DDI Cross-Domain Integration (DDI-CDI) Overview

Arofan Gregory, Flavio Rizzolo

DDI-CDI Working Group

EDDI 2022

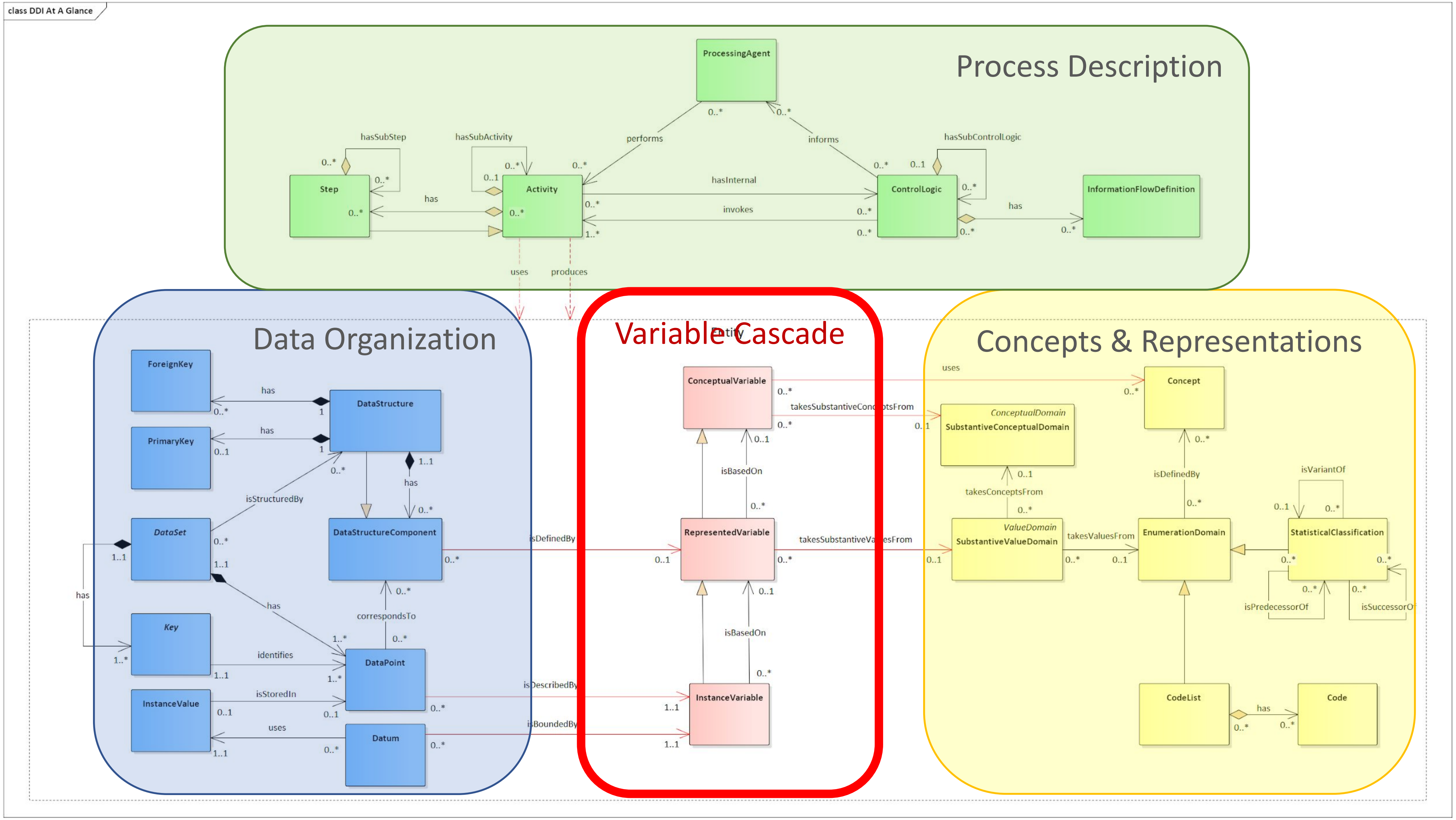
Why a New Specification?

- The volume of research data is increasing exponentially
  - New sources
  - New formats/structures
- The use of data across domain boundaries is increasing
  - “Grand challenges” (e.g., COVID-19, climate change)
  - New technologies and new approached (e.g., AI, machine learning)
- Problems of scale demand machine-actionability
  - For metadata harvesting
  - For navigating data at all levels and across domain boundries

A New DDI Specification

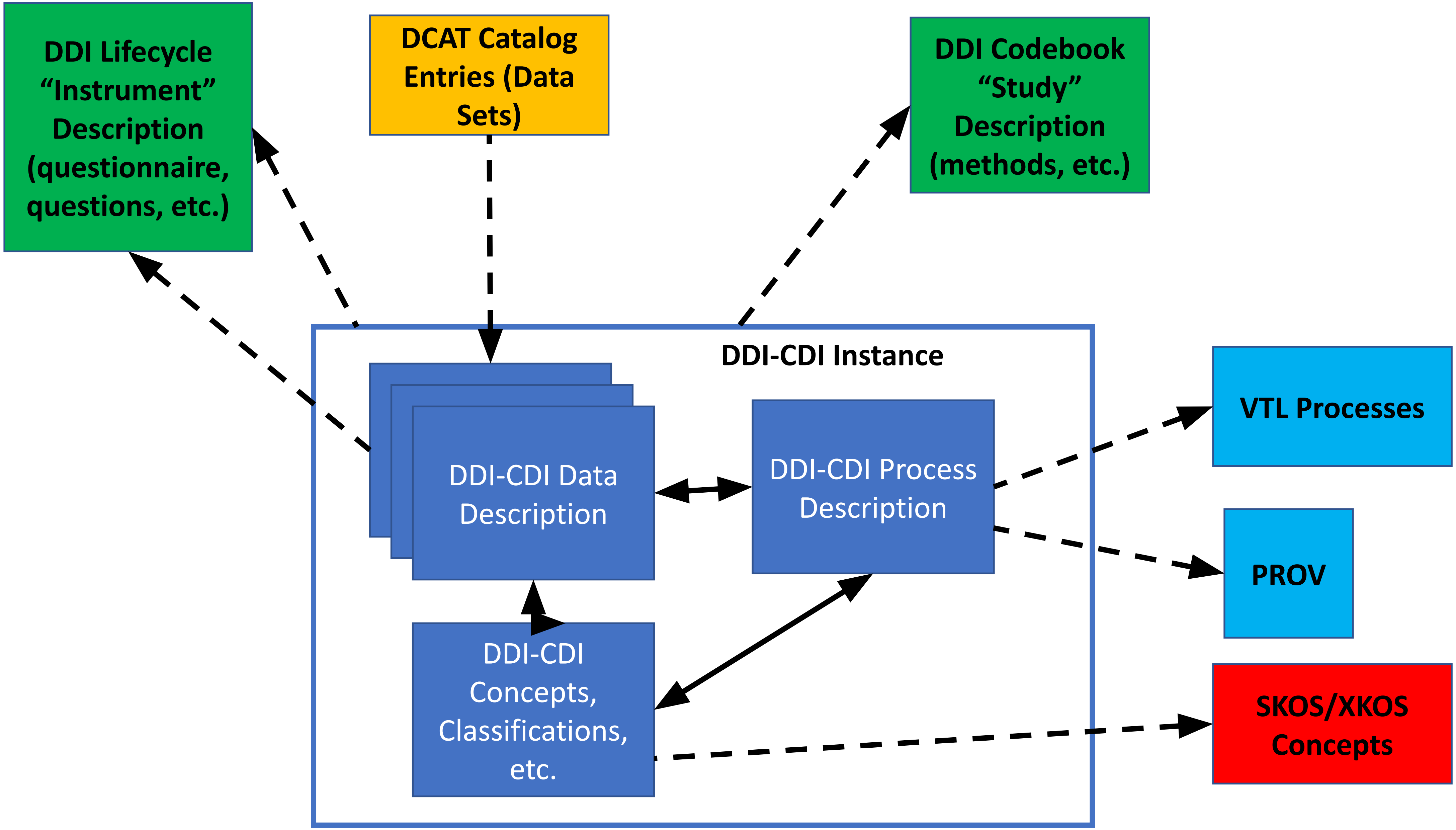
- DDI-CDI is a new specification
  - Currently in final revision
  - Release for public review in Q1 2023
- DDI-CDI is an implementation of the “DDI 4”/”DDI Moving Forward” model
  - Specific focus on cross-domain data integration
  - Model-based standard
  - XML and other syntax representations supported
  - Designed to be machine-actionable
- Complementary to other DDI specifications
  - Works with DDI Codebook and DDI Lifecycle
  - Extends metadata coverage to support integration with other domain data
  - Can work with other (non-DDI) domain metadata specifications

DDI-CDI at-a-glance

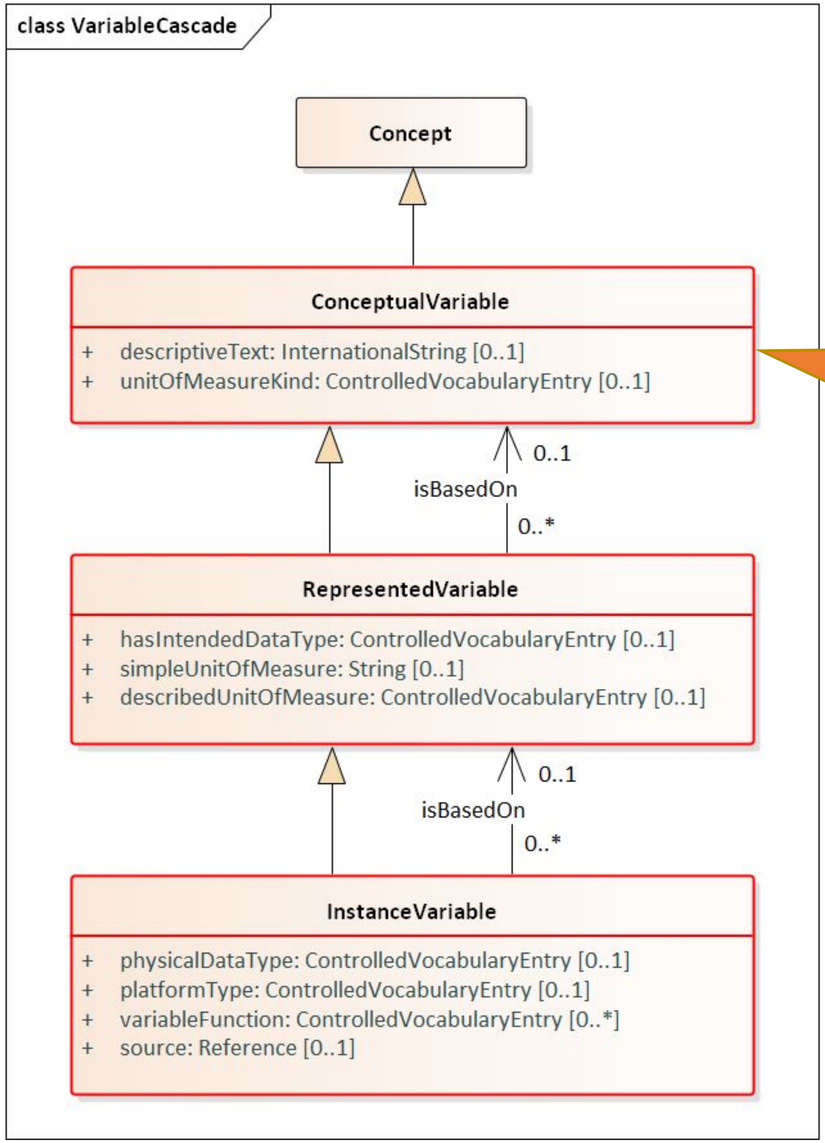


Connecting Standard Metadata Sets

- In real-world implementations, it is typical for there to be several different metadata standards used for different functions (examples):
  - DCAT/Schema.org for discoverability
  - DDI-C or DDI-L for granular data description
  - PROV for provenance
  - SKOS for concepts
  - Etc.
- These disconnected models must work together
  - DDI-CDI provides a framework for integrating this metadata
  - Allows for the native use by reference of external standards, or translation into the DDI-CDI form

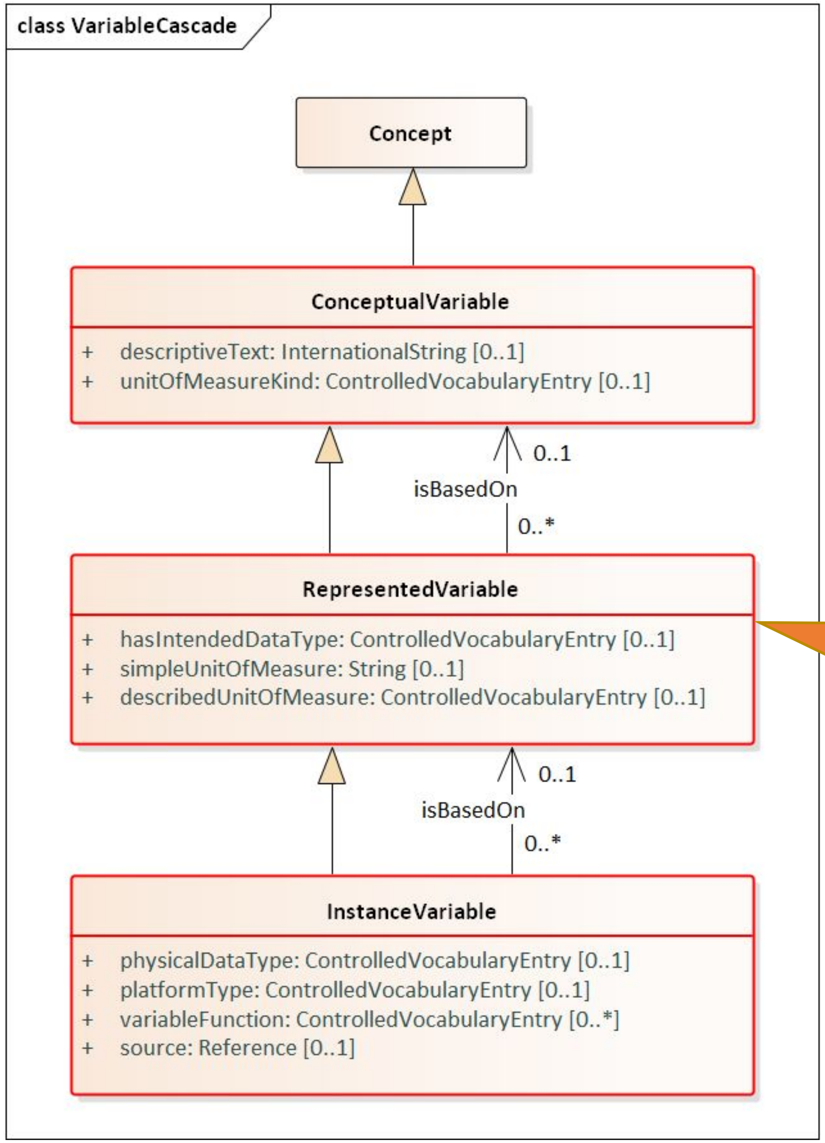

DDI Lifecycle "Instrument" Description (questionnaire, questions, etc.)
DCAT Catalog Entries (Data Sets)
DDI Codebook "Study" Description (methods, etc.)
DDI-CDI Instance
DDI-CDI Data Description
DDI-CDI Process Description
DDI-CDI Concepts, Classifications, etc.
VTL Processes
PROV
SKOS/XKOS Concepts

DDI-CDI variable cascade – Conceptual


class VariableCascade
Concept
ConceptualVariable
+ descriptiveText: InternationalString [0..1]
+ unitOfMeasureKind: ControlledVocabularyEntry [0..1]
0..1
isBasedOn
0..*
RepresentedVariable
+ hasIntendedDataType: ControlledVocabularyEntry [0..1]
+ simpleUnitOfMeasure: String [0..1]
+ describedUnitOfMeasure: ControlledVocabularyEntry [0..1]
0..1
isBasedOn
0..*
InstanceVariable
+ physicalDataType: ControlledVocabularyEntry [0..1]
+ platformType: ControlledVocabularyEntry [0..1]
+ variableFunction: ControlledVocabularyEntry [0..*]
+ source: Reference [0..1]

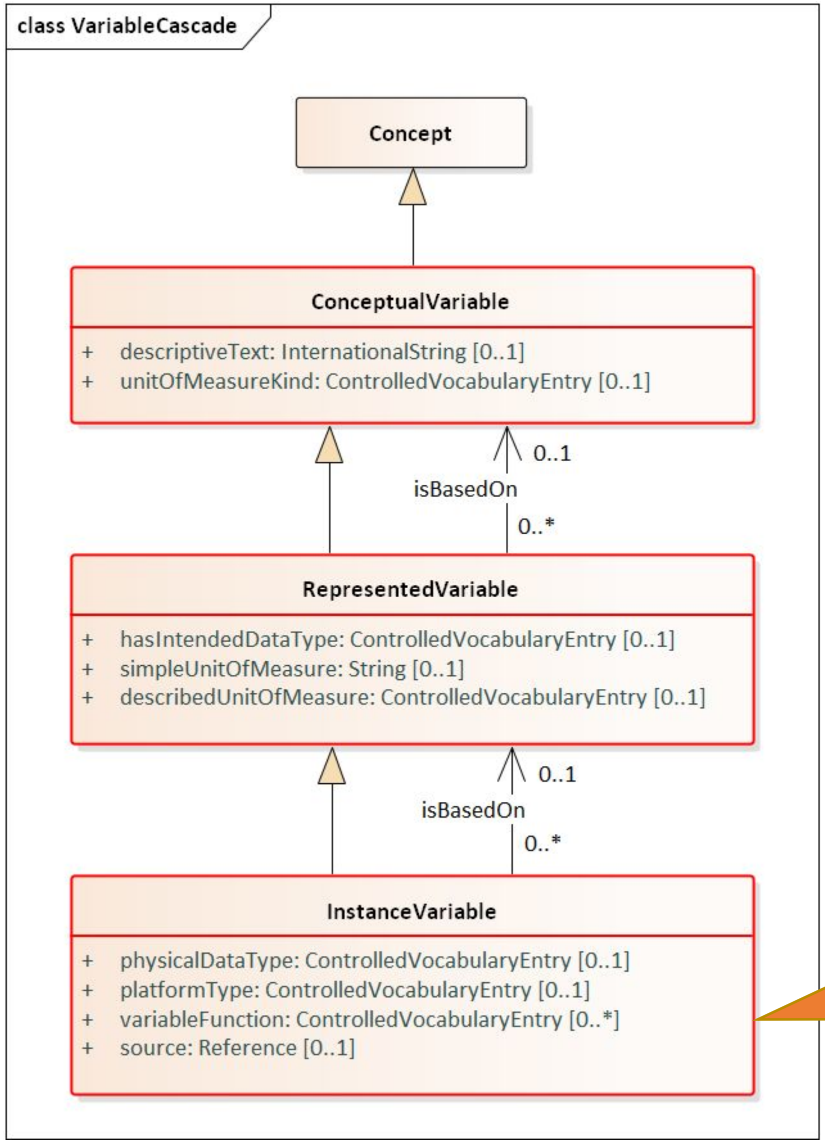
- **Variable descriptions at a high level, e.g. conceptual domains**
- **Early design data capture/ intake**
- **Broad search and discovery**
- **Least specific/Most reusable**

DDI-CDI variable cascade – Representation


class VariableCascade
Concept
ConceptualVariable
+ descriptiveText: InternationalString [0..1]
+ unitOfMeasureKind: ControlledVocabularyEntry [0..1]
0..1
isBasedOn
0..*
RepresentedVariable
+ hasIntendedDataType: ControlledVocabularyEntry [0..1]
+ simpleUnitOfMeasure: String [0..1]
+ describedUnitOfMeasure: ControlledVocabularyEntry [0..1]
0..1
isBasedOn
0..*
InstanceVariable
+ physicalDataType: ControlledVocabularyEntry [0..1]
+ platformType: ControlledVocabularyEntry [0..1]
+ variableFunction: ControlledVocabularyEntry [0..*]
+ source: Reference [0..1]

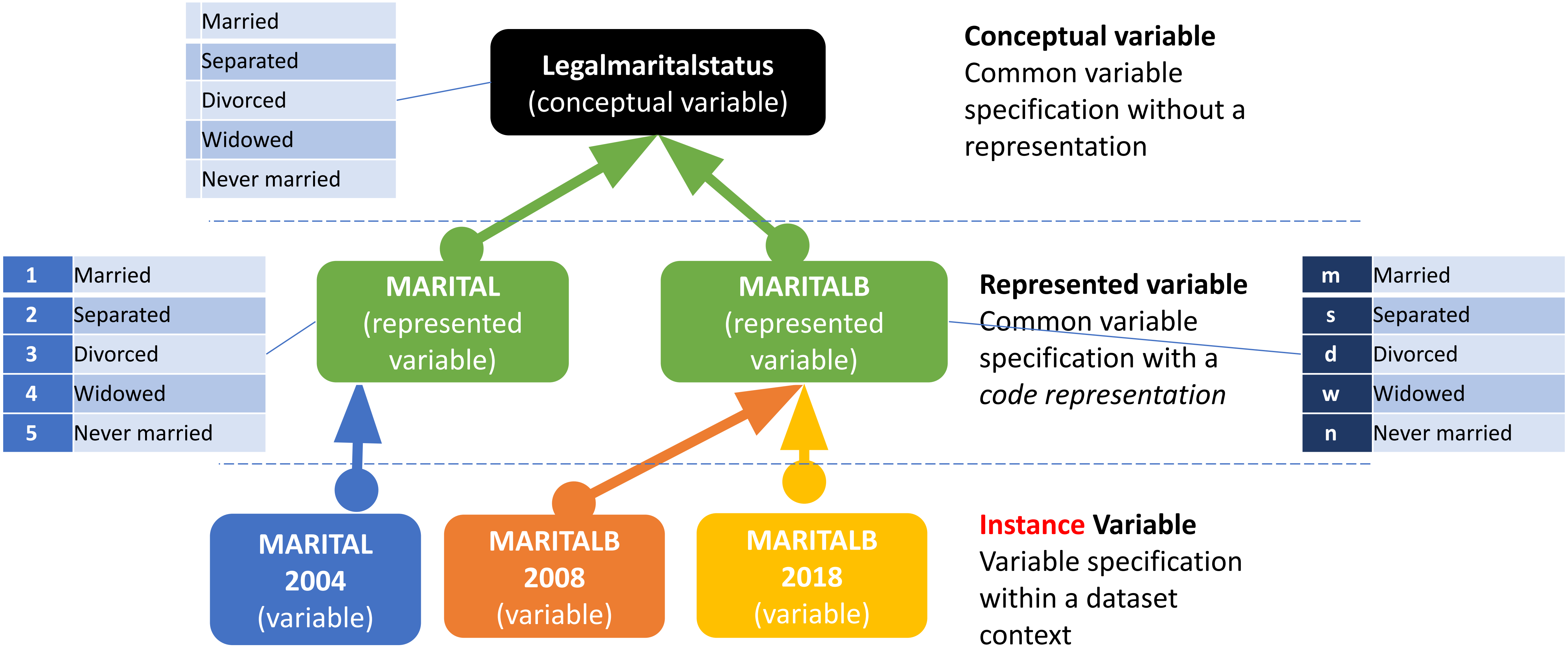
- **Variable descriptions at a detailed level, e.g. value domains**
- **Advanced design for all stages of data lifecycle**
- **Specific search and discovery**
- **More specific/Less reusable**

DDI-CDI variable cascade – Instance


class VariableCascade
Concept
ConceptualVariable
+ descriptiveText: InternationalString [0..1]
+ unitOfMeasureKind: ControlledVocabularyEntry [0..1]
0..1
isBasedOn
0..*
RepresentedVariable
+ hasIntendedDataType: ControlledVocabularyEntry [0..1]
+ simpleUnitOfMeasure: String [0..1]
+ describedUnitOfMeasure: ControlledVocabularyEntry [0..1]
0..1
isBasedOn
0..*
InstanceVariable
+ physicalDataType: ControlledVocabularyEntry [0..1]
+ platformType: ControlledVocabularyEntry [0..1]
+ variableFunction: ControlledVocabularyEntry [0..*]
+ source: Reference [0..1]

- **Physical data description, e.g. physical data types**
- **Use of a variable in specific data instances**
- **Data search and discovery**
- **No reusable**

Example: comparability and traceability



Married
Separated
Divorced
Widowed
Never married

1	Married
2	Separated
3	Divorced
4	Widowed
5	Never married

m	Married
s	Separated
d	Divorced
w	Widowed
n	Never married

Conceptual variable
Common variable specification without a representation

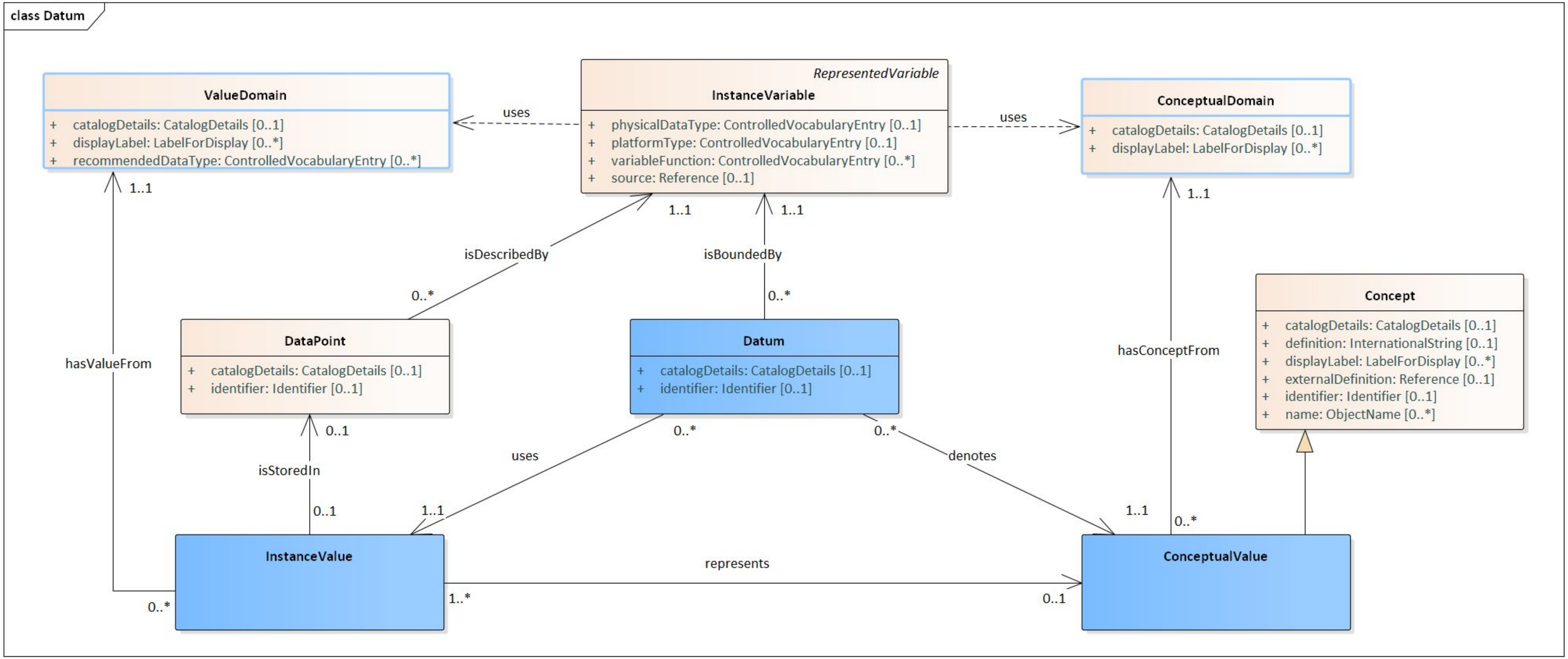
Represented variable
Common variable specification with a *code representation*

Instance Variable
Variable specification within a dataset context

Application: comparability and traceability

- Two variables in different data sets might:
  - Measure the same concept differently
  - Measure the same concept in the same way with different physical representations
  - Exist identically in two data sets, but with no formal link
- In all of these cases, understanding the variables at each level (conceptual, representational, and actual) provides a strong basis for programmatically identifying them as potential points for joining data sets

Datums – Holding data, variables and concepts together



class Datum

ValueDomain
- + catalogDetails: CatalogDetails [0..1]
- + displayLabel: LabelForDisplay [0..*]
- + recommendedDataType: ControlledVocabularyEntry [0..*]

RepresentedVariable
InstanceVariable
- + physicalDataType: ControlledVocabularyEntry [0..1]
- + platformType: ControlledVocabularyEntry [0..1]
- + variableFunction: ControlledVocabularyEntry [0..*]
- + source: Reference [0..1]

ConceptualDomain
- + catalogDetails: CatalogDetails [0..1]
- + displayLabel: LabelForDisplay [0..*]

Concept
- + catalogDetails: CatalogDetails [0..1]
- + definition: InternationalString [0..1]
- + displayLabel: LabelForDisplay [0..*]
- + externalDefinition: Reference [0..1]
- + identifier: Identifier [0..1]
- + name: ObjectName [0..*]

DataPoint
- + catalogDetails: CatalogDetails [0..1]
- + identifier: Identifier [0..1]

Datum
- + catalogDetails: CatalogDetails [0..1]
- + identifier: Identifier [0..1]

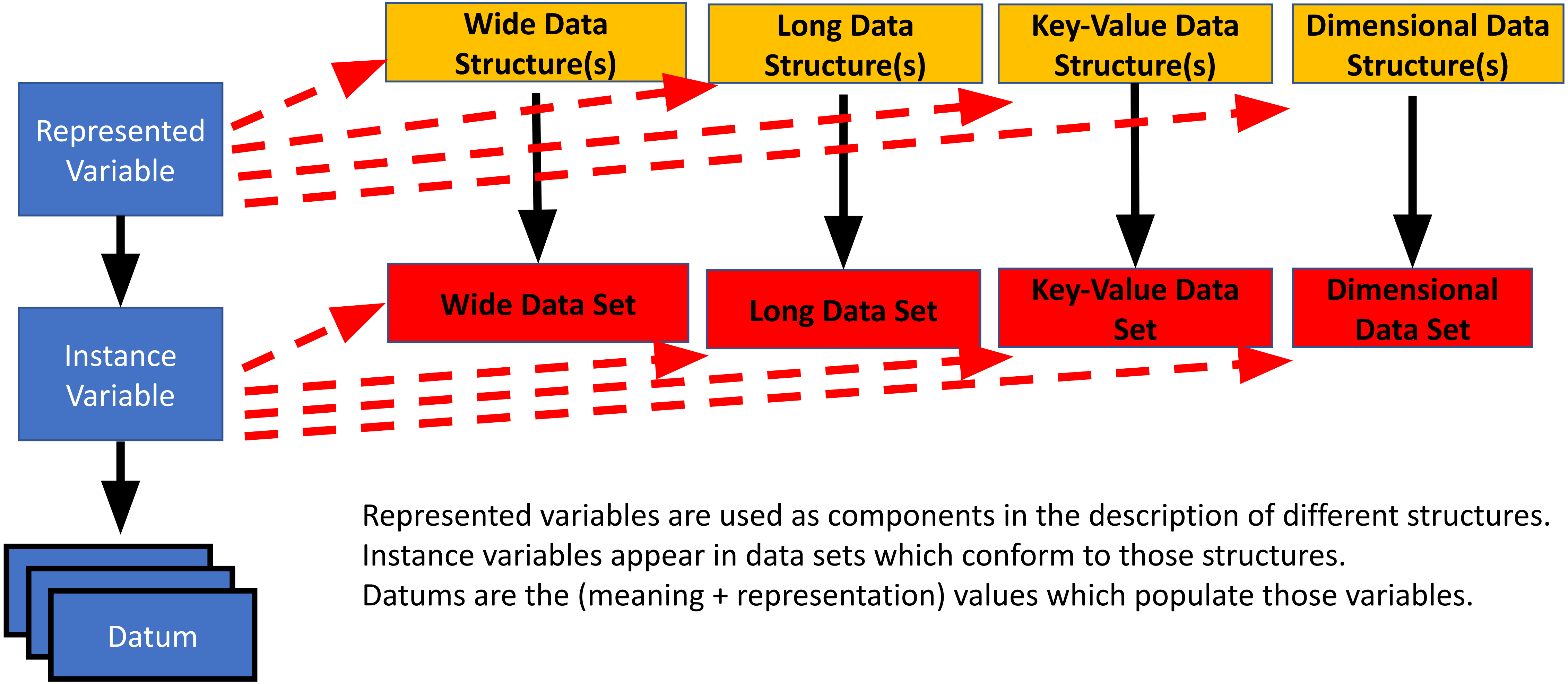
InstanceValue

ConceptualValue

DDI-CDI data description – Data structures

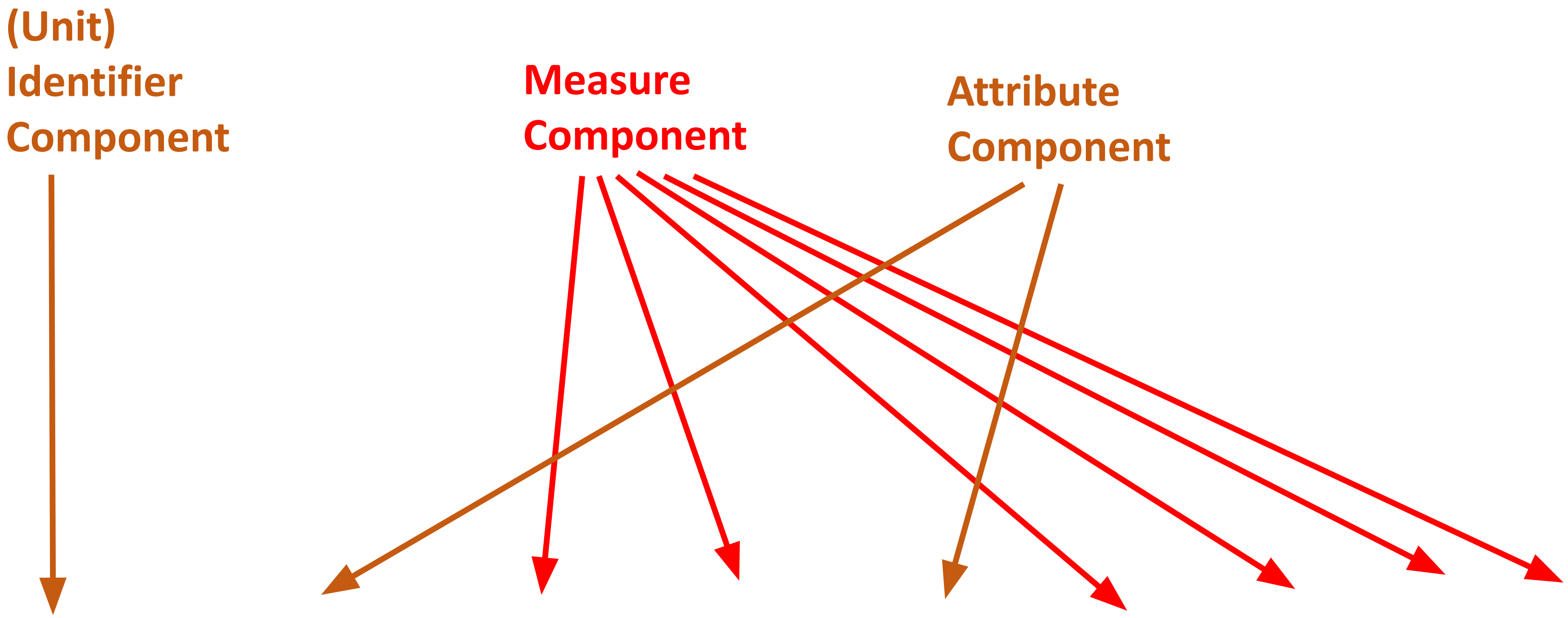
- DDI-CDI can describe four different data structures
  - Wide – as with unit records
  - Long – as with event or stream data
  - Key value – as in a key-value store
  - Dimensional – as with aggregate data

Datums and Variables: Reuse in Different Structures/Data Sets



Represented variables are used as components in the description of different structures.
Instance variables appear in data sets which conform to those structures.
Datums are the (meaning + representation) values which populate those variables.

Example 1: data in wide form



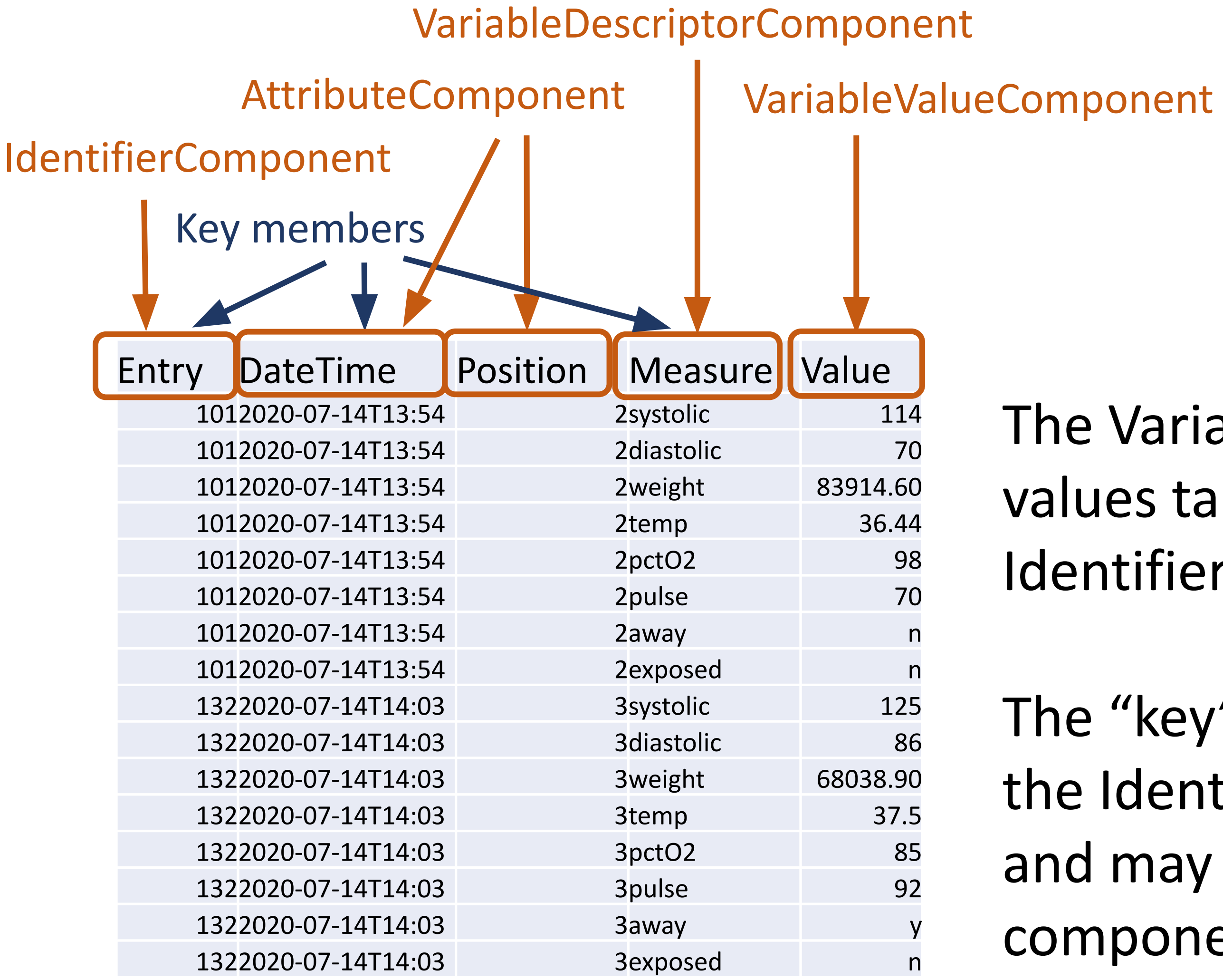
(Unit) Identifier Component

Measure Component

Attribute Component

entry	datetime	systolic	diastolic	position	weight	temp	pctO2	pulse
101	2020-07-14T13:54	114	70	2	83914.6	36.44	98	70
132	2020-07-14T14:03	125	86	3	68038.9	37.5	85	92

Example1: data in long form



Entry	DateTime	Position	Measure	Value
101	2020-07-14T13:54	2	systolic	114
101	2020-07-14T13:54	2	diastolic	70
101	2020-07-14T13:54	2	weight	83914.60
101	2020-07-14T13:54	2	temp	36.44
101	2020-07-14T13:54	2	pctO2	98
101	2020-07-14T13:54	2	pulse	70
101	2020-07-14T13:54	2	away	n
101	2020-07-14T13:54	2	exposed	n
132	2020-07-14T14:03	3	systolic	125
132	2020-07-14T14:03	3	diastolic	86
132	2020-07-14T14:03	3	weight	68038.90
132	2020-07-14T14:03	3	temp	37.5
132	2020-07-14T14:03	3	pctO2	85
132	2020-07-14T14:03	3	pulse	92
132	2020-07-14T14:03	3	away	y
132	2020-07-14T14:03	3	exposed	n

The Variable Descriptor Component has values taken from the list of non-Unit Identifiers in the wide data set.

The “key” for each value is composed from the Identifier and the Variable Descriptor, and may include non-transposed components, e.g. DateTime.

Application: cross domain integration

- Integrating data across domains involves both dealing with different kinds of discipline's structures and vocabularies
    - Sensor data streams in tall structures
    - Survey data in wide structures
    - Administrative summary data in cubes
- A standard also needs to be discipline agnostic.
    - Vocabularies need to be referenced, not built in
        - (e.g. "question")
- A standard needs to be able to at least reference metadata in other disciplines standards.
    - This, of course, presents challenges for machine actionability.

Status

- Almost-final draft
- Browsable field-level documentation and syntax representations: https://ddi-alliance.bitbucket.io/DDI-CDI/DDI-CDI_2022-10-06/doc/_build/index.html
- Beta-level implementation at UKDA (see Tools session)
- Public review will include a webinar to explain the review process and the specification