IMPLEMENTOR'S GUIDE FOR SDMX FORMAT STANDARDS

(VERSION 1.0)

Initial Release September 2004
First Revision December 2004

http://www.sdmx.org/

1 INTRODUCTION

This guide exists to provide information to implementors of the SDMX format standards – SDMX-ML and SDMX-EDI. This document is intended to provide information which will help users of SDMX understand and implement the standards. It is not normative, and it does not provide any rules for the use of the standards, such as those found in *SDMX-ML: Schema and Documentation* and *SDMX-EDI: Syntax and Documentation.*

This document is organized into parts:

- A guide to the SDMX Information Model
- Statement of differences in functionality supported by the different formats and syntaxes
- Best practices for use of SDMX formats

2 SDMX INFORMATION MODEL FOR FORMAT IMPLEMENTORS

2.1 Introduction

The purpose of this section is to provide an introduction to the SDMX Information Model for those whose primary interest is in the use of the XML or EDI formats. For those wishing to have a deeper understanding of the Information Model, the full SDMX Information Model document provides a more in-depth view, along with UML diagrams and supporting explanation. For those who are unfamiliar with key families, an appendix to the SDMX Information Model provides a tutorial which may serve as a useful introduction.

The SDMX Information Model is used to describe the basic data and metadata structures used in all of the SDMX data formats. There is a primary division between time series and cross-sectional data and the metadata which describes the structure of that data. The Information Model concerns itself with statistical data and its structural metadata, and that is what is described here. Both structural metadata and data have some additional metadata in common, related to their management and administration. These aspects of the data model are not addressed here.

This information model is consistent with the GESMES/TS version 3.0 Data Model, with these exceptions:

- the “sibling group” construct has been generalized to permit *any* dimension or dimensions to be wildcarded, and not just frequency, as in GESMES/TS. It has been renamed a “group” to distinguish it from the “sibling group” where only frequency is wildcarded. The set of allowable partial “group” keys must be declared in the key family, and attributes may be attached to any of these group keys;
- the section on data representation is now a convention, to support interoperability with EDIFACT-syntax implementations;
- cross-sectional data formats are derived from the model, and some supporting features for deriving cross-sectional and time-series views of a single data set structure have been added to the structural metadata descriptions.

Clearly, this is not a coincidence - the intention is that the GESMES/TS Data Model becomes the foundation for not only the EDIFACT messages, but also the XML used for web dissemination.

Note that in the descriptions below, text in courier and italicised are the names used in the information model (e.g. *`DataSet`*).

2.2 Fundamental Parts of the Information Model

The statistical information in SDMX is broken down into two fundamental parts - structural metadata (comprising the *`KeyFamily`*, and associated *`Concept`*s and *`Code Lists`*) – see Framework for Standards -, and observational data (The *`DataSet`*). This is an important distinction, with specific terminology associated with each part. Data - which is typically a set of numeric observations at specific points in time - is organized into. data sets (*`DataSet)`* These data sets are structured according to a specific key family (*`KeyFamily)`*, and are described in the data flow definition (*`DataFlowDefinition)`* The key family describes the metadata that allows an understanding of what is expressed in the data set, whilst the data flow definition provides the identifier and other important information (such as the periodicity or reporting) that is common to all of its component data sets.

2.3 Data Set

Data sets are made up of a number of time series or sections (the cross-sectional organization of observations at a single point in time). In addition to the numeric observation (*`Observation`*)and the related date (*`TimePeriod`*), which are the core of the time series, there may be attributes

(`AttributeValue`) indicating the status of the observation, e.g. whether the value is a normal or break value, etc. These attributes may be optional (or “conditional”), and may have coded or free text values. They may pertain to any part of the data set - each observation might have a different value for the attribute, or there might be only a single attribute value describing the entire data set, or for each time series, etc.

Each time series can be identified by the values of its dimensions. Time series data can be seen as n-dimensional. A given time series will have exactly one value (`KeyValue`), of the set of permissible values, for each of its dimensions (`Dimension`), and a set of observations (`Observation`): one value for each specific point in time (`TimePeriod`). A specific time series might have dimensions of "frequency", "topic", "stock or flow", "reporting country", etc., with a single corresponding value for each dimension. Taken together, this set of values uniquely identifies the time series within its data set, and is called the time series key (`TimeSeriesKey`).

Cross-sectional representations of the data may be derived from the same key family from which time-series representations are structured, so long as the needed additional structural metadata is provided. This functionality allows multiple measures to be declared in the key family, associated with the representational values of one dimension. When data is structured to represent a set of multiple observations at a single point in time, the “section” – one or more observations for each declared measure – replaces the series in the data structure. Each measure carries at least one dimension of the key (the “measure dimension”) at the observation level, while the time period is attached at a higher level in the data structure (the Group level – see below). The remainder of the key is found at the Section level (or above), similar to the way in which it is attached at the Series level for time series data structures.

Support for cross-sectional data representation is not as complete as that for representing time-series data. The intended functionality is to allow key families which are to be used to represent cross-sectional data to be created with this application in mind. Because time-series data representations are also possible for any key family which has time period as a concept, these data structures may also be derived from the key family. The functional result is that two complementary types of data structures may be provided: the needed cross-sectional view, and the time-series oriented view which may be useful to systems which may not be configured to process data in any other fashion. The key family created to support cross-sectional structuring of data will support the predictable (and thus, automatable) transformation of data from the cross-sectional structure into the time-series structure.

Data sets may be organized into “groups” of time series or sections (*GroupKey*); this is a particularly useful mechanism for attaching metadata to the data. One such group is called the “sibling group”, which shares dimension values for all but the frequency dimension (the frequency dimension is said to be “wildcarded”). In the key family, all legitimate groups are declared and named. All members of the group will share key values for a stated set of dimensions. Attributes may be attached at this level in the data formats, as are the shared key values for those formats where message size is an issue. In cross-sectional formats, time period (a period or point in time) is attached at the group level.

The key family is a description of all the metadata needed to understand the data set structure. This includes identification of the dimensions (*Dimension*) according to standard statistical terminology, the key structure (*KeyDescriptor*), the attributes (*MetadataAttribute*) associated with the data set, the code-lists (*CodeList*) that enumerate valid values for each dimension and coded attribute (*CodedAttribute*), information about whether attributes are required or optional and coded or free text. Given the metadata in the key family, all of the data in the data set becomes meaningful.

It is also possible to associate annotations (*Annotation*) with both the structures described in key families and the observations contained in the data set. These annotations are a slightly atypical form of documentation, in that they are used to describe both the data itself - like other attributes - but also may be used to describe other metadata. An example of this is methodological information about some particular dimension in a key family structure, attached as an annotation to the description of that dimension. Regular “footnotes” attached to the data as documentation should be declared as attributes in the appropriate places in a key family – annotations are irregular documentation which may need to be attached at many points in the key family or data set.

The following section provides more complete definitions of the SDMX Information Model as it relates to statistical data, for easy reference by syntax implementors.

2.4 Attachment Levels and Data Formats

It is worth looking briefly at the available formats in light of the discussion above:

- SDMX-ML and SDMX-EDI both have a format for describing key families, concepts, and codelists.

- In SDMX-EDI, there is a single message format for transmitting data-related messages. This format allows for very compact expression of different types of packages of information: just data, just documentation, delete messages, etc. This format is time-series-oriented. Time is specified either as a range for a set of observation values with a known frequency, or is associated on a one-to-one basis with observation values.
- In SDMX-ML, the Generic Data Message requires that all key values be specified at the Series level, and that attribute values be attached at the level in which they are assigned in the key family (if any attribute values are to be transmitted). This is a time-series-oriented format, which requires that a time be specified for each observation value.
- In SDMX-ML, the Compact Data Message requires the values of keys to be specified at the Series level. Attribute values are specified at the level assigned to them in the key family, if provided. This is a time-series-oriented format, which associates time with observations either on a one-for-one basis, or expressed as a range for a set of observations with a known frequency.
- In the SDMX-ML Utility Data Message, all key values are attached at the Series level, and all attribute values are attached at the level of assignment in the key family. Attribute values must be provided – there is no concept of a "delete" message or partial message (for updates, documentation-only, etc.) as there is for other data formats. This is a time-series-oriented format which requires that time be specified for each observation on a one-for-one basis.
- In the SDMX-ML Cross-Sectional Data Message, attachment levels vary more than in other formats. Key values may be attached at any level or combination of levels, as declared in the key family, with the exception that time is always attached at the group level for those key families which use time as a concept. Key values may be attached at the observation level for each type of declared measure. Attribute values may be provided at any of the levels assigned in the key family. This is the only non-time-series-oriented format.
- SDMX-ML has a Query message, but discussion of the attachment levels is not relevant for this message.

2.5 Concepts, Definitions, Properties and Rules

This section provides a common language and framework for describing statistical data exchanges.

1. A **period (`TimePeriod`)**is a *time reference,* which may be either a time period or a point in time.

2. An **observation (`Observation`)** is the value, at a particular period, of a particular variable (sometimes called the “observed phenomenon”).

3. To be useful, an observation must have more information relating to it than just a value and an associated period. Information about observations is called **metadata**.

4. The characteristics of observations that make up the metadata are known as **statistical concepts (`Concept`)** (eg, reporting country). The use of a statistical concept in a key family (`KeyFamily`) is either *coded* or *uncoded*.

- A **coded statistical concept usage** takes values from a **code list (`CodeList`)** of valid values. For example, a coded statistical concept called “reporting country” might be created, taking its values from the ISO list of country codes. A code list may supply the values of more than one statistical concept.

- An **uncoded statistical concept usage** takes its values as free form text (eg, time series title).

5. A **time series** is a time-ordered vector of observations (`Observation`).

6. If a time series has time intervals between its observations, this interval determines the **frequency** of the time series.

7. **Data exchange context** is the framework in which two or more partners agree to:

- exchange one or more identified sets of data and related attributes (“**exchanged time series**”; **ETS**).

- use one or more key families to serve this requirement;

- possibly, comply with some business and implementation agreements.

8. **Structural definitions maintenance agency** is an institution that devises key families.

9. The exchanged time series (ETS) is a collection of **data flow definitions** (`DataFlowDefinition`) for which instances, known as **data sets (`DataSet`)**are exchanged.

10. Each data flow definition takes its structure from exactly one **key family (**`KeyFamily`**)**.

11. Each data flow definition is uniquely identified within an ETS by a **data flow identifier.**

12. Each key family **links** exactly one code list to each coded statistical concept usage defined in that key family.

13. Each key family is uniquely identified by a structural definitions maintenance agency using a unique **key family identifier**.

14. Each key family has a **key structure (***`KeyDescriptor`***)**, namely an ordered set of coded statistical concept usages whose combination of values uniquely *identifies* each time series within a data set.

- The coded statistical concept usages assigned as members of a key family's key structure are called the **dimensions (*`Dimension`*)** of the key family.

- **Measure dimensions (***`MeasureTypeDimension`***)** are a specialized class of dimension. The codes which represent a measure dimension correspond one-to-one with a set of declared cross-sectional measures (*`Measure`*). Measure dimensions only exist in key families which describe cross sectional presentation of data.

- No key family is permitted to assign a particular coded statistical concept usage as a dimension more than once. (The same codelist may be used to represent more than one statistical concept within the key family, however.)

- Only coded statistical concept usages are permitted to be dimensions of a key structure.

- **Frequency** must be assigned as a dimension (*`FrequencyDimension`*) in every key family which uses the concept of Time (*`TimeDimension`*). (Note that most central institutions devising structural definitions have decided, in order to facilitate frequency's identification in a homogeneous manner, to define frequency as the first dimension of the key structure.)

- Each time series takes a value (`KeyValue`) for every dimension of the key family to which the series belongs.

- The meaning attached to the value of one dimension or attribute is not permitted to depend upon the values of any other dimensions, with the exception of "measure" dimension and "unit" attribute described above.

- The list of values uniquely identifying a time series within a data set is called the **key** of the time series (`TimeSeriesKey`).

- Within the ETS a time series is uniquely identified by a data set identifier combined with the time series key. (Note that in the information model the identifier of the data set is the data flow identifier qualified by time).

- Within a data set, an observation is identified by a time series key (`TimeSeriesKey`).combined with a time period (`TimePeriod`).

15. Within a data set, a set of time series whose keys differ only in the value taken by the frequency dimension is called a **sibling group**. Within an ETS, a sibling group is uniquely identified by a data set identifier combined with the sibling group key (`GroupKey`). A set of time series whose keys differ along some other dimension value or values are termed a **group**.

16. In addition to the dimensions, each key family assigns a set of statistical concept usages that *qualify* the observations within the key family. The members of this set of statistical concept usages are called the **attributes (**`MetadataAttribute`**)** of the key family.

- No key family is permitted to assign a particular statistical concept usage as an attribute more than once.

- No statistical concept usage may be assigned as both an attribute and a dimension of the same key family.

- Each key family has a property for each of its attributes that determines whether:

- the attribute takes an independent value for each *observation* in the data set
- the attribute takes an independent value for each *time series* in the data set
- the attribute takes an independent value for each *sibling group* in the data set
- the attribute takes a single value for the entire *data set.*

This property uniquely identifies the **attachment level** of the attribute for the key family.

- Within a given key family, each attribute is considered either *mandatory* or *conditional*. (A *conditional* attribute is one where the value may be supplied based on conditions external to the formal relationships described by key family: functionally, it is a value which is “optional”.)
  - a **mandatory attribute** is an attribute which must take a value, otherwise the corresponding observation(s), which it refers to, is (are) not considered meaningful enough (eg, the "status" of an observation or the units to which a whole time series is expressed).
  - a **conditional attribute** is permitted to take empty values within the key family.

- Annotations (`Annotation`) are irregular documentation which may need to be attached at many points in the key family or data set.

17. Each key family has the following properties:

- **Identifier:** It provides a unique identification within the set of key families specified by a structural definitions maintenance agency.

- **Name:** This is a non-unique identifier meant to be more intuitive than the Identifier.

- **Description:** Description of the purpose and domain covered by the key family.

18. Each data set has the following properties:

- **Identifier**: It provides a unique identification within an ETS (in the information model this is the identifier of the `DataFlowDefinition`)

- **Description:** Description of the purpose and domain covered by the data set (in the information model this description is part of the DataFlowDefinition).

- **Key family**: Key family describing the structure of the data set.

19. Each statistical concept (`Concept`) has the following properties:

- **Identifier**: It provides a unique identification within the set of statistical concepts specified by a structural definitions maintenance agency.

- **Name:** This is a non-unique identifier meant to be more intuitive than the Identifier.

- **Description:** Description of the meaning and purpose of the statistical concept.

20. IEach uncoded statistical concept usage has the following properties:

- **Type:** Alpha, alphanumeric, numeric.

- **Maximum length:** The maximum number of characters in the text values of the concept.

- **Decimals:** The number of digits that appear after the decimal point in the number

21. Each code list has the following properties:

- **Identifier:** It provides a unique identification within the set of code lists specified by a structural definitions maintenance agency.

- **Name:** This is a non-unique identifier meant to be more intuitive than the Identifier .

- **Description:** Description of the purpose of the code list.

- **Code value length:** Either an exact or a maximum number of characters and a type (ie, numeric or alphanumeric) must be specified.

22. Each code in a code list has the following properties:

- **Identifier:** It provides a unique identification within the code list specified by a structural definitions maintenance agency

- **Name:** This is a non-unique identifier meant to be more intuitive than the Identifier .

- **Description:** It uniquely describes the code value.

3 SDMX-ML AND SDMX-EDI: COMPARISON OF EXPRESSIVE CAPABILITIES AND FUNCTION

SDMX offers several equivalent formats for describing data and structural metadata, optimized for use in different applications. Although all of these formats are derived directly from the SDMX Information Model, and are thus equivalent, the syntaxes used to express the model place some restrictions on their use. Also, different optimizations provide different capabilities. This section describes these differences, and provides some rules for applications which may need to support more than one SDMX format or syntax.

3.1 Format Optimizations and Differences

The following section provides a brief overview of the differences between the various SDMX formats.

Structure Definition

- The SDMX-ML Structure Message supports the use of annotations to the structure, which is not supported by the SDMX-EDI syntax.
- The SDMX-ML Structure Message allows for the structures on which a key family depends – that is, codelists and concepts – to be either included in the message or to be referenced by the message containing the key family. XML syntax is designed to leverage URIs and other Internet-based referencing mechanisms, and these are used in the SDMX-ML message. This option is not available to those using the SDMX-EDI structure message.

Validation

- SDMX-EDI – as is typical of EDIFACT syntax messages – leaves validation to dedicated applications ("validation" being the checking of syntax, datatyping, and adherence of the data message to the structure as described in the structural definition.)

- The SDMX-ML Generic Data Message also leaves validation above the XML syntax level to the application.
- The SDMX-ML Compact Data and Cross-Sectional Data Messages will allow validation of XML syntax and datatyping to be performed with a generic XML parser, and enforce agreement between the structural definition and the data to a moderate degree with the same tool.
- The SDMX-ML Utility Data Message leverages a generic XML parser to perform the most complete degree of validation at all levels. (Note that dependencies between and among coded dimension and attribute values are not captured in the structural definition, and therefore must always be validated by the application.)

Update and Delete Messages and Documentation Messages

- The SDMX-ML Utility Data Message must always provide a complete update of the data set, ands thererefore cannot be used to perform deletions. Further, it cannot be used to send documentation without the corresponding data. All other SDMX data messages allow for both delete messages and messages consisting of only data or only documentation.

Character Encodings

- All SDMX-ML messages use the UTF-8 encoding, while SDMX-EDI uses the ISO 8879-1 character encoding. There is a greater capacity with UTF-8 to express some character sets (see the [Reference the SDMX-EDI appendix here]). Many transformation tools are available which allow XML instances with UTF-8 encodings to be expressed as ISO 8879-1-encoded characters, and to transform UTF-8 into ISO 8879-1. Such tools should be used when transforming SDMX-ML messages into SDMX-EDI messages and vice-versa.

Data Typing

The XML syntax and EDIFACT syntax have different data-typing mechanisms. The section below provides a set of conventions to be observed when support for messages in both syntaxes is required.

3.2 Data Types

The XML syntax has a very different mechanism for data-typing than the EDIFACT syntax, and this difference may create some difficulties for applications which support both EDIFACT-based and XML-based SDMX data formats. This section provides a set of conventions for the expression in data in all formats, to allow for clean interoperability between them.

It should be noted that this section does not address character encodings – it is assumed that conversion software will include the use of transformations which will map between the ISO 8879-1 encoding of the SDMX-EDI format and the UTF-8 encoding of the SDMX-ML formats.

Note that the following conventions may be followed for ease of interoperation between EDIFACT and XML representations of the data and metadata. For implementations in which no transformation between EDIFACT and XML syntaxes is forseen, the restrictions below need not apply.

23. Identifiers are:

- Maximum 18 characters;
- Any of A..Z (upper case alphabetic), 0..9 (numeric), _ (underbar);
- The first character is alphabetic.

24. Names are:

- Maximum 70 characters.
- From ISO 8859-1 character set (including accented characters)

25. Descriptions are:

- Maximum 350 characters;
- From ISO 8859-1 character set.

26. Code values are:

- Maximum 18 characters;
- Any of A..Z (upper case alphabetic), 0..9 (numeric), _ (underscore), / (solidus, slash), = (equal sign), - (hyphen);

However, code values providing values to a dimension must use only the following characters: A..Z (upper case alphabetic), 0..9 (numeric), _ (underscore)

27. Observation values are:

- Decimal numerics (signed only if they are negative);
- The maximum number of significant figures is:
  - 15 for a positive number

  - 14 for a positive decimal or a negative integer
  - 13 for a negative decimal
- Scientific notation may be used.

28. Uncoded statistical concept text values are:
- Maximum 1050 characters;
- From ISO 8859-1 character set.

29. Time series keys:

In principle, the maximum permissible length of time series keys used in a data exchange does not need to be restricted. However, for working purposes, an effort is made to limit the maximum length to 35 characters; in this length, also one (separator) position is included between all successive dimension values; this means that the maximum length allowed for a pure series key (concatenation of dimension values) can be less than 35 characters. The separator character is a colon (“:”) by conventional usage.

4 SDMX-ML AND SDMX-EDI BEST PRACTICES

4.1 Reporting and Dissemination Guidelines

4.1.1 Central Institutions and Their Role in Statistical Data Exchanges

Central institutions are the organisations to which other partner institutions "report" statistics. These statistics are used by central institutions either to compile aggregates and/or they are put together and made available in a uniform manner (e.g. on-line or on a CD-ROM or through file transfers). Therefore, central institutions receive data from other institutions and, usually, they also "disseminate" data to individual and/or institutions for end-use. Within a country, a NSI or a national central bank (NCB) plays, of course, a central institution role as it collects data from other entities and it disseminates statistical information to end users. In SDMX the role of central institution is very important: every statistical message is based on underlying structural definitions (statistical concepts, code lists, key families) which have been devised by a particular agency, usually a central institution. Such an institution plays the role of the reference "structural definitions maintenance agency" for the corresponding messages which are exchanged. Of course, two institutions could exchange data using/referring to structural information devised by a third institution.

Central institutions can play a double role:

- collecting and further disseminating statistics;
- devising structural definitions for use in data exchanges.

4.1.2 Defining Key Families

The following guidelines are suggested for building a key family. However, it is expected that these guidelines will be considered by central institutions when devising new key family definitions.

- ***Avoid dimensions that are not appropriate for all the time series in the key family.*** If some dimensions are not appropriate for some series then consider moving these series to a new key family in which these dimensions are dropped from the key structure[1].

- ***Avoid composite dimensions.*** Each dimension should correspond to a single characteristic of the data, not to a combination of characteristics.

- ***Avoid creating a new code list where one already exists.*** It is highly recommended that structural definitions and code lists be consistent with internationally agreed standard methodologies, wherever they exist, e.g., System of National Accounts 1993; Balance of Payments Manual, Fifth Edition; Monetary and Financial Statistics Manual; Government Finance Statistics Manual, etc. When setting-up a new data exchange, the following order of priority is suggested when considering the use of code lists:

- international standard code lists;
- international code lists supplemented by other international and/or regional institutions;
- standardised lists used already by international institutions;
- new code lists agreed between two international or regional institutions;
- new specific code lists.

The same code list can be used for several statistical concepts, within a key family or across key families.

- ***Key family definition.*** The following items have to be specified by a structural definitions maintenance agency when defining a new key family:

[1] If it is decided not to create a separate key family then, for the set of time series for which the dimension is not relevant, a value such as "non-applicable", "non-defined" "all" or "total" has to be assigned to this dimension.

- Key family identification:
    - key family identifier
    - key family name

- A list of coded statistical concepts assigned as dimensions of the key family. For each:
    - statistical concept identifier
    - statistical concept name
    - ordinal number of the dimension in the key structure
    - code list identifier
- A list of statistical concepts assigned as attributes for the key family. For each:
    - statistical concept identifier
    - statistical concept name
    - code list identifier if the concept is coded
    - assignment status: mandatory or conditional
    - attachment level
    - maximum text length for the uncoded concepts
    - maximum code length for the coded concepts
- A list of the code lists used in the key family. For each:
    - code lists identifier
    - code list name
    - code values and descriptions

- ***Definition of data flow definitions.*** Two (or more) partners performing data exchanges in a certain context need to agree on:
    - the list of data set identifiers they will be using;
    - for each data flow:
        - its content and description
        - the relevant key family definition

- ***Mandatory attributes.*** Once the key structure of a key family has been decided, then the set of mandatory attributes of this key family has to be defined. In general, some statistical concepts are necessary across all key families to qualify the contained information. Examples of these are:
    - Reference area
    - Frequency (always a dimension)
    - A descriptive title (see also comment below)
    - Collection (e.g. end of period, averaged or summed over period)
    - Unit (e.g. currency of denomination)
    - Unit multiplier (e.g. expressed in millions)
    - Availability (which institutions can a series become available to)
    - Decimals (i.e. number of decimal digits used in a time series)
    - Observation Status (e.g. estimate, provisional, *normal*)

Therefore, those concepts which are not dimensions within a key family have to be present in that key family as mandatory attributes. Moreover, additional attributes may be considered as mandatory when a specific key family is defined.

4.1.3 Time and Frequency

While it is not required that a key family designed to provide only cross-sectional presentations have the concept of Time associated with the data it describes, this is a very unusual case. For all key families which use the Time concept, it is strongly recommended that the concept Frequency also be used in the key family as a dimension. While this may not seem to be important for some publishers and disseminators of statistics, the lack of a Frequency can create difficulties with many systems in the presentation and processing of statistics.

Conventionally, Frequency is the first dimension in the key. Frequency is typically a value from the following list, although it may be necessary to make additions to this list for specific uses:

A	Annual
B	Business (often not supported)
D	Daily
E	Event (often not supported)
H	Semi-annual
M	Monthly
Q	Quarterly
W	Weekly

For reasons related to backward compatibility with existing systems, there is a corresponding concept of “TIME_FORMAT”, which is needed in the formats to describe how time is formatted. TIME_FORMAT is included in the key family as a required series-level attribute. However, when the key family definition is serialised as SDMX-EDI the TIME_FORMAT is declared as a dimension (which, in sequence, is placed immediately after the time dimension), and not as an attribute. In the SDMX-ML rendering it is declared as defined in the key family (i.e. as a series-level attribute).

In the XML representation, the TIME_FORMAT is usually a value taken from the following list (meanings as defined in ISO 8601):

P1Y – Annual
P6M – Semi-annual
P3M – Quarterly
P1M – Monthly
P7D – Weekly
P1D – Daily
PT1M – Minutely

For SDMX-EDI, there is a syntax-specific list of relevant codes taken from the code list associated with the UN/EDIFACT TDID data element 2379 - Date or time or period format code.

Applications processing time ranges in either SDMX-EDI or SDMX-ML must know how to iterate time such as knowledge of leap years and 53 weeks years. For the latter the calculation of weeks is according to ISO 2017 (simply put, this states that the first week in a year is the week that contains the first Thursday of the year).

Time ranges are expressed in SDMX-ML simply by omitting the time value from the observation for all except the first observation (supported only by the CompactData message). In SDMX-EDI the time period is expressed as a time range by declaring begin and end periods. This can be used to validate whether all the observations are present for the time series. As the SDMX-ML declares only the beginning period, it is recommended that the time period is also present in the last observation, so that a similar validation can be performed.

Additional attributes may be necessary to specify such items as whether the time period specified is the beginning or end of period, etc. For example, a monthly series may contain observations taken at the beginning, or the middle, or end of the month, and it may be important for this metadata to be attached as an attribute.

If a key family which uses time does not use the concept of Frequency, then it cannot use certain specific features of the formats (such as expressing time ranges in SDMX-EDI and the CompactData message in SDMX-ML.)

4.1.4 Exchanging Attributes

4.1.4.1 Attributes on series, sibling and data set level

- *Static properties*.
    - Upon creation of a series the sender has to provide to the receiver values for all mandatory attributes. In case they are available, values for conditional attributes should also be provided. Whereas initially this information may be provided by means other than SDMX-ML or SDMX-EDI messages (e.g. paper, telephone) it is expected that partner institutions will be in a position to provide this information in SDMX-ML or SDMX-EDI format over time.
    - A centre may agree with its data exchange partners special procedures for authorising the setting of attributes' initial values.
    - Attribute values at a data set level are set and maintained exclusively by the centre administrating the exchanged data set.
- *Communication of changes to the centre*.
    - Following the creation of a series, the attribute values do not have to be reported again by senders, as long as they do not change.
    - Whenever changes in attribute values for a series (or sibling group) occur, the reporting institutions should report either all attribute values again (this is the recommended option) or only the attribute values which have changed. This applies both to the mandatory and the conditional attributes. For example, if a previously reported value for a conditional attribute is no longer valid, this has to be reported to the centre.
    - A centre may agree with its data exchange partners special procedures for authorising modifications in the attribute values.
- *Communication of observation level attributes “observation status”, "observation confidentiality", "observation pre-break"*
    - In SDMX-EDI, the observation level attribute “observation status” is part of the fixed syntax of the ARR segment used for observation reporting. Whenever an observation is exchanged, the corresponding observation status must also be exchanged attached to the observation, regardless of whether it has changed or not since the previous data exchange. This rule also applies to the use of the SDMX-ML formats, although the syntax does not necessarily require this.
    - If the “observation status” changes and the observation remains unchanged, both components would have to be reported.
    - For key families having also the observation level attributes “observation confidentiality” and "observation pre-break" defined, this rule applies to these attribute as well: if an institution receives from another institution an observation with an observation status attribute only attached, this means that the associated observation

confidentiality and pre-break observation attributes either never existed or from now they do not have a value for this observation.[1]

4.2 Best Practices for Batch Data Exchange

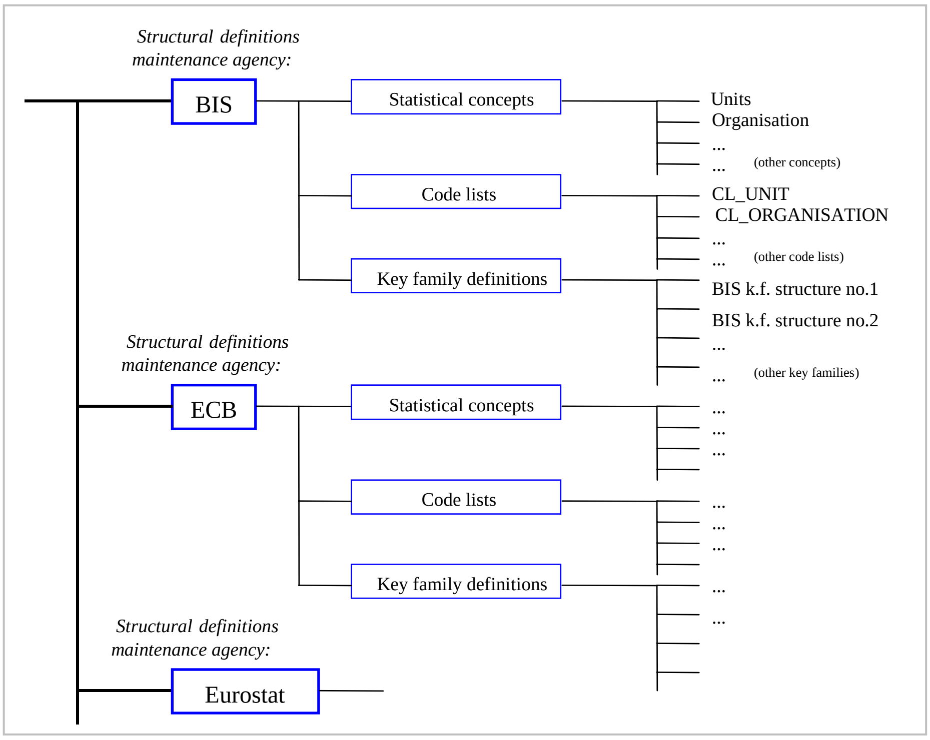
Batch data exchange – the exchange and maintenance of entire databases between counterparties – is an activity that often employs SDMX-EDI formats, and might also use the SDMX-ML CompactDataMessage. The following points apply equally to both formats.

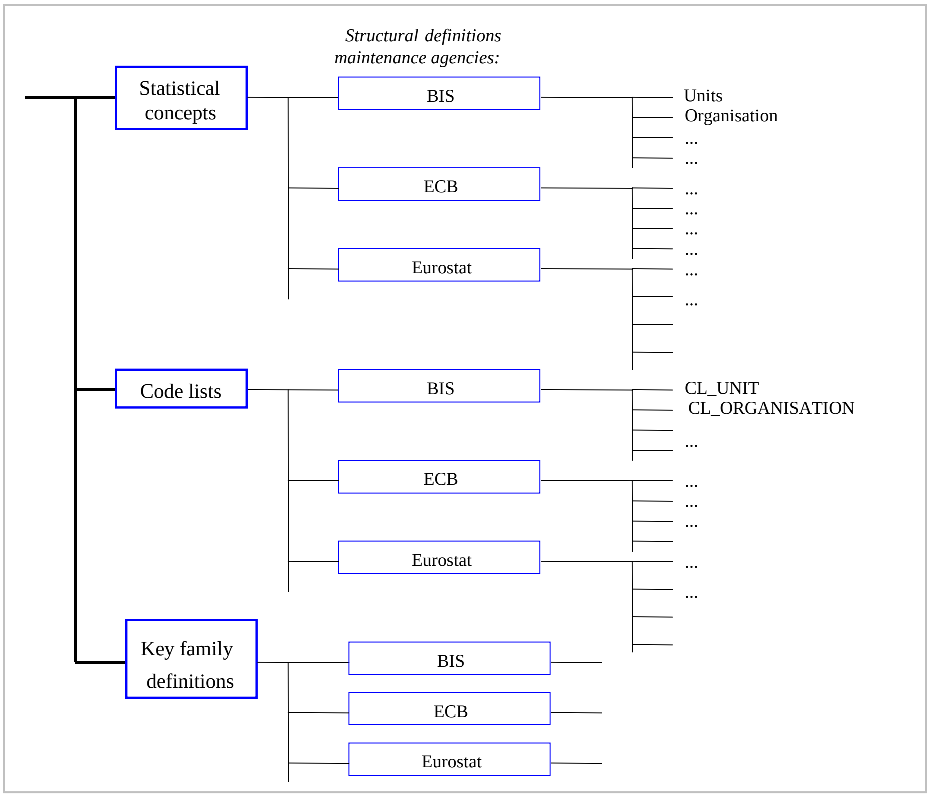
4.2.1 More Than One Central Institutions Involved in a Data Exchange

In the paragraph discussing the role of central institutions, it was mentioned that though, usually, a central institution administrates the exchange of data sets based on the structural definitions it devises. There may be other cases in which a third institution's structural definitions could be used in a data exchange. In this case, the central institution administrating this data set(s) should take care (possibly in co-operation with the corresponding structural definitions maintenance agency) that the necessary structural definitions become known to all data exchange partners involved and that the corresponding SDMX structural definition messages are properly maintained and, if necessary, appropriately updated.

SDMX gives the possibility to partner institutions to design generic data exchange systems which can take into account the role of central institutions in devising structural definitions. In principle, each institution should design its system in such a way that could cope with an environment in which more than one structural definitions maintenance agency could exist. For example, the following figures describe alternative ways to organise structural definitions assuming the existence of three central institutions (e.g. BIS, ECB, Eurostat). In practice, more central institutions could be envisaged and, therefore, more central branches in the tree; including possibly even the home institution (e.g. a central bank or statistical institute) if the home institution plays a role in "devising" structural definitions within a user community.

[1] However, this logic does not apply to the observation comment attribute. If it is not received in an interchange and if it had previously existed, that previously received value should be still kept in receiver's database (the "updates and revisions" principle applies).


Structural definitions maintenance agency:
BIS
Statistical concepts
Units
Organisation
...
... (other concepts)
Code lists
CL_UNIT
CL_ORGANISATION
...
... (other code lists)
Key family definitions
BIS k.f. structure no.1
BIS k.f. structure no.2
...
... (other key families)
Structural definitions maintenance agency:
ECB
Statistical concepts
Code lists
Key family definitions
Structural definitions maintenance agency:
Eurostat


Structural definitions maintenance agencies:
Statistical concepts
BIS
Units
Organisation
ECB
Eurostat
Code lists
BIS
CL_UNIT
CL_ORGANISATION
ECB
Eurostat
Key family definitions
BIS
ECB
Eurostat

4.2.2 Positioning of the Dimension "Frequency"

The position of the “frequency” dimension is unambiguously identified in the key family definition. Moreover, most central institutions devising structural definitions have decided to assign to this dimension the first position in the key structure. This facilitates the easy identification of this dimension, something that it is necessary to frequency's crucial role in several database systems and in attaching attributes at the sibling group level.

4.2.3 Identification of Key Families

In order to facilitate the easy and immediate recognition of the structural definition maintenance agency that defined a key family, most central institutions devising structural definitions use the first characters of the key family identifiers to identify their institution: e.g. BIS_MACRO, EUROSTAT_BOP_01, ECB_BOP1, etc.

4.2.4 Identification of the Data Sets

In order to facilitate the easy and immediate recognition of the institution administrating a data set, many central institutions prefer to use the first characters of the data set identifiers to identify their institution: e.g. BIS_MACRO, ECB_BOP1, ECB_BOP1T, etc.

4.2.5 Special Issues

4.2.5.1 <u>"Frequency" related issues</u>

- ***Special frequencies.*** The issue of data collected at special (regular or irregular) intervals at a lower than daily frequency (e.g. 24 or 36 or 48 observations per year, on irregular days during the year) is not extensively discussed here. However, for data exchange purposes:

  - such data can be mapped into a series with daily frequency; this daily series will only hold observations for those days on which the measured event takes place;

  - if the collection intervals are regular, additional values to the existing frequency code list(s) could be added in the future.

- ***Tick data.*** The issue of data collected at irregular intervals at a higher than daily frequency (e.g. tick-by-tick data) is not discussed here either. However, for data exchange purposes, such series can already be exchanged in the SDMX-EDI format by using the option to send observations with the associated time stamp.