
D 2.2.3A: European Open Source Metadata Editor Developers' Guide v.1.0

Title	D 2.2.3A: European Open Source Metadata Editor Developers' Guide v1.0
Creator	Marzia Grasso and Max Craglia
Creation date	03/12/2010
Date of last revision	20/12/2010
Subject	European Open Source Metadata Editor (EUOSME) Developers' guide
Status	<input type="checkbox"/> Draft <input checked="" type="checkbox"/> Final
Publisher	EuroGEOSS
Type	Text
Description	Developer guide of the European Open Source Metadata Editor
Contributor	Angelo Quaglia
Format	pdf
Source	
Rights	<input type="checkbox"/> Restricted <input checked="" type="checkbox"/> Public
Identifier	D2_2_3A_EUOSME_Developers_Guide.doc
Language	En
Relation	EuroGEOSS European Open Source Metadata Editor D 2.2.3
Coverage	If applicable

TABLE OF CONTENTS

1	Introduction	8
2	Quick start	8
2.1	Prerequisites	8
2.1.1	System requirements	9
2.1.2	Java SDK	9
2.1.3	Eclipse and Google Plugin for Eclipse	9
2.1.4	Google Web Toolkit SDK	9
2.1.5	EUOSME	9
2.2	Run the application	9
2.2.1	Run locally in development mode	10
2.2.2	Compile and run in production mode	10
3	Project Organization	10
3.1	Configuration settings	10
3.2	Packages	11
3.2.1	Client package	12
3.2.2	Callback package	13
3.2.3	CodeLists package	14
3.2.4	I18n package (Internationalization)	16
3.2.5	Images package	17
3.2.6	ISO19115 package	17
3.2.7	ISO19115 UI package	18
3.2.8	UserGuides package	20
3.2.9	Widgets package	20
3.2.10	XML sources package	21
3.2.11	Server package	21
3.3	Referenced libraries	22
3.4	Inherited modules	22
3.5	War	23
4	ISO19115 represented classes	24
4.1	Path identification	24
4.2	Table structure	26

4.3	CI_Address	26
4.4	CI_Citation	27
4.5	CI_Contact	27
4.6	CI_Date	28
4.7	CI_OnlineResource	29
4.8	CI_ResponsibleParty	29
4.9	CI_Telephone	30
4.10	Distance	30
4.11	DQ_ConformanceResult	30
4.12	DQ_Element	31
4.13	EX_GeographicBoundingBox	31
4.14	MD_Constraints	31
4.15	MD_Identifier	32
4.16	MD_Keywords	32
4.16.1	MD_Keywords_DataThemes	32
4.16.2	MD_Keywords_Gemet	32
4.16.3	MD_Keywords_INSPIRE	33
4.17	MD_LegalConstraints	33
4.18	MD_Resolution	33
4.19	TM_Primitive	34
5	EUOSME classes	34
5.1	Main Panel	34
5.2	Tabs	35
5.3	Tab Classification	35
5.4	Tab Conformity	35
5.5	Tab Constraints	36
5.6	Tab Geographic	36
5.7	Tab Identification	36
5.8	Tab Keyword	37
5.9	Tab Metadata	37
5.10	Tab Organization	38
5.11	Tab Quality	38
5.12	Tab Temporal	38
6	Widgets	39

6.1	CI	39
6.2	CIMultiple	40
6.3	Boolean	41
6.4	CharacterString	42
6.5	CharacterStringLong	43
6.6	CharacterStringMultiple	43
6.7	CodeList	44
6.8	CodeListFree	45
6.9	CodeListMultiple	46
6.10	DatImpl	46
6.11	GeoBoundsMultiple	47
7	Services	48
8	Customization	49
8.1	Write your own widget to represent a metadata element (extend to full ISO)	49
8.1.1	Example of adding new elements (Extension of CI_Citation)	49
8.2	Internationalization and locale	52
8.3	Maps	54

TABLES

Table 1:	Configuration parameters	11
Table 2:	Main packages	11
Table 3:	Client package files	12
Table 4:	Callback package files	13
Table 5:	Example of definition of a ClientBundle	14
Table 6:	Example of results calling code list service	15
Table 7:	CodeLists package files	15
Table 8:	l18n package files	16
Table 9:	Images package files	17
Table 10:	ISO19115 package files	17
Table 11:	ISO19115 UI package files	19
Table 12:	UserGuides package files	20
Table 13:	Widgets package files	20
Table 14:	XML sources package files	21

Table 15: Server package files.....	21
Table 16: Inherited modules.....	22
Table 17: EUOSMEGWT/war/ files.....	23
Table 18: CI_Address elements.....	27
Table 19: CI_Citation elements.....	27
Table 20: CI_Contact elements.....	28
Table 21: CI_Date elements.....	28
Table 22: CI_OnlineResource elements.....	29
Table 23: CI_ResponsibleParty elements.....	29
Table 24: CI_Telephone elements.....	30
Table 25: Distance elements.....	30
Table 26: DQ_ConformanceResult elements.....	30
Table 27: DQ_Element elements.....	31
Table 28: EX_GeographicBoundingBox elements.....	31
Table 29: MD_Constraints elements.....	32
Table 30: MD_Identifier elements.....	32
Table 31: MD_Keywords elements.....	32
Table 32: MD_LegalConstraints elements.....	33
Table 33: MD_Resolution elements.....	34
Table 34: TM_Primitive elements.....	34
Table 35: Tab Classification elements.....	35
Table 36: Tab Conformity elements.....	36
Table 37: Tab Constraints elements.....	36
Table 38: Tab Geographic elements.....	36
Table 39: Tab Identification elements.....	37
Table 40: Tab Keywords elements.....	37
Table 41: Tab Metadata elements.....	37
Table 42: Tab Organization elements.....	38
Table 43: Tab Quality elements.....	38
Table 44: Tab Temporal elements.....	39
Table 45: List of RPC methods.....	49

FIGURES

Figure 1: MD_Metadata class in UML	25
Figure 2: Example of CI.....	40
Figure 3: Example of CIMultiple	41
Figure 4: Example of Boolean	42
Figure 5: Example of CharacterString	43
Figure 6: Example of CharacterStringLong	43
Figure 7: Example of CharacterStringMultiple.....	44
Figure 8: Example of CodeList.....	44
Figure 9: Example of CodeListFree.....	46
Figure 10: Example of CodeListMultiple.....	46
Figure 11: Example of DateImpl.....	47
Figure 12: Example of GeoBoundsMultiple.....	48

ACRONYMS AND ABBREVIATIONS

Abbreviation	Name
EUOSME	European Open Source Metadata Editor
GWT	Google Web Toolkit
XML	Extensible Mark-up Language
ISO	International Organization for Standardization
FF	Mozilla Firefox
IE	Internet Explorer
UI	User Interface
RPC	A remote procedure call
UTF-8	Unicode Transformation Format, 8 bit
GEMET	General Environmental Multilingual Thesaurus
XPath	XML Path Language
UML	Unified Modeling Language
JSON	JavaScript Object Notation

1 INTRODUCTION

The European Open Source Metadata Editor (EUOSME) is a web application written in Java and based on Google Web Toolkit (GWT) libraries¹. EUOSME has been developed as part of the EuroGEOSS project (www.eurogeoss.eu) to help create metadata compliant with the INSPIRE Directive (2007/2/EC)¹ and the INSPIRE Metadata Regulation (1205/2008)². More specifically, this implementation allows to describe a spatial data set, a spatial data set series or a spatial data service compliant with the standards ISO 19115:2003 (corrigendum 2003/Cor.1:2006)ⁱⁱ and ISO 19119:2005ⁱⁱⁱ. It is therefore an implementation of the INSPIRE Metadata Technical Guidelines based on these two ISO standards, and published on the INSPIRE web site³. This editor builds on the experience acquired in the development of the INSPIRE Metadata Implementing Rules, and includes the INSPIRE Metadata Validator Service available from the INSPIRE EU Geo-portal (<http://www.inspire-geoportal.eu/>).

This editor is an Open Source evolution of the one currently available through the eu-geoportal, with extended features on semantic queries. The editor is available through the Open Source Observatory & Repository Europe (www.osor.eu). A User Guide is also available from the EuroGEOSS Web site (http://www.eurogeoss.eu/Documents/EuroGEOSS_D_2_2_3.pdf) and subsequent releases. The purpose of this Developers' Guide is to allow developers to extend and customise the metadata editor, and by updating the code and related documentation on OSOR, extend the capabilities of the user community.

2 QUICK START

2.1 Prerequisites

In order to use the application you have to do the following tasks:

- Install Java SDK
- Install Eclipse or your favorite Java IDE
- Install Google Plugin for Eclipse
- Download Google Web Toolkit and unzip in your favorite directory
- Import EUOSME project in your Java IDE workspace

¹ <http://eur-lex.europa.eu/JOHtml.do?uri=OJ:L:2007:108:SOM:EN:HTML>

² <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32008R1205:EN:NOT>

³ http://inspire.jrc.ec.europa.eu/documents/Metadata/INSPIRE_MD_IR_and_ISO_v1_2_2_20100616.pdf

2.1.1 System requirements

EUOSME is designed to run on systems that meet the following requirements:

- Operating system: Windows Vista/XP/2000, Mac OS X 10.4+ (Tiger or Leopard), or Linux with GTK+ 2.2.1+
- Hardware: ~100MB of free disk space, 512MB RAM

2.1.2 Java SDK

You will need the Java SDK version 1.5 or later. Download it from Sun Oracle site <http://www.oracle.com/technetwork/java/index.html>.

2.1.3 Eclipse and Google Plugin for Eclipse

<http://www.eclipse.org/downloads/> (Eclipse)

<http://code.google.com/intl/it/eclipse/docs/download.html> (Google Plugin for Eclipse)

The Google Plugin for Eclipse adds functionality to Eclipse for creating and developing GWT applications.

2.1.4 Google Web Toolkit SDK

The Google Web Toolkit SDK contains the core libraries and compiler. Download the last version from <http://code.google.com/intl/it/webtoolkit/download.html>. We are using the GWT SDK 2.1.

2.1.5 EUOSME

Download the latest version of EUOSME from www.osor.eu

2.2 Run the application

Once you have installed GWT and Eclipse you can import EUOSME into your workspace:

- Open Eclipse
- Click on File > Import
- Select General > Existing projects into workspace
- Browse the local system and find the folder where you have extracted the project files.

2.2.1 *Run locally in development mode*

In Eclipse right-click on your web application project and select `Run As > Web Application` from the popup menu.

This creates a Web Application launch configuration for you and launches it. The web application launch configuration will start a local web server and GWT development mode server.

Once the browser plugin is installed, navigate to the URL again and the starter application will load in development mode.

2.2.2 *Compile and run in production mode*

To run the application in production mode, firstly you have to compile the application by right-clicking the project and choosing `Google > GWT Compile`.

This command invokes the GWT compiler which generates a number of JavaScript and HTML files in the folder `war`.

To see the final application, open the file `EUOSMEGWT.html` in your web browser. Instead, if you want to deploy your application to production, you could do that by serving the HTML and JavaScript files in the `war` directory from your web servers.

3 PROJECT ORGANIZATION

Under the main project directory there are the following directories:

- `src`: contains production Java source
- `doc`: contains the documents in textual and html format related to Metadata Regulation (it is used to help the user through the editing of a metadata)
- `lib`: contains the java libraries referenced in the application
- `war`: is the web application and it contains static resources (host page, style sheet and images)

3.1 Configuration settings

To configure EUOSME web application, you can modify the settings in the `web.xml` file at `/war/WEB-INF/web.xml`.

You could change the following parameters, related to the servlet named `InitServiceProxyImpl`:

Table 1: Configuration parameters

Name	Purpose	Possible values	Default value
metadataType	Default metadata type loaded (see DataTypes enum class in Client package)	DATASET, DATASET_SERIES, DATA_SERVICE	DATASET
appMode	Set the profile used, it enable or disable elements to be shown (see AppModes enum class in Client package)	DEFAULT, RDSI, GEOPORTAL	GEOPORTAL
rpcCodeList	If true, the application uses the online code lists; if false, the application uses the local resources (see CodeLists package)	true/false	false
apiMapstraction	Indicates the type of maps to show (see Maps)	google/openlayers	openlayers
showAll	The elements are organized using the support of disclosure panels. If true, all the disclosure panels are opened; if false, the disclosure panels are opened only if the content is required	true/false	false

3.2 Packages

These are the main modules and packages:

- Java source files are in the directory: EUOSMEGWT\src\eu\europa\ec\jrc\euosme\gwt
- The module is defined in the XML file: EUOSMEGWT\src\eu\europa\ec\jrc\euosme\gwt\EUOSMEGWT.gwt.xml
- The project root package is: eu.europa.ec.jrc.euosme.gwt
- The client-side code packages are in the folder: EUOSMEGWT \src\eu\europa\ec\jrc\euosme\gwt\client
- The server-side code package is in the folder: EUOSMEGWT \src\eu\europa\ec\jrc\euosme\gwt\server
- The logical module name is: eu.europa.ec.jrc.euosme.gwt.client.EUOSMEGWT

Table 2: Main packages

Name	Package	Purpose
Client	eu.europa.ec.jrc.euosme.gwt.client	Fundamental classes used in client-side code with the EntryPoint class, the utilities class and resources (see Client package)
Callback	eu.europa.ec.jrc.euosme.gwt.client.callback	It contains the asynchronous callback objects to be notified when an RPC has completed (see Callback package)
CodeLists	eu.europa.ec.jrc.euosme.gwt.client.codeLists	ClientBundle resources with the offline version of code lists (see CodeLists package)

i18n	eu.europa.ec.jrc.euosme.gwt.client.i18n	Internationalization support using static string localization method relied on standard Java properties files (see I18n package (Internationalization))
Images	eu.europa.ec.jrc.euosme.gwt.client.images	Image resources (see Images package)
ISO19115	eu.europa.ec.jrc.euosme.gwt.client.iso19115	Modules that reproduce some elements of ISO 19115 (see ISO19115 package)
ISO19115 UI	eu.europa.ec.jrc.euosme.gwt.client.iso19115.ui	User interface classes: main panel and tabs (see ISO19115 UI package)
UserGuides	eu.europa.ec.jrc.euosme.gwt.client.userGuides	User guides (see UserGuides package)
Widgets	eu.europa.ec.jrc.euosme.gwt.client.widgets	Base modules: each module represents a widget to be used to create ISO complex elements (see Widgets package)
XML Sources	eu.europa.ec.jrc.euosme.gwt.client.XMLsources	XML resources (see XML sources package)
Server	eu.europa.ec.jrc.euosme.gwt.server	Contains the server-side code for RPC (see Server package)

3.2.1 Client package

Source folder: eu.europa.ec.jrc.euosme.gwt.client

This package includes classes used for generic purposes.

Table 3: Client package files

File	Purpose
AppModes.java	Set of constants to define a particular profile: <code>GEOPORTAL</code> follows INSPIRE regulation, <code>NORMAL</code> supports ISO generic structure, etc.
CheckFunctions.java	Fixed set of constants that defines the type of functions used to check the input fields (electronic mail address, URL, text string, integer number, double number)
DataTypes.java	Set of constants to define different metadata types (dataset, dataset series, data service)
EUOSMEGWT.java	Entry point class
InitServiceProxy.java	Remote service interface used to read the configuration settings
InitServiceProxyAsync.java	Remote service asynchronous interface for the above mentioned service
MyResources.java	Client Bundle resources (images, code lists and xml sources)

MyDateBox.java	It extends GWT <code>DateBox</code> ⁴ solving the known issue 4532 ⁵ on IE.
MyDockLayoutPanel.java	It extends GWT <code>DockLayoutPanel</code> ⁶ setting the unit of measure to Pixel
RESTfulWebServiceException.java	RPC custom exception class
RESTfulWebServiceProxy.java	Remote service interface (synchronous) for generic RPC
RESTfulWebServiceProxyAsync.java	Remote service asynchronous interface for generic RPC
Utilities.java	Utilities class with common methods

3.2.2 Callback package

Source folder: eu.europa.ec.jrc.euosme.gwt.client.callback

This package groups the classes implemented to receive a response from a remote procedure call.

If an RPC is successful, then `onSuccess(Object)` is called, otherwise `onFailure(Throwable)` is called.

Each callable asynchronous method corresponds to a method in the correlated service interface. The asynchronous method always takes an `AsyncCallback<T>` as its last parameter, where `T` is the return type of the correlated synchronous method.

Table 4: Callback package files

File	Purpose
CodeListRpcCallback.java	Contains the implementation of the callback class to obtain a code list (and put the values in a defined ListBox)
DataThemesCallback.java	Callback from INSPIRE data themes service
InitRpcCallback.java	Callback from Configuration service
InspireServiceRpcCallback.java	Callback from INSPIRE service (to get UUID or to get HTML version of the metadata)

⁴ Google Web Toolkit Javadoc for class `DateBox`

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.1/com/google/gwt/user/datepicker/client/DateBox.html>

⁵ GWT Known issue 4532: Whole UI disappear in IE 7 when we Hover over the menubar menu item

<http://code.google.com/p/google-web-toolkit/issues/detail?id=4532>

⁶ Google Web Toolkit Javadoc for class `DockLayoutPanel`

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.1/com/google/gwt/user/client/ui/DockLayoutPanel.html>

SuggestCallback.java	Callback from GEMET service, used to get the keywords related to a defined repository
SuggestListCallback.java	Callback from GEMET service, used to get the list of available repositories
ValidationRpcCallback.java	Callback from Validation service

3.2.3 CodeLists package

Source folder: eu.europa.ec.jrc.euosme.gwt.client.codeLists

GWT has a mechanism for embedding text files into the application by specifying an associated `TextResource`⁷ within a `ClientBundle`⁸. The main advantage of that is that `ClientBundle` is compatible with GWT's `i18n`⁹ module and this approach offers a quickly management of offline resources. In this case we are talking about code lists.

Suppose you have defined a resource (in this application the resources are specified in the interface `client/myResource.java`):

Table 5: Example of definition of a ClientBundle

```
@Source("codeLists/11.txt")
public TextResource codeList11();
```

For each possible value of the `locale` deferred-binding property, the `ClientBundle` generator will look for variations of the specified filename in a manner similar to that of Java's `ResourceBundle`.

Suppose the `locale` were set to `fr_FR`. The generator would look for files in the following order:

- `codeLists/11_fr_FR.txt`
- `codeLists/11_fr.txt`
- `codeLists/11.txt`

⁷ Google Web Toolkit Javadoc for class `TextResource`

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.1/com/google/gwt/resources/client/TextResource.html>

⁸ Google Web Toolkit Javadoc for class `ClientBundle`

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.1/com/google/gwt/resources/client/ClientBundle.html>

⁹ Google Web Toolkit Javadoc for class `i18n client`

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.1/com/google/gwt/i18n/client/package-summary.html>

We have created a web service ad hoc with translated code lists. The application calls the web service, passing 3-alphabetic digits code in ISO 639 of the locale settings, and the result is a json file like the following:

Table 6: Example of results calling code list service

```
{ "@uri": "http://.../codelists/11/values",
  "value": {
    "@uri": "http://.../codelists/11/values/130/",
    "code": "test",
    "idCl": { "@uri": ".../codelists/11/values/130/idCl/" },
    "id": "130",
    "name": "test"
  }
}
```

Offline version of results from code list service

Table 7: CodeLists package files

File	Purpose
10[_locale].txt	Service type of data service
11[_locale].txt	Suggested organisation name for point of contact and responsible party
12[_locale].txt	Suggested conditions for access and use of spatial data sets and service
2[_locale].txt	Resource language (LanguageCode in ISO639, recommended 3-alphabetic digits code)
3[_locale].txt	Code for function performed by the online resource (CI_OnLineFunctionCode in ISO19115)
4[_locale].txt	Responsible party role (CI_RoleCode in ISO19115)
5[_locale].txt	Type of date (CI_DateTypeCode in ISO19115)
6[_locale].txt	Spatial data service type (MD_ScopeCode in ISO19115)
7[_locale].txt	Topic category code (MD_TopicCategoryCode in ISO19115)
8[_locale].txt	Subject matter used to group similar keywords (MD_KeywordTypeCode in ISO19115)
9[_locale].txt	Access constraints applied to assure the protection of privacy or intellectual property, and any special restrictions or limitations on

	obtaining the resource or metadata (MD_RestrictionCode in ISO19115)
--	---

3.2.4 I18n package (Internationalization)

Source folder: eu.europa.ec.jrc.euosme.gwt.client.i18n

The user interface of EUOSME has been translated into 22 of the 23 official languages of the European Union using the official translations of the legislation regarding metadata. The Irish language has been omitted because the official translation of the regulation has not been made available yet.

GWT internationalization support provides a variety of techniques to internationalize strings, typed values, and classes.

We have chosen the static string internationalization technique because it is the most efficient way to localize the application for different locales in terms of runtime performance. This approach is called "static" because it refers to creating tags that are matched up with human readable strings at compile time. At compile time, mappings between tags and strings are created for all languages defined in the module.

Static string internationalization uses traditional Java `.properties` files to manage translating tags into localized values. Main properties file and the corresponding `Constants/Messages` sub interface definition files are placed in this package.

The `Constants` interface allows localizing constant values in a type-safe manner.

The `Messages` interface allows to substitute parameters into messages and to even re-order those parameters for different locales as needed. The format of the messages in the properties files follows the specification in Java `MessageFormat`. The interface you create will contain a Java method with parameters matching those specified in the format string.

In addition, the `Messages` interface supports `Plural Forms` to accurately reflect text changes based on the count of something.

Both `Constants` and `Messages` use traditional Java properties files, with one notable difference: properties files used with GWT should be encoded as UTF-8 and may contain Unicode characters directly.

Table 8: I18n package files

File	Purpose
iso19115Constants.java	Constants file
iso19115Constants[_locale].properties	Constants properties file translated in each language
iso19115Messages.java	Messages file
iso19115Messages[_locale].properties	Messages properties file translated in each language

3.2.5 Images package

Source folder: eu.europa.ec.jrc.euosme.gwt.client.images

Table 9: Images package files

File	Purpose
loaderb16.gif	Loading image

3.2.6 ISO19115 package

Source folder: eu.europa.ec.jrc.euosme.gwt.client.iso19115

This package contains all the ISO 19115 / 19119 elements used by the application. Other elements can be created or the existing ones extended. See Write your own widget to represent a metadata element (extend to full ISO) for more details.

Table 10: ISO19115 package files

File	Purpose
CI_Address.java	Location (see In each class, “Full ISO” label, also defines if the interface covers all ISO 19115 elements or the developer has to add other elements to complete the ISO structure. For the INSPIRE profile only the relevant elements are shown, following the INSPIRE regulation. The other ISO elements are hidden. CI_Address) of the responsible person or organization
CI_Citation.java	Standardized resource reference (see CI_Citation)
CI_Contact.java	Information about the responsible person and/or organization (see CI_Contact)
CI_Date.java	Date (see CI_Date)
CI_OnlineResource.java	Information about on-line sources (see CI_OnlineResource)
CI_ResponsibleParty.java	Point of contact or responsible party (see CI_ResponsibleParty)
CI_Telephone.java	Telephone numbers for contacting the responsible individual or organization (see CI_Telephone)
Distance.java	Ground sample distance (see Distance)
DQ_ConformanceResult.java	Information about the outcome of evaluating the obtained value (or set of values) against a specified acceptable conformance quality level (see DQ_ConformanceResult)

DQ_Element.java	Quantitative quality information (see DQ_Element)
EX_GeographicBoundingBox.java	Geographic position of the dataset (see EX_GeographicBoundingBox)
MD_Constraints.java	Restrictions on the access and use of a resource or metadata (see MD_ConstraintsMD_)
MD_Identifier.java	Value uniquely identifying an object within a namespace (see MD_Identifier)
MD_Keywords.java	Provides category keywords, their type, and reference source (see MD_Keywords)
MD_Keywords_DataThemes.java	List of INSPIRE data themes (see MD_Keywords_DataThemes)
MD_Keywords_Gemet.java	List of keywords from GEMET (see MD_Keywords_Gemet)
MD_Keywords_INSPIRE.java	Group of tools to choose a keyword INSPIRE compliant (see MD_Keywords_INSPIRE)
MD_LegalConstraints.java	Restrictions and legal prerequisites for accessing and using the resource or metadata (see MD_LegalConstraints)
MD_Resolution.java	Level of detail expressed as a scale factor or a ground distance (see MD_Resolution)
TM_Primitive.java	Defines a period in the time frame (see TM_Primitive)

3.2.7 ISO19115 UI package

Source folder: eu.europa.ec.jrc.euosme.gwt.client.iso19115.ui

The data input is simplified using tabs. Each tab contains a set of fields and the grouping follows the pattern of the INSPIRE Metadata Regulation.

All the java classes of this section have their own UiBinder¹⁰ XML markup to improve the application efficiency, distinguish the layout from the coding.

¹⁰ GWT Declarative layout with UIBinder, <http://code.google.com/intl/it-IT/webtoolkit/doc/latest/DevGuideUiBinder.html>

Table 11: ISO19115 UI package files

File	Purpose
MainPanel.java and sub files	This is the base class (see Main Panel)
TabClassification.java and TabClassification.ui.xml	Classification of spatial data and services tab (see Tab Classification)
TabConformity.java and TabConformity.ui.xml	Conformity tab (see Tab Conformity)
TabConstraints.java and TabConstraints.ui.xml	Constraint related to access and use tab (see Tab Constraints)
TabGeographic.java and sub files	Geographic location tab (see Tab Geographic)
TabIdentification.java and TabIdentification.ui.xml	Identification tab (See Tab Identification)
TabKeyword.java and TabKeyword.ui.xml	Keywords tab (See Tab Keyword)
TabMetadata.java and TabMetadata.ui.xml	Metadata on metadata tab (see Tab Metadata)
TabOrganization.java and TabOrganization.ui.xml	Organizations responsible for the establishment, management, maintenance and distribution of spatial data sets and services (see Tab Organization)
TabQuality.java and TabQuality.ui.xml	Quality and validity tab (see Tab Quality)
Tabs.java and Tabs.ui.xml	This class includes a panel to groups all the pages (see Tabs)
TabTemporal.java and TabTemporal.ui.xml	Temporal reference tab (see Tab Temporal)

3.2.8 *UserGuides package*

Source folder: eu.europa.ec.jrc.euosme.gwt.client.userGuides

Table 12: UserGuides package files

File	Purpose
eurlex[_locale].htm	HTML version of legislation regarding metadata

3.2.9 *Widgets package*

Source folder: eu.europa.ec.jrc.euosme.gwt.client.widgets

These are the base classes to construct an element.

Table 13: Widgets package files

File	Purpose
Boolean.java and Boolean.ui.xml	True/false widget (see Boolean)
CharacterString.java and CharacterString.ui.xml	Text box widget (see CharacterString)
CharacterStringLong.java and CharacterStringLong.ui.xml	Text area widget, for long texts (see CharacterStringLong)
CharacterStringMultiple.java and CharacterStringMultiple.ui.xml	Text box with a table to add more than one string (CharacterStringMultiple)
CI.java and CI.ui.xml	This is the base class for a single element with a disclosure panel and a label, an help button and the widget that compose the metadata element (see CI)
CIMultiple.java and CIMultiple.ui.xml	For elements with repeating occurrences (see CIMultiple)
CodeList.java and CodeList.ui.xml	List box widget (see CodeList)
CodeListFree.java and CodeListFree.ui.xml	List box widget with a free text input box (see CodeListFree)
CodeListMultiple.java and CodeListMultiple.ui.xml	List box widget with a table to add more than one item from the list (CodeListMultiple)
DateImpl.java and DateImpl.ui.xml	Date box with calendar (see DateImpl)
GeoBoundsMultiple.java and GeoBoundsMultiple.ui.xml	Table with bounding boxes (see GeoBoundsMultiple)

3.2.10 XML sources package

Source folder: eu.europa.ec.jrc.euosme.gwt.client.XMLsources

These files are used by the application as a template, especially when the application is called to duplicate elements. For example, if the user clicks “Add responsible party”, the application points to the first responsible party element in the XML tree, and use it as a sample structure to replicate the code.

Table 14: XML sources package files

File	Purpose
dataset.xml	Template of INSPIRE spatial dataset
series.xml	Template of INSPIRE spatial dataset series
service.xml	Template of INSPIRE spatial data service

3.2.11 Server package

Source folder: eu.europa.ec.jrc.euosme.gwt.server

Table 15: Server package files

File	Purpose
DownloadServlet.java	Servlet used to save a file edited with EUOSME
InitServiceProxyImpl.java	Servlet to read configuration settings
LoadServlet.java	This servlet loads a file from local machine
RESTfulWebServiceProxyImpl.java	Servlet used to manage all RESTful services

3.3 Referenced libraries

Library	Logical name	Purpose
FileUpload 1.2.1 API	org.apache.commons.fileupload	The Commons FileUpload package makes it easy to add robust, high-performance, file upload capability to your servlets and web applications
Commons IO 1.4 API	org.apache.commons.io	This package defines utility classes for working with streams, readers, writers and files
GemetClient	eu.inspire.geoportal.viewclient.cache.gemet	Used to get the topmost concepts of GEMET web service
GWT 2.1.0		GWT SDK
AjaxLoader API Library for GWT 1.1.0	com.google.gwt.ajaxloader	The classes in this package provide GWT access to the google.load function in JavaScript to load other Google Ajax libraries.
Maps API Library for GWT 1.1.0	com.google.gwt.maps	The Google Maps API lets you embed Google Maps in your own web pages with JavaScript. The API provides a number of utilities for manipulating maps (just like on the http://maps.google.com web page) and adding content to the map through a variety of services, allowing you to create robust maps applications on your website.

3.4 Inherited modules

When module inherits other modules, their source paths are combined so that each module will have access to the translatable source it requires.

Table 16: Inherited modules

Module	Logical name	Contents
User	com.google.gwt.user.User	Core GWT functionality

I18n	com.google.gwt.i18n.I18N	Internationalization support for GWT applications
HTTP	com.google.gwt.http.HTTP	Low-level HTTP communications library
JSON	com.google.gwt.json.JSON	JSON creation and parsing
XML	com.google.gwt.xml.XML	XML document creation and parsing
Resources	com.google.gwt.resources.Resources	Classes for aggregating static resources into bundles
GoogleMaps	com.google.gwt.maps.GoogleMaps	The Google Maps API lets you embed Google Maps in your own web pages with JavaScript
AjaxLoader	com.google.gwt.ajaxloader.AjaxLoader	Allows dynamically loading an API from GWT code

3.5 War

The war directory is the deployment image of your web application. It is in the standard expanded war format recognized by a variety of Java web servers, including Tomcat, Jetty, and other J2EE servlet containers. It contains a variety of resources:

- Static content you provide, such as the host HTML page
- GWT compiled output
- Java class files and jar files for server-side code
- A web.xml file that configures your web app and any servlets

A detailed description of the war format is beyond the scope of this document, but here are the basic pieces you will want to know about:

Table 17: EUOSMEGWT/war/ files

Directory or file	Purpose
/EUOSMEGWT.html	The host HTML page that loads the application.
/EUOSMEGWT.css	Static style sheet that styles the application
/euosmegwt/	The module directory where the GWT compiler writes output and files on the public path are copied.
/euosmegwt/euosmegwt.nocache.js	The "selection script". This script must be loaded from the host HTML to load the GWT module into the page.
/images/	Images used in the application
/scripts/	JavaScript libraries such as Mapstraction and OpenLayers
/temp/	Temporary folder used to create files on the fly
/userguide/	User guides

/WEB-INF/	All non-public resources
/WEB-INF/web.xml	Configuration file (see Configuration settings)
/WEB-INF/classes	Java compiled class files live here to implement server side functionality. If you are using an IDE, set the output directory to this folder.
/WEB-INF/lib/	Library dependencies that the server code needs.
/WEB-INF/lib/gwt-servlet.jar	Used from servlets using GWT RPC, you will need to place a copy of gwt-servlet.jar here.
/wel/	European Commission templates

4 ISO19115 REPRESENTED CLASSES

This section describes a subset of ISO 19115 classes, starting from the elements required for the implementation of the INSPIRE Metadata Technical Guidelines. The full ISO model has not been implemented in this project, but we count on the developers' community to contribute to the further evolution of the project.

4.1 Path identification

Each metadata element that has a value (string, number, etc.) is represented in EUOSME with a specific html input element (such as a text box or a list box, etc.).

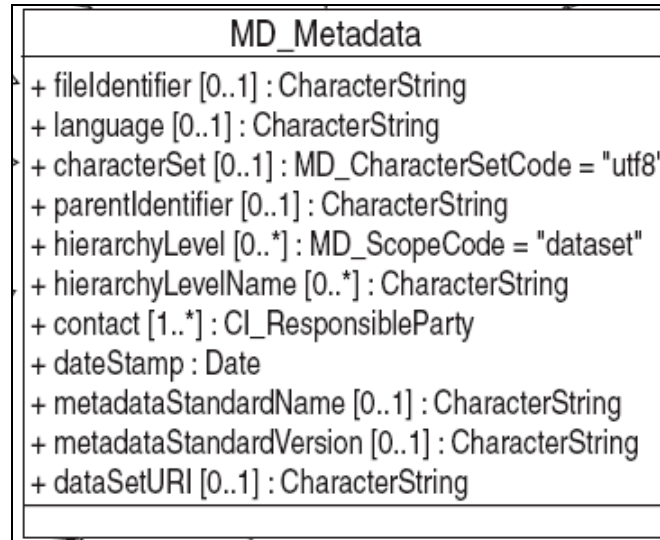
Each html input element has its own identifier.

In EUOSME the primary purpose of this identifier is to uniquely identify the element and also to supply a quickly access to the XML structure of the metadata. For this reason, the identifier represents the position of the element/node in the hierarchical structure of the final XML navigating through it. This approach is similar to XPath but simpler because it refers only to the name of the node and to its position. Like XPath, this method models an XML document as a tree of nodes.

There is an element node for every element in the XML document. An element node has a qualified name and it has an associated set of attribute nodes. Every node other than the root node has exactly one parent, which is either an element node or the root node. A root node or an element node is the parent of each of its child nodes.

For example the metadata is represented in ISO19115 by this UML diagram:

Figure 1: MD_Metadata class in UML



Have a look for example at the language element: it is the language used for documenting metadata and, as you can see, this element is optional (could be present zero or one time).

In ISO/TS 19139:2005 the language element has been extended and instead the use of `CharacterString`, it is recommended the use of `LanguageCode` (that is a `CodeList`). See the portion of an XML document explained below:

```

<?xml version="1.0" encoding="UTF-8"?>
<gmd:MD_Metadata
  xsi:schemaLocation="http://www.isotc211.org/2005/gmd
  http://schemas.opengis.net/iso/19139/20060504/gmd/gmd.xsd"
  xmlns:gmd="http://www.isotc211.org/2005/gmd" xmlns:gco="http://www.isotc211.org/2005/gco"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gml="http://www.opengis.net/gml"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <gmd:fileIdentifier>
    <gco:CharacterString>7ed9da1b.xml</gco:CharacterString>
  </gmd:fileIdentifier>
  <gmd:language>
    <gmd:LanguageCode
  codeList="http://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_19139_Schemas/resources/Code1
  
```

```
ist/ML_gmxCodelists.xml#LanguageCode" codeListValue="eng">eng</gmd:LanguageCode>
</gmd:language>
...
```

Using the mechanism mentioned above, the language will be identified by this path (note the use of lowercase):

```
md_metadata[1].language[1].languagecode[1]
```

Where [1] stands for the first children of the named item: in the example, navigate through the root MD_Metadata and through the first children element named language and then ending to the first children element of language node, named LanguageCode.

This identification is used for the elements which have a text value. So the identification through a path could be extended also for attributes.

We distinguish attributes from nodes, using the symbol @ before the qualified name.

For example, the element LanguageCode has two attributes: codeList and codeListValue. These attributes have these identifiers:

```
md_metadata[1].language[1].languagecode[1].@codelist[1]
md_metadata[1].language[1].languagecode[1].@codelistvalue[1]
```

The use of the [] symbols that indicate the position, in this case, and also in the case of the root element, could be omitted but for conformity with the other elements, it has been maintained.

4.2 Table structure

The following tables, which explain the collections of elements of each metadata entity, have these headers:

- Name, is the name of the metadata element
- Definition, a brief summary of the meaning of the element
- O/C stands for obligation or condition, metadata entities and elements shall be considered to be mandatory, conditional or optional as specified in the applicable profile
- Occur stands for maximum occurrence: Single occurrences are shown by “1”; repeating occurrences are represented by “N”. Fixed number occurrences other than one are allowed, and will be represented by the corresponding number.
- INSPIRE equal to Y indicates that the corresponding element is shown in the INSPIRE profile
- Class is the java class that represents the element (a link to each class is provided)
- Path is a partial identifier of the element (the full path is done starting from the root class)

In each class, “Full ISO” label, also defines if the interface covers all ISO 19115 elements or the developer has to add other elements to complete the ISO structure. For the INSPIRE profile only the relevant elements are shown, following the INSPIRE regulation. The other ISO elements are hidden.

4.3 CI_Address

Physical and email address at which the organization or individual may be contacted.

Full ISO: Yes.

Table 18: CI_Address elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
deliveryPoint	address line for the location	O	N	N	CharacterStringMultiple	.deliverypoint[n].characterstring[1]
city	city of the location	O	1	N	CharacterString	.city[1].characterstring[1]
administrativeArea	state, province of the location	O	1	N	CharacterString	.administrativearea[1].characterstring[1]
postalCode	ZIP or other postal code	O	1	N	CharacterString	.postalcode[1].characterstring[1]
country	country of the physical address	O	1	N	CharacterString	.country[1].characterstring[1]
electronicMailAddress	address of the electronic mailbox of the responsible organization or individual	O	N	Y	CharacterStringMultiple	.electronicmailaddress[n].characterstring[1]

4.4 CI_Citation

This class is used in many tabs:

- Tab Identification: citation data for the resource;
- Tab Keyword: name of the formally registered thesaurus or a similar authoritative source of keywords;
- Tab Conformity: citation of product specification or user requirement against which data is being evaluated (see DQ_ConformanceResult)

Full ISO: No.

Table 19: CI_Citation elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
title	name by which the cited resource is known	M	1	Y	CharacterString	.title[1].characterstring[1]
alternateTitle	short name or other language name by which the cited information is known	O	N	N	CharacterString	.alternatetitle[1].characterstring[1]
date	reference date for the cited resource	M	N	Y	CIMultiple of CI_Date	.date[1].ci_date[1]
identifier	value uniquely identifying an object within a namespace	O	N	Y	CIMultiple of MD_Identifier	.identifier[1].rs_identifier[1]

4.5 CI_Contact

It is a child of CI_ResponsibleParty and it refers to Information required enabling contact with the responsible person and/or organization

Full ISO: Yes.

Table 20: CI_Contact elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
phone	telephone numbers at which the organization or individual may be contacted	O	1	N	CI_Telephone	.phone[1].ci_telephone[1]
address	physical and email address at which the organization or individual may be contacted	O	1	Y	In each class, "Full ISO" label, also defines if the interface covers all ISO 19115 elements or the developer has to add other elements to complete the ISO structure. For the INSPIRE profile only the relevant elements are shown, following the INSPIRE regulation. The other ISO elements are hidden. CI_Address	.address[1].ci_address[1]
onlineResource	on-line information that can be used to contact the individual or organization	O	1	Y	CI_OnlineResource	.onlineresource[1].ci_onlineresource[1]
hoursOfService	time period (including time zone) when individuals can contact the organization or individual	O	1	N	CharacterString	.hoursofservice[1].characterstring[1]
contactInstructions	supplemental instructions on how or when to contact the individual or organization	O	1	N	CharacterString	.contactinstructions[1].characterstring[1]

4.6 CI_Date

It is a child of CI_Citation and it is a reference date for the cited resource.

Full ISO: Yes.

Table 21: CI_Date elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
date	reference date for the cited resource	M	1	Y	DateImpl	.date[1].date[1]
dateType	event used for reference date	M	1	Y	CodeList with ID 5	.datetype[1].ci_datetypecode[1]

4.7 CI_OnlineResource

It is a child of CI_Contact and it refers to information about on-line sources from which the dataset, specification, or community profile name and extended metadata elements can be obtained.

It is also used into Tab Identification to identify the information about online sources from which the resource can be obtained.

Full ISO: Yes.

Table 22: CI_OnlineResource elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
linkage	location (address) for on-line access using a Uniform Resource Locator address or similar	M	1	Y	CharacterString	.linkage[1].url[1]
protocol	connection protocol to be used	O	1	N	CharacterString	.protocol[1].characterstring[1]
applicationProfile	name of an application profile that can be used with the online resource	O	1	N	CharacterString	.applicationprofile[1].characterstring[1]
name	name of the online resource	O	1	N	CharacterString	.name[1].characterstring[1]
description	detailed text description of what the online resource is/does	O	1	N	CharacterString	.description[1].characterstring[1]
function	code for function performed by the online resource	O	1	N	CodeList with ID 3	.function[1].ci_onlinefunctioncode[1]

4.8 CI_ResponsibleParty

It is included in:

- Tab Metadata: point of contact(s) associated with the resource(s).
- Tab Organization: responsible party for the metadata information

Individual name or organisationName or positionName must be specified.

The role of a point of contact is automatically set.

Full ISO: Yes.

Table 23: CI_ResponsibleParty elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
individualName	name of the responsible person surname, given name, title separated by a delimiter	C	1	N	CharacterString	.individualname[1].characterstring[1]
organisationName	name of the responsible organization	C	1	Y	CharacterString	.organisationname[1].characterstring[1]
positionName	role or position of the responsible person	C	1	N	CharacterString	.positionname[1].characterstring[1]
contactInfo	address of the responsible party	O	1	Y	CI_Contact	.contactinfo[1].ci_contact[1]
role	function performed by the responsible party	M	1	Y	CodeList with ID 4	.role[1].ci_rolecode[1]

4.9 CI_Telephone

Telephone numbers for contacting the responsible individual or organization. It is a child of CI_Contact.

Full ISO: Yes.

Table 24: CI_Telephone elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
voice	telephone number by which individuals can speak to the responsible organization or individual	O	N	N	CharacterStringMultiple	<code>.voice[1] characterstring[1]</code>
facsimile	telephone number of a facsimile machine for the responsible organization or individual	O	N	N	CharacterStringMultiple	<code>.facsimile[1] characterstring[1]</code>

4.10 Distance

Ground sample distance. It is a child of MD_Resolution. This class is documented in ISO/TS 19103.

Full ISO: Yes.

Table 25: Distance elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
distance	distance	M	1	Y	CharacterString	<code>.distance[1]</code>
uom	unit of measure	M	1	Y	CharacterString	<code>.distance[1].@uom</code>

4.11 DQ_ConformanceResult

Information about the outcome of evaluating the obtained value (or set of values) against a specified acceptable conformance quality level.

Child of DQ_Element.

Full ISO: Yes.

Table 26: DQ_ConformanceResult elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
specification	citation of product specification or user requirement against which data is being evaluated	M	1	Y	CI_Citation	<code>.specification[1].ci_citation[1]</code>
explanation	explanation of the meaning of conformance for this result	M	1	Y	CharacterString	<code>.explanation[1].characterstring[1]</code>
pass	indication of the conformance result where 0 = fail and 1 = pass	M	1	Y	Boolean	<code>.pass[1].boolean[1]</code>

4.12 DQ_Element

Aspect of quantitative quality information.

It appears in the Tab Conformity.

Full ISO: No.

Table 27: DQ_Element elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
measureidentification	code identifying a registered standard procedure	O	1	Y	MD_Identifier	.measureidentification[1].rs_identifier[1]
result	value (or set of values) obtained from applying a data quality measure or the outcome of evaluating the obtained value (or set of values) against a specified acceptable conformance quality level	M	2	Y	DQ_ConformanceResult	.result[1].dq_conformanceresult[1]

4.13 EX_GeographicBoundingBox

Geographic position of the dataset.

Full ISO: Yes.

Table 28: EX_GeographicBoundingBox elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
westBoundLongitude	western-most coordinate of the limit of the dataset extent, expressed in longitude in decimal degrees (positive east)	M	1	Y	CharacterString	.westboundlongitude[1].decimal[1]
eastBoundLongitude	eastern-most coordinate of the limit of the dataset extent, expressed in longitude in decimal degrees (positive east)	M	1	Y	CharacterString	.eastboundlongitude[1].decimal[1]
southBoundLatitude	southern-most coordinate of the limit of the dataset extent, expressed in latitude in decimal degrees (positive north)	M	1	Y	CharacterString	.southboundlatitude[1].decimal[1]
northBoundLatitude	northern-most, coordinate of the limit of the dataset extent expressed in latitude in decimal degrees (positive north)	M	1	Y	CharacterString	.northboundlatitude[1].decimal[1]

4.14 MD_Constraints

Restrictions on the access and use of a resource or metadata.

It is included in Tab Constraints.

Full ISO: Yes.

Table 29: MD_Constraints elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
useLimitation	limitation affecting the fitness for use of the resource or metadata	O	N	Y	CharacterString	.useLimitation[1].characterstring[1]

4.15 MD_Identifier

Value uniquely identifying an object within a namespace.

Full ISO: No

Table 30: MD_Identifier elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
code	alphanumeric value identifying an instance in the namespace	M	1	Y	CharacterString	.code[1].characterstring[1]
codeSpace	name or identifier of the person or organization responsible for namespace	O	1	Y	CharacterString	.codespace[1].characterstring[1]
version	version identifier for the namespace	O	1	N	CharacterString	.version[1].characterstring[1]

4.16 MD_Keywords

Provides category keywords, their type, and reference source.

Full ISO: Yes.

Table 31: MD_Keywords elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
keyword	commonly used word(s) or formalized word(s) or phrase(s) used to describe the subject	M	N	Y	CharacterString	.keyword[1].characterstring[1]
Type	subject matter used to group similar keywords	O	1	Y	CodeList with ID 8	.type[1].md_keywordtypecode[1]
ThesaurusName	name of the formally registered thesaurus or a similar authoritative source of keywords	O	1	Y	CI_Citation	.thesaurusname[1].ci_citation[1]

4.16.1 MD_Keywords_DataThemes

This class proposes a list with the INSPIRE data themes. To obtain this list the application calls the RESTful service that gets the topmost concepts using the GemetClient library.

4.16.2 MD_Keywords_Gemet

This class lets the user to choose a keyword from all the available repository of GEMET.

4.16.3 MD_Keywords_INSPIRE

This class includes other classes so the user could choose the keyword from:

- INSPIRE data theme
- GEMET
- Alternatively, type a free keyword.

It is used in the Tab Keyword.

4.17 MD_LegalConstraints

Restrictions and legal prerequisites for accessing and using the resource or metadata.

OtherConstraints is used if the user does not fill accessConstraints or useConstraints elements.

It is included in Tab Constraints.

Full ISO: Yes.

Table 32: MD_LegalConstraints elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
accessConstraints	access constraints applied to assure the protection of privacy or intellectual property, and any special restrictions or limitations on obtaining the resource or metadata	O	N	N	CodeListMultiple with ID 9	<code>.accessconstraints[1].md_restrictioncode[1]</code>
useConstraints	constraints applied to assure the protection of privacy or intellectual property, and any special restrictions or limitations or warnings on using the resource or metadata	O	N	Y	CodeListMultiple with ID 9	<code>.useconstraints[1].md_restrictioncode[1]</code>
otherConstraints	other restrictions and legal prerequisites for accessing and using the resource or metadata	C	N	Y	CharacterStringMultiple	<code>.otherconstraints[1].characterstring[1]</code>

4.18 MD_Resolution

Level of detail expressed as a scale factor or a ground distance.

Input equivalentScale or distance.

Full ISO: Yes.

Table 33: MD_Resolution elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
equivalentScale	level of detail expressed as the scale of a comparable hardcopy map or chart	C	1	Y	CharacterString	<code>.equivalentscale[1].md_representativefraction[1].denominator[1].integer[1]</code>
distance	ground sample distance	C	1	Y	Distance	<code>.distance[1]</code>

4.19 TM_Primitive

TM_Primitive is class representing a non-decomposed element of geometry or topology. This ISO class is fully documented in ISO 19108. Full ISO: No.

Table 34: TM_Primitive elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
beginDate	starting date	M	1	Y	DateImpl	<code>.extent[1].timeperiod[1].beginposition[1]</code>
endDate	ending date	M	1	Y	DateImpl	<code>.extent[1].timeperiod[1].endposition[1]</code>

5 EUOSME CLASSES

ISO Elements are grouped into different tabs.

5.1 Main Panel

This class contains the entire user interface and specifically includes:

- HTML header
- A standard menu bar widget (GWT MenuBar¹¹)
- A panel that adds user-positioned splitters between each of its child widgets (GWT SplitLayoutPanel¹²) with:
 - Tabs

¹¹ Google Web Toolkit Javadoc of Class MenuBar

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.1/com/google/gwt/user/client/ui/MenuBar.html>

¹² Google Web Toolkit Javadoc of Class SplitLayoutPanel

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.1/com/google/gwt/user/client/ui/SplitLayoutPanel.html>

- a panel that stacks its children vertically, displaying only one at a time, with a header for each child which the user can click to display (GWT StackLayoutPanel¹³) with:
 - A standard hierarchical tree widget (GWT Tree¹⁴) with the XML source of the file
 - HTML preview of the metadata

HTML preview of the metadata is obtained through a RPC mechanism that call the related INSPIRE service.

5.2 Tabs

It is a panel that represents a tabbed set of pages, each of which contains another widget, in particular one of the tabs defined below.

5.3 Tab Classification

Topic category is displayed only for Spatial Data Sets and Dataset series, while Service type is used only for metadata related to Spatial Data Services.

Table 35: Tab Classification elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
topicCategory	main theme(s) of the dataset	M	N	Y	CodeListMultiple with ID 10	md_metadata[1].identificationinfo[1].MD_DataIdentification[1].topiccategory[1].md_topiccategorycode[1]
serviceType (services ¹⁵)	a service type name from a registry of services	M	1	Y	CodeList with ID 10	md_metadata[1].identificationinfo[1].SV_ServiceIdentification[1].servicetype[1].localname[1]

5.4 Tab Conformity

The requirements relating to the conformity and the degree of conformity shall be addressed by the current tab.

¹³ Google Web Toolkit Javadoc of Class StackLayoutPanel

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.1/com/google/gwt/user/client/ui/StackLayoutPanel.html>

¹⁴ Google Web Toolkit Javadoc for class Tree

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.1/com/google/gwt/user/client/ui/Tree.html>

¹⁵ ISO 19119:2005

Table 36: Tab Conformity elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
report	quantitative quality information for the data specified by the scope	O	1	Y	DQ_Element	md_metadata[1].dataqualityinfo[1].dq_dataquality[1].report[1].dq_do mainconsistency[1]

5.5 Tab Constraints

This tab is used to add constraints related to access and use.

Table 37: Tab Constraints elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
useLimitation	limitation affecting the fitness for use of the resource or metadata	O	N	N	MD_Constraints	md_metadata[1].identificationinfo[1].MD_DataIdentifi cation[1].resourceconstraints[1].md_constraints[1]
legalConstraints	Restrictions and legal prerequisites for accessing and using the resource or metadata	O	N	Y	MD_LegalConstraints	md_metadata[1].identificationinfo[1].MD_DataIdentifi cation[1].resourceconstraints[2].md_legalconstraints [1]

5.6 Tab Geographic

This tab is used to satisfy the requirement for a geographic location. That is done with the utilization of a map (see Maps for more details) and a widget ad hoc named GeoBoundsMultiple that allows the automatic addition of a bounding box.

Table 38: Tab Geographic elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
geographicElement	provides geographic component of the extent of the referring object	O	N	Y	GeoBoundsMultiple	md_metadata[1].identificationinfo[1].MD_DataIdentifica tion[1].extent[1].ex_extent[1].geographicElement[1]

5.7 Tab Identification

Identification tab displays differently for Spatial Datasets and Spatial Data Services.

Table 39: Tab Identification elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
citation	citation data for the resource(s)	M	1	Y	CI_Citation	md_metadata[1].identificationinfo[1].MD_DataIdentification[1].citation[1].ci_citation[1]
abstract	brief narrative summary of the content of the resource(s)	M	1	Y	CharacterStringLong	md_metadata[1].identificationinfo[1].MD_DataIdentification[1].abstract[1].characterstring[1]
hierarchyLevel	scope to which the metadata applies	O	N	Y	CodeList with ID 6	md_metadata[1].hierarchyLevel[1].md_scopecode[1]
onLine	information about online sources from which the resource can be obtained	O	N	Y	CI_Multiple of CI_OnlineResource	md_metadata[1].distributioninfo[1].md_distribution[1].transferoptions[1].md_digitaltransferoptions[1].online[1].ci_onlineresource[1]
language	language(s) used within the dataset	M	N	Y	CodeListMultiple with ID 2	md_metadata[1].identificationinfo[1].MD_DataIdentification[1].language[1].languagecode[1]
operatesOn (services)	provides information on the datasets that the service operates on	O	N	Y	CharacterStringMultiple	md_metadata[1].identificationinfo[1].SV_ServiceIdentification[1].operateson[1]

5.8 Tab Keyword

This tab lets the user to enter keywords that describe the resource. We have made a class ad hoc to simplify the addition of keywords as a free text or originated from a controlled vocabulary. In the INSPIRE profile is requested at least one keyword from the general environmental multilingual thesaurus (GEMET) describing the relevant spatial data theme.

Table 40: Tab Keywords elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
descriptiveKeywords	provides category keywords, their type, and reference source	O	N	Y	MD_Keywords_INSPIRE	md_metadata[1].identificationinfo[1].MD_DataIdentification[1].descriptivekeywords[1].md_keywords[1]

5.9 Tab Metadata

This tab groups information about metadata on metadata, in particular the metadata point of contact, date and language.

Table 41: Tab Metadata elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
fileIdentifier	unique identifier for this metadata file	O	1	Y	CharacterString	md_metadata[1].fileIdentifier[1].characterstring[1]
language	language used for documenting metadata	O	1	Y	CodeList with ID 2	md_metadata[1].language[1].languagecode[1]

pointOfContact	identification of, and means of communication with, person(s) and organization(s) associated with the resource(s)	O	N	Y	CIMultiple of CI_ResponsibleParty	md_metadata[1].contact[1].ci_responsibleparty[1]
dateStamp	date that the metadata was created	M	1	Y	DateImpl	md_metadata[1].datestamp[1].date[1]

5.10 Tab Organization

Organization groups the elements related to the organisation responsible for the establishment, management, maintenance and distribution of the resource.

Table 42: Tab Organization elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
contact	party responsible for the metadata information	M	N	Y	CIMultiple of CI_ResponsibleParty	md_metadata[1].identificationinfo[1].MD_DataIdentification[1].pointofcontact[1].ci_responsibleparty[1]

5.11 Tab Quality

The requirements relating to the quality and validity of spatial data shall be addressed by this tab.

Table 43: Tab Quality elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
lineage	non-quantitative quality information about the lineage of the data specified by the scope	O	1	Y	CharacterStringLong	md_metadata[1].dataqualityinfo[1].dq_dataquality[1].lineage[1].li_lineage[1].statement[1].characterstring[1]
spatialResolution	factor which provides a general understanding of the density of spatial data in the dataset	O	N	Y	CIMultiple of MD_Resolution	md_metadata[1].identificationinfo[1].MD_DataIdentification[1].spatialresolution[1].md_resolution[1]

5.12 Tab Temporal

This tab element addresses the requirement to have information on the temporal dimension of the data.

Table 44: Tab Temporal elements

Name	Definition	O/C	Occur.	INSPIRE	Class	Path
temporalElement	provides temporal component of the extent of the referring object	O	N	Y	CI Multiple of TM_Primitive	md_metadata[1].identificationinfo[1].MD_DataIdentification[1].extent[1].ex_extent[1].temporalelement[1].ex_temporalextent[1]
publicationDate	this is the date of publication of the resource when available, or the date of entry into force	O	1	Y	CI_Date	md_metadata[1].identificationinfo[1].MD_DataIdentification[1].citation[1].ci_citation[1].date[1].ci_date[1]
revisionDate	this is the date of last revision of the resource, if the resource has been revised	O	1	Y	CI_Date	md_metadata[1].identificationinfo[1].MD_DataIdentification[1].citation[1].ci_citation[1].date[2].ci_date[1]
creationDate	this is the date of creation of the resource	O	1	Y	CI_Date	md_metadata[1].identificationinfo[1].MD_DataIdentification[1].citation[1].ci_citation[1].date[3].ci_date[1]

6 WIDGETS

This section explains the widgets which are the basis for building a metadata element. The names used recall ISO 19115 data types that include primitive pre-defined types and user-definable types (such as `CharacterString`¹⁶ or `CodeList`¹⁷ data type).

Each widget is composed by these common elements:

- A `GWT DisclosurePanel`¹⁸ that consists of a header and a content panel that discloses the content when a user clicks on the header.
- A header with a label and if the element is required the symbol of mandatory (*).
- A content panel with a group of widget or a single widget
- A button that supplies information about the element

6.1 CI

The content panel contains a group of metadata elements, an error label (shown if requested) and a button to remove this panel.

In case of multiple elements, for example, point of contact can occur more than once and this button appears for the second point of contact and above.

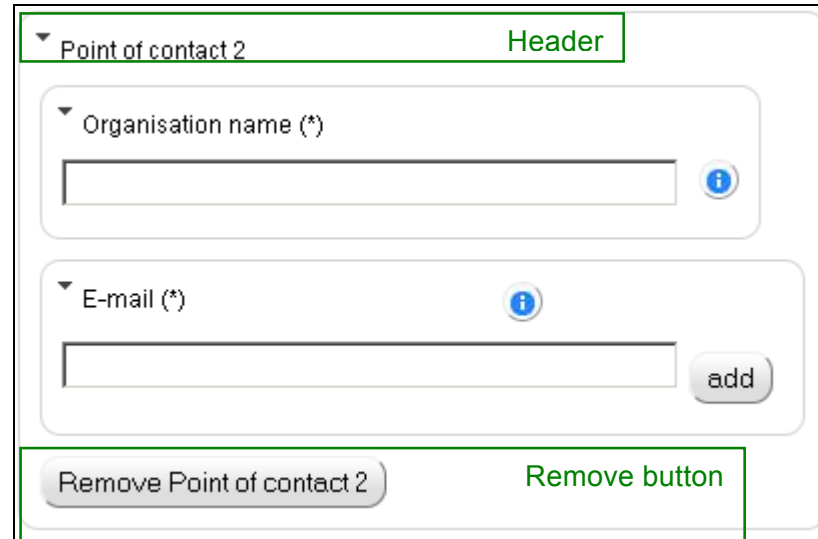
¹⁶ `CharacterString` represents a free text value

¹⁷ `CodeList` is used to describe a more open enumeration. `CodeList` is a flexible enumeration. Code lists are useful for expressing a long list of potential values. If the elements of the list are completely known, an enumeration should be used; if the only likely values of the elements are known, a code list should be used.

¹⁸ Google Web Toolkit Javadoc for class `DisclosurePanel`

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.0/com/google/gwt/user/client/ui/DisclosurePanel.html>

Figure 2: Example of CI



Point of contact 2 Header

Organisation name (*)

E-mail (*) add

Remove Point of contact 2 Remove button

6.2 CIMultiple

This class is used for elements composed by a group of fields that could be replicated. For example, the responsible party can occur more than once in the metadata and a responsible party is composed by some elements (organization name, role...). A disclosure panel with a common header and a button to add elements of this type (for example, to add another responsible party) composes this widget.

Figure 3: Example of CIMultiple



The screenshot displays a web form titled "Responsible party (*)" with a dropdown arrow. Inside this container, there is a sub-section "Responsible party 1 (*)" which contains three input fields: "Organisation name (*)" (a text box with an information icon), "E-mail (*)" (a text box with an "add" button and an information icon), and "Responsible party role (*)" (a dropdown menu with the text "--select a value" and an information icon). At the bottom of the "Responsible party 1" section is a button labeled "Add Responsible party".

6.3 Boolean

This widget presents a list of choices to the user, either as a list box (GWT `ListBox`¹⁹) with three values:

- Empty

¹⁹ Google Web Toolkit Javadoc for class `ListBox`

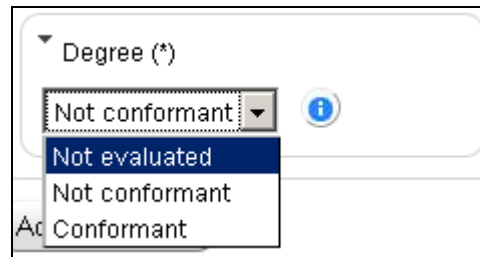
<http://google-web-toolkit.googlecode.com/svn/javadoc/2.0/com/google/gwt/user/client/ui/ListBox.html>

- True
- False

It is used for example in the Tab Conformity to identify the degree:

- not evaluated (empty value)
- not conformant (false)
- conformant (true)

Figure 4: Example of Boolean



6.4 CharacterString

`CharacterString` is a standard single-line text box (GWT `TextBox`²⁰) that usually contains a string or a number.

The type of possible values is defined by the parameter `CheckFunction` that could have these values:

- Electronic mails address (`electronicMailAddress`)
- URL (`URL`)
- Sequence of characters and numbers (`normal`)
- Integer number (`integer`)
- Double number (`doublenum`)

²⁰ Google Web Toolkit Javadoc for class `TextBox`

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.0/com/google/gwt/user/client/ui/TextBox.html>

Figure 5: Example of CharacterString

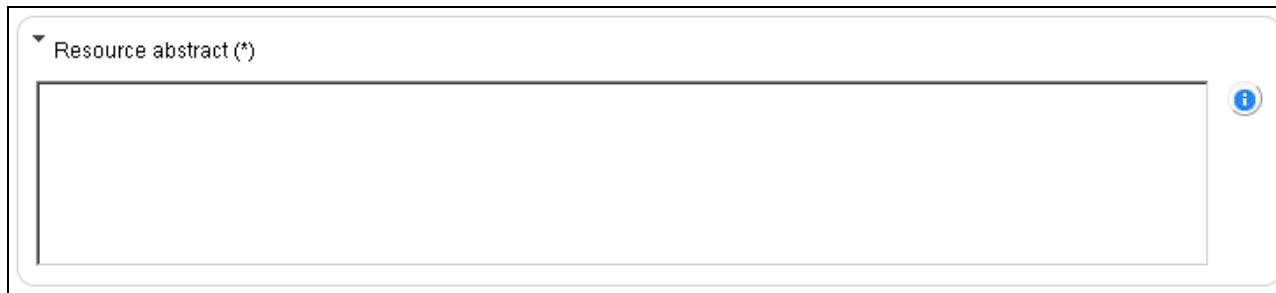


A screenshot of a web form element. It features a label 'Resource title (*)' with a small downward-pointing triangle to its left. Below the label is a single-line text input field. To the right of the input field is a circular blue information icon containing a white lowercase 'i'.

6.5 CharacterStringLong

A text box that allows multiple lines of text (long string) to be entered (GWT `TextArea`²¹).

Figure 6: Example of CharacterStringLong



A screenshot of a web form element. It features a label 'Resource abstract (*)' with a small downward-pointing triangle to its left. Below the label is a large, multi-line text area. To the right of the text area is a circular blue information icon containing a white lowercase 'i'.

6.6 CharacterStringMultiple

`CharacterStringMultiple` represents a group of `CharacterString`.

For this reason, it is composed by:

- A `CharacterString` object with a button that allows the user to add the typed string
- A table (GWT `FlexTable`²²) to hold the added string and, for each value, a button to remove the item

²¹ Google Web Toolkit Javadoc for class `TextArea`

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.0/com/google/gwt/user/client/ui/TextArea.html>

²² Google Web Toolkit Javadoc for class `FlexTable`

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.1/com/google/gwt/user/client/ui/FlexTable.html>

Figure 7: Example of CharacterStringMultiple

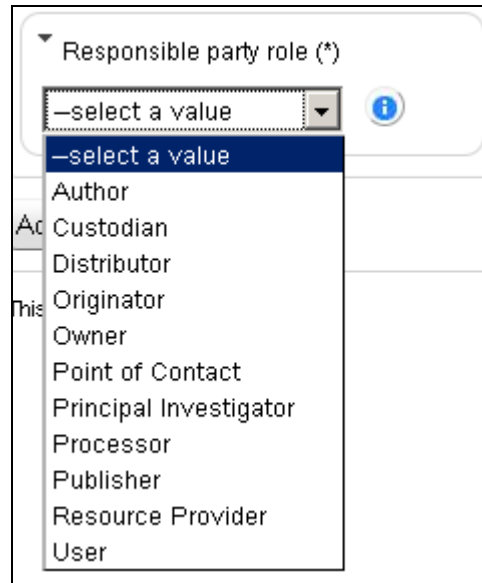


The image shows a user interface for a 'CharacterStringMultiple' widget. It is titled 'E-mail (*)' with a dropdown arrow on the left and an information icon on the right. Below the title, there are two rows of text, each followed by a circular 'x' button for removal. The first row contains 'mario.rossi@jrc.ec.europa.eu' and the second row contains 'mario.bianchi@jrc.ec.europa.eu'. At the bottom, there is a text input field containing 'another.value@email.com' and a rounded 'add' button to its right.

6.7 CodeList

`CodeList` is a list box that displays lists of available options from the *codelist service* (see [Section 7 of this document](#)) or from the offline resources (`CodeLists` package). The developer could choose from online or offline, changing the value of the public field *rpcCodeList* into `Main Panel` class (false stands for offline resources, true stands for online resources obtained by a client server communication).

Figure 8: Example of CodeList



6.8 CodeListFree

A `CodeListFree` is a text box that displays a pre-configured set of selections that match the user's input (GWT `SuggestBox`²³). Each suggests box is associated with a single `SuggestOracle`²⁴. The `SuggestOracle` is used to provide a set of selections given a specific query string and is attached to a code list resource.

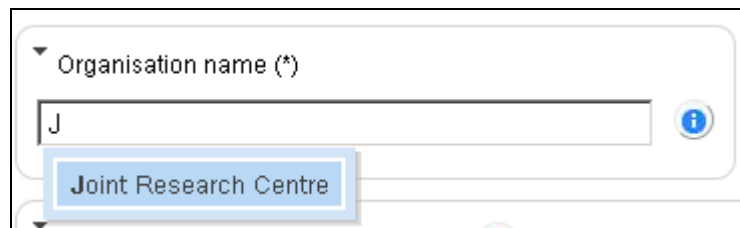
²³Google Web Toolkit Javadoc for class `SuggestBox`

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.0/com/google/gwt/user/client/ui/SuggestBox.html>

²⁴Google Web Toolkit Javadoc for class `SuggestOracle`

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.0/com/google/gwt/user/client/ui/SuggestOracle.html>

Figure 9: Example of CodeListFree



The screenshot shows a form titled "Organisation name (*)". It features a text input field containing the letter "J". Below the input field, a dropdown menu is open, displaying the option "Joint Research Centre". An information icon (i) is located to the right of the input field.

6.9 CodeListMultiple

It is similar to `CharacterStringMultiple` but the element to add to the table is a `CodeList` and not a `CharacterString`.

Figure 10: Example of CodeListMultiple



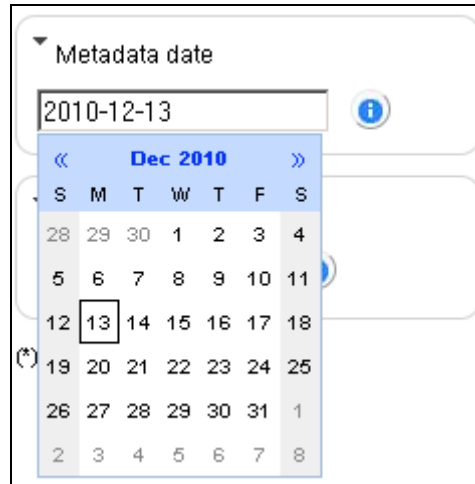
The screenshot shows a form titled "Resource language (*)". It displays a list of selected languages: "eng" and "ita", each with a small "x" button to its right. Below this list, there is a dropdown menu with "italian" selected, an "add" button, and an information icon (i).

6.10 DateImpl

`DateImpl` is a text box that shows a GWT `DatePicker`²⁵ when the user focuses on it.

²⁵ <http://google-web-toolkit.googlecode.com/svn/javadoc/1.6/com/google/gwt/user/datepicker/client/DatePicker.html>

Figure 11: Example of DateImpl



6.11 GeoBoundsMultiple

This widget ensures the addition of geographic bounding boxes (north, south, east and west) into a flex table. The values of latitudes and longitudes are taken from the map or typed as a free text by the user.

Figure 12: Example of GeoBoundsMultiple

Geographic bounding box (*)

North Bound Latitude	East Bound Longitude	South Bound Latitude	West Bound Longitude
70.45078916539954	50.1231934817182	36.490256037843984	-37.76741248171768

7 SERVICES

EUOSME is supported by different online services through GWT RPC framework.

This library let the application transparently makes calls to web services and let EUOSME to provide access to these server side resources.

In order to do that, we have defined the following classes:

- An interface for your service that extends `GWT RemoteService`²⁶ and lists all RPC methods named `RESTfulWebServiceProxy` in the Client package
- A class to implement the server-side code that extends `GWT RemoteServiceServlet`²⁷ and implements the interface created above: `RESTfulWebServiceProxyImpl` in Server package
- An asynchronous interface to RPC service to be called from the client-side code, named `RESTfulWebServiceProxyAsync` in the Client package

The methods defined are:

²⁶ Google Web Toolkit Javadoc of Interface Remote Service

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.1/com/google/gwt/user/client/rpc/RemoteService.html>

²⁷ Google Web Toolkit Javadoc of Class RemoteServiceServlet

<http://google-web-toolkit.googlecode.com/svn/javadoc/2.1/com/google/gwt/user/server/rpc/RemoteServiceServlet.html>

Table 45: List of RPC methods

Name	Purpose	Result
invokeGetRESTfulWebService	It is used to contact the code lists service (see CodeList) and the repository service (see MD_Keywords)	Json for the code lists and Xml for the repositories
invokeGemetService	Get top most concepts related to INSPIRE data themes	Map (object that maps keys to values)
invokeUpdateRESTfulWebService	Update the resources related to the code lists (see CodeLists package)	Json
invokeValidationService	Invoke the validation service passing the XML file related to the metadata	Html
invokeInspireMetadataConverterService	Call the service named Metadata Converter that given a metadata in XML, returns the html representation of it	Html
invokeInspireUUIDService	Get the UUID code (see MD_Identifier)	Xml

8 CUSTOMIZATION

8.1 Write your own widget to represent a metadata element (extend to full ISO)

8.1.1 Example of adding new elements (Extension of CI_Citation)

In the current version of EUOSME the metadata element CI_Citation is partially implemented: only elements INSPIRE compliant have been considered (title, alternateTitle, date and identifier).

To complete the full ISO representation of CI_Citation we have to add the following elements:

Name	Definition	O/C	Occur.	Class	Path
edition	version of the cited resource	O	1	CharacterString	<code>.edition[1].characterstring[1]</code>
editionDate	date of the edition	O	1	DateImpl	<code>.editiondate[1].date[1]</code>
citedResponsibleParty	name and position information for an individual or organization that is responsible for the resource	O	N	CI Multiple of CI_ResponsibleParty	<code>.citedresponsibleparty[1].ci_responsibleparty[1]</code>
presentationForm	mode in which the resource is represented	O	N	New CodeListMultiple containing CI_PresentationFormCode values	<code>.presentationForm[1].ci_presentationformcode[1]</code>
series	information about the series, or aggregate dataset, of which the dataset	O	1	New class named	<code>.series[1].ci_series[1]</code>

	is a part			CI_Series	
otherCitationDetails	other information required to complete the citation that is not recorded elsewhere	O	1	CharacterString	<code>othercitationdetails[1].characterstring[1]</code>
collectiveTitle	common title with holdings note	O	1	CharacterString	<code>collectivetitle[1].characterstring[1]</code>
ISBN	international Standard Book Number	O	1	CharacterString	<code>isbn[1].characterstring[1]</code>
ISSN	international Standard Serial Number	O	1	CharacterString	<code>issn[1].characterstring[1]</code>

As you can see, only one class has to be created and only one new code list.

About the new class `CI_Series` it will contain known classes:

Name	Definition	O/C	Occur.	Class	Path
name	name of the series, or aggregate dataset, of which the dataset is a part	O	1	CharacterString	<code>.name[1].characterstring[1]</code>
issueidentification	information identifying the issue of the series	O	1	CharacterString	<code>.issueidentification[1].characterstring[1]</code>
page	details on which pages of the publication the article was published	O	1	CharacterString	<code>.page[1].characterstring[1]</code>

The efforts spent to complete the metadata entity of the example are: add a new class and a new code list and reuse the existing classes. So it could be considered as minimal.

Let me explain in detail at java programming code, what the developer has to do:

- Add the code list values related to `CI_PresentationFormCode` to the code list service and to the local resources (see `CodeLists` package).
- Add the constants as the labels in the properties files `iso19115Constants` (see `I18n` package (Internationalization)):

```
# Description: FIELD TITLE: Edition
editionValue=Version of the cited resource
# Description: FIELD TITLE: Edition date
editionDateValue=Date of the edition
...
```

- Create new fields, related to the above mentioned elements, in class `CI_Citation`:

```
/** version of the cited resource */
CharacterString editionObj = new CharacterString(constants.editionValue(), "", false,
CheckFunctions.normal);
/** date of the edition */
DateImpl editionDateObj = new DateImpl(constants.editionDateValue(), "", false);
...
```

- Add these new fields (highlighted in yellow below) to the fields group panel of the composite object `CI_Citation`:

```

public CI_Citation(String label, boolean required, boolean multiple) {
    super(label, required, multiple);
    // Show the fields if required
    fieldsGroup.add(titleObj);
    fieldsGroup.add(alternateTitleObj);
    fieldsGroup.add(identifierContainerObj);
    fieldsGroup.add(dateContainerObj);
    fieldsGroup.add(editionObj);
    fieldsGroup.add(editionDateObj);
    ...
    // Set interface
    setInterface(-1);
}

```

- Add the corresponding checks (done during validation) to the function `myCheck` of the class:

```

@Override
public void myCheck() {
    if (this.getParent().isVisible()) {
        if (titleObj.isVisible()) titleObj.myCheck();
        if (alternateTitleObj.isVisible()) alternateTitleObj.myCheck();
        if (identifierObj.isVisible()) identifierObj.myCheck();
        if (dateObj.isVisible()) dateObj.myCheck();
        if (editionObj.isVisible()) editionObj.myCheck();
        if (editionDateObj.isVisible()) editionDateObj.myCheck();
        ...
    }
}

```

- Add the corresponding paths in the function `setFormName` of the class

```

@Override
public void setFormName(String name) {
    super.setFormName(name);
    titleObj.setFormName(name + ".title[1].characterstring[1]");
    alternateTitleObj.setFormName(name + ".alternatetitle[1].characterstring[1]");
    identifierObj.setFormName(name + ".identifier[1].rs_identifier[1]");
    identifierContainerObj.setFormName(name + ".identifier[1]");
    dateObj.setFormName(name + ".date[1].ci_date[1]");
    dateContainerObj.setFormName(name + ".date[1]");
}

```

```
        editionObj.setFormName(name + ".edition[1].characterstring[1]");  
        editionDateObj.setFormName(name + ".editiondate[1].date[1]");  
        ...  
    }
```

- If you want to differentiate the visualization of the elements (hiding, showing or something else) you can do that, in the function `setInterface` of the class `CI_Citation`

```
@Override  
public void setInterface(int i) {  
    if (EUOSMEGWT.appMode.equalsIgnoreCase(AppModes.GEOPORTAL.toString())) {  
        // setRequired: true, the element was not required in ISO  
        // but the developer wants to put it mandatory for the current  
        // application mode  
        editionObj.setRequired(true);  
        // setVisible: false, hide the element  
        editionDateObj.setVisible(false);  
        // removeDisclosure is used to remove the disclosurePanel from  
        // the element, this in case of multiplicity, to simplify the  
        // interface (too much disclosure panels, disorienting the users)  
        dateObj.removeDisclosure();  
        ...  
    }  
}
```

- Check the XML templates and add the XML code related to the element. This code will be used as a sample to create the final XML.
- Create new class `CI_Series`, following the same steps as this class implementation.

8.2 Internationalization and locale

GWT represents locale as a client property whose value can be set either using a meta tag embedded in the host page or in the query string of the host page's URL.

Adding a locale means extending the set of values of the locale client property using the `<extend-property name="locale" values=""/>` element in the module `EUOSMEGWT.gwt.xml`.

The current release supports 22 languages:

- Bulgarian (`bg`)
- Czech (`cs`)

- Danish ([da](#))
- Dutch ([nl](#))
- English ([en](#)) that is the default language
- Estonian ([et](#))
- Finnish ([fi](#))
- French ([fr](#))
- German ([de](#))
- Greek ([el](#))
- Hungarian ([hu](#))
- Italian ([it](#))
- Latvian ([lv](#))
- Lithuanian ([lt](#))
- Maltese ([mt](#))
- Polish ([pl](#))
- Portuguese ([pt](#))
- Romanian ([ro](#))
- Slovak ([sk](#))
- Slovene ([sl](#))
- Spanish ([es](#))
- Swedish ([sv](#))

To extend the application to another language the developer needs also to add the following properties file with the right extension (for the related locale):

- `\src\eu\europa\ec\jrc\euosme\gwt\client\i18n\iso19115Constants.properties`
- `\src\eu\europa\ec\jrc\euosme\gwt\client\i18n\iso19115Messages.properties`
- `\src\eu\europa\ec\jrc\euosme\gwt\client\userGuides\eurlex.html`
- `\src\eu\europa\ec\jrc\euosme\gwt\client\iso19115\ui\MainPanelmainPanelUiBinderImplGenMessages.properties`
- `\src\eu\europa\ec\jrc\euosme\gwt\client\iso19115\ui\TabGeographicTabGeographicUiBinderImplGenMessages.properties`

In addition to the module and the properties files the developer must also add the values in the code lists service (offline and online).

To update the offline version of the code list, the user could use the function `invokeUpdateRESTfulWebService` available through the server side code.

8.3 Maps

The dynamic map is implemented using the Mapstraction²⁸ JavaScript library with OpenLayers²⁹ as the provider.

The Mapstraction library has been chosen because:

- It abstracts from the map provider
- it is open-source and released under the BSD License

Out-of-the-box the library defaults the provider to Google Maps (using Google Maps API version 2) therefore in order to use it, you have to register for a Google Maps API key.

The Google map terms explicitly indicates in particular that you have to explicitly state in your Maps API Implementation's terms of use that, by using your Maps API Implementation, your users are agreeing to be bound by Google's Terms of Use.

Since Google Terms Of Service are quite restrictive - mainly with reference to the clause that "your Maps API Implementation must not...operate only behind a firewall or only on an internal network (except during the development and testing phase)" - we have chosen the Mapstraction library which abstracts from a specific map implementation making it very easy to change the map provider.

However, if you want to use Google Maps, it is easy to switch back to the default by following these steps:

- Set the parameter `apiMapstraction` to "google" in the file `web.xml` into `\war\WEB-INF\` folder.
- Note that in order to use the Maps API, you need to apply for a Google Maps API key. Running with no key specified will work with `localhost` for development purposes, but you will need to apply for your own key to deploy to a website.
- If you are not working in a local host, you have to add the API key before loading the API so, before the line:

```
| AjaxLoader.loadApi("maps", "2", new Runnable() {});
```
- Put this line:

```
| AjaxLoader.init("Type your API key here");
```

ⁱ Google, Google Web Toolkit version 2.1.0, at <http://code.google.com/intl/it-IT/webtoolkit/overview.html> [accessed at December 2010]

ⁱⁱ Technical Committee ISO/TC 211, *Geographic information/Geomatics*, 2003-05-08, ISO 19115:2003, Geographic information – Metadata and Technical Committee ISO/TC 211, *Geographic information/Geomatics*, 2006-07-01, ISO 19115:2003/Cor.1:2006(E), International Standard ISO 19115:2003 Technical corrigendum 1

ⁱⁱⁱ Technical Committee ISO/TC 211, *Geographic information/Geomatics*, 2005-02-10, ISO 19119:2005, Geographic information – Services

²⁸ Mapstraction, JavaScript mapping abstraction library version 2, at <http://www.mapstraction.com/> [accessed at December 2010]

²⁹ OpenLayers, pure JavaScript library for displaying map data version 2.10, at <http://openlayers.org/> [accessed at December 2010]