Spatially-Aware Information Retrieval on the Internet

SPIRIT is funded by EU IST Programme
Contract Number: IST-2001-35047

Report on Metadata Models and Mark-up Languages

Deliverable number:	D2 6101
Deliverable type:	R
Deliverable nature:	PU
Contributing WP:	WP 6
Contractual date of delivery:	30th September 2002
Actual date of delivery:	30th September 2002
Authors:	Bénédicte Bucher, IGN
Keywords:	Metadata, Metadata Models, Mark-up languages

Abstract: The goal of this deliverable is to investigate appropriate mark-up languages and metadata standards for SPIRIT.

Contents

Executive Summary

SPIRIT (Spatially-Aware Information Retrieval on the Internet) project aims to improve users' access to spatially related information on the Web. It focuses on a better interpretation of the spatial context of a resource, which matches user request for information. Therefore, specific requirements for metadata need to be explored.

It is anticipated that the SPIRIT will make use of existing metadata schemas, with a possibility of adapting them to the prototype, and produce new metadata schemas dedicated to the prototype functions as well as acquire the corresponding metadata. SPIRIT will rely on widely adopted standards in particular from ISO and W3C.

This report investigates techniques for defining metadata in the SPIRIT context.

Section 2.1 introduces the main concepts about metadata and mark up languages. Sections 2.2 and 2.3 briefly expose SPIRIT's concern for metadata with regards to specific requirements for metadata used within SPIRIT. The following types of resource are distinguished: geographic data sets and Web documents. The need for metadata to support spatially-aware information retrieval and result ranking, together with ontologies, is emphasized.

Sections 3 and 4 present the elements, models and languages, we may use to implement our metadata and the possibilities these languages provide us with for defining our own structures. Specifically, we underline that the ISO19 115 model for geographic metadata may be used to depict not only geographic data sets but also Web documents.

Section 5 concludes this delivery by summarizing choices that have to be made concerning metadata and by sketching a draft model for SPIRIT metadata.

In the appendix, examples of available metadata can be found. They show the gap between the theory developed for storing and querying metadata, and the existing metadata sets.

D2 6101

Metadata and mark-up languages

1. Introduction

Accessing information resources is an activity that has radically changed over the past ten years, thanks to growing usage of the Internet and the Web. The new paradigm of a "Semantic Web" has further modified the access scenario, by stating ambitious goals - many operations involving semantics, that are currently performed by the user himself or not performed at all, should in the future be performed automatically.

The technical foundations of a semantic Web are expressive, shareable and machine-readable semantics. Practically, this necessitates the definition of modalities for exchanging more and more structured information. These modalities are *explicit models* (thesaurus, schema,…), *mark up languages* and *metadata*.

SPIRIT works on users access to spatially related information on the Web. It focuses on a better interpretation of the spatial context of a resource, to match it with a user's information request. Therefore, specific requirements for metadata are to be explored. SPIRIT will use existing metadata models, possibly adapt them for the prototype, and produce new metadata standards/schemas dedicated to the prototype functions. It will rely on widely adopted standards.

2. Metadata: components and functionality

2.1. Models, mark-up languages and metadata

The Semantic Web

Tim Berners-Lee has introduced the notion of Semantic Web as "an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [Berners-Lee et al. 01].

This objective calls for the possibility to perform specific operations on information documents. These operations may be the exchange of information documents (possibly with complex semantics) between computer and human components, and the discovery of information documents.

The exchange of information documents operates at different levels, from physical signals to meaning. Low levels like operating systems, communication protocols and data formats are handled by the Internet. Interoperability at these levels allows data exchange.

But information exchange needs interoperability at a higher level: models and languages.

Models

There are two functions of models: to define structures and to define vocabularies.

Models are used to define structures, i.e. information containers, and to specify the types of data used to fill these containers. A structure of data is often dedicated to a specific type of operation.

Models are also used to define agreements on terms resulting in specific vocabularies. A vocabulary may rely on a structure, e.g. on objects or on entities and relationships. An ontology is such a particular model of concepts. A thesaurus is a model of words, concepts, and relationships between words and concepts (a word *represents* a concept), between words and words (a word *is synonymous of* another word) and between concepts and concepts (a concept *is decomposed into* other concepts).

Language

Languages support authorship of "machine-readable" documents.

A language is structured according to a particular data model. For instance the Java language refers to an object-oriented model. The information containers are the classes, the interfaces, the objects and their attributes and methods. The types of data are the primitive types defined in Java like *int*, *string* and *boolean*.

Documents written in languages structured after according to a given model can be exploited with specific operations associated to this model. For instance an XML document can be parsed thanks to the tags structure, and may be exchanged due to the nature of its elementary datatype, the *string*.

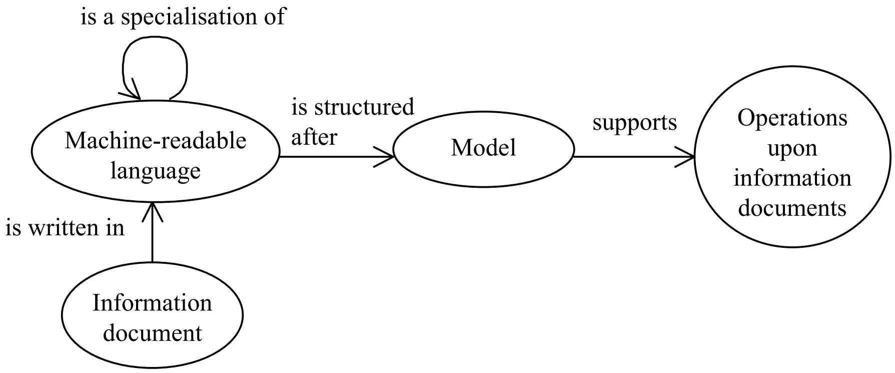


Figure 1. **Elements involved in the creation of information documents exchangeable on the Web.**

Not only do languages refer to structuring models, but also they may refer to vocabularies. When using a term defined in a vocabulary, the author of a document may mention which vocabulary is used, but it is up to the reader to interpret correctly the term.

From the Web to the Semantic Web

The Web is defined around the mark up language HTML. This language refers to a simple model that organizes information for a limited set of operations performed by browsers (displaying pages, activating hypertext links) and by search engines (indexing HEAD sections). It does not support specific vocabularies.

In contrast with HTML, the Semantic Web calls for a less ambiguous exchange of information, which should be supported by common vocabularies available on the Web as well as implementation languages to encode statements made with these vocabularies.

The Semantic Web also calls for the automation of many operations composing resource discovery. Such automation should rely on data structured according to a model dedicated to these operations. In other words, the Semantic Web needs structured languages not only dedicated to presentation but to other operations. And it also needs specific data dedicated to these operations. These are metadata.

Metadata

Metadata are often defined as “data about data”, but we also propose the following pragmatic definition:

“Metadata are data dedicated to support operations about resources that *cannot* be performed on the resources themselves.” The “*cannot*” has several meanings: these operations can not be performed on the resource because the needed information is not explicit in the resource, these operations can not be easily performed on the resources because the needed information is ill-structured in the resources, these operations can not be performed on the resources because they are not available, etc.

Part of the information retrieval thus consists in building standard models dedicated to operations that make up information retrieval and exchange, e.g. the description of resources or the expression and interpretation of a user's need. The *Semantic Web* now refers to a global effort, defined by the W3C as follows: "The Semantic Web is the abstract representation of data on the World Wide Web, based on the RDF standards and other standards to be defined. It is being developed by the W3C, in collaboration with a large number of researchers and industrial partners."

Proposals for metadata models are always divided into an abstract model, and an implementation of this abstract model using a standard machine-readable language as shown in Figure 2. The abstract model expresses how to formalize information, and the implementation model expresses how to encode statements produced with respect to the abstract model.

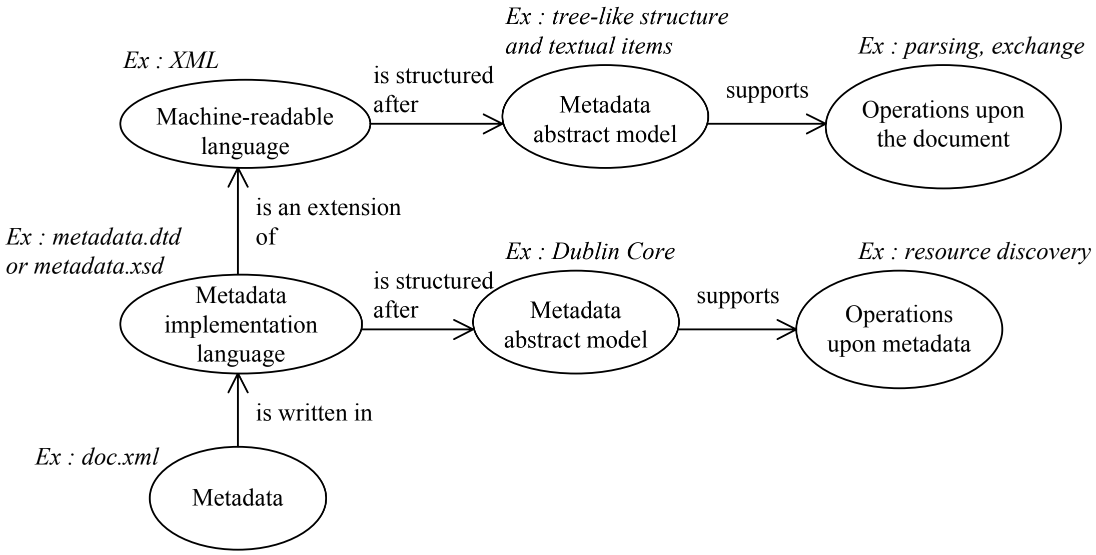


Figure 2. **Metadata model and implementation language.**

The components of an abstract metadata model are:

- A set of metadata elements, or fields of description, organized into aggregates. These are the properties of the resource.
- The definition of range domains that should be used to document each metadata field. It might be a simple datatype like a "free text", an "integer", or it can be an item in a given vocabulary.

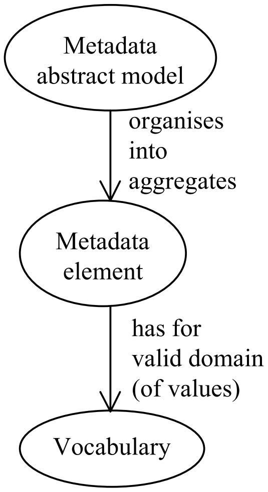


Figure 3. **Components of a metadata model.**

For example, let us examine the metadata containers defined in HTML. The generic metadata model of HTML is as follows: a metadata element has a *name* and *content*. The names may be specified in *profiles* and the content can be specified in a *scheme*. The following example is taken from the HTML Specification of the W3C :

```
<HEAD profile="http://www.acme.com/profiles/core">
        <TITLE>How to complete Memorandum cover sheets</TITLE>
        <META name="author" content="John Doe">
        <META name="copyright" content="&copy; 1997 Acme Corp.">
        <META name="keywords" content="corporate,guidelines,cataloging">
        <META name="date" content="1994-11-06T08:49:37+00:00">
        <META scheme="ISBN"  name="identifier" content="0-8230-2355-9">
</HEAD>
```

2.2. SPIRIT metadata activities

In this section, we enumerate SPIRIT activities regarding metadata. We use SPIRIT MD to refer to SPIRIT metadata. In section 2.3, the requirements regarding metadata and mark up languages are defined.

We distinguish two types of activities: the first kind of activities take place at the beginning of the metadata life cycle and lead to the creation of a metadata model, schema, and database, the second type of activities are the exploitation of metadata. They are summed up in Figure 5 below.

The creation of metadata sets

- **Content specification**

The first activity is the specification of the content of metadata, i.e. what aspects of SPIRIT resources shall be described. Answer to this question depends on two factors: the resources themselves and the intended use of the description. The intended use is discussed in detail in this section.

Several types of resources should be described as depicted in Figure 4: Web sites and geographical data sets.

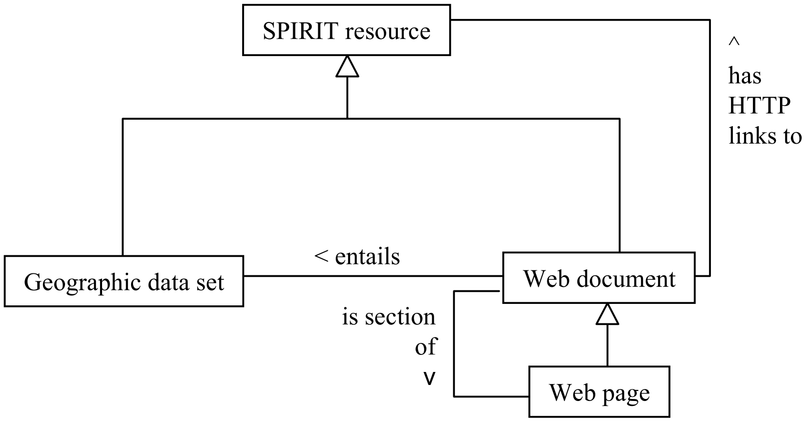


Figure 4. **Types of SPIRIT resources.**

Part of the metadata content should be generic items common to all types of resources, including the spatial extent. The structure of such items may be found by comparing metadata models for both types of resources. Other items of metadata are specific to a type of resource. For instance, the geometric accuracy is specific to the type of geographic data set.

In the content specification, it must be specified how the defined metadata elements, e.g. “spatial extent” are to be documented on specific resources.

- **Models design**

An abstract model for SPIRIT MD must be designed. It is organized into several models.

The resource description model should organize the description of resources inside SPIRIT. It should comprise some generic items, some items specific to the description of geographic data sets and some items specific to the description of Web pages. These items must be compliant with existing standards for these types of resources that are described in section 3. In this model, metadata may be associated to a resource or to a collection of resources.

A metadata-querying model is also needed to support the matching of a user's query with a ranked set of resources. It should thus represent:

- Expressions of user queries.
- Contextual data necessary to interface the user’s representation of geographic space and of their need with the representation of geographic space in the metadata.
- Expressions of a request on the metadata DB.
- Criteria for ranking resources.

Designing this model needs input from the "Metadata" WP and also from the "Ontologies" WP. A draft model is sketched in section 4.3.

- **Acquisition**

There are several scenarios for metadata acquisition:

- To reuse already available metadata. This might be data that are already identified as "metadata", or this can be data that are not yet labeled as "metadata", and thus are often not structured.
- To derive metadata by means of Data Interpretation and Data Mining.
- To allow “qualified” users to annotate resources (cf Open Directory).

The reuse of existing metadata is impeded by the lack of metadata. In Europe it is not mandatory to document GDB (Geo-Database) after standards and most GDB have no metadata. This is also true on the Web. The appendices give some examples of available metadata illustrating this issue. Specifically, we emphasize that an effort of metadata acquisition must be undertaken. Also, it is important that metadata should rely on data interpretation.

SPIRIT intends to rely on Data Interpretation and Data Mining to gather metadata about web documents when metadata are not yet available. For instance, the spatial coverage of websites is seldom explicit in metadata enclosed within the HTML source (see examples in the Appendix B). Making the spatial context of a resource explicit is an important task in SPIRIT. This could mean to scan the content of HTML META tags named "abstract", "description" and "keywords", and look for expressions that are of the form: "relationship + geographic zone".

This will be the goal of the deliverable D14 (“Extraction of Semantic Annotations from Textual Web Pages”) and deliverable D29 (“Functional Prototype Implementing the Annotation Methods”).

Possible exploitation of the metadata depends on the degree of automation of the acquisition process. For instance, if it is low and cannot be undertaken for all resources, it would be difficult to ground resource discovery on these metadata.

- **Implementation**

The abstract model must be translated into an implementation model. Depending on the chosen implementation language, specific operations will be provided on the encoded model. An implementation in a mark-up language, as shown in Figure 2, is useful for the following operations:

- Metadata exchange.
- Metadata browsing by a human person.

Moreover, specific event-driven parsing programs can be written that are dedicated to a specific XML structure.

The implementation of the metadata also raises the question of where to store the metadata. There are several possibilities:

- Metadata might be stored in the resources.
- Metadata might be encoded as annotations external to the resources and attached to part of the resource.

If metadata are stored in a DB, the implementation is not limited to the specialization of a mark up language. It is also necessary to define the logical structure of the metadata DB.

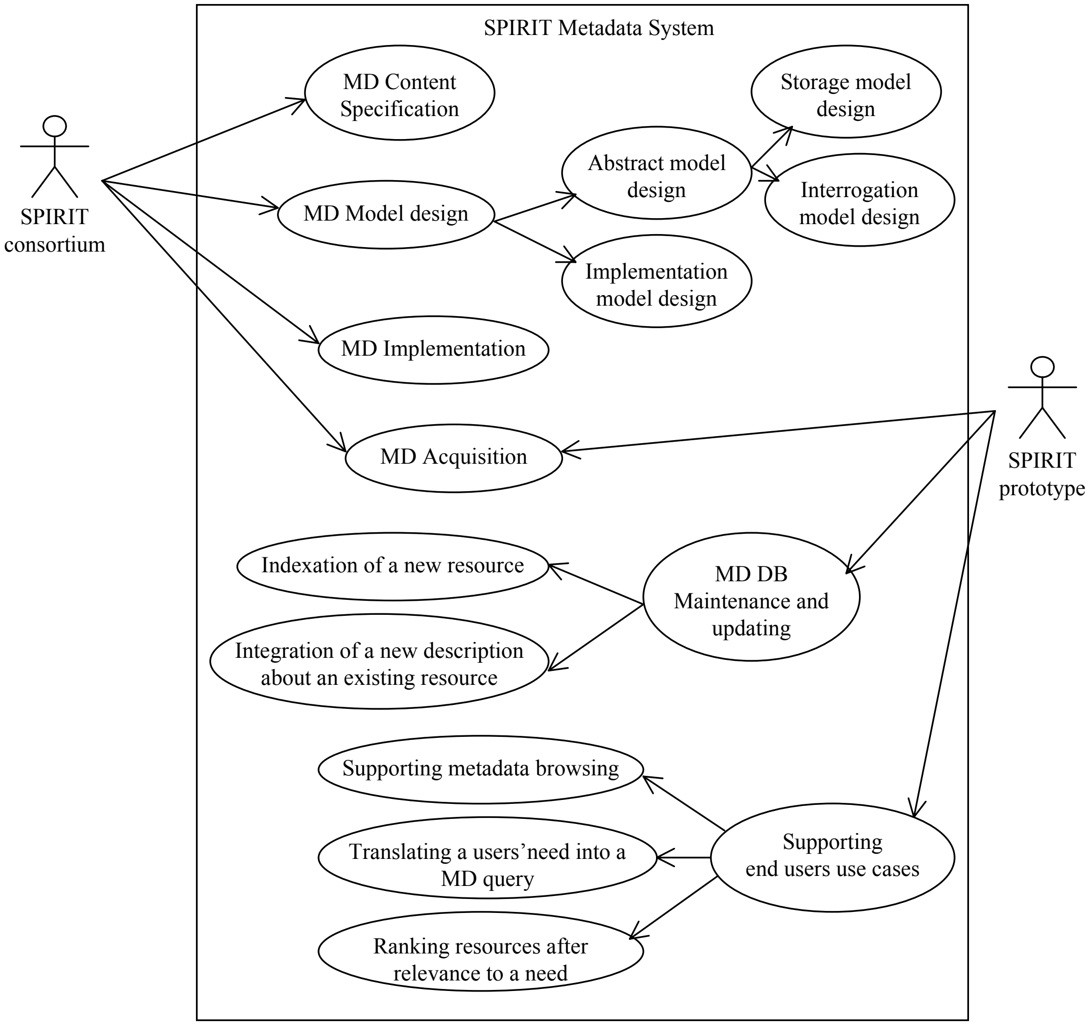

SPIRIT Metadata System
SPIRIT consortium
SPIRIT prototype
MD Content Specification
MD Model design
Abstract model design
Storage model design
Interrogation model design
Implementation model design
MD Implementation
MD Acquisition
MD DB Maintenance and updating
Indexation of a new resource
Integration of a new description about an existing resource
Supporting end users use cases
Supporting metadata browsing
Translating a users'need into a MD query
Ranking resources after relevance to a need

Figure 5. **Use cases of SPIRIT metadata system (during the whole life cycle of this system).**

The exploitation of metadata sets

Whereas the preceding activities are linked to the creation of the MD system, the following activities are use-cases of the MD system provided it exists. In this domain, some choices are still to be made, as we explain in this section.

Information retrieval is often detailed as follows (GSDI Cookbook, ETeMII Metadata Users' Needs):

- *resources discovery* : the user expresses a need for info and SPIRIT answers by a list of resources indexed in SPIRIT.
- *resources exploration* : the user may ask to know more about some resources that appear in the list.
- *resources exploitation* : the user may ask to retrieve/use some resources.

We detail these three stages in SPIRIT.

• Resource discovery

The resource discovery stage consists of defining a set of keys that can be used in an index of resources to retrieve resources of potential interest for the user. The type of “keys” may vary depending on the indexing solution chosen by the system. If the index is a metadata DB, a key is a query on the metadata DB. If the index is a spatial index, the “key” is a query for a geographic footprint in this index. The geographic footprint that is used to represent a resource in the spatial index is actually a specific metadata of the resource. If the index is a textual index, like in Google or Glass [Sanderson 00], the key is a set of words.

In the cases where a metadata DB or a spatial index is used for index, metadata alone are not enough to support resource discovery. Indeed the expression of the user's need and the metadata documentation seldom use the same terms. For instance, the user may speak a different language. He may also use his own representation of geographic space that does not fit with the representation used by the metadata schema. To translate the expression of a user need into an expression built in the metadata querying model, i.e. into a valid formal query, contextual databases are needed that translate user terms into metadata terms. This translation is of two types:

- It concerns with the value used to document metadata elements. For instance, the user may specify a place name as the required coverage, whereas this field of metadata is documented with a set of geographic coordinates.
- It may also concern with the translation of the information need into the fields of the metadata. For instance, a user needs “data to prepare a bicycle ride”, which may be translated into “data such as schema entails roads and paths, and scale is more than 1:25 000”.

In the following, we will use the term contextual database to refer to data needed in the discovery process, to formalize the user's need. Contextual databases are often ontologies.

SPIRIT focuses on answering two types of user needs:

- type 1 : "geographic data sets about some aspects of a geographic zone"
- type 2 : "information about *something* related to a geographic zone". *Something* might be "a garage", "an airport", and “a vegetal species". *Related to* can be more specifically "near", "reachable by car from", and "in".

SPIRIT will propose a more accurate interpretation of the users' words to express *something*, *related to* and *geographic zone*. This relies much on the contextual databases and on the metadata-querying model as shown in Figure 6.

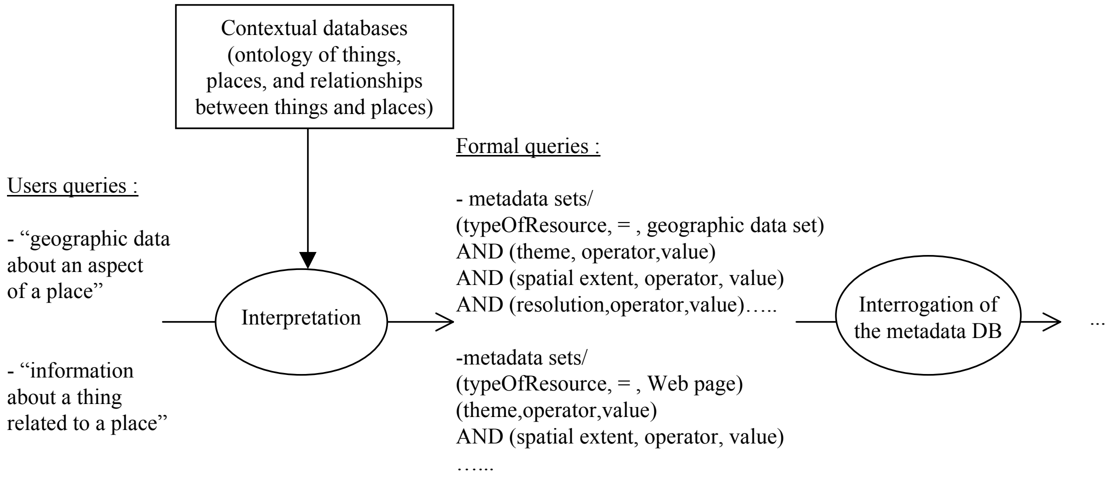

Contextual databases (ontology of things, places, and relationships between things and places)
Users queries :
- “geographic data about an aspect of a place”
- “information about a thing related to a place”
Interpretation
Formal queries :
- metadata sets/
(typeOfResource, = , geographic data set)
AND (theme, operator,value)
AND (spatial extent, operator, value)
AND (resolution,operator,value)…..
-metadata sets/
(typeOfResource, = , Web page)
(theme,operator,value)
AND (spatial extent, operator, value)
…...
Interrogation of the metadata DB
...

Figure 6. **First step of the discovery process : interpretation of user's query into a formal query that may be presented to the metadata sets.**

For instance, a query "tourism in England" should be interpreted as: "metadata sets where spatial extent is included in England and theme is tourism or a sub theme of tourism." A query "weather forecast on big cities in Japan" should be interpreted as: "metadata sets where spatial extent includes {Tokyo, Osaka,…} and theme is “weather forecast".

Contextual databases should support inferences like:

- "about tourism" implies "theme *belongs to* s1" where s1 is a set of themes that are sub themes of tourism;
- "tourism in zone" implies "spatial extent *is included in* zone";
- "weather forecast on zone" implies "spatial extent *includes* zone";
- "big cities in Japan" are the cities : Tokyo,…..

The links between this and the metadata description model are the valid formal queries supported by the interrogation model, i.e. the fields of metadata, the corresponding valid operators and values, and the vocabularies supported by the description model as shown in Figure 7.

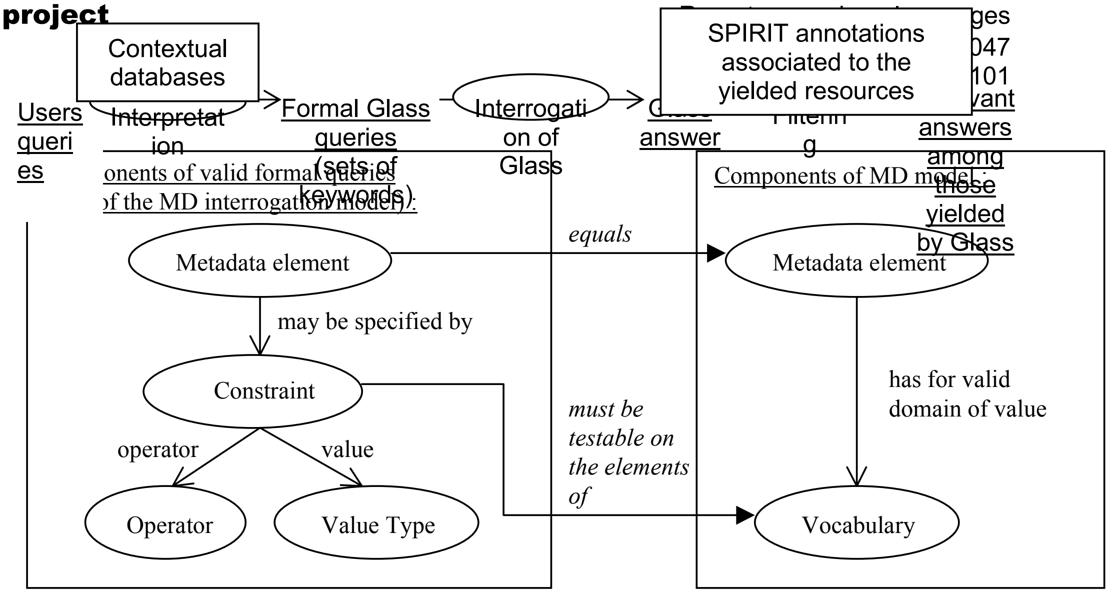

Contextual databases
Users queries
Interpretation
Formal Glass queries (sets of keywords)
Interrogation of Glass
Glass answer
Filtering
SPIRIT annotations associated to the yielded resources
Relevant answers among those yielded by Glass
Components of valid formal queries (of the MD interrogation model):
Metadata element
may be specified by
Constraint
operator
value
Operator
Value Type
equals
must be testable on the elements of
Components of MD model :
Metadata element
has for valid domain of value
Vocabulary

Figure 7. **Linking between the interrogation model and the MD model.**

In the case where a textual index is used, like that of Google or Glass [Sanderson 00], contextual databases are still needed to translate the user need into a key adapted to the index, which is a set of keywords as shown in

Figure 8.

- **Relevance assessment and resources ranking**

No matter how the resource discovery is performed (thanks to a metadata DB or a search engine), metadata can also be used to assess the relevance of the retrieved resources as shown in

Figure 8.

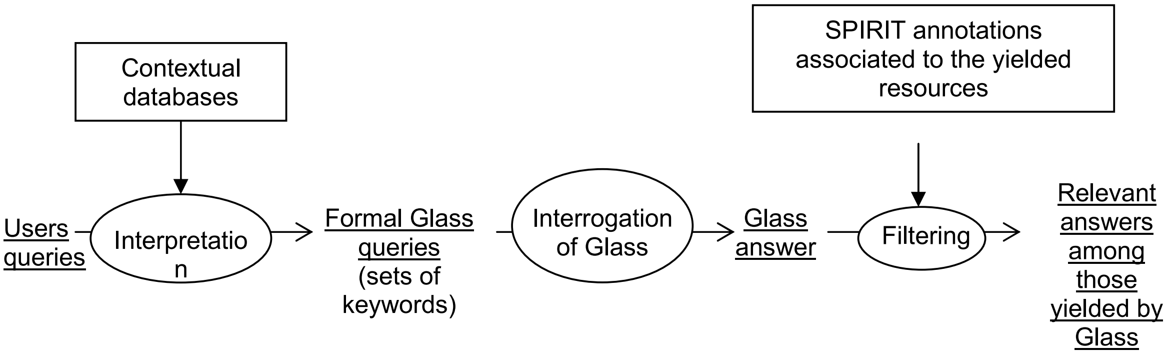

Contextual databases
SPIRIT annotations associated to the yielded resources
Users queries
Interpretation
Formal Glass queries (sets of keywords)
Interrogation of Glass
Glass answer
Filtering
Relevant answers among those yielded by Glass

Figure 8. **Filtering of answers yielded by Glass using annotation metadata.**

Moreover, SPIRIT will rank the items answering the user's query as shown in Figure 9. This ranking should also be supported by metadata elements.

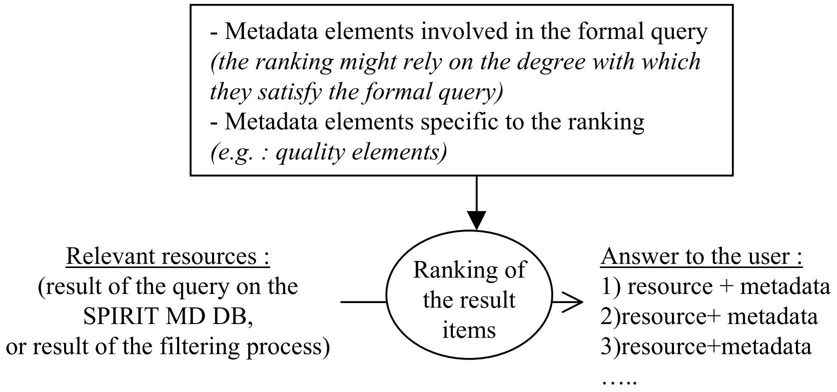

- Metadata elements involved in the formal query
(the ranking might rely on the degree with which they satisfy the formal query)
- Metadata elements specific to the ranking
(e.g. : quality elements)
Relevant resources :
(result of the query on the SPIRIT MD DB,
or result of the filtering process)
Ranking of the result items
Answer to the user :
1) resource + metadata
2)resource+ metadata
3)resource+metadata
…..

Figure 9. **The ranking of the results.**

- **Displaying resources for the user**

A last exploitation of metadata is to help the user understand a resource that has been selected as a result to his query.

Metadata may contain information about how to display the resource. Annotations often contain information to assist the user in understanding the content of resources. In fact, the W3C effort about metadata is dedicated to enhancing collaboration between different users authoring different resources.

- **Maintenance**

The last exploitation activity, which is not part of information retrieval, is the maintenance. The MD database must be extensible in order to allow the following operations:

- integration of new resources;
- modification of descriptions of existing resources;
- creation of new operations, like applying change of spatial coordinates to geographic data sets, or mapping several results on one map.

2.3. Metadata requirements

This section sums up requirements general or related to SPIRIT objectives. These requirements concern the metadata model, as well as the mark-up language.

A major requirement concerning metadata is their interoperability: metadata created in SPIRIT should be usable by other systems, and most SPIRIT's operations should be applicable to classical metadata models. This is ensured by the conformance to existing standards.

Another requirement is the scalability of the metadata. Descriptions can occurred at several levels: a collection of resources, a resource or a portion of resource, which implies management of relationships between metadata. This is handled in standard models like RDF and ISO19115.

It is also important to build an extensible model to possibly further integrate new elements. It implies avoiding redundancy in the storage model. This is handled in standard models.

These are general requirements. Other requirements are specific to SPIRIT.

In the context of SPIRIT, the model of metadata should support the explicit representation of the spatial context of a resource. More precisely, it should comprise an element describing the geographic footprint of a resource. The footprint instances will be used to build a spatial index of resources. It should also represent other information that is used for ranking.

Contextual databases are also needed as shown Figure 6. An ontology of places, an ontology of "things", and an ontology of relationships between things and places are required. These ontologies also help to meet requirements, regarding the interpretation of users' queries and the ranking.

3. Models for metadata

This section describes the main metadata standards regarding SPIRIT resources. For each standard, we specify its status (de facto standard, proposed recommendation, recommendation), and go into detail on some specific parts.

3.1. Highly generic models : RDF(S), Topic Map (ISO 13250)

RDF (Resource Description Framework)

RDF is a highly generic model to describe resources. It is a recommendation of the W3C.

RDF is a semantic net like description format. It relies on two basic notions:

- A resource: it is anything that can have a Unique Resource Identifier (URI). A resource may be anonymous. Examples of resources are a web page, a service off-line, a museum, a theory.
- A property: a characteristic that may be attached to a resource. It must have a name. Examples of properties are "author", "price", "spatial extent".

A RDF description is composed of statements. A statement, as depicted in Figure 10 employs resources and properties. It has three parts: [subject, predicate, object]. The subject is a resource, the predicate a property attached to the resource, and the object is either a resource or a literal.

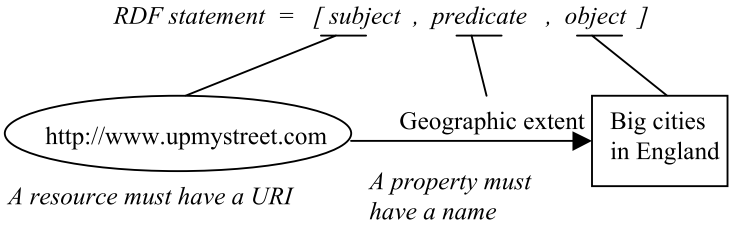

RDF statement = [subject , predicate , object]
http://www.upmystreet.com
Geographic extent
Big cities in England
A resource must have a URI
A property must have a name

Figure 10. **Graphical representation of a RDF statement.**

RDF statements as such are not shareable. Resources are identified through their URLs, but the name of a property might refer to different properties for the author of the statement and for the reader.

To improve this situation, the W3C has defined a mechanism to define unambiguous vocabularies, similar to the *namespace* mechanism proposed in XML. A vocabulary is built as a schema, possibly of the type RDFS (RDF Schema). Such a schema specifies properties by their names and the restriction of usage. The domain of a property is the set of elements that have a property. The range of a property is the set of possible values this property has. The schema also specifies types of resources (*classes*) identified by their names. A property defined in a schema is itself a resource because it has a URL which is composed of the URL[1] of the schema and the name of the property in the schema.

We may, for instance, define a schema that specifies the following properties:

- The domain of the property "type" is any resource, and its range is the set of instances of the classes "SPIRIT resource" or "geographic zone".
- The domain of the property "geographic extent" is the set of resources that have the "type" "SPIRIT resource". Its range is the set of resources that have the "type" "geographic zone". In other words, geographic extent of a SPIRIT resource is a geographic zone.
- The domain of the property "name" is the set of resources of "type" "geographic zone", and its range is the datatype string.
- The domain of the property "coordinates" is a set of resources of "type" "geographic zone", and its range is the datatype tuples of pair of floats.

We may then identify our schema by the abbreviated name *os* and write shareable RDF statements like: [http://www.upmystreet.com, *os*:type, *os*:SPIRITresource].

[1] The URL of a schema often looks like a URL but this is not necessarily a valid URL. Moreover, in a document, a schema is referred to by an abbreviated name, which is associated to the URL in the document. For instance, the W3C has defined particular RDFSs. Their abbreviated names are usually *rdf* and *rdfs*.

The RDF standard is extensible. The definition of a RDFS may rely on other RDFSs. Especially, rdf and rdfs are useful to define new RDFS. This contributes to the extensibility of models defined as RDFS.

Extended schemas may be written as RDF statements. From this on, a RDF statement might be:

- a description of an existing resource, i.e. a metadata;
- or a creation of a resource, e.g. definition of a class in a schema, creation of an instance of a class, creation of a whole ontology.

The fact that RDF statements may be used as building bricks for ontology has led to defining extension of RDF dedicated to the representation of complex knowledge constructs. Besides *rdf* (the basic RDF schema) and *rdfs* (the schema RDFS), the schema DAML+OIL is such a tool dedicated to the RDF representation of ontology [DAML+OIL, 2001]. It is proposed by a group of people from DARPA and from the European IST who work on agent mark-up languages. They form the "Joint Committee". The OWL Web Ontology Language is being designed by the W3C Web Ontology Working Group as a revision of the DAML+OIL web ontology language [OWL 2002].

Topic Map

The Topic Map model is an ISO standard dedicated to organizing browsable indexes for a large collection of resources. A topic map contains *topics* (topic link) that are related to subjects in the real world. Any topic is of one or more *topic types*. For instance the topic "Paris" is of topic types "Keyword"' and "Spatial extent". Any topic might be related to several resources. Each resource is an *occurrence* of the topic.

An occurrence is based on an *occurrence role* (examples of occurrence roles are: illustration, definition, example). It is also specified by an *occurrence role type* that is a reference to a topic in the map.

In a topic map topic associations describe the relationships between topics. They may be of *topic association types*. Topic association types are described as topics.

What is specific in this standard is the distinction of several categories of knowledge:

- reality or resource depicted (that are occurrences of topics);
- layer of indexing (the topics);
- relationships between reality and index.

Every category has a structure. The structure of reality itself is not explicit. The structure of the index is represented by topic types, topic associations, topic association roles. The structure of the relationships between the indexed reality and the index is represented by occurrence roles, and occurrence role types.

Currently, many attempts are made to represent Topic Maps in RDF (topics are resources, associations are properties). The only element of a Topic Map that has not yet a corresponding element in RDF is the "occurrence". So far, Topic Maps are mostly used to build user browsable indexes of resources rather than structured metadata.

3.2. A model dedicated to the description of Web documents: The Dublin Core

The Dublin Core (DC) model is a recommendation issued by an open group: the Dublin Core Metadata Initiative. It is dedicated to textual documents. Since web pages are mostly HTML documents, i.e. sort of textual documents, the DC is often seen as a model to depict Web pages.

In Figure 11 the definition of the Dublin Core elements extracted from "Dublin Core Metadata Element Set, Version 1.1: Reference Description " is given. Each element is described by its name, a definition and a comment. We keep only those comments that are important in the specific context of SPIRIT. All elements are optional and can be repeated several times.

The DC can be extended in two ways: by decomposing its elements or by adding new elements.

Element (Name when different)	Definition	Comment
Title	A name given to the resource	
Creator	An entity primarily responsible for making the content of the resource	
Subject (Subject and Keywords)	The topic of the content of the resource	Typically, a Subject will be expressed as keywords, key phrases or classification codes that describe a topic of the resource. Recommended best practice is to select a value from a controlled vocabulary or formal classification scheme.
Description	An account of the content of the resource.	Description may include but is not limited to: an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content.
Publisher	An entity responsible for making the resource available	
Contributors	An entity responsible for making contributions to the content of the resource.	
Date	A date associated with an event in the life cycle of the resource.	
Type	The nature or genre of the content of the resource.	Type includes terms describing general categories, functions, genres, or aggregation levels for content. Recommended best practice is to select a value from a controlled vocabulary (for example, the working draft list of Dublin Core Types [DCT1]). To describe the physical or digital manifestation of the resource, use the FORMAT element.
Format	The physical or digital manifestation of the resource.	
Identifier	An unambiguous reference to the resource within a given context.	
Source	A reference to a resource from which the present resource is derived.	
Language	A language of the intellectual content of the resource.	
Relation	A reference to a related resource.	
	The extent or scope of the content of the resource.	Coverage will typically include spatial location (a place name or geographic coordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a

Coverage		named administrative entity). Recommended best practice is to select a value from a controlled vocabulary (for example, the Thesaurus of Geographic Names [TGN]) and that, where appropriate, named places or time periods be used in preference to numeric identifiers such as sets of coordinates or date ranges.
Rights (Rights Management)	Information about rights held in and over the resource.	

Figure 11. **Dublin Core Metadata elements.**

3.3. The ISO 19115 draft standard for geographical metadata

Several models for geographical metadata have been proposed, de facto standards and official standards. In the last years, metadata models working groups have tried to federate their work and to produce a unique metadata model. This will soon be the ISO19115 standard for geographical metadata. So far, ISO has released a draft standard for geographical metadata. We describe in this section the main elements of this draft standard.

In the context of SPIRIT it is important to highlight an observation brought by the authors of this draft: "though this International Standard is applicable to digital data, its principles can be extended to many other forms of geographic data such as maps, charts, and textual documents as well as non-geographic data."

ISO19115 introduces the following notions :

- A metadata element is a discrete unit of metadata (an attribute or an association in UML terminology).
- A metadata entity is a set of metadata elements describing the same aspect of data (a class in UML terminology). A metadata element is unique within a metadata entity.
- A metadata section is a subset of metadata that consists in a collection of related metadata entities and metadata elements (a package in UML terminology).

Moreover, ISO defines specific datatypes that should be used to document the model. Some are defined in other ISO standards. Some are CodeLists.

The draft standard defines several metadata sections (or packages) that may be depicted as one or more entities. Figure 12 shows the relationships between the high-level metadata entities in ISO19115.

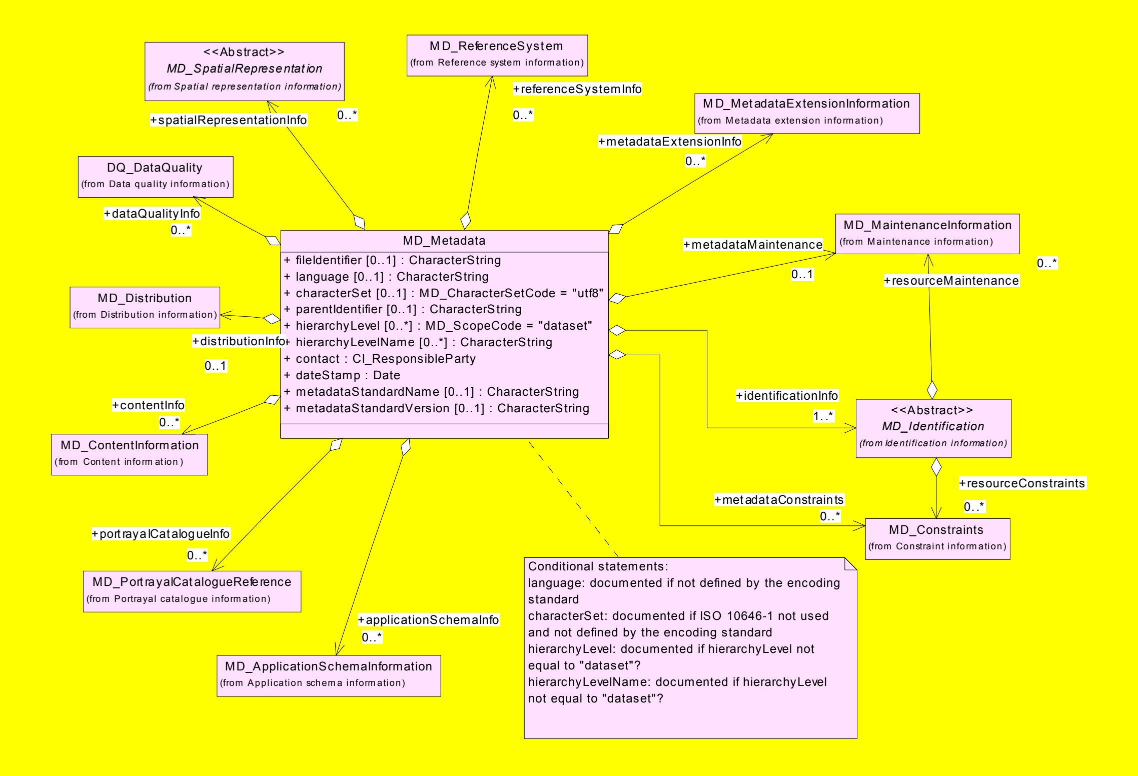

<<Abstract>>
MD_SpatialRepresentation
(from Spatial representation information)
MD_ReferenceSystem
(from Reference system information)
+referenceSystemInfo
0..*
MD_MetadataExtensionInformation
(from Metadata extension information)
+spatialRepresentationInfo
0..*
+metadataExtensionInfo
0..*
DQ_DataQuality
(from Data quality information)
+dataQualityInfo
0..*
MD_Metadata
+ fileIdentifier [0..1] : CharacterString
+ language [0..1] : CharacterString
+ characterSet [0..1] : MD_CharacterSetCode = "utf8"
+ parentIdentifier [0..1] : CharacterString
+ hierarchyLevel [0..*] : MD_ScopeCode = "dataset"
+ hierarchyLevelName [0..*] : CharacterString
+ contact : CI_ResponsibleParty
+ dateStamp : Date
+ metadataStandardName [0..1] : CharacterString
+ metadataStandardVersion [0..1] : CharacterString
MD_MaintenanceInformation
(from Maintenance information)
+metadataMaintenance
0..1
0..*
+resourceMaintenance
MD_Distribution
(from Distribution information)
+distributionInfo
0..1
+identificationInfo
1..*
<<Abstract>>
MD_Identification
(from Identification information)
+contentInfo
0..*
MD_ContentInformation
(from Content information)
+resourceConstraints
0..*
+metadataConstraints
0..*
MD_Constraints
(from Constraint information)
+portrayalCatalogueInfo
0..*
MD_PortrayalCatalogueReference
(from Portrayal catalogue information)
+applicationSchemaInfo
0..*
MD_ApplicationSchemaInformation
(from Application schema information)
Conditional statements:
language: documented if not defined by the encoding standard
characterSet: documented if ISO 10646-1 not used and not defined by the encoding standard
hierarchyLevel: documented if hierarchyLevel not equal to "dataset"?
hierarchyLevelName: documented if hierarchyLevel not equal to "dataset"?

Figure 12. **ISO19115 Metadata entity set information ([ISO 01] p. 19 Figure A.1)**

The entity **MD_Identification** contains information to uniquely identify the data. It is detailed in Appendix A..

The entity **MD_Constraints** contains information concerning the restrictions placed on data.

The entity **DQ_DataQuality** contains a general assessment of the quality of the dataset.

It is detailed in Appendix A and might be used in SPIRIT in the ranking of the resources.

The entity **MD_MaintenanceInformation** contains information about the scope and frequency of updating data.

This information might be used in the ranking of the resources in SPIRIT.

The entity **MD_SpatialRepresentation** contains information concerning the mechanisms used to represent spatial information in a dataset. This class is specialised into two subclasses : MD_GridSpatialRepresentation and MD_VectorSpatialRepresentation.

The entity **MD_ReferenceSystem** contains the description of the spatial and temporal reference system(s) used in a dataset. It is detailed in Appendix A.

The entity **MD_ContentInformation** contains information identifying the feature catalogue used (MD_FeatureCatalogueDescription) and/or information describing the content of a coverage dataset (MD_CoverageDescription). It is detailed in Appendix A.

The entity **MD_PortrayalCatalogueReference** contains information identifying the portrayal catalogue used by the dataset.

The entity **MD_Distribution** contains information about the distributor of, and options for obtaining, a resource. It has zero or more attribute "transferOptions" which depict the unit of distribution (tiles, layers, geographic areas etc. in which data is available) and zero or more attributes "onLine" depicting online sources from which the data may be obtained.

The entity **MD_MetadataExtensionInformation** contains information about user specified extensions.

The entity **MD_ApplicationSchemaInformation** contains information about the application schema used to build a dataset.

The entity **EX_Extent** contains information about the application schema used to build a dataset.

Core elements

ISO defines a set of core metadata elements selected from all the elements appearing in the standard. This set is shown in the table below.

Element name (MD_Metadata….)	**Comment and domain**
Dataset title (M) (MD_Identification.citation > CI_Citation.title)	Name by which the resource is known (Free text)
Dataset reference date (M) (MD_Identification.citation > CI_Citation > CI_Date.date and CI_Date.dateType)	Reference date for the resource (date in year,month,day) and event used for reference date (in a Codelist :{creation,publication,revision})
Dataset responsible party (O) (MD_Identification.pointOfContact > CI_ResponsibleParty)	Person or organization "responsible for" the resource
Geographic location of the dataset (by four coordinates or by geographic identifier) (C) (MD_DataIdentification.geographicBox or MD_DataIdentification.geographicIdentifier)	
Dataset language (M) (MD_DataIdentification.lauguage)	
Dataset character set (C) (MD_DataIdentification.characterSet)	
Dataset topic category (M) (MD_DataIdentification.topicCategory)	Main theme(s) of the dataset (in a CodeList : {farming, biota, boundaries, climatology/meteorology/atmosphere,…})
Spatial resolution of the dataset (O) (MD_DataIdentification.spatialResolution > MD_Resolution.equivalentScale or MD_Resolution.distance)	Factors which provide a general understanding of the density of spatial data in the dataset (A specific datatype composed of equivalentScale (fraction) and distance)
Abstract describing the dataset (M) (MD_Identification.abstract)	Brief narrative summary of the content of the resource (Free text)
Distribution format (O) (MD_Distribution > MD_Distributor > MD_Format.name and MD_Format.version)	Description of the computer language construct that specifies the representation of the data objects (both, name and version of the format, are free text)
Spatial representation type (O) (MD_DataIdentification .spatialRepresentationType)	Method used to spatially represent geographic information (In a CodeList : {vector,grid,textTable,tin,stereoModel,video})
Reference system (O) (MD_ReferenceSystem)	Information about the reference system (Aggregated Class depicted in B3.4)
Lineage statement (O) (DQ_DataQuality > LI_Lineage.statement)	General explanation of the data producer's knowledge about the lineage of the dataset (Free text)
On-line resource (O) (MD_Distribution > MD_DigitalTransferOption.onLine > CI_OnlineResource)	
Metadata file identifier (O) (.fileIdentifier)	
Metadata standard name (O) (.metadataStandardName)	
Metadata standard version (O) (.metadataStandardVersion)	
Metadata language (C) (.language)	
Metadata character set (C) (.characterSet)	
Metadata point of contact (M) (.contact > CI_ResponsibleParty)	Party responsible for the creation of the metadata
Metadata date stamp (M) (.dateStamp)	

Figure 13. **ISO 19115 Core metadata for geographic datasets ([ISO 01] p.15)**

This Core metadata set is quite similar to the Dublin Core metadata model, though it details some information specific to spatial data: spatial representation, spatial resolution and spatial reference system. These elements, as underlined by the authors of the standard, might be relevant to describe resources other than geographic data. They seem relevant to describe SPIRIT resources.

3.4. Models for (geographic) Web services

Resources on the Web might be services. There are standards to describe services, which do not necessarily relate to the Web bases ones. They aim at different operations:

- Services discovery.
- Services invocation.
- Services chaining.

Models are being designed to describe web services. The Web Services Description Language (WSDL) model is an implemented model depicted further. The ISO TC211 has produced a draft standard, ISO/DIS19119, on the topic "Geographic Information-Services". This standard is not limited to web services. The OpenGIS Consortium has released specifications for Web Map Services [OGC 02]. A *service* is described by:

- A type (in a list defined by OGC : Web Feature Server, Web Map Server, ..).
- A human-readable title.
- An abstract.
- Keywords (currently no controlled vocabulary has been defined).
- A point of contact.
- Access restrictions.

A service relies on several operations. An *operation* is described by:

- A name.
- A human-readable description.
- A list of parameters.
- For each parameter a list of possible valid values.

A service is also associated with a *content*. This content is a type of geo-data exposed by the service.

4. Implementation languages for metadata

This section depicts the XML language, used to encode information on the Web, and how metadata are to be implemented with it. XML is a recommendation of the W3C.

4.1. The XML language

XML documents

XML (eXtended Mark-up Language) is a mark-up language, similar to HTML. These languages form a specialisation of SGML (Standardised Generalised Mark-up Language). SGML uses tags to structure a document. A tag has a name, some tags may have attributes, and there are different types of tags: starting, closing, empty starting and empty closing.

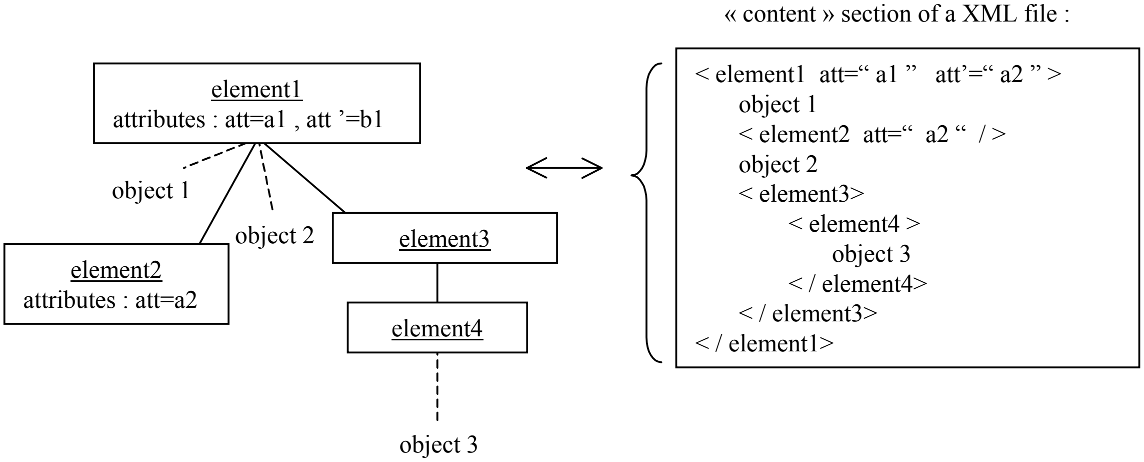
Example of starting tag : <name attribute1=value1 attribute2=value2>.

Mark-up languages are not dedicated to the storage of data but to the exchange of information. They are associated with a specific type of operation: parsing (i.e. reading). Every character in an XML document is to be parsed. If characters inside an XML document are not supposed to be parsed, for instance because they contain some characters that have a given meaning in XML, they should be written inside a tag delimited by "<![CDATA[" and "]]>".

Contrary to HTML, XML does not propose a fixed structure, i.e. a limited set of tags, dedicated to presentation. Rather it supports authoring of structures. Yet we'll see that XML imposes some syntactic rules, some of which are a meta-structure.

An XML document is organized in several parts. The prolog gives information about the models used in the XML document (version of XML, ISO character encoding system, and namespaces). It also entails operating instructions that give information about how to process the document with existing programs. The last part is the content itself. Each part is subject to specific syntactic rules.

XML imposes a specific meta-structure of a tree on the content, as illustrated on Figure 14. A non-empty tag always comes with a closing tag. In the content, all tags are nested in one element called the root, and all non-empty tags are nested within each other.



```
< element1  att=“ a1 ”   att’=“ a2 ” >
     object 1
     < element2  att=“  a2 “  / >
     object 2
     < element3>
          < element4 >
               object 3
          < / element4>
     < / element3>
< / element1>
```

Figure 14. **Tree structure of the content of a XML document.**

XML allows the description of information subject for some conditions. The description has a tree structure and its items are strings. These strings are:

- names of elements and attributes;
- values of attributes (always quoted);
- data between tags.

DTD and XML Schema

It is possible to define the tags used in an XML document, so that they may be reused in another XML document with the same structural form. This is done through writing a DTD (Document Type Definition), or through the definition of an XML Schema (XSD).

The DTD defines tags by their names and attributes; it also specifies the proper nesting of tags. A DTD may also introduce specific variables ("entities") that will be referred to in the XML documents written with respect to this DTD. When an XML document uses tags defined in a DTD, it is mentioned in its "prolog" section by the following tag: <!DOCTYPE *nameOfDTD* SYSTEM="*URL*of*DTD*">.

Another way of defining the tags used in an XML document is to write an XML Schema. This schema specifies new types for the attributes of an XML document. It is a mechanism similar to RDFS. A XMLS document may contribute to defining a Namespace. It is written in XML and has for root element the element "*xs*:schema".

Example :

```
<?xml version = "1.0"?>
<xs:schema  xmlns:xs1="URLduschéma 1"
            xmlns:xs2="URLdemonschéma 2"
            targetNamespace="URLdemonEspace"
            xmlns="URLdemonEspace">
….
</xs:schema>
```

An XML document may refer to several namespaces, and each namespace may be defined in one or more XSD. It also may refer to specific XSD.

XSD has many pros DTD have not:

- A DTD is written in mark-up language but not in XML, an XSD is an XML document.
- A DTD does not allow introducing new types of data, an XSD does.
- A DTD might not be reused in another DTD, nor can several DTDs be combined in an XML document, XSD can.

Operations associated to the XML structure

XML has a tree structure that may be handled in two general ways.

The first one consists of mapping the whole XML document into a generic tree object, whose elements (node,…) have attributes of the values of the document's tags. This generic tree can then be manipulated in general operations, e.g. displaying the structure.

The second one is the event-driven parsing of XML document, which relies on a correspondence between each tag and an operation that should be performed when reading it.

The XML structure is not optimized for storing and querying documents but solutions have been provided to map XML documents with a relational DBMS. A detailed presentation of these solutions may be found in [Van Zwol 02].

4.2. Implementation of metadata models in XML

XML may be used, as a machine-readable language to encode metadata structured after models presented in section 3.

Implementation of RDF

The encoding of RDF description relies on the specific namespaces : rdf and rdfs.

(rdf=http://www.w3.org./…..rdf-syntax-ns#, rdfs=" http://www.w3.org./…..rdf-schema#")

These schemas translate in XML the structure of RDF statements.

An RDF description can be encoded as a stand alone XML document. The root element of such document is then "rdf:description". Let us underline that it might also be an element defined as a subClass of rdf:description in a specific XSD. For instance, "rdfs:class", "daml:ontology".

An RDF description can also be encoded inside the resource it describes. XML document can entail some metadata about the document itself structured after the RDF model. The element enclosing these metadata in the document is rdf:RDF. It contains a sequence of elements named rdf:description.

The simplest way to write in XML a RDF statement [subject, predicate, object] is the following:

```
<rdf:description  rdf:about=URIofTheSubject>
    <predicate>   object     </predicate>
</rdf:description>
```

The attribute "about" is followed by a URL, if the resource exists. When the resource is created in the same XML document this attribute is replaced as follows: <*rdf*:description *rdf*:ID="Name">.

A description without *rdf*:about nor *rdf*:ID is said to describe an anonymous resource.

The object might be a literal or a resource. If it is a resource it may be detailed as a rdf:description element:

```
<rdf:description  about=URIofTheSubject>
    <predicate>
         < rdf:description  about=URIofTheObject>
          ….
         < /rdf:description>
    </predicate>
</rdf:description>
```

Usually, statements relating to the same subject are grouped inside the rdf:description tags.

```
<rdf:description  about=URIofTheSubject>
    <predicate1>      object1      </predicate1>
    <predicate2>
          <rdf:description about=URIofTheObject>
           …
          </rdf:description>
     </predicate2>
</rdf:description>
```

A lighter way of encoding RDF statements consists in integrating objects in the predicate tags:

```
<rdf:description  about=URIofTheSubject>
    <predicate1 "object1">
    <predicate2  rdf:resource="URIofTheObject">
</rdf:description>
```

If the object of predicate2 is a resource created in the document, its URI is usually of the form "#NameOfTheResourceInTheDocument".

An even lighter encoding consists in integrating in the *rdf*:description tag those statements whose objects are not resources:

```
<rdf:description  about=URIofTheSubject   predicate1="object1"    predicate3="object3">
     <predicate2  rdf:resource="URIofTheObject">
```

```
</rdf:description>
```

Implementation of DAML+OIL

The DAML+OIL model is encoded in XML as the schema RDF and RDFS are, using the *daml* namespace.

A DAML+OIL description is actually a specific rdf:description defined by the tag *daml*:Ontology.

The tag *daml*:Class provides for the definition of classes in such ontologies.

DAML introduces useful tags for rich descriptions and definitions of classes. An important tag is *daml*:Restriction that defines a class as the set of all things that satisfy the conditions enclosed in the tags *daml*:Restriction. For instance, the following XML lines state that a Person may have at most one occupation that is a FullTimeOccupation:

```
<daml:Class rdf:about="#Person">
  <rdfs:subClassOf>
    <daml:Restriction daml:maxCardinalityQ="1">
      <daml:onProperty rdf:resource="#hasOccupation"/>
      <daml:hasClassQ rdf:resource="#FullTimeOccupation"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

DAML also introduces specific types like *daml*:TransitiveProperty, and interesting properties like *daml*:disjointWith.

The Topic Map model also has been implemented in XML [TopicMaps.Org 00].

Implementation of the Dublin Core

A proposed recommendation of the DCMI (Dublin Core Metadata Initiative) defines an implementation of the DC elements in XML. This encoding is a specialisation of RDF encoding in XML.

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF PUBLIC "-//DUBLIN CORE//DCMES DTD 2001 11 28//EN"
"http://dublincore.org/documents/2001/11/28/dcmes-xml/dcmes-xml-dtd.dtd">
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description rdf:about="http://..../">
   <dc:title>Up My Street </dc:title>
   <dc:source rdf:resource="http://.../"/>
   <dc:creator>Lulu </dc:creator>
   <dc:format>HTML</dc:format>
```

<dc:identifier>http://www.UpMyStreet.com</dc:identifier>

<dc:description>A web site proposing thematical information about areas defined as neighbours of a street </dc:description>

</rdf:Description>

Implementation of services description

WSDL (Web Services Description Language) is an XML based language dedicated to describing the location of a Web Service and the operations it exposes.

A WSDL document has four types of elements depicted below.

1. The element <types> contains definition of types used by the service.

2. The element <message> contains the definition of a message that may be exchanged through the service interface. This element has child elements that are the parameters of the message. Each is depicted in an element <part name="…" type="…">. There can be several messages inside a WSDL document.

3. The element <portType> describes a Web service by the operations it exposes. An operation is described by its name and possible input and output. An input or output is itself described by a name and a message. There are four types of operations :

    - one-way : the operation may only receive messages. The operation has one child <input..>
    - request-response : the operation can receive a request and return a response. The operation has for sub-elements, in this order : <input…><output…>
    - solicit-response : the operation can send a request and wait for a response. It is described by the sub-elements, in this order : <output…><input…>
    - notification : the operation can send a message but does take no response. It is described by a sub-element <output..>

4. The element <binding> defines the message format and protocol details for each port. For instance, it might be a binding using SOAP.

4.3. A draft SPIRIT metadata model

To conclude this part, we sketch what a SPIRIT metadata model could look like. It is important not only to define **metadata elements** for a SPIRIT metadata model but also to define **controlled vocabularies** that will be the valid range domains for these metadata elements. The definition of these vocabularies may be linked to the definition of ontologies in SPIRIT (see WP3).

Selection of relevant elements from existing models

The Dublin Core and the ISO 19115 metadata models define useful elements to describe resources in SPIRIT.

Some elements of ISO 19115 are dedicated to the description of the metadata themselves:

- file identifier
- standard name and version
- character set
- language
- point of contact
- date.

These elements have no equivalent in the DC. Yet, in the XML implementation of the DC, some elements are always documented - file identifier, standard name and version and character set.

We focus on DC and ISO19115 elements that are dedicated to the description of the resources (see table below). Some elements are similar in their denomination but not necessarily in the definition of their range domain. Some are similar and grouped together. Yet the notion of elements is precisely defined in ISO, but not in DC, where elements are actually metadata categories that might contain several sub elements. Moreover, elements may have different range domains depending on the model they are defined in.

Generic name	ISO 19115 element	DC element	SPIRIT
Title	Name given to the resource (Free text)	Idem	Idem
Keywords	"topic category"	"Subject"	Idem (from controlled vocabularies that include ISO Codelist)
Summary	"Abstract" (Free text)	"Description" (may include : abstract)	Statements built with the ontologies of things and places
Date	Date	Date	Idem There may be several dates, each associated to a different event in the resource life-cycle
Coverage	Location (geographicBox or geographicIdentifier)	Coverage (spatial and temporal)	Location (ISO 19115 definition)
Language	Language of the intellectual content of the resource	Idem	Idem (there might be several translations)
Person responsible for the resource	Dataset responsible party	-	Persons responsible for the content, and for the resource
	-	"Creator"	
		"Contributor"	
Resource provider	"on-line resource"	"Publisher"	Idem
Source	"lineage statement" (free text)	"source"	Idem
Relation	-	"relation"	"Contains", "give access to",…
Type		Type	Idem
Format	Distribution format	Format	Idem
Spatial representation type	Spatial representation type		A Web site may also have such a type
Reference system	Reference system	-	A Web site may also have a reference system
Spatial resolution	Spatial resolution of the dataset	-	A Web site may also have a spatial resolution
Identifier		Identifier	
Character set	Character set		
Rights		Rights	

Figure 15. **Elements taken from the ISO 19115 core elements or the Dublin Core that would be useful to describe SPIRIT resources. These elements are dedicated to the description of the resource and not to the description of the metadata themselves.**

To illustrate the idea of using these metadata elements to depict SPIRIT resources, let us take practical examples.

In the website "upmystreet.com" the spatial representation type could be "low vector" because spatiality is represented by geographic objects (the streets). The reference systems could be the following: geographic identifier reference system (ZIP codes), geographic identifier reference system (names of towns).

In a weather forecast website, a resolution of value "big cities" would depict that the site provides weather forecasts in Europe by a map of temperatures for large European cities.

Vocabularies are needed to support the designation of the following items:

- geographic features like city, bridge, roads;
- geographic places with terms like "New York city", "the British museum", "Mont Blanc";
- thematic information like weather forecast, tourism, car renting, airport, and the relationships between these terms.

These vocabularies should be compliant with existing vocabularies, such as the ISO topic category codeList. Moreover, there are relationships between these vocabularies - the designation of a geographic place comprises the designation of the geographic feature it corresponds to.

Describing non static information resources

Web documents are not necessarily static resources, like a data set depicted in ISO19115 or a textual document depicted in DC. They are more complex information resources since they allow queries. We need to have a model of services offered by SPIRIT resources. For instance, we may use a model of services depicted by their input and corresponding output as shown in Figure 16.

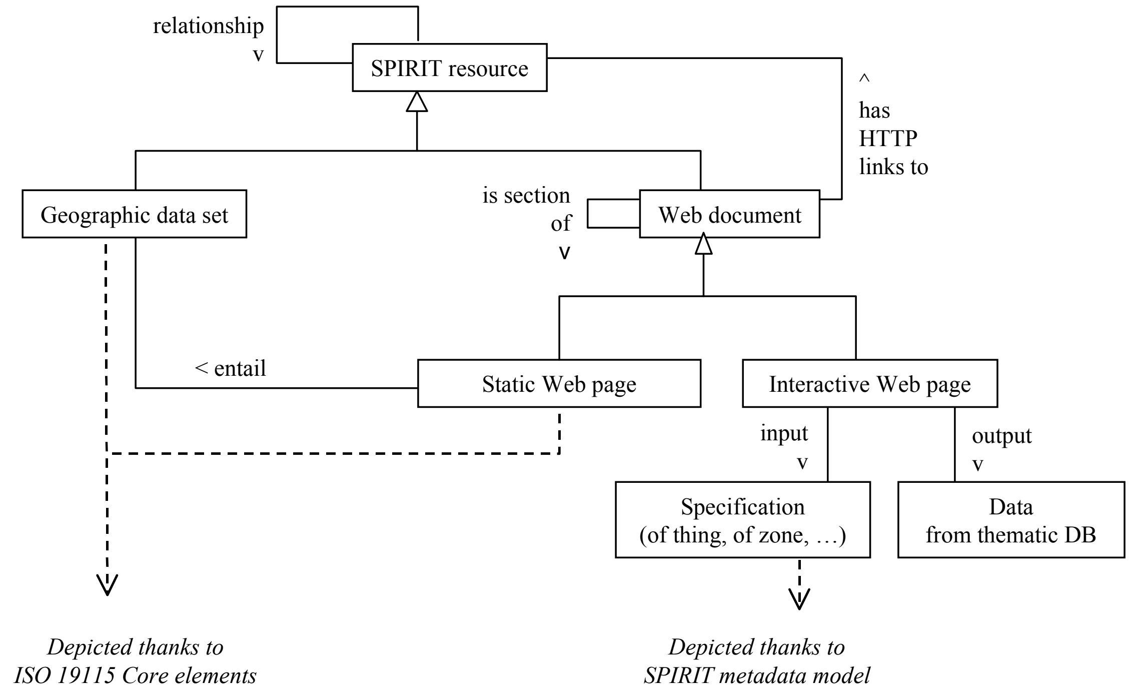


Figure 16. **A general model of SPIRIT resources and their description.**

Services maybe described at a generic level by the type of underlying function. Generic services could be:

- Locating = to provide location information about a thematic information (e.g. : a map of prehistoric sites in a given country, a site to find flats to rent in a given city).
- Context giving = to provide thematic information about a given location (e.g. : the site upmystreet.com).
- Mapping = to draw plans, maps.
- Routing = to tell how to go from one place to another.
- Access to geographic data sets = a portal diffusing geographic data sets.

This is not an exclusive partition. Services that perform mere mapping can answer a user's need for routing (if the site provides a map centred on a mail address, it helps you reach this address).

For instance, inputs and outputs of a locating service may be described with the following elements:

- Input:
  - *retrievedThematicInfo* which information may be located (ex: flats to rent), and how it might be specified (e.g. : field=price, operator=comprised between, value=(n euros, m euros)). The controlled vocabulary for these metadata is the ontology of things.
  - *contextZone* of the request : some sites allow specification of the spatial zone of research. For instance, on a Parisian site for flat renting you might specify 1 to 3

"arrondissements" where to look for a flat. The controlled vocabulary for this metadata is the ontology of places.

  - RequestFormat.
- Output:The output is a resource depicted by the metadata on table Figure 15.

Metadata dedicated to the format of the input and output could be useful to automatically perform the request and interpret the answer. This would be useful for the case, where SPIRIT would retrieve information from different Web sites and display all the results on one map. For instance, if the answer is a gif format, SPIRIT cannot further exploit it. But it might be a textual format that can be recovered in the HTML answer, provided the metadata depict which tags embrace it.

5. Conclusion

The preceding sections have presented tools that can be used in SPIRIT to organize and store a MD database. We also stressed some tasks that must be undertaken: specifying MD content, designing the model, implementing and acquiring MD. As a conclusion, we underline the main difficulties SPIRIT has to face with respect to metadata, and we sketch a draft model of metadata for SPIRIT resources.

The following questions concerning metadata need answering in the short term.

The first one is what operations do we intend to perform with metadata among these described in section 2.2:

- First step of resource discovery: metadata should be organized into a DB that would also be an index of resources.
- Relevance assessing: the content of metadata should hold information necessary to assess the relevance of the resource in the user context.
- Resources ranking.
- Resource displaying.

In SPIRIT objectives, it has been stated that spatial indexing of the resources will be provided based on a geographic footprint of the resources. Moreover, ranking will also be provided based on this footprint.

The next question is the design of the model. It depends on answering the preceding question and on the inference capacities of the ontologies.

In particular, the definition of the metadata "geographic footprint" requires careful design. Web documents do not have unique and unambiguous spatial extent as geographic data do.

Last, we have to define how to acquire these metadata. In particular, acquiring the geographic footprint of Web documents will be a complex process. The acquisition process will rely on the ontologies.

6. References

[Beckett et al. 01] Dave Beckett ,Eric Miller, Dan Brickley, Expressing Simple Dublin Core in RDF/XML, Dublin Core Metadata Initiative Proposed Recommendation, november 2001 http://dublincore.org/documents/dcmes-xml/

[Berners-Lee et al. 01] Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific American, May 2001

[DAML+OIL, 201] DAML+OIL (March 2001) Reference Description. Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. W3C Note 18 December 2001. Latest version is available at http://www.w3.org/TR/daml+oil-reference.

[Horrocks et al. 01] Ian Horrocks, Frank van Harmelen, Peter Patel-Schneider, Tim Berners-Lee, Dan Brickley, Dan Connolly, Mike Dean, Stefan Decker, Dieter Fensel, Pat Hayes, Jeff Heflin, Jim Hendler, Ora Lassila, Deb McGuinness, Lynn Andrea Stein, The DAML+OIL language, March 2001, http://www.daml.org/2001/03/daml+oil-index.html

[ISO 99]ISO, *ISO IEC 13250*, *Topic Maps*, December 1999

[ISO 00] ISO TC 211, *ISO/DIS 19112, Geographic information, Spatial referencing by geographic identifiers,* December 2000

[ISO 01] ISO TC211, *ISO DIS 19115 Geographic Information - Metadata*, 13 August 2001

[OGC 02] OpenGIS Consortium, *OpenGIS® Web Map Server Interfaces Implementation Specification,* Jeff de la Beaujardière (ed), January 2002

[OWL 02] OWL Web Ontology Language 1.0 Abstract Syntax, Peter F. Patel-Schneider Ian Horrocks, Frank van Harmelen (Eds.), W3C Working Draft, 29. July 2002, http://www.w3.org/TR/2002/WD-owl-absyn-20020729

[Sanderson 00] Mark Sanderson, *GLASS*, http://dis.shef.ac.uk/mark/GLASS/ , 2000

[TopicMaps.Org 00] TopicMaps.Org Authoring Group, *XML Topic Maps (XTM) 1.0 Core Deliverables*, December 2000, http://www.topicmaps.org/xtm/1.0/core.html

[Van Zwol 02] Roelof Van Zwol, *Modelling and searching web-based document collections*, Ph.D. Thesis of the University of Twente, Enschede, The Netherlands, 2002

[W3C 00] W3C, *Extensible Mark-up Language (XML) 1.0*, W3C Recommendation, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler (Eds), 6 October 2000 http://www.w3.org/TR/2000/REC-xml-20001006

[W3C 02] W3C, *RDF Vocabulary Description Language 1.0: RDF Schema*, W3C Working Draft, Dan Brickley, R.V. Guha (Eds), 30 April 2002 http://www.w3.org/TR/2002/WD-rdf-schema-20020430/

7. Appendix A. “ISO 19115 Draft Standard”.

In this appendix, we describe elements from the ISO19115 draft standard.

Identification information

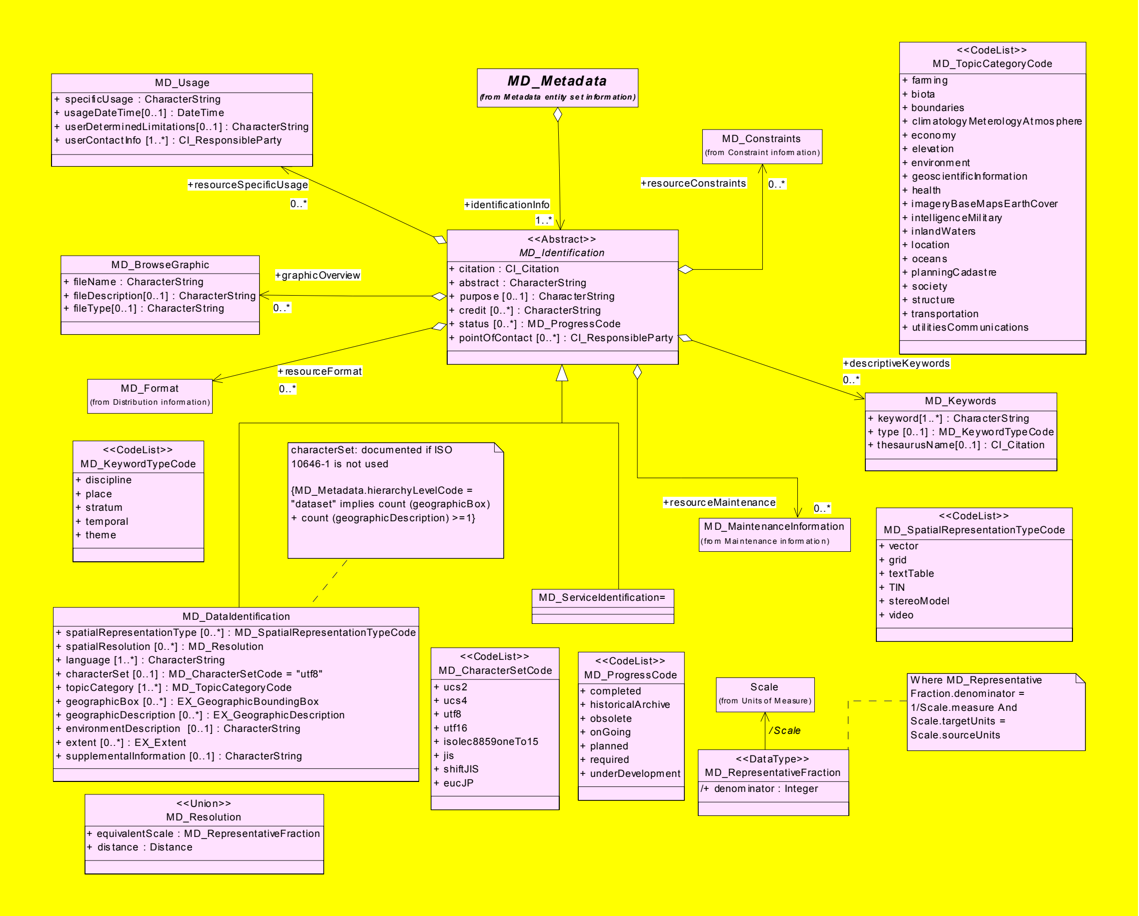

MD_Usage
+ specificUsage : CharacterString
+ usageDateTime[0..1] : DateTime
+ userDeterminedLimitations[0..1] : CharacterString
+ userContactInfo [1..*] : CI_ResponsibleParty
MD_Metadata
(from Metadata entity set information)
<<CodeList>>
MD_TopicCategoryCode
+ farming
+ biota
+ boundaries
+ climatologyMeterologyAtmosphere
+ economy
+ elevation
+ environment
+ geoscientificInformation
+ health
+ imageryBaseMapsEarthCover
+ intelligenceMilitary
+ inlandWaters
+ location
+ oceans
+ planningCadastre
+ society
+ structure
+ transportation
+ utilitiesCommunications
MD_Constraints
(from Constraint information)
+resourceSpecificUsage
0..*
+identificationInfo
1..*
+resourceConstraints
0..*
<<Abstract>>
MD_Identification
+ citation : CI_Citation
+ abstract : CharacterString
+ purpose [0..1] : CharacterString
+ credit [0..*] : CharacterString
+ status [0..*] : MD_ProgressCode
+ pointOfContact [0..*] : CI_ResponsibleParty
MD_BrowseGraphic
+ fileName : CharacterString
+ fileDescription[0..1] : CharacterString
+ fileType[0..1] : CharacterString
+graphicOverview
0..*
+resourceFormat
0..*
MD_Format
(from Distribution information)
+descriptiveKeywords
0..*
MD_Keywords
+ keyword[1..*] : CharacterString
+ type [0..1] : MD_KeywordTypeCode
+ thesaurusName[0..1] : CI_Citation
<<CodeList>>
MD_KeywordTypeCode
+ discipline
+ place
+ stratum
+ temporal
+ theme
characterSet: documented if ISO 10646-1 is not used
{MD_Metadata.hierarchyLevelCode = "dataset" implies count (geographicBox) + count (geographicDescription) >=1}
+resourceMaintenance
0..*
MD_MaintenanceInformation
(from Maintenance information)
<<CodeList>>
MD_SpatialRepresentationTypeCode
+ vector
+ grid
+ textTable
+ TIN
+ stereoModel
+ video
MD_ServiceIdentification=
MD_DataIdentification
+ spatialRepresentationType [0..*] : MD_SpatialRepresentationTypeCode
+ spatialResolution [0..*] : MD_Resolution
+ language [1..*] : CharacterString
+ characterSet [0..1] : MD_CharacterSetCode = "utf8"
+ topicCategory [1..*] : MD_TopicCategoryCode
+ geographicBox [0..*] : EX_GeographicBoundingBox
+ geographicDescription [0..*] : EX_GeographicDescription
+ environmentDescription [0..1] : CharacterString
+ extent [0..*] : EX_Extent
+ supplementalInformation [0..1] : CharacterString
<<CodeList>>
MD_CharacterSetCode
+ ucs2
+ ucs4
+ utf8
+ utf16
+ isolec8859oneTo15
+ jis
+ shiftJIS
+ eucJP
<<CodeList>>
MD_ProgressCode
+ completed
+ historicalArchive
+ obsolete
+ onGoing
+ planned
+ required
+ underDevelopment
Scale
(from Units of Measure)
/Scale
<<DataType>>
MD_RepresentativeFraction
/+ denominator : Integer
Where MD_Representative Fraction.denominator = 1/Scale.measure And Scale.targetUnits = Scale.sourceUnits
<<Union>>
MD_Resolution
+ equivalentScale : MD_RepresentativeFraction
+ distance : Distance

Figure 17. **ISO19115 Identification information ([ISO 01] p.20 Figure A.2)**

The MD_Identification entity may be specified (subclassed) as MD_DataIdentification when used to identify data and as MD_ServiceIdentification when used to identify a service.

Metadata used to describe a service are introduced in ISO 19119. ISO19115 detail metadata about data. More precisely, these metadata may describe geographic datasets, aggregations of datasets, features, or attributes of features.

The entity MD_Keywords is composed of three types of elements:

- The keywords are specific words attached to the resource, they are free CharacterString.
- The types are more generic words defining the domain of the resource, they belong to a CodeList.
- The thesaurusName is the reference to external thesaurus.

The element topicCategory of the entity MD_DataIdentification depicts the theme of features represented in the dataset.

A specific datatype defined in ISO19115 should be used to document the extent : EX_Extent. This type is specialised in three types: EX_BoundingPolygon (a geometric object), EX_GeographicBoundingBox, and EX_GeographicDescription (characterised by a geographicIdentifier) as shown Figure 18.

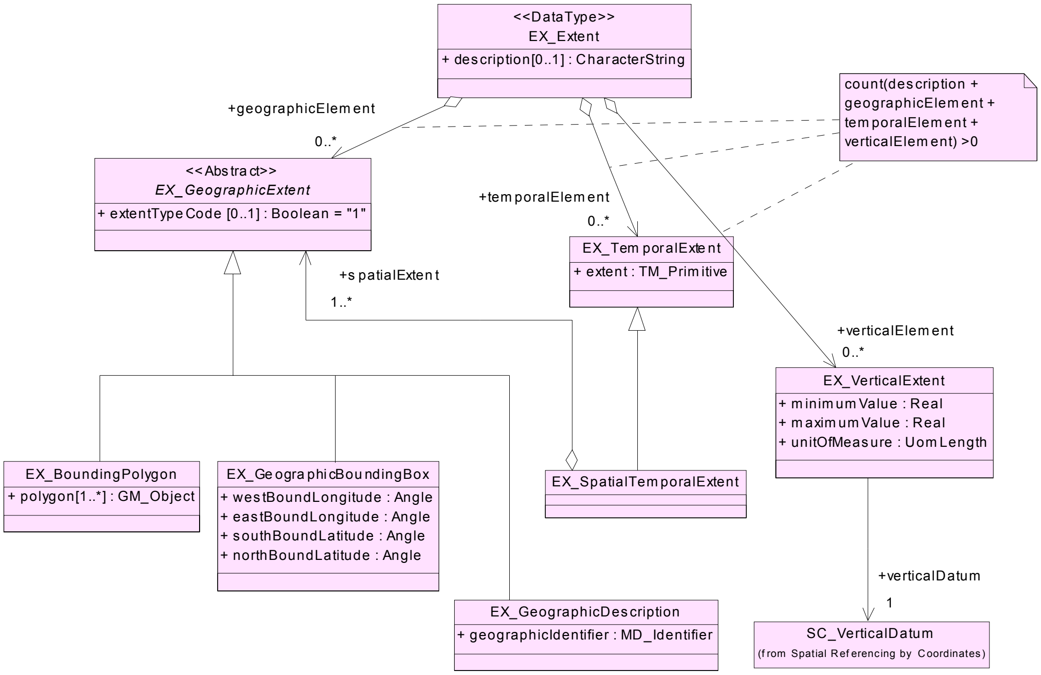

<<DataType>>
EX_Extent
+ description[0..1] : CharacterString
count(description + geographicElement + temporalElement + verticalElement) >0
+geographicElement
0..*
<<Abstract>>
EX_GeographicExtent
+ extentTypeCode [0..1] : Boolean = "1"
+temporalElement
0..*
EX_TemporalExtent
+ extent : TM_Primitive
+spatialExtent
1..*
+verticalElement
0..*
EX_VerticalExtent
+ minimumValue : Real
+ maximumValue : Real
+ unitOfMeasure : UomLength
EX_BoundingPolygon
+ polygon[1..*] : GM_Object
EX_GeographicBoundingBox
+ westBoundLongitude : Angle
+ eastBoundLongitude : Angle
+ southBoundLatitude : Angle
+ northBoundLatitude : Angle
EX_SpatialTemporalExtent
+verticalDatum
1
EX_GeographicDescription
+ geographicIdentifier : MD_Identifier
SC_VerticalDatum
(from Spatial Referencing by Coordinates)

Figure 18. **ISO 19115 Extent information datatype ([ISO 01] p. 33 figure A.15)**

Data quality information

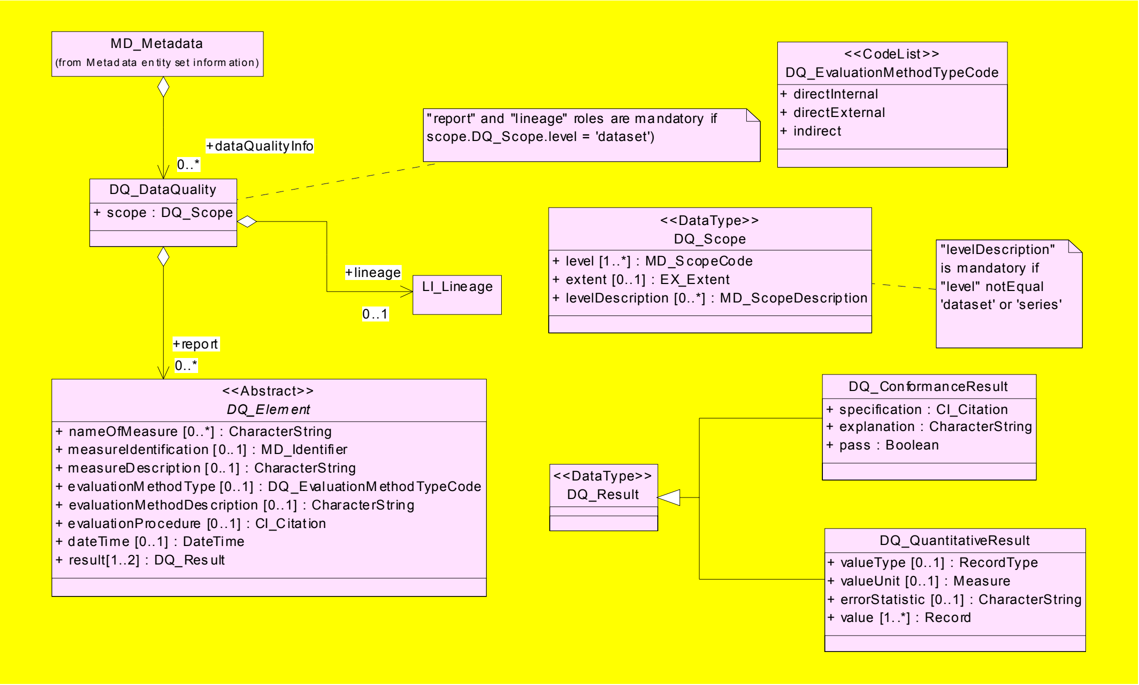

MD_Metadata
(from Metadata entity set information)
+dataQualityInfo
0..*
DQ_DataQuality
+ scope : DQ_Scope
"report" and "lineage" roles are mandatory if scope.DQ_Scope.level = 'dataset')
<<CodeList>>
DQ_EvaluationMethodTypeCode
+ directInternal
+ directExternal
+ indirect
+lineage
0..1
LI_Lineage
<<DataType>>
DQ_Scope
+ level [1..*] : MD_ScopeCode
+ extent [0..1] : EX_Extent
+ levelDescription [0..*] : MD_ScopeDescription
"levelDescription" is mandatory if "level" notEqual 'dataset' or 'series'
+report
0..*
<<Abstract>>
DQ_Element
+ nameOfMeasure [0..*] : CharacterString
+ measureIdentification [0..1] : MD_Identifier
+ measureDescription [0..1] : CharacterString
+ evaluationMethodType [0..1] : DQ_EvaluationMethodTypeCode
+ evaluationMethodDescription [0..1] : CharacterString
+ evaluationProcedure [0..1] : CI_Citation
+ dateTime [0..1] : DateTime
+ result[1..2] : DQ_Result
<<DataType>>
DQ_Result
DQ_ConformanceResult
+ specification : CI_Citation
+ explanation : CharacterString
+ pass : Boolean
DQ_QuantitativeResult
+ valueType [0..1] : RecordType
+ valueUnit [0..1] : Measure
+ errorStatistic [0..1] : CharacterString
+ value [1..*] : Record

Figure 19. **ISO19115 Data quality information ([ISO 01]p.22 Figure A.4.)**

The DQ_DataQuality entity contains the scope of the quality assessment. There may be several DQ-DataQuality entities in a set of metadata describing a dataset, some related to a series the dataset belongs to, some related to the dataset itself, some related to features in the dataset.

A DQ_DataQuality entity is an aggregate of LI_Lineage and DQ_Element.

For geographic data sets, the lineage information is very useful to assess the quality of a dataset. It is represented as an aggregation of sources and process steps involved in the production of the data set.

The DQ_Element may be specialised in one of the following types : completeness, logical consistency, positional accuracy. These types are defined as subclasses of the abstract class DQ_Element.

Reference system information

There may be several MD_ReferenceSystem entities in a metadata set.

A MD_ReferenceSystem entity is related to an entity RS_ReferenceSystem. This may be of several subtypes :

- a temporal reference system, which type is documented in ISO 19108 "temporal schema",
- a spatial coordinate reference system, which type is documented in ISO 19111 "spatial referencing by coordinates",
- a spatial reference system using geographic identifiers, which type is documented in ISO19112 "spatial referencing by geographic identifiers".

A spatial reference system using geographic identifiers is a structured collection of location types, instances of which have a geographic identifier [ISO 00]. Three important distinct notions are :

- location : an identifiable place in the real world.
- spatial reference : uniquely identifiable description of position in the real world.
- geographic identifier : spatial reference in the form of a label or code that identifies a position in the real world.

A geographic identifier identifies exactly one location. In return a location is identified by one or more geographic identifiers.

Content information

There may be several MD_ContentInformation entities in a metadata set. A MD_ContentInformation entity might be of two different subtypes, as shown Figure 20.

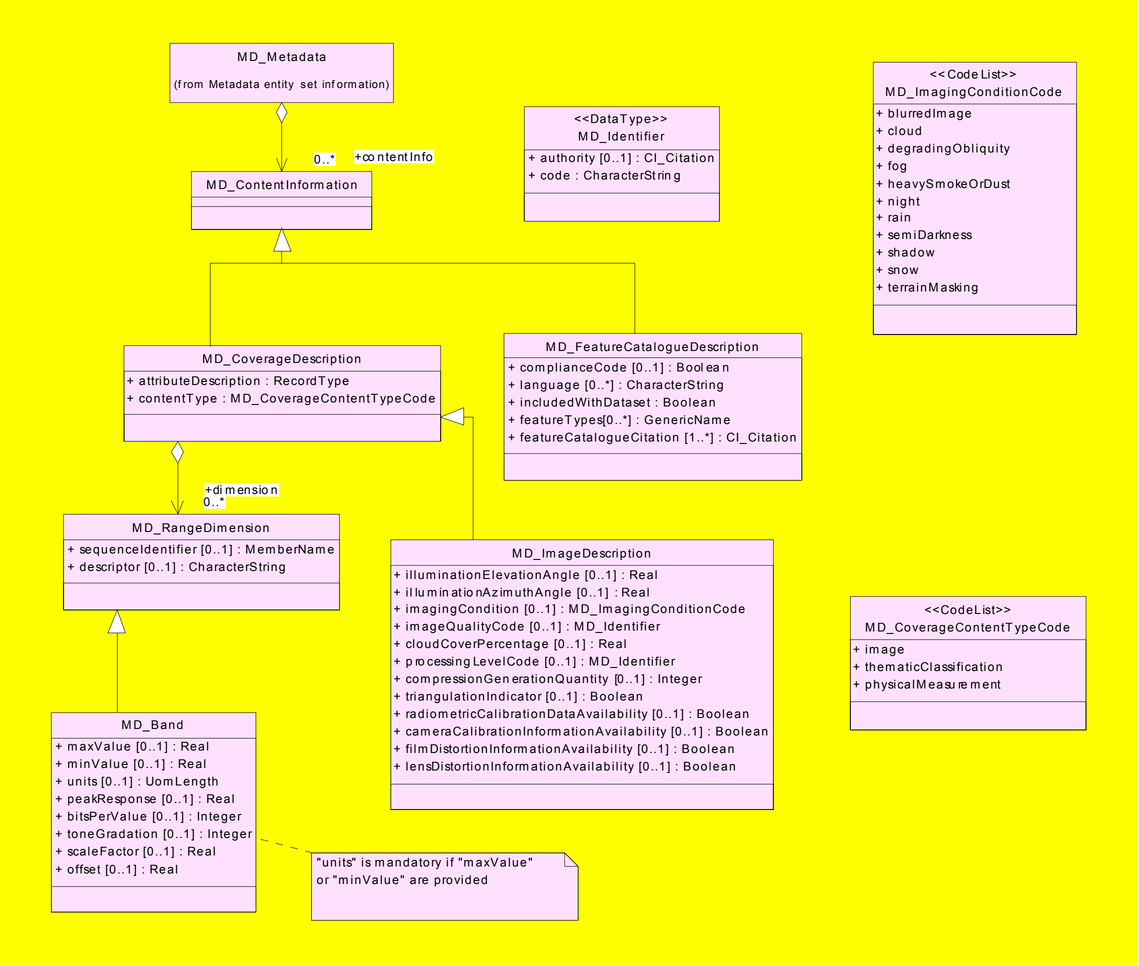

MD_Metadata
(from Metadata entity set information)
0..*
+contentInfo
MD_ContentInformation
<<DataType>>
MD_Identifier
+ authority [0..1] : CI_Citation
+ code : CharacterString
<<CodeList>>
MD_ImagingConditionCode
+ blurredImage
+ cloud
+ degradingObliquity
+ fog
+ heavySmokeOrDust
+ night
+ rain
+ semiDarkness
+ shadow
+ snow
+ terrainMasking
MD_CoverageDescription
+ attributeDescription : RecordType
+ contentType : MD_CoverageContentTypeCode
MD_FeatureCatalogueDescription
+ complianceCode [0..1] : Boolean
+ language [0..*] : CharacterString
+ includedWithDataset : Boolean
+ featureTypes[0..*] : GenericName
+ featureCatalogueCitation [1..*] : CI_Citation
+dimension
0..*
MD_RangeDimension
+ sequenceIdentifier [0..1] : MemberName
+ descriptor [0..1] : CharacterString
MD_ImageDescription
+ illuminationElevationAngle [0..1] : Real
+ illuminationAzimuthAngle [0..1] : Real
+ imagingCondition [0..1] : MD_ImagingConditionCode
+ imageQualityCode [0..1] : MD_Identifier
+ cloudCoverPercentage [0..1] : Real
+ processingLevelCode [0..1] : MD_Identifier
+ compressionGenerationQuantity [0..1] : Integer
+ triangulationIndicator [0..1] : Boolean
+ radiometricCalibrationDataAvailability [0..1] : Boolean
+ cameraCalibrationInformationAvailability [0..1] : Boolean
+ filmDistortionInformationAvailability [0..1] : Boolean
+ lensDistortionInformationAvailability [0..1] : Boolean
<<CodeList>>
MD_CoverageContentTypeCode
+ image
+ thematicClassification
+ physicalMeasurement
MD_Band
+ maxValue [0..1] : Real
+ minValue [0..1] : Real
+ units [0..1] : UomLength
+ peakResponse [0..1] : Real
+ bitsPerValue [0..1] : Integer
+ toneGradation [0..1] : Integer
+ scaleFactor [0..1] : Real
+ offset [0..1] : Real
"units" is mandatory if "maxValue" or "minValue" are provided

Figure 20. **ISO19115 Content information ([ISO 01] p.28 figure A.10)**

The attribute of the class MD_FeatureCatalogueDescription has the following meanings:

- attribute complianceCode specifies whether the feature catalogue complies with ISO19110,
- attribute includedWithDataset indicates whether the feature catalogue is included with the dataset,
- attribute featureTypes are the features from the catalogue that occur in the dataset. The type of this attribute is the generic name of the feature,
- featureCatalogueCitation is the complete bibliographic reference to one or more external feature catalogues.

Metadata extension information

Extended metadata elements should be defined using the schema Figure 21.

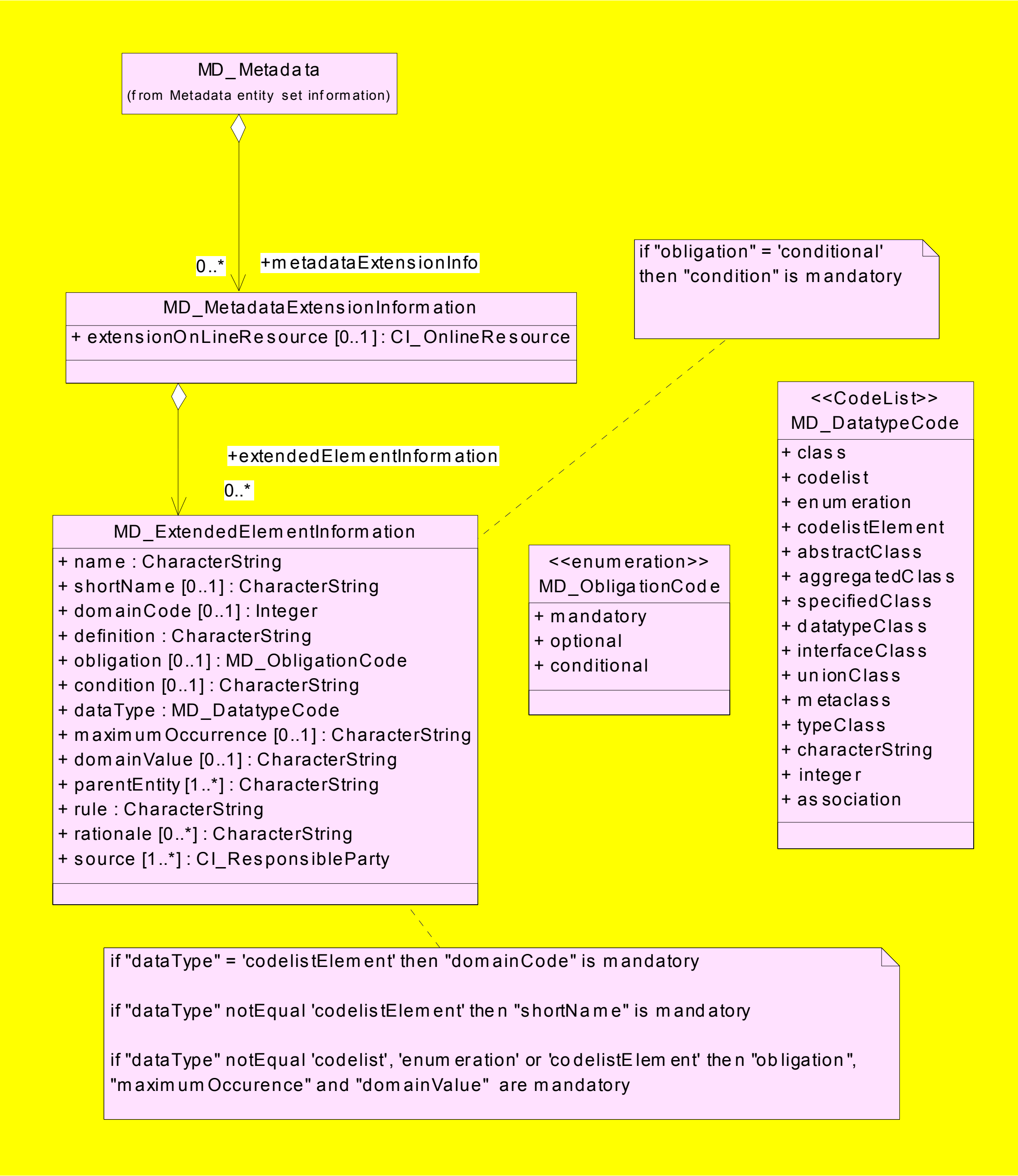

MD_Metadata
(from Metadata entity set information)
0..*
+metadataExtensionInfo
MD_MetadataExtensionInformation
+ extensionOnLineResource [0..1] : CI_OnlineResource
if "obligation" = 'conditional'
then "condition" is mandatory
<<CodeList>>
MD_DatatypeCode
+ class
+ codelist
+ enumeration
+ codelistElement
+ abstractClass
+ aggregatedClass
+ specifiedClass
+ datatypeClass
+ interfaceClass
+ unionClass
+ metaclass
+ typeClass
+ characterString
+ integer
+ association
+extendedElementInformation
0..*
MD_ExtendedElementInformation
+ name : CharacterString
+ shortName [0..1] : CharacterString
+ domainCode [0..1] : Integer
+ definition : CharacterString
+ obligation [0..1] : MD_ObligationCode
+ condition [0..1] : CharacterString
+ dataType : MD_DatatypeCode
+ maximumOccurrence [0..1] : CharacterString
+ domainValue [0..1] : CharacterString
+ parentEntity [1..*] : CharacterString
+ rule : CharacterString
+ rationale [0..*] : CharacterString
+ source [1..*] : CI_ResponsibleParty
<<enumeration>>
MD_ObligationCode
+ mandatory
+ optional
+ conditional
if "dataType" = 'codelistElement' then "domainCode" is mandatory
if "dataType" notEqual 'codelistElement' then "shortName" is mandatory
if "dataType" notEqual 'codelist', 'enumeration' or 'codelistElement' then "obligation",
"maximumOccurence" and "domainValue" are mandatory

Figure 21. **ISO 19115 Metadata extension information ([ISO 01] p.31 figure A.13)**

The attribute extensionOnLineResource depicts information about on-line sources containing the community profile name and the extended elements.

8. Appendix B. "Metadata Examples".

In this appendix, we discuss some examples of metadata in order to show what currently available metadata looks like.

8.1. Metadata about some IGN data sets

Hereafter is a model, close to ISO19115, used to implement metadata at the COGIT laboratory. To depict some data sets using this model, we had to gather information from paper documents and from DB management files. It was done on a small set of data because it was a long process.

The definition of effective metadata databases for IGN is still a work in progress.

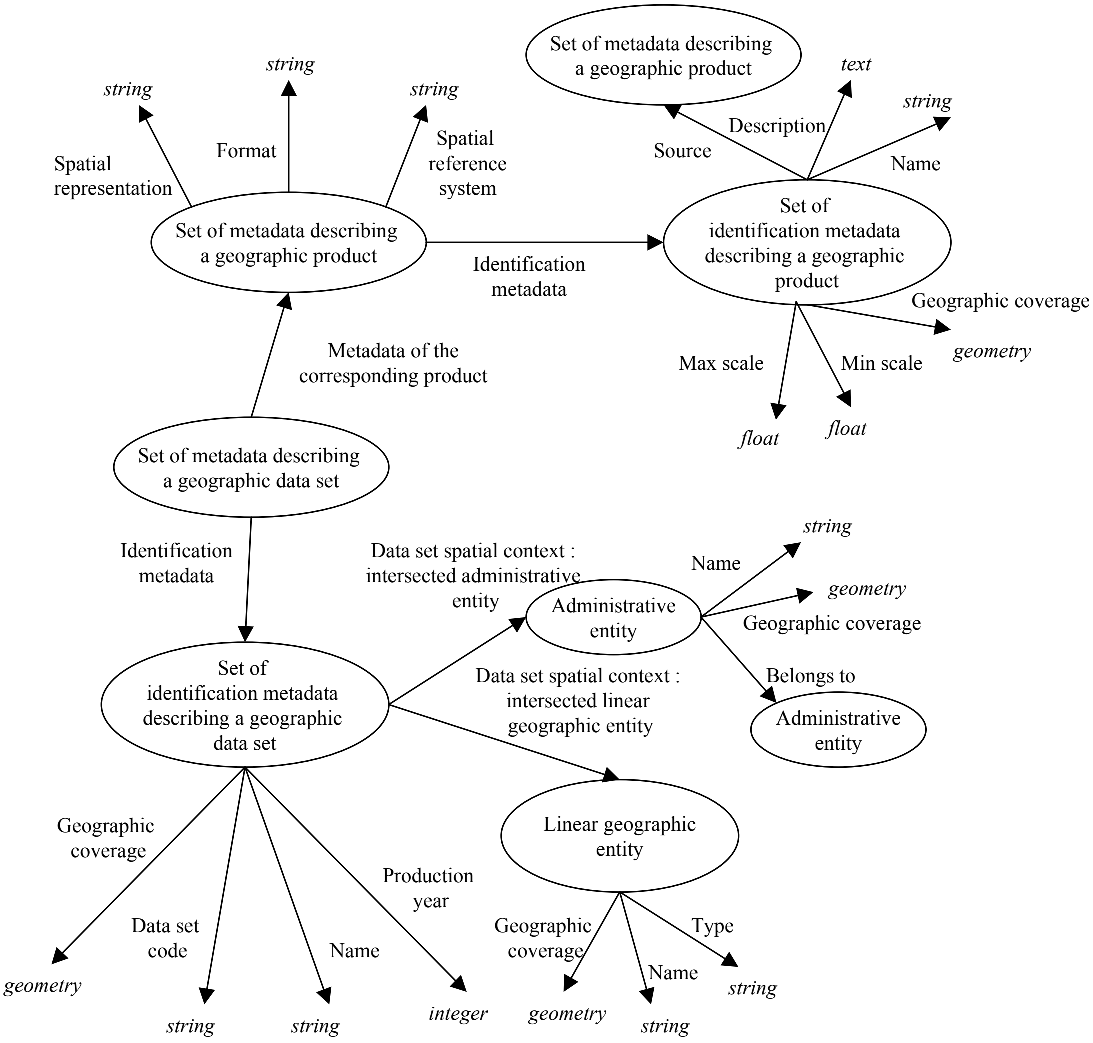


Figure 22. **Metadata model used to depict some IGN data sets.**

This generic schema may be used to describe specific data sets. For instance, we create hereafter an entity from the «Set of metadata describing a geographic data set » class : metadata of a map covering Beauvais city, taken from the IGN product « Série bleue ».

```
<rdf :RDF>
<rdf:Description about= « metadata_set_Beauvais »>
< Metadata of the corresponding product  rdf:resource= « metadata_product_SerieBleue »/>
< Identification metadata  rdf:resource= « IDmetadata_set_Beauvais »/>
</rdf :Description>

<rdf:Description about=  « metadata_product_SerieBleue »>
    <spatial representation  map / >
<format paper />
<spatial reference system lambert I />
<Product identification metadata
rdf :resource= «http://…../IDmetadata_ product_seriebleue.html »/>
</rdf :Description>

<rdf:Description about = « IDmetadata_product_SerieBleue »>
    <Source rdf :resource= « metadata_product_BDTopo » />
<Description Product constituting the root map of France, describing topographic details />
<Name Série Bleue />
<Geographic coverage (X1,Y1,X2,Y2,….) />
<Min scale 1/25000 />
<Max scale 1/25000 />
<rdf :seq>
<rdf :li> <Key word topography  /> </rdf :li>
<rdf :li> <Key word root map /> </rdf :li>
<rdf :li> <Key word hiking /> </rdf :li>
    </rdf :seq>
    <rdf :seq>
<rdf :li> <Theme Communication tracks /> </rdf :li>
<rdf :li> <Theme Public equipment  /> </rdf :li>
<rdf :li> <Theme Road equipment  /> </rdf :li>
<rdf :li> <Theme ground occupation /> </rdf :li>
<rdf :li> <Theme Administrative limits /> </rdf :li>
<rdf :li> < Theme Toponymy /> </rdf :li>
<rdf :li> < Theme Vegetation /> </rdf :li>
<rdf :li> < Theme Hydrography /> </rdf :li>
<rdf :li> < Theme Buildings /> </rdf :li>
<rdf :li> < Theme Diverse limits /> </rdf :li>
<rdf :li> < Theme Relief description /> </rdf :li>
    </rdf :seq>
</rdf :Description>

<rdf:Description about=  « IDmetadata_set_Beauvais »>
```

```
    < Geographic coverage (X1,Y1,X2,Y2,….) />
    < Data set code 2211E  />
    < Name Beauvais  />
    < Production year 1996 />

<rdf :seq>
<rdf :li> < Data set spatial context _intersected administrative entity
rdf :resource= « city/Beauvais»> </rdf :li>
<rdf :li> < Data set spatial context _intersected administrative entity
rdf :resource= « city/SaintLucien»> </rdf :li>
    </rdf :seq>
<rdf :seq>
<rdf :li> < Data set spatial context _intersected linear geographic entity
rdf :resource= « river/Therain »> </rdf :li>
<rdf :li> < Data set spatial context _intersected linear geographic entity
rdf :resource= « road/A16 »> </rdf :li>
    </rdf :seq>
</rdf :description>

<rdf:Description about= « city/Beauvais »>
    <Name Beauvais />
    <Geographic coverage (X1,Y1,X2,Y2….) />
    < rdf :seq>
<rdf :li> <BelongsTo rdf :resource= « department/Oise» /> </rdf :li>
<rdf :li> <BelongsTo rdf :resource= « region/Picardie » /> </rdf :li>
<rdf :li> <BelongsTo rdf :resource=  « country/France » /></rdf :li>
    </rdf :seq>
</rdf :description>

<rdf:Description about= « river/Therain  »>
    < Geographic coverage (X1,Y1,X2,Y2….) />
    <Name Therain />
    <Type River />
</rdf :description>
</rdf :RDF>
```

8.2. Metadata about Web documents

Here are metadata that are attached in the HTML source of Web pages.

- Metadata of the IGN Web site.

```
------------------------------
<BASE HREF="http://www.ign.fr/">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//FR" "http://www.w3.org/TR/REC-
html40/strict.dtd">
<HTML>
<HEAD>
<META NAME="Generator" CONTENT="IGN-SDOG-SG/GAZINOR2">
 <META NAME="Rating" CONTENT="General">
 <META HTTP-EQUIV="Cache-Control" CONTENT="post-check=86400,pre-
check=2592000">
 <META HTTP-EQUIV="Content-Language" CONTENT="fr-FR">
 <META HTTP-EQUIV="Vary" CONTENT="Content-Language">
 <META HTTP-EQUIV="Content-Script-Type" CONTENT="text/javascript">
 <META HTTP-EQUIV="PICS-Label" CONTENT='(PICS-1.1
"http://www.rsac.org/ratingsv01.html" l gen true comment "RSACi North America Server" by
"webmaster@ign.fr" for "http://www.ign.fr/" on "1997.10.31T09:36-0800" r (n 0 s 0 v 0 l 0))'>
 <META NAME="DESCRIPTION" CONTENT="Production et vente de cartes, photographies
aériennes, bases de données géographiques, France (organisme officiel) et export.
Prestations de formation, conseil, géodésie, topographie et cartographie">
 <META NAME="ABSTRACT" CONTENT="Production et vente de cartes, photographies
aériennes, bases de données géographiques, France (organisme officiel) et export.
Prestations de formation, conseil, géodésie, topographie et cartographie">
 <META NAME="KEYWORDS" CONTENT="ign, IGN, institut, géographie, géographique,
France, cartographie, topographie, géodésie, geodesie, orthophotographie, geographique,
geographie, sciences, terre, base, bases, données, donnees, sig, SIG, gps, GPS,
télédetection, navigation, imagerie, spatiale, carte, photographie, aérienne.">
</HEAD>
<BODY…….>
[The resource itself]
</BODY>
</HTML>
------------------------------------
```

When you follow some link of the page, for instance the one labelled "Acheter en ligne des cartes, des photos ... ", you reach a Web page with new metadata encoded in the HTML source:

```
<META NAME="DESCRIPTION" CONTENT="Découvrez la géographie avec les cartes, (leur
histoire, leur fabrication et leur utilisation avec un GPS) les plans de ville, atlas, guides,
ouvrages cartographiques, ainsi que les photos aériennes.">
 <META NAME="ABSTRACT" CONTENT="Découvrez la géographie avec les cartes, (leur
histoire, leur fabrication et leur utilisation avec un GPS) les plans de ville, atlas, guides,
ouvrages cartographiques, ainsi que les photos aériennes.">
```

```
<META NAME="KEYWORDS" CONTENT="carte, photographie, aérienne, guide, atlas, plan, ville, ouvrage, cartographique, géographie, photo, cartes, aériennes, aerienne, guides , ign, IGN">
```

In this example, the encoded metadata are not enough. For instance, they do not hold information about the information that the user can find on this site :

- to learn what is : maps production, geodesy, RGF93…
- to buy : maps, data sets
- to find : the coordinates of a commune, coordinates transformation tools…

- Metadata of a UK site to access geographic data :

```
<html>
<head>
<title>Ask GIraffe</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
```

In this example there are no encoded metadata. Even the title does not give clue. Yet the Web site is very efficient and interesting.

- Metadata of a UK site to find information about a given area :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd">
<html>
<head>
    <title>UpMyStreet &#187; The real-life guide to your neighbourhood</title>
    <meta http-equiv="content-type" content="text/html; charset=iso-8859-1">
    <meta name="keywords" content=", postcode, jobs, cars, motors, homes, houses for sale, property for sale, property, nearest, up my street, WAP, wap.upmystreet.com, restaurants, pubs, taxi, schools, classifieds, business finder, Crimes, Crimes solved, Primary schools, league tables, local bargains, community, what's on, things to do, take away, postcode, area profiles, CACI, ACORN, Health, upmystreet.com, local council, MP, MEP, Top Schools, Education, Analysis, house prices, property prices, school performance tables, GCSE results, A level results, council tax, council tax rates, crime statistics, england, statistics, uk, wales, Great Britain, britain, local information, government information, lender, mortgage rates, local authority, local education authority, buying a house, house buying, estate agents, moving house, house, flat, statistics, democracy, cars for sale, local jobs, Find My Nearest, moving homes, on the move, new home, street maps, local maps, maps, uk maps, regional maps">
    <meta name="description" content="All the facts about your area, including Property prices, school results, council contacts, crime statistics, your MP, street maps, a local business finder - Find My Nearest and classified services including used cars, jobs, items for sale and local property listings">
….
</head>
….
```

There are lots of metadata in the same field that could be restructured. It is not explicit, except in the title, that the user specifies in his query the street where he lives.

- Metadata of mappy, a web site for routing and locating :

```
<html><head>
<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<base href="http://www3.mappy.com/sidDUUoKTsXfx55Xg8o/">
<meta name="description" content="Mappy.fr (ITI), votre guide routier : calcul d'itinéraires pour des voyages en Europe et en France, plans de paris et des grandes villes, plan d'accès">
```

<meta name="abstract" content="Mappy.fr (ITI), votre guide routier : calcul d'itinéraires pour des voyages en Europe et en France, plans de paris et des grandes villes, plan d'accès">

<meta name="keywords" content="adresse,atlas,calcul itinéraires,carte,carte Bordeaux,carte France,carte Grenoble,carte interactive,carte Lille,carte Lyon,carte Marseille,carte Nantes,carte Nice,carte Paris,carte Strasbourg,carte Toulouse,cartes,cartographie,cartographique,code postal,communes,Deplacement,deplacement Europe,Europe,France,guide,guide routier,guide touristique,hôtels,Itineraire,itineraire France,localiser,mappy,paris,plan Bordeaux,plan Lille,plan Lyon,plan Marseille,plan Nantes,plan Nice,plan Paris,plan Rennes,plan Strasbourg,plan Toulouse,plans d'accès,Proximite,region France,region Paris,restaurant,route,routier,rue,rue Paris,se rendre,tourisme,ville,Voyage">

<title>Guide routier</title>

It is not specified what query the user might express and that two different types of service are provided: drawing a map or determining a route.