SDMX 3.0 – Enhanced artefacts

January 25, 2021

Alvaro Diez Soto
Deputy Head of Unit/Eurostat
Objectives

SDMX 3.0 new features require changes in the information model, this presentation will focus on changes that will allow:

- Simplify the representation of hierarchies of codes and the linkage to other elements of the information model (Hierarchical code lists).
- Simplify and enhance the use of constraints (Constraints).
- Simplify and enhance the mapping capabilities of SDMX (Structure Sets).
SDMX—Hierarchical code lists

- Code list in SDMX supports a simple hierarchy of Codes, and restricts any child Code to having just one parent Code.
- Often, the Code list used are derived from a more complex coding scheme that cannot be represented with a simple code list.
- A Hierarchical Code List (HCL) is a code list that can represent complex hierarchies:
  - It does not create code lists and codes, just references codes in existing code lists.
  - The referenced codes are listed in an object called a Hierarchy.
  - There can be multiple Hierarchies within a HCL.
SDMX – Hierarchical code lists – Issues

- No link between the Hierarchical code lists and its hierarchies and any other object in the SDMX information model
- The link to be created, has to be done at the level of each hierarchy within a HCL, the HCL is just a collection of hierarchies
SDMX 3.0 – Hierarchical code lists

- **Change 1**: Remove the Hierarchical Codelist and make the Hierarchy a Maintainable Artefact.

- **Change 2**: Change the Information Model to include a link (association) between a Hierarchy and the object to which it is linked (e.g., a Dimension).

- **Change 3**: Change the Information Model to include a link (association) between a Hierarchy and the context in which the previous linking is performed (e.g., a Dataflow including the above Dimension).
Hierarchical code lists– SDMX 2.1 vs SDMX 3.0

SDMX 2.1

SDMX 3.0
SDMX– Constraints

- A constraint is used to:
  - further restrict a code list by defining a subset of permitted or restricted values or to define full or partial series keys allowed or restricted (ContentConstraint)
  - define slices of the full set of data and to which metadata can be attached in a dataset or metadata set (AttachmentConstraint)

- ContentConstraints are semantically split in two types:
  - Allowed ContentConstraint: to describe the allowed values that DSD dimensions may take within datasets
  - Actual ContentConstraint: to present the actual data that an organisation has available in an SDMX web service
SDMX – Constraints – Issues

- Organization of constraints has to be reviewed in view of:
  - Complexity of the model
  - Lack of use of some of the implemented features
  - Align with the advancements of the SDMX REST API to ease their use
  - Adapt to the changes proposed to MSDs and metadatasets in SDMX 3.0
  - Consider missing features like the capability to add Attribute constraining rules to DataKeysets
  - Difficulty in the specification of complex content constraint (constraining long list of codes)
SDMX 3.0 – Constraints(1/2)

- **Change 1**: AttachmentConstraints are deprecated, due to the lack of use cases.
- **Change 2**: The semantic split of ContentConstraint is removed. ActualConstraints are replaced by the use of the data availability queries.
- **Change 3**: Split the ContentConstraint into DataConstraint and MetadataConstraint as there are two different target groups for constrains (data and metadata)
- **Change 4**: Simplify the metadata related cube region to reflect the changes in the new MSDs.
SDMX 3.0 – Constraints(2/2)

- **Change 5:** Move the validity period from the Constraint to the constraining terms (CubeRegion, DataKeySet and MetadataTargetRegion).

- **Change 6:** Consider latest stable versions of Constraints per Constrainable artefact as effective Constraints.

- **Change 7:** Attributes to be allowed in DataKeySets, within each KeySet.

- **Change 8:** Allow using the character “%” to wildcard any character(s), either to the right of a specified prefix, or any other part of a Value, and include all items matching that pattern.

- **Change 9:** Allow excluding the root Code when using cascading.
SDMX– Structure Set

- A *StructureSet* allows components in one structure to be mapped to components in another structure of the same type.
- The *StructureSet* can contain one or more “maps” and can define related structures (via the association +relatedStructure) which group related>DataStructureDefinitions, MetadataStructureDefinitions, DataflowDefinitions, MetadataflowDefinitions.
- Much of the current demand for the Structure Set is to support mapping between datasets, and classification system.
SDMX– Structure Set – Issues (1/2)

- The structure set is overly complex and cumbersome to maintain as it is a container for multiple map types.
- It is not possible to match source values based on starts with, or patterns.
- Common use cases are not supported, due to lack of support for many to many rules.
Support for mapping from non SDMX compliant identifiers is limited and does not provide a reusable solution, which duplicates maintenance.

There is no provision for mappings from datasets which use a different representation for time.

Mapping rules which span a number of time periods require the same mapping rule duplicated for each period.
SDMX 3.0 – Structure Set (1/2)

- **Change 1:** Break down the different mappings within a Structure Set into their own independent maintainable structure types.

- **Change 2:** Allow start-with and regular expression in the definition of mappings.

- **Change 3:** Change the cardinality of the mapping data structures to allow N-N mapping.
SDMX 3.0 – Structure Set (2/2)

- **Change 4:** Build a mechanism to allow mapping from non-sdmx data source by allowing the mapping from valueList (list of values similar to codelist but without the restrictions on the types imposed by SDMX).

- **Change 5:** Allow the Component Map, which maps the source Dimension/Attribute to the target Dimension/Attribute, to specify that it is a Time based Mapping.

- **Change 6:** Introduce optional validity periods for each mapping.
SDMX 3.0 – New possibilities...

- The changes in HCL, Constraints and Structure Set will improve the usage of SDMX by allowing for example:
  - Users to define complex hierarchies for data dissemination and to link within the standard these hierarchies to other objects. As now is part of the standard, it will improve interoperability.
  - Define a constraint for a global code list like CL_ACTIVITY that contains different classifications with prefix (NACE, ISIC) by simply using the wildcarding NACE% (discriminated union).
  - Make possible the mapping of non SDMX artefacts to SDMX artefacts with the improved capabilities of the structure set.
Any Questions?