Distr. GENERAL

25 April 2013

WP.9

ENGLISH ONLY

UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE
CONFERENCE OF EUROPEAN STATISTICIANS

EUROPEAN COMMISSION
STATISTICAL OFFICE OF THE
EUROPEAN UNION (EUROSTAT)

ORGANISATION FOR ECONOMIC COOPERATION
AND DEVELOPMENT (OECD)
STATISTICS DIRECTORATE

Work Session on Statistical Metadata
(Geneva, Switzerland, 6-8 May 2013)
Topic (i): Metadata standards and models

DESIGNING A METADATA REPOSITORY

Prepared by Chris Nelson & Matt Nelson, Metadata Technology Ltd.

I. Introduction

1. Metadata has many forms, it is everywhere, it is authored in many ways using different tools and is consequently stored in many forms, it is often not in a centralised accessible resource and consequently linking the metadata to the construct or data slice to which it relates at a granular level is difficult.

2. There are many metadata standards which have models supporting the metadata required to support, to a greater or less extent, different aspects of the Generic Statistical Business Process Model (GSBPM): DDI (Data Documentation Initiative); SDMX (Statistical Data and Metadata Exchange); Neuchâtel; ISO/IEC 11179; and more recently, GSIM (Generic Statistical Information Model), and many others (see http://www1.unece.org/stat/platform/pages/viewpage.action?pageId=14319930). The SDMX standard (ISO 17369:2013) has also standardised the way data can be queried and retrieved from a database and the related structures from a structural metadata repository.

3. However, to date there is very little support for standardising the way metadata repositories can work with what SDMX calls "referential metadata" such as the metadata concerned with data quality frameworks, or the metadata that relates directly to data but which is authored independent from the data and is not stored with the data in the database. So, even with SDMX, which has a very generic and powerful model for supporting this type of metadata, building a metadata repository that is able to link these "referential metadata" to the data or the structural metadata to which they relate, and to link to these metadata in the responses to queries for data or structural metadata queries, has challenges.

4. The metadata repository Metadata Technology is building has a novel approach on how to achieve this for data queries. Whilst the metadata repository supports SDMX (adding relevant metadata URLs to the response from structure and data queries), the design issues

discussed in this paper are relevant to the use of any formal model that supports queries for data and structural metadata and the attachment of these metadata in any query response.

II. Scope of this Paper

5. Unless otherwise stated in the context of this paper "metadata" means what SDMX calls referential or footnote metadata such as may be used in data quality frameworks as used by Eurostat, IMF,ILO, OECD and many others, and other metadata that is relevant to actual data disseminated in a "data set" but is not an integral part of the data as stored in the statistical database.

6. The metadata repository we are building supports the SDMX Information Model which means it supports metadata that is structured according an SDMX Metadata Structure Definition (MSD). The design has focussed on real use cases that have resulted from a number of projects where we have been asked to develop an MSD and projects where we have developed a metadata repository to service specific needs.

7. We have given careful thought to these use cases as they raise interesting issues that must be solved when designing and building a generic metadata repository that can be used by any organisation that needs to author, collect, manage, and disseminate metadata. Some of these issues are related directly to SDMX, but many are quite generic in nature and relate to any metadata repository that is built to support a specific standard, rather than the needs of a specific organisation.

8. The scope of this paper is to identify each of these issues, to discuss the issue and possible solutions, and how this has influenced the design approach taken by Metadata Technology.

III. Design Issues

9. As already mentioned most of the design issues are independent of any particular standard and relate to any metadata repository that is built for use by multiple organisations or communities. These are:

(a) How to query for metadata: in SDMX, for example, the standard is very clear on how to query for data but there is no equivalent simple query format for metadata that lends itself to use by web clients.

(b) What should the metadata repository return when queried: should this be a link to the metadata or the actual metadata.

(c) How does the web client get informed of the presence of metadata: should this be separated from the data set or embedded with the data, even though it was authored independent of the data.

(d) Does the data store need to know about the metadata repository or can the data store be totally de-coupled from the metadata repository.

(e) How does the GUI know precisely to which object metadata is attached: it is simple to attach metadata to data as data has a precise key, but metadata related to structural metadata can be more complex and it is necessary to have a unique and unambiguous way of identifying the “context” of the metadata.

(f) What input and output formats should be supported for the metadata: for SDMX systems clearly SDMX will need to be supported but there are other more web friendly formats such as JSON for which SDMX has developed a data format but with limited support for metadata. Also pdf could be a popular format for longer reports, or HTML.

(g) For SDMX, how best to support the authoring of the Metadata Structure Definition (MSD): the MSD is very powerful but also extremely generic and it is easy to author MSDs that will result in metadata that will not be understood by the metadata repository or other generic software (see issue (e)). It is necessary to ensure only meaningful MSDs are developed that Ican be understood unambiguously by the metadata repository and software processing the metadata returned from the repository.

(h) How to attach the same metadata to multiple objects: metadata often travels through the statistical lifecycle and may be used by multiple artefacts in this lifecycle.

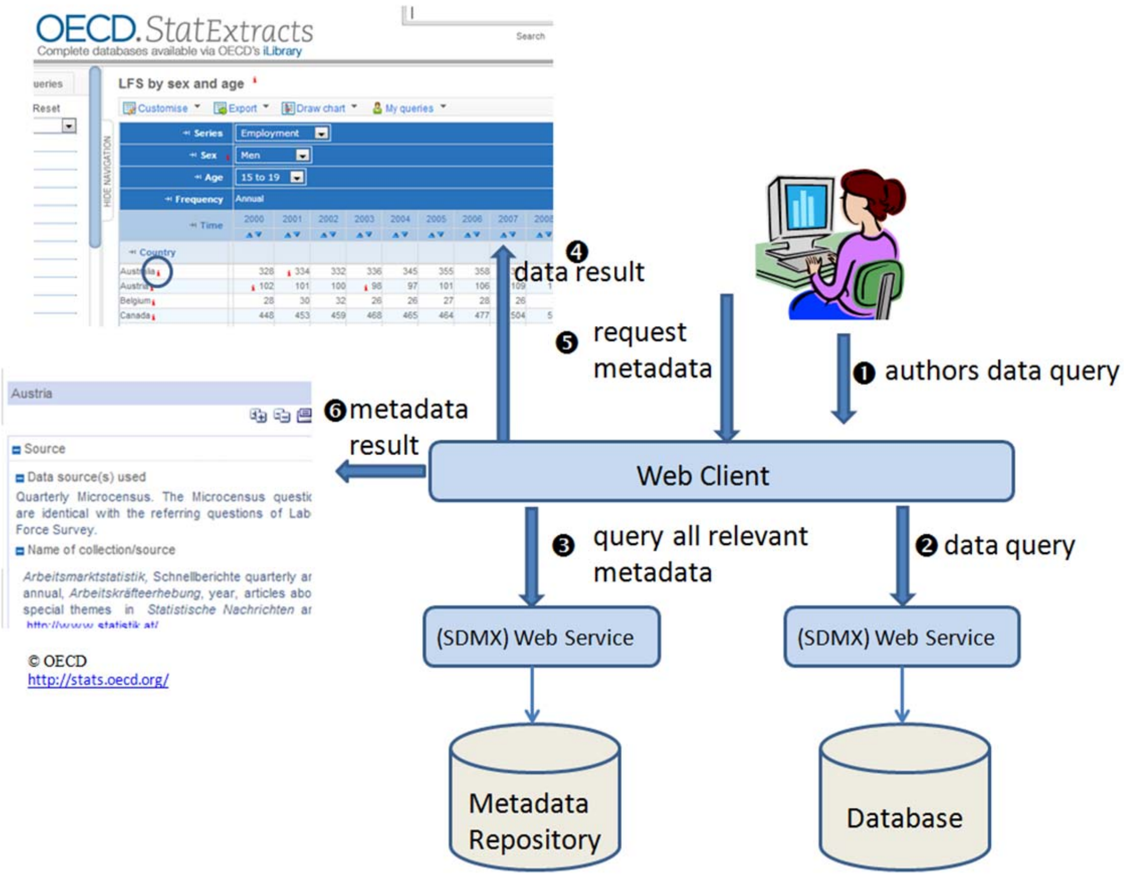
IV. Discussion of the Issues

A. Querying for Data and Related Metadata

10. The issue here is how best to support an application that requires data and any related referential metadata. It is probable that neither the application (such as a web client) nor the user will know how to query for data and related referential metadata in the same query, and, at least for SDMX, such a query is not possible and it is also not possible to combine any referential metadata with the data in the same response.

11. There are three basic ways that data and related referential metadata can be queried. Note that although the example is taken from the OECD it does not imply that the exact mechanism used at the OECD relates to any of the scenarios described below.

1. Scenario 1



Series	Employment								
Sex	Men								
Age	15 to 19								
Frequency	Annual								
Time	2000	2001	2002	2003	2004	2005	2006	2007	2008
Country									
Australia	328	334	332	336	345	355	358		
Austria	102	101	100	98	97	101	106	109	1
Belgium	28	30	32	26	26	27	28	26	
Canada	448	453	459	468	465	464	477	504	5

Quarterly Microcensus. The Microcensus questic are identical with the referring questions of Lab Force Survey.

Arbeitsmarktstatistik, Schnellberichte quarterly ar annual, *Arbeitskräfteerhebung*, year, articles abo special themes in *Statistische Nachrichten* ar http://www.statistik.at/

Figure 1: Web client is responsible for querying the metadata repository

12. Here the web client queries for the data (2), obtains the result, determines the “key” for each possible point at which metadata can be available, and queries the metadata repository for all these points (3). There are two possible types of query response:

(a) one that returns all the metadata

(b) one that responds “true” or “false”.

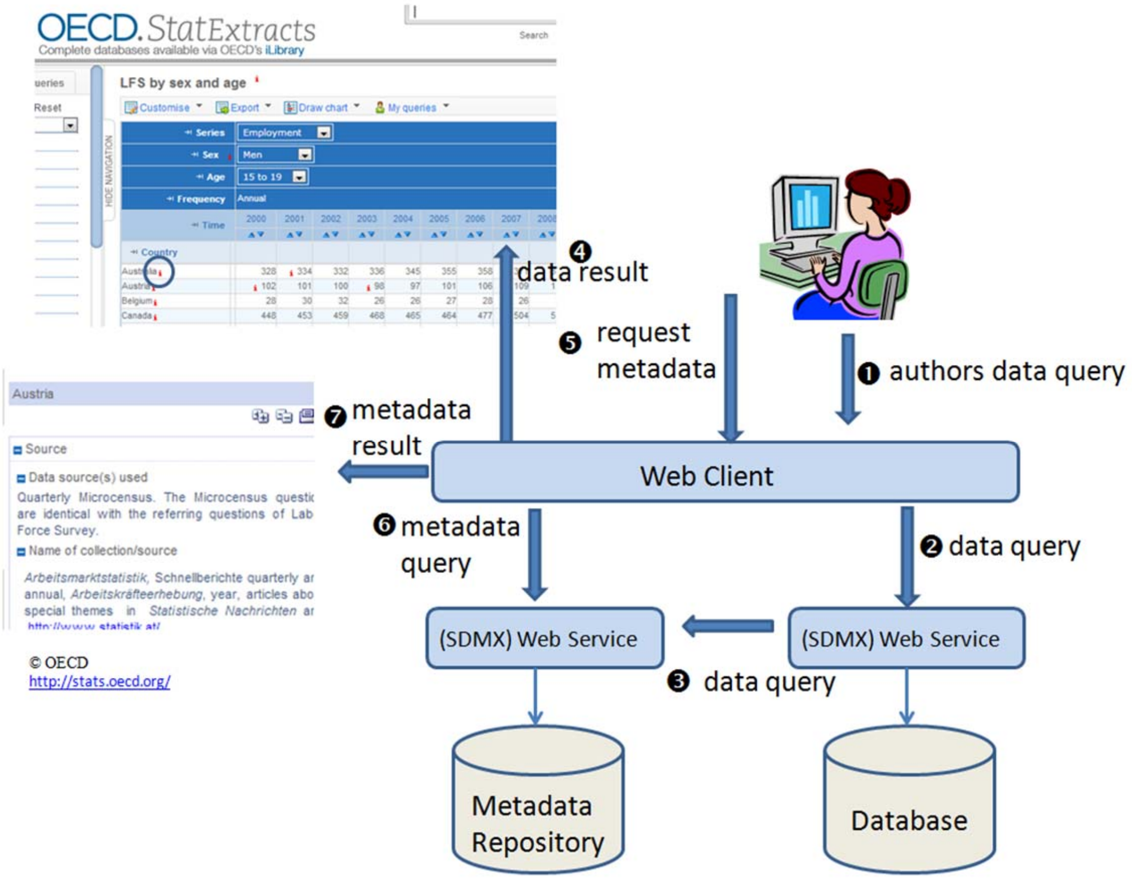
13. Both result in the web client placing an “i” at the point at which there is metadata (4). When the user clicks on the “i” (5) the web client retrieves the metadata, either from its own stored result (response (a) above) or from a query to the metadata repository (response (b) above).

14. The disadvantages of this approach are:

(a) The web client needs know how to query for the metadata.

(b) The web client needs to make multiple queries to the metadata service, one for each data point.

2. Scenario 2



OECD.StatExtracts
Complete databases available via OECD's iLibrary

LFS by sex and age

Series	Employment							
Sex	Men							
Age	15 to 19							
Frequency	Annual							
Time	2000	2001	2002	2003	2004	2005	2006	2007
Country								
Australia	328	334	332	336	345	355	358	[illegible]
Austria	102	101	100	98	97	101	106	109
Belgium	28	30	32	26	26	27	28	26
Canada	448	453	459	468	465	464	477	504

Austria

Source

Data source(s) used

Quarterly Microcensus. The Microcensus questic are identical with the referring questions of Lab Force Survey.

Name of collection/source

Arbeitsmarktstatistik, Schnellberichte quarterly ar annual, *Arbeitskräfteerhebung*, year, articles abo special themes in *Statistische Nachrichten* ar http://www.statistik.at/

© OECD
http://stats.oecd.org/

Figure 2: Database service is responsible for querying the metadata repository

15. A better scenario is to enrich the data response with metadata points. In SDMX this can be achieved by annotating the dataset, series, and observations. Each annotation can link back to a URL from where the actual metadata can be retrieved.

16. In this implementation the database web service receives a data query (2), and it queries for the metadata. The same data query is passed to the metadata repository (3) and the metadata repository responds with all the metadata indexes which are relevant to the data query. The metadata indexes contain the URL from where the metadata can be retrieved, together with information concerning to which part of the dataset it attaches.

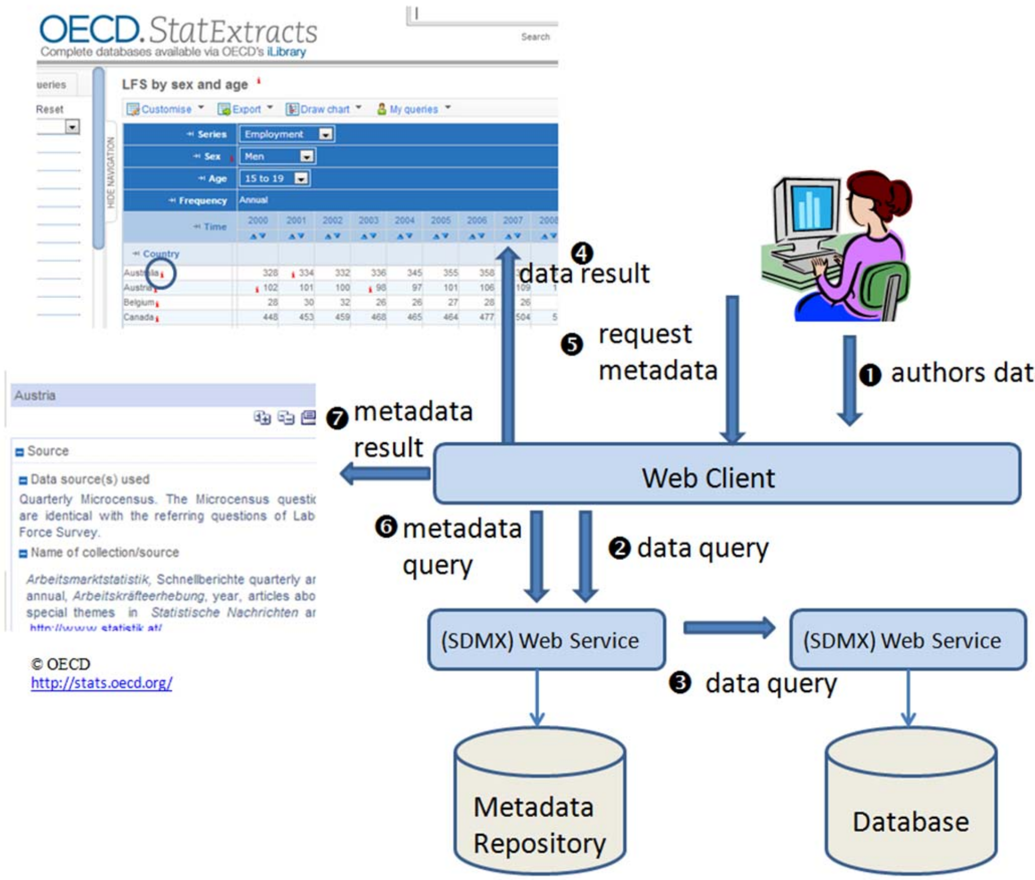
17. The data repository is then responsible for annotating the data response at the correct data locations to indicate the presence of additional metadata. The web client then receives the response, and can use the metadata points to indicate to the user that extra metadata is available (4). When the user wishes to view the additional metadata (5), the web client makes an additional query to the metadata repository (6). The web client knows where to get the metadata from, as it is included in the annotation URL. The requested metadata is returned (7).

18. The disadvantages of this approach are:

(a) The database needs to know about the metadata repository, it needs to query the metadata repository, and it needs to know how to interpret the response and embed the information into the dataset.

(b) It is not possible to retrofit the metadata repository onto existing data web services without enhancing the data web service with the functionality described in (a) above.

3. Scenario 3



OECD.StatExtracts
Complete databases available via OECD's iLibrary

LFS by sex and age

Series	Employment								
Sex	Men								
Age	15 to 19								
Frequency	Annual								
Time	2000	2001	2002	2003	2004	2005	2006	2007	2008
Country									
Australia	328	334	332	336	345	355	358		
Austria	102	101	100	98	97	101	106	109	1
Belgium	28	30	32	26	26	27	28	26	
Canada	448	453	459	468	465	464	477	504	5

Austria

Source

Data source(s) used

Quarterly Microcensus. The Microcensus questic are identical with the referring questions of Lab Force Survey.

Name of collection/source

Arbeitsmarktstatistik, Schnellberichte quarterly ar annual, *Arbeitskräfteerhebung*, year, articles abo special themes in *Statistische Nachrichten* ar http://www.statistik.at/

© OECD
http://stats.oecd.org/

Figure 3: Metadata repository acts as a proxy on top of the data service

19. This scenario looks very similar to scenario 2 but it is architecturally quite different. The difference here is that the metadata repository is in control and passes the data query to the database. The database responds with data and the metadata repository augments this response by adding the metadata. In this way, the metadata web service acts as a proxy on top of one (or even more than one) data service. The metadata repository is responsible for passing the data query onto the data web service, whilst querying its own repository for any relevant metadata. The metadata repository is then responsible for enriching the data message with metadata indexes, as in scenario 2.

20. It is important to note that enriching the data message leads to a separate problem as embedding metadata with data may not be supported easily by the standard being used. Thankfully, SDMX does have the annotation construct that can be used for this.

21. The major advantage of the scenario 3 approach is that the metadata repository can be retro-fitted to existing database systems because the database does not need to have any knowledge of the existence of the metadata repository. Whilst this is also true of scenario 1 this scenario 3 relieves the web client of having to know of the existence of the metadata repository and how to query it.

B. Embedding Metadata with Data

22. Different standards will work in different ways. For SDMX it is possible to embed some forms of metadata in a data set but this is limited, and such metadata needs to be described in a Data Structure Definition (DSD). This then leads to the coupling of the data structure with the metadata structure and this is not a wise way forward and will lead to serious software interoperability problems. However, SDMX does support an extension mechanism in the form of Annotations. Many artefacts in SDMX are "Annotable" including Data Set, Series, Observation.

23. The Annotation is structured so it is possible to identify the type of annotation, the title, and various other constructs for the content. Whilst the ability to annotate objects may seem that SDMX has introduced anarchy as annotations are essentially only processible if the receiver knows exactly the way the annotation is used, it does, nevertheless, give SDMX communities the ability to support all sorts of use cases that are not predictable and so are not catered for explicitly in the standard.

24. There are four major design decisions required for the use of annotations:

(a) What should be passed back to the application in the annotation, the metadata or a link to it? This is easy to resolve as the annotation is quite limited in what it can contain and, in any case, metadata can be quite large and the end user may not want to view or process it. So, the design choice we made is to pass back the link to the metadata and sufficient information to support (b) below.

(b) Which constructs in the annotation should be used and for which purpose? There is a need to give sufficient information to the application to know what sort of metadata is being linked and to which object or set of objects in the data set the metadata pertains.

(c) To which constructs in the data set should the annotations be attached? The metadata can be authored to "attach" to any key, set of keys, partial keys, observations.

(d) Should the metadata repository resolve the partial key and place annotations in each series key that matches the partial key. For example, in SDMX terms the metadata shown in figures 1-3 relate to a specific partial key (Country = Austria). So, should the metadata link(s) be attached to all of the series that contain Country = Austria, or should this partial key be given to the application for it to decide how best to process it? The design choice we made is to place annotations at the observation, or series level if the full key is defined. If a partial key is defined, the annotation is placed at the dataset level, and the application can decide how best to use this or to present it to the user. Note that the metadata shown in figures 1-3 would be attached to a partial key and clearly in the case of this web client, it is shown only in the context of that partial key (i.e. country=Austria).

C. Metadata Output Formats

25. The SDMX web services guidelines for the REST data query states that the output format requested should be specified in the HTTP Header as part of the "content negotiation". This is the chosen approach for the Metadata Technology metadata repository. This approach allows different output formats to be added as users request them. Initial support will be for JSON as this is a friendly format for web developers, and SDMX. Whilst the SDMX format is a part of the SDMX standard, the JSON format is not standardised by SDMX, so it will be necessary to develop a format. If a metadata set JSON format is developed later by SDMX then clearly this will be added. Note that using the Sdmx Source open source components (www.sdmxsource.org) makes it very easy to add new input and output formats to systems built with the Sdmx Source architecture.

D. Attaching metadata to non-data points

26. So far, the discussion has centred around metadata attached to data. However, a lot of metadata is data independent and concerns the process of collecting and disseminating data. These metadata are collected as part of quality frameworks and SDMX has identified and harmonised the identity and meaning of number of concepts that support these frameworks. This harmonisation process took as input already existing quality frameworks used by a variety of institutions. Whilst these "cross domain concepts" are important and organisations are encouraged to use them, these are "guidelines" and a metadata repository must allow organisations to define and describe their own metadata concepts.

27. So, attaching metadata to non-data points is something a metadata repository needs to support. The big problem here is how to identify the "key" of the non-data point. This is supported very well in the SDMX MSD but needs careful thought about what can and what cannot be supported.

28. For instance, attaching metadata to a Concept or to a Data Provider has semantic meaning in the SDMX Information Model, but it would not be meaningful to author metadata which is to be attached to a Concept in the context of Data Provider: whilst this is perfectly valid in an MSD and may have meaning to an individual organisation, it has no semantic meaning in the SDMX Information Model and therefore cannot be processed intelligently by a generic metadata repository.

29. For this reason it is necessary to define a "profile" of what makes sense and what does not make sense, and to ensure that any metadata structure definition that is accessed by the metadata repository fits this profile.

E. Maintaining Metadata

1. Couple or de-couple the structural metadata repository

30. The Metadata Technology metadata repository is de-coupled from the structural metadata, which is retrieved from any SDMX-compliant structural metadata repository, such as an SDMX Registry. The validation process of a Metadata Set and the GUI to support authoring of metadata both retrieve the MSD from the structural metadata repository. However, these structural metadata repositories are unaware of the "profile" of the MSD that can be supported by a metadata repository. For instance, the SDMX Registry has no restrictions on what is allowed in an MSD save for that allowed by the SDMX standard.

31. So not only is the authoring of an MSD using a GUI for the SDMX Registry more complex and less user friendly than a GUI that restricts choices to those supported by the metadata repository, but it is possible that a “non meaningful” MSD is authored. Consequently, it could be better to support the definition of an MSD in the metadata repository as not only would this would give the user a more intuitive and simple GUI for maintaining MSDs, it would lead the user down a path that ensures the definition of a meaningful MSD.

32. This does not mean that the metadata repository must also be a structural metadata repository. The design could be to use the structural metadata repository for accessing all structural metadata (such as code lists and concepts) except for the MSD itself. Nor does this mean that the MSD is “hidden” in the metadata repository. The MSD contained in the metadata repository could also be in the structural metadata, but this would be held as a “stub” with a URI link to the (SDMX query) service that will return the MSD from the metadata repository. This in turn has the implication that the metadata repository must be able to process an SDMX structure query for an MSD and respond with an SDMX MSD.

2. Marking Up Metadata

33. Metadata are often “marked up” with headings, bold, italics, and tables. Regardless of how this is achieved in an editor (there are may such editors available) the metadata needs to be stored in such a way that it can be searched using textual searches and be able to be converted to formats required by the user, such pdf, SDMX, JSON.

3. Organising Metadata

34. Identification, storage, and retrieval of metadata are important to metadata authors, managers, and applications. It is necessary to have a design that:

(a) Allows metadata to be identified uniquely and to be found easily. A lot of metadata is textual in nature and this is often better supported in a NoSQL database which has good in-built indexing and textual query support.

(b) Supports metadata versioning. This could be draft and final versions or metadata for different reporting periods. The choice of how to organise this (all together with the same Id or split with different ids) can be left to the metadata manager, but it is necessary to store the metadata in such a way that historicity is navigable. Note that SDMX supports the identification of a reporting period as part of the metadata identifiers and this is useful if it is required to store and manage these reporting periods in a metadata repository.

(c) Allows the same metadata to be shared between different objects. An example of this in SDMX is the metadata that is input to or output from processes. These metadata may be output from one process and used as input by another process. It would be useful to support queries that can track these metadata through the statistical lifecycle.

35. The solution to (b) and (c) adopted by the Metadata Technology metadata repository is to separate the metadata itself from the identity information thus allowing multiple indexes to the same metadata.

V. Summary

36. Getting the design decisions right are crucial to the development process. The more the problems are analysed before the actual design and program coding starts, the quicker will be the overall development. However, one thing that we have learned in the work we have done with SDMX and metadata is that there will always be organisations that have different metadata requirements than has been experienced so far. On the other hand, we have also learned that the SDMX MSD and related metadata set is flexible enough to support these, it’s just a matter of working out how best to use it. Having said this we may find some things that would make life easier or will aid interoperability for metadata systems, and if we do then we will submit a request for enhancing the standard.