# DDI-CDI Overview

EDDI, December 2024

Arofan Gregory

Chair, DDI-CDI WG

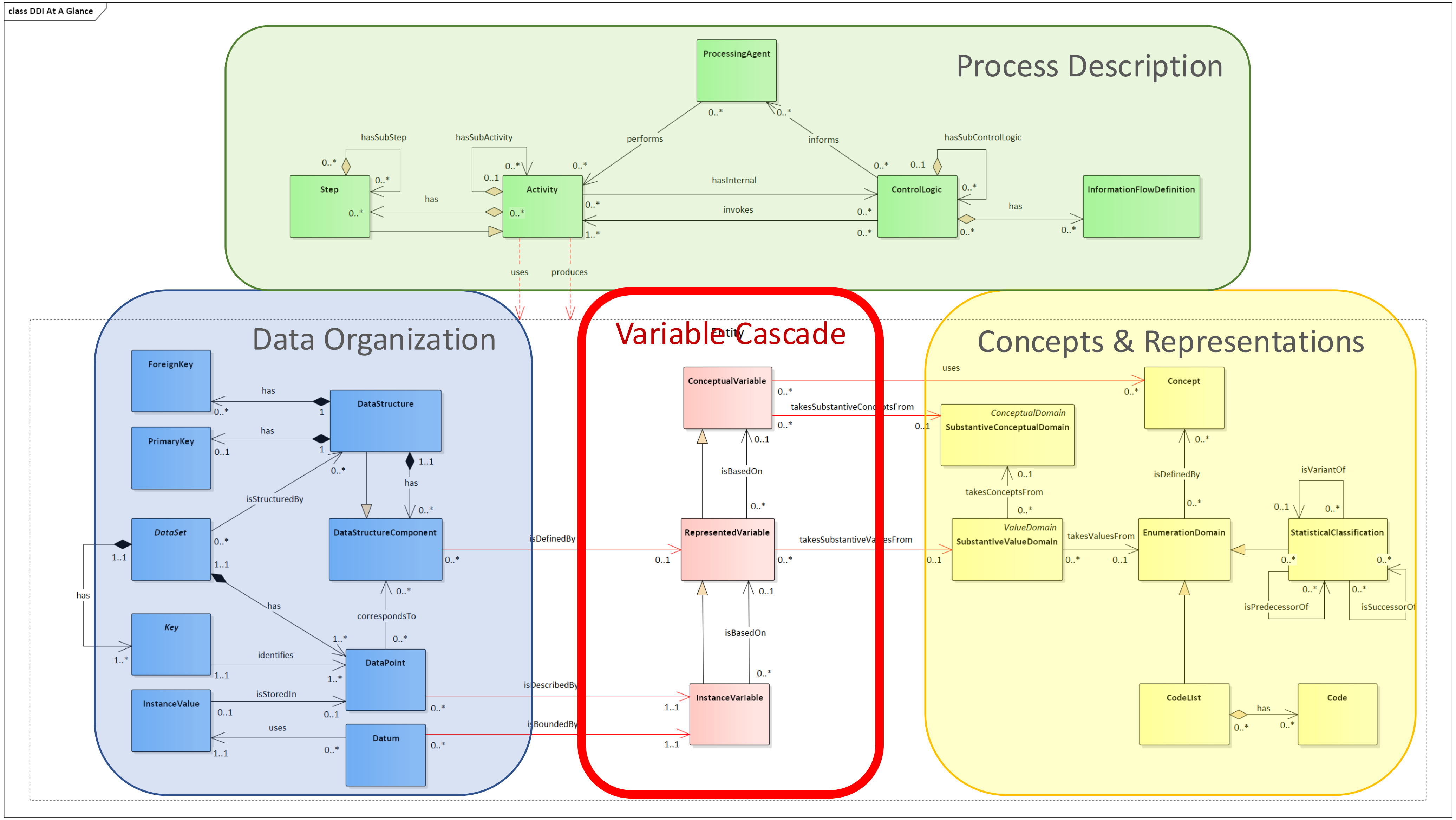
# A New DDI Specification!

- DDI-CDI is a new specification
    - Vote just completed – 1.0 specification approved
    - Now in publication process – not yet officially released
- DDI-CDI is an implementation of the “DDI 4”/”DDI Moving Forward” model
    - Specific focus on cross-domain data integration
    - Model-based standard – Canonical XMI describes UML model
    - XML and other syntax representations supported (JSON-LD and Turtle)
    - 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 directly with other (non-DDI) domain metadata specifications (esp. RDF)
- Designed to be implemented using subsets (“profiles”)

# DDI-CDI Functionality

- Structural description across diverse sources/types of data
  - Wide (rectangular) data
  - Tall (long), event data, sensors
  - Key-vale data (big data, No SQL data)
  - Multi-dimensional data cubes, indicators, time series
- Describes provenance of data between different structures/forms
  - Processing framework
  - Relies on other standards (PROV-O, SDTL, etc.)
- Describes data at an atomic level
  - Variables
  - Datums/cells

# DDI-CDI at-a-glance



Courtesy of the amazing Flavio Rizzolo, Statistics Canada

# Specification and Documentation

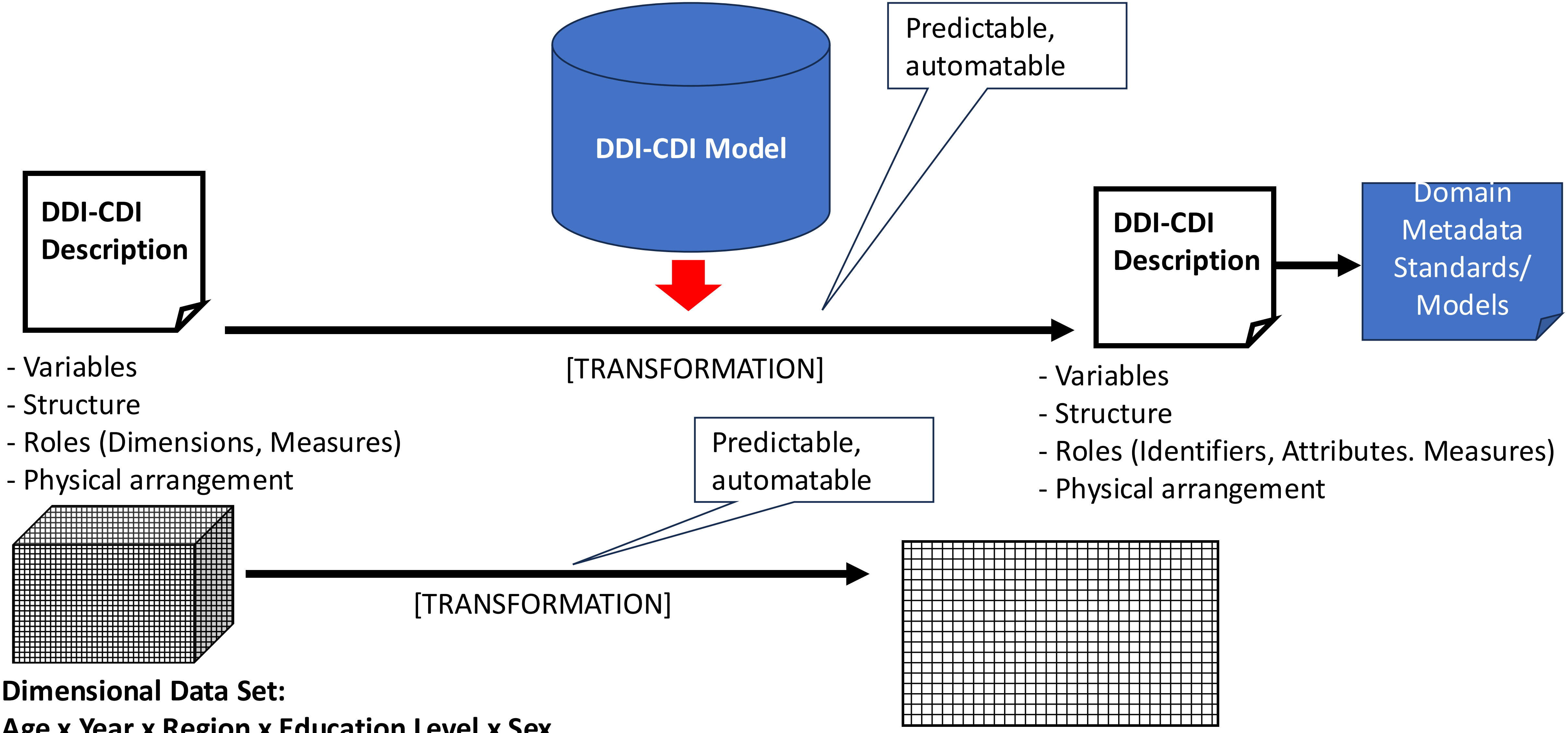
- High-level specification: https://github.com/ddi-cdi/ddi-cdi/blob/main/build/high-level-documentation/DDI-CDI_Model_Specification.pdf
- Full specification download: https://github.com/ddi-cdi/ddi-cdi/archive/refs/tags/v1.0-rc3.zip
- Field-level documentation: https://ddi-cdi.github.io/ddi-cdi_v1.0-rc3/field-level-documentation/

# What Is the Purpose?

- To enable data disseminators to describe the data they provide in a way that makes it comprehensible to those from other domains
- This often requires a re-structuring of the data to agree with a target data set to be integrated with
  - E.g., using existing environmental data from a domain-external source (EEA air quality) to provide a context for domain data (the European Social Survey)
  - Sensor data is re-structured to be integrated with survey data (it is also processed to create temporal and geographical alignments, but this is in additional to needed structural transformations)
- DDI-CDI allows “data sets” to be processed *as if* they were a pool of atomic pieces to be re-combined
  - The provider does not need to atomize their data
  - The receiver can still perform re-combination
- Bridges the gap between data structures and logical/conceptual content

# Summarizing the Problem

- Data providers cannot anticipate all of the capabilities and requirements of data users
- Data providers understand (and can describe) their data as *they* understand it
- The provided data structures contain *implicit* information about the values which comprise them
  - Data users may not understand this implicit knowledge
- DDI-CDI makes this information understandable across structures by giving a standard model of how this implicit structural information is expressed across structural types
- The provider describes what they understand – the user can derive what they need from this description
- Requires that there be a strong focus on *variables* and *datums*
- *NOTE:* RDF descriptions of individual datums can be used to perform this re-combining function, but fail the "practicality" test:
  - It is difficult to manage vast pools of individual observations
  - The "bibliographic" culture of data dissemination discourages this level of data management: everyone understands "data sets"



DDI-CDI Model

Predictable, automatable

DDI-CDI Description

- Variables
- Structure
- Roles (Dimensions, Measures)
- Physical arrangement

[TRANSFORMATION]

DDI-CDI Description

Domain Metadata Standards/ Models

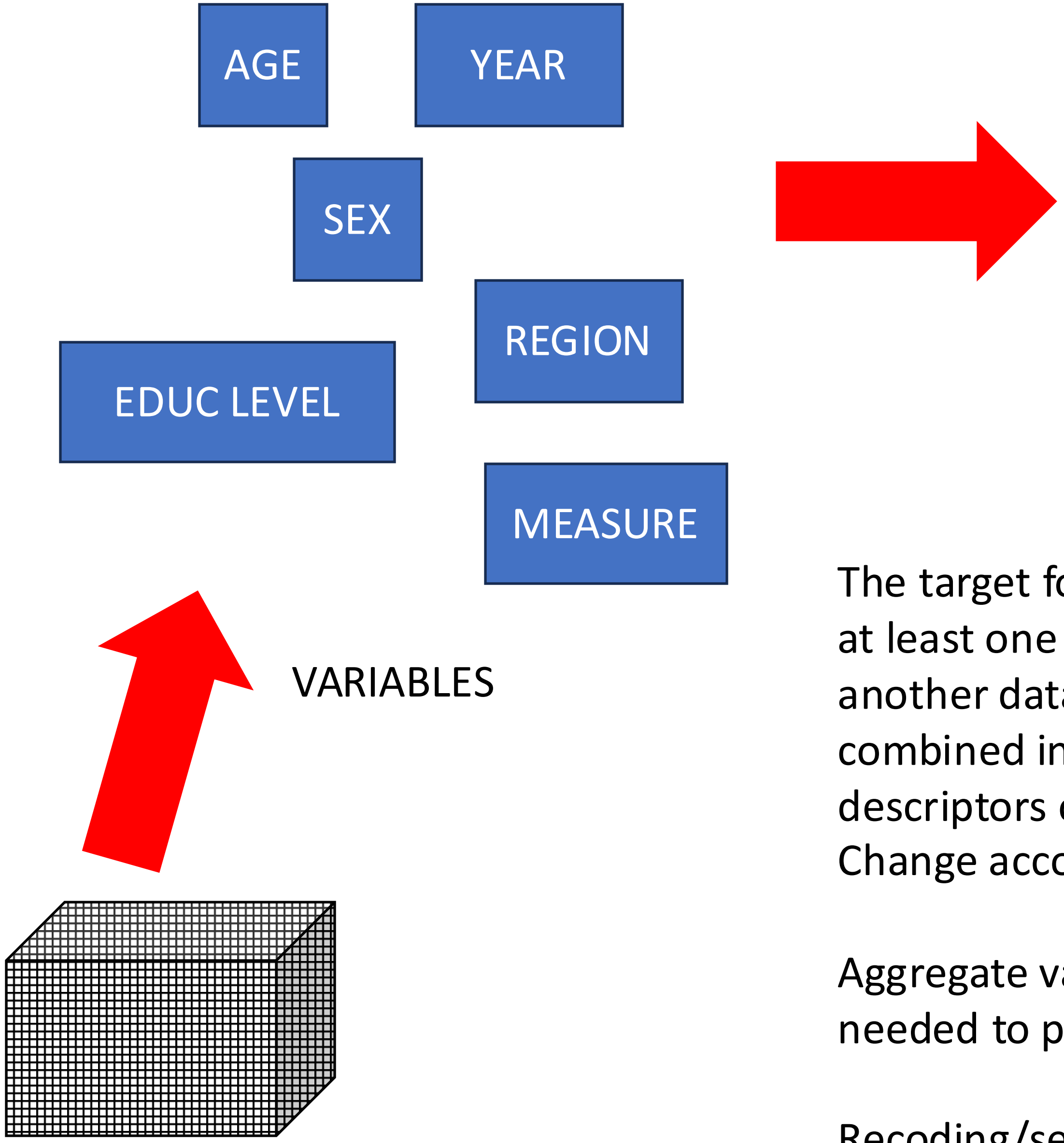
- Variables
- Structure
- Roles (Identifiers, Attributes. Measures)
- Physical arrangement

Predictable, automatable

[TRANSFORMATION]

**Dimensional Data Set:**
**Age x Year x Region x Education Level x Sex**

**Wide Data Set: Dimensions viewed as Variables (properties)**



| AGE | YEAR | SEX | REGION | EDUC LEVEL | MEASURE |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |

The target format is wide, with the dimensions treated as variables (properties): at least one must be a place where integration can be performed with another data set (typically time and geography). Other variables may be combined into compound identifiers, or may be treated as additional descriptors or measures, depending on what they are. The roles of variables Change according to the structures – the meanings/values do not.

Aggregate values would be repeated to align with micro-data records as needed to provide the "complete" records for analysis.

Recoding/semantic transformations are handled separately as appropriate to the structural re-arrangement.

# Alignment/Combination with other Standards

- DDI-CDI is designed to work in the context of other standards
- We are looking at:
    - Schema.org (discovery)
    - DCAT (cataloguing)
    - PROV-O (provenance/process)
      SKOS/XKOS (concepts and codelists)
    - I-ADOPT/O&M
    - DDI Codebook/DDI Lifecycle
    - ODRL
- Model-driven standard (UML, in Canonical XMI) can produce different equivalent syntaxes
    - XML
    - RDF (Turtle, JSON-LD)
    - Python
    - ShEx, SHACL
    - Others
- Part of the Cross-Domain Interoperability Framework (CDIF) being developed by WorldFAIR

# DDI-CDI is “Semantics-Neutral”

- Describes structures and the *roles* played by concepts in describing the data
- The Concepts themselves can come from any type of controlled vocabulary
  - SKOS/XKOS
  - Ontologies (OWL, RDF-S, etc.)
  - Classifications, codelists, thesauri
- Semantics are *domain-specific* – *structures* are not!

# Important Upcoming Feature

- We have a very active group looking at how qualitative data can be managed alongside quantitative data
    - The model draws heavily on relevant W3C standards for annotation and “deep linking” (for segment selection)
    - The idea is that mixed-method collections could be managed as unitary *logical* ones
        - E.g., images of a patient’s brain could be held alongside relevant quantitative measures
        - Qualitative inputs to quantitative data sets could be described (like any other process)
    - Focus is on qualitative data as *data for management and reuse* – not on qualitative social science methods (etc.)

# Early Adopters of the Specification

- **UKDA:** “Product Builder” application based on description of all variables in the archive in DDI-CDI
- **ESS “Climate Neutral and Smart Cities”:** Implementation of DDI-CDI Process model in combination with DDI Lifecycle for integrated social science and climate/environment data
- **WorldFAIR “Cross-Domain Interoperability Framework” (CDIF):** Implementation of DDI-CDI as part of a package of RDF vocabularies used for cross-domain FAIR exchange of data & resource

# Tools and Projects

- **Sikt:** Stata/SPSS-to-DDI-CDI Converter, SQL Tools
- **High-Value Data Services:** Wrappering of municipal “open data” from Socrata with DDI-CDI metadata (also DDI Codebook)
- Implementation in **Harvard Dataverse**
- **UNSD:** SDMX Web Services-to-DDI-CDI prototype (part of CDIF work)
- **DDI Alliance-SDMX Alignment (UN/ECE Modern Stats):** Mapping SDMX data cubes to DDI-CDI dimensional data
- **Climate Adapt for EOSC:** Integrating data with CDIF
- **EOSC OSCARS:** X-Ray Absorption Spectroscopy
- Other **WorldFAIR+** Projects (CDIF)
- “Nectar” spreadsheet tool (**DDI Developer’s Group**)
- DDI-CDI/**Croissant** transformations

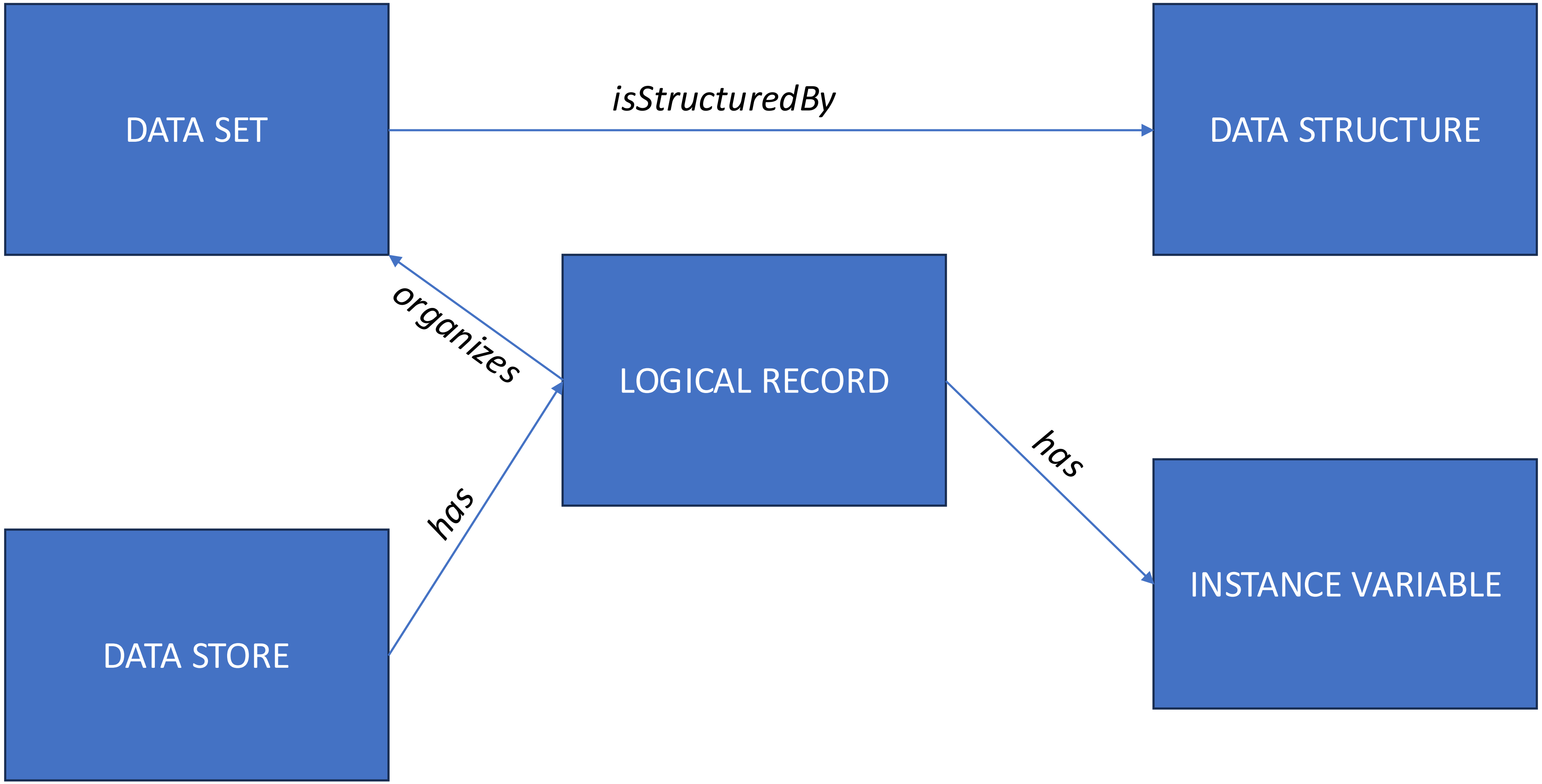
# Discussion Slides

# Brief Tour of Core Data Description Features

- Data sets, logical records, and variables
- The Variable Cascade
- Wide Data Structures
- Long Data Structures
- Dimensional Data Structures
- Physical Representation of Data

- Here we use the CDIF profile as an example, which does not show all of the available classes, and makes a very restricted use of the variable cascade
  - Requires only Instance Variables, and allows for optional use of Represented Variables (Instance Variables can be used in their stead as they are a sub-class of Represented Variables in the model)
- Other profiles employ more of the variable cascade

# DESCRIBE THE DATA SET OR SERVICE



DATA SET

*isStructuredBy*

DATA STRUCTURE

*organizes*

LOGICAL RECORD

*has*

DATA STORE

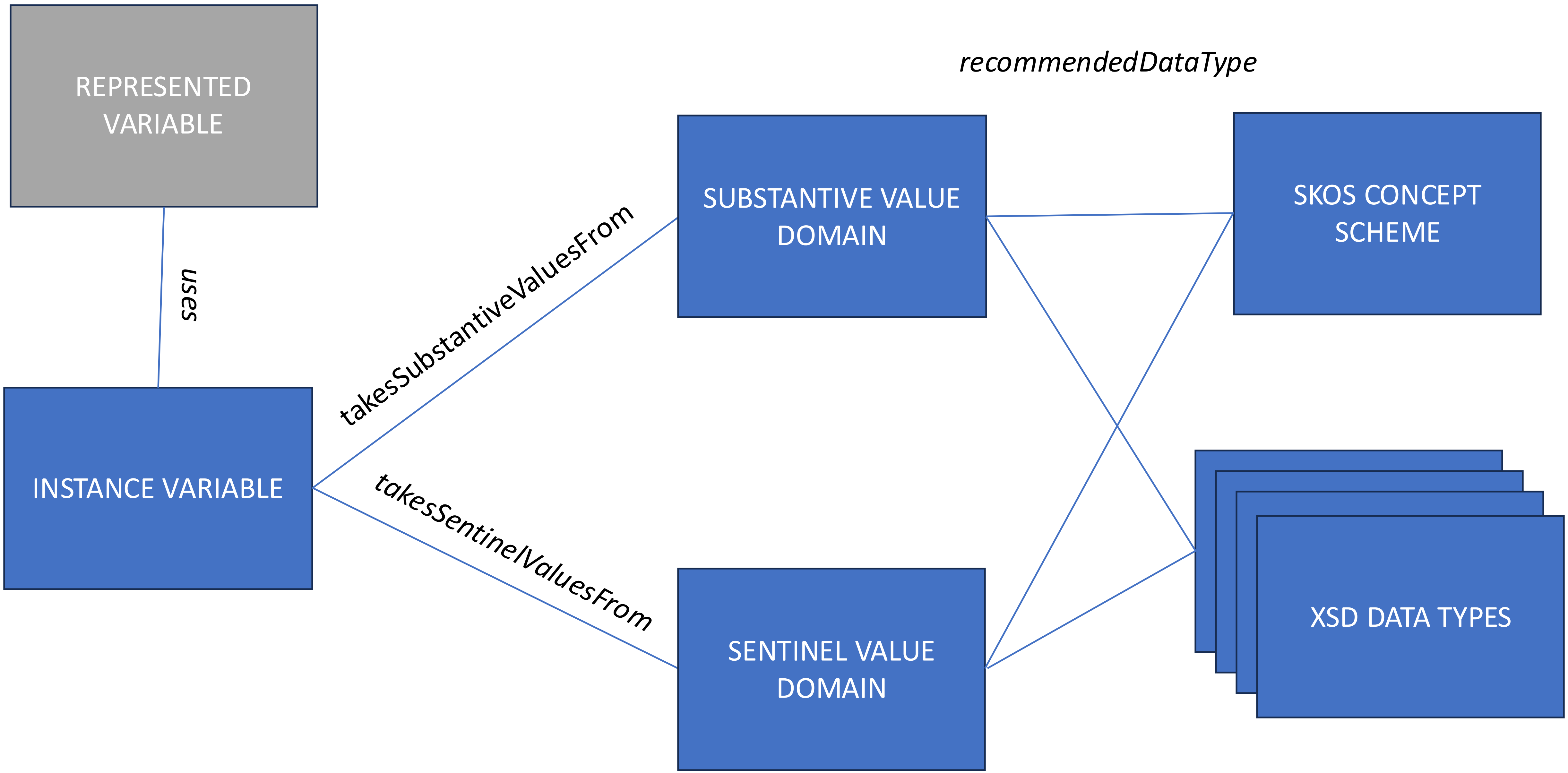
*has*

INSTANCE VARIABLE

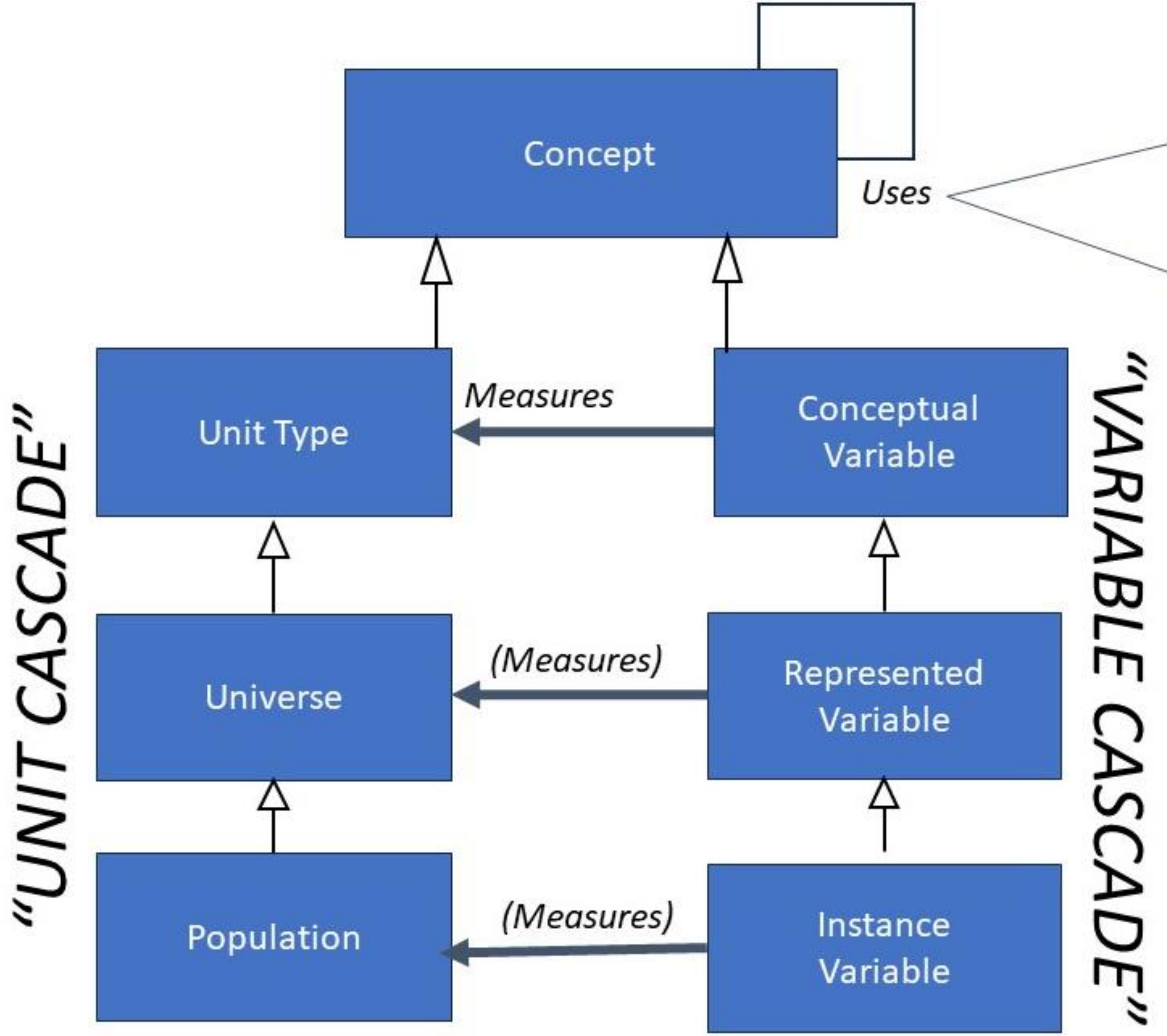
# Variables and the Variable Cascade

- A variable is defined by a (formal) Concept
  - Together, these form a reusable metadata construct termed a “Conceptual Variable”
- Variables have representations
  - May be a data type
  - May be categorical (typically using a code list of some type)
  - A Conceptual Variable with added description of its representation is a reusable construct termed a “Represented Variable”
- Variables appear in data as the definition of sets of values (in wide data, a column, etc.)
  - They are templates for a sub-set of the values provided
  - They are non-reusable applications of Represented Variables, and are termed “Instance Variables”
- These different types of variables can be associated with the real-world subject of observation, at different levels of specificity (unit type, universe, population)

# DESCRIBE THE VARIABLES



# The Variable Cascade in the  Model



Concept

Uses

Unit Type

Measures

Conceptual Variable

Universe

(Measures)

Represented Variable

Population

(Measures)

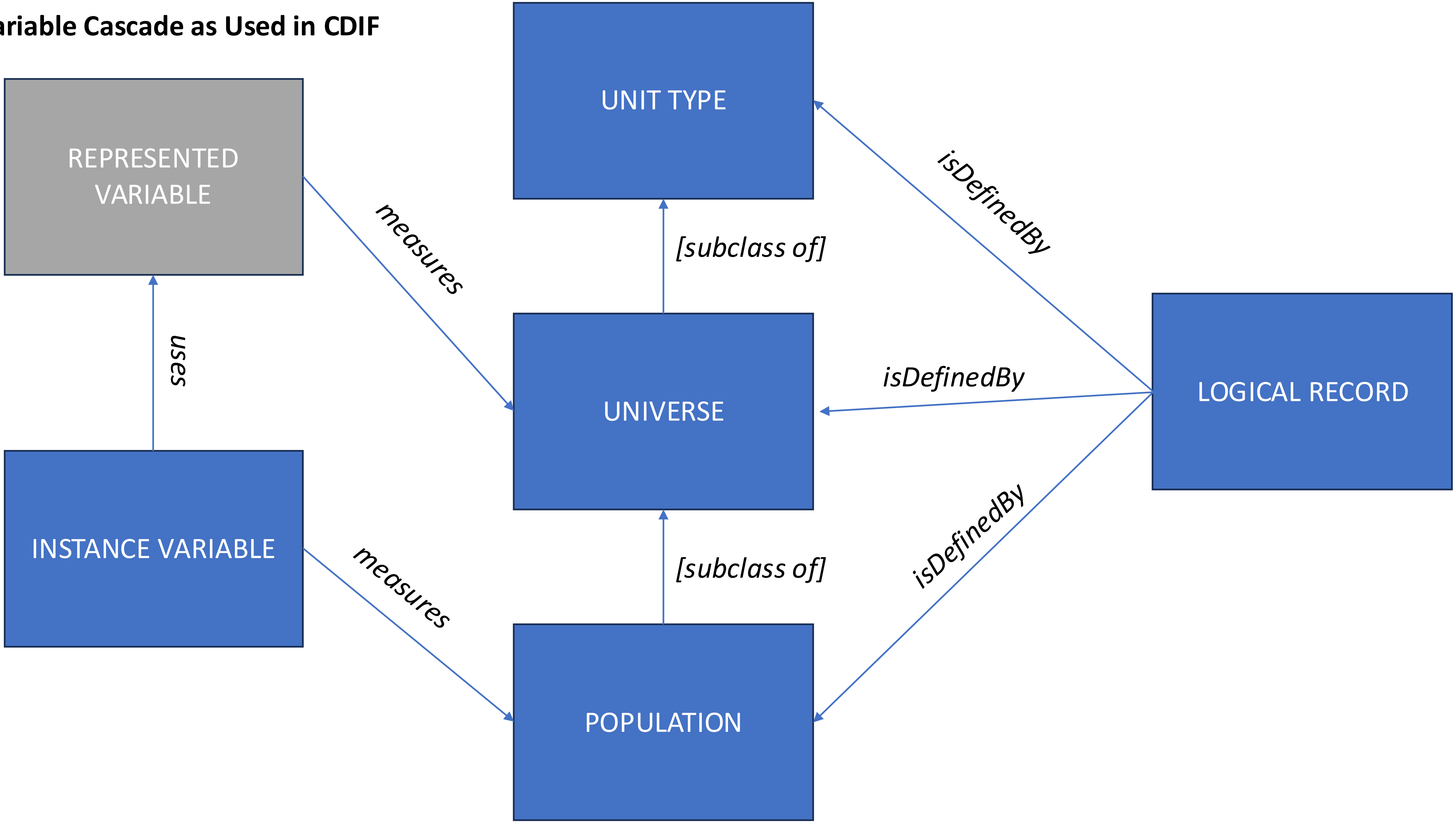
Instance Variable

“UNIT CASCADE”

“VARIABLE CASCADE”

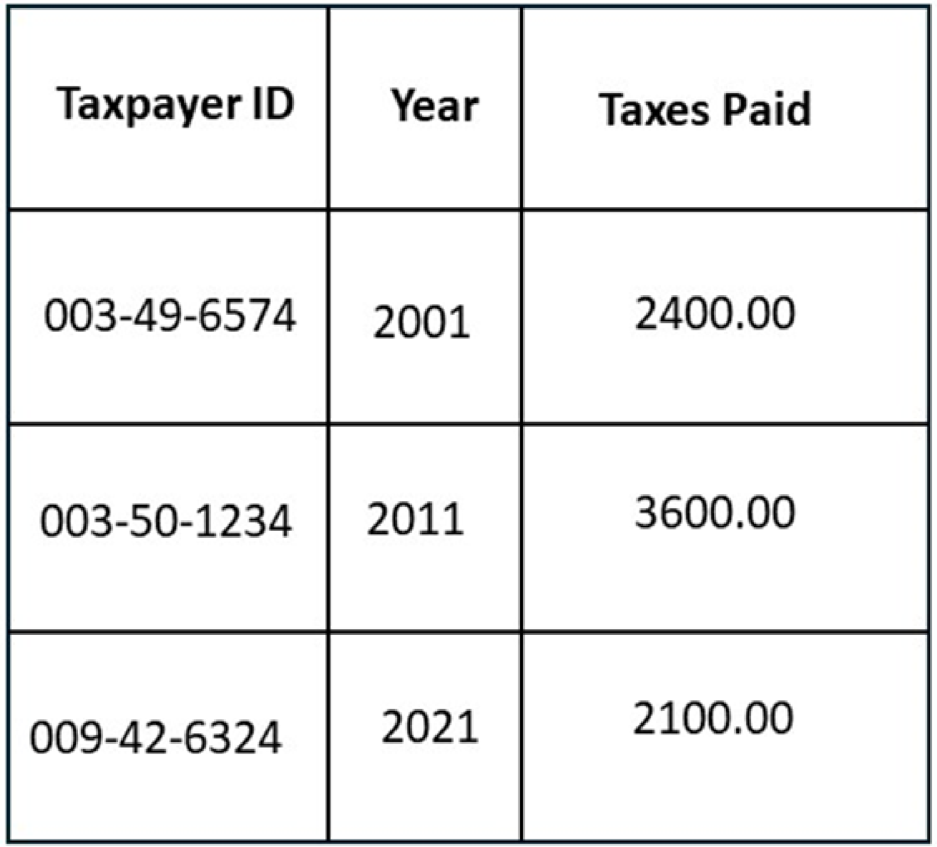
The “uses” association depicts a set of relationships between concepts at different levels of the cascade on both sides, but is not used between them. It is inherited by all classes in this picture.

## Variable Cascade as Used in CDIF



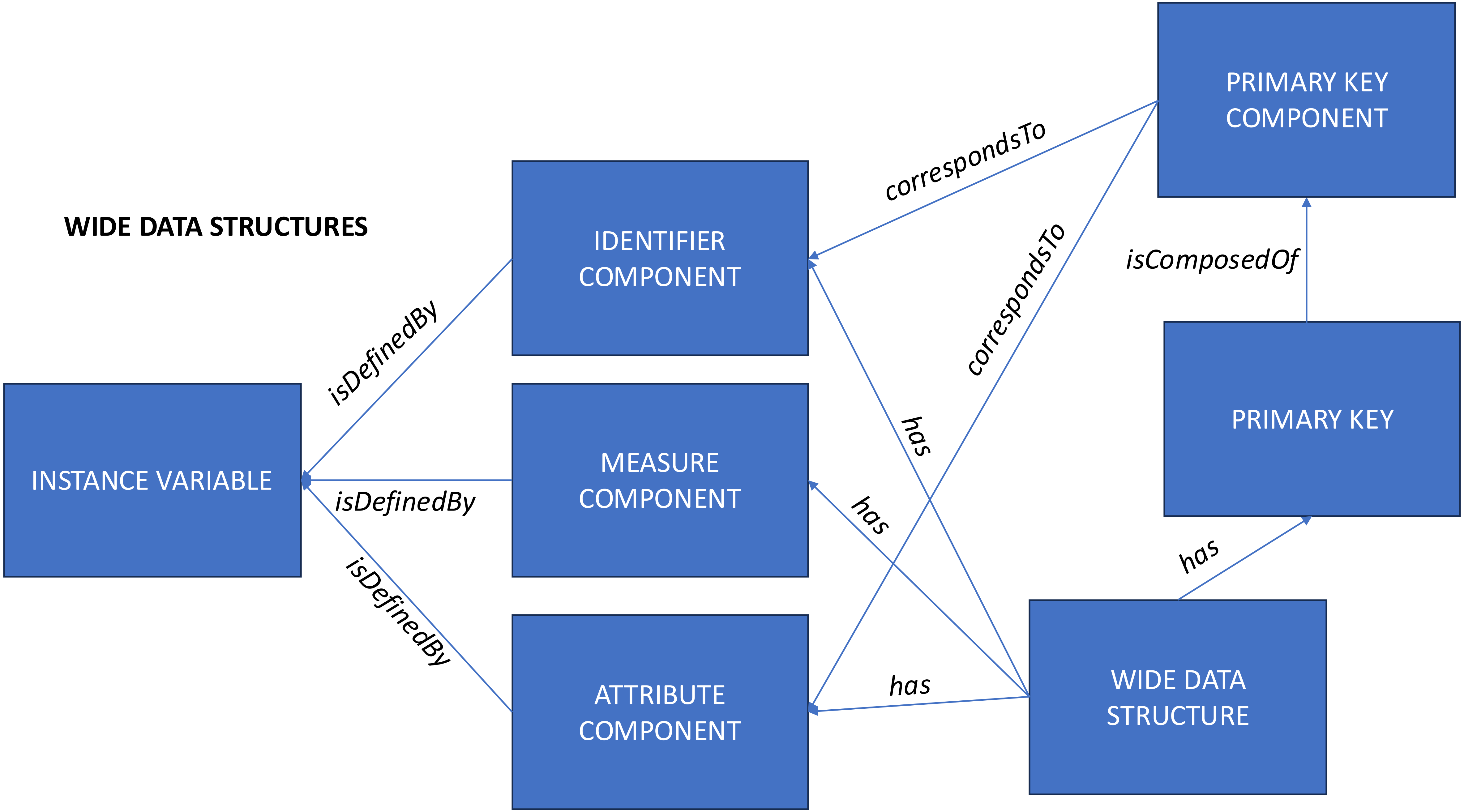
# Wide Data Example

- “Wide” data is also described as “unit-record data”
  - Each row describes one observed unit using a set of variables (columns)
- Here, Taxpayer ID, Year, and Taxes Paid are variables
- At least one variable holds the observation value (Taxes Paid)
- Other variables provide a “key” for identifying the observation value(s)
  - Taxpayer ID + Year
- Other variables can act as descriptors (none in this example)



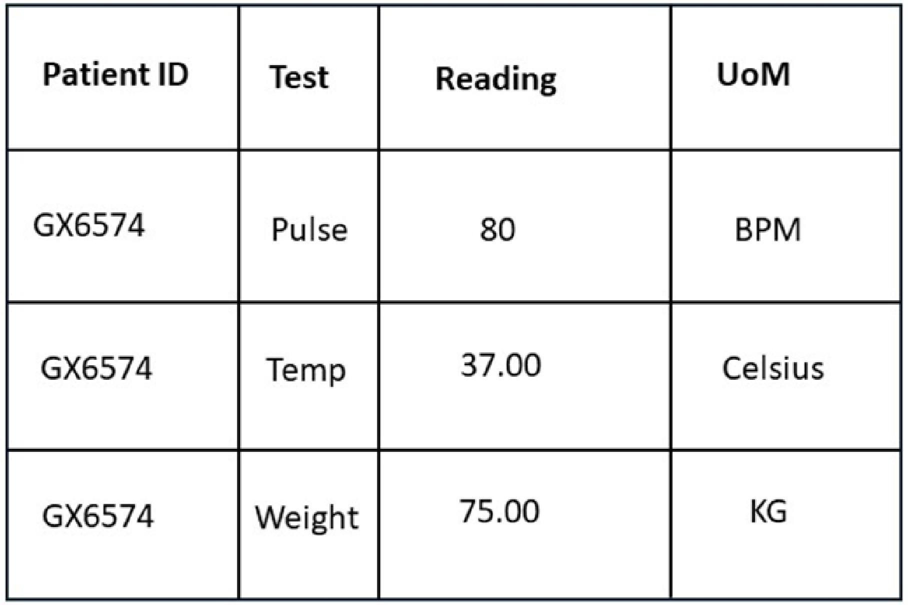
| Taxpayer ID | Year | Taxes Paid |
|---|---|---|
| 003-49-6574 | 2001 | 2400.00 |
| 003-50-1234 | 2011 | 3600.00 |
| 009-42-6324 | 2021 | 2100.00 |

## WIDE DATA STRUCTURES

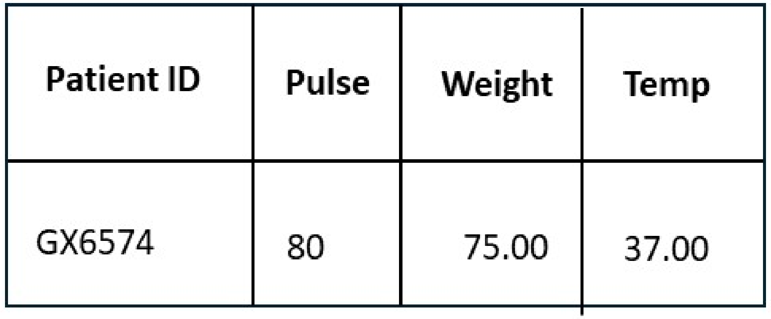


# Long Data Example

- Long data is often seen as the output of sensors, in event registers, and similar “streams” of data
- Each entry is a value plus identifiers and descriptors to tell you what the logical content is
  - The smaller table below shows how this data could be understood as Wide data, with a “logical” variable in each column
  - Note that the UoM (unit of measure) is part of the variable descriptions in this example (and thus not shown)

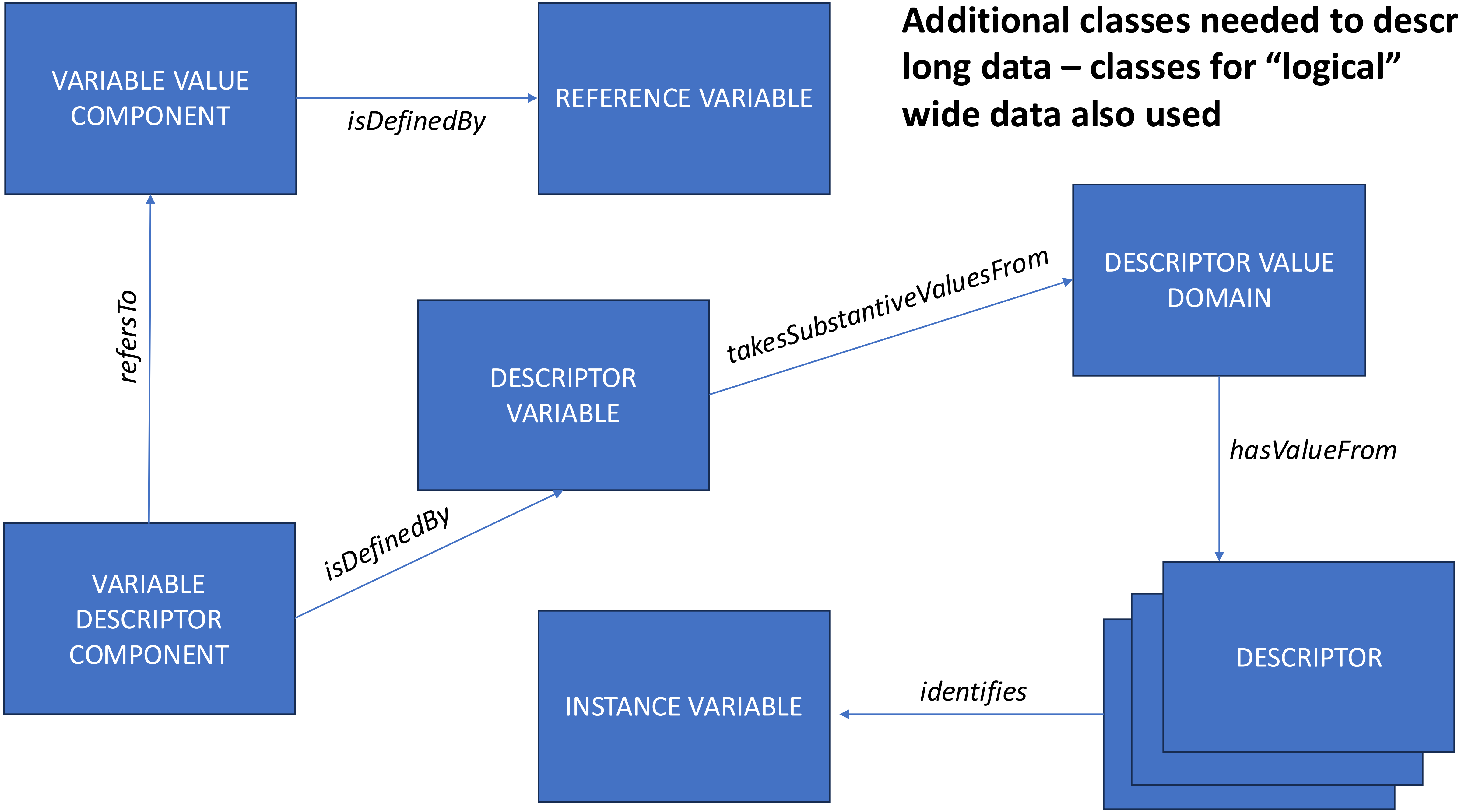


| Patient ID | Test | Reading | UoM |
|---|---|---|---|
| GX6574 | Pulse | 80 | BPM |
| GX6574 | Temp | 37.00 | Celsius |
| GX6574 | Weight | 75.00 | KG |



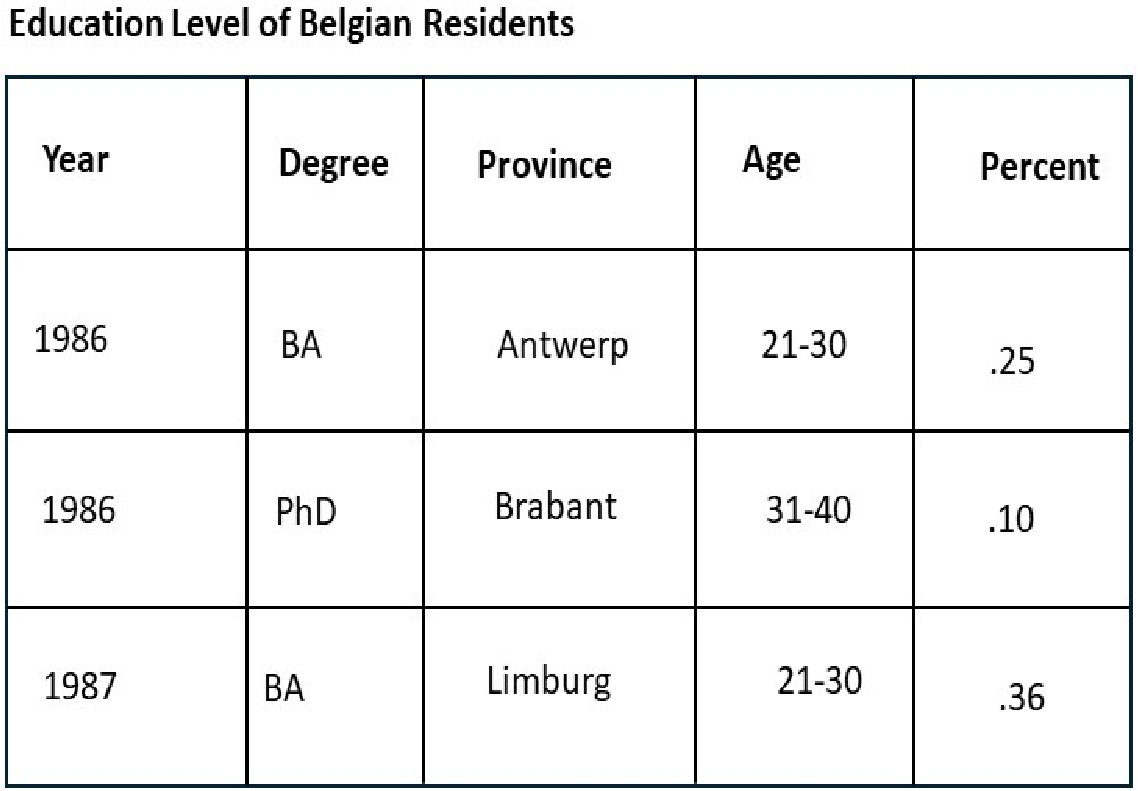
| Patient ID | Pulse | Weight | Temp |
|---|---|---|---|
| GX6574 | 80 | 75.00 | 37.00 |

## Additional classes needed to describe long data – classes for “logical” wide data also used



# Dimensional Data Example

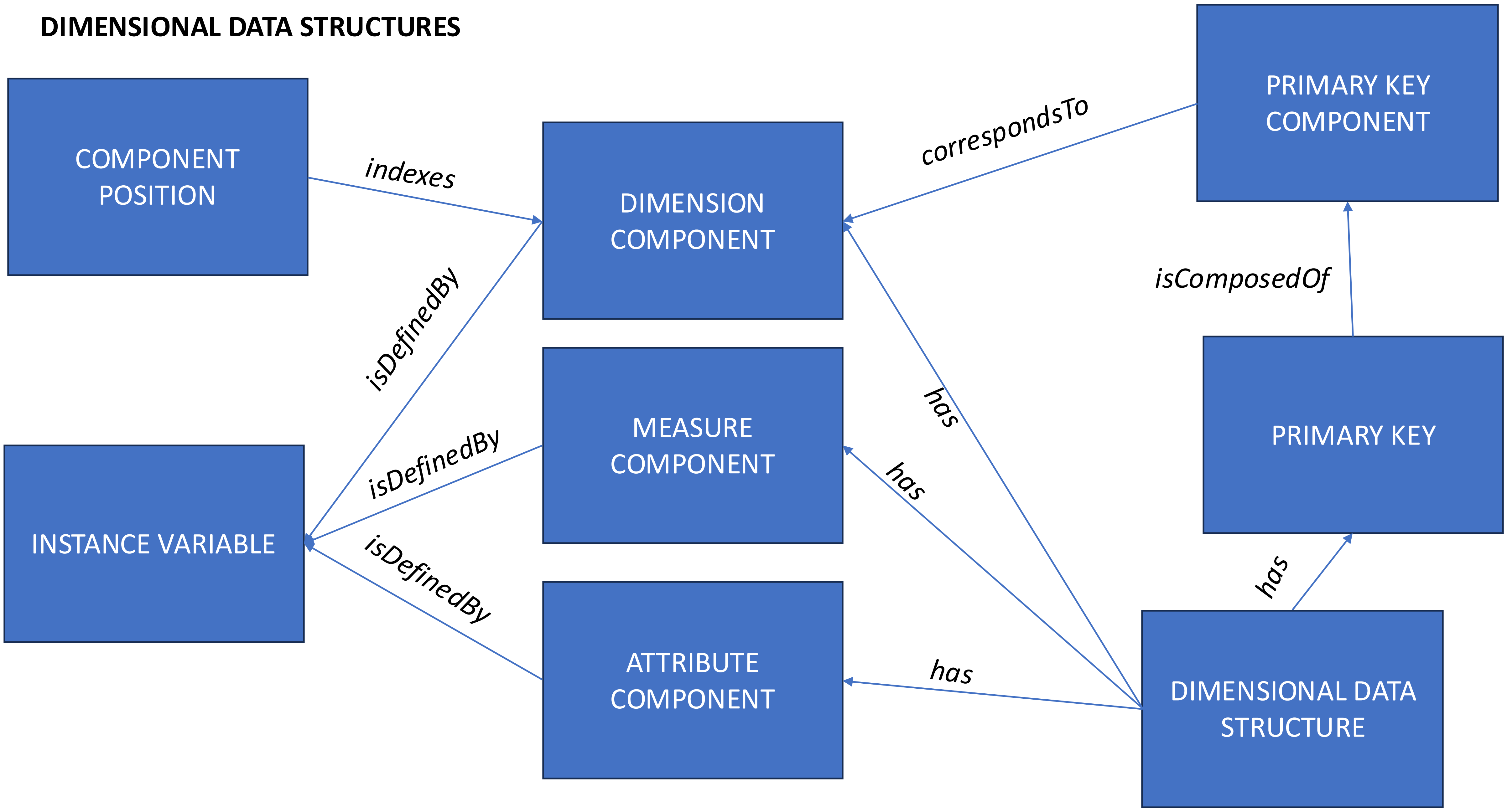
- Each observation is identified with a structured key, composed of dimensions
- Dimensions often have computable values (time, hierarchical categories, etc.) which can be used to “roll up” and “roll down” the values
  - Think OLAP cubes
- In practice, the dimensions of keys are often presented in a standard sequence for convenience and ease of computation



Education Level of Belgian Residents

| Year | Degree | Province | Age | Percent |
|---|---|---|---|---|
| 1986 | BA | Antwerp | 21-30 | .25 |
| 1986 | PhD | Brabant | 31-40 | .10 |
| 1987 | BA | Limburg | 21-30 | .36 |

# DIMENSIONAL DATA STRUCTURES



COMPONENT POSITION

DIMENSION COMPONENT

PRIMARY KEY COMPONENT

MEASURE COMPONENT

PRIMARY KEY

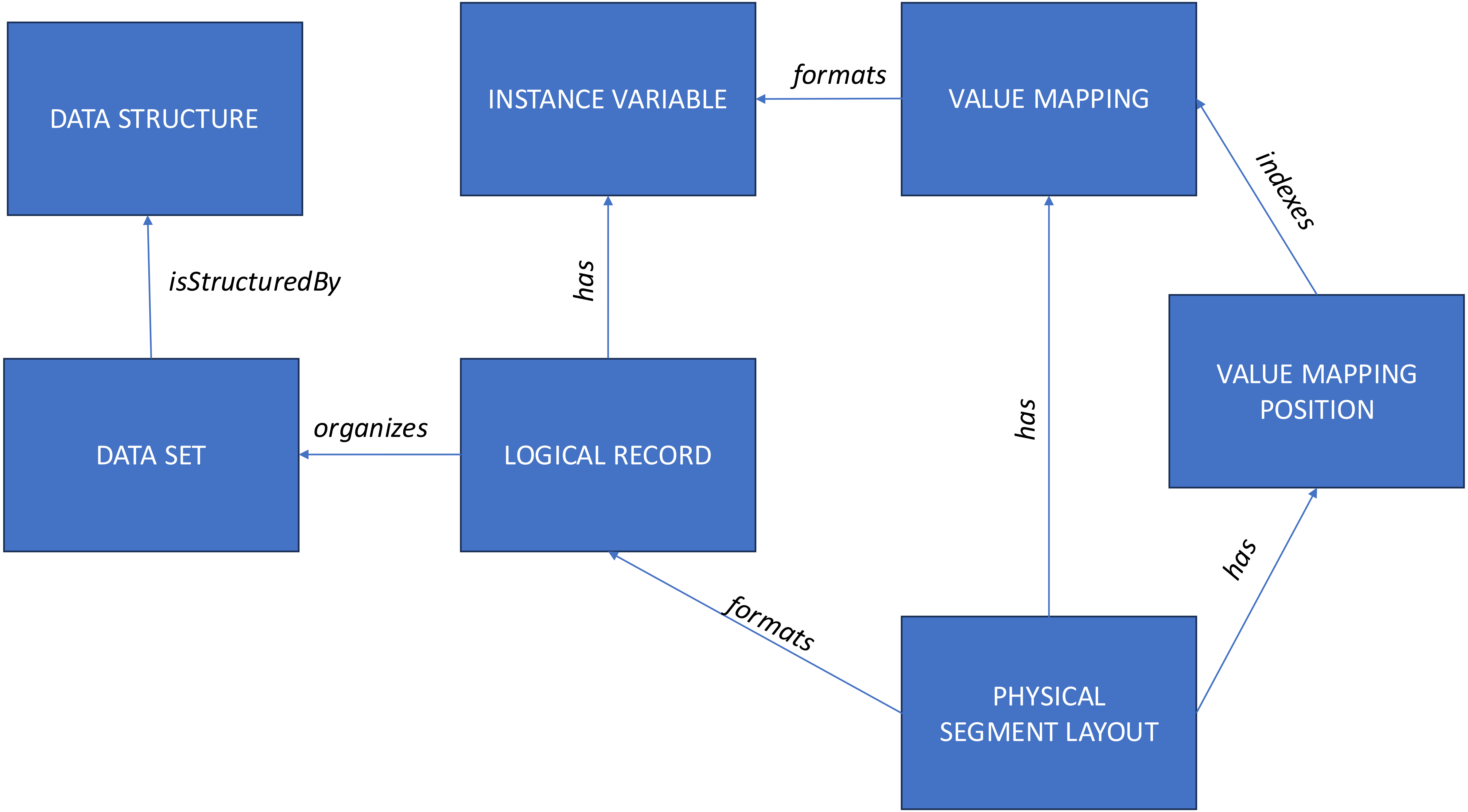
INSTANCE VARIABLE

ATTRIBUTE COMPONENT

DIMENSIONAL DATA STRUCTURE

# Describing Physical Data

- Assumption is that data is in a text-based form (delimited ASCII, CSV, etc.)
- Physical data references a specific type of structure. It holds data formed into a single type of Logical Record, and then maps the Instance Variables in that record to positions/fields within a Physical Record Layout.
- This is a simple description which works for a wide range of data, but does not cover all possibilities
    - Single record layout
    - Single logical record


DATA STRUCTURE
INSTANCE VARIABLE
formats
VALUE MAPPING
isStructuredBy
has
indexes
has
VALUE MAPPING POSITION
DATA SET
organizes
LOGICAL RECORD
has
formats
PHYSICAL SEGMENT LAYOUT

# Other Significant Parts of the Model

- This brief introduction does not cover the Process part of DDI-CDI
  - Can describe Activities, Steps, and Sub-Steps in a process flow – useful for describing business-level flows and chaining together other standard descriptions of processing (SDTL, PROV, etc.)
  - Can describe “black box” processes with a set of inputs, outputs, completion criteria and “playbook” functions
- This introduction does not describe Datums
  - Individual data values within data
  - Powerful, but potentially complex
- These features not currently used in CDIF

# DDI-CDI Sample Generator

- https://ddialliance.github.io/ddi-cdi-sample-generator/