

INTEGRATED HIERARCHICAL METADATA PROPOSAL: SERIES, LAYER, ENTITY AND ATTRIBUTE METADATA

Alaitz Zabala¹ and Joan Masó²

¹Departament de Geografia, Universitat Autònoma de Barcelona, UAB. a.zabala@miramon.uab.es

²Centre de Recerca Ecològica i Aplicacions Forestals, CREAM. joan.maso@uab.es

ABSTRACT

ISO 19115:2003 Metadata standard allows generating a set of metadata for different hierarchical levels. There are still few attempts to apply this metadata standard to every level. People consider that series and layer metadata are solved correctly using this approach but when the model is tested to feature or attributes type levels it generates a huge degree of redundancy. It is neither clear how the definition of the feature and attribute types has to be done.

A proposal defining father-and-son relationships between series and layer metadata is presented. Layer elements can inherit, modify or extent the series value. Definition of entity and attribute metadata is carried out using another approach based on the pre-standard ISO 19109. The proposal suggests general rules for application schema applied to a vector layer and a set of objects (with necessary metadata) that can be used to describe a relational database of thematic attributes.

1. INTRODUCTION: APPROACHES

Metadata are the bases for the exchange, cataloguing and geospatial information searching. ISO 19115:2003 Metadata standard [1] defines which information is part of geographical information metadata. The aim of the paper is to describe, using the existing standards (approved or pre-standards) the metadata for a vector layer (or a geodatabase) associated to a relational database containing thematic attributes of the entities (see Fig. 1). The layer can be included in a dataset series (spatial data that share similar characteristics of theme, source data, resolution an methodology).

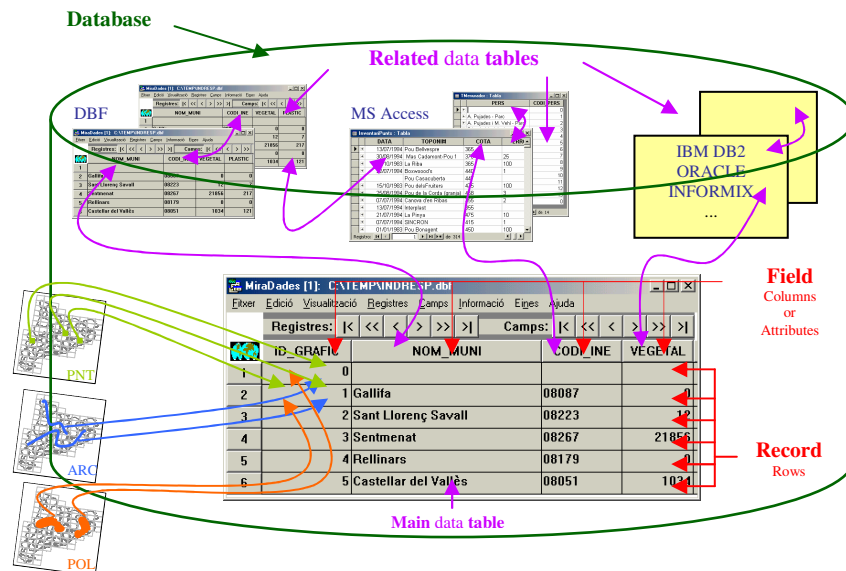


Fig. 1. Graphical representation of the data model of a vector layer associated to one relational database containing thematic attributes and that is included in a dataset series

1.1. ISO 19115:2003

ISO 19115:2003 Metadata standard [1] is a conceptual model that allows to describe geographical information metadata at different hierarchical level, particularly series, layers (dataset), and also types of entities (feature type), types of

attributes (attribute type), entities (feature instances) and attributes (attribute instances). This approach drives to generate a complete metadata set for each series, layer, feature type, feature instance or even attributes to describe.

The same standard suggests another approach to describe entity and attribute metadata (not possible for series or dataset metadata hierarchical level). This second approach provides a mechanism to reference a feature catalogue (that describes the entities and attributes) and some other metadata like the language of the catalogue or the list of entities of the catalogue represented in the layer (optional). It can also be indicated whether this catalogue of entities is compliant to standard ISO 19110 Methodology for cataloguing feature [2] (recommended). We will show you that this approach is not entirely satisfactory for describing our model

1.2. ISO/FDIS 19109 and ISO 19110:2005

The standard ISO 19115:2003 does not define how the description of the entities and attributes (and its relationships) of the layer (dataset) has to be implemented. Although some other standards for geographic metadata included the definition of entities and attributes, ISO considers this aspect to be treated in other standard of the 19100 series, not into the metadata standard.

General description of the entities and their attributes is carried out in the Final Draft International Standard *ISO 19109 Rules for application schema* [3], which includes the General Feature Model (GFM). This pre-standard defines the rules to create and describe an application schema and includes the principles for definition of entities and attributes (see Fig. 2). Application schemas provide the formal description of the structure of the data and of the contents required by one or more applications.

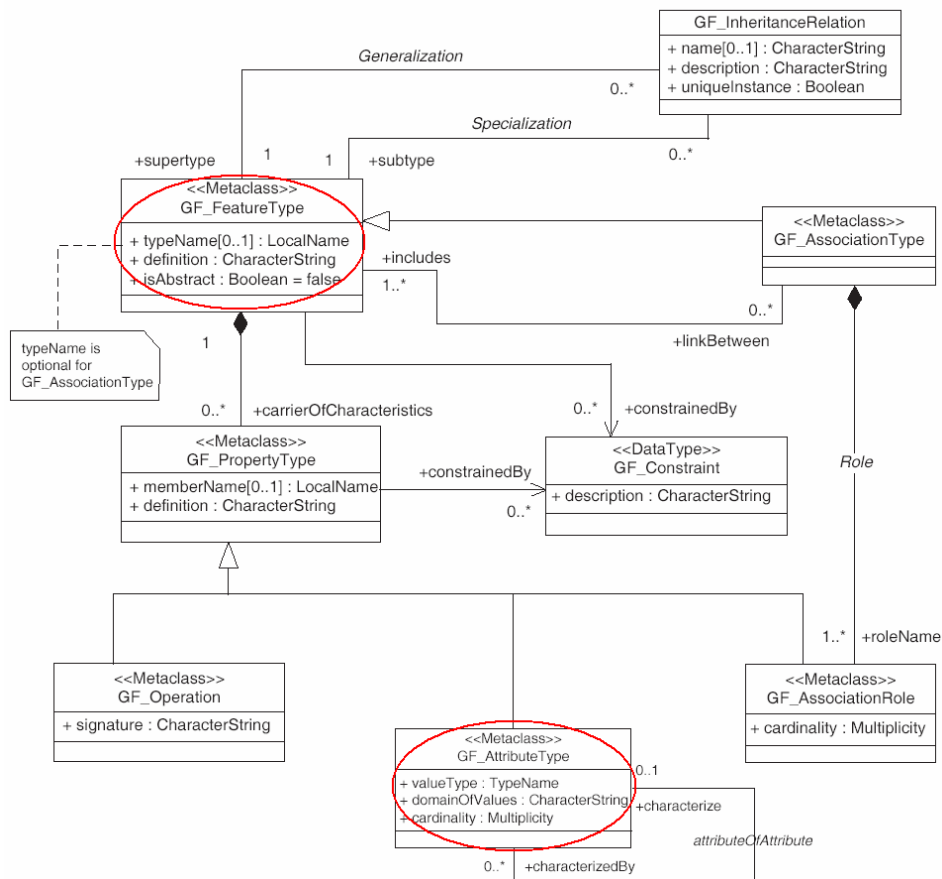


Fig. 2. Fragment of the General Feature Model (ISO 19109) (Source: UML diagrams of the standard ISO 19109)

The standard *ISO 19110:2005 Methodology for feature cataloguing* [2] (see Fig. 3) defines the methodology for cataloguing of types of entities (feature types) and is an implementation of the GFM (19110 uses concepts that are realizations of GFM).

Although ISO 19110 is inspired in the principles of ISO 19109, there are some aspects that have been simplified. There are differences between the general model described by ISO 19109 and the model implemented in ISO 19110. Particularly the definition of attributes of attributes, which appears in the general model, is not included in the model of

the feature catalogue. This characteristic is important to fully describe the thematic attributes of a vector layer contained in a relational database. Also 19110 do not allow to describe geometrical properties for features but ISO 19109 do include them through a `GF_SpatialAttributeType` and its child `GM_Object` (from 19107).

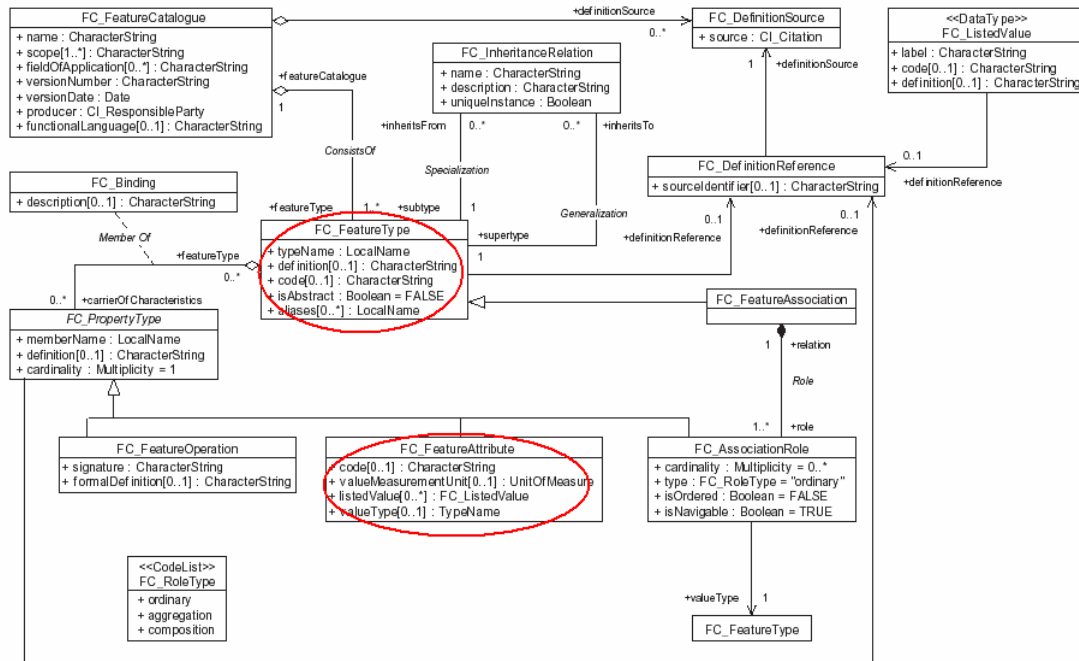


Fig. 3. Conceptual model of Feature Catalogue (ISO 19110) (Source: standard ISO 19110)

1.3. CSDGM – FGDC

The North American *Content Standard for Digital Geospatial Metadata* (CSDGM, developed by FGDC [4] before the ISO 19100 series of standards) considers in section 5 (Entity and Attribute Information) how to describe entities and their attributes.

The element ‘attribute of attribute value’ of the standard allows defining additional information for an entity with a determined value in an attribute. For example an entity ‘well’ may have an attribute ‘product’. If the value of the attribute ‘product’ is ‘water’, then this attribute can have other attributes that determine the characteristics of the water (pH, salinity, etc.).

This approach is nearer to the problem considered here than the approach of ISO 19115:2003, but anyway it does not permit a complete description of tables, fields and relationships between tables of the relational model. Particularly it does not have any concept equivalent to ‘tables’.

1.4. FGDC to ISO Metadata Crosswalk

The document *FGDC to ISO Metadata Crosswalk (v.4.0)* [5] suggests equivalences among the elements of the standard CSDGM of FGDC and those of ISO 19115:2003 (and other standards, related to ISO-TC211). This crosswalk links the elements of the section 5. Entity and Attribute Information (of CSDGM) with the elements of feature catalogue (ISO 19110).

This document does not indicate to which element of the ISO 19100 series of standards the element ‘attribute of attribute value’ of the North American standard must be related. This omission may be due to the fact the crosswalk uses 19110 to describe section 5 of CSDGM, and 19110 does not consider the element ‘attributeOfAttribute’ of the model GFM (ISO 19109).

1.5. ISO/DTS 19139

The Draft Technical Specification *ISO 19139 Metadata – XML Implementation* [6] is a technical specification that is not approved yet although it seems to have the support of most of ISO member countries (according to a recent vote,

October 2004). It intends to develop an XML implementation of the metadata model described by ISO 19115:2003 [5] and, as such, a set of XML declaring XML elements to describe layer metadata (xsd files).

However, ISO 19139 incorporates elements that were not present in the abstract metadata model ISO 19115:2003. Relating to the discussed subject it is important to stand out that ISO 19139 incorporates directly descriptions of entities (featureType) and attributes (attributeType) based on the GFM (ISO 19109). These elements are introduced as MD_Metadata compound element child.

Following this proposal, for each set of metadata describing a layer it is possible to describe the entities and attributes used in this layer. Besides incorporates in a natural way the concept, based on ISO 19109, which allows to describe attributes of attributes, essential aspect that allows us in the present paper to carry out a proposal to fully describe the tables, fields and relationships of the relational database.

The present paper describes an implementation that includes an integrated hierarchical metadata proposal: series, layer, entity and attribute metadata. Two different approaches are used for different hierarchical levels.

1.6. ISO/CD 19136

The Committee Draft *ISO 19136 Geography Markup Language* [7] is data model to generate GML application schemas. The standard explains the rules to define those application schemas and also the necessary objects to define geographic entities directly related to the feature. Every feature can have a set of geometric and non geometric properties, expressed in XML objects.

This standard has been previously developed by Open Geospatial Consortium (OGC) and it is in its third version. Version 3.2 it is pending to be approved by ISO as the 19136 IS. The standard incorporates the possibility to define metadata at the feature collection, feature instance or attribute instance level.

You can to define series and dataset metadata and you must describe which feature and attribute types are included in the dataset, and, if necessary, feature and attribute type metadata.

2. PROPOSED MODEL FOR HIERARCHICAL METADATA

The aim is to define a complete metadata model in all hierarchical levels of metadata from series to feature type and attribute type (we exclude not feature and attribute instances). The attribute types that we want to describe are those attributes characterizing a feature and included in a relational database (see Fig. 1).

Datasets we are trying to describe are vectors datasets with only one feature type and which are parts of a dataset series due to the necessity of splitting the space to avoid too extensive files.

2.1. Series and Dataset level metadata

ISO 19115:2003 Metadata standard defines which information is part of metadata and their relationships and dependences, allowing the definition of metadata for different hierarchical levels like layer, series of layers, feature type, feature, attribute type or even attribute level. At first sight, this approach would imply generating a complete set of metadata for each series, layer, feature type or attribute type to be described.

Annex G of ISO 19115 standard discusses metadata implementation. According to hierarchy, it exposes that the definition of general metadata can be supplemented by spatially specific metadata that, when required, either inherits or overrides the general case (G.1.3). This situation is also represented in the Annex H of the same document, where an example of metadata hierarchical levels is presented. This annex states that only metadata exceptions are defined at lower levels, so it is not necessary to generate the full registry of metadata for each level but to link particular spatial values to the general value that they inherit.

Conceptually the metadata registry is complete for each metadata hierarchical level, but at the *implementation* level most of the metadata elements are not present at both levels but only at the general one.

Layer metadata elements inherit the value of series metadata which is part of. It is necessary to allow layer metadata to set a particular value, superseding the value of the series or to add it to the series one. Our proposal defines which metadata has to support series-layer inheritance and in which way.

In this proposal, metadata elements are classified according to the type of inheritance between the series and the datasets. This paper explains metadata elements classification and exemplifies it using core metadata elements.

2.1.1. Metadata elements with different values for series and dataset level

Some metadata elements have sense at two levels of metadata because they are documenting different aspects. These elements can be grouped again into two subgroups regarding the necessity of dataset value being controlled.

Controlled elements: these elements may have different content for series and dataset level, but the general (series) level content restrict the dataset level content. Examples:

- *Metadata date stamp (MD_Metadata > dateStamp)*: the date that the metadata was created. This element has a mandatory element 'century' and three optional elements: year, month and day. Thus, a date may be more or less 'defined', meaning that the temporal extent referred by the 'dateStamp' element content may be only the century or until day. Dataset metadata content for this element must be included in the series metadata content. For example if series dateStamp is century: 20th, year: 96, dataset content can not be century: 20th, year: 95. A possible value for dataset metadata in the same example would be century: 20th, year: 96 (there is no ore definition) or century: 20th, year: 96, month: January.
- *Dataset reference date (MD_Metadata > MD_DataIdentification.citation > CI_Citation.date)*: reference date for the dataset or for the series. The restriction for dataset value is the same than the explained for *Metadata date stamp* element: dataset reference date must be 'included' in series reference date.
- *Geographic location of the dataset by four coordinates (MD_Metadata > MD_DataIdentification.extent > EX_Extent > EX_GeographicExtent > EX_GeographicBoundingBox)*: Geographic location for the series must contain all geographic datasets location.
- *Geographic location of the dataset by geographic identifier (MD_Metadata > MD_DataIdentification.extent > EX_Extent > EX_GeographicExtent > EX_GeographicDescription)*: Geographic location for the series must contain all geographic datasets location. It can be controlled if *geographic description* is in a gazetteer that can control relation (inclusion for example) between different identifiers.

Not controlled elements: these elements may have different content for series and dataset level, but the general (series) level content do not restrict the dataset level content. Examples:

- *Metadata file identifier (MD_Metadata.fileIdentifier)*: is a unique identifier for each metadata file. Dataset metadata links to series metadata using the file identifier of the series. Both have to be defined and different.
- *Dataset title (MD_Metadata > MD_DataIdentification.citation > CI_Citation.title)*: Dataset and series dataset title are different and usually complementary because the series title contains the general title and the dataset title contains the spatially particular content of the title, for example the name of a representative place (if the dataset is a tile). An application should merge series and dataset metadata to present them to the user, so the user can see both titles at the same time, because they bring different information.

2.1.2. Metadata elements with series metadata content that is mandatory but extendable

Dataset content for those metadata elements is always the same than the series content (which becomes 'mandatory'). If necessary, dataset content for these metadata elements can be extended with more information.

- *Abstract describing the dataset (MD_Metadata > MD_DataIdentification.abstract)*: The dataset series may have some explanation for the abstract. Each dataset related to this series inherit the value from the series and can add some extra information to the abstract. An application should merge series and dataset metadata to present them to the user, so the user can see the combination of both abstracts.
- *Metadata point of contact (MD_Metadata.contact > CI_ResponsibleParty)*: Organizations related with metadata. General organizations related with all the datasets of the series are related in the series level, and each dataset may define additional organizations. An application should merge series and dataset metadata to present them to the user, so the user can see all the organizations related to metadata at the same time.
- *Lineage (MD_Metadata.DQ_DataQuality.lineage > LI_Lineage)*:
 - *Lineage statement (statement)*: Abstract explanation applies here.
 - *Source description (source > LI_Source.description)*: Abstract explanation applies here.

- *Process step (processStep > LI_ProcessStep)*: some process steps may be defined at series level. They are inherited by each layer, which can define other process steps defining its production process. An application can show processes to the user sorted by date (*dateTime*), so dataset processes may be mixed with series process at dataset level where the user can see all of them and can modify only those defined for this particular dataset. Processes with no date-time defined are sorted at the beginning of the list.

2.1.3. Metadata elements with series metadata content that is mandatory if defined

Series content for those metadata is mandatory and cannot be specified or extended by dataset metadata. Only when all layers have the same value, the series one is set. If the series does not specify the content it means that there is no common value so the dataset can specify the content.

- *Reference System (MD_Metadata > MD_ReferenceSystem)*
- *Spatial resolution of the dataset (MD_Metadata > MD_DataIdentification.spatialResolution > MD_Resolution.equivalentScale or MD_Resolution.distance)*

2.1.4. Metadata elements with series metadata content that is mandatory but may be modified

Series content for those metadata is mandatory but can be modified by dataset metadata. It means that if the series level contains a value, dataset must contain also a value, which can be the series values or other one.

- *Report (MD_Metadata.DQ_DataQuality.report > DQ_Element)*: series metadata may define a general value, for example the mean of the error of all the datasets included in the series. For those dataset that a specific value is not defined the general value applies. For datasets with a specific value, the general value is 'overwritten'.

2.2. Entity and Attribute Type Description

In order to describe Entity and Attribute metadata, it is necessary to describe the entities and its attributes first. ISO gives at least two approximations. ISO 19110 Methodology for feature cataloguing [2] gives us a way to define features and attributes but has neither support to describe geometrical attributes nor to define attributes of an attribute. On the other hand, ISO 19109 Rules for Application Schema [3] gives us a General Feature Model that defines a set of abstract objects that can be extended to fulfil our aim.

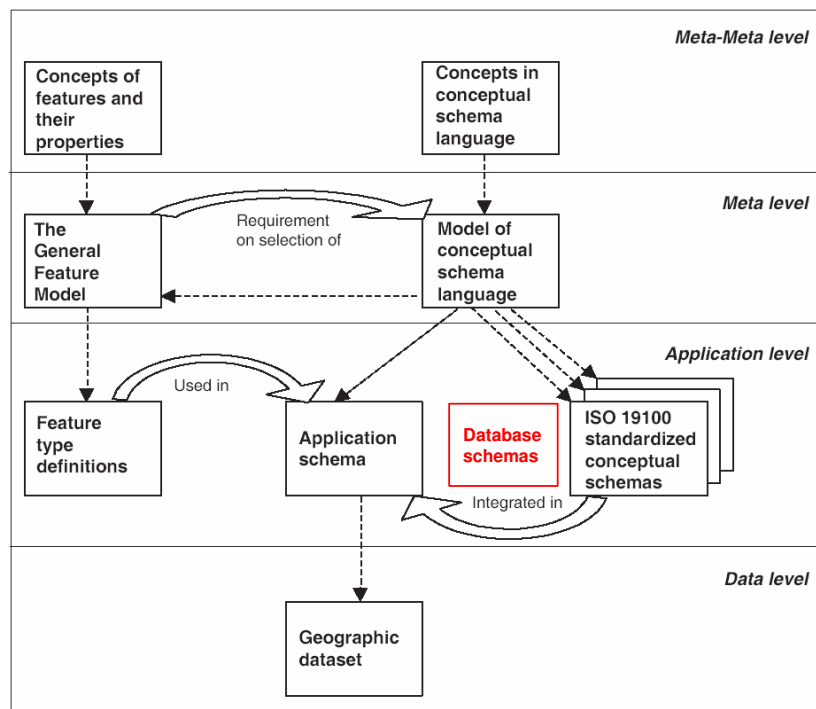


Fig. 4. The 4-layer architecture (Source: standard ISO 19109 extended to consider thematic database schemas)

In the General Feature Model it is possible to define GF_FeatureType's. Every feature type could have properties (GF_PropertyType) that can be of different kind, and particularly attributes (GF_AttributeType). Attributes can be of different kinds and particularly thematic attributes (GF_ThematicAttributeType).

The General Feature Model is not at the *Application Level* but in the *Meta Level* (see Fig. 4 and Annex B of ISO 19109 for more information about *The 4-layer architecture*). It means that the elements of the GFM are *Metaclasses* and they are not directly usable in an Application Schema. They have to be used to generate *Classes* that are already in the Application Level and can be used in the application schema.

ISO 19100 standards define different standardized conceptual schemas that can be integrated in the application schema. ISO 19109 describes which dependencies are necessary to describe different kind of attributes, for example the TM_Object (from Temporal Objects) is used to describe the GF_TemporalAttributeType (see ISO 19109 section 7.4).

The standard does not describe which schema has to be used to describe thematic attributes. So it is necessary to define a conceptual schema in the application level to integrate it into the application schema to describe thematic attributes. When you try to apply relational database concepts to an application schema you realize that there is a lack of reusable objects at the application level for some of these concepts (like tables) neither in SQL ISO-9075:1999 nor in vendor products [9]. This paper defines some new data types (a new schema) based on database structure to define thematic attributes of features from GFM (see “Database schemas” in Fig. 4). We are going to define two new data types that will be used as thematic attributes: DB_RowSet and DB_Field.

2.2.1. DB_RowSet

In our model thematic attributes are stored in fields of database tables that are combined forming a relational database [10]. Every feature is directly related with some rows of a database table that is called “main table”. Similarly, ISO 19125-2 explains that a feature table’s columns represent feature attributes, while rows represent individual features [8]. Any table could be seen as a set of rows like SQL language does. It means that one group of thematic attributes of a particular feature will be derived as a DB_RowSet type attribute.

The proposed DB_RowSet data type is a complex data type with several child elements. Those are (see Fig. 5):

- *Name*: the name of the table. It has to be unique within the application schema. It is the reference to find the rowset containing the attributes, so it may be, for example, the path and filename of a DBF table.
- *Description*: it is a description about the content of this table or rowset.
- *Type*: the type of rowset. For example DBF table, SQL query, ODBC table, etc. It is implemented as a code list with predefined values (DB_RowSetTypeCode).
- *Relation type*: the type of relation between this rowset and the previous element of which it is a child (a feature type or another attribute). It is implemented as a code list (DB_RelationTypeCode). Possible values are shown in Fig. 5. The difference between the normal values and those relating to a thesaurus is that the thesaurus has to be complete, that means that every value in the previous table is related to rows in this table.
- *Record number*: number of records in the rowset.
- *Field number*: number of fields forming the rowset.
- *Character set*: it is the MD_CharacterSetCode code list.

Every rowset is related to ‘field number’ fields (in a way that we are going to explain next).

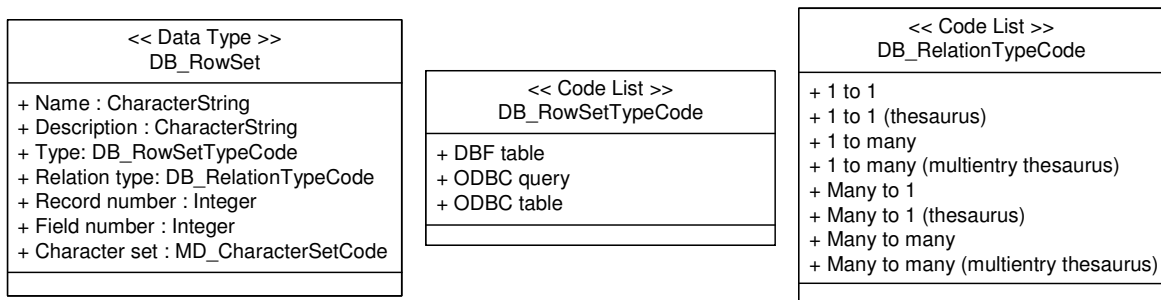


Fig. 5. The DB_RowSet data type

2.2.2. DB_Field

The proposed rowset has many fields (or columns). The DB_Field data type is a complex data type with several child elements. Those are (see Fig. 6):

- *Name*: the name of the field. It has to be unique within the rowset.

- *Description*: it is a description about the content of this field.
- *Type*: it describes which the field content type is, for example character, number, data, etc. It is implemented as a code list with predefined values (DB_FieldTypeCode).
- *Size*: this is the size (in bytes) of the field.
- *Precision*: number of significant figures used to save floating point numbers.
- *Treatment*: code list that can be set to ordinal, quantitative continuous or categorical content. The treatment of a field is usually related to the field type, for example character fields usually have categorical content. But sometimes may be interesting to redefine it, for example when a character string field contains ordinal or even numerical codes. It is implemented as a code list (DB_FieldTreatmentCode).
- *Units*: When the field is treated as a quantitative continuous value, it defines the units of the field.
- *Quality*: some reports of the quality of field content may be included. It is DQ_Element type element.
- *Is geometric or topologic*: some fields are marked because they contain geometric or topologic information, for example the area of each polygon, the longitude of an arc or the arcs in a node. It is implemented as a code list (DB_GeoTopoCode).
- *Is related*: It is a Boolean attribute explained in the following section.

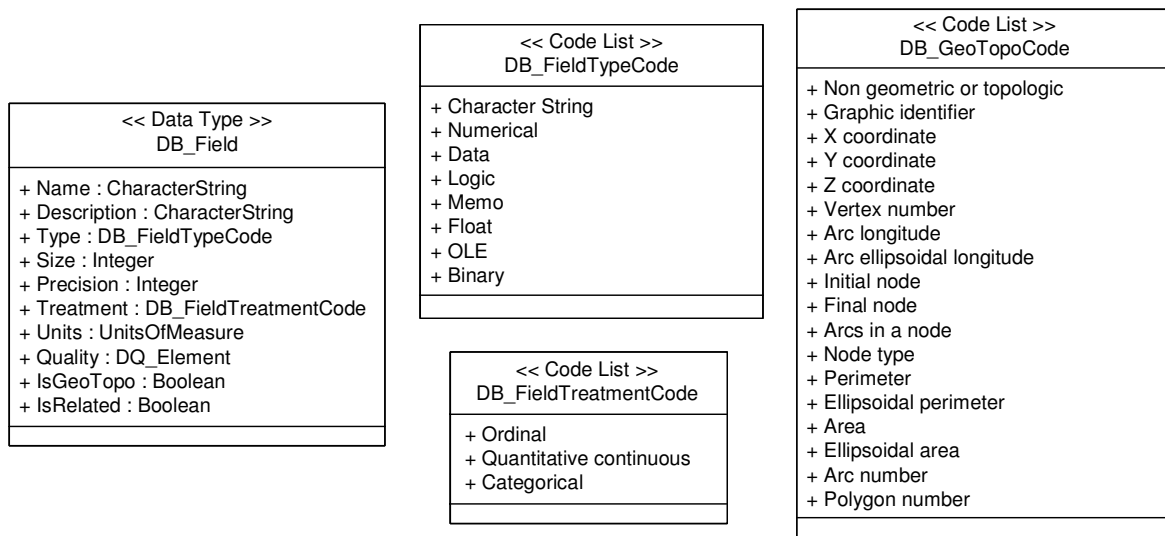


Fig. 6. The DB_Field data type

2.2.3. Relations between DB_RowSet and DB_Field

Each rowset is formed by ‘field number’ of fields and some of these fields may be related with other tables (see Fig. 7). This relation can be defined in a simple but common way by linking the values of one field in the source table with the values of one field in the target table. The target table is represented by another rowset, having its own field types. *Is Related* has been added as a new attribute to the DB_Field type to indicate when this is the related field (usually only once in a table) between the current rowset and the previous parent element (a feature type or another DB_Field attribute).

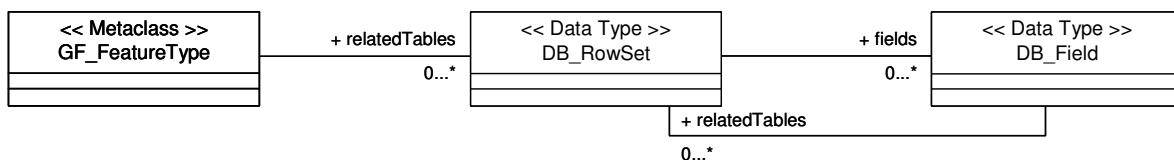


Fig. 7. Relations among Feature Types, DB_RowSets and DB_Fields

2.3. Entity and Attribute Type Level Metadata

An approach based on hierarchical levels may be also used for features and attributes, generating a schema similar to that explained for series and datasets. It exist some agreement in the scientific community that this approach is to complex and unnecessary. We also think that a whole set of metadata for features and attributes is not required other than metadata included in DB_RowSet and DB_Field data types.

In the description of features and attributes we have already defined some metadata elements. For example the quality of an attribute or the character set of the rowset may be defined. The model could be extended to include some other

metadata elements for features and attributes if necessary. The rest of the undefined elements are assumed to be automatically inherited from the metadata layer ones.

3. FINAL CONSIDERATIONS

In case of relations between tables using a combined key field it is not possible to relate the destination rowset to one of the key fields in the origin of the relation (to which of them?). For this reason it will be necessary to define a new attribute type which relates to rowsets and indicates which fields of each rowset are linked. In case that a table is related from more than one field, it is described only once and a n identifier is assigned to it. This identifier is used later to indicate the other links to the same table to avoid redundancies and circularity.

A new implementation that should be considered is the multilingual approach to content of free text metadata elements. ISO 19115 proposes a structure to handle multilingual metadata. Everywhere on ISO 19115 where 'free text' is used, the class PT_FreeText can be used. In the same way, in the character string type elements in the database schema (DB_RowSet and DB_Field elements) the class PT_FreeText may be used to support multiple instances of information in different languages.

4. CONCLUSIONS

A general description of multilevel metadata for a vector layer related to a relational database of thematic attributes is possible applying metadata ISO19115 concepts for series and layer levels. ISO 19109 concepts have to be included to define new reusable objects to describe attribute types present on the layer. These objects define tables, fields and relations between them. Some metadata elements could be added to the description of these objects. This approach has to rest on a precise inheritance roles to avoid redundancy. An application can use this rule to show the user the whole set of metadata for each level and object.

5. REFERENCES

- [1]. International Organization for Standardization: International Standard: Geographic information – Metadata. ISO 19115:2003. Technical Committee 211 (2003)
- [2]. International Organization for Standardization: Draft International Standard: Geographic information – Methodology for feature cataloguing. ISO 19110:2005. Technical Committee 211 (2005)
- [3]. International Organization for Standardization: Draft International Standard: Geographic information – Rules for application schema. ISO/DIS 19109. Technical Committee 211 (2002)
- [4]. Federal Geographic Data Committee: Content Standard for Digital Geospatial Metadata. CSDGM Version 2: FGDC-STD-001-1998. Washington (1998)
- [5]. Federal Geographic Data Committee: ISO-FGDC-METADATA-CROSSWALK-V4 (2003)
- [6]. International Organization for Standardization: Draft Technical Specification: Geographic information – Metadata – XML schema implementation. ISO/PTDS 19139. Technical Committee 211 (2004)
- [7]. International Organization for Standardization: Committee Draft: Geography Markup Language (GML). ISO/CD 19136. Technical Committee 211 (2005)
- [8]. International Organization for Standardization: International Standard: 2004 Geographic information - Simple feature access - Part 2: SQL option. ISO 19125-2:2004. Technical Committee 211 (2004)
- [9]. Mahnke, W. Generic Components in Object-Relational Database Systems. In: Proc. Int. Colloquium "Software Reuse - Requirements, Technologies and Applications" (Sonderforschungsbereich 501, Kaiserslautern), (2003)
- [10] ISO-9075 SQL Part 1: SQL/Framework (1994)

BIOGRAPHY OF THE PRESENTING AUTHOR

Alaitz Zabala

M.S. degrees in Remote Sensing Applications and GIS from the Institute for Space Studies of Catalonia at Barcelona in 2001.

She was graduated in Environmental Sciences in 2002 by the Autonomous University of Barcelona. Her final year degree project “Generation and Management of metadata for environmental cartographic datasets” introduced her in the field of research of the geographical information metadata.

She has been developing metadata tools for documenting metadata in the MiraMon Project since 2001 and she is co-author of the MiraMon metadata model.

She participated in the definition of the strategy to migrate the IDEC (Catalan SDI) metadata tools to the new ISO-19139 proposals.

She has been working in the Department of Geography from the Autonomous University of Barcelona since 2001, and she currently is a Research Assistant in the same department. Now she is carrying out the studies to obtain Ph.D. degree on geography and focusing her research in the effects of the lossy compression techniques in the digital image classification.