

# **DDI-Lifecycle (Model Driven) Specification (Development Draft)**

## **Class Description**

**April 2015**

---

# DDI-Lifecycle (Model Driven) Specification (Development Draft): Class Description: April 2015

DDI-Lifecycle (Model Driven) Specification (Development Draft)  
Copyright © 2015 DDI Alliance. All Rights Reserved

<http://www.ddialliance.org/>

The DDI-Lifecycle (Model Driven) Specification (Development Draft) is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

The DDI-Lifecycle (Model Driven) Specification (Development Draft) is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License [<http://www.gnu.org/licenses/lgpl-3.0.html>] for more details.

---

# Table of Contents

Preface .....	xxii
Development of the Model .....	xxii
Funding .....	xxii
Further information .....	xxii
How this document is organized .....	xxii
How to contact us .....	xxiii
Acknowledgements .....	xxiii
1. Introduction .....	1
.....	1
A. Structure of the DDI-Lifecycle (MD) Model .....	1
B. Library of Classes .....	1
C. Functional Views .....	1
D. Model Constructs and Their Relationships .....	2
E. Extension .....	3
F. Managing the Library .....	4
Provisional Library Organization .....	4
G. Versioning the Library .....	5
H. Example of a Functional View .....	5
2. Primitives (Full Review) .....	7
xs:boolean .....	7
Definition .....	7
xs:Date .....	7
Definition .....	7
xs:Decimal .....	7
Definition .....	7
xs:Integer .....	7
Definition .....	7
xs:String .....	8
Definition .....	8
xs:anyURI .....	8
Definition .....	8
xs:double .....	8
Definition .....	8
xs:float .....	8
Definition .....	8
xs:NMTOKEN .....	9
Definition .....	9
xs:NMTOKENS .....	9
Definition .....	9
xs:dateTime .....	9
Definition .....	9
xs:language .....	10
Definition .....	10
xhtml:BlkNoForm.mix .....	10
Definition .....	10
3. ComplexDataTypes (Full Review) .....	11
.....	11
AccessLocation .....	13
Definition .....	13
Properties .....	13
Address .....	13

Based on DDI 3.2 .....	13
Definition .....	13
Example .....	14
Properties .....	14
AgentAssociation .....	16
Based on DDI 3.2 .....	16
Definition .....	16
Properties .....	16
Relationships .....	16
AgentId .....	16
Definition .....	16
Example .....	17
Properties .....	17
AnnotationDate .....	17
Extends .....	17
.....	17
Definition .....	17
Properties .....	17
AudioSegment .....	18
Based on DDI 3.2 .....	18
Definition .....	18
Properties .....	18
BasedOnObject .....	18
Based on DDI 3.2 .....	18
Definition .....	18
Properties .....	19
Relationships .....	19
BibliographicName .....	19
Extends .....	19
Based on DDI 3.2 .....	19
Definition .....	19
Properties .....	20
CharacterOffset .....	20
Based on DDI 3.2 .....	20
Definition .....	20
Properties .....	20
CodeValueType .....	20
Based on DDI 3.2 .....	20
Definition .....	20
Properties .....	21
CommandCode .....	22
Based on DDI 3.2 .....	22
Definition .....	22
Properties .....	22
Relationships .....	22
ConditionalText .....	23
Extends .....	23
Based on DDI 3.2 .....	23
Definition .....	23
Properties .....	23
ContactInformation .....	24
Based on DDI 3.2 .....	24
Definition .....	24
Properties .....	24

Content .....	25
Based on DDI 3.2 .....	25
Definition .....	25
Properties .....	25
ContentDateOffset .....	26
Extends .....	26
Based on DDI 3.2 .....	26
Definition .....	26
Properties .....	26
CorrespondenceType .....	27
Based on DDI 3.2 .....	27
Definition .....	27
Properties .....	27
CountryCode .....	27
Abstract .....	27
Based on DDI 3.2 .....	27
Definition .....	28
Properties .....	28
Date .....	28
Based on DDI 3.2 .....	28
Definition .....	28
Example .....	28
Explanatory Notes .....	28
Properties .....	28
DynamicText .....	30
Based on DDI 3.2 .....	30
Definition .....	30
Properties .....	30
ElectronicMessageSystem .....	30
Based on DDI 3.2 .....	30
Definition .....	30
Example .....	31
Properties .....	31
Email .....	31
Based on DDI 3.2 .....	31
Definition .....	31
Example .....	32
Properties .....	32
Form .....	32
Based on DDI 3.2 .....	32
Definition .....	33
Properties .....	33
HistoricalDate .....	33
Based on DDI 3.2 .....	33
Definition .....	33
Properties .....	33
Image .....	34
Based on DDI 3.2 .....	34
Definition .....	34
Properties .....	34
ImageArea .....	35
Based on DDI 3.2 .....	35
Definition .....	35
Properties .....	35

IndividualName .....	35
Based on DDI 3.2 .....	35
Definition .....	35
Properties .....	36
InternationalCodeValueType .....	38
Extends .....	38
Based on DDI 3.2 .....	38
Definition .....	38
Properties .....	38
InternationalIdentifier .....	39
Based on DDI 3.2 .....	39
Definition .....	39
Properties .....	39
InternationalString .....	40
Based on DDI 3.2 .....	40
Definition .....	40
Properties .....	40
Label .....	40
Extends .....	40
Based on DDI 3.2 .....	40
.....	40
Definition .....	40
Properties .....	41
LineParameter .....	41
Based on DDI 3.2 .....	41
Definition .....	41
Properties .....	41
LiteralText .....	42
Extends .....	42
Based on DDI 3.2 .....	42
Definition .....	42
Properties .....	42
LocalId .....	43
Definition .....	43
Example .....	43
Properties .....	43
LocationName .....	43
Extends .....	43
Based on DDI 3.2 .....	43
Definition .....	44
Properties .....	44
Name .....	44
Extends .....	44
Based on DDI 3.2 .....	44
Definition .....	44
Properties .....	44
OrganizationName .....	45
Extends .....	45
Based on DDI 3.2 .....	45
Definition .....	45
Properties .....	45
PairedCodeValueType .....	46
Extends .....	46
Definition .....	46

Example .....	46
Properties .....	46
Point .....	46
Based on DDI 3.2 .....	46
Definition .....	46
Properties .....	46
Polygon .....	47
Based on DDI 3.2 .....	47
Definition .....	47
Properties .....	47
PrivateImage .....	48
Extends .....	48
Based on DDI 3.2 .....	48
Definition .....	48
Properties .....	48
ProprietaryInfo .....	48
Based on DDI 3.2 .....	48
Definition .....	48
Properties .....	49
Range .....	49
Based on DDI 3.2 .....	49
Definition .....	49
Properties .....	49
RangeValue .....	49
Extends .....	49
Based on DDI 3.2 .....	50
Definition .....	50
Properties .....	50
ReferenceDate .....	50
Extends .....	50
Based on DDI 3.2 .....	50
Definition .....	50
Properties .....	50
Relationship .....	51
Based on DDI 3.2 .....	51
Definition .....	51
Properties .....	51
Relationships .....	51
ResourceIdentifier .....	51
Extends .....	51
Definition .....	51
Properties .....	52
Segment .....	52
Based on DDI 3.2 .....	52
Definition .....	52
Properties .....	52
Software .....	53
Based on DDI 3.2 .....	53
Definition .....	53
Properties .....	53
SpatialCoordinate .....	54
Based on DDI 3.2 .....	54
Definition .....	54
Properties .....	54

SpecificSequence .....	55
Definition .....	55
Properties .....	55
StandardKeyValuePair .....	55
Based on DDI 3.2 .....	55
Definition .....	55
Properties .....	55
Statistic .....	56
Based on DDI 3.2 .....	56
Definition .....	56
Properties .....	56
String .....	56
Based on DDI 3.2 .....	56
Definition .....	56
Properties .....	57
StructuredString .....	58
Based on DDI 3.2 .....	58
Definition .....	58
Properties .....	58
Telephone .....	58
Based on DDI 3.2 .....	58
Definition .....	58
Example .....	58
Properties .....	58
Text .....	59
Extends .....	59
Based on DDI 3.2 .....	59
Definition .....	59
Properties .....	59
TextContent .....	60
Abstract .....	60
Based on DDI 3.2 .....	60
Definition .....	60
Properties .....	60
TextualSegment .....	60
Based on DDI 3.2 .....	60
Definition .....	60
Properties .....	60
URI .....	61
Based on DDI 3.2 .....	61
Definition .....	61
Properties .....	61
URL .....	61
Based on DDI 3.2 .....	61
Definition .....	61
Properties .....	61
Value .....	62
Based on DDI 3.2 .....	62
Definition .....	62
Properties .....	62
VideoSegment .....	63
Based on DDI 3.2 .....	63
Definition .....	63
Properties .....	63

XMLPrefixMap .....	63
Based on DDI 3.2 .....	63
Definition .....	64
Properties .....	64
4. Agents (Full Review) .....	65
.....	65
Agent .....	65
Extends .....	65
Abstract .....	66
.....	66
Definition .....	66
Example .....	66
Explanatory Notes .....	66
Properties .....	66
AuthorizationSource .....	66
Extends .....	66
Based on DDI 3.2 .....	66
Definition .....	67
Example .....	67
Properties .....	67
Relationships .....	67
Individual .....	68
Extends .....	68
Definition .....	68
Example .....	68
Properties .....	68
Machine .....	69
Extends .....	69
Definition .....	69
Example .....	69
Properties .....	69
Organization .....	70
Extends .....	70
.....	70
Definition .....	70
Example .....	70
Explanatory Notes .....	70
Properties .....	70
Relation .....	71
Extends .....	71
Based on DDI 3.2 .....	71
Definition .....	71
Example .....	71
Properties .....	71
Relationships .....	72
5. Collections (Full Review) .....	73
.....	73
Collection .....	74
Extends .....	74
Abstract .....	74
Definition .....	74
Example .....	74
Properties .....	74
Relationships .....	74

CollectionCorrespondence .....	75
Extends .....	75
Abstract .....	75
Definition .....	75
Properties .....	75
Relationships .....	75
Member .....	76
Extends .....	76
Abstract .....	76
Definition .....	76
Properties .....	76
MemberCorrespondence .....	77
Extends .....	77
Abstract .....	77
Definition .....	77
Explanatory Notes .....	77
Properties .....	77
Relationships .....	78
OrderRelation .....	78
Extends .....	78
Abstract .....	78
Definition .....	78
Properties .....	78
Relationships .....	79
OrderedCollectionCorrespondence .....	80
Extends .....	80
Corresponds to GSIM .....	80
Definition .....	80
Relationships .....	80
OrderedMemberCorrespondence .....	81
Extends .....	81
Abstract .....	81
Definition .....	81
Example .....	81
Relationships .....	81
6. Conceptual (Full Review) .....	83
.....	83
Category .....	84
Extends .....	84
Based on DDI 3.2 .....	84
Definition .....	84
Concept .....	84
Extends .....	84
Based on DDI 3.2 .....	84
.....	84
Corresponds to GSIM .....	84
Definition .....	84
ConceptParentChild .....	84
Extends .....	84
Definition .....	84
Relationships .....	85
ConceptPartWhole .....	85
Extends .....	85
Definition .....	85

Relationships .....	85
ConceptSystem .....	86
Extends .....	86
Abstract .....	86
Based on DDI 3.2 .....	86
Corresponds to GSIM .....	86
Definition .....	86
Example .....	86
Relationships .....	86
ConceptSystemCorrespondence .....	87
Extends .....	87
Definition .....	87
Relationships .....	87
ConceptualDomain .....	88
Extends .....	88
Corresponds to GSIM .....	88
Definition .....	88
Explanatory Notes .....	88
Properties .....	88
ConceptualVariable .....	89
Extends .....	89
Based on DDI 3.2 .....	89
Corresponds to GSIM .....	89
Definition .....	89
Example .....	89
Relationships .....	89
DescribedConceptualDomain .....	90
Extends .....	90
Corresponds to GSIM .....	90
Definition .....	90
Example .....	90
Explanatory Notes .....	90
EnumeratedConceptualDomain .....	90
Extends .....	90
Corresponds to GSIM .....	90
Definition .....	90
Example .....	90
Explanatory Notes .....	90
Relationships .....	90
InstanceVariable .....	91
Extends .....	91
Corresponds to GSIM .....	91
Definition .....	91
Example .....	91
Explanatory Notes .....	91
Properties .....	91
Relationships .....	91
Population .....	92
Extends .....	92
Corresponds to GSIM .....	92
Definition .....	92
Example .....	92
Relationships .....	93
RepresentedVariable .....	93

Extends .....	93
Definition .....	93
Example .....	93
Relationships .....	93
SimilarConcept .....	94
Extends .....	94
Based on DDI 3.2 .....	94
Definition .....	94
Relationships .....	94
Unit .....	95
Extends .....	95
Corresponds to GSIM .....	95
Definition .....	95
Example .....	95
Properties .....	95
UnitType .....	95
Extends .....	95
Corresponds to GSIM .....	96
Definition .....	96
Explanatory Notes .....	96
Relationships .....	96
Universe .....	96
Extends .....	96
Based on DDI 3.2 .....	96
Corresponds to GSIM .....	96
Definition .....	96
Example .....	97
Explanatory Notes .....	97
Properties .....	97
Relationships .....	97
7. CoreProcess (Full Review) .....	98
.....	98
Act .....	98
Extends .....	98
Abstract .....	98
Definition .....	99
Example .....	99
Binding .....	99
Based on DDI 3.2 .....	99
Definition .....	99
Properties .....	99
ContainsTemporalRelation .....	100
Extends .....	100
Definition .....	100
Relationships .....	100
ControlConstruct .....	100
Extends .....	100
Abstract .....	100
Based on DDI 3.2 .....	100
Corresponds to GSIM .....	100
Definition .....	101
Example .....	101
EqualTemporalRelation .....	101
Extends .....	101

Definition .....	101
Relationships .....	101
FinishesTemporalRelation .....	101
Extends .....	101
Definition .....	102
Relationships .....	102
IfThenElse .....	102
Extends .....	102
Based on DDI 3.2 .....	102
Definition .....	102
Example .....	102
Explanatory Notes .....	102
Properties .....	103
Relationships .....	103
Input .....	103
Extends .....	103
Definition .....	103
Example .....	104
Properties .....	104
Loop .....	104
Extends .....	104
Based on DDI 3.2 .....	104
Definition .....	104
Explanatory Notes .....	104
Properties .....	104
Relationships .....	105
MeetsTemporalRelation .....	105
Extends .....	105
Definition .....	105
Relationships .....	105
Output .....	106
Extends .....	106
Definition .....	106
Example .....	106
Explanatory Notes .....	106
OverlapsTemporalRelation .....	106
Extends .....	106
Definition .....	106
Relationships .....	106
PredecessorTemporalRelation .....	107
Extends .....	107
Definition .....	107
Relationships .....	107
ProcessStep .....	107
Extends .....	107
Abstract .....	108
Corresponds to GSIM .....	108
Definition .....	108
Example .....	108
Explanatory Notes .....	108
Properties .....	108
Relationships .....	109
RepeatUntil .....	109
Extends .....	109

Based on DDI 3.2 .....	109
Definition .....	110
Properties .....	110
Relationships .....	110
RepeatWhile .....	110
Extends .....	110
Based on DDI 3.2 .....	110
Definition .....	110
Properties .....	110
Relationships .....	111
Sequence .....	111
Extends .....	111
Based on DDI 3.2 .....	111
Definition .....	111
Properties .....	111
Relationships .....	112
Service .....	112
Extends .....	112
Corresponds to GSIM .....	112
Definition .....	112
Properties .....	112
Relationships .....	112
StartsTemporalRelation .....	113
Extends .....	113
Definition .....	113
Relationships .....	113
TemporalRelation .....	113
Definition .....	113
Explanatory Notes .....	113
Relationships .....	114
8. Discovery (Full Review) .....	115
.....	115
Access .....	116
Extends .....	116
Based on DDI 3.2 .....	116
Definition .....	116
Properties .....	116
Annotation .....	118
Extends .....	118
Based on DDI 3.2 .....	118
Definition .....	118
Properties .....	118
BoundingBox .....	121
Extends .....	121
Definition .....	121
Properties .....	121
Coverage .....	122
Extends .....	122
Based on DDI 3.2 .....	122
Definition .....	122
Properties .....	122
Relationships .....	122
SpatialCoverage .....	123
Extends .....	123

.....	123
Definition .....	123
Example .....	123
Properties .....	123
Relationships .....	124
TemporalCoverage .....	124
Extends .....	124
Definition .....	124
Properties .....	124
TopicalCoverage .....	125
Extends .....	125
Based on DDI 3.2 .....	125
.....	125
Definition .....	125
Properties .....	125
9. Identification (Full Review) .....	126
.....	126
AnnotatedIdentifiable .....	126
Extends .....	126
Abstract .....	126
Definition .....	126
Properties .....	127
Relationships .....	128
Identifiable .....	128
Abstract .....	128
Definition .....	128
Properties .....	128
10. Representations (Full Review) .....	130
.....	130
CategorySet .....	130
Extends .....	130
Based on DDI 3.2 .....	131
Corresponds to GSIM .....	131
Definition .....	131
Example .....	131
Explanatory Notes .....	131
Relationships .....	131
ClassificationFamily .....	131
Extends .....	131
Corresponds to GSIM .....	131
Definition .....	131
Explanatory Notes .....	132
Relationships .....	132
ClassificationIndex .....	132
Extends .....	132
Corresponds to GSIM .....	132
Definition .....	132
Explanatory Notes .....	132
Properties .....	133
Relationships .....	134
ClassificationIndexEntry .....	134
Extends .....	134
Corresponds to GSIM .....	134
Definition .....	134

Explanatory Notes .....	135
Properties .....	135
ClassificationItem .....	135
Extends .....	135
Corresponds to GSIM .....	135
Definition .....	135
Explanatory Notes .....	136
Properties .....	136
Relationships .....	137
ClassificationSeries .....	138
Extends .....	138
Corresponds to GSIM .....	138
Definition .....	138
Properties .....	138
Relationships .....	139
Code .....	139
Extends .....	139
Based on DDI 3.2 .....	139
Corresponds to GSIM .....	139
Definition .....	139
Properties .....	139
Relationships .....	140
CodeList .....	140
Extends .....	140
Based on DDI 3.2 .....	140
Corresponds to GSIM .....	140
Definition .....	140
Relationships .....	140
CorrespondenceTable .....	141
Extends .....	141
Corresponds to GSIM .....	141
Definition .....	141
Explanatory Notes .....	141
Properties .....	141
Relationships .....	142
DataType .....	143
Extends .....	143
Definition .....	144
Example .....	144
Properties .....	144
DescribedSentinelValueDomain .....	144
Extends .....	144
Definition .....	144
Relationships .....	145
DescribedSubstantiveValueDomain .....	145
Extends .....	145
Definition .....	145
Relationships .....	145
DescribedValueDomain .....	145
Extends .....	145
Corresponds to GSIM .....	145
Definition .....	145
Example .....	146
Relationships .....	146

Designation .....	146
Extends .....	146
Abstract .....	146
Corresponds to GSIM .....	146
Definition .....	146
Example .....	146
Explanatory Notes .....	146
Properties .....	146
Relationships .....	147
EnumeratedSentinelValueDomain .....	147
Extends .....	147
Definition .....	147
Relationships .....	147
EnumeratedSubstantiveValueDomain .....	148
Extends .....	148
Definition .....	148
Relationships .....	148
EnumeratedValueDomain .....	148
Extends .....	148
Corresponds to GSIM .....	148
Definition .....	148
Example .....	148
Relationships .....	148
IndexOrder .....	149
Extends .....	149
Definition .....	149
Relationships .....	149
Level .....	149
Extends .....	149
Based on DDI 3.2 .....	150
Corresponds to GSIM .....	150
Definition .....	150
Explanatory Notes .....	150
Relationships .....	150
LevelParentChild .....	150
Extends .....	150
Definition .....	151
Relationships .....	151
Map .....	151
Extends .....	151
Corresponds to GSIM .....	151
Definition .....	151
Properties .....	151
Relationships .....	152
Node .....	152
Extends .....	152
Abstract .....	152
Corresponds to GSIM .....	152
Definition .....	153
Example .....	153
Explanatory Notes .....	153
Relationships .....	153
NodeParentChild .....	153
Extends .....	153

Definition .....	153
Relationships .....	154
NodePartWhole .....	154
Extends .....	154
Definition .....	154
Relationships .....	154
NodeSet .....	155
Extends .....	155
Abstract .....	155
Corresponds to GSIM .....	155
Definition .....	155
Example .....	155
Relationships .....	155
NodeSetParentChild .....	157
Extends .....	157
Definition .....	157
Relationships .....	157
NodeSetPartWhole .....	158
Extends .....	158
Definition .....	158
Relationships .....	158
Sign .....	159
Extends .....	159
Definition .....	159
Example .....	159
Explanatory Notes .....	159
Properties .....	159
Relationships .....	159
StatisticalClassification .....	160
Extends .....	160
Corresponds to GSIM .....	160
Definition .....	160
Explanatory Notes .....	160
Properties .....	160
Relationships .....	162
ValueDomain .....	164
Extends .....	164
Abstract .....	164
Based on DDI 3.2 .....	164
Corresponds to GSIM .....	164
Definition .....	164
Example .....	164
Explanatory Notes .....	164
Properties .....	164
Vocabulary .....	165
Extends .....	165
Based on DDI 3.2 .....	165
Definition .....	165
Example .....	165
Properties .....	165
11. Utility (Full Review) .....	167
.....	167
Note .....	167
Extends .....	167

Based on DDI 3.2 .....	167
Definition .....	167
Properties .....	167
12. DataCapture (Partial Review) .....	169
.....	169
Capture .....	170
Extends .....	170
Abstract .....	170
Definition .....	170
Example .....	170
Relationships .....	170
ExternalAid .....	171
Extends .....	171
Based on DDI 3.2 .....	171
Definition .....	171
Example .....	171
Explanatory Notes .....	171
Properties .....	171
Instruction .....	171
Extends .....	171
Based on DDI 3.2 .....	171
Definition .....	172
Example .....	172
Properties .....	172
InstrumentComponent .....	172
Extends .....	172
Abstract .....	172
Definition .....	172
Explanatory Notes .....	172
Relationships .....	173
ResponseDomain .....	173
Extends .....	173
Definition .....	173
Example .....	173
Explanatory Notes .....	173
Relationships .....	173
13. Processing (Partial Review) .....	175
.....	175
Command .....	176
Extends .....	176
Based on DDI 3.2 .....	176
Definition .....	176
Properties .....	176
CommandFile .....	176
Extends .....	176
Based on DDI 3.2 .....	176
Definition .....	176
Properties .....	177
Parameter .....	177
Extends .....	177
Based on DDI 3.2 .....	177
Definition .....	177
Properties .....	177
Relationships .....	178

StructuredCommand .....	178
Extends .....	178
Based on DDI 3.2 .....	178
Definition .....	178
14. Agents Functional View .....	179
.....	179
15. Discovery Functional View .....	180
.....	180

---

## List of Tables

3.1. RDF Mapping (AnnotationDate) .....	17
3.2. RDF Mapping (Label) .....	40
4.1. RDF Mapping (Agent) .....	66
4.2. RDF Mapping (Organization) .....	70
6.1. RDF Mapping (Concept) .....	84
8.1. RDF Mapping (SpatialCoverage) .....	123
8.2. RDF Mapping (TopicalCoverage) .....	125

---

# Preface

The intent of the Model-Based DDI 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 the General Statistical Information Model.

## 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

## 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.

## Further information

Further information on this is available at the DDI Moving Forward Project site [<https://ddi-alliance.atlassian.net/wiki/pages/viewpage.action?pageId=491703>].

## How this document is organized

The introduction provides a high level view of the structure of the DDI-Lifecycle§ (MD) 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, or GSIM or other standard.

- 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
- Mapping to Other standards

## How to contact us

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](mailto:wlt@umn.edu).

This document and associated files were generated from Build-2015-04-08 available at <http://lion.ddialliance.org/builds> using Docbook version 5.0 (docbook-xsl-1.76.1)

## 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
- Arofan Gregory
- Barry Radler
- Brigitte Mathiak
- Bryan Fitzpatrick
- Chris Seymour
- Chuck Humphrey
- Dan Smith
- Daniel Gillman
- Denis Grofils
- Dennis Wegener
- Elizabeth Hostetter
- Flavio Rizzolo
- Franck Cotton
- Guillaume Duffes
- Helen Toole
- Herve L'Hours
- Hilde Orten
- Ingo Barkow
- Jannik Jensen
- 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 Jensby
- Mary Vardigan
- Merja Karjalainen
- Michael Witt
- Oleg Volguine
- Oliver Hopt
- Olof Olsson
- Ornulf Risnes
- Richard Cyganiak
- Ron Nakao
- Sam Hume
- Sanda Ionescu
- Simon Wall
- Sophia Kuan
- Steve McEachern
- Stuart Weibel
- Therese Lalor
- Thomas Bosch
- Wendy Thomas
- William Block
- Wolfgang Zenk-Moltgen

---

# Chapter 1. Introduction

The intent of the Model-Based DDI 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 (GWIM).

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.

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'

## A. Structure of the DDI-Lifecycle (MD) 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.

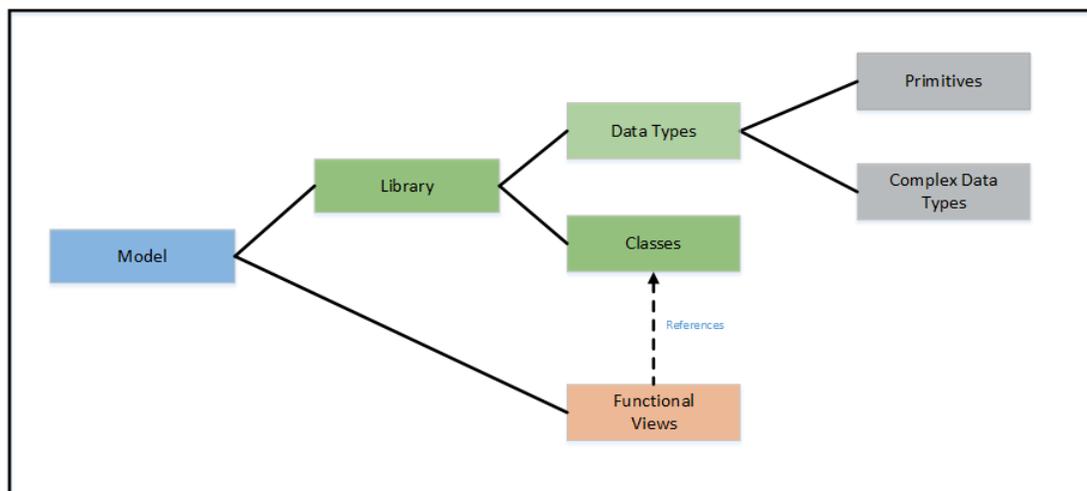


Figure 1. DDI-Lifecycle (MD) Model and its components

## B. Library of Classes

The Library of Classes encompasses the entire DDI-Lifecycle (MD) 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.

## C. 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

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

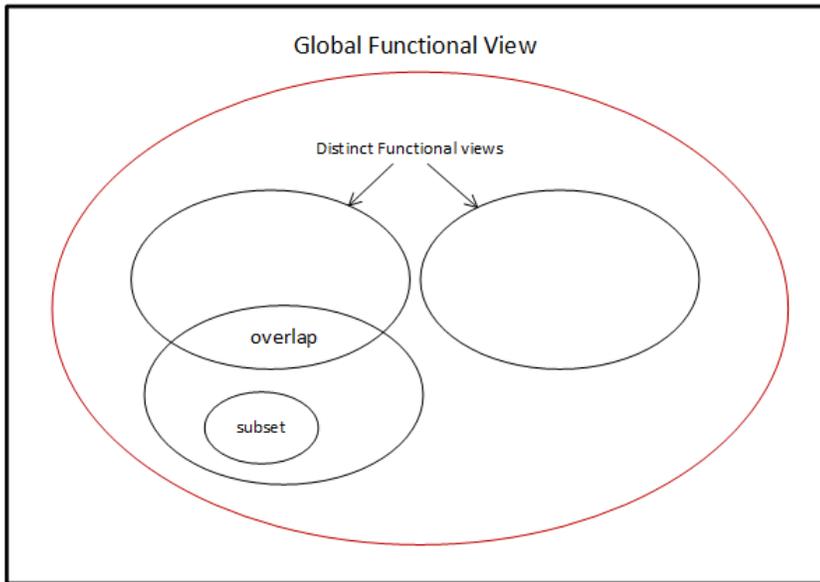


Figure 2. Interoperability of Functional Views

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.

## D. 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.

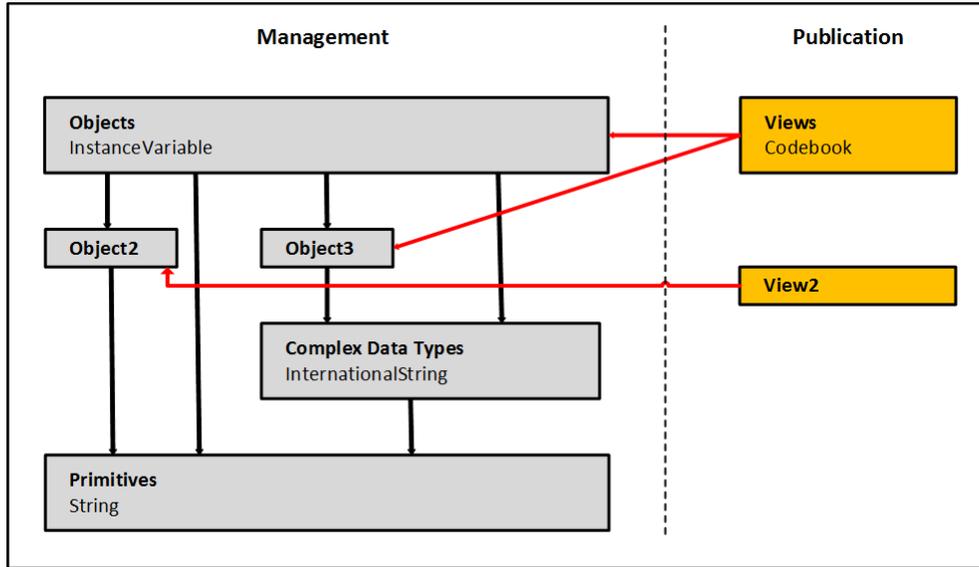


Figure 3. DDI-Lifecycle (MD) Organization of the Model

## E. 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.

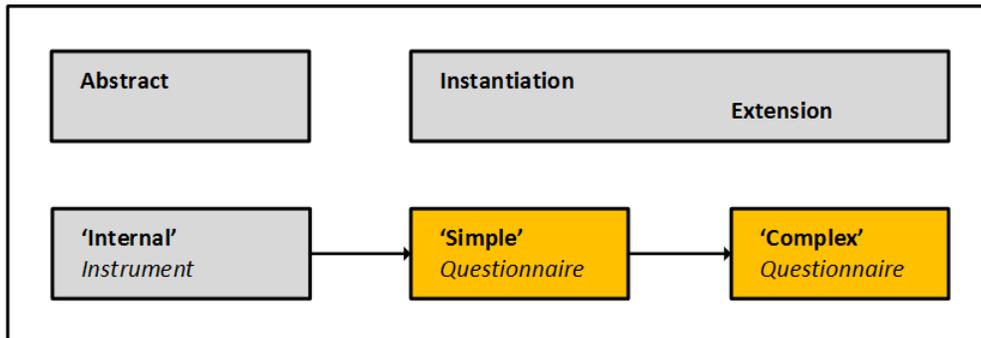


Figure 4. Extensions in DDI-Lifecycle (MD)

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.

## F. 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

## G. 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.

## H. 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.

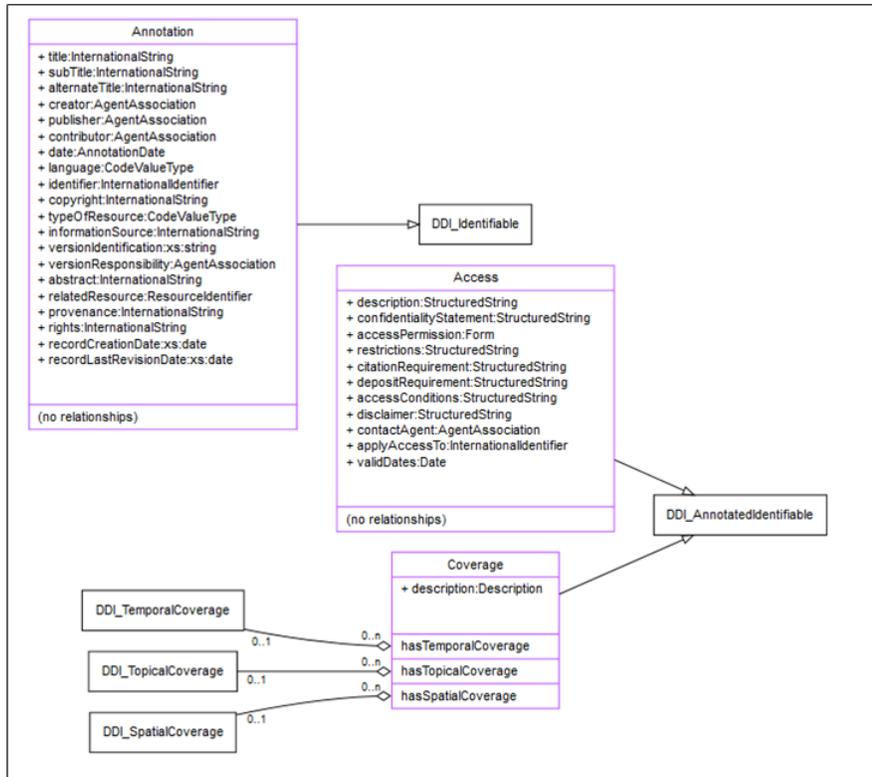


Figure 5. Example Functional View

---

# Chapter 2. Primitives (Full Review)

## xs:boolean

### Definition

A boolean has the value space required to support the mathematical concept of binary-valued logic: {true, false}. Further details at w3.org [<http://www.w3.org/TR/xmlschema-2/#boolean>]

## xs>Date

### Definition

The value space of date consists of top-open intervals of exactly one day in length on the timelines of dateTime, beginning on the beginning moment of each day (in each timezone), i.e. '00:00:00', up to but not including '24:00:00' (which is identical with '00:00:00' of the next day). For nontimezoned values, the top-open intervals disjointly cover the nontimezoned timeline, one per day. For timezoned values, the intervals begin at every minute and therefore overlap. A "date object" is an object with year, month, and day properties just like those of dateTime objects, plus an optional timezone-valued timezone property. (As with values of dateTime timezones are a special case of durations.) Just as a dateTime object corresponds to a point on one of the timelines, a date object corresponds to an interval on one of the two timelines as just described. Timezoned date values track the starting moment of their day, as determined by their timezone; said timezone is generally recoverable for canonical representations. [Definition:] The recoverable timezone is that duration which is the result of subtracting the first moment (or any moment) of the timezoned date from the first moment (or the corresponding moment) UTC on the same date. ·recoverable timezone-s are always durations between '+12:00' and '-11:59'. This "timezone normalization" (which follows automatically from the definition of the date ·value space·) is explained more in Lexical representation (§3.2.9.1). For example: the first moment of 2002-10-10+13:00 is 2002-10-10T00:00:00+13, which is 2002-10-09T11:00:00Z, which is also the first moment of 2002-10-09-11:00. Therefore 2002-10-10+13:00 is 2002-10-09-11:00; they are the same interval. Further details at w3.org [<http://www.w3.org/TR/xmlschema-2/#date>]

## xs:Decimal

### Definition

Decimal represents a subset of the real numbers, which can be represented by decimal numerals. The ·value space· of decimal is the set of numbers that can be obtained by multiplying an integer by a non-positive power of ten, i.e., expressible as  $i \times 10^{-n}$  where  $i$  and  $n$  are integers and  $n \geq 0$ . Precision is not reflected in this value space; the number 2.0 is not distinct from the number 2.00. The ·order-relation· on decimal is the order relation on real numbers, restricted to this subset. Further details at w3.org [<http://www.w3.org/TR/xmlschema-2/#decimal>]

## xs:Integer

### Definition

Integer is derived from decimal by fixing the value of fraction Digits to be 0 and disallowing the trailing decimal point. This results in the standard mathematical concept of the integer numbers. The value space

of integer is the infinite set  $\{\dots, -2, -1, 0, 1, 2, \dots\}$ . The base type of integer is decimal. Further details at w3.org [<http://www.w3.org/TR/xmlschema-2/#integer>]

## xs:String

### Definition

The string datatype represents character strings in XML. The *value space* of string is the set of finite-length sequences of characters (as defined in [XML 1.0 (Second Edition)]) that *match* the Char production from [XML 1.0 (Second Edition)]. A character is an atomic unit of communication; it is not further specified except to note that every character has a corresponding Universal Character Set code point, which is an integer. Further details at w3.org [<http://www.w3.org/TR/xmlschema-2/#string>]

## xs:anyURI

### Definition

anyURI represents a Uniform Resource Identifier Reference (URI). An anyURI value can be absolute or relative, and may have an optional fragment identifier (i.e., it may be a URI Reference). This type should be used to specify the intention that the value fulfills the role of a URI as defined by [RFC 2396], as amended by [RFC 2732]. The mapping from anyURI values to URIs is as defined by the URI reference escaping procedure defined in Section 5.4 Locator Attribute of [XML Linking Language] (see also Section 8 Character Encoding in URI References of [Character Model]). This means that a wide range of internationalized resource identifiers can be specified when an anyURI is called for, and still be understood as URIs per [RFC 2396], as amended by [RFC 2732], where appropriate to identify resources. Further details at w3.org [<http://www.w3.org/TR/xmlschema-2/#anyURI>]

## xs:double

### Definition

The double datatype is patterned after the IEEE double-precision 64-bit floating point type [IEEE 754-1985]. The basic *value space* of double consists of the values  $m \times 2^e$ , where  $m$  is an integer whose absolute value is less than  $2^{53}$ , and  $e$  is an integer between -1075 and 970, inclusive. In addition to the basic *value space* described above, the *value space* of double also contains the following three special values: positive and negative infinity and not-a-number (NaN). The order-relation on double is:  $x < y$  iff  $y - x$  is positive for  $x$  and  $y$  in the value space. Positive infinity is greater than all other non-NaN values. NaN equals itself but is *incomparable* with (neither greater than nor less than) any other value in the *value space*. A literal in the lexical space representing a decimal number  $d$  maps to the normalized value in the *value space* of double that is closest to  $d$ ; if  $d$  is exactly halfway between two such values then the even value is chosen. This is the best approximation of  $d$  ([Clinger, WD (1990)], [Gay, DM (1990)]), which is more accurate than the mapping required by [IEEE 754-1985]. Further details at w3.org [<http://www.w3.org/TR/xmlschema-2/#double>]

## xs:float

### Definition

float is patterned after the IEEE single-precision 32-bit floating point type [IEEE 754-1985]. The basic *value space* of float consists of the values  $m \times 2^e$ , where  $m$  is an integer whose absolute value is less

than  $2^{24}$ , and  $e$  is an integer between -149 and 104, inclusive. In addition to the basic `·value space·` described above, the `·value space·` of float also contains the following three special values: positive and negative infinity and not-a-number (NaN). The order-relation on float is:  $x < y$  iff  $y - x$  is positive for  $x$  and  $y$  in the value space. Positive infinity is greater than all other non-NaN values. NaN equals itself but is `·incomparable·` with (neither greater than nor less than) any other value in the `·value space·`. Further details at w3.org [<http://www.w3.org/TR/xmlschema-2/#float>]

## xs:NMTOKEN

### Definition

NMTOKEN represents the NMTOKEN attribute type from [XML 1.0 (Second Edition)]. The `·value space·` of NMTOKEN is the set of tokens that `·match·` the Nmtoken production in [XML 1.0 (Second Edition)]. The `·lexical space·` of NMTOKEN is the set of strings that `·match·` the Nmtoken production in [XML 1.0 (Second Edition)]. The `·base type·` of NMTOKEN is token. Further details at w3.org [<http://www.w3.org/TR/xmlschema-2/#NMTOKEN>]

## xs:NMTOKENS

### Definition

NMTOKENS represents the NMTOKENS attribute type from [XML 1.0 (Second Edition)]. The `·value space·` of NMTOKENS is the set of finite, non-zero-length sequences of `·NMTOKEN·`s. The `·lexical space·` of NMTOKENS is the set of space-separated lists of tokens, of which each token is in the `·lexical space·` of NMTOKEN. The `·itemType·` of NMTOKENS is NMTOKEN. Further details at w3.org [<http://www.w3.org/TR/xmlschema-2/#NMTOKENS>]

## xs:dateTime

### Definition

dateTime values may be viewed as objects with integer-valued year, month, day, hour and minute properties, a decimal-valued second property, and a boolean timezoned property. Each such object also has one decimal-valued method or computed property, `timeOnTimeline`, whose value is always a decimal number; the values are dimensioned in seconds, the integer 0 is 0001-01-01T00:00:00 and the value of `timeOnTimeline` for other dateTime values is computed using the Gregorian algorithm as modified for leap-seconds. The `timeOnTimeline` values form two related "timelines", one for timezoned values and one for non-timezoned values. Each timeline is a copy of the `·value space·` of decimal, with integers given units of seconds. The value space of dateTime is closely related to the dates and times described in ISO 8601. For clarity, the text above specifies a particular origin point for the timeline. It should be noted, however, that schema processors need not expose the `timeOnTimeline` value to schema users, and there is no requirement that a timeline-based implementation use the particular origin described here in its internal representation. Other interpretations of the `·value space·` which lead to the same results (i.e., are isomorphic) are of course acceptable. All timezoned times are Coordinated Universal Time (UTC, sometimes called "Greenwich Mean Time"). Other timezones indicated in lexical representations are converted to UTC during conversion of literals to values. "Local" or untimezoned times are presumed to be the time in the timezone of some unspecified locality as prescribed by the appropriate legal authority; currently there are no legally prescribed timezones which are durations whose magnitude is greater than 14 hours. The value of each numeric-valued property (other than `timeOnTimeline`) is limited to the maximum value within the interval determined by the next-higher property. For example, the day value can never be

32, and cannot even be 29 for month 02 and year 2002 (February 2002). Further details at w3.org [<http://www.w3.org/TR/xmlschema-2/#dateTime>]

## xs:language

### Definition

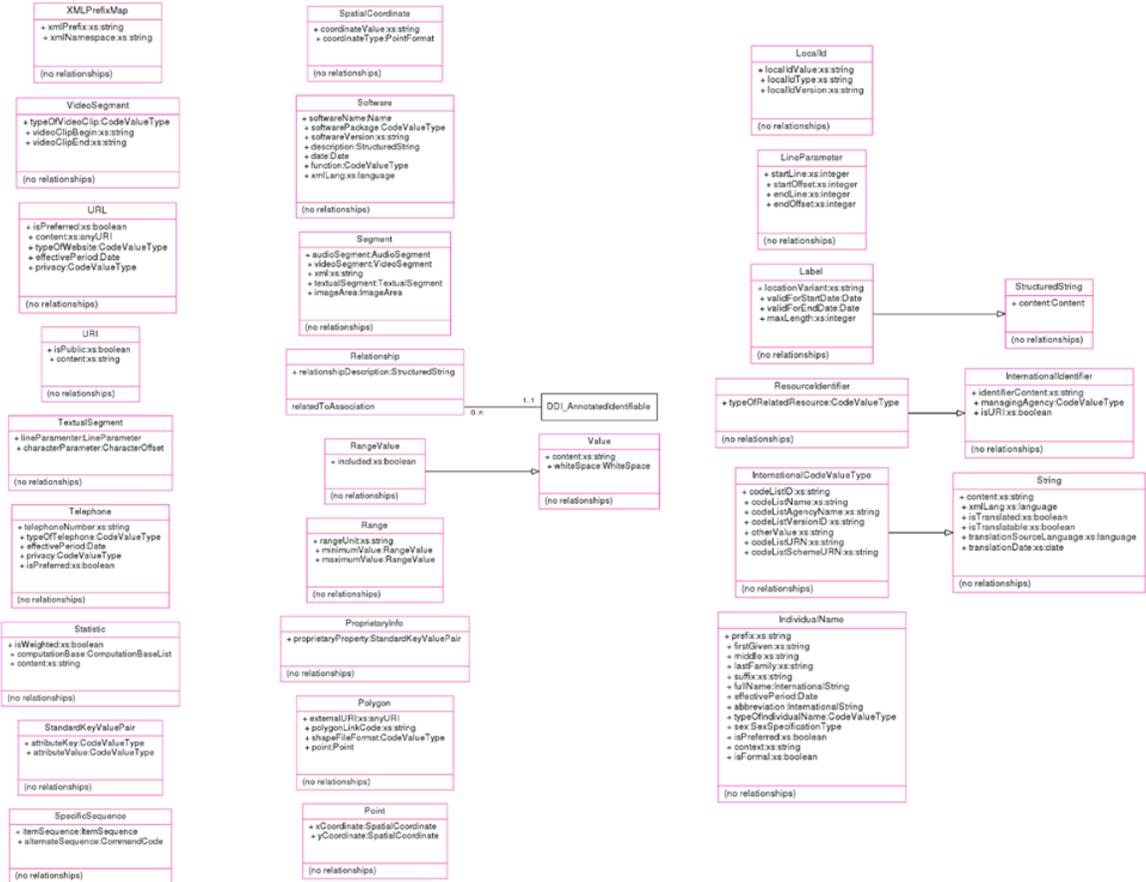
Language represents formal natural language identifiers, as defined by BCP 47 (currently represented by RFC 4646 and RFC 4647 or its successor(s)). The value space and lexical space of language are the set of all strings that conform to the pattern `[a-zA-Z]{1,8}(-[a-zA-Z0-9]{1,8})*`. This is the set of strings accepted by the grammar given in RFC 3066, which is now obsolete; the current specification of language codes is more restrictive. The base type of language is token. Note: The regular expression above provides the only normative constraint on the lexical and value spaces of this type. The additional constraints imposed on language identifiers by BCP 47 and its successor(s), and in particular their requirement that language codes be registered with IANA or ISO if not given in ISO 639, are not part of this datatype as defined here. Note: BCP 47 specifies that language codes "are to be treated as case insensitive; there exist conventions for capitalization of some of the subtags, but these **MUST NOT** be taken to carry meaning." Since the language datatype is derived from string, it inherits from string a one-to-one mapping from lexical representations to values. The literals 'MN' and 'mn' (for Mongolian) therefore correspond to distinct values and have distinct canonical forms. Users of this specification should be aware of this fact, the consequence of which is that the case-insensitive treatment of language values prescribed by BCP 47] does not follow from the definition of this datatype given here; applications which require case-insensitivity should make appropriate adjustments. Note: The empty string is not a member of the -value space- of language. Some constructs which normally take language codes as their values, however, also allow the empty string. The attribute `xml:lang` defined by XML is one example; there, the empty string overrides a value which would otherwise be inherited, but without specifying a new value. The language datatype has the following constraining facets with the values shown; these facets may be specified in the derivation of new types, if the value given is at least as restrictive as the one shown: `pattern = [a-zA-Z]{1,8}(-[a-zA-Z0-9]{1,8})*`  
`whiteSpace = collapse` Further details at w3.org [<http://www.w3.org/TR/xmlschema11-2/#language>]

## xhtml:BlkNoForm.mix

### Definition

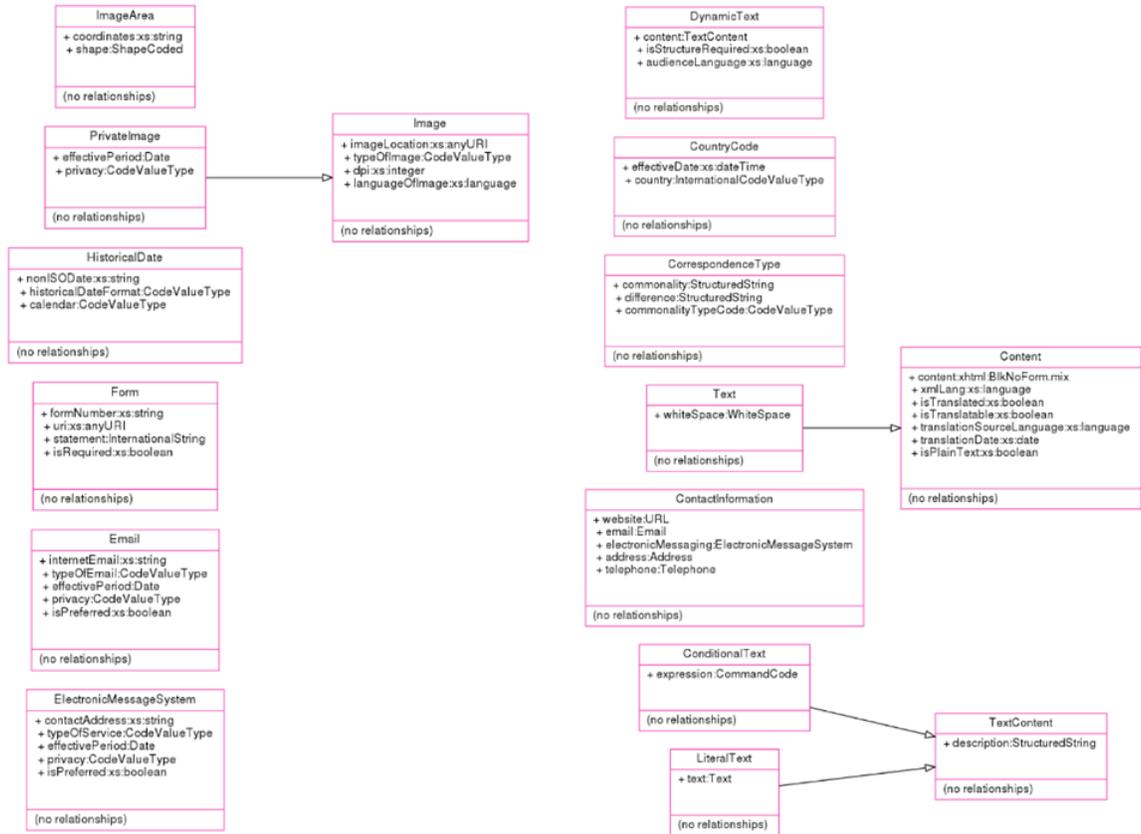
The following xhtml tags are available for use in Content: `address`, `blockquote`, `pre`, `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `hr`, `div`, `p`, `a`, `abbr`, `acronym`, `cite`, `code`, `dfn`, `em`, `kbd`, `q`, `samp`, `strong`, `var`, `b`, `big`, `i`, `small`, `sub`, `sup`, `tt`, `br`, `span`, `dl`, `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., `xhtml:div`. Further details at w3.org [<http://www.w3.org/1999/xhtml/>]

# Chapter 3. ComplexDataTypes (Full Review)

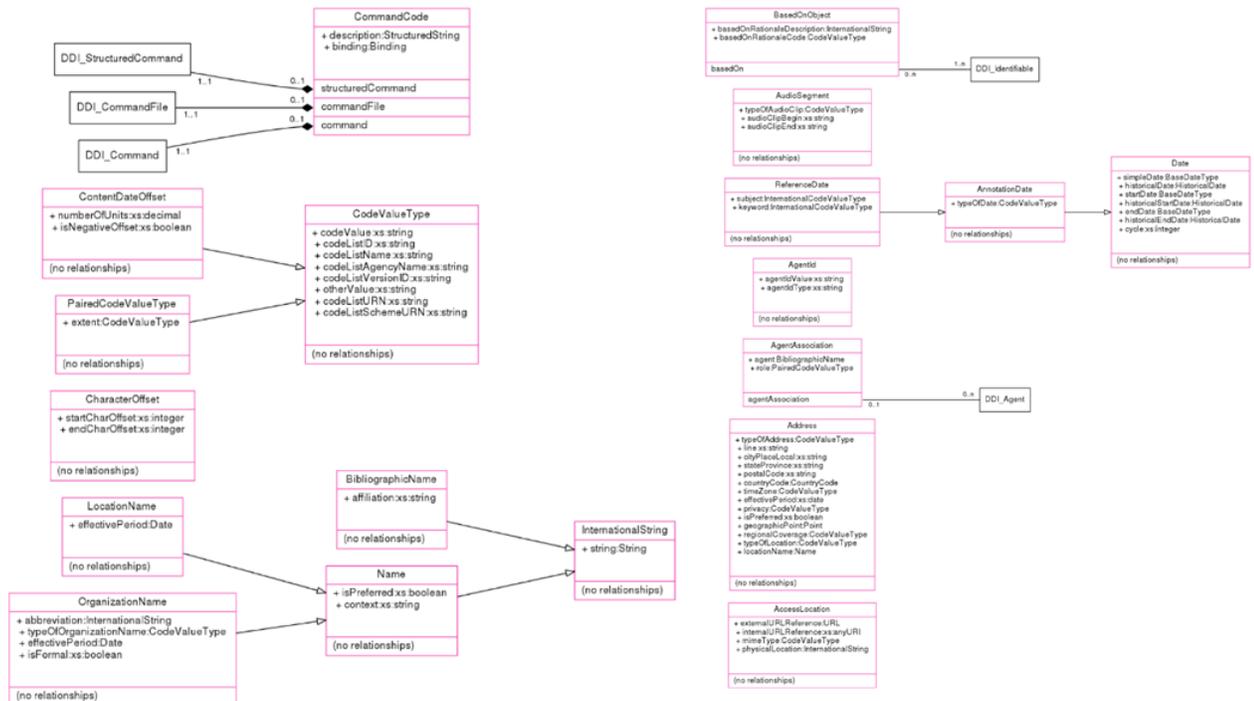


Complex Data Types (1/3)

# ComplexDataTypes (Full Review)



## Complex Data Types (2/3)



## Complex Data Types (3/3)

# AccessLocation

## Definition

A set of access information for a Machine including external and internal URL, mime type, and physical location

## Properties

### externalURLReference

Datatype	URL
Cardinality	0..n
An external URL	

### internalURLReference

Datatype	xs:anyURI
Cardinality	0..1
The internal URL.	

### contentType

Datatype	CodeValueType
Cardinality	0..1

### physicalLocation

Datatype	InternationalString
Cardinality	0..n
The physical location of the machine	

# Address

## Based on DDI 3.2

a:AddressType

## Definition

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.

## Example

For example: 1. OFFICE, ABS HOUSE, 45 Benjamin Way, Belconnen, Canberra, ACT, 2617, AU 2. OFFICE, Institute of Education, 20 Bedford Way, London, WC1H 0AL, UK

## Properties

### typeOfAddress

Datatype	CodeValueType
Cardinality	0..1
Indicates address type (i.e. home, office, mailing, etc.)	

### line

Datatype	xs:string
Cardinality	0..n
Number and street including office or suite number. May use multiple lines.	

### cityPlaceLocal

Datatype	xs:string
Cardinality	0..1
City, place, or local area used as part of an address.	

### stateProvince

Datatype	xs:string
Cardinality	0..1
A major subnational division such as a state or province used to identify a major region within an address.	

### postalCode

Datatype	xs:string
Cardinality	0..1
Postal or ZIP Code	

### countryCode

Datatype	CountryCode
Cardinality	0..1
Country of the location	

### timeZone

Datatype	CodeValueType
----------	---------------

Cardinality	0..1
Time zone of the location expressed as code.	

## effectivePeriod

Datatype	xs:date
Cardinality	0..1
Clarifies when the identification information is accurate.	

## privacy

Datatype	CodeValueType
Cardinality	0..1
Specify the level privacy for the address as public, restricted, or private. Supports the use of an external controlled vocabulary	

## isPreferred

Datatype	xs:boolean
Cardinality	0..1
Set to "true" if this is the preferred location for contacting the organization or individual.	

## geographicPoint

Datatype	Point
Cardinality	0..1
Geographic coordinates corresponding to the address.	

## regionalCoverage

Datatype	CodeValueType
Cardinality	0..1
The region covered by the agent at this address	

## typeOfLocation

Datatype	CodeValueType
Cardinality	0..1
The type or purpose of the location (i.e. regional office, distribution center, home)	

## locationName

Datatype	Name
Cardinality	0..1
Name of the location if applicable.	

# AgentAssociation

## Based on DDI 3.2

r:ContributorType

## Definition

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

## Properties

### agent

Datatype	BibliographicName
Cardinality	0..1
Full name of the contributor. Language equivalents should be expressed within the International String structure.	

### role

Datatype	PairedCodeValueType
Cardinality	0..n
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.	

## Relationships

### agentAssociation

Target	Agent
Type	Neither
Source Cardinality	0..1
Target Cardinality	0..n
Reference to an agent as described by any object that is a member of the abstract type Agent.	

## AgentId

### Definition

Persistent identifier for a researcher using a system like ORCID

## Example

ORCID

## Properties

### agentIdValue

Datatype	xs:string
Cardinality	1..1
The identifier for the agent.	

### agentIdType

Datatype	xs:string
Cardinality	1..1
The identifier system in use.	

## AnnotationDate

## Extends

This class extends Date

**Table 3.1. RDF Mapping (AnnotationDate)**

Label	Type	URI
date	owl:sameAs	<a href="http://purl.org/dc/elements/1.1/date">http://purl.org/dc/elements/1.1/date</a>
date	owl:sameAs	<a href="http://purl.org/dc/terms/date">http://purl.org/dc/terms/date</a>

## Definition

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.

## Properties

### typeOfDate

Datatype	CodeValueType
Cardinality	0..n
Use to specify the type of date. This may reflect the refinements of dc:date such as dateAccepted, dateCopyrighted, dateSubmitted, etc.	

# AudioSegment

## Based on DDI 3.2

r:AudioType

### Definition

Describes the type and length of the audio segment.

### Properties

#### typeofAudioClip

Datatype	CodeValueType
Cardinality	0..1
The type of audio clip provided. Supports the use of a controlled vocabulary.	

#### audioClipBegin

Datatype	xs:string
Cardinality	0..1
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

Datatype	xs:string
Cardinality	0..1
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.	

# BasedOnObject

## Based on DDI 3.2

r:BasedOnObjectType

### Definition

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 class, 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

### basedOnRationaleDescription

Datatype	InternationalString
Cardinality	0..1
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.	

### basedOnRationaleCode

Datatype	CodeValueType
Cardinality	0..1
RationaleCode is primarily for internal processing flags within an organization or system. Supports the use of an external controlled vocabulary.	

## Relationships

### basedOn

Target	Identifiable
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..n

## BibliographicName

### Extends

This class extends InternationalString

### Based on DDI 3.2

r:BibliographicNameType

### Definition

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.

## Properties

### affiliation

Datatype	xs:string
Cardinality	0..1
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.	

## CharacterOffset

### Based on DDI 3.2

r:CharacterParameterType

### Definition

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

## Properties

### startCharOffset

Datatype	xs:integer
Cardinality	0..1
Number of characters from beginning of the document, indicating the inclusive start of the text range.	

### endCharOffset

Datatype	xs:integer
Cardinality	0..1
Number of characters from the beginning of the document, indicating the inclusive end of the text segment.	

## CodeValueType

### Based on DDI 3.2

r:CodeValueType

### Definition

Allows for string content which may be taken from an externally maintained controlled vocabulary (code value). If the content is from a controlled vocabulary provide the code value, as well as a reference to the code list 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 CodeValue structure. Use of shared controlled vocabularies helps support interoperability and machine actionability.

## Properties

### codeValue

Datatype	xs:string
Cardinality	1..1
The actual value.	

### codeListID

Datatype	xs:string
Cardinality	0..1
The ID of the code list (controlled vocabulary) that the content was taken from.	

### codeListName

Datatype	xs:string
Cardinality	0..1
The name of the code list.	

### codeListAgencyName

Datatype	xs:string
Cardinality	0..1
The name of the agency maintaining the code list.	

### codeListVersionID

Datatype	xs:string
Cardinality	0..1
The version number of the code list (default is 1.0).	

### otherValue

Datatype	xs:string
Cardinality	0..1
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.	

### codeListURN

Datatype	xs:string
Cardinality	0..1

The URN of the codelist.
--------------------------

## codeListSchemeURN

Datatype	xs:string
Cardinality	0..1
If maintained within a scheme, the URN of the scheme containing the codelist.	

# CommandCode

## Based on DDI 3.2

r:CommandCodeType

## Definition

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

### description

Datatype	StructuredString
Cardinality	0..1
A description of the purpose and use of the command code provided. Supports multiple languages.	

### binding

Datatype	Binding
Cardinality	0..n
Defines the link between the OutParameter of an external object to an InParameter of this CommandCode.	

## Relationships

### structuredCommand

Target	StructuredCommand
Type	Composition
Source Cardinality	0..1
Target Cardinality	1..1

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

## commandFile

Target	CommandFile
Type	Composition
Source Cardinality	0..1
Target Cardinality	1..1
Identifies and provides a link to an external copy of the command, for example, a SAS Command Code script.	

## command

Target	Command
Type	Composition
Source Cardinality	0..1
Target Cardinality	1..1
This is an in-line provision of the command itself.	

# ConditionalText

## Extends

This class extends TextContent

## Based on DDI 3.2

d:ConditionalTextType

## Definition

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).

## Properties

### expression

Datatype	CommandCode
Cardinality	0..1
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.	

# ContactInformation

## Based on DDI 3.2

a:ContactInformationType

### Definition

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

#### website

Datatype	URL
Cardinality	0..n
The URL of the Agent's website	

#### email

Datatype	Email
Cardinality	0..n
Email contact information	

#### electronicMessaging

Datatype	ElectronicMessageSystem
Cardinality	0..n
Electronic messaging other than email	

#### address

Datatype	Address
Cardinality	0..n
The address for contact.	

#### telephone

Datatype	Telephone
Cardinality	0..n
Telephone for contact	

# Content

## Based on DDI 3.2

r:ContentType

## Definition

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 `xmlLang`. 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

### content

Datatype	xhtml:BlkNoForm.mix
Cardinality	1..n
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., xhtml:div. See DDI Technical Manual Part I for additional details.	

### xmlLang

Datatype	xs:language
Cardinality	0..1
Indicates the language of content.	

### isTranslated

Datatype	xs:boolean
Cardinality	0..1
Indicates whether content is a translation (true) or an original (false).	

### isTranslatable

Datatype	xs:boolean
Cardinality	0..1
Indicates whether content is translatable (true) or not (false).	

### translationSourceLanguage

Datatype	xs:language
----------	-------------

Cardinality	0..n
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.	

## translationDate

Datatype	xs:date
Cardinality	0..1
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.	

## isPlainText

Datatype	xs:boolean
Cardinality	0..1
Indicates that the content is to be treated as plain text (no formatting). You may use DDIPprofile 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.	

# ContentDateOffset

## Extends

This class extends CodeValueType

## Based on DDI 3.2

r:ContentDateOffsetType

## Definition

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.

## Properties

### numberOfUnits

Datatype	xs:decimal
Cardinality	0..1
The number of units to off-set the date for this item expressed as a decimal.	

### isNegativeOffset

Datatype	xs:boolean
Cardinality	0..1

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.

# CorrespondenceType

## Based on DDI 3.2

cm:CorrespondenceType

## Definition

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

## Properties

### commonality

Datatype	StructuredString
Cardinality	0..1
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.	

### difference

Datatype	StructuredString
Cardinality	0..1
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.	

### commonalityTypeCode

Datatype	CodeValueType
Cardinality	0..n
Commonality expressed as a term or code. Supports the use of an external controlled vocabulary. If repeated, clarify each external controlled vocabulary used.	

# CountryCode

## Abstract

This class is abstract

## Based on DDI 3.2

r:CountryCodeType

## Definition

Provides means of expressing a code/term for the country plus an optional valid date.

## Properties

### effectiveDate

Datatype	xs:dateTime
Cardinality	0..1
If it is important to specify the date that this code is effective in order to accurately capture a name or similar change, specify that here.	

### country

Datatype	InternationalCodeValueType
Cardinality	1..1
The code or term used to designate the country. If a term, indicate the language.	

## Date

### Based on DDI 3.2

r:DateType

## Definition

Provides the structure of a Date element, which allows a choice between single, simple dates (of BaseDateType) or date ranges. If the Date element contains a range, Cycle may be used to indicate occurrence of this range within a series of ranges as an integer identifying the cycle, i.e. the 4th wave of a data collection cycle would have. BaseDateType allows for different date time combinations to provide a simple and convenient mechanism to specify different date and time values with a machine actionable format specified by regular expressions.

## Example

## Explanatory Notes

simpleDate allows one of a set of date-time (YYYY-MM-DDTHH:MM:SS), date (YYYY-MM-DD), year-month (YYYY-MM), year (YYYY), and duration (PnYnMnDnHnMnS), which is formatted according to ISO 8601 and backed supported by regular expressions in the BaseDateType.

## Properties

### simpleDate

Datatype	BaseDateType
----------	--------------

Cardinality	0..1
A single point in time. If a duration is expressed as a SimpleDate it is defining a period of time without a designated Start or End date.	

## historicalDate

Datatype	HistoricalDate
Cardinality	0..1
A simple date expressed in a historical date format, including a specification of the date format and calendar used.	

## startDate

Datatype	BaseDateType
Cardinality	0..1
Start of a date range. If there is a start date with no end date provided, this implies that the end date is unknown but that the date being recorded is not just a simple date but a range of unknown duration.	

## historicalStartDate

Datatype	HistoricalDate
Cardinality	0..1
A start date expressed in a historical date format, including a specification of the date format and calendar used.	

## endDate

Datatype	BaseDateType
Cardinality	0..1
End of a date range which may or may not have a known start date.	

## historicalEndDate

Datatype	HistoricalDate
Cardinality	0..1
An end date expressed in a historical date format, including a specification of the date format and calendar used.	

## cycle

Datatype	xs:integer
Cardinality	0..1
Use to indicate occurrence of this range within a series of ranges as an integer identifying the cycle, i.e. the 4th wave of a data collection cycle would have	

# DynamicText

## Based on DDI 3.2

d:DynamicTextType

### Definition

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

#### content

Datatype	TextContent
Cardinality	1..n
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

Datatype	xs:boolean
Cardinality	0..1
If textual structure (e.g. size, color, font, etc.) is required to understand the meaning of the content change value to "true".	

#### audienceLanguage

Datatype	xs:language
Cardinality	0..1
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 within the question text.	

# ElectronicMessageSystem

## Based on DDI 3.2

a:InstantMessagingType

### Definition

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.

## Example

Skype account, etc.

## Properties

### contactAddress

Datatype	xs:string
Cardinality	0..1
Account identification for contacting	

### typeOfService

Datatype	CodeValueType
Cardinality	0..1
Indicates the type of service used. Supports the use of a controlled vocabulary.	

### effectivePeriod

Datatype	Date
Cardinality	0..1
Time period during which the account is valid.	

### privacy

Datatype	CodeValueType
Cardinality	0..1
Specify the level privacy for the address as public, restricted, or private. Supports the use of an external controlled vocabulary.	

### isPreferred

Datatype	xs:boolean
Cardinality	0..1
Set to "true" if this is the preferred address.	

## Email

### Based on DDI 3.2

r:EmailType

### Definition

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

## Example

info@ddialliance.org; ex.ample@somewhere.org

## Properties

### internetEmail

Datatype	xs:string
Cardinality	0..1
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.	

### typeOfEmail

Datatype	CodeValueType
Cardinality	0..1
Code indicating the type of e-mail address. Supports the use of an external controlled vocabulary. (e.g. home, office)	

### effectivePeriod

Datatype	Date
Cardinality	0..1
Time period for which the e-mail address is valid.	

### privacy

Datatype	CodeValueType
Cardinality	0..1
Indicates the level of privacy	

### isPreferred

Datatype	xs:boolean
Cardinality	0..1
Set to true if this is the preferred email	

## Form

### Based on DDI 3.2

a:FormType

## Definition

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

### formNumber

Datatype	xs:string
Cardinality	0..1
The number or other means of identifying the form.	

### uri

Datatype	xs:anyURI
Cardinality	0..1
The URN or URL of the form.	

### statement

Datatype	InternationalString
Cardinality	0..1
A statement regarding the use, coverage, and purpose of the form.	

### isRequired

Datatype	xs:boolean
Cardinality	0..1
Set to "true" if the form is required. Set to "false" if the form is optional.	

## HistoricalDate

### Based on DDI 3.2

r:HistoricalDateType

## Definition

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

## Properties

### nonISODate

Datatype	xs:string
----------	-----------

Cardinality	1..1
This is the date expressed in a non-ISO compliant structure. Primarily used to retain legacy content or to express non-Gregorian calendar dates.	

## historicalDateFormat

Datatype	CodeValueType
Cardinality	0..1
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.	

## calendar

Datatype	CodeValueType
Cardinality	0..1
Specifies the type of calendar used (e.g., Gregorian, Julian, Jewish).	

# Image

## Based on DDI 3.2

r:ImageType

## Definition

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

## Properties

### imageLocation

Datatype	xs:anyURI
Cardinality	0..1
A reference to the location of the image using a URI.	

### typeOfImage

Datatype	CodeValueType
Cardinality	0..1
Brief description of the image type. Supports the use of an external controlled vocabulary.	

### dpi

Datatype	xs:integer
Cardinality	0..1

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

## languageOfImage

Datatype	xs:language
Cardinality	0..1
Language of image.	

## ImageArea

### Based on DDI 3.2

r:ImageAreaType

### Definition

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

#### coordinates

Datatype	xs:string
Cardinality	0..1
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

Datatype	ShapeCoded
Cardinality	1..1
A fixed set of valid responses includes Rectangle, Circle, and Polygon.	

## IndividualName

### Based on DDI 3.2

a:IndividualNameType

### Definition

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

### prefix

Datatype	xs:string
Cardinality	0..1
Title that precedes the name of the individual, such as Ms., or Dr.	

### firstGiven

Datatype	xs:string
Cardinality	0..1
First (given) name of the individual	

### middle

Datatype	xs:string
Cardinality	0..n
Middle name or initial of the individual	

### lastFamily

Datatype	xs:string
Cardinality	0..1
Last (family) name /surname of the individual	

### suffix

Datatype	xs:string
Cardinality	0..1
Title that follows the name of the individual, such as Esq.	

### fullName

Datatype	InternationalString
Cardinality	0..1
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.	

### effectivePeriod

Datatype	Date
----------	------

Cardinality	0..1
Clarifies when the name information is accurate.	

## abbreviation

Datatype	InternationalString
Cardinality	0..1
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.	

## typeOfIndividualName

Datatype	CodeValueType
Cardinality	0..1
The type of individual name provided. the use of a controlled vocabulary is strongly recommended. At minimum this should include, e.g. PreviousFormalName, Nickname (or CommonName), Other.	

## sex

Datatype	SexSpecificationType
Cardinality	0..1
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.	

## isPreferred

Datatype	xs:boolean
Cardinality	0..1
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".	

## context

Datatype	xs:string
Cardinality	0..1
A name may be specific to a particular context, i.e. common usage, business, social, etc.. Identify the context related to the specified name.	

## isFormal

Datatype	xs:boolean
Cardinality	0..1
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.

## InternationalCodeValueType

### Extends

This class extends String

### Based on DDI 3.2

r:InternationalCodeValueType

### Definition

Allows for string content which may be taken from an externally maintained controlled vocabulary (code value). If the content is from a controlled vocabulary provide the code value, as well as a reference to the code list from which the value is taken. This differs from a CodeValue only by the provision of a language-location specification. DDI uses the International CodeValue in cases where controlled vocabularies are assumed to be highly language specific, such as nationally maintained subject headings, thesauri, or keywords derived from the content of documents. The ability to identify language is especially important when supporting searches by external language-specific search engines. Provide as many of the identifying attributes as needed to adequately identify the controlled vocabulary.

### Properties

#### codeListID

Datatype	xs:string
Cardinality	0..1
The ID of the code list (controlled vocabulary) that the content was taken from.	

#### codeListName

Datatype	xs:string
Cardinality	0..1
The name of the code list.	

#### codeListAgencyName

Datatype	xs:string
Cardinality	0..1
The name of the agency maintaining the code list.	

#### codeListVersionID

Datatype	xs:string
Cardinality	0..1

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

## otherValue

Datatype	xs:string
Cardinality	0..1
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.	

## codeListURN

Datatype	xs:string
Cardinality	0..1
The URN of the codelist.	

## codeListSchemeURN

Datatype	xs:string
Cardinality	0..1
If maintained within a scheme, the URN of the scheme containing the codelist.	

# InternationalIdentifier

## Based on DDI 3.2

r:InternationalIdentifierType

## Definition

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

### identifierContent

Datatype	xs:string
Cardinality	0..1
An identifier as it should be listed for identification purposes.	

### managingAgency

Datatype	CodeValueType
Cardinality	0..1
The identification of the Agency which assigns and manages the identifier, i.e., ISBN, ISSN, DOI, etc.	

## isURI

Datatype	xs:boolean
Cardinality	0..1
Set to "true" if Identifier is a URI	

# InternationalString

## Based on DDI 3.2

r:InternationalStringType

## Definition

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

### string

Datatype	String
Cardinality	0..n
A non-formatted string of text with an attribute that designates the language of the text. Repeat This class to express the same content in another language.	

## Label

## Extends

This class extends StructuredString

## Based on DDI 3.2

r:LabelType

**Table 3.2. RDF Mapping (Label)**

Label	Type	URI
rdfs:label	rdf:type	<a href="http://www.w3.org/2000/01/rdf-schema#label">http://www.w3.org/2000/01/rdf-schema#label</a>

## Definition

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

## Properties

### locationVariant

Datatype	xs:string
Cardinality	0..1
Indicate the locality specification for content that is specific to a geographic area. May be a country code, sub-country code, or area name.	

### validForStartDate

Datatype	Date
Cardinality	0..1
Allows for the specification of a starting date for the period that this label is valid. The date must be formatted according to ISO 8601.	

### validForEndDate

Datatype	Date
Cardinality	0..1
Allows for the specification of an ending date for the period that this label is valid. The date must be formatted according to ISO 8601.	

### maxLength

Datatype	xs:integer
Cardinality	0..1
A positive integer indicating the maximum number of characters in the label.	

## LineParameter

### Based on DDI 3.2

r:LineParameterType

### Definition

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

## Properties

### startLine

Datatype	xs:integer
----------	------------

Cardinality	0..1
Number of lines from beginning of the document.	

## startOffset

Datatype	xs:integer
Cardinality	0..1
Number of characters from start of the line specified in StartLine.	

## endLine

Datatype	xs:integer
Cardinality	0..1
Number of lines from beginning of the document.	

## endOffset

Datatype	xs:integer
Cardinality	0..1
Number of characters from the start of the line specified in EndLine.	

# LiteralText

## Extends

This class extends TextContent

## Based on DDI 3.2

d:LiteralTextType

## Definition

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

## Properties

### text

Datatype	Text
Cardinality	0..1
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 xmspace to "preserve".

## LocalId

### Definition

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.

### Example

The name of a variable within a dataset is a localId. Software systems might use their own identifier systems for performance reasons (e.g. incrementing integers).

## Properties

### localIdValue

Datatype	xs:string
Cardinality	1..1
Value of the local ID.	

### localIdType

Datatype	xs:string
Cardinality	1..1
Type of identifier, specifying the context of the identifier.	

### localIdVersion

Datatype	xs:string
Cardinality	0..1
Version of the Local ID.	

## LocationName

### Extends

This class extends Name

### Based on DDI 3.2

a:LocationNameType

## Definition

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

## Properties

### effectivePeriod

Datatype	Date
Cardinality	0..1
The time period for which this name is accurate and in use.	

## Name

## Extends

This class extends InternationalString

## Based on DDI 3.2

r:NameType

## Definition

A reusable type assigned to an element with the naming convention XxxName e.g. OrganizationName at selected locations where the element may be assumed to be administered by a registry or is otherwise shared. This is a human understandable name (word, phrase, or mnemonic) that reflects the ISO/IEC 11179-5 naming principles. An item administered by a registry should have at least one name. Names within an administered registry should follow the naming conventions of the registry. If more than one name is provided the context of each name should be noted and one name selected as the preferred name. See ISO/IEC 11179-5 Information Technology - Metadata Registries (MDR) Part 5: naming and identification principles. ISO/IEC1179-5:2005(E).

## Properties

### isPreferred

Datatype	xs:boolean
Cardinality	0..1
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".	

### context

Datatype	xs:string
Cardinality	0..1
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.	

# OrganizationName

## Extends

This class extends Name

## Based on DDI 3.2

a:OrganizationNameType

## Definition

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.

## Properties

### abbreviation

Datatype	InternationalString
Cardinality	0..1
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.	

### typeOfOrganizationName

Datatype	CodeValueType
Cardinality	0..1
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.	

### effectivePeriod

Datatype	Date
Cardinality	0..1
The time period for which this name is accurate and in use.	

### isFormal

Datatype	xs:boolean
Cardinality	0..1
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.

## PairedCodeValueType

### Extends

This class extends CodeValueType

### Definition

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

### Example

When used to assign a role to an actor within a specific activity this term would express the degree of contribution. Contributor with Role=Editor and extent=Lead.

### Properties

#### extent

Datatype	CodeValueType
Cardinality	0..1
Describes the extent to which the parent term applies for the specific case using an external controlled vocabulary.	

## Point

### Based on DDI 3.2

r:PointType

### Definition

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

### Properties

#### xCoordinate

Datatype	SpatialCoordinate
Cardinality	0..1
An X coordinate (latitudinal equivalent) value and format expressed using the Spatial Coordinate structure.	

## yCoordinate

Datatype	SpatialCoordinate
Cardinality	0..1
A Y coordinate (longitudinal equivalent) value and format expressed using the Spatial Coordinate structure.	

## Polygon

### Based on DDI 3.2

r:PolygonType

### Definition

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

#### externalURI

Datatype	xs:anyURI
Cardinality	0..1
Note that ExternalURI points to the boundary file location.	

#### polygonLinkCode

Datatype	xs:string
Cardinality	0..1
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

Datatype	CodeValueType
Cardinality	0..1
The format of the shape file existing at the location indicated by the sibling ExternalURI element.	

#### point

Datatype	Point
----------	-------

Cardinality	4..n
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.	

## PrivateImage

### Extends

This class extends Image

### Based on DDI 3.2

a:PrivateImageType

### Definition

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.

### Properties

#### effectivePeriod

Datatype	Date
Cardinality	0..1
The period for which this image is effective/valid.	

#### privacy

Datatype	CodeValueType
Cardinality	0..1
Specify the level privacy for the image as public, restricted, or private. Supports the use of an external controlled vocabulary.	

## ProprietaryInfo

### Based on DDI 3.2

r:ProprietaryInfoType

### Definition

Contains information proprietary to the software package which produced the data file. This is expressed as a set of key(name)-value pairs.

## Properties

### proprietaryProperty

Datatype	StandardKeyValuePair
Cardinality	0..n
A structure that supports the use of a standard key value pair. Note that this information is often not interoperable and is provided to support the use of the metadata within specific systems.	

## Range

### Based on DDI 3.2

r:RangeType

### Definition

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

## Properties

### rangeUnit

Datatype	xs:string
Cardinality	0..1
Specifies the units in the range.	

### minimumValue

Datatype	RangeValue
Cardinality	0..1
Minimum value in the range.	

### maximumValue

Datatype	RangeValue
Cardinality	0..1
Maximum value in the range.	

## RangeValue

### Extends

This class extends Value

## Based on DDI 3.2

r:RangeValueType

### Definition

Describes a bounding value of a string.

### Properties

#### included

Datatype	xs:boolean
Cardinality	0..1
Set to "true" if the value is included in the range.	

## ReferenceDate

### Extends

This class extends AnnotationDate

## Based on DDI 3.2

r:ReferenceDateType

### Definition

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.

### Properties

#### subject

Datatype	InternationalCodeValueType
Cardinality	0..n
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.	

#### keyword

Datatype	InternationalCodeValueType
Cardinality	0..n

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.

## Relationship

### Based on DDI 3.2

r:RelationshipType

### Definition

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

#### relationshipDescription

Datatype	StructuredString
Cardinality	0..1
A description of the nature of the relationship between the parent element of the relationship item and the DDI object to which it is related.	

### Relationships

#### relatedToAssociation

Target	AnnotatedIdentifiable
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Reference to the item within the DDI Instance to which this item is related.	

## ResourceIdentifier

### Extends

This class extends InternationalIdentifier

### Definition

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.

## Properties

### typeOfRelatedResource

Datatype	CodeValueType
Cardinality	0..n
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.	

## Segment

### Based on DDI 3.2

r:SegmentType

### Definition

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

### audioSegment

Datatype	AudioSegment
Cardinality	0..n
Describes the type and length of the audio segment.	

### videoSegment

Datatype	VideoSegment
Cardinality	0..n
Describes the type and length of the video segment.	

### xml

Datatype	xs:string
Cardinality	0..n
An X-Pointer expression identifying a node in the XML document.	

### textualSegment

Datatype	TextualSegment
Cardinality	0..n

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

## imageArea

Datatype	ImageArea
Cardinality	0..n
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.	

# Software

## Based on DDI 3.2

r:SoftwareType

## Definition

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

## Properties

### softwareName

Datatype	Name
Cardinality	0..n
The name of the software package, including its producer.	

### softwarePackage

Datatype	CodeValueType
Cardinality	0..1
A coded value from a controlled vocabulary, describing the software package.	

### softwareVersion

Datatype	xs:string
Cardinality	0..1
The version of the software package. Defaults to '1.0'.	

### description

Datatype	StructuredString
Cardinality	0..1
A description of the content and purpose of the software. May be expressed in multiple languages and supports the use of structured content.	

**date**

Datatype	Date
Cardinality	0..1
Supported date of the software package with, at minimum, a release date if known.	

**function**

Datatype	CodeValueType
Cardinality	0..n
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.	

**xmlLang**

Datatype	xs:language
Cardinality	0..1
Language (human language) of the software package.	

**SpatialCoordinate****Based on DDI 3.2**

r:SpatialCoordinateType

**Definition**

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****coordinateValue**

Datatype	xs:string
Cardinality	0..1
The value of the coordinate expressed as a string.	

**coordinateType**

Datatype	PointFormat
Cardinality	1..1
Identifies the type of point coordinate system using a controlled vocabulary. Point formats include decimal degree, degrees minutes seconds, decimal minutes, meters, and feet.	

# SpecificSequence

## Definition

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

### itemSequence

Datatype	ItemSequence
Cardinality	1..1
Identifies the type of sequence to use. Values include InOrderOfAppearance, Random, Rotate, and Other.	

### alternateSequence

Datatype	CommandCode
Cardinality	0..1
Information on the command used to generate an alternative means of determining sequence changes. If used, the ItemSequenceType should be "Other".	

# StandardKeyValuePair

## Based on DDI 3.2

r:StandardKeyValuePairType

## Definition

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

### attributeKey

Datatype	CodeValueType
Cardinality	0..1
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

Datatype	CodeValueType
----------	---------------

Cardinality	0..1
The value assigned to the named Key expressed as a string. Supports the use of an external controlled vocabulary.	

## Statistic

### Based on DDI 3.2

pi:StatisticType

### Definition

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

## Properties

### isWeighted

Datatype	xs:boolean
Cardinality	0..1
Set to "true" if the statistic is weighted using the weight designated in VariableStatistics.	

### computationBase

Datatype	ComputationBaseList
Cardinality	0..1
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.	

### content

Datatype	xs:string
Cardinality	1..1

## String

### Based on DDI 3.2

r:StringType

### Definition

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, `translationSourceLanguage` to indicate the source languages used in creating this translation, and `translationDate`.

## Properties

### content

Datatype	xs:string
Cardinality	1..1
value of this string	

### xmlLang

Datatype	xs:language
Cardinality	0..1
Indicates the language of content. Note that <code>xmlang</code> allows for a simple 2 or 3 character language code or a language code extended by a country code , for example <code>en-au</code> for English as used in Australia.	

### isTranslated

Datatype	xs:boolean
Cardinality	0..1
Indicates whether content is a translation ( <code>true</code> ) or an original ( <code>false</code> ).	

### isTranslatable

Datatype	xs:boolean
Cardinality	0..1
Indicates whether content is translatable ( <code>true</code> ) or not ( <code>false</code> ). An example of something that is not translatable would be a MNEMONIC of an object or a number.	

### translationSourceLanguage

Datatype	xs:language
Cardinality	0..n
List the language code of the source. Repeat of multiple language sources are used.	

### translationDate

Datatype	xs:date
Cardinality	0..1
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.	

# StructuredString

## Based on DDI 3.2

r:StructuredStringType

### Definition

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

#### content

Datatype	Content
Cardinality	1..n
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.	

# Telephone

## Based on DDI 3.2

a:TelephoneType

### Definition

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.

### Example

+12 345 67890123

### Properties

#### telephoneNumber

Datatype	xs:string
Cardinality	0..1
The telephone number including country code if appropriate.	

## typeOfTelephone

Datatype	CodeValueType
Cardinality	0..1
Indicates type of telephone number provided (home, fax, office, cell, etc.). Supports the use of a controlled vocabulary.	

## effectivePeriod

Datatype	Date
Cardinality	0..1
Time period during which the telephone number is valid.	

## privacy

Datatype	CodeValueType
Cardinality	0..1
Specify the level privacy for the telephone number as public, restricted, or private. Supports the use of an external controlled vocabulary.	

## isPreferred

Datatype	xs:boolean
Cardinality	0..1
Set to "true" if this is the preferred telephone number for contact.	

## Text

### Extends

This class extends Content

### Based on DDI 3.2

d:TextType

### Definition

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

### Properties

#### whiteSpace

Datatype	WhiteSpace
Cardinality	0..1

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".

## TextContent

### Abstract

This class is abstract

### Based on DDI 3.2

d:TextContentType

### Definition

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 element Description to all members using TextContent as an extension base.

### Properties

#### description

Datatype	StructuredString
Cardinality	0..1
A description of the content and purpose of the text segment. May be expressed in multiple languages and supports the use of structured content.	

## TextualSegment

### Based on DDI 3.2

r:TextualType

### Definition

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

#### lineParameter

Datatype	LineParameter
Cardinality	1..1
Specification of the line and offset for the beginning and end of the segment.	

## characterParameter

Datatype	CharacterOffset
Cardinality	1..1
Specification of the character offset for the beginning and end of the segment.	

## URI

### Based on DDI 3.2

pi:URIType

### Definition

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

### Properties

#### isPublic

Datatype	xs:boolean
Cardinality	0..1
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.	

#### content

Datatype	xs:string
Cardinality	1..1

## URL

### Based on DDI 3.2

a:URLType

### Definition

A web site URL

### Properties

#### isPreferred

Datatype	xs:boolean
----------	------------

Cardinality	0..1
Set to "true" if this is the preferred URL.	

**content**

Datatype	xs:anyURI
Cardinality	1..1
The content of the URL	

**typeOfWebsite**

Datatype	CodeValueType
Cardinality	0..1
The type of URL for example personal, project, organization, division, etc.	

**effectivePeriod**

Datatype	Date
Cardinality	0..1
The period for which this URL is valid.	

**privacy**

Datatype	CodeValueType
Cardinality	0..1
Indicates the privacy level of this URL	

**Value****Based on DDI 3.2**

r.ValueType

**Definition**

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

**Properties****content**

Datatype	xs:string
Cardinality	1..1
The actual content of this value as a string	

## whiteSpace

Datatype	WhiteSpace
Cardinality	0..1
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".	

## VideoSegment

### Based on DDI 3.2

r:VideoType

### Definition

Describes the type and length of the video segment.

### Properties

#### typeOfVideoClip

Datatype	CodeValueType
Cardinality	0..1
The type of video clip provided. Supports the use of a controlled vocabulary.	

#### videoClipBegin

Datatype	xs:string
Cardinality	0..1
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

Datatype	xs:string
Cardinality	0..1
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.	

## XMLPrefixMap

### Based on DDI 3.2

pr:XMLPrefixMapType

## Definition

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

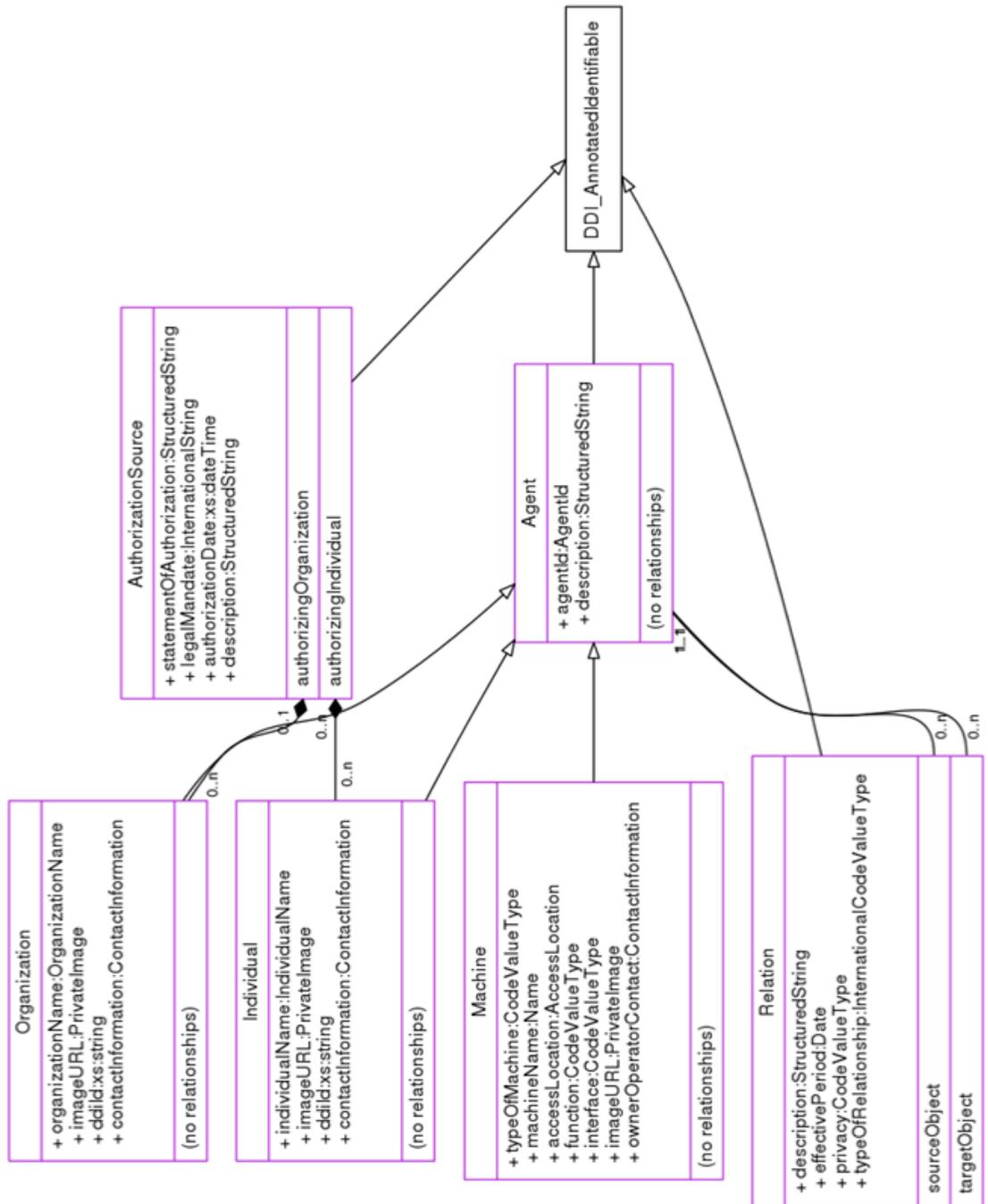
### xmlPrefix

Datatype	xs:string
Cardinality	0..1
Specify the exact prefix used.	

### xmlNamespace

Datatype	xs:string
Cardinality	0..1
Specify the namespace which the prefix represents.	

# Chapter 4. Agents (Full Review)



## Agent

## Extends

This class extends AnnotatedIdentifiable

## Abstract

This class is abstract

**Table 4.1. RDF Mapping (Agent)**

Label	Type	URI
Agent	skos:closeMatch	http://xmlns.com/foaf/0.1/Agent
Agent	skos:closeMatch	http://www.w3.org/ns/prov#Agent

## Definition

An actor that performs a role in relation to a process.

## Example

Analyst performing edits on data, interviewer conducting an interview, a relational database management system managing data, organization publishing data on a regular basis.

## Explanatory Notes

foaf:Agent is: An agent (eg. person, group, software or physical artifact) prov:Agent is An agent is something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity.

## Properties

### agentId

Datatype	AgentId
Cardinality	0..n
An identifier within a specified system for specifying an agent	

### description

Datatype	StructuredString
Cardinality	0..1
Multilingual description allowing for internal formatting using XHTML tags.	

## AuthorizationSource

### Extends

This class extends AnnotatedIdentifiable

### Based on DDI 3.2

r:AuthorizationSourceType

## Definition

Identifies the authorizing agency and allows for the full text of the authorization (law, regulation, or other form of authorization).

## Example

May be used to list authorizations from oversight committees and other regulatory agencies.

## Properties

### statementOfAuthorization

Datatype	StructuredString
Cardinality	0..1
Text of the authorization (law, mandate, approved business case).	

### legalMandate

Datatype	InternationalString
Cardinality	0..1
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.	

### authorizationDate

Datatype	xs:dateTime
Cardinality	0..1
Identifies the date of Authorization.	

### description

Datatype	StructuredString
Cardinality	0..1

## Relationships

### authorizingOrganization

Target	Organization
Type	Composition
Source Cardinality	0..1
Target Cardinality	0..n

References the authorizing agency described as an organization
--

## authorizingIndividual

Target	Individual
Type	Composition
Source Cardinality	0..n
Target Cardinality	0..n
References the authorizing individual.	

# Individual

## Extends

This class extends Agent

## Definition

A person who acts, or is designated to act towards a specific purpose.

## Example

Analyst performing edits on data, interviewer conducting an interview.

## Properties

### individualName

Datatype	IndividualName
Cardinality	1..n
The name of an individual broken out into its component parts of prefix, first/given name, middle name, last/family/surname, and suffix.	

### imageURL

Datatype	PrivateImage
Cardinality	0..n
The URL of an image of the individual.	

### ddild

Datatype	xs:string
Cardinality	0..n

The agency identifier of the individual according to the DDI Alliance agent registry.

## contactInformation

Datatype	ContactInformation
Cardinality	0..1

# Machine

## Extends

This class extends Agent

## Definition

Mechanism or computer program used to implement a process.

## Example

SAS program, photocopier

## Properties

### typeOfMachine

Datatype	CodeValueType
Cardinality	0..1
The kind of machine used - software, web service, physical machine, from a controlled vocabulary	

### machineName

Datatype	Name
Cardinality	0..1
The name of the machine	

### accessLocation

Datatype	AccessLocation
Cardinality	0..1
The locations where the machine can be access	

### function

Datatype	CodeValueType
Cardinality	0..n

The function of the machine
-----------------------------

**interface**

Datatype	CodeValueType
Cardinality	0..n

**imageURL**

Datatype	PrivateImage
Cardinality	0..n

**ownerOperatorContact**

Datatype	ContactInformation
Cardinality	0..1

**Organization****Extends**

This class extends Agent

**Table 4.2. RDF Mapping (Organization)**

Label	Type	URI
Organization	skos:closeMatch	<a href="http://www.w3.org/ns/org#Organization">http://www.w3.org/ns/org#Organization</a>

**Definition**

A framework of authority designated to act toward some purpose.

**Example**

U.S. Census Bureau, University of Michigan/ISR, Norwegian Social Data Archive

**Explanatory Notes**

related to org:Organization which is described as "Represents a collection of people organized together into a community or other..."

**Properties****organizationName**

Datatype	OrganizationName
Cardinality	1..n

Names by which the organization is known.
---

## imageURL

Datatype	PrivateImage
Cardinality	0..n
The URL of an image of the organization.	

## ddild

Datatype	xs:string
Cardinality	0..n
The agency identifier of the organization as registered at the DDI Alliance register.	

## contactInformation

Datatype	ContactInformation
Cardinality	0..1

# Relation

## Extends

This class extends AnnotatedIdentifiable

## Based on DDI 3.2

a:RelationType

## Definition

Describes the relationship between any two organizations or individuals, or an individual and an organization. This is a pairwise relationship and relationships may be unidirectional. Identifies the Source organization or individual and the Target organization or individual, describes the relationship, provides a keyword to classify the relationship, provides and effective period for the relationship, allows for addition information to be provided, and can contain a privacy specification.

## Example

An individual is an employee of an organization.

## Properties

### description

Datatype	StructuredString
Cardinality	0..1

A description of the relationship. May be expressed in multiple languages and supports the use of structured content.

## effectivePeriod

Datatype	Date
Cardinality	0..n
Time period during which this relationship is valid.	

## privacy

Datatype	CodeValueType
Cardinality	0..1
Specifies the level of privacy for the relationship specification as public, restricted, or private. Supports the use of an external controlled vocabulary.	

## typeOfRelationship

Datatype	InternationalCodeValueType
Cardinality	0..1
A brief textual identification of the type of relation. Supports the use of an external controlled vocabulary.	

# Relationships

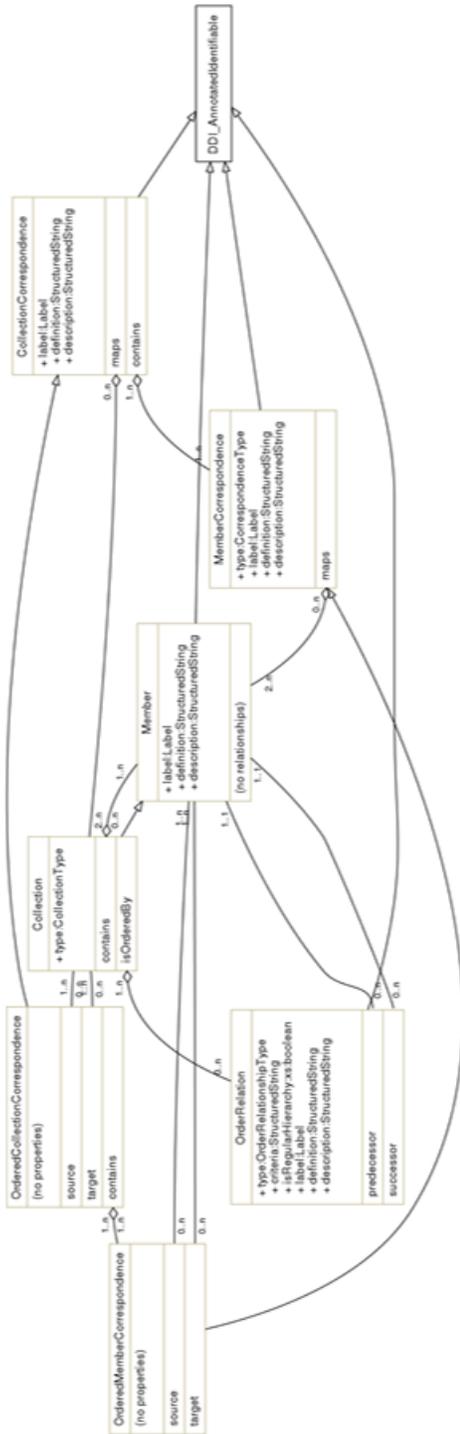
## sourceObject

Target	Agent
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Identifies the Source organization or individual in the relationship. Source to Target provides a directional perception when defining the relationship.	

## targetObject

Target	Agent
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Identifies the Target organization or individual in the relationship. The Target object describes its role in relationship to the Source object.	

# Chapter 5. Collections (Full Review)



# Collection

## Extends

This class extends Member

## Abstract

This class is abstract

## Definition

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.

## Example

Node Sets, Schemes, Groups, Concept Systems are all types of Collections.

## Properties

### type

Datatype	CollectionType
Cardinality	0..1
Whether the collection is a bag or a set.	

## Relationships

### contains

Target	Member
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	1..n
Members of the collection.	

### isOrderedBy

Target	OrderRelation
Type	Aggregation
Source Cardinality	1..n

Target Cardinality	0..n
Total or partial order relation associated with the collection.	

# CollectionCorrespondence

## Extends

This class extends `AnnotatedIdentifiable`

## Abstract

This class is abstract

## Definition

Generic (untyped) relationship between collections.

## Properties

### label

Datatype	Label
Cardinality	0..n
A display label for the <code>CollectionCorrespondence</code> . May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.	

### definition

Datatype	StructuredString
Cardinality	0..1
A definition of the <code>CollectionCorrespondence</code> . May be expressed in multiple languages and supports the use of structured content.	

### description

Datatype	StructuredString
Cardinality	0..1
A description of the purpose or use of the <code>CollectionCorrespondence</code> . May be expressed in multiple languages and supports the use of structured content.	

## Relationships

### maps

Target	Collection
--------	------------

Type	Aggregation
Source Cardinality	0..n
Target Cardinality	2..n
Set of collections that participate in the correspondence. When members of a single collection are mapped, the collection has to appear twice as target.	

## contains

Target	MemberCorrespondence
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	1..n
Correspondences between members of the collections in maps.	

# Member

## Extends

This class extends `AnnotatedIdentifiable`

## Abstract

This class is abstract

## Definition

Generic class representing members of a collection.

## Properties

### label

Datatype	Label
Cardinality	0..n
A display label for the Member. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.	

### definition

Datatype	StructuredString
Cardinality	0..1

A definition of the Member. May be expressed in multiple languages and supports the use of structured content.
--

## description

Datatype	StructuredString
Cardinality	0..1
A description of the purpose or use of the Member. May be expressed in multiple languages and supports the use of structured content.	

# MemberCorrespondence

## Extends

This class extends `AnnotatedIdentifiable`

## Abstract

This class is abstract

## Definition

Generic (untyped) relationship between members of collections.

## Explanatory Notes

Used to define equivalence and similarity between members of collections.

## Properties

### type

Datatype	CorrespondenceType
Cardinality	0..1
Type of correspondence in terms of commonalities and differences between two members.	

### label

Datatype	Label
Cardinality	0..n
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.	

### definition

Datatype	StructuredString
----------	------------------

Cardinality	0..1
A definition of the MemberCorrespondence. May be expressed in multiple languages and supports the use of structured content.	

## description

Datatype	StructuredString
Cardinality	0..1
A description of the purpose or use of the MemberCorrespondence. May be expressed in multiple languages and supports the use of structured content.	

## Relationships

### maps

Target	Member
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	2..n
Set of members that participate in the correspondence.	

## OrderRelation

### Extends

This class extends AnnotatedIdentifiable

### Abstract

This class is abstract

### Definition

Binary relation over members in a collection (set or bag) that is always reflexive, antisymmetric, and transitive. It can also be either total or partial. It must contain like items.

## Properties

### type

Datatype	OrderRelationshipType
Cardinality	0..1
Whether the order relation is total or partial.	

**criteria**

Datatype	StructuredString
Cardinality	0..1
Intensional definition of the order criteria (e.g. alphabetical, numerical, increasing, decreasing, etc.)	

**isRegularHierarchy**

Datatype	xs:boolean
Cardinality	0..1
Indicates whether the tree defined by the order relation is regular or not. i.e., all leaves are at the same level..	

**label**

Datatype	Label
Cardinality	0..n
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.	

**definition**

Datatype	StructuredString
Cardinality	0..1
A definition of the OrderRelation. May be expressed in multiple languages and supports the use of structured content.	

**description**

Datatype	StructuredString
Cardinality	0..1
A description of the purpose or use of the OrderRelation. May be expressed in multiple languages and supports the use of structured content.	

**Relationships****predecessor**

Target	Member
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Member that appears before the successor according to the type of order defined by the relation.	

**successor**

Target	Member
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Member that appears after the predecessor according to the type of order defined by the relation.	

**OrderedCollectionCorrespondence****Extends**

This class extends `CollectionCorrespondence`

**Corresponds to GSIM**

Correspondence Table

**Definition**

Generic (untyped) ordered relationship between collections.

**Relationships****source**

Target	Collection
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..n
The collection(s) from which the correspondence is made.	

**target**

Target	Collection
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..n
The collection(s) to which the correspondence is directed. It could be the same as the source when mapping members of a single collection.	

**contains**

Target	OrderedMemberCorrespondence
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	1..n
Ordered correspondences between members of collections in source and target.	

**OrderedMemberCorrespondence****Extends**

This class extends MemberCorrespondence

**Abstract**

This class is abstract

**Definition**

Ordered relationship between members of collections.

**Example**

Used to define an order of precedence between members of collections.

**Relationships****source**

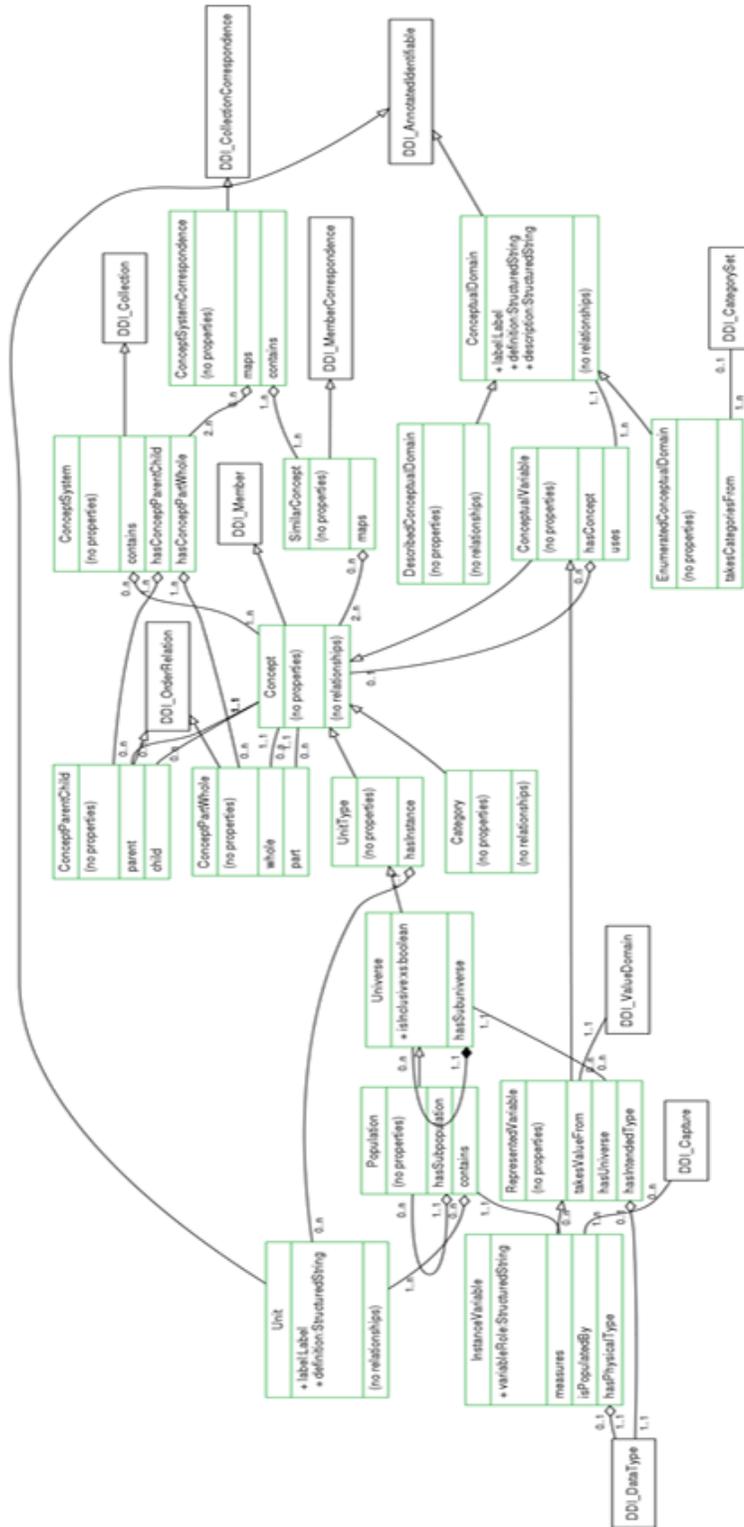
Target	Member
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..n
Source members of the correspondence.	

**target**

Target	Member
Type	Neither
Source Cardinality	0..n

Target Cardinality	1..n
Target members of the correspondence.	

# Chapter 6. Conceptual (Full Review)



# Category

## Extends

This class extends Concept

## Based on DDI 3.2

l:CategoryType

## Definition

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

# Concept

## Extends

This class extends Member

## Based on DDI 3.2

c:ConceptType

**Table 6.1. RDF Mapping (Concept)**

Label	Type	URI
Concept	skos:closeMatch	<a href="http://www.w3.org/2004/02/skos/core#Concept">http://www.w3.org/2004/02/skos/core#Concept</a>

## Corresponds to GSIM

Concept

## Definition

Unit of thought differentiated by characteristics [GSIM 1.1]

# ConceptParentChild

## Extends

This class extends OrderRelation

## Definition

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

## Relationships

### parent

Target	Concept
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Specialization of "predecessor" in OrderRelation	

### child

Target	Concept
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Specialization of "successor" in OrderRelation	

## ConceptPartWhole

### Extends

This class extends OrderRelation

### Definition

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

## Relationships

### whole

Target	Concept
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Specialization of "predecessor" in OrderRelation.	

**part**

Target	Concept
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Specialization of "successor" in OrderRelation.	

**ConceptSystem****Extends**

This class extends Collection

**Abstract**

This class is abstract

**Based on DDI 3.2**

c:ConceptSchemeType

**Corresponds to GSIM**

Concept System

**Definition**

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

**Example**

Here are 2 examples 1) Concept of Sex: Male, Female, Other 2) ISIC (the list is too long to write down)

**Relationships****contains**

Target	Concept
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	1..n
The relationship to the concepts contained in the concept system	

## hasConceptParentChild

Target	ConceptParentChild
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	0..n
Specialization of isOrderedBy in Collection.	

## hasConceptPartWhole

Target	ConceptPartWhole
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	0..n
Specialization of isOrderedBy in Collection.	

# ConceptSystemCorrespondence

## Extends

This class extends CollectionCorrespondence

## Definition

## Relationships

### maps

Target	ConceptSystem
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	2..n
Specialization of maps in CollectionCorrespondence. When Concepts of a single ConceptSystem are mapped, the Concept has to appear twice as target.	

### contains

Target	SimilarConcept
--------	----------------

Type	Aggregation
Source Cardinality	1..n
Target Cardinality	1..n
Specialization of contains in CollectionCorrespondence for mapping similar concepts.	

## ConceptualDomain

### Extends

This class extends AnnotatedIdentifiable

### Corresponds to GSIM

Conceptual Domain

### Definition

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

### Explanatory Notes

[GSIM 1.1]

### Properties

#### label

Datatype	Label
Cardinality	0..n
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.	

#### definition

Datatype	StructuredString
Cardinality	0..1
A description of the content and purpose of the Conceptual Domain. May be expressed in multiple languages and supports the use of structured content.	

#### description

Datatype	StructuredString
Cardinality	0..1

A description of the purpose or use of a concept. May be expressed in multiple languages and supports the use of structured content.

# ConceptualVariable

## Extends

This class extends Concept

## Based on DDI 3.2

c:ConceptualVariableType

## Corresponds to GSIM

Variable

## Definition

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

## Example

Here are 3 examples - 1. Sex of person 2. Number of employees 3. Value of endowment

## Relationships

### hasConcept

Target	Concept
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	0..1
Reference to a Concept that is being used	

### uses

Target	ConceptualDomain
Type	Neither
Source Cardinality	1..n
Target Cardinality	1..1
Identifies the ConceptualDomain containing the set of concepts used to describe the ConceptualVariable.	

# DescribedConceptualDomain

## Extends

This class extends ConceptualDomain

## Corresponds to GSIM

Described Conceptual Domain

## Definition

A Conceptual Domain defined by an expression.

## Example

For example: All real numbers between 0 and 1. [GSIM 1.1]

## Explanatory Notes

[GSIM 1.1]

# EnumeratedConceptualDomain

## Extends

This class extends ConceptualDomain

## Corresponds to GSIM

Enumerated Conceptual Domain

## Definition

A Conceptual Domain expressed as a list of Categories.

## Example

For instance, the Sex Categories: 'Male' and 'Female' [GSIM 1.1]

## Explanatory Notes

[GSIM 1.1]

## Relationships

### takesCategoriesFrom

Target	CategorySet
--------	-------------

Type	Neither
Source Cardinality	1..n
Target Cardinality	0..1

## InstanceVariable

### Extends

This class extends RepresentedVariable

### Corresponds to GSIM

Instance Variable

### Definition

The use of a Represented Variable within a Data Set.

### Example

The InstanceVariable is to describe actual instances of data that have been collected. Here are 3 examples: 1) Gender: Dan Gillman has gender <m, male>, Arofan Gregory has gender<m, male>, etc. 2) Number of employees: Microsoft has 90,000 employees; IBM has 433,000 employees, etc. 3) Endowment: Johns Hopkins has endowment of <3, \$1,000,000 and above>, Yale has endowment of <3, \$1,000,000 and above>, etc.

### Explanatory Notes

It can inherit information from Represented Variable (which again can inherit from Conceptual Variable)

### Properties

#### variableRole

Datatype	StructuredString
Cardinality	0..1
An Instance Variable can take different roles, e.g. Identifier, Measure and Attribute. Note that DataStructure takes care of the ordering of Identifiers.	

### Relationships

#### measures

Target	Population
--------	------------

Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Set of specific units (people, entities, objects, events), usually in a given time and geography, being measured.	

## isPopulatedBy

Target	Capture
Type	Neither
Source Cardinality	1..n
Target Cardinality	0..n
Means of obtaining data for the variable.	

## hasPhysicalType

Target	DataType
Type	Aggregation
Source Cardinality	0..1
Target Cardinality	1..1

# Population

## Extends

This class extends Universe

## Corresponds to GSIM

Population

## Definition

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

## Example

Here are three examples: 1. Canadian adult persons residing in Canada on 13 November 1956 2. US computer companies at the end of 2012 3. Universities in Denmark 1 January 2011.

## Relationships

### hasSubpopulation

Target	Population
Type	Aggregation
Source Cardinality	1..1
Target Cardinality	0..n
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.	

### contains

Target	Unit
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	1..n
Units in the Population	

## RepresentedVariable

### Extends

This class extends ConceptualVariable

### Definition

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

### Example

The pair (Number of Employees, Integer), where "Number of Employees" is the characteristic of the population (Variable) and "Integer" is how that measure will be represented (Value Domain).

## Relationships

### takesValueFrom

Target	ValueDomain
Type	Neither
Source Cardinality	0..n

Target Cardinality	1..1
The representation (Value Domain) of the variable.	

## hasUniverse

Target	Universe
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
The defined class of people, entities, events, or objects to be measured.	

## hasIntendedType

Target	DataType
Type	Aggregation
Source Cardinality	0..1
Target Cardinality	1..1

## SimilarConcept

### Extends

This class extends MemberCorrespondence

### Based on DDI 3.2

c:SimilarConceptType

### Definition

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.

### Relationships

#### maps

Target	Concept
Type	Aggregation
Source Cardinality	0..n

Target Cardinality	2..n
Specialization of maps in MemberCorrespondence for Concepts that are similar. Used to assist in disambiguation of concepts.	

## Unit

### Extends

This class extends AnnotatedIdentifiable

### Corresponds to GSIM

Unit

### Definition

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

### Example

Here are 3 examples - 1. Individual US person (i.e., Arofan Gregory, Dan Gillman, Barack Obama, etc.) 2. Individual US computer companies (i.e., Microsoft, Apple, IBM, etc.) 3. Individual US universities (i.e., Johns Hopkins, University of Maryland, Yale, etc.) [GSIM 1.1]

### Properties

#### label

Datatype	Label
Cardinality	0..n
A display label for the Unit. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.	

#### definition

Datatype	StructuredString
Cardinality	0..1
A description of the content and purpose of the Unit. May be expressed in multiple languages and supports the use of structured content.	

## UnitType

### Extends

This class extends Concept

## Corresponds to GSIM

Unit Type

### Definition

A Unit Type is a class of objects of interest.

### Explanatory Notes

A Unit Type is used to describe a class or group of Units based on a single characteristic with no specification of time and geography. For example, the Unit Type of “Person” groups together a set of Units based on the characteristic that they are ‘Persons’. It concerns not only Unit Types used in dissemination, but anywhere in the statistical process. E.g. using administrative data might involve the use of a fiscal unit. [GSIM 1.1]

### Relationships

#### hasInstance

Target	Unit
Type	Aggregation
Source Cardinality	1..1
Target Cardinality	0..n
Objects of interests that are instances of this UnitType.	

## Universe

### Extends

This class extends UnitType

### Based on DDI 3.2

c:UniverseType

## Corresponds to GSIM

Population

### Definition

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

## Example

Here are three examples: 1. Canadian adults (not limited to those residing in Canada) 2. Computer companies 3. Universities

## Explanatory Notes

If the Universe consist of people a number of factors may be used in defining membership in the Universe, such as age, sex, race, residence, income, veteran status, criminal convictions, etc. The universe may consist of elements other than persons, such as housing units, court cases, deaths, countries, etc. A universe may be described as "inclusive" or "exclusive". Not to be confused with Population, which includes the specification of time and geography.

## Properties

### isInclusive

Datatype	xs:boolean
Cardinality	0..1
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".	

## Relationships

### hasSubuniverse

Target	Universe
Type	Composition
Source Cardinality	1..1
Target Cardinality	0..n
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	



## Definition

An Act is a type of ControlConstruct. An Act has many subtypes including an Instruction, a Question, an Instrument and a StudyUnit. Both Acts and ControlConstructs are triggered when the conditions of a ControlConstruct are met.

## Example

QuestionConstructType is a type of Act.

## Binding

### Based on DDI 3.2

r:BindingType

## Definition

A structure used to bind the content of a parameter declared as the source to a parameter declared as the target. For example, binding the output of a question to the input of a generation instruction. Question A has an OutParameter X. Generation Instruction has an InParameter Y used in the recode instruction. Binding defines the content of InParameter Y to be whatever is provided by OutParameter X for use in the calculation of the recode.

## Properties

### sourceParameter

Datatype	xs:string
Cardinality	1..1
A structure used to bind the content of a parameter declared as the source to a parameter declared as the target. For example, binding the output of a question to the input of a generation instruction. Question A has an OutParameter X. Generation Instruction has an InParameter Y used in the recode instruction. Binding defines the content of InParameter Y to be whatever is provided by OutParameter X for use in the calculation of the recode.[Referenced object not explicit]	

### targetParameter

Datatype	xs:string
Cardinality	1..1
A structure used to bind the content of a parameter declared as the source to a parameter declared as the target. For example, binding the output of a question to the input of a generation instruction. Question A has an OutParameter X. Generation Instruction has an InParameter Y used in the recode instruction. Binding defines the content of InParameter Y to be whatever is provided by OutParameter X for use in the calculation of the recode.[Referenced object not explicit]	

# ContainsTemporalRelation

## Extends

This class extends TemporalRelation

## Definition

## Relationships

### contains

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
A contains B (D)	

### during

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
B is during A (d)	

# ControlConstruct

## Extends

This class extends ProcessStep

## Abstract

This class is abstract

## Based on DDI 3.2

d:ControlConstructType

## Corresponds to GSIM

Process Control

## Definition

A ControlConstruct is used in the definition of the sequence of execution of process steps.

## Example

ControlConstructs, such as IfThenElse and While, could be used to define a sequence of questions in a questionnaire.

## EqualTemporalRelation

### Extends

This class extends TemporalRelation

## Definition

## Relationships

### equalA

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
A is coextensive with B	

### equalB

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
B is coextensive with A	

## FinishesTemporalRelation

### Extends

This class extends TemporalRelation

## Definition

## Relationships

### finishedBy

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
A is finished by B (F)	

### finishes

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
B finishes A (f)	

## IfThenElse

### Extends

This class extends ControlConstruct

### Based on DDI 3.2

d.IfThenElseType

### Definition

IfThenElse describes an if-then-else decision type of control construct. IF the stated condition is met, the THEN clause is triggered, otherwise the ELSE clause is triggered.

### Example

(add example)

### Explanatory Notes

Contains an IfCondition (the condition that must be met to trigger the Then clause), a ThenConstructReference (indicating the construct to invoke if the condition is met), an

ElseConstructReference (indicating the construct to invoke if the condition is not met), and an ElseIf structure allowing the expression of multiple conditions to invoke multiple branching. The command to be executed if the ifCondition is true is assumed to be represented by the inherited relationship between ControlConstruct and ProcessStep.

## Properties

### ifCondition

Datatype	CommandCode
Cardinality	1..1
The condition which must be met to trigger the Then clause, expressed as a CommandCode. The condition is an expression in the programming language used in the instrument.	

## Relationships

### hasElse

Target	ControlConstruct
Type	Composition
Source Cardinality	0..1
Target Cardinality	1..1
The ControlConstruct to be triggered if ifCondition is false.	

### hasThen

Target	ControlConstruct
Type	Composition
Source Cardinality	1..1
Target Cardinality	1..1
The ControlConstruct to be triggered if ifCondition is true.	

## Input

### Extends

This class extends Parameter

### Definition

Input to a process step, either a type of an object or an instance.

## Example

Input variables for imputation.

## Properties

### limitArrayIndex

Datatype	xs:NMTOKENS
Cardinality	0..1
When the Input represents an array of items, this attribute specifies the index identification of the items within the zero-based array which should be treated as input parameters. If not specified, the full array is treated as the input parameter.	

## Loop

### Extends

This class extends ControlConstruct

### Based on DDI 3.2

d:LoopType

### Definition

Describes an action which loops until a limiting condition is met.

### Explanatory Notes

The ControlConstruct contained in the Loop operates on the LoopVariable until the LoopWhile condition is met, and then control is handed back to the containing control construct.

## Properties

### initialValue

Datatype	xs:integer
Cardinality	0..1
The command used to set the initial value for the process. Could be a simple value.	

### loopWhile

Datatype	CommandCode
Cardinality	0..1

The command used to determine whether the "LoopWhile" condition is met.
---

## stepValue

Datatype	xs:integer
Cardinality	0..1
The command used to set the incremental or step value for the process. Could be a simple value.	

## Relationships

### hasLoop

Target	ControlConstruct
Type	Composition
Source Cardinality	1..1
Target Cardinality	1..1
The ControlConstruct to be repeated if loopWhile is true.	

## MeetsTemporalRelation

### Extends

This class extends TemporalRelation

### Definition

## Relationships

### meets

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
A meets B (m)	

### isMetBy

Target	ProcessStep
--------	-------------

Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
B is met by A (M)	

## Output

### Extends

This class extends Parameter

### Definition

Output to a process step, either a type of an object or an instance.

### Example

Imputed variable

### Explanatory Notes

[The actual use could be clarified, e.g. link to the relevant document. ]

## OverlapsTemporalRelation

### Extends

This class extends TemporalRelation

### Definition

### Relationships

#### overlaps

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
A overlaps B (o)	

**isOverlappedBy**

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
B is overlapped by A (O)	

**PredecessorTemporalRelation****Extends**

This class extends TemporalRelation

**Definition****Relationships****predecessor**

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
A precedes B. A ends before B begins.	

**successor**

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
B is preceded by A. B begins after A ends.	

**ProcessStep****Extends**

This class extends AnnotatedIdentifiable

## Abstract

This class is abstract

## Corresponds to GSIM

Process Step

## Definition

Work package performed by a service to transform inputs to outputs considering rules as defined in the control construct.

## Example

Imputation of a survey variable for missing data values.

## Explanatory Notes

Substeps can be expressed using Sequence. Sequence is a container of process steps. The order of a Sequence is temporal. Each sequence has zero or more TemporalRelations. A TemporalRelation specifies a temporal order between two process steps. The "sum" of these TemporalRelations defines the order of a Sequence.

## Properties

### label

Datatype	Label
Cardinality	0..1

### description

Datatype	StructuredString
Cardinality	0..1

### definition

Datatype	StructuredString
Cardinality	0..1

### binding

Datatype	Binding
Cardinality	0..1

A structure used to bind the content of a parameter declared as the source to a parameter declared as the target. The binding may be defined as part of the ProcessStep in which case it is called "in line". The binding may also be defined by a ProcessStep user such as an Instance Variable which might get its value from the ProcessStep. In this case we say the ProcessStep or a set of ProcessSteps that form a processing pipeline is reusable and "declarative" only.

## Relationships

### isPerformedBy

Target	Service
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1

### hasInput

Target	Input
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	1..1

### hasOutput

Target	Output
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	1..1

## RepeatUntil

### Extends

This class extends ControlConstruct

### Based on DDI 3.2

d:RepeatUntilType

## Definition

Specifies a ControlConstruct to be repeated until a specified condition is met. Before each iteration the condition is tested. When the condition is met, control passes back to the containing control construct.

## Properties

### untilCondition

Datatype	CommandCode
Cardinality	1..1
Information on the command used to determine whether the "Until" condition is met.	

## Relationships

### hasUntil

Target	ControlConstruct
Type	Composition
Source Cardinality	1..1
Target Cardinality	1..1
The ControlConstruct to be repeated if untilCondition is not met.	

## RepeatWhile

### Extends

This class extends ControlConstruct

### Based on DDI 3.2

d:RepeatWhileType

## Definition

Specifies a ControlConstruct to be repeated while a specified condition is met. Before each iteration the condition is tested. When the condition is not met, control passes back to the containing control construct.

## Properties

### whileCondition

Datatype	CommandCode
Cardinality	1..1

Information on the command used to determine whether the "While" condition is met.
--

## Relationships

### hasWhile

Target	ControlConstruct
Type	Composition
Source Cardinality	1..1
Target Cardinality	1..1
The ControlConstruct to be repeated if whileCondition is true.	

## Sequence

### Extends

This class extends ControlConstruct

### Based on DDI 3.2

d:SequenceType

### Definition

Provides a sequence order for operations expressed as control constructs. The sequence can be typed to support local processing or classification flags and alternate sequencing instructions (such as randomize for each respondent).

## Properties

### typeOfSequence

Datatype	CodeValueType
Cardinality	0..n
Provides the ability to "type" a sequence for classification or processing purposes. Supports the use of an external controlled vocabulary.	

### constructSequence

Datatype	SpecificSequence
Cardinality	0..1
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.	

## Relationships

### isOrderedBy

Target	TemporalRelation
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	0..n

## Service

### Extends

This class extends ProcessStep

### Corresponds to GSIM

Business Service

### Definition

A means of performing 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). (source: GSIM)

## Properties

### interface

Datatype	CodeValueType
Cardinality	0..1
Specifies how to communicate with the service.	

### location

Datatype	CodeValueType
Cardinality	0..1
Specifies where the service can be accessed.	

## Relationships

### hasAgent

Target	Agent
--------	-------

Type	Neither
Source Cardinality	0..n
Target Cardinality	0..n

## StartsTemporalRelation

### Extends

This class extends TemporalRelation

### Definition

### Relationships

#### starts

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
A starts B (s)	

#### isStartedBy

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
B is started by A (S)	

## TemporalRelation

### Definition

### Explanatory Notes

Sequence is a container of process steps. The order of a Sequence is temporal. Each sequence has zero or more TemporalRelations. A TemporalRelation specifies a temporal order between two process steps.

The "sum" of these TemporalRelations defines the order of a Sequence. Between any two process steps seven temporal relationships are possible based on [Allen's interval algebra](http://www.ics.uci.edu/~alspaugh/cls/shr/allen.html). For example, between two process steps A might precede B and B might be preceded by A in which case A end before B begins and B begins after A ends. However, in this event A and B never meet. If A were to meet B and B was to be met by A then A would end at exactly the instant B begins with no gap between them and B would begin exactly when A ends with no gap between them. Altogether between A and B in Allen's interval algebra there are 7 possible relationships: A precedes B (p) and B is preceded by A (P) A meets B (m) and B is met by A (M) A overlaps B (o) and B is overlapped by A (O) A is finished by B (F) and B finishes A (f) A contains B (D) and B is during A (d) A starts B (s) and B is started by A (S) A and B equal (e) each other The set of TemporalRelations between pairs of process steps in a Sequence are assumed to be consistent. Although consistency across an Allen set of TemporalRelations between pairs of process steps can be defined, no constraints are provided here. Instead the burden of being consistent is placed on the user. That being said, we assist the user here with a TemporalRelation that can be used with less thought. This is a fuzzy Allen relation that can be used to order pairs of process steps and achieve consistency perhaps with less thought. Thus, It is possible to skirt the complexity of these temporal relationships and speak in fuzzy terms about the temporal relationship that obtains between A and B. In fuzzy terms there might a predecessor / successor relationship in which we "overlook" and don't specify if between the end of A and the start of B there is a gap. This fuzzy predecessor/successor relationship pair has been added and is designated: A precedes B (pf) and B is preceded by A (Pf) Using this fuzzy TemporalRelation wi might define a complete order across three process steps A, B and C in a Sequence as follows: A precedes B (pf) and B is preceded by A (Pf) B precedes C (pf) and C is preceded by B (Pf) Because transitivity applies, we need not specify: A precedes C (pf) and C is preceded by A (Pf).

## Relationships

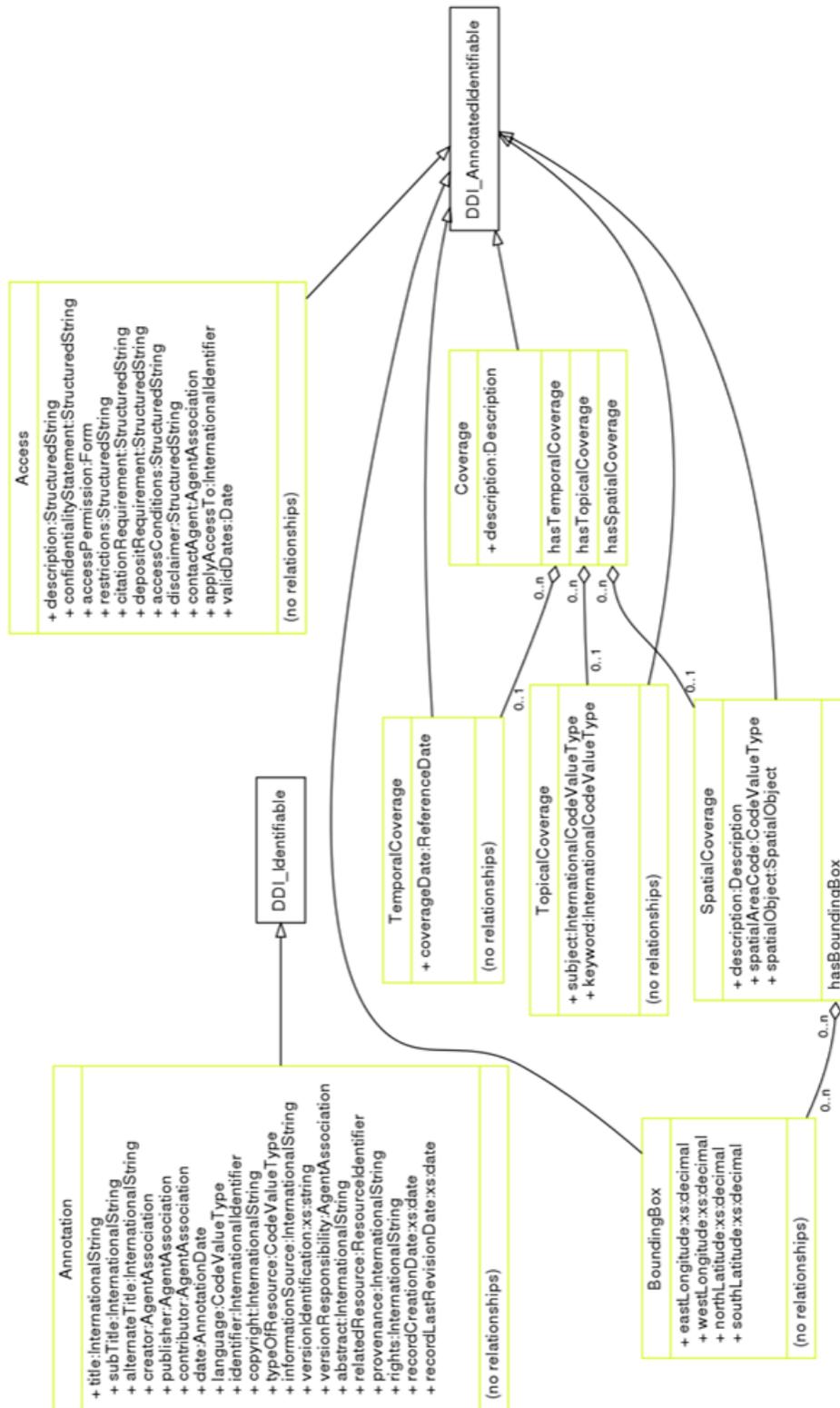
### fuzzyPredecessor

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
A precedes B whether or not B begins before A ends	

### fuzzySuccessor

Target	ProcessStep
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
B succeeds A whether or not A ends before B begins	

# Chapter 8. Discovery (Full Review)



# Access

## Extends

This class extends `AnnotatedIdentifiable`

## Based on DDI 3.2

a:AccessType

## Definition

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.

## Properties

### description

Datatype	StructuredString
Cardinality	0..1
A description of the content and purpose of the access description. May be expressed in multiple languages and supports the use of structured content.	

### confidentialityStatement

Datatype	StructuredString
Cardinality	0..1
A statement regarding the confidentiality of the related data or metadata.	

### accessPermission

Datatype	Form
Cardinality	0..n
A link to a form used to provide access to the data or metadata including a statement of the purpose of the form.	

### restrictions

Datatype	StructuredString
Cardinality	0..1
A statement regarding restrictions to access. May be expressed in multiple languages and supports the use of structured content.	

**citationRequirement**

Datatype	StructuredString
Cardinality	0..1
A statement regarding the citation requirement. May be expressed in multiple languages and supports the use of structured content.	

**depositRequirement**

Datatype	StructuredString
Cardinality	0..1
A statement regarding depositor requirements. May be expressed in multiple languages and supports the use of structured content.	

**accessConditions**

Datatype	StructuredString
Cardinality	0..1
A statement regarding conditions for access. May be expressed in multiple languages and supports the use of structured content.	

**disclaimer**

Datatype	StructuredString
Cardinality	0..1
A disclaimer regarding the liability of the data producers or providers. May be expressed in multiple languages and supports the use of structured content.	

**contactAgent**

Datatype	AgentAssociation
Cardinality	0..n
The agent to contact regarding access including the role of the agent.	

**applyAccessTo**

Datatype	InternationalIdentifier
Cardinality	0..n
Identification for an object covered by the access description. This may be any annotated object (collection, publication, identifiable object).	

**validDates**

Datatype	Date
Cardinality	0..1
The date range or start date of the access description.	

# Annotation

## Extends

This class extends Identifiable

## Based on DDI 3.2

r:CitationType

## Definition

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

## Properties

### title

Datatype	InternationalString
Cardinality	0..1
Full authoritative title. List any additional titles for this item as AlternativeTitle.	

### subTitle

Datatype	InternationalString
Cardinality	0..n
Secondary or explanatory title.	

### alternateTitle

Datatype	InternationalString
Cardinality	0..n
An alternative title by which a data collection is commonly referred, or an abbreviation for the title.	

### creator

Datatype	AgentAssociation
Cardinality	0..n
Person, corporate body, or agency responsible for the substantive and intellectual content of the described object.	

### publisher

Datatype	AgentAssociation
Cardinality	0..n

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

**contributor**

Datatype	AgentAssociation
Cardinality	0..n
The name of a contributing author or creator, who worked in support of the primary creator given above.	

**date**

Datatype	AnnotationDate
Cardinality	0..n
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.	

**language**

Datatype	CodeValueType
Cardinality	0..n
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.	

**identifier**

Datatype	InternationalIdentifier
Cardinality	0..n
An identifier or locator. Contains identifier and Managing agency (ISBN, ISSN, DOI, local archive). Indicates if it is a URI.	

**copyright**

Datatype	InternationalString
Cardinality	0..n
The copyright statement.	

**typeOfResource**

Datatype	CodeValueType
Cardinality	0..n
Provide the type of the resource. This supports the use of a controlled vocabulary. It should be appropriate to the level of the annotation.	

**informationSource**

Datatype	InternationalString
----------	---------------------

Cardinality	0..n
The name or identifier of source information for the annotated object.	

## versionIdentification

Datatype	xs:string
Cardinality	0..1
Means of identifying the current version of the annotated object.	

## versionResponsibility

Datatype	AgentAssociation
Cardinality	0..n
The agent responsible for the version. May have an associated role.	

## abstract

Datatype	InternationalString
Cardinality	0..1
An a abstract (description) of the annotated object.	

## relatedResource

Datatype	ResourceIdentifier
Cardinality	0..n
Provide the identifier, managing agency, and type of resource related to This class.	

## provenance

Datatype	InternationalString
Cardinality	0..n
A statement of any changes in ownership and custody of the resource since its creation that are significant for its authenticity, integrity, and interpretation.	

## rights

Datatype	InternationalString
Cardinality	0..n
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.	

## recordCreationDate

Datatype	xs:date
Cardinality	0..1

Date the record was created
-----------------------------

## recordLastRevisionDate

Datatype	xs:date
Cardinality	0..1
Date the record was last revised	

# BoundingBox

## Extends

This class extends AnnotatedIdentifiable

## Definition

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.

## Properties

### eastLongitude

Datatype	xs:decimal
Cardinality	1..1
The easternmost coordinate expressed as a decimal between the values of -180 and 180 degrees	

### westLongitude

Datatype	xs:decimal
Cardinality	1..1
The westernmost coordinate expressed as a decimal between the values of -180 and 180 degrees	

### northLatitude

Datatype	xs:decimal
Cardinality	1..1
The northernmost coordinate expressed as a decimal between the values of -90 and 90 degrees.	

### southLatitude

Datatype	xs:decimal
Cardinality	1..1
The southernmost latitude expressed as a decimal between the values of -90 and 90 degrees	

# Coverage

## Extends

This class extends `AnnotatedIdentifiable`

## Based on DDI 3.2

r:CoverageType

## Definition

Coverage information for an annotated object. Includes coverage information for temporal, topical, and spatial coverage.

## Properties

### description

Datatype	Description
Cardinality	0..n
A generic description including temporal, topical, and spatial coverage that is the equivalent of <code>dc:coverage</code> (the refinement base of <code>dcterms:spatial</code> and <code>dcterms:temporal</code> . Use specific coverage content for detailed information.	

## Relationships

### hasTemporalCoverage

Target	TemporalCoverage
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	0..1
The dates and time periods described by the contents of the annotated object.	

### hasTopicalCoverage

Target	TopicalCoverage
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	0..1

The topics covered by the contents of the annotated object. These may be expressed by subject classification systems and structured or unstructured keywords.

## hasSpatialCoverage

Target	SpatialCoverage
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	0..1
Description of the spatial (geographic) coverage of the contents of the annotated object.	

## SpatialCoverage

### Extends

This class extends AnnotatedIdentifiable

**Table 8.1. RDF Mapping (SpatialCoverage)**

Label	Type	URI
description	skos:related	<a href="http://purl.org/dc/terms/spatial">http://purl.org/dc/terms/spatial</a>

### Definition

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).

### Example

A country, a neighborhood, the inside of a polygon on the surface of the earth, along a street, at a particular intersection, or perhaps even in a certain orbit around the planet Mars.

### Properties

#### description

Datatype	Description
Cardinality	0..1
A textual description of the spatial coverage to support general searches.	

#### spatialAreaCode

Datatype	CodeValueType
----------	---------------

Cardinality	0..n
Supports the use of a standardized code such as ISO 3166-1, the Getty Thesaurus of Geographic Names, FIPS-5, etc.	

## spatialObject

Datatype	SpatialObject
Cardinality	0..1
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

### hasBoundingBox

Target	BoundingBox
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	0..n
The north and south latitudes and east and west longitudes that define the spatial coverage area.	

## TemporalCoverage

### Extends

This class extends AnnotatedIdentifiable

### Definition

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.

### Properties

#### coverageDate

Datatype	ReferenceDate
Cardinality	0..n
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).	

# TopicalCoverage

## Extends

This class extends `AnnotatedIdentifiable`

## Based on DDI 3.2

`r:TopicalCoverageType`

**Table 8.2. RDF Mapping (TopicalCoverage)**

Label	Type	URI
subject	owl:sameAs	<a href="http://purl.org/dc/terms/subject">http://purl.org/dc/terms/subject</a>
keyword	owl:sameAs	<a href="http://purl.org/dc/terms/subject">http://purl.org/dc/terms/subject</a>

## Definition

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.

## Properties

### subject

Datatype	<code>InternationalCodeValueType</code>
Cardinality	0..n
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 <code>InternationalCodeValue</code> and may indicate the language of the code used.	

### keyword

Datatype	<code>InternationalCodeValueType</code>
Cardinality	0..n
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 <code>InternationalCodeValue</code> and may indicate the language of the code used.	

---

# Chapter 9. Identification (Full Review)



## AnnotatedIdentifiable

### Extends

This class extends Identifiable

### Abstract

This class is abstract

### Definition

Used to identify objects for purposes of internal and/or external referencing. Elements of this type are versioned. Provides administrative metadata about the object in addition to what is provided

by Identifiable, including more details on the versioning of the object. Most objects except for the ComplexDataTypes will inherit AnnotatedIdentifiable.

## Properties

### versionResponsibility

Datatype	xs:string
Cardinality	0..1
Contributor who has the ownership and responsibility for the current version.	

### versionRationale

Datatype	xs:string
Cardinality	0..1
The reason for making this version of the object.	

### versionDate

Datatype	xs:dateTime
Cardinality	0..1
The date and time the object was changed.	

### isUniversallyUnique

Datatype	xs:boolean
Cardinality	1..1
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. Default value is false.	

### isPersistent

Datatype	xs:boolean
Cardinality	1..1
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 there will be no more changes to the current version. Default value is false.	

### localId

Datatype	LocalId
Cardinality	0..n
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.	

## basedOnObject

Datatype	BasedOnObject
Cardinality	0..1
The object/version that This class version is based on.	

## Relationships

### hasAnnotation

Target	Annotation
Type	Aggregation
Source Cardinality	0..1
Target Cardinality	1..n
Provides annotation information on the object to support citation and crediting of the creator(s) of the object.	

## Identifiable

### Abstract

This class is abstract

### Definition

Used to identify objects for purposes of internal and/or external referencing. Elements of this type are versioned. Most objects except for the ComplexDataTypes will inherit from Identifiable or the more specialised AnnotatedIdentifiable.

## Properties

### agency

Datatype	xs:string
Cardinality	1..1
This is the registered agency code with optional sub-agencies separated by dots. For example, diw.soep, ucl.qss, abs.essg.	

### id

Datatype	xs:string
Cardinality	1..1
The ID of the object. This must conform to the allowed structure of the DDI Identifier and must be unique within the declared scope of uniqueness (Agency or Maintainable).	

## version

Datatype	xs:string
Cardinality	1..1
The version number of the object. The version number is incremented whenever the non-administrative metadata contained by the object changes.	



## Based on DDI 3.2

l:CategorySchemeType

## Corresponds to GSIM

Category Set

## Definition

A Category Set is a type of Node Set which groups Categories.

## Example

For example: Male, Female

## Explanatory Notes

The Categories in a Category Set typically have no assigned Designations (Codes).

## Relationships

### hasCategory

Target	Category
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	1..n
Specialization of contains in NodeSet for Categories.	

## ClassificationFamily

### Extends

This class extends Collection

## Corresponds to GSIM

Classification Family

## Definition

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]

## Explanatory Notes

Different classification databases may use different types of Classification Families and have different names for the families, as no standard has been agreed upon. [GSIM1.1]

## Relationships

has

Target	ClassificationIndex
Type	Neither
Source Cardinality	0..n
Target Cardinality	0..n
ClassificationIndexes associated to the ClassificationFamily.	

groups

Target	ClassificationSeries
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	0..1
Specialization of contains in Collection	

## ClassificationIndex

### Extends

This class extends Collection

### Corresponds to GSIM

Classification Index

### Definition

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]

## Explanatory Notes

A Classification Index shows the relationship between text found in statistical data sources (responses to survey questionnaires, administrative records) and one or more Statistical Classifications. A Classification

Index may be used to assign the codes for Classification Items to observations in statistical collections. A Statistical Classification is a subtype of Node Set. The relationship between Statistical Classification and Classification Index can also be extended to include the other Node Set types - Code List and Category Set. [GSIM Statistical Classification Model]

## Properties

### releaseDate

Datatype	Date
Cardinality	0..1
Date when the current version of the Classification Index was released.	

### maintenanceUnit

Datatype	InternationalString
Cardinality	0..1
The unit or group of persons within the organisation responsible for the Classification Index, i.e. for adding, changing or deleting Classification Index Entries.	

### contactPersons

Datatype	InternationalString
Cardinality	0..1
Person(s) who may be contacted for additional information about the Classification Index.	

### publications

Datatype	InternationalString
Cardinality	0..n
A list of the publications in which the Classification Index has been published.	

### languages

Datatype	InternationalString
Cardinality	0..n
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.	

### corrections

Datatype	InternationalString
Cardinality	0..n
Verbal summary description of corrections, which have occurred within the Classification Index. Corrections include changing the item code associated with an Classification Index Entry.	

## codingInstructions

Datatype	InternationalString
Cardinality	0..n
Additional information which drives the coding process for all entries in a Classification Index.	

## Relationships

### isOrderedBy

Target	IndexOrder
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	1..n
Specialization of isOrderedBy in Collection.	

### groups

Target	ClassificationIndexEntry
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	1..n
Specialization of contains in Collection	

## ClassificationIndexEntry

### Extends

This class extends Member

### Corresponds to GSIM

Classification Index Entry

### Definition

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.

## Explanatory Notes

A Classification Item is a subtype of Node. The relationship between Classification Item and Classification Index Entry can also be extended to include the other Node types - Code Item and Category Item.

## Properties

### text

Datatype	InternationalString
Cardinality	1..n
Text describing the type of object/unit or object property.	

### validfrom

Datatype	Date
Cardinality	0..1
Date from which the Classification Index Entry became valid. The date must be defined if the Classification Index Entry belongs to a floating Classification Index.	

### validto

Datatype	Date
Cardinality	0..1
Date at which the Classification Index Entry became invalid. The date must be defined if the Classification Index Entry belongs to a floating Classification Index and is no longer valid.	

### codingInstructions

Datatype	InternationalString
Cardinality	0..n
Additional information which drives the coding process. Required when coding is dependent upon one or many other factors.	

## ClassificationItem

### Extends

This class extends Node

### Corresponds to GSIM

Classification Item

### Definition

A Classification Item represents a Category at a certain Level within a Statistical Classification.

## Explanatory Notes

A Classification Item defines the content and the borders of the Category. A Unit can be classified to one and only one item at each Level of a Statistical Classification.

## Properties

### isValid

Datatype	xs:boolean
Cardinality	0..1
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.	

### isGenerated

Datatype	xs:boolean
Cardinality	0..1
Indicates whether or not the item has been generated to make the level to which it belongs complete	

### explanatoryNotes

Datatype	StructuredString
Cardinality	0..n
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

Datatype	InternationalString
Cardinality	0..n
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.	

### changeLog

Datatype	InternationalString
Cardinality	0..1
Describes the changes, which the item has been subject to during the life time of the actual Statistical Classification.	

## changeFromPreviousVersion

Datatype	InternationalString
Cardinality	0..1
Describes the changes, which the item has been subject to from the previous version to the actual Statistical Classification	

## validDate

Datatype	Date
Cardinality	0..1
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	

## officialName

Datatype	Name
Cardinality	1..1
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	

## Relationships

### caseLaw

Target	AuthorizationSource
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	0..n
Case law rulings related to the Classification Item.	

### groups

Target	ClassificationIndexEntry
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	0..n
ClassificationIndexEntries related to the ClassificationItem.	

## excludes

Target	ClassificationItem
Type	Neither
Source Cardinality	0..n
Target Cardinality	0..n
Classification Items to which the excluded cases belong (as described in explanatoryNotes).	

# ClassificationSeries

## Extends

This class extends Collection

## Corresponds to GSIM

Classification Series

## Definition

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).

## Properties

### context

Datatype	StructuredString
Cardinality	0..1
ClassificationSeries can be designed in a specific context.	

### objectsOrUnitsClassified

Datatype	StructuredString
Cardinality	1..1
A ClassificationSeries is designed to classify a specific type of object/unit according to a specific attribute.	

### subjectAreas

Datatype	StructuredString
Cardinality	1..1
Areas of statistics in which the ClassificationSeries is implemented.	

## owners

Datatype	String
Cardinality	0..1
The statistical office or other authority, which created and maintains the StatisticalClassification (s) related to the ClassificationSeries. A ClassificationSeries may have several owners.	

## keywords

Datatype	StructuredString
Cardinality	0..n
A ClassificationSeries can be associated with one or a number of keywords.	

## Relationships

### groups

Target	StatisticalClassification
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	1..n
Specialization of contains in Collection for StatisticalClassifications.	

## Code

### Extends

This class extends Designation

### Based on DDI 3.2

l:CodeType

### Corresponds to GSIM

Code Item

### Definition

A Designation for a Category.

## Properties

### value

Datatype	Value
----------	-------

Cardinality	1..1
Specified value of the code	

## Relationships

### hasCategory

Target	Category
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	1..1
A definition for the code.	

## CodeList

### Extends

This class extends NodeSet

### Based on DDI 3.2

l:CodeListType

### Corresponds to GSIM

Code List

### Definition

A list of Codes and associated Categories. May be flat or hierarchical.

## Relationships

### hasCode

Target	Code
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	1..n
Specialization of contains in NodeSet for Codes.	

## references

Target	CategorySet
Type	Neither
Source Cardinality	1..n
Target Cardinality	0..1
CategorySet associated with the CodeList.	

## represents

Target	EnumeratedValueDomain
Type	Neither
Source Cardinality	1..n
Target Cardinality	0..n
Enumerated Value Domain represented by the CodeList.	

# CorrespondenceTable

## Extends

This class extends `OrderedCollectionCorrespondence`

## Corresponds to GSIM

Correspondence Table

## Definition

A Correspondence Table expresses a relationship between two NodeSets.

## Explanatory Notes

CorrespondenceTables are typically used in specializations of NodeSet, like Statistical Classifications. For instance, it can relate two versions from the same Classification Series; Statistical Classifications from different Classification Series; a variant and the version on which it is based; or, different versions of a variant. In the first and last examples, the Correspondence Table facilitates comparability over time.

## Properties

### owners

Datatype	String
Cardinality	0..1

The statistical office, other authority or section that created and maintains the Correspondence Table. A Correspondence Table may have several owners.

## **maintenanceUnit**

Datatype	String
Cardinality	0..1
The unit or group of persons who are responsible for the Correspondence Table, i.e. for maintaining and updating it.	

## **contactPersons**

Datatype	String
Cardinality	0..n
The person(s) who may be contacted for additional information about the Correspondence Table.	

## **publications**

Datatype	StructuredString
Cardinality	0..n
A list of the publications in which the Correspondence Table has been published.	

## **effectivePeriod**

Datatype	Date
Cardinality	0..1
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**

## **contains**

Target	Map
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	1..n
Set of mappings between nodes that participate in the correspondence.	

## **source**

Target	NodeSet
Type	Neither
Source Cardinality	0..n

Target Cardinality	1..n
The NodeSet(s) from which the correspondence is made. Specialization of source in OrderedCollectionCorrespondence.	

## target

Target	NodeSet
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..n
The NodeSet(s) to which the correspondence is directed. Specialization of target in OrderedCollectionCorrespondence.	

## sourceLevel

Target	Level
Type	Neither
Source Cardinality	0..n
Target Cardinality	0..1
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

Target	Level
Type	Neither
Source Cardinality	0..n
Target Cardinality	0..1
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.	

# Data Type

## Extends

This class extends `AnnotatedIdentifiable`

## Definition

Set of distinct values, characterized by properties of those values, and by operations on those values. (From ISO/IEC 11404 - General purpose datatypes)

## Example

1) Temperature can be measured in Fahrenheit, Celsius, and Kelvin. For the first 2 units of measure, one may take differences but not ratios. It is wrong to say 25 deg F is a third as warm as 75 deg F. Same with Celsius. However, 75 deg F is 50 deg F warmer than 25 deg F. Same with Celsius. Same with Kelvin. However, in Kelvin, 25 deg K IS a third as warm as 75 deg K. Therefore, we say Celsius and Fahrenheit measures have an Interval datatype, whereas Kelvin measures are Ratio. 2) Take the typical Sex Codes classification <0, male> <1, female> Does the fact that 0 and 1 have an ordering mean anything here? You don't want it to, as codes are arbitrary, so you declare the Nominal datatype. This corresponds to the fact that male and female as categories have no ordering. But, the declaration makes it clear. 3) An example that is a little more confusing is educational attainment. To keep it simple, let the categories and codes be as follows: <1, elementary> <2, high school> <3, college> <4, post-graduate> One could say these values (or categories) have an ordering, thus you would assign Ordinal to this. But, Ordinal with numeric codes suggests to people they can take averages, which leads to loads of confusion. This is not Interval data, and that means differences are not necessarily comparable. Taking averages requires comparable intervals, and doing so here leads to nonsense. Finally, it is an interesting question whether assigning Nominal to this might work just as well. The ordering is not so well-defined. 4) A preference scale on the other hand needs to be Ordinal, as preferences have an intrinsic order. As we said above, the codes are not important, so you could have the following (simple) scale: <3, dislike> <1, neutral> <2, like> The codes are arbitrary. Typically, though, we choose them as mnemonics to coincide with the available computation. But, look what will happen with averages here! 5) Currencies are often listed in units.hundredths. For example, 5 dollars and 22 cents is 5.22. This leads to the usage of Real datatypes, yet a Scaled datatype (see ISO/IEC 11404 for a detailed explanation) is far more appropriate. There is a difference between the datatype in some application (often currencies can only be represented as Real numbers) and the intended one (Scaled in the case of currencies). So, the RV has the intended datatype specified, whereas the IV has the application one.

## Properties

### scheme

Datatype	InternationalString
Cardinality	1..1
ISO 11404, Excel, SAS, R, etc.	

## Described Sentinel Value Domain

### Extends

This class extends DescribedValueDomain

### Definition

A described value domain whose values are used only for the processing of data after capture but before dissemination.

## Relationships

### measures

Target	InstanceVariable
Type	Neither
Source Cardinality	0..1
Target Cardinality	0..n

## DescribedSubstantiveValueDomain

### Extends

This class extends DescribedValueDomain

### Definition

A described value domain whose values are associated with the scientific questions of interest in a study.

## Relationships

### measures

Target	RepresentedVariable
Type	Neither
Source Cardinality	0..1
Target Cardinality	0..n

## DescribedValueDomain

### Extends

This class extends ValueDomain

### Corresponds to GSIM

Value Domain

### Definition

A Value Domain defined by an expression. [GSIM 1.1]

## Example

All real decimal numbers between 0 and 1 specified in Arabic numerals. [GSIM 1.1]

## Relationships

### references

Target	DescribedConceptualDomain
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Corresponding conceptual definition given by an DescribedConceptualDomain.	

## Designation

### Extends

This class extends AnnotatedIdentifiable

### Abstract

This class is abstract

## Corresponds to GSIM

Designation

## Definition

The name given to an object for identification.

## Example

A linking of the term “Unemployment” to an particular underlying concept.

## Explanatory Notes

The representation of a concept by a sign (e.g., string, pictogram, bitmap) which denotes it; a property of concept

## Properties

### label

Datatype	Label
----------	-------

Cardinality	0..n
A display label for the Designation. May be expressed in multiple languages. Repeat for labels with different content, for example, labels with differing length limitations.	

## description

Datatype	StructuredString
Cardinality	0..1
A description of the purpose or use of the Designation. May be expressed in multiple languages and supports the use of structured content.	

## Relationships

### hasVocabulary

Target	Vocabulary
Type	Composition
Source Cardinality	1..n
Target Cardinality	0..n
The Sign associated with the Designation is drawn from a Vocabulary	

## EnumeratedSentinelValueDomain

### Extends

This class extends EnumeratedValueDomain

### Definition

An enumerated value domain whose values are used only for the processing of data after capture but before dissemination.

## Relationships

### measures

Target	InstanceVariable
Type	Neither
Source Cardinality	0..1
Target Cardinality	0..n

# EnumeratedSubstantiveValueDomain

## Extends

This class extends EnumeratedValueDomain

## Definition

An enumerated value domain whose values are associated with the scientific questions of interest in a study.

## Relationships

### measures

Target	RepresentedVariable
Type	Neither
Source Cardinality	0..1
Target Cardinality	0..n

# EnumeratedValueDomain

## Extends

This class extends ValueDomain

## Corresponds to GSIM

Enumerated Value Domain

## Definition

A Value domain expressed as a list of categories and associated codes.

## Example

Sex categories expressed as Codes <m, male>; <f, female>; <o, other>.

## Relationships

### references

Target	EnumeratedConceptualDomain
--------	----------------------------

Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Corresponding conceptual definition given by an EnumeratedConceptualDomain.	

## IndexOrder

### Extends

This class extends OrderRelation

### Definition

Indexing order, defined either by predecessor-successor pairs or by a criteria (e.g. alphabetical, in code order, etc.)

## Relationships

### precedingEntry

Target	ClassificationIndexEntry
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
Specialization of predecessor in OrderRelation for ClassificationIndexEntries.	

### followingEntry

Target	ClassificationIndexEntry
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
Specialization of successor in OrderRelation for ClassificationIndexEntries.	

## Level

### Extends

This class extends Collection

## Based on DDI 3.2

l:LevelType

## Corresponds to GSIM

Level

## Definition

The Level describes the nesting structure of a hierarchical collection.

## Explanatory Notes

A Level often is associated with a Concept, which defines it. In a hierarchical structure of a NodeSet the nodes of each Level but the highest are aggregated to the nearest higher Level. A linear structure has only one Level.

## Relationships

### groups

Target	Node
Type	Aggregation
Source Cardinality	1..1
Target Cardinality	1..n
Specialization of contains in Collection for Nodes.	

### isDefinedBy

Target	Concept
Type	Neither
Source Cardinality	0..n
Target Cardinality	0..1
Associated concept that provides the conceptual definition of the level.	

## LevelParentChild

### Extends

This class extends OrderRelation

## Definition

Parent-child specialization of OrderRelation between Levels within a NodeSet. The inherited type property is set to "total" to specify that the parent-child relationships among Levels in any given NodeSet define a linear sequence.

## Relationships

### parent

Target	Level
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
Specialization of predecessor in OrderRelation for parents of Levels.	

### child

Target	Level
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
Specialization of successor in OrderRelation for children of Levels.	

## Map

### Extends

This class extends OrderedMemberCorrespondence

### Corresponds to GSIM

Map

## Definition

## Properties

### validFrom

Datatype	Date
----------	------

Cardinality	0..1
Date from which the Map became valid. The date must be defined if the Map belongs to a floating CorrespondenceTable.	

## validTo

Datatype	Date
Cardinality	0..1
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

### mapsSource

Target	Node
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	1..n
Specialization of source in OrderedMemberCorrespondence for Nodes.	

### mapsTarget

Target	Node
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	1..n
Specialization of target in OrderedMemberCorrespondence for Nodes.	

## Node

### Extends

This class extends Member

### Abstract

This class is abstract

### Corresponds to GSIM

Node

## Definition

A combination of a category and related attributes.

## Example

Within a CodeList (a subtype of NodeSet) each Code is a Node and contains the value of the code and the category associated with it.

## Explanatory Notes

A Node is created as a Category, Code or Classification Item for the purpose of defining the situation in which the Category is being used. Nodes can not be reused.

## Relationships

### contains

Target	Designation
Type	Aggregation
Source Cardinality	0..n
Target Cardinality	0..n
Representation(s) of the related Category.	

### takesMeaningFrom

Target	Category
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Category providing meaning to the Node.	

## NodeParentChild

### Extends

This class extends OrderRelation

### Definition

Parent-child specialization of OrderRelation between Nodes within a NodeSet. The inherited type property is set to "partial" to specify that the parent-child relationships among Nodes define a tree structure.

## Relationships

### parent

Target	Node
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Specialization of predecessor in OrderRelation for parents of Nodes.	

### child

Target	Node
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
Specialization of successor in OrderRelation for children of Nodes.	

## NodePartWhole

### Extends

This class extends OrderRelation

### Definition

Part-whole specialization of OrderRelation between Nodes within a NodeSet. The inherited type property is set to "partial" to specify that the part-whole relationships among Nodes define a tree structure.

## Relationships

### whole

Target	Node
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Specialization of predecessor in OrderRelation for a part-whole relation between Nodes.	

## part

Target	Node
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
Specialization of successor in OrderRelation for a part-whole relation between Nodes.	

## NodeSet

### Extends

This class extends Collection

### Abstract

This class is abstract

### Corresponds to GSIM

Node Set

### Definition

A NodeSet is a set of Nodes, which could be organized into a hierarchy of Levels.

### Example

Here are 2 examples: 1) Sex Categories Male Female Other 2) Sex Codes <m, male> <f, female> <o, other>

### Relationships

#### isBasedOn

Target	Concept
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..n
Associated concept within a system.	

#### contains

Target	Node
--------	------

Type	Aggregation
Source Cardinality	1..n
Target Cardinality	1..n
Specialization of contains in Collection for Nodes.	

## hasLevel

Target	Level
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	0..n
Specialization of contains in Collection for Levels.	

## hasNodeParentChild

Target	NodeParentChild
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	0..n
Specialization of isOrderedBy in Collection for parent-child relationships between Nodes.	

## hasNodePartWhole

Target	NodePartWhole
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	0..n
Specialization of isOrderedBy in Collection for part-whole relationships between Nodes.	

## hasNodeSetParentChild

Target	NodeSetParentChild
Type	Aggregation
Source Cardinality	1..n

Target Cardinality	0..n
Specialization of isOrderedBy in Collection for parent-child relationships between NodeSets.	

## hasNodeSetPartWhole

Target	NodeSetPartWhole
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	0..n
Specialization of isOrderedBy in Collection for part-whole relationships between NodeSets.	

## hasLevelParentChild

Target	LevelParentChild
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	0..n
Specialization of isOrderedBy in Collection for parent-child relationships between Levels.	

# NodeSetParentChild

## Extends

This class extends OrderRelation

## Definition

Parent-child specialization of OrderRelation between NodeSets. The inherited type property is set to "partial" to specify that the parent-child relationships among NodeSets define a tree structure.

## Relationships

### parent

Target	NodeSet
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1

Specialization of predecessor in OrderRelation for parents of NodeSets.

## child

Target	NodeSet
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
Specialization of successor in OrderRelation for children of NodeSets.	

# NodeSetPartWhole

## Extends

This class extends OrderRelation

## Definition

Part-whole specialization of OrderRelation between NodeSets. The inherited type property is set to "partial" to specify that the part-whole relationships among NodeSets define a tree structure.

## Relationships

### whole

Target	NodeSet
Type	Neither
Source Cardinality	0..n
Target Cardinality	1..1
Specialization of predecessor in OrderRelation for a part-whole relation between NodeSets.	

### part

Target	NodeSet
Type	Neither
Source Cardinality	0..1
Target Cardinality	1..1
Specialization of successor in OrderRelation for a part-whole relation between NodeSets.	

# Sign

## Extends

This class extends `AnnotatedIdentifiable`

## Definition

Something that suggests the presence or existence of a fact, condition, or quality.

## Example

The terms “Unemployment” and “Arbeitslosigkeit”

## Explanatory Notes

It is a perceivable object. This class is used to denote a Concept as a Designation.

## Properties

### value

Datatype	StructuredString
Cardinality	1..1
The text representation	

### label

Datatype	Label
Cardinality	0..n
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.	

### description

Datatype	StructuredString
Cardinality	0..1
A description of the content and purpose of the object. May be expressed in multiple languages and supports the use of structured content.	

## Relationships

### denotes

Target	Designation
--------	-------------

Type	Neither
Source Cardinality	1..1
Target Cardinality	1..1
The kind of relation that links a Sign to a Designation is a "denotation".	

## StatisticalClassification

### Extends

This class extends NodeSet

### Corresponds to GSIM

Statistical Classification

### Definition

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)

### Explanatory Notes

The Categories are defined with reference to one or more characteristics of a particular universe of units of observation. A Statistical Classification may have a flat, linear structure or may be hierarchically structured, such that all Categories at lower Levels are sub-Categories of Categories at the next Level up. Categories in Statistical Classifications are represented in the information model as Classification Items. (Source: GSIM StatisticalClassification)

### Properties

#### introduction

Datatype	StructuredString
Cardinality	0..1
The introduction provides a detailed description of the Statistical Classification, the background for its creation, the classification variable and objects/units classified, classification rules etc. (Source: GSIM StatisticalClassification)	

#### releaseDate

Datatype	Date
Cardinality	0..1
Date the Statistical Classification was released	

## terminationDate

Datatype	Date
Cardinality	0..1
Date on which the Statistical Classification was superseded by a successor version or otherwise ceased to be valid. (Source: GSIM Statistical Classification)	

## validDate

Datatype	Date
Cardinality	0..1
The date the statistical classification enters production use.	

## isCurrent

Datatype	xs:boolean
Cardinality	0..1
Indicates if the Statistical Classification is currently valid.	

## isFloating

Datatype	xs:boolean
Cardinality	0..1
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)	

## changeFromBase

Datatype	StructuredString
Cardinality	0..1
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)	

## purposeOfVariant

Datatype	StructuredString
Cardinality	0..1
If the Statistical Classification is a variant, notes the specific purpose for which it was developed. (Source: GSIM StatisticalClassification/Purpose of variant)	

## copyright

Datatype	String
----------	--------

Cardinality	0..n
Copyright of the statistical classification.	

## updateChanges

Datatype	StructuredString
Cardinality	0..n
Summary description of changes which have occurred since the most recent classification version or classification update came into force.	

## availableLanguage

Datatype	xs:language
Cardinality	0..n
A list of languages in which the Statistical Classification is available. Repeat for each language.	

## Relationships

### hasClassificationItem

Target	ClassificationItem
Type	Aggregation
Source Cardinality	1..n
Target Cardinality	1..n
Specialization of contains in NodeSet for ClassificationItems.	

### isMaintainedBy

Target	Organization
Type	Neither
Source Cardinality	0..n
Target Cardinality	0..n
Organization, agency, or group within an agency responsible for the maintenance and upkeep of the statistical classification.	

### hasDistribution

Target	OtherMaterial
Type	Aggregation
Source Cardinality	0..n

Target Cardinality	0..n
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.	

## variantOf

Target	StatisticalClassification
Type	Neither
Source Cardinality	0..n
Target Cardinality	0..1
Statistical Classification on which the current variant is based, and any subsequent versions of that Statistical Classification to which it is also applicable.	

## has

Target	ClassificationIndex
Type	Neither
Source Cardinality	1..n
Target Cardinality	0..n
ClassificationIndex(es) related to the StatisticalClassification.	

## predecessor

Target	StatisticalClassification
Type	Neither
Source Cardinality	0..1
Target Cardinality	0..1
Statistical Classification superseded by the actual Statistical Classification (for those Statistical Classifications that are versions or updates).	

## successor

Target	StatisticalClassification
Type	Neither
Source Cardinality	0..1
Target Cardinality	0..1

Statistical Classification that supersedes the actual Statistical Classification (for those Statistical Classifications that are versions or updates),

## ValueDomain

### Extends

This class extends AnnotatedIdentifiable

### Abstract

This class is abstract

### Based on DDI 3.2

r:RepresentationType

### Corresponds to GSIM

Value Domain

### Definition

The permitted range of values for a characteristic of a variable. [GSIM 1.1]

### Example

Age categories with a numeric code list; Age in years; Young, Middle-aged and Old

### Explanatory Notes

The values can be described by enumeration or by an expression. Value domains can be either substantive/sentinel, or described/enumeration

### Properties

#### unitOfMeasurement

Datatype	xs:string
Cardinality	0..1
The unit in which the data values are measured (kg, pound, euro).	

#### label

Datatype	Label
Cardinality	0..n
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.	

## definition

Datatype	StructuredString
Cardinality	0..1
A definition of the object. May be expressed in multiple languages and supports the use of structured content.	

## description

Datatype	StructuredString
Cardinality	0..1
A description of the content and purpose of the object. May be expressed in multiple languages and supports the use of structured content.	

# Vocabulary

## Extends

This class extends ConceptSystem

## Based on DDI 3.2

c:VocabularyType

## Definition

A vocabulary is an established list of standardized terminology for use in indexing and retrieval of information.

## Example

A dictionary or thesauri, specialist vocabulary such as HASET, MESH

## Properties

### abbreviation

Datatype	InternationalString
Cardinality	0..n
Abbreviation of vocabulary title.	

### location

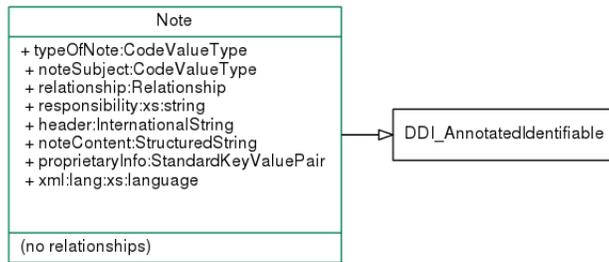
Datatype	URI
Cardinality	0..1
Location of external resource providing information about the vocabulary.	

## comments

Datatype	StructuredString
Cardinality	0..n
Information for the user regarding the reasons for use of the vocabulary and appropriate usage constraints.	

---

# Chapter 11. Utility (Full Review)



## Note

### Extends

This class extends `AnnotatedIdentifiable`

### Based on DDI 3.2

r:NoteType

### Definition

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.

## Properties

### typeOfNote

Datatype	CodeValueType
Cardinality	0..1
Specifies the type of note. Supports the use of a controlled vocabulary.	

### noteSubject

Datatype	CodeValueType
Cardinality	0..1
The subject of the note.	

### relationship

Datatype	Relationship
----------	--------------

Cardinality	0..n
Reference to one or more identifiable objects which the note is related to.	

## responsibility

Datatype	xs:string
Cardinality	0..1
The person or agency responsible for adding the note.	

## header

Datatype	InternationalString
Cardinality	0..1
A brief label or heading for the note contents.	

## noteContent

Datatype	StructuredString
Cardinality	0..1
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 formatting, DDI strongly recommends the use of local extensions or the Key/Value pair structure in ProprietaryInfo whenever possible.	

## proprietaryInfo

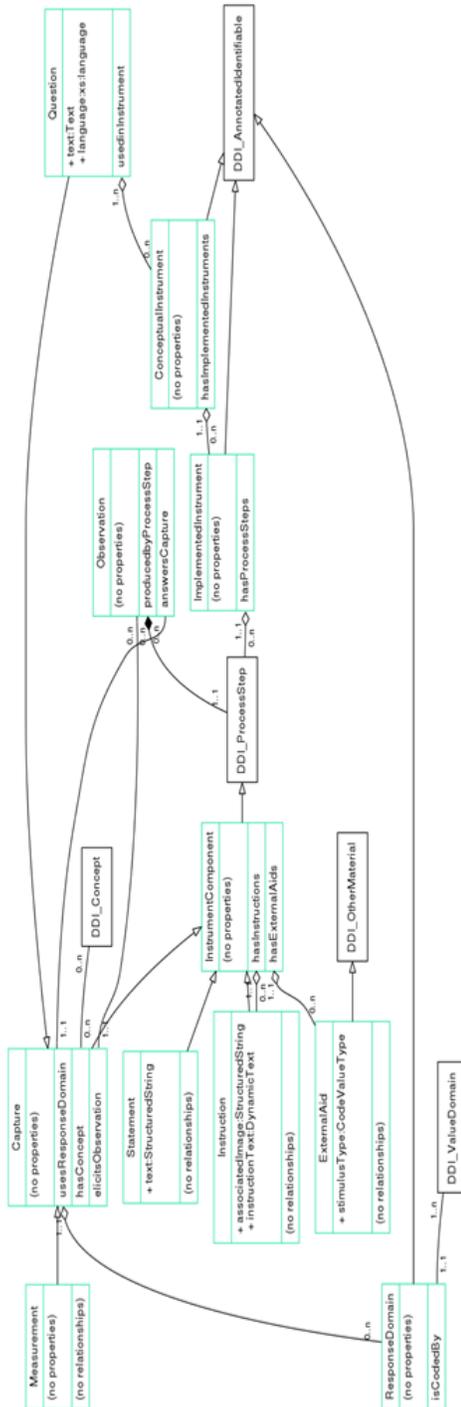
Datatype	StandardKeyValuePair
Cardinality	0..1
A set of actions related to the object as described by a set of name-value pairs. This would commonly be used in a case where additional information needs to be recorded regarding the content of a new element or attribute that has not yet been added to the 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.	

## xml:lang

Datatype	xs:language
Cardinality	0..1
Indicates the language of content. Note that xml:lang 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.	

# Chapter 12. DataCapture (Partial Review)

N.B. Not all classes in diagram are for review



# Capture

## Extends

This class extends InstrumentComponent

## Abstract

This class is abstract

## Definition

A Capture is an abstract object. Concrete objects that extend Capture describe the means of obtaining research data.

## Example

Question; blood pressure reading; MRI device; thermometer; web service

## Relationships

### usesResponseDomain

Target	ResponseDomain
Type	Aggregation
Source Cardinality	1..1
Target Cardinality	0..n

### hasConcept

Target	Concept
Type	Neither
Source Cardinality	0..n
Target Cardinality	0..n
Capture has a Concept	

### elicitsObservation

Target	Observation
Type	Neither
Source Cardinality	1..1

Target Cardinality	0..n
A Question elicits an Observation	

## ExternalAid

### Extends

This class extends OtherMaterial

### Based on DDI 3.2

r:OtherMaterialType

### Definition

Any external stimulus material used in an instrument that aids or facilitates data capture, or that is presented to a respondent and about which measurements are made.

### Example

Image, link, external aid, stimulus, physical object.

### Explanatory Notes

ExternalAid was removed from Statement to clean up and distinguish that object. Multiple ExternalAids can be used in an instrument design via ProcessStep. And then later...tried to introduce another object variously referred to as ExternalAid or OtherMaterial to refer to external stimulus materials used in data capture instruments (e.g. images, URNs, other stimulus material). But OtherMaterial already exists as an object in Simple Codebook. Where should OtherMaterial live? Can it live in two models at once?

### Properties

#### stimulusType

Datatype	CodeValueType
Cardinality	0..1

## Instruction

### Extends

This class extends InstrumentComponent

### Based on DDI 3.2

d:InstructionType

## Definition

Provides the content and description of data capture instructions. Contains the "how to" information for administering an instrument.

## Example

Completion instructions in self-administered mail questionnaire, information for administering a blood pressure measurement, interviewer instructions for a CATI questionnaire, guidance for communicating between an interviewer and a respondent (note MIDUS cognitive assessment example).

## Properties

### associatedImage

Datatype	StructuredString
Cardinality	0..1
An image associated with the Instruction, located at the provided URN or URL.	

### instructionText

Datatype	DynamicText
Cardinality	0..n
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.	

## InstrumentComponent

### Extends

This class extends ProcessStep

### Abstract

This class is abstract

## Definition

InstrumentComponent is an abstract object which extends ProcessStep. The purpose of InstrumentComponent is to provide a common parent for Capture (e.g., Question, Measure), Statement, and Instructions.

## Explanatory Notes

InstrumentComponent is never instantiated. It is a semantically meaningful parent for objects that exist in an instrument.

## Relationships

### hasInstructions

Target	Instruction
Type	Aggregation
Source Cardinality	1..1
Target Cardinality	0..n
An instrument component can have zero to many instructions	

### hasExternalAids

Target	ExternalAid
Type	Aggregation
Source Cardinality	1..1
Target Cardinality	0..n

## ResponseDomain

### Extends

This class extends AnnotatedIdentifiable

### Definition

The possible list of values that are allowed by a Capture.

### Example

Yes/No, Male/Female, Age in years, Open-ended text, Temperature, BP reading

### Explanatory Notes

The response domain links the value domain of a variable to the set of response categories in a question.

## Relationships

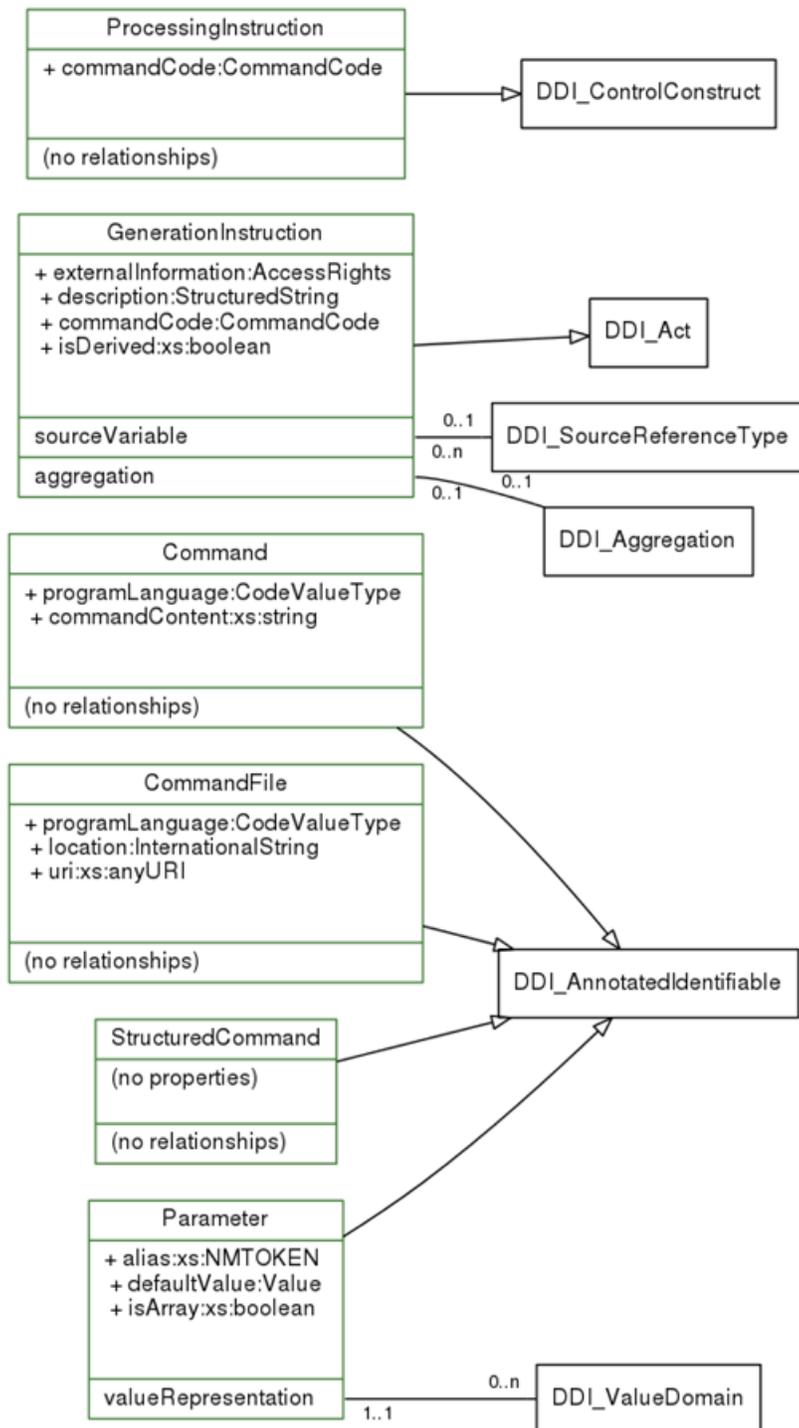
### isCodedBy

Target	ValueDomain
Type	Neither

Source Cardinality	1..1
Target Cardinality	1..n

# Chapter 13. Processing (Partial Review)

N.B. Not all classes in diagram are for review



# Command

## Extends

This class extends AnnotatedIdentifiable

## Based on DDI 3.2

r:CommandType

## Definition

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

### programLanguage

Datatype	CodeValueType
Cardinality	0..1
Designates the programming language used for the command. Supports the use of a controlled vocabulary.	

### commandContent

Datatype	xs:string
Cardinality	0..1
Content of the command itself expressed in the language designated in Programming Language.	

# CommandFile

## Extends

This class extends AnnotatedIdentifiable

## Based on DDI 3.2

r:CommandFileType

## Definition

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

### programLanguage

Datatype	CodeValueType
Cardinality	0..1
Designates the programming language used for the command. Supports the use of a controlled vocabulary.	

### location

Datatype	InternationalString
Cardinality	0..1
A description of the location of the file. This may not be machine actionable. It supports a description expressed in multiple languages.	

### uri

Datatype	xs:anyURI
Cardinality	0..1
The URL or URN of the command file.	

## Parameter

### Extends

This class extends AnnotatedIdentifiable

### Based on DDI 3.2

r:ParameterType

### Definition

A parameter is a structure that specifically identifies a source of input or output information so that it can be use pragmatically.

## Properties

### alias

Datatype	xs:NMTOKEN
Cardinality	0..1
If the content of the parameter is being used in a generic set of code as an alias for the value of an object in a formula (for example source code for a statistical program) enter that name here. This provides a link from the identified parameter to the alias in the code.	

## defaultValue

Datatype	Value
Cardinality	0..1
Provides a default value for the parameter if there is no value provided by the object it is bound to or the process that was intended to produce a value.	

## isArray

Datatype	xs:boolean
Cardinality	0..1
If set to "true" indicates that the content of the parameter is a delimited array rather than a single object and should be processed as such.	

## Relationships

### valueRepresentation

Target	ValueDomain
Type	Neither
Source Cardinality	1..1
Target Cardinality	0..n
If the content of the parameter contains representational content, such as codes, provide the representation here. ValueRepresentation is the abstract head of a substitution group and may be replaced with any valid substitution for ValueRepresentation. Inclusion of the ValueRepresentation is recommended if you will be sharing data with others as it provides information on the structure of what they can expect to receive when the parameter is processed.	

## StructuredCommand

### Extends

This class extends AnnotatedIdentifiable

### Based on DDI 3.2

r:StructuredCommandType

### Definition

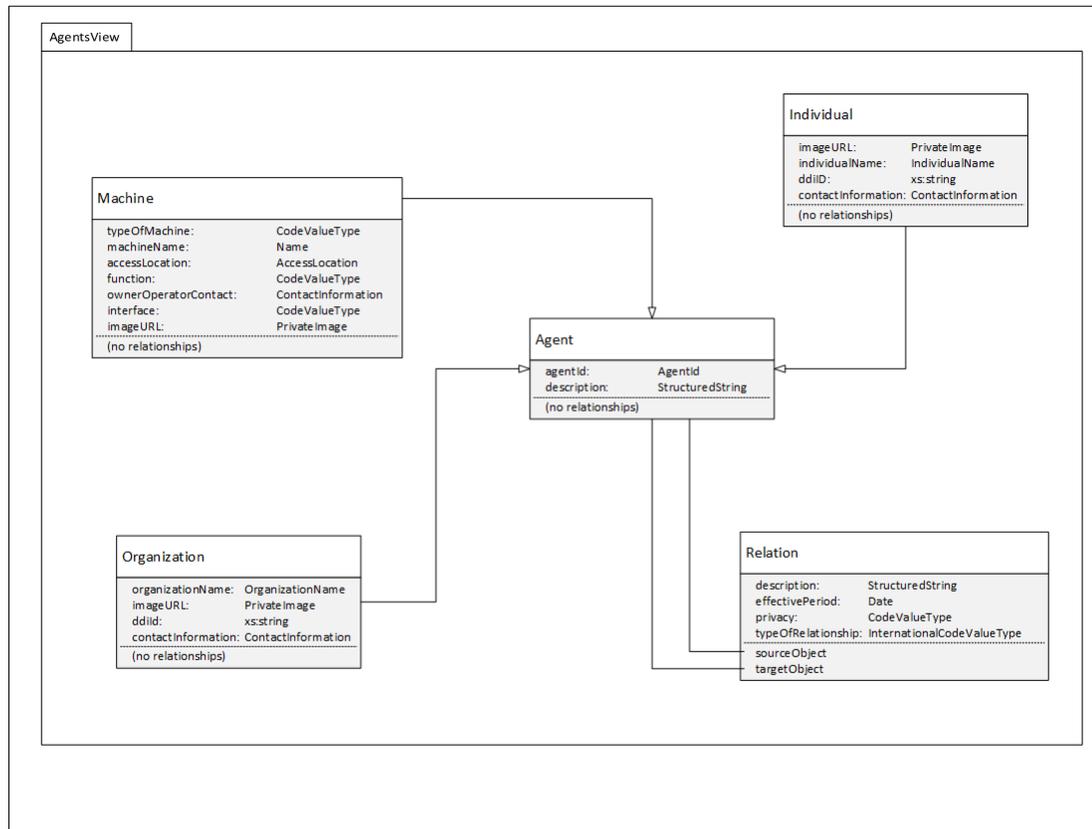
This type structures an empty stub which is used as the basis for extensions added using external namespaces such as MathML. The DDI 3.0 extension methodology is used here - a new module is declared, and the StructuredCommand element is used as the head of a substitution group to insert whatever XML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

# Chapter 14. Agents Functional View

The Agents Functional View provides the objects required to define Organizations, Individuals, and Machines and their interrelationships. Definitions include names and contact information for all agents. Sets of descriptive information are time-stamped to allow for change over time or periods where multiple sets of information are valid. Relationships between any two Agents are represented by a separate descriptions of pair-wise relationships for designated periods of time.

The Agents Functional View references the following classes

- Address
- Agent
- ContactInformation
- Email
- Individual
- ElectronicMessageSystem
- Machine
- Organization
- Relation
- Telephone



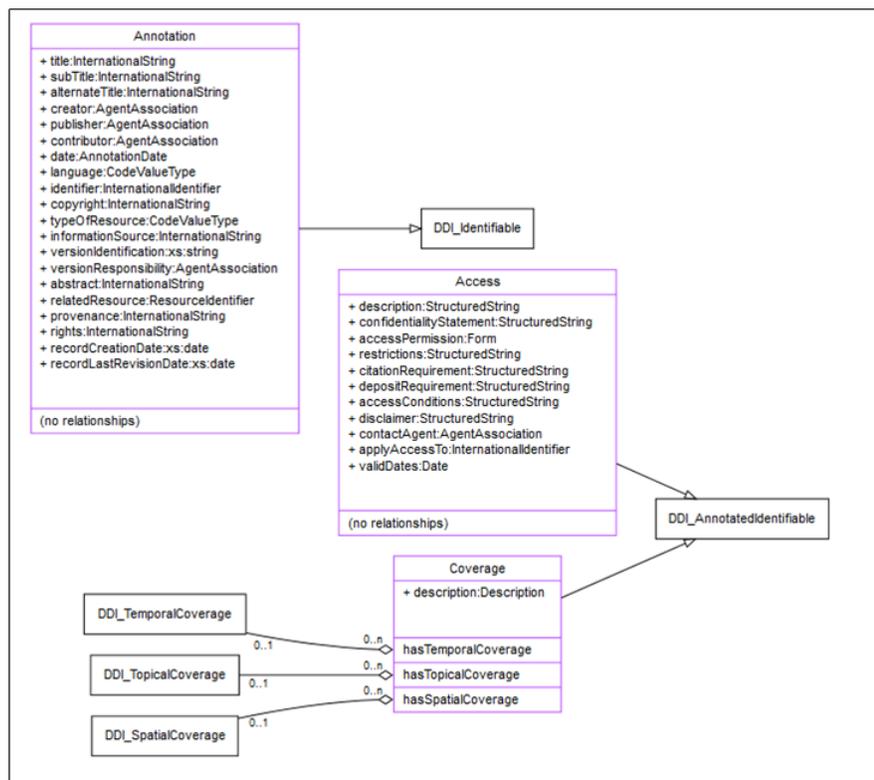
Agents Functional View

# Chapter 15. Discovery Functional View

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 [<http://www.ddialliance.org/Specification/RDF#disco>] as development continues.

The Discovery Functional View references the following classes

- Access
- Annotation
- Coverage



Discovery Functional View