# A Framework for Classifying Scientific Metadata

Helena Galhardas*, Eric Simon and Anthony Tomasic
INRIA
Domaine de Voluceau - Rocquencourt 78153 Le Chesnay France
email: First-Name.Last-Name@inria.fr



### Abstract

The scientific community, public organizations and administrations have generated a large amount of data concerning the environment. There is a need to allow sharing and exchange of this type of information by various kinds of users including scientists, decision-makers and public authorities. Metadata arises as the solution to support these requirements. We present a formal framework for classification of metadata that will give a uniform definition of what metadata is, how it can be used and where it must be used. This framework also provides a procedure for classifying elements of existing metadata standards.

## 1 Introduction

Nowadays, scientific data is considered of prime importance. Governments, organizations, the scientific community and the public in general are more and more concerned on building applications that handle scientific data. Nevertheless, scientific applications have some properties that make them difficult to be handled. The amount of scientific data that needs to be shared by scientists is very large. It is around two orders of magnitude above the size of the data involved in bank on-line transaction systems [5]. Scientific applications have a multi-disciplinary character as they cover, for example, scientific calculus, statistical data analysis, decision support, management of natural risks and support to regulations. Since a wide variety of communities is involved, scientific data comprises many different types of data. Thus, we can have: satellite images, maps, time series, descriptive text documents or management data. Scientific data is usually heterogeneous and often distributed over heterogeneous software and hardware platforms, because it is produced by many different sources. Moreover, different levels of users (scientists, decision makers or public agencies and organizations) with distinct skills access to these distributed data sets. Consequently, data is used in multiple and distinct ways. Another important characteristic of most of this data is that it is ruled by scientific laws and it can be transformed by mathematical models. Some of the data transformation methods are required to be re-used by different users in the same community or by users that belong to a different community.

Users of scientific data have to face some problems due to the above mentioned properties of scientific applications. The first task users have to accomplish is to find relevant data according to their interests. This might be difficult either because data is not properly referenced by data suppliers, is duplicated and not consistently maintained or because it is insufficient and require additional processing. Secondly, the extraction of the required information from the data sources might be a difficult process. The main reasons for such an obstacle are the heterogeneous nature of the underlying data and systems (requiring format translations), inherent administrative procedures (data might be private), and the costs associated to pre-processing data before it can be used. Once retrieved, data may be hard to be used. Due to lacking of data classification by data suppliers, data sets may be incompatible or inconsistent. Finally, the quality of extracted data may be difficult to evaluate. It is often hard to compare data produced by distinct sources and using different scientific models with no documentation about the data production process.

### 1.1 Metadata

A possible solution to eliminate or reduce the user problems previously mentioned is to publish auxiliary information in addition to the one accessible in data sets. Some examples of this additional information will follow. Searching useful data sources for answering a user question is easier if each data source publishes its location and a summary of its contents. Retrieving data from the selected sources is possible only if one is aware of

*Founded by "Instituto Superior Técnico" - Technical University of Lisbon and by a JNICT fellowship of Program PRAXIS XXI (Portugal)

its structure (e.g. schema of a database). The user must be able to choose the appropriate scientific model to apply to the retrieved data to produce new data. So, information about the interfaces of available transformation models and its uses must also be publicly available. A scientist may want to repeat the same experiment several times and record its results. Thus, historical information must be stored and be available for interpretation. Finally, scientists also need to decide whether data has enough quality to be used. Information about data accuracy and the methods used to produce it will support that decision.

This auxiliary information is usually called **metadata**. Although there is an intuitive explanation of what metadata is - *structured data about data* - there is no clear definition that describes what the notion of metadata encompasses. Many people define metadata with different meanings. In the database domain, metadata is all the information that is stored in the database dictionary. In Digital Libraries, metadata provides information about a data source content in order to support the efficient search and retrieval of documents while releasing users from being aware of the entire data source content. Within the community of environmental scientists, reference [4] claims that, besides its usefulness in the process of data source discovery, metadata is also essential to the effective use of discovered data sources, by providing the mechanisms for interoperability across protocol domains. For the rest of this paper, we use environmental scientist community as a prototypical community for scientific data management. Even within the environmental community, there is not a common definition and different metadata formats emerged from distinct disciplines. From an analysis of the existing standards, one concludes that there is not and there will never exist a common metadata format. This is mainly due to the heterogeneous nature of environmental communities. Nevertheless, a *common denominator metadata format* as UDK model [4] or CEO metadata guidelines [3] exists. UDK model proposes a class hierarchy of metadata based on environmental data types, and CEO guidelines supply a minimum set of metadata to be used by several environmental communities. In parallel, a qualitative classification of the metadata used during the environmental data production process, namely to support data location and evaluation, can be found in [2].

Despite these efforts, there still does not exist a way to formally explain to data producers and users **what** metadata is, **how** can it be used to solve user problems and **when** it should be used during the environmental data production and management process. To address these three points, we propose a framework for the formal classification of metadata. This framework should be described orthogonally to the data types involved. In addition, this formal definition should provide an unambiguous method for comparing and evaluating metadata standards. By accessing to such a classification of existing metadata standards, users and producers will then be sure of using the more appropriate ones.

### 1.2 Contributions

The main contributions of this paper are:

- Definition of a framework for classification of metadata
- Application of this framework to the production of environmental data
- Definition of a procedure for classifying elements of a metadata standard.
- Application of the framework and procedure against one existing metadata standard and one metadata model.

The next section will describe formally the framework for classification of metadata. Section 3 explains what types of metadata, from the framework, are used through the environmental data production and management. The fourth section introduces a methodology for classifying metadata elements from metadata standards. This methodology is exemplified for a metadata format (the US Federal Geographic Data Committee [6]) and a metadata model (UDK model [4]).

## 2 Framework for classification of metadata

Our classification of metadata is based on first-order logic. We use as example a small extract of the domain of oceanography based in [7] where we restrict ourselves to relational data sets. The base relations are *wind-data(id, speed, direction)* and *wave-data(id, height, direction, period)*. In our knowledge domain, information about the sea conditions (*wave-data*) is obtained through the application of a scientific model (called *predicted-wave*) to the wind speed and direction.

We briefly introduce the basic notions of a first-order logic [1] that will be used. A *first-order logic* is composed of a *theory* and an *interpretation.* A *theory* consists of an alphabet, a first-order language $L$, a set of axioms and a set of inference rules. The *alphabet* is composed of: variables $(x, y, z, \ldots)$, constants $(a, b, c, \ldots)$, function symbols $(f, g, \ldots)$, predicates $(p, q, \ldots)$, connectives ($\vee, \wedge$ and $\neg$), quantifiers ($\exists, \forall$) and punctuation symbols ( ",", "(" and ")"). A *well-formed formula* is a collection of symbols from the theory that behave according to the rules of a first-order logic. A *fact* is a well-formed formula applied to constants of the alphabet. The first-order language $L$, given by an alphabet, is composed by the set of all well-formed formulas built from the symbols of the alphabet. $L$ is concerned with the syntactic or structural aspects of the corresponding first-order logic. An *interpretation* gives a meaning to constants, function symbols and predicate symbols. It is a domain of discourse composed by elements that are assigned to constants; variables range over them; function symbols are mapped into them; and predicates are assigned to relations between them. Interesting interpretations are those for which a well-formed formula expresses a true statement.

The main idea underlying the classification of metadata using first-order logic is to write well-formed formulas and give them an interpretation.

During the modeling of metadata, we will need to introduce new first-order logics. Before their introduction, we explain the notation used and some assumptions taken. The basic first-order theory has a language that is called $L^0$. Constants in $L^0$ are symbols like *1*, *"wave information"*, and *http://www.wave.com* that are interpreted as the real data values. Thus, we use the classical definition of logical interpretation. We then construct from $L^0$ a new first-order logic (with language $L^1$) that is obtained by a restricted form of reification[1]. The interpretation of this new first-order logic states that predicates and function symbols from the $L^0$ alphabet become constants of the new first-order theory alphabet. $L^1$ may contain other new constants, however it cannot contain any constant of $L^0$. So, *1* is not a constant in $L^1$ but *Integer* will be a constant in $L^1$ indicating type information. The same reasoning is applied to $L^1$ to construct a new first-order logic $L^2$. Thus, the constants of $L^2$ are the predicate symbols of $L^1$. We avoid an infinite sequence of $L^0$, $L^1$, $L^2$... by using reification to reduce the language hierarchy to a limited number of levels.

To handle the expression of facts from multiple languages, we define the *union* of first-order languages as follows. $L^u$ is a new first-order logic that is the union of the languages of $L^0$ and $L^1$. The interpretation of $L^u$ is the union of the interpretations of $L^0$ and $L^1$. Thus, the constants of $L^u$ are the union of the constants of $L^0$ and the constants of $L^1$. Similarly, $L^{uu}$ is the union between $L^2$ and $L^u$. The set of constants of $L^{uu}$ is then composed by the constants of $L^0$, of $L^1$ and of $L^2$.

### 2.1 Modeling Data

In the subsequent sections we use the above definitions to model standards of data. Figure 1 shows the general strategy of this modeling. The horizontal plans on the right side represent data and data transformations. The column on the left hand side shows the metadata associated with each plan and the fragments of logic used to model this metadata. The theory level of data corresponds to environmental data sets. It corresponds to a first-order theory and its language $L^0$. The interpretation level represents the data that is really stored. It is the interpretation of the previous theory. We assume that the interpretation includes text documents, databases, images and files at the same level as integers and reals. The activity level represents the use of environmental data by applications. At this level, we consider that both data (basic data *B* or derived data *D*) and data transformations (*tf1* and *tf2*) can be searched and used by different users.

For example, facts in $L^0$ are:

**Example 2.1**

```
wind-data(1, 50, ''South'')
wave-data(1, 2, ''South-East'', 5)
```

[1]The formula $\forall x \forall y, y(x)$ is a well-formed formula in a second-order logic, because it quantifies over predicate names. To reify this formula, predicate names are mapped into constants and the formula is rewritten as $\forall x, \forall y, p(y, x)$.

Thus, in language $L^0$, we represent data. These facts match the base relations *wind-data(id, speed, direction)* and *wave-data(id, height, direction, period)*.

### 2.2 Modeling Metadata

Now, we want to model the representation of the data. This means we want to represent some information about the schema of a database. (In such a case, we will be talking about metadata that is localized on the left side rectangle of Figure 1.) We define a first-order logic called $L^1$. Following on our example, facts in $L^1$ are:

**Example 2.2**

```
predicates(wind-data)
attributes(wind-data, id, Integer)
attributes(wind-data, speed, Integer)
attributes(wind-data, direction, String)
predicates(wave-data)
attributes(wave-data, id, Integer)
attributes(wave-data, height, Integer)
attributes(wave-data, direction, String)
attributes(wave-data, period, Integer).
```

Through the predicates *predicates* and *attributes* in $L^0$, we are modeling the structure of the data which corresponds to the definition of part of the meta-schema.

The notion of index can also be represented as predicates in $L^1$. Indexing *wave-data* by direction, for instance, corresponds to facts in $L^0$ of the form:

**Example 2.3**

```
direction-wave-data(''South'', 209),
```

where "209" is the identification of a *wave-data* record that has direction "South". In contrast to other definitions of index (that consider it as metadata), by defining an index we are just describing one more data property. But, if we want to add some knowledge about its structure, then again we have to move to $L^1$ and add the following facts:

**Example 2.4**

```
predicates(direction-wave-data)
attributes(direction-wave-data, direction, String)
attributes(direction-wave-data, waveid, Integer)
indexes(direction-wave-data,wave-data,direction)
```

These facts correspond to predicates in $L^1$. The only difference is that *predicates* and *attributes* are "unary" predicates with respect to *direction-wave-data* and *indexes* is a "binary" predicate with respect to *direction-wave-data* and *wave-data*, that are both predicates in $L^0$. The former ones represent the structure of the index and the later defines the index.

At the activity level of Figure 1, classifying data within a hierarchy makes it easier for users to search for interesting data sets. In our example, let us assume that both data about waves and about wind are considered indicators for weather forecasting. This can be modeled as facts in $L^1$:

**Example 2.5**

```
indicators(wave-data)
indicators(wind-data).
```

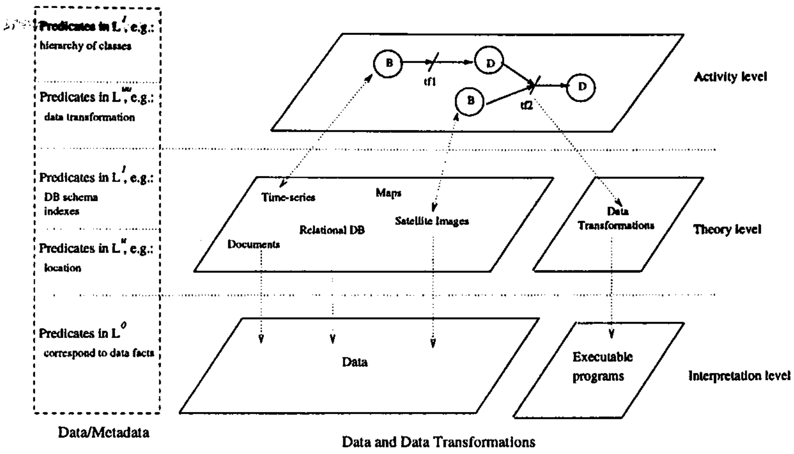


Figure 1: Data and Metadata plans

But, we may also say that *weather-forecasting* is a more generic class of data that encloses *indicators*. This would imply to define the predicate:

**Example 2.6**

```
weather-forecasting(indicators)
```

that is a predicate in $L^2$, because *indicators* is a predicate in $L^1$. A different way of describing the same thing and avoid predicates in $L^2$ is to reify *indicators* and *weather-forecasting* and treat them as constants in $L^1$. So, instead of Example 2.6, we use a new predicate *subtype* in $L^1$ to represent the following facts in $L^1$:

**Example 2.7**

```
subtype(weather-forecasting, indicators)
subtype(indicators, wave-data)
subtype(indicators, wind-data).
```

From the analysis of the examples, notice that there are several predicates at $L^1$ that model aspects of the same data. For instance, *wave-data* properties are modeled, in $L^1$, through the predicates: *predicates*, *attributes*, *indexes* and *subtype*. Although they are all predicates, we can distinguish two levels of predicates in $L^1$: those that are *minimum* describe the structure of data (like *predicates* and *attributes*) and the others that add some more knowledge about the properties of data (like *indexes* and *subtype*). Minimum predicates in $L^1$ represent the minimum model needed to reconstruct the lower level, $L^0$. The *additional* predicates in $L^1$ describe some semantics associated to data.

### 2.3 Mixing Data and Metadata

Another kind of metadata that we call *demi-metadata* [8] arises from the fact that we need to associate data values to $L^0$ predicates. Referring to Figure 1, we are at the theory level. If we want to state that all data about waves can be obtained from a certain URL, we will add the fact:

**Example 2.8**

```
location(wave-data, http://www.ocean.com/wave.html).
```

This fact is neither a fact in $L^0$ nor a fact in $L^1$ because it uses *wave-data*, a constant in $L^1$, and *http://www.ocean.com/wave.html*, a constant in $L^0$. To deal with this situation, we need to go to $L^u$ and *location* is a fact in $L^u$. We have demi-metadata every time we associate constant values in $L^0$ to predicates in $L^0$. Intuitively, this happens each time we want to describe properties that are independent of the data itself.

### 2.4 Modeling Data Transformations

Referring again to the activity level in Figure 1, we model the transformation of data by the signatures of the programs that accomplish these transformations. Independently of its type, data is classified as basic or derived. Basic data is transformed through data transformation activities and generates derived data. Transformation activities, in general, have input arguments, return an output and correspond to the execution of a scientific model or to some human interpretation. Each transformation activity, applied to some data (basic or derived) will produce new data called derived data. In our example, wave conditions can be calculated from wind conditions through the application of a prediction model called *predicted-wave*. This corresponds to the following fact in $L^1$:

**Example 2.9**

```
predicted-wave(wind-data, wave-data)
```

and to the fact:

**Example 2.10**

```
data-transformation(predicted-wave, wind-data,
                    wave-data).
```

*Data-transformation* models properties of a predicate (*predicted-wave*) and two constants in $L^1$. In other words, *data-transformation* uses a constant in $L^2$ and two constants in $L^1$, and thus belongs to

$L^{uu}$. The semantics of the particular data transformation *predicted-wave* is modeled in $L^1$ through its interpretation which is the executable program. $L^{uu}$ models the way the data transformation is applied. An alternative way of doing it would be to reify *predicted-wave* and to consider it as a constant in $L^1$. The predicate *data-transformation* would then be a predicate in $L^1$. We chose the first approach of being at $L^{uu}$ as it seemed a more natural representation.

### 2.5 Framework for classifying Metadata

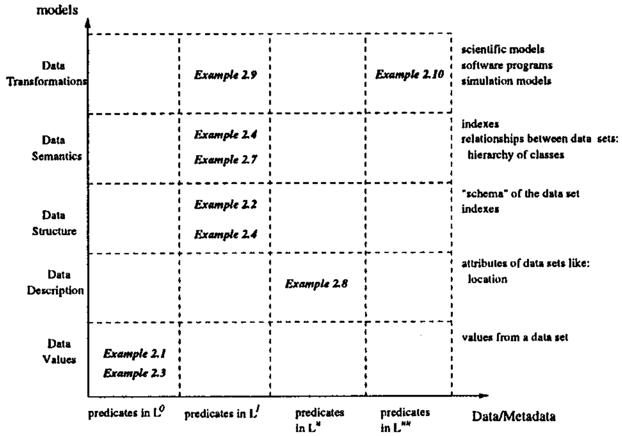


Figure 2: Framework for classifying metadata

Through the presented sequence of examples, we were able to define and characterize several formal types of metadata. Figure 2 illustrates a grid where the $x$ axis contains the formal types of metadata (*what is metadata*) and the $y$ axis represents the kinds of knowledge modeled by those types of metadata. The $y$ axis has informal examples on the right side of the figure. Each point in the grid corresponds to one or more examples given in this section. For each kind of informal data/metadata in $y$, one finds the formal way to represent it as data/metadata in the $x$ axis.

For example, the collection of indexes, i.e., the predicate *indexes* in $L^1$, provides data semantics about the relationship between data elements. This predicate is classified as data semantics on the $y$ axis and as a predicate in $L^1$ on the $x$ axis. As another example, the structure of each index is also modeled, i.e., predicates *predicates* and *attributes.* These predicates are classified as data structure on the $y$ axis and as predicates in $L^1$ on the $x$ axis. We will use the same grid-based framework to explain when the different types of metadata should be used in the environmental data production and management and to apply the classification against metadata standards.

## 3 Application of the framework to environmental data production

Figure 3 summarizes the activities that support the environmental data management and production process [8]. In this figure, environmental data produced in data sources is documented and *published* by data producers. This way, users (data consumers) *locate* and *extract* the appropriate data sources to answer their requirements. Data consumers access and use the data to eventually produce new data or to *aggregate* it at a higher level of abstraction. Users can also *analyze* the retrieved data and draw conclusions on its utility or interpret it and generate new "value-added" data. Data that is produced as a result of user interactions is also published.

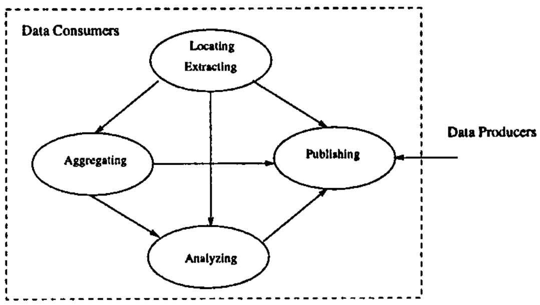


Figure 3: Environmental data production and management activities

These activities are accomplished through the use of metadata. Our purpose in this section is to analyze the types of metadata that have to be used during each of the activities: locating, extracting, aggregating and analyzing data. For this analysis, we use the procedure described in Section 2.

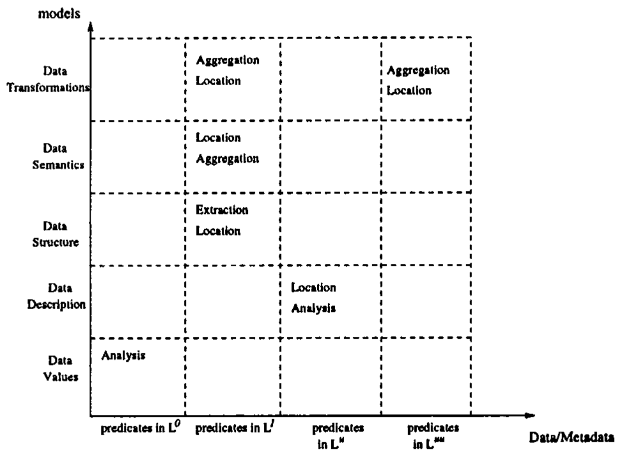


Figure 4: Use of metadata during data production and management activities

To locate a certain data set, we may use several types of data. We can use a description of the data set to locate data on a local server (for example an attribute called *location* that has an URL assigned to it). This location query is modeled by $location(x, y) \wedge localserver(y)$ which is a well formed formula in $L^u$. Thus, since the locating activity uses the *location* predicate which is a data description predicate, this activity is classified as $L^u$ on the $x$ axis and data description on the $y$ axis. Other locating activities use an appropriate index over the data set (data structure and data semantics over $L^1$) or the usual transformation that is used to obtain it (data transformations over $L^{uu}$).

To extract data, one must be aware of its structural details represented by predicates in $L^1$. The aggregation of data is done through the execution of data transformations by using predicates in $L^1$ and $L^{uu}$. Data analysis is needed in order to evaluate the utility of data. For that, one may use descriptors of data (like the description of the measurement techniques used to obtain it) that are represented as predicates in $L^u$ or samples of data that are represented as facts in $L^0$. Figure 4 summarizes this classification.

## 4 Application of the framework for classifying metadata standards

The procedure followed for classifying the elements of metadata formats and models, according to our grid-based framework, is:

Given a part of a model or standard:

- model it in a first-order logic
- classify it according to the types of metadata in the $x$ axis
- look, in the $y$ axis, for the purpose of that fragment of metadata
- fill in a given $(x, y)$ coordinate.

We will apply our framework against the UDK metadata model and the FDGC metadata format using this procedure. Firstly, we will give a brief overview of each of these approaches. Then, we will follow the classification procedure to characterize the associated metadata.

### 4.1 UDK model

UDK (*Umwelt-Datenkatalog*) [4] or Environmental Data Catalogue is a meta-information system and navigation tool that documents collections of environmental data produced by the German states and other sources. Three types of objects are distinguished in UDK: environmental objects, environmental data objects and UDK (metadata) objects. A (real-world) *environmental object* corresponds to a physical entity like a river and is described by a collection of *environmental data objects.* Each environmental data object is associated with *UDK (metadata) objects* that describe its format and contents. Environmental data objects can be organized through a hierarchy of classes with inheritance of attributes. There are seven important classes: *project data* that corresponds to environmental impact studies, *empirical data* that includes measuring series, *data about facilities* (which factories or buildings are involved), *maps, expertises and reports*, *product data* and *model data* that corresponds to simulation results. Orthogonally, UDK metadata objects and environmental data objects are linked through semantic graphs called catalogues. These catalogues describe *part-of* and *responsible of* relationships between metadata objects and environmental data objects.

The procedure followed for identifying and classifying UDK metadata is the one described above. By using it, we will be able to represent $L^0$, $L^1$, $L^u$ and $L^{uu}$ for the UDK model. The examples used here are based on reference [5].

An environmental data object describes, for instance a set of measurements made to capture the concentration of oxygen in a river. This corresponds to the fact in $L^0$:

**Example 4.1**

```
oxygen-concentration(''Thames'', 50, Jan/97)
```

where the constants stand for the river name, the oxygen concentration level and the sampling date.

UDK environmental data objects belong to classes and each class has a set of attributes. Facts correspond to instances of classes. So, *oxygen-concentration* is an environmental data class. It is represented by the following predicates in $L^1$:

**Example 4.2**

```
environmental-data-classes(oxygen-concentration)
attributes(oxygen-concentration, river-name, String)
attributes(oxygen-concentration, conct-level, Integer)
attributes(oxygen-concentration, date, Date)
```

These facts describe the structure of data.

To state that *oxygen-concentration* is a sub-class of the class *measurements* and that *measurements* is a sub-class of *empirical-data,* we can use the following facts in $L^1$:

**Example 4.3**

```
sub-class(empirical-data, measurements)
sub-class(measurements, oxygen-concentration)
```

that result from reifying *empirical-data* and *measurements* as constants in $L^1$. The *subclass* predicate describes semantics of data.

UDK metadata objects add information about environmental data objects and they can exist at several levels of aggregation of environmental data objects. For example, it may be useful to store which is the name of the organization responsible for measuring the concentration of oxygen. This corresponds to the fact in $L^u$:

**Example 4.4**

```
responsible(oxygen-concentration, ''Company OC'')
```

if we want to state that "Company OC" is the responsible for all measurements of oxygen concentration. But we will be also talking about a UDK metadata object if we say that "Company OC" is the company that is responsible for the measure of oxygen concentration in the Thames river. And this corresponds to the fact in $L^0$:

**Example 4.5**

```
oxygen-concentration(''Thames'', 50, Jan/97,
                     ''Company OC'')
```

This means that UDK metadata objects are mapped into predicates in $L^0$ and in $L^u$. Both of them correspond to data description.

Figure 5 resumes the classification that we have done by example.

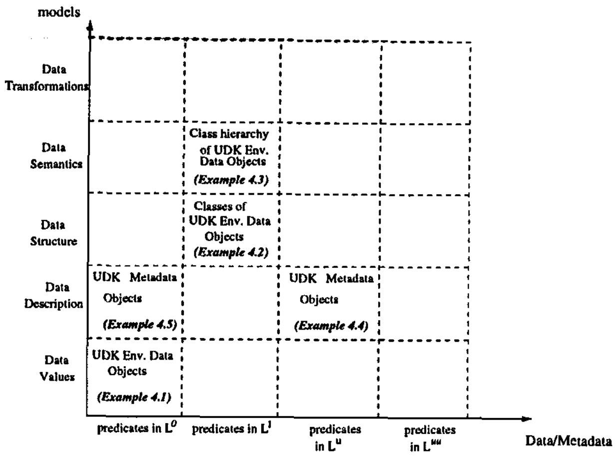


Figure 5: Classification of UDK model metadata

## 4.2 FDGC metadata format

The FGDC (US Federal Geographic Data Committee) standard [6], as it is common known, is in fact called *Content Standards for Digital Geospatial Metadata* (CSDGM) and it was approved by FDGC in 1994. This standard is composed by a set of terminology and definitions described by metadata descriptive elements. Its main purpose is to help to determine the availability, suitability and means to access to geospatial data (although it can also serve to other types of environmental data). The CSDGM metadata elements are organized in seven groups according to the information they provide: Identification, Data Quality, Spatial Data Organization, Spatial Reference, Entity and Attribute, Distribution and Metadata Reference. A short example, taken from the standard, of the syntax and semantic description of the *Identification* group is:

```
Syntax:
  Identification_Information = Citation + Description
                               + ...
  Citation = Citation_Information
  Description = Abstract + Purpose + ...
  Citation_Information = 1{Originator}n +
                         Publication_Date + ...
Semantics:
  Citation - information to be used to reference
   the data set. Compound.
  Originator - the name of an organization or
   individual that developed the data set. If
   the name of editors or compilers are provided,
   the name must be followed by "(ed.)" or
   "(comp.)" respectively. Type: text; Domain:
   "Unknown" free text
  Publication_Date - the date when the data set
   is published or otherwise made available for
   release. Type: date; Domain: "Unknown",
   "Unpublished material" free date
  Description - a characterization of the data
   set, including its intended use and limitations.
   Compound.
    Abstract - a brief narrative summary of
     the data set. Type: text; Domain: free text
    Purpose - a summary of the intentions with
     which the data set was developed. Type: text;
     Domain: free text
...
```

For classifying the associated metadata, we will follow the same procedure as we did for the UDK model. Let us suppose that one wants to store satellite images from beaches. The element *Originator* from the *Identification* group of metadata, makes a correspondence between the data set (or a particular data element) and a data value. It is represented as a predicate in $L^u$ (or by a predicate in $L$):

### Example 4.6

```
originator(beach-image, ''Unknown'')
beach-image(''Miami Beach'', XOFA16, ''Unknown'')
```

where *X0FA16* is the identifier of the image.

The Entity and Attribute group of metadata describes the entities, the attributes and the attribute domains of the data set. This corresponds to the conceptual schema of a database. It is then represented by facts in $L^1$, like:

### Example 4.7

```
entities(beach-image)
attributes(beach-image, image-id, Integer)
attributes(beach-image, image, Image)
```

The other distinct group of metadata is Metadata Reference that describes the other metadata groups. We may want to store the freshness of metadata elements. For instance, the freshness of *Originator* is represented by the fact:

### Example 4.8

```
freshness(originator, January/97)
```

which is a fact in $L^{uu}$(if *originator* was considered as a fact in $L^u$) or is a fact in $L^0$ (if *originator* was considered as being embedded in a fact in $L^0$).

Entity and Attribute metadata elements describe data structure. Metadata Reference elements still describe data, even though they describe metadata. The rest of the FGDC Metadata groups add some information about data so they describe data. The whole metadata classification is illustrated in figure 6.

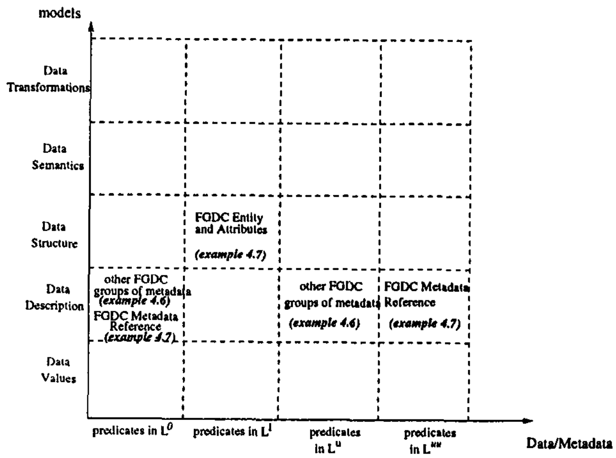


Figure 6: Classification of the FGDC metadata standard

## 5 Conclusions

To summarize what was done through this paper, we firstly built, by-example, a formal classification of metadata based on first-order logic. We identified three types of metadata (predicates in $L^1$, predicates in $L^u$ and predicates in $L^{uu}$). Through a grid we defined exactly the meaning of these kinds of metadata. We also identified the types of metadata, from the framework, that are used through the environmental data production and management. Finally, we applied the framework against an environmental metadata model and a standard format and defined a procedure for classifying them.

## References


[1] Jon Barwise. *Mathematical Logic.* North Holland, 1977.

[2] Kate Beard. A Structure for Organizing Metadata Collection. In *Third International Conference/Workshop Integrating GIS and Environmental Modeling*, Santa Fe, New Mexico, USA, January 1996. http://ncgia.ncgia.ucsb.edu:80/conf/SANTA_FE_CD-ROM/sf_papers/beard_kate/metadatapaper.html.

[3] Centre for Earth Observation Programme of the European Commission. *Recommendations on Metadata*, May 1997.

[4] Oliver Guenther and Agnès Voisard. Metadata in Geographic and Environmental Data Management. In W. Klas and A. Sheth, editors, *Managing Multimedia Data: Using Metadata to Integrate and Apply Digital Data.* McGraw Hill, 1997.

[5] Olivier Guenther. *Environmental Information Systems.* Springer-Verlag, 1998.

[6] NSDI. The Federal Geographic Data Comittee Home Page. http://www.fgdc.gov/, June 1997.

[7] Eric Simon. Personal Communication, December 1997.

[8] Anthony Tomasic and Eric Simon. Improving Access to Environmental Data using Context Information. *SIGMOD Record*, 1997.