Statistical Data and Metadata Exchange Initiative

Draft SDMX Technical Standards (Version 2.0) - Disposition Log – Project Team

No.	Organis-ation	Document/ Standard	Line No.	Problem/Issue	Suggested Solution	Disposition
1	Project Team	Registry Interfaces logical specification	general	In the document "Framework for SDMX technical standards, version 2) it is stated (line 331) that "data structure definition" is a synonym for "key family". In the registry interfaces document there are frequent references to "metadata structure definitions" eg in line 318 "What key family or metadata structure definition is used by the registered data?" However, the wording of lines 221-225 leaves the reader wondering what kind of metadata are being referred to here. Note that in line 277 the document refers to "Reference Metadata Structure Definitions".	- restate definition of "key family" (eg in section 3) - introduce "reference metadata" into definition of "metadata structure definition" - review all use of these terms for possible ambiguity. - consider using the term "Reference Metadata Structure Definitions" throughout, to avoid ambiguity.	Comment noted. It is important that the terminology of the model be used, but the text will be clarified as suggested where appropriate.
2	Project Team	Registry Interfaces logical specification	general (see below for examples)	In several places, the document seems to go beyond the description of a logical specification to describe how the specification might be implemented.	Either: 1) drop the description of possible implementations and limit the document to a pure logical specification or 2) it should be made clear where an implementation option is included by way of explanation or as an	Comment noted and agreed. The "implementation" status of the document is indicated by a designation of many sections as non-normative. This has been discussed, and will be clarified, to indicate which

No.	Organis-ation	Document/ Standard	Line No.	Problem/Issue	Suggested Solution	Disposition
					example.	sections are specification (normative) and which are more explanatory (non-normative)
3	Project Team	Registry Interfaces logical specification	246 356-374 962-992	Why are email and HTTP post the only subscription/notification mechanisms mentioned? Could RSS also be envisaged?

This may be an example of an area where the document goes too far towards describing an implementation of the SDMX registry, rather than describing the standard. | The document could limit itself to describing the content of a notification/subscription, as in section 11.1. | A standard format for notifications is needed, to support the possibility for generic applications. RSS was considered, but it lacks sufficient granularity in terms of the model. The use of RSS is actually complementary to the envisaged use of subscription/notify-cation, and its use in addition to the specified mechanism is anticipated. |
| 4 | Project Team | Registry Interfaces logical specification | 481-482 | Reference to HTTP GET is too implementation specific. Other mechanisms could be envisaged. | Remove, or cite as example only (see above). | Comment noted – this will be changed to include the other standard option (SOAP), and it will be pointed out that this is a minimal requirement. |
| 5 | Project Team | Registry Interfaces logical specification | 513 | "Structural metadata definition": in this paragraph, | If so, then use standard terms "key families" or | This is a general term for both key |

No.	Organis-ation	Document/ Standard	Line No.	Problem/Issue	Suggested Solution	Disposition
				does this mean "key families"?	"data structure definitions" (see above)	families and metadata structure definitions. We will clarify.
6	Project Team	Registry Interfaces logical specification	513-514	Reference to HTTP GET and SOAP is implementation specific	Remove, or cite as example only (see above).	A minimal standard implementation is needed, but the document will changed to indicate this.
7	Project Team	Registry Interfaces logical specification	580	What is the meaning of the reference to "service level agreements in those scenarios that are based on legal directives". An SLA would normally imply that there was no legal act, since a legal act would itself specify what was expected from a data provider. Also, neither an SLA nor a legal act can be "imposed" – they are always mutual agreements.	Delete this sentence, or reword along the lines : "data provisioning can reflect at a technical level the data transmission provisions of a legal act or other agreement".	Comment noted – text will be changed.
8	Project Team	Registry Interfaces logical specification	621-624	Reference to REST datasource and protocol: this is not explained and in any case it is implementation specific.	Remove, or cite as example only (see above).	This is a minimum standard requirement, and will be made consistent with similar cases.
9	Project Team	Registry Interfaces logical specification	653	"Key Family Definition": what is the difference between a "Key Family" (= Data Structure Definition, see above) and "Key Family Definition".	If some difference is intended, it should be explained. Otherwise just use "Key Family" here.	Comment noted – will fix.
10	Project Team	Registry Interfaces logical specification	719-731	The description of how to specify publication and release dates is confusing	Either 1) drop this part or	The combination of periodicity and offset provides for

No.	Organis-ation	Document/ Standard	Line No.	Problem/Issue	Suggested Solution	Disposition
				and probably not useable as it stands. In the context of SDMX, the distinction between "publication" and "release" is unclear: surely they would be the same? One might use the concept of "reference period" (or date) and the permitted period until data are transmitted, eg "5 months after reference period".	2) replace with attributes - reference period - expected delay before release/transmission (but be aware that this is very hard to implement due to the oddities of the calendar…)	the definition of periodic time points on which data will be released. The Tolerance attribute specifies the acceptable delay between specified and actual release dates. This full semantic is needed to support the computability of a wide variety of variations within institutions.
11	Project Team	Registry Interfaces logical specification	817ff	An example would be very useful here.	Insert example	This just shows how the registration message is constructed from fields in the data set, including its header. Additional examples are important, and will be provided either as part of the specification or as part of other documentation.
12	Project Team	Registry Interfaces logical specification	963-971	These lines are repeated as lines 974-982.		Comment noted – will fix.
13	Project Team	SDMX Information Model: UML conceptual Design	274, 280, 913	It is said that for some classes the attribute compartment is suppressed "to aid clarity". This is useful but, rather than aiding clarity, it improves diagrams	Change "aid clarity" with "improve readability".	The attribute compartment is suppressed either when there are no attributes (other than those

No.	Organis-ation	Document/ Standard	Line No.	Problem/Issue	Suggested Solution	Disposition
				readability. In general, I think that omitting information makes things less clear.		inherited) or when the class is referenced on a diagram other than the diagram on which it is first introduced and described. This was the intent of "aid clarity". Additional explanation, as above, will be added to the text.
14	Project Team	SDMX Information Model: UML conceptual Design	421	There is no relationship between data use and publication.	Add a "uses" link between the "Uses Data" use case and the "Publish data" one, and between "Uses metadata" and "Publish Reference Metadata".	Agree that there is a link and it will be added to the diagram.
15	Project Team	SDMX Information Model: UML conceptual Design	581	"ComponentStructure" is mentioned, but it is not part of the described diagram.	Change with "Structure"?	Agree. This will be changed.
16	Project Team	SDMX Information Model: UML conceptual Design	614	In the "Feature" column, the "Attribute" class is sometimes reported in italic fonts, to indicate that it's an abstract class, and sometimes in plain fonts.	Check for consistency: always report "Attribute" in italic fonts.	This will be done.
17	Project Team	SDMX Information Model: UML conceptual Design	1032	The term "EntityDimension" to indicate the subject to whom the data refers could be misleading because is too generic and can be confused with the "Entity" of the Entity-relationship model.	Call it "TableSubject Dimension"	The term "Entity" is a recognized term for this type of information.

No.	Organis- ation	Document/ Standard	Line No.	Problem/Issue	Suggested Solution	Disposition
18	Project Team	SDMX Information Model: UML conceptual Design	1124	It is not clear what the "CountDimension" is good for.	Add a rationale for this dimension type.	This will be explained in the text.
19	Project Team	SDMX Information Model: UML conceptual Design	1159	The "UncodedMeasure" class is reported twice.	The second occurrence should be "UncodedXSMeasure"	Agree. This will be changed.
20	Project Team	SDMX Information Model: UML conceptual Design	1218	The "describedBy" feature for the "DataSet" Class does not exist.	Change it with "defines", as in the diagram.	Agree. This will be changed.
21	Project Team	SDMX Information Model: UML conceptual Design	1797, 1831, 1906 (first row)	The term CodeSet is not reported in any diagram.	Maybe it should be "CodeAssociation"	This should be Hierarchical Code Scheme and will be changed in the text.
22	Project Team	SDMX Information Model: UML conceptual Design	2225-2226	What happens with root classes, with no parent? What is the lifecycle of a class?	Clarify	All classes in the model have some parent, the ultimate parent is the Maintenance Agency. The Maintenance Agency is the only concrete class that does not itself have a parent. The maintenance agency is not in such an aggregation – in the model it just exists.

The class itself does not have a lifecycle, only the instances have lifecycles – the refers to the class |

No.	Organis-ation	Document/ Standard	Line No.	Problem/Issue	Suggested Solution	Disposition
						instance. The model does not specify lifecycle events.
23	Project Team	SDMX Information Model: UML conceptual Design	2230	The diagram is inconsistent with the one in figure 12.	"ComponentStructure" should be "Structure"	Agreed. This will be changed.
24	Project Team	SDMX-ML: Schema and Documentation	242-243	The sentence is not clear.	Define the concept of "validation profile".	Comment noted – we will clarify.
25	Project Team	SDMX-ML: Schema and Documentation	271-273	The sentence is not clear.	Define the concept of "external entity or identifier".	This means "external to the specification" and will be clarified.
26	Project Team	SDMX-ML: Schema and Documentation	887-1063	The concept of time is absent in the "CategorySchemeType". The requirement of having temporal category schemes is fundamental in many contexts. See for example the need of keeping track of the temporal evolution of hierarchical code lists.	Insert this concept, where appropriate, in abstract and concrete classes modeling hierarchical categories. At least in those modeling hierarchical code lists.	Comment noted. Attributes for capturing time on hierarchical codelists will be added.
27	Project Team	SDMX-ML: Schema and Documentation	1018-1020	The usage of alias in this context is unclear.	Describe more in detail what is a reference from a hierarchical code list and why alias simplify it.	Comment noted – this will be clarified. The Alias allows for the assignment of external names to specific nodes when the SDMX-ML ID of the node doesn't match an external scheme for technical reasons.
28	Project Team	Schemas – samples	General	Examples are incomplete. Optional attributes are mostly left out and there is only a query example for data sets.	Produce examples covering all the attributes mentioned in the schema definitions.	Comment noted. Examples will be extended, either in the specification or

No.	Organis-ation	Document/ Standard	Line No.	Problem/Issue	Suggested Solution	Disposition
					Produce different data set query examples, covering many different types of "DataWhere" conditions.	in additional documentation.
29	Project Team	SDMX-ML: Schema and Documentation f	2122	A reference is made to TimePeriodType; however there appears to be a missing convention for the specification of TIME format in the Query message.	Include the specification of TIME format in the Query message schema	XML Schema requires the use of ISO time formats, which specifies time formats.
30	Project Team	GenericSample.xml		The special status Time dimension in the Generic Data schema makes it unusable for non-time-series data. There is need for a 'more' generic GenericData message by treating Time in the same way as other other concepts/dimensions	Remove the special status of time in the GenericData schema.	Agreed.
31	Project Team	SDMX-ML: Schema and Documentation	1957	Query message allows the requesting of Data from more than one dataset; however the GenericData schema only allows data from one dataset	Allow multiple dataset data in GenericData message.	This is possible using MessageGroups, with the problem that all of the datasets take only one header. This will be corrected in the 2.0 schemas.
32	Project Team	QuerySample.xml		2 frequencies 'A' and 'M' are specified in the And node of the Query Message. Conceptually data observations cannot be attributable to 2 different frequencies at the same time	Remove one of the frequencies in this example XML.	Noted, and will be fixed.
33	Project Team	SDMX Web Services Guidelines	215-296	Web service method names are specified here; however	Expand this documentation to	The "parameters" are always

No.	Organis-ation	Document/ Standard	Line No.	Problem/Issue	Suggested Solution	Disposition
				there is no specification of parameters and naming standards for developing web services with SDMX. This is important so that users can call web services for SDMX data using standard methods, parameters and corresponding names. This is also important for Registry implementation.	include standard, methods, parameters required and corresponding standard naming conventions.	expressed as SDMX-ML types, because SOAP RPC is not supported. This will be documented better to show how this mechanism works for each standard function.
34	Project Team	SDMX Web Services Guidelines		There are no specifications for compression and securisation (encryption) for transmitting SDMX-ML between systems. These standards would help implementation and optimization of data sharing using SDMX-ML	Develop specifications for compression and securisation (encryption) for transmitting SDMX-ML.	This point will be discussed, but some technical difficulties will need exploration and testing. This requirement may not be met in the current version of the spec.