gesis
Leibniz Institute for the Social Sciences

Create Your Own Information Systems on the Basis of DDI-Lifecycle

4th Annual European DDI User Conference

(EDDI 2012)

03.12.2012

Thomas Bosch

M.Sc. (TUM)
Ph.D. student
boschthomas.blogspot.com
GESIS - Leibniz Institute for the Social Sciences

Matthäus Zloch

M.Sc.
Ph.D. student

GESIS - Leibniz Institute for the Social Sciences

Introduction of Attendees

- What is your name?
- What's your organization?
- Why are you here?
  - What are you mostly interested in?
  - What do you expect from this tutorial?
  - …

Appr. 30 seconds

Goals of This Presentation

- Introduction to DDI-Lifecycle
  - Give a DDI Lifecycle overview
  - Show you (basic) elements of the DDI-Lifecycle and the main structure
  - How does Identification, Versioning, Maintenance work?
  - DDI Discovery Vocabulary “disco”
  - How to use the DDI Documentation
- Individual software project
  - Introduce you to basics of software architecture design
  - Show you how the software architecture of your project may look like
  - Show you how a software project leverages DDI and the disco model
  - Introduce you to a project template for using the disco model in the back-end

Outline

Matthäus

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy
- Software Architecture
  - Multitier
  - MVC
- Missy Data Model
  - Extendable Data Model

Thomas

- DDI-L Overview
- Identification, Versioning, Maintenance
- DDI-L Main Structures
- DDI Instance
- Study Unit
- Conceptual Component
- Logical Products
- Data Collection
- disco-model

Outline

- Persistence Layer
  - Programming Interface
  - Types of physical data storages
  - Examples
- Outlook
- Physical Data Product
- Physical Instance
- Archive
- DDIProfile
- DDI Serialization Examples
  - DDI-XML
  - DDI-RDF / disco-spec
  - Relational-DB

Outline

Matthäus

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy

Presentation

- Software Architecture
  - Multitier
  - MVC
- Missy Data Model
  - Extendable Data Model

Thomas

- DDI-L Overview
- Identification, Versioning, Maintenance
- DDI-L Main Structures
- DDI Instance
- Study Unit

- Conceptual Component
- Logical Products
- Data Collection
- disco-model

Business Layer

Outline

- Persistence Layer
  - Programming Interface
  - Types of physical data storages
  - Examples
- Physical Data Product
- Physical Instance
- Archive
- DDIProfile

Data Storage Access

- Outlook
- DDI Serialization Examples
  - DDI-XML
  - DDI-RDF / disco-spec
  - Relational-DB

Data Storage

Feel free to ask questions at any time!

What comes next?

- Before we start with the use-case Missy…
- Thomas, can you give us a top-level overview of DDI?
  - You might also include how DDI treats identification, versioning and maintenance…

What comes next?

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy

Presentation

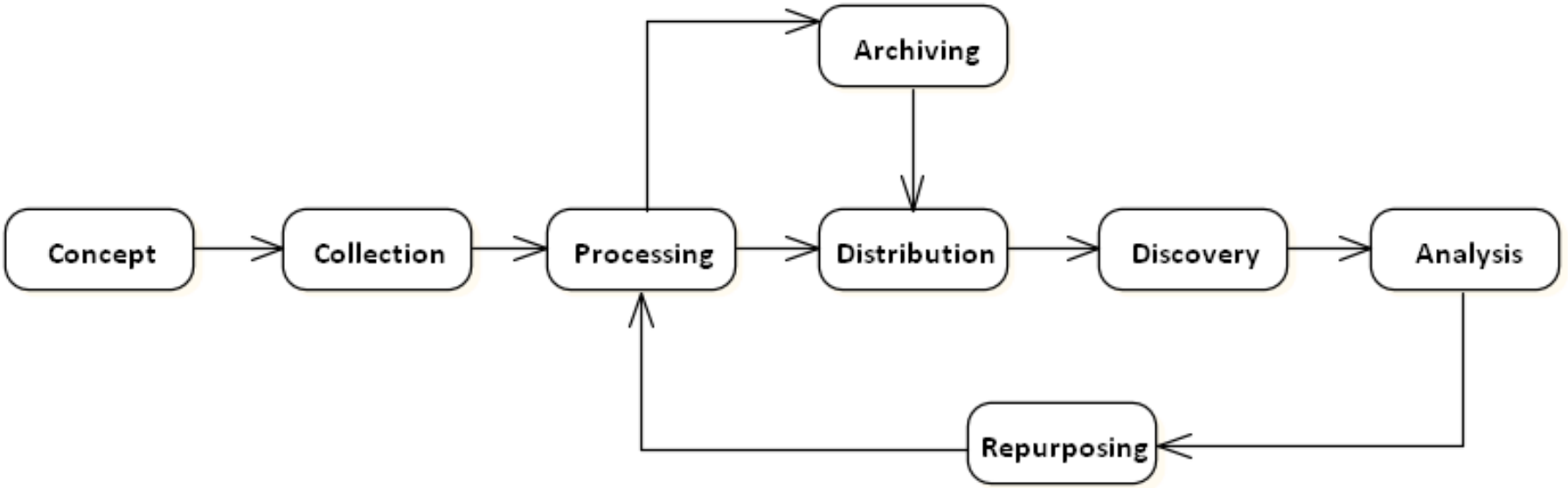
- **DDI-L Overview**
- Identification, Versioning, Maintenance
- DDI-L Main Structures
- DDI Instance
- Study Unit

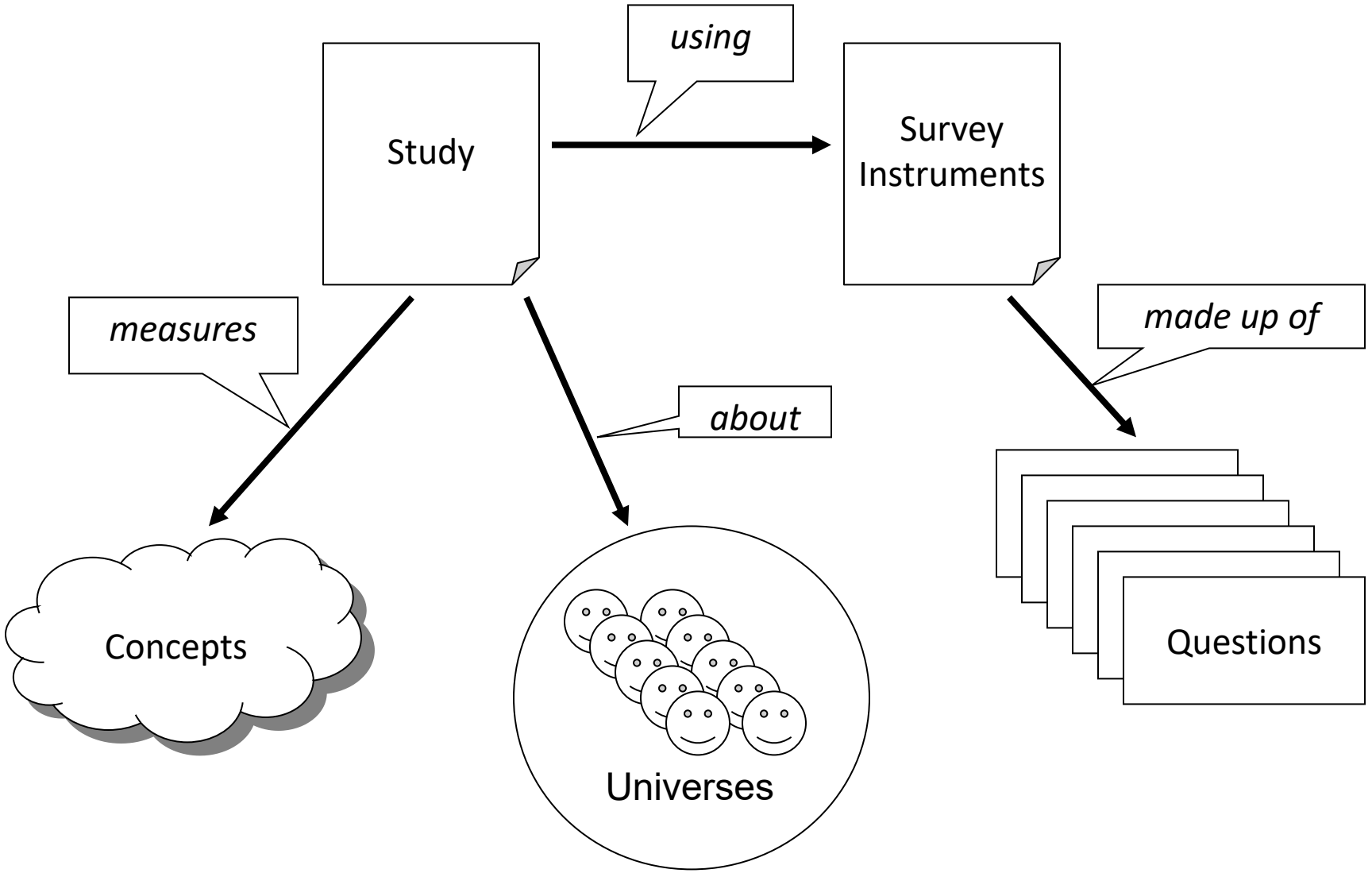
- Software Architecture
  - Multitier
  - MVC
- Missy Data Model
  - Extendable Data Model

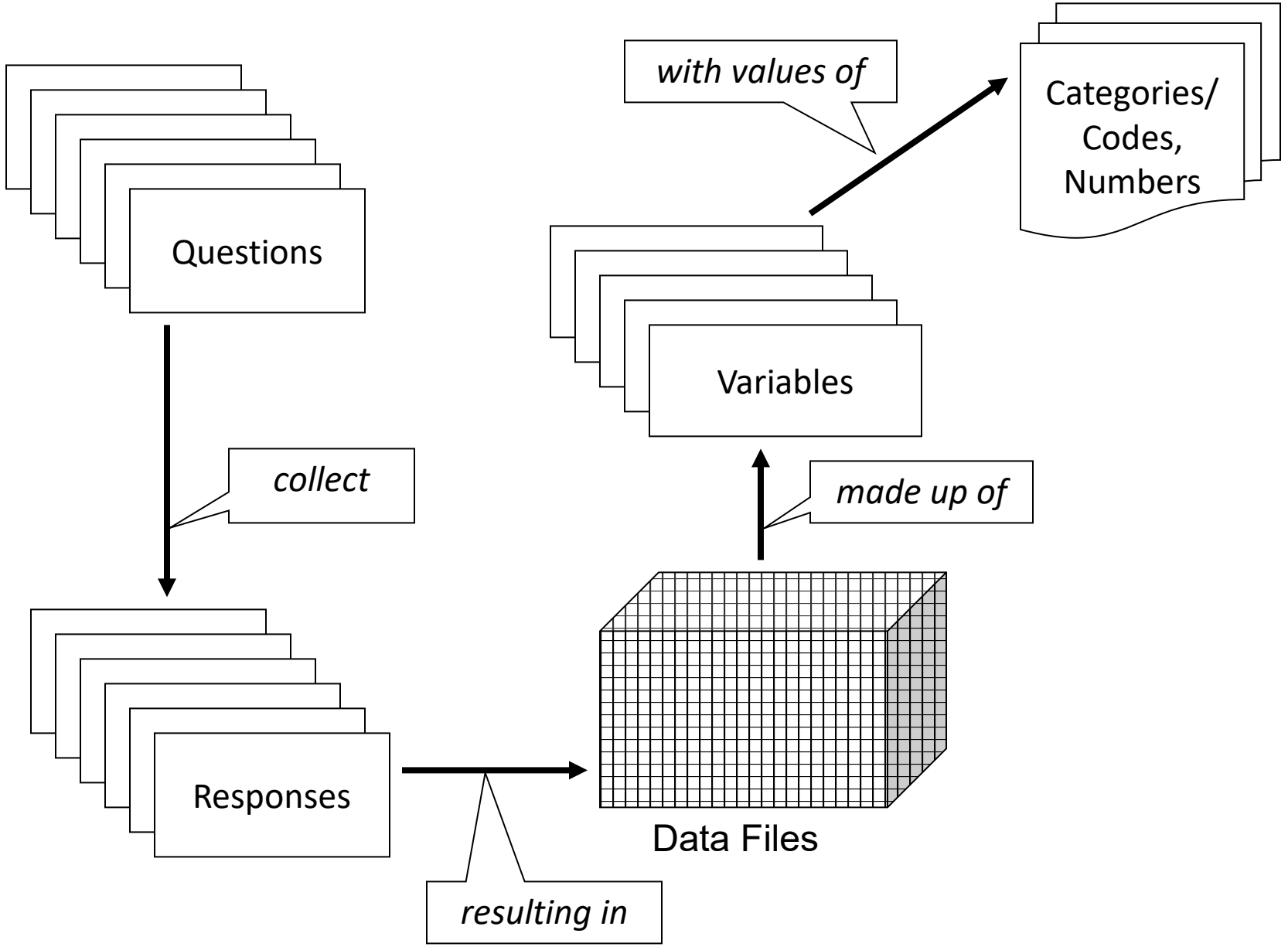
- Conceptual Component
- Logical Products
- Data Collection
- disco-model

Business Layer

DDI-L Stages


Archiving
Concept
Collection
Processing
Distribution
Discovery
Analysis
Repurposing


Study
using
Survey Instruments
measures
about
made up of
Concepts
Universes
Questions


Questions
collect
Responses
resulting in
Data Files
made up of
Variables
with values of
Categories/
Codes,
Numbers

What comes next?

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy

Presentation

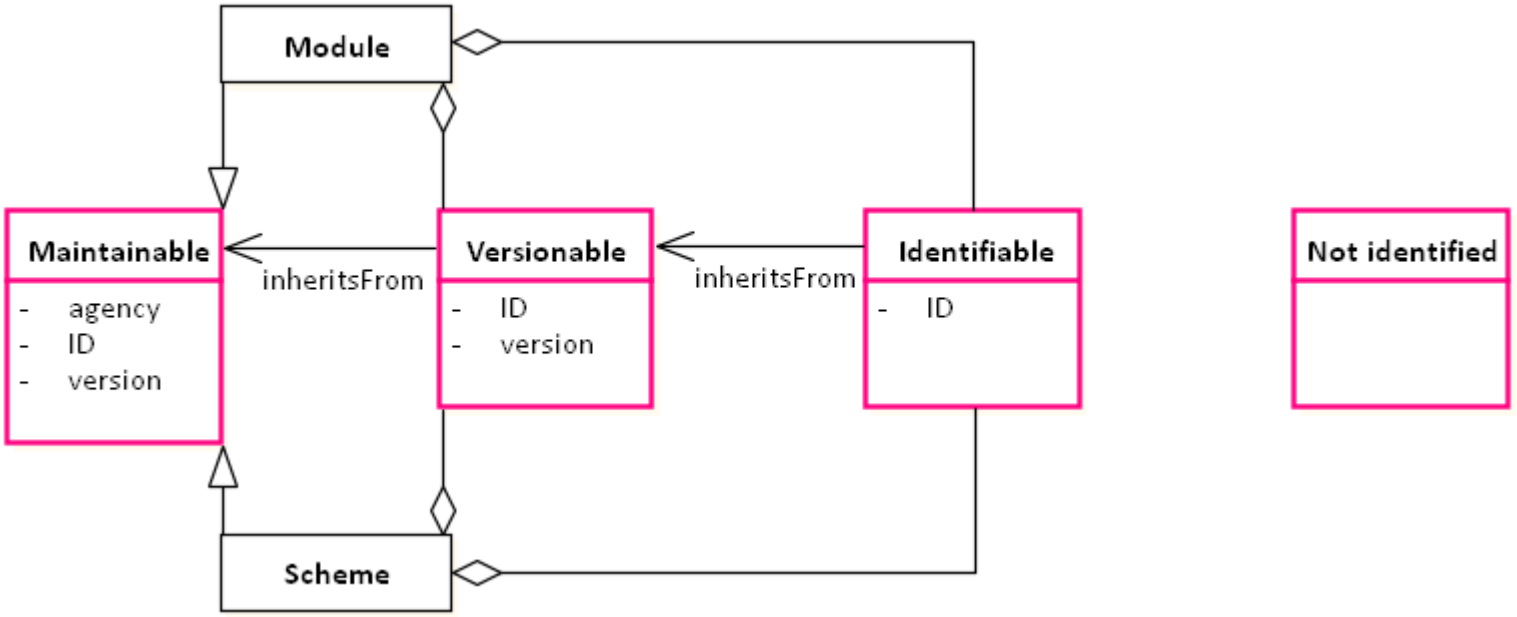
- DDI-L Overview
- **Identification, Versioning, Maintenance**
- DDI-L Main Structures
- DDI Instance
- Study Unit

- Software Architecture
  - Multitier
  - MVC
- Missy Data Model
  - Extendable Data Model

- Conceptual Component
- Logical Products
- Data Collection
- disco-model

Business Layer

Element Types



IDs must be unique within maintainables

Not identified elements are contained by other element types

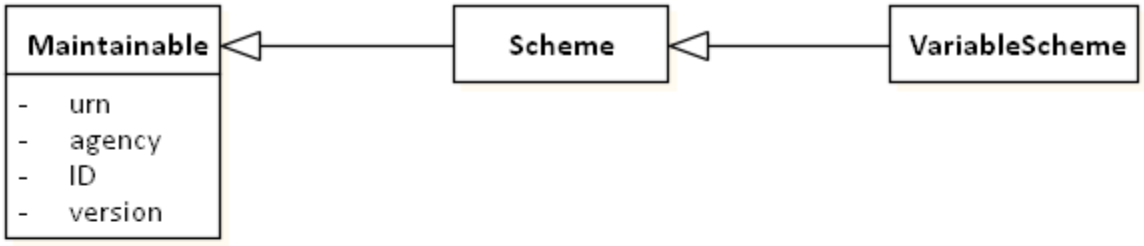
Identification

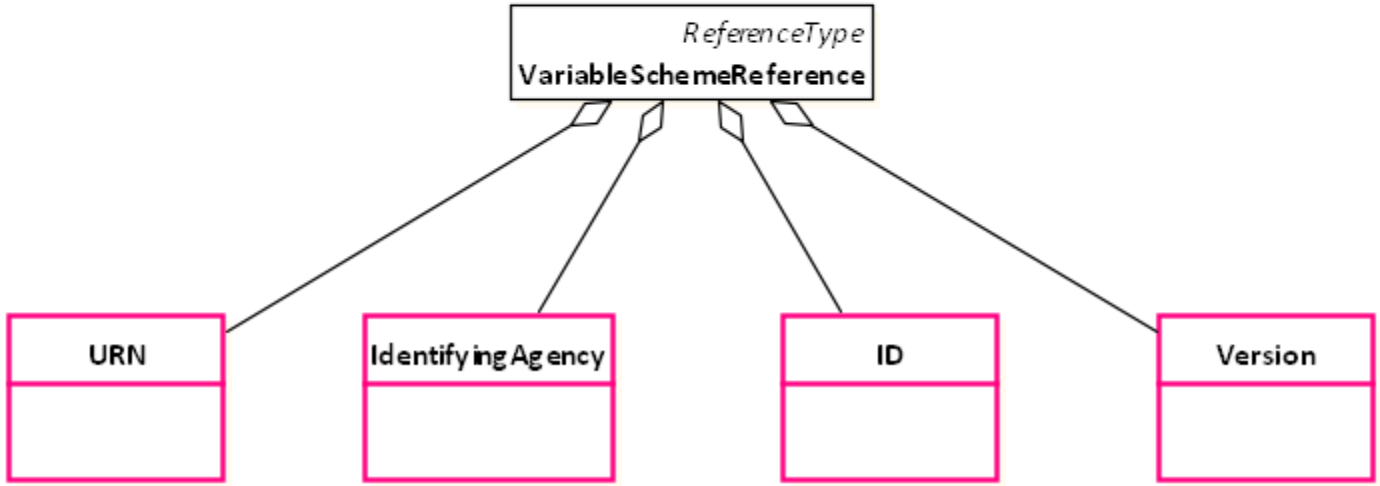
Agency, ID, Version

or

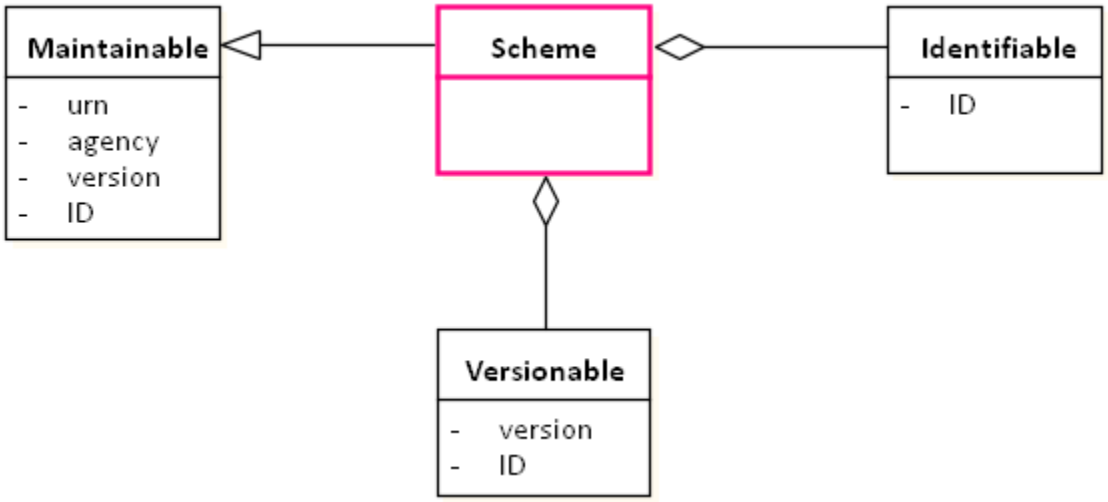
URN (= Agency + ID + Version)

Referencing Maintainables





Schemes



Very often, identifiable and versionable elements are maintained in parent schemes

Maintenance Rules

Maintenance Agency

- identified by a reserved code (based on its domain name; similar to it's website and e-mail)
- **Registration** of maintenance agency (e.g. GESIS)

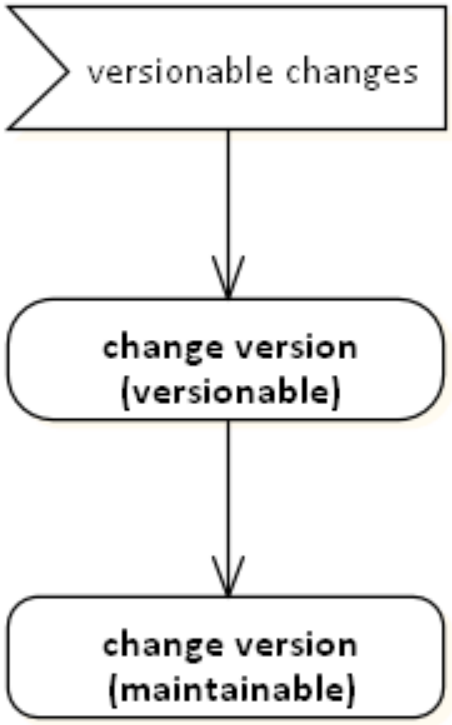
Ownership

- Maintenance agencies own the objects they maintain
- Only they are allowed to change or version the objects

Reference

- Other organizations may reference external items in their own schemes, but may not change those items
- You can make a copy which you change and maintain, but once you do that, you own it!

Versioning Rules



Identifiable (DDI 3.1 XML)

```
<DataItem id="AB347">
  …
</DataItem>
```

Versionable (DDI 3.1 XML)

```
<Variable id="V1" version="1.1.0" versionDate="20012-11-12">
   <VersionResponsibility>
      Thomas Bosch
   </VersionResponsibility>
  <VersionRationale>
      Spelling Correction
   </VersionRationale>
   …
</Variable >
```

Maintainable (DDI 3.1 XML)

```
<VariableScheme
   id="VarSch01"
   agency ="us.mpc"
   version="1.4.0"
   versionDate="2009-02-12">
  …
</VariableScheme>
```

URN (DDI 3.1)

```
urn:ddi:us.mpc:VariableScheme.
VarSch01.1.4.0:Variable.V1.1.1.0
```

URN (DDI 3.1)

urn:ddi:us.mpc:VariableScheme.VarSch01.1.4.0:Variable.V1.1.1.0

URN (DDI 3.1)

urn:ddi:us.mpc:VariableScheme.VarSch01.1.4.0:Variable.V1.1.1.0

Maintaining agency

URN (DDI 3.1)

urn:ddi:us.mpc:VariableScheme.VarSch01.1.4.0:Variable.V1.1.1.0

Maintainable (element type, ID, version)

URN (DDI 3.1)

urn:ddi:us.mpc:VariableScheme.VarSch01.1.4.0:Variable.V1.1.1.0

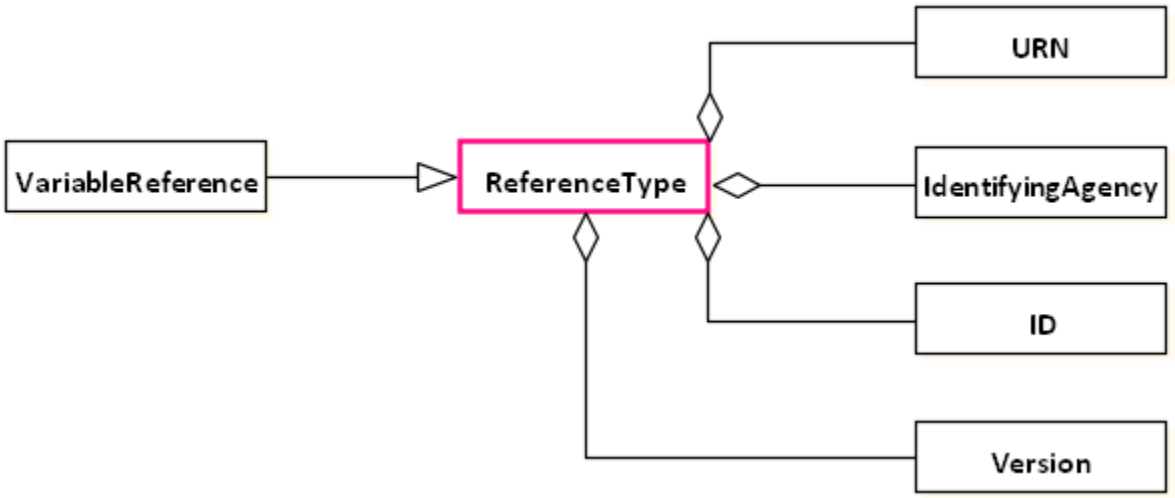
Identifiable (element type, ID, version number)

ID must be unique within maintainable

URN Resolution

- DNS-based resolution for DDI services
  - URN can be resolved by DNS means
  - http://registry.ddialliance.org/
    - details and library
    - list of available DNS services for this DDI agency
- URNs are mapped to local URLs
- XML Catalog can be used as simple resolution mechanism

Referencing (DDI 3.1)



(Agency, ID, Version) vs. URN

Often, these are **inherited** from a maintainable object

Referencing (DDI 3.1 + 3.2)

- **Internal**
  - In same DDI Instance
  - (agency, ID, version)
    - Maintainable (agency, ID, version)
    - Identifiable (ID), versionable (ID, version)
- **External**
  - To other DDI Instances
  - **URN > (agency, ID, version)**

Internal Reference (DDI 3.1)

```
<VariableReference isReference="true" isExternal="false"
    lateBound="false">
  <Scheme isReference="true" isExternal="false"
    lateBound="false">
   <ID>VarSch01</ID>
   <IdenftifyingAgency>us.mpc</IdentifyingAgency>
   <Version>1.4.0</Version>
  </Scheme>
  <ID>V1</ID>
  <IdenftifyingAgency>us.mpc</IdentifyingAgency>
  <Version>1.1.0</Version>
</VariableReference>
```

External Reference (DDI 3.1)

```
<VariableReference isReference="true" isExternal="true"
    lateBound="false">

    <urn>
        urn:ddi:us.mpc:VariableScheme.VarSch01.1.4.0:Variable.
        V1.1.1.0
    </urn>

</VariableReference>
```

Identifiers (DDI 3.2)

- **Why new IDs?**
  - better support of persistent identification (less semantics, context information)
- **URN or (agency, ID, version)**
- **Version changes?**
  - Administrative metadata → No
  - Payload metadata → yes
- **No inheritance**
  - Why? (implementation problems, …)
- **Deprecated vs. Canonical IDs**
  - Deprecated (no version of maintainables)
  - Mechanism for roundtrip available between DDI 3.2 canonical ID, DDI 3.2 deprecated ID, DDI 3.1 ID

Canonical URN (DDI 3.2)

urn:ddi:us.mpc:VariableScheme.VarSch01.1.4.0:Variable.V1.1.1.0

No maintainables

Canonical URN (DDI 3.2)

`urn:ddi:us.mpc:Variable.V1.1.1.0`

No maintainables

Canonical URN (DDI 3.2)

`urn:ddi:us.mpc:Variable.V1.1.1.0`

Agency, ID, version (Identifiable, Versionable)

Canonical URN (DDI 3.2)

`urn:ddi:us.mpc:Variable.V1.1.1.0`

ID unique in agency (not maintainable)

Canonical URN (DDI 3.2)

`urn:ddi:us.mpc:Variable.V1.1.1.0`

No element types

Canonical URN (DDI 3.2)

`urn:ddi:us.mpc:V1.1.1.0`

No element types

Canonical URN (DDI 3.2)

`urn:ddi:us.mpc:Variable.V1.1.1.0`

Colon is separator

Canonical URN (DDI 3.2)

urn:ddi:us.mpc:Variable.V1:1.1.0

Colon is separator

Canonical URN (DDI 3.2)

`urn:ddi:us.mpc:V1:1.1.0`

No maintainables

Agency, ID, version (Identifiable, Versionable)

ID unique in agency (not maintainable)

No element types

Colon is separator

What comes next?

- This was the DDI Overview with identification, versioning, and maintenance

- But, what is the use-case actually?
- What do you want to use DDI for?

What comes next?

- **Missy – General Information**
  - Requirements to Developers
  - Use-Cases in Missy

Presentation

- DDI-L Overview
- Identification, Versioning, Maintenance
- DDI-L Main Structures
- DDI Instance
- Study Unit

- Software Architecture
  - Multitier
  - MVC
- Missy Data Model
  - Extendable Data Model

- Conceptual Component
- Logical Products
- Data Collection
- disco-model

Business Layer

General Information About Missy

- Missy – Microdata Information System

- Largest household survey in Europe
- Provides detailed information about individual datasets
- Currently consists of “microcensus" survey, which is comprised of statistics about, e.g.
  - general population in Germany
  - situation about employment market
    - occupation, professional education, income, legal insurance, etc.

General Information About Missy

- May be split in two parts
  - Missy Web (end-user front-end part)
  - Missy Editor for documentation (back-end part)

- Consists of approx. 500 Variables & Questions
- Captures 25 years, since 1973

Missy 3

- That's what it's all about! The "next generation Missy"

- Integration of further surveys, e.g. EU-SILC, EU-LFS, …
- Implementation of Missy Editor as a Web-Application

Use-Cases in Missy

- General Information about survey “microcensus”
- Variables by thematic classification and year
- List of variables by year
- List of generated variables by year
- Show details of a variable with statistics
- Variable-Time Matrix
  - List of variables by thematic classification and year (selectable)

- Questionnaire Catalogue

Use-Cases in Missy

- General Information about survey “microcensus”
- Variables by thematic classification and year
- List of variables by year
- List of generated variables by year
- Show details of a variable with statistics
- Variable-Time Matrix
  - List of variables by thematic classification and year (selectable)

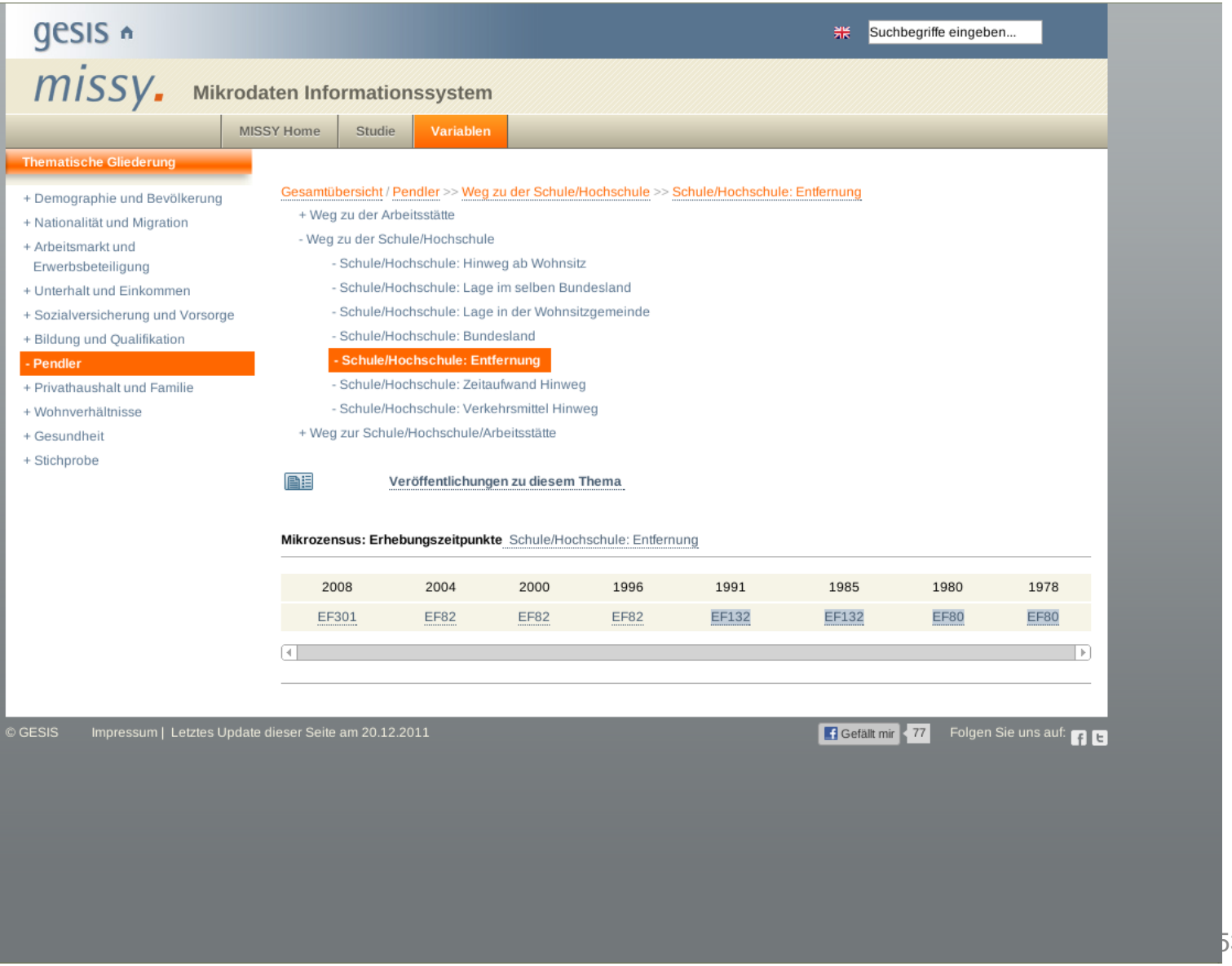
- Questionnaire Catalogue
- **In future: browse by survey and country!**

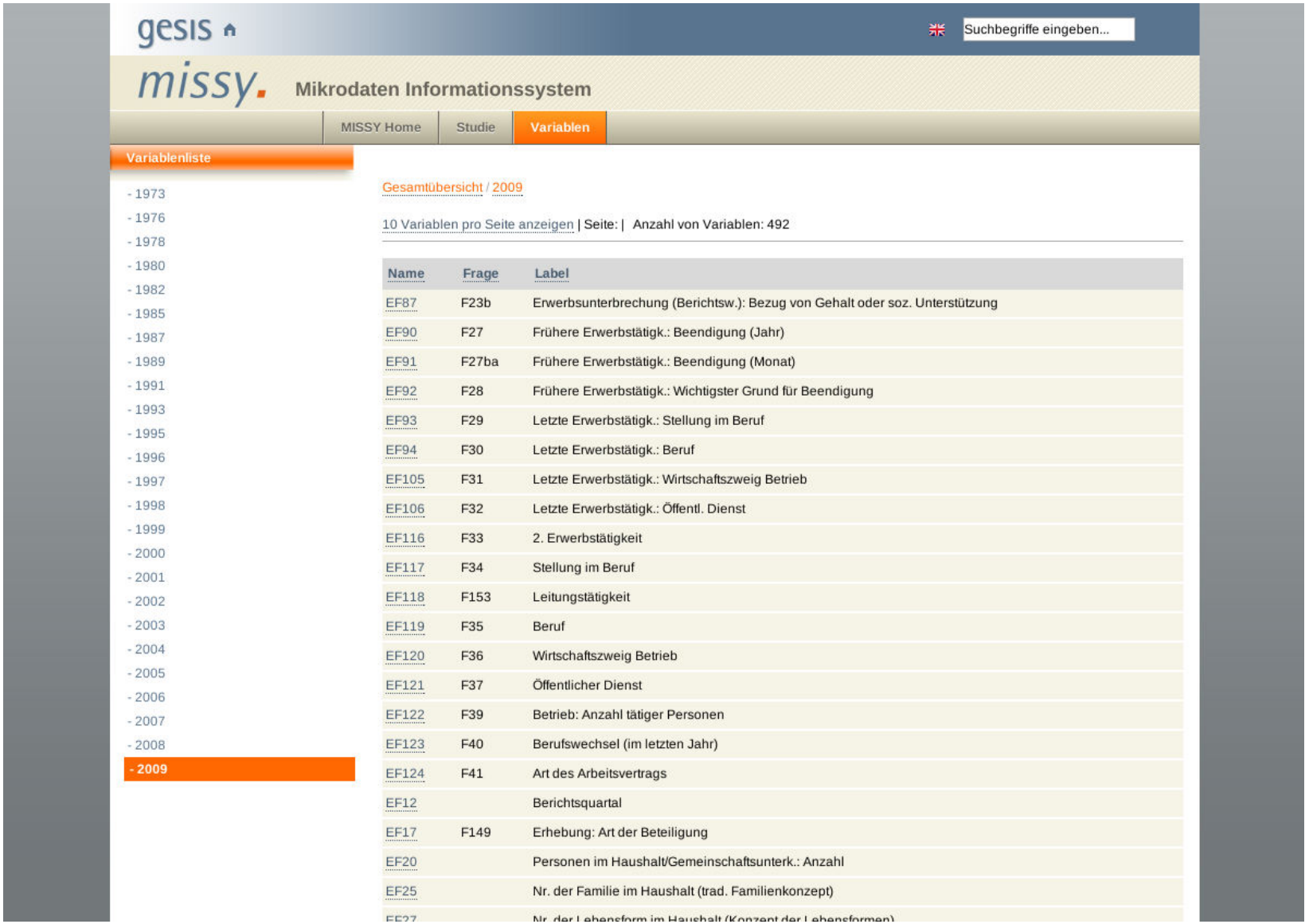
Use-Cases in Missy

- General Information about survey “microcensus”
- Variables by thematic classification and year
- List of variables by year
- List of generated variables by year
- **Show details of a variable with statistics**
- Variable-Time Matrix
  - List of variables by thematic classification and year (selectable)

- Questionnaire Catalogue


gesis
Suchbegriffe eingeben...
missy. Mikrodaten Informationssystem
MISSY Home
Studie
Variablen
über MISSY
Auswahl Datensatz
GML
Impressum
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
Mitglied der Leibniz-Gemeinschaft
MISSY - Mikrodaten-Informationssystem
Das Mikrodaten-Informationssystem MISSY bietet detaillierte Informationen zu Individualdatensätzen an. Im jetzigen Entwicklungsstadium stehen zunächst Metainformationen zum Mikrozensus, einer Erhebung der amtlichen Statistik, zur Verfügung.
Der Mikrozensus ist die amtliche Repräsentativstatistik über die Bevölkerung und den Arbeitsmarkt, an der jährlich 1 Prozent aller Haushalte in Deutschland beteiligt sind. Für Forschungszwecke stehen faktisch anonymisierte Scientific Use Files zur Verfügung, bei denen es sich um eine 70-Prozent-Stichprobe der Originaldaten handelt.
MISSY hat zum Ziel, die Verwendung der Mikrozensus Scientific Use Files für empirische Forschungsarbeiten zu erleichtern, und stellt hierzu ein umfassendes Online-Informationsangebot zur Verfügung. Das Datenangebot von MISSY beinhaltet detaillierte Informationen zu den Mikrozensus Scientific Use Files seit 1973:
Informationen auf Variablenebene finden Sie im Bereich
> Variablen
Hier haben Sie Zugang über:
Thematische Gliederung
Variablenliste
Variablen-Zeitpunkte-Matrix
Hintergrundinformationen zu den Mikrozensen (z.B. Studienbeschreibung, Fragebögen, Mikrodaten-Tools, Literatur etc.) finden Sie im Bereich
> Studie
Die wichtigsten Links und Informationen pro Erhebungsjahr zusammengefasst:
2009 2008 2007 2006 2005 2004 2003 2002 2001 2000 1999 1998
1997 1996 1995 1993 1991 1989 1987 1985 1982 1980 1978 1976
1973 1962-69 (GESIS-Files)
GESIS Quicklinks
Seite auswählen...
fdz German Microdata Lab
Keine Nachrichten in dieser Ansicht.


gesis
Suchbegriffe eingeben...
missy.
Mikrodaten Informationssystem
MISSY Home
Studie
Variablen
Thematische Gliederung
+ Demographie und Bevölkerung
+ Nationalität und Migration
+ Arbeitsmarkt und Erwerbsbeteiligung
+ Unterhalt und Einkommen
+ Sozialversicherung und Vorsorge
+ Bildung und Qualifikation
- Pendler
+ Privathaushalt und Familie
+ Wohnverhältnisse
+ Gesundheit
+ Stichprobe
Gesamtübersicht / Pendler >> Weg zu der Schule/Hochschule >> Schule/Hochschule: Entfernung
+ Weg zu der Arbeitsstätte
- Weg zu der Schule/Hochschule
- Schule/Hochschule: Hinweg ab Wohnsitz
- Schule/Hochschule: Lage im selben Bundesland
- Schule/Hochschule: Lage in der Wohnsitzgemeinde
- Schule/Hochschule: Bundesland
- Schule/Hochschule: Entfernung
- Schule/Hochschule: Zeitaufwand Hinweg
- Schule/Hochschule: Verkehrsmittel Hinweg
+ Weg zur Schule/Hochschule/Arbeitsstätte
Veröffentlichungen zu diesem Thema
Mikrozensus: Erhebungszeitpunkte Schule/Hochschule: Entfernung
2008 2004 2000 1996 1991 1985 1980 1978
EF301 EF82 EF82 EF82 EF132 EF132 EF80 EF80
© GESIS Impressum | Letztes Update dieser Seite am 20.12.2011
Gefällt mir 77
Folgen Sie uns auf:



gesis
Suchbegriffe eingeben...
missy. Mikrodaten Informationssystem
MISSY Home | Studie | Variablen
Variablenliste
- 1973
- 1976
- 1978
- 1980
- 1982
- 1985
- 1987
- 1989
- 1991
- 1993
- 1995
- 1996
- 1997
- 1998
- 1999
- 2000
- 2001
- 2002
- 2003
- 2004
- 2005
- 2006
- 2007
- 2008
- 2009

Gesamtübersicht / 2009

10 Variablen pro Seite anzeigen | Seite: | Anzahl von Variablen: 492

Name	Frage	Label
EF87	F23b	Erwerbsunterbrechung (Berichtsw.): Bezug von Gehalt oder soz. Unterstützung
EF90	F27	Frühere Erwerbstätigk.: Beendigung (Jahr)
EF91	F27ba	Frühere Erwerbstätigk.: Beendigung (Monat)
EF92	F28	Frühere Erwerbstätigk.: Wichtigster Grund für Beendigung
EF93	F29	Letzte Erwerbstätigk.: Stellung im Beruf
EF94	F30	Letzte Erwerbstätigk.: Beruf
EF105	F31	Letzte Erwerbstätigk.: Wirtschaftszweig Betrieb
EF106	F32	Letzte Erwerbstätigk.: Öffentl. Dienst
EF116	F33	2. Erwerbstätigkeit
EF117	F34	Stellung im Beruf
EF118	F153	Leitungstätigkeit
EF119	F35	Beruf
EF120	F36	Wirtschaftszweig Betrieb
EF121	F37	Öffentlicher Dienst
EF122	F39	Betrieb: Anzahl tätiger Personen
EF123	F40	Berufswechsel (im letzten Jahr)
EF124	F41	Art des Arbeitsvertrags
EF12		Berichtsquartal
EF17	F149	Erhebung: Art der Beteiligung
EF20		Personen im Haushalt/Gemeinschaftsunterk.: Anzahl
EF25		Nr. der Familie im Haushalt (trad. Familienkonzept)
EF27		Nr. der Lebensform im Haushalt (Konzept der Lebensformen)

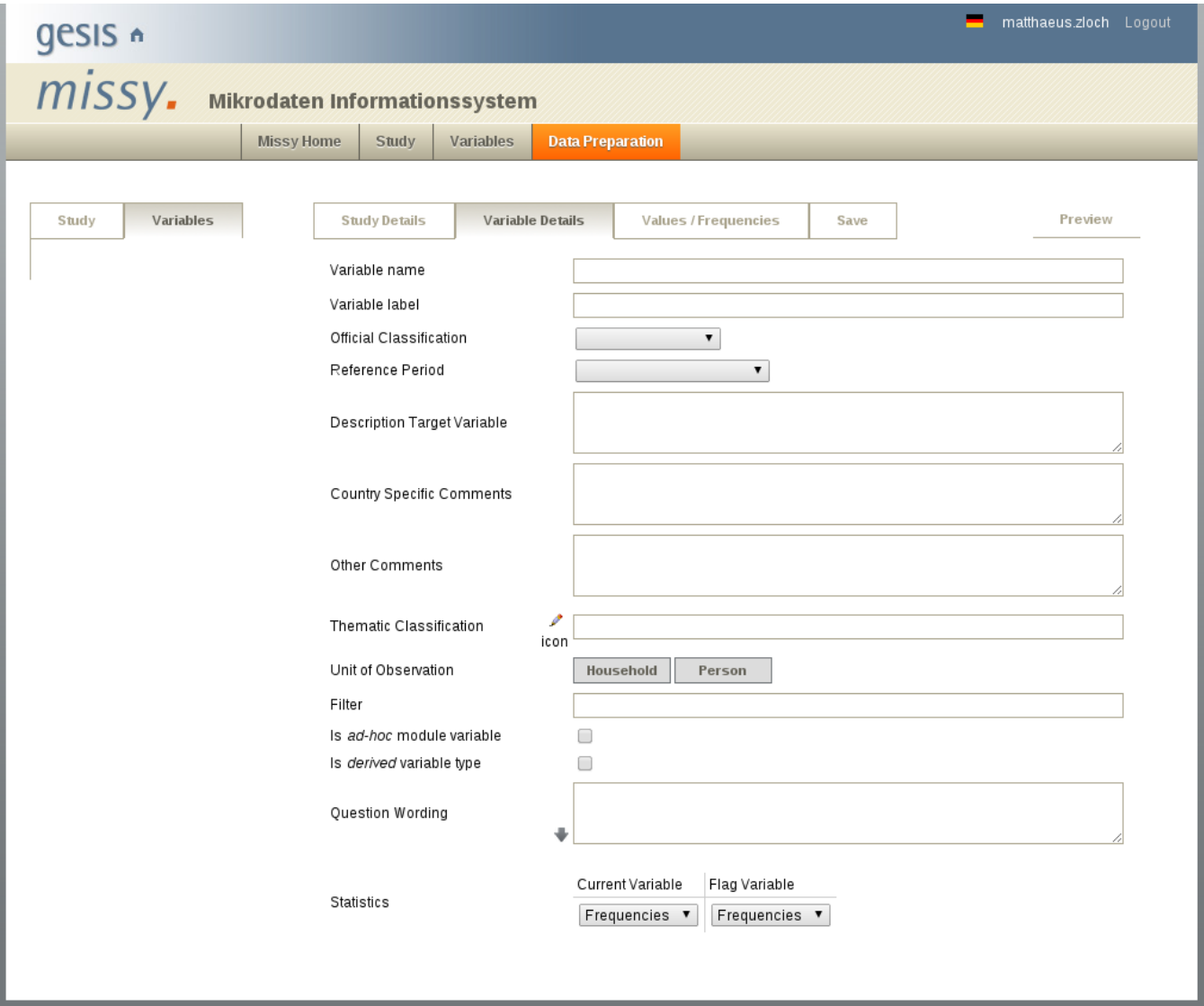
EF117	F34	Stellung im Beruf
EF118	F153	Leitungstätigkeit
EF119		

F35 Beruf (KldB92)

Thematische Gliederung:	Arbeitsmarkt und Erwerbsbeteiligung >> Erwerbsbeteiligung >> Gegenwärtige Erwerbsbeteiligung: Beschäftigungsmerkmale der ersten Erwerbstätigkeit >> Berufliche Platzierung >> Beruf (ab MZ1996)
Andere Erhebungszeitpunkte für diese Variable:	2009 2008 2007 2006 2005 2004 2003 2002 2001 2000 1999 1998 1997 1 EF119 EF119 EF119 EF119 EF119 EF128 EF128 EF128 EF128 EF128 EF128 EF128 EF128 EF
Variablenname:	EF119
Erhebungszeitraum:	2009
Fragebogen:	Erhebungsbogen
Substichprobe:	
Auswahlsatz:	1%
Fragenummer:	35
Fragetext:	Welchen Beruf üben Sie aus?
Erläuterungen zur Frage im Anhang:	-
Filteranweisung:	Anweisung 18: Person (auch Schüler/-in) ist 15 Jahre oder älter.
Filterangaben:	Für Erwerbstätige, auch für geringfügig Beschäftigte (einschl. Personen in Altersteilzeit, die sich nicht mehr am Arbeitsplatz befinden).
Filterangaben (formal):	
Auskunftspflicht:	ja
Anmerkung:	* Kategorienänderung: Zusammenfassung aufgehoben: 224/225/229. * Kategorienänderung: Zusammenlegung von Zusammenfassungen: 331/332 und 344/349. * Kategorienänderung: Herausnahme von Codes: 374/376/378 -> 374.

Häufigkeitsauszählung:

Value Label	Value	Frequency	%	Valid %
Landwirte, Pflanzenschützer	11	1784	0,36%	0,78%
Winzer	12	109	0,02%	0,04%
Landarbeitskräfte	13	314	0,06%	0,13%
Mithelfende Familienangehörige in der Landwirtschaft, a.n.g.	14	316	0,06%	0,13%
Tier-, Pferde-, Fischwirte	23	252	0,05%	0,11%
Tierpfleger und verwandte Berufe, a.n.g.	24	178	0,03%	0,07%
31/32: Verwalter in der Land- und Tierwirtschaft(31); Land-, Tierwirtschaftsberater, Agraringenieure, Agrartechniker(32)	32	165	0,03%	0,07%


gesis
matthaeus.zloch Logout
missy. Mikrodaten Informationssystem
Missy Home
Study
Variables
Data Preparation
Study
Variables
Study Details
Variable Details
Values / Frequencies
Save
Preview
Variable name
Variable label
Official Classification
Reference Period
Description Target Variable
Country Specific Comments
Other Comments
Thematic Classification
icon
Unit of Observation
Household
Person
Filter
Is ad-hoc module variable
Is derived variable type
Question Wording
Statistics
Current Variable
Flag Variable
Frequencies
Frequencies

Requirements to Software Developers

- Complex data model to represent use-cases
- Focus lies on reusability
- Flexible Web Application Framework and modern architecture
  - Service-oriented
  - Semantic Web technologies
- Creation of an abstract framework and architecture
  - should be well designed to be able to be extended and reusable

What comes next?

- We have seen Missy as an use-case with several fields which holds values

- I would like to use DDI
- What are the main structures of DDI-L, Thomas?

What comes next?

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy

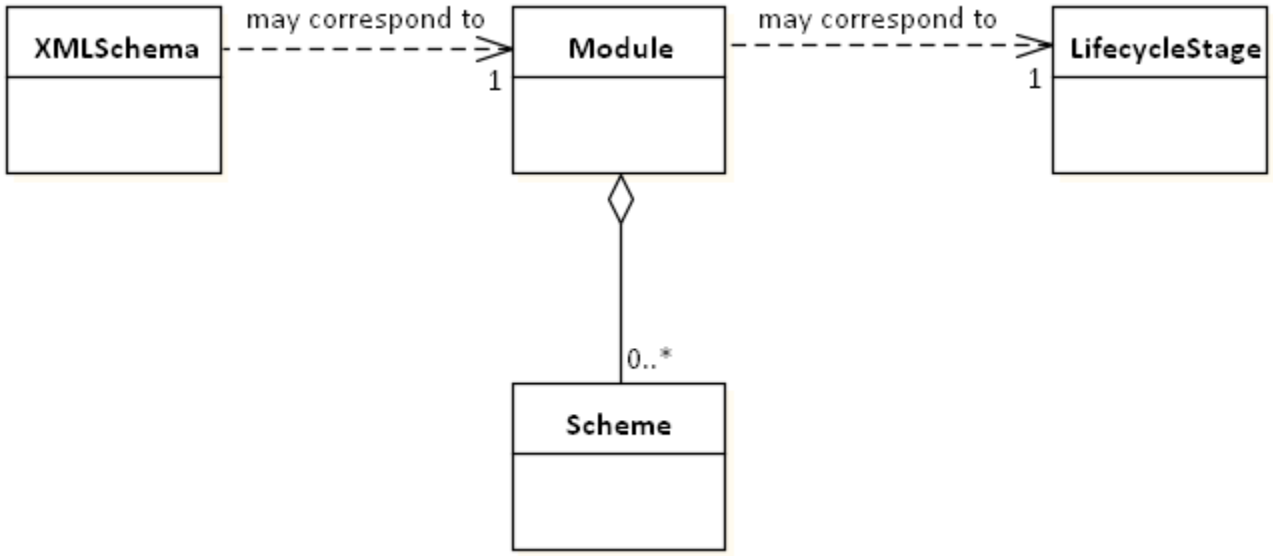
Presentation

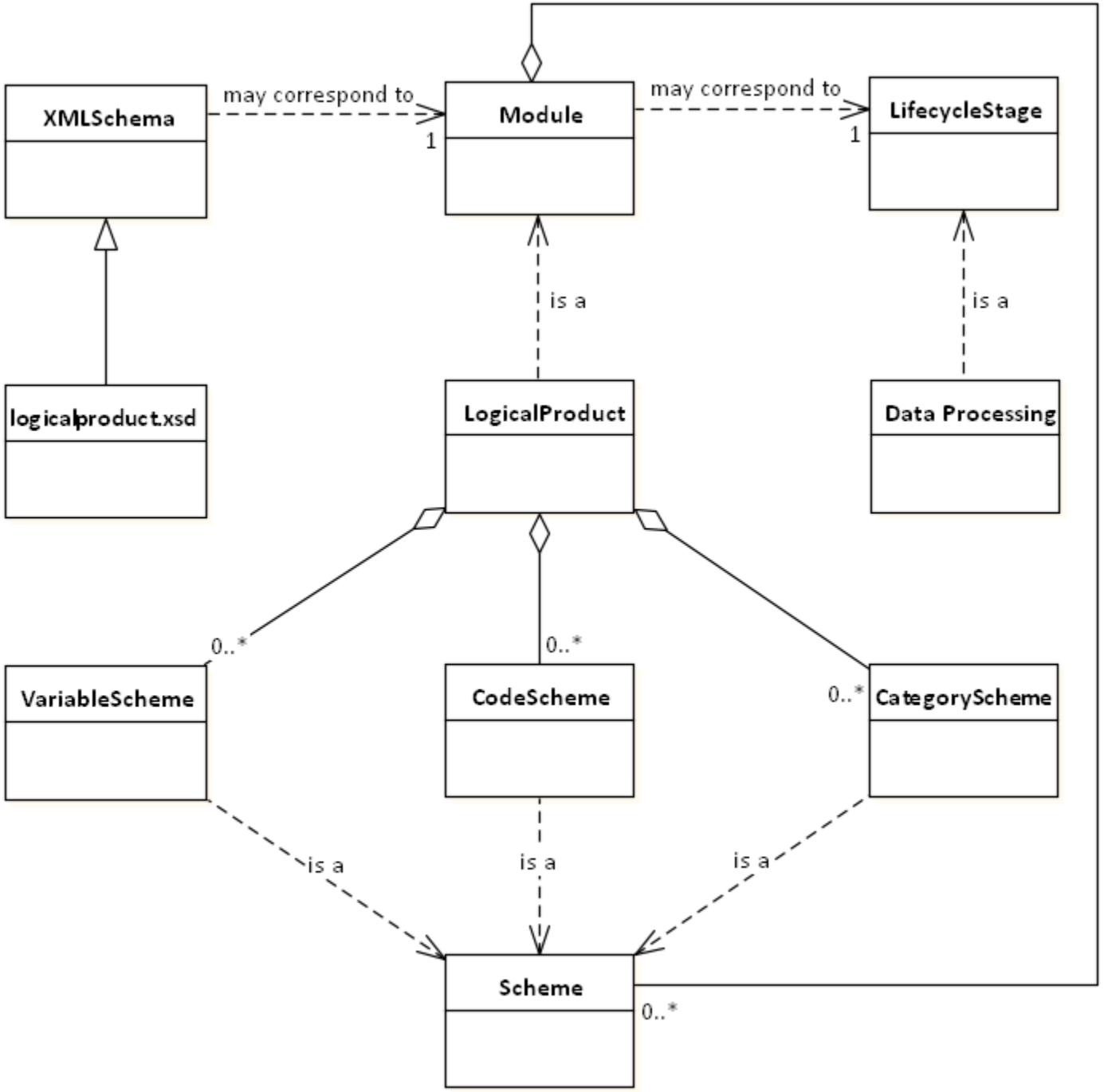
- DDI-L Overview
- Identification, Versioning, Maintenance
- **DDI-L Main Structures**
- DDI Instance
- Study Unit

- Software Architecture
  - Multitier
  - MVC
- Missy Data Model
  - Extendable Data Model

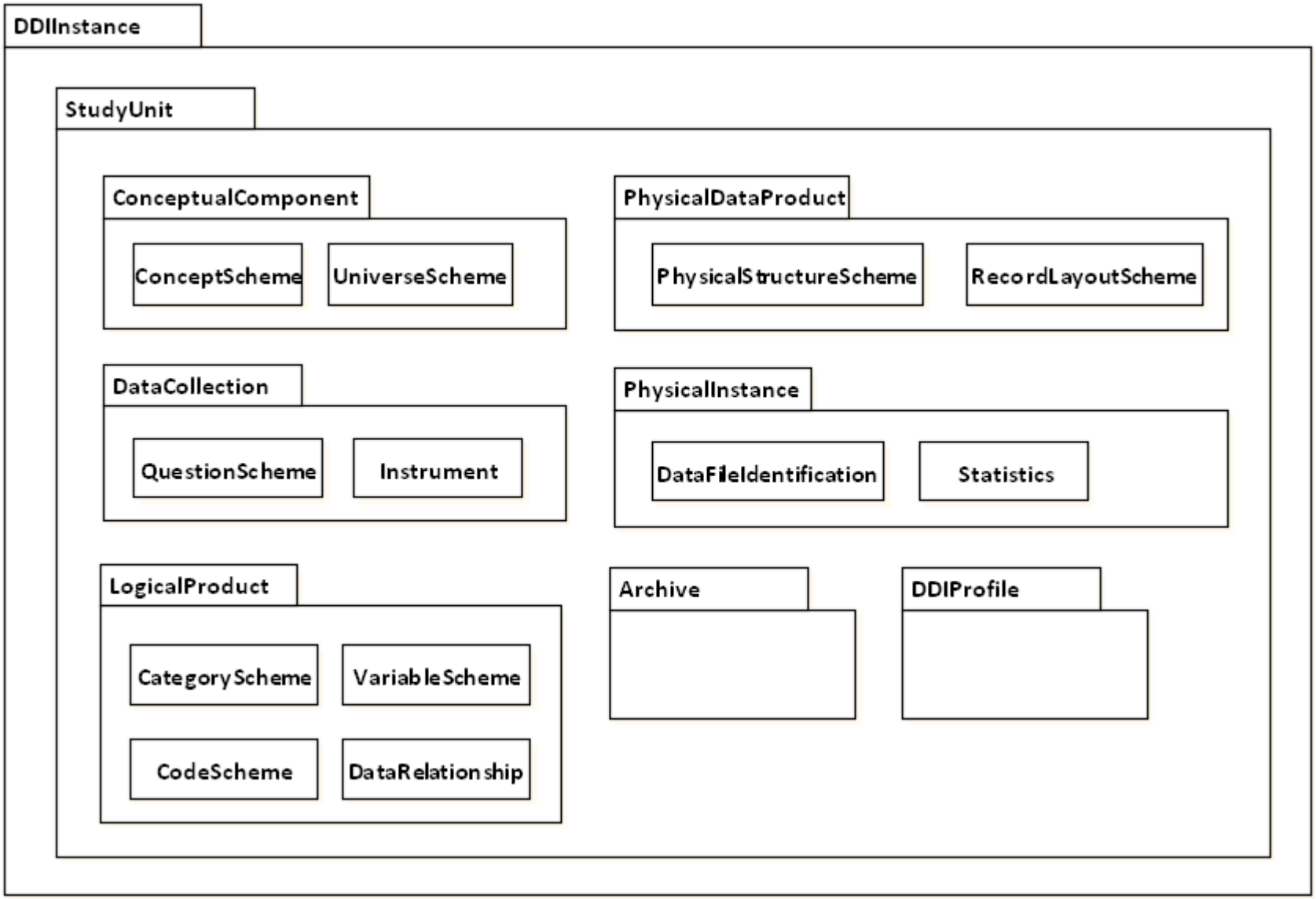
- Conceptual Component
- Logical Products
- Data Collection
- disco-model

Business Layer


XMLSchema
may correspond to
1
Module
may correspond to
1
LifecycleStage
0..*
Scheme


XMLSchema
may correspond to
1
Module
may correspond to
1
LifecycleStage
is a
is a
logicalproduct.xsd
LogicalProduct
Data Processing
0..*
0..*
0..*
VariableScheme
CodeScheme
CategoryScheme
is a
is a
is a
Scheme
0..*

Modules


DDIInstance
StudyUnit
ConceptualComponent
ConceptScheme
UniverseScheme
PhysicalDataProduct
PhysicalStructureScheme
RecordLayoutScheme
DataCollection
QuestionScheme
Instrument
PhysicalInstance
DataFileIdentification
Statistics
LogicalProduct
CategoryScheme
VariableScheme
CodeScheme
DataRelationship
Archive
DDIProfile

What comes next?

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy

Presentation

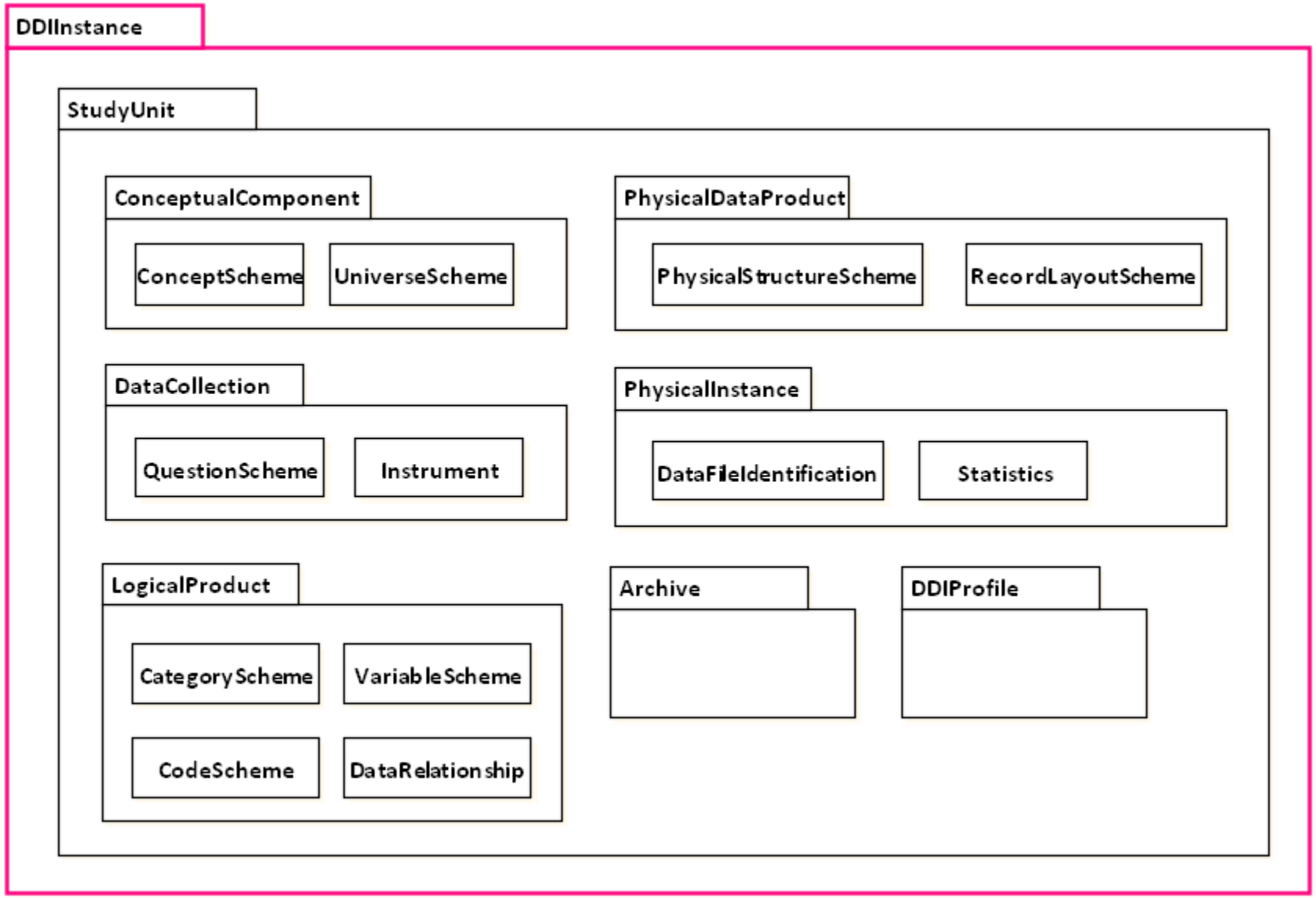
- DDI-L Overview
- Identification, Versioning, Maintenance
- DDI-L Main Structures
- **DDI Instance**
- Study Unit

- Software Architecture
  - Multitier
  - MVC
- Missy Data Model
  - Extendable Data Model

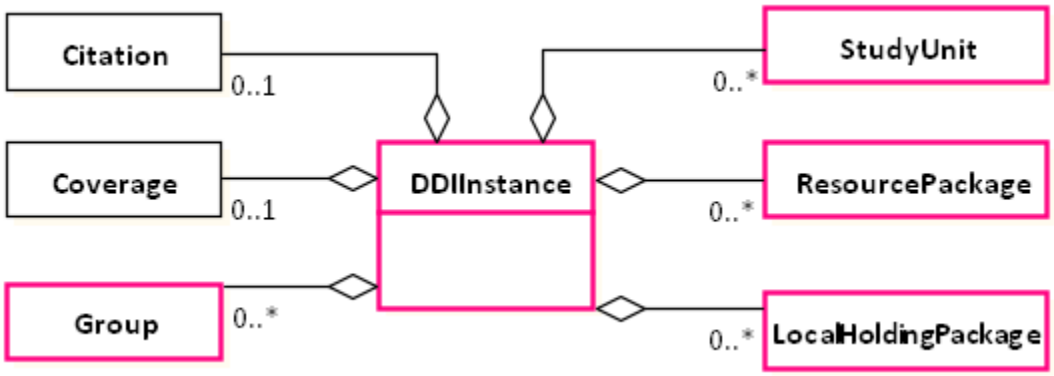
- Conceptual Component
- Logical Products
- Data Collection
- disco-model

Business Layer

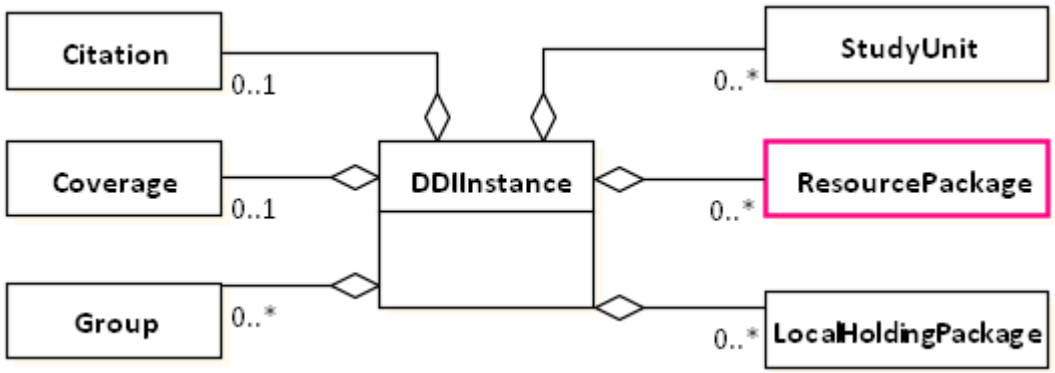
DDIInstance



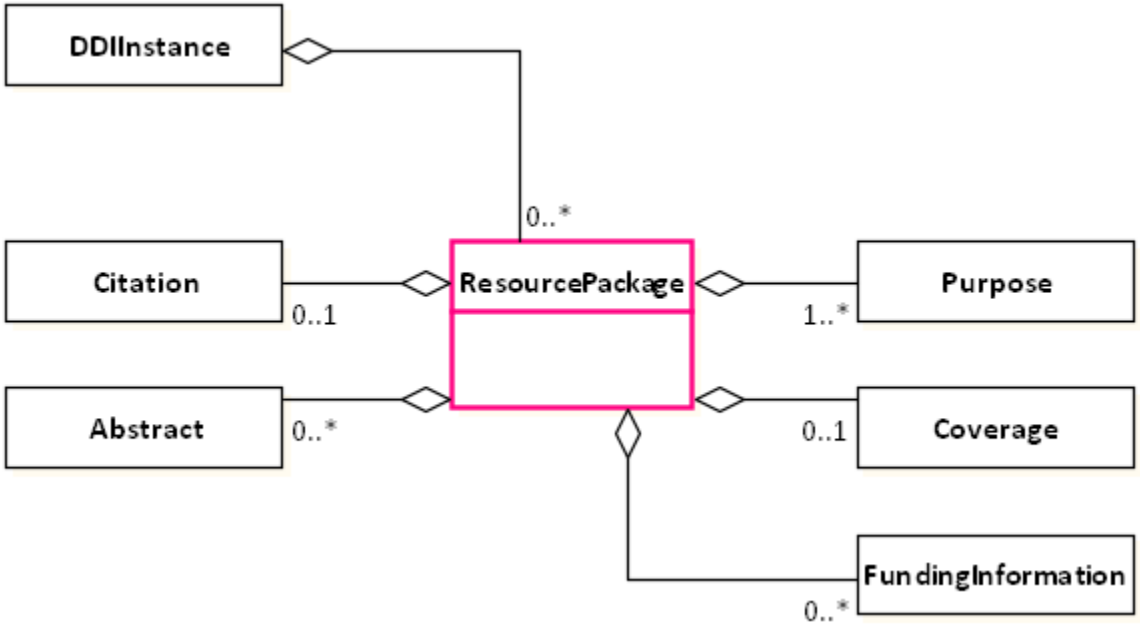
DDIInstance



ResourcePackage



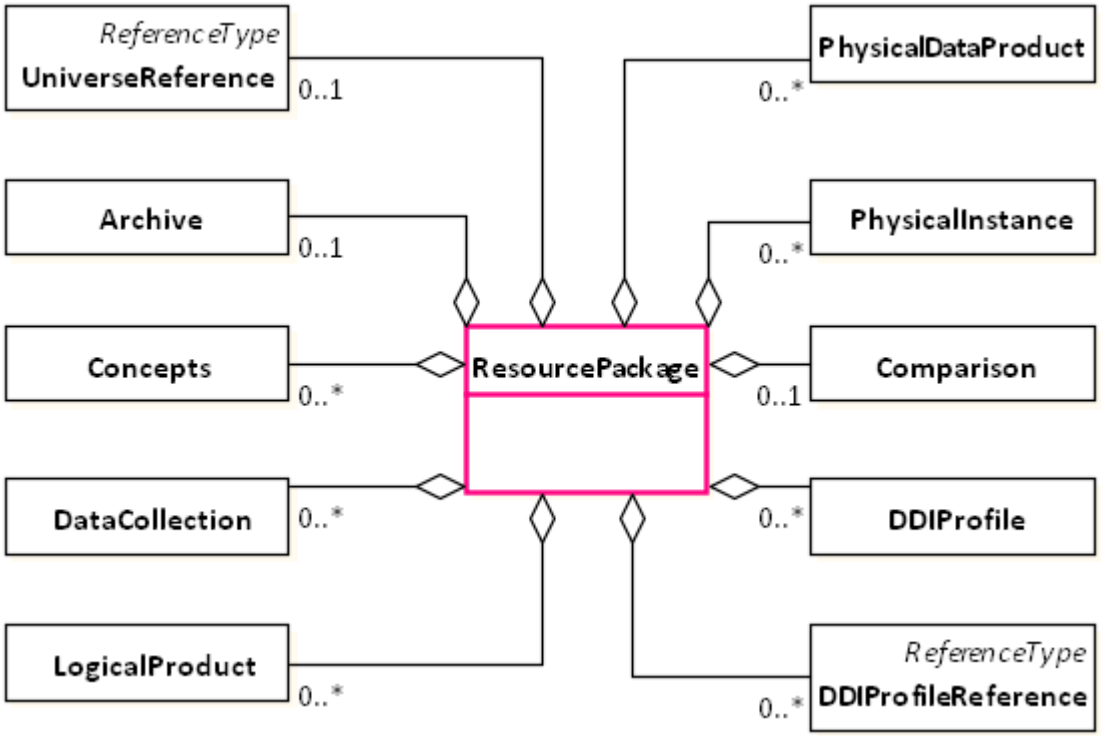
ResourcePackage



Allows packaging of any maintainable item as a resource item

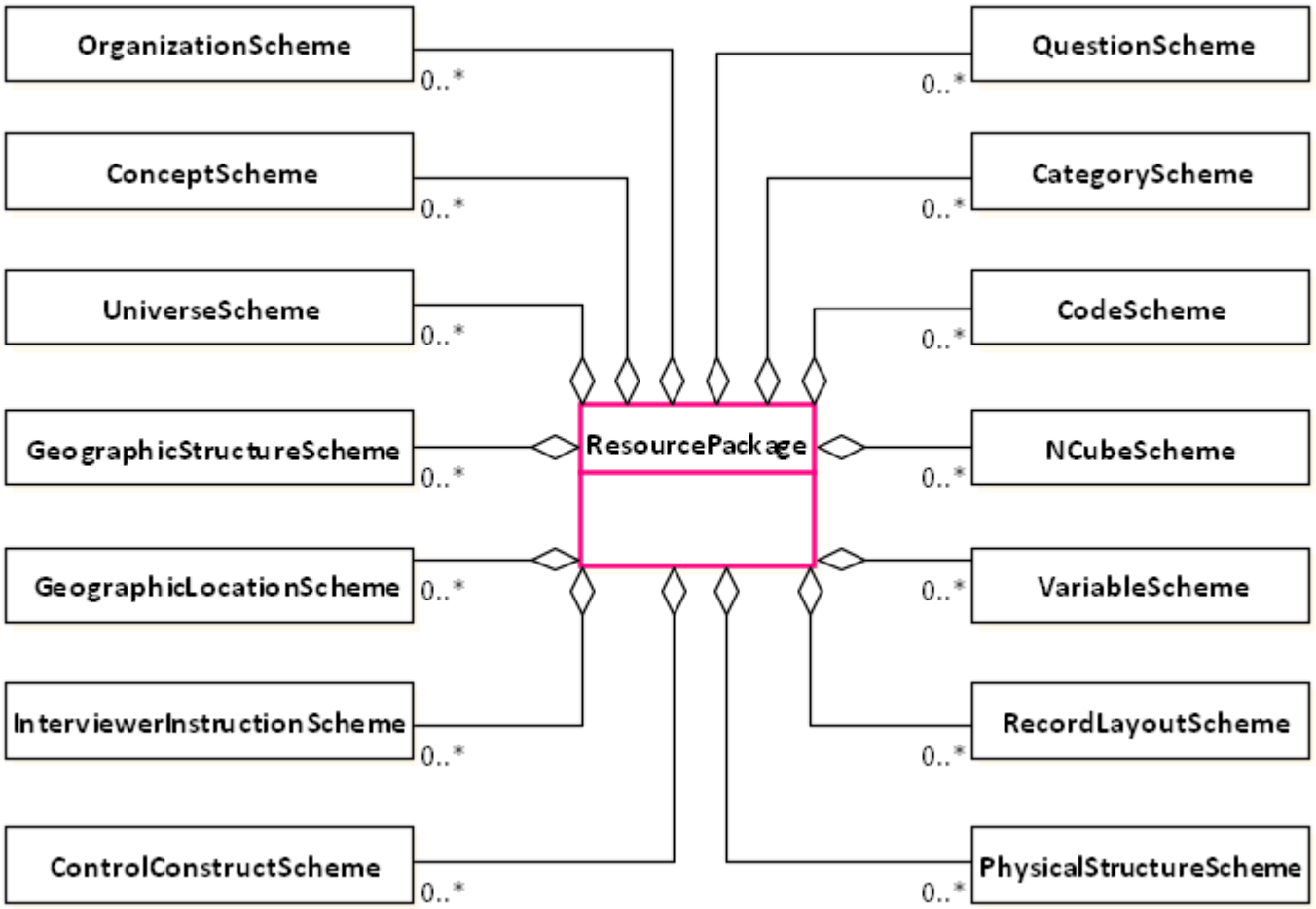
Structure to publish **non-study-specific** materials for reuse

ResourcePackage



Any module (except: StudyUnit, Group, LocalHoldingPackage)

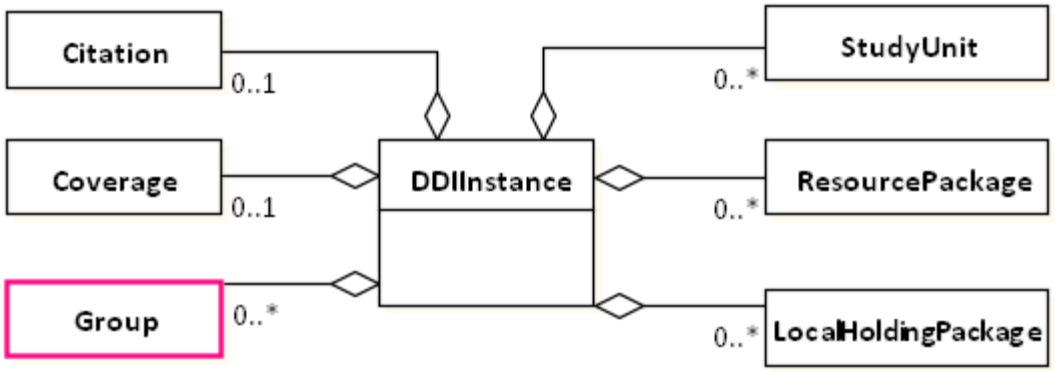
ResourcePackage



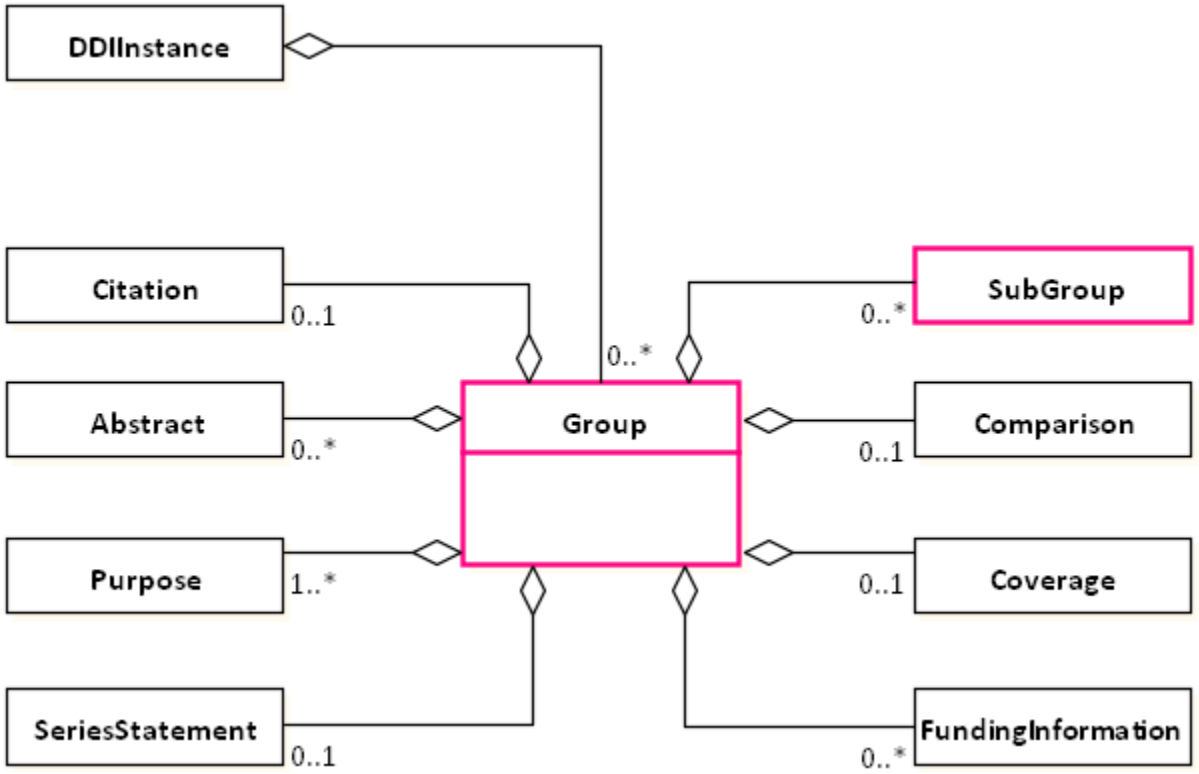
Any Scheme

Internal or external references to schemes and elements within schemes

Group

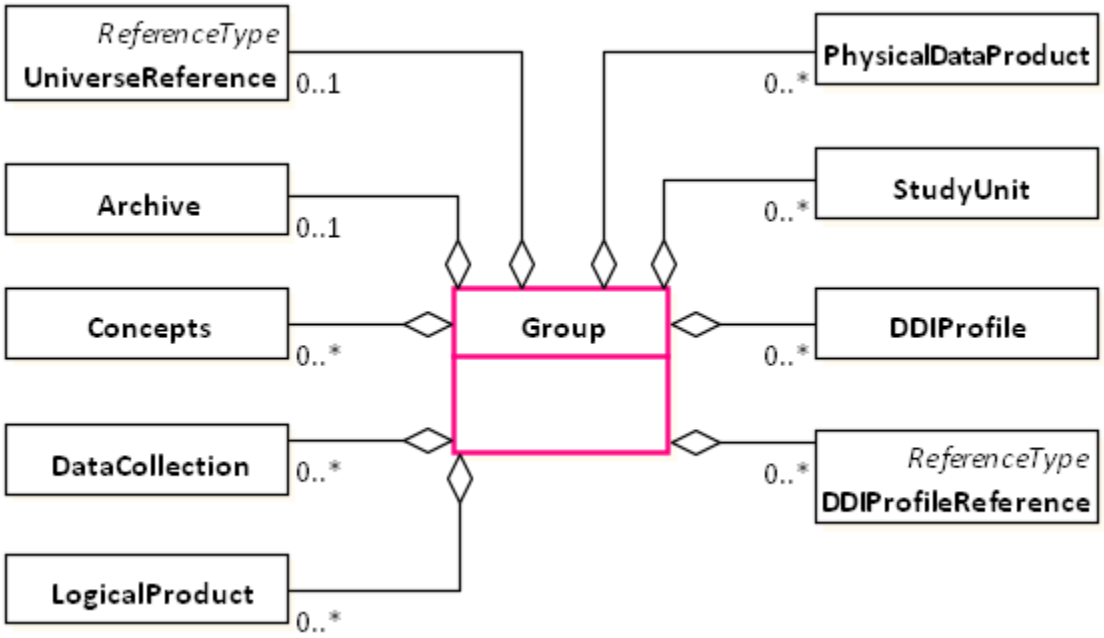

Citation
0..1
Coverage
0..1
Group
0..*
DDIInstance
StudyUnit
0..*
ResourcePackage
0..*
LocalHoldingPackage
0..*

Group

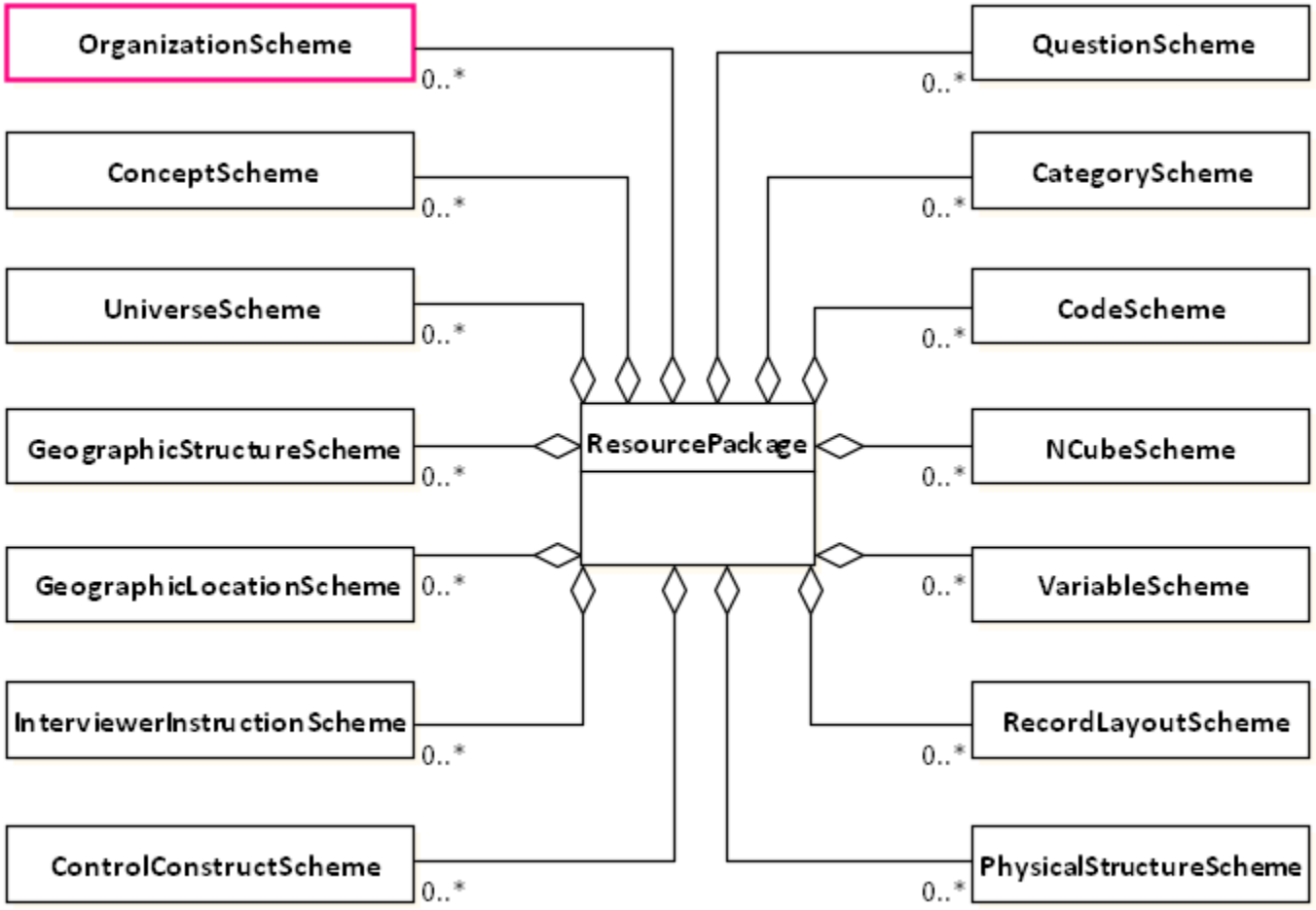


Groups allow inheritance

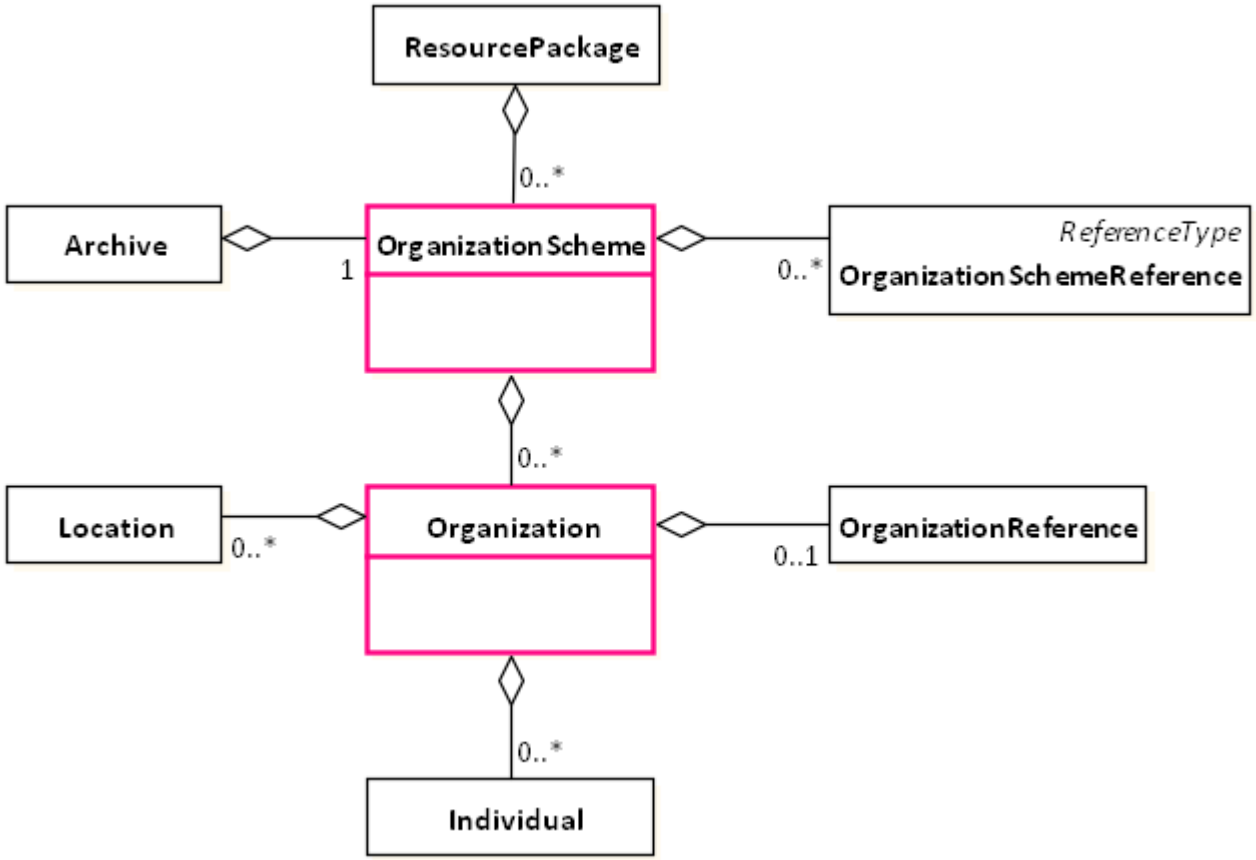
Group


ReferenceType
UniverseReference
0..1
Archive
0..1
Concepts
0..*
DataCollection
0..*
LogicalProduct
0..*
Group
PhysicalDataProduct
0..*
StudyUnit
0..*
DDIProfile
0..*
ReferenceType
DDIProfileReference
0..*

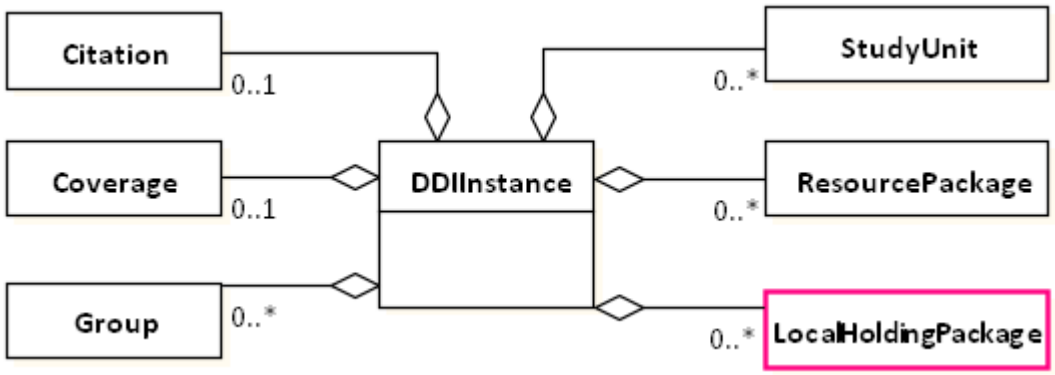
Organization



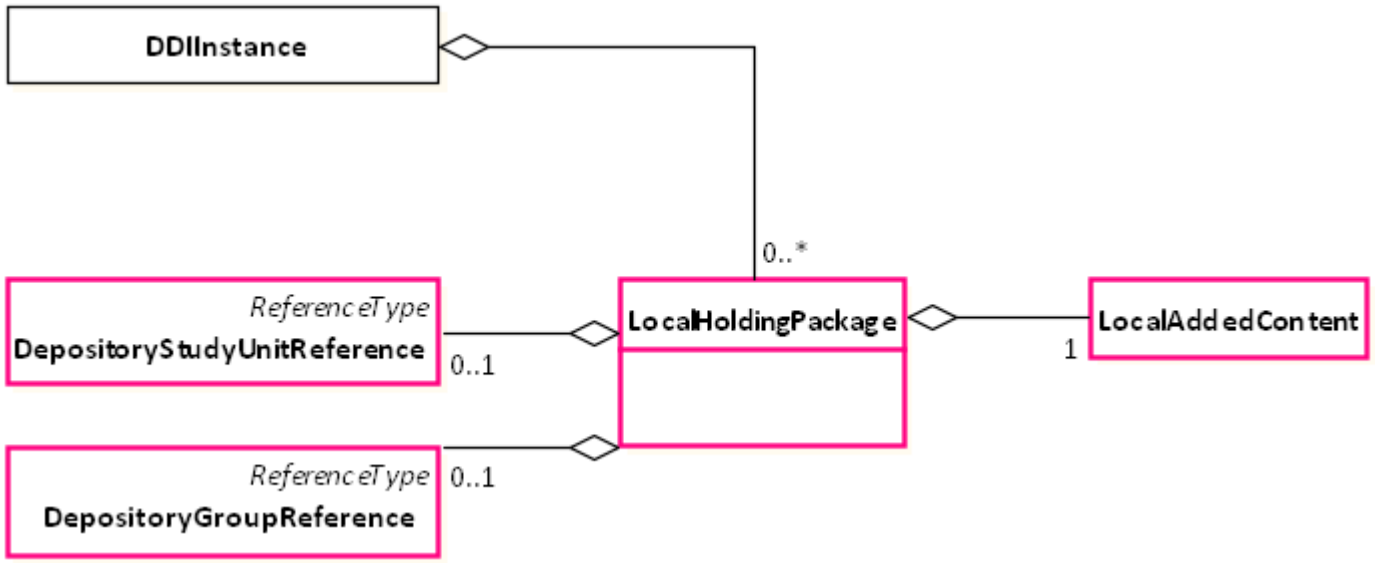
Organization



LocalHoldingPackage



LocalHoldingPackage



- Add local content to a deposited study unit or group
- without changing the version of the study unit, group

What comes next?

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy

Presentation

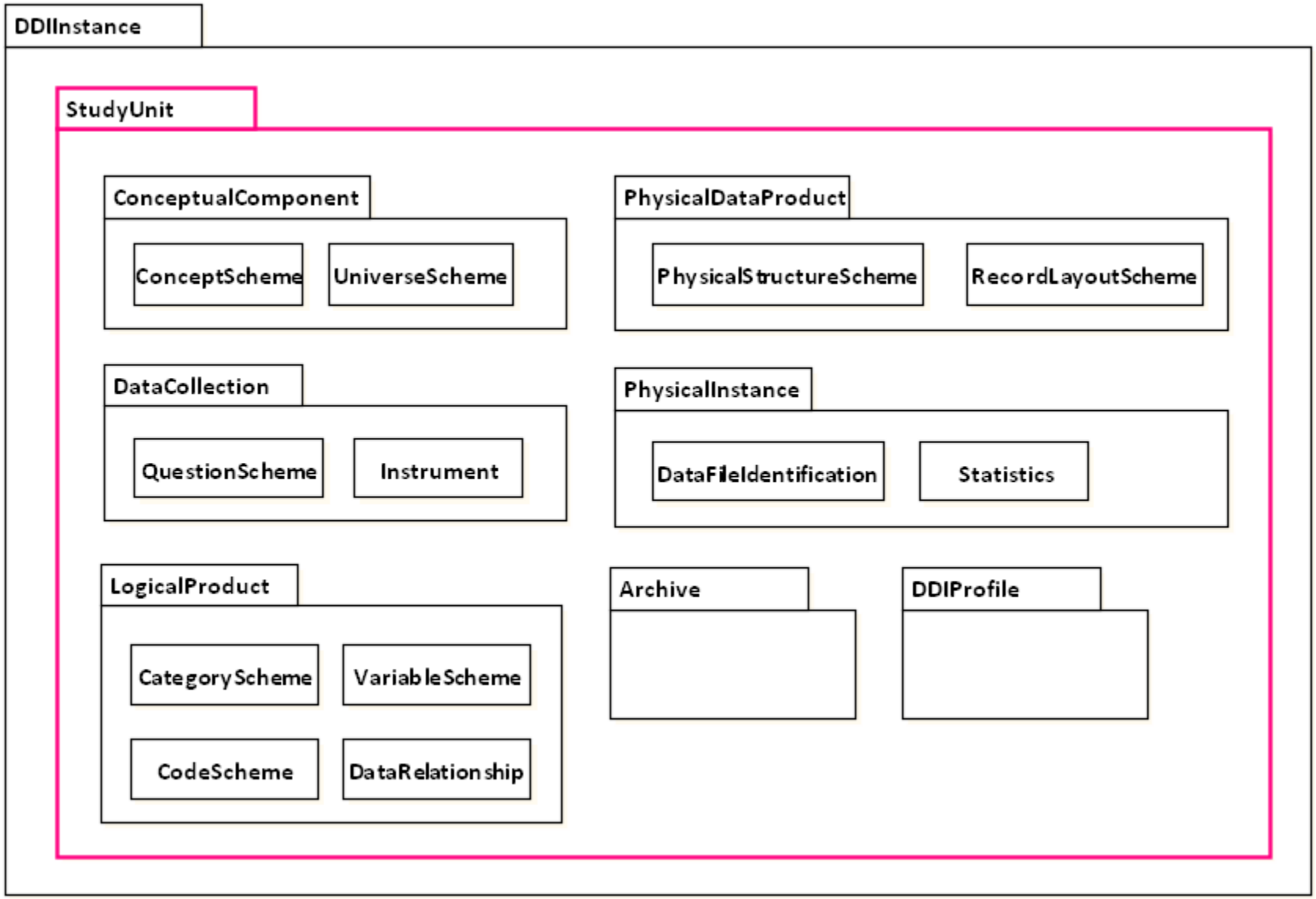
- DDI-L Overview
- Identification, Versioning, Maintenance
- DDI-L Main Structures
- DDI Instance
- **Study Unit**

- Software Architecture
  - Multitier
  - MVC
- Missy Data Model
  - Extendable Data Model

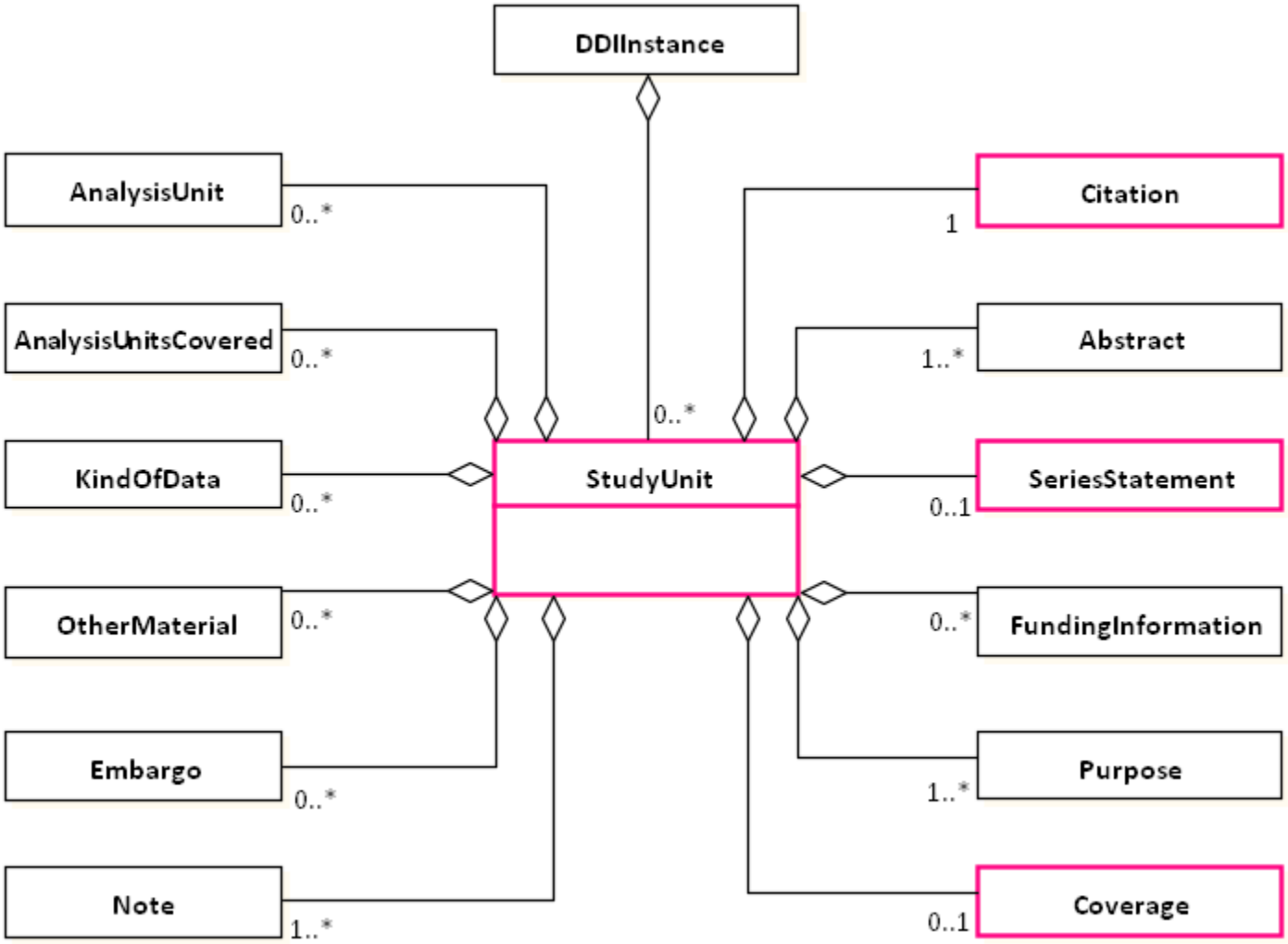
- Conceptual Component
- Logical Products
- Data Collection
- disco-model

Business Layer

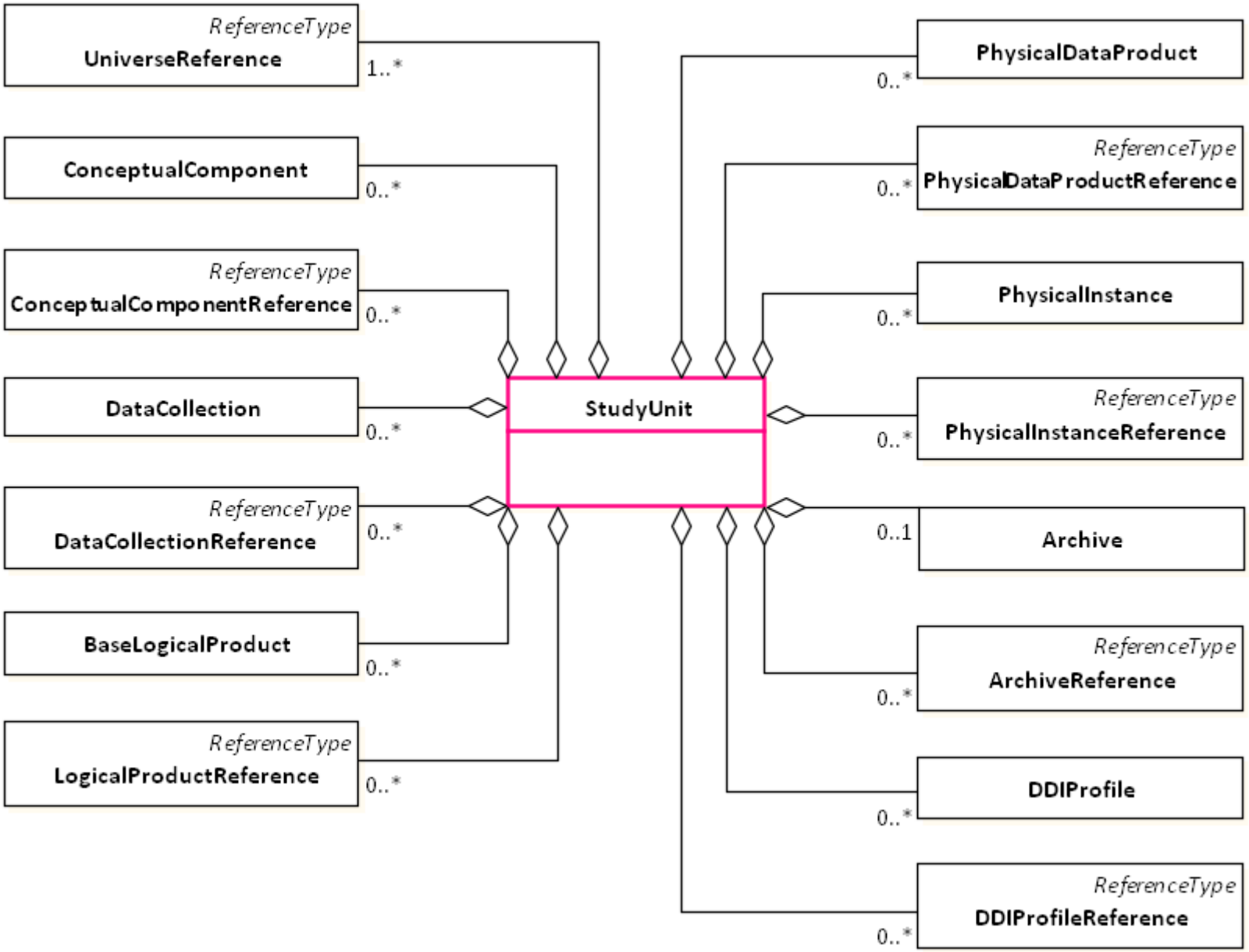
StudyUnit


DDIInstance
StudyUnit
ConceptualComponent
ConceptScheme
UniverseScheme
PhysicalDataProduct
PhysicalStructureScheme
RecordLayoutScheme
DataCollection
QuestionScheme
Instrument
PhysicalInstance
DataFileIdentification
Statistics
LogicalProduct
CategoryScheme
VariableScheme
CodeScheme
DataRelationship
Archive
DDIProfile

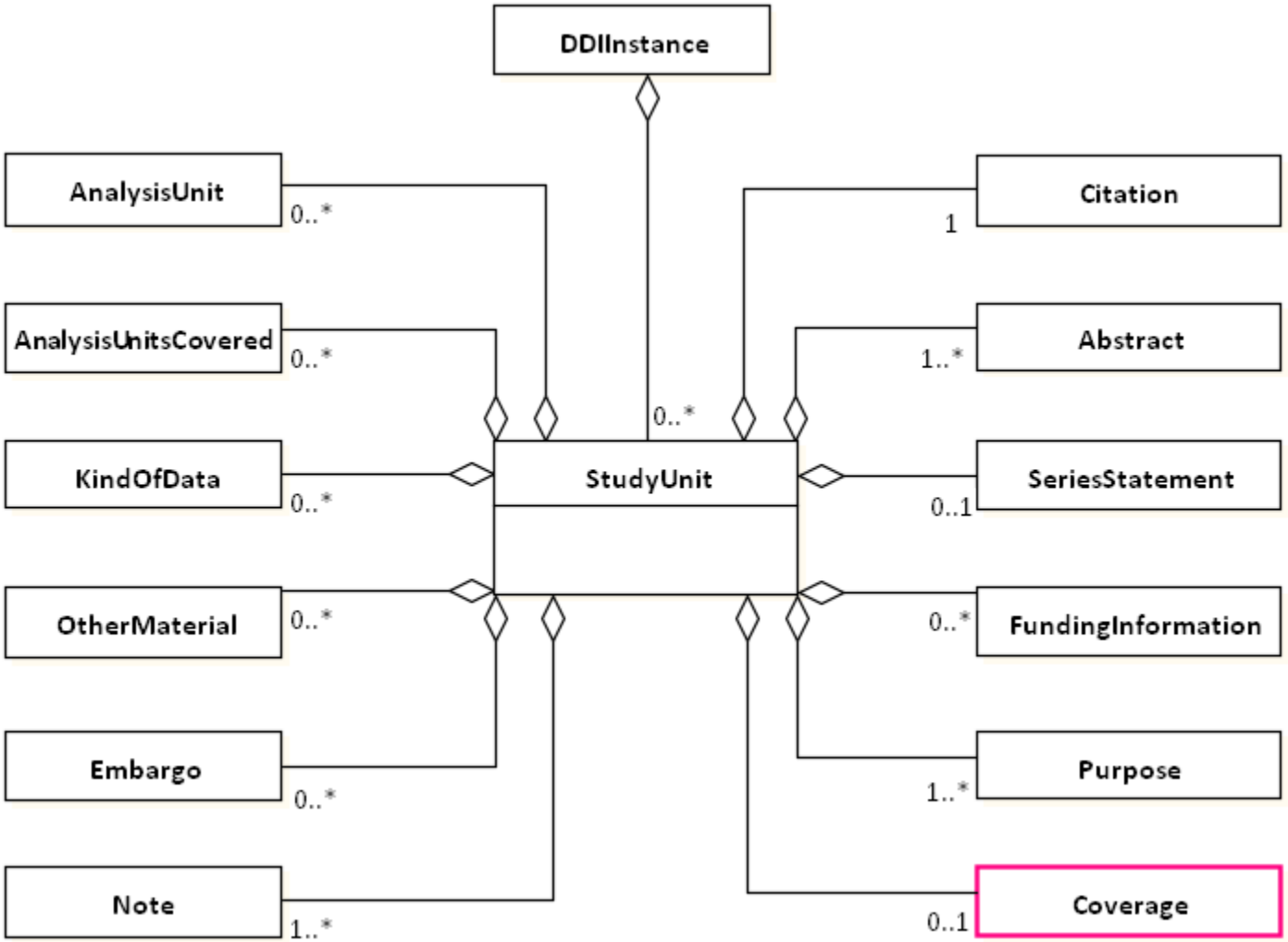
StudyUnit


DDIInstance
AnalysisUnit
0..*
AnalysisUnitsCovered
0..*
KindOfData
0..*
OtherMaterial
0..*
Embargo
0..*
Note
1..*
0..*
StudyUnit
Citation
1
Abstract
1..*
SeriesStatement
0..1
FundingInformation
0..*
Purpose
1..*
Coverage
0..1

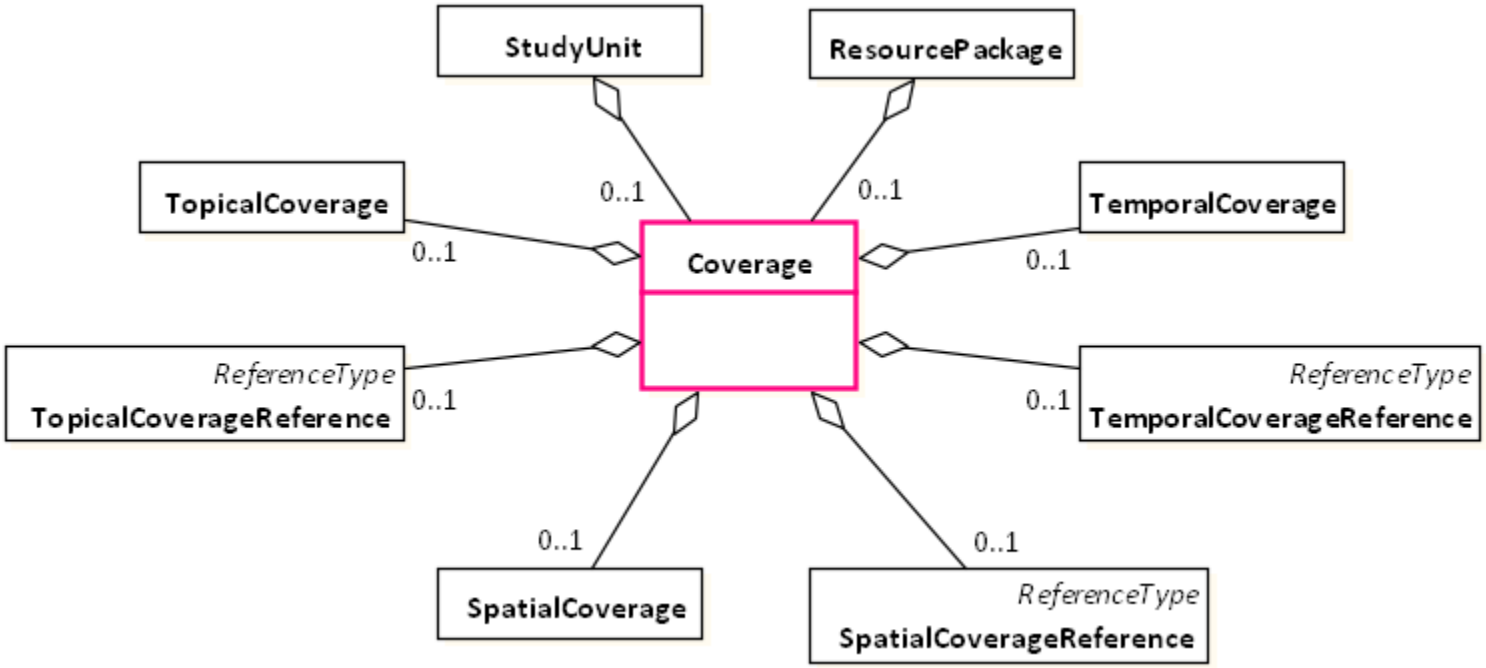
StudyUnit


ReferenceType
UniverseReference
1..*
ConceptualComponent
0..*
ReferenceType
ConceptualComponentReference
0..*
DataCollection
0..*
ReferenceType
DataCollectionReference
0..*
BaseLogicalProduct
0..*
ReferenceType
LogicalProductReference
0..*
StudyUnit
PhysicalDataProduct
0..*
ReferenceType
PhysicalDataProductReference
0..*
PhysicalInstance
0..*
ReferenceType
PhysicalInstanceReference
0..*
0..1
Archive
ReferenceType
ArchiveReference
0..*
DDIProfile
0..*
ReferenceType
DDIProfileReference
0..*

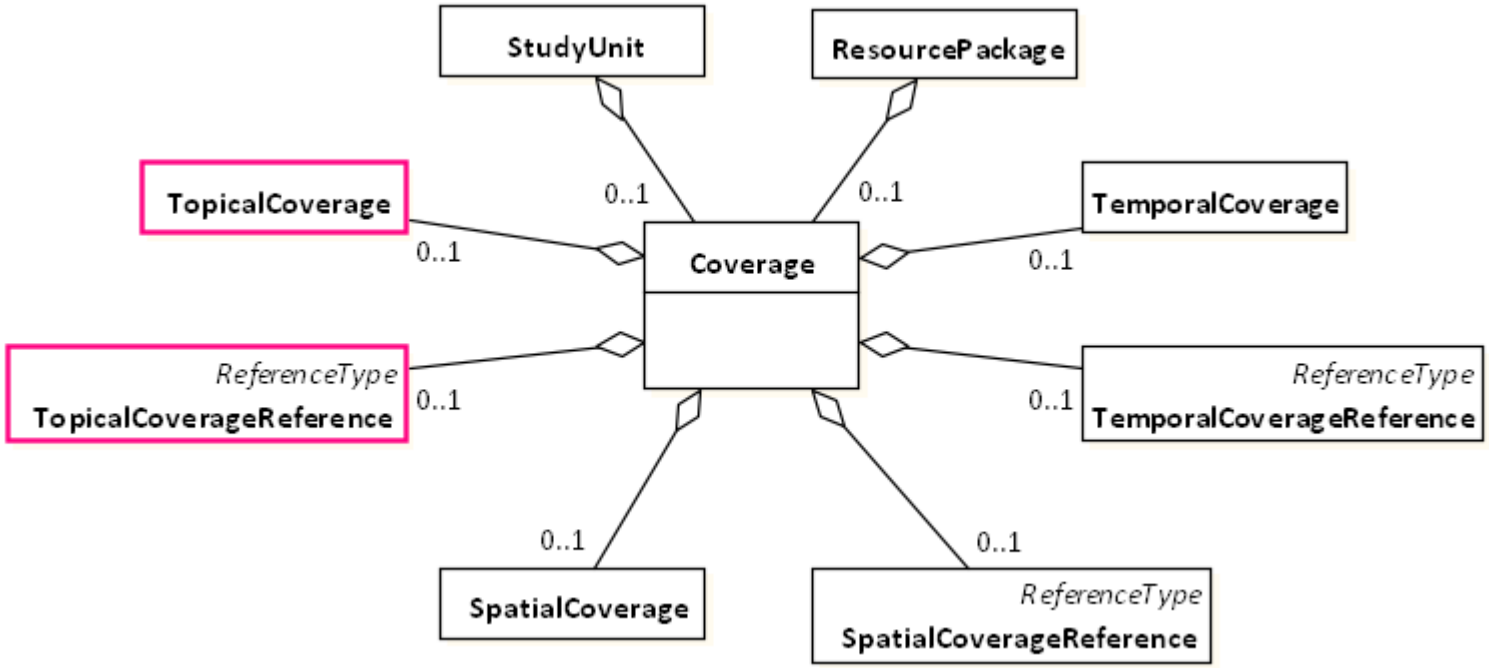
Coverage


DDIInstance
AnalysisUnit
0..*
AnalysisUnitsCovered
0..*
KindOfData
0..*
OtherMaterial
0..*
Embargo
0..*
Note
1..*
0..*
StudyUnit
Citation
1
Abstract
1..*
SeriesStatement
0..1
FundingInformation
0..*
Purpose
1..*
Coverage
0..1

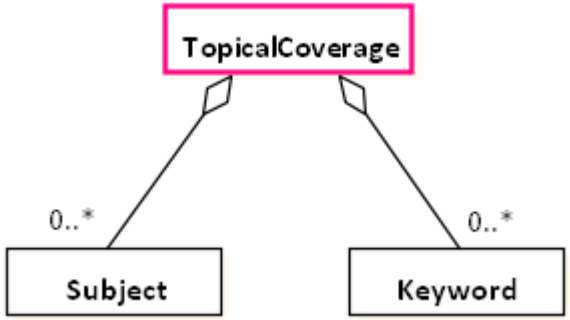
Coverage


StudyUnit
ResourcePackage
TopicalCoverage
TemporalCoverage
Coverage
ReferenceType
TopicalCoverageReference
ReferenceType
TemporalCoverageReference
SpatialCoverage
ReferenceType
SpatialCoverageReference
0..1
0..1
0..1
0..1
0..1
0..1
0..1
0..1

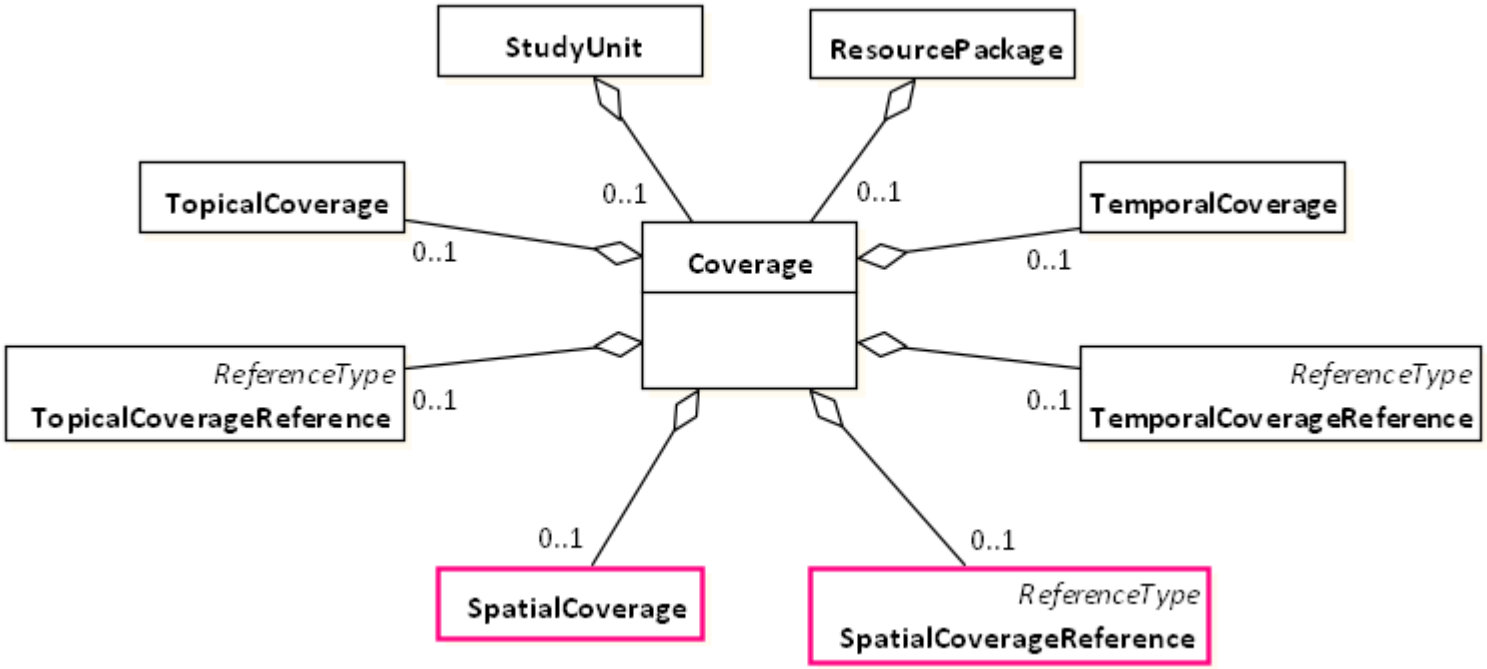
TemporalCoverage



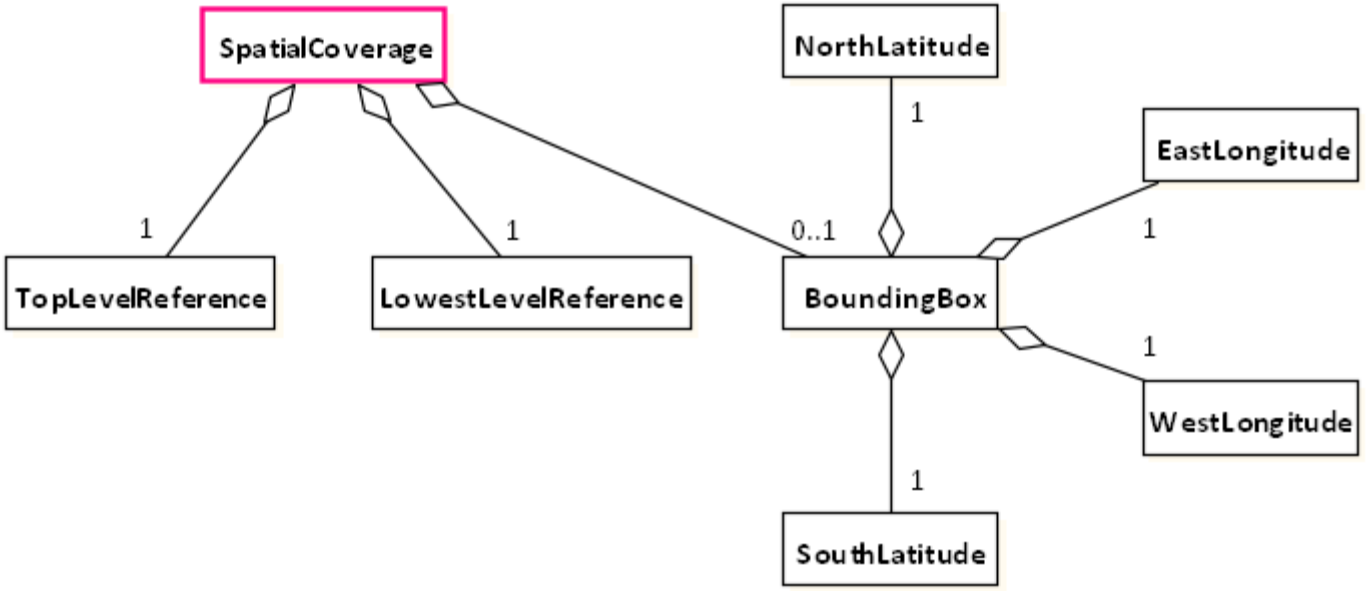
TopicalCoverage



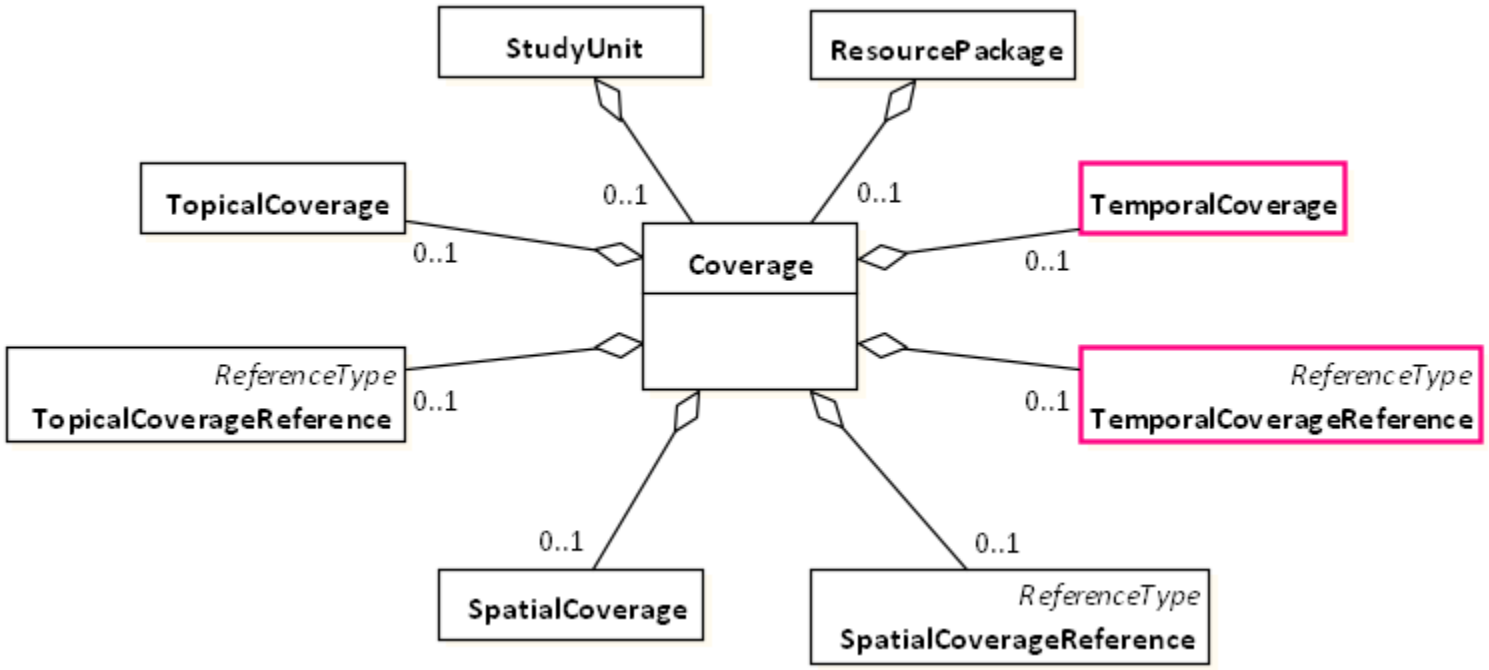
SpatialCoverage



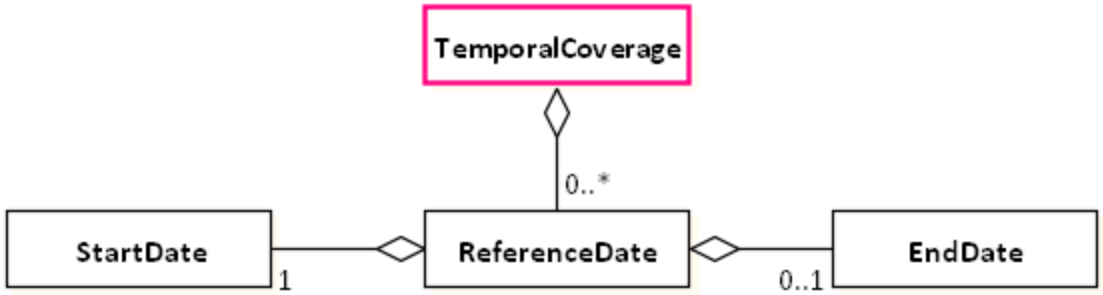
SpatialCoverage



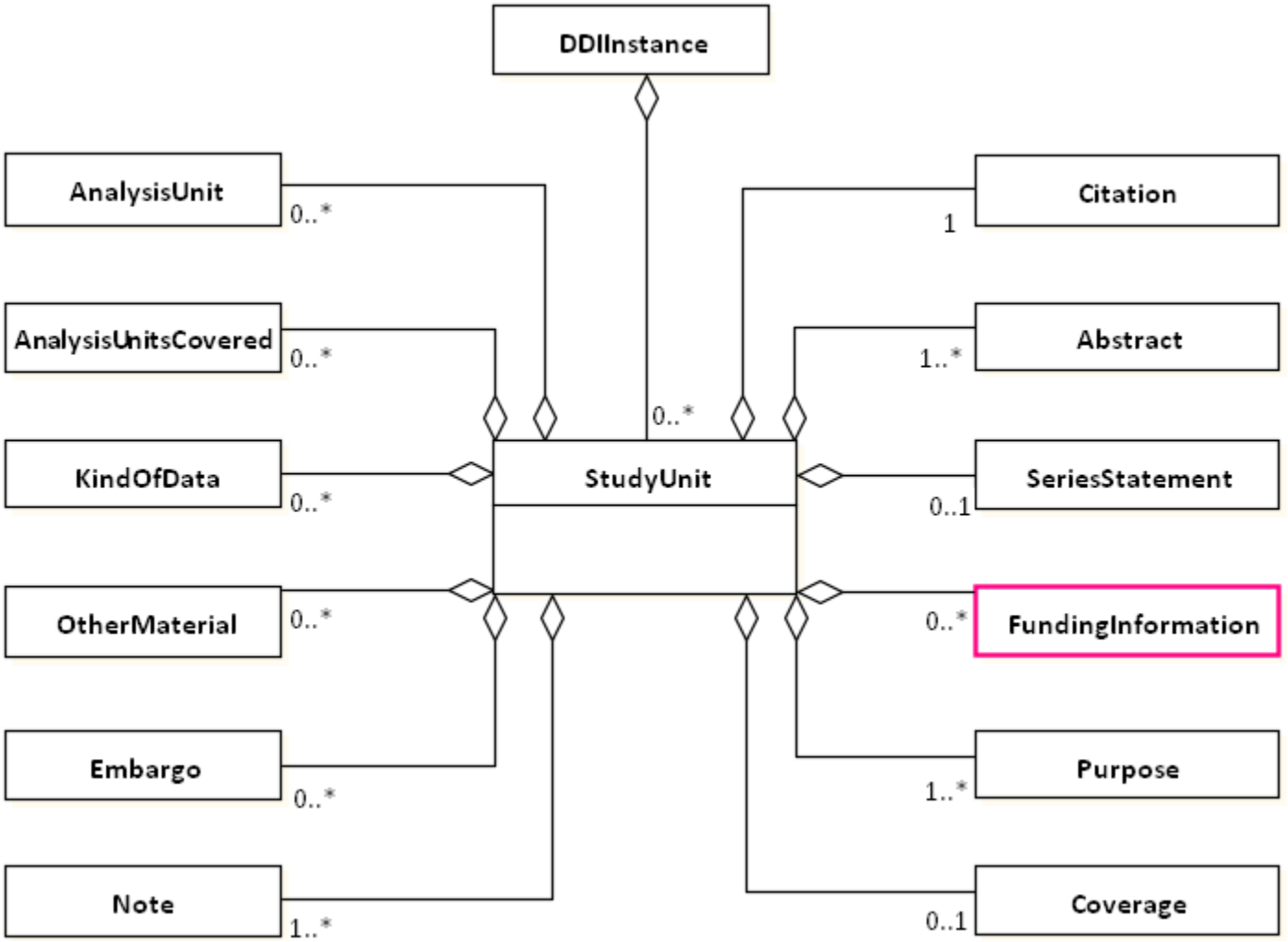
TopicalCoverage



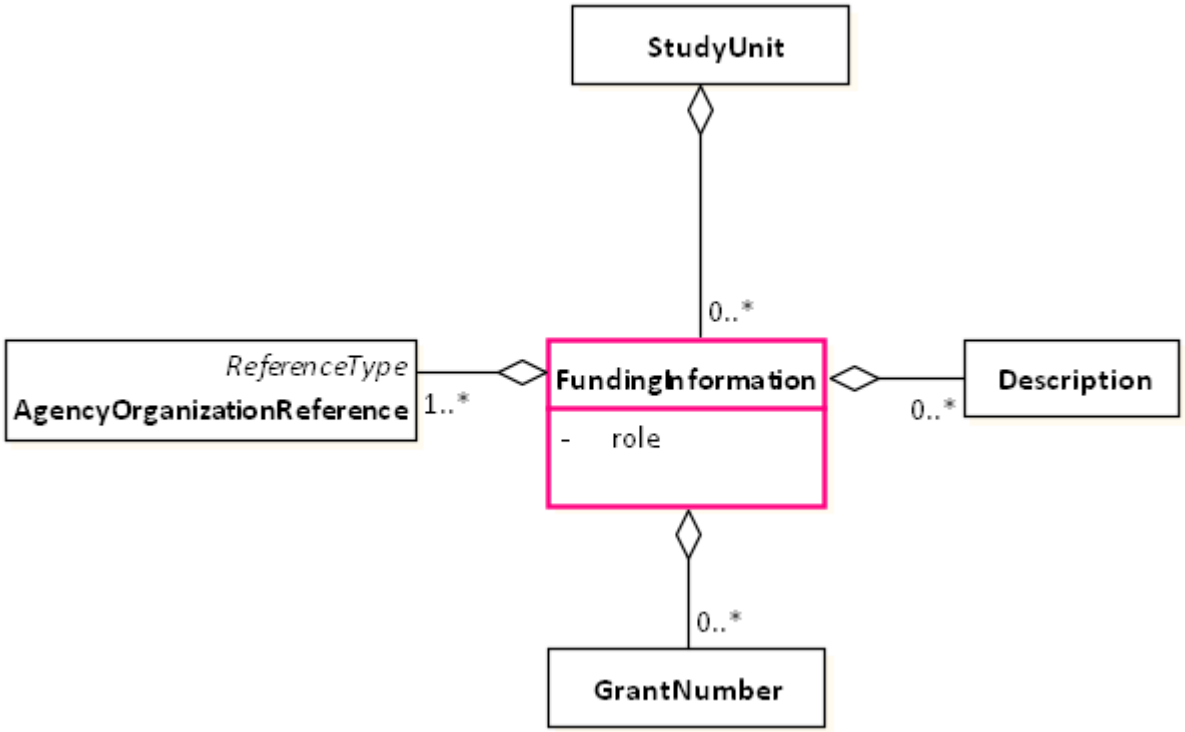
TemporalCoverage



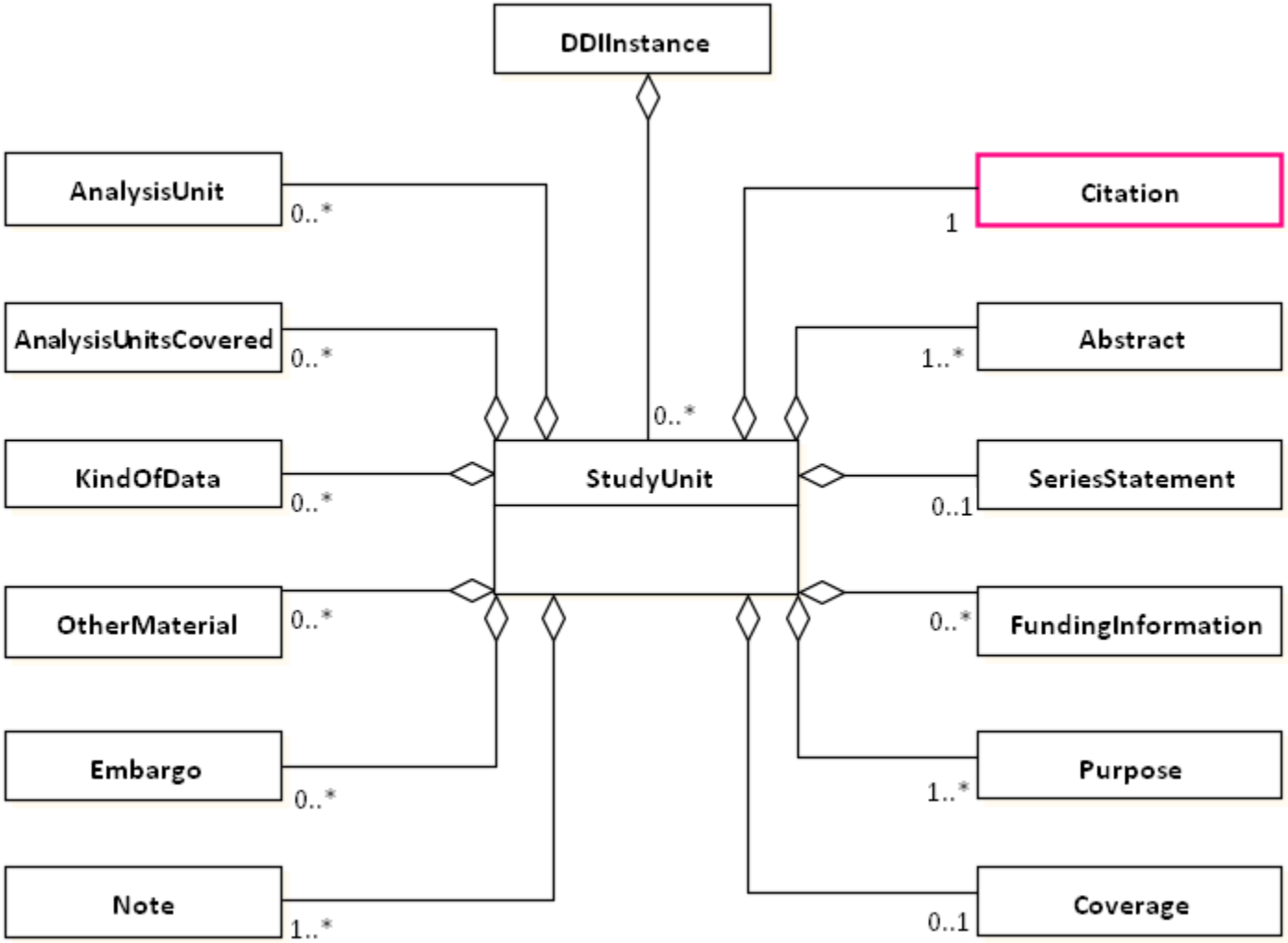
FundingInformation



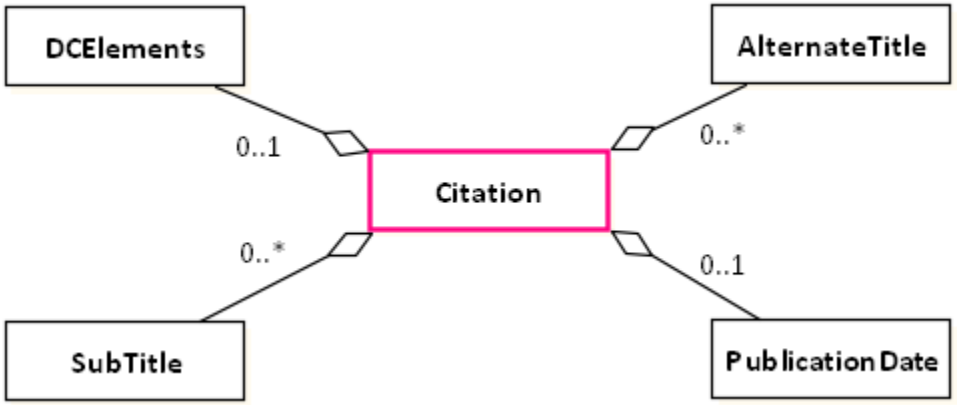
FundingInformation



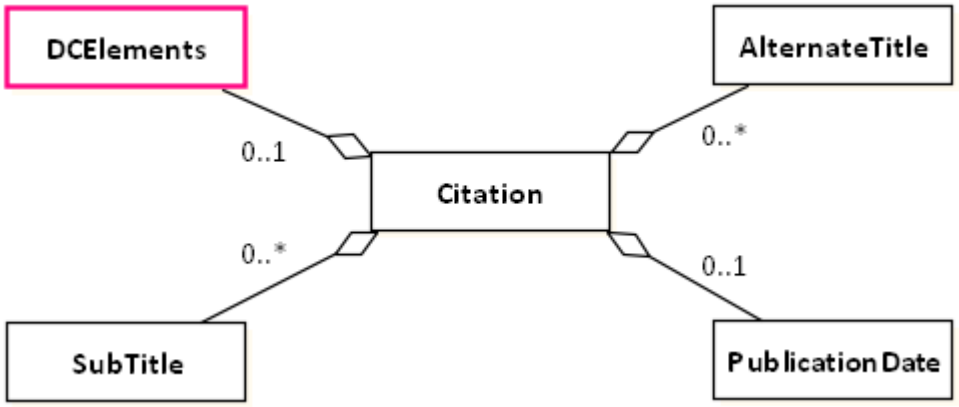
Citation


DDIInstance
AnalysisUnit
0..*
AnalysisUnitsCovered
0..*
KindOfData
0..*
OtherMaterial
0..*
Embargo
0..*
Note
1..*
0..*
StudyUnit
Citation
1
Abstract
1..*
SeriesStatement
0..1
FundingInformation
0..*
Purpose
1..*
Coverage
0..1

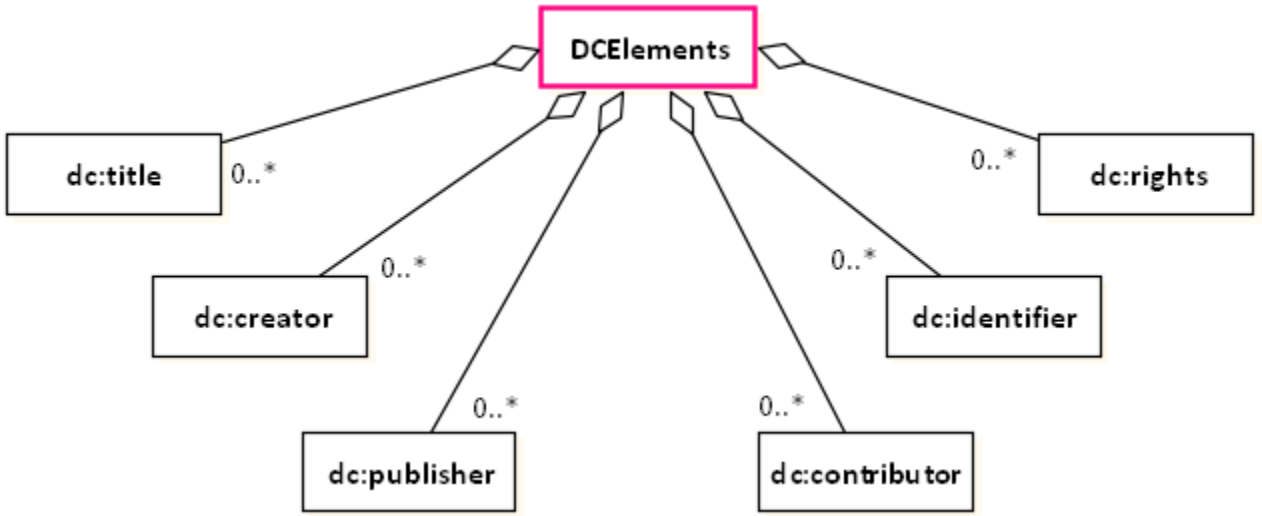
Citation


DCElements
AlternateTitle
0..1
0..*
Citation
0..*
0..1
SubTitle
Publication Date

Citation


DCElements
AlternateTitle
0..1
0..*
Citation
0..*
0..1
SubTitle
Publication Date

Citation


DCElements
dc:title
0..*
dc:creator
0..*
dc:publisher
0..*
dc:contributor
0..*
dc:identifier
0..*
dc:rights
0..*

How the presentation is continued

- Ok, I have shown you
  - the main structures as well as
  - the main modules of DDI-Lifecycle

- If you want to use DDI you have to have a model or idea of your software architecture
- How would your software architecture look like?

What comes next?

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy

Presentation

- DDI-L Overview
- Identification, Versioning, Maintenance
- DDI-L Main Structures
- DDI Instance
- Study Unit

- **Software Architecture**
  - Multitier
  - MVC
- Missy Data Model
  - Extendable Data Model

- Conceptual Component
- Logical Products
- Data Collection
- disco-model

Business Layer

What makes Missy interesting

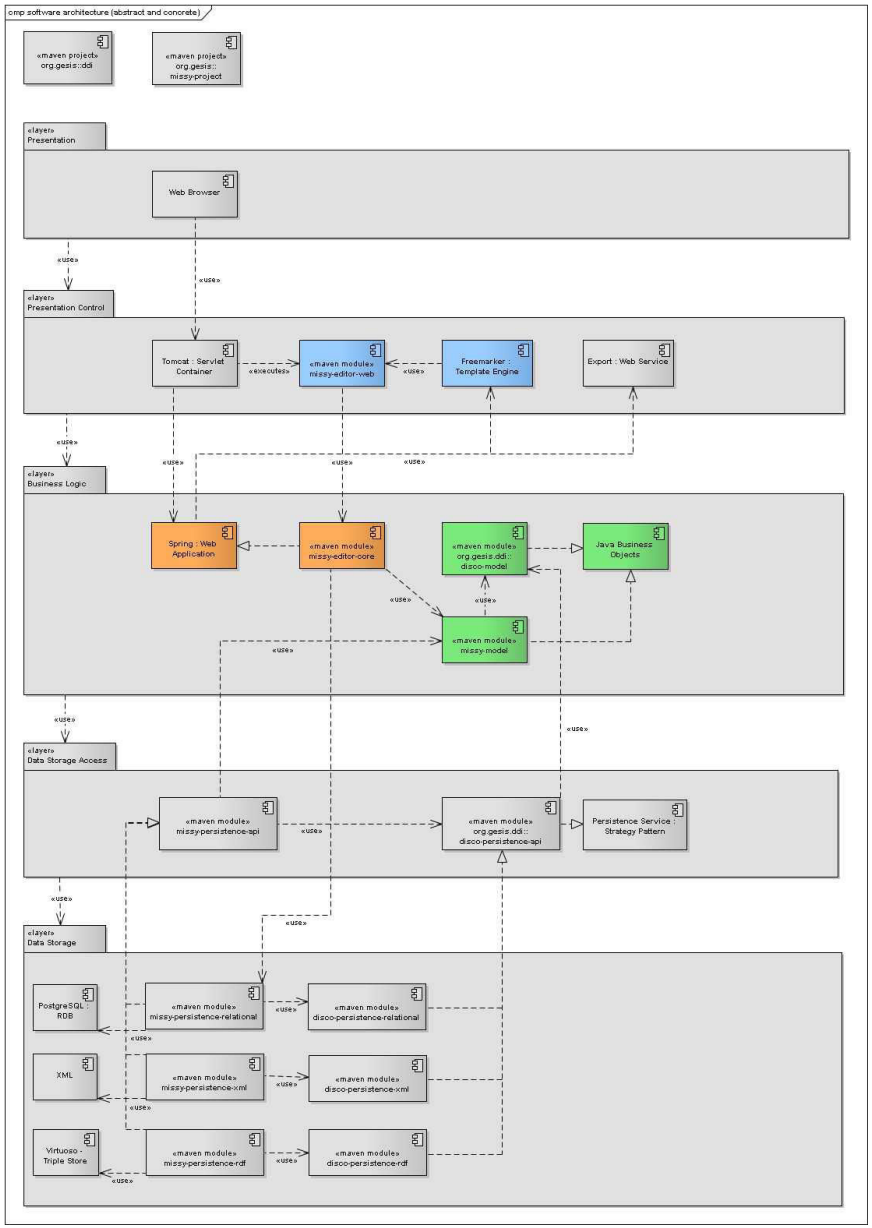
- Cross-linking between different multilingual studies
  - enabled by the common data model
  - provides different use-cases
- Missy leverages DDI as its back-end data model and exchange format
- Modern web project architecture, e.g. Multitier, MVC, etc.

- Is designed to be published as open-source project with an API to persist DDI data

Software Architecture

- Standard technologies to develop software
  - a multitier architecture
  - Model-View-Controller (MVC-Pattern)
  - Project management software, e.g. Maven

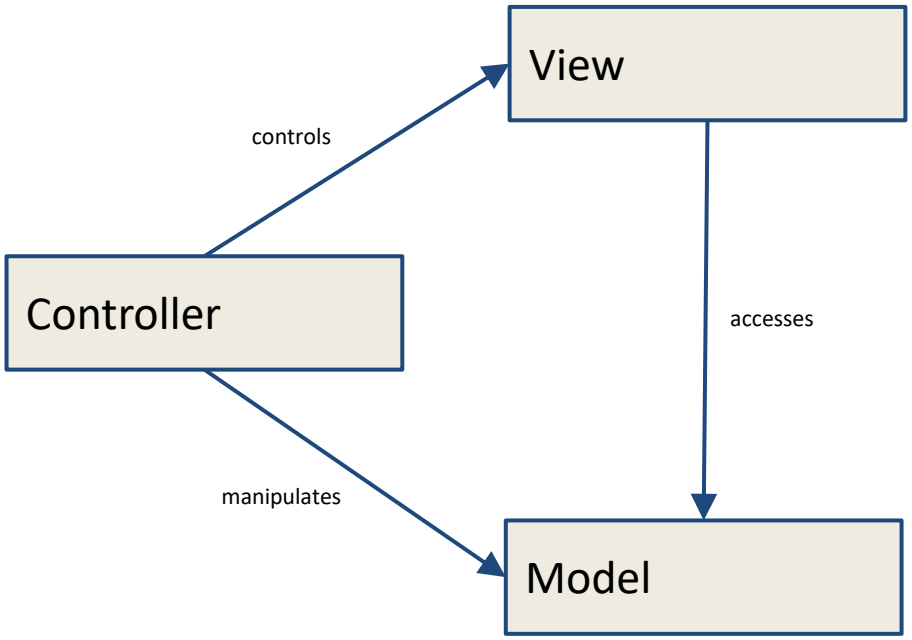
- Multitier architecture separates the project into logical parts
  - presentation
  - business logic or application processing
  - data management
  - persistence
  - …


cmp software architecture (abstract and concrete)
«maven project» org.gesis::ddi
«maven project» org.gesis:: missy-project
«layer» Presentation
Web Browser
«use»
«layer» Presentation Control
Tomcat : Servlet Container
«executes»
«maven module» missy-editor-web
Freemarker : Template Engine
Export : Web Service
«layer» Business Logic
Spring : Web Application
«maven module» missy-editor-core
«maven module» org.gesis.ddi:: disco-model
Java Business Objects
«maven module» missy-model
«layer» Data Storage Access
«maven module» missy-persistence-api
«maven module» org.gesis.ddi:: disco-persistence-api
Persistence Service : Strategy Pattern
«layer» Data Storage
PostgreSQL : RDB
«maven module» missy-persistence-relational
«maven module» disco-persistence-relational
XML
«maven module» missy-persistence-xml
«maven module» disco-persistence-xml
Virtuoso - Triple Store
«maven module» missy-persistence-rdf
«maven module» disco-persistence-rdf

Model-View-Controller

- separation of
  - representation of information
  - interactions with the data
  - components
- the key is to have logically separated parts, where people might work collaboratively

MVC – Interactions



Model

- Represents the business objects, i.e. real life concepts, and their relations to each other
- Sometimes also includes some business logic

- Independent of the presentation and the controls

View

- Responsible for the presentation of information to the user
- Provides actions, so that the user can interact with the application

- "Knows" the model and can access it

- Usually does not display the whole model, but rather special "views" or use-cases of it

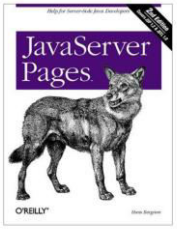
Controller

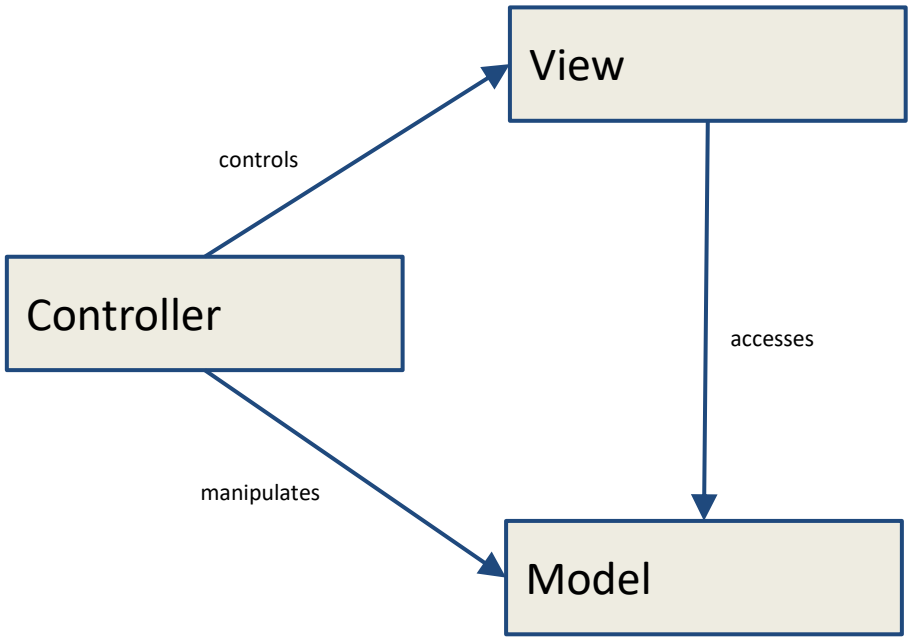
- Controls the presentations, i.e. views
- Receives actions from users
  - interprets/evaluates them
  - acts accordingly

- Manipulates the data / model
- Usually each view has its own controller

- Be warned of code reuse of controllers
  - controllers are most often the throwaway components

View Technologies

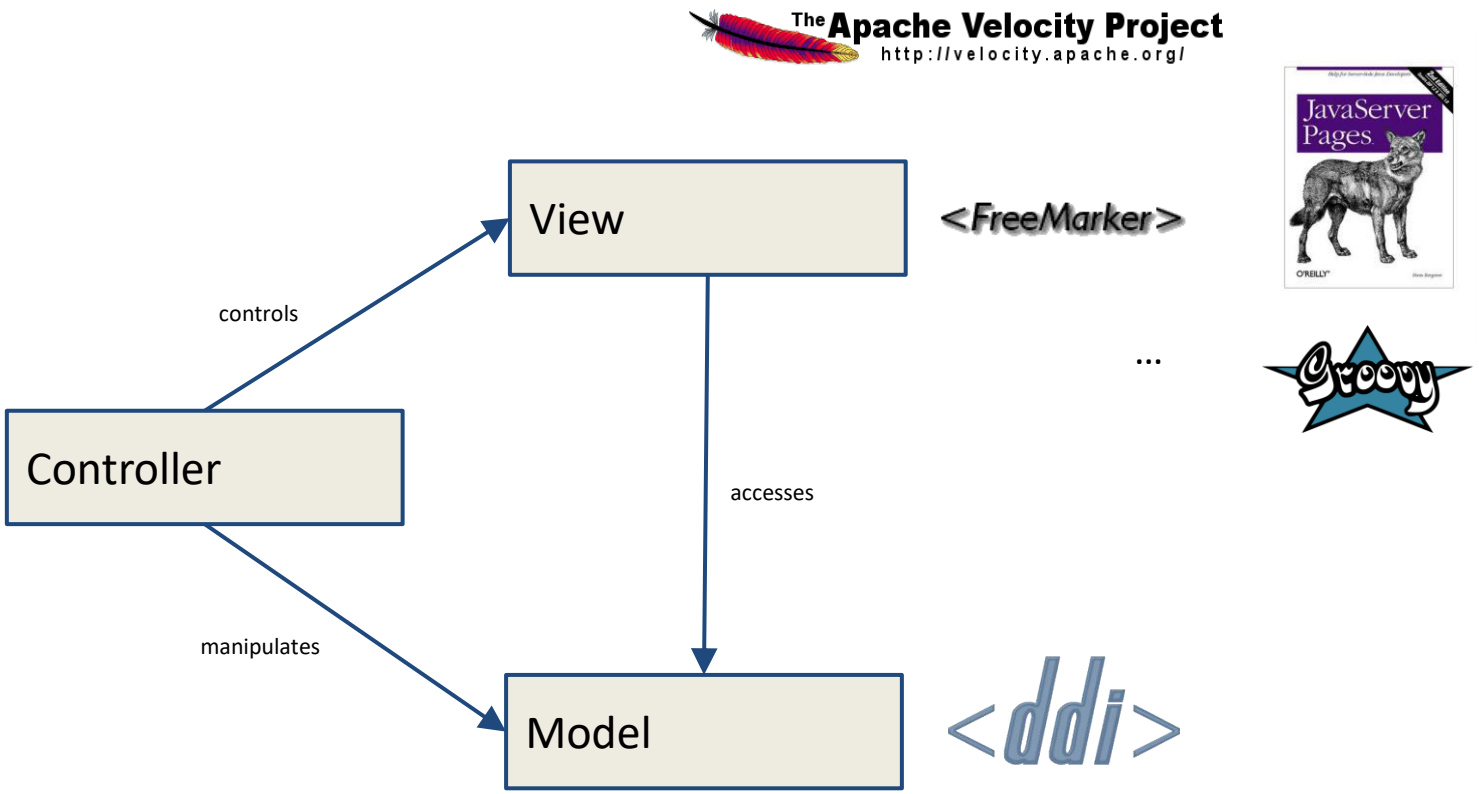




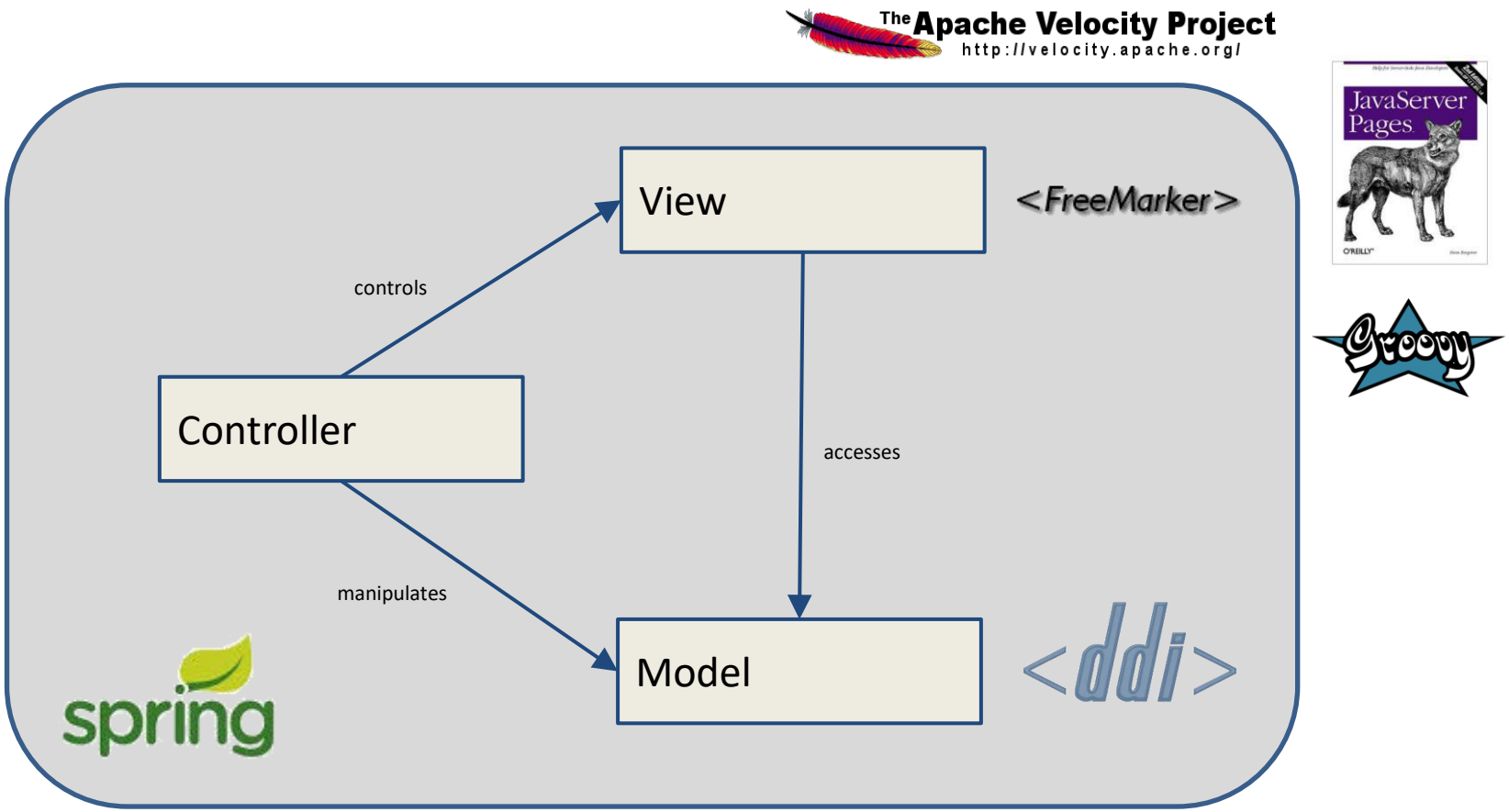
<FreeMarker>

...

Data Model


The Apache Velocity Project
http://velocity.apache.org/
JavaServer Pages
View
<FreeMarker>
controls
...
Groovy
Controller
accesses
manipulates
Model
<ddi>

Missy Technologies



What we have discussed so far

- Brief introduction to Missy and the use-cases
- With which technologies a web application framework might be build up today
- How a modern software architecture looks like

How the presentation is continued

- What the DDI-L core modules are and how they are organized

- Introduction the disco model and
- How to extend it

What comes next?

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy

Presentation

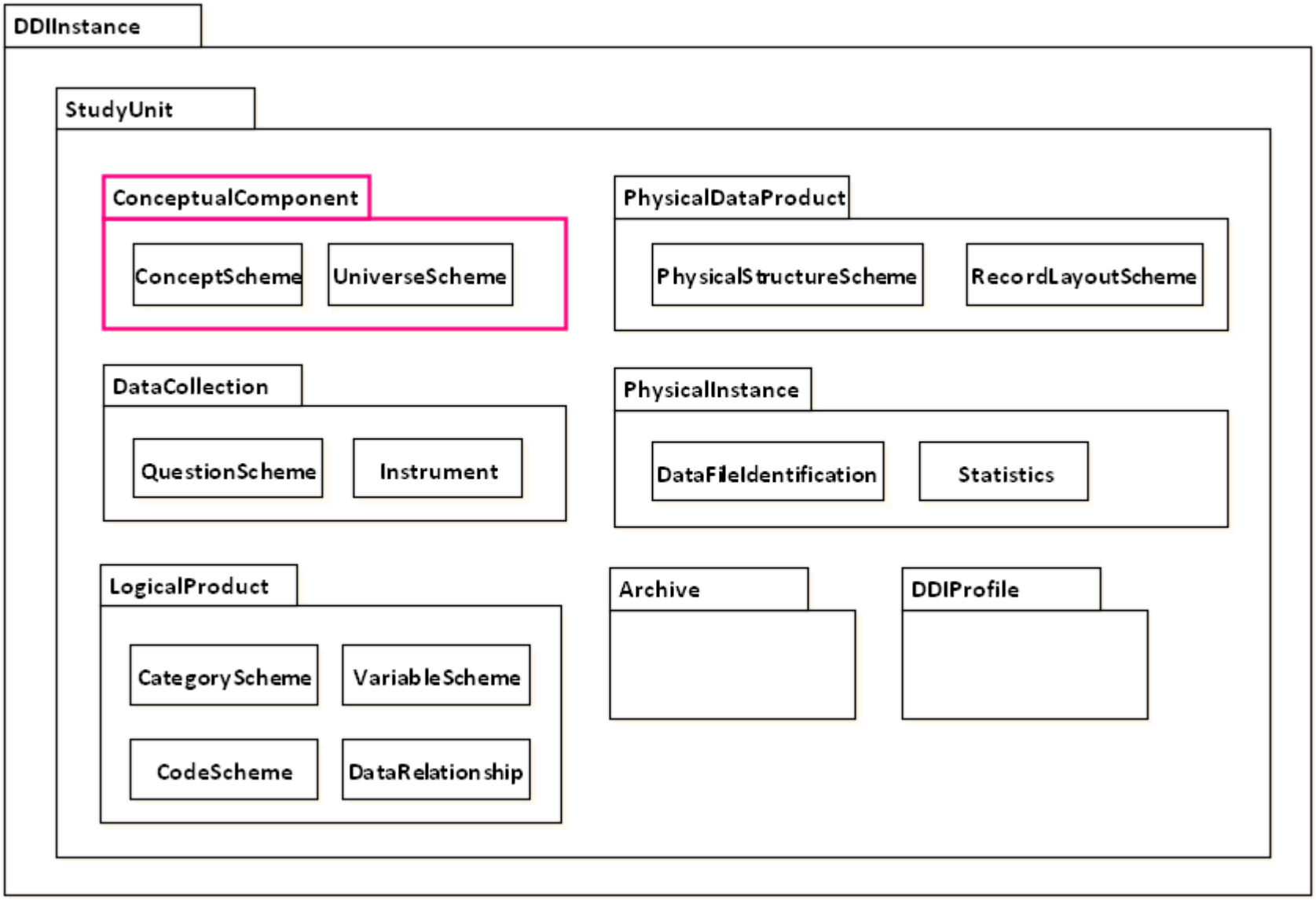
- DDI-L Overview
- Identification, Versioning, Maintenance
- DDI-L Main Structures
- DDI Instance
- Study Unit

- Software Architecture
  - Multitier
  - MVC
- Missy Data Model
  - Extendable Data Model

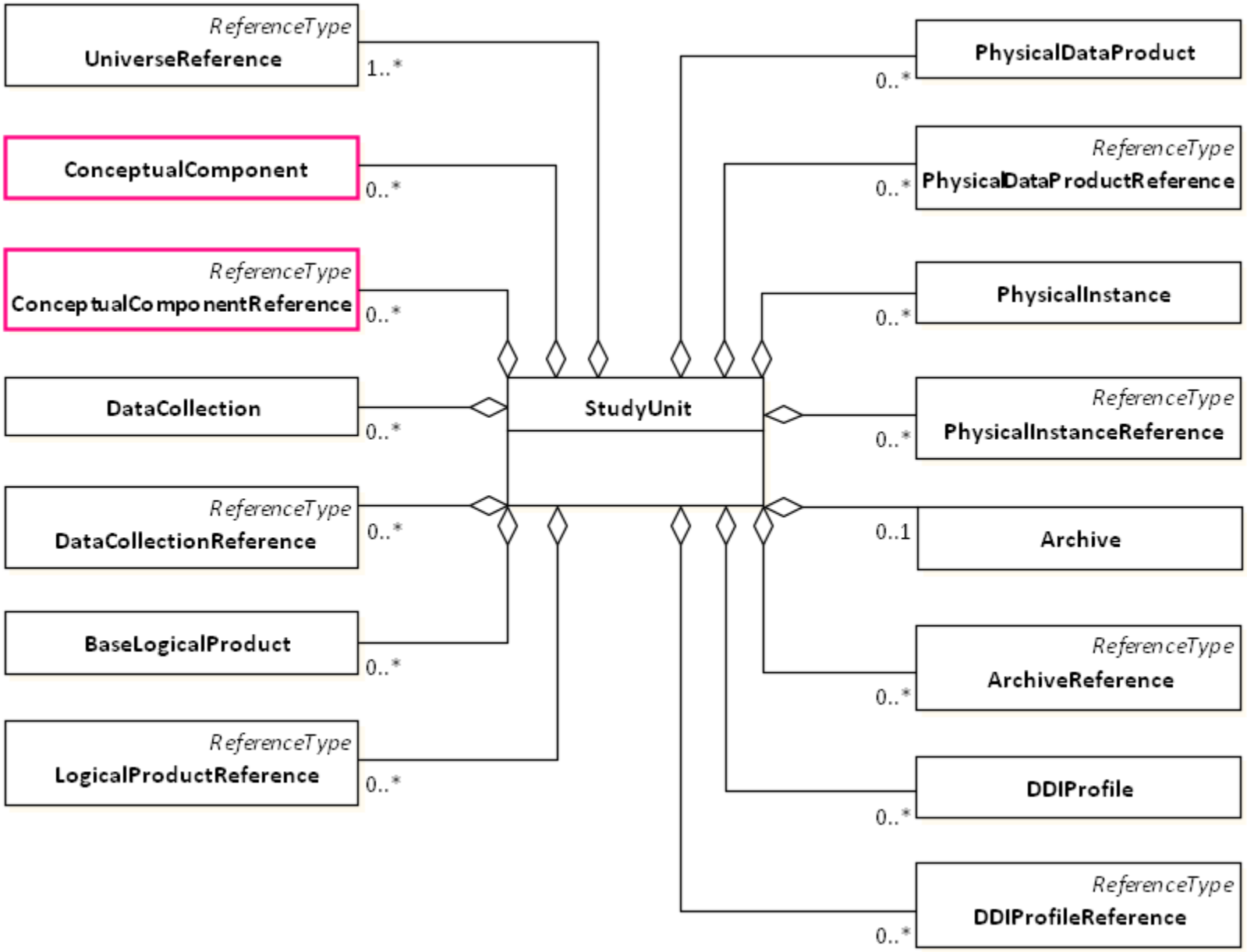
- **Conceptual Component**
- Logical Products
- Data Collection
- disco-model

Business Layer

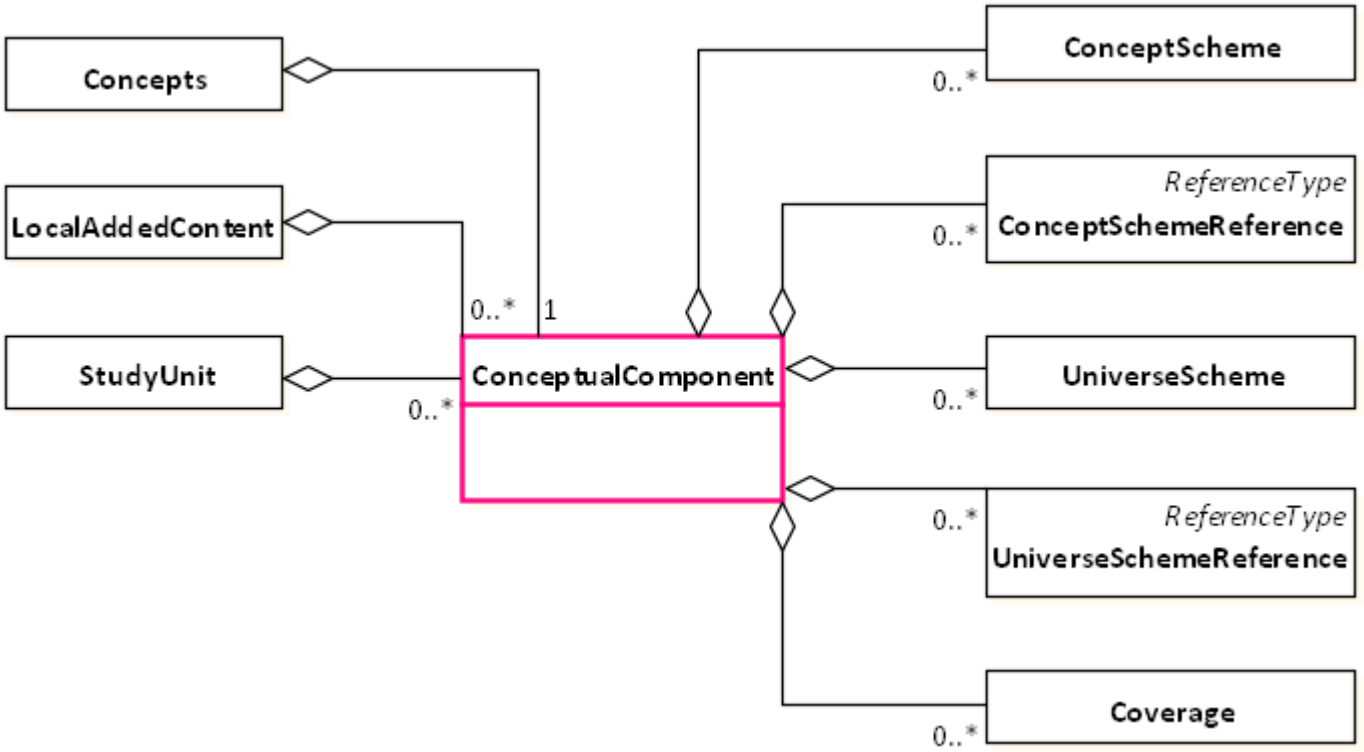
ConceptualComponent



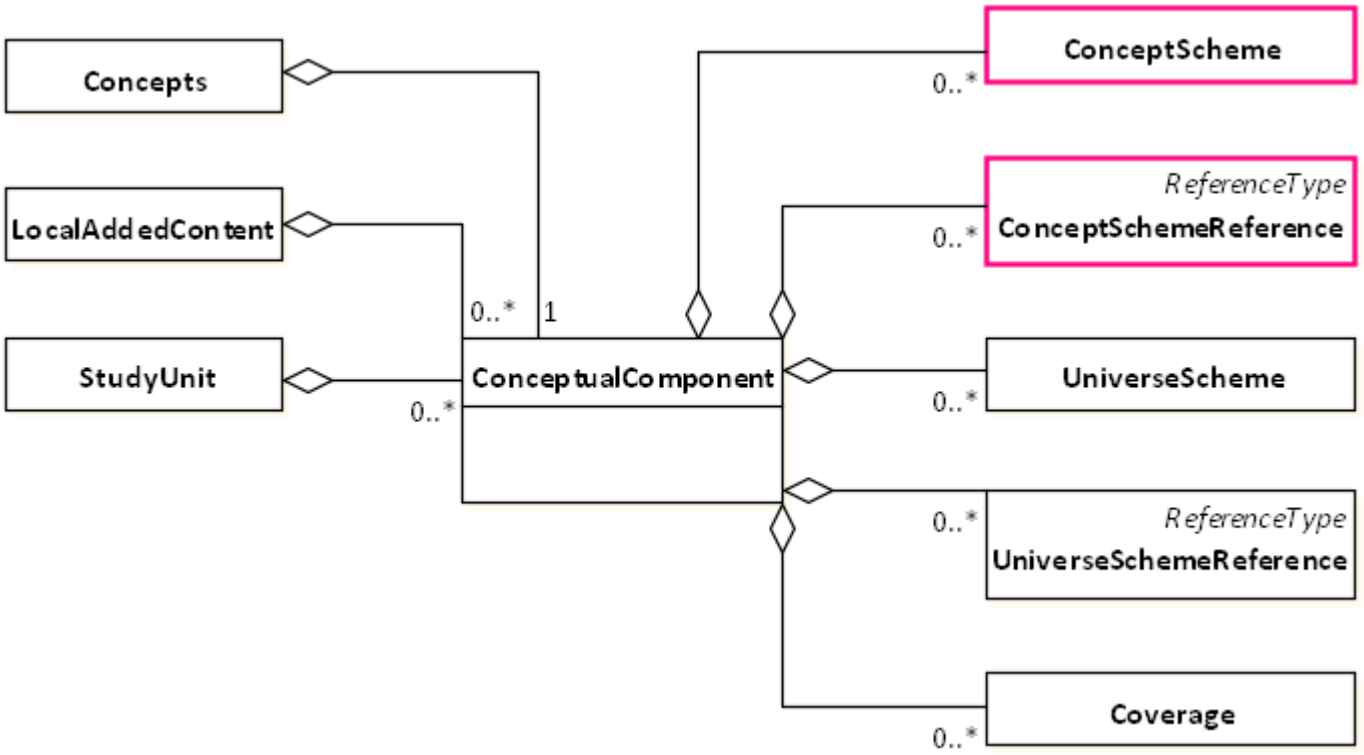
ConceptualComponent



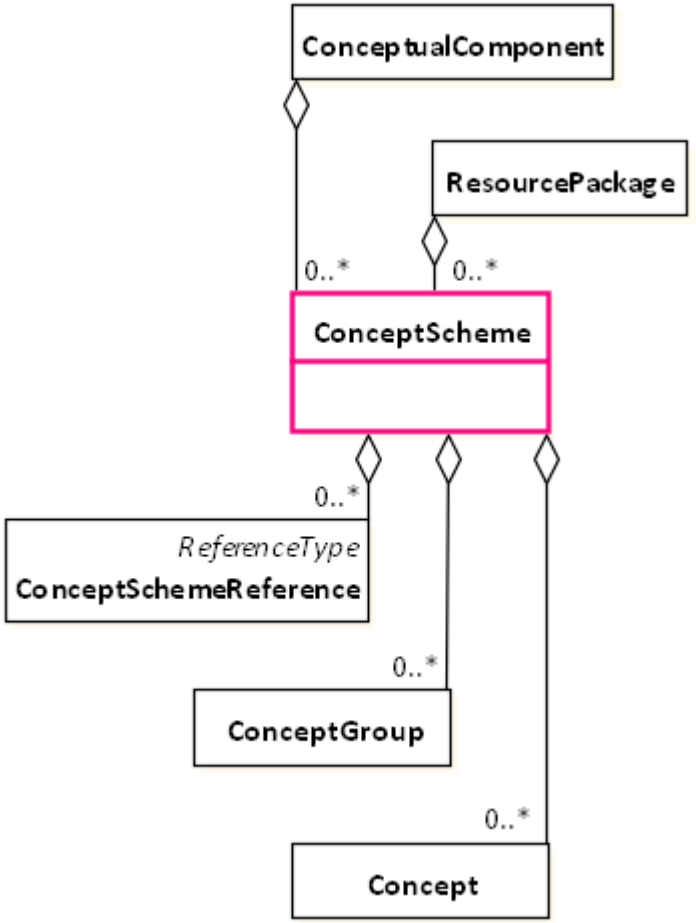
ConceptualComponent



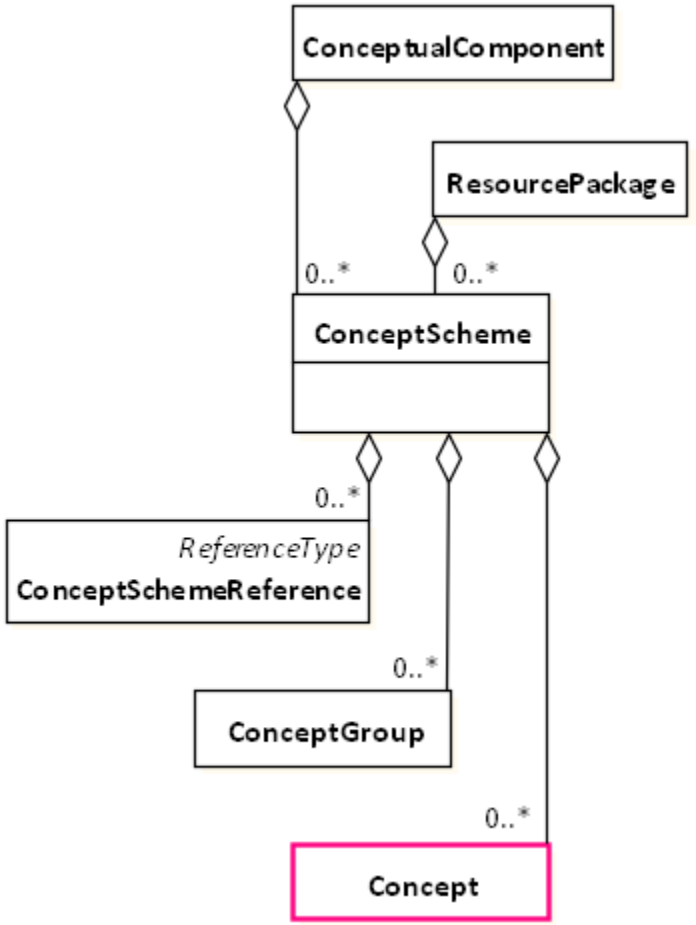
ConceptScheme



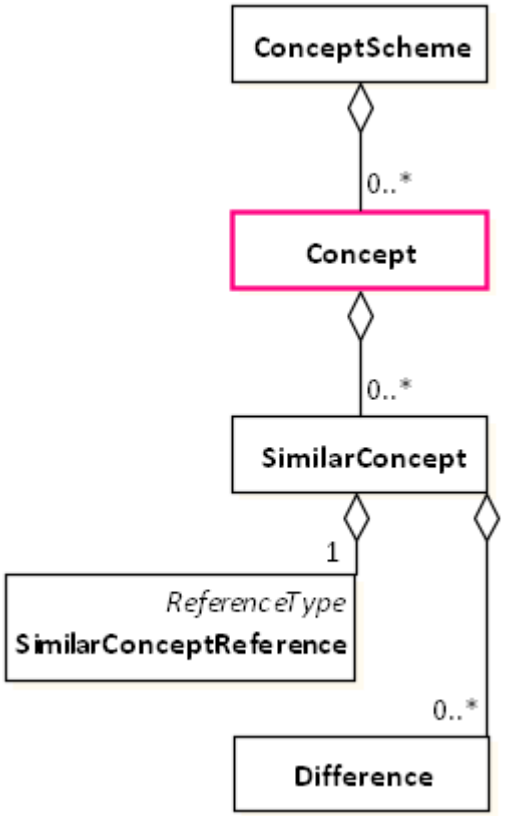
ConceptScheme



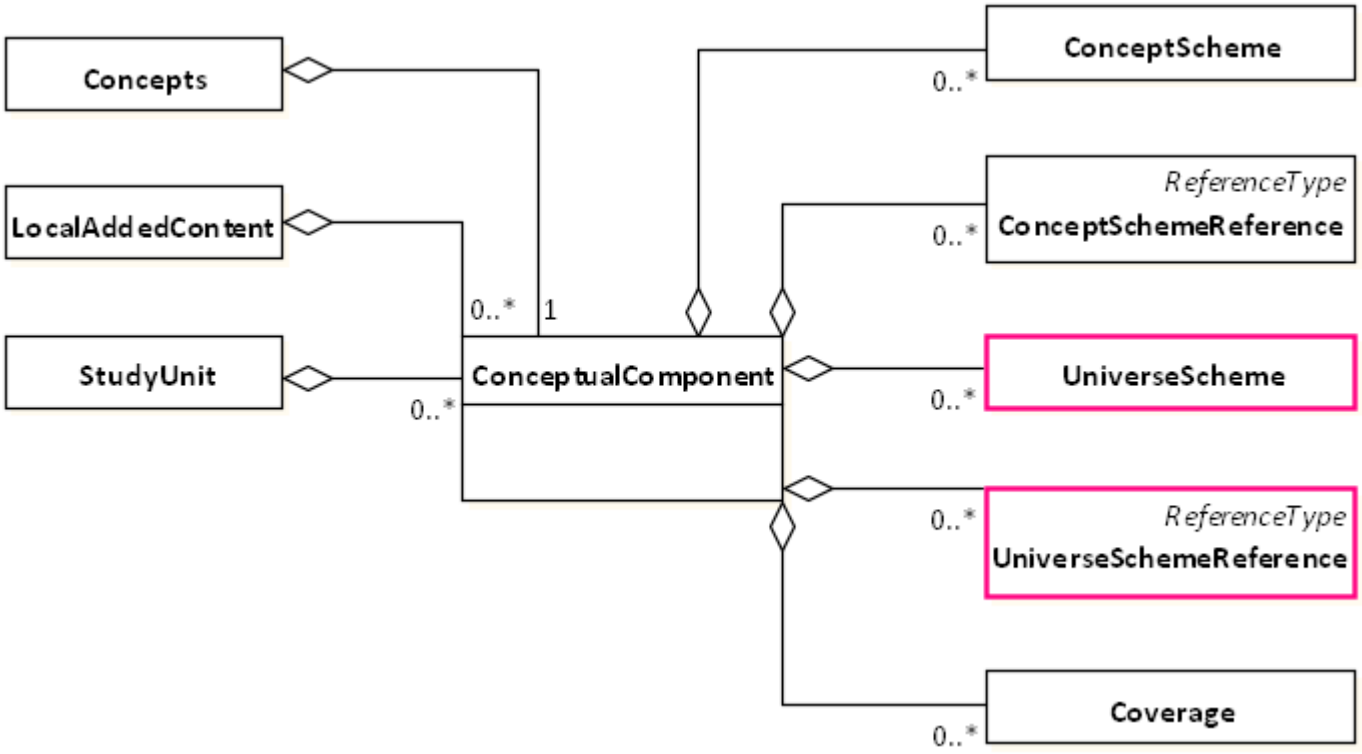
Concept


ConceptualComponent
ResourcePackage
0..*
0..*
ConceptScheme
0..*
ReferenceType
ConceptSchemeReference
0..*
ConceptGroup
0..*
Concept

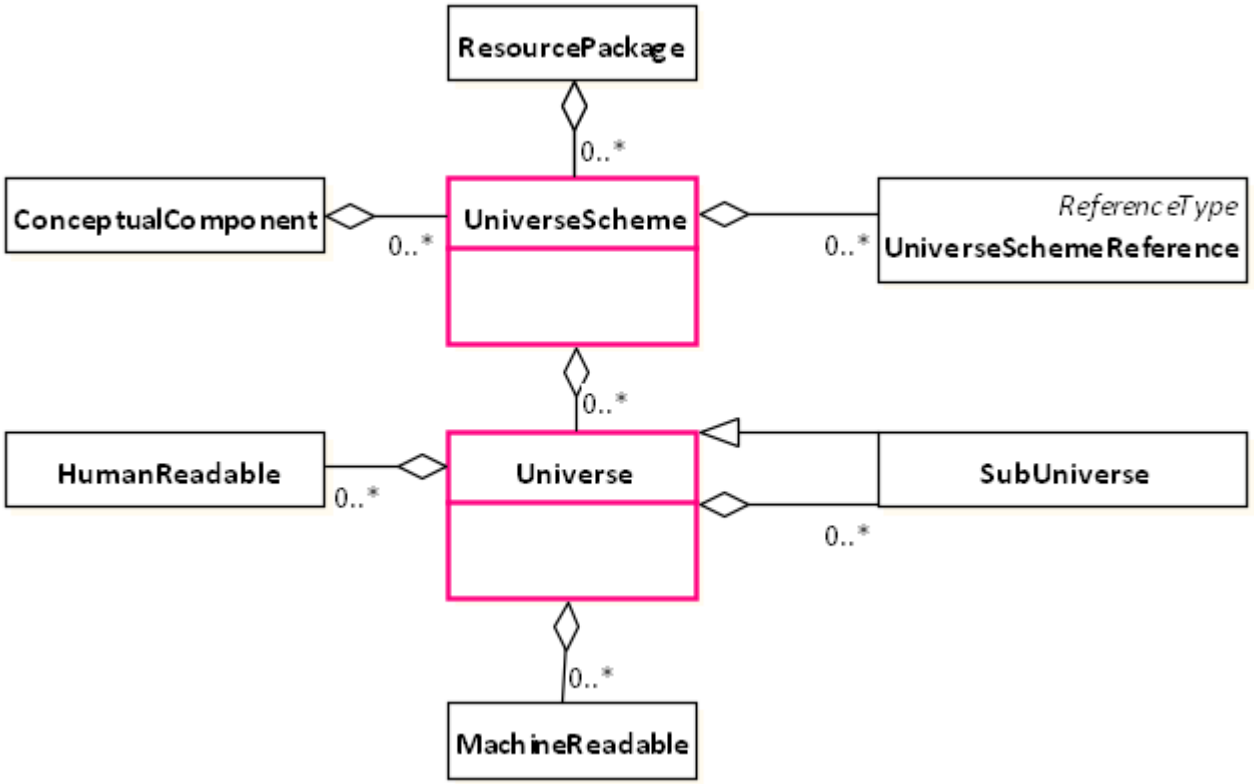
Concept


ConceptScheme
0..*
Concept
0..*
SimilarConcept
1
ReferenceType
SimilarConceptReference
0..*
Difference

Universe


Concepts
LocalAddedContent
StudyUnit
ConceptualComponent
0..*
1
0..*
0..*
ConceptScheme
0..*
ReferenceType
ConceptSchemeReference
0..*
UniverseScheme
0..*
ReferenceType
UniverseSchemeReference
0..*
Coverage
0..*

Universe


ResourcePackage
0..*
ConceptualComponent
0..*
UniverseScheme
0..*
ReferenceType
UniverseSchemeReference
0..*
HumanReadable
0..*
Universe
0..*
SubUniverse
0..*
MachineReadable

What comes next?

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy

Presentation

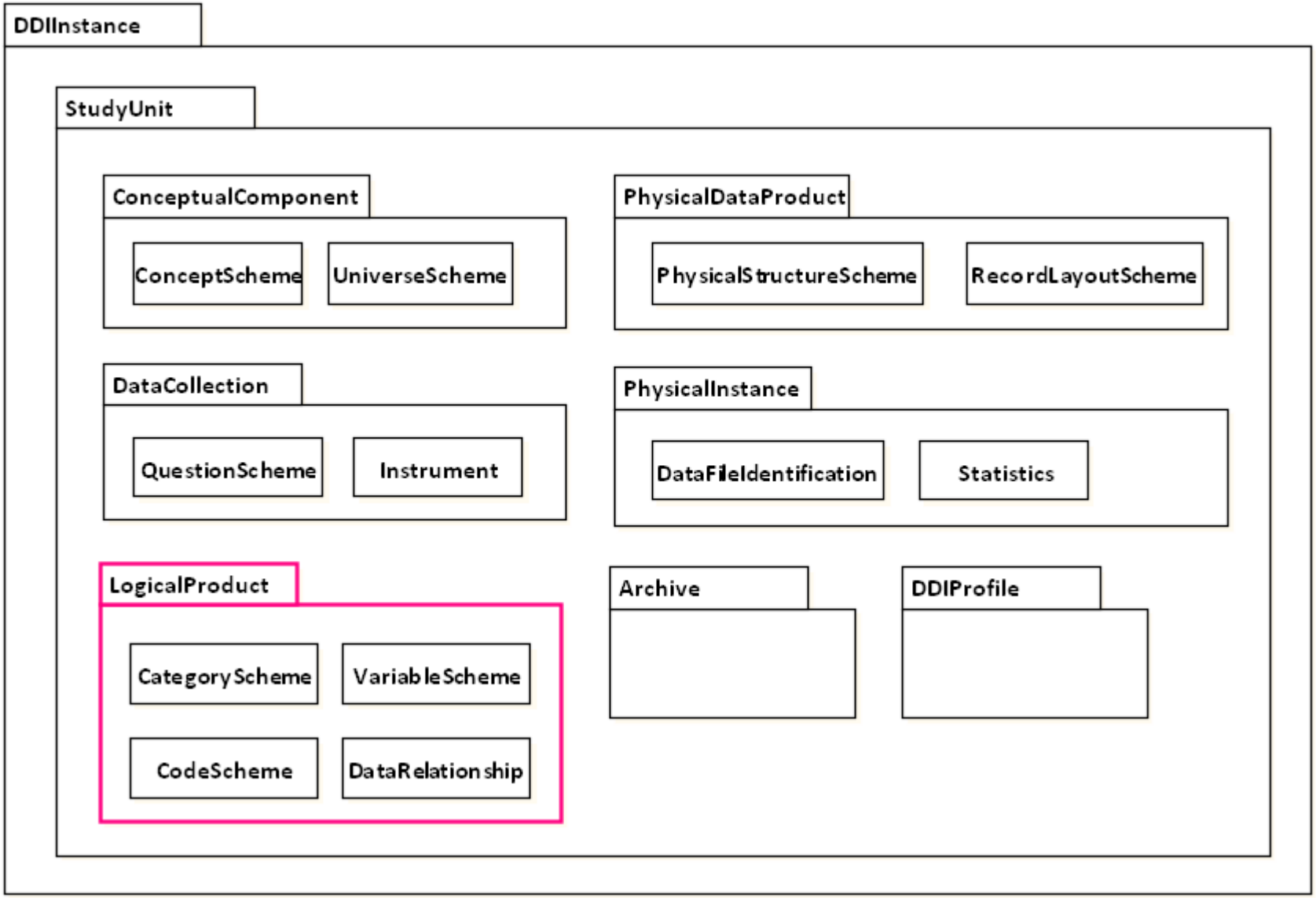
- DDI-L Overview
- Identification, Versioning, Maintenance
- DDI-L Main Structures
- DDI Instance
- Study Unit

- Software Architecture
  - Multitier
  - MVC
- Missy Data Model
  - Extendable Data Model

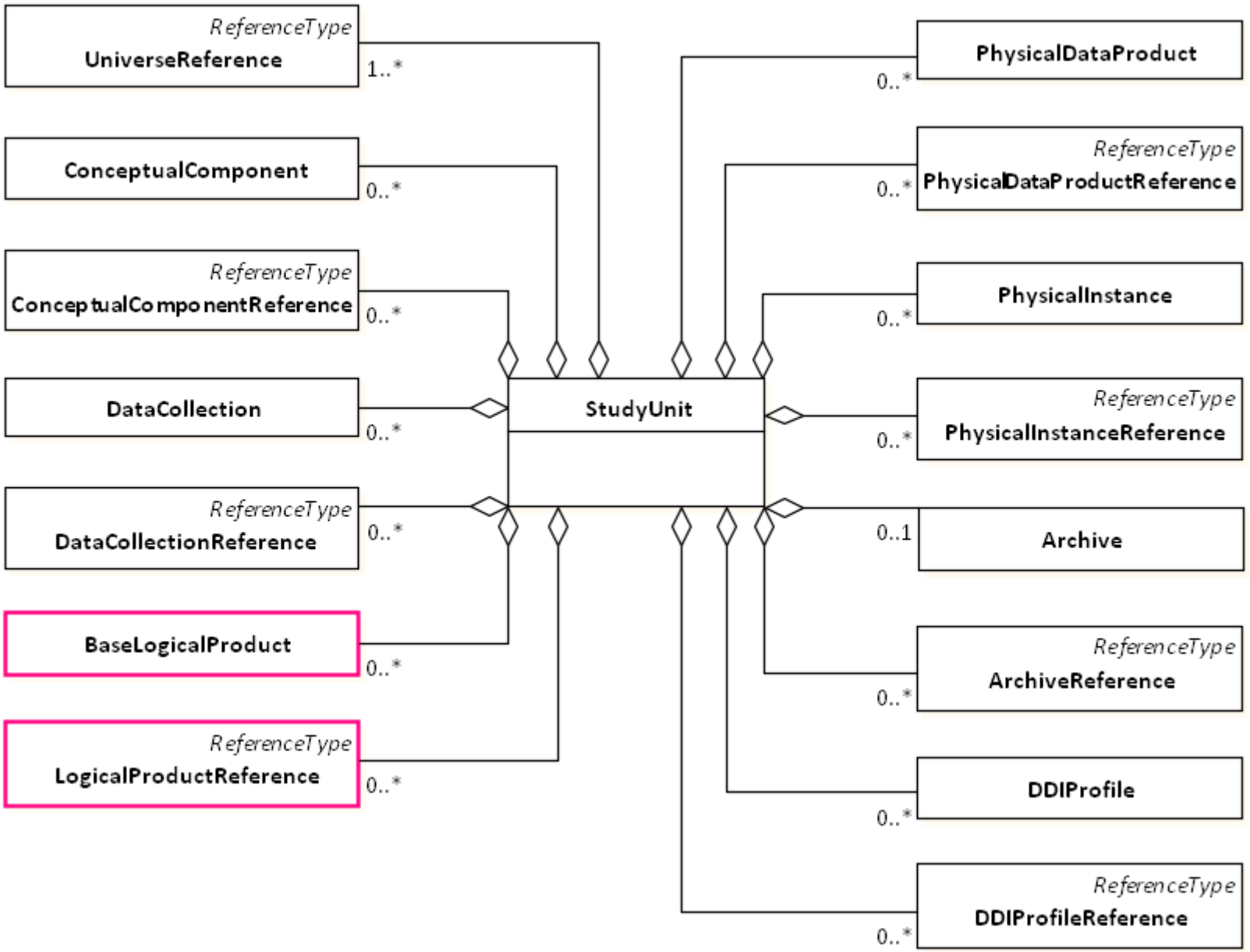
- Conceptual Component
- **Logical Products**
- Data Collection
- disco-model

Business Layer

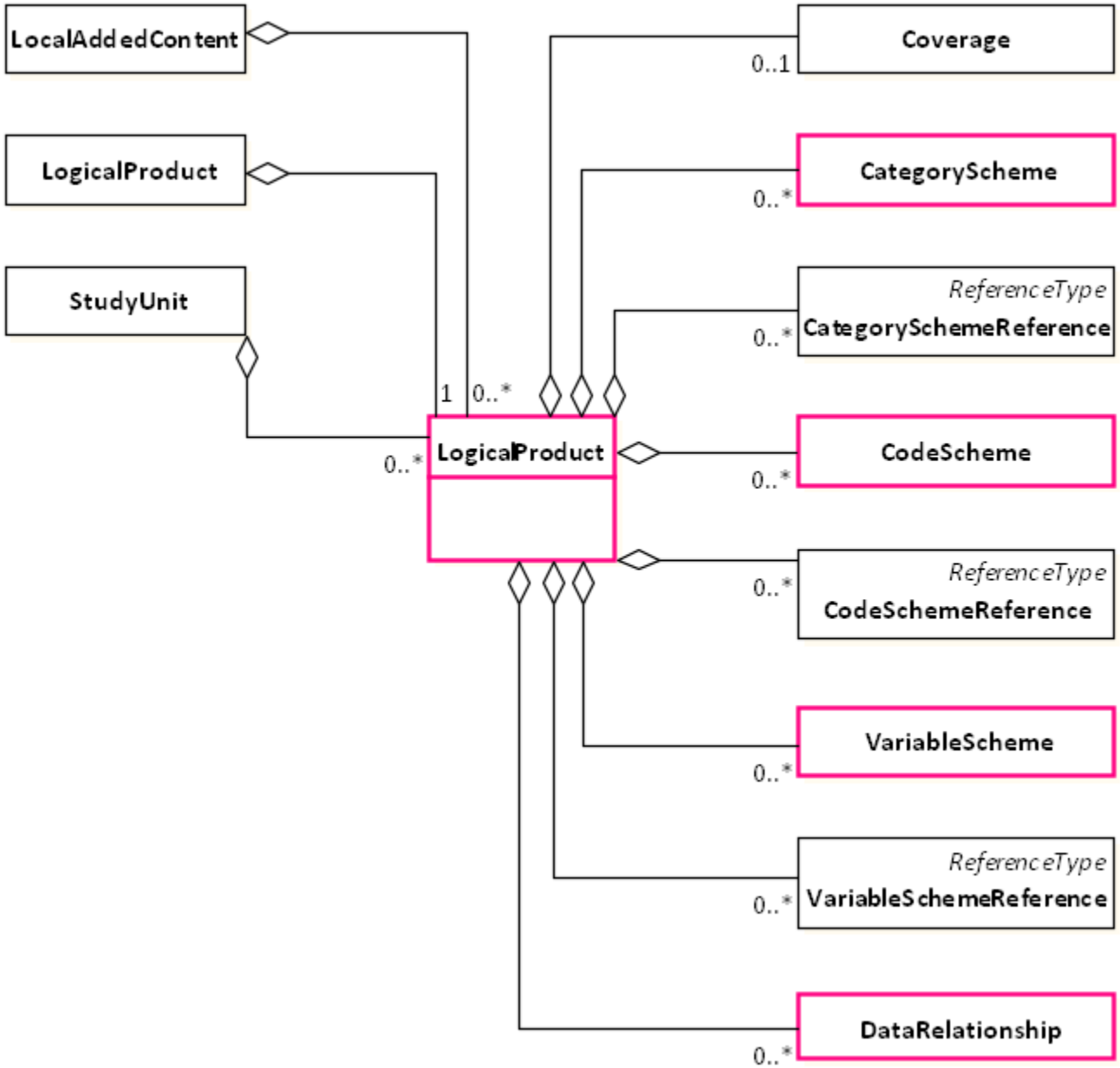
LogicalProduct



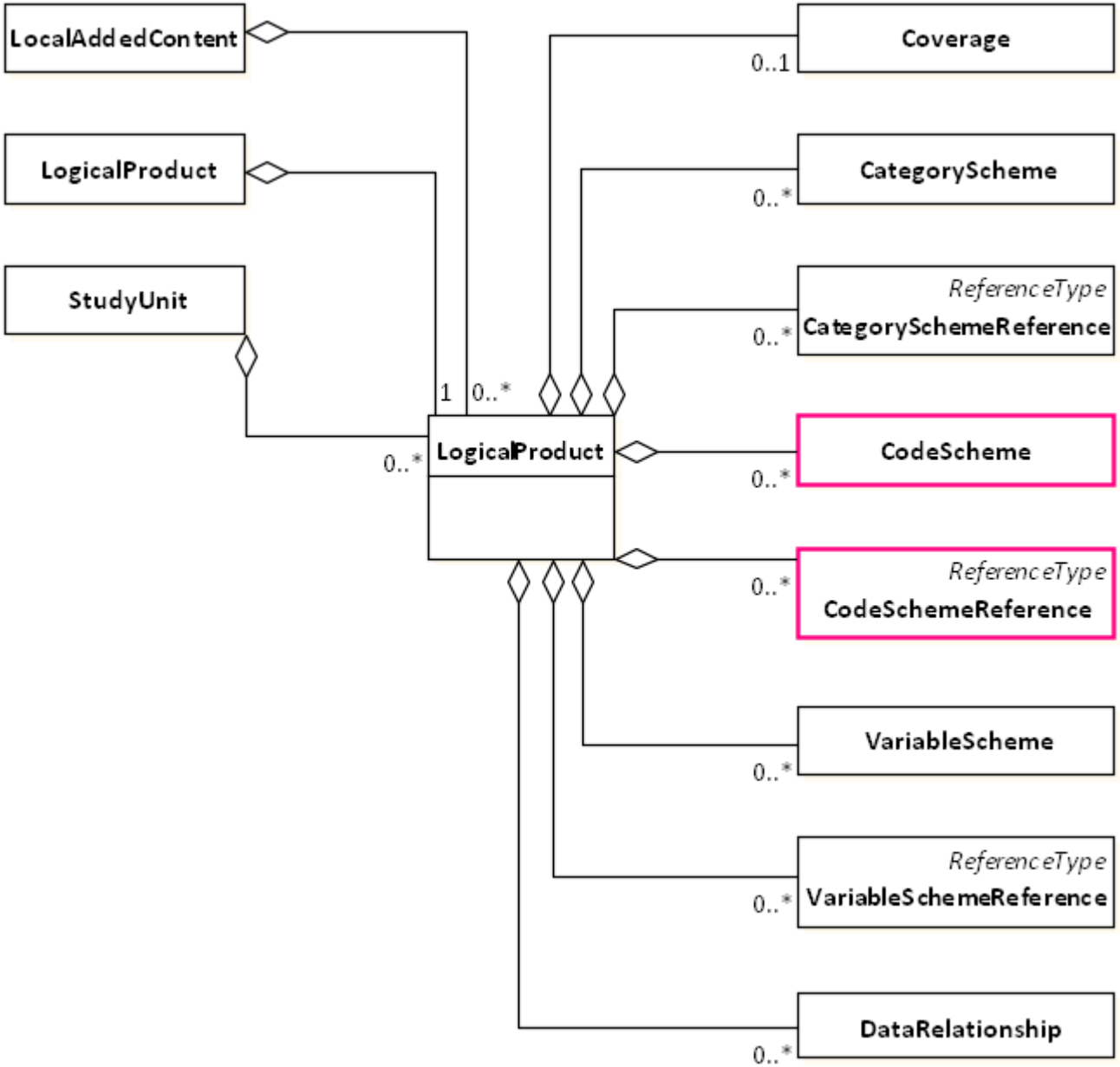
LogicalProduct



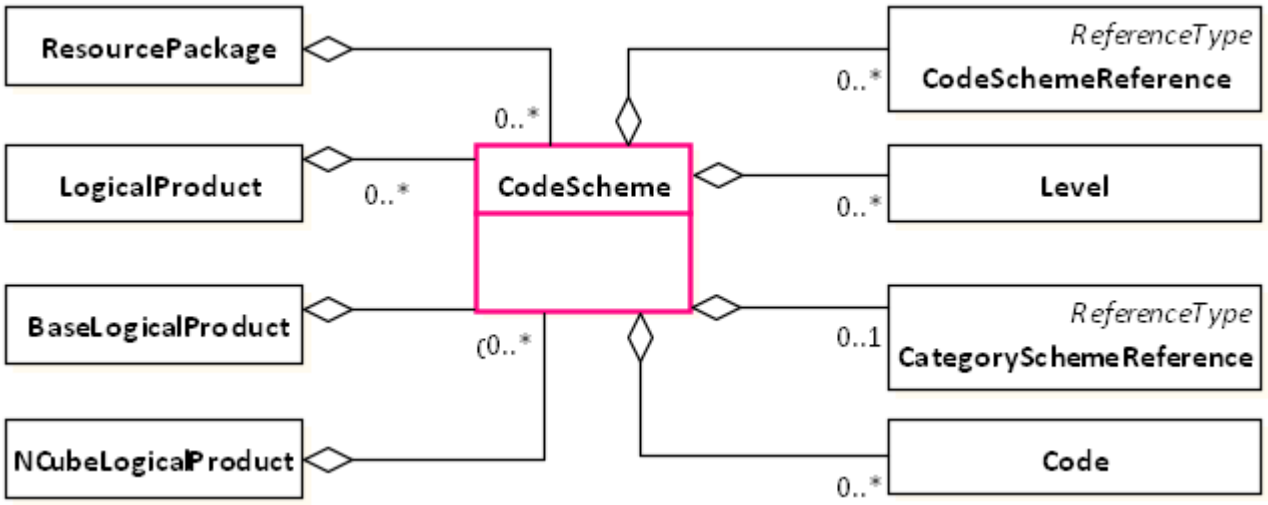
LogicalProduct



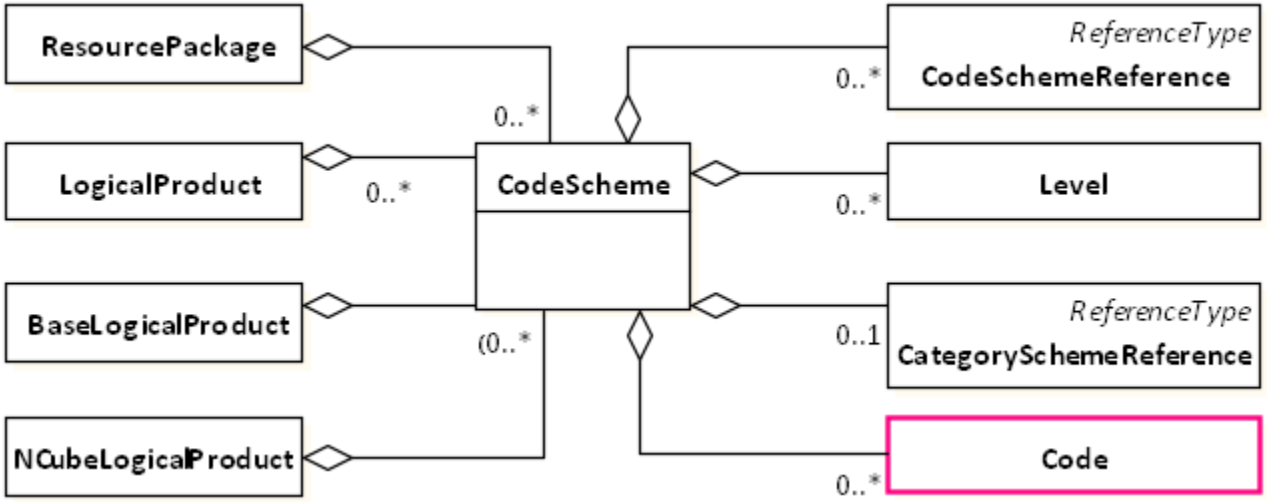
CodeScheme


LocalAddedContent
LogicalProduct
StudyUnit
1
0..*
0..*
LogicalProduct
Coverage
0..1
CategoryScheme
0..*
ReferenceType
CategorySchemeReference
0..*
CodeScheme
0..*
ReferenceType
CodeSchemeReference
0..*
VariableScheme
0..*
ReferenceType
VariableSchemeReference
0..*
DataRelationship
0..*

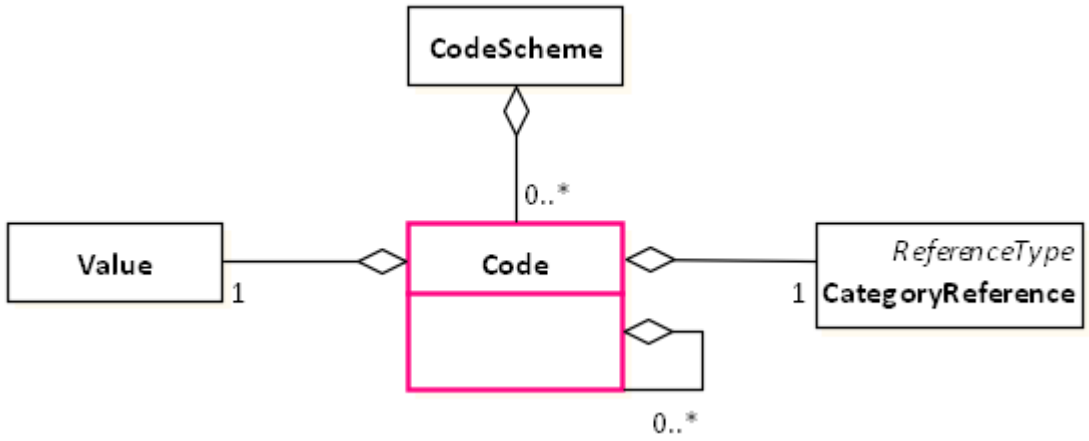
CodeScheme


ResourcePackage
LogicalProduct
BaseLogicalProduct
NCubeLogicalProduct
CodeScheme
0..*
0..*
0..*
0..*
ReferenceType
CodeSchemeReference
Level
ReferenceType
CategorySchemeReference
Code
0..*
0..*
0..1
0..*

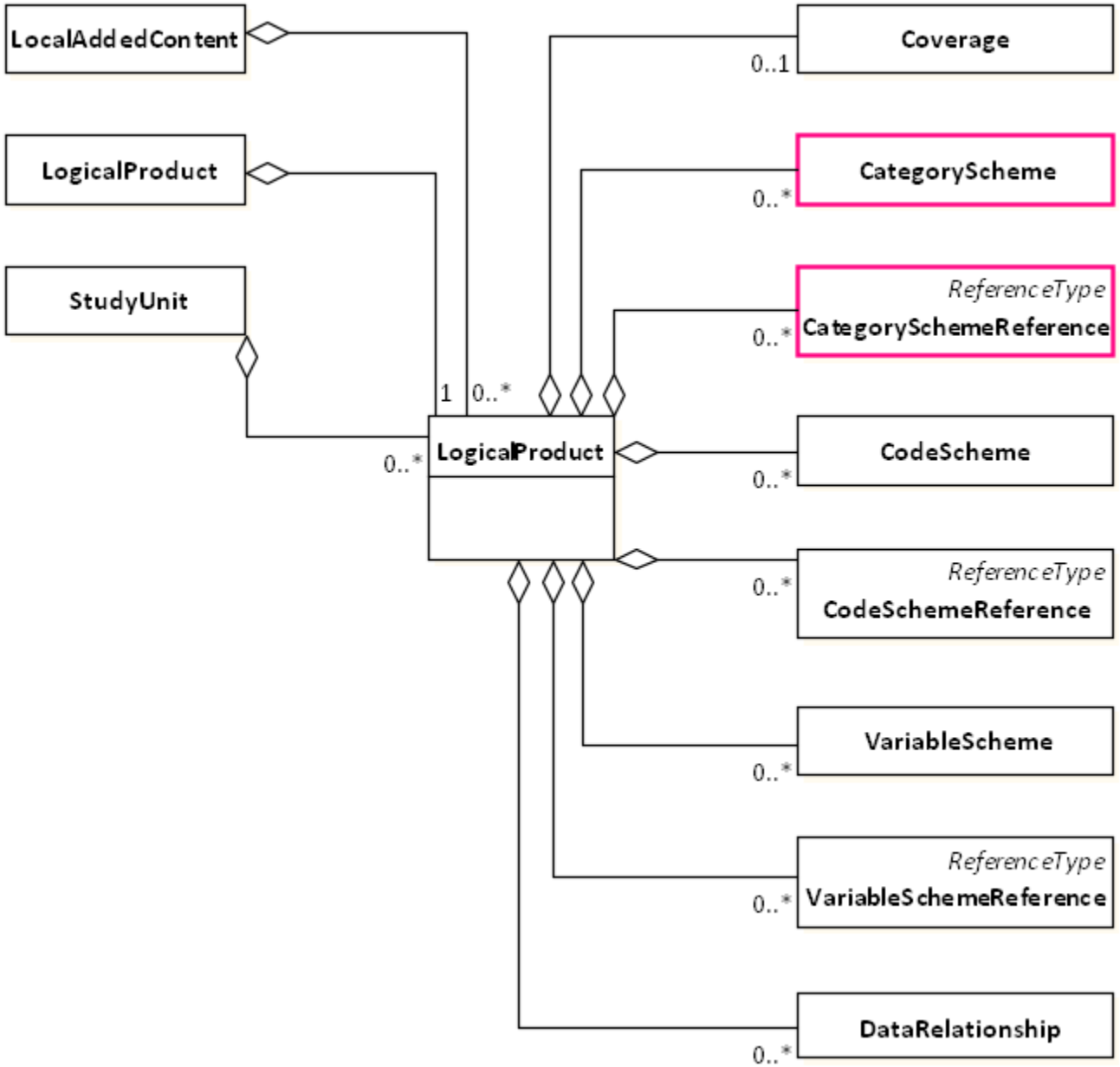
Code


ResourcePackage
LogicalProduct
BaseLogicalProduct
NCubeLogicalProduct
CodeScheme
0..*
0..*
(0..*
0..*
ReferenceType
CodeSchemeReference
0..*
Level
0..*
ReferenceType
CategorySchemeReference
0..1
Code
0..*

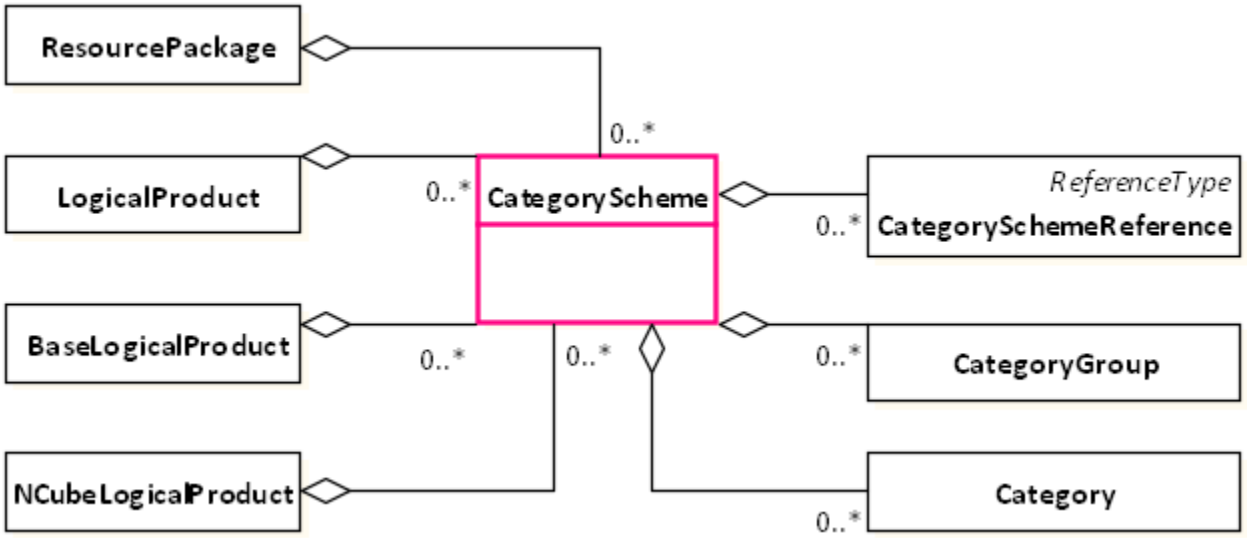
Code


CodeScheme
0..*
Value
1
Code
1
ReferenceType
CategoryReference
0..*

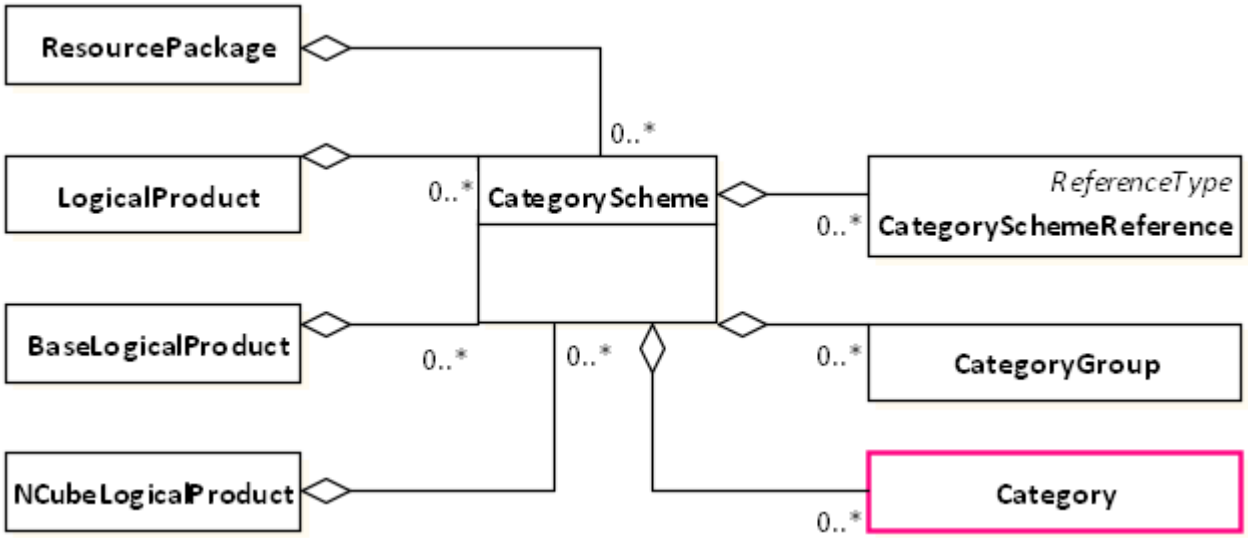
CategoryScheme



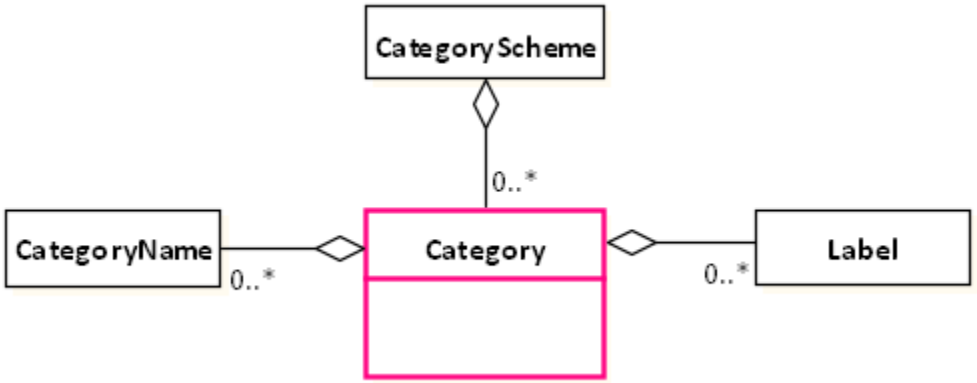
CategoryScheme



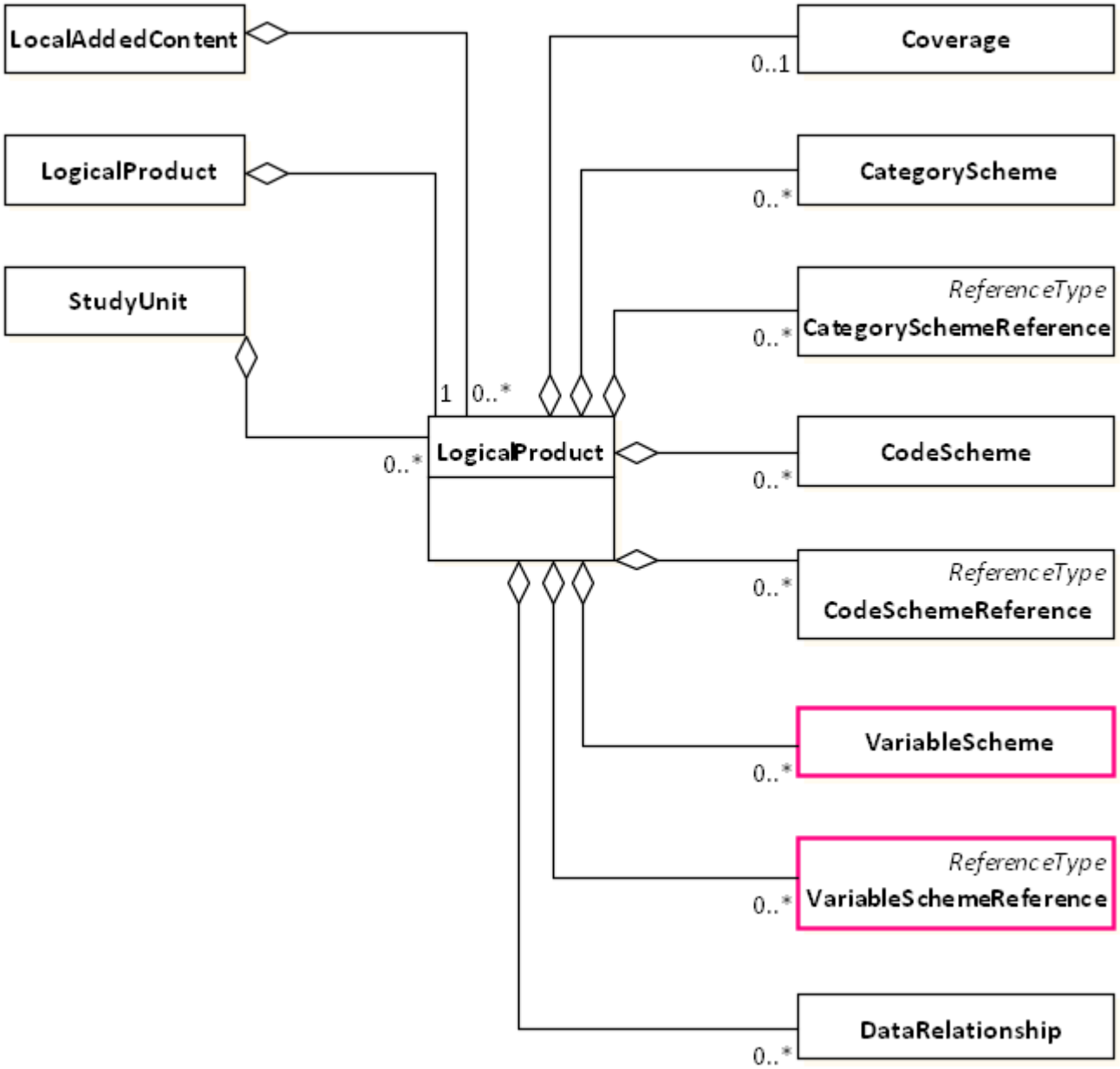
Category


ResourcePackage
LogicalProduct
BaseLogicalProduct
NCubeLogicalProduct
CategoryScheme
ReferenceType
CategorySchemeReference
CategoryGroup
Category
0..*
0..*
0..*
0..*
0..*
0..*
0..*
0..*

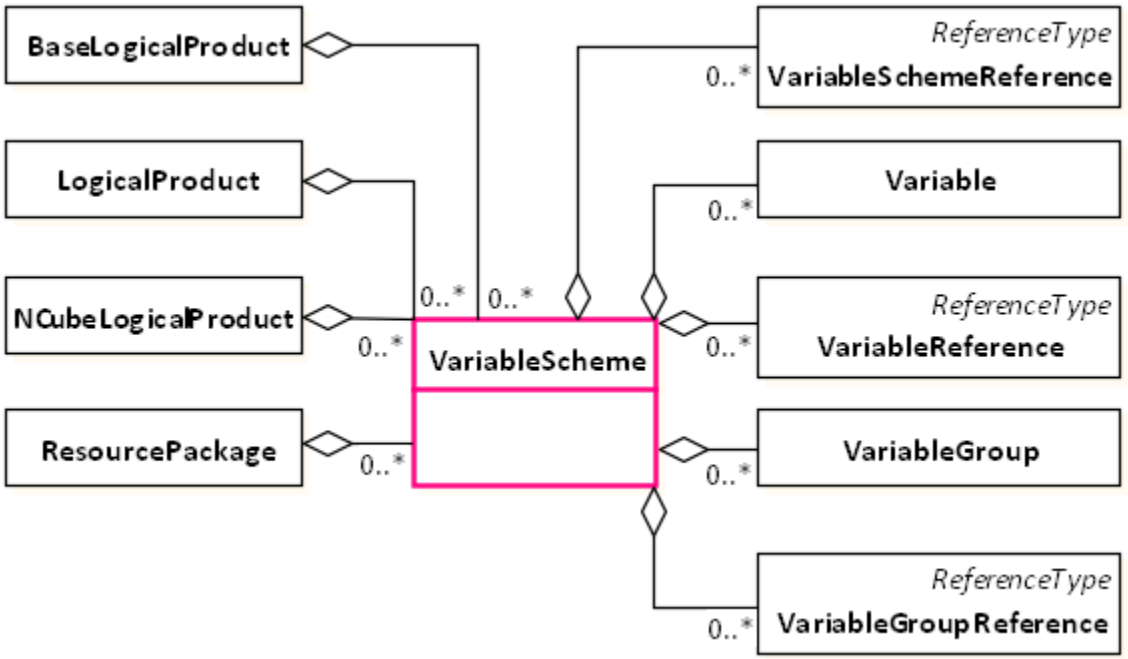
Category



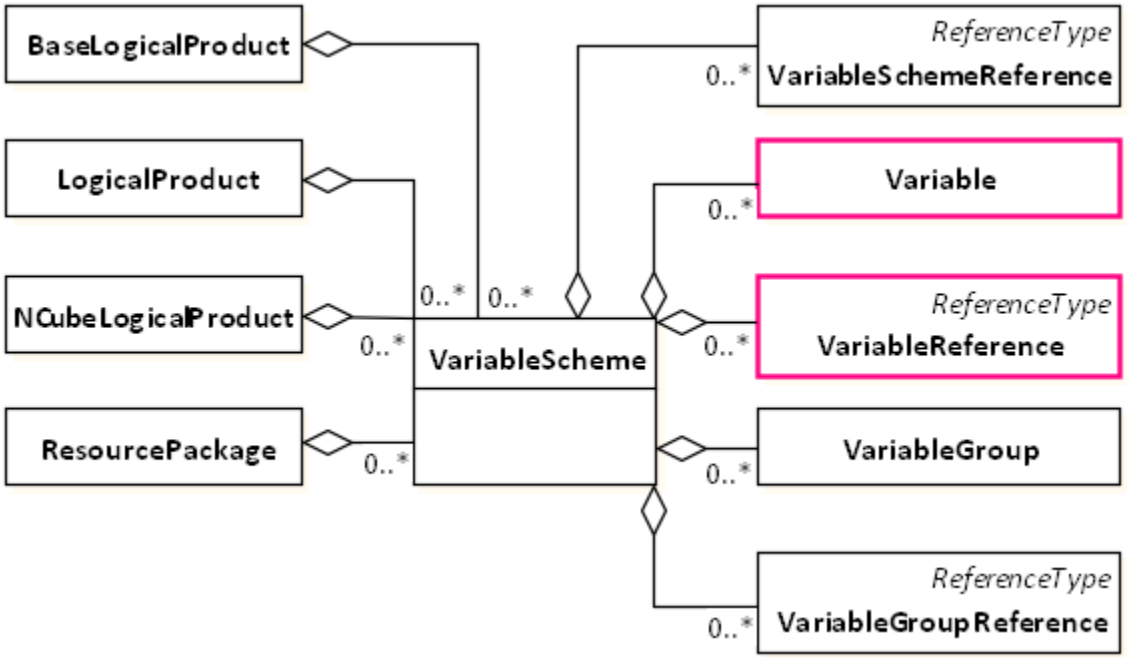
VariableScheme



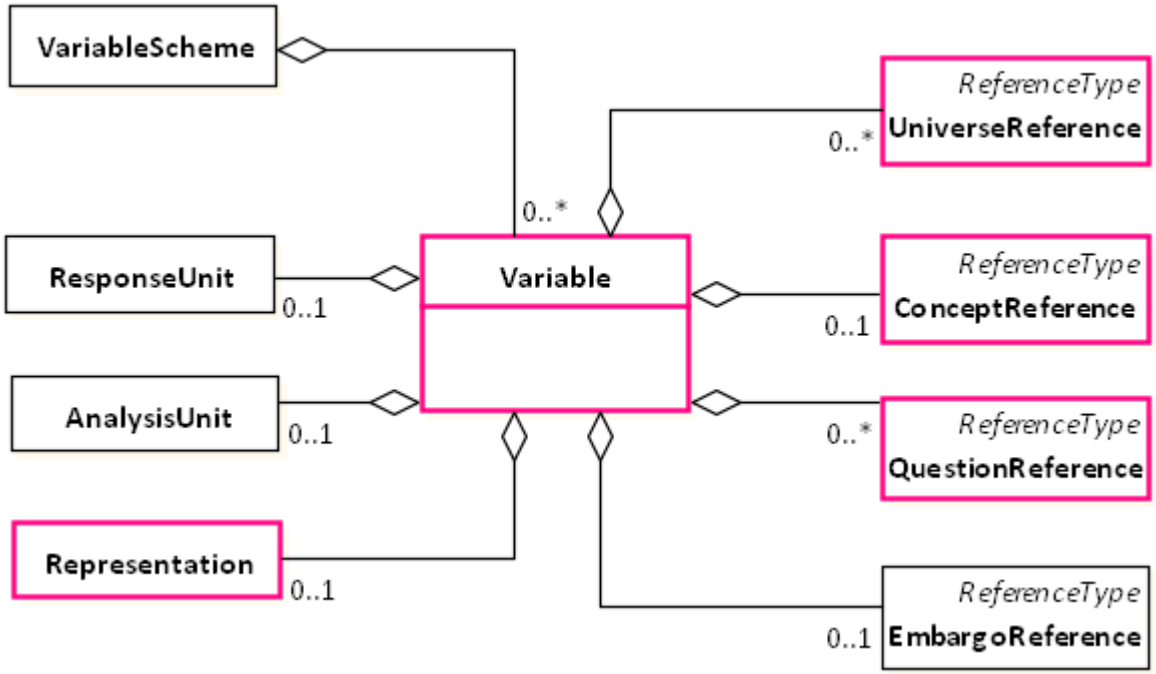
VariableScheme



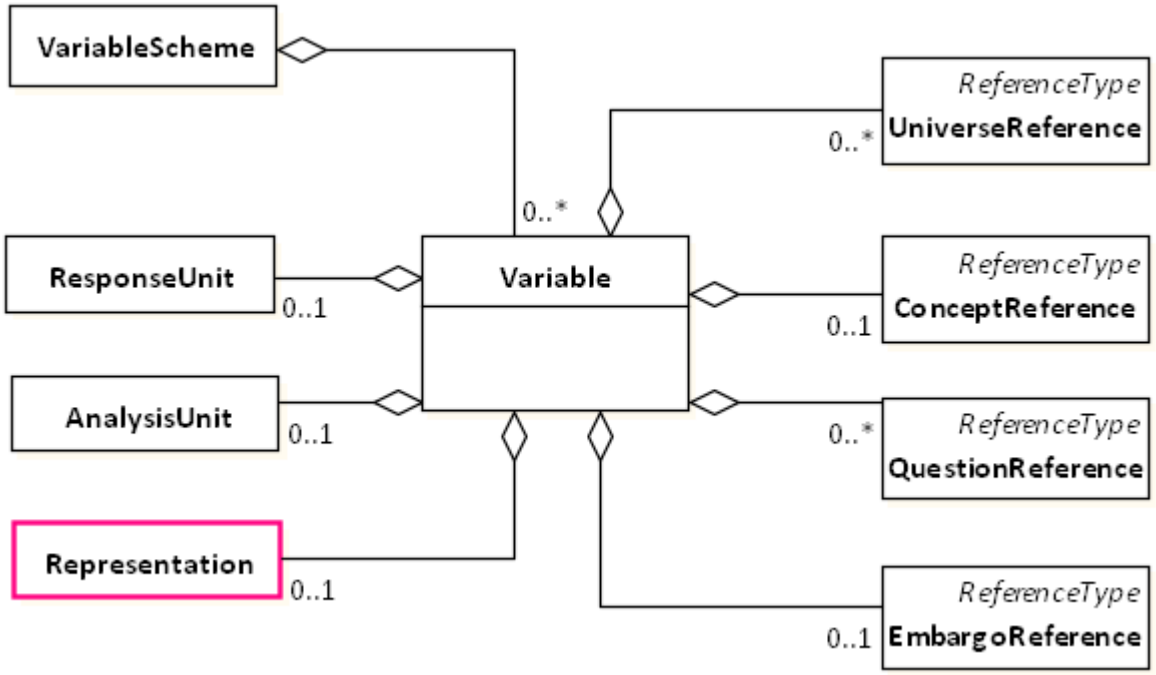
Variable


BaseLogicalProduct
LogicalProduct
NCubeLogicalProduct
ResourcePackage
VariableScheme
0..*
ReferenceType
VariableSchemeReference
Variable
ReferenceType
VariableReference
VariableGroup
ReferenceType
VariableGroupReference

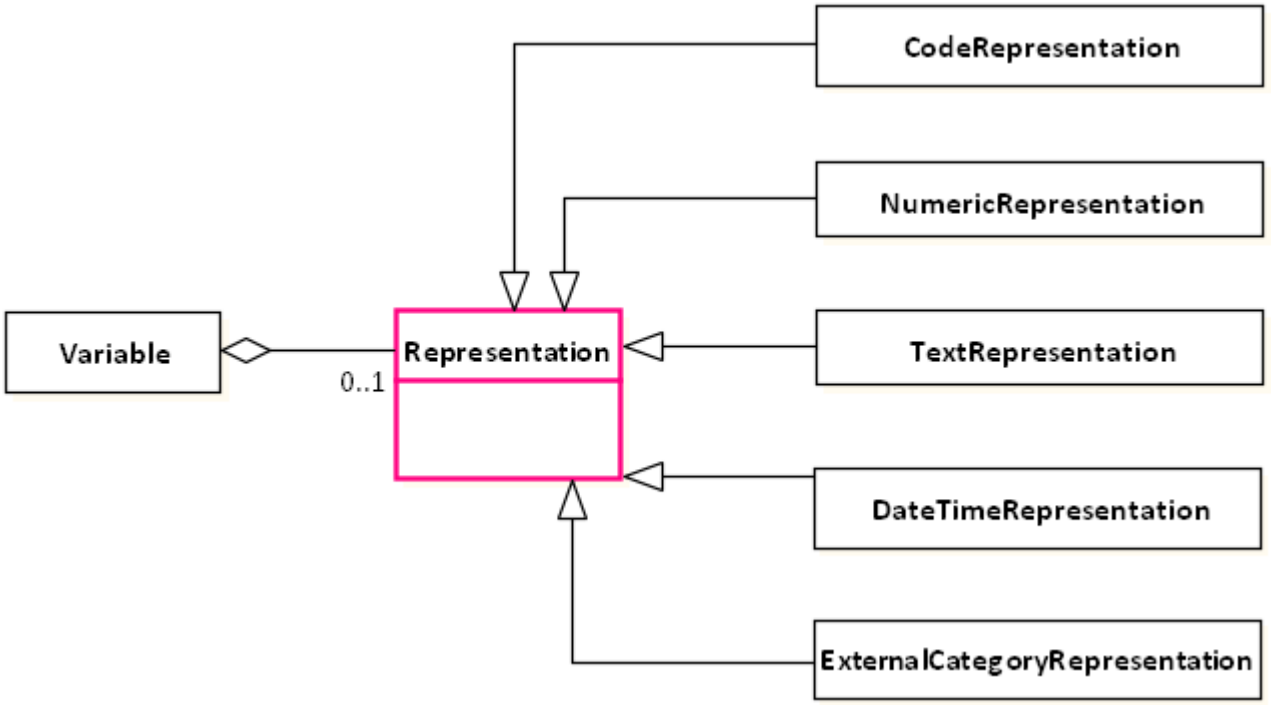
Variable


VariableScheme
0..*
ReferenceType
UniverseReference
0..*
ResponseUnit
0..1
Variable
ReferenceType
ConceptReference
0..1
AnalysisUnit
0..1
0..*
ReferenceType
QuestionReference
Representation
0..1
ReferenceType
EmbargoReference
0..1

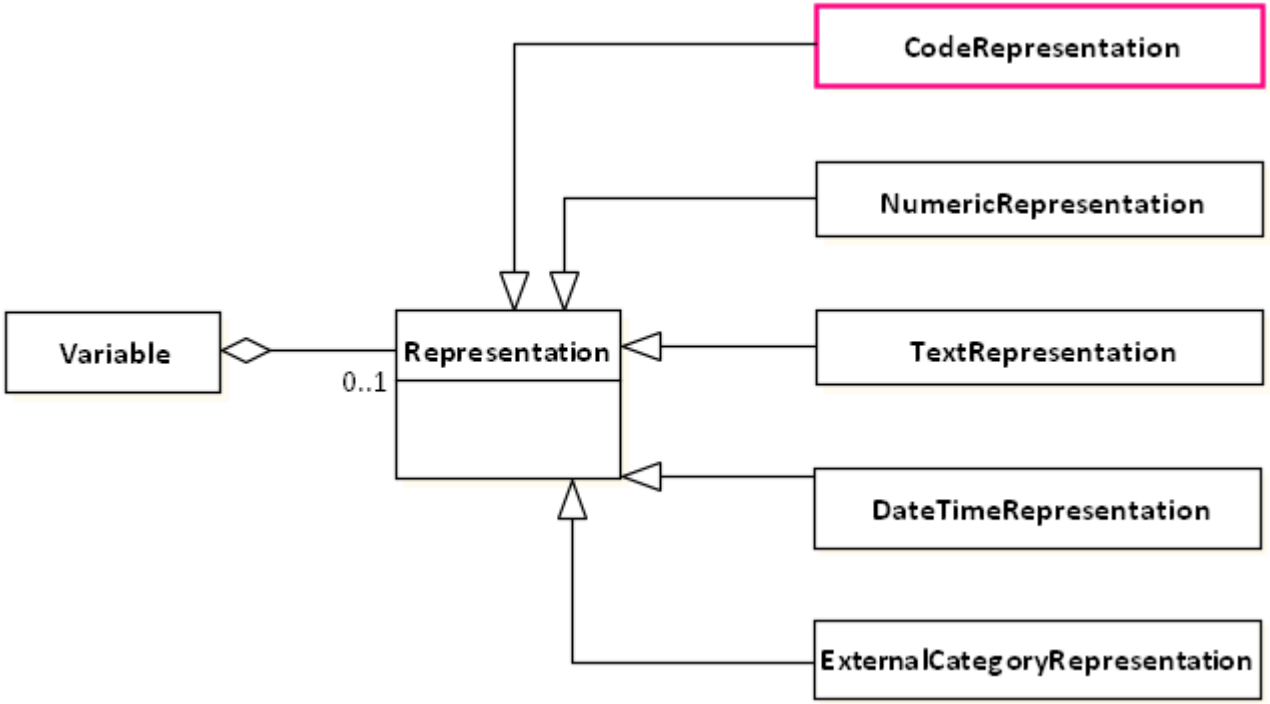
Representation



Representation

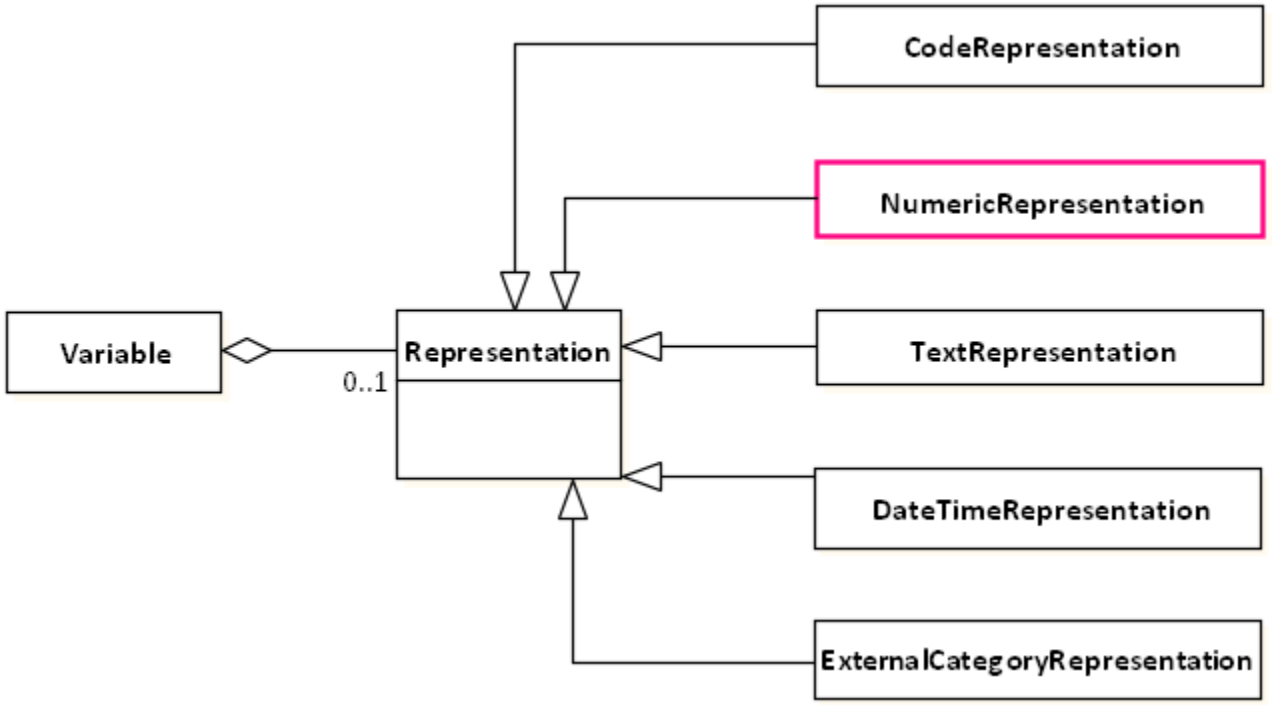


CodeRepresentation

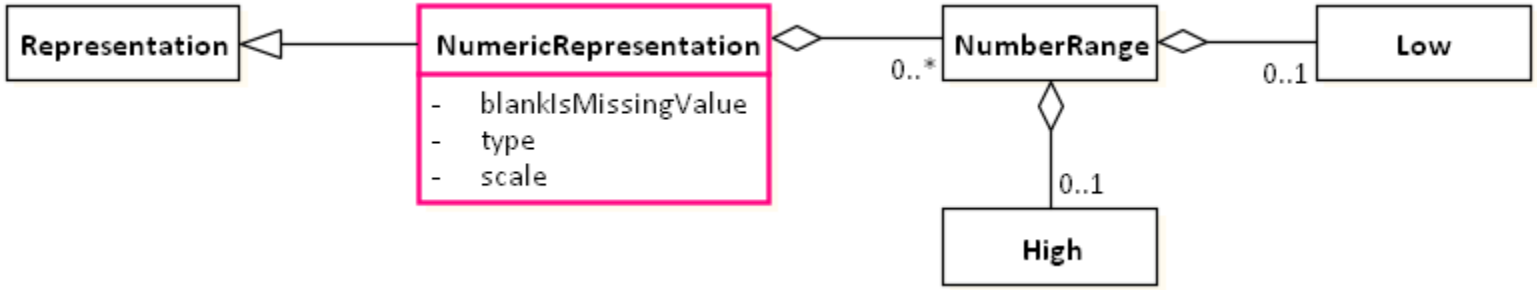


CodeRepresentation

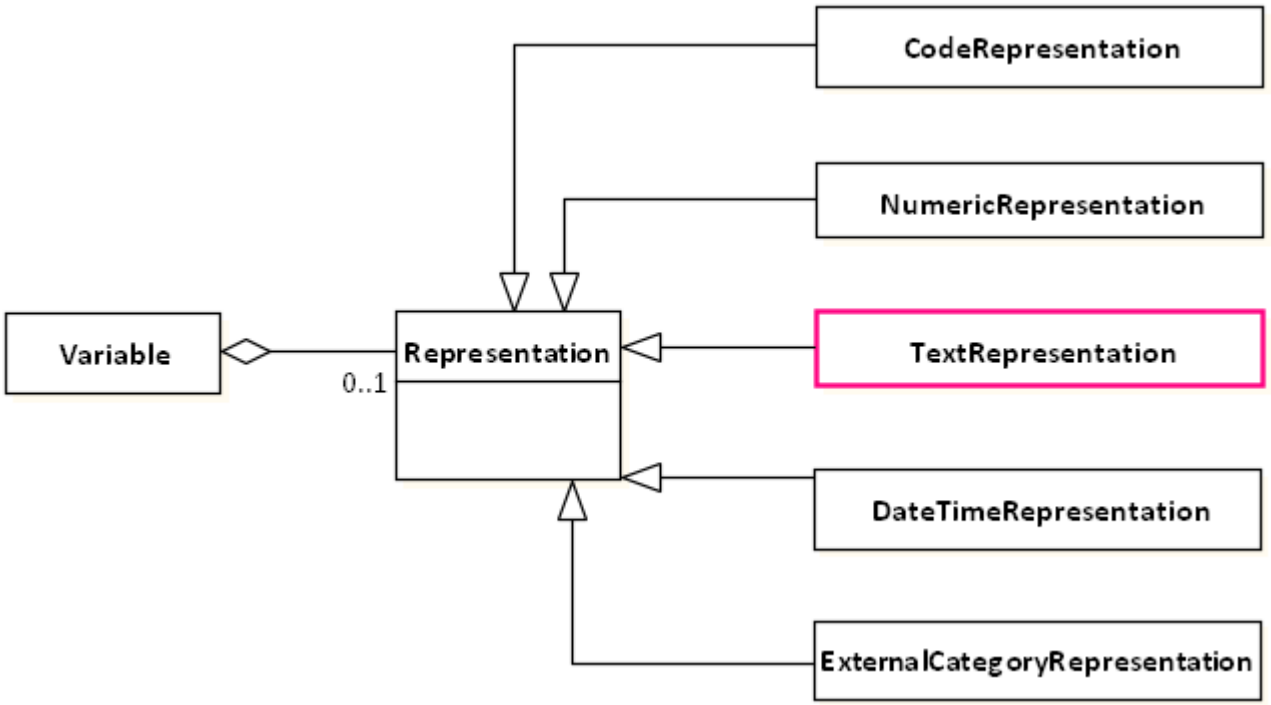
NumericRepresentation



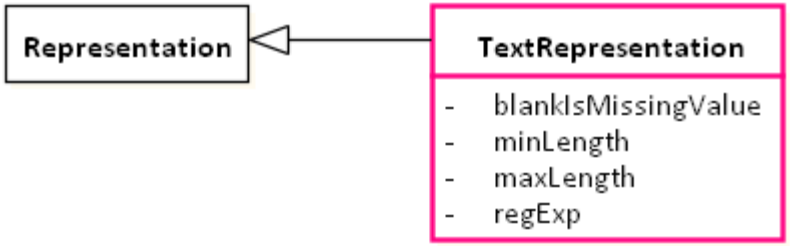
NumericRepresentation



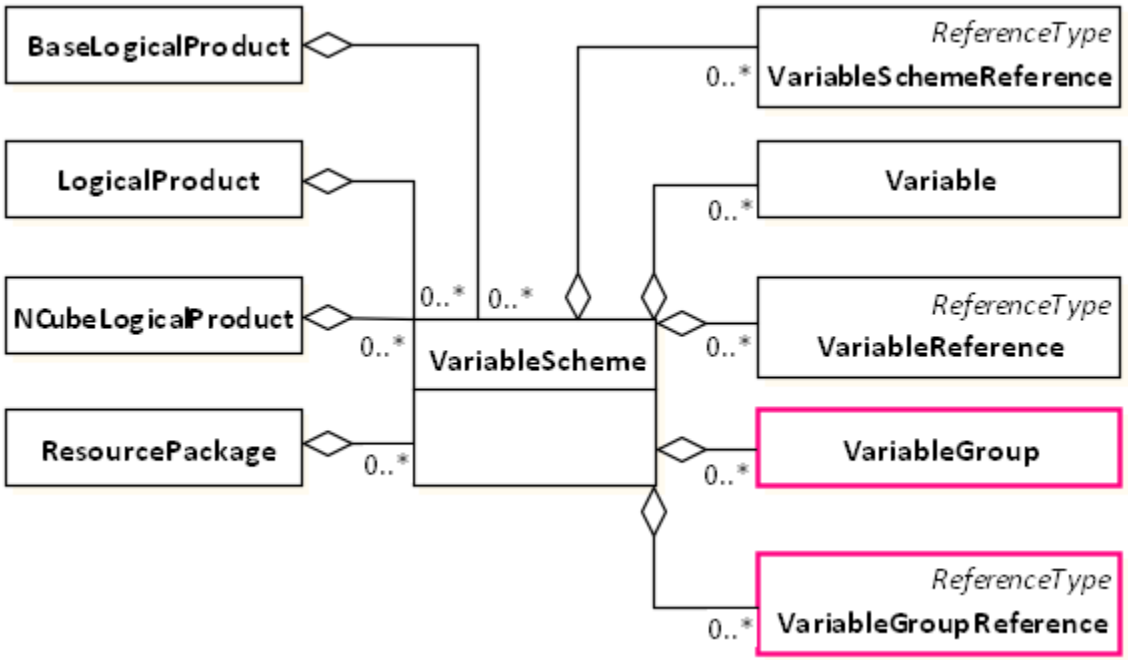
TextRepresentation



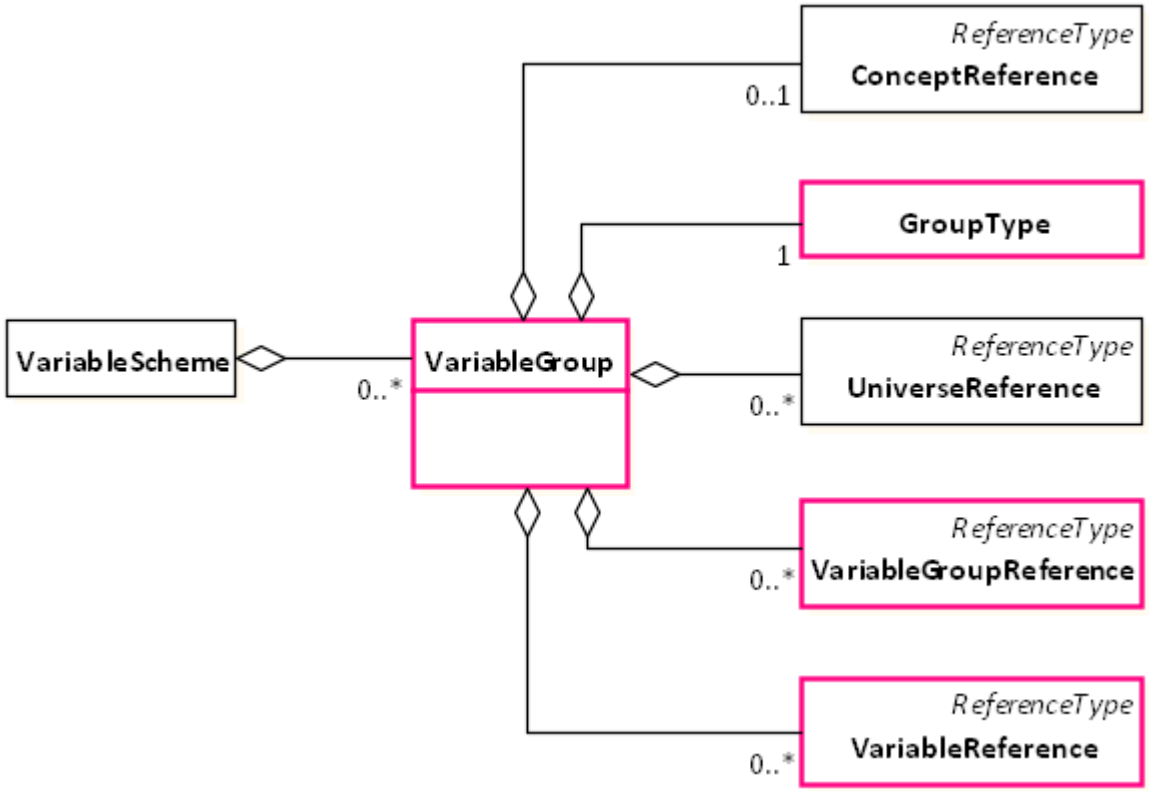
TextRepresentation



VariableGroup



VariableGroup



What comes next?

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy

Presentation

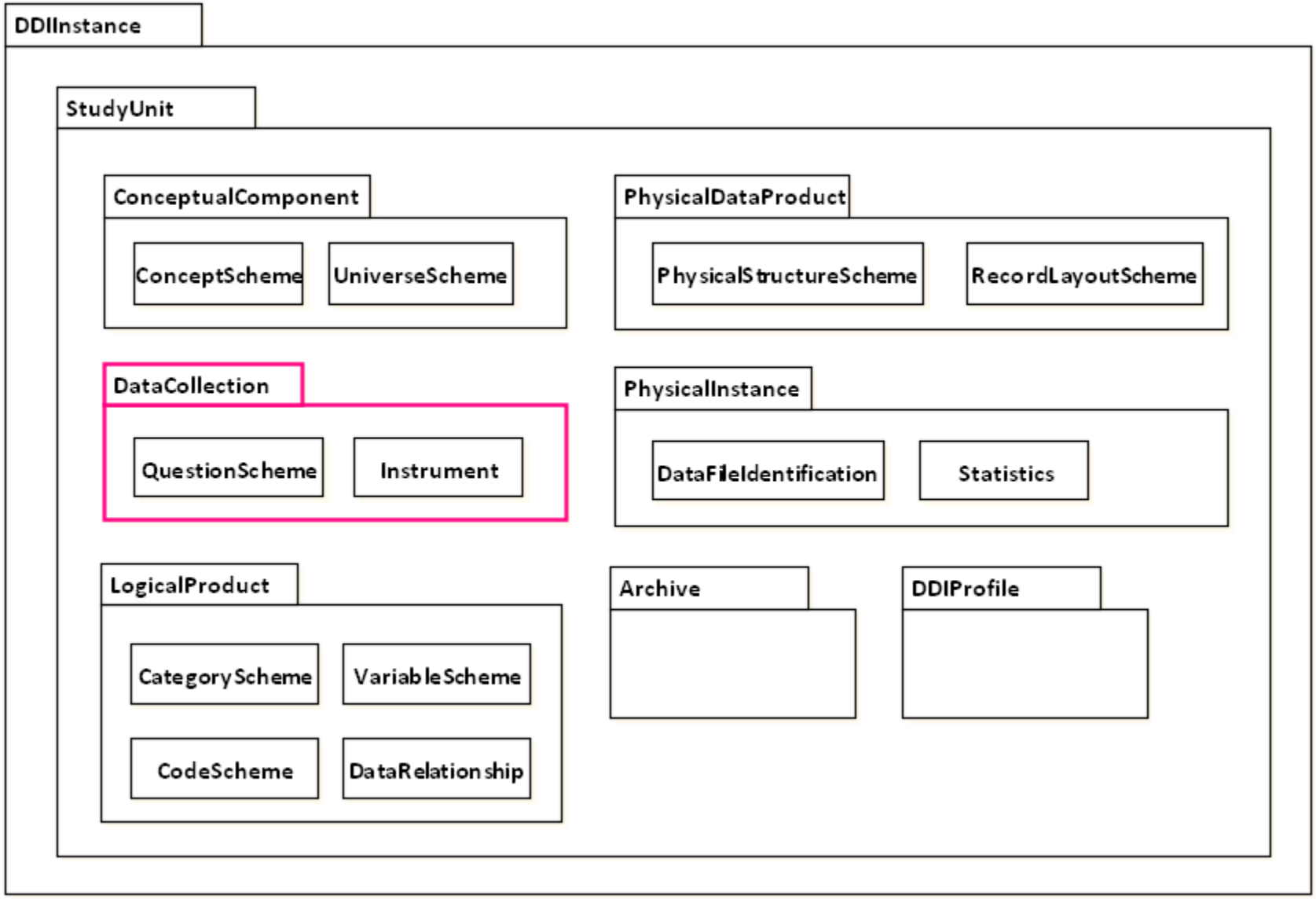
- DDI-L Overview
- Identification, Versioning, Maintenance
- DDI-L Main Structures
- DDI Instance
- Study Unit

- Software Architecture
  - Multitier
  - MVC
- Missy Data Model
  - Extendable Data Model

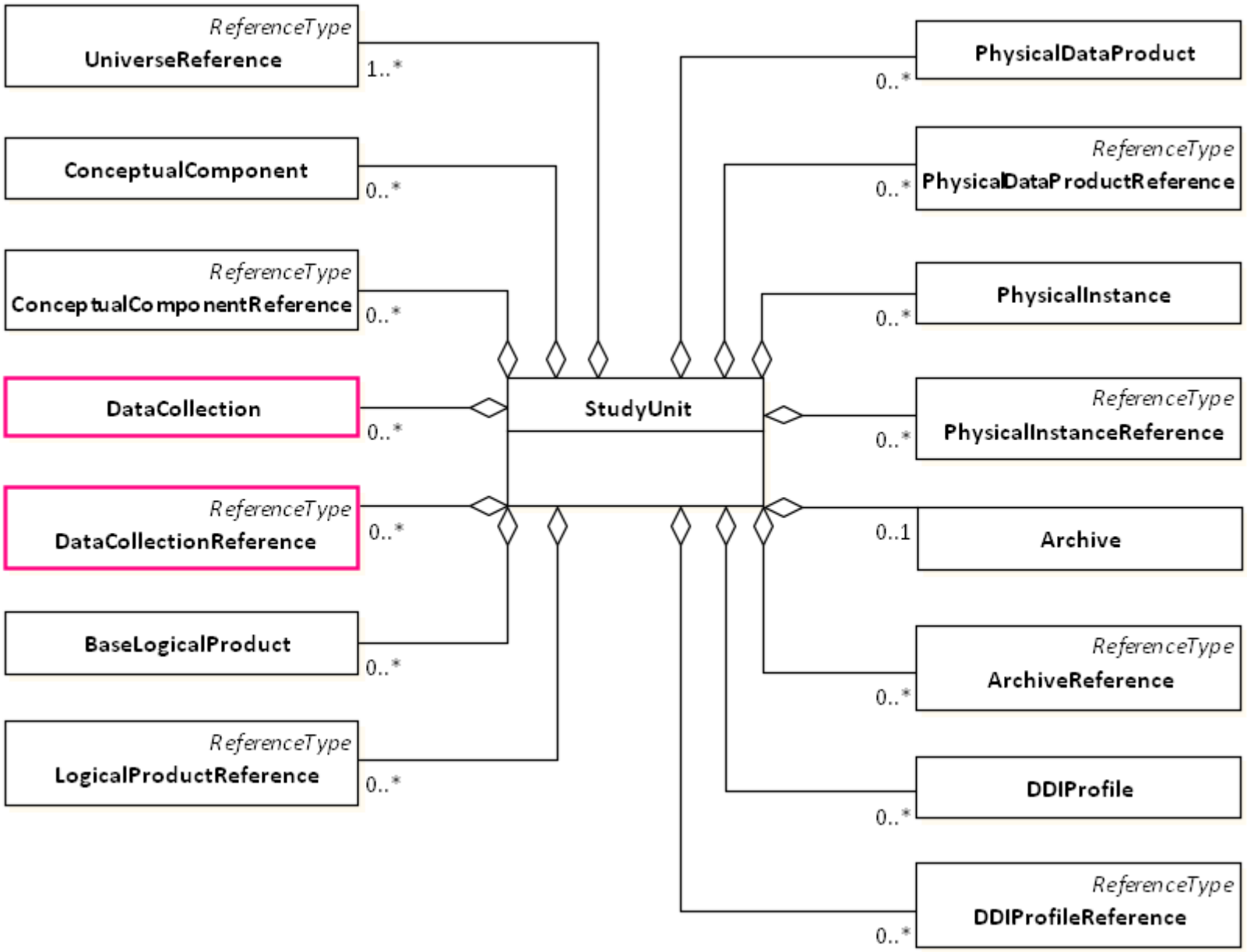
- Conceptual Component
- Logical Products
- **Data Collection**
- disco-model

Business Layer

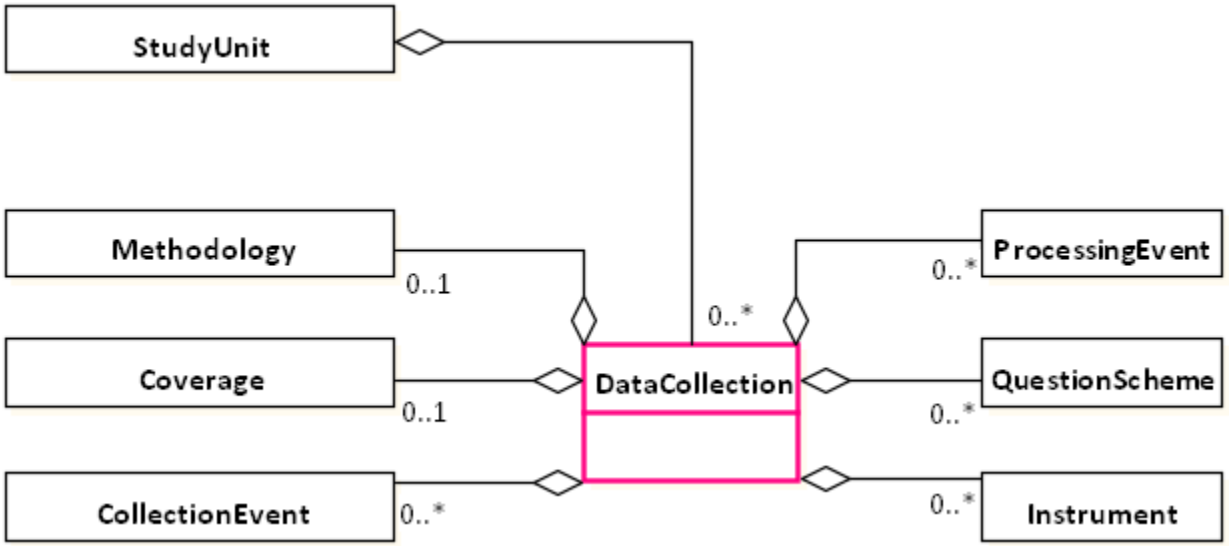
DataCollection



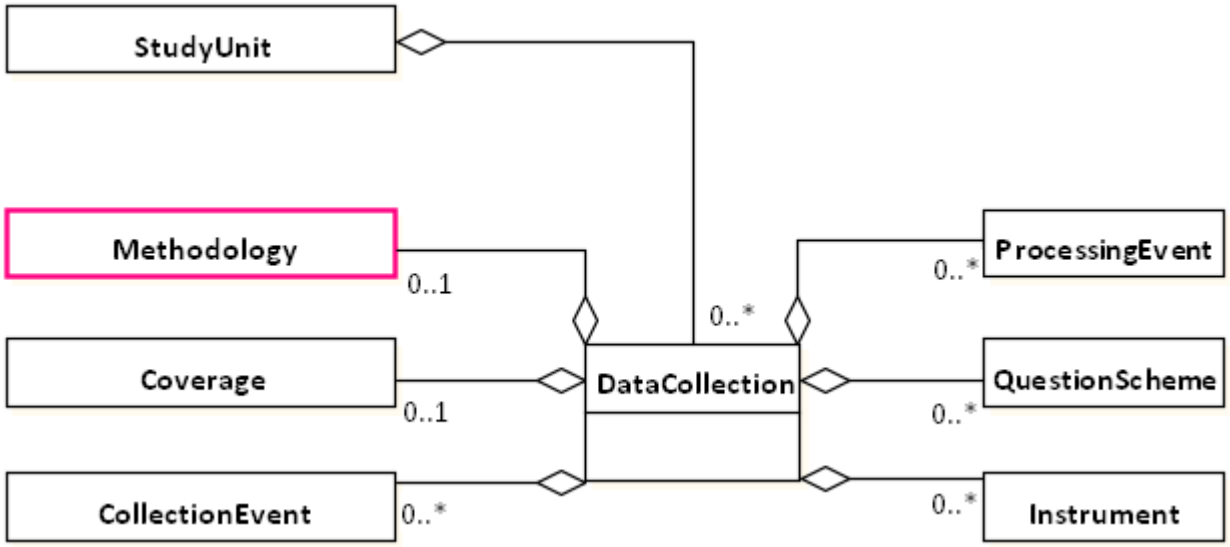
DataCollection



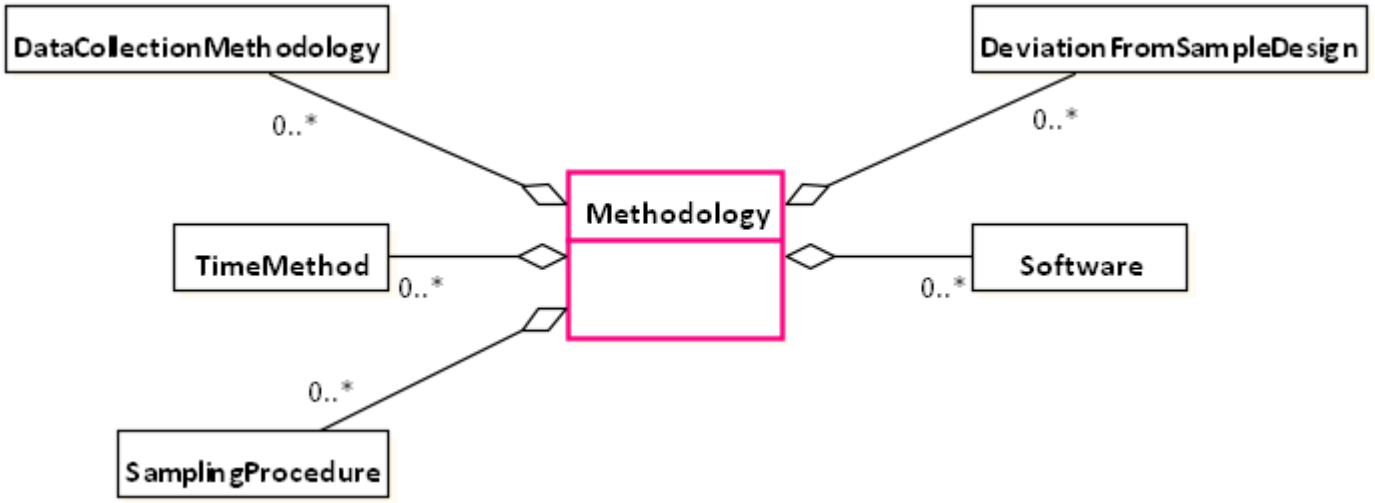
DataCollection



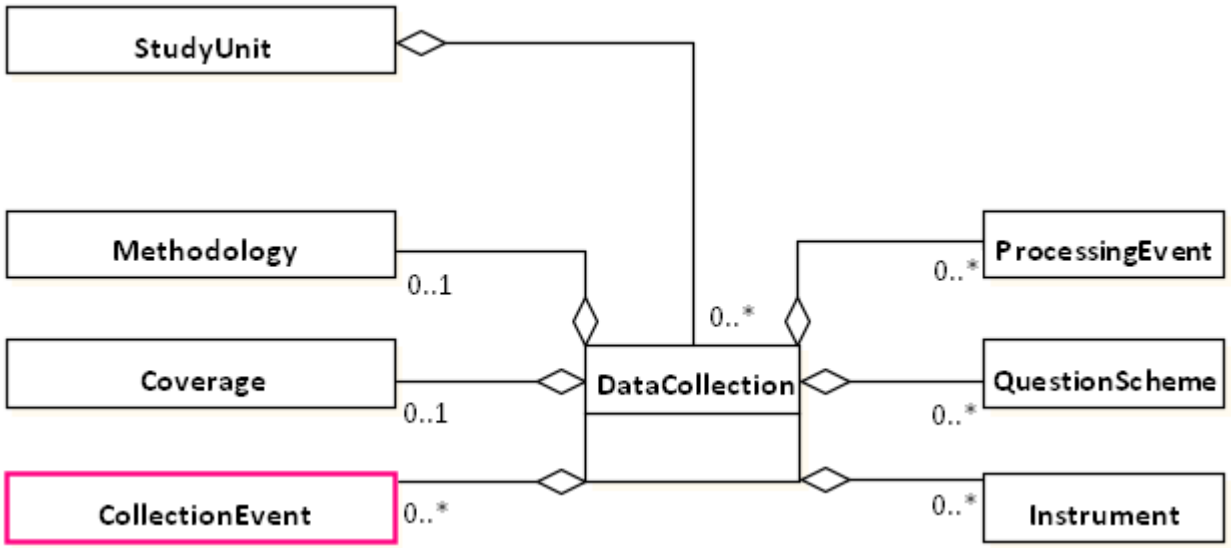
Methodology



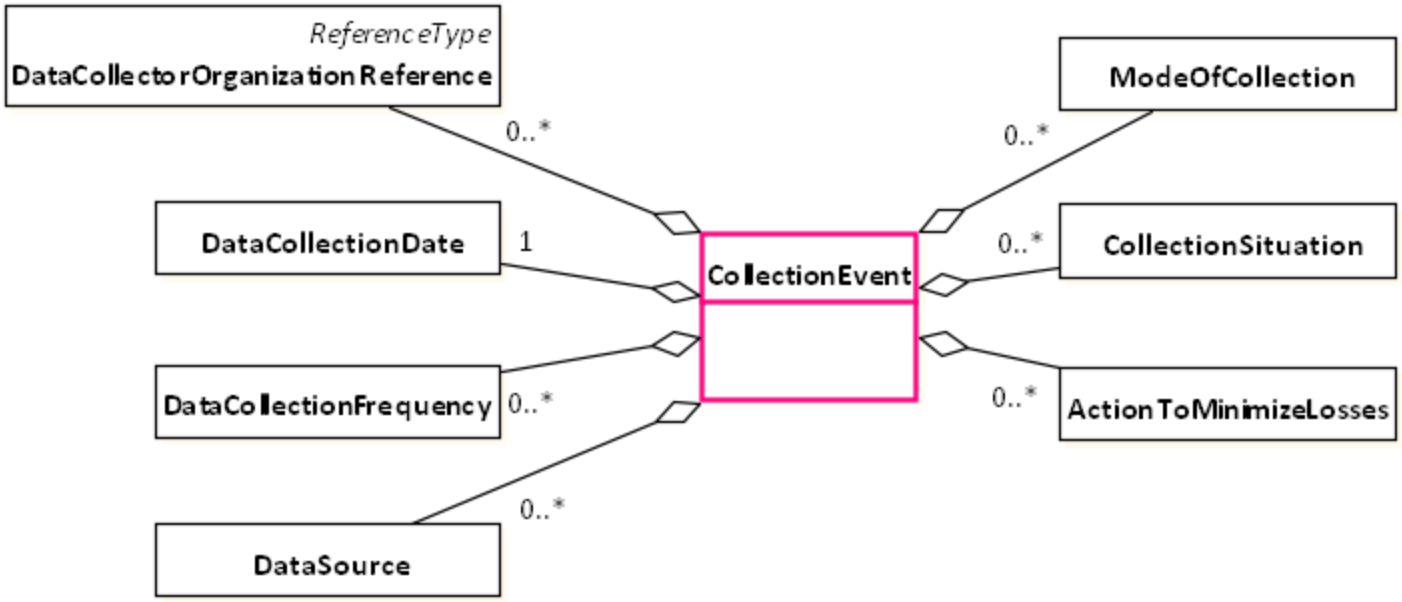
Methodology



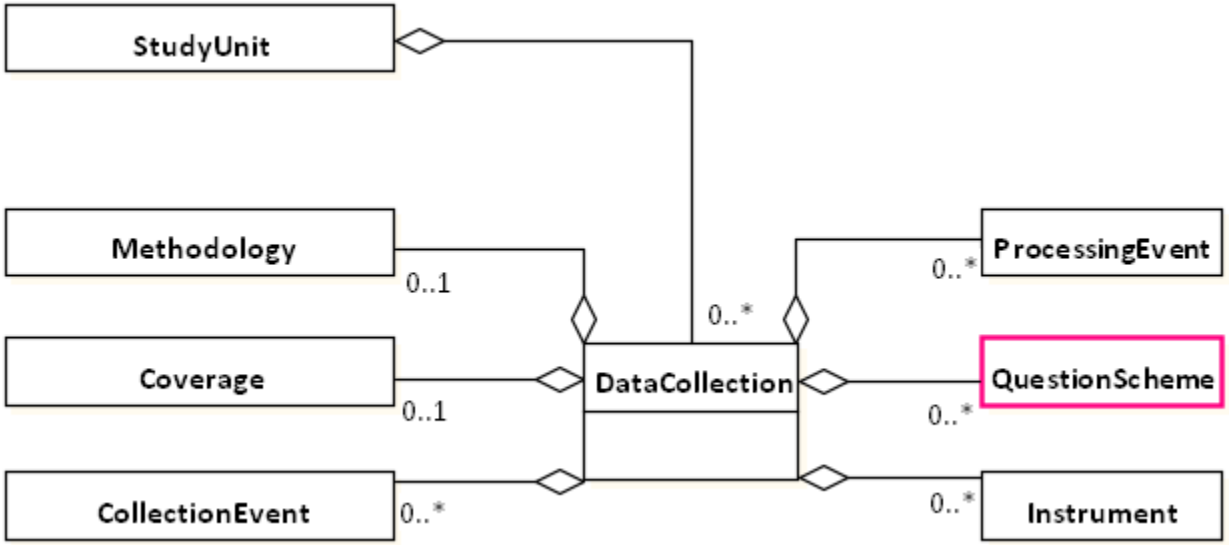
CollectionEvent



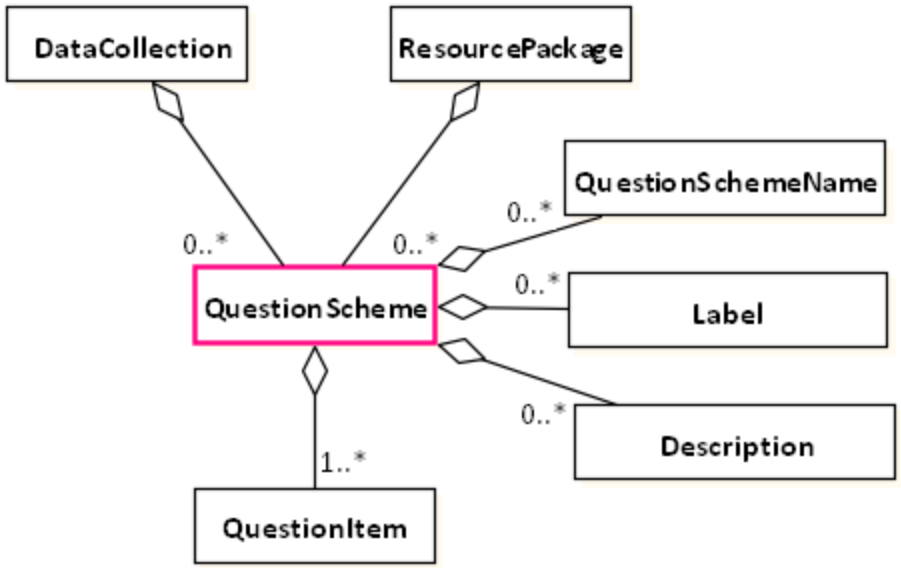
CollectionEvent



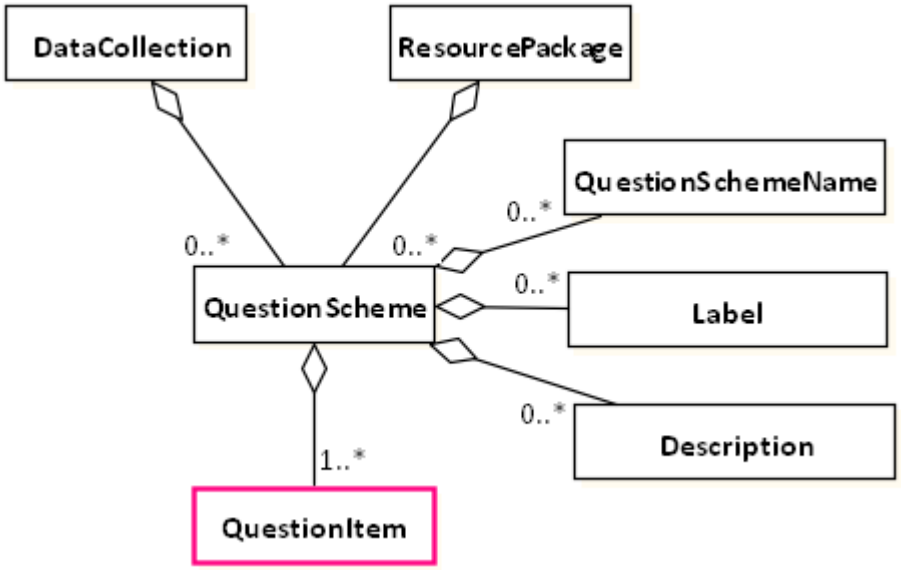
QuestionScheme



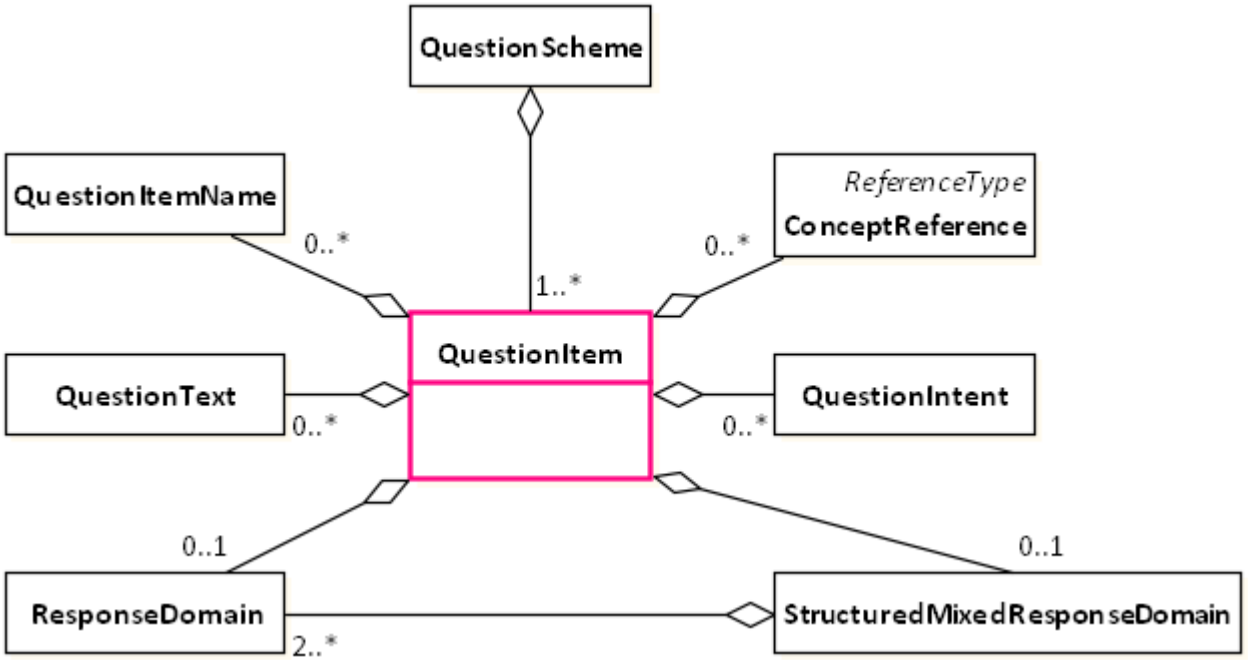
QuestionScheme



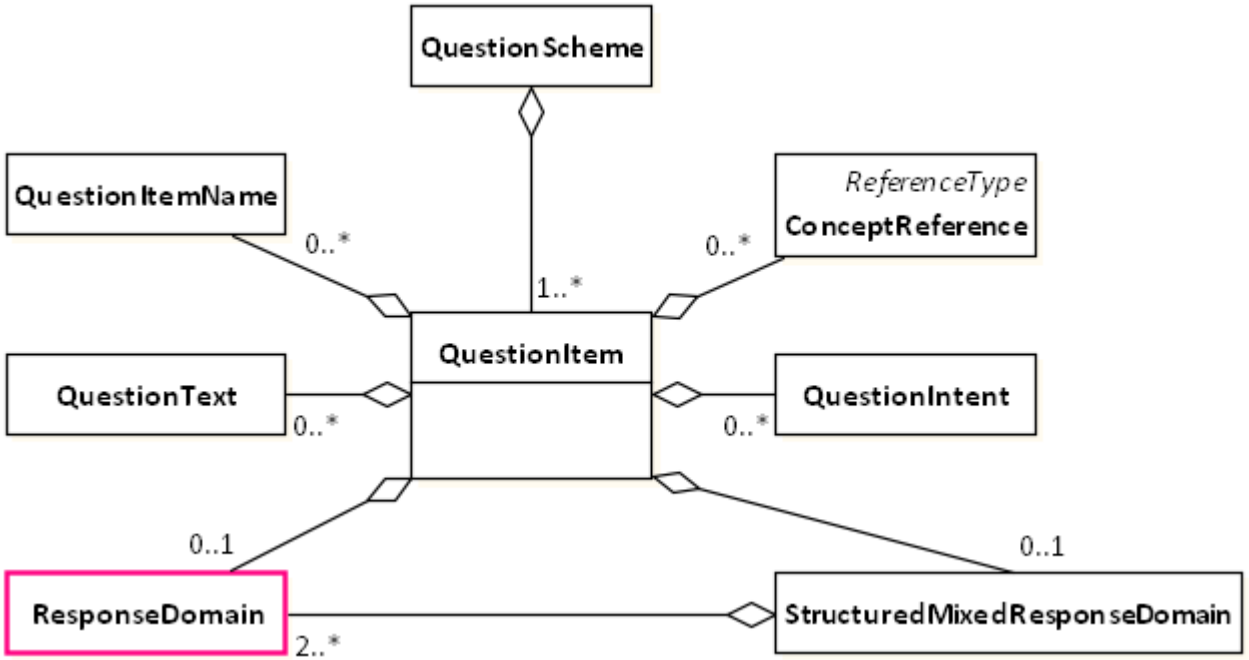
QuestionItem



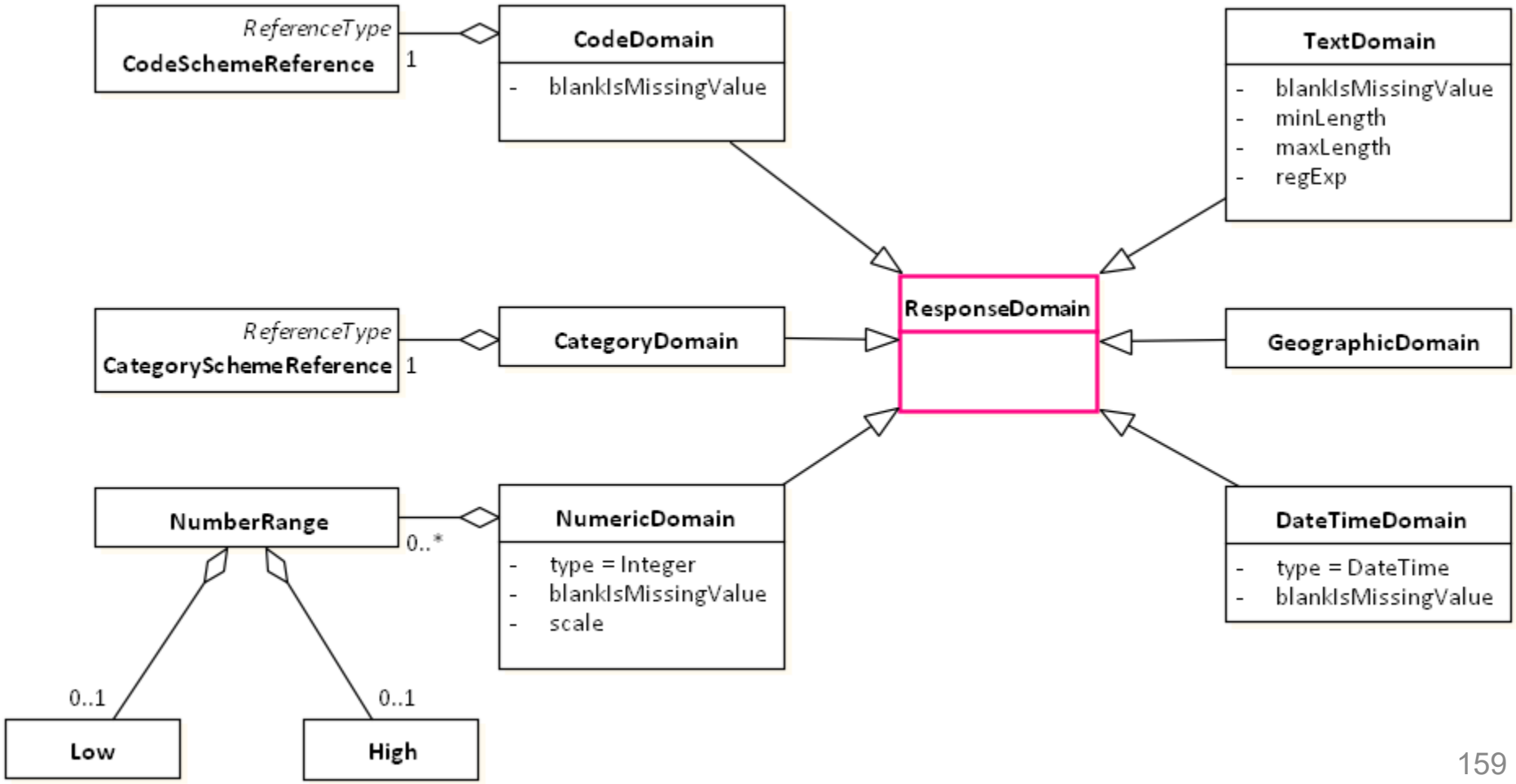
QuestionItem



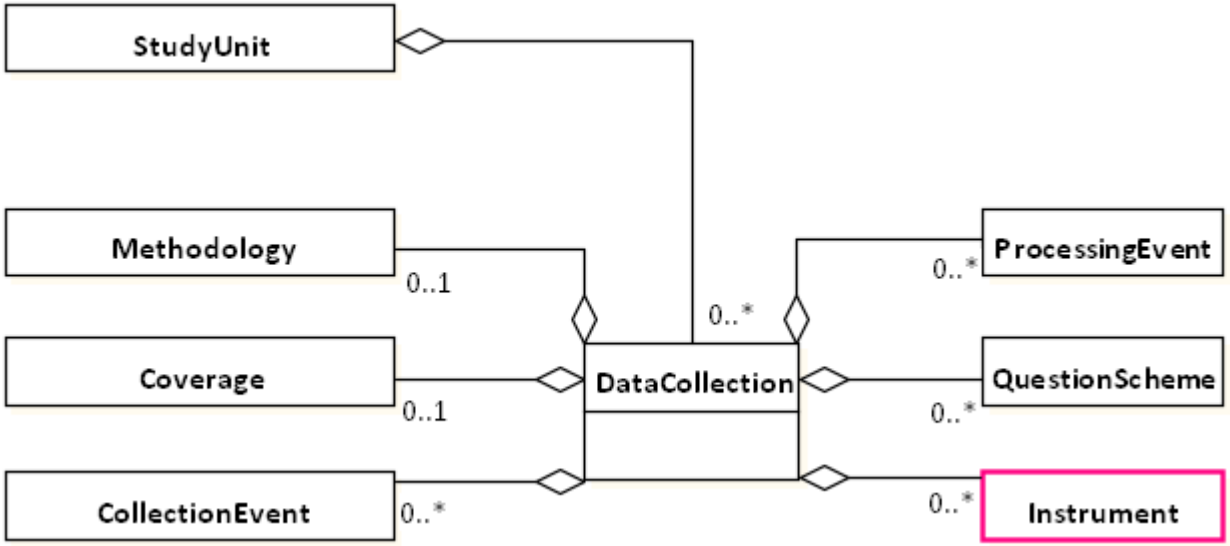
Response Domain



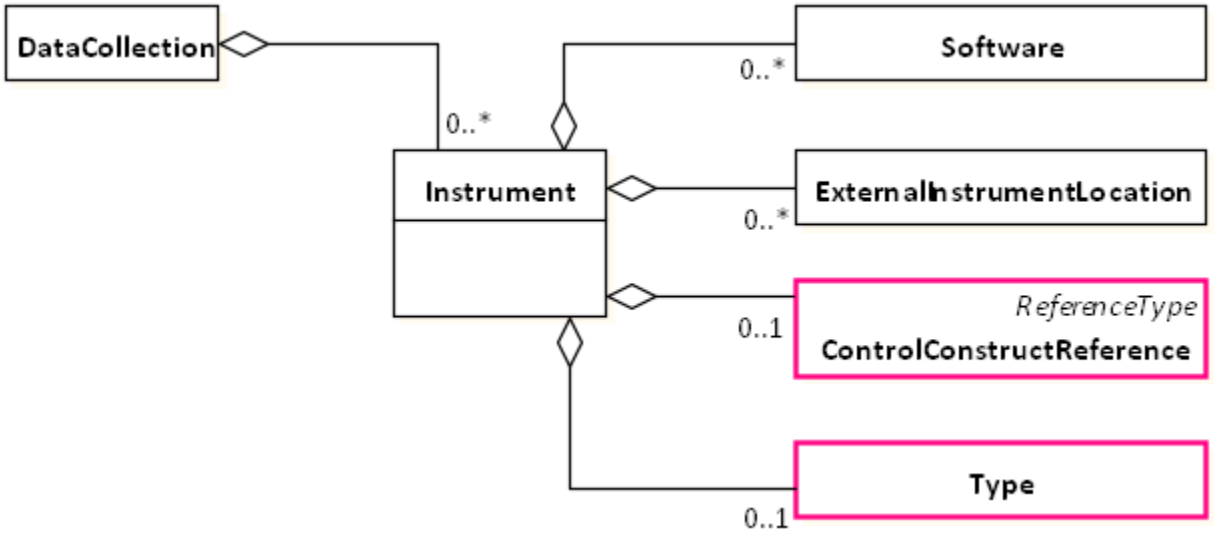
Response Domain



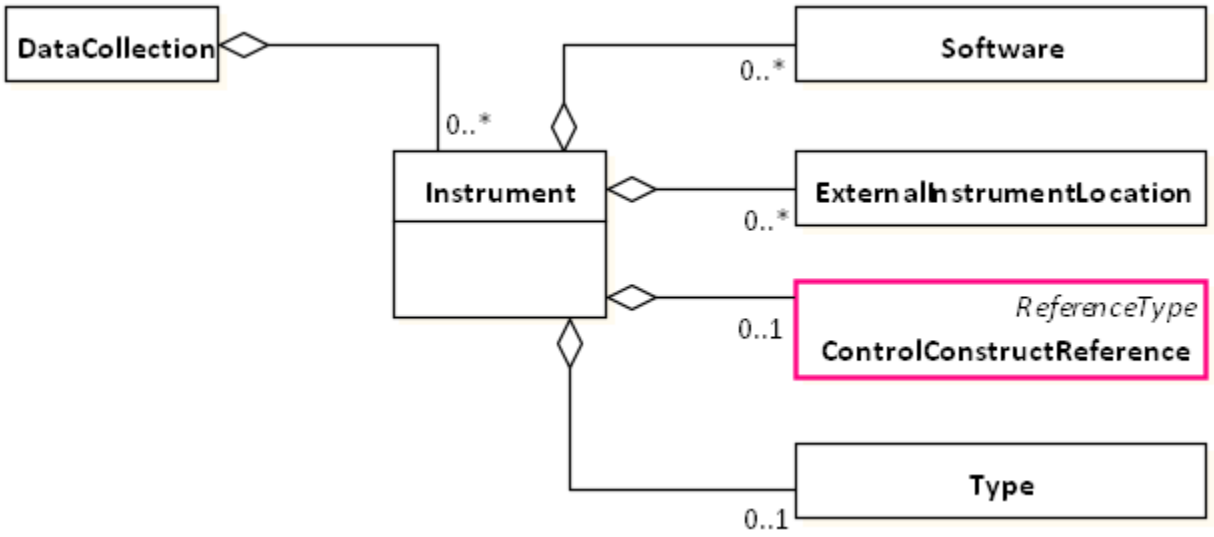
Instrument


StudyUnit
Methodology
Coverage
CollectionEvent
DataCollection
ProcessingEvent
QuestionScheme
Instrument
0..1
0..1
0..*
0..*
0..*
0..*
0..*

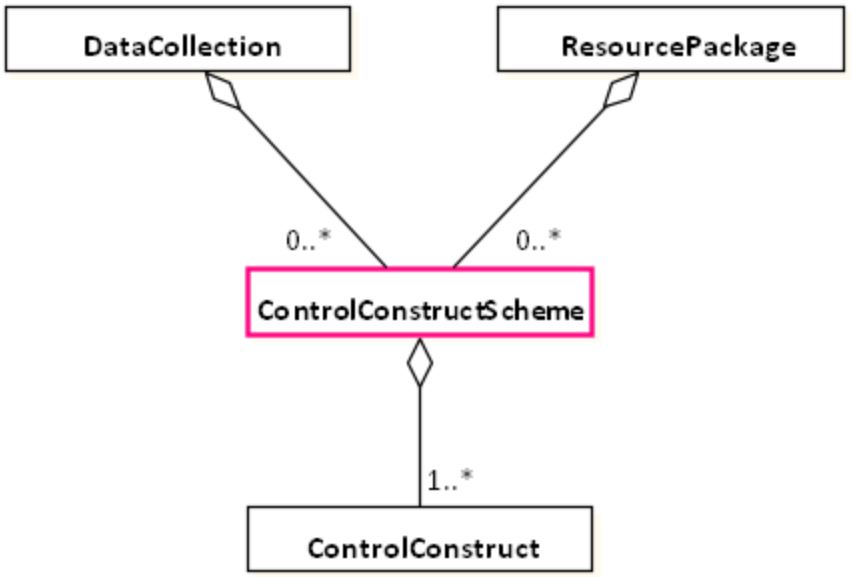
Instrument


DataCollection
0..*
Instrument
0..*
Software
0..*
ExternalInstrumentLocation
0..1
ReferenceType
ControlConstructReference
0..1
Type

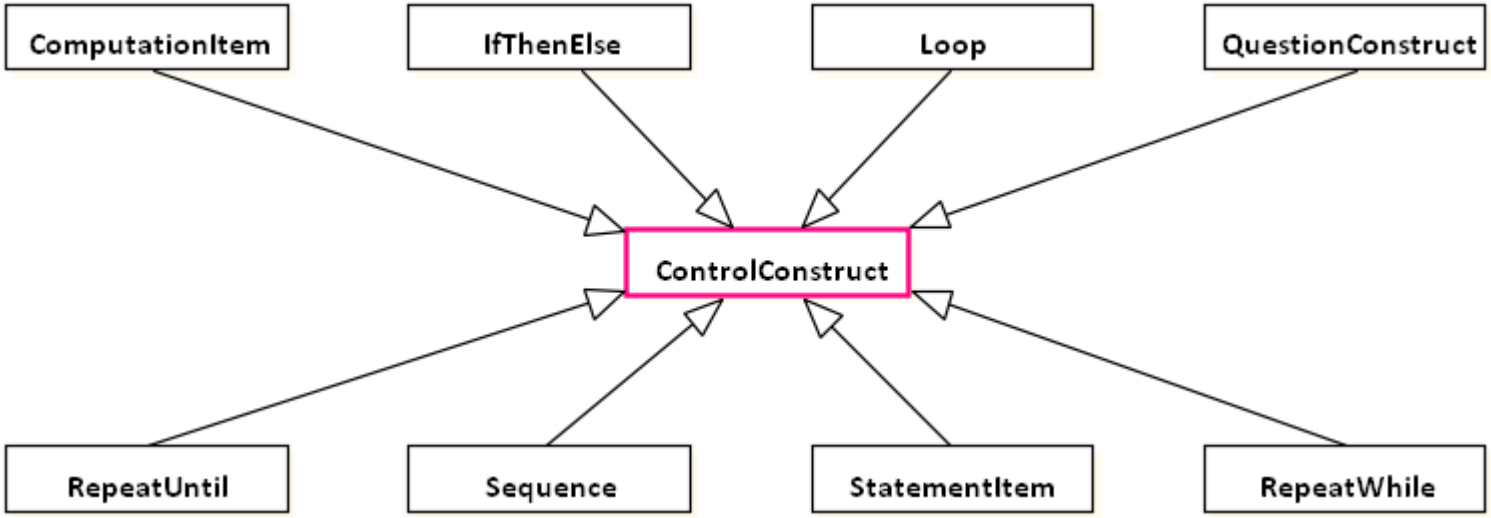
ControlConstruct



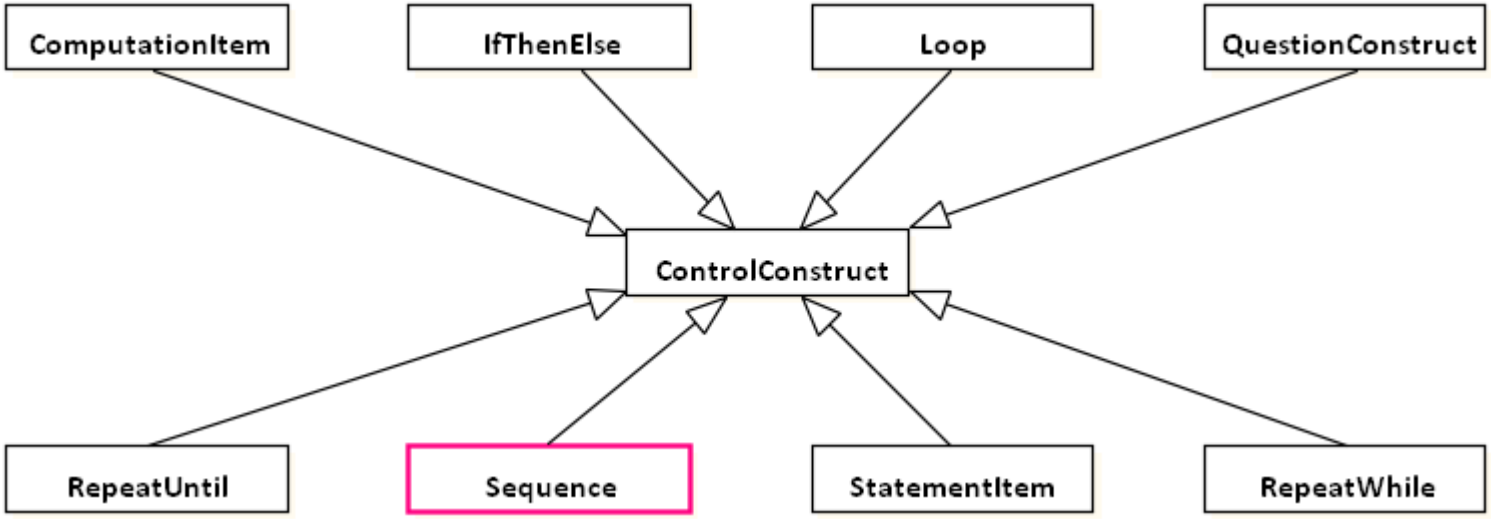
ControlConstructScheme



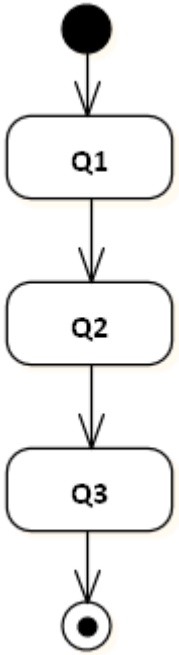
ControlConstruct



Sequence


ComputationItem
IfThenElse
Loop
QuestionConstruct
ControlConstruct
RepeatUntil
Sequence
StatementItem
RepeatWhile

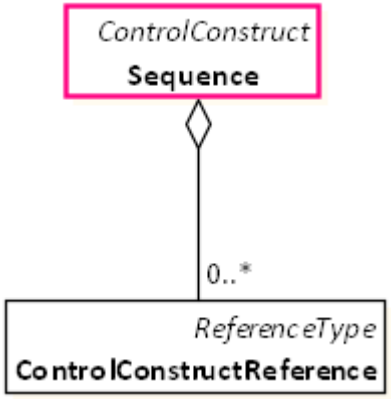
Sequence (bahavioral)



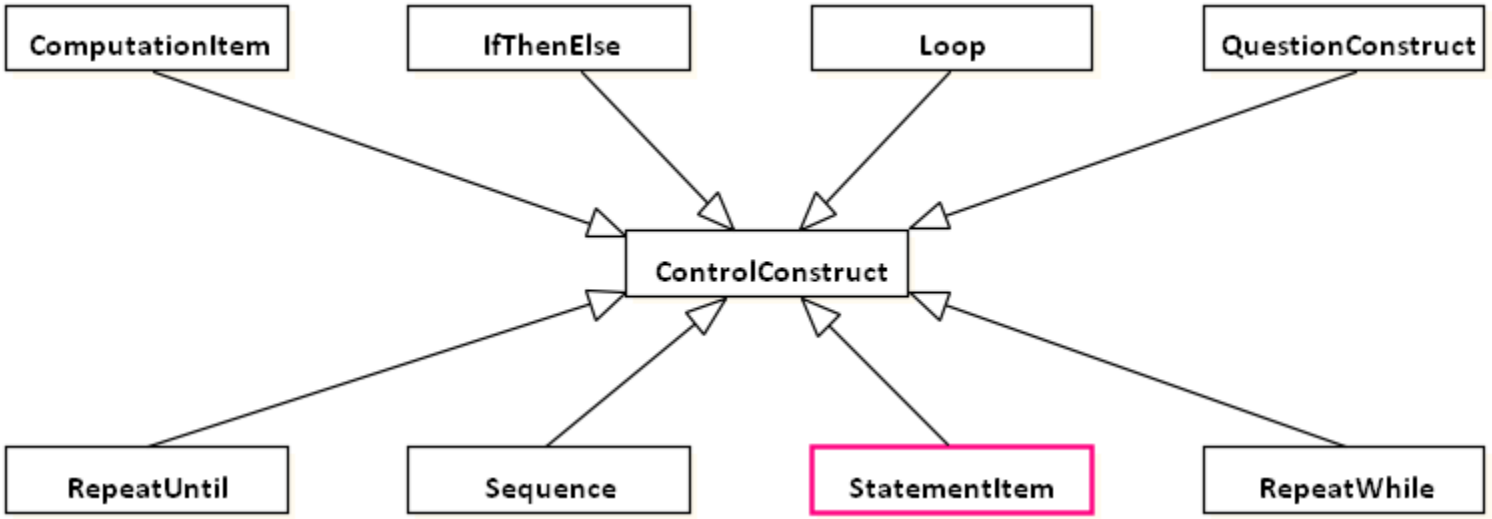
Sequence (XML)

```
<Sequence>
   <ControlConstructReference> [Q1] </ControlConstructReference>
   <ControlConstructReference> [Q2] </ControlConstructReference>
   <ControlConstructReference> [Q3] </ControlConstructReference>
</Sequence>
```

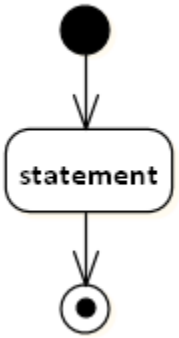
Sequence (structural)



StatementItem



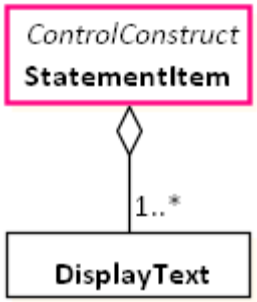
StatementItem (behavioral)



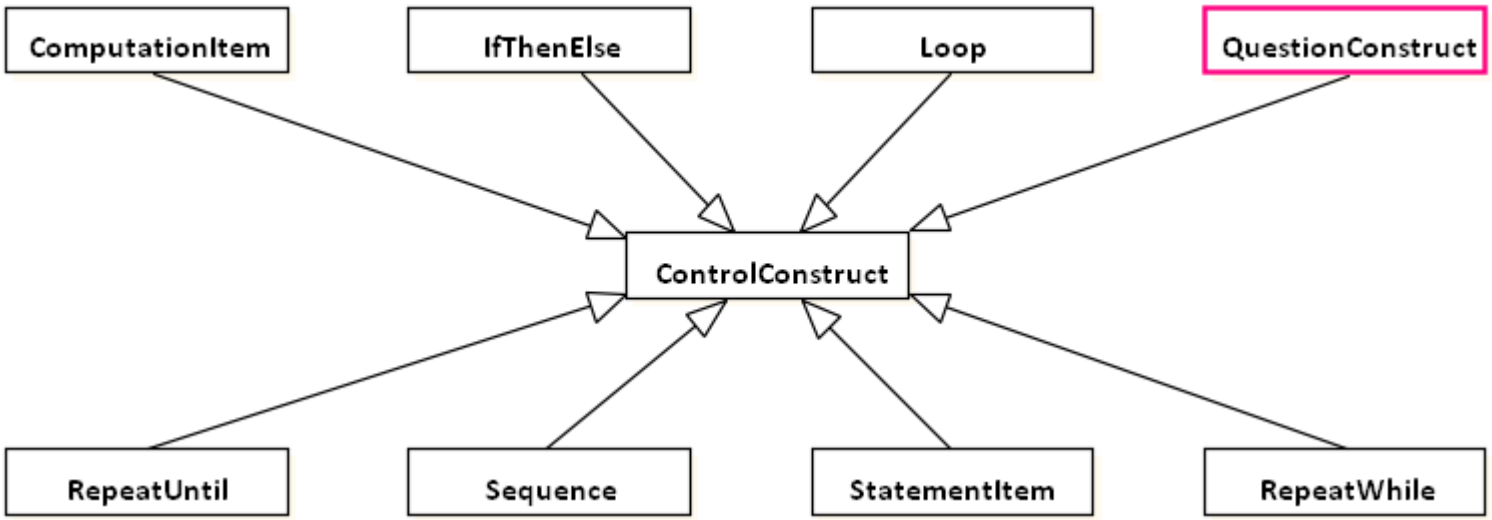
StatementItem (XML)

```
<StatementItem>
   <DisplayText>
      statement
   </DisplayText>
</StatementItem>
```

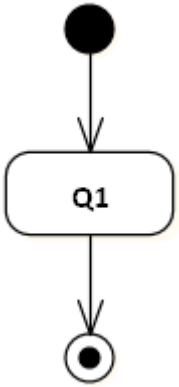
StatementItem (structural)



QuestionConstruct



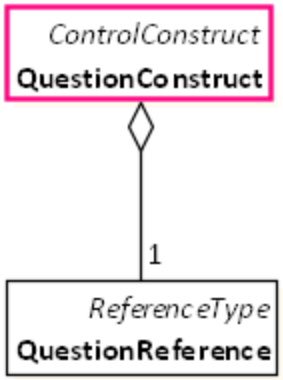
QuestionConstruct (behavioral)



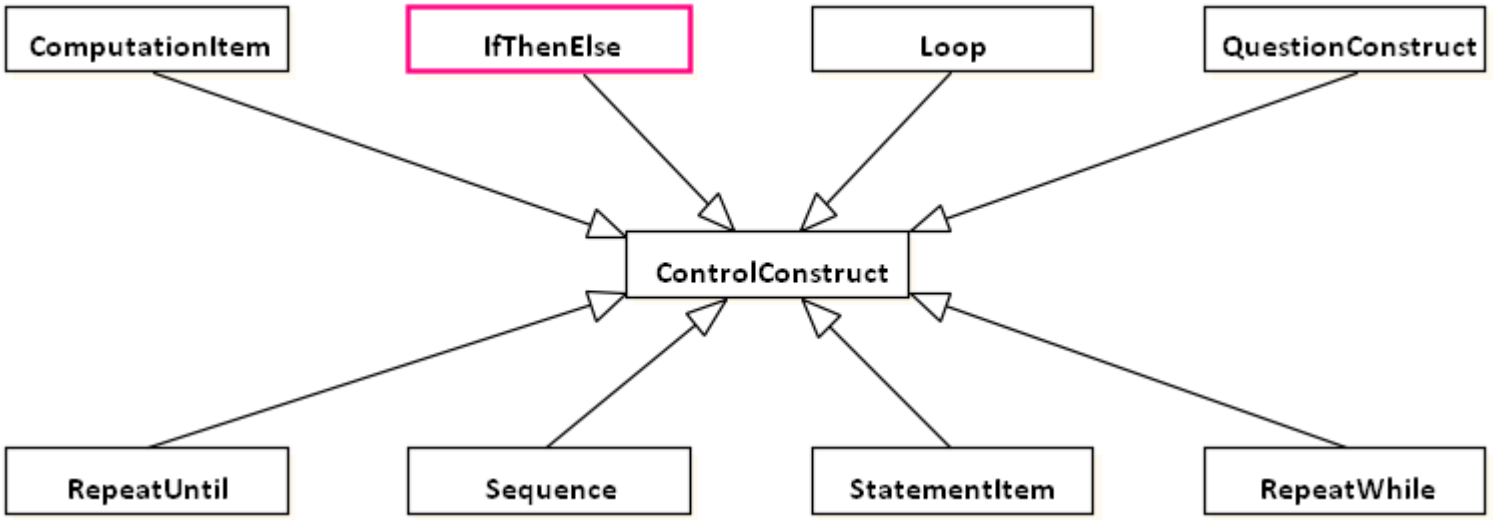
QuestionConstruct (XML)

```
<QuestionConstruct>
  <QuestionReference> [Q1] </QuestionReference>
</QuestionConstruct>
```

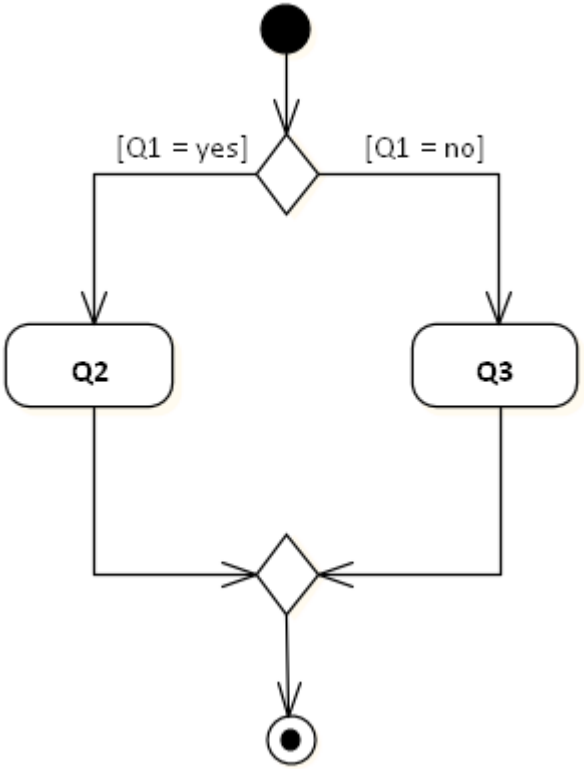
QuestionConstruct (structural)



IfThenElse


ComputationItem
IfThenElse
Loop
QuestionConstruct
ControlConstruct
RepeatUntil
Sequence
StatementItem
RepeatWhile

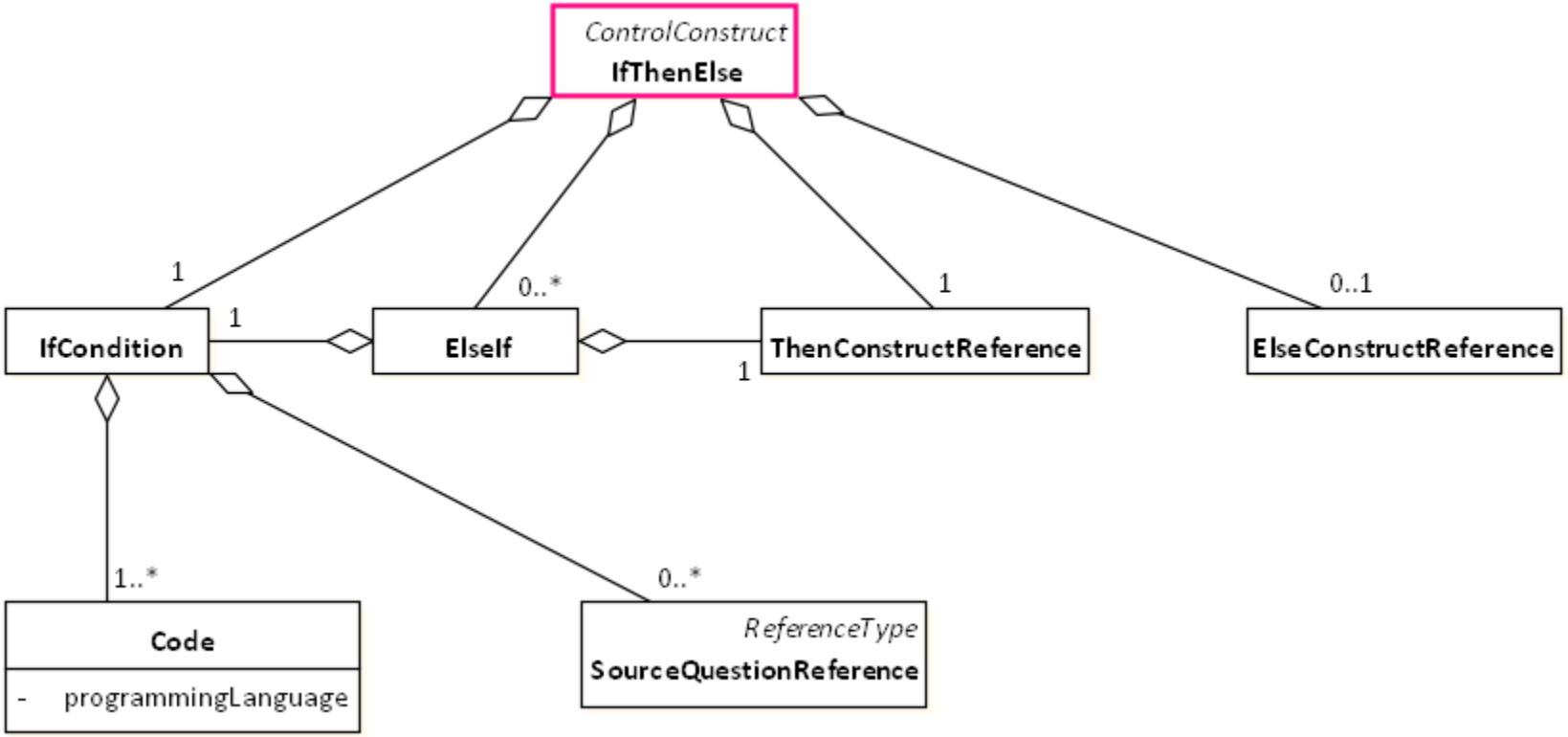
IfThenElse (behavioral)



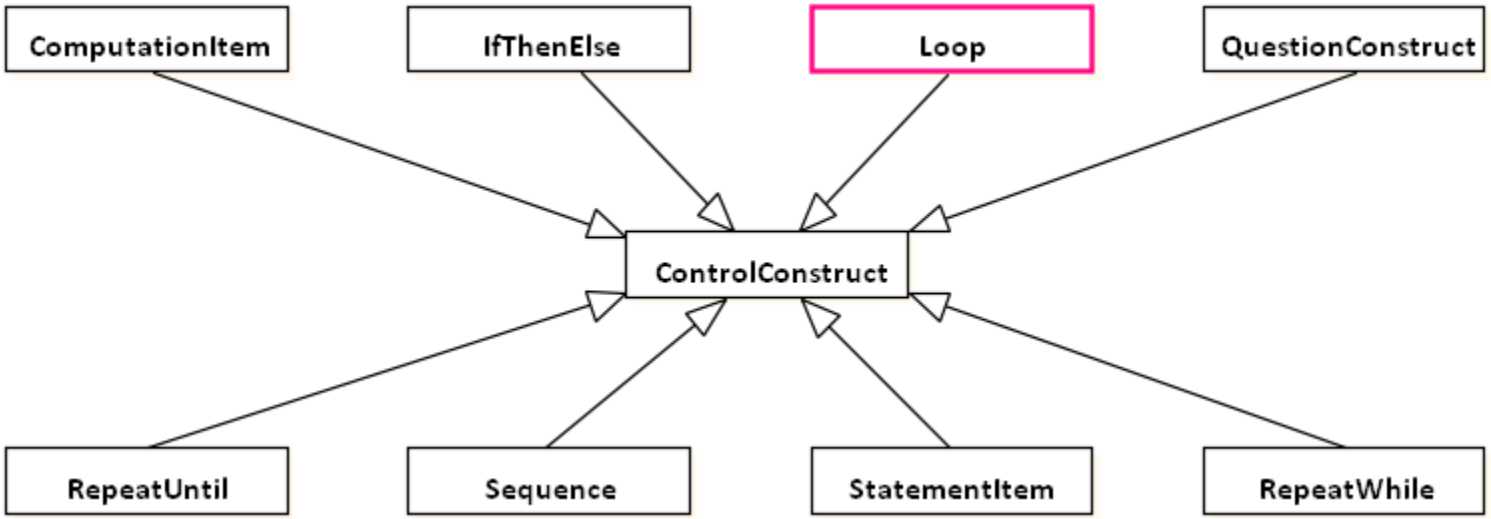
IfThenElse (XML)

```
<IfThenElse>
   <IfCondition>
      <Code programmingLanguage="…"> yes </Code>
      <SourceQuestionReference> [Q1] </SourceQuestionReference>
   </IfCondition>
   <ThenConstructReference> [Q2] </ThenConstructReference>
   <ElseIf>
      <IfCondition>
         <Code programmingLanguage="…"> no </Code>
         <SourceQuestionReference> [Q1] </SourceQuestionReference>
      </IfCondition>
      <ThenConstructReference> [Q3] </d:ThenConstructReference>
   </ElseIf>
</IfThenElse>
```

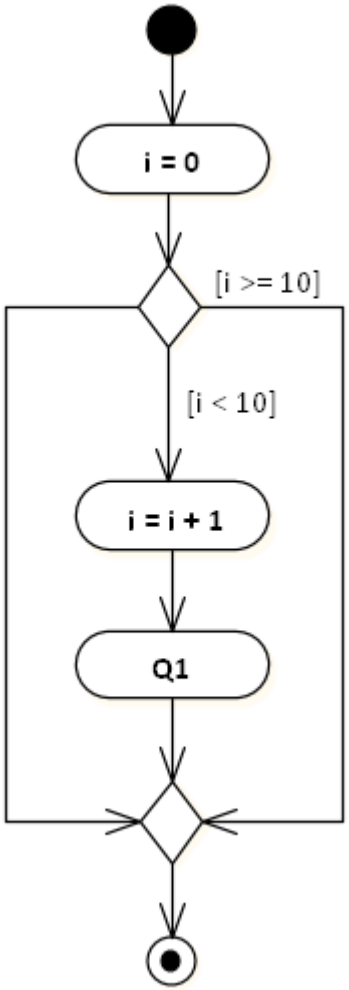
IfThenElse (structural)



Loop


ComputationItem
IfThenElse
Loop
QuestionConstruct
ControlConstruct
RepeatUntil
Sequence
StatementItem
RepeatWhile

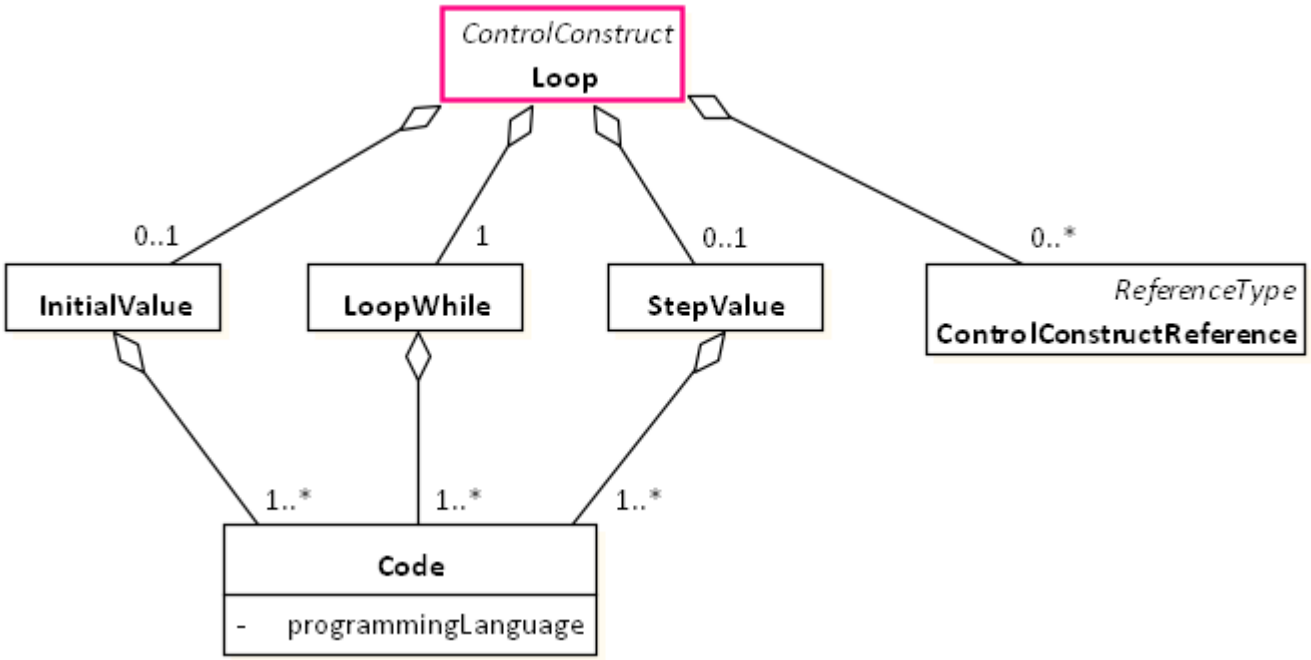
Loop (behavioral)



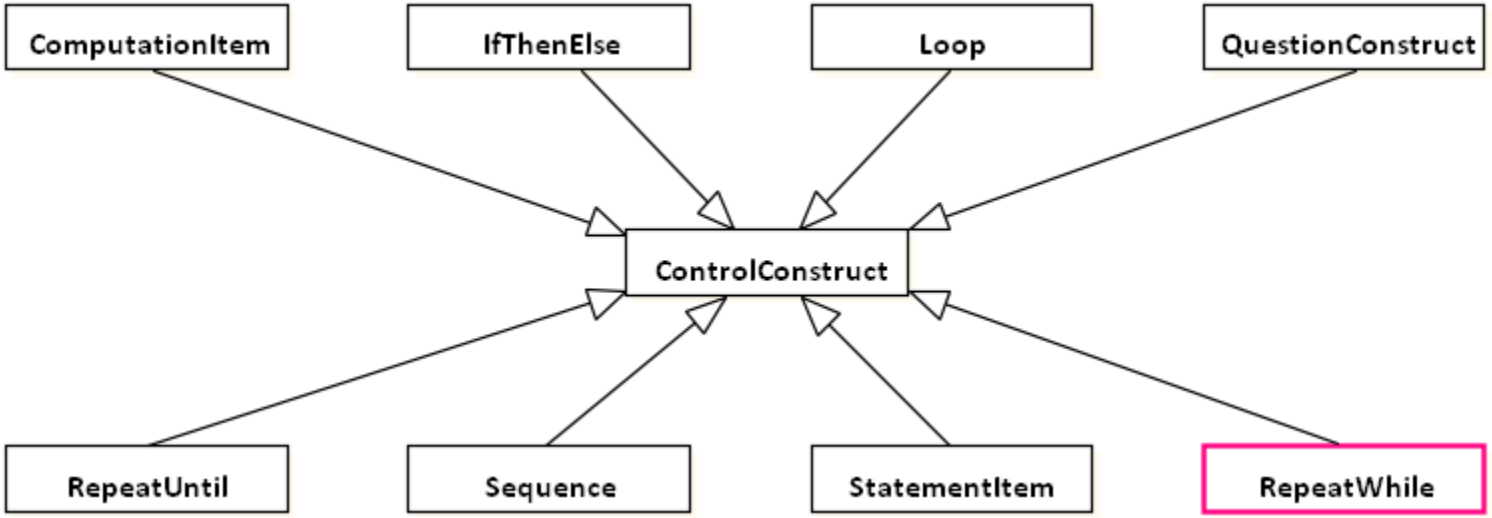
Loop (XML)

```
<Loop>
   <InitialValue>
      <Code programmingLanguage="Java"> i = 0 </Code>
   </InitialValue>
   <LoopWhile>
      <Code programmingLanguage="Java"> i < 10 </Code>
   </LoopWhile>
   <StepValue>
      <Code programmingLanguage="Java"> i = i + 1 </Code>
   </StepValue>
   <ControlConstructReference> [Q1] </ControlConstructReference>
</Loop>
```

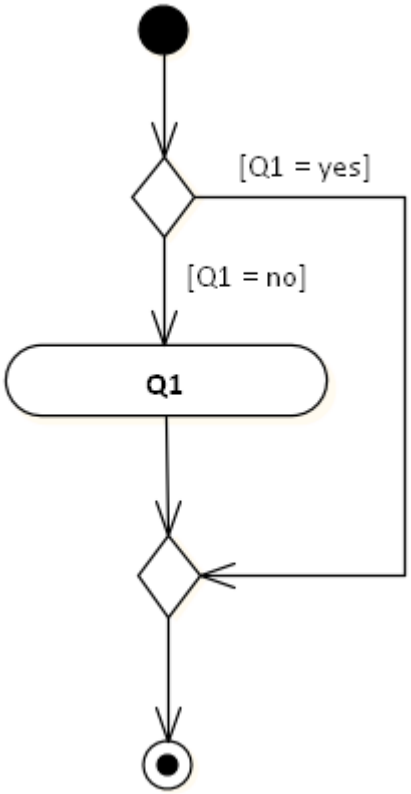
Loop (structural)



RepeatWhile



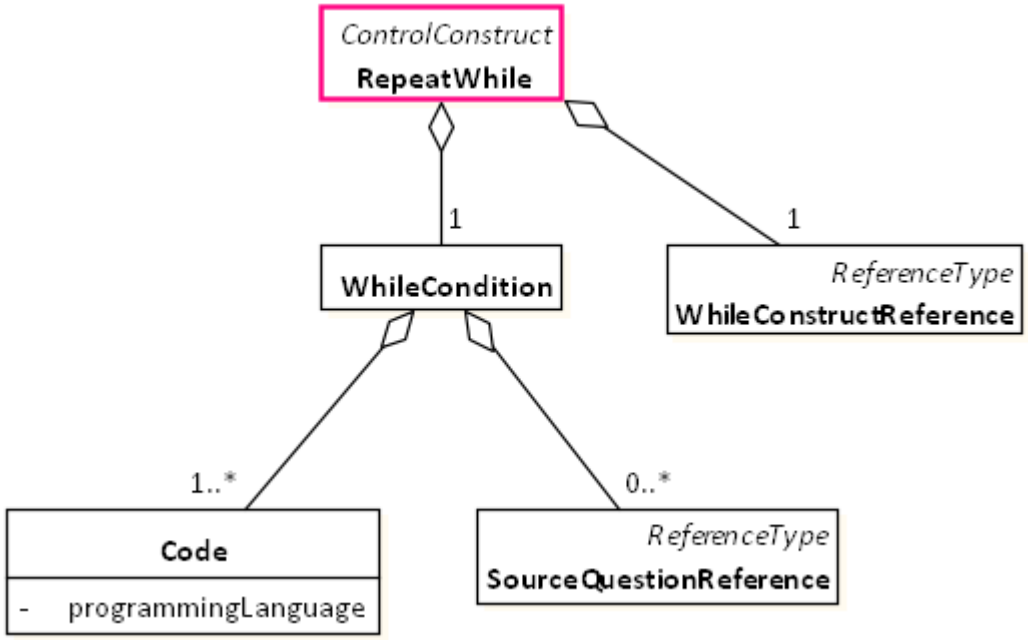
RepeatWhile (behavioral)



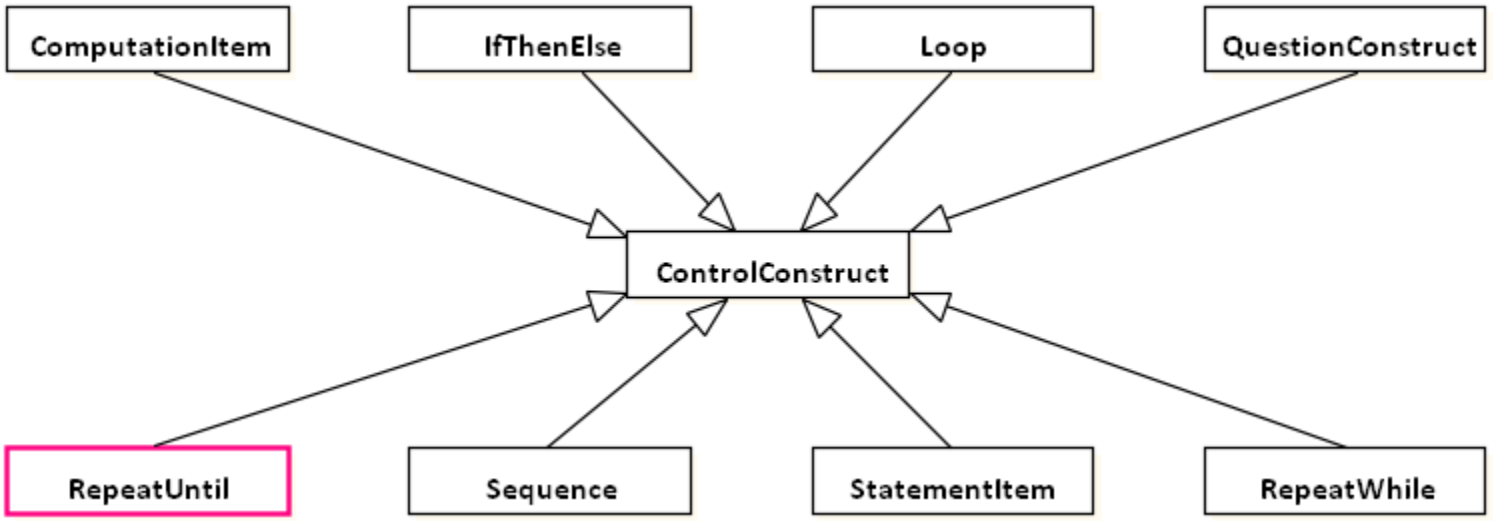
RepeatWhile (XML)

```
<RepeatWhile>
   <WhileCondition>
      <Code programmingLanguage="Neutral"> no </Code>
      <SourceQuestionReference> [Q1] </ SourceQuestionReference >
   </WhileCondition>
   <WhileConstructReference> [Q1] </WhileConstructReference>
</RepeatWhile>
```

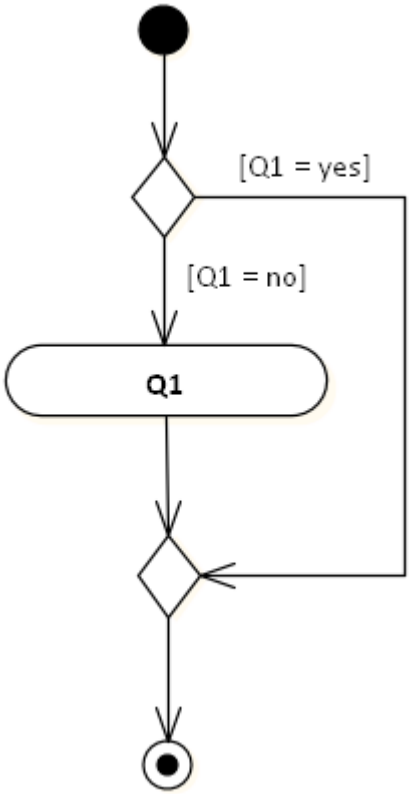
RepeatWhile (structural)



RepeatUntil



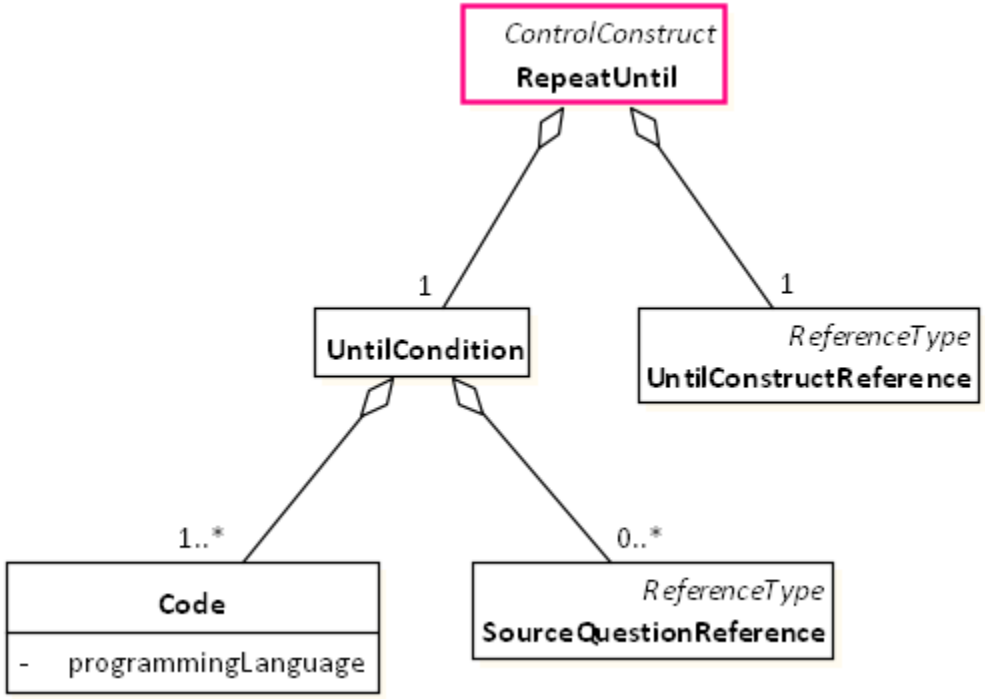
RepeatUntil (behavioral)



RepeatUntil (XML)

```
<RepeatUntil>
   <UntilCondition>
      <Code programmingLanguage="Neutral"> yes </Code>
      <SourceQuestionReference> [Q1] </SourceQuestionReference>
   </UntilCondition>
   <UntilConstructReference> [Q1] </UntilConstructReference>
</d:RepeatUntil>
```

RepeatUntil (structural)



What comes next?

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy

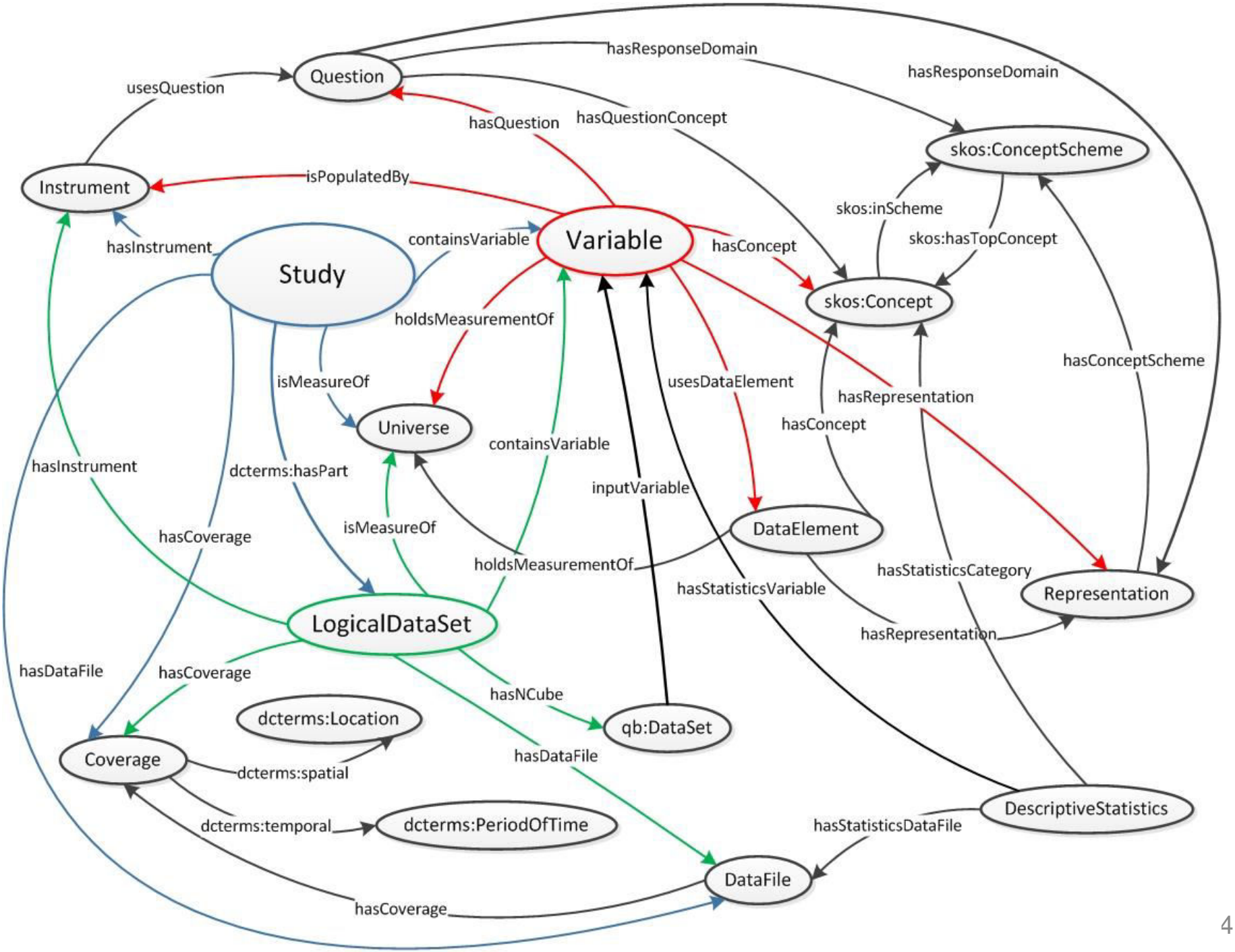
Presentation

- DDI-L Overview
- Identification, Versioning, Maintenance
- DDI-L Main Structures
- DDI Instance
- Study Unit

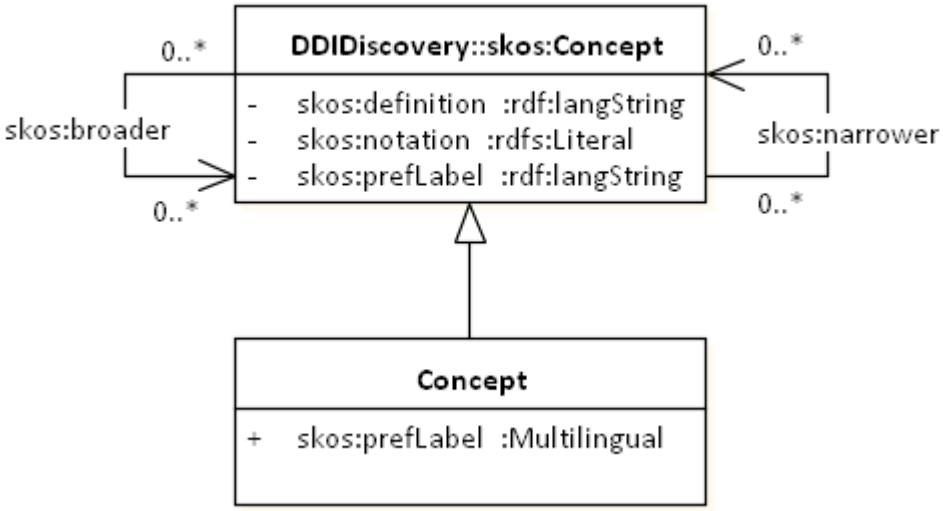
- Software Architecture
  - Multitier
  - MVC
- Missy Data Model
  - Extendable Data Model

- Conceptual Component
- Logical Products
- Data Collection
- **disco-model**

Business Layer


Question
hasResponseDomain
hasResponseDomain
usesQuestion
hasQuestion
hasQuestionConcept
skos:ConceptScheme
Instrument
isPopulatedBy
skos:inScheme
skos:hasTopConcept
hasInstrument
containsVariable
Variable
hasConcept
Study
skos:Concept
holdsMeasurementOf
hasConceptScheme
isMeasureOf
usesDataElement
hasRepresentation
Universe
hasConcept
containsVariable
hasInstrument
dcterms:hasPart
inputVariable
DataElement
isMeasureOf
hasCoverage
holdsMeasurementOf
hasStatisticsCategory
Representation
hasStatisticsVariable
LogicalDataSet
hasRepresentation
hasDataFile
hasCoverage
hasNCube
dcterms:Location
qb:DataSet
Coverage
dcterms:spatial
hasDataFile
DescriptiveStatistics
dcterms:temporal
dcterms:PeriodOfTime
hasStatisticsDataFile
DataFile
hasCoverage

How to extend the disco model?



What we have seen so far

- An introduction to the core DDI modules
- A short introduction to the disco-model, which covers the most important parts of DDI-Lifecycle and DDI-Codebook

How the presentation is continued

- Now that we know the data model…
- How can it be used for the Missy use-case, Matthäus?
- Can we create a mapping of the fields in the disco-model and the fields in Missy?

What comes next?

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy

Presentation

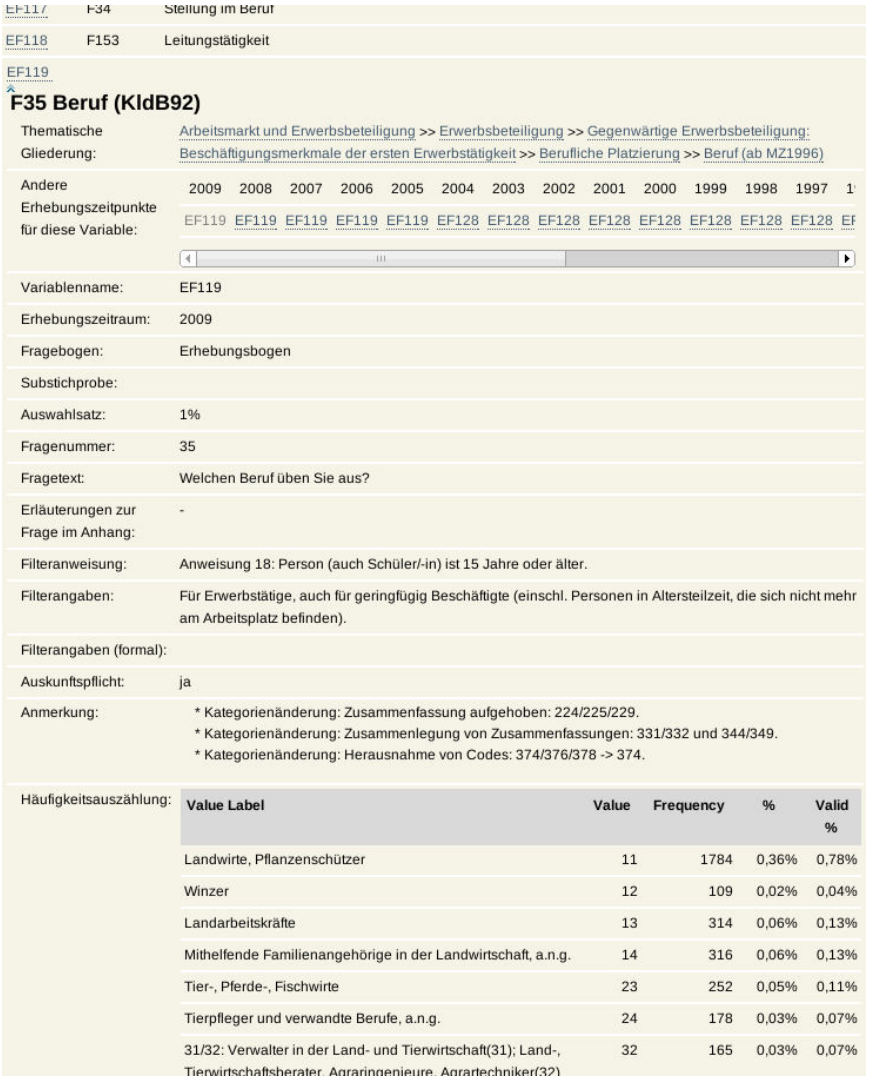
- DDI-L Overview
- Identification, Versioning, Maintenance
- DDI-L Main Structures
- DDI Instance
- Study Unit

- Software Architecture
  - Multitier
  - MVC
- **Missy Data Model**
  - Extendable Data Model

- Conceptual Component
- Logical Products
- Data Collection
- disco-model

Business Layer

Use-case “variable details”



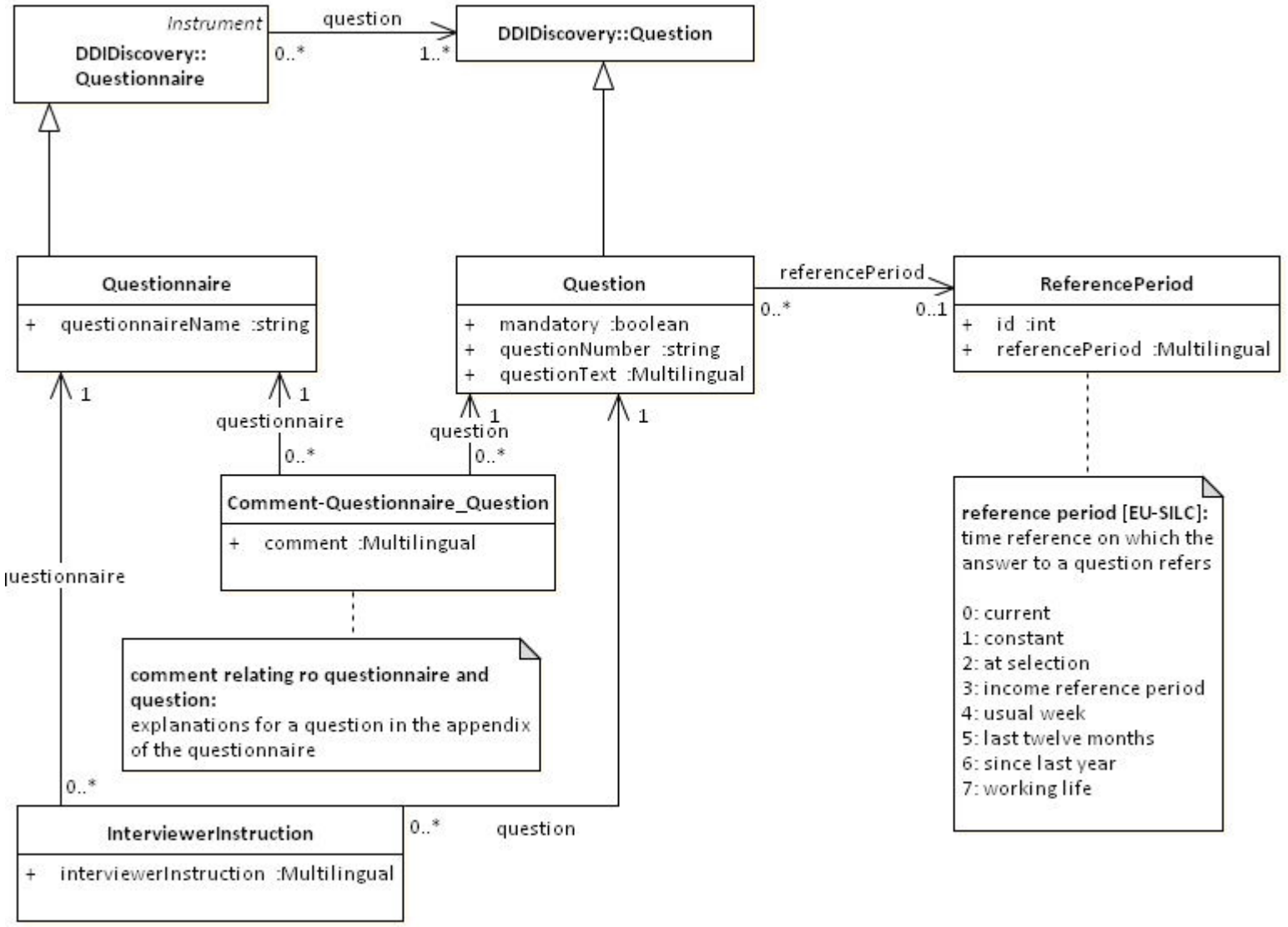
EF117	F34	Stellung im Beruf
EF118	F153	Leitungstätigkeit
EF119		

F35 Beruf (KldB92)

Thematische Gliederung:	Arbeitsmarkt und Erwerbsbeteiligung >> Erwerbsbeteiligung >> Gegenwärtige Erwerbsbeteiligung: Beschäftigungsmerkmale der ersten Erwerbstätigkeit >> Berufliche Platzierung >> Beruf (ab MZ1996)
Andere Erhebungszeitpunkte für diese Variable:	2009 2008 2007 2006 2005 2004 2003 2002 2001 2000 1999 1998 1997 1 EF119 EF119 EF119 EF119 EF119 EF128 EF128 EF128 EF128 EF128 EF128 EF128 EF128 EF
Variablenname:	EF119
Erhebungszeitraum:	2009
Fragebogen:	Erhebungsbogen
Substichprobe:	
Auswahlsatz:	1%
Fragenummer:	35
Fragetext:	Welchen Beruf üben Sie aus?
Erläuterungen zur Frage im Anhang:	-
Filteranweisung:	Anweisung 18: Person (auch Schüler/-in) ist 15 Jahre oder älter.
Filterangaben:	Für Erwerbstätige, auch für geringfügig Beschäftigte (einschl. Personen in Altersteilzeit, die sich nicht mehr am Arbeitsplatz befinden).
Filterangaben (formal):	
Auskunftspflicht:	ja
Anmerkung:	* Kategorienänderung: Zusammenfassung aufgehoben: 224/225/229. * Kategorienänderung: Zusammenlegung von Zusammenfassungen: 331/332 und 344/349. * Kategorienänderung: Herausnahme von Codes: 374/376/378 -> 374.

Häufigkeitsauszählung:

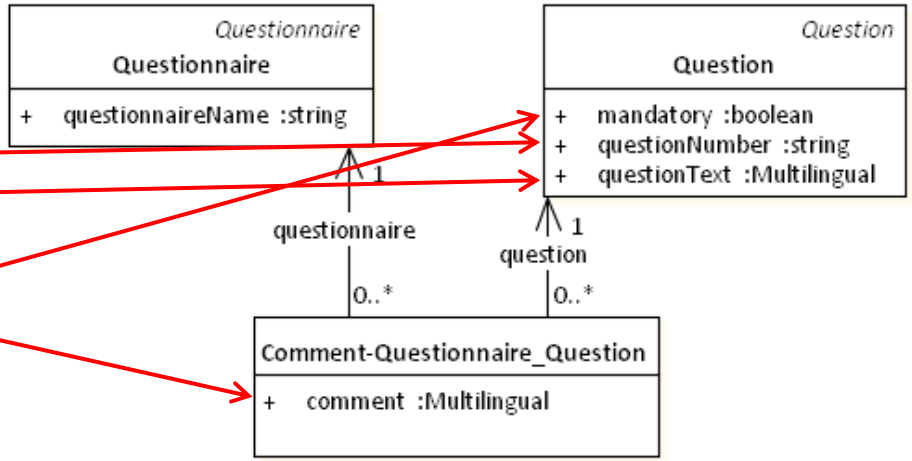
Value Label	Value	Frequency	%	Valid %
Landwirte, Pflanzenschützer	11	1784	0,36%	0,78%
Winzer	12	109	0,02%	0,04%
Landarbeitskräfte	13	314	0,06%	0,13%
Mithelfende Familienangehörige in der Landwirtschaft, a.n.g.	14	316	0,06%	0,13%
Tier-, Pferde-, Fischwirte	23	252	0,05%	0,11%
Tierpfleger und verwandte Berufe, a.n.g.	24	178	0,03%	0,07%
31/32: Verwalter in der Land- und Tierwirtschaft(31); Land-, Tierwirtschaftsberater, Agraringenieure, Agrartechniker(32)	32	165	0,03%	0,07%

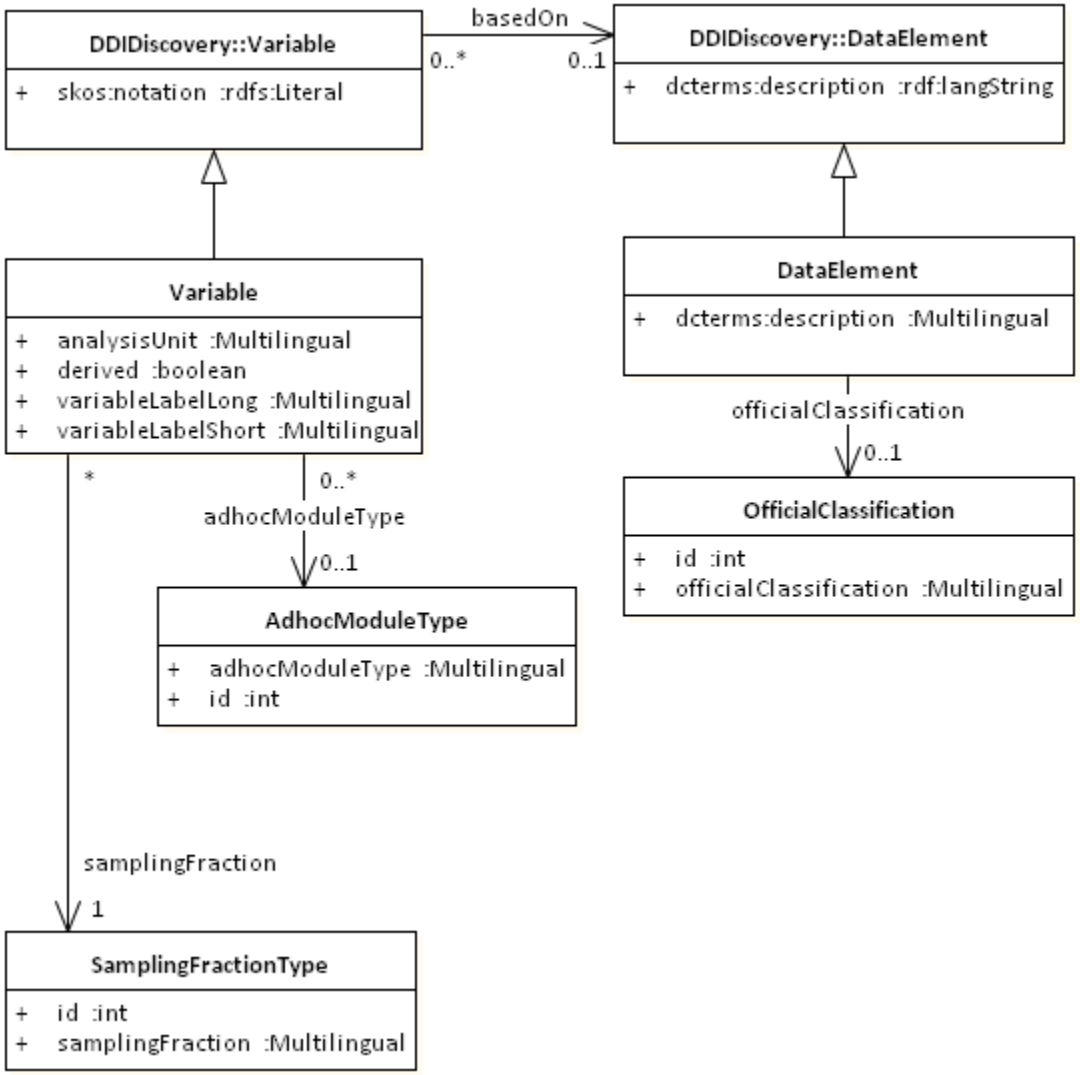

Instrument
DDIDiscovery::
Questionnaire
question
0..*
1..*
DDIDiscovery::Question
Questionnaire
+ questionnaireName :string
Question
+ mandatory :boolean
+ questionNumber :string
+ questionText :Multilingual
referencePeriod
0..*
0..1
ReferencePeriod
+ id :int
+ referencePeriod :Multilingual
1
1
questionnaire
0..*
1
question
0..*
1
Comment-Questionnaire_Question
+ comment :Multilingual
questionnaire
comment relating ro questionnaire and question:
explanations for a question in the appendix of the questionnaire
reference period [EU-SILC]:
time reference on which the answer to a question refers
0: current
1: constant
2: at selection
3: income reference period
4: usual week
5: last twelve months
6: since last year
7: working life
0..*
InterviewerInstruction
+ interviewerInstruction :Multilingual
0..*
question

Andere Erhebungszeitpunkte für diese Variable:	2009 2008 2007 2006 2005 2004 2003 2002 2001 2000 1999 1998 EF119 EF119 EF119 EF119 EF119 EF128 EF128 EF128 EF128 EF128 EF128 EF128
Variablenname:	EF119
Erhebungszeitraum:	2009
Fragebogen:	Erhebungsbogen
Substichprobe:	
Auswahlsatz:	1%
Fragenummer:	35
Fragetext:	Welchen Beruf üben Sie aus?
Erläuterungen zur Frage im Anhang:	-
Filteranweisung:	Anweisung 18: Person (auch Schüler/-in) ist 15 Jahre oder älter.
Filterangaben:	Für Erwerbstätige, auch für geringfügig Beschäftigte (einschl. Personen in Altersteilzeit, die sich am Arbeitsplatz befinden).
Filterangaben (formal):	
Auskunftspflicht:	ja
Anmerkung:	* Kategorienänderung: Zusammenfassung aufgehoben: 224/225/229. * Kategorienänderung: Zusammenlegung von Zusammenfassungen: 331/332 und 344/349. * Kategorienänderung: Herausnahme von Codes: 374/376/378 -> 374.

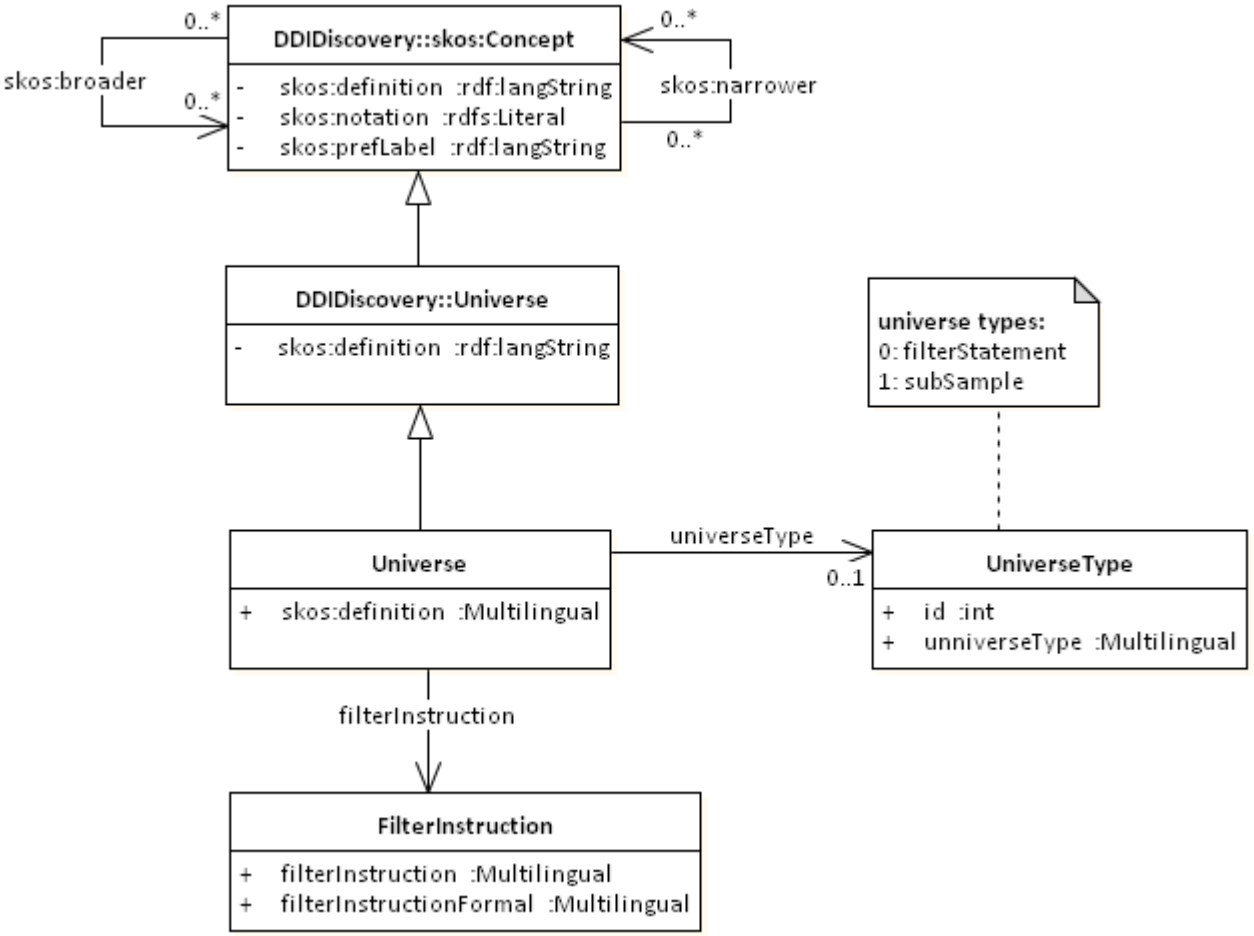
Häufigkeitsauszählung:

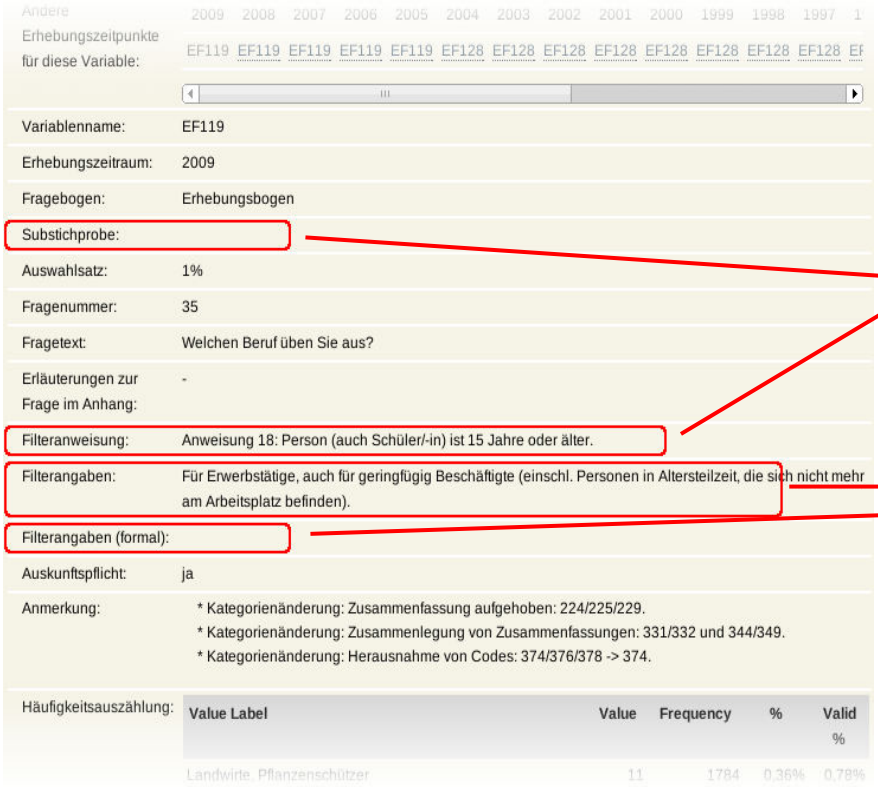
Value Label	Value	Frequency	%
Landwirte, Pflanzenschützer	11	1784	0.3

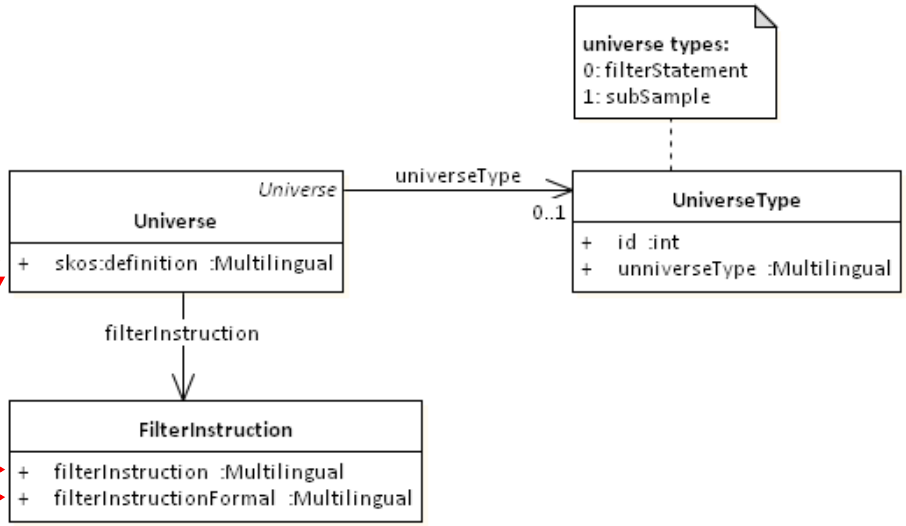



DDIDiscovery::Variable
+ skos:notation :rdfs:Literal
basedOn
0..*
0..1
DDIDiscovery::DataElement
+ dcterms:description :rdf:langString
Variable
+ analysisUnit :Multilingual
+ derived :boolean
+ variableLabelLong :Multilingual
+ variableLabelShort :Multilingual
DataElement
+ dcterms:description :Multilingual
officialClassification
0..1
OfficialClassification
+ id :int
+ officialClassification :Multilingual
*
0..*
adhocModuleType
0..1
AdhocModuleType
+ adhocModuleType :Multilingual
+ id :int
samplingFraction
1
SamplingFractionType
+ id :int
+ samplingFraction :Multilingual


Gliederung:
Beschäftigungsmerkmale der ersten Erwerbstätigkeit >> Berufliche Platzierung >>
Andere Erhebungszeitpunkte für diese Variable:
2009 2008 2007 2006 2005 2004 2003 2002 2001 2000
EF119 EF119 EF119 EF119 EF119 EF128 EF128 EF128 EF128 EF128
Variablenname: EF119
Erhebungszeitraum: 2009
Fragebogen: Erhebungsbogen
Substichprobe:
Auswahlsatz: 1%
Fragenummer: 35
Fragetext: Welchen Beruf üben Sie aus?
Erläuterungen zur Frage im Anhang: -
Filteranweisung: Anweisung 18: Person (auch Schüler/in) ist 15 Jahre oder älter
DDIDiscovery::Variable
+ skos:notation :rdfs:Literal
Variable
+ analysisUnit :Multilingual
+ derived :boolean
+ variableLabelLong :Multilingual
+ variableLabelShort :Multilingual
*
samplingFraction
1
SamplingFractionType
+ id :int
+ samplingFraction :Multilingual


DDIDiscovery::skos:Concept
- skos:definition :rdf:langString
- skos:notation :rdfs:Literal
- skos:prefLabel :rdf:langString
0..*
skos:broader
0..*
0..*
skos:narrower
0..*
DDIDiscovery::Universe
- skos:definition :rdf:langString
universe types:
0: filterStatement
1: subSample
Universe
+ skos:definition :Multilingual
universeType
0..1
UniverseType
+ id :int
+ unniverseType :Multilingual
filterInstruction
FilterInstruction
+ filterInstruction :Multilingual
+ filterInstructionFormal :Multilingual



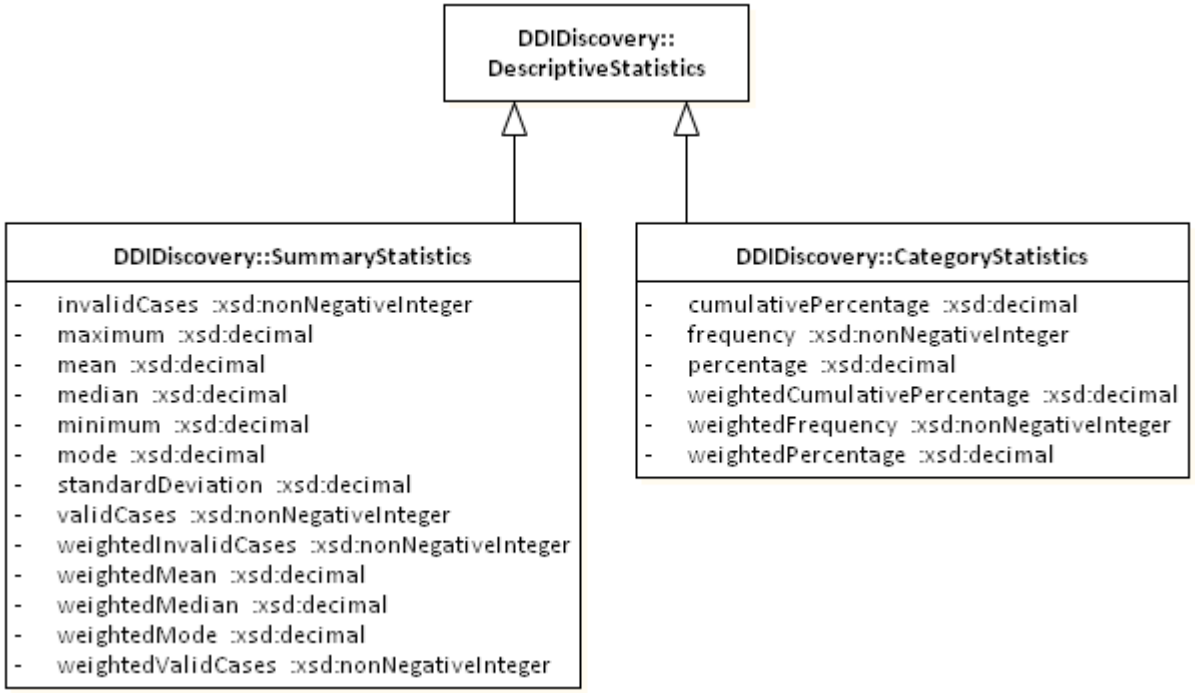


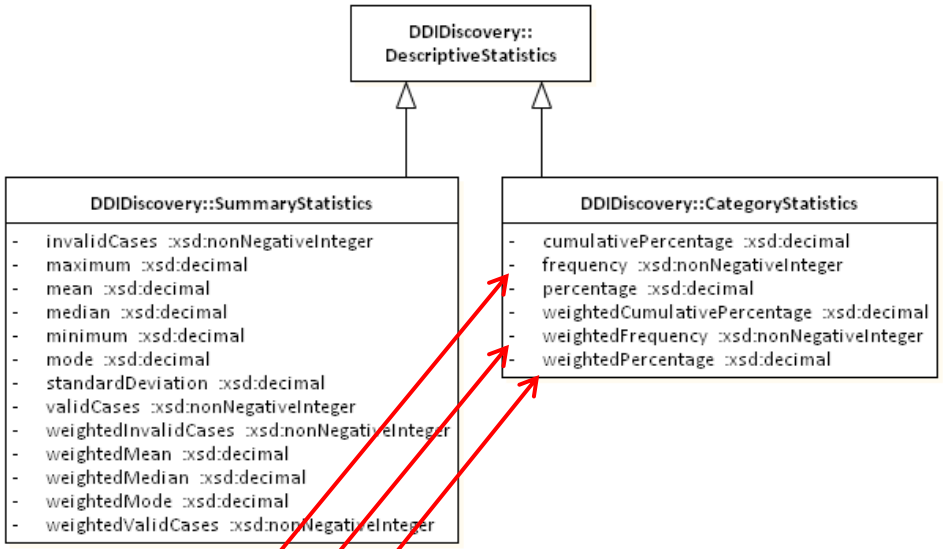
Sub-Sample:

- is-a Sub-Universe with the UniverseType „subSample“

Filter Statement

- is-a Sub-Universe with the UniverseType „filterStatement“

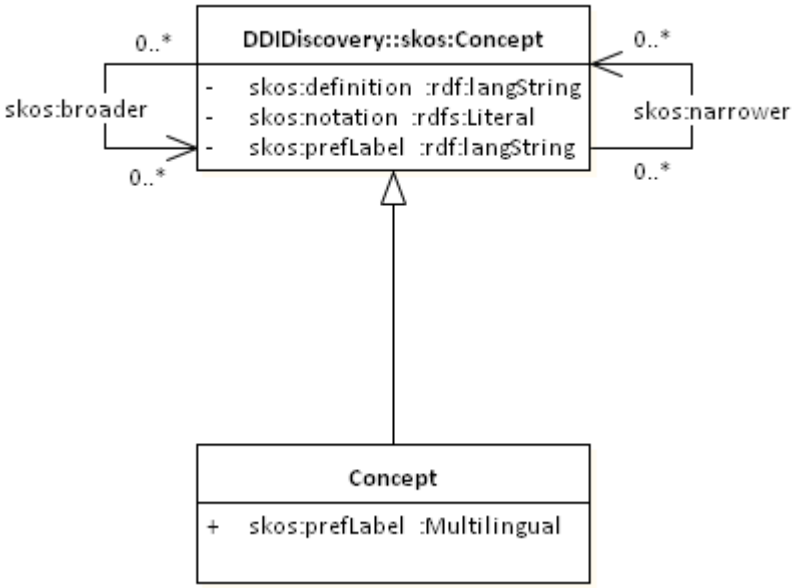

DDIDiscovery::
DescriptiveStatistics
DDIDiscovery::SummaryStatistics
- invalidCases :xsd:nonNegativeInteger
- maximum :xsd:decimal
- mean :xsd:decimal
- median :xsd:decimal
- minimum :xsd:decimal
- mode :xsd:decimal
- standardDeviation :xsd:decimal
- validCases :xsd:nonNegativeInteger
- weightedInvalidCases :xsd:nonNegativeInteger
- weightedMean :xsd:decimal
- weightedMedian :xsd:decimal
- weightedMode :xsd:decimal
- weightedValidCases :xsd:nonNegativeInteger
DDIDiscovery::CategoryStatistics
- cumulativePercentage :xsd:decimal
- frequency :xsd:nonNegativeInteger
- percentage :xsd:decimal
- weightedCumulativePercentage :xsd:decimal
- weightedFrequency :xsd:nonNegativeInteger
- weightedPercentage :xsd:decimal



Erläuterungen zur Frage im Anhang:	-
Filteranweisung:	Anweisung 18: Person (auch Schüler/-in) ist 15 Jahre oder älter.
Filterangaben:	Für Erwerbstätige, auch für geringfügig Beschäftigte (einschl. Personen in Altersteilzeit, die sich nicht mehr am Arbeitsplatz befinden).
Filterangaben (formal):	
Auskunftspflicht:	ja
Anmerkung:	* Kategorienänderung: Zusammenfassung aufgehoben: 224/225/229. * Kategorienänderung: Zusammenlegung von Zusammenfassungen: 331/332 und 344/349. * Kategorienänderung: Herausnahme von Codes: 374/376/378 -> 374.

Häufigkeitsauszählung:

Value Label	Value	Frequency	%	Valid %
Landwirte, Pflanzenschützer	11	1784	0,36%	0,78%
Winzer	12	109	0,02%	0,04%
Landarbeitskräfte	13	314	0,06%	0,13%
Mithelfende Familienangehörige in der Landwirtschaft, a.n.g.	14	316	0,06%	0,13%
Tier-, Pferde-, Fischwirte	23	252	0,05%	0,11%
Tierpfleger und verwandte Berufe, a.n.g.	24	178	0,03%	0,07%
31/32: Verwalter in der Land- und Tierwirtschaft/31: Land-	32	165	0,03%	0,07%


DDIDiscovery::skos:Concept
- skos:definition :rdf:langString
- skos:notation :rdfs:Literal
- skos:prefLabel :rdf:langString
0..*
skos:broader
0..*
0..*
skos:narrower
0..*
Concept
+ skos:prefLabel :Multilingual

EF118 F153 Leitungstätigkeit

EF119

F35 Beruf (KldB92)

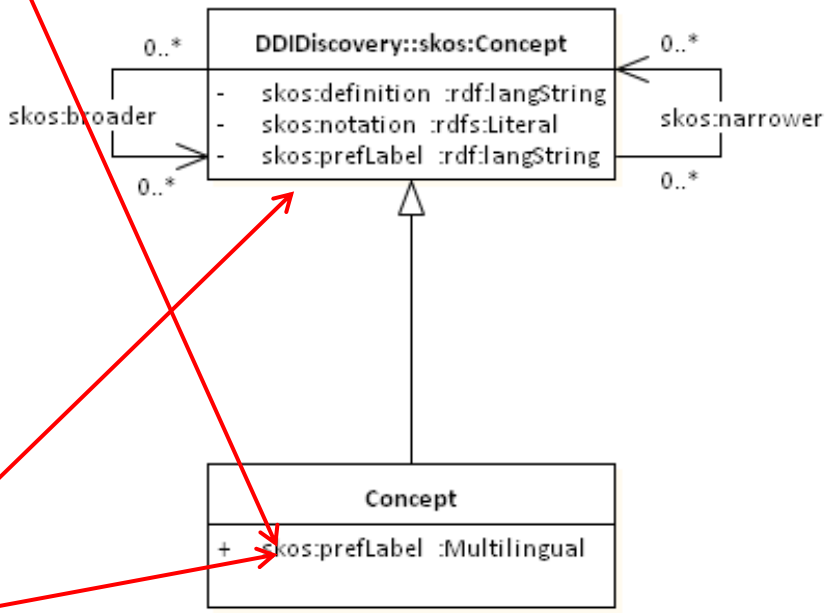
Thematische Gliederung:	Arbeitsmarkt und Erwerbsbeteiligung >> Erwerbsbeteiligung >> Gegenwärtige Erwerbsbeteiligung: Beschäftigungsmerkmale der ersten Erwerbstätigkeit >> Berufliche Platzierung >> Beruf (ab MZ1996)

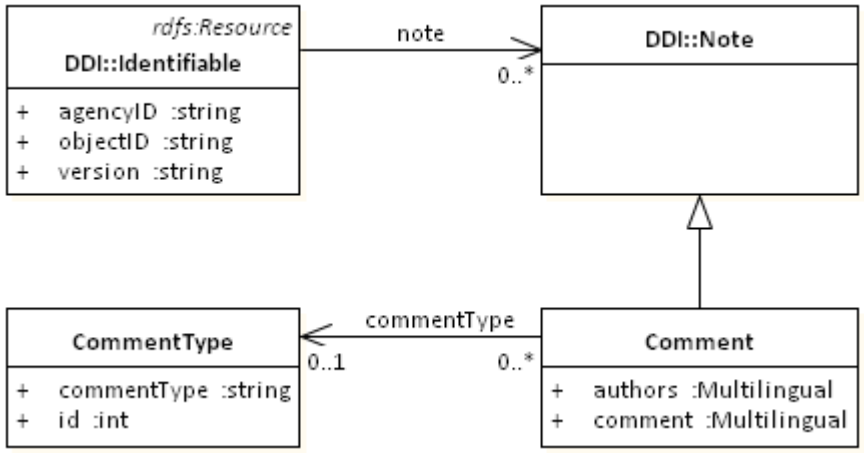
Andere Erhebungszeitpunkte für diese Variable:	2009	2008	2007	2006	2005	2004	2003	2002	2001	2000	1999	1998	1997	1
	EF119	EF119	EF119	EF119	EF119	EF128	EF128	EF128	EF128	EF128	EF128	EF128	EF128	E

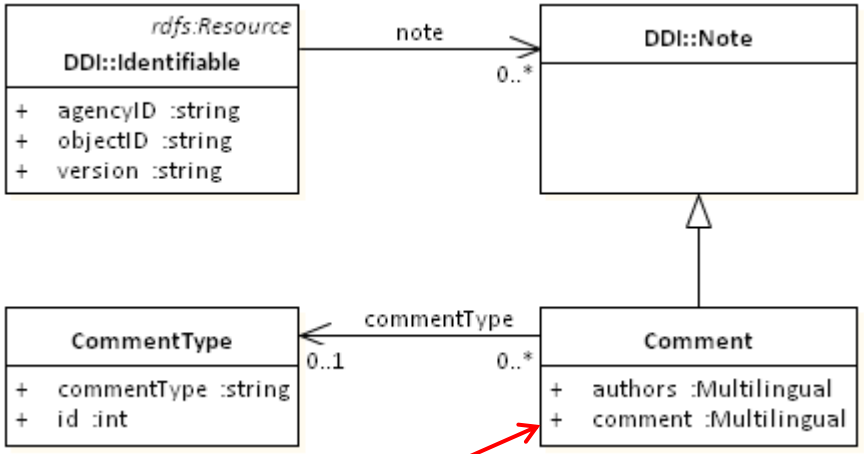
Variablenname:	EF119
Erhebungszeitraum:	2009
Fragebogen:	Erhebungsbogen
Substichprobe:	
Auswahlsatz:	1%
Fragenummer:	35
Fragetext:	Welchen Beruf üben Sie aus?
Erläuterungen zur Frage im Anhang:	-
Filteranweisung:	Anweisung 18: Person (auch Schüler/-in) ist 15 Jahre oder älter.
Filterangaben:	Für Erwerbstätige, auch für geringfügig Beschäftigte (einschl. Personen in Altersteilzeit, die sich nicht meh am Arbeitsplatz befinden).
Filterangaben (formal):	
Auskunftspflicht:	ja
Anmerkung:	* Kategorienänderung: Zusammenfassung aufgehoben: 224/225/229. * Kategorienänderung: Zusammenlegung von Zusammenfassungen: 331/332 und 344/349. * Kategorienänderung: Herausnahme von Codes: 374/376/378 -> 374.

Häufigkeitsauszählung:

Value Label	Value	Frequency	%	Valid %
Landwirte, Pflanzenschützer	11	1784	0,36%	0,78%
Winzer	12	109	0,02%	0,04%
Landarbeitskräfte	13	314	0,06%	0,13%
Mithelfende Familienangehörige in der Landwirtschaft, a.n.g.	14	316	0,06%	0,13%
Tier-, Pferde-, Fischwirte	23	252	0,05%	0,11%



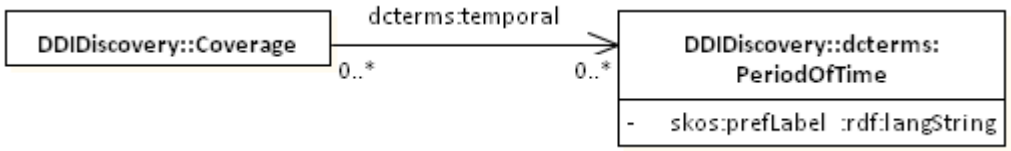

rdfs:Resource
DDI::Identifiable
+ agencyID :string
+ objectID :string
+ version :string
note
0..*
DDI::Note
CommentType
+ commentType :string
+ id :int
commentType
0..1
0..*
Comment
+ authors :Multilingual
+ comment :Multilingual



Fragenummer:	35
Fragetext:	Welchen Beruf üben Sie aus?
Erläuterungen zur Frage im Anhang:	-
Filteranweisung:	Anweisung 18: Person (auch Schüler/-in) ist 15 Jahre oder älter.
Filterangaben:	Für Erwerbstätige, auch für geringfügig Beschäftigte (einschl. Personen in Altersteilzeit, die sich n am Arbeitsplatz befinden).
Filterangaben (formal):	
Auskunftspflicht:	ja
Anmerkung:	* Kategorienänderung: Zusammenfassung aufgehoben: 224/225/229. * Kategorienänderung: Zusammenlegung von Zusammenfassungen: 331/332 und 344/349. * Kategorienänderung: Herausnahme von Codes: 374/376/378 -> 374.

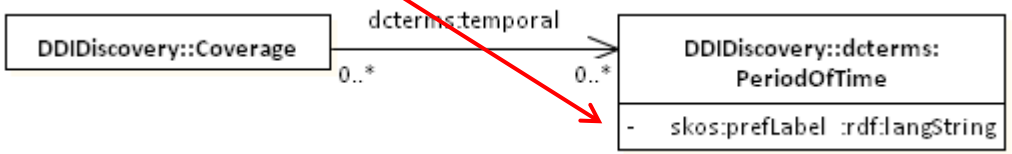
Häufigkeitsauszählung:

Value Label	Value	Frequency	%
Landwirte, Pflanzenschützer	11	1784	0,36%
Winzer	12	109	0,02%


DDIDiscovery::Coverage
dcterms:temporal
0..*
0..*
DDIDiscovery::dcterms:
PeriodOfTime
- skos:prefLabel :rdf:langString

F35 Beruf (KldB92)

Thematische Gliederung:	Arbeitsmarkt und Erwerbsbeteiligung >> Erwerbsbeteiligung >> Gegenwärtige Erwerbsbeteiligung: Beschäftigungsmerkmale der ersten Erwerbstätigkeit >> Berufliche Platzierung >> Beruf (ab MZ1996)
Andere Erhebungszeitpunkte für diese Variable:	2009 2008 2007 2006 2005 2004 2003 2002 2001 2000 1999 1998 1997 1 EF119 EF119 EF119 EF119 EF119 EF128 EF128 EF128 EF128 EF128 EF128 EF128 EF128 EF
Variablenname:	EF119
Erhebungszeitraum:	2009
Fragebogen:	Erhebungsbogen
Substichprobe:	
Auswahlsatz:	1%
Fragenummer:	35
Fragetext:	Welchen Beruf üben Sie aus?
Erläuterungen zur Frage im Anhang:	-
Filteranweisung:	Anweisung 18: Person (auch Schüler/-in) ist 15 Jahre oder älter.
Filterangaben:	Für Erwerbstätige, auch für geringfügig Beschäftigte (einschl. Personen in Altersteilzeit; die sich nicht mehr



What we have seen so far

- disco-model
  - designed for the discovery use-case
  - provides object types, properties and data type properties designed for discovery use-case
- One Missy use-case and all the data type and object properties that are covered
- The idea of how theoretically the disco can be extended

How the presentation is continued

- But, what if a project has specific requirements to the model, that are not covered by the disco model?
- How are we going to implement that?

What comes next?

- Missy – General Information
  - Requirements to Developers
  - Use-Cases in Missy

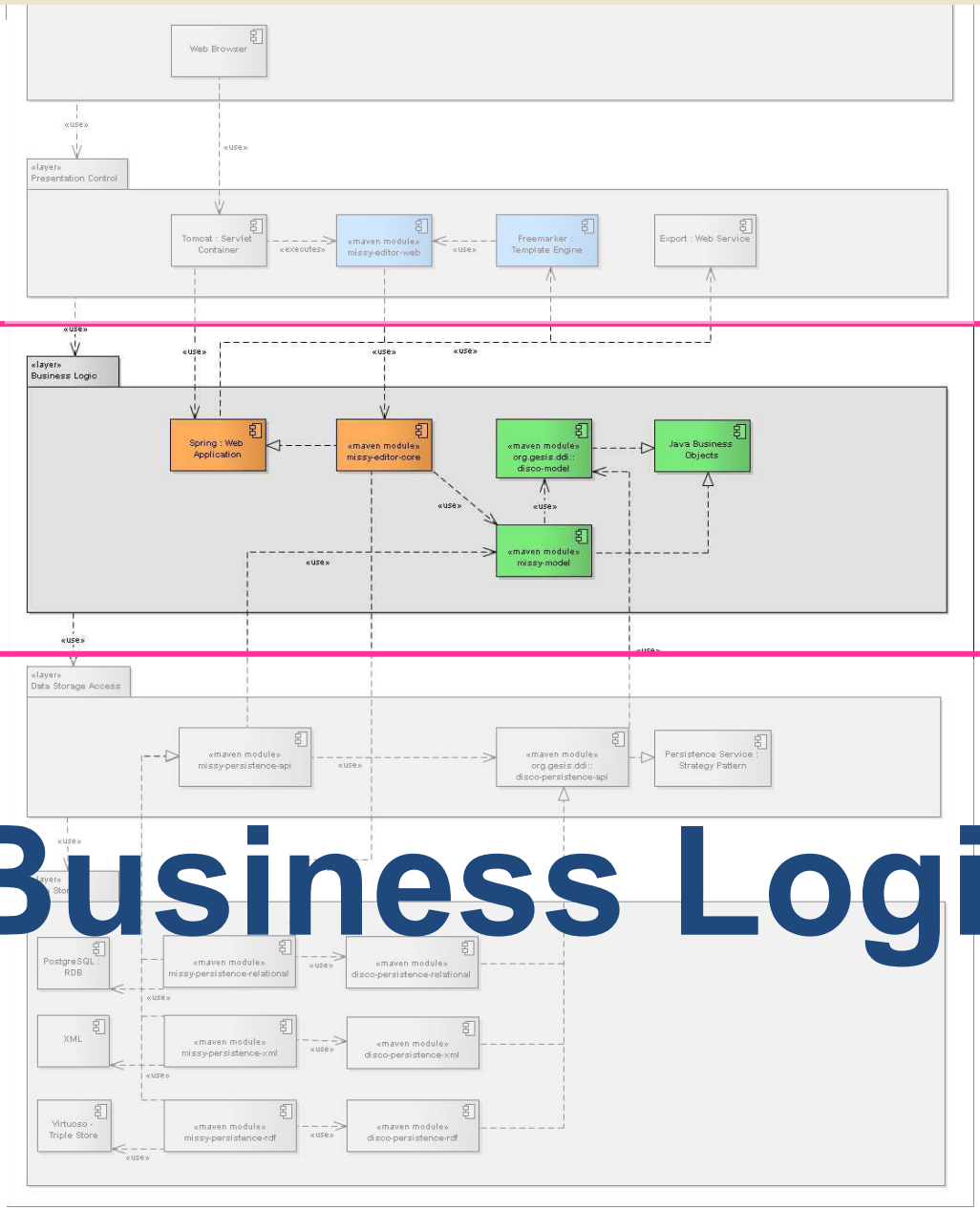
Presentation

- DDI-L Overview
- Identification, Versioning, Maintenance
- DDI-L Main Structures
- DDI Instance
- Study Unit

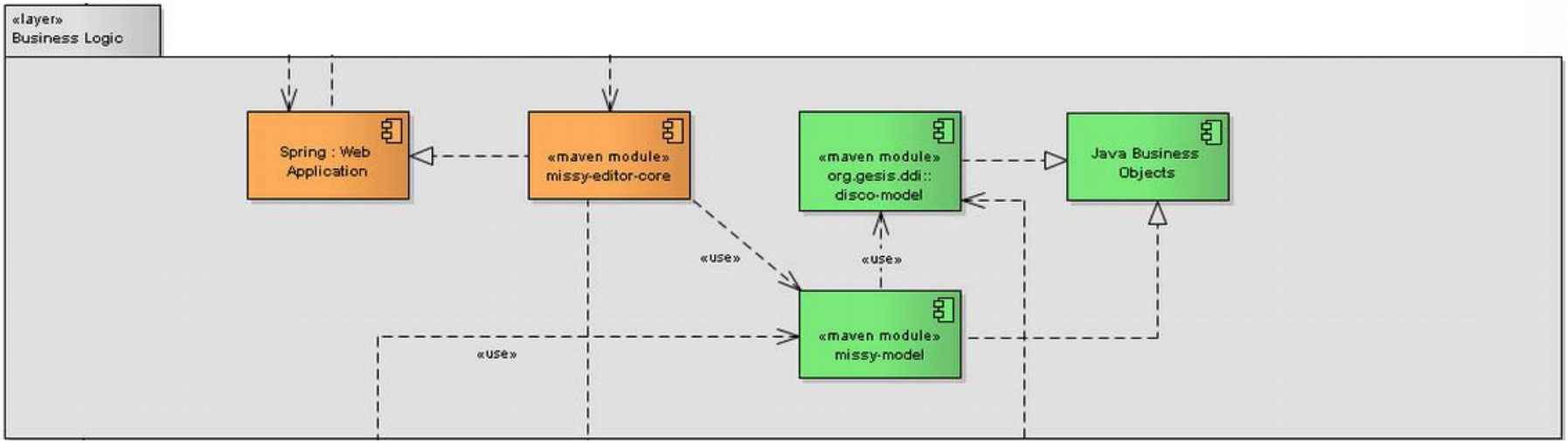
- Software Architecture
  - Multitier
  - MVC
- Missy Data Model
  - **Extendable Data Model**

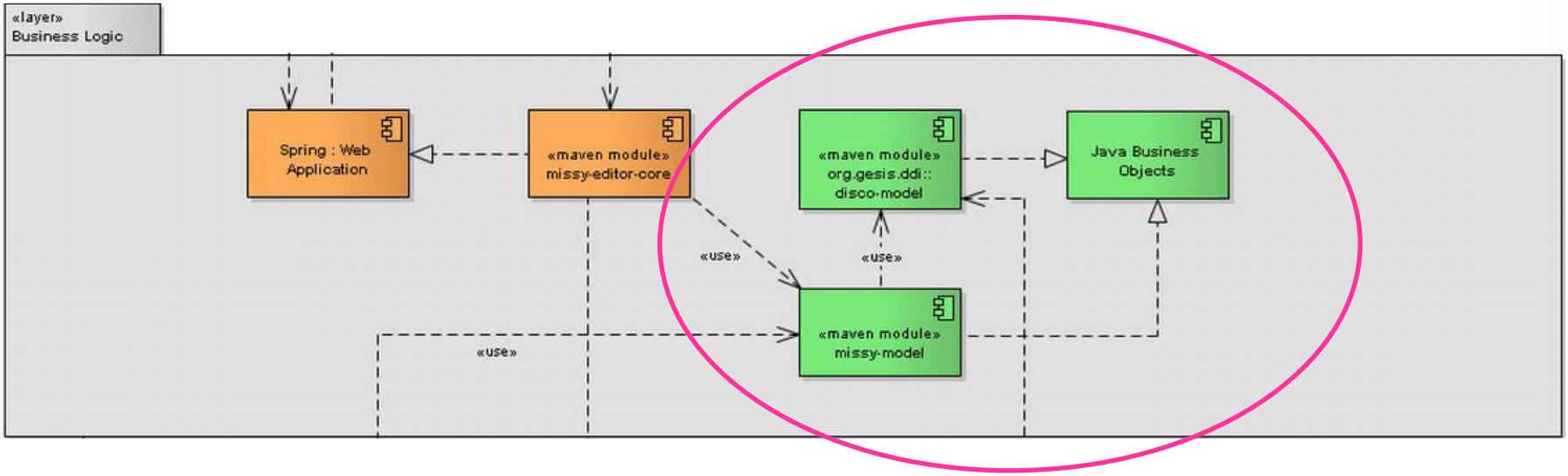
- Conceptual Component
- Logical Products
- Data Collection
- disco-model

Business Layer



Business Logic


«layer»
Business Logic
Spring : Web Application
«maven module» missy-editor-core
«maven module» org.gesis.ddi:: disco-model
Java Business Objects
«use»
«use»
«maven module» missy-model
«use»


«layer»
Business Logic
Spring : Web Application
«maven module» missy-editor-core
«maven module» org.gesis.ddi:: disco-model
Java Business Objects
«use»
«use»
«maven module» missy-model
«use»

disco-model API

- The disco-model implementation is designed to be extended
- For one project's individual requirements

- Speaking in MVC-pattern language: the model is separated and independent from the views and the controllers

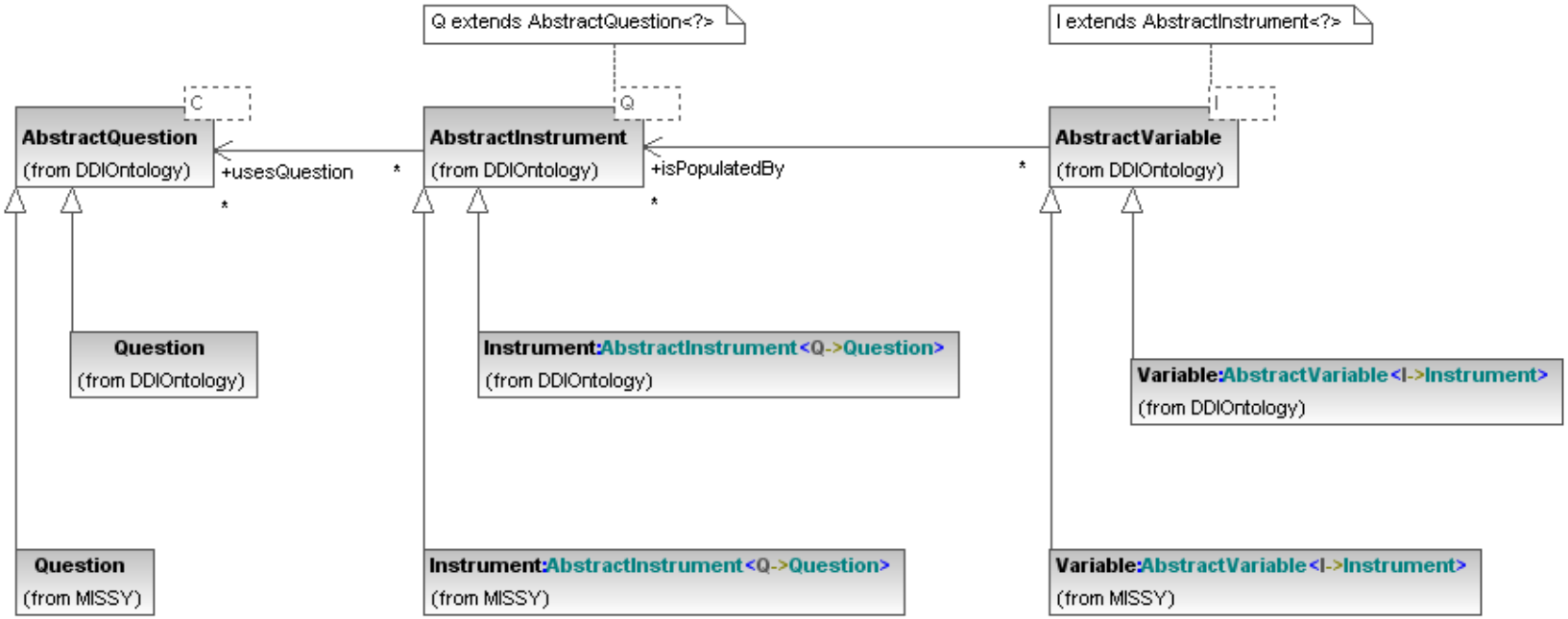
disco-model API

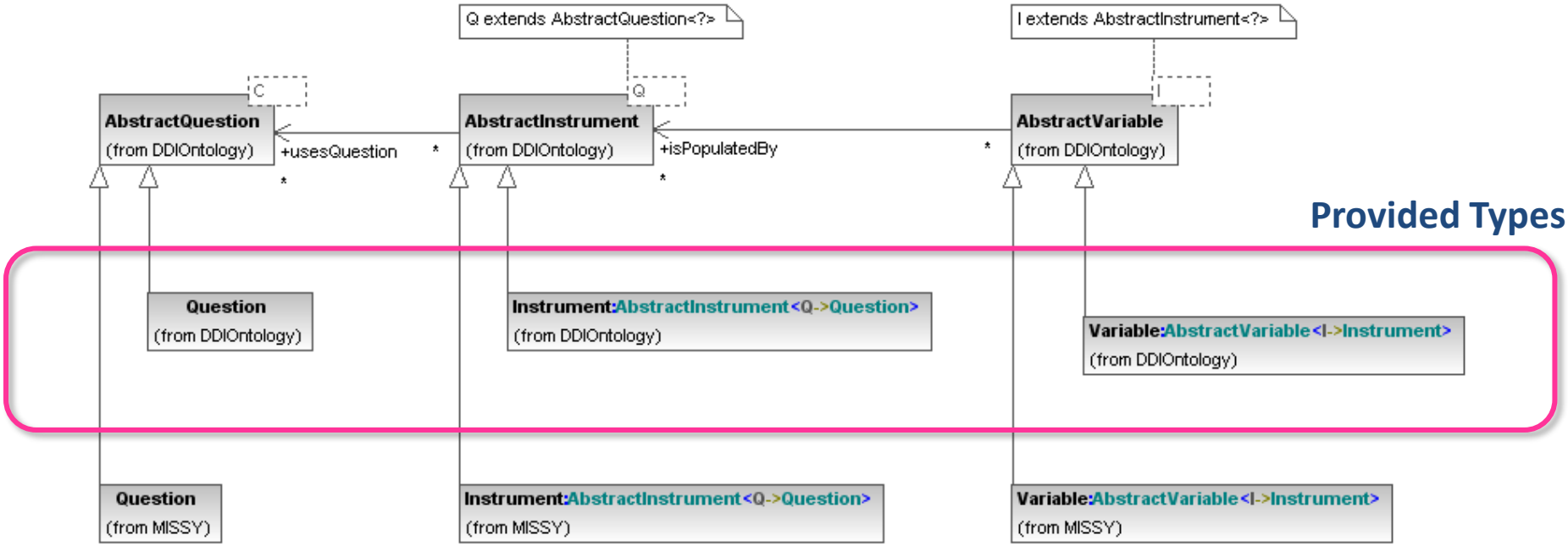
- abstract classes
  - define the basic elements, i.e. data type and object properties, that are provided and covered by the disco-model
  - may be extended, but cannot be instantiated
  - e.g. AbstractVariable
- model classes
  - represent the actual entity types of disco-model
  - may be instantiated and reused
  - e.g. Variable

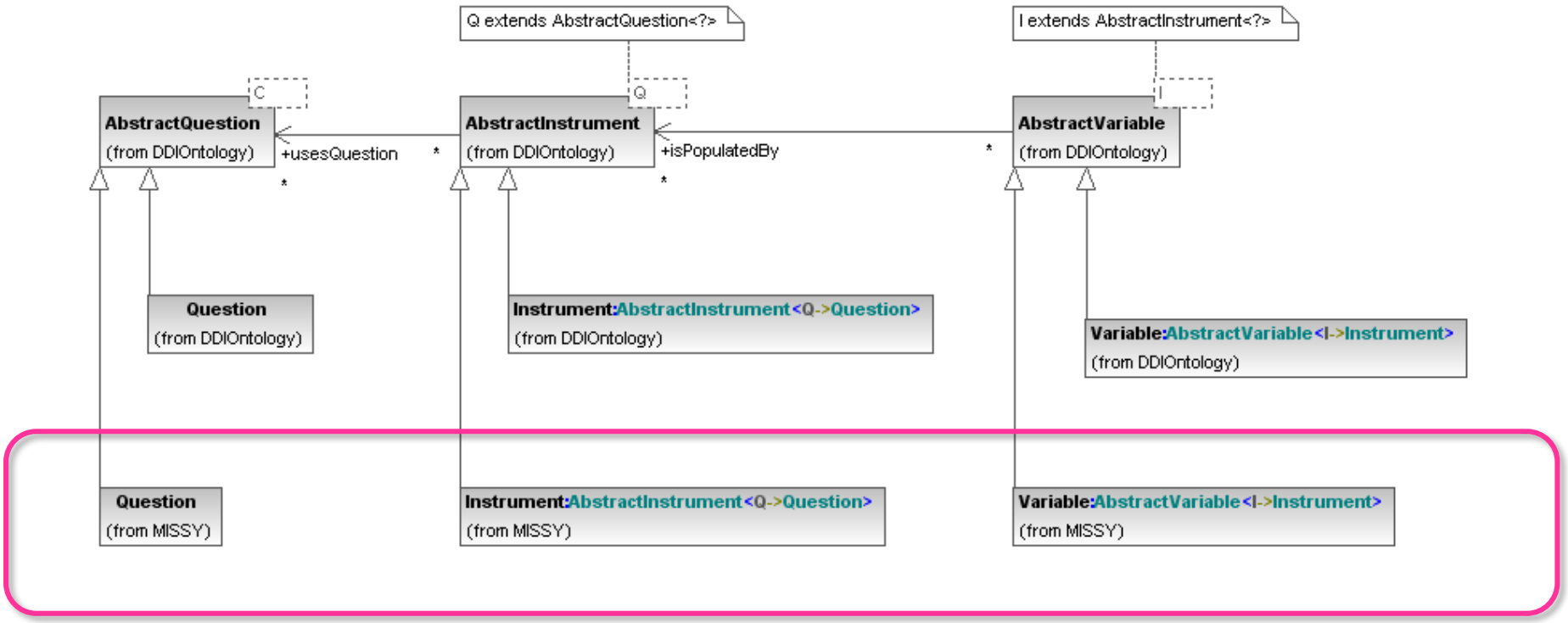
disco-model API

- abstract classes
  - define the basic elements, i.e. data type and object properties, that are provided and covered by the disco-model
  - may be extended, but cannot be instantiated
  - e.g. AbstractVariable
- model classes
  - represent the actual entity types of disco-model
  - may be instantiated and reused
  - e.g. Variable

- Implementation of the **Template Pattern**


Q extends AbstractQuestion<?>
I extends AbstractInstrument<?>
C
Q
I
AbstractQuestion
(from DDIOntology)
+usesQuestion
*
*
AbstractInstrument
(from DDIOntology)
+isPopulatedBy
*
*
AbstractVariable
(from DDIOntology)
Question
(from DDIOntology)
Instrument:AbstractInstrument<Q->Question>
(from DDIOntology)
Variable:AbstractVariable<I->Instrument>
(from DDIOntology)
Question
(from MISSY)
Instrument:AbstractInstrument<Q->Question>
(from MISSY)
Variable:AbstractVariable<I->Instrument>
(from MISSY)


Q extends AbstractQuestion<?>
I extends AbstractInstrument<?>
C
Q
I
AbstractQuestion
(from DDIOntology)
+usesQuestion
*
*
AbstractInstrument
(from DDIOntology)
+isPopulatedBy
*
*
AbstractVariable
(from DDIOntology)
Provided Types
Question
(from DDIOntology)
Instrument:AbstractInstrument<Q->Question>
(from DDIOntology)
Variable:AbstractVariable<I->Instrument>
(from DDIOntology)
Question
(from MISSY)
Instrument:AbstractInstrument<Q->Question>
(from MISSY)
Variable:AbstractVariable<I->Instrument>
(from MISSY)



Project specific extensions

Project Specific Extensions

- Projects may just simply use
- and extend the defined classes
  - Introduce new fields and methods, such as getter/setter
  - Change the type of existing fields when they overwrite the return type (in Java this is only possible in case of covariant returns)

How is the presentation continued

- Now that I have the data model… and I can extend it
- I just need to store my data somewhere…

- Are there any persistence mechanisms that are offered by the DDI-Lifecycle model?

Outline

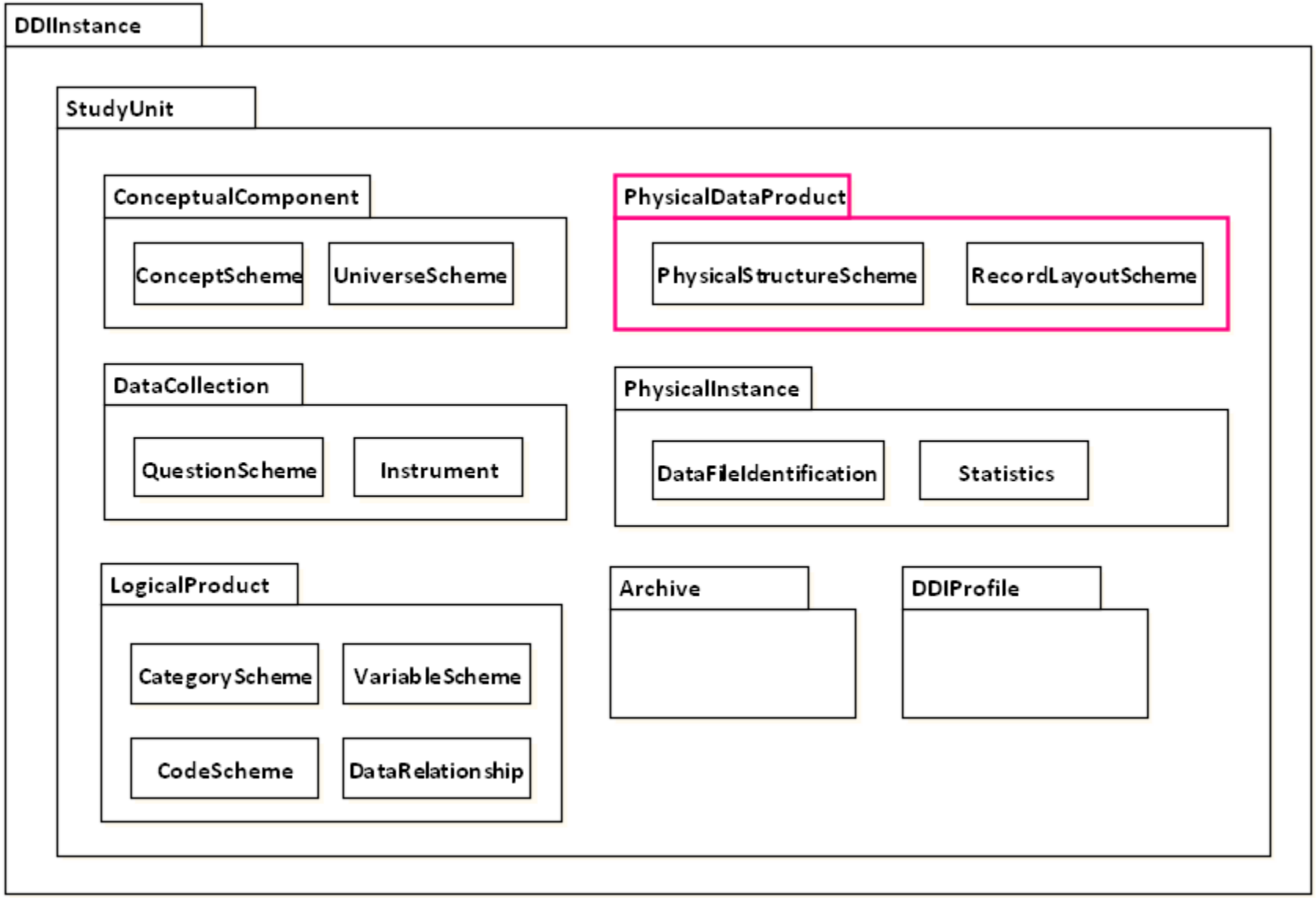
- Persistence Layer
  - Programming Interface
  - Types of physical data storages
  - Examples
- **Physical Data Product**
- Physical Instance
- Archive
- DDIProfile

Data Storage Access

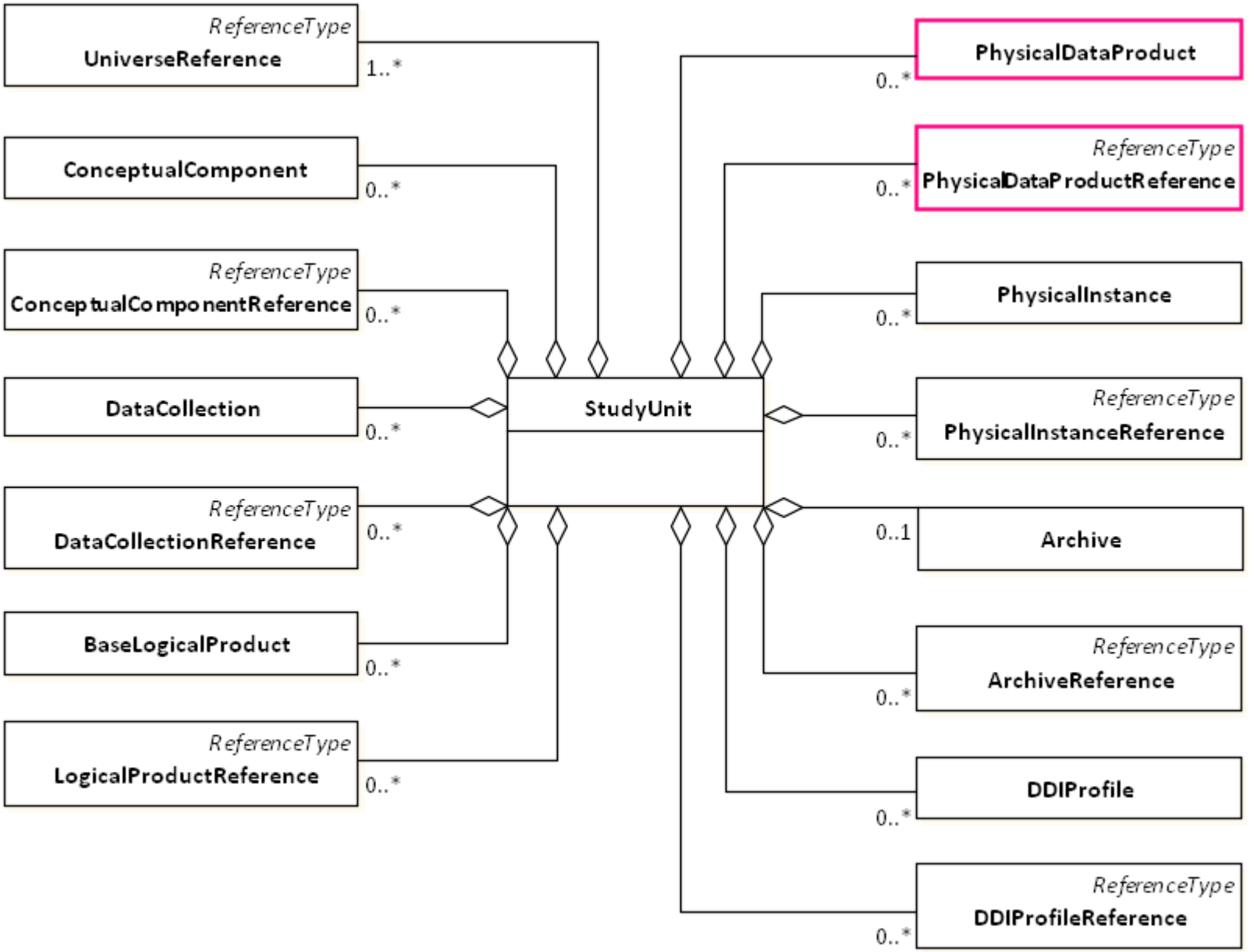
- Outlook
- DDI Serialization Examples
  - DDI-XML
  - DDI-RDF / disco-spec
  - Relational-DB

Data Storage

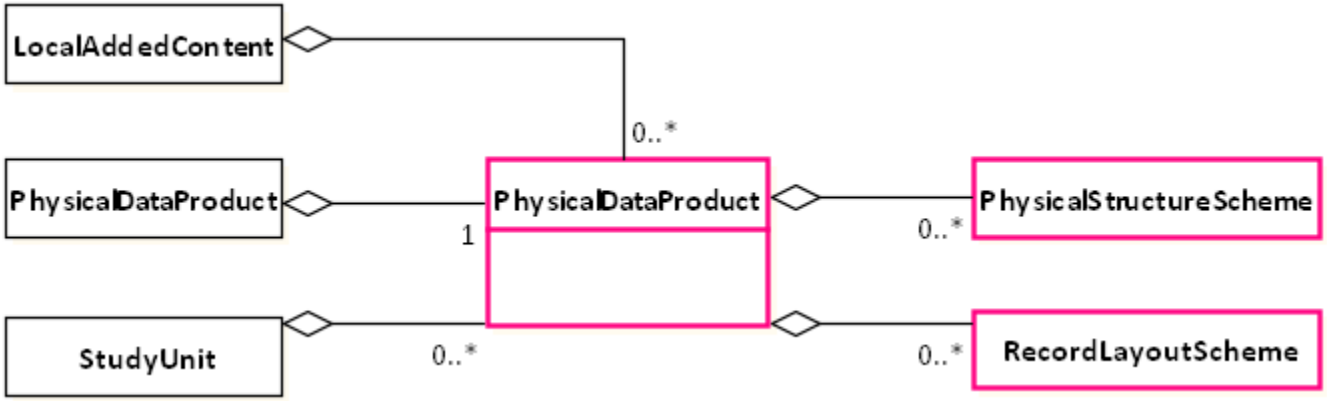
PhysicalDataProduct



PhysicalDataProduct



PhysicalDataProduct



Outline

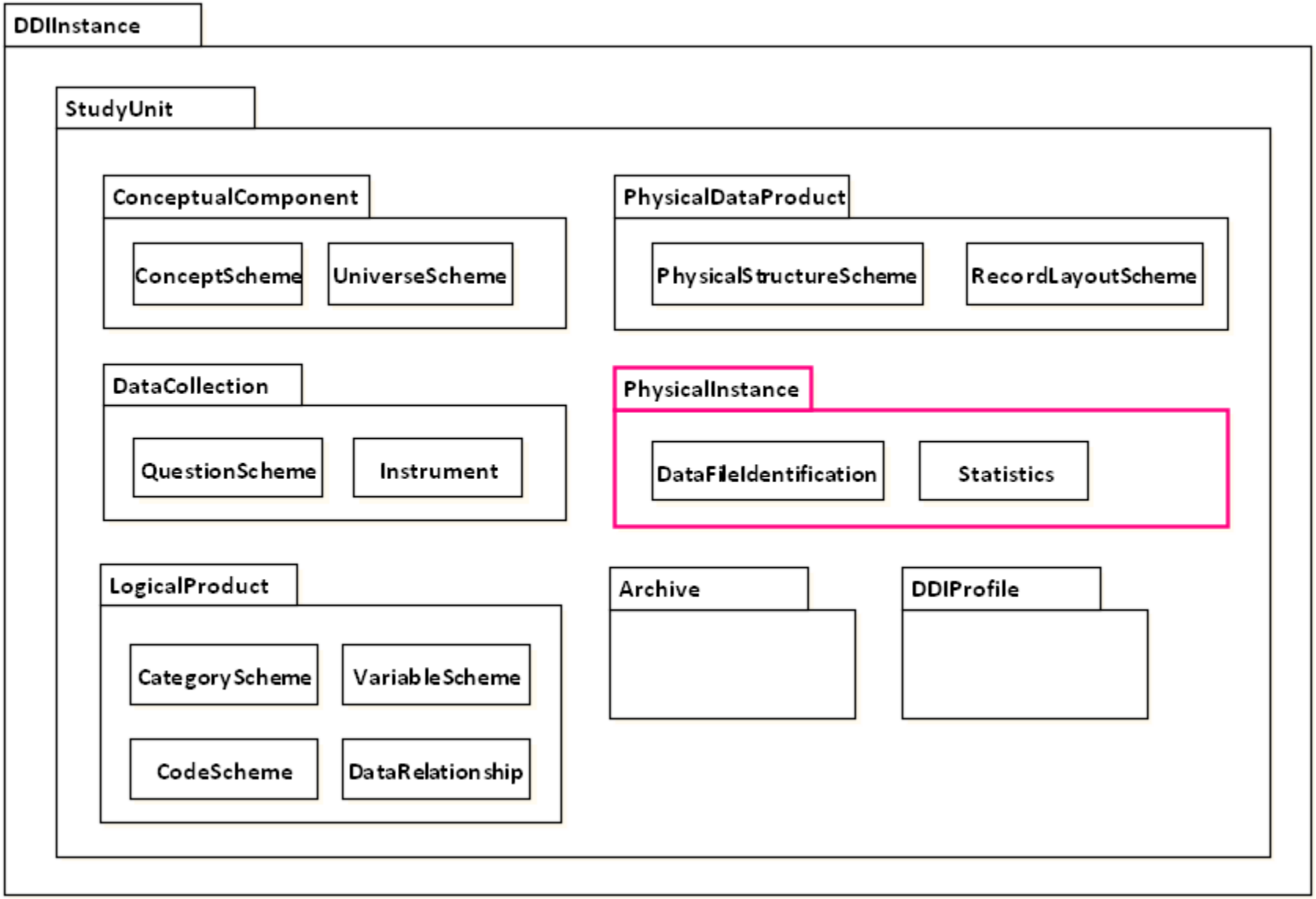
- Persistence Layer
  - Programming Interface
  - Types of physical data storages
  - Examples
- Physical Data Product
- **Physical Instance**
- Archive
- DDIProfile

Data Storage Access

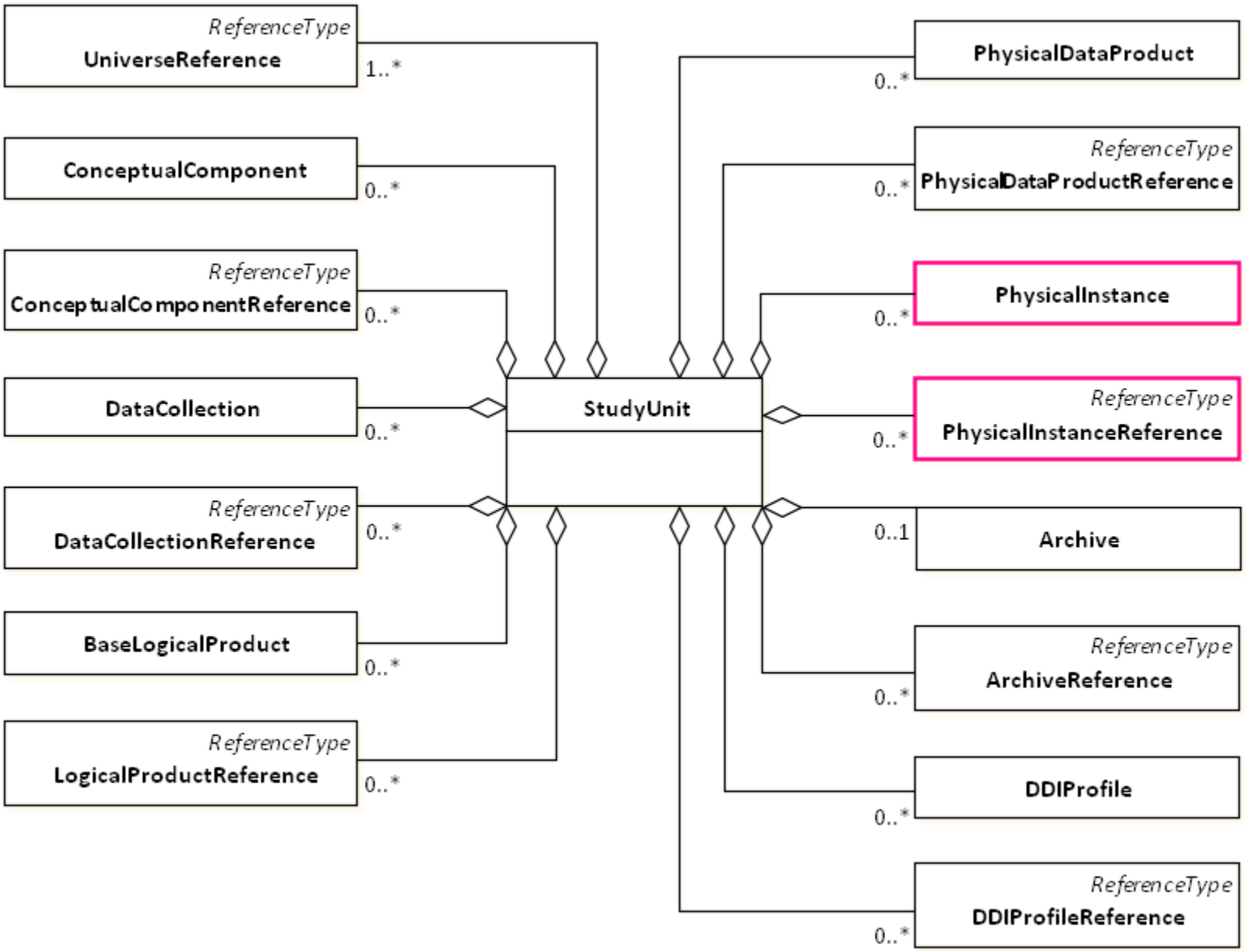
- Outlook
- DDI Serialization Examples
  - DDI-XML
  - DDI-RDF / disco-spec
  - Relational-DB

Data Storage

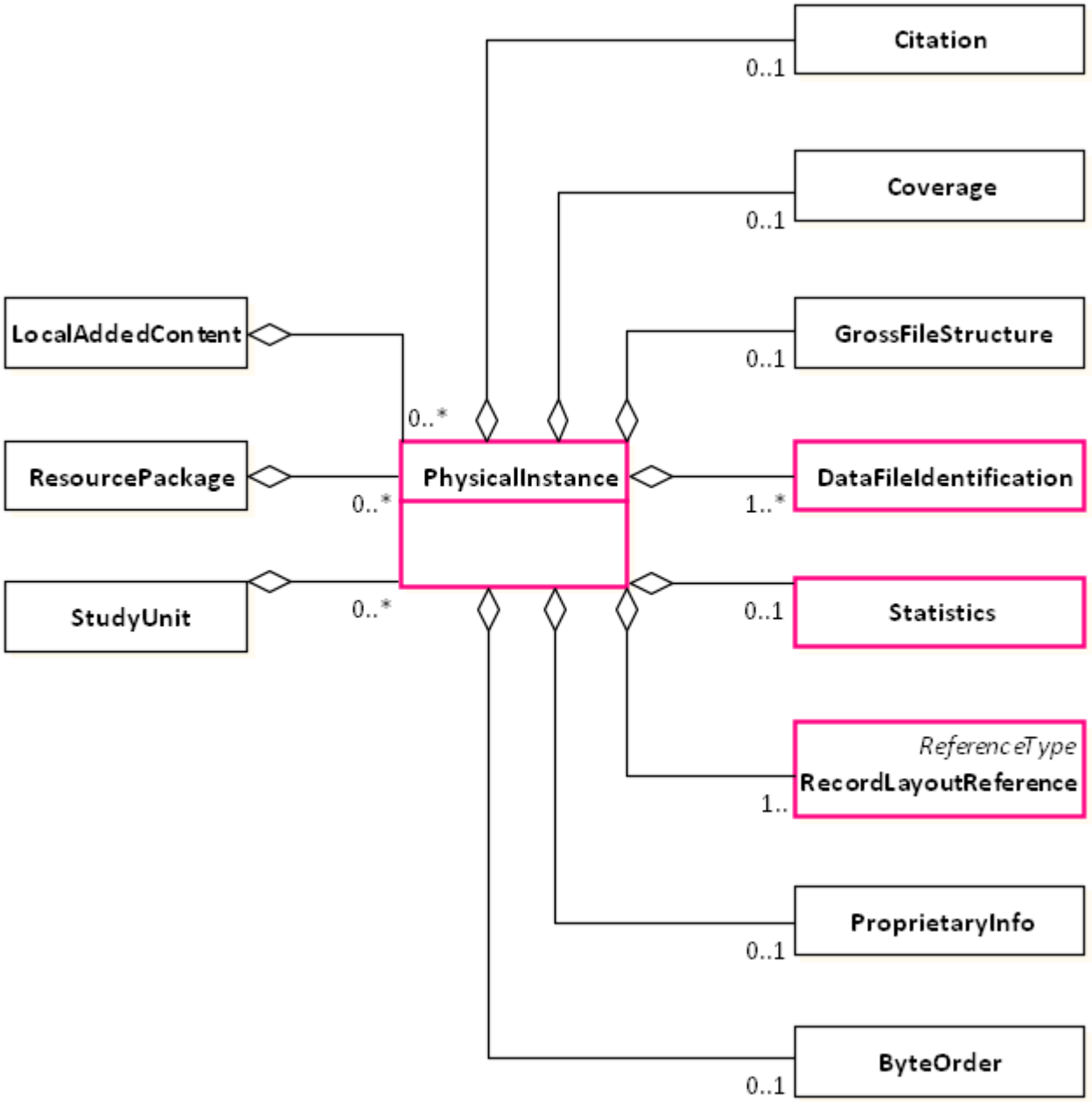
PhysicalInstance



PhysicalInstance



PhysicalInstance



Outline

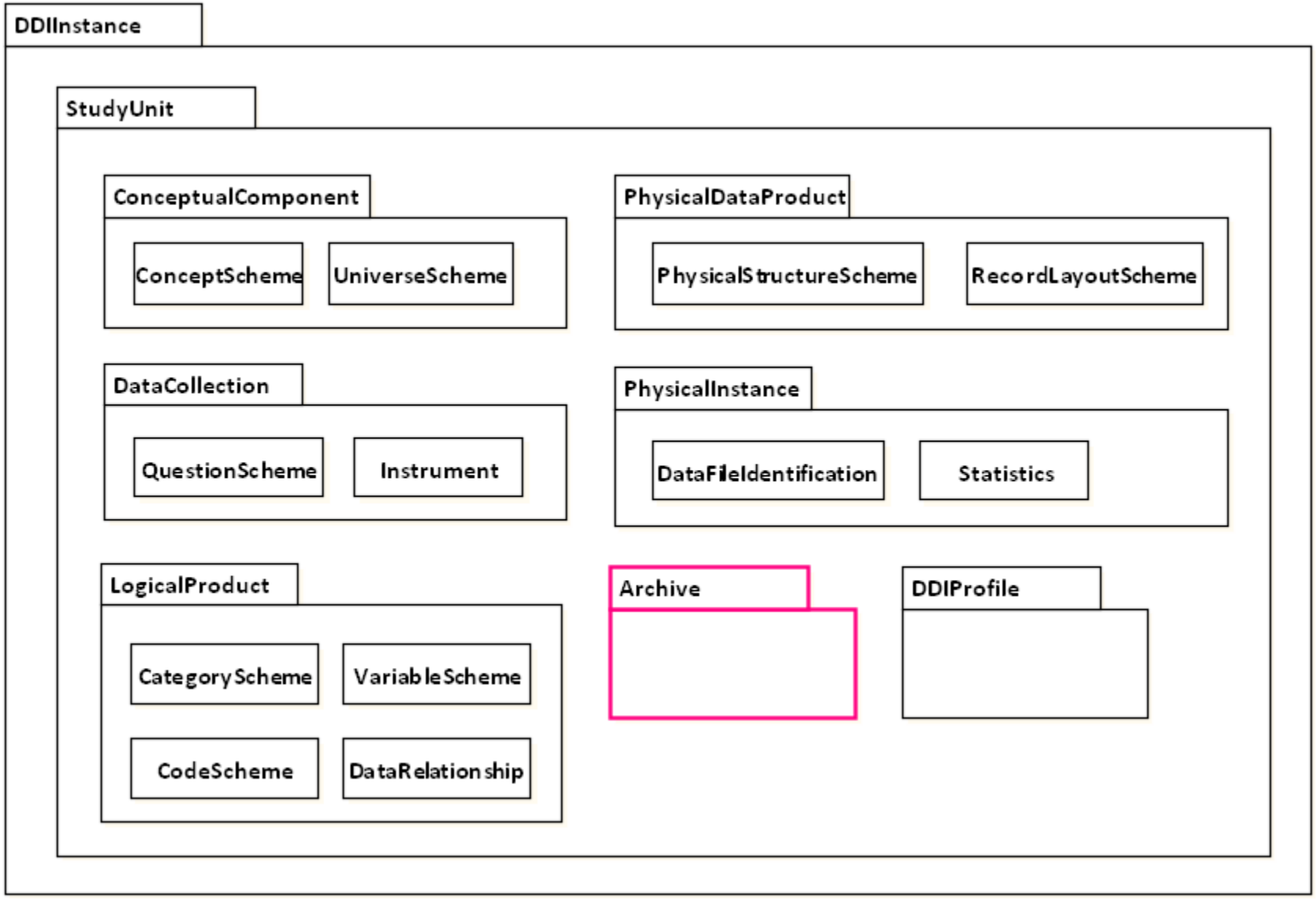
- Persistence Layer
  - Programming Interface
  - Types of physical data storages
  - Examples
- Physical Data Product
- Physical Instance
- **Archive**
- DDIProfile

Data Storage Access

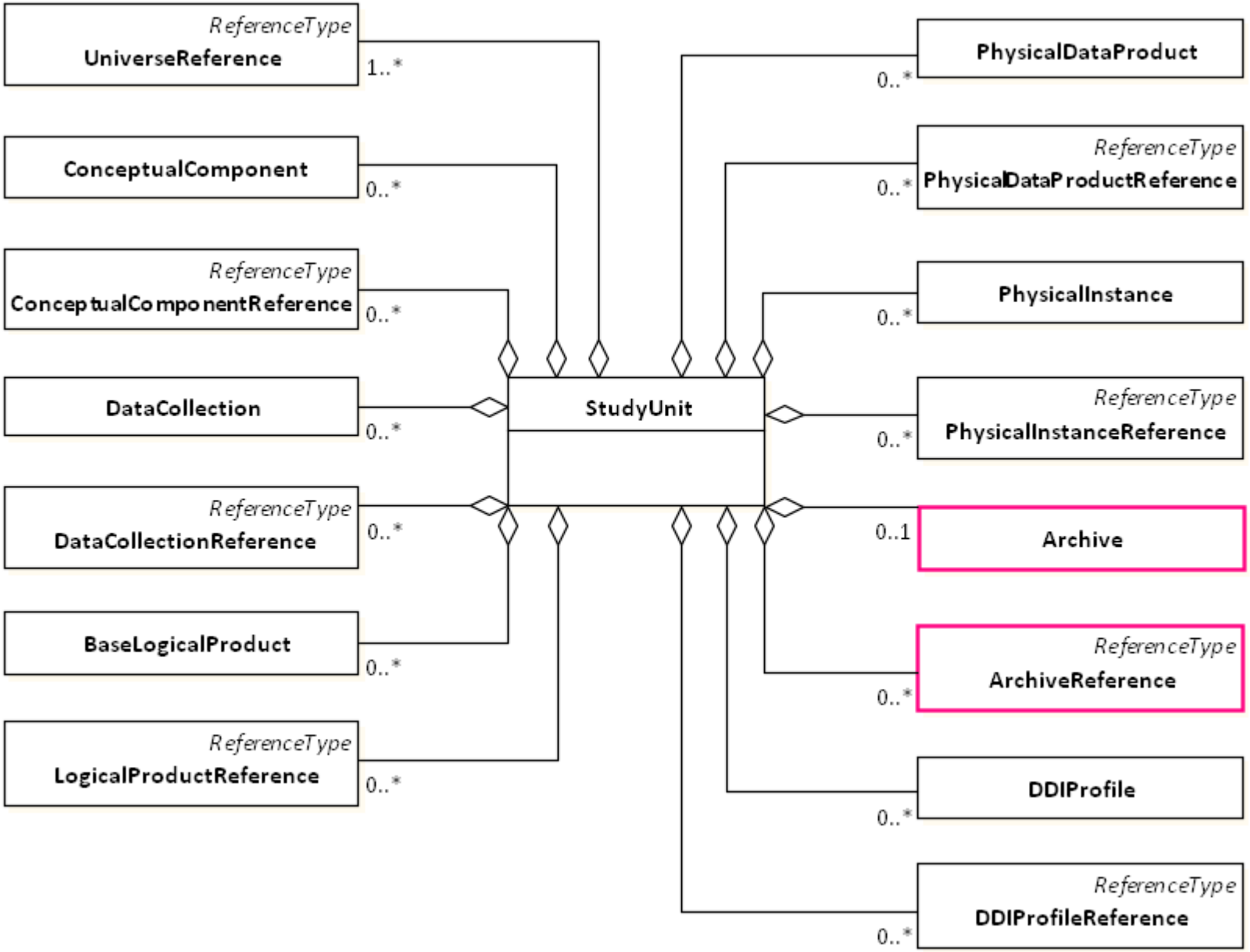
- Outlook
- DDI Serialization Examples
  - DDI-XML
  - DDI-RDF / disco-spec
  - Relational-DB

Data Storage

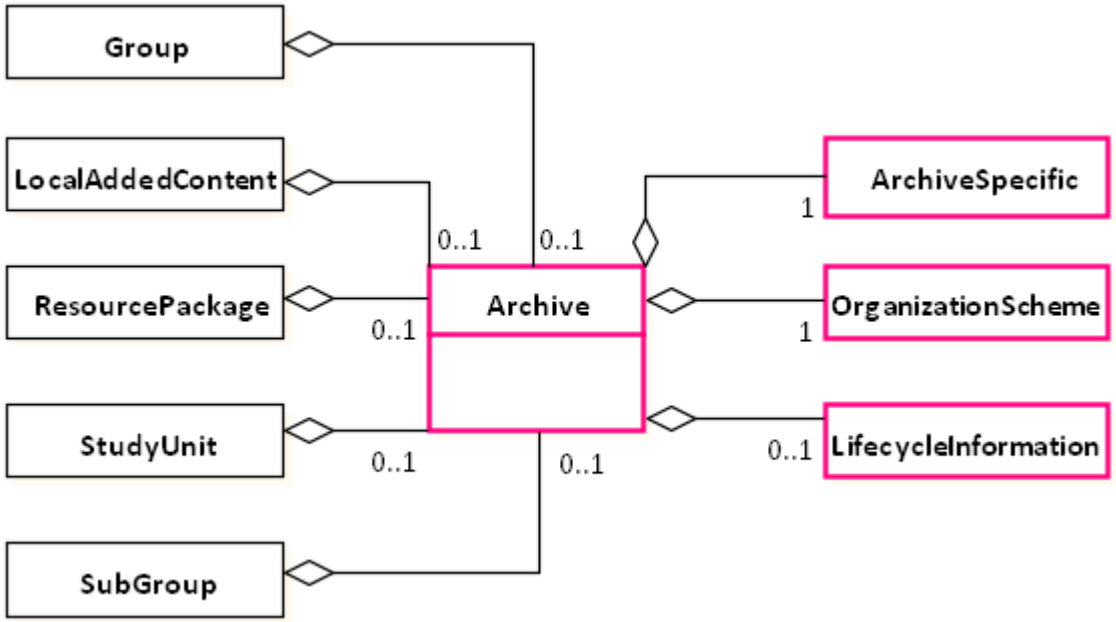
Archive


DDIInstance
StudyUnit
ConceptualComponent
ConceptScheme
UniverseScheme
PhysicalDataProduct
PhysicalStructureScheme
RecordLayoutScheme
DataCollection
QuestionScheme
Instrument
PhysicalInstance
DataFileIdentification
Statistics
LogicalProduct
CategoryScheme
VariableScheme
CodeScheme
DataRelationship
Archive
DDIProfile

Archive


ReferenceType
UniverseReference
1..*
ConceptualComponent
0..*
ReferenceType
ConceptualComponentReference
0..*
DataCollection
0..*
ReferenceType
DataCollectionReference
0..*
BaseLogicalProduct
0..*
ReferenceType
LogicalProductReference
0..*
StudyUnit
PhysicalDataProduct
0..*
ReferenceType
PhysicalDataProductReference
0..*
PhysicalInstance
0..*
ReferenceType
PhysicalInstanceReference
0..*
0..1
Archive
ReferenceType
ArchiveReference
0..*
DDIProfile
0..*
ReferenceType
DDIProfileReference
0..*

Archive



Contains persistent lifecycle events

Contains archive-specific information (archive reference, access, funding)

Outline

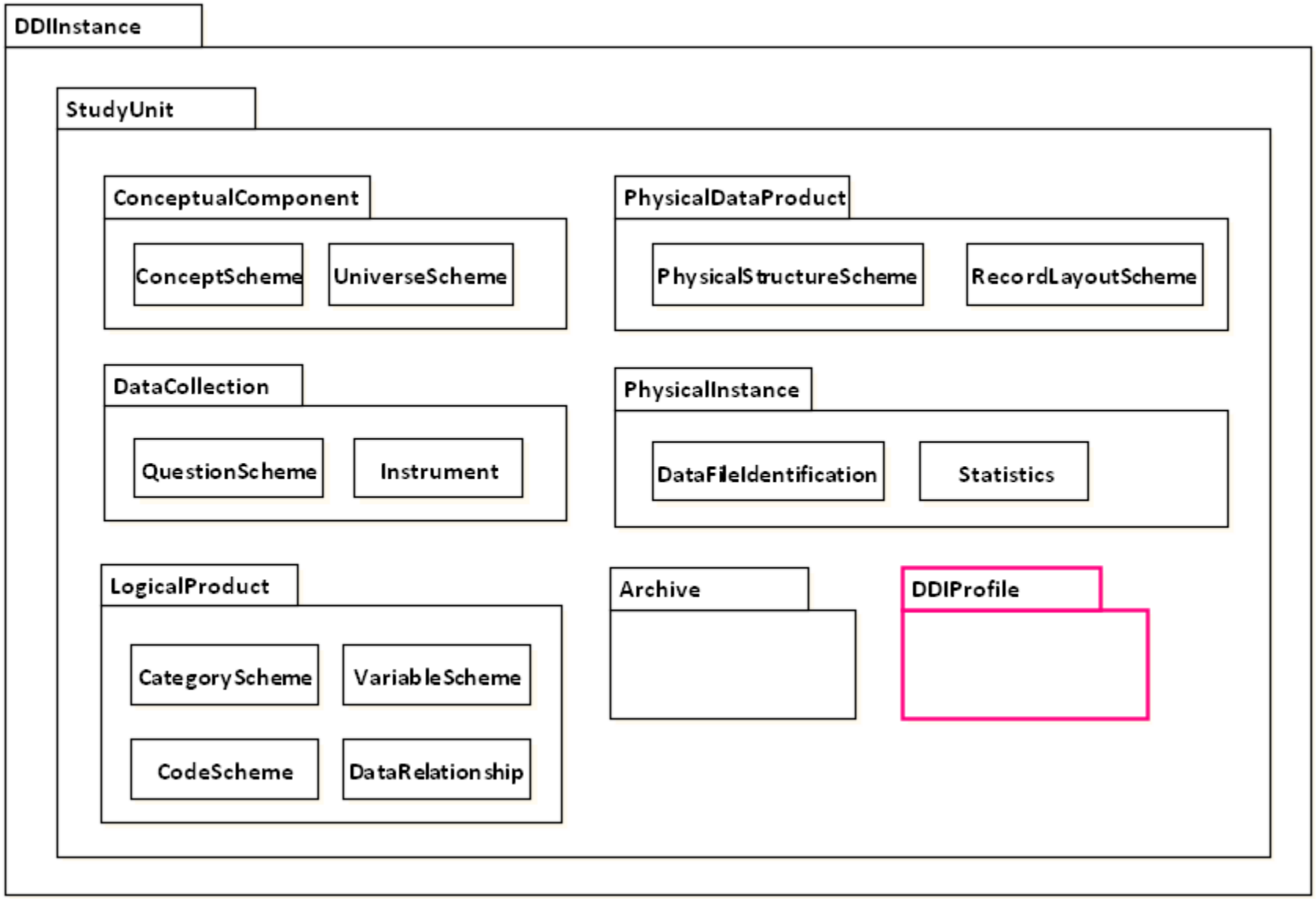
- Persistence Layer
  - Programming Interface
  - Types of physical data storages
  - Examples
- Physical Data Product
- Physical Instance
- Archive
- **DDIProfile**

Data Storage Access

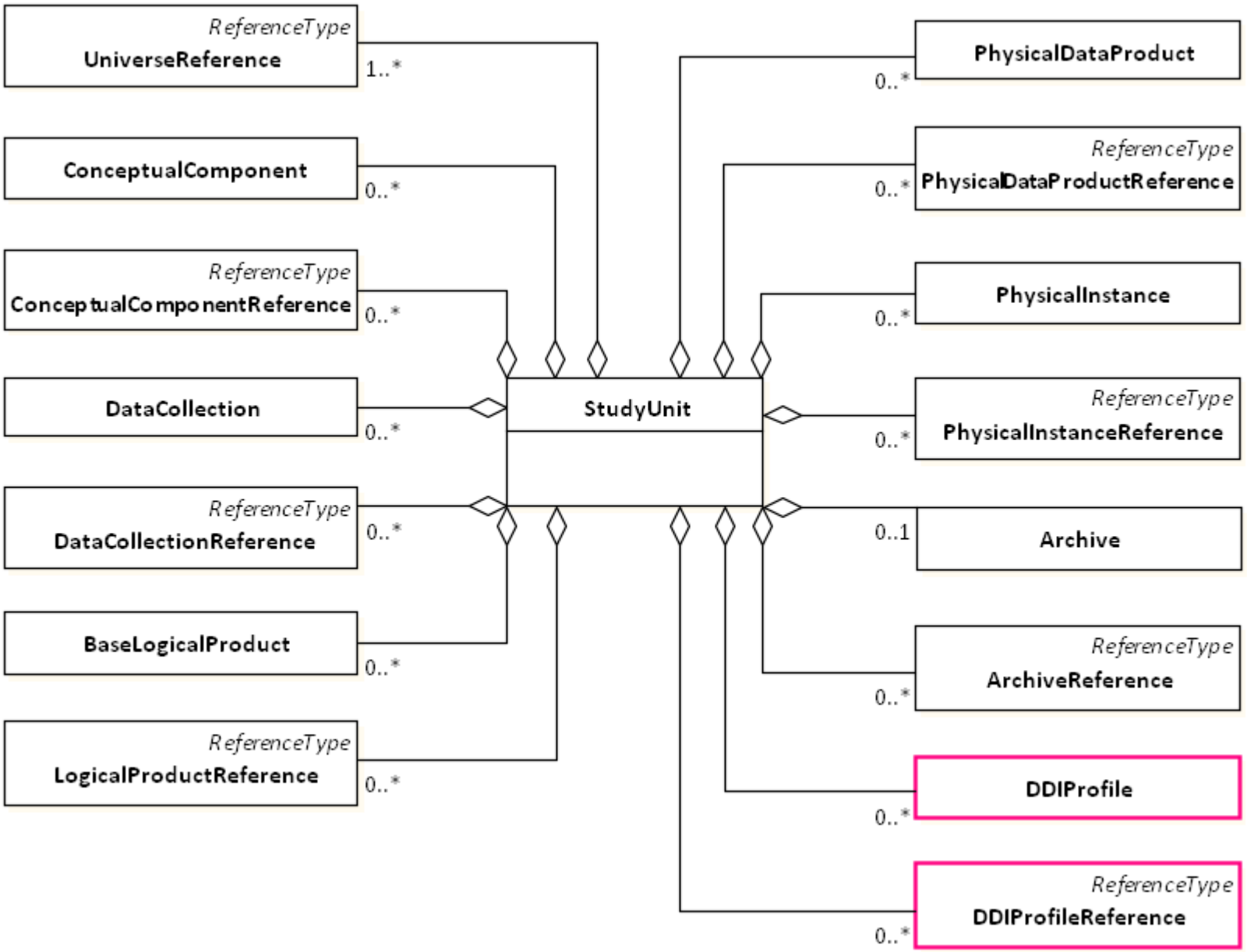
- Outlook
- DDI Serialization Examples
  - DDI-XML
  - DDI-RDF / disco-spec
  - Relational-DB

Data Storage

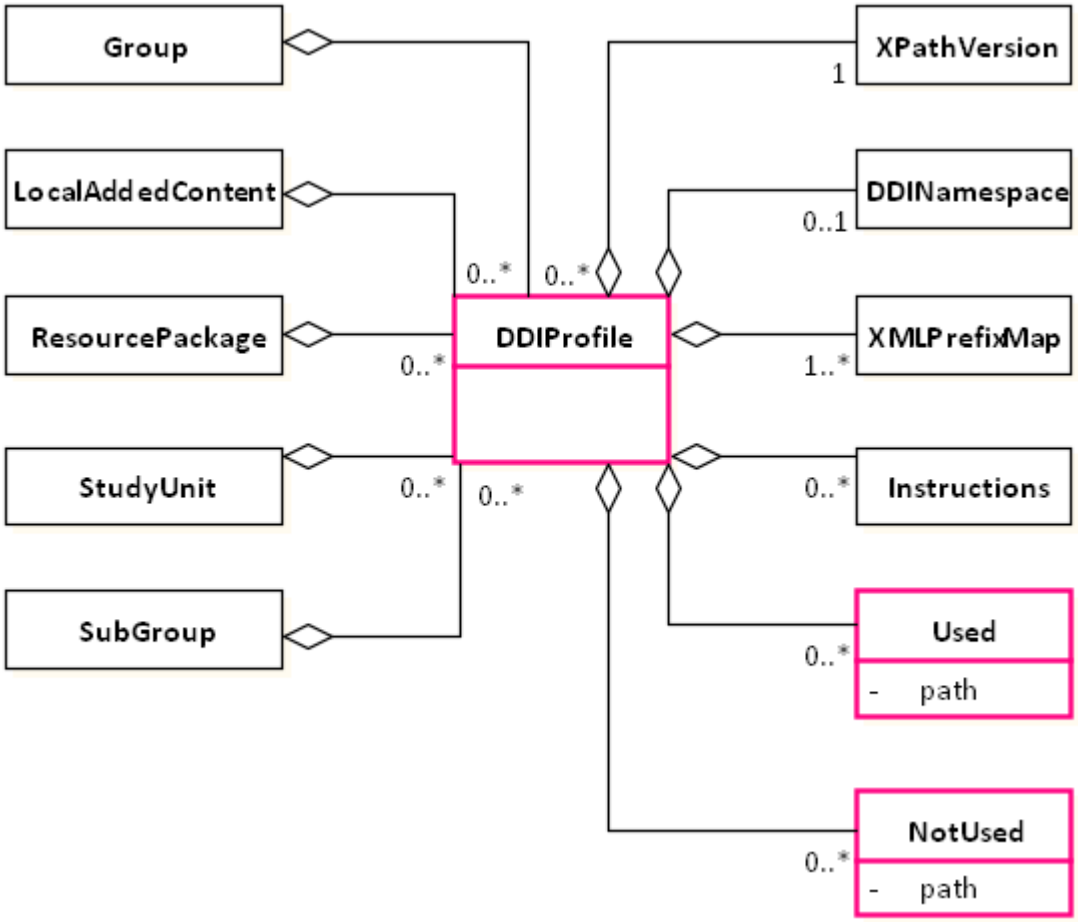
DDIProfile


DDIInstance
StudyUnit
ConceptualComponent
ConceptScheme
UniverseScheme
PhysicalDataProduct
PhysicalStructureScheme
RecordLayoutScheme
DataCollection
QuestionScheme
Instrument
PhysicalInstance
DataFileIdentification
Statistics
LogicalProduct
CategoryScheme
VariableScheme
CodeScheme
DataRelationship
Archive
DDIProfile

DDIProfile


ReferenceType
UniverseReference
1..*
ConceptualComponent
0..*
ReferenceType
ConceptualComponentReference
0..*
DataCollection
0..*
ReferenceType
DataCollectionReference
0..*
BaseLogicalProduct
0..*
ReferenceType
LogicalProductReference
0..*
StudyUnit
PhysicalDataProduct
0..*
ReferenceType
PhysicalDataProductReference
0..*
PhysicalInstance
0..*
ReferenceType
PhysicalInstanceReference
0..*
0..1
Archive
ReferenceType
ArchiveReference
0..*
DDIProfile
0..*
ReferenceType
DDIProfileReference
0..*

DDIProfile



Describes which DDI-L elements you use in your institution's DDI flavor

How the presentation is continued

- How can the disco-model be persisted?

and

- How can it be done in a project with individual data model elements?
- How do we do this in the Missy application?

Outline

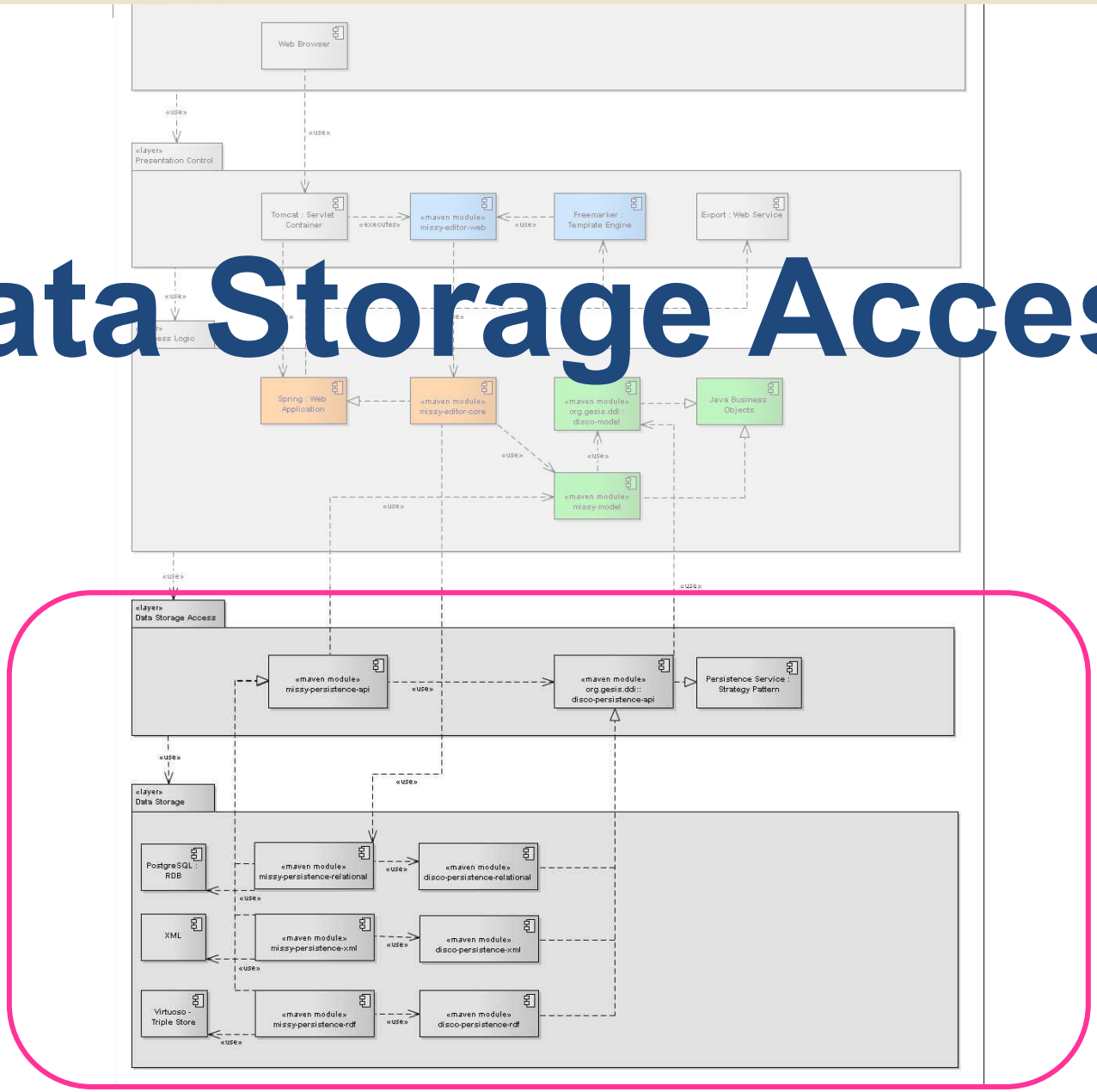
- **Persistence Layer**
  - Programming Interface
  - Types of physical data storages
  - Examples
- Physical Data Product
- Physical Instance
- Archive
- DDIProfile

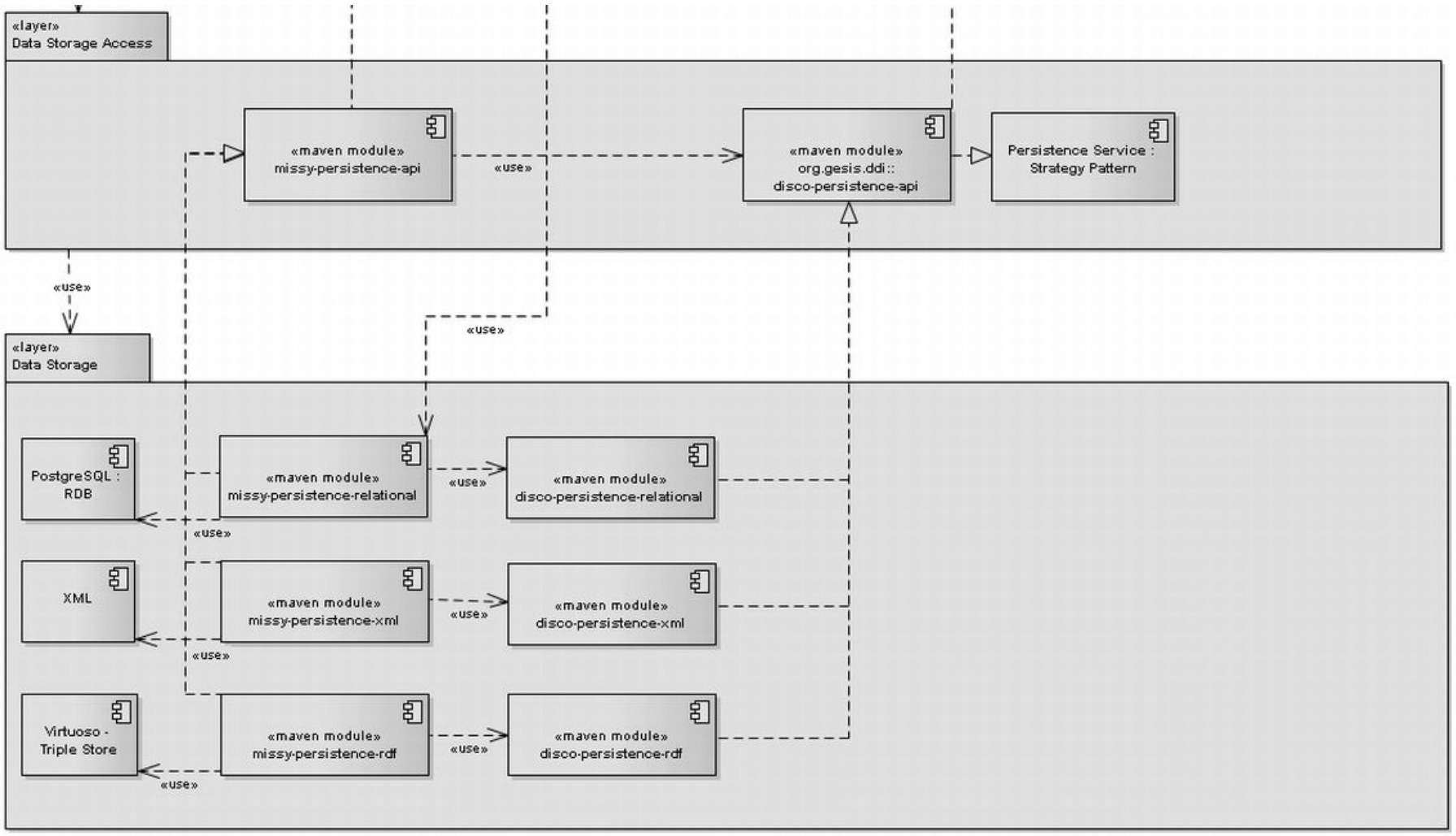
Data Storage Access

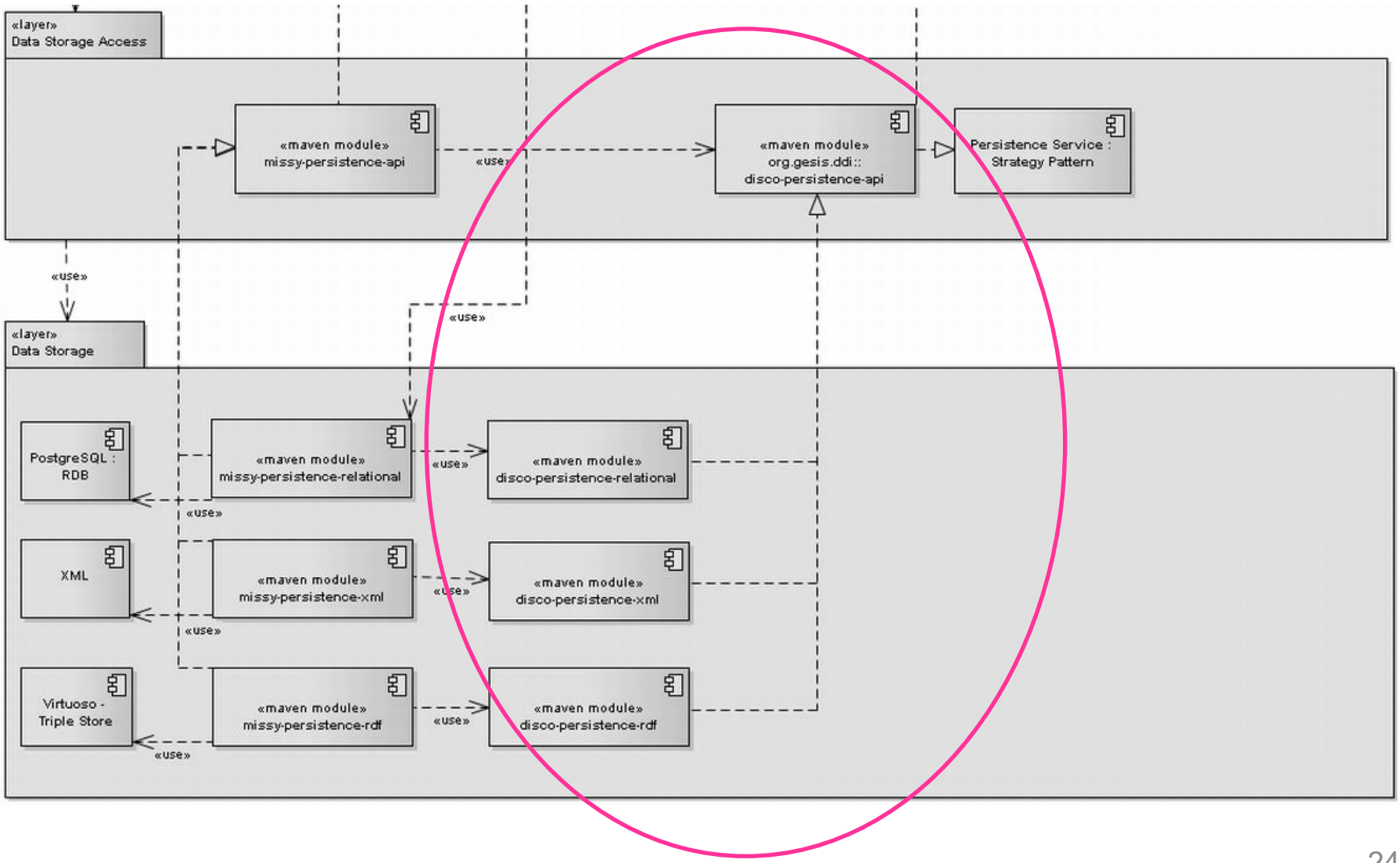
- Outlook
- DDI Serialization Examples
  - DDI-XML
  - DDI-RDF / disco-spec
  - Relational-DB

Data Storage

Data Storage Access




«layer»
Data Storage Access
«maven module»
missy-persistence-api
«use»
«maven module»
org.gesis.ddi::
disco-persistence-api
Persistence Service :
Strategy Pattern
«use»
«use»
«layer»
Data Storage
PostgreSQL :
RDB
«maven module»
missy-persistence-relational
«use»
«maven module»
disco-persistence-relational
«use»
XML
«maven module»
missy-persistence-xml
«use»
«maven module»
disco-persistence-xml
«use»
Virtuoso -
Triple Store
«maven module»
missy-persistence-rdf
«use»
«maven module»
disco-persistence-rdf
«use»


«layer»
Data Storage Access
«maven module»
missy-persistence-api
«use»
«maven module»
org.gesis.ddi::
disco-persistence-api
Persistence Service :
Strategy Pattern
«use»
«use»
«layer»
Data Storage
PostgreSQL :
RDB
«maven module»
missy-persistence-relational
«use»
«maven module»
disco-persistence-relational
«use»
XML
«maven module»
missy-persistence-xml
«use»
«maven module»
disco-persistence-xml
«use»
Virtuoso -
Triple Store
«maven module»
missy-persistence-rdf
«use»
«maven module»
disco-persistence-rdf
«use»

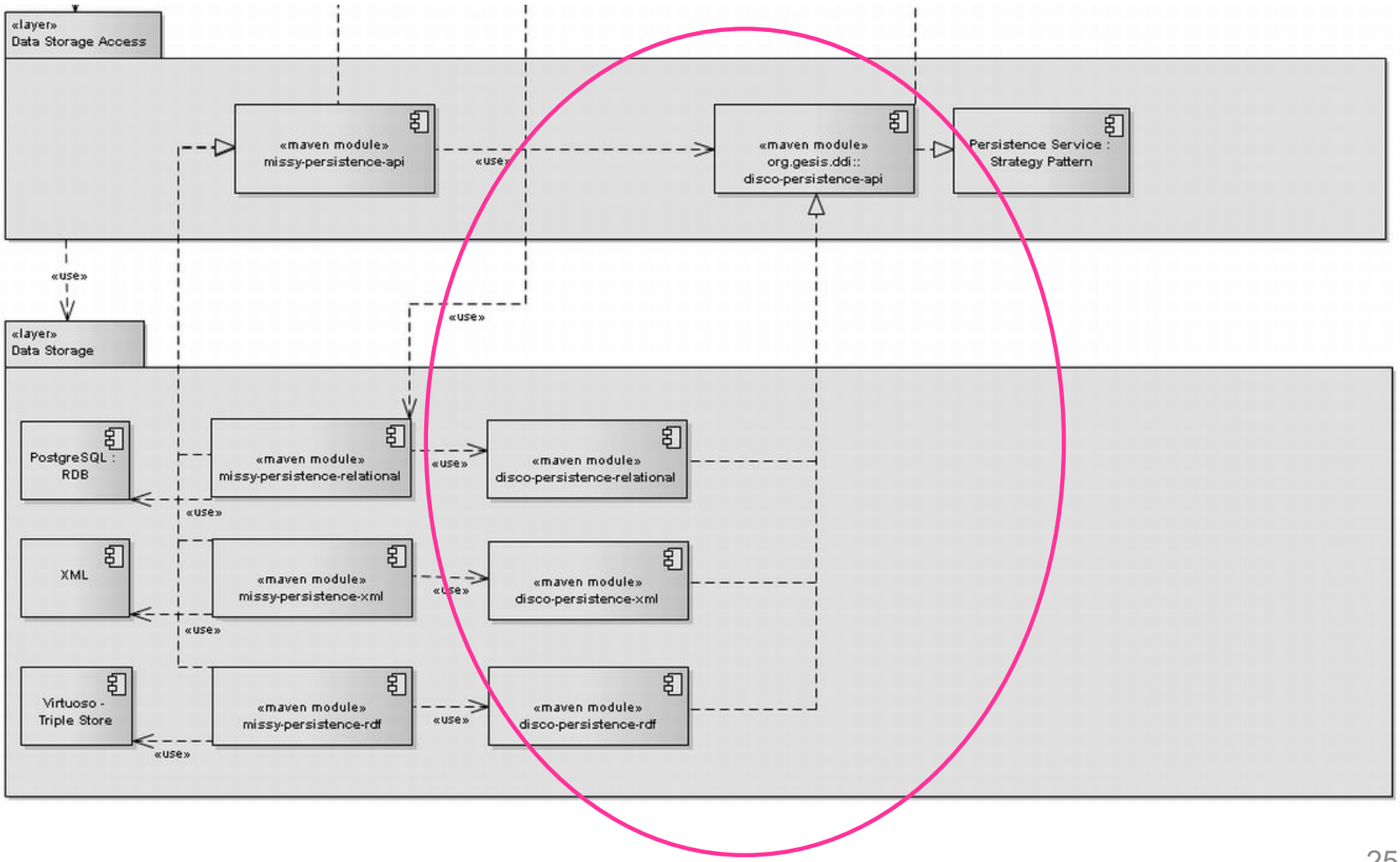
Persistence Layer – Motivation

- In a well-defined software architecture the application itself does not need to know *how* the data is stored
- It just needs to know the API
  - Methods that are provided to access and store objects
  - May be abstracted away from the actual implementation

- An actual implementation or strategy can just be a matter of configuration

Persistence Layer – Strategy Pattern

- Implementation of the Strategy Pattern
  - simple interface
  - actual implementations are the different strategies
- A strategy is an implementation of the actual type of persistence or physical storage, respectively
  - e.g. DDI-L-XML, DDI-RDF, XML-DB, Relational-DB, etc.


«layer»
Data Storage Access
«maven module»
missy-persistence-api
«use»
«maven module»
org.gesis.ddi::
disco-persistence-api
Persistence Service :
Strategy Pattern
«use»
«use»
«layer»
Data Storage
PostgreSQL :
RDB
«maven module»
missy-persistence-relational
«use»
«maven module»
disco-persistence-relational
«use»
XML
«maven module»
missy-persistence-xml
«use»
«maven module»
disco-persistence-xml
«use»
Virtuoso -
Triple Store
«maven module»
missy-persistence-rdf
«use»
«maven module»
disco-persistence-rdf
«use»

Modules

- disco-persistence-api
  - **Defines** persistence functionality for model components regardless of the actual type of physical persistence
- disco-persistence-relational
  - **Implements** the persistence functionality defined in disco-persistence-api with respect to the usage of relational DBs
- disco-persistence-xml
  - **Implements** the persistence functionality defined in disco-persistence-api with respect to the usage of DDI-XML
- disco-persistence-rdf
  - **Implements** the persistence functionality defined in disco-persistence-api with respect to the usage of the disco-specification

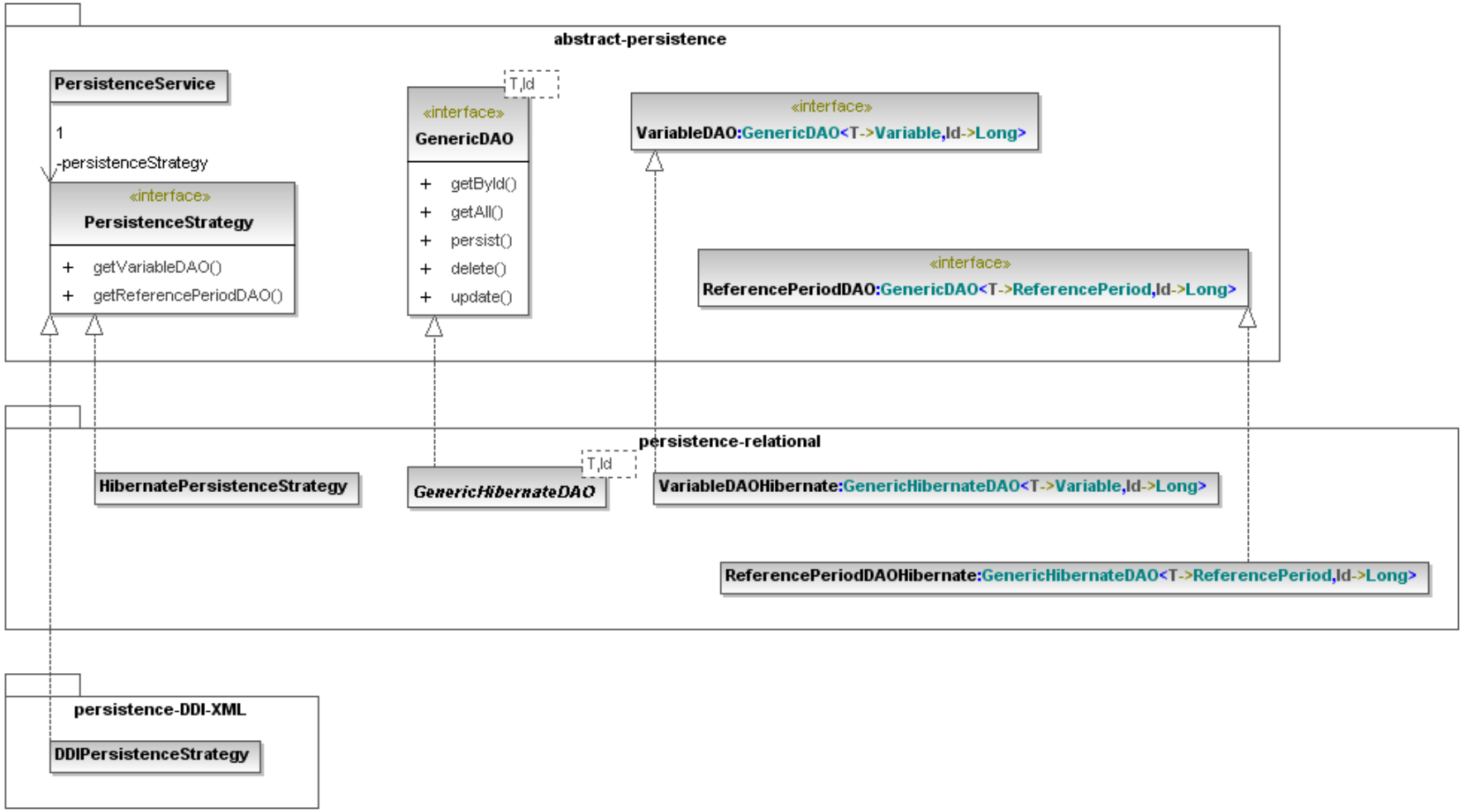
Data Access Object

- A DAO is an object that is responsible for providing methods for reading and storing of objects in our model
- A DAO is again implemented against an interface

- For each business object in the data model there exists a DAO

- Persistence strategy interface defines methods for obtaining specific DAOs (data access objects)

DAO Pattern


abstract-persistence
PersistenceService
1
-persistenceStrategy
«interface»
PersistenceStrategy
+ getVariableDAO()
+ getReferencePeriodDAO()
T,Id
«interface»
GenericDAO
+ getById()
+ getAll()
+ persist()
+ delete()
+ update()
«interface»
VariableDAO:GenericDAO<T->Variable,Id->Long>
«interface»
ReferencePeriodDAO:GenericDAO<T->ReferencePeriod,Id->Long>
persistence-relational
HibernatePersistenceStrategy
T,Id
GenericHibernateDAO
VariableDAOHibernate:GenericHibernateDAO<T->Variable,Id->Long>
ReferencePeriodDAOHibernate:GenericHibernateDAO<T->ReferencePeriod,Id->Long>
persistence-DDI-XML
DDIPersistenceStrategy

disco-persistence-api

- Provides an API and implementations for the disco-model
- PersistenceService
  - Available in the disco-persistence-api
  - Encapsulates the actual strategy
  - Strategy can be instantiated and injected into the PersistenceService

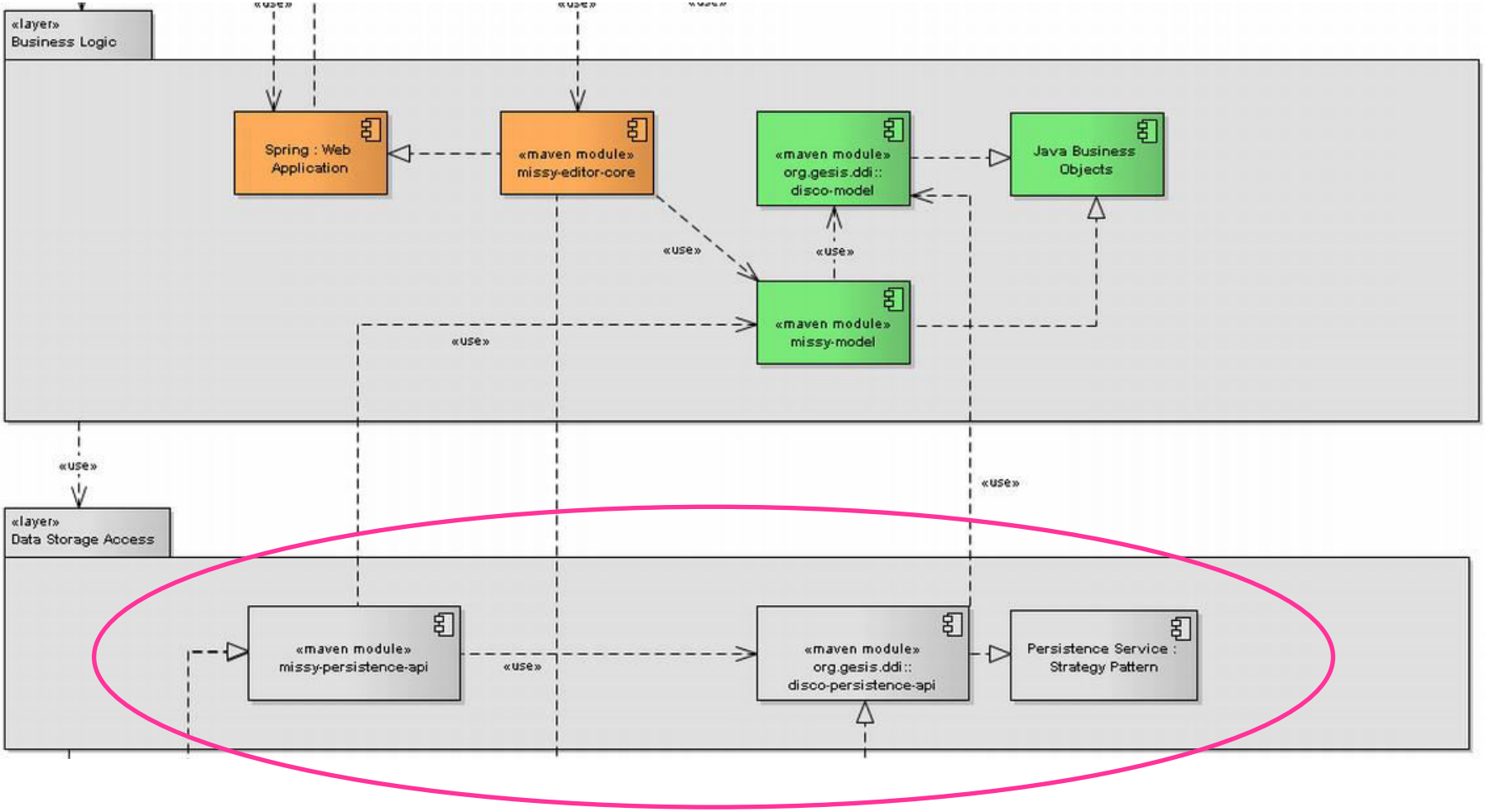
Extendable disco-persistence-api

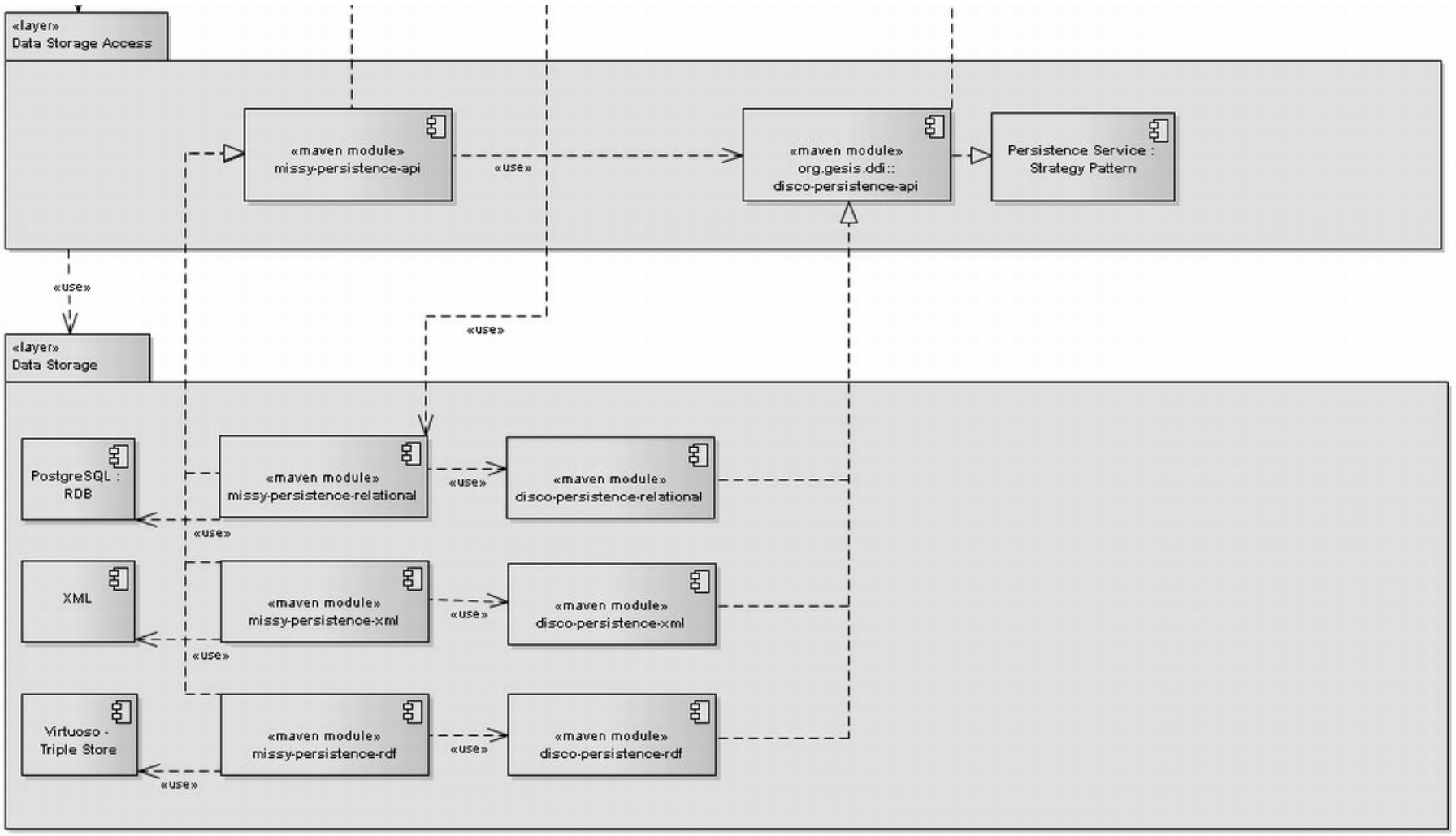
- But, projects have own requirements to the data model
  - => new business objects are included
  - => DAOs need to be written for reading and storing these objects
  - => these DAOs also have to implement the appropriate type of persistence used by that project

- => this API has to be extendable

Extendable disco-persistence-api

- But, projects have own requirements to the data model
  - => new business objects are included
  - => DAOs need to be written for reading and storing these objects
  - => these DAOs also have to implement the appropriate type of persistence used by that project
- => this API has to be extendable
- Luckily it is, because of the interface structure!
- Projects only need to implement the defined methods


«layer»
Business Logic
Spring : Web Application
«maven module»
missy-editor-core
«maven module»
org.gesis.ddi::
disco-model
Java Business Objects
«use»
«use»
«maven module»
missy-model
«use»
«use»
«use»
«layer»
Data Storage Access
«maven module»
missy-persistence-api
«use»
«maven module»
org.gesis.ddi::
disco-persistence-api
Persistence Service : Strategy Pattern


«layer»
Data Storage Access
«maven module»
missy-persistence-api
«use»
«maven module»
org.gesis.ddi::
disco-persistence-api
Persistence Service :
Strategy Pattern
«use»
«use»
«layer»
Data Storage
PostgreSQL :
RDB
«maven module»
missy-persistence-relational
«use»
«maven module»
disco-persistence-relational
«use»
XML
«maven module»
missy-persistence-xml
«use»
«maven module»
disco-persistence-xml
«use»
Virtuoso -
Triple Store
«maven module»
missy-persistence-rdf
«use»
«maven module»
disco-persistence-rdf
«use»

Missy Specific Modules

- missy-persistence-api
  - **Defines additional** persistence functionality for model components regardless of the actual type of physical persistence
- missy-persistence-relational
  - **Implements additional** persistence functionality for model components with respect to the usage of relational DBs
- missy-persistence-xml
  - **Implements additional** persistence functionality for model components with respect to the usage of DDI-XML
- missy-persistence-rdf
  - **Implements additional** persistence functionality for model components with respect to the usage of the disco-specification

What we have seen so far

- Implementation of a persistence API for the disco model
- How the persistence API can be extended in individual projects

How the presentation is continued

- How does, on the persistence level, the serialization of data stored in DDI really look like?

Outline

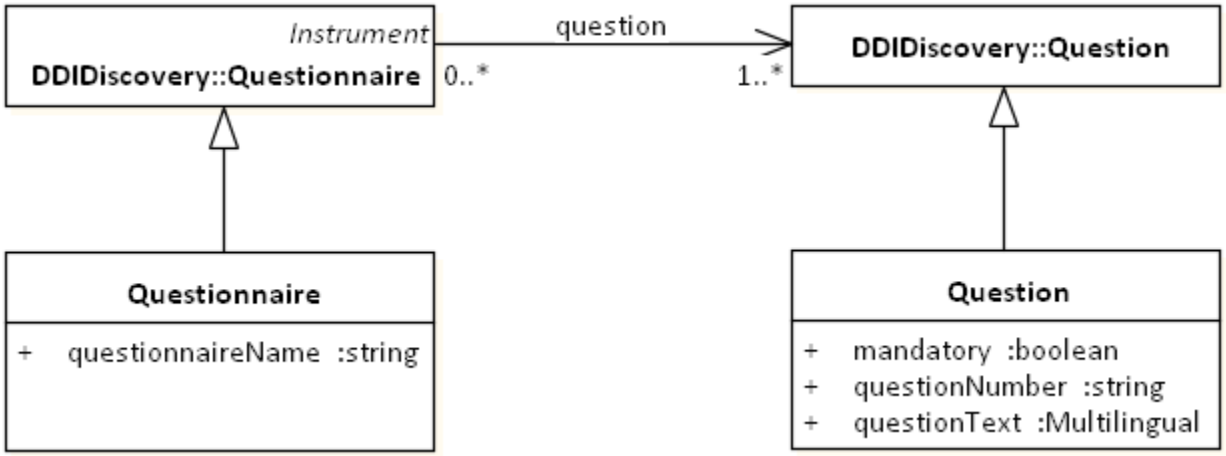
- Persistence Layer
  - Programming Interface
  - Types of physical data storages
  - Examples
- Physical Data Product
- Physical Instance
- Archive
- DDIProfile

Data Storage Access

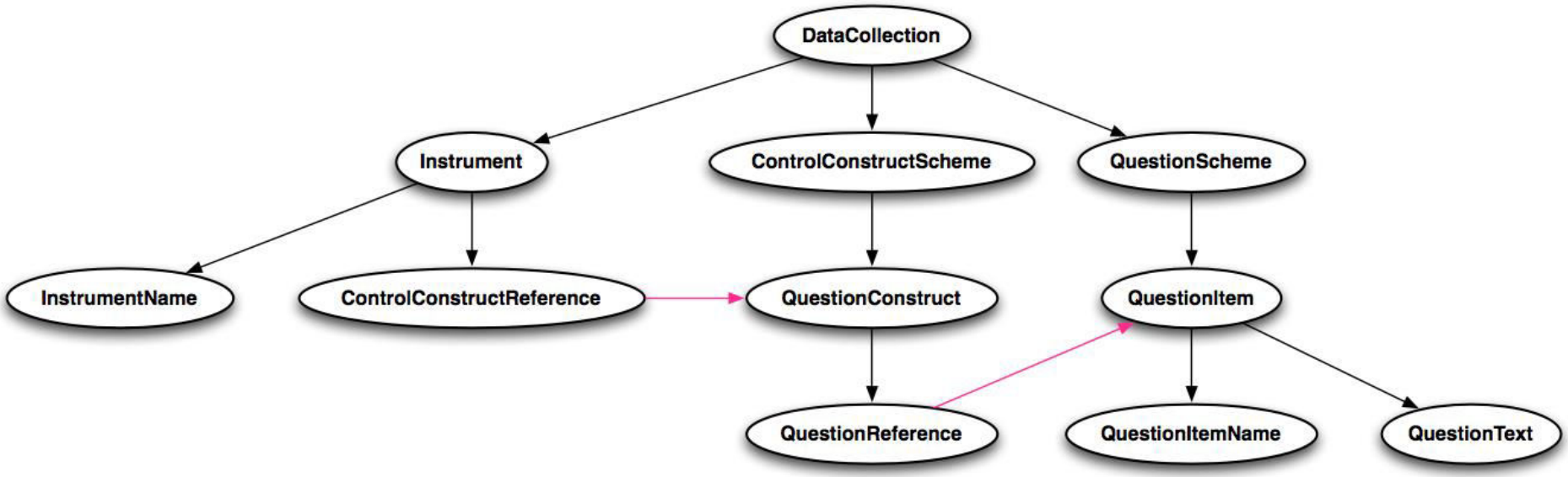
- Outlook
- **DDI Serialization Examples**
  - **DDI-XML**
  - **DDI-RDF / disco-spec**
  - Relational-DB

Data Storage

Questionnaire - Question



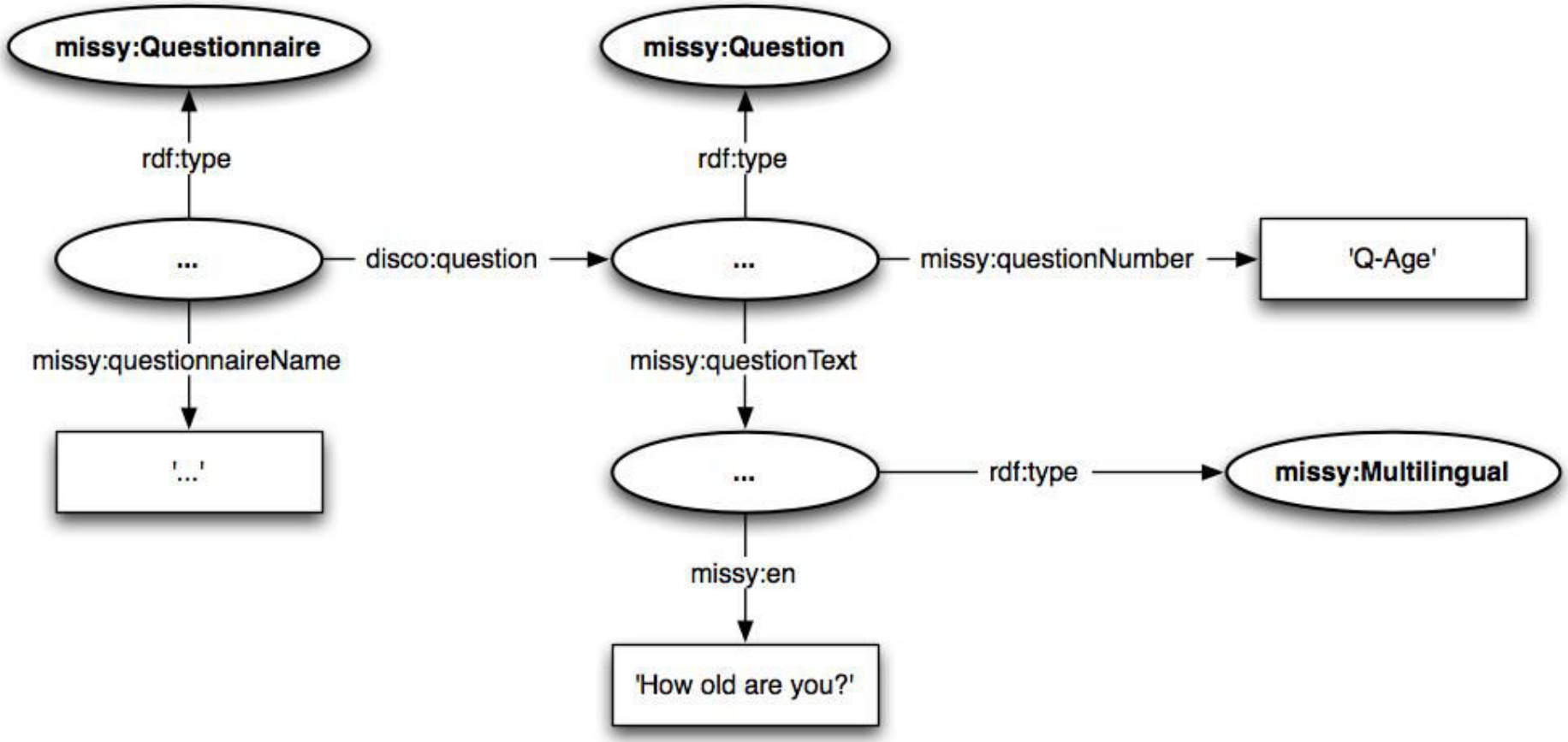
Questionnaire – Question (DDI 3.1 XML)



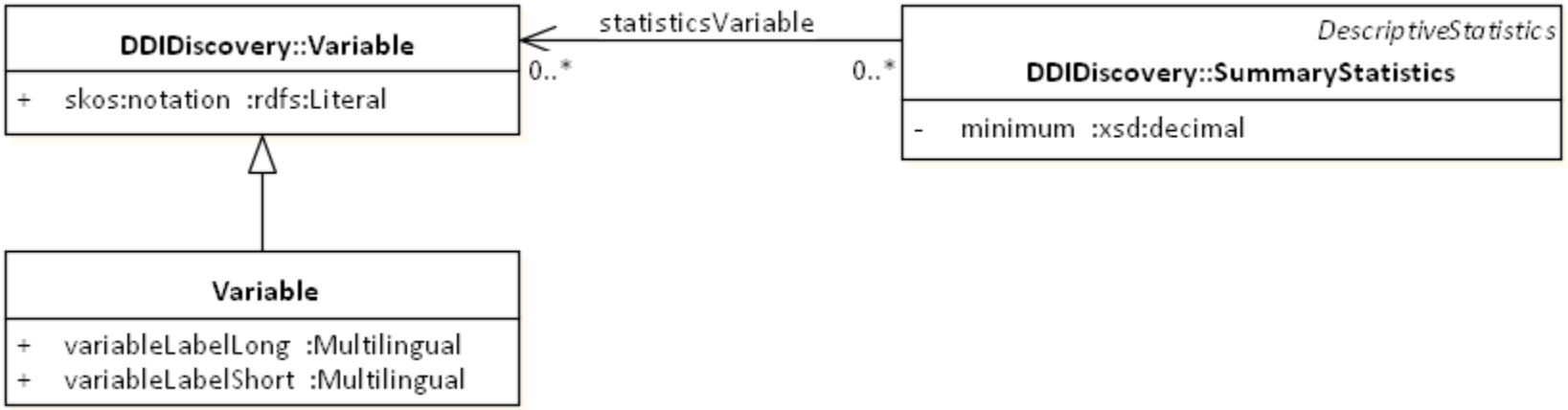
DataCollection

- Instrument
  - IntrumentName
  - ControlConstructReference
- ControlCostructScheme
  - QuestionConstruct
    - QuestionReference
- QuestionScheme
  - QuestionItem
    - QuestionItemName
    - QuestionText

Questionnaire – Question (DDI-RDF)



Variable - SummaryStatistics

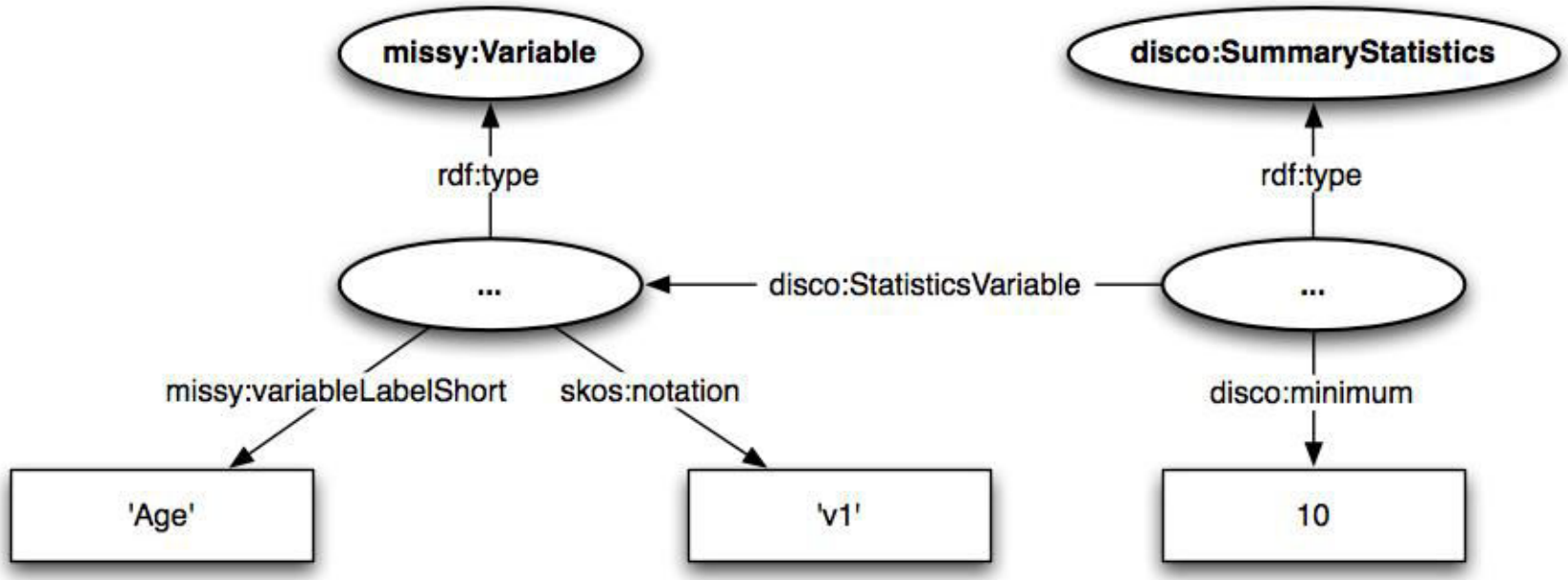


Variable – SummaryStatistics (DDI 3.1 XML)

PhysicalInstance
  Statistics
    VariableStatistics
      VariableReference
      SummaryStatistic
        SummaryStatisticType
          'Minimum'
        Value
          10

Variable
  VariableName
    'v1'
  Label
    'Age'

Variable – SummaryStatistics (DDI-RDF)



Outline

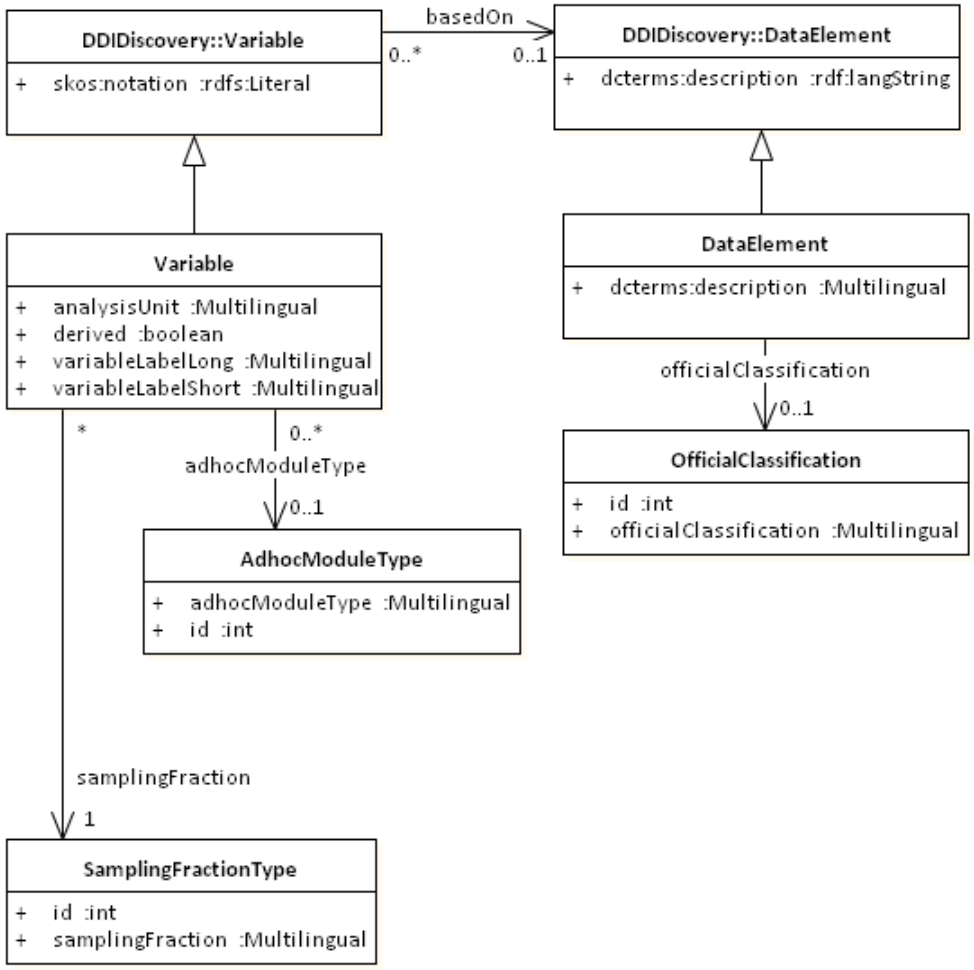
- Persistence Layer
  - Programming Interface
  - Types of physical data storages
  - Examples
- Physical Data Product
- Physical Instance
- Archive
- DDIProfile

Data Storage Access

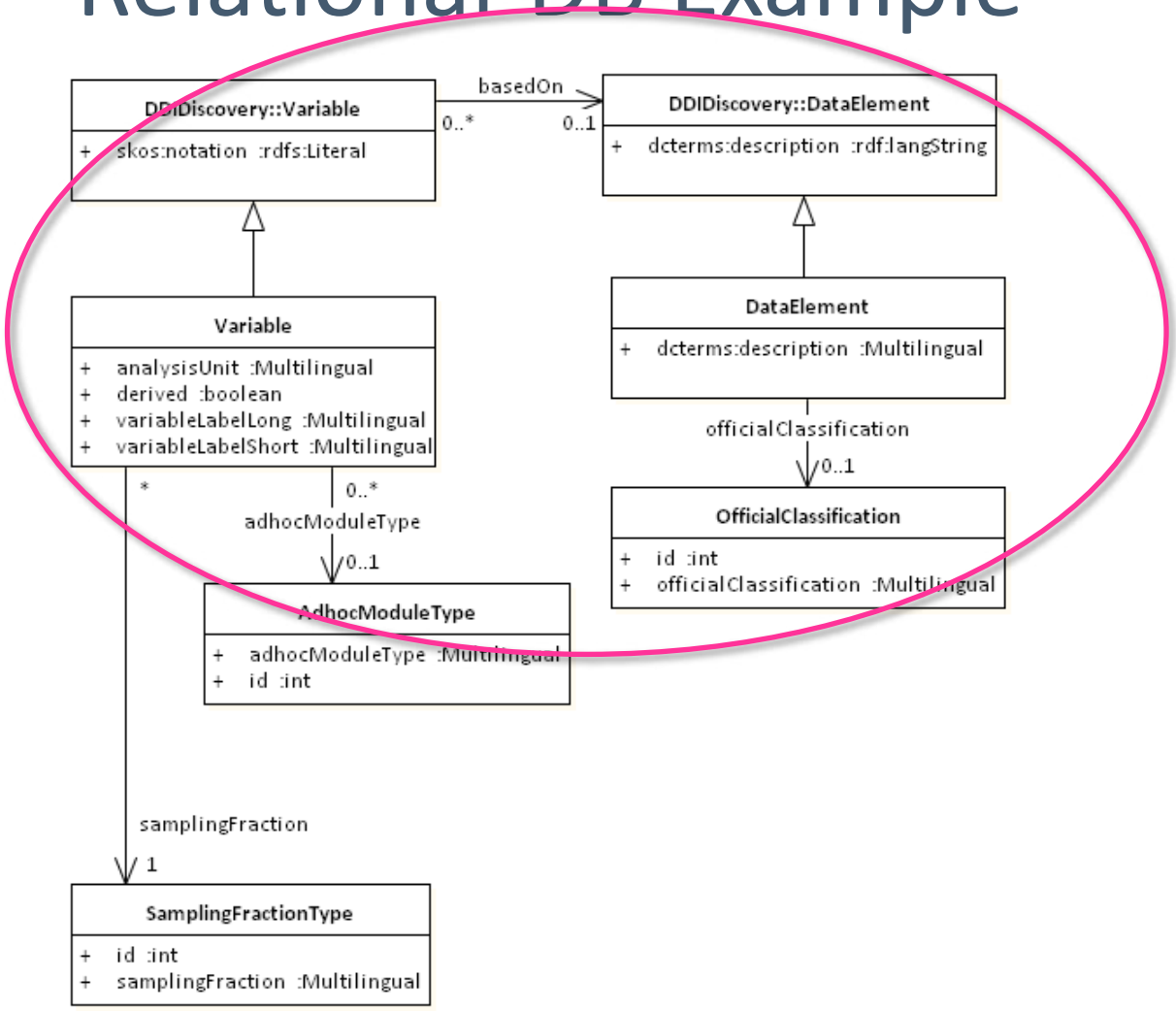
- Outlook
- **DDI Serialization Examples**
  - DDI-XML
  - DDI-RDF / disco-spec
  - **Relational-DB**

Data Storage

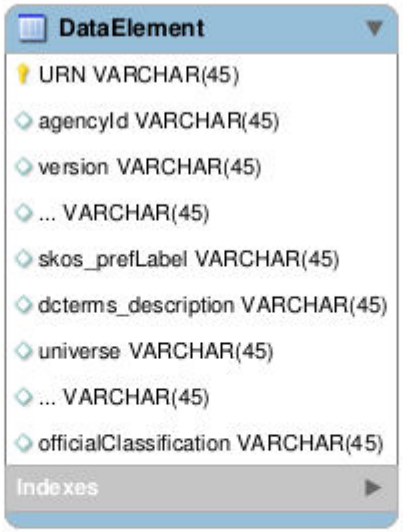
Relational-DB Example

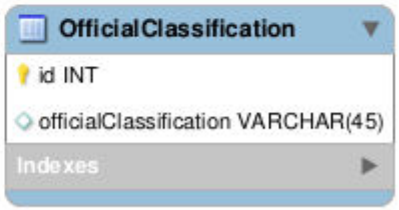


Relational-DB Example

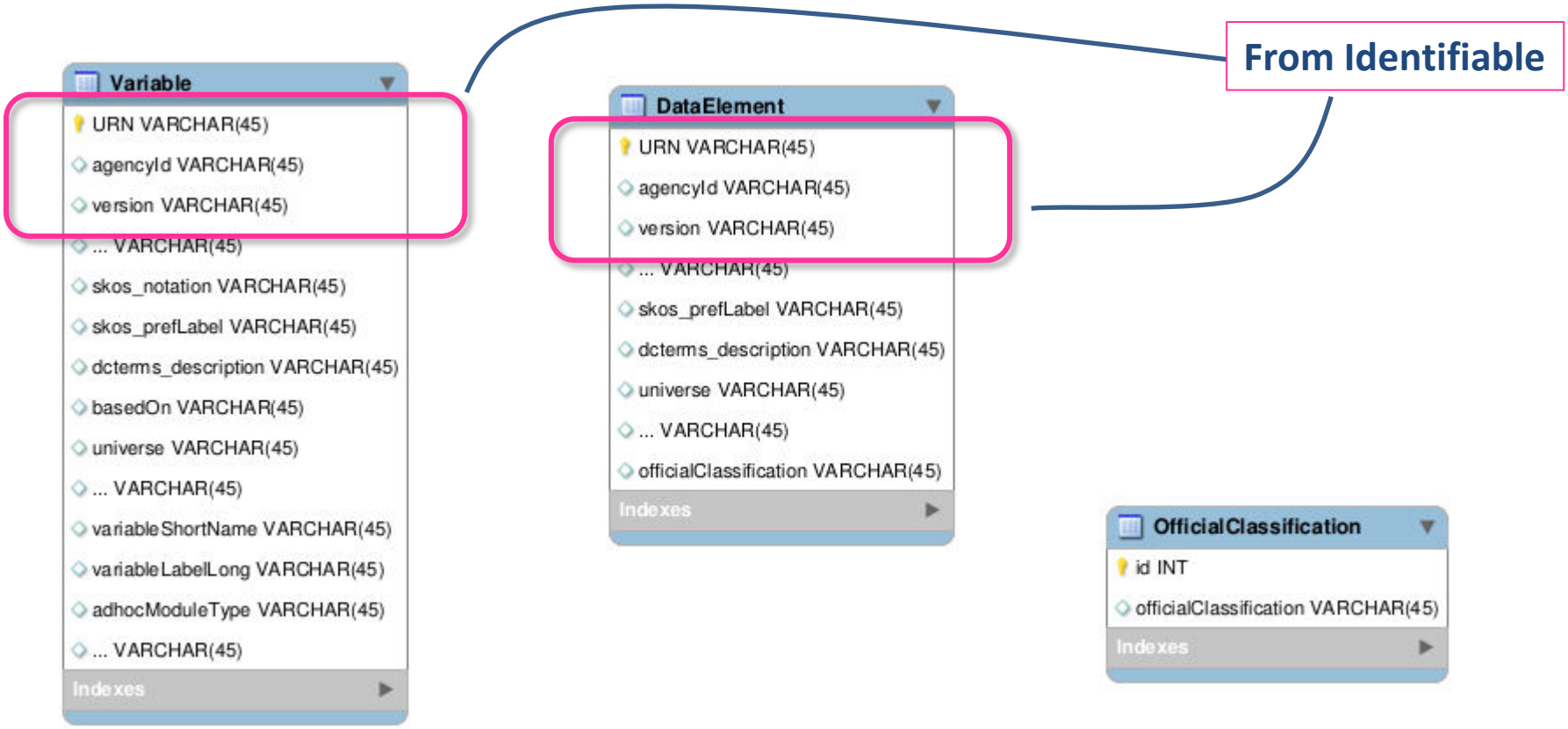


Relational-DB Example

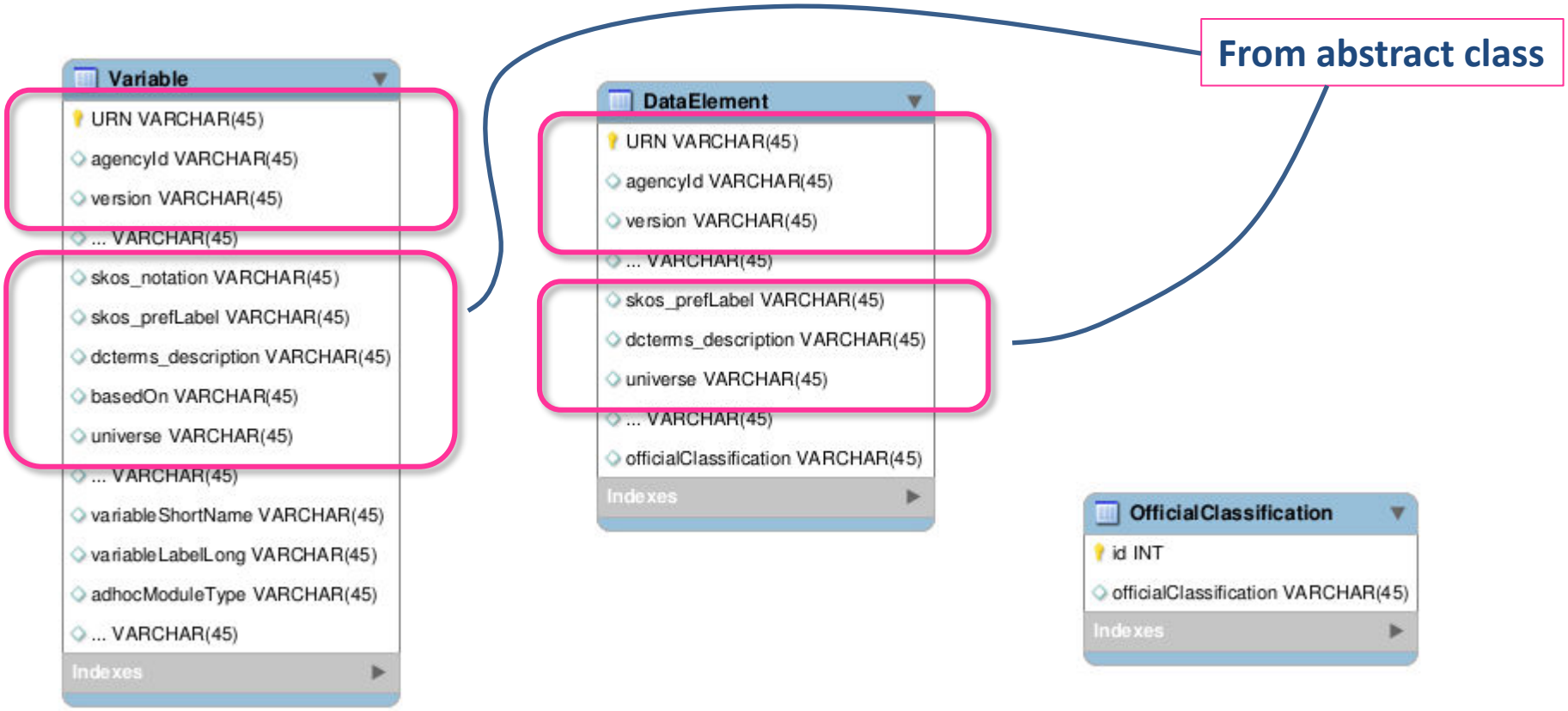




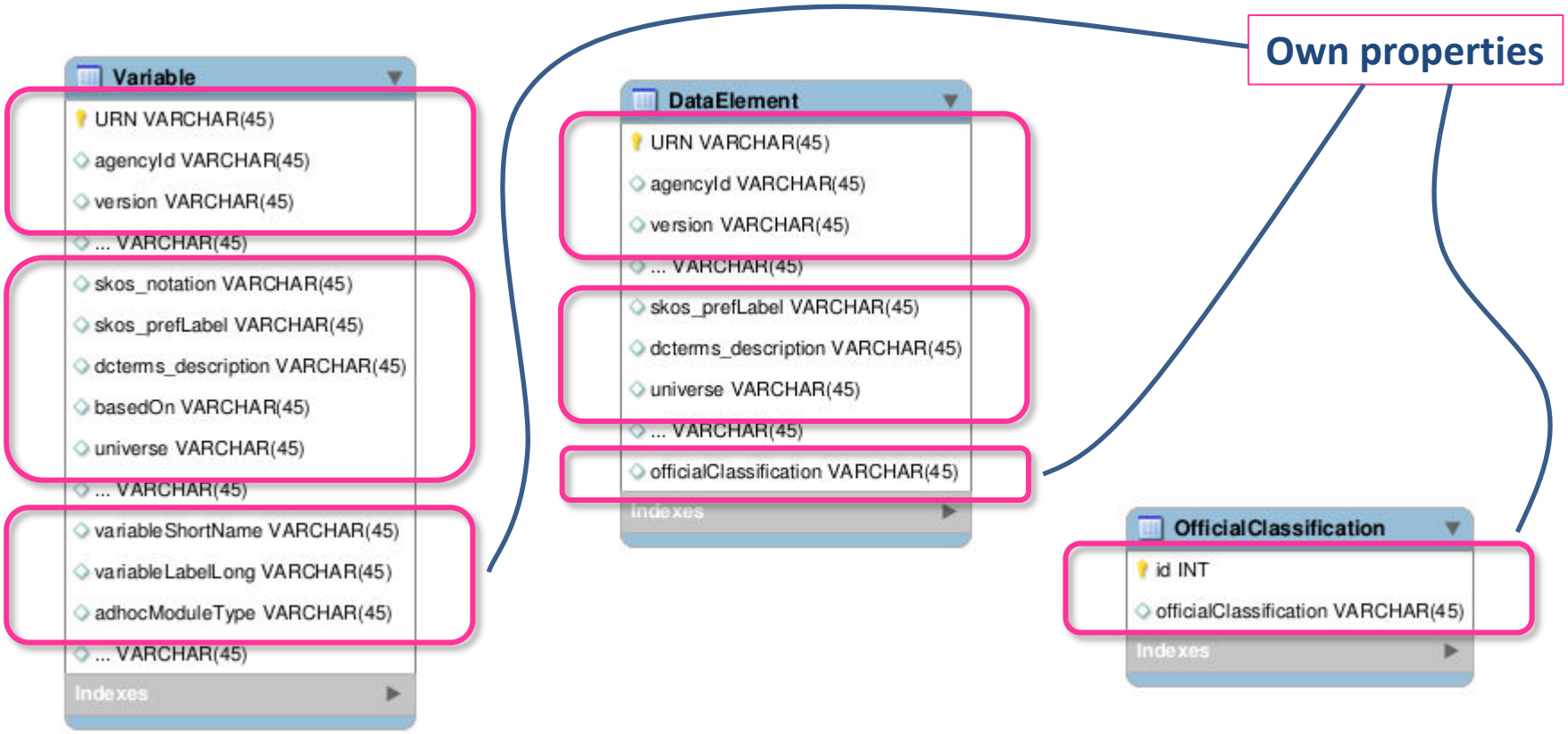
Inheritance of Properties



Inheritance of Properties



Declaration of own Properties



What comes next?

- What are you going to do with the MISSY project?
- Is there ongoing work?

Outline

- Persistence Layer
  - Programming Interface
  - Types of physical data storages
  - Examples
- Physical Data Product
- Physical Instance
- Archive
- DDIProfile

Data Storage Access

- **Outlook**
- DDI Serialization Examples
  - DDI-XML
  - DDI-RDF / disco-spec
  - Relational-DB

Data Storage

Outlook

- YES! There is ongoing work!

- Use the API in other projects, e.g. GESIS internal projects like StarDat
- Publish the persistence API as open-source on github

- Two presentations at IASSIST 2013
- One poster at IASSIST 2013

Poster

- Shows detailed information regarding Missy
  - Multitier Architecture, MVC
  - Employed Technologies
  - Specific Maven Modules
  - Project Home
- Excerpts from disco-model and the mapping of the fields

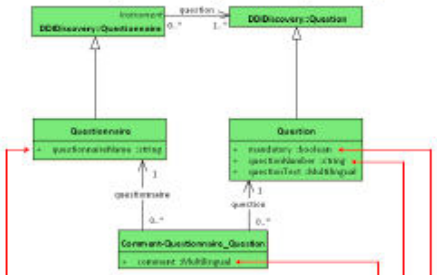
gesis
Leibniz Institute for the Social Sciences

Towards a DDI Backend Architecture

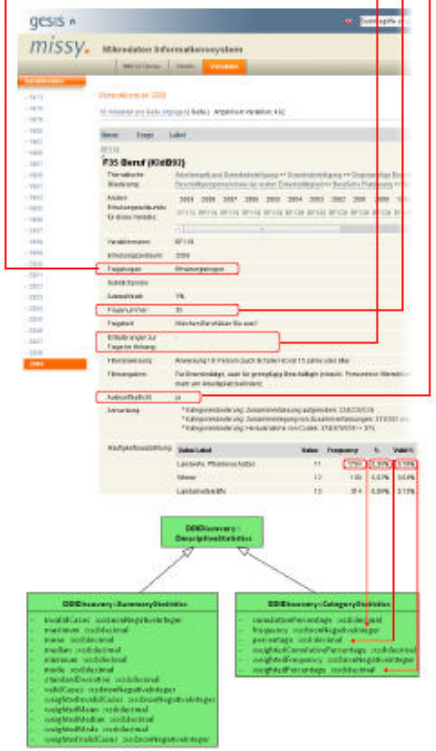
Sharing DDI-related Software Modules

Matthäus Zloch, Thomas Bosch, Dennis Wegener
{matthaeus.zloch, thomas.bosch, dennis.wegener}@gesis.org

Data Model (DDI Discovery Vocabulary + MISSY)



MISSY Web – Variable Details



Microdata Information System (MISSY)

- Study documentation on variable level
- Studies: german microcensus, EU-SILC, EU-LFS, EVS, ...

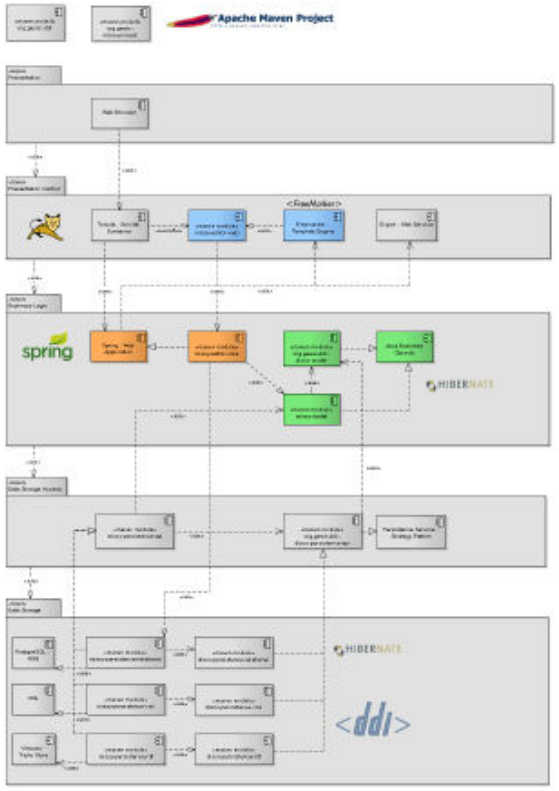
Data models

- DDI Discovery Vocabulary as a basis
- Project-specific data models

GitHub Repository: https://github.com/misay-project/disco-model-impl

Multitier Software Architecture

- Reusable modules
- Usage of divers software development patterns (e.g. MVC, strategy, DAO)



Thank you for your attention…

- Feel free to download the sources from GitHub:
  https://github.com/missy-project

- Give us feedback!
- Feel free to criticize!

Thomas Bosch
GESIS - Leibniz Institute for the Social Sciences
thomas.bosch@gesis.org
boschthomas.blogspot.com

Matthäus Zloch
GESIS - Leibniz Institute for the Social Sciences
matthaeus.zloch@gesis.org