**Implementing the SDMX Istat strategy**

*WP3 – Enhancement of the Istat Metadata Management System*

D3.2.2 - Software development of the structural metadata management system

D3.2.3 – Software development of a web GUI to browse and download SDMX structural metadata

Technical and functional documentation

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

***Table of contents***

[1 Introduction 3](#_Toc419733615)

[2 Architecture of the application 6](#_Toc419733616)

[3 Technical features 8](#_Toc419733617)

[4 Main functions 12](#_Toc419733618)

# Introduction

The document describes the technical features and functionalities of the web GUI for creating, importing, managing, browsing and downloading the SDMX structural metadata (artefacts) , that is based on the use and enhancement of some modules of the Eurostat’s SDMX- Reference Infrastructure.

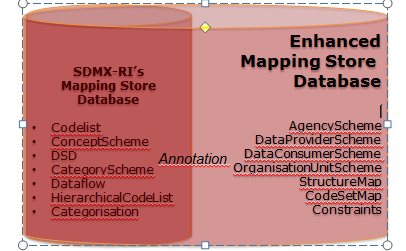
The artefacts that can be managed and browsed by the web GUI are those already managed in the SDMX-RI (CodeList, ConceptScheme, DSD, CategoryScheme, Dataflow, Categorization), and, in addition,

* *AgencyScheme,* ItemScheme artefact representing a set of Agencies.
* *DataProviderScheme,* ItemsScheme 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 selections 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 allows to correlate two CodeLists by linking together their respective codes.
  + The *StructureMap,* artefact that allows to correlate two keyfamilies (DSDs) by linking together their respective dimensions and/or attributes

It’s possible to import artefacts in SDMX format and also to import items of the ItemSchemes (CodeLists, ConceptSchemes, CategorySchemes ,AgencyScheme, DataProviderScheme, DataConsumerScheme) in csv format. It’s possible to export artefacts in SDMX format, and the items of the ItemSchemes in csv format; furthermore is possible to export CodeLists and DSDs (Data StructureDefinitions) in the xml format that can be used in order to create hypercubes and common dimensions inside the .Stat’s datawarehouse..

It’s also possible possible to manage, view, browse and download 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 for the extension of the old database scheme with the artefacts foreseen for the management thorugh the web gui and the enhanced SDMX-RI web service, in order to create a real 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 ISTAT the Single Exit Point’s Mapping Store database has been implemented in the MS SQL Server 2008 RDBMS environment) 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 datawarehouse) against the SDMX data structures.

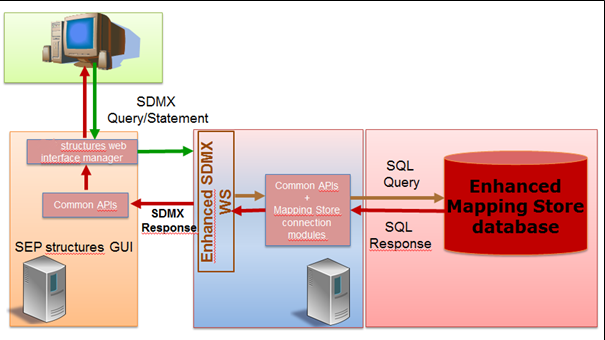
The software is based on the reuse and enhancement of some modules of the SDMX Reference Infrastructure (SDMX-RI): and particularly the Common APIs (SdmxSource .NET)

At present the web gui can work in two modes, according to the value of a specific parameter of the *web.config* file:

* .*Reading mode,* through which is possible to browse, search and download artefacts
* *Management mode,* through which, in addition to the reading functionalities, is possible to import, insert, update, delete artefacts, according with their state (i.e. final/not final for update; referenced/not referenced by other artefacts for deletion)

# 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 web GUI**

In this architecture, the web GUI is based on the following layers:

* The *interface manager layer*, on which are based the interfaces for selecting artefacts, browsing the general information and the detail of a single artefact, downloading and uploading the artefacts
* The *SDMX query/statement generation layer*, through which the selections made by the users on the web GUI are translated in SDMX registry queries for retrieving artefacts on the base of the options selected (lists of artefacts; artefacts comprehensive of their details i.e. items; artefacts together with their referenced artefacts;…), or in SDMX statements for inserting or deleting artefacts.
* The *SDMX web service* to which the query/statements are submitted. This web service is an extended version of the SDMX web service containing, in addition to the methods for querying data and structural metadata (SDMX Artefacts), also a method for submitting artefacts. Is possible to upload and download through the web service artefacts in SDMX v. 2.0 and 2.1.The queries/statements are then submitted to the web service and then processed according to the SDMX rules (i.e. is not possible to change a finalised artefact; is not possible to delete artefacts referenced by other artefacts, etc…)
* *Layer for artefacts handling*, that is based on the SDMX Common APIs (SdmxSource .NET) through which the artefacts returned by the SDMX web service are uploaded to SDMX objects in order to be presented or handled through the web GUI

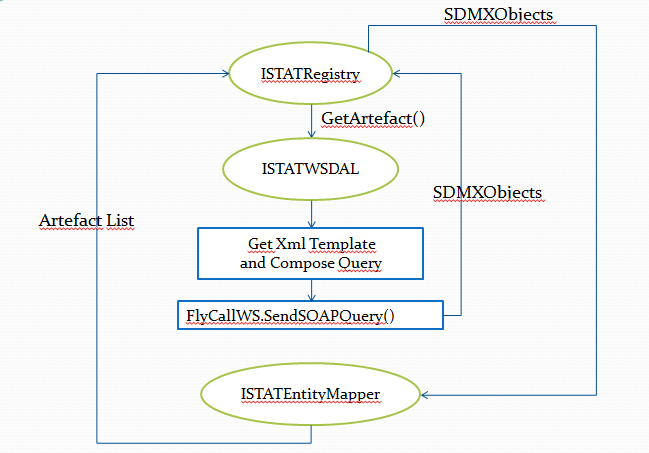
# Technical features

The application has been developed in C# language using the ASP. NET technology for 4.0 .NET framework. Some parts of the user interface have been developed by using also jquery technology (for instance the feature that, when is opened a pop-up window, the underlying page is blocked until the closure of the pop-up itself). The lists of artefacts and the lists of items of ItemSchemes are presented inside parametric grids defined in a general way and populated on the base of the user’s selections. Each selection produces a post-back command through which are generated the queries to be submitted to the SDMX web service and is managed the result through the SDMXObjects of the SdmxSource .NET; the content of the SDMXObjects is then passed to the interface objects (grid bindings text-box, combo-box, etc..)

As far as concerns the back-end modules, the two main functional flows for browsing and managing artefacts, are the following:

* *Artefact retrieval*

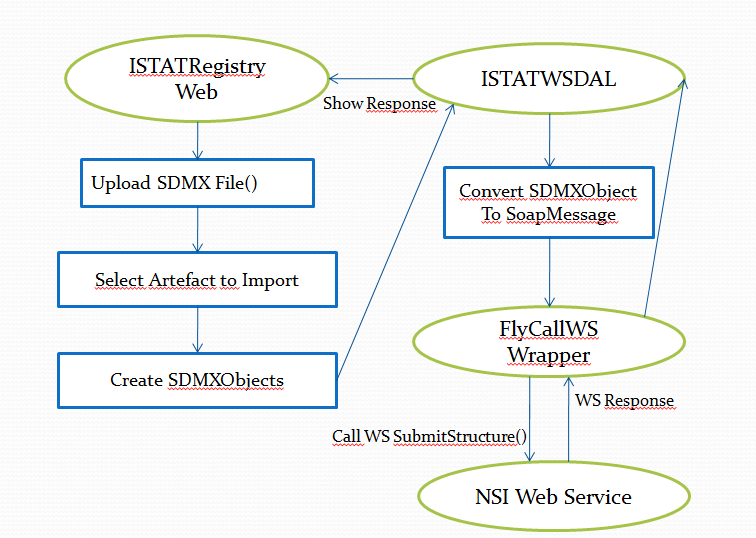
In order to retrieve the artefacts of interest, once selected from the GUI the type of artefact or a single artefact, the related parameters are passed to the ISTATWSDAL module, that, on the base of the SDMX-ML query templates, creates the SDMX-ML query and submit it to the web service through the FlyCallWS.SendSOAPQuery() function. The web service response is passed to the SDMXObiects module and, from it, uploaded to the IstatEntityMapper module, that maps the artefacts information contained in the SDMXObjects against the objects used for populating the web gui (as the grids containing the lists of artefacts or items).

****

**Fig.3 – Artefact retrieval modules workflow**

* *Submit Artefact*

After uploaded the SDMX file containing the artefacts to be imported, the web GUI populates an SDMXObiects in order to allow end users to select the artefacts to be imported, and then the selected artefacts through the SDMXObject are passed to the ISTATWSDAL (the class layer that “wraps” the interaction to the web service) that converts it in a XML stream and then submits this stream to the “SubmitStructure” method of the web service.

****

**Fig.4 – Submit artefact modules workflow**

In applicative terms, the ASP.NET solution is composed by the following 5 projects:

* *ISTATEntity:* this project manages the entities used by the Web Gui (implemented with the only properties needed for visualization and storicization); it also contains the classes used for populating the grids (DataBind)
* *ISTATEntityMapper:* this project, that consists in a only one class named “EntityMapper”, that maps the data retrieved by the SdmxObjects against the ISTATEntity objects.
* *ISTATRegistry:* the main ASP .NET project
* *ISTATUtils:* Project that contains Utility class for the localization of the application (LocalizedUtils)
* *ISTATWSDAL:* Project implementing the wrapper for accessing to the web service. It exposes the interfaces for the artefacts retrieval.

Furthermore, have been developed also the following user controls, reused in more than one page of the web GUI:

|  |  |
| --- | --- |
| **Control** | **Description** |
| *AddText* | Manages the “Name” and “Description” items for all the aretefacts and is used by all the pages showing the details of the artefacts. |
| *ArtefactDelete* | Control that manages the artefact’s deletion |
| *ControlAnnotation* | Control that manages the insert/update of the annotations |
| *DuplicateArtefact* | Control that manages the duplication of an artefact |
| *FileDownload3* | Control that manages the download of an artefact |
| *GetCodeList* | Control allowing to retrieve the Codelist Id |
| *GetConcept* | Control allowing to retrieve the Ids for ConcencptSchemes and Concepts |
| *GetDataflow* | Control allowing to retrieve the Dataflow Id |
| *GetDSD* | Control allowing to retrieve the DSD Id |
| *SearchBar* | Control used to filter the list of artefacts |
| *UserPopUp* | Control for the visualization of the application messages |

Main functions

The main functions of the web GUI are the following:

* Import SDMX structure files containing artefacts
* Management (insert, update, delete) of the following artefacts and of the related annotations at all levels (general artefact’s information, artefact’s items, components, other types of artefact’s detail):
  + *Codelist*
  + *Concept Scheme*
  + *DSD*
  + *Category Scheme*
  + *Dataflow*
  + *Categorization*
  + *Agency Scheme*
  + *Data Provider Scheme*
  + *Data Consumer Scheme*
  + *Organization Unit Scheme*
  + *Content Constraint*
  + *StructureSet (StructureMap, CodeSetMap)*
* Export of the artefacts in SDMX v. 2.0 and 2.1 formats
* Csv export of the list of items of the ItemSchemes, and also XML export in .Stat compliant formats for DSD and Codelist

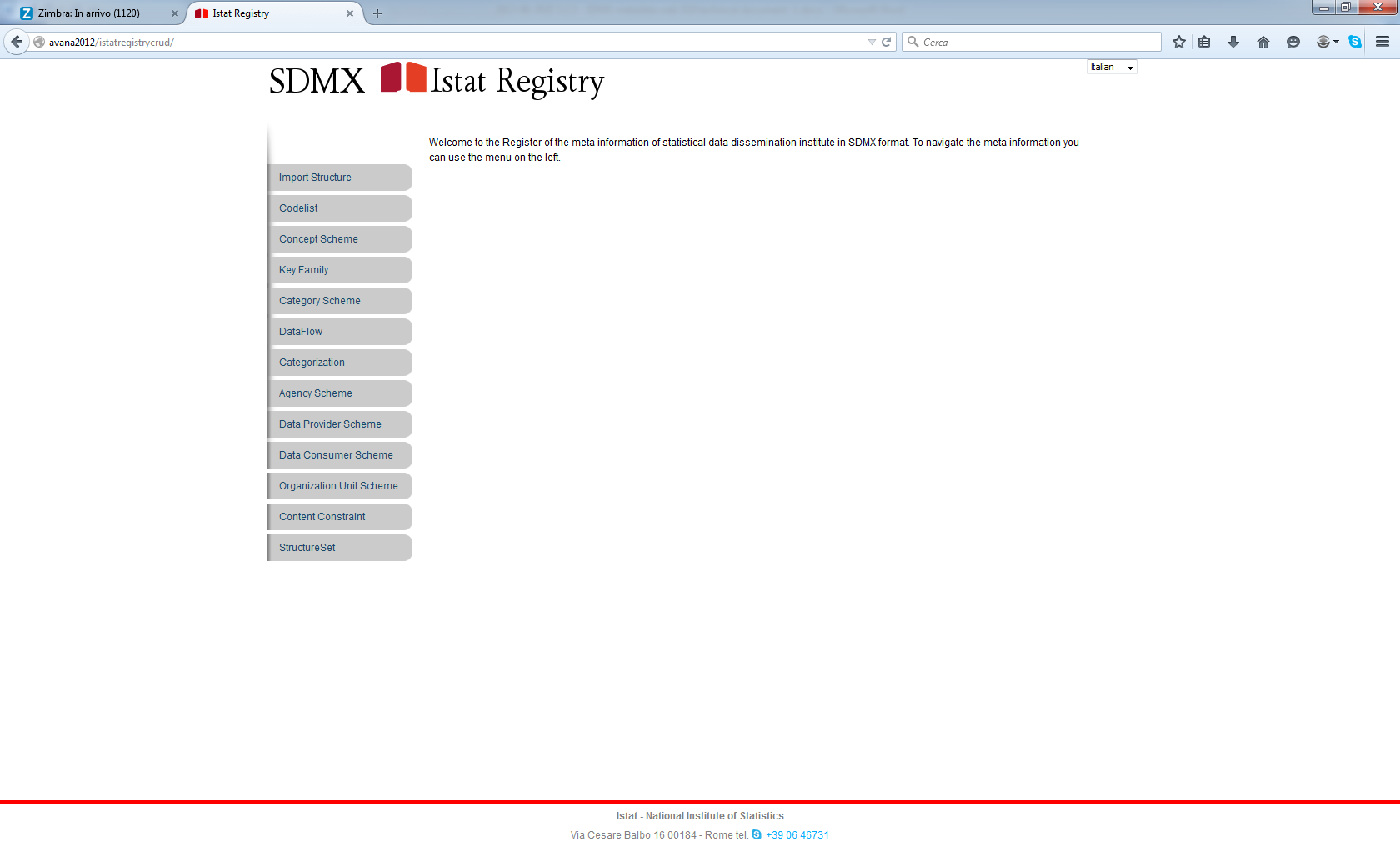
The application is structured in a generalized way, and the layout is customizable through specific style-sheets. The localization of the interface is configurable through resource files specific for each language.

Furthermore, is configurable through a parameter in the *web.config* file a visualization mode of the application in which:

* users can only browse and download the artefacts (*reading mode*)
* the artefacts can be changed and modified (*management mode*).

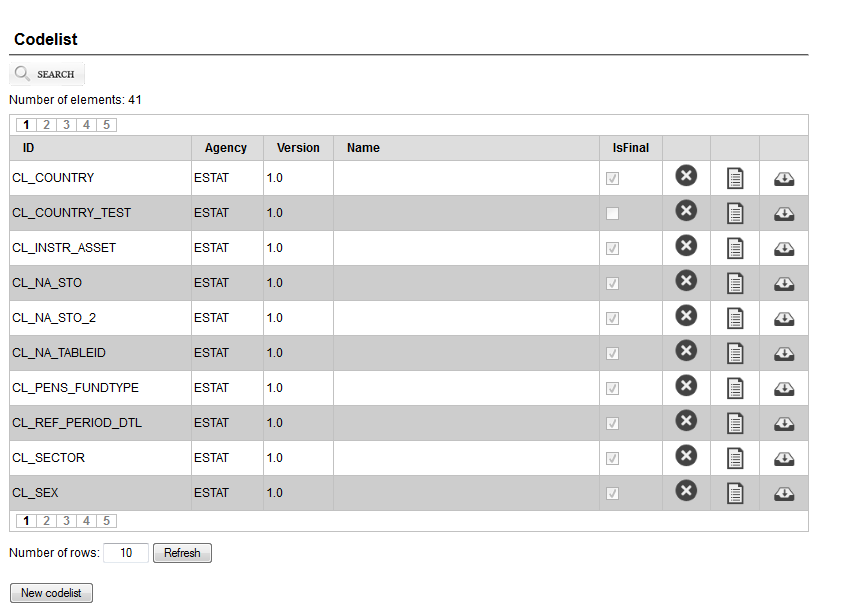
The main menu, containing the list of the artefacts that can be managed, is shown in figure 5:

In order to connect the web gui to the enhanced SDMX-RI web service, in the *web.config* file has to be set the *“WSEndPoint”* parameter with the URL of the web service’s endpoint.



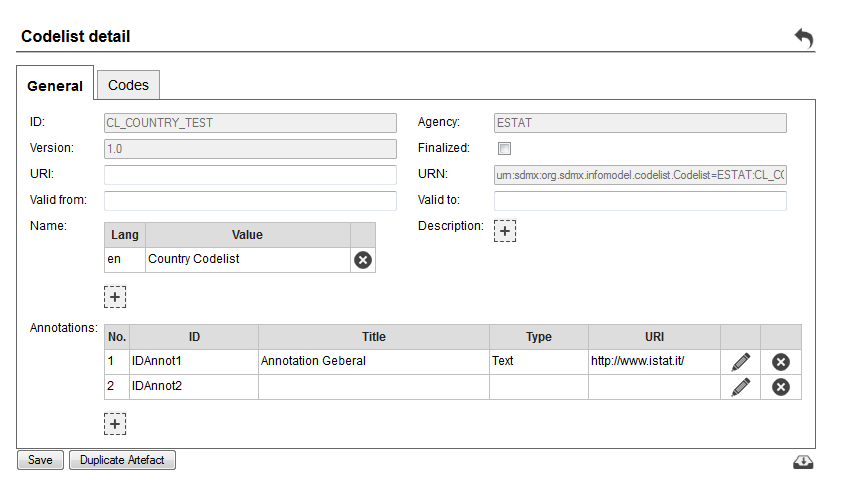
**Figure 5 – Main menu of the web GUI**

In figure 6 is shown the list of artefacts related to a specific type (in this case the codelists): this view is the same for all types of artefact and from this interface is possible to delete, edit and download the artefacts.



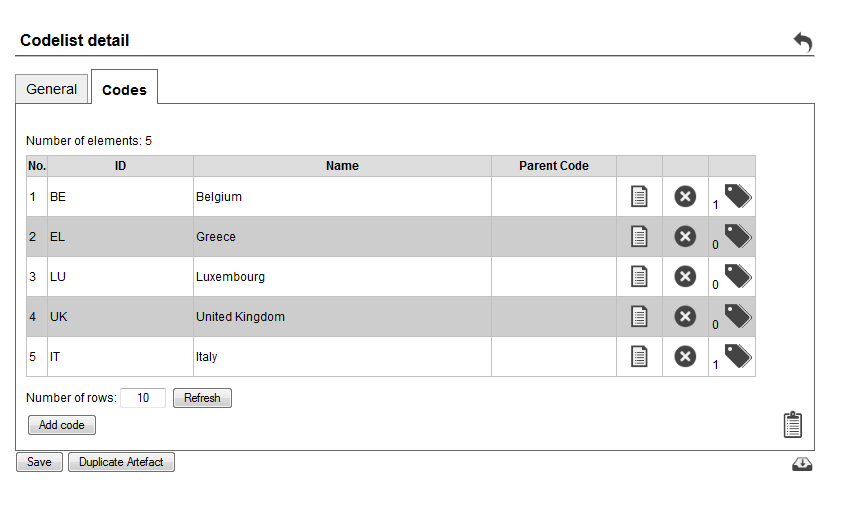
**Figure 6 – List of artefacts**

In figure 7 is shown the interface containing the general information of the artefacts (Id, Agency, Version, Name,….) and also this interface is the same for all the artefacts



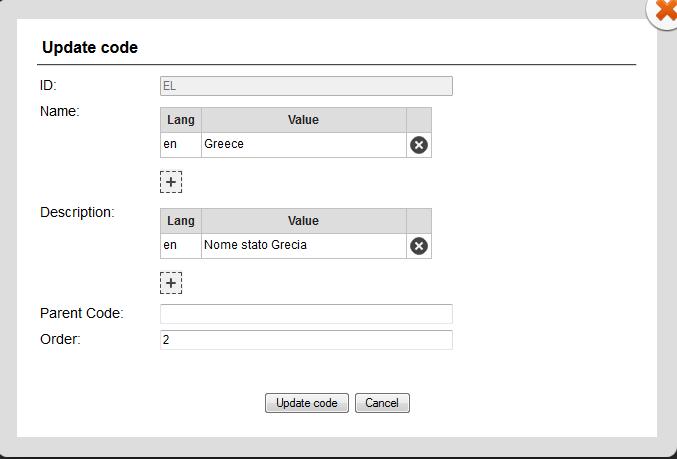
**Figure 7 – Artefact’s general informations**

For each ItemScheme artefact, in the detail tab is visible, into a specific grid, the list of the items with their name in the selected language, and the buttons for deleting, editing, adding items and for each item the button for managing the related annotations



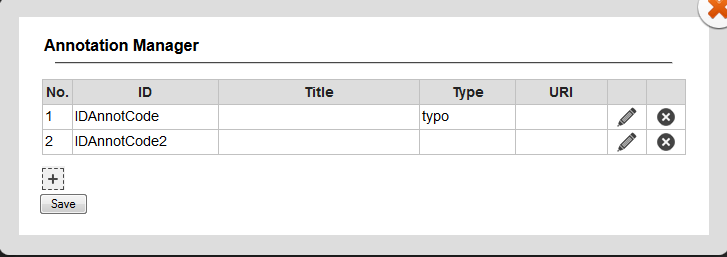
**Figure 8 – ItemScheme artefact detail**

In figure 9 there’s the interface for the management of the single item (insert, update), containing the information related to ID, name (multilingual)



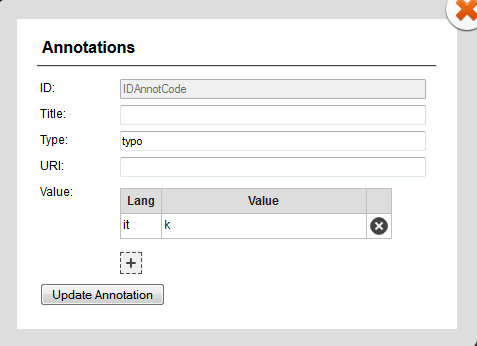
**Figure 9 – Item detail**

As far as concerns the management of the annotations for the artefacts and also for their elements as items, components and other types of detail, in figure 10a is represented the menu showing the list of annotations for each element, and the buttons for inserting/editing/deleting annotations



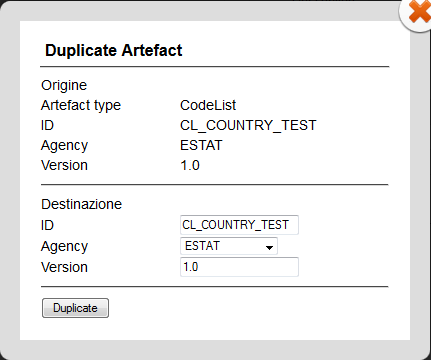
**Figure 10a – Annotations manager menu**

In figure 10b there’s the interface for inserting or updating the annotations together with all their details:



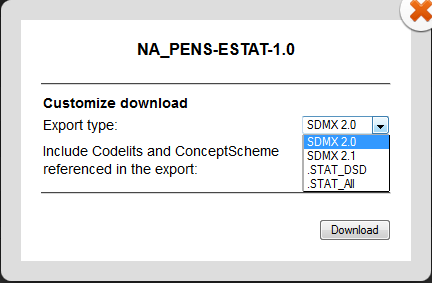
**Figure 10b – Detail of Annotations**

In figure 11 is shown the interface for the artefacts duplication, that is implemented for each artefact: in this interface the users can change the Id, Agency and Version of the artefact that will be saved as “not final” artefact.

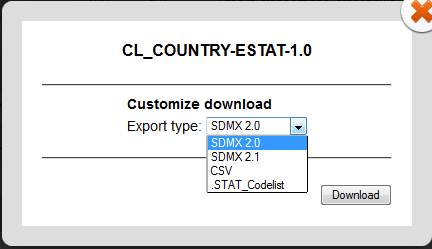


**Figure 11 – Artefacts duplication interface**

In figure 12a and in figure 12b there’s the export interface respectively for DSD and Codelists (as an example for ItemSchemes): the export formats are SDMX-ML v. 2.0, v. 2.1, csv (for ItemSchemes only), and XML for .Stat (only for codelists and DSDs).



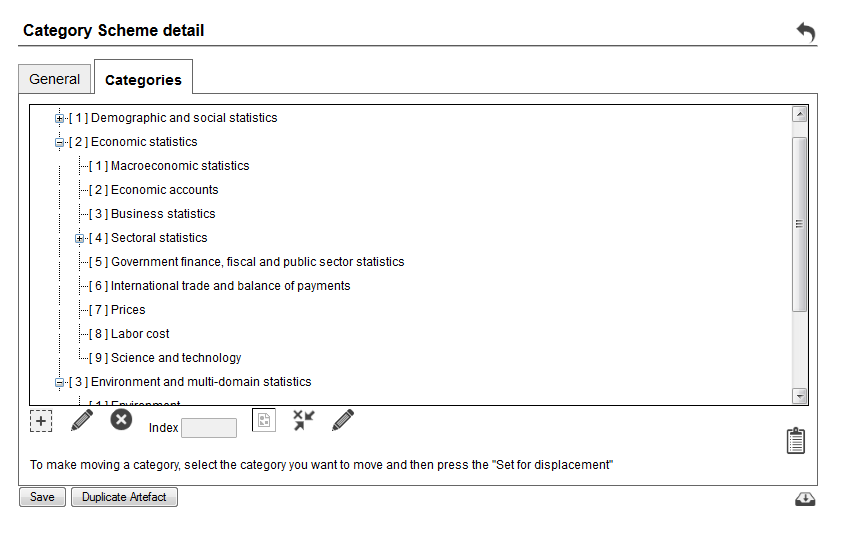
**Figure 12a – DSD export interface**



**Figure 12b – Codelist export interface**

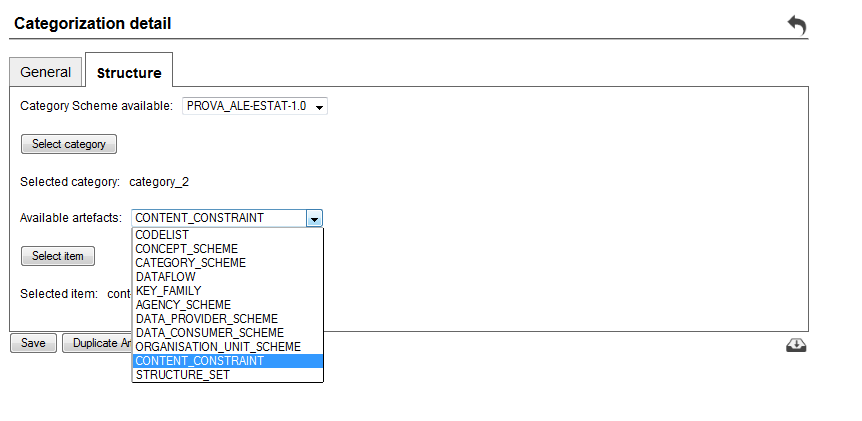
As far as concerns the SDMX artefacts different from the ItemSchemes (Codelist, ConceptScheme,AgencyScheme, DataProviderScheme,DataConsumerScheme), the detail interface changes according to the type of artefact.

In figure 13 is shown the detail interface for the CategoryScheme, that effectively is an ItemScheme but in which the set of items (categories) is shown through a hierarchical tree representation, and are shown the buttons to to create, update a category and also change the position and the hierarchical level of the category iteself.



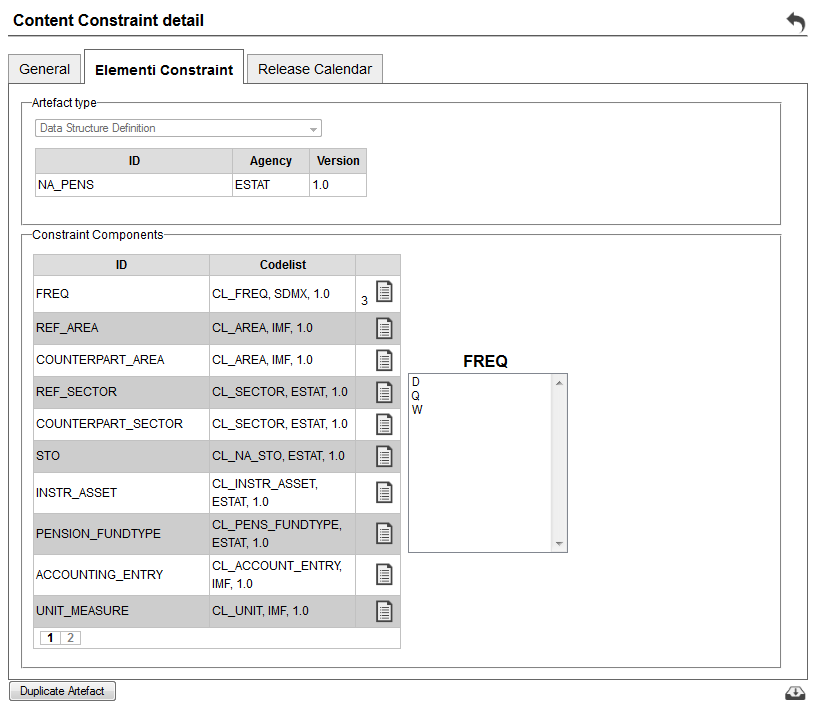
**Figure 13 – CategoryScheme detail**

As far as concerns Categorisations, in the detail the interface represents the relationship among a selected category and a specified artefact



**Figure 13 – Categorisation detail**

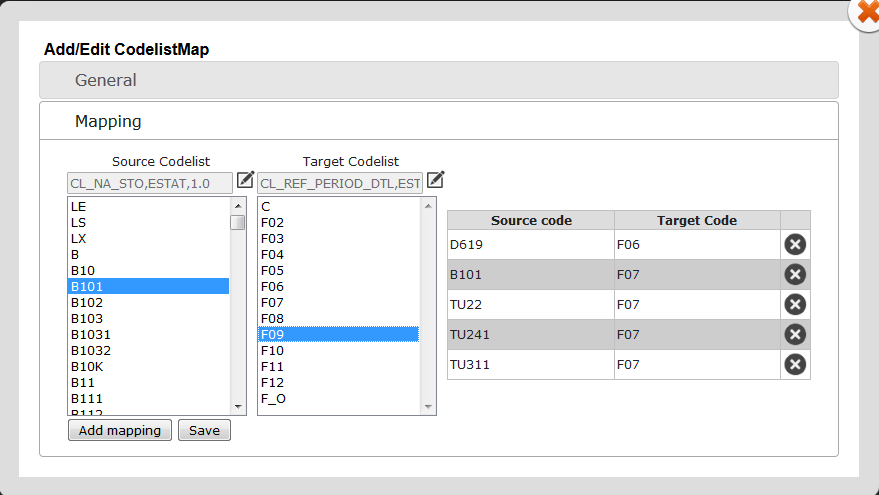
In the detail of a constraint there’s the indication of the selected codes of the codelists linked to the DSD’s components as shown in the following figure:



**Figure 14 – Constraint’s detail**

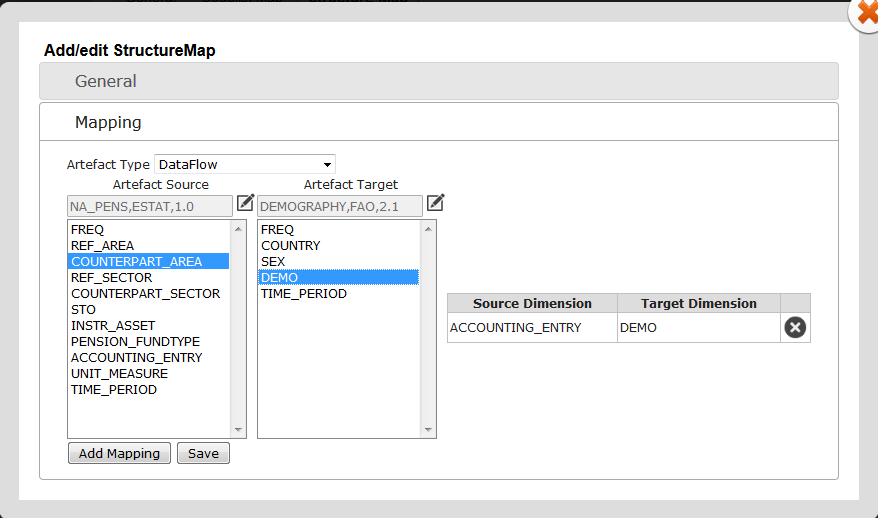
Furthermore, there’s the detail of the StructureSet artefact in which there’s the list of CodeListMaps (mappings among the codes of the codelists) and of the StructureMaps (mappings among components of the DSDs)

For each CodeListMap the detail interface is shown in figure 15, in which is possible to select the source and target Codelist, and, once selected them, are shown the related items in order to define the relationship among them.



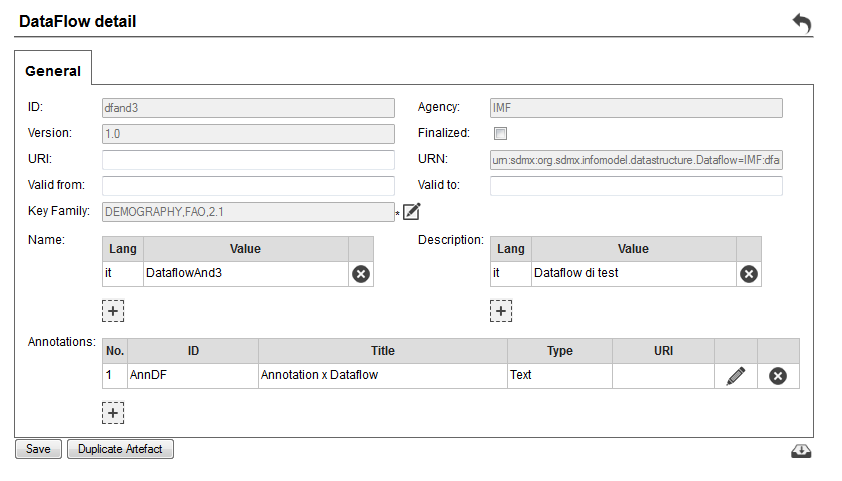
**Figure 15 – CodelistMap detail**

In figure 16 is shown the detail interface for the StructureMap, in which is possible to select the source and target DSD and, once selected them, are shown the related components in order to define the relationship among them.



**Figure 16 – StructureMap detail**

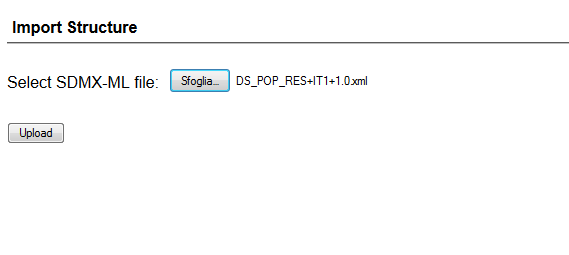
In figure 17 is shown the detail interface for the Dataflow, in which is possible to view and select the DSD describing the structure of the dataflow itself in terms of components (Dimensions, Attributes and Measures), and of the related Codelists



**Figure 17 – Dataflow detail**

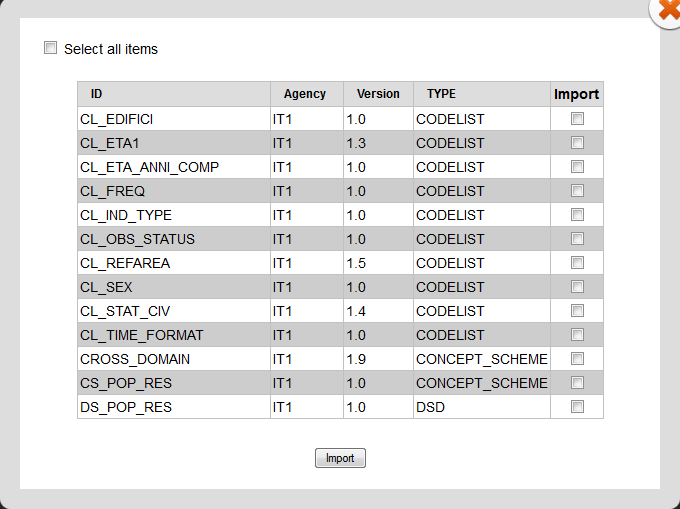
As far as concerns the import of SDMX artefacts, from the main menu is possible to select the “ImportStructure” function.

In the “ImportStructure” page is possible to select the SDMX structure file and to upload it by clicking on the related button.



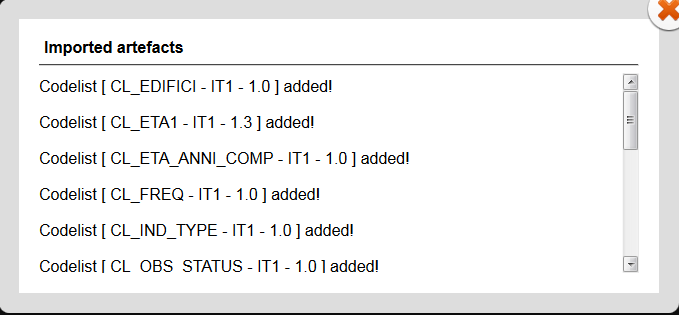
**Figure 18 – SDMX structure file upload page**

Once uploaded the structure file, the system shows the list of artefacts contained into the file, with the possibility to select the artefacts to be imported



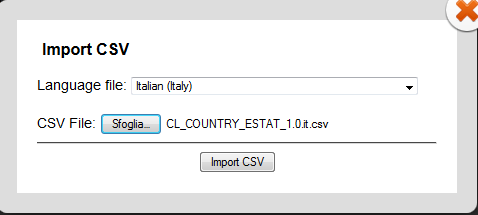
**Figure 19 – SDMX artefacts import page**

Clicking on the “import” button, the artefacts are stored into the Mapping Store database, and the system returns a report about the SDMX import of the artefacts.



**Figure 20 – SDMX artefacts import report**

The csv import of the items of the ItemSchemes can be performed by clicking on the “import” button icon in the detail of the artefact. The system returns a popup in which the users have to indicate the csv file containing the items to be imported, and also the import language



**Figure 21 – csv import popup**

Once completed the import, the system returns a report screen on which is indicated if this task has been performed successfully or not and, in this second case, the rows affected by errors.