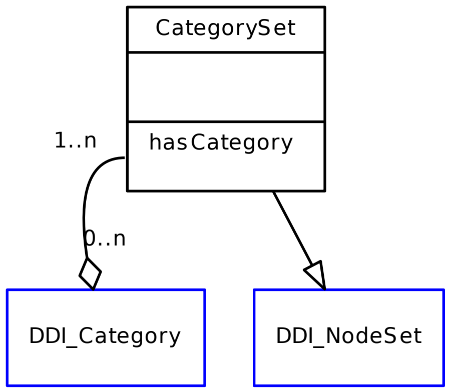
Relationships

Name	Type	Cardinality	allways external
hasCategory	Category	1..n	no

hasCategory

Specialization of contains in NodeSet for Categories.

Graph



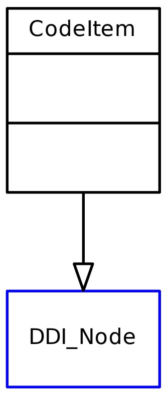
7.2.4 CodeItem

A type of Node contained in a CodeList that has a Code associated to a Category. In addition, a CodeItem can have a number of optional Designations. All Designations associated to the Code Item, including the Code, are synonyms, i.e. are associated with the same Concept.

Extends

:ref:‘Node

Graph



7.2.5 CodeList

A list of Codes and associated Categories. May be flat or hierarchical.

Extends

NodeSet

Relationships

Name	Type	Cardinality	allways external
contains	CodeItem	1..n	no
references	CategorySet	1..n	no
represents	ValueDomain	1..n	no

contains

Specialization of contains in NodeSet for CodeItems.

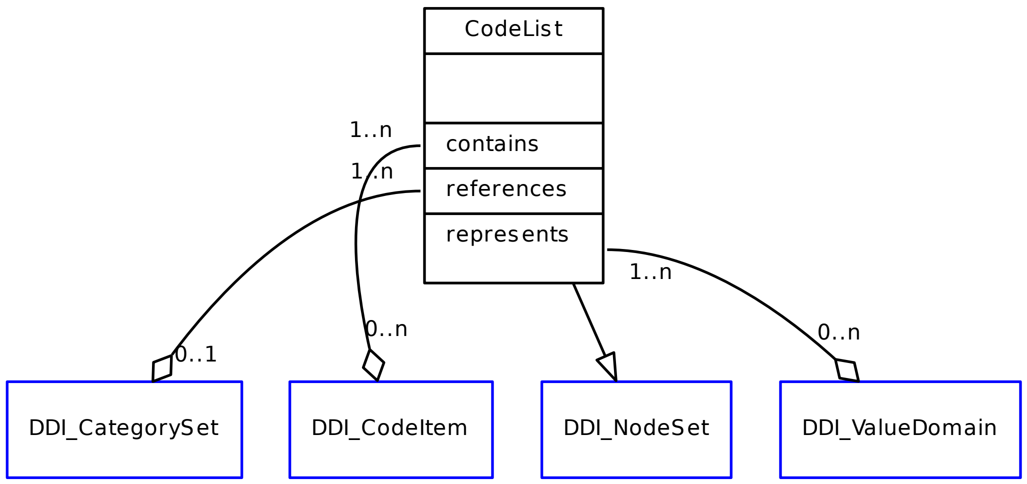
references

CategorySet associated with the CodeList.

represents

Enumerated Value Domain represented by the CodeList.

Graph



7.2.6 Concept

Unit of thought differentiated by characteristics [GSIM 1.1]

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
definition	StructuredString	0..1
name	Name	0..n

definition

Natural language statement conveying the meaning of a concept, differentiating it from other concepts. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

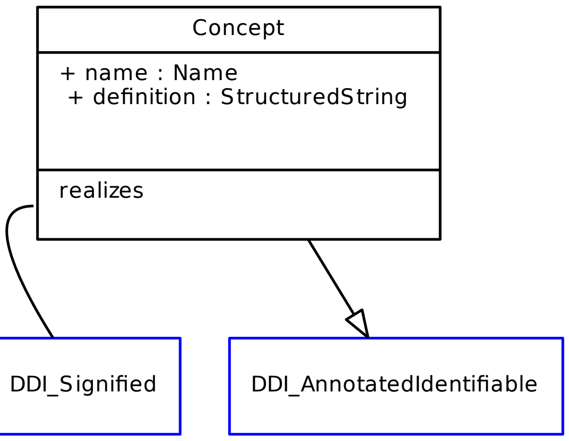
Relationships

Name	Type	Cardinality	allways external
realizes	Signified	0..n	yes

realizes

A concept can play the role of a signified when there is a designation that denotes it.

Graph



7.2.7 ConceptualInstrument

Design plan for creating a data capture tool.

Extends

Workflow

Properties

Name	Type	Cardinality
description	StructuredString	0..1
displayLabel	DisplayLabel	0..n
hasAnnotation	Annotation	0..1
name	Name	0..n
usage	StructuredString	0..1

description

A description of the purpose or use of the Member. May be expressed in multiple languages and supports the use of structured content.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

hasAnnotation

Allows for complete annotation of the Conceptual Instrument

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

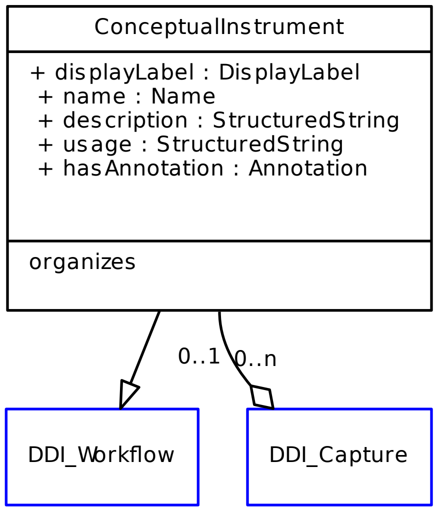
Relationships

Name	Type	Cardinality	allways external
organizes	Capture	0..1	no

organizes

Refers to the RepresentedMeasures and RepresentedQuestions used by the ConceptualInstrument.

Graph


ConceptualInstrument
+ displayLabel : DisplayLabel
+ name : Name
+ description : StructuredString
+ usage : StructuredString
+ hasAnnotation : Annotation
organizes
0..1
0..n
DDI_Workflow
DDI_Capture

7.2.8 ConceptualVariable

The use of a Concept as a characteristic of a Universe intended to be measured [GSIM 1.1]

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
name	Name	0..n

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

Relationships

Name	Type	Cardinality	allways external
realizes	Member	0..n	yes
takesSentinelConceptsFrom	SentinelConceptualDomain	0..n	no
takesSubstantiveConceptsFrom	SubstantiveConceptualDomain	1..n	no
usesConcept	Concept	0..n	no
usesUnitType	UnitType	0..n	no

realizes

Class can play the role of Member within a Collection

takesSentinelConceptsFrom

Identifies the ConceptualDomain containing the set of sentinel concepts used to describe the ConceptualVariable.

takesSubstantiveConceptsFrom

Identifies the ConceptualDomain containing the set of substantive concepts used to describe the ConceptualVariable.

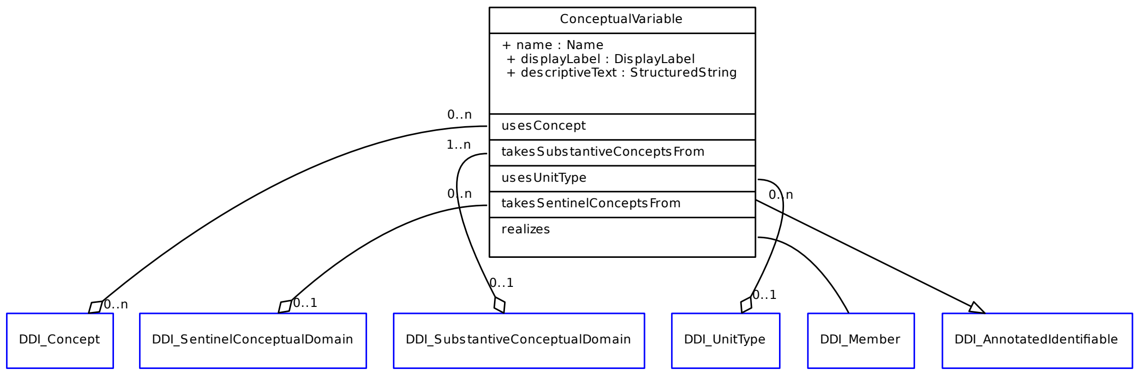
usesConcept

Reference to a Concept that is being used

usesUnitType

Identifies the UnitType associated with the ConceptualVariable

Graph


ConceptualVariable
+ name : Name
+ displayLabel : DisplayLabel
+ descriptiveText : StructuredString
usesConcept
takesSubstantiveConceptsFrom
usesUnitType
takesSentinelConceptsFrom
realizes
0..n
1..n
0..n
0..n
0..n
0..1
0..1
0..1
DDI_Concept
DDI_SentinelConceptualDomain
DDI_SubstantiveConceptualDomain
DDI_UnitType
DDI_Member
DDI_AnnotatedIdentifiable

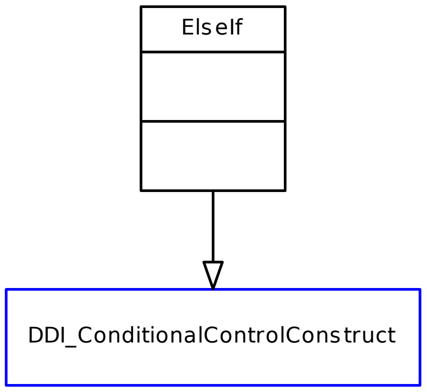
7.2.9 ElseIf

Conditional Control Construct that is evaluated if the condition in the parent IfThenElse is false. It cannot exist without a parent IfThenElse.

Extends

:ref:‘ConditionalControlConstruct

Graph



7.2.10 ExternalAid

Any external material used in an instrument that aids or facilitates data capture, or that is presented to a respondent and about which measurements are made.

Extends

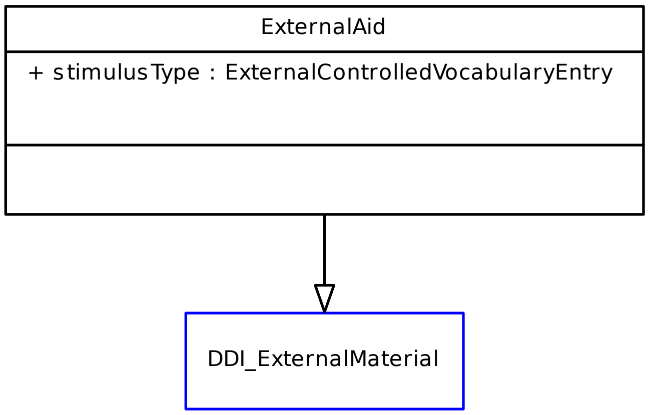
ExternalMaterial

Properties

Name	Type	Cardinality
stimulusType	ExternalControlledVocabularyEntry	0..1

stimulusType

Graph



7.2.11 ExternalMaterial

ExternalMaterial describes the location, structure, and relationship to the DDI metadata instance for any material held external to that instance. This includes citations to such material, an external reference to a URL (or other URI), and a statement about the relationship between the cited ExternalMaterial the contents of the DDI instance. It should be used as follows: As an extension base for specific external materials found within DDI (such as an External Aid); as a target object from a relationship which clarifies its role within a class; or as the target of a relatedResource within an annotation.

Extends

Identifiable

Properties

Name	Type	Cardinality
citationOfExternalMaterial	Annotation	0..1
descriptiveText	StructuredString	0..1
externalURLReference	anyURI	0..n
externalURNReference	anyURI	0..1
mimeType	ExternalControlledVocabularyEntry	0..1
relationshipDescription	StructuredString	0..n
segment	Segment	0..n
typeOfMaterial	ExternalControlledVocabularyEntry	0..1

citationOfExternalMaterial

Bibliographic citation for the external resource.

descriptiveText

A description of the referenced material. This field can map to a Dublin Core abstract. Note that Dublin Core does not support structure within the abstract element. Supports multiple languages and optional structured content.

externalURLReference

Contains a URL which indicates the location of the cited external resource.

externalURNReference

Contains a URN which identifies the cited external resource.

mimeType

Provides a standard Internet MIME type for use by processing applications.

relationshipDescription

Describes the reason the external material is being related to the DDI metadata instance.

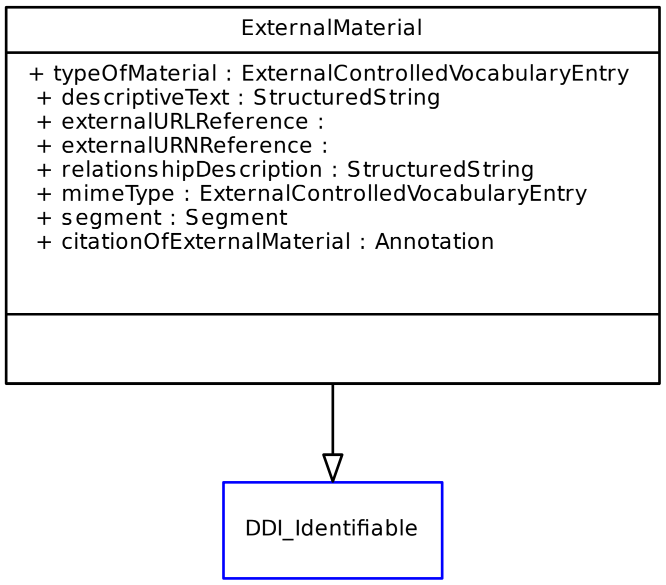
segment

Can describe a segment within a larger object such as a text or video segment.

typeOfMaterial

Designation of the type of material being described. Supports the use of a controlled vocabulary.

Graph


ExternalMaterial
+ typeOfMaterial : ExternalControlledVocabularyEntry
+ descriptiveText : StructuredString
+ externalURLReference :
+ externalURNReference :
+ relationshipDescription : StructuredString
+ mimeType : ExternalControlledVocabularyEntry
+ segment : Segment
+ citationOfExternalMaterial : Annotation
DDI_Identifiable

7.2.12 IfThenElse

IfThenElse describes an if-then-else decision type of control construct. If the stated condition is met, then the associated Workflow Sequence in contains (inherited from Conditional Control Construct) is triggered, otherwise the Workflow Sequence that is triggered is the one associated via elseContains.

Extends

ConditionalControlConstruct

Relationships

Name	Type	Cardinality	allways external
elseContains	WorkflowSequence	0..n	no
hasElseIf	ElseIf	1..1	no

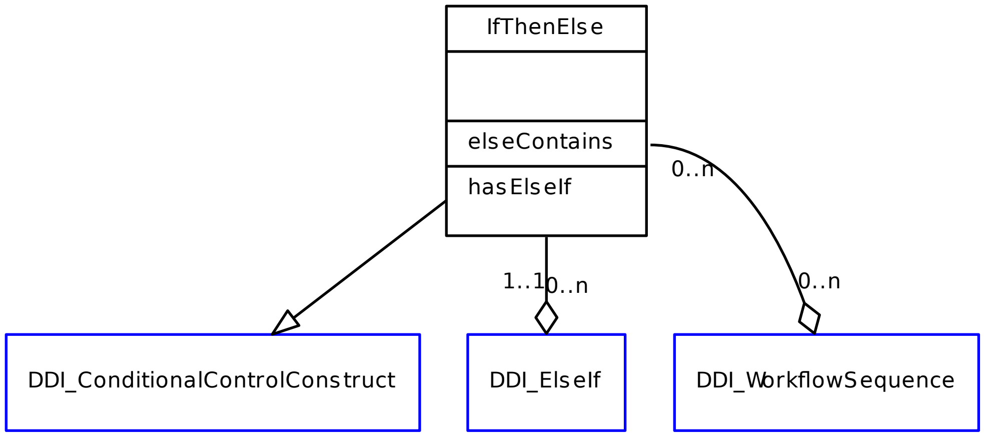
elseContains

Optional sequence of Process Steps that are triggered when the condition is false.

hasElseIf

Set of ElseIf that are evaluated if the condition is false.

Graph



7.2.13 ImplementedInstrument

ImplementedInstruments are mode and/or unit specific.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n
externalInstrumentLocation	anyURI	0..n
name	Name	0..n
typeOfInstrument	ExternalControlledVocabularyEntry	0..1
usage	StructuredString	0..n

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

externalInstrumentLocation

A reference to an external representation of the the data collection instrument, such as an image of a questionnaire or programming script.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

typeOfInstrument

Specification of the type of instrument according to the classification system of the documentor.

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

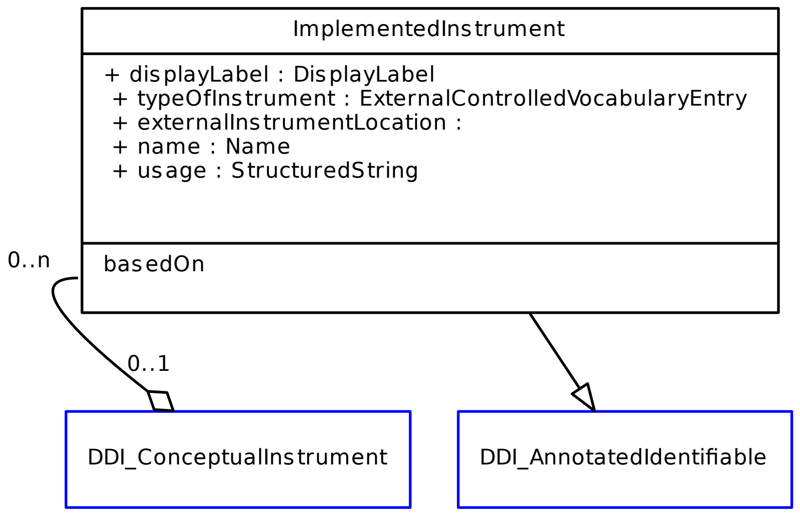
Relationships

Name	Type	Cardinality	allways external
basedOn	ConceptualInstrument	0..n	no

basedOn

The ConceptualInstrument which informs the design of the ImplementedInstrument.

Graph


ImplementedInstrument
+ displayLabel : DisplayLabel
+ typeOfInstrument : ExternalControlledVocabularyEntry
+ externalInstrumentLocation :
+ name : Name
+ usage : StructuredString
basedOn
0..n
0..1
DDI_ConceptualInstrument
DDI_AnnotatedIdentifiable

7.2.14 InputParameter

Generic container for an input instance to a Process Step.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
type	ExternalControlledVocabularyEntry	0..n

type

Type of object the parameter accepts (DDI object, Datatype, etc.). If not present the parameter is untyped.

‘

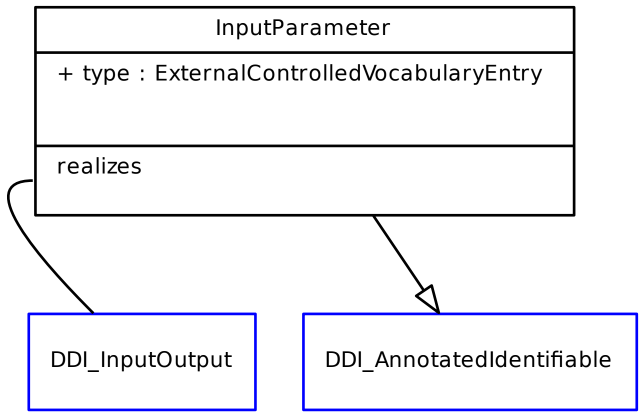
Relationships

Name	Type	Cardinality	allways external
realizes	InputOutput	0..n	yes

realizes

Class in the Process Pattern to be realized by Input Parameter.

Graph


InputParameter
+ type : ExternalControlledVocabularyEntry
realizes
DDI_InputOutput
DDI_AnnotatedIdentifiable

7.2.15 InstanceQuestion

An instance question is an instantiation of a represented question,to be used as an Act in the process steps that define a survey questionnaire.

Extends

InstrumentComponent

Properties

Name	Type	Cardinality
name	Name	0..n

name

The name of a question as used in an Instrument. Redefined to provide a more useful description. A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

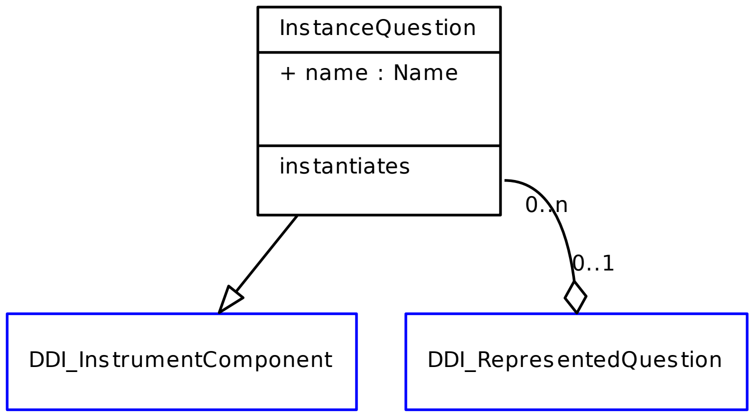
Relationships

Name	Type	Cardinality	allways external
instantiates	RepresentedQuestion	0..n	no

instantiates

The question being used.

Graph



7.2.16 Instruction

Provides the content and description of data capture instructions. Contains the “how to” information for administering an instrument.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n
instructionText	DynamicText	0..n
name	Name	0..n

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

instructionText

The content of the Instruction text provided using DynamicText. Note that when using Dynamic Text, the full InstructionText must be repeated for multi-language versions of the content. The InstructionText may also be repeated to provide a dynamic and plain text version of the instruction. This allows for accurate rendering of the instruction in a non-dynamic environment like print.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

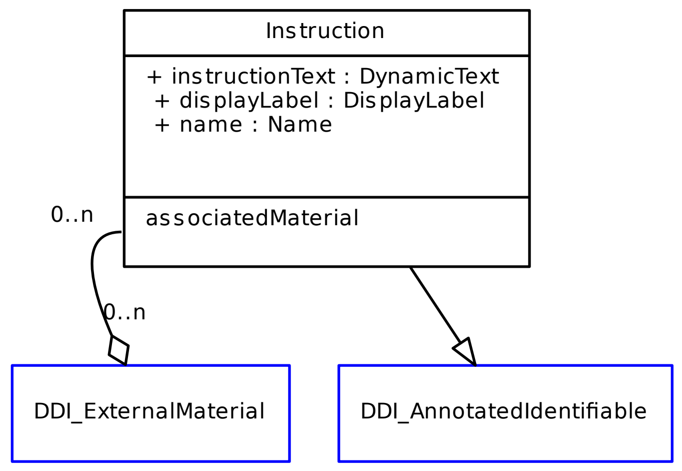
Relationships

Name	Type	Cardinality	allways external
associatedMaterial	ExternalMaterial	0..n	no

associatedMaterial

An image or other external material associated with the Instruction, located at the provided URN or URL.

Graph


Instruction
+ instructionText : DynamicText
+ displayLabel : DisplayLabel
+ name : Name
associatedMaterial
0..n
0..n
DDI_ExternalMaterial
DDI_AnnotatedIdentifiable

7.2.17 InstrumentCode

Extends

InstrumentComponent

Properties

Name	Type	Cardinality
commandCode	CommandCode	0..1
purposeOfCode	ExternalControlledVocabularyEntry	0..1

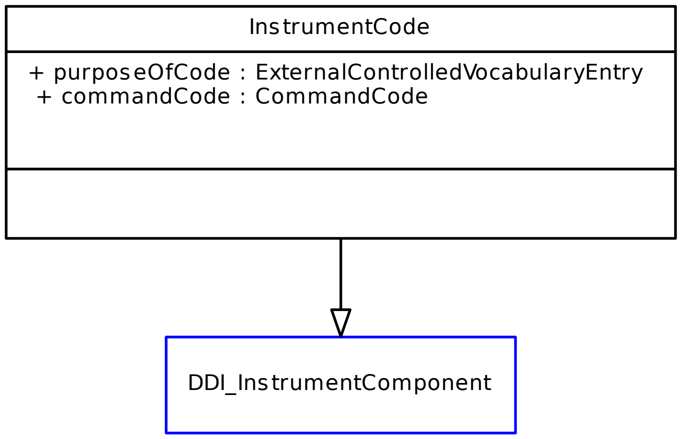
commandCode

Describes the code used to execute the command using the options of inline textual description, inline code, and/or an external file.

purposeOfCode

The purpose of the code (e.g., quality control, edit check, checksums, compute filler text, compute values for use in administering the instrument)

Graph


InstrumentCode
+ purposeOfCode : ExternalControlledVocabularyEntry
+ commandCode : CommandCode
DDI_InstrumentComponent

7.2.18 Level

The Level describes the nesting structure of a hierarchical collection.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
groups	Node	1..1	no
isDefinedBy	Concept	0..n	no
realizes	Collection	0..n	yes

groups

Realization of contains in Collection for Nodes.

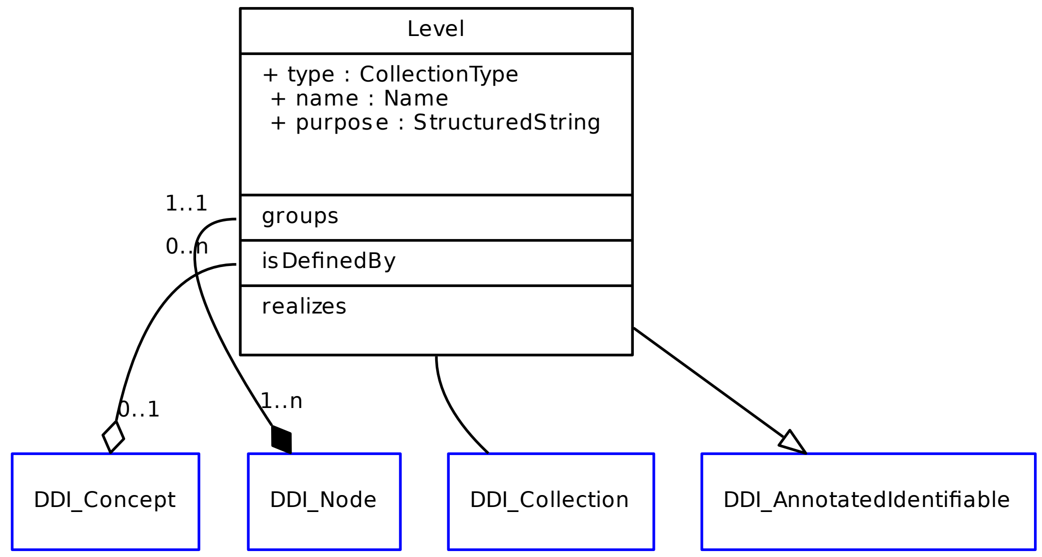
isDefinedBy

Associated concept that provides the conceptual definition of the level.

realizes

Class of the Collection pattern realized by this class.

Graph



7.2.19 Loop

Iterative control structure to be repeated a specified number of times based on one or more conditions. Inside the loop, one or more Workflow Steps are evaluated and processed in the order they appear.

Extends

ConditionalControlConstruct

Properties

Name	Type	Cardinality
initialValue	CommandCode	0..1
stepValue	CommandCode	0..1

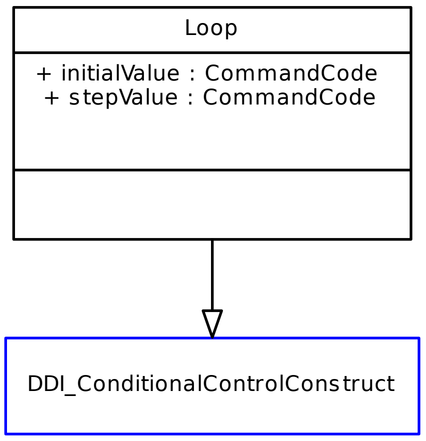
initialValue

The command used to set the initial value for the process. Could be a simple value.

stepValue

The command used to set the incremental or step value for the process. Could be a simple value.

Graph


Loop
+ initialValue : CommandCode
+ stepValue : CommandCode
DDI_ConditionalControlConstruct

7.2.20 OrderedIntervalPair

Ordered pair of Process Steps in a Temporal Interval Relation.

Extends

Identifiable

Relationships

Name	Type	Cardinality	allways external
realizes	OrderedPair	0..n	yes
source	WorkflowStep	0..n	no
target	WorkflowStep	0..n	no

realizes

Class of the Collection pattern realized by this class.

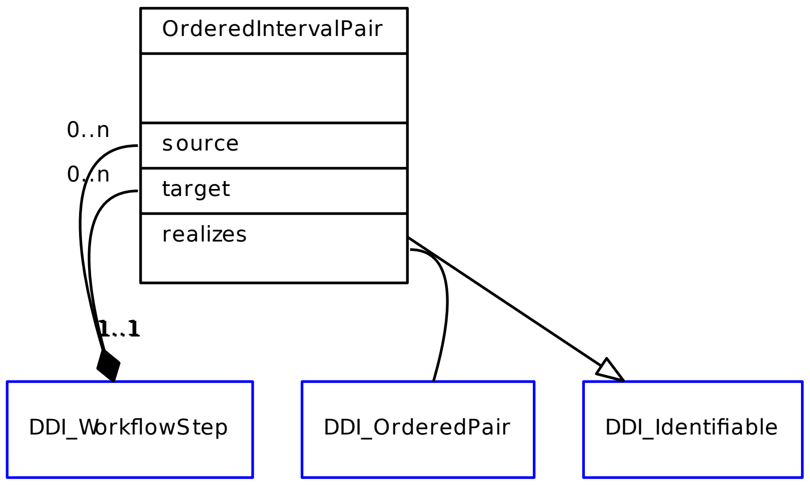
source

First Process Step in the Ordered Pair. Process Step that either starts or ends before the Process Step in the target.

target

Second Process Step in the Ordered Pair. Process Step that either starts or ends after the Process Step in the source.

Graph



7.2.21 OutputParameter

Generic container for an output instance of a Process Step.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
type	ExternalControlledVocabularyEntry	0..n

type

Type of object the parameter accepts (DDI object, Datatype, etc.). If not present the parameter is untyped.

‘

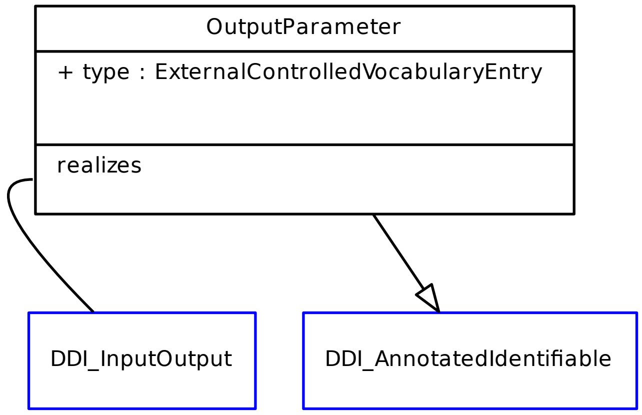
Relationships

Name	Type	Cardinality	allways external
realizes	InputOutput	0..n	yes

realizes

Class of the Process Pattern to be realized by Output Parameter.

Graph


OutputParameter
+ type : ExternalControlledVocabularyEntry
realizes
DDI_InputOutput
DDI_AnnotatedIdentifiable

7.2.22 Parameters

Collection of Input and Output Parameters of Workflow Steps.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
type	CollectionType	0..1

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
input	InputParameter	0..n	no
output	OutputParameter	0..n	no
realizes	Interface	0..n	yes

input

Realization of contains in Interface for Input Parameters.

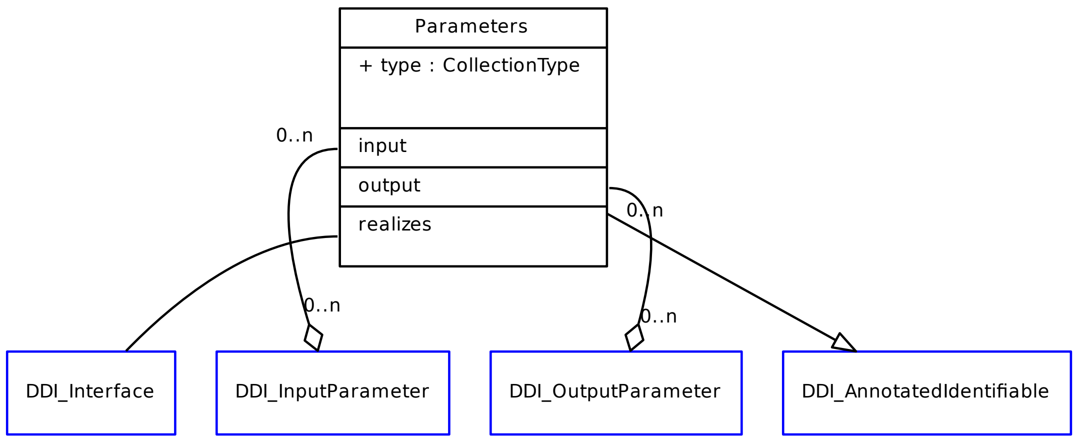
output

Realization of contains in Interface for Output Parameters.

realizes

Class in the Process Pattern realized by Parameters.

Graph



7.2.23 PrecedesIntervalRelation

Representation of the precedes relation in Allen’s interval algebra.

We say that an interval A precedes another interval B if and only if A finishes before B begins.

More precisely, A.end < B.start.

Instead of saying that A precedes B we can also say that B is preceded by A (converse).

Extends

TemporalIntervalRelation

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	1..1
symmetry	SymmetryType	1..1
transitivity	TransitivityType	1..1
usage	StructuredString	0..1

reflexivity

Fixed to Anti-Reflexive

semantics

Fixed to Temporal-Precedes

symmetry

Fixed to Anti-Symmetric

transitivity

Fixed to Transitive

usage

Explanation of the ways in which some decision or object is employed. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality	allways external
contains	OrderedIntervalPair	0..n	no
realizes	StrictOrderRelation	0..n	yes
structures	WorkflowSequence	0..n	no

contains

Pairs of Process Steps in the temporal relation.

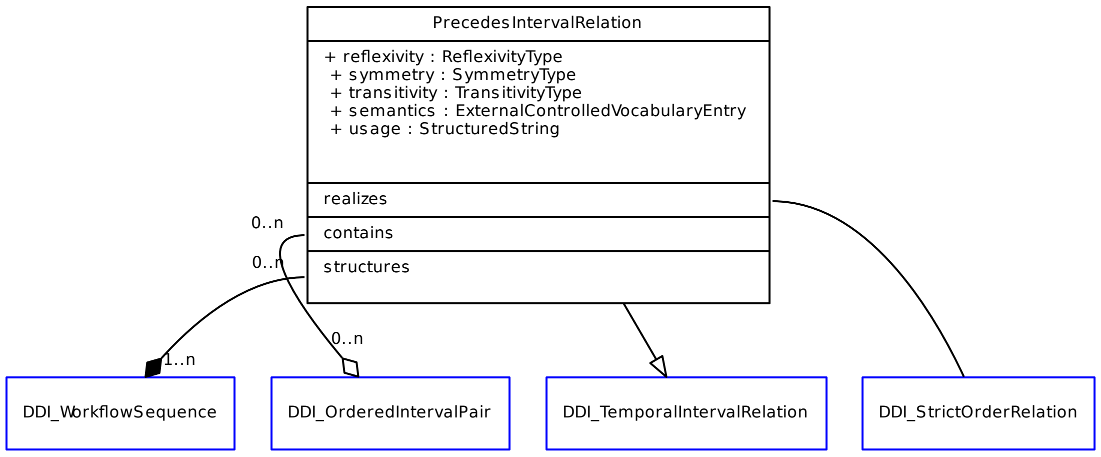
realizes

Class of the Collection pattern realized by this class.

structures

Sequence to which the Temporal Interval Relation is applied.

Graph


PrecedesIntervalRelation
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
+ usage : StructuredString
realizes
contains
structures
0..n
0..n
0..n
1..n
DDI_WorkflowSequence
DDI_OrderedIntervalPair
DDI_TemporalIntervalRelation
DDI_StrictOrderRelation

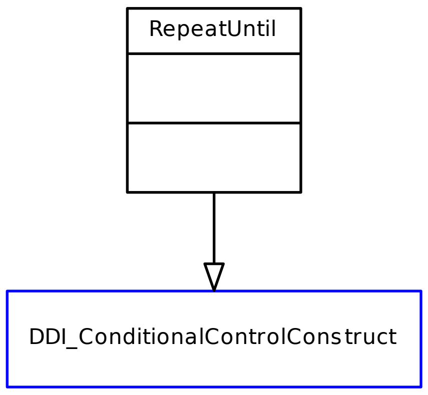
7.2.24 RepeatUntil

Iterative control structure to be repeated until a specified condition is met. Before each iteration the condition is tested. If the condition is not met, the associated Workflow Sequence in contains (inherited from Conditional Control Construct) is triggered. When the condition is met, control passes back to the containing Workflow Step.

Extends

:ref:‘ConditionalControlConstruct

Graph



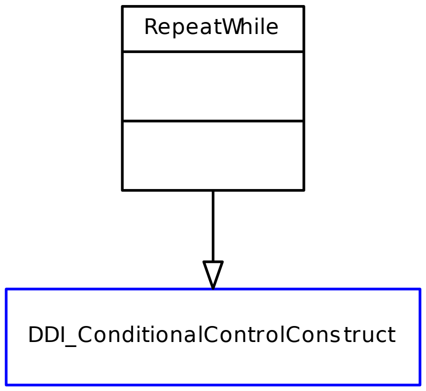
7.2.25 RepeatWhile

Iterative control structure to be repeated while a specified condition is met. Before each iteration the condition is tested. If the condition is met, the associated Workflow Sequence in contains (inherited from Conditional Control Construct) is triggered. When the condition is not met, control passes back to the containing Workflow Step.

Extends

:ref:‘ConditionalControlConstruct

Graph



7.2.26 RepresentedQuestion

The description of a reusable question, that can be used as a template that describes the components of a question

Extends

Capture

Properties

Name	Type	Cardinality
estimatedResponseTimeInSeconds	Real	0..1
questionIntent	StructuredString	0..1
questionText	DynamicText	0..n

estimatedResponseTimeInSeconds

An estimation of the number of seconds required to respond to the question. Used for estimating overall questionnaire completion time.

questionIntent

The purpose or intent of the question.

questionText

The text of the question which may be literal or dynamic (altered to personalize the question text) in terms of content.

‘

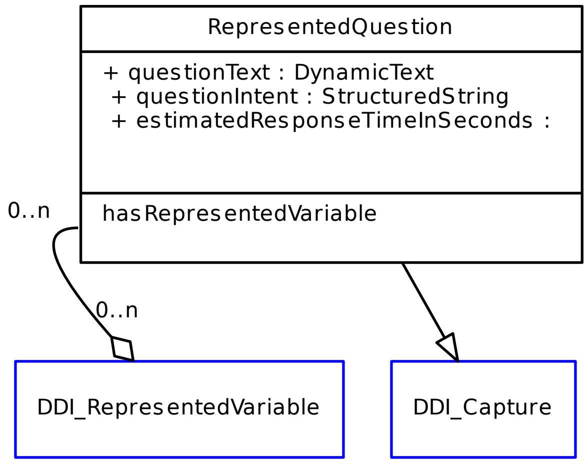
Relationships

Name	Type	Cardinality	allways external
hasRepresentedVariable	RepresentedVariable	0..n	no

hasRepresentedVariable

An optional link to a represented variable which can be used by each instance variable created by a use of this question.

Graph


RepresentedQuestion
+ questionText : DynamicText
+ questionIntent : StructuredString
+ estimatedResponseTimeInSeconds :
hasRepresentedVariable
0..n
0..n
DDI_RepresentedVariable
DDI_Capture

7.2.27 RepresentedVariable

A combination of a characteristic of a universe to be measured and how that measure will be represented.

Extends

ConceptualVariable

Properties

Name	Type	Cardinality
hasIntendedDataType	ExternalControlledVocabularyEntry	0..1
unitOfMeasurement	String	0..1

hasIntendedDataType

The data type intended to be used by this variable. Supports the optional use of an external controlled vocabulary.

unitOfMeasurement

The unit in which the data values are measured (kg, pound, euro).

‘

Relationships

Name	Type	Cardinality	allways external
basedOnConceptualVariable	ConceptualVariable	0..n	no
measures	Universe	0..n	no
takesSubstantiveValuesFrom	SubstantiveValueDomain	0..n	no

basedOnConceptualVariable

The ConceptualVariable that can be shared by a set of multiple RepresentedVariables. Indicates comparability in ConceptualDomain and UnitType.

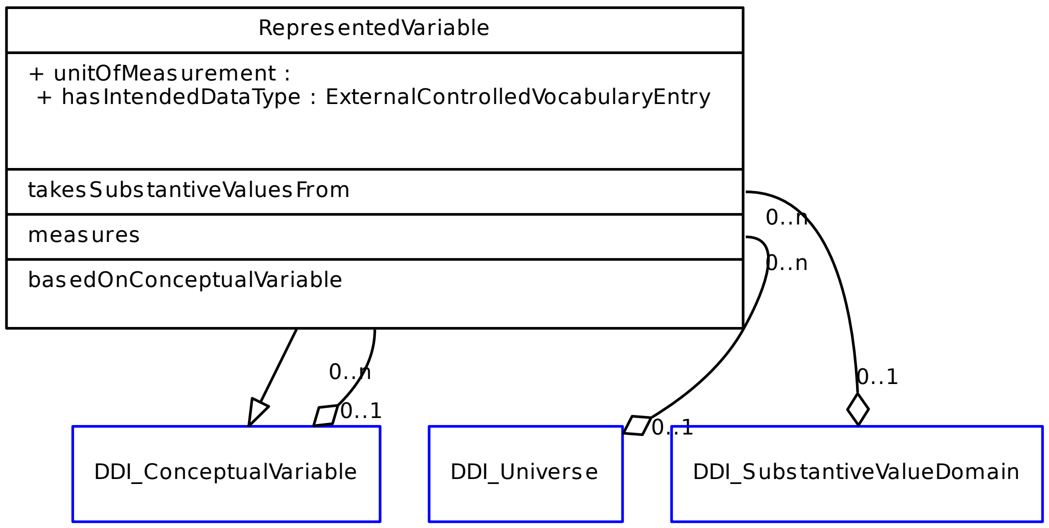
measures

The defined class of people, entities, events, or objects to be measured.

takesSubstantiveValuesFrom

The substantive representation (SubstantiveValueDomain) of the variable. This is equivalent to the relationship “Measures” in GSIM although GSIM makes no distinction between substantive and sentinel values

Graph


RepresentedVariable
+ unitOfMeasurement :
+ hasIntendedDataType : ExternalControlledVocabularyEntry
takesSubstantiveValuesFrom
measures
basedOnConceptualVariable
0..n
0..1
0..n
0..1
0..n
0..1
DDI_ConceptualVariable
DDI_Universe
DDI_SubstantiveValueDomain

7.2.28 ResponseDomain

The possible list of values that are allowed by a Capture.

Extends

AnnotatedIdentifiable

Relationships

Name	Type	Cardinality	allways external
takesSentinelResponsesFrom	SentinelValueDomain	0..n	no
takesSubstantiveResponsesFrom	SubstantiveValueDomain	0..n	no

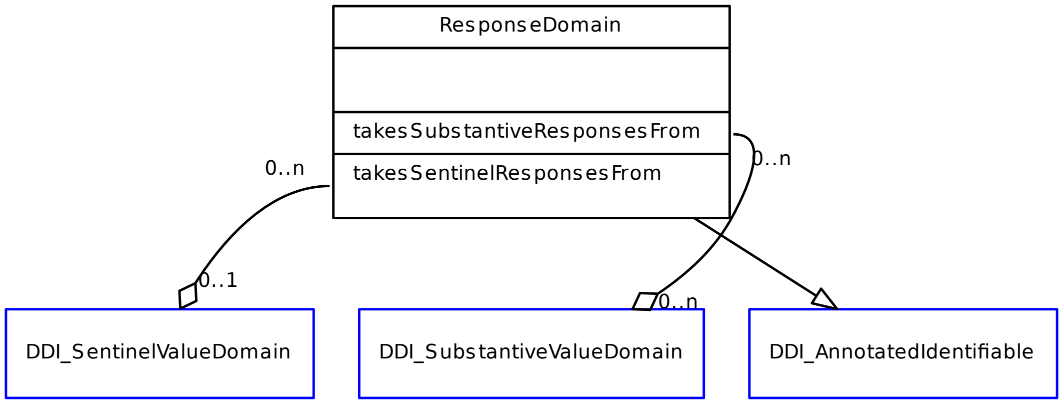
takesSentinelResponsesFrom

The sentinel values (e.g. missing codes) are drawn from this SubstantiveValueDomain

takesSubstantiveResponsesFrom

The values of interest are drawn from these SubstantiveValueDomains

Graph


ResponseDomain
takesSubstantiveResponsesFrom
takesSentinelResponsesFrom
0..n
0..1
0..n
0..n
DDI_SentinelValueDomain
DDI_SubstantiveValueDomain
DDI_AnnotatedIdentifiable

7.2.29 SentinelConceptualDomain

Description or list of possible sentinel concepts , e.g. missing values.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality	allways external
describedConceptualDomain	ValueAndConceptDescription	0..n	no
enumeratedConceptualDomain	CategorySet	0..n	no

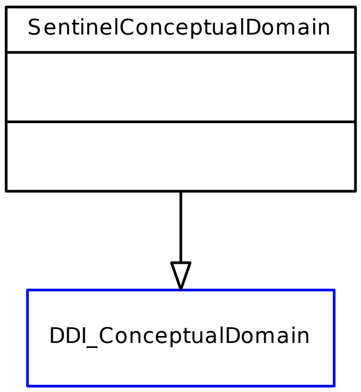
describedConceptualDomain

A description of the concepts in the domain. A numeric domain might use a logical expression to be machine actionable a text domain might use a regular expression to describe strings that describe the concepts.

enumeratedConceptualDomain

The CategorySet containing the concepts in the domain.

Graph



7.2.30 SentinelValueDomain

The Value Domain for a sentinel conceptual domain. Sentinel values are defined in ISO 11404 as “element of a value space that is not completely consistent with a datatype’s properties and characterizing operations...”. A common example would be codes for missing values.

Extends

ValueDomain

Relationships

Name	Type	Cardinality	allways external
describedValueDomain	ValueAndConceptDescription	0..n	no
enumeratedValueDomain	CodeList	0..n	no
takesConceptsFrom	SentinelConceptualDomain	0..n	no

describedValueDomain

A formal description of the set of valid values - for described value domains.

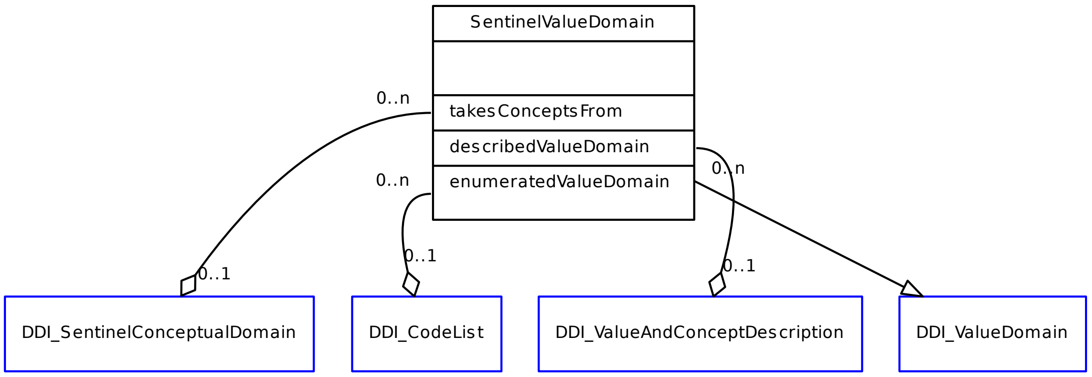
enumeratedValueDomain

A CodeList enumerating the set of valid values.

takesConceptsFrom

Corresponding conceptual definition given by a SentinelConceptualDomain.

Graph



7.2.31 Statement

A Statement is human readable text or referred material.

Extends

InstrumentComponent

Properties

Name	Type	Cardinality
purposeOfStatement	ExternalControlledVocabularyEntry	0..1
statementText	DynamicText	0..n

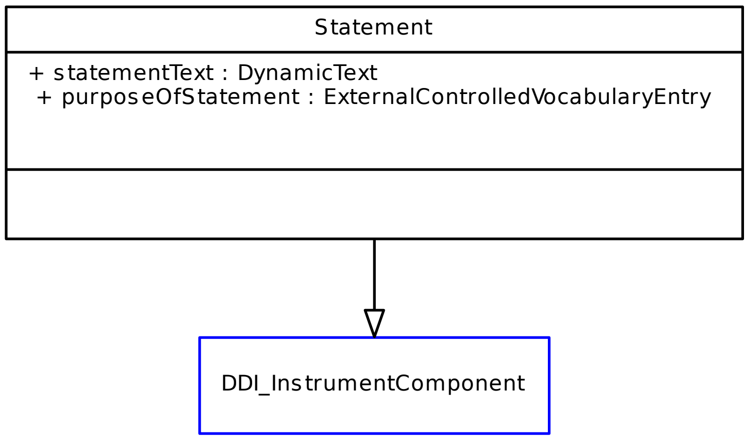
purposeOfStatement

Describes the use of the statement within the instrument. For example, a section divider, welcome statement, or other text.

statementText

Structured human-readable text that allows for dynamic content. For example, the insertion of a name or gender specific pronoun. Repeat ONLY for capturing the same content in multiple languages.

Graph


Statement
+ statementText : DynamicText
+ purposeOfStatement : ExternalControlledVocabularyEntry
DDI_InstrumentComponent

7.2.32 SubstantiveConceptualDomain

Set of valid Concepts. The Concepts can be described by either enumeration or by an expression.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n

displayLabel

A display label for the Conceptual Domain. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

‘

Relationships

Name	Type	Cardinality	allways external
describedConceptualDomain	ValueAndConceptDescription	0..n	no
enumeratedConceptualDomain	CategorySet	0..n	no

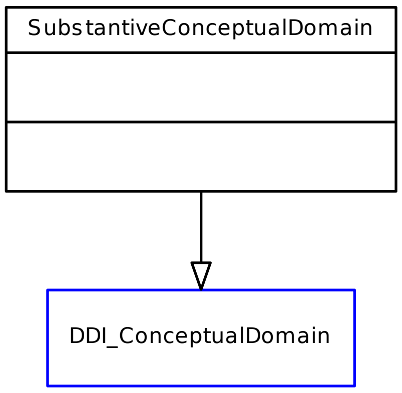
describedConceptualDomain

A description of the concepts in the domain. A numeric domain might use a logical expression to be machine actionable a text domain might use a regular expression to describe strings that describe the concepts.

enumeratedConceptualDomain

The CategorySet containing the concepts in the domain.

Graph



7.2.33 SubstantiveValueDomain

The Value Domain for a substantive conceptual domain.

Extends

ValueDomain

Relationships

Name	Type	Cardinality	allways external
describedValueDomain	ValueAndConceptDescription	0..n	no
enumeratedValueDomain	CodeList	0..n	no
takesConceptsFrom	SubstantiveConceptualDomain	0..n	no

describedValueDomain

A formal description of the set of valid values - for described value domains.

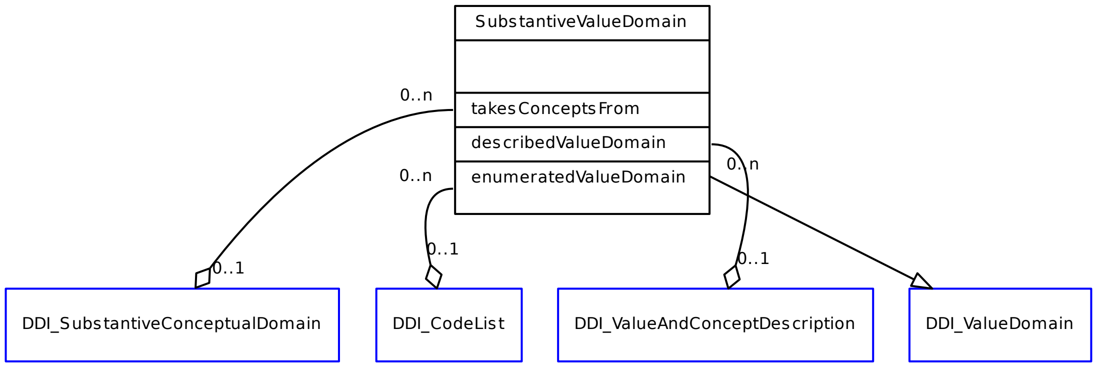
enumeratedValueDomain

A CodeList enumerating the set of valid values.

takesConceptsFrom

Corresponding conceptual definition given by an SubstantiveConceptualDomain.

Graph



7.2.34 UnitType

A Unit Type is a class of objects of interest.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
name	Name	0..1

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

Relationships

Name	Type	Cardinality	allways external
realizes	Member	0..n	yes
usesConcept	Concept	0..n	no

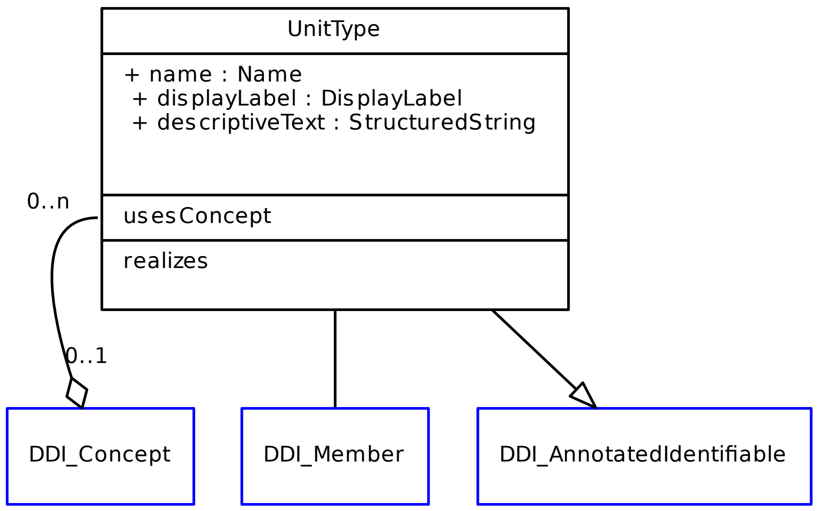
realizes

Class can be used in the role of Member within a Collection

usesConcept

Reference to the Concept that is being used

Graph


UnitType
+ name : Name
+ displayLabel : DisplayLabel
+ descriptiveText : StructuredString
usesConcept
realizes
0..n
0..1
DDI_Concept
DDI_Member
DDI_AnnotatedIdentifiable

7.2.35 Universe

A defined class of people, entities, events, or objects, with no specification of time and geography, contextualizing a Unit Type

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
isInclusive	Boolean	0..1
name	Name	0..n

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

isInclusive

The default value is “true”. The description statement of a universe is generally stated in inclusive terms such as “All persons with university degree”. Occasionally a universe is defined by what it excludes, i.e., “All persons except those with university degree”. In this case the value would be changed to “false”.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

Relationships

Name	Type	Cardinality	allways external
hasSubuniverse	Universe	1..1	no
narrowsUnitType	UnitType	0..n	no
realizes	Member	0..n	yes
usesConcept	Concept	0..n	no

hasSubuniverse

Universes can have sub universes. A sub-universe class provides a definition to the universes contained within it. For example the Sub-Universe Class of Gender for the Universe Population may contain the Universe Males and the Universe Females

narrowsUnitType

Reference to the Unit Type that the Universe definition narrows.

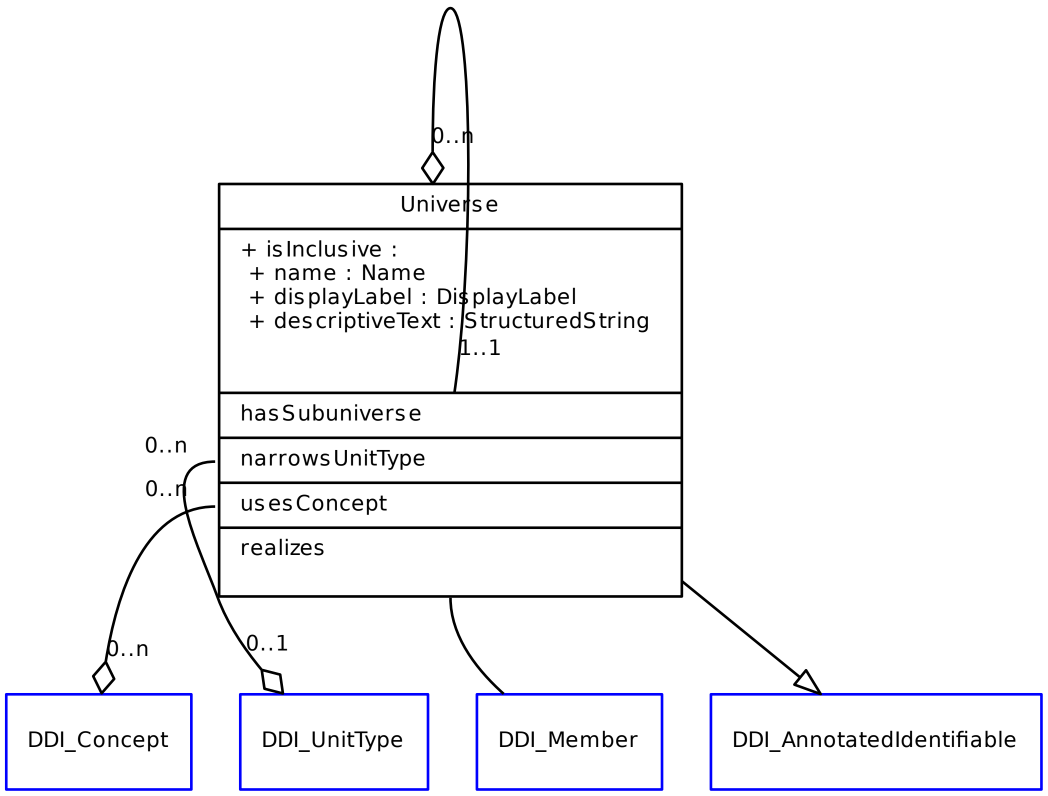
realizes

Class can be used in the role of Member within a Collection

usesConcept

Reference to the Concept that is being used

Graph



7.2.36 ValueAndConceptDescription

A formal description of a set of values.

Extends

Identifier

Properties

Name	Type	Cardinality
classificationLevel	CategoryRelationCode	0..1
description	StructuredString	0..1
logicalExpression	String	0..1
maximumValueExclusive	String	0..1
maximumValueInclusive	String	0..1
minimumValueExclusive	String	0..1
minimumValueInclusive	String	0..1
regularExpression	String	0..1

classificationLevel

Indicates the type of relationship, nominal, ordinal, interval, ratio, or continuous. Use where appropriate for the representation type.

description

A formal description of the set of values.

logicalExpression

A logical expression where the values of "x" making the expression true are the members of the set of valid values. Example: x >0 describes the real numbers greater than 0

maximumValueExclusive

A string denotFrom https://www.w3.org/TR/tabular-metadata/ 5.11.2 maxExclusive: "An atomic property that contains a single number or string that is the maximum valid value (exclusive). The value of this property becomes the maximum exclusive annotation for the described datatype. See Value Constraints in [tabular-data-model] for details." DDI3.2 handles this with a Boolean isInclusive attribute. ing the maximumpossible value (excluding this value)

maximumValueInclusive

A string denoting the maximum possible value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 maximum: "An atomic property that contains a single number or string that is the maximum valid value (inclusive); equivalent to maxInclusive. The value of this property becomes the maximum annotation for the described datatype. See Value Constraints in [tabular-data-model] for details."

minimumValueExclusive

A string denoting the minimum possible value (excluding this value) From https://www.w3.org/TR/tabular-metadata/ 5.11.2 minExclusive: "An atomic property that contains a single number or string that is the minimum valid value (exclusive). The value of this property becomes the minimum exclusive annotation for the described datatype. See Value Constraints in [tabular-data-model] for details." DDI3.2 handles this with a Boolean isInclusive attribute.

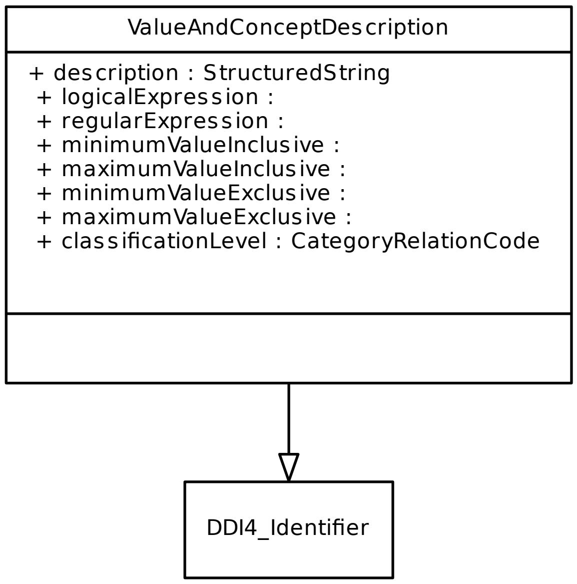
minimumValueInclusive

A string denoting the minimum possible value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 minimum: "An atomic property that contains a single number or string that is the minimum valid value (inclusive); equivalent to minInclusive. The value of this property becomes the minimum annotation for the described datatype. See Value Constraints in [tabular-data-model] for details."

regularExpression

A regular expression where strings matching the expression belong to the set of valid values.

Graph


ValueAndConceptDescription
+ description : StructuredString
+ logicalExpression :
+ regularExpression :
+ minimumValueInclusive :
+ maximumValueInclusive :
+ minimumValueExclusive :
+ maximumValueExclusive :
+ classificationLevel : CategoryRelationCode
DDI4_Identifier

7.2.37 WorkflowSequence

Sequencing of Workflow Steps that can be defined either by Temporal or Order Relations.

The sequence can be typed to support local processing or classification flags and alternate sequencing instructions (such as randomize for each respondent).

Extends

ControlConstruct

Properties

Name	Type	Cardinality
constructSequence	SpecificSequence	0..1
type	CollectionType	0..1
typeOfSequence	ExternalControlledVocabularyEntry	0..n

constructSequence

Describes alternate ordering for different cases using the SpecificSequence structure. If you set the sequence to anything other than order of appearance the only allowable children are QuestionConstruct or Sequence. Contents must be randomizable.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

typeOfSequence

Provides the ability to “type” a sequence for classification or processing purposes. Supports the use of an external controlled vocabulary.

‘

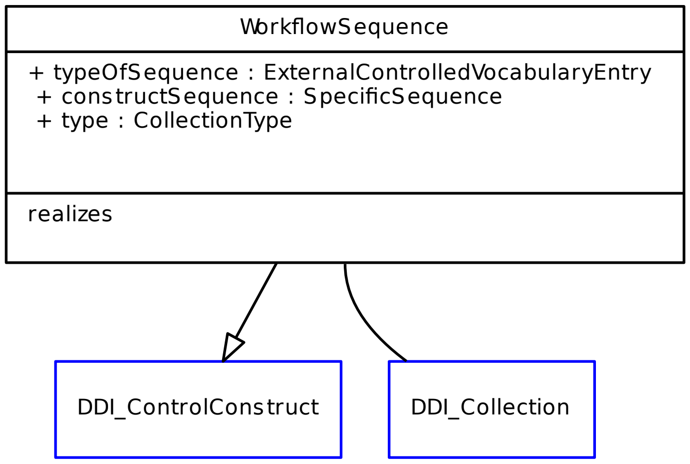
Relationships

Name	Type	Cardinality	allways external
realizes	Collection	0..n	yes

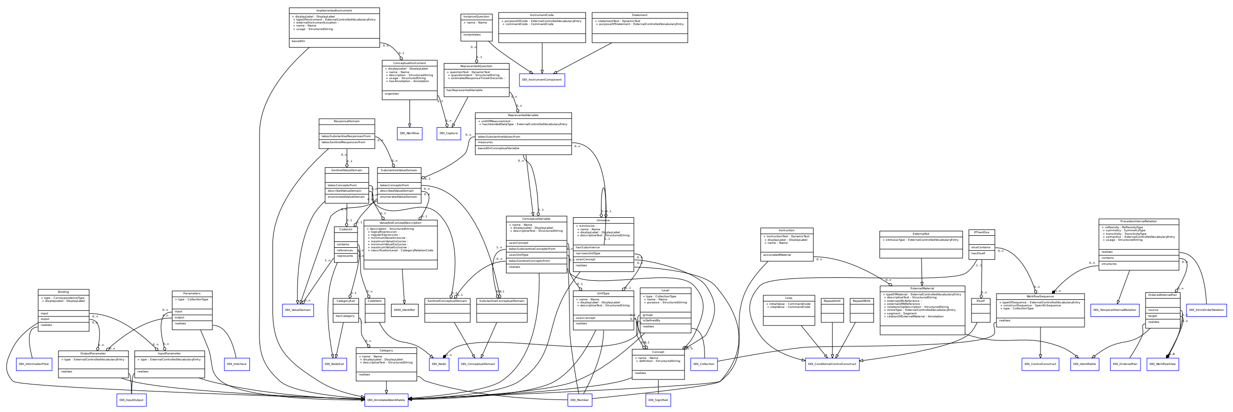
realizes

Class in the Collections pattern realized by this class. A Workflow Sequence can be viewed as a Collection of Workflow Steps.

Graph


WorkflowSequence
+ typeOfSequence : ExternalControlledVocabularyEntry
+ constructSequence : SpecificSequence
+ type : CollectionType
realizes
DDI_ControlConstruct
DDI_Collection

7.2.38 Graph



7.3 DiscoveryView

Purpose: The Discovery View provides basic information on Annotation, Coverage, and Access. The current view is limited to this basic set of objects and is provided to test out these structures. The Discovery View will modify as more content is available. The future intent is to make this view closer to DISCO as development continues. Currently allows the creation of catalogs of collection items.

Use Cases: Archive collection of bibliographic records and related information for items in the collection.

Target Audience: Distributors

Included Classes: Access, BoundingBox, CatalogOfItems, CollectionItem, Coverage, FundingInformation, SeriesStatement, SpatialCoverage, TemporalCoverage, TopicalCoverage

General Documentation: Point of entry is CatalogOfItems.

A functional view is a collection of classes in DDI that covers a functional use case. These are not namespaces.

7.3.1 Access

Describes access to the annotated object. This item includes a confidentiality statement, descriptions of the access permissions required, restrictions to access, citation requirements, depositor requirements, conditions for access, a disclaimer, any time limits for access restrictions, and contact information regarding access.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
accessConditions	StructuredString	0..1
accessPermission	Form	0..n
applyAccessTo	InternationalIdentifier	0..n
citationRequirement	StructuredString	0..1
confidentialityStatement	StructuredString	0..1
contactAgent	AgentAssociation	0..n
depositRequirement	StructuredString	0..1
disclaimer	StructuredString	0..1
purpose	StructuredString	0..1
restrictions	StructuredString	0..1
validDates	DateRange	0..1

accessConditions

A statement regarding conditions for access. May be expressed in multiple languages and supports the use of structured content.

accessPermission

A link to a form used to provide access to the data or metadata including a statement of the purpose of the form.

applyAccessTo

Identification for an object covered by the access description. This may be any annotated object (collection, publication, identifiable object).

citationRequirement

A statement regarding the citation requirement. May be expressed in multiple languages and supports the use of structured content.

confidentialityStatement

A statement regarding the confidentiality of the related data or metadata.

contactAgent

The agent to contact regarding access including the role of the agent.

depositRequirement

A statement regarding depositor requirements. May be expressed in multiple languages and supports the use of structured content.

disclaimer

A disclaimer regarding the liability of the data producers or providers. May be expressed in multiple languages and supports the use of structured content.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text

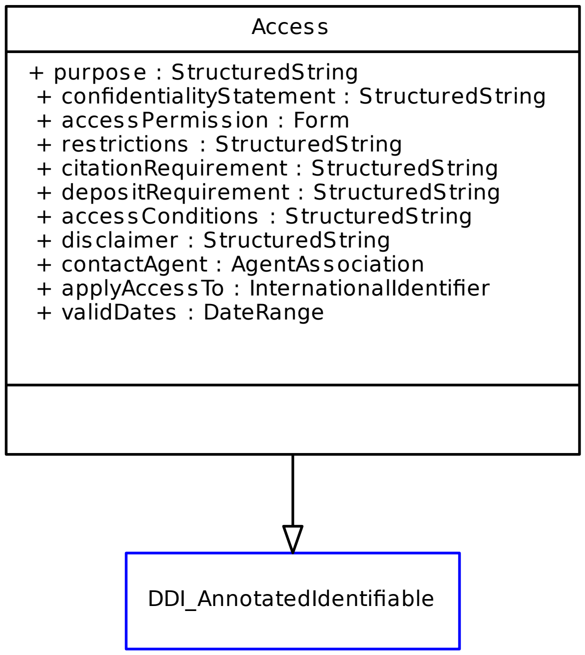
restrictions

A statement regarding restrictions to access. May be expressed in multiple languages and supports the use of structured content.

validDates

The date range or start date of the access description.

Graph


Access
+ purpose : StructuredString
+ confidentialityStatement : StructuredString
+ accessPermission : Form
+ restrictions : StructuredString
+ citationRequirement : StructuredString
+ depositRequirement : StructuredString
+ accessConditions : StructuredString
+ disclaimer : StructuredString
+ contactAgent : AgentAssociation
+ applyAccessTo : InternationalIdentifier
+ validDates : DateRange
DDI_AnnotatedIdentifiable

7.3.2 BoundingBox

A type of Spatial coverage describing a rectangular area within which the actual range of location fits. A BoundingBox can be described by 4 numbers, or two x,y coordinates - the maxima of the north, south, east, and west coordinates found in the area.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
eastLongitude	Real	1..1
northLatitude	Real	1..1
southLatitude	Real	1..1
westLongitude	Real	1..1

eastLongitude

The easternmost coordinate expressed as a decimal between the values of -180 and 180 degrees

northLatitude

The northernmost coordinate expressed as a decimal between the values of -90 and 90 degrees.

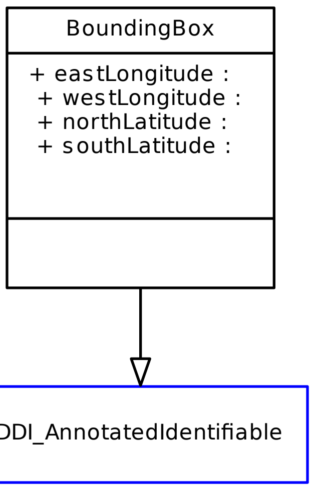
southLatitude

The southermost latitude expressed as a decimal between the values of -90 and 90 degrees

westLongitude

The westernmost coordinate expressed as a decimal between the values of -180 and 180 degrees

Graph



7.3.3 CatalogOfItems

Means of creating a catalog of items in a collection or sub-sets of a larger catalog

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
contains	CollectionItem	0..n	no
hasSubCollection	CatalogOfItems	0..n	no
realizes	Collection	0..n	yes

contains

Constrains member types to CollectionItem

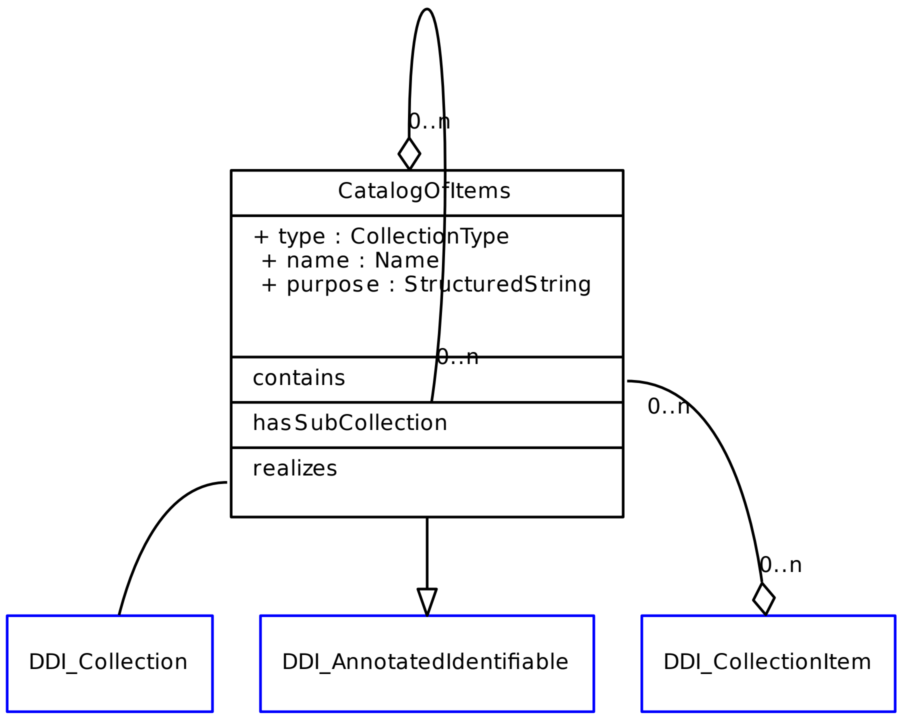
hasSubCollection

May contain sub-collections of catalogs of items

realizes

Realizes the pattern collection

Graph



7.3.4 CollectionItem

A collection item is designed to provide a bibliographic, coverage and related description of an item in a collection.

Extends

AnnotatedIdentifiable

Relationships

Name	Type	Cardinality	allways external
belongsToSeries	SeriesStatement	0..n	no
describesAccess	Access	0..n	no
hasCoverage	Coverage	0..n	no
hasFunding	FundingInformation	0..n	no
realizes	Member	0..n	yes

belongsToSeries

Collection item belongs to this series

describesAccess

Information on how to access the collection item

hasCoverage

Describes the temporal, topical, and spatial coverof the colleciton item

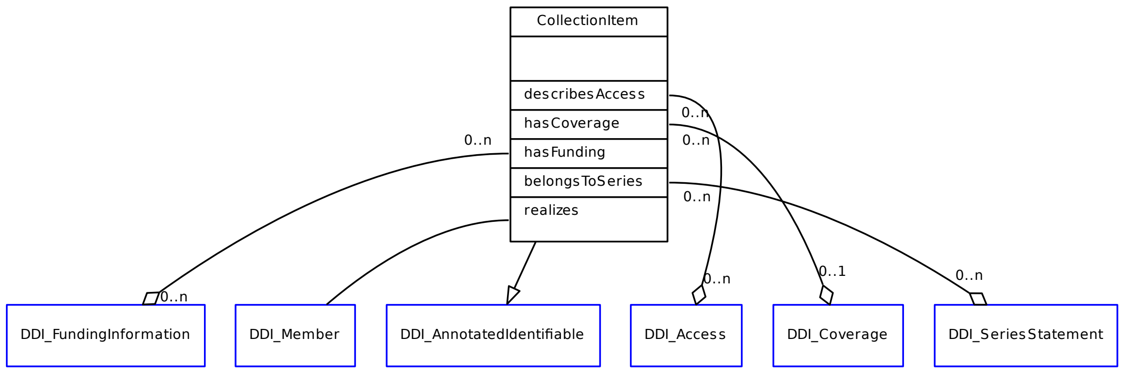
hasFunding

Information concerning the funding source for the colleciton item

realizes

Collection Item can act as the member of a collection

Graph



7.3.5 Coverage

Coverage information for an annotated object. Includes coverage information for temporal, topical, and spatial coverage.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
overview	StructuredString	0..1

overview

A generic description including temporal, topical, and spatial coverage that is the equivalent of dc:coverage (the refinement base of dcterms:spatial and dcterms:temporal. Use specific coverage content for detailed information. Short natural language account of the information obtained from the combination of properties and relationships associated with an object. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality	allways external
hasSpatialCoverage	SpatialCoverage	0..n	no
hasTemporalCoverage	TemporalCoverage	0..n	no
hasTopicalCoverage	TopicalCoverage	0..n	no

hasSpatialCoverage

Description of the spatial (geographic) coverage of the contents of the annotated object.

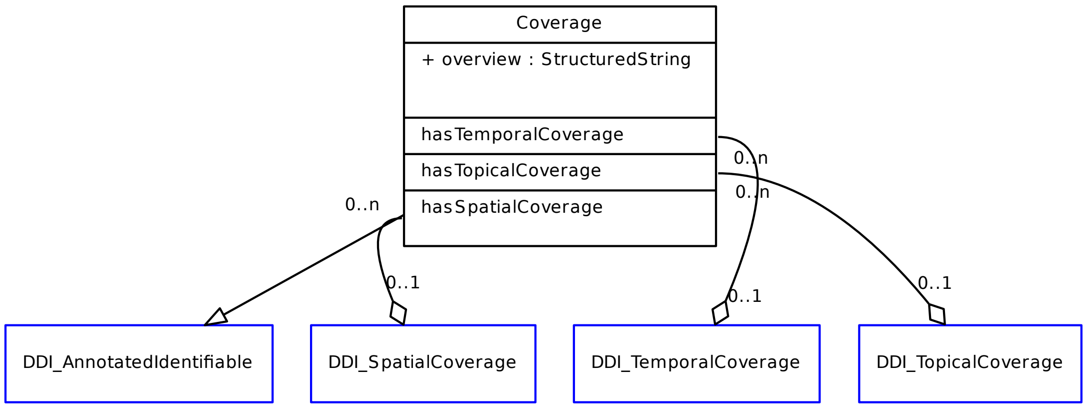
hasTemporalCoverage

The dates and time periods described by the contents of the annotated object.

hasTopicalCoverage

The topics covered by the contents of the annotated object. These may be expressed by subject classification systems and structured or unstructured keywords.

Graph



7.3.6 FundingInformation

Provides information about the individual, agency and/or grant(s) which funded the described entity. Lists a reference to the agency or individual as described by a DDI Agent, the role of the funder, the grant number(s) and a description of the funding activity.

Extends

Identifiable

Properties

Name	Type	Cardinality
funderRole	ExternalControlledVocabularyEntry	0..1
grantNumber	String	0..n
purpose	StructuredString	0..1

funderRole

Role of the funding organization or individual. Supports the use of a controlled vocabulary.

grantNumber

The identification code of the grant or other monetary award which provided funding for the described object.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured tex
‘

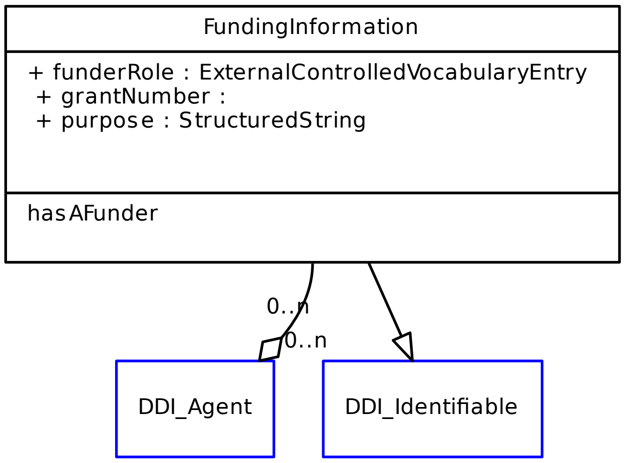
Relationships

Name	Type	Cardinality	allways external
hasAFunder	Agent	0..n	yes

hasAFunder

Funding has been provided by

Graph


FundingInformation
+ funderRole : ExternalControlledVocabularyEntry
+ grantNumber :
+ purpose : StructuredString
hasAFunder
0..n
0..n
DDI_Agent
DDI_Identifiable

7.3.7 SeriesStatement

Describes a series by providing citation information and coverage information.

Extends

Identifiable

Properties

Name	Type	Cardinality
citationOfSeries	Annotation	0..n

citationOfSeries

Bibliographic citation of the Series

‘

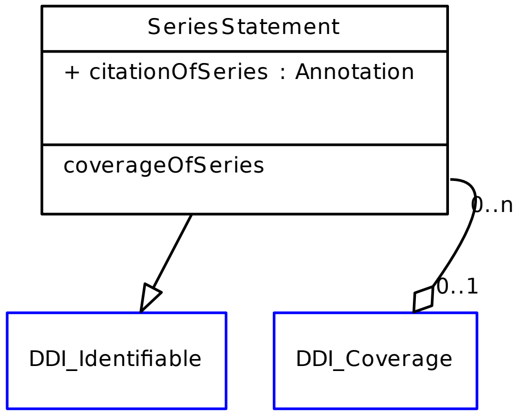
Relationships

Name	Type	Cardinality	allways external
coverageOfSeries	Coverage	0..n	no

coverageOfSeries

Temporal, topical, and spatial coverage of the series

Graph



7.3.8 SpatialCoverage

A description of spatial coverage (geographic coverage) of the annotated object. Spatial coverage is described using a number of objects that support searching by a wide range of systems (geospatial coordinates, geographic classification systems, and general systems using dcterms:spatial.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
description	StructuredString	0..1
spatialAreaCode	ExternalControlledVocabularyEntry	0..n
spatialObject	SpatialObject	0..1

description

A textual description of the spatial coverage to support general searches.

spatialAreaCode

Supports the use of a standardized code such as ISO 3166-1, the Getty Thesaurus of Geographic Names, FIPS-5, etc.

spatialObject

Indicates the most discrete spatial object type identified for a single case. Note that data can be collected at a geographic point (address) and reported as such in a protected file, and then aggregated to a polygon for a public file.

‘

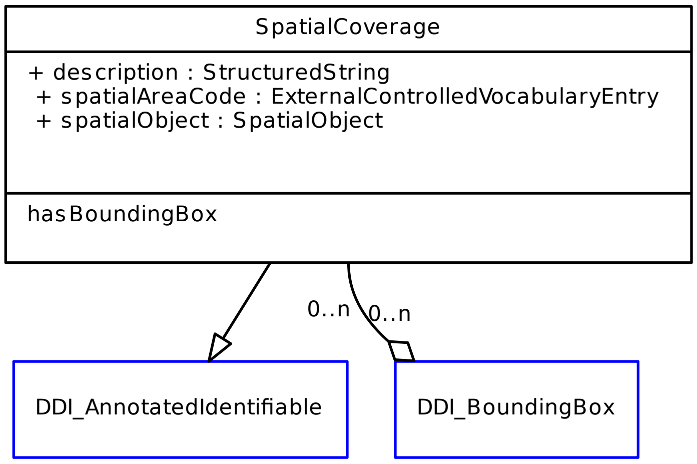
Relationships

Name	Type	Cardinality	allways external
hasBoundingBox	BoundingBox	0..n	no

hasBoundingBox

The north and south latitudes and east and west longitudes that define the spatial coverage area.

Graph


SpatialCoverage
+ description : StructuredString
+ spatialAreaCode : ExternalControlledVocabularyEntry
+ spatialObject : SpatialObject
hasBoundingBox
0..n
0..n
DDI_AnnotatedIdentifiable
DDI_BoundingBox

7.3.9 TemporalCoverage

Describes the date or time period covered by the annotated object. Allows for the use of a specifying the type of coverage date as well as associated subjects or keywords.

Extends

AnnotatedIdentifiable

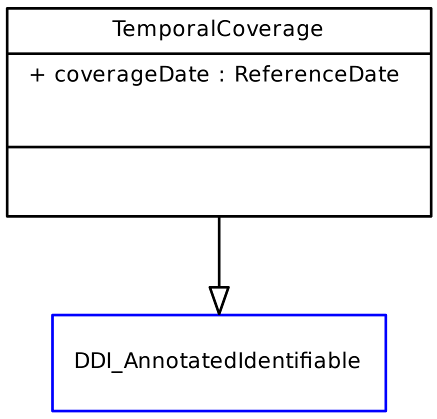
Properties

Name	Type	Cardinality
coverageDate	ReferenceDate	0..n

coverageDate

A date referencing a specific aspect of temporal coverage. The date may be typed to reflect coverage date, collection date, referent date, etc. Subject and Keywords may be associated with the date to specify a specific set of topical information (i.e. Residence associated with a date 5 years prior to the collection date).

Graph



7.3.10 TopicalCoverage

Describes the topical coverage of the module using Subject and Keyword. Note that upper level modules should include all the members of lower level modules. Subjects are members of structured classification systems such as formal subject headings in libraries. Keywords may be structured (e.g. TheSoz thesauri) or unstructured and reflect the terminology found in the document and other related (broader or similar) terms.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
keyword	ExternalControlledVocabularyEntry	0..n
subject	ExternalControlledVocabularyEntry	0..n

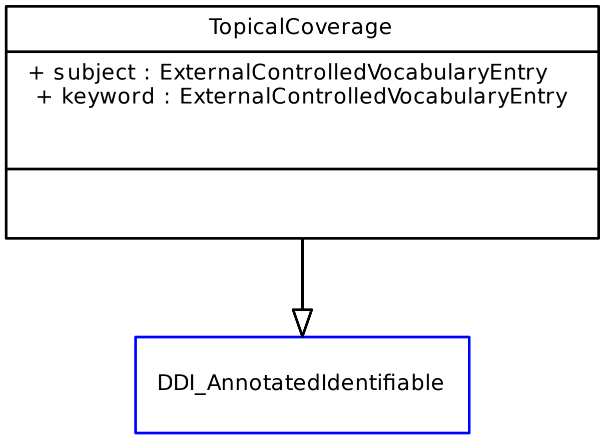
keyword

A keyword that describes the topical coverage of the content of the annotated object. Keywords may be structured (e.g. TheSoz thesauri) or unstructured and reflect the terminology found in the document and other related (broader or similar) terms. Uses and InternationalCodeValue and may indicate the language of the code used.

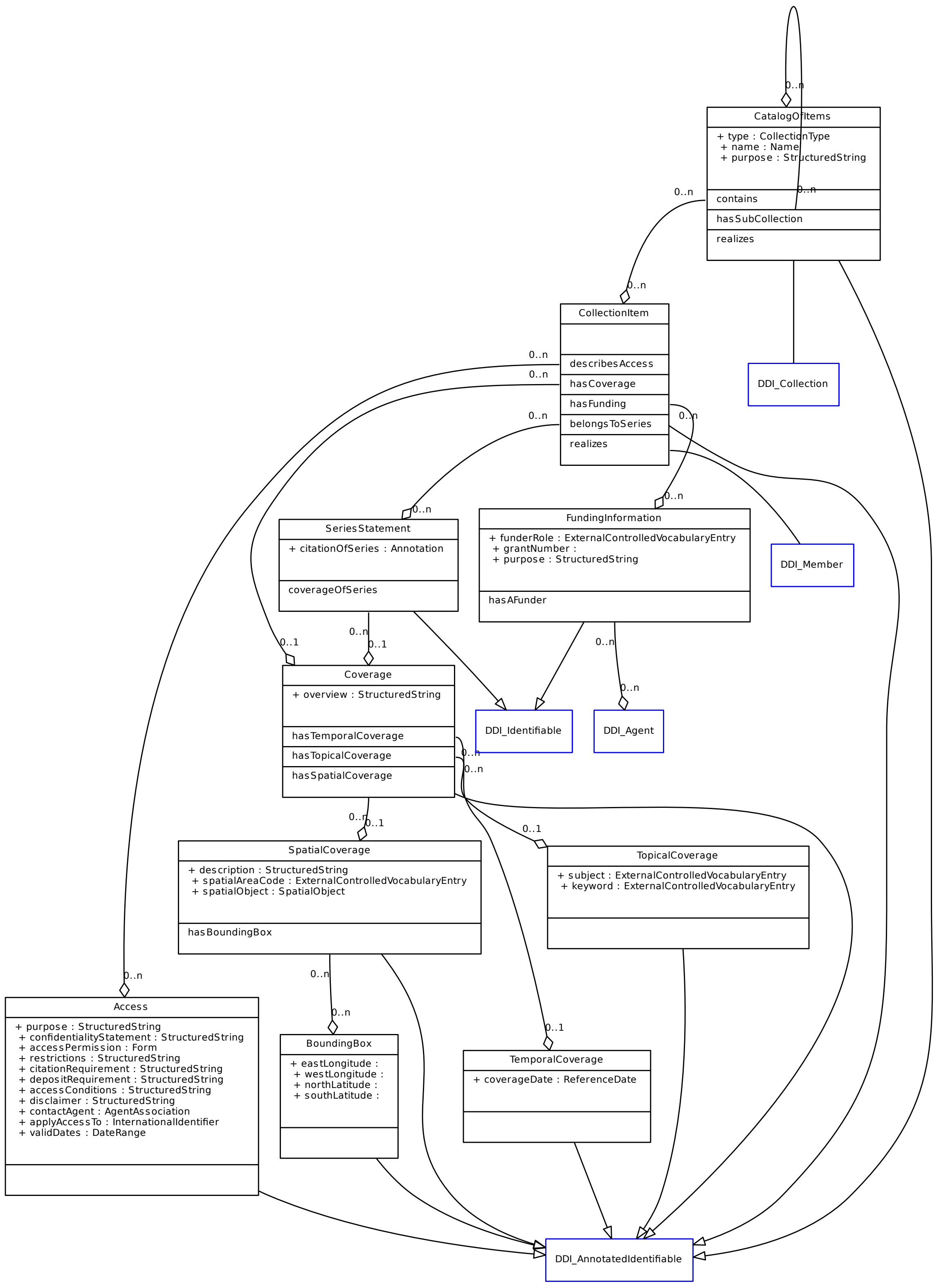
subject

A subject that describes the topical coverage of the content of the annotated object. Subjects are members of structured classification systems such as formal subject headings in libraries. Uses and InternationalCodeValue and may indicate the language of the code used.

Graph


TopicalCoverage
+ subject : ExternalControlledVocabularyEntry
+ keyword : ExternalControlledVocabularyEntry
DDI_AnnotatedIdentifiable

7.3.11 Graph


CatalogOfItems
+ type : CollectionType
+ name : Name
+ purpose : StructuredString
contains
hasSubCollection
realizes
0..n
0..n
0..n
0..n
DDI_Collection
CollectionItem
describesAccess
hasCoverage
hasFunding
belongsToSeries
realizes
0..n
0..n
0..n
0..n
DDI_Member
SeriesStatement
+ citationOfSeries : Annotation
coverageOfSeries
0..n
FundingInformation
+ funderRole : ExternalControlledVocabularyEntry
+ grantNumber :
+ purpose : StructuredString
hasAFunder
0..n
0..n
0..n
DDI_Identifiable
DDI_Agent
Coverage
+ overview : StructuredString
hasTemporalCoverage
hasTopicalCoverage
hasSpatialCoverage
0..1
0..n
0..1
0..n
0..n
SpatialCoverage
+ description : StructuredString
+ spatialAreaCode : ExternalControlledVocabularyEntry
+ spatialObject : SpatialObject
hasBoundingBox
0..n
0..1
TopicalCoverage
+ subject : ExternalControlledVocabularyEntry
+ keyword : ExternalControlledVocabularyEntry
0..1
Access
+ purpose : StructuredString
+ confidentialityStatement : StructuredString
+ accessPermission : Form
+ restrictions : StructuredString
+ citationRequirement : StructuredString
+ depositRequirement : StructuredString
+ accessConditions : StructuredString
+ disclaimer : StructuredString
+ contactAgent : AgentAssociation
+ applyAccessTo : InternationalIdentifier
+ validDates : DateRange
0..n
BoundingBox
+ eastLongitude :
+ westLongitude :
+ northLatitude :
+ southLatitude :
0..n
0..n
TemporalCoverage
+ coverageDate : ReferenceDate
0..1
DDI_AnnotatedIdentifiable

7.4 DataDictionaryView

Purpose: The Data Dictionary View provides the limited information on a data file structure including: the name of the data file, name of variable, physical location (order) of variable, type of variable (character, numeric, etc.), variable label, variable universe/population, substantive (valid) and sentinel (invalid) values with labels if appropriate, and variable concept.

Use Cases: This is a core section of most codebooks as well as often being supplied as a structured document as import to various statistical package set-up files.

Target Audience: Creators of data files for access and distributors.

Included Classes: Access, AttributeRole, Category, CategorySet, Code, CodeItem, CodeList, Concept, Coverage DataPoint, DataRecord, DataStore, Datum, FundingInformation, IdentifierRole, InstanceVariable, Level, LogicalRecordLayout, MeasureRole, PhysicalLayoutOrder, PhysicalLayoutOrderPairs, Population, RectangularLayout (physicalLayout), SegmentByText, SentinalConceptualDomain, SentinelValueDomain, StructureDescription, SpatialCoverage, SubstantiveConceptualDomain, SubstantiveValueDomain, TemporalCoverage, TopicalCoverage, Unit, UnitType, Universe, ValueAndConceptDescription, ValueMapping, Viewpoint

Restricted Classes: The following classes contain relationships NOT found in this view. (If the instance of the related class is available in another functional view or other DDI store it may be identified with an external reference): BoundingBox from SpatialCoverage RepresentedVariable from InstanceVariable ConceptualVariable from InstanceVariable

General Documentation: InstanceVariable contains a relationship to a RepresentedVariable and ConceptualVariable. These two classes have not been included in the Data Dictionary View. The specific content of these two classes are found in the InstanceVariable. Reference may be made instances of these two classes held externally to the Data Dictionary View instance. The entry point for this view is generally the StructureDescription which identifies the DataStore (leading to logical and physical descriptions) and the specific layout information (CSV, Rectangular, etc.).

A functional view is a collection of classes in DDI that covers a functional use case. These are not namespaces.

7.4.1 Access

Describes access to the annotated object. This item includes a confidentiality statement, descriptions of the access permissions required, restrictions to access, citation requirements, depositor requirements, conditions for access, a disclaimer, any time limits for access restrictions, and contact information regarding access.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
accessConditions	StructuredString	0..1
accessPermission	Form	0..n
applyAccessTo	InternationalIdentifier	0..n
citationRequirement	StructuredString	0..1
confidentialityStatement	StructuredString	0..1
contactAgent	AgentAssociation	0..n
depositRequirement	StructuredString	0..1
disclaimer	StructuredString	0..1
purpose	StructuredString	0..1
restrictions	StructuredString	0..1
validDates	DateRange	0..1

accessConditions

A statement regarding conditions for access. May be expressed in multiple languages and supports the use of structured content.

accessPermission

A link to a form used to provide access to the data or metadata including a statement of the purpose of the form.

applyAccessTo

Identification for an object covered by the access description. This may be any annotated object (collection, publication, identifiable object).

citationRequirement

A statement regarding the citation requirement. May be expressed in multiple languages and supports the use of structured content.

confidentialityStatement

A statement regarding the confidentiality of the related data or metadata.

contactAgent

The agent to contact regarding access including the role of the agent.

depositRequirement

A statement regarding depositor requirements. May be expressed in multiple languages and supports the use of structured content.

disclaimer

A disclaimer regarding the liability of the data producers or providers. May be expressed in multiple languages and supports the use of structured content.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text

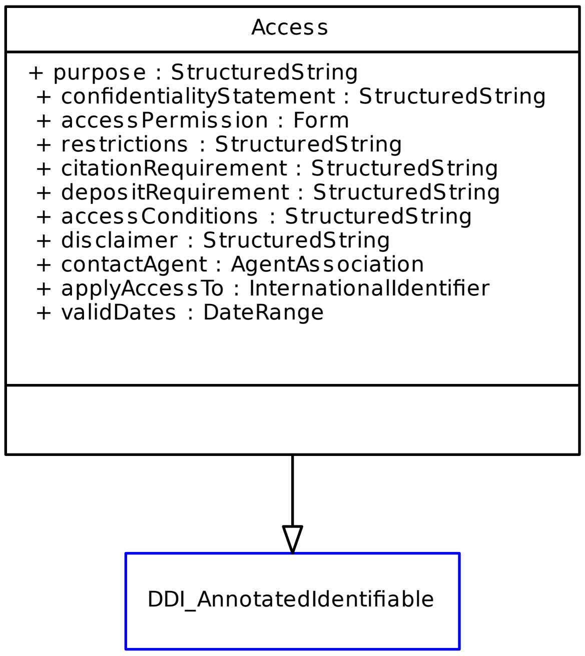
restrictions

A statement regarding restrictions to access. May be expressed in multiple languages and supports the use of structured content.

validDates

The date range or start date of the access description.

Graph


Access
+ purpose : StructuredString
+ confidentialityStatement : StructuredString
+ accessPermission : Form
+ restrictions : StructuredString
+ citationRequirement : StructuredString
+ depositRequirement : StructuredString
+ accessConditions : StructuredString
+ disclaimer : StructuredString
+ contactAgent : AgentAssociation
+ applyAccessTo : InternationalIdentifier
+ validDates : DateRange
DDI_AnnotatedIdentifiable

7.4.2 Agent

An actor that performs a role in relation to a process.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
agentId	AgentId	0..n
image	PrivateImage	0..n
purpose	StructuredString	0..1

agentId

An identifier within a specified system for specifying an agent

image

References an image using the standard Image description. In addition to the standard attributes provides an effective date (period), the type of image, and a privacy ranking.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

‘

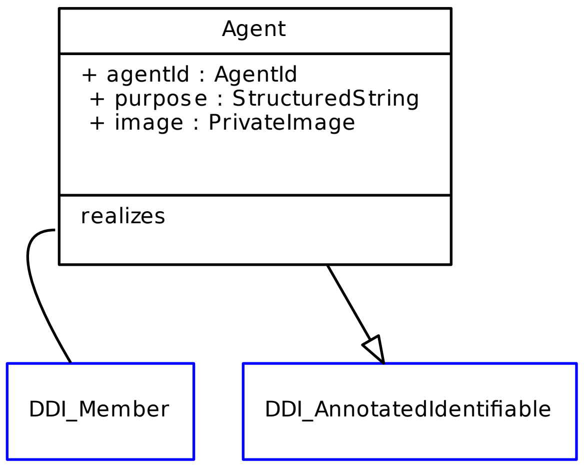
Relationships

Name	Type	Cardinality	allways external
realizes	Member	0..n	yes

realizes

An Agent can perform the role of a Member in an Agent Registry

Graph



7.4.3 AttributeRole

A MeasureRole identifies an InstanceVariable as being an attribute within a ViewPoint.

Extends

ViewpointRole

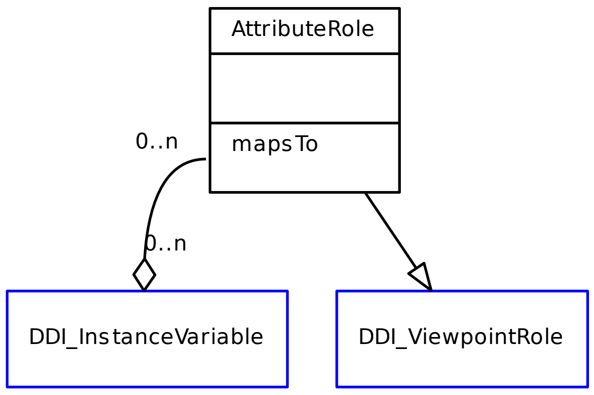
Relationships

Name	Type	Cardinality	allways external
mapsTo	InstanceVariable	0..n	no

mapsTo

The InstanceVariables being assigned the role of attribute

Graph



7.4.4 Category

A Concept whose role is to define and measure a characteristic.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
name	Name	0..n

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

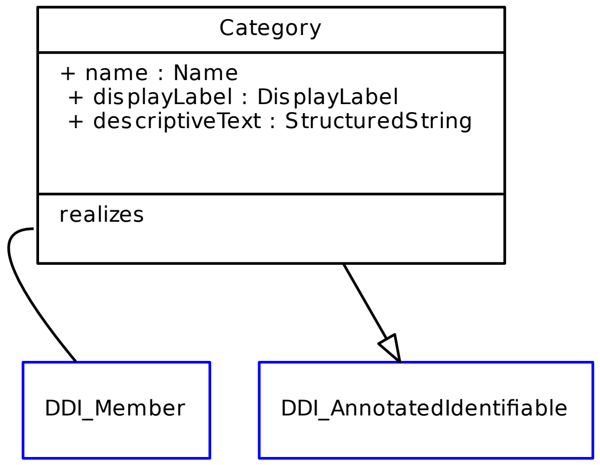
Relationships

Name	Type	Cardinality	allways external
realizes	Member	0..n	yes

realizes

Class can play the role of Member within a Collection

Graph


Category
+ name : Name
+ displayLabel : DisplayLabel
+ descriptiveText : StructuredString
realizes
DDI_Member
DDI_AnnotatedIdentifiable

7.4.5 CategorySet

A Category Set is a type of Node Set which groups Categories.

Extends

NodeSet

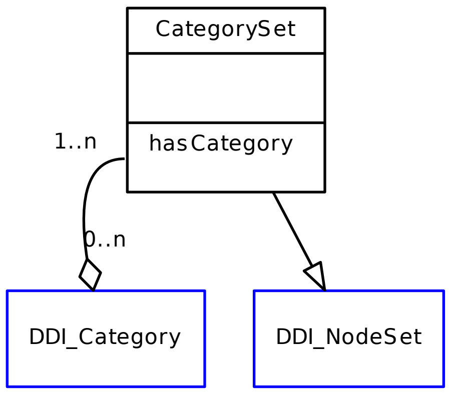
Relationships

Name	Type	Cardinality	allways external
hasCategory	Category	1..n	no

hasCategory

Specialization of contains in NodeSet for Categories.

Graph



7.4.6 Code

A Designation for a Category.

Extends

Designation

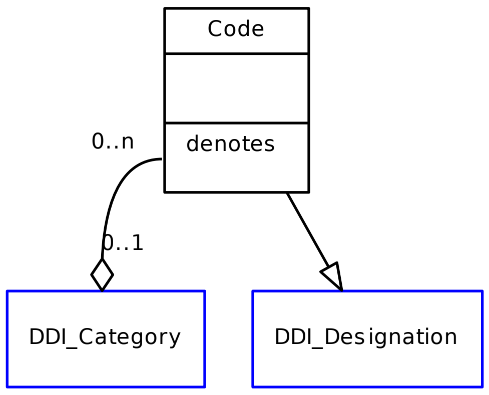
Relationships

Name	Type	Cardinality	allways external
denotes	Category	0..n	no

denotes

A definition for the code. Specialization of denotes for Categories.

Graph



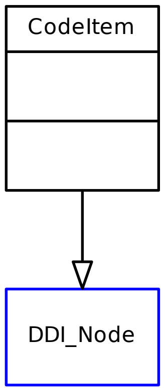
7.4.7 CodeItem

A type of Node contained in a CodeList that has a Code associated to a Category. In addition, a CodeItem can have a number of optional Designations. All Designations associated to the Code Item, including the Code, are synonyms, i.e. are associated with the same Concept.

Extends

:ref:‘Node

Graph



7.4.8 CodeList

A list of Codes and associated Categories. May be flat or hierarchical.

Extends

NodeSet

Relationships

Name	Type	Cardinality	allways external
contains	CodeItem	1..n	no
references	CategorySet	1..n	no
represents	ValueDomain	1..n	no

contains

Specialization of contains in NodeSet for CodeItems.

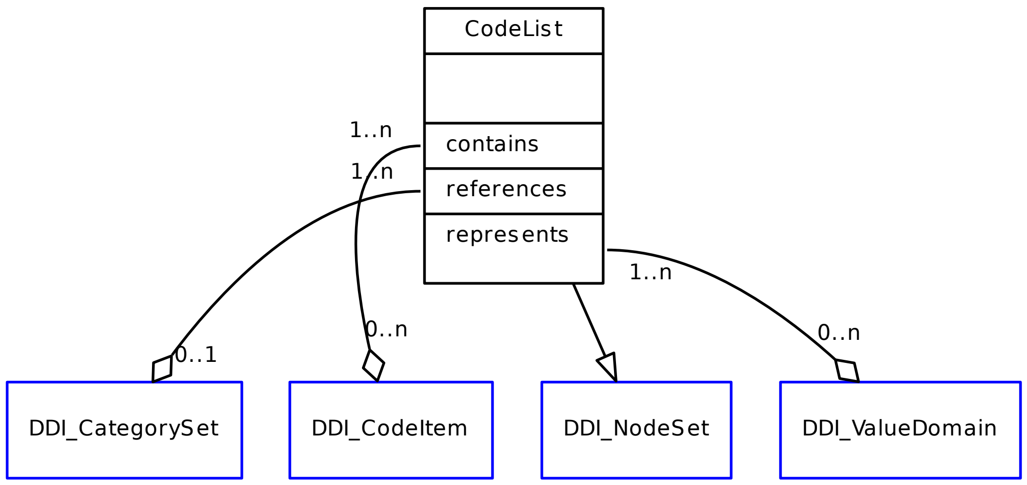
references

CategorySet associated with the CodeList.

represents

Enumerated Value Domain represented by the CodeList.

Graph



7.4.9 Concept

Unit of thought differentiated by characteristics [GSIM 1.1]

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
definition	StructuredString	0..1
name	Name	0..n

definition

Natural language statement conveying the meaning of a concept, differentiating it from other concepts. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

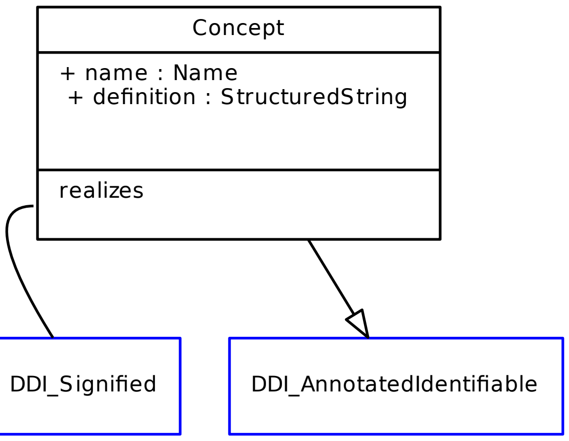
Relationships

Name	Type	Cardinality	allways external
realizes	Signified	0..n	yes

realizes

A concept can play the role of a signified when there is a designation that denotes it.

Graph



7.4.10 Coverage

Coverage information for an annotated object. Includes coverage information for temporal, topical, and spatial coverage.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
overview	StructuredString	0..1

overview

A generic description including temporal, topical, and spatial coverage that is the equivalent of dc:coverage (the refinement base of dcterms:spatial and dcterms:temporal. Use specific coverage content for detailed information. Short natural language account of the information obtained from the combination of properties and relationships associated with an object. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality	allways external
hasSpatialCoverage	SpatialCoverage	0..n	no
hasTemporalCoverage	TemporalCoverage	0..n	no
hasTopicalCoverage	TopicalCoverage	0..n	no

hasSpatialCoverage

Description of the spatial (geographic) coverage of the contents of the annotated object.

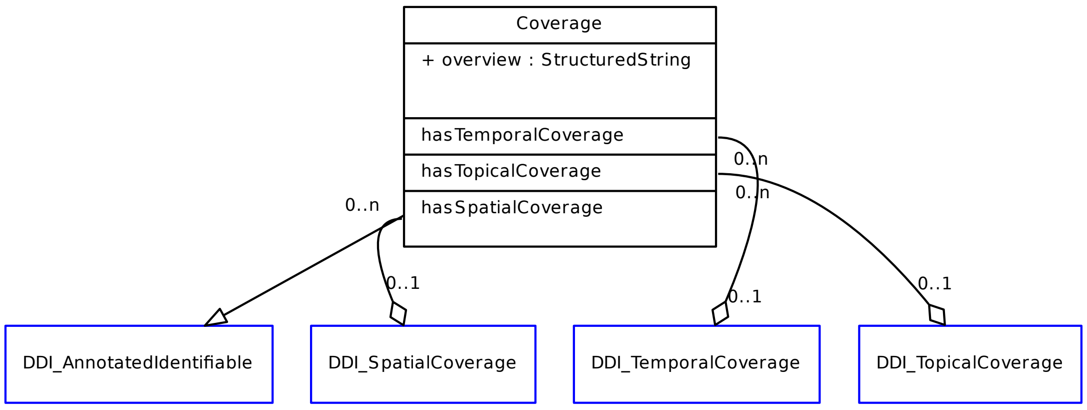
hasTemporalCoverage

The dates and time periods described by the contents of the annotated object.

hasTopicalCoverage

The topics covered by the contents of the annotated object. These may be expressed by subject classification systems and structured or unstructured keywords.

Graph



7.4.11 DataPoint

A DataPoint is a container for a Datum.

Extends

AnnotatedIdentifiable

Relationships

Name	Type	Cardinality	allways external
has	Datum	0..n	no
isDescribedBy	InstanceVariable	0..n	no
realizes	Member	0..n	yes

has

The Datum populating the DataPoint.

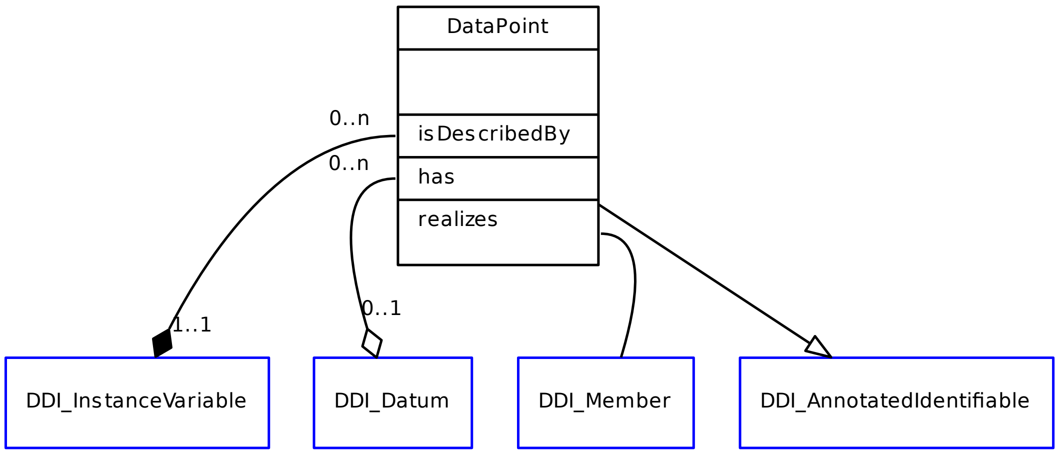
isDescribedBy

The InstanceVariable delimits the values which can populate a DataPoint.

realizes

Classes which realize the class Member fufill the role of Member e.g. they may have relations defined on them, for example one DataPoint follows another in a Record.

Graph



7.4.12 DataRecord

A Record is a Collection of DataPoints with an optional OrderRelation.

[Name of the class needs to be changed to Record].

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	1..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
contains	DataPoint	1..1	no
hasLogicalLayout	LogicalRecordLayout	0..n	no
isViewedFrom	Viewpoint	1..n	no
realizes	Collection	0..n	yes

contains

The set of DataPoints composing the DataRecord

hasLogicalLayout

Structure (type) of the Data Record.

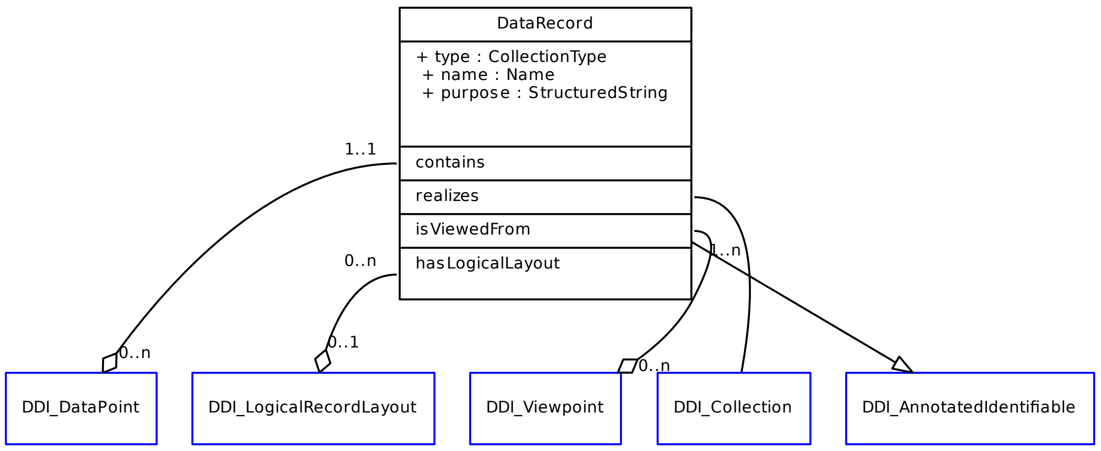
isViewedFrom

A DataRecord may have multiple ViewPoints, each of which assigns among roles identifier, measure, and attribute to the InstanceVariables associated with each DataPoint.

realizes

Allows the DataStucture to function as a Collection, enabling, for example, ordering of its components.

Graph



7.4.13 DataStore

An organized collection of data

Additional information: Using its datapoints and their processing instructions, a datastore can host questionnaire and variable datum wiring them together to form processing pipelines in all shapes and forms. Also, in line with a spreadsheet, a datastore can host multiple data structures. So a datastore can contain unit data as well as a pivot table that represents the unit data dimensionally. Likewise, the same datastore may contain both a normalized recordset in which entities are columns and its denormalized counterpart in which the rows are entities. Given processing instructions, one can losslessly be constructed from the other.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
characterSet	String	0..1
dataStoreType	ExternalControlledVocabularyEntry	0..1
name	Name	0..n
purpose	StructuredString	0..1
recordCount	Integer	0..1
type	CollectionType	0..1

characterSet

Default character set used in the Data Store.

dataStoreType

The type of DataStore. Could be delimited file, fixed record length file, relational database, etc. Points to an external definition which can be part of a controlled vocabulary maintained by the DDI Alliance.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

recordCount

The number of records in the Data Store.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
contains	DataRecord	0..1	no
realizes	Collection	0..n	yes
references	LogicalRecordLayout	0..1	no

contains

Data in the form of Data Records contained in the Data Store.

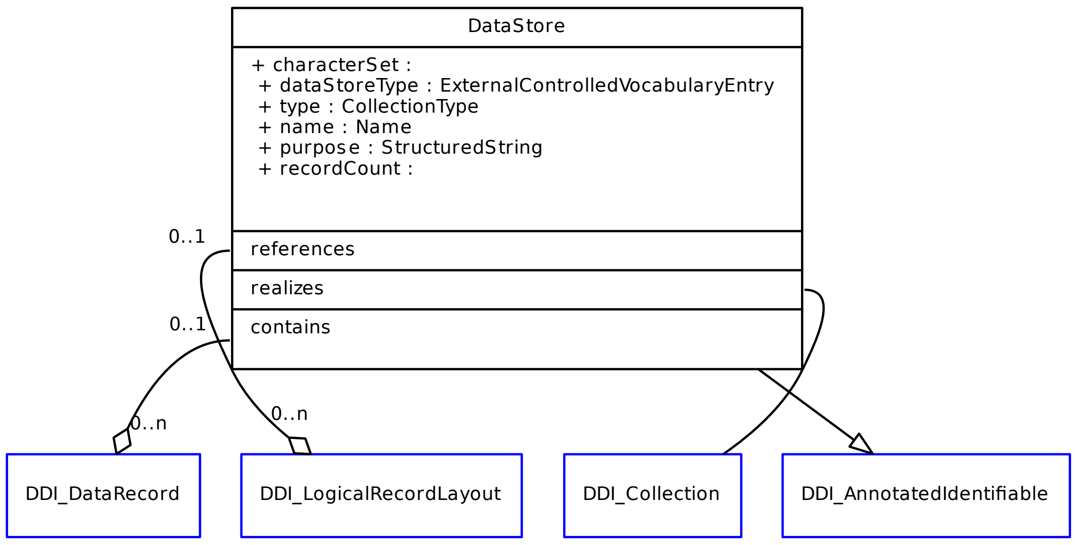
realizes

Allows the DataStore to function as a Collection, enabling, for example, ordering of its components.

references

Can point to one or more descriptions of the structure (type) of the Data Record.

Graph



7.4.14 Datum

A Datum is the designation of a concept with a notion of equality defined.

Extends

Designation

Relationships

Name	Type	Cardinality	allways external
isBoundedBy	InstanceVariable	0..n	no
isConstrainedOf	ValueDomain	0..n	no

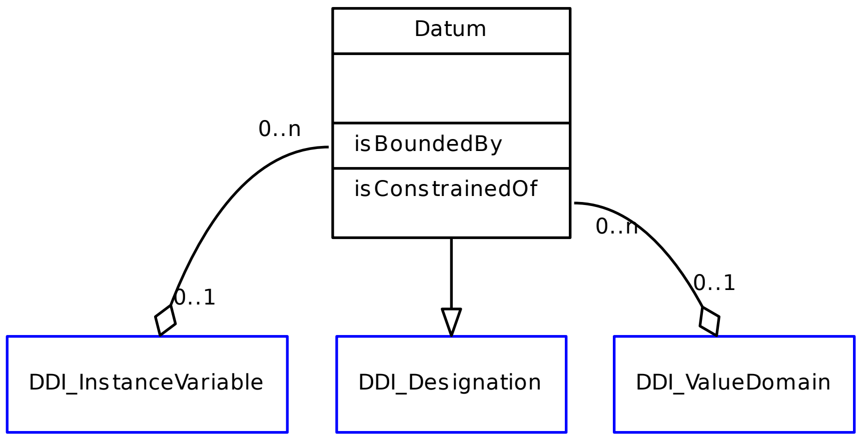
isBoundedBy

A Datum is bounded by an InstanceVariable.

isConstrainedOf

A Datum is drawn from a set of values, either substantive or sentinel

Graph



7.4.15 FundingInformation

Provides information about the individual, agency and/or grant(s) which funded the described entity. Lists a reference to the agency or individual as described by a DDI Agent, the role of the funder, the grant number(s) and a description of the funding activity.

Extends

Identifiable

Properties

Name	Type	Cardinality
funderRole	ExternalControlledVocabularyEntry	0..1
grantNumber	String	0..n
purpose	StructuredString	0..1

funderRole

Role of the funding organization or individual. Supports the use of a controlled vocabulary.

grantNumber

The identification code of the grant or other monetary award which provided funding for the described object.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured tex

‘

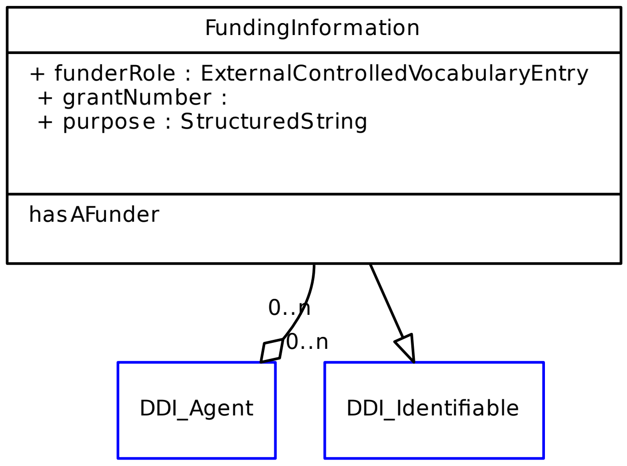
Relationships

Name	Type	Cardinality	allways external
hasAFunder	Agent	0..n	no

hasAFunder

Funding has been provided by

Graph


FundingInformation
+ funderRole : ExternalControlledVocabularyEntry
+ grantNumber :
+ purpose : StructuredString
hasAFunder
0..n
0..n
DDI_Agent
DDI_Identifiable

7.4.16 IdentifierRole

An IdentifierRole identifies an InstanceVariable as being an identifier within a ViewPoint.

Extends

ViewpointRole

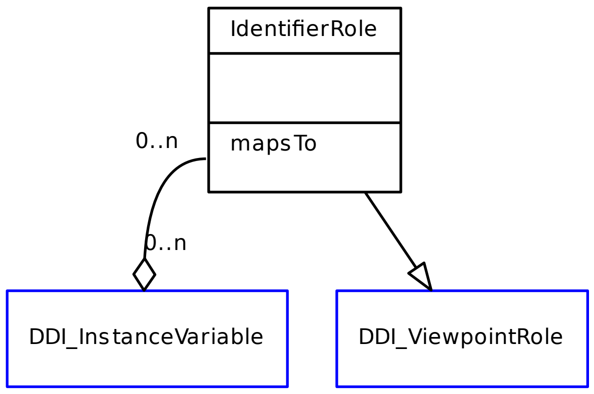
Relationships

Name	Type	Cardinality	allways external
mapsTo	InstanceVariable	0..n	no

mapsTo

The InstanceVariables being assigned the role of identifier

Graph



7.4.17 InstanceVariable

The use of a Represented Variable within a Data Set.

Extends

RepresentedVariable

Properties

Name	Type	Cardinality
physicalDataType	ExternalControlledVocabularyEntry	0..1
variableRole	StructuredString	0..1

physicalDataType

The data type of this variable. Supports the optional use of an external controlled vocabulary.

variableRole

An Instance Variable can take different roles, e.g. Identifier, Measure and Attribute. Note that DataStructure takes care of the ordering of Identifiers.

‘

Relationships

Name	Type	Cardinality	allways external
basedOnConceptualVariable	ConceptualVariable	0..n	no
basedOnRepresentedVariable	RepresentedVariable	0..n	no
measures	Population	0..n	no
takesSentinelValuesFrom	SentinelValueDomain	0..n	no

basedOnConceptualVariable

The ConceptualVariable that can be shared by multiple InstanceVariables. Indicates comparability ConceptualDomain and UnitType. If a relationship to a RepresentedVariable is defined there should be no relationship to a Conceptual-Variable.

basedOnRepresentedVariable

The RepresentedVariable that can be shared by multiple InstanceVariables. Indicates comparability in substantive representation, Universe, and indirectly ConceptualDomain. If a relationship to a RepresentedVariable is defined there should be no relationship to a ConceptualVariable.

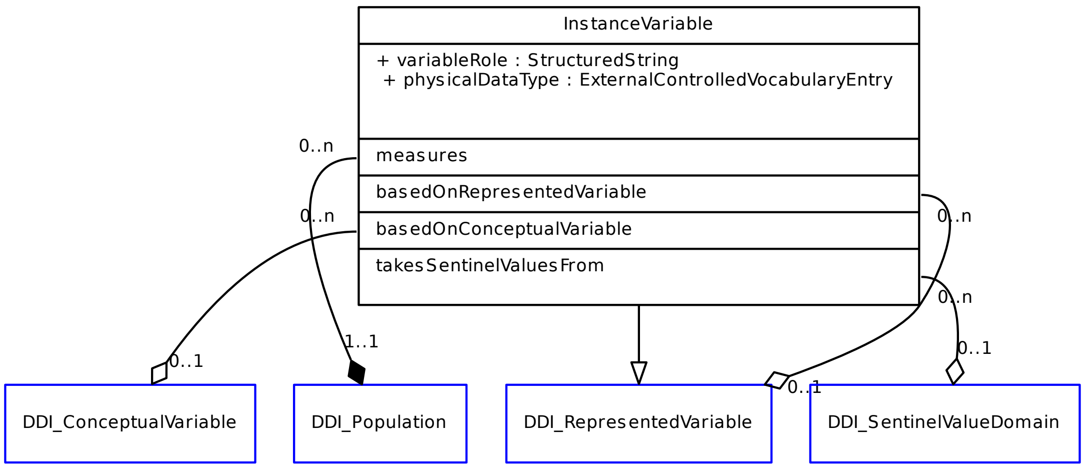
measures

Set of specific units (people, entities, objects, events), usually in a given time and geography, being measured. Can be a specialization of the Universe measured by a related RepresentedVariable.

takesSentinelValuesFrom

The association to the possible sentinel (missing) values.

Graph



7.4.18 Level

The Level describes the nesting structure of a hierarchical collection.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
groups	Node	1..1	no
isDefinedBy	Concept	0..n	no
realizes	Collection	0..n	yes

groups

Realization of contains in Collection for Nodes.

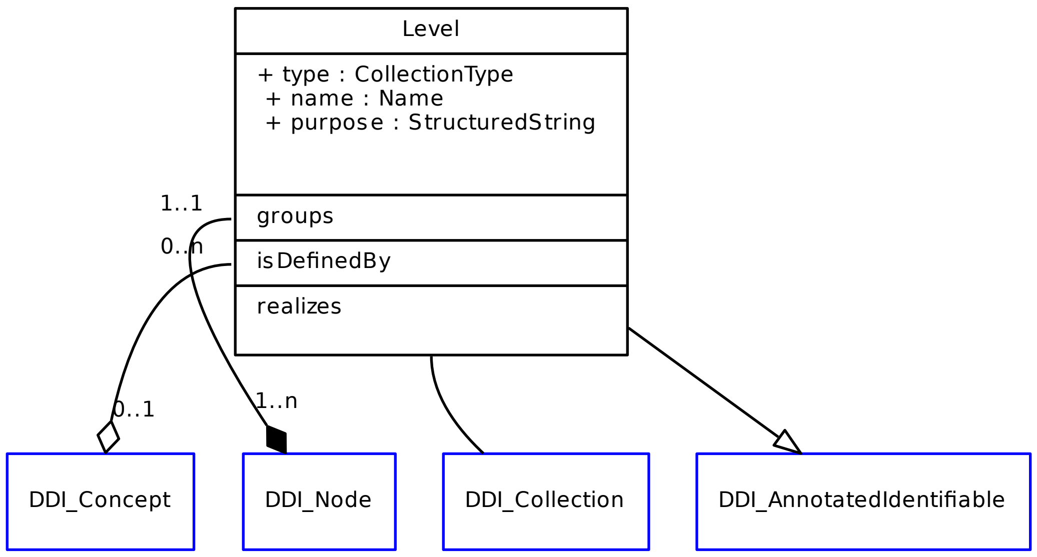
isDefinedBy

Associated concept that provides the conceptual definition of the level.

realizes

Class of the Collection pattern realized by this class.

Graph



7.4.19 LogicalRecordLayout

Collection of Instance Variables that function as a record type describing the structure of individual records.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
contains	InstanceVariable	0..1	no
isViewedFrom	Viewpoint	0..n	no
nests	LogicalRecordLayout	0..n	no
realizes	Collection	0..n	yes

contains

Instance Variable(s) defining this record layout.

isViewedFrom

Viewpoint associated with the record layout

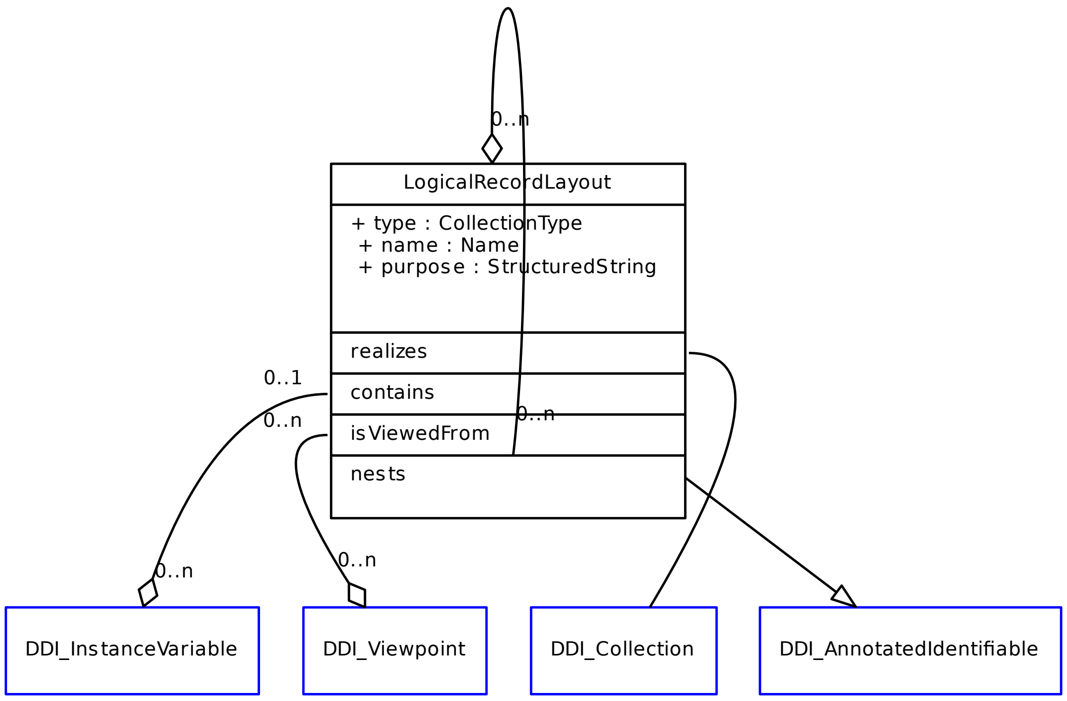
nests

LogicalRecordLayout nested in a LogicalRecordLayout

realizes

Class of the Collection pattern realized by this class.

Graph



7.4.20 MeasureRole

A MeasureRole identifies an InstanceVariable as being a measure within a ViewPoint.

Extends

ViewpointRole

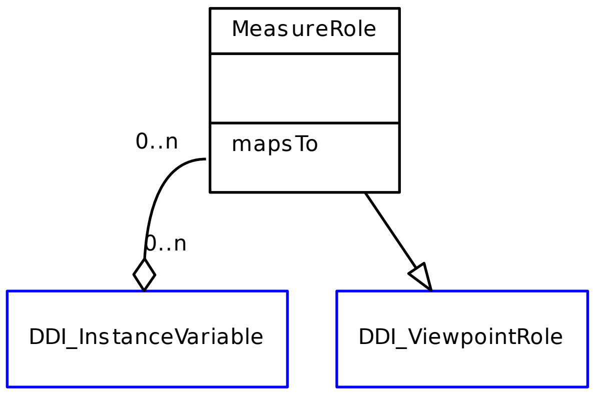
Relationships

Name	Type	Cardinality	allways external
mapsTo	InstanceVariable	0..n	no

mapsTo

The InstanceVariables being assigned the role of measure

Graph



7.4.21 PhysicalLayoutOrder

A realization of OrderRelation that is used to describe the sequence of Value Mappings in a Physical Layout.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
criteria	StructuredString	0..1
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
transitivity	TransitivityType	1..1

criteria

Intensional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)

reflexivity

Fixed to Reflexive

semantics

Controlled vocabulary for the order relation semantics. It should contain, at least, the following: Self_Or_Descendant_Of, Part_Of, Less_Than_Or_Equal_To, Subtype_Of, Subclass_Of.

symmetry

Fixed to Anti_Symmetric

transitivity

Fixed to Transitive

‘

Relationships

Name	Type	Cardinality	allways external
contains	PhysicalLayoutOrderedPair	0..n	no
realizes	OrderRelation	0..n	yes
structures	PhysicalLayout	0..n	no

contains

Physical layout ordered pairs describing the Value Mappings sequence.

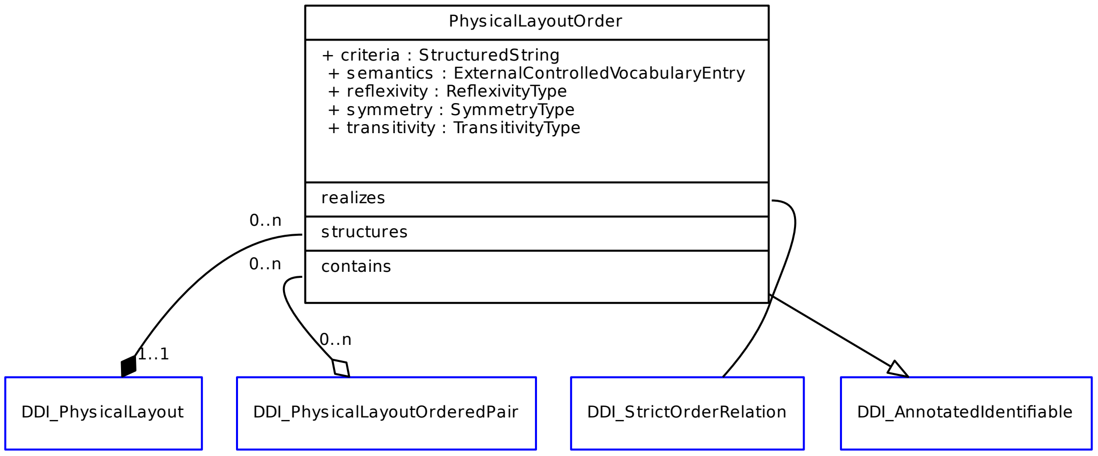
realizes

Class of the Collection pattern realized by this class.

structures

The Value Mappings in a Physical Layout to be sequenced.

Graph


PhysicalLayoutOrder
+ criteria : StructuredString
+ semantics : ExternalControlledVocabularyEntry
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
realizes
structures
contains
0..n
0..n
1..1
0..n
DDI_PhysicalLayout
DDI_PhysicalLayoutOrderedPair
DDI_StrictOrderRelation
DDI_AnnotatedIdentifiable

7.4.22 PhysicalLayoutOrderedPair

The pair of Value Mappings in a Physical Layout which are being placed in a sequence.

Extends

Identifiable

Relationships

Name	Type	Cardinality	allways external
realizes	OrderedPair	0..n	yes
source	ValueMapping	0..n	no
target	ValueMapping	0..n	no

realizes

Class of the Collection pattern realized by this class.

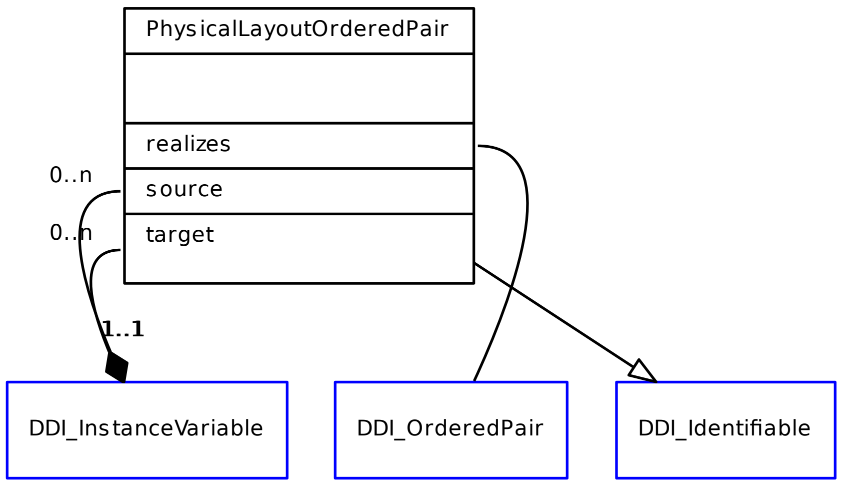
source

Specialization of source in OrderedPair for Value Mappings in a Physical Layout.

target

Specialization of target in OrderedPair for Value Mappings in a Physical Layout.

Graph



7.4.23 Population

Set of specific units (people, entities, objects, events), usually in a given time and geography.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
name	Name	0..n

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

Relationships

Name	Type	Cardinality	allways external
contains	Unit	0..n	no
hasSubpopulation	Population	1..1	no
narrowsUniverse	Universe	0..n	no
realizes	Member	0..n	yes
usesConcept	Concept	0..n	no

contains

Units in the Population

hasSubpopulation

Populations can have sub populations. For example the Sub-Universe Class of Gender for the Universe Population of Canada in 2011 may contain the Universe Canadian Males in 2011 and the Universe Canadian Females in 2011.

narrowsUniverse

Reference to a Universe that the Population narrows

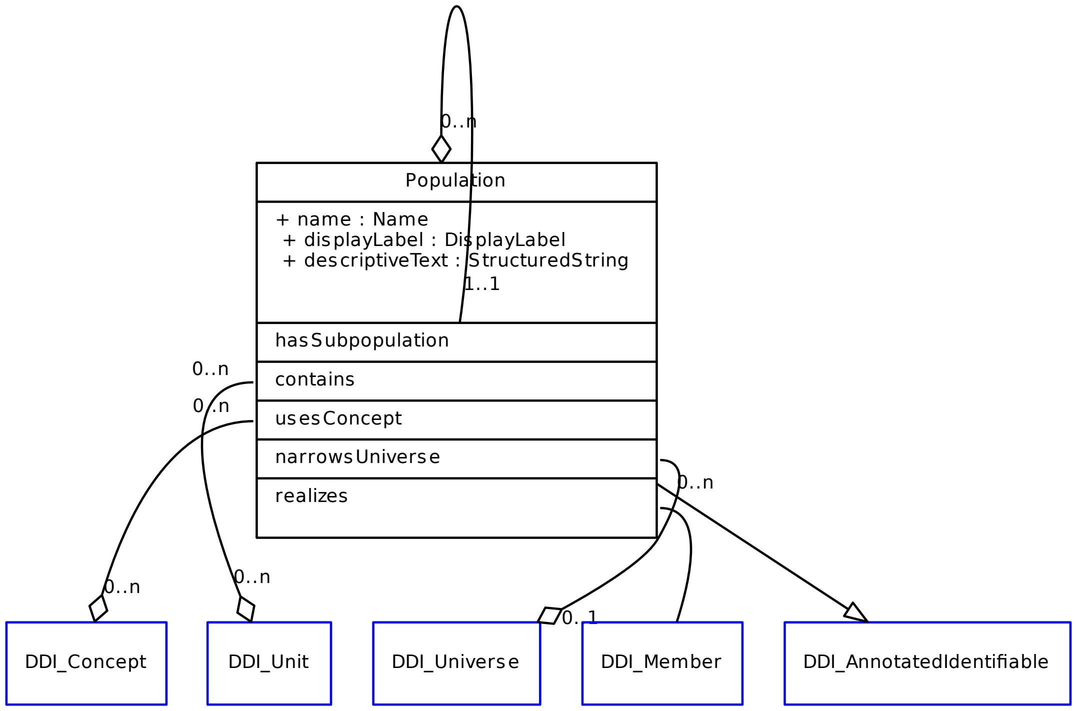
realizes

Class can be used in the role of Member within a Collection

usesConcept

Reference to the Concept that is being used

Graph



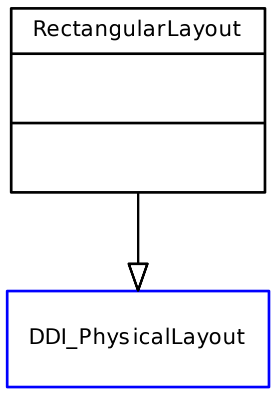
7.4.24 RectangularLayout

RectangularLayout supports the description of unit-record (“wide”) data sets, where each row in the data set provides a group of values for variables all relating to a single unit. The columns will contain data relating to the values for a single variable.

Extends

:ref:‘PhysicalLayout

Graph



7.4.25 SegmentByText

Defines the location of a segment of text.

Extends

PhysicalSegmentLocation

Properties

Name	Type	Cardinality
definedByCharacterOffset	CharacterOffset	0..n
definedByLineParameters	LineParameter	0..n

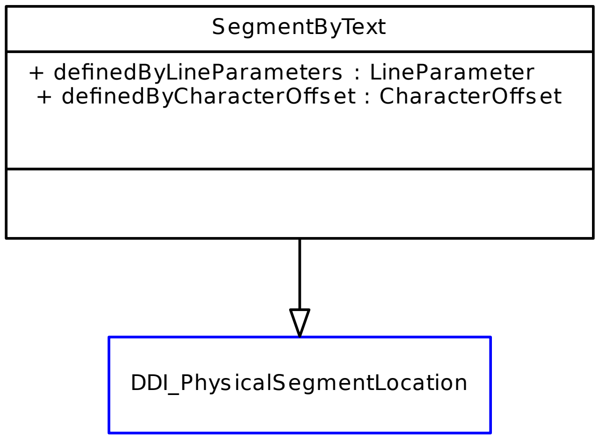
definedByCharacterOffset

Points to parameter pairs using start character (from the beginning of the document) and end character

definedByLineParameters

Points to parameters using start line and character withing line and end line and character within that line

Graph


SegmentByText
+ definedByLineParameters : LineParameter
+ definedByCharacterOffset : CharacterOffset
DDI_PhysicalSegmentLocation

7.4.26 SentinelConceptualDomain

Description or list of possible sentinel concepts , e.g. missing values.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality	allways external
describedConceptualDomain	ValueAndConceptDescription	0..n	no
enumeratedConceptualDomain	CategorySet	0..n	no

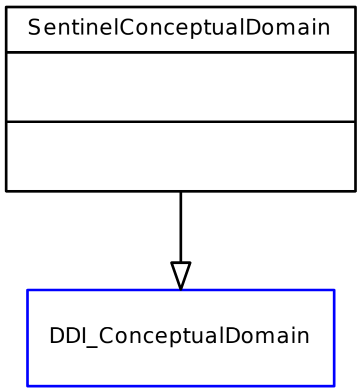
describedConceptualDomain

A description of the concepts in the domain. A numeric domain might use a logical expression to be machine actionable a text domain might use a regular expression to describe strings that describe the concepts.

enumeratedConceptualDomain

The CategorySet containing the concepts in the domain.

Graph



7.4.27 SentinelValueDomain

The Value Domain for a sentinel conceptual domain. Sentinel values are defined in ISO 11404 as "element of a value space that is not completely consistent with a datatype's properties and characterizing operations...". A common example would be codes for missing values.

Extends

ValueDomain

Relationships

Name	Type	Cardinality	allways external
describedValueDomain	ValueAndConceptDescription	0..n	no
enumeratedValueDomain	CodeList	0..n	no
takesConceptsFrom	SentinelConceptualDomain	0..n	no

describedValueDomain

A formal description of the set of valid values - for described value domains.

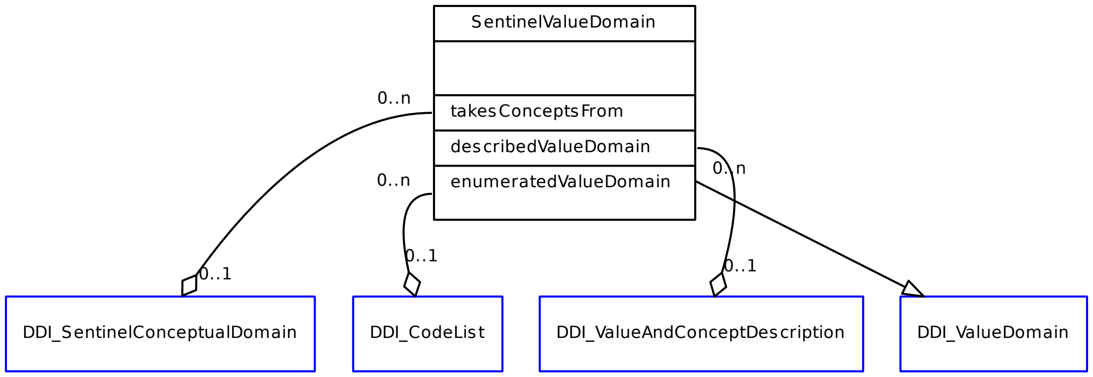
enumeratedValueDomain

A CodeList enumerating the set of valid values.

takesConceptsFrom

Corresponding conceptual definition given by a SentinelConceptualDomain.

Graph



7.4.28 SpatialCoverage

A description of spatial coverage (geographic coverage) of the annotated object. Spatial coverage is described using a number of objects that support searching by a wide range of systems (geospatial coordinates, geographic classification systems, and general systems using dcterms:spatial.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
description	StructuredString	0..1
spatialAreaCode	ExternalControlledVocabularyEntry	0..n
spatialObject	SpatialObject	0..1

description

A textual description of the spatial coverage to support general searches.

spatialAreaCode

Supports the use of a standardized code such as ISO 3166-1, the Getty Thesaurus of Geographic Names, FIPS-5, etc.

spatialObject

Indicates the most discrete spatial object type identified for a single case. Note that data can be collected at a geographic point (address) and reported as such in a protected file, and then aggregated to a polygon for a public file.

‘

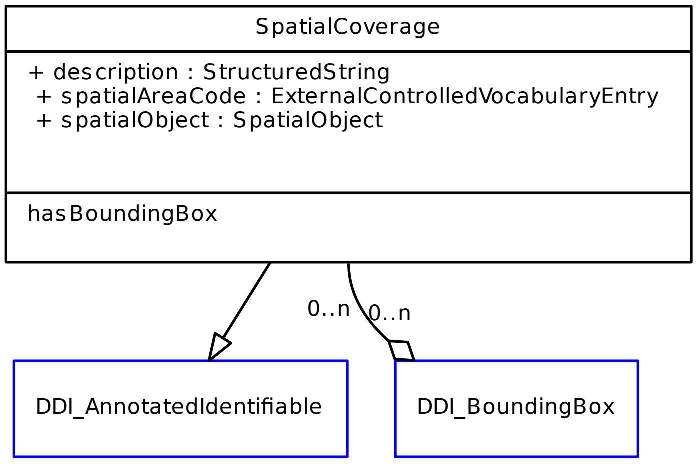
Relationships

Name	Type	Cardinality	allways external
hasBoundingBox	BoundingBox	0..n	yes

hasBoundingBox

The north and south latitudes and east and west longitudes that define the spatial coverage area.

Graph


SpatialCoverage
+ description : StructuredString
+ spatialAreaCode : ExternalControlledVocabularyEntry
+ spatialObject : SpatialObject
hasBoundingBox
0..n
0..n
DDI_AnnotatedIdentifiable
DDI_BoundingBox

7.4.29 SubstantiveConceptualDomain

Set of valid Concepts. The Concepts can be described by either enumeration or by an expression.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n

displayLabel

A display label for the Conceptual Domain. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.

‘

Relationships

Name	Type	Cardinality	allways external
describedConceptualDomain	ValueAndConceptDescription	0..n	no
enumeratedConceptualDomain	CategorySet	0..n	no

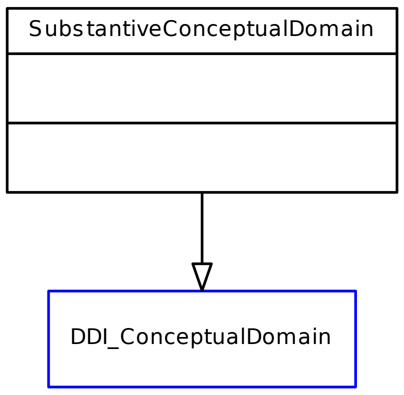
describedConceptualDomain

A description of the concepts in the domain. A numeric domain might use a logical expression to be machine actionable a text domain might use a regular expression to describe strings that describe the concepts.

enumeratedConceptualDomain

The CategorySet containing the concepts in the domain.

Graph



7.4.30 SubstantiveValueDomain

The Value Domain for a substantive conceptual domain.

Extends

ValueDomain

Relationships

Name	Type	Cardinality	allways external
describedValueDomain	ValueAndConceptDescription	0..n	no
enumeratedValueDomain	CodeList	0..n	no
takesConceptsFrom	SubstantiveConceptualDomain	0..n	no

describedValueDomain

A formal description of the set of valid values - for described value domains.

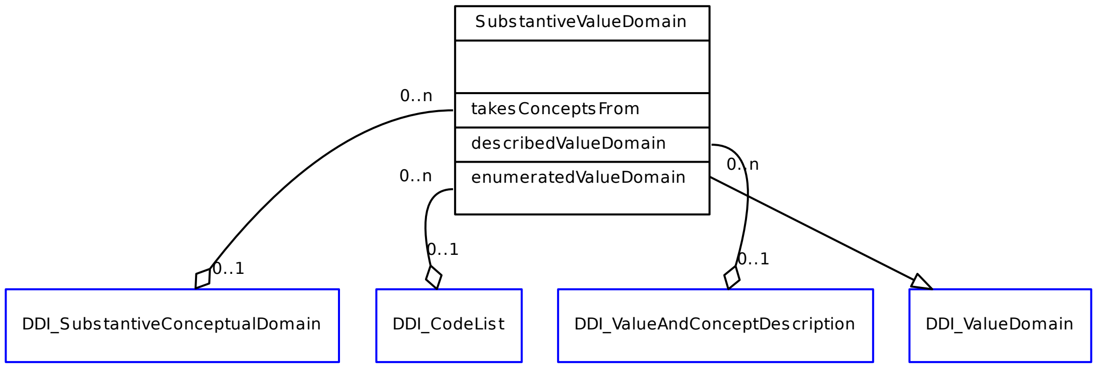
enumeratedValueDomain

A CodeList enumerating the set of valid values.

takesConceptsFrom

Corresponding conceptual definition given by an SubstantiveConceptualDomain.

Graph



7.4.31 TemporalCoverage

Describes the date or time period covered by the annotated object. Allows for the use of a specifying the type of coverage date as well as associated subjects or keywords.

Extends

AnnotatedIdentifiable

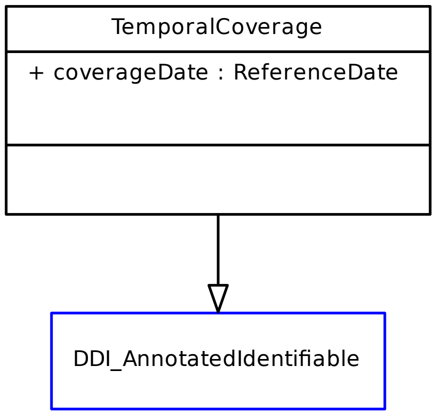
Properties

Name	Type	Cardinality
coverageDate	ReferenceDate	0..n

coverageDate

A date referencing a specific aspect of temporal coverage. The date may be typed to reflect coverage date, collection date, referent date, etc. Subject and Keywords may be associated with the date to specify a specific set of topical information (i.e. Residence associated with a date 5 years prior to the collection date).

Graph



7.4.32 TopicalCoverage

Describes the topical coverage of the module using Subject and Keyword. Note that upper level modules should include all the members of lower level modules. Subjects are members of structured classification systems such as formal subject headings in libraries. Keywords may be structured (e.g. TheSoz thesauri) or unstructured and reflect the terminology found in the document and other related (broader or similar) terms.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
keyword	ExternalControlledVocabularyEntry	0..n
subject	ExternalControlledVocabularyEntry	0..n

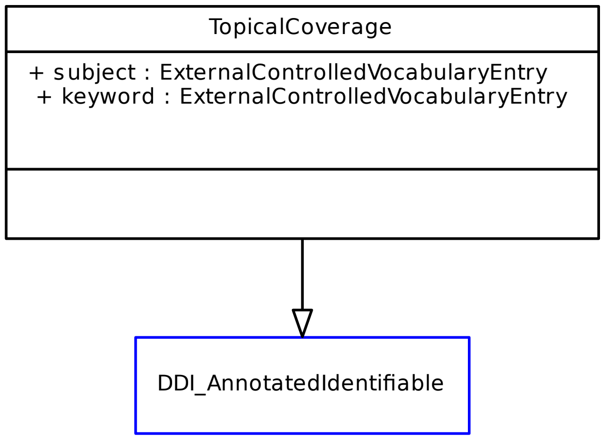
keyword

A keyword that describes the topical coverage of the content of the annotated object. Keywords may be structured (e.g. TheSoz thesauri) or unstructured and reflect the terminology found in the document and other related (broader or similar) terms. Uses and InternationalCodeValue and may indicate the language of the code used.

subject

A subject that describes the topical coverage of the content of the annotated object. Subjects are members of structured classification systems such as formal subject headings in libraries. Uses and InternationalCodeValue and may indicate the language of the code used.

Graph


TopicalCoverage
+ subject : ExternalControlledVocabularyEntry
+ keyword : ExternalControlledVocabularyEntry
DDI_AnnotatedIdentifiable

7.4.33 Unit

The object of interest in a process step related to the collection or use of observational data.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
displayLabel	DisplayLabel	0..n
name	Name	0..n

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

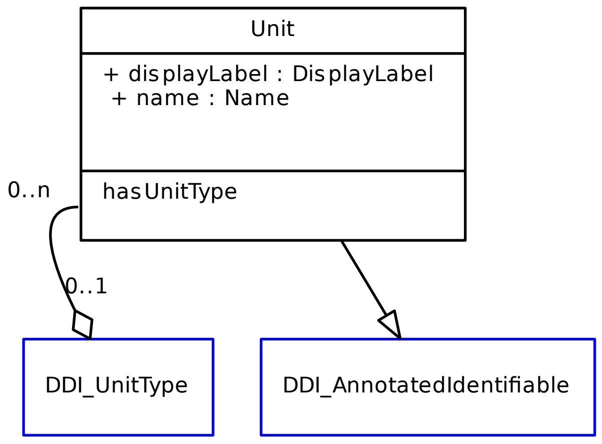
Relationships

Name	Type	Cardinality	allways external
hasUnitType	UnitType	0..n	no

hasUnitType

The UnitType of the Unit

Graph



7.4.34 UnitType

A Unit Type is a class of objects of interest.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
name	Name	0..1

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

Relationships

Name	Type	Cardinality	allways external
realizes	Member	0..n	yes
usesConcept	Concept	0..n	no

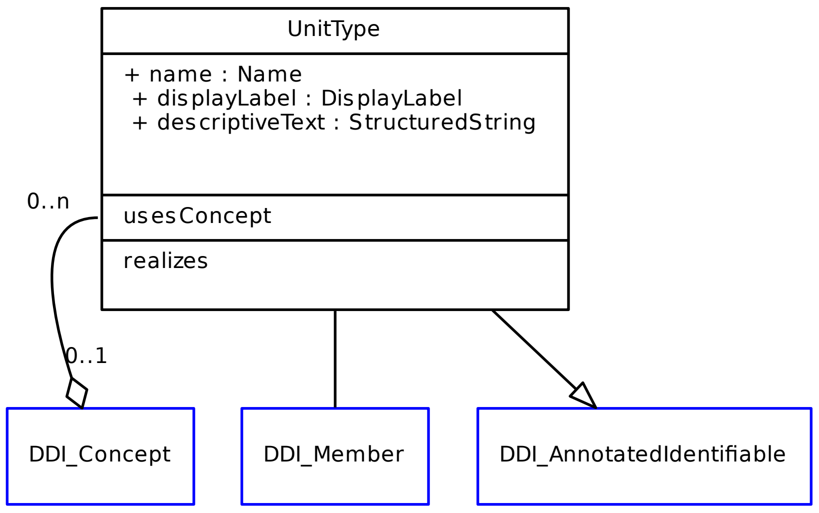
realizes

Class can be used in the role of Member within a Collection

usesConcept

Reference to the Concept that is being used

Graph


UnitType
+ name : Name
+ displayLabel : DisplayLabel
+ descriptiveText : StructuredString
usesConcept
realizes
0..n
0..1
DDI_Concept
DDI_Member
DDI_AnnotatedIdentifiable

7.4.35 Universe

A defined class of people, entities, events, or objects, with no specification of time and geography, contextualizing a Unit Type

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
descriptiveText	StructuredString	0..1
displayLabel	DisplayLabel	0..n
isInclusive	Boolean	0..1
name	Name	0..n

descriptiveText

A short natural language account of the characteristics of the object.

displayLabel

A structured display label providing a fully human readable display for the identification of the object. Supports the use of multiple languages and structured text.

isInclusive

The default value is “true”. The description statement of a universe is generally stated in inclusive terms such as “All persons with university degree”. Occasionally a universe is defined by what it excludes, i.e., “All persons except those with university degree”. In this case the value would be changed to “false”.

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

‘

Relationships

Name	Type	Cardinality	allways external
hasSubuniverse	Universe	1..1	no
narrowsUnitType	UnitType	0..n	no
realizes	Member	0..n	yes
usesConcept	Concept	0..n	no

hasSubuniverse

Universes can have sub universes. A sub-universe class provides a definition to the universes contained within it. For example the Sub-Universe Class of Gender for the Universe Population may contain the Universe Males and the Universe Females

narrowsUnitType

Reference to the Unit Type that the Universe definition narrows.

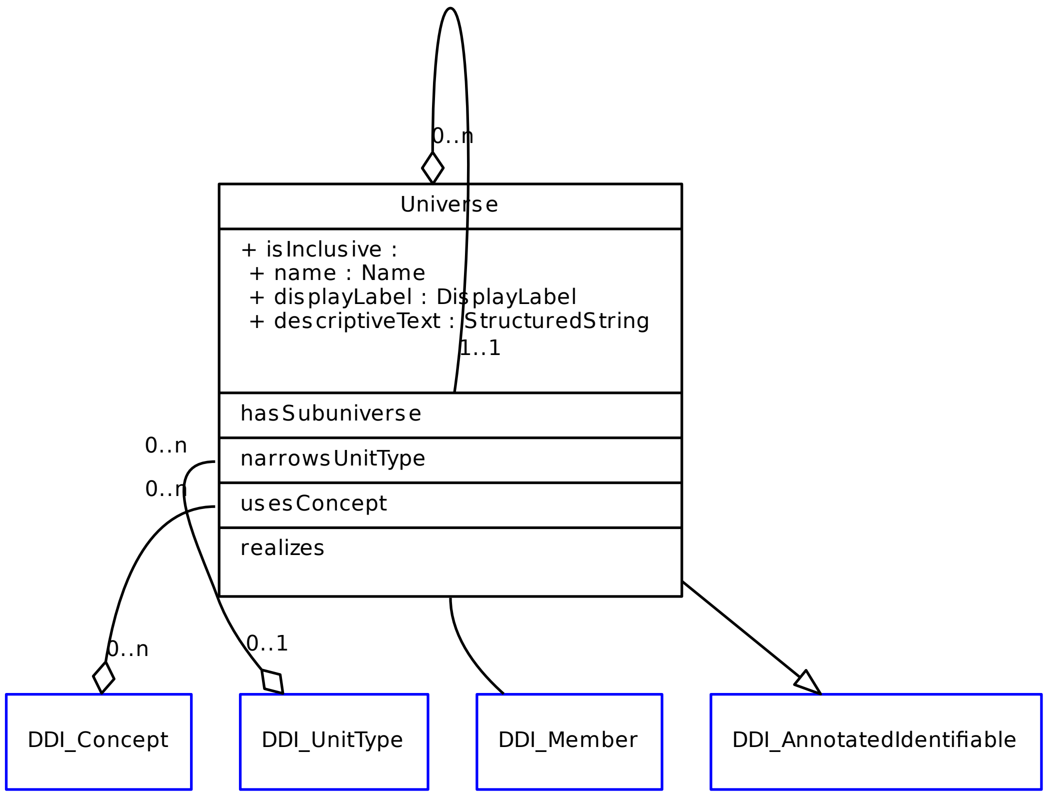
realizes

Class can be used in the role of Member within a Collection

usesConcept

Reference to the Concept that is being used

Graph



7.4.36 ValueAndConceptDescription

A formal description of a set of values.

Extends

Identifier

Properties

Name	Type	Cardinality
classificationLevel	CategoryRelationCode	0..1
description	StructuredString	0..1
logicalExpression	String	0..1
maximumValueExclusive	String	0..1
maximumValueInclusive	String	0..1
minimumValueExclusive	String	0..1
minimumValueInclusive	String	0..1
regularExpression	String	0..1

classificationLevel

Indicates the type of relationship, nominal, ordinal, interval, ratio, or continuous. Use where appropriate for the representation type.

description

A formal description of the set of values.

logicalExpression

A logical expression where the values of "x" making the expression true are the members of the set of valid values. Example: x >0 describes the real numbers greater than 0

maximumValueExclusive

A string denotFrom https://www.w3.org/TR/tabular-metadata/ 5.11.2 maxExclusive: "An atomic property that contains a single number or string that is the maximum valid value (exclusive). The value of this property becomes the maximum exclusive annotation for the described datatype. See Value Constraints in [tabular-data-model] for details." DDI3.2 handles this with a Boolean isInclusive attribute. ing the maximumpossible value (excluding this value)

maximumValueInclusive

A string denoting the maximum possible value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 maximum: "An atomic property that contains a single number or string that is the maximum valid value (inclusive); equivalent to maxInclusive. The value of this property becomes the maximum annotation for the described datatype. See Value Constraints in [tabular-data-model] for details."

minimumValueExclusive

A string denoting the minimum possible value (excluding this value) From https://www.w3.org/TR/tabular-metadata/ 5.11.2 minExclusive: "An atomic property that contains a single number or string that is the minimum valid value (exclusive). The value of this property becomes the minimum exclusive annotation for the described datatype. See Value Constraints in [tabular-data-model] for details." DDI3.2 handles this with a Boolean isInclusive attribute.

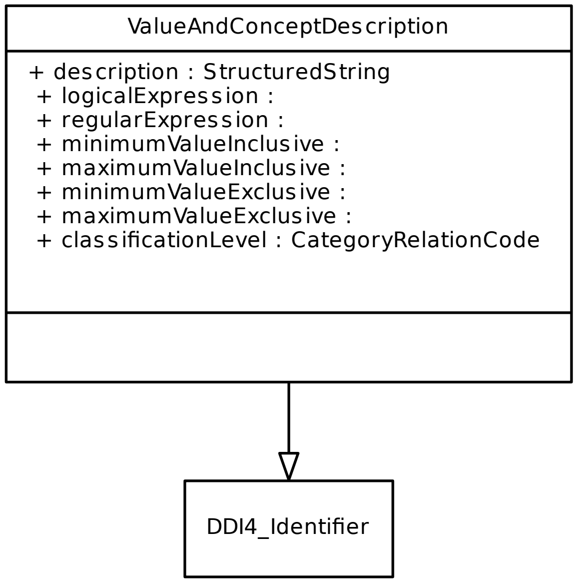
minimumValueInclusive

A string denoting the minimum possible value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 minimum: “An atomic property that contains a single number or string that is the minimum valid value (inclusive); equivalent to minInclusive. The value of this property becomes the minimum annotation for the described datatype. See Value Constraints in [tabular-data-model] for details.”

regularExpression

A regular expression where strings matching the expression belong to the set of valid values.

Graph


ValueAndConceptDescription
+ description : StructuredString
+ logicalExpression :
+ regularExpression :
+ minimumValueInclusive :
+ maximumValueInclusive :
+ minimumValueExclusive :
+ maximumValueExclusive :
+ classificationLevel : CategoryRelationCode
DDI4_Identifier

7.4.37 ValueMapping

Provides physical characteristics for an InstanceVariable as part of a PhysicalLayout

Extends

Identifiable

Properties

Name	Type	Cardinality
decimalPositions	Integer	0..1
defaultDecimalSeparator	String	0..1
defaultDigitGroupSeparator	String	0..1
defaultValue	String	0..1
format	ExternalControlledVocabularyEntry	0..1
length	Integer	0..1
maximumLength	Integer	0..1
minimumLength	Integer	0..1
nullSequence	String	0..1
numberPattern	String	0..1
physicalDatatype	ExternalControlledVocabularyEntry	0..1
required	Boolean	0..1
scale	Integer	0..1

decimalPositions

The number of decimal positions expressed as an integer. Used when the decimal position is implied (no decimal separator is present) See DDI 3.2 ManagedNumericRepresentation@decimalPositions

defaultDecimalSeparator

The string separating the integer part from the fractional part of a decimal or real number. In W3C part of the datatype format From https://www.w3.org/TR/tabular-metadata/ tabular 6.4.2 decimalChar: "A string whose value is used to represent a decimal point within the number. The default value is ".". If the supplied value is not a string, implementations MUST issue a warning and proceed as if the property had not been specified."

defaultDigitGroupSeparator

A string separating groups of digits (for readability). In W3C part of the datatype format From https://www.w3.org/TR/tabular-metadata/ tabular 6.4.2 groupChar: "A string whose value is used to group digits within the number. The default value is null. If the supplied value is not a string, implementations MUST issue a warning and proceed as if the property had not been specified."

defaultValue

A default string indicating the value to substitute for an empty string. From https://www.w3.org/TR/tabular-metadata/ Inherited 5.7 default "An atomic property holding a single string that is used to create a default value for the cell in cases where the original string value is an empty string. See Parsing Cells in [tabular-data-model] for more details. If not specified, the default for the default property is the empty string, "". The value of this property becomes the default annotation for the described column."

format

This defines the format of the physical representation of the value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 format: "An atomic property that contains either a single string or an object that defines the format of a value of this type, used when parsing a string value as described in Parsing Cells in [tabular-data-model]. The value of this property becomes the format annotation for the described datatype." See https://www.w3.org/TR/tabular-metadata/ Tabular 6.4.2 'Formats for numeric datatypes' this may include decimalChar, groupChar, pattern "By default, numeric values must be in the formats defined in [xmlschema11-2]. It is not uncommon for numbers within tabular data to be formatted for human consumption, which may involve using commas for decimal points, grouping digits in the number using commas, or adding percent signs to the number." See https://www.w3.org/TR/tabular-metadata/ Tabular 6.4. Formats for Booleans " Boolean values may be represented in many ways aside from the standard 1 and 0 or true and false." See https://www.w3.org/TR/tabular-metadata/ 6.4.4. Formats for dates and times "By default, dates and times are assumed to be in the format defined in [xmlschema11-2]. However dates and times are commonly represented in tabular data in other formats." See https://www.w3.org/TR/tabular-metadata/ 6.4.5 Formats for durations "Durations MUST be formatted and interpreted as defined in [xmlschema11-2], using the [ISO8601] format -?PnYnMnDTnHnMnS. For example, the duration P1Y1D is used for a year and a day; the duration PT2H30M for 2 hours and 30 minutes." See https://www.w3.org/TR/tabular-metadata/ 6.4.6 Formats for other types "If the datatype base is not numeric, boolean, a date/time type, or a duration type, the datatype format annotation provides a regular expression for the string values, with syntax and processing defined by [ECMASCRIPT]. If the supplied value is not a valid regular expression, implementations MUST issue a warning and proceed as if no format had been provided." From DDI3.2 ManagedNumericRepresentation@format "A format for number expressed as a string." From DDI3.2 ManagedDateTimeRepresentation_DateFieldFormat "Describes the format of the date field, in formats such as YYYY/MM or MM-DD-YY, etc. If this element is omitted, then the format is assumed to be the XML Schema format corresponding to the type attribute value."

length

The length of the physical representation of the value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 length: "A numeric atomic property that contains a single integer that is the exact length of the value. The value of this property becomes the length annotation for the described datatype. See Length Constraints in [tabular-data-model] for details." Corresponds to DDI2.5 var/location/width and DDI 3.2 PhysicalLocation/Width

maximumLength

The largest possible value of the length of the physical representation of the value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 maxLength: "A numeric atomic property that contains a single integer that is the maximum length of the value. The value of this property becomes the maximum length annotation for the described datatype. See Length Constraints in [tabular-data-model] for details."

minimumLength

The smallest possible value for the length of the physical representation of the value. From https://www.w3.org/TR/tabular-metadata/ 5.11.2 minLength: "An atomic property that contains a single integer that is the minimum length of the value. The value of this property becomes the minimum length annotation for the described datatype. See Length Constraints in [tabular-data-model] for details."

nullSequence

A string indicating a null value. From https://www.w3.org/TR/tabular-metadata/ 4.3 null — "the string or strings which cause the value of cells having string value matching any of these values to be null." From Inherited 5.7 null: "An atomic property giving the string or strings used for null values within the data. If the string value of the cell is equal to any one of these values, the cell value is null. See Parsing Cells in [tabular-data-model] for more details. If not specified, the default for the null property is the empty string "". The value of this property becomes the null annotation for the described column."

numberPattern

A pattern description of the format of a numeric value. In W3C part of the datatype format From https://www.w3.org/TR/tabular-metadata/ tabular 6.4.2 pattern: "A number format pattern as defined in [UAX35] http://www.unicode.org/reports/tr35/tr35-31/tr35-numbers.html#Number_Format_Patterns . Implementations MUST recognise number format patterns containing the symbols 0, #, the specified decimalChar (or "." if unspecified), the specified groupChar (or "," if unspecified), E, +, % and ‰. Implementations MAY additionally recognise number format patterns containing other special pattern characters defined in [UAX35]. If the supplied value is not a string, or if it contains an invalid number format pattern or uses special pattern characters that the implementation does not recognise, implementations MUST issue a warning and proceed as if the property had not been specified. f the datatype format annotation is a single string, this is interpreted in the same way as if it were an object with a pattern property whose value is that string. If the groupChar is specified, but no pattern is supplied, when parsing the string value of a cell against this format specification, implementations MUST recognise and parse numbers that consist of: an optional + or - sign, … Implementations MAY also recognise numeric values that are in any of the standard-decimal, standard-percent or standard-scientific formats listed in the Unicode Common Locale Data Repository. …"

physicalDatatype

The base datatype of the physical representation. An integer InstanceVariable might, for example, be stored as a floating point number. From https://www.w3.org/TR/tabular-metadata/ Inherited 5.7 datatype: "An atomic property that contains either a single string that is the main datatype of the values of the cell or a datatype description object. If the value of this property is a string, it MUST be the name of one of the built-in datatypes defined in section 5.11.1 Built-in Datatypes and this value is normalized to an object whose base property is the original string value. If it is an object then it describes a more specialized datatype. If a cell contains a sequence (i.e. the separator property is specified and not null) then this property specifies the datatype of each value within that sequence. See 5.11 Datatypes and Parsing Cells in [tabular-data-model] for more details. The normalized value of this property becomes the datatype annotation for the described column. "

required

If True a value is required for this variable. NOTE: this might be better at the InstanceVariable or higher in the variable cascade. From https://www.w3.org/TR/tabular-metadata/ Inherited 5.7 required: "A boolean atomic property taking a single value which indicates whether the cell value can be null. See Parsing Cells in [tabular-data-model] for more details. The default is false, which means cells can have null values. The value of this property becomes the required annotation for the described column."

scale

The scale of the number expressed as an integer (for example a number expressed in 100's, 5 = 500 would have a scale of 100). From DDI 3.2 ManagedNumericRepresentation@scale:

'

Relationships

Name	Type	Cardinality	allways external
describesSegmentLocation	PhysicalSegmentLocation	0..n	no
formatsInstanceVariable	InstanceVariable	0..n	no
realizes	Member	0..n	yes

describesSegmentLocation

Uses a PhysicalSegmentLocation to describe where in the physical record a segment representing the InstanceVarible is. This could be, for example, described as a start position and end position value for characters in a text record via the SegmentByText extension of PhysicalSegmentLocation.

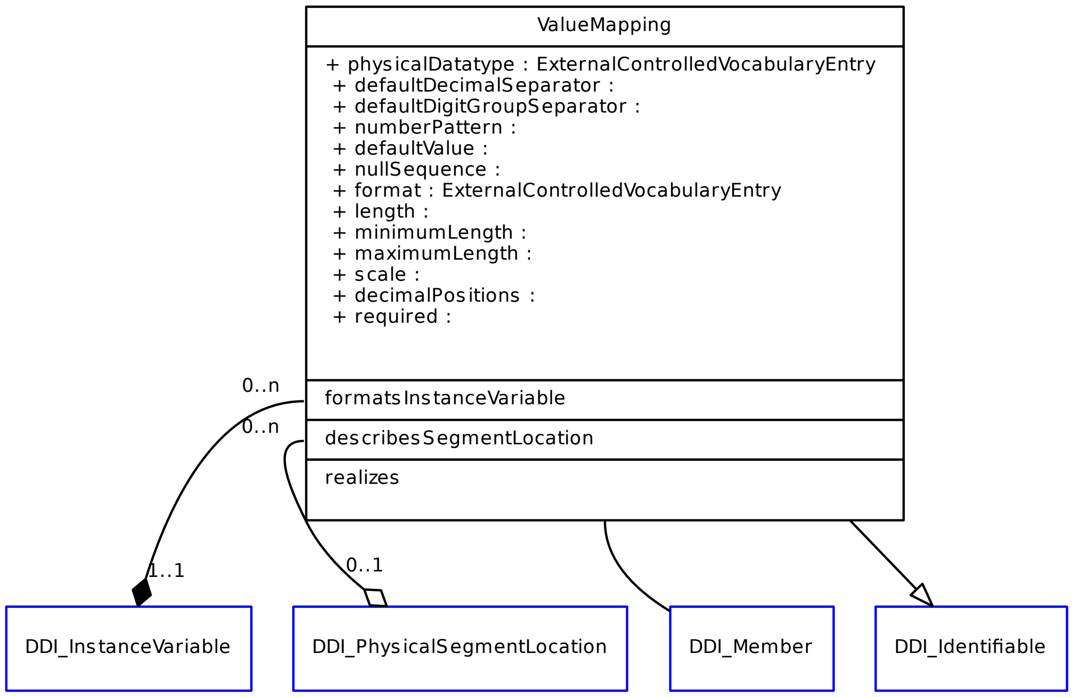
formatsInstanceVariable

Describes the physical representation of the InstanceVariable

realizes

ValueMappings are members of the PhysicalLayout.

Graph



7.4.38 Viewpoint

The assignment of measure, identifier and attribute roles to InstanceVariables

Extends

AnnotatedIdentifiable

Relationships

Name	Type	Cardinality	allways external
hasAttributeRole	AttributeRole	1..n	no
hasIdentifierRole	IdentifierRole	1..1	no
hasMeasureRole	MeasureRole	1..1	no

hasAttributeRole

The InstanceVaribles functioning as attributes

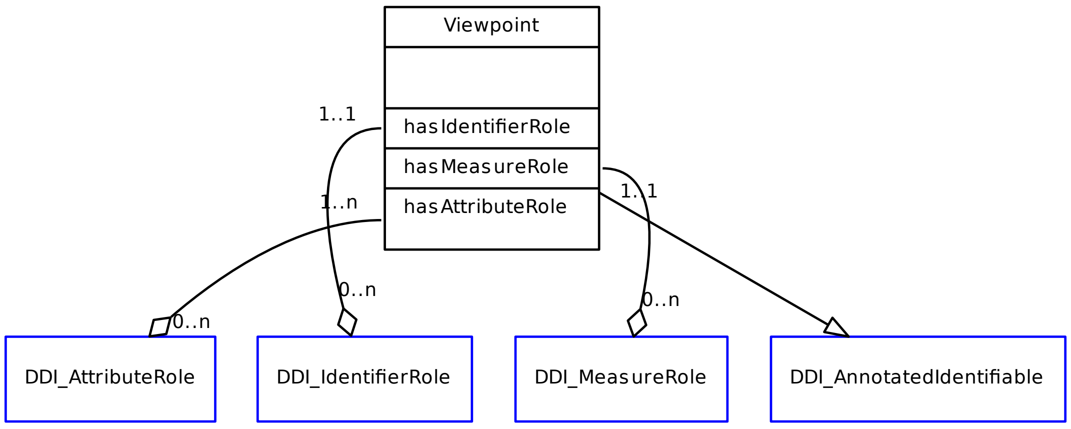
hasIdentifierRole

The InstanceVaribles functioning as identifiers

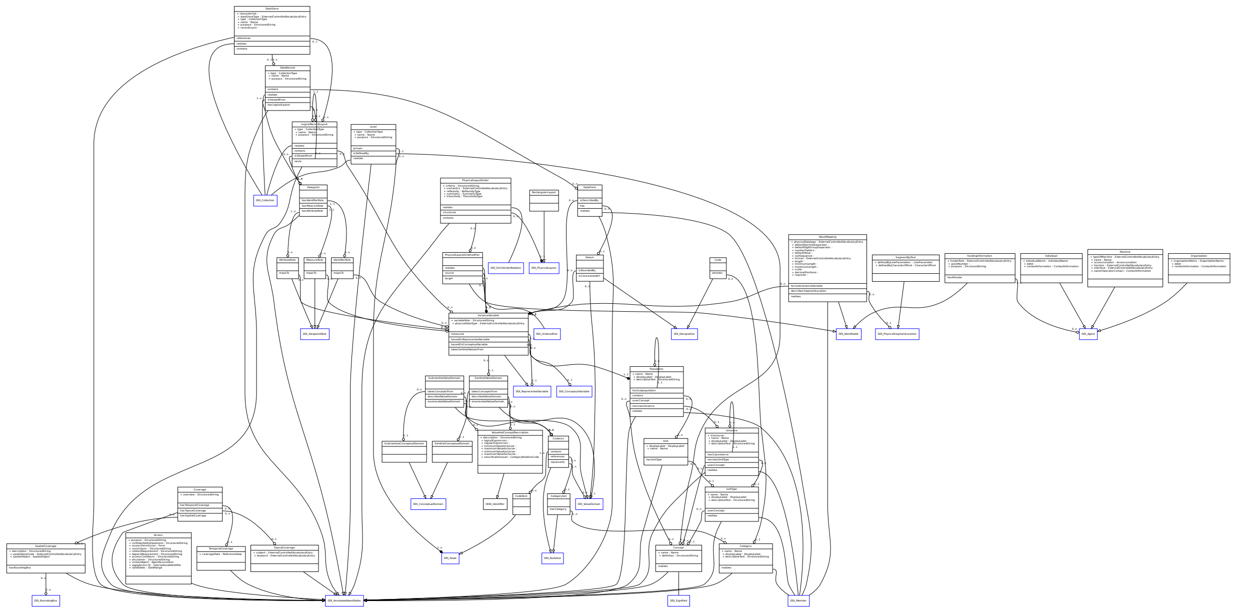
hasMeasureRole

The InstanceVaribles functioning as measures

Graph



7.4.39 Graph



7.5 AgentRegistryView

Purpose: The Agent Registry View supports the creation of a listing of Agents (Organization, Individual, Machine) for the purpose of maintenance and reuse. The listing of Agents can be organized in pair-wise descriptions within Agent Relationships. These may be typed and provided with effective date ranges either as a set of relationships of a specified type or as individual pairs.

Use Cases: Maintaining a listing of all Organizations and Individuals related to a project or statistical activities, i.e. Individuals and Organizations related to the IPUMS projects at the Minnesota Population Center or the Departments and Individuals involved in the production of the Australian Census within the Australian Bureau of Statistics.

Target Audience: Organizations working with a number of Individuals and Organizations whose descriptive information and interrelationships change over time.

Included Classes: Agent Registry, Agent Listing, Agent Relationships, Organization, Individual, Machine, Agent Similarity, Agent Similarity Pair, Agent Hierarchy, Agent Hierarchy Pair, Access, Funding Information, External Material, Coverage, Bounding Box.

Restricted Classes: All classes are included in their entirety, there are no restrictions.

General Documentation: The primary class is the Agent Registry which contains Agent Listings and Agent Relationships. Agents may be of any type supported by DDI at the time of this release. Agent Relationships are described by Agent Similarity specified with Agent Similarity Pair (unordered pair of agents) or Agent Hierarchy specified with Agent Hierarchy Pair (ordered pair reflecting a hierarchical relationship between the agents). The Functional View includes the standard document description and Access, Coverage, Funding Information, External Materials, and Bounding Box classes have been added to support a full description of the instance to support discovery and access.

A functional view is a collection of classes in DDI that covers a functional use case. These are not namespaces.

7.5.1 Access

Describes access to the annotated object. This item includes a confidentiality statement, descriptions of the access permissions required, restrictions to access, citation requirements, depositor requirements, conditions for access, a disclaimer, any time limits for access restrictions, and contact information regarding access.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
accessConditions	StructuredString	0..1
accessPermission	Form	0..n
applyAccessTo	InternationalIdentifier	0..n
citationRequirement	StructuredString	0..1
confidentialityStatement	StructuredString	0..1
contactAgent	AgentAssociation	0..n
depositRequirement	StructuredString	0..1
disclaimer	StructuredString	0..1
purpose	StructuredString	0..1
restrictions	StructuredString	0..1
validDates	DateRange	0..1

accessConditions

A statement regarding conditions for access. May be expressed in multiple languages and supports the use of structured content.

accessPermission

A link to a form used to provide access to the data or metadata including a statement of the purpose of the form.

applyAccessTo

Identification for an object covered by the access description. This may be any annotated object (collection, publication, identifiable object).

citationRequirement

A statement regarding the citation requirement. May be expressed in multiple languages and supports the use of structured content.

confidentialityStatement

A statement regarding the confidentiality of the related data or metadata.

contactAgent

The agent to contact regarding access including the role of the agent.

depositRequirement

A statement regarding depositor requirements. May be expressed in multiple languages and supports the use of structured content.

disclaimer

A disclaimer regarding the liability of the data producers or providers. May be expressed in multiple languages and supports the use of structured content.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured text

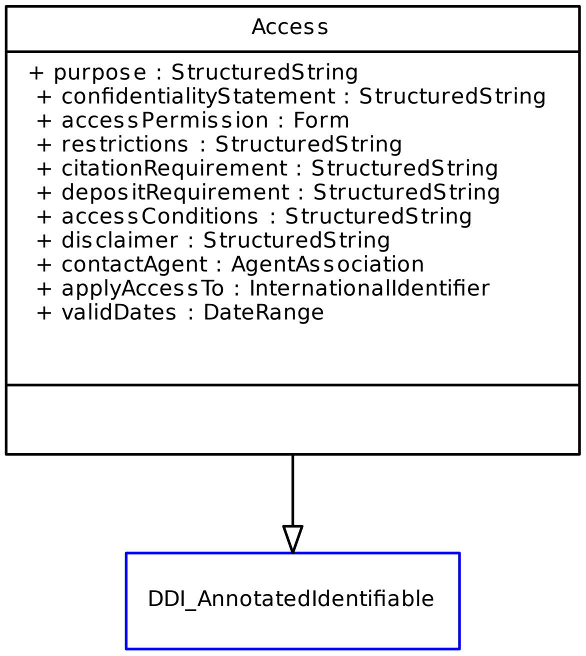
restrictions

A statement regarding restrictions to access. May be expressed in multiple languages and supports the use of structured content.

validDates

The date range or start date of the access description.

Graph


Access
+ purpose : StructuredString
+ confidentialityStatement : StructuredString
+ accessPermission : Form
+ restrictions : StructuredString
+ citationRequirement : StructuredString
+ depositRequirement : StructuredString
+ accessConditions : StructuredString
+ disclaimer : StructuredString
+ contactAgent : AgentAssociation
+ applyAccessTo : InternationalIdentifier
+ validDates : DateRange
DDI_AnnotatedIdentifiable

7.5.2 AgentHierarchy

A set of relationships between pairs of Agents that is hierarchical (Parent/Child) in nature. Uses the pattern for Immediate Precedence Relation where the relation over members in the Agent Registry is characterized by the following: Reflexivity=Anti_Reflexive; Symmetry=Anti_Symmedtric; Transitivity=Anti_Transitive. It must contain like items (Agents).

Extends

AgentBinary

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1

reflexivity

Fixed to Anti_Reflexive

semantics

Controlled vocabulary for the ImmediatePrecedenceRelation semantics. It should contain, at least, the following: Parent_Of, Child_Of, Next, Previous, Instance_Of, Temporal_Meets.

symmetry

Fixed to Anti_Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown.

transitivity

Fixed to Anti_Transitive

‘

Relationships

Name	Type	Cardinality	allways external
contains	AgentHierarchyPair	0..n	no
realizes	ImmediatePrecedenceRelation	0..n	yes
structures	AgentListing	0..n	no

contains

Pair of Agents whose relationship is hierarchical in nature.

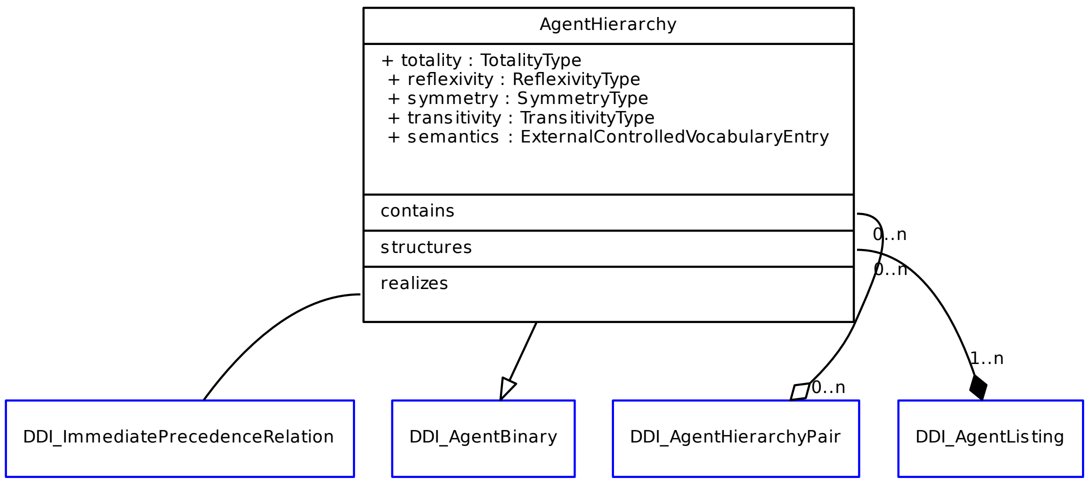
realizes

Uses the pattern of Immediate Precedence Relation

structures

The Agent Listing or Listings whose Agents are grouped by pairs to support a variety of structures

Graph


AgentHierarchy
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
contains
structures
realizes
0..n
0..n
1..n
0..n
DDI_ImmediatePrecedenceRelation
DDI_AgentBinary
DDI_AgentHierarchyPair
DDI_AgentListing

7.5.3 AgentHierarchyPair

Used to define a pair of agents in a hierarchical structure where the source is the parent and the target is the child.

Extends

Identifiable

Properties

Name	Type	Cardinality
effectiveDates	DateRange	0..1

effectiveDates

The effective dates of the relation. A structured DateRange with start and end Date (both with the structure of Date and supporting the use of ISO and non-ISO date structures); Use to relate a period with a start and end date.

‘

Relationships

Name	Type	Cardinality	allways external
realizes	OrderedPair	0..n	yes
source	Agent	0..n	no
target	Agent	0..n	no

realizes

Uses the pattern of an OrderedPair

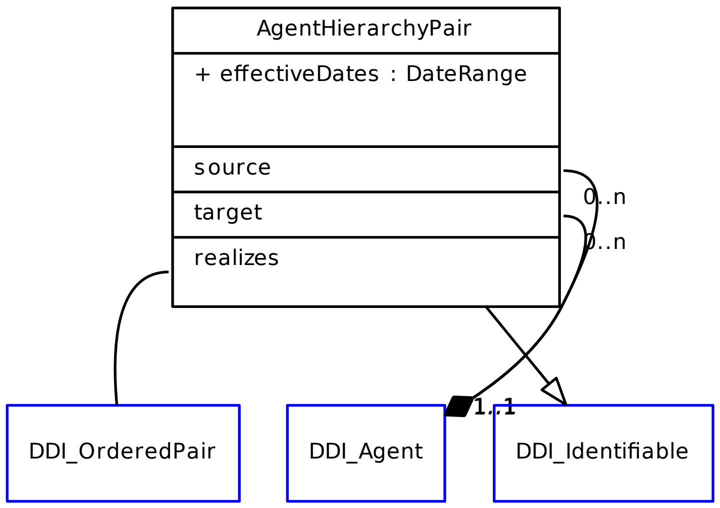
source

The Parent agent in the hierarchical relationship

target

The Child agent in the hierarchical relationship

Graph



7.5.4 AgentListing

A listing of Agents of any type

Extends

Identifiable

Properties

Name	Type	Cardinality
maintainer	AgentAssociation	0..1
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

maintainer

The agent responsible for maintaining the Agent Listing

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of the Agent Listing. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
contains	Agent	0..n	no
realizes	Collection	0..n	yes

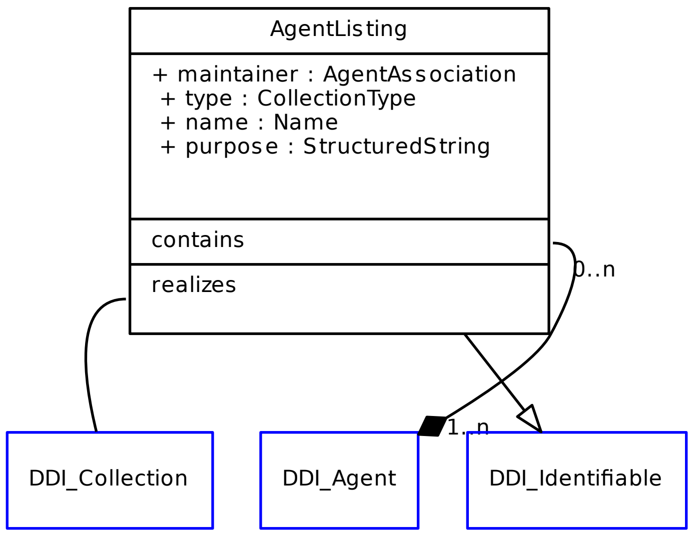
contains

The Agents contained in the Listing

realizes

Uses the pattern Collection

Graph



7.5.5 AgentRegistry

A collection of Agent Listings and/or AgentRelationships describing individual agents and their relationships to each other over time.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
purpose	StructuredString	0..1
usage	StructuredString	0..1
validDates	DateRange	0..1

purpose

Explanation of the intent of creating and maintaining this registry. Supports the use of multiple languages and structured text.

usage

Explanation of the ways in which the contents of the registry, Agent Listing and Agent Relationships can be employed. Supports the use of multiple languages and structured text.

validDates

A structured DateRange with start and end Date (both with the structure of Date and supporting the use of ISO and non-ISO date structures); Use to relate a period with a start and end date.

‘

Relationships

Name	Type	Cardinality	allways external
hasListing	AgentListing	0..n	no
hasRelationships	AgentRelationships	0..n	no
realizes	Collection	0..n	yes

hasListing

Specialization of the contains relation in Collection. Identifies the specific Agent Listings defined in the Registry.

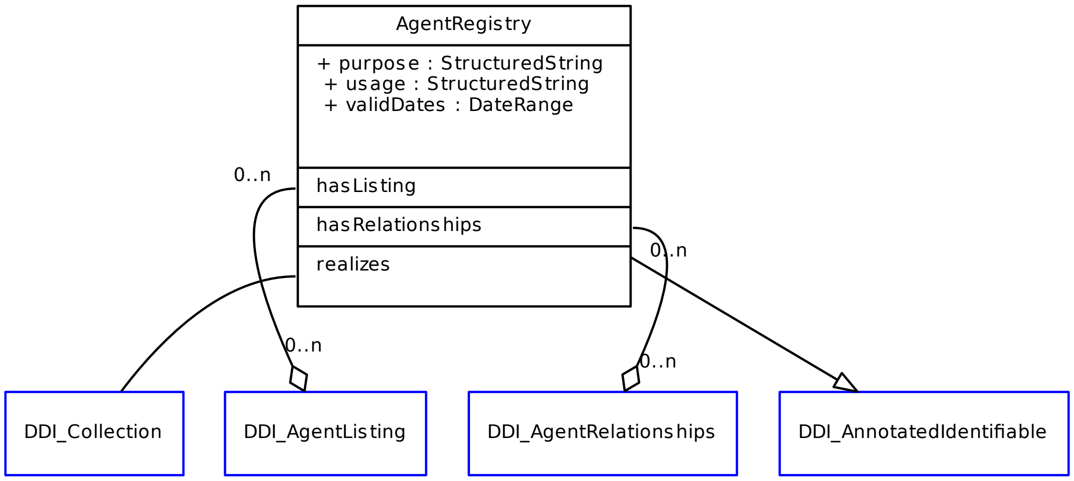
hasRelationships

Specialization of the contains relation in Collection. Identifies the specific Agent Relationships defined in the Registry.

realizes

Uses the pattern of Collection

Graph



7.5.6 AgentRelationships

A collection of relationships between pairs of Agents.

‘

Properties

Name	Type	Cardinality
maintainer	AgentAssociation	0..1
name	Name	0..n
purpose	StructuredString	0..1
type	CollectionType	0..1

maintainer

The agent who maintains the relationship information

name

A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

purpose

Explanation of the intent of the collection of Agent Relationships. Supports the use of multiple languages and structured text.

type

Whether the collection is a bag or a set: a bag is a collection with duplicates allowed, a set is a collection without duplicates.

‘

Relationships

Name	Type	Cardinality	allways external
contains	AgentBinary	0..n	no
realizes	Collection	0..n	yes

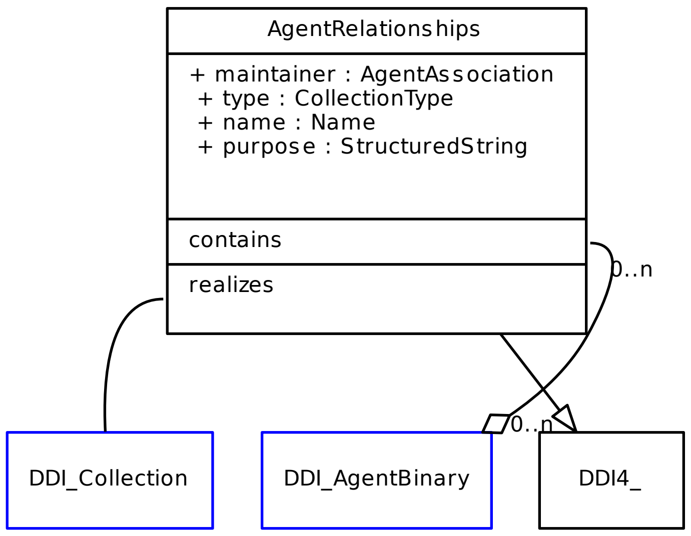
contains

Agent Hierarchy or Agent Similarity sets described in the Agent Relationship

realizes

Uses the pattern for a Collection

Graph



7.5.7 AgentSimilarity

A specific type of Agent Binary relationship that is used when the intent is to express the similarity between a pair of agents. Uses the pattern for EquivalenceRelation. Each pair must contain like items. The relation of the members in the collection is characterized by the following assignments: Reflexivity=Reflexive; Symmetry=Semetric; Transitivity=Transitive.

Extends

AgentBinary

Properties

Name	Type	Cardinality
reflexivity	ReflexivityType	1..1
semantics	ExternalControlledVocabularyEntry	0..1
symmetry	SymmetryType	1..1
totality	TotalityType	1..1
transitivity	TransitivityType	1..1

reflexivity

Fixed to Reflexive

semantics

Controlled vocabulary for the equivalence relation semantics. It should contain, at least, the following: Same_As, Similar_To, Congruent_To, Compatible_With, Equidistant_From, Temporal_Equals.

symmetry

Fixed to Symmetric

totality

Controlled Vocabulary to specify whether the relation is total, partial or unknown

transitivity

Fixed to Transitive

‘

Relationships

Name	Type	Cardinality	allways external
contains	AgentSimilarityPair	0..n	no
realizes	EquivalenceRelation	0..n	yes
structures	AgentListing	0..n	no

contains

Unordered pairs of Agents

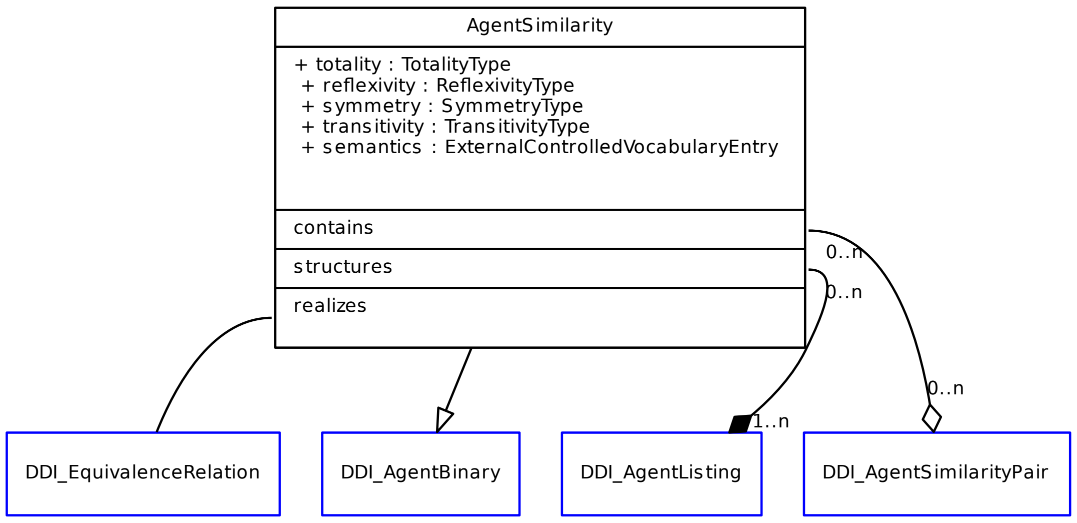
realizes

Reflects the pattern of EquivalenceRelation

structures

AgentListing or Listings whose Agents are grouped by pairs to support a variety of structures.

Graph


AgentSimilarity
+ totality : TotalityType
+ reflexivity : ReflexivityType
+ symmetry : SymmetryType
+ transitivity : TransitivityType
+ semantics : ExternalControlledVocabularyEntry
contains
structures
realizes
0..n
0..n
0..n
1..n
DDI_EquivalenceRelation
DDI_AgentBinary
DDI_AgentListing
DDI_AgentSimilarityPair

7.5.8 AgentSimilarityPair

Describes simple pair of similar agents of any type (i.e. Organization, Individual, Machine, etc.). Uses the pattern for an UnorderedPair.

Extends

Identifiable

Properties

Name	Type	Cardinality
effectiveDates	DateRange	0..1

effectiveDates

The effective dates of the relation. A structured DateRange with start and end Date (both with the structure of Date and supporting the use of ISO and non-ISO date structures); Use to relate a period with a start and end date.

‘

Relationships

Name	Type	Cardinality	allways external
maps	Agent	0..n	no
realizes	UnorderedPair	0..n	yes

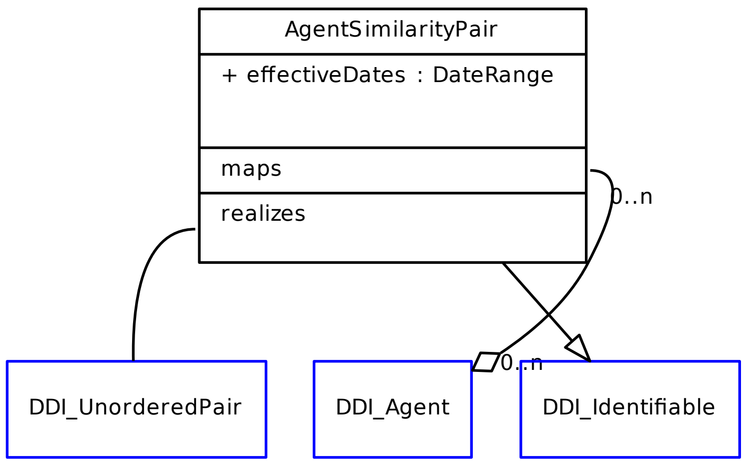
maps

Two Agents that are similar within the context of the type of relationship.

realizes

Reflects the pattern of an UnorderedPair

Graph



7.5.9 BoundingBox

A type of Spatial coverage describing a rectangular area within which the actual range of location fits. A BoundingBox can be described by 4 numbers, or two x,y coordinates - the maxima of the north, south, east, and west coordinates found in the area.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
eastLongitude	Real	1..1
northLatitude	Real	1..1
southLatitude	Real	1..1
westLongitude	Real	1..1

eastLongitude

The easternmost coordinate expressed as a decimal between the values of -180 and 180 degrees

northLatitude

The northernmost coordinate expressed as a decimal between the values of -90 and 90 degrees.

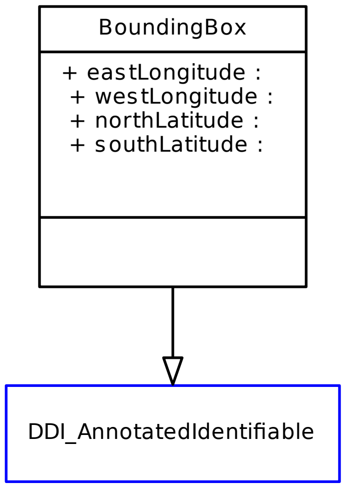
southLatitude

The southermost latitude expressed as a decimal between the values of -90 and 90 degrees

westLongitude

The westernmost coordinate expressed as a decimal between the values of -180 and 180 degrees

Graph



7.5.10 Coverage

Coverage information for an annotated object. Includes coverage information for temporal, topical, and spatial coverage.

Extends

AnnotatedIdentifiable

Properties

Name	Type	Cardinality
overview	StructuredString	0..1

overview

A generic description including temporal, topical, and spatial coverage that is the equivalent of dc:coverage (the refinement base of dcterms:spatial and dcterms:temporal. Use specific coverage content for detailed information. Short natural language account of the information obtained from the combination of properties and relationships associated with an object. Supports the use of multiple languages and structured text.

‘

Relationships

Name	Type	Cardinality	allways external
hasSpatialCoverage	SpatialCoverage	0..n	yes
hasTemporalCoverage	TemporalCoverage	0..n	yes
hasTopicalCoverage	TopicalCoverage	0..n	yes

hasSpatialCoverage

Description of the spatial (geographic) coverage of the contents of the annotated object.

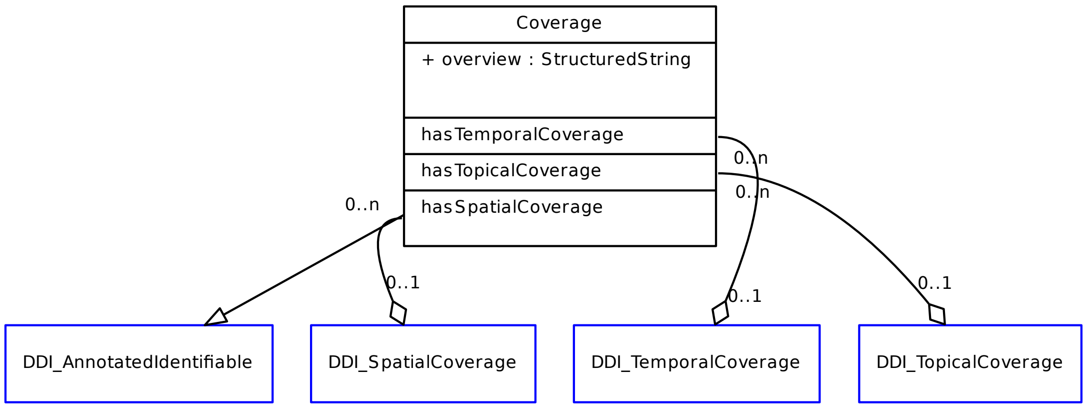
hasTemporalCoverage

The dates and time periods described by the contents of the annotated object.

hasTopicalCoverage

The topics covered by the contents of the annotated object. These may be expressed by subject classification systems and structured or unstructured keywords.

Graph



7.5.11 ExternalMaterial

ExternalMaterial describes the location, structure, and relationship to the DDI metadata instance for any material held external to that instance. This includes citations to such material, an external reference to a URL (or other URI), and a statement about the relationship between the cited ExternalMaterial the contents of the DDI instance. It should be used as follows: As an extension base for specific external materials found within DDI (such as an External Aid); as a target object from a relationship which clarifies its role within a class; or as the target of a relatedResource within an annotation.

Extends

Identifiable

Properties

Name	Type	Cardinality
citationOfExternalMaterial	Annotation	0..1
descriptiveText	StructuredString	0..1
externalURLReference	anyURI	0..n
externalURNReference	anyURI	0..1
mimeType	ExternalControlledVocabularyEntry	0..1
relationshipDescription	StructuredString	0..n
segment	Segment	0..n
typeOfMaterial	ExternalControlledVocabularyEntry	0..1

citationOfExternalMaterial

Bibliographic citation for the external resource.

descriptiveText

A description of the referenced material. This field can map to a Dublin Core abstract. Note that Dublin Core does not support structure within the abstract element. Supports multiple languages and optional structured content.

externalURLReference

Contains a URL which indicates the location of the cited external resource.

externalURNReference

Contains a URN which identifies the cited external resource.

mimeType

Provides a standard Internet MIME type for use by processing applications.

relationshipDescription

Describes the reason the external material is being related to the DDI metadata instance.

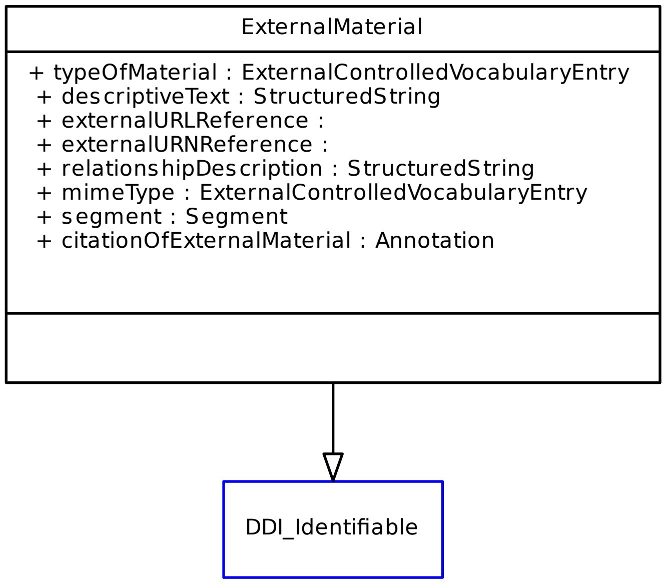
segment

Can describe a segment within a larger object such as a text or video segment.

typeOfMaterial

Designation of the type of material being described. Supports the use of a controlled vocabulary.

Graph


ExternalMaterial
+ typeOfMaterial : ExternalControlledVocabularyEntry
+ descriptiveText : StructuredString
+ externalURLReference :
+ externalURNReference :
+ relationshipDescription : StructuredString
+ mimeType : ExternalControlledVocabularyEntry
+ segment : Segment
+ citationOfExternalMaterial : Annotation
DDI_Identifiable

7.5.12 FundingInformation

Provides information about the individual, agency and/or grant(s) which funded the described entity. Lists a reference to the agency or individual as described by a DDI Agent, the role of the funder, the grant number(s) and a description of the funding activity.

Extends

Identifiable

Properties

Name	Type	Cardinality
funderRole	ExternalControlledVocabularyEntry	0..1
grantNumber	String	0..n
purpose	StructuredString	0..1

funderRole

Role of the funding organization or individual. Supports the use of a controlled vocabulary.

grantNumber

The identification code of the grant or other monetary award which provided funding for the described object.

purpose

Explanation of the intent of some decision or object. Supports the use of multiple languages and structured tex
'

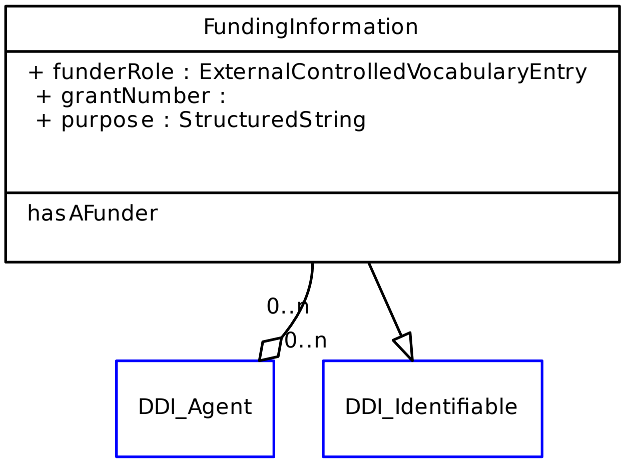
Relationships

Name	Type	Cardinality	allways external
hasAFunder	Agent	0..n	no

hasAFunder

Funding has been provided by

Graph


FundingInformation
+ funderRole : ExternalControlledVocabularyEntry
+ grantNumber :
+ purpose : StructuredString
hasAFunder
0..n
0..n
DDI_Agent
DDI_Identifiable

7.5.13 Individual

A person who acts, or is designated to act towards a specific purpose.

Extends

Agent

Properties

Name	Type	Cardinality
contactInformation	ContactInformation	0..1
ddiId	String	0..n
individualName	IndividualName	0..n

contactInformation

Contact information for the individual including location specification, address, URL, phone numbers, and other means of communication access. Sets of information can be repeated and date-stamped.

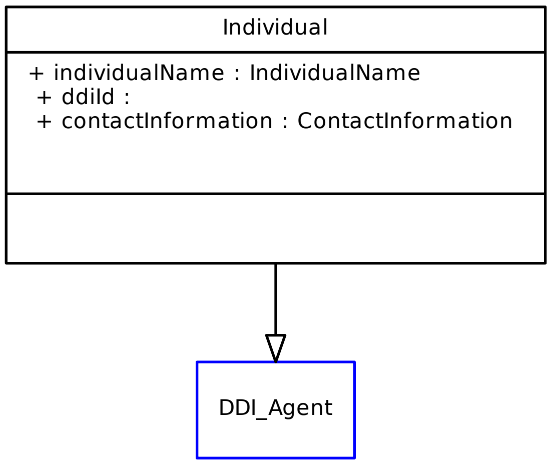
ddiId

The agency identifier of the individual according to the DDI Alliance agent registry.

individualName

The name of an individual broken out into its component parts of prefix, first/given name, middle name, last/family/surname, and suffix.

Graph


Individual
+ individualName : IndividualName
+ ddiId :
+ contactInformation : ContactInformation
DDI_Agent

7.5.14 Machine

Mechanism or computer program used to implement a process.

Extends

Agent

Properties

Name	Type	Cardinality
accessLocation	AccessLocation	0..1
function	ExternalControlledVocabularyEntry	0..n
interface	ExternalControlledVocabularyEntry	0..n
name	Name	0..n
ownerOperatorContact	ContactInformation	0..1
typeOfMachine	ExternalControlledVocabularyEntry	0..1

accessLocation

The locations where the machine can be access

function

The function of the machine

interface

Specified the machine interface. Supports the use of a controlled vocabulary.

name

The name of the machine. A linguistic signifier. Human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. If more than one name is provided provide a context to differentiate usage.

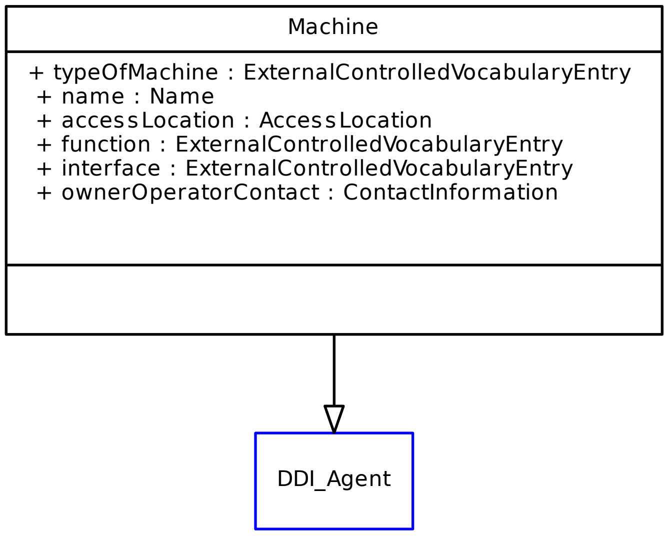
ownerOperatorContact

Contact information for the owner/operator including location specification, address, URL, phone numbers, and other means of communication access. Sets of information can be repeated and date-stamped.

typeOfMachine

The kind of machine used - software, web service, physical machine, from a controlled vocabulary

Graph


Machine
+ typeOfMachine : ExternalControlledVocabularyEntry
+ name : Name
+ accessLocation : AccessLocation
+ function : ExternalControlledVocabularyEntry
+ interface : ExternalControlledVocabularyEntry
+ ownerOperatorContact : ContactInformation
DDI_Agent

7.5.15 Organization

A framework of authority designated to act toward some purpose.

Extends

Agent

Properties

Name	Type	Cardinality
contactInformation	ContactInformation	0..1
ddiId	String	0..n
organizationName	OrganizationName	1..n

contactInformation

Contact information for the organization including location specification, address, URL, phone numbers, and other means of communication access. Sets of information can be repeated and date-stamped.

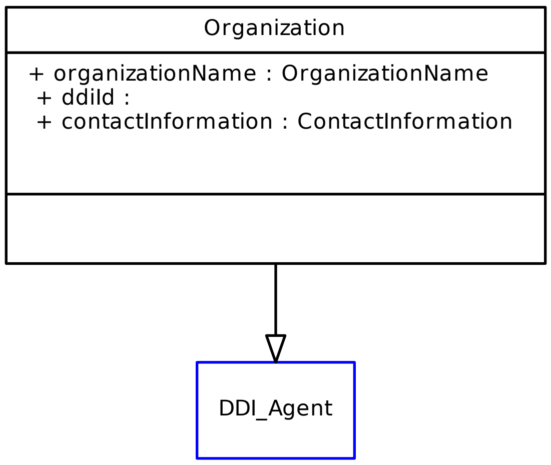
ddiId

The agency identifier of the organization as registered at the DDI Alliance register.

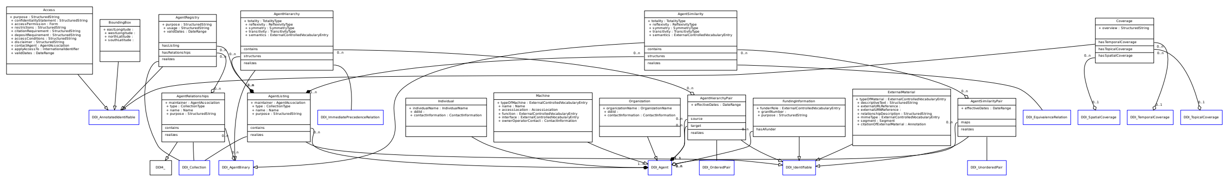
organizationName

Names by which the organization is known.

Graph


Organization
+ organizationName : OrganizationName
+ ddiId :
+ contactInformation : ContactInformation
DDI_Agent

7.5.16 Graph



CHAPTER

EIGHT

GLOSSARY

Abstract class In programming languages, an abstract type is a type in a nominative type system which cannot be instantiated directly. Its only purpose is for other classes to extend (see http://en.wikipedia.org/wiki/Abstract_type).

Binding Data binding is a way to un/serialize objects across programs, languages, and platforms. The structure and the data remain consistent and coherent throughout the journey, and no custom formats or parsing are required (see http://en.wikipedia.org/wiki/XML_data_binding).

Canonical Conforming to a general rule or acceptable procedure (see http://www.merriam-webster.com/dictionary/canonical).

Codebook A codebook is a type of document used for gathering and storing codes. Originally codebooks were often literally books, but today codebook is a byword for the complete record of a series of codes, regardless of physical format. See Wikipedia – http://en.wikipedia.org/wiki/Codebook. DDI Codebook is the development line of the DDI specification that reflects the content of codebooks.

Complex Data Types An extended data type is a user-defined definition of a primitive data type. The following primitive data types can be extended: boolean, integer, real, string, date and container (see http://www.axaptapedia.com/Extended_Data_Types).

Content Capture Content capture refers to the process of harvesting content from other sources. For DDI 4 development, content is being captured using the Drupal content management system to permit machine-processing.

Content Modeler One of the group of people determining the requirements for a Functional View and then identifying the set of objects needed to meet those requirements. In this process they may also need to describe new objects.

Data Modeler These people work with Content Modelers to insure that new objects are consistent with the objects accepted into the approved model. Data Modelers also arrange objects into the final namespace structure of the published model.

DDI The Data Documentation Initiative (DDI) is an effort to create an international standard for describing data from the social, behavioral, and economic sciences. Versions upto DDi-Lifecycl are expressed solely in XML in, the DDI metadata specification now supports the entire research data life cycle. DDI metadata accompanies and enables data conceptualization, collection, processing, distribution, discovery, analysis, repurposing, and archiving.

Drupal Drupal is a free and open-source content management framework written in PHP and distributed under the GNU General Public License. It is also used for knowledge management and business collaboration (see http://en.wikipedia.org/wiki/Drupal).

Enterprise Architect Enterprise Architect is a visual modeling and design tool based on the OMG UML. The platform supports: the design and construction of software systems; modeling business processes; and modeling industry based domains. It is used by businesses and organizations to not only model the architecture of their systems, but to process the implementation of these models across the full application development life-cycle (see http://en.wikipedia.org/wiki/Enterprise_Architect_(Visual_Modeling_Platform)).

Extension Extension is the inheritance of one object's properties and relationships from another object. It also has a semantic relationship – an extending object provides a specialized use of the extended object. Extensions are used within the DDI-published packages to provide relationships between objects as they increase in complexity to meet increasingly complex functionality. Thus, a "simple" version of a questionnaire object might be extended into a more complex object, describing a more complex questionnaire.

Framework The basic structure of something; a set of ideas or facts that provide support for something; a supporting structure; a structural frame (see http://www.merriam-webster.com/dictionary/framework).

Functional View In DDI 4, a functional view identifies a set of objects that are needed to perform a specific task. It primarily consists of a set of references to specific versions of objects. Functional Views are the method used to restrict the portions of the model that are used, and as such they function very much like DDI profiles in DDI 3.*.

GLBPM The Generic Longitudinal Business Process Model is a reference model of the process of longitudinal and repeat cross-sectional data collection, describing the activities undertaken and mapping these to their typical inputs and outputs, which would then be described using DDI Lifecycle. See http://www.ddialliance.org/system/files/GenericLongitudinalBusinessProcessModel.pdf .

GSBPM The Generic Statistical Business Process Model describes and defines the set of business processes needed to produce official statistics. It provides a standard framework and harmonised terminology to help statistical organisations to modernise their statistical production processes, as well as to share methods and components. The GSBPM can also be used for integrating data and metadata standards, as a template for process documentation, for harmonizing statistical computing infrastructures, and to provide a framework for process quality assessment and improvement See http://www1.unece.org/stat/platform/display/GSBPM/Generic+Statistical+Business+Process+Model.

GSIM The Generic Statistical Information Model provides the information object framework supporting all statistical production processes such as those described in the Generic Statistical Business Process Model (GSBPM), giving the information objects agreed names, defining them, specifying their essential properties, and indicating their relationships with other information objects. It does not, however, make assumptions about the standards or technologies used to implement the model. See http://www1.unece.org/stat/platform/display/gsim/Generic+Statistical+Information+Model

Identification In the DDI specifications, each object is uniquely identified.

Instantiate To create an object of a specific class (see http://en.wiktionary.org/wiki/instantiate).

Library The Object Library for DDI 4 encompasses the entire DDI 4.0 model, but without any specific schemas or vocabularies for Functional Views. Objects contain primitives and extended primitives and are the building blocks used to construct the Functional Views. Objects are organized into packages in the Library.

Lifecycle The research data lifecycle is a set of processes that begins at study inception and progresses through data collection, data publication, data archiving, and beyond. DDI created a lifecycle model in 2004 to describe this flow (see http://www.ddialliance.org/system/files/Concept-Model-WD.pdf).

Management package DDI View packages containing library constructs – primitives, extended primitives, objects, and functional views – which are organized thematically.

Metadata Data about data.

Modeling The representation, often mathematical, of a process, concept, or operation of a system, often implemented by a computer program. For DDI Views development, we are using the Unified Modeling Language (UML) to model the specification.

Namespace A grouping of objects that allows for objects with the same name to be differentiated. The full name of an object is a combination of its namespace and its name within the namespace.

Object In the DDI4 model objects are representations of real entities. Objects have properties and relationships. An example might include an "Author" object that might have a Name attribute and an affiliation property linking to an "Organization" object.

Ontology A formal representation of knowledge (see http://en.wikipedia.org/wiki/Ontology_%28information_science%29).

OWL Web Ontology Language," a semantic mark-up language for publishing and sharing ontologies on the World Wide Web" (see http://www.w3.org/TR/owl-ref/).

Package /UML Package A grouping of objects in the Library (see namespace).

Platform A pre-existing environment in which to represent data and metadata (see for example http://en.wikipedia.org/wiki/Computing_platform).

Primitive A basic type is a data type provided by a programming language as a basic building block. Most languages allow more complicated composite types to be recursively constructed starting from basic types.

rdf The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications[1] originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax notations and data serialization formats. It is also used in knowledge management applications.

xml Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable.

INDEX

A

Abstract class, **645**
Anti-reflexive, **42**
Anti-symmetric, **42**
Anti-transitive, **42**

B

Binary relation, **42**
Binding, **645**

C

Canonical, **645**
Class, **41**
Codebook, **645**
Complex Data Types, **645**
Conceivable, **41**
Connected graph, **42**
Content Capture, **645**
Content Modeler, **645**

D

Data Modeler, **645**
DDI, **645**
Drupal, **645**

E

Enterprise Architect, **645**
Extension, **646**

F

Framework, **646**
Functional View, **646**

G

GLBPM, **646**
Graph, **42**
GSBPM, **646**
GSIM, **646**

I

Identification, **646**
Instantiate, **646**
Intransitive relation, **42**

L

Library, **646**
Lifecycle, **646**

M

Management package, **646**
Metadata, **646**
Modeling, **646**

N

Namespace, **646**

O

Object, **41**, **646**
Ontology, **647**
OWL, **647**

P

Package /UML Package, **647**
Partial order, **42**
Path, **42**
Perceivable, **41**
Platform, **647**
Primitive, **647**

R

rdf, **647**
Reflexive relation, **42**
Relation, **41**
Relationship, **41**

S

Symmetric relation, **42**

T

Total relation, **42**
Transitive relation, **42**
Tree, **42**

Tuple, 41

X

xml, 647