

DDI-CDI and Other Standards

Overall goal

To define general patterns for translating between different data models (SDMX, schema.org, etc) using DDI-CDI as the “link” between those standards

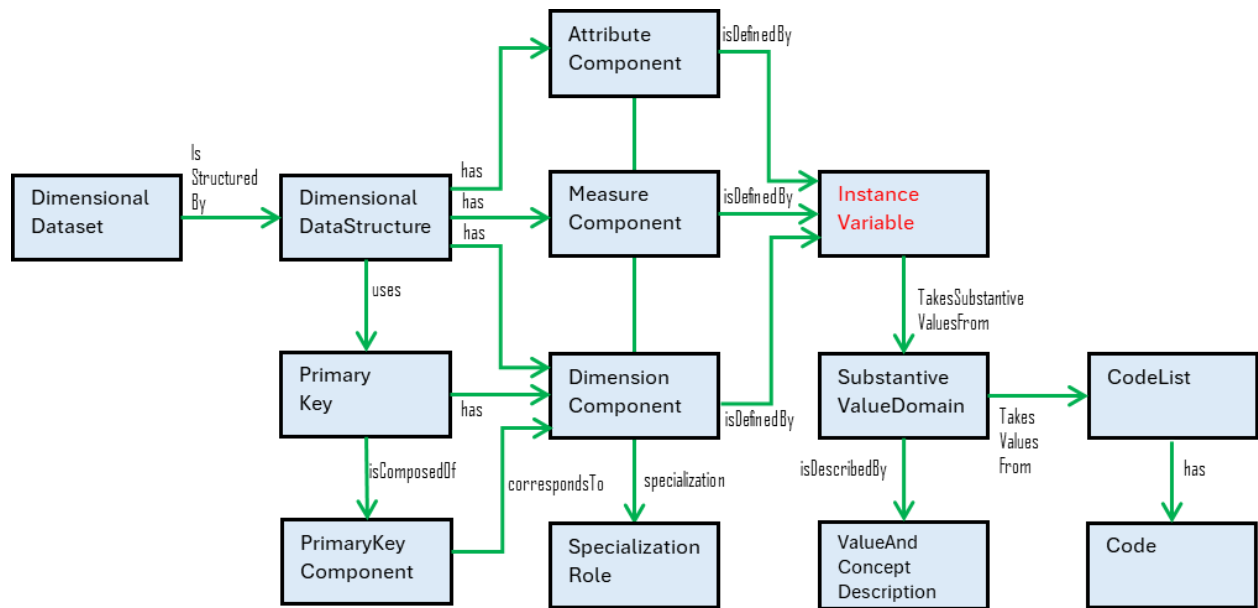


Figure 1: DDI-CDI Dimensional Data Integration Profile

1) Making dataset “integration-ready”

Source datasets are described using the DDI-CDI vocabulary (more specifically, using the DDI-CDI profile described above).

This requires mapping specific objects in the source dataset to individuals of specific classes in the DDI-CDI vocabulary.

Accompanying the DDI-CDI data descriptions, the dataset will be converted into a wide-form tabular array (with columns corresponding to each dimension, measure, and attribute component), serialized as csv.

2) Describing the target integrated dataset

The target data structure will be specified using the same DDI-DCI profile. The target dataset will specify a dimensional data structure, with its components and corresponding represented variables and value domains.

This requires mapping the main “entities” and “relations” of the source model (e.g., SDMX artifacts or schema.org classes or properties) to the relevant subset of DDI-DCI classes and properties

- 3) Mappings between source and target represented variables
- 4) Mappings between source and target codes for each variable whose value domain is given by a code list.
- 5) Automating data integration

A python script will be developed to automate the integration of each source dataset into a unified target dataset.

Concrete deliverable

1. A revised CDI-DDI data integration profile in ttl format
[Available in Git](#)
2. CDI-DDI “connector” mappings, i.e., mappings between the main “entities” and “relations” of the source models of the two use cases to the classes or properties of the DDI-DCI data integration profile
 - a. SDMX
 - b. Data Commons (based on schema.org)
3. Description of source schemas of SDG indicator data using the DDI-DCI data integration profile
 - a. SDMX: [Global SDG SDMX Data Structure Definition + Code lists](#)
 - b. Data Commons: UN Data Commons for the SDGs schema
4. A Python script that that uses all the artifacts above to convert a concrete example of an SDG dataset in SDMX format into schema.org.

Use case 1: SDMX

1. **An XML message including with full SDMX message from ILO, including:**
 - Data table

- DSD
 - Code lists
 - Concept scheme
2. **Sample_target_1.ttl**: An RDF file showing the *desired* DDI-CDI output for **one or two rows** of the input data, using the UNData DDI-CDI subset and extensions
 3. A python script that builds the full target TTL from the SDMX input data

A stylized version of the SDMX information model is illustrated in the following diagram.

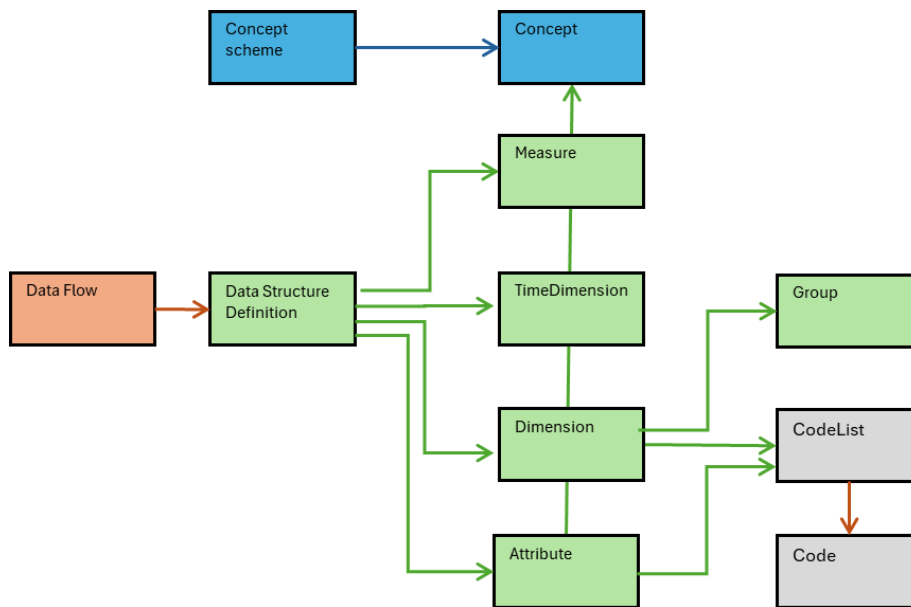


Figure 2: Simplified view of the SDMX information model

A mapping specifying the most important correspondences between the “classes” of the SDMX information model and the DDI-CDI classes is presented below.

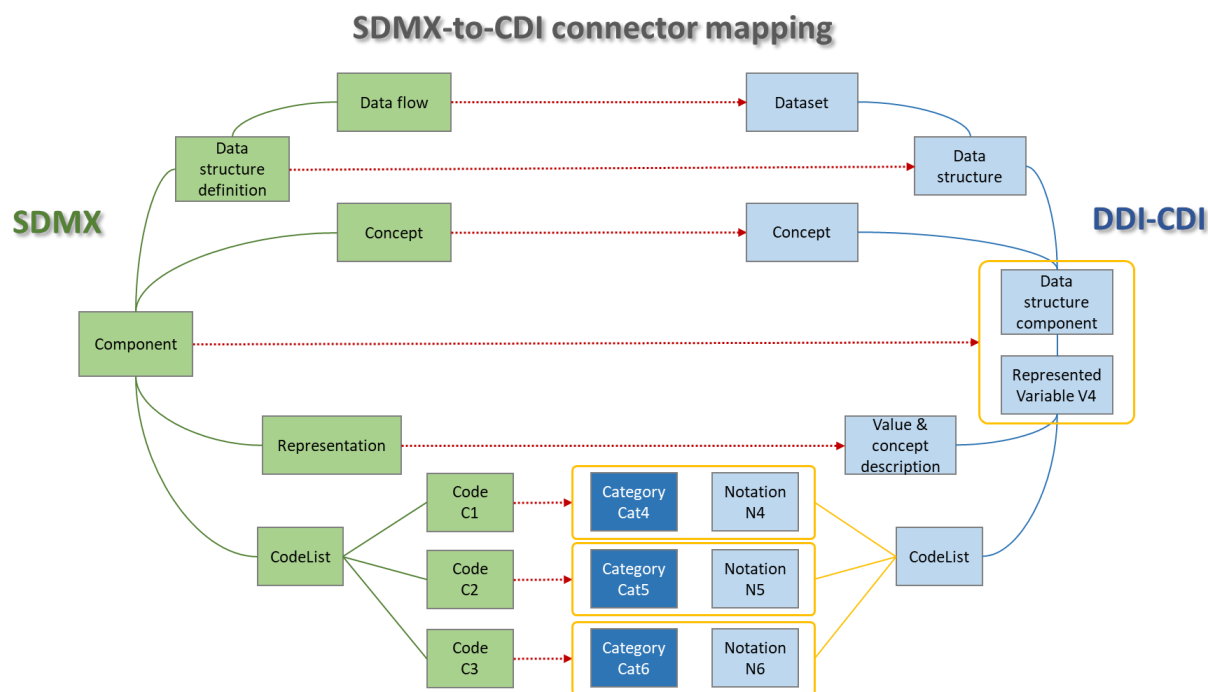


Figure 2: SDMX-to-DDI-CDI “connector” mapping

Example 0: Proto-SDMX

A representation in Excel format of the SDMX artifacts describing the global SDG dataset can be downloaded from the [SDMX Global](#) registry. These artifacts are available in this [file](#). The main artifacts are: (1) Concept Scheme, (2) Data Flow, (3) Data structure definition, and (4) Code lists.

- The **Concept Scheme** is simply a list of variables (concepts), such as Reference Area, Age, Value or Unit of Measure.
- The **Dataflow** is a “package” that identifies the dataset itself, understood as a collection of observations (records) organized according to a specific data structure definition.
- The **Data Structure Definition (DSD)** identifies a selection of variables (concepts) from the Concept Scheme which are contained in the dataset, alongside their roles as dimensions, attributes or measures. When applicable, the DSD also identifies the code lists used to represent their specific values in the dataset.
- The **code lists** referenced in the DSD for coded variables (concepts) specify the possible codes used to represent the domain of values that those variables can take in the data.

Based on these files, a complete description of a dataset that conforms to the Proto-SDMX Data Structure Definition can be translated into a DDI-CDI profile.

Use case 2: Data Commons / schema.org

1. **As set of mcf files from Google Data Commons**
 2. **Sample_target_2.ttl**: An RDF file showing *desired* DDI-CDI output for **one or two observations** of the input data
 3. A python that builds the target ttl from the input data
-
- Identify the “types” of mappings and transformations involved in the production of an integrated dataset
 - Brainstorm on how to codify those mappings and transformations
 - SSSOM ? XKOS ? Data Commons mapping utilities? Other?
 - Prototype: automated data integration workflow based on DDI-CDI representation of source datasets and proposed mappings.

CASE 1: SDMX

Notes about data formats

In addition to describing the data using different standards, data providers often publish their data using different physical formats, e.g. SDMX XML, CSV, Parquet, and others. This creates different issues for data consumers:

- Same as with metadata, the data needs to be organized in a single data format before integration, namely a DDI-CDI data format.
- DDI-CDI doesn't include any specific data format in the specification, which means a number of formats could be used.
- Guidelines need to be produced to determine how to map the physical data structures in CDI to the most common data formats.
- Once all the data sources are in this common format, we need to determine the best way to actually perform the data integration, including all data transformation required.
 - One possibility is to read each of the data sources into in-memory structures, e.g. data frames, and use some language-specific integration process.
 - When integrating large data sets, the first option might be unfeasible, in which case another possibility would be to use the native functionality of the DDI-CDI data format to perform the data integration. This native functionality will usually be dependent on specific tools or a specific language, e.g. SQL in a RDBMS.
 - Once the data integration is completed, the output, integrated dataset will be stored in the DDI-CDI data format for sharing.

End of day outputs

- Outputs
- Summary doc
- Checklist of deliverables
- Questions
 - Priorities
 - Resources
 - Open issues
 - Next steps