**Implementing the SDMX Istat strategy**

*WP2 – Enhancement of the SDMX infrastructure and its integration into Istat information systems*

D2.2.1–Software module that implements an enhancement of the Web Service Provider module of the SDMX-RI

Technical and functional documentation

|  |  |
| --- | --- |
| **Version:** | 1.0 |
| **Date:** | 30/04/2015 |
| **Person in charge:** | Alessio Cardacino |
| **Approved by:** |  |

***Table of contents***

[1 Introduction 3](#_Toc419726179)

[2 Architecture of the application 5](#_Toc419726180)

[3 Technical features 7](#_Toc419726181)

Introduction

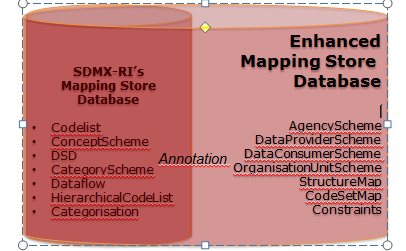
The document describes the technical features and functionalities of the enhanced SDMX-RI web service for querying and submitting (inserting, replacing, deleting) the SDMX structural metadata (artefacts)

The artefacts that can be queried and submitted are the following:

* *CodeList,* ItemScheme artefact representing a set of Codes
* *ConceptScheme,* ItemScheme artefact representing a set of Concepts
* *DSD,* artefact representing thestructure of data in terms of Dimensions (identifying components), Attributes (describing documents), Measures (component indicating the observed value)
* *CategoryScheme,* ItemScheme artefact representing a set of Categories
* *Dataflow,* artefact representing conceptually set of data structured according to a specific DSD
* *HierarchicalCodeList* artefact composed by hierarchies of codes coming from selected codelists
* *Categorization,* artefact linking a category of a specific CategoryScheme and a selected artefact
* *AgencyScheme,* ItemScheme artefact representing a set of Agencies
* *DataProviderScheme* ItemScheme artefact representing a set of Data Provider organizations
* *DataConsumerScheme* ItemScheme artefact representing a set of Data Consumer organizations
  + *OrganizationUnitScheme* ItemScheme artefact representing a set of units inside an organization
  + *Constraint:* artefact representing the selection of codes of the codelists linked to the components of a specific DSD or of a DSD referenced by a dataflow
  + The *CodeSetMap:* artefact that allow to correlate two CodeLists by linking together their respective codes
  + The *StructureMap* artefact that allow to correlate two DSDs by linking together their respective dimensions and/or attributes

it’s also possible to query and submit (insert, replace, delete) the artefacts and the related items together with their annotations

The SDMX artefacts are stored inside the new version of the SDMX-RI’s Mapping Store Database, published by Eurostat at the end of 2014, on the base of a joint work performed among ISTAT and Eurostat itself for the extension of the old Mapping Store scheme in order to create a more rich repository for SDMX structural metadata, as shown in the figure 1



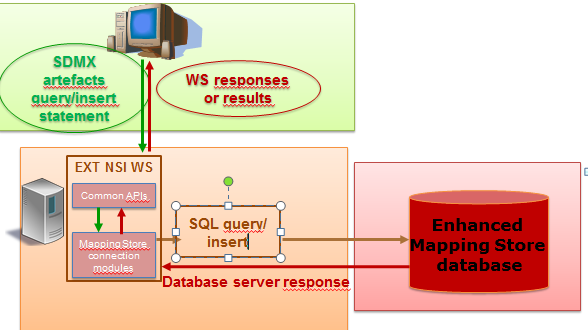
**Fig.1 – List of artefacts added in the new Mapping Store database schema**

The mapping store database is in fact an SDMX repository based on a relational DBMS in which are stored the structural metadata that are represented in the SDMX Information Model through some specific objects (artefacts) very well defined e described by the related XML schemas. Furthermore, in the Mapping Store are also stored the information related to the mapping of local data stored into a dissemination database (in the case of ISTAT, the I.Stat data warehouse) against the SDMX data structures.

The enhanced web service is based on the enhancement of some modules of the SDMX Reference Infrastructure (SDMX-RI): and particularly the Common APIs (SdmxSource .NET) and the libraries for interfacing the mapping store database (as the MappingStore .dll library)

Architecture of the application

The application architecture on which is based the web GUI is shown in the figure 2:

****

**Fig.2 – Application architecture of the enhanced SDMX web service**

The SDMX web service is based on the following layers:

* The *interface layer*, that exposes the methods (available into the WSDL) for the artefacts retrieval and submit (insert, replace. delete). The SubmitStructure method developed by ISTAT is exposed on the “SdmxService” endpoint of the enhanced SDMX web service and allows to submit structure messages for v 2.0 and 2.1 versions of SDMX. ISTAT has also implemented the methods for retrieving artefacts not made available by the SDMX-RI web service, as *Constraints*, *OrganizationSchemes* (*AgencyScheme, DataProviderScheme, DataConumerScheme, OrganizationUnitScheme*), StructureSet (*StructureMap,CodeSetMap*)
* The *information model layer*, based on the use of *the Common APIs* (SDMXSource .NET): inside these classes are managed in central memory the retrieved or submitted artefacts. About the artefact retrieval, the artefacts are passed to the interface layer in order to be returned, as SDMX-ML stream, to the user’s machines that performed the queries. As far as concerns the upload of the artefacts, from the interface layer the SDMX-ML stream containing artefacts is transformed into the related SDMX Objects containing the various artefacts.
* The *interface to the persistent layer* (the *enhanced mapping store database*), is the set of classes (as MappingStore.dll or MappingStoreRetrieval.dll) that allows to insert delete or retrieve SDMX artefacts inside the MappingStore database. These classes are connected to SQL stored procedures in order to perform the tasks above indicated. Particularly as far as concerns the MappingStoreRetrieval.dll, ISTAT has implemented the extension for the retrieval of the artefacts that can’t be queried through the standard version of the SDMX-RI web service: *Constraints*, *OrganizationSchemes* (*AgencyScheme, DataProviderScheme, DataConumerScheme, OrganizationUnitScheme*), *StructureSet* (*StructureMap,CodeSetMap*)

Technical features

The enhancements of the web service have been developed in C# language for 4.0 .NET framework, according to the development platform of the standard version of the SDMX-RI web service

The changes performed for the enhancement of the web service are the following:

* *Implementation of the new web service interfaces*

In table 1 are indicated the changes for the implementation of these new interfaces. These changes are related to the addition of the methods to retrieve the artefacts not previously foreseen by the SDMX-RI, and also to the addition of the SubmitStructure method. The SubmitStructure method takes in input an SDMX structure message that is passed to a method of the new SubmitStructure.dll

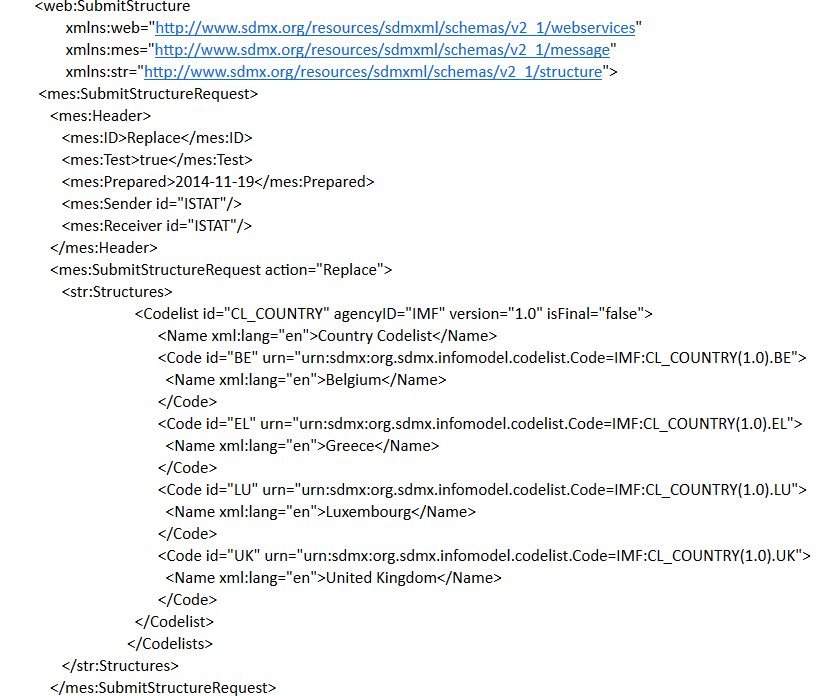
**Table 1 – Changes for the implementation of new web services interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project** | **File path** | **Method/Property** | **Description** |
| Controllers | Controllers/Extension/inputExtension.cs | GetSoapOperation() | Modified method for implementing the retrieval of: -*Constraints*  *OrganisationSchemes*  *StructureSet*  and for implementing the SubmitStructure method |
| NsiWsSoap | NsiWsSoap/NSIStdV21Service.cs |  | Implemented new methods:  *GetConstraint*  *GetOrganisationScheme*  *GetStructureSet*  *SubmitStructure* |
| NsiWsSoap | NsiWsSoap/INSIStdV21Service.cs |  | Implemented new interfaces for the new methods:  *GetConstraint*  *GetOrganisationScheme*  *GetStructureSet*  *SubmitStructure* |
| SdmxAPI | SDMX.API.Constant.QueryMessageType.cs | \_instances | Added new elements in the dictionary for querying Constraints, OrganisationScheme  ,StructureSet |
| SdmxAPI | SDMX.API.Constant.QueryMessageType.cs |  | Added new constants for querying Constraints, OrganisationScheme  ,StructureSet |

* *Submit structures*

The new SubmitStructure.dll developed by ISTAT validates the structure message and transforms it in an object of SDMXObjects type. This object is passed to the MappingStore.dll component that parses the object’s content and, according to the “action” parameter (that can assume the values “Replace” and “Delete”), saves it into the Mapping Store database or delete it.

In the figure 3 is shown an example of SDMX message to be passed to the for inserting an artefact (in this case a codelist) inside the Mapping Store, using the “Replace” action:



**Figure 3–SDMX message example for inserting an artefact through the SubmitStructure method**

In the figure 4 is shown an example of SDMX message for the deletion of an artefact (in this case a Codelist) from the Mapping Store, using the “Delete” action:



**Figure 4–SDMX message example for deleting an artefact through the SubmitStructure method**

The updates made to the SDMX-RI web service in order to implement the *SubmitStructure* web method are indicated in the Table 2:

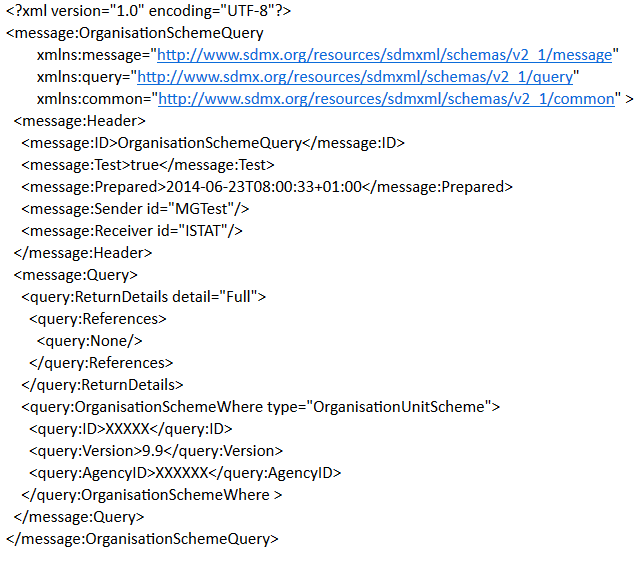
**Table 2 – Software Updates for the implementation of the SubmitStructure web method**

|  |  |  |
| --- | --- | --- |
| **Project** | **File path** | **Description** |
| SubmitStructure |  | Creation of the new project |
| NsiWsSoap | NsiWsSoap/INSIStdV21Service.cs | Implementation of a new “SubmitStructure” interface |
| NsiWsSoap | NsiWsSoap/NSIStdV21Service.cs | Implementation of a new “SubmitStructure” project |

* *Artefact retrieval*

As previously indicated, have been performed the changes to the *MappingStoreRetrieval.dll* in order to allow the retrieval of the artefacts that can’t be queried through the standard version of the *SDMX-RI* web service: *Constraints*, *OrganizationSchemes* (*AgencyScheme, DataProviderScheme, DataConumerScheme, OrganizationUnitScheme*), *StructureSet* (*StructureMap,CodeSetMap*)

An example of an SDMX structure query for retrieving an OrganizationUnitScheme (that is a type of artefact implemented in the enhanced version of the web service) is shown in figure 5:



**Figure 5–SDMX message example for querying an artefact implemented in the enhanced SDMX web service**

The set of changes, categorized for the implemented artefacts, is shown in the table 3

**Table 3 – Set of changes of the MappingStoreRetrieval.dll for the retrieval of the new managed artefacts**

| **Artefact** | **Implemented classes** | **Implemented methods on existing classes** |
| --- | --- | --- |
| *AgencyScheme* | MappingStoreretrieval.Engine.AgencySchemeRetrievalEngine.cs  MappingStoreretrieval.Constants.AgencySchemeConstant.cs | File: MappingStoreretrieval.Manager.AdvancedStructureRetriever.cs  Method:GetMutableAgencyScheme() |
| File: MappingStoreretrieval.Manager.AdvancedStructureRetriever.cs  Method:GetMutableAgencySchemeObjects() |
| File: MappingStoreretrieval.Manager.MappingStoreRetrievalManager.cs  Method:GetMutableAgencyScheme() |
| File: MappingStoreretrieval.Manager.MappingStoreRetrievalManager.cs  Method:GetMutableAgencySchemeObjects() |
| *DataConsumerScheme* | MappingStoreretrieval.Engine.DataConsumerSchemeRetrievalEngine.cs  MappingStoreretrieval.Constants.DataConsumerSchemeConstant.cs | File: MappingStoreretrieval.Manager.AdvancedStructureRetriever.cs  Method:GetMutableDataConsumerScheme() |
| File: MappingStoreretrieval.Manager.AdvancedStructureRetriever.cs  Method:GetMutableDataConsumerSchemeObjects() |
| File: MappingStoreretrieval.Manager.MappingStoreRetrievalManager.cs  Method:GetMutableDataConsumerScheme() |
| File: MappingStoreretrieval.Manager.MappingStoreRetrievalManager.cs  Method:GetMutableDataConsumerSchemeObjects() |
| *DataProviderScheme* | MappingStoreretrieval.Engine.DataProviderSchemeRetrievalEngine.cs  MappingStoreretrieval.Constants.DataProviderSchemeConstant.cs | File: MappingStoreretrieval.Manager.AdvancedStructureRetriever.cs  Method:GetMutableDataProviderScheme() |
| File: MappingStoreretrieval.Manager.AdvancedStructureRetriever.cs  Method:GetMutableDataProviderSchemeObjects() |
| File: MappingStoreretrieval.Manager.MappingStoreRetrievalManager.cs  Method:GetMutableDataProviderScheme() |
| File: MappingStoreretrieval.Manager.MappingStoreRetrievalManager.cs  Method:GetMutableDataProviderSchemeObjects() |
| *OrganisationUnitScheme* | MappingStoreretrieval.Engine.OrganisationUnitSchemeRetrievalEngine.cs  MappingStoreretrieval.Constants.OrganisationUnitSchemeConstant.cs | File: MappingStoreretrieval.Manager.AdvancedStructureRetriever.cs  Method:GetMutableOrganisationUnitScheme()tSchemeObjects() |
| File: MappingStoreretrieval.Manager.AdvancedStructureRetriever.cs  Method:GetMutableOrganisationUnitSchemeObjects() |
| File: MappingStoreretrieval.Manager.MappingStoreRetrievalManager.cs  Method:GetMutableOrganisationUnitScheme() |
| File: MappingStoreretrieval.Manager.MappingStoreRetrievalManager.cs  Method:GetMutableOrganisationUni |
| *StructureSet* | MappingStoreretrieval.Engine.StructureSetRetrievalEngine.cs  MappingStoreretrieval.Constants.StructureSetConstant.cs | File: MappingStoreretrieval.Manager.AdvancedStructureRetriever.cs  Method:GetMutableStructureSet() |
| File: MappingStoreretrieval.Manager.AdvancedStructureRetriever.cs  Method:GetMutableStructureSetObjects() |
| File: MappingStoreretrieval.Manager.MappingStoreRetrievalManager.cs  Method:GetMutableStructureSet() |
| File: MappingStoreretrieval.Manager.MappingStoreRetrievalManager.cs  Method:GetMutableStructureSetObjects() |
| *ContentConstraint* | MappingStoreretrieval.Engine.ContentConstraintRetrievalEngine.cs  MappingStoreretrieval.Constants.ContentConstraintConstant.cs | File: MappingStoreretrieval.Manager.AdvancedStructureRetriever.cs  Method:GetMutableContentConstraint() |
| File: MappingStoreretrieval.Manager.MappingStoreRetrievalManager.cs  Method:GetMutableContentConstraintObjects() |
| File: MappingStoreretrieval.Manager.AdvancedStructureRetriever.cs  Method:GetMutableContentConstraintObjects() |
| File: MappingStoreretrieval.Manager.MappingStoreRetrievalManager.cs  Method:GetMutableContentConstraint() |
|  |
| *Annotations* | MappingStoreretrieval.Engine.AnnotationRetrievalEngine.cs  MappingStoreretrieval.Constants.AnnotationConstant.cs  MappingStoreretrieval.Builder.AnnotationCommandBuilder.cs | MappingStoreretrieval.Engine.ArtefactRetrieverEngine.cs  MappingStoreretrieval.Engine.CategorisationRetrievalEngine.cs MappingStoreretrieval.Engine.ConceptSchemeRetrievalEngine.cs MappingStoreretrieval.Engine.DsdRetrievalEngine.cs MappingStoreretrieval.Engine.HierarchicalCodeListRetrievealEngine.cs MappingStoreretrieval.Engine.DsdRetrievalEngine.cs MappingStoreretrieval.Engine.HierarchicalItemSchemeRetrievalEngine.cs |