Semantic integration of thematic geographic information in a multimedia context

by

Antonio Navarrete Terrasa

Ph.D. Thesis

Doctorate in Computer Science and Communication
Department of Technology

Advisor: Dr. Josep A. Blat Gimeno

Universitat Pompeu Fabra

Barcelona, June 2006

Dipòsit legal: B.48968-2006
ISBN: 978- 84-690-3513-9

A Chus

Acknowledgments/Agradecimientos/Agraïments

Esta tesis ha supuesto un largo camino que empezó hace ya casi siete años. No sé cuántas veces habré soñado el llegar a este momento de poder daros las gracias a todos los que me habéis apoyado durante este tiempo.

Vull començar donant les gràcies a en Josep Blat, el meu director, per haver-me donat l'oportunitat d'iniciar aquesta aventura que va començar un dia en què em va oferir venir a Barcelona a participar en l'inici dels Estudis d'Informàtica a la Pompeu. Moltes gràcies Josep per haver confiat en mi i per haver-me deixat fer la tesi que jo volia, donant-me tota la llibertat i facilitats.

També vull donar les gràcies a en Maurici Ruiz. Gràcies per convidar-me a fer una estada de dos mesos al Servei de SIG de la Universitat de les Illes Balears durant els quals vaig aprendre moltíssim sobre la part pràctica d'aquest món dels SIG. Gràcies per les converses que vàrem tenir llavors i després, durant bona part del procés de la tesi. I gràcies també a na Jero i en Tomeu.

Muito obrigado a António Câmara e as pessoas do GASA. Unfortunately I have forgotten almost all the Portuguese I learnt and I cannot write theses words in your language. Many thanks to Professor António Câmara for inviting me to Lisbon, where I could meet a group of very nice people and I could participate in an encouraging environment. Thanks also to the guys of GASA and Y-dreams, especially to Miguel Remédio, João and Manuel, that introduced me to the Portuguese life and made me enjoy your wonderful city.

Vull també donar les gràcies a tots els companys i amics del Grup de Tecnologies Interactives amb els quals he compartit no només projectes, sinó molts moments bons i dolents durant tots aquests anys. Gracias a Dani, Jesús I., Claudia y *en* Dai a los que ya conocía de Mallorca, y también a Ginés, Jesús V., Sofía, Sergio y muy especialmente a Rocío y Alejandro (¡ahora te toca a ti!). Gracias también a Nadjet por sus correcciones con el inglés. I a la resta d'amics del Departament, sobre tot a Marcelo, Jesús B. i Juan. El millor d'haver vingut a Barcelona ha estat conèixer-vos a tots vosaltres.

También quiero acordarme de Virginia. Muchas gracias porque tú fuiste quien me animó a cruzar el charco para empezar esta aventura.

Y por supuesto, muchas gracias a mi familia: a mis abuelos, a la Tita y Carlos, a mis hermanos Sebastián y Celia, y sobre todo a mis padres. Cuántas veces me habréis preguntado si me faltaba mucho para terminar la tesis. Pues, ¡ya está acabada! Muchas gracias por vuestro apoyo. Y gracias también a mi "nueva" familia: Ruperto, Isabel, Isa, Xisco, Nieves, Miguel Ángel y a los más nuevos todavía, Luis y Eva; y también a Cris y Juanlu.

Y dejo para el final a la persona más importante de todas. Muchas gracias Chus por haber estado ahí en los momentos más difíciles, que no han sido pocos. Y también en los buenos, que han sido muchos más. Sin ti esta tesis no hubiera acabado nunca. No tengo palabras para agradecerte cuánto me has ayudado.

Abstract

Geographic datasets represent reality through a set of thematic entities that are often not precisely defined and that may be understood in different ways by different subjects. In this context, integrating geographic information from diverse datasets presents a significant challenge from the semantic point of view. A solution to this problem based on ontologies and Description Logic is proposed in this thesis. A semantic framework has been defined whose core is an ontology that represents the thematic concepts in a repository of datasets as well as their relations. This ontology is built from the application ontologies of the datasets being inserted in the repository through a merging process. A semi-automatic merging method is proposed in this work, where three different mapping algorithms have also been developed to generate a list of suggested mappings that a domain expert can accept or modify. This semantic framework supports the definition of semantic services that go beyond the functionalities provided by current catalogues of geographic information. In particular, one of the three semantic services defined in this thesis consists in the integration of the thematic information from different datasets in a new one. Finally, the semantic framework and services have been used in the context of indexing and retrieving geo-referenced multimedia elements (still images and video sequences) based on their thematic geographic content.

Resumen

Los *datasets* geográficos representan la realidad mediante un conjunto de entidades temáticas que a menudo no están definidas de una manera precisa y que diferentes sujetos pueden entender de distintas formas. En este contexto, la integración de información geográfica proveniente de diversas fuentes presenta un importante reto desde el punto de vista semántico. En esta tesis se propone una solución a este problema basada en ontologías y Lógica de Descripción. Se ha definido un marco semántico cuyo núcleo es una ontología que representa los conceptos temáticos en un repositorio de *datasets*, así como las relaciones entre dichos conceptos. La ontología se construye mediante un proceso de fusión (*merging*) de las ontologías de aplicación de los *datasets* que se han insertado en el repositorio. En este trabajo se propone un método semi-automático de *merging*, para el que se han desarrollado tres algoritmos diferentes con el objetivo de generar una lista de sugerencias de operaciones de mapeado que un experto podrá aceptar o modificar. Este marco semántico permite la definición de servicios semánticos que van más allá de las funcionalidades que los actuales catálogos de información geográfica ofrecen. En concreto, uno de los tres servicios semánticos definidos en esta tesis consiste en la integración en un nuevo *dataset* de información temática proveniente de diversas fuentes. Finalmente, el marco semántico y sus servicios se utilizarán en un sistema de indexación y recuperación de elementos multimedia geo-referenciados (imágenes estáticas y secuencias de vídeo) a partir de su contenido geográfico temático.

Resum

Els *datasets* geogràfics representen la realitat mitjançant un conjunt d'entitats temàtiques que sovint no estan definides d'una manera precisa i que diferents subjectes poden entendre de distintes formes. En aquest context, la integració d'informació geogràfica provinent de diverses fonts presenta un important repte des del punt de vista semàntic. En aquesta tesis es proposa una solució a aquest problema basada en ontologies i Lògica de Descripció. S'ha definit un marc semàntic el nucli del qual és una ontologia que representa els conceptes temàtics en un repositori de *datasets*, així com les relacions entre aquests conceptes. L'ontologia es construeix mitjançant un procés de fusió (*merging*) de les ontologies d'aplicació dels *datasets* que s'han inserit al repositori. En aquest treball es proposa un mètode semi-automàtic de merging, per al qual s'han desenvolupat tres algorismes diferents amb l'objectiu de generar una llista de suggeriments d'operacions de mapejat que un expert podrà acceptar o modificar. Aquest marc semàntic permet la definició de serveis semàntics que van més enllà de les funcionalitats que els actuals catàlegs d'informació geogràfica ofereixen. En concret, un dels tres serveis semàntics definits en aquesta tesi consisteix en la integració en un nou *dataset* d'informació temàtica provinent de diverses fonts. Finalment, el marc semàntic i els seus serveis s'utilitzaran en un sistema d'indexació i recuperació d'elements multimedia geo-referenciats (imatges estàtiques i seqüències de vídeo) a partir del seu contingut geogràfic temàtic.

Table of contents

List of figures

List of tables

1 Introduction and objectives

In the first section of this chapter we describe the main elements of the problem that motivates this thesis. In the second section we introduce the main aspects of our approach to achieve semantic integration of thematic geographic information. The objectives of this thesis are formulated in the third section. Finally, the structure of this thesis is presented in the fourth section.

1.1 Description of the problem

In this section we describe the problem that motivates this thesis. As the title of the thesis suggests, the main part of the problem that we address is the semantic integration of thematic geographic information. In subsection 1.1.1 we define semantic integration, also usually referred to as semantic interoperability, of geographic information (GI). We also formulate the different types of semantic heterogeneities that will be addressed in our work. In subsection 1.1.2 we briefly discuss what we understand by thematic geographic information and some of its specific traits. As the title also suggests, a second aspect of the problem refers to a multimedia context where semantic integration of thematic geographic information will be applied. This context is briefly described in subsection 1.1.3.

1.1.1 Semantic integration of Geographic Information

During the 1970s and 1980s the organizations that needed to deal with geographic/cartographic information developed their own Geographic Information Systems (GIS), which almost always were based on proprietary commercial products. These organizations collected their data that was rarely acquired from digital sources, and that was seldom shared with other organizations. Using the metaphor of Bishr in (Bishr 1998), these early GIS can be considered as “islands of information”. In the last 10-15 years, this situation has dramatically changed. The Internet enables users to share information and to avoid the inefficient and redundant old ways of working. In this new context, isolated and monolithic proprietary GIS have evolved into the so-called *Interoperable GIS* (Goodchild et al. 1999).

From a software engineering perspective, interoperability means open systems that can integrate software components from different developers. This has enabled new software products to enter the market, breaking the strong tie that bounded organizations to their GIS vendors. In particular, the open source movement has been very important in the last few years. Numerous open source tools have recently appeared related to different aspects of geographic information: GIS, spatial databases, web map servers, analysis tools, among others[1]. Open GeoSpatial Consortium[2], OGC, formerly known as Open GIS Consortium, has had a prominent role promoting interoperability in geospatial software by developing specifications at different levels. This enables developers to build software by integrating different modules that are compliant with OGC specifications.

From an information perspective, the word interoperability points out the need to share information. While information was produced and maintained locally, it was sufficiently unambiguous for its respective small community. But when this information has been produced by others this is not true any more, and consequently we have to deal with heterogeneous information. According to (Bishr 1998), people in the GI community recognize facts in the real world, categorize them creating a mental model and represent them in a dataset. From this process we can identify three different types of heterogeneity. *Syntactic heterogeneity* refers to the way the data is encoded in datasets: datasets may have different data formats, their spatial data may be represented through different models (vector or raster), or they may refer to different spatial coordinate systems. The Geographic Markup Language (GML) (OGC 2004) is an OGC specification oriented to provide a common format for representing geographic information avoiding syntactic heterogeneity. *Structural heterogeneity* refers to the way the mental model is represented in the dataset, for instance using different thematic attributes. Metadata describes the structure of the representation schema in the dataset and is an important tool to deal with structural heterogeneity. Finally, *semantic heterogeneity* refers to the fact that different agents (persons or organizations) may use different mental models, i.e. they categorize the real world in different ways. Note that these categories correspond to thematic concepts, and consequently we can observe that semantics is mainly related to the thematic component of the geographic information.

Semantic interoperability, or *semantic integration*, refers to the mechanisms that enable agents to share and integrate information from different sources overcoming semantic heterogeneity. The problem of semantic integration has motivated an important research area in Geographic Information Science (GISc), with a prominent presence in scientific journals and conferences.

Solutions for semantic integration usually rely on ontologies since they provide a formal specification of the mental model underneath datasets. But apart from defining ontologies, semantic services are also necessary to integrate data from different datasets. This usually means dealing with different ontologies or mental models. Logic in which ontologies are founded on also provides the basis for these services.

We can identify three levels of semantic heterogeneity, mainly based on the classification provided by (KnowledgeWeb Consortium 2005):

1. The *syntactic level* refers to the fact that different ontologies may be expressed in different languages as OWL or KIF.

[1] See http://www.freegis.org and http://www.opensourcegis.org for wide catalogues of free/open source geospatial tools

[2] http://www.opengeospatial.org

2. The *terminological level* comprises mismatches in names of concepts. Examples of mismatches at this level are synonymy, polysemy, different languages (English, Spanish, etc.) or derivatives (prefixes or suffixes).
3. The *conceptual level* includes mismatches related to the content of the ontology. There are two main types of conceptual discrepancies: *metaphysical* differences, which refer to how the world is "broken into pieces" (i.e., what entities, properties and relations are represented in an ontology); and *epistemic* differences, which are related to how we understand these entities, i.e. what assertions are made about them. In particular, metaphysical discrepancies comprise three types of differences: coverage (different ontologies cover different portions of the real world), granularity (one ontology provides a more detailed description of the same concepts than the other) and perspective (two ontologies are the result of observing the real world from different points of view, as it is the typical case of different disciplines).

The conceptual level is clearly the most complex one. We are particularly interested in the terminological (except multilinguality) and the conceptual levels, covering both metaphysical and epistemic differences. Multilinguality will not be addressed in this thesis, and we will concentrate on concepts expressed only in English. Regarding the syntactic level, we avoid possible discrepancies by representing all our ontologies in OWL. A conversion from/to other ontology languages is out of the scope of our work.

Semantic interoperability is very related to the notion of the Semantic Web. This is an initiative aiming at representing the information available over the web in a way understandable not only by humans but by machines too; this will enable machines to use the web not only to display information as today they do, but for more "intelligent" purposes, supporting sharing and reusing data across different applications or communities. Tim Berners-Lee, the creator of the world wide web and one of the main promoters of this initiative, conceives the 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. 2001). Egenhofer points out the need to define the so-called Semantic Geospatial Web (Egenhofer 2002). According to Egenhofer, this would be based on a framework comprising multiple spatial and thematic ontologies, as well as on a canonical form for specifying geospatial queries. Furthermore, in the last years there has been a growing importance of service-oriented architectures. In this context, there has also been an initiative aiming at formally specifying geoprocessing services, which perform some kind of computation or analysis on the geospatial data. See (Lutz 2005; Lutz and Klien 2006) and the projects meanInGs[3] and ACE-GIS[4] for more details.

Also directly related to interoperability (although unfortunately not semantic) is the concept of spatial data infrastructure (SDI). An SDI is defined as the infrastructure that provides the framework for the optimization of the creation, maintenance and distribution of geographic information at different organizational levels (regional, national or global) and involving both public and private institutions (Nebert 2001). One of the main services that an SDI provides is dataset discovery. Metadata of datasets, compliant with standard metadata schemas, can be searched by catalogues compliant to

[3] http://www.meanings.de

[4] http://www.acegis.net

the OGC specification (more details can be found in Chapter 3). However, metadata standards only provide simple semantics through keywords, which are the basis of the services for finding datasets offered by catalogues. But this does not address the problem of semantic heterogeneity, as we discuss in more depth in Section 1.2.

1.1.2 On Thematic Geographic Information

We can differentiate three main components in geographic information: spatial, thematic and temporal. Regarding the spatial component, it is clear that geographic information is referenced to a space (the Earth's surface) by means of geographic coordinates. The thematic component refers to the information of the physical or abstract entities represented in that geographic space. Finally, these entities are not static in time: they may move in space or the value of their thematic properties may change. In our work we deal with the spatial and thematic components and do not consider their time variations. For simplicity, the term *geospatial* is commonly used to refer to the spatial component of the geographic information. Furthermore, wherever we use the term spatial here we refer to the geographic space, and consequently to the spatial component of the geographic information. Likewise, we will indistinctly use the terms *geothematic information*, *thematic geographic information,* or simply *thematic information* to denote the thematic component of geographic information. Note that the problem of semantic interoperability described above is mainly referred to thematic geographic information, as the title of this thesis indicates.

The spatial information can be structured according to two different models: *object-based* and *field-based*. On the one hand, according to (Worboys 1995), the field-based model treats geographic information as a spatial distribution that may be formalized as a mathematical function from a spatial framework (for example, a regular grid) to an attribute domain; examples of fields are topographic altitude, temperature or rainfall. On the other hand, the object-based model treats the information space as populated by discrete, identifiable entities, each with a geo-reference. Examples of objects are roads or buildings. The concept of field is usually related to *fiat boundaries*. Fiat, as opposed to *bona-fide*, boundaries are those that do not correspond to genuine discontinuities in the world and "exist only in virtue of the different sorts of demarcations effected cognitively and behaviorally by human beings" (Smith and Mark 1998). According to (Winter 1998), objects have the property of being located in space, and consequently they have its location as an attribute; fields are continuous phenomena and characterize space by properties related to location. Note that this dichotomy between field and object is very related to the two types of structures for representing space in GIS: raster and vector. While a raster representation divides the space in equal-size cells, vector datasets represent the geometry of the geographic features. Fields are usually represented through raster datasets, while object-based information is usually represented through vector datasets.

Regarding the thematic component, thematic variables (also called thematic attributes especially when talking about datasets) can be classified as either *qualitative* or *quantitative* according to their range of values. Quantitative variables have a range of numeric values that can be either continuous, as in the case of temperature or precipitation, or discrete, as in population. The value can be either a direct measure, as it would be the case in population or temperature, or it can be calculated from it, as in population density or per capita income. Furthermore, quantitative variables can be classified in a set of classes or intervals. For instance, the range of temperature can be divided in three classes “warm”, “medium” and “cold”. Each of them corresponds to an interval of numerical values. This way, a *classified quantitative* variable (also called ordinal variable) has a range that consists of a set of disjoint intervals in a numerical range. On the other hand, values for qualitative variables (also called nominal variables) lie within a discrete nominal range: each value is typically associated to a term. Land cover or geomorphology are examples of qualitative datasets.

We are interested on both spatial models as well as qualitative and classified quantitative thematic variables. However, we define a restriction: each dataset only contains one thematic variable, and consequently each spatial unit in the dataset only has one thematic value. This restriction is independent from the spatial model, since the spatial unit can be either a cell or a feature with its geometry. This is a common restriction in the majority of formal approaches in semantic interoperability in order to avoid structural heterogeneity. Nevertheless, if we have a (typically vector) dataset with several thematic attributes, we can identify them as different logical datasets, one for each thematic attribute, and thus, the restriction is not as important as it might seem at first.

1.1.3 Our multimedia context

The second part of the problem that motivates this thesis is related to the application of semantic integration of thematic geographic information in other contexts, following the philosophy of the Semantic Web. In particular, we focus on the problem of indexing and retrieving geo-referenced multimedia elements according to their thematic geographic content.

In fact, we have been working during more than ten years in different aspects of hypermedia modelling. In particular, we have been interested in looking for ways of improving the visualization of geographic information by means of video. This involves an automatic construction of hypervideos that can be navigable according to their thematic geographic content, where the relevant video segments contain references to the thematic GI they are showing.

There is a necessity for representing the thematic GI to be used in the process of indexing and retrieving those videos, where the thematic GI may come from different sources. In this context, we remarked the lack of a suitable semantic framework to achieve semantic integration of this thematic GI. The development, grounding and evaluation of a semantic framework for integrating thematic GI has become the main issue of this thesis. As a second aspect of the problem addressed in this thesis we also consider the application of our approach in the context of indexing and retrieving videos (and also still images) according to their thematic content.

1.2 Overview of our approach

Let us start first with a paradigmatic example of semantic heterogeneity. Let us consider a mapping agency that builds a dataset of land use in Europe. It probably has different needs from a local organization producing a dataset of land use in a specific small area, such as, for instance, the Serra de Tramuntana in Majorca. At a European level, the dataset producer will be interested in depicting general categories, while the local organization will probably use concepts closer to the local reality and perhaps linked to local regulations. For instance, while the European dataset author may consider a category "protected natural areas", the local one may be interested in depicting what particular type of protection each natural area has received, such as for instance *Àrea natural d'especial interès* (natural area of special interest), which is a type of protection in the Balearic law.

In this typical case, we see that each organization structures the dataset in a different way. The entities or categories are chosen according to their particular needs. This example illustrates that, although both projects share the same reality, they would represent it in a different way. This is an example of metaphysical semantic heterogeneity, particularly showing granularity differences.

The level of detail, that is typically directly affected by scale as well as by the minimum size considered for spatial units, is an example of a factor determining the type of representation, and in fact the mental model underneath the dataset. But there may be other factors. For instance, when an expert on ecology and an expert on geomorphology observe a piece of the real world to produce a dataset, they focus on different categories. While the former may categorize a piece of land as "coniferous forest", the latter may categorize it as "alluvial plane". This is another example of metaphysical semantic heterogeneity, now produced by different perspectives.

Different datasets may also present epistemic semantic heterogeneity. For instance, different dataset producers (and consumers) may have different ideas of what a "land suitable for building" or "land at fire hazard" mean. Our approach supports the definition of different models for a thematic concept.

All the datasets mentioned in these examples will obviously be correct, but they have been produced for different purposes, with different mental models and consequently different representation schemas. It has to be remarked that, from our point of view, an interoperability approach cannot be based on constraining the freedom of dataset producers to represent GI as they need. Contrary to this, it has to provide mechanisms supporting datasets as they have been created by the producers, without changing their internal representation schema to support a variety of users and uses.

The basis of our work is to provide a framework to represent the semantic relations among the concepts appearing in the different datasets that form a repository; these datasets may have been produced by different authors, and structured according to different representation schemas. For instance, if a dataset uses the entity "pine forest" and another one uses the entity "woodland", there is a semantic relation between them, in the sense that "pine forest" is a particular type of "woodland".

From a semantic point of view, we can identify the representation schema of a dataset with an application ontology, in terms of Guarino's classification of types of ontologies (Guarino 1998) (some basic notions on ontologies are introduced in Chapter 2). Although there is a shared conceptualization in the domain of geosciences that enables their members to communicate with each other, each dataset presents its own ontology that commits to the conceptualization in a different way.

According to these premises, these application ontologies have to be expressed in a formal way. Then, a higher level ontology is obtained through a *merging* process to represent the knowledge in the overall repository of datasets by specifying the semantic relations among the different dataset application ontologies. This higher level ontology does not claim to describe all the semantics in all the subfields of geography, but instead, to describe in a precise and formal way the contents of the repository. This ontology is the basis for defining semantic services (also referred to as semantic queries in the course of this thesis) that will enable users to find datasets containing certain thematic entities, translate a dataset to another vocabulary and, more important, integrate data from different datasets in a new one. None of these services is supported by current catalogues. This shows that our approach provides new relevant functionalities.

Summarizing, our approach consists in defining a semantic framework that comprises three main elements: an ontology that represents the thematic knowledge in the GI repository made of different datasets, merging methods and several semantic services enabling external agents or applications to use it.

We have developed a prototype tool, OntoGIS, that implements our semantic framework, including the management of the repository, merging methods and the three types of semantic queries. OntoGIS will be described in Appendix A.

We will distinguish among *quantitative* and *qualitative* datasets depending on the type of thematic variable. Thematic interoperability has to be addressed in different ways depending on whether we deal with quantitative or qualitative datasets. In particular, we will see that semantic interoperability with qualitative thematic datasets is more complex than with quantitative ones.

The following subsections briefly discuss each of the main elements presented here, although they will be covered in depth in several chapters of the thesis.

1.2.1 Quantitative datasets

The main semantic issues that have to be represented for a quantitative dataset are the main theme of the dataset (for instance temperature) and the units of measure (for instance Celsius degrees or Fahrenheit degrees). The main theme can also be characterized by other properties related to how the samples have been obtained. For instance, following on the temperatures example, we should distinguish whether a dataset shows maximum or minimum temperatures. And in this last case, it may also be relevant to distinguish whether the minimum value of a location represents the minimal value in the whole year or, for instance, if it comes from an yearly average of daily minimal values.

In any case, the semantic heterogeneity among two datasets of temperatures is minimal since the meaning of the values is not ambiguous. In the case of a classified quantitative variable (for instance, intervals of temperature corresponding to “warm”, “medium” and “cold”) a semantic problem arises, since two datasets of the same theme may use different sets of intervals. It is important to remark that classes with the same name in different datasets might not share the same meaning. For instance, the class “cold” has often different meanings in two datasets (through different thresholds). We can observe that every class or interval in a classified quantitative dataset could have its own meaning.

Our representation model deals with classified quantitative datasets by supporting the description of the meaning of the intervals defined in each dataset. The meaning of each class is given by its minimum and maximum values. A quantitative theme as temperature may have different classifications, each one with its own set of classes or intervals. A class “cold” of the theme “temperature” only has an unambiguous meaning when we identify it with its threshold values.

Note that in our model, reasoning about a quantitative dataset is done according to a particular classification. In this context, a numerical value such as 25.87 ºC is per se useless, unless it is attached to the properties of a specific class. For that reason, we do not consider pure quantitative datasets with raw non-classified values in our model. Thus, from now on, we will use “quantitative” and “classified” as synonyms and wherever we say “quantitative datasets” in fact we are referring to “classified quantitative datasets”.

1.2.2 Qualitative datasets

Unlike quantitative datasets, the value that a qualitative dataset associates to a geographic location does not correspond to the numeric measurement of a thematic variable. Instead, a qualitative dataset associates a concept to each location, where this concept represents a quality of the thematic variable. Some examples of qualitative datasets are land cover, geomorphology or types of soil, while examples of qualitative values for these datasets are “pine forest”, “alluvial plain” or “pedocal soil” respectively. Although the value physically stored in the data file is an integer number, it is associated to a term (or set of terms), that defines its meaning. From a conceptual point of view, the value of a qualitative dataset is not a number itself but the concept that the term or terms represent. The terms are usually shown in the legends of the maps, so that humans understand the meaning of each value. Sometimes these values may be grouped in order to show a hierarchy. Figure 1 shows an example of a two-levels hierarchy of types of coast, where the six physical values in the lower level are classified as either “*Costa Baixa*” (low coast) or “*Costa amb penya-segat*” (cliffs) in the upper level.

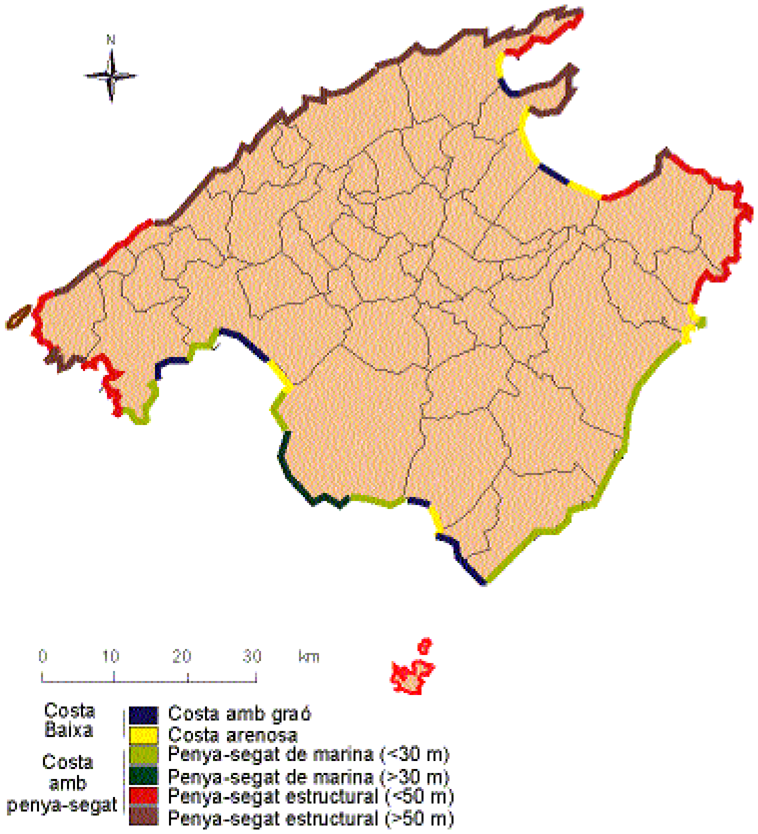

N
0
10
20
30
km
Costa Baixa
Costa amb penya-segat
Costa amb graó
Costa arenosa
Penya-segat de marina (<30 m)
Penya-segat de marina (>30 m)
Penya-segat estructural (<50 m)
Penya-segat estructural (>50 m)

Figure 1. Coastal model of Majorca. Source: Atles de les Illes Balears (CD-ROM)

Each of the qualitative values corresponds in fact to a geographic theme, which is more specific than the main theme of the dataset. As a further example, the “urban area” value in a land use dataset is a geographic theme, which could be the main one in other datasets, where different subtypes of urban areas would be values. Additionally, this example illustrates how qualitative values from different datasets are typically related, in this case through an is-a relation, and how this relation conforms a taxonomy of themes.

As it has already been discussed, different authors that produce different datasets of the same theme will specialize it using different sets of values tailored to their specific needs. Our model has to represent the relations among these different organizations of values, expressing the main different themes and thematic values used in each of them and how they are related. Each theme, either the main theme or a thematic value, is characterized through an ontology class. It is worth noting here that the most typical relation is the specialization (is-a), but it is not the only one. Relations of equivalence, disjointness and property restrictions can also be set.

1.2.3 Vocabularies

Several normalized vocabularies exist for some sub-fields of geosciences. The CORINE vocabulary for land cover and land use (Bossard et al. 2000) is a good example. These vocabularies consist of a set of terms that are related by means of the three usual thesauri operations: broader term, narrower term and synonym term.

However, datasets producers must not be forced to use normalized values of these vocabularies. The needs of a dataset producer may not be well covered by the vocabulary, and thus s/he has to use other thematic concepts not included in the vocabulary. But vocabularies have the advantage of being known by the community, and consequently, they provide a shared meaning for thematic concepts. Consequently, it is very useful to integrate these vocabularies into the ontology: vocabulary terms can be converted into ontology classes, broader terms into superclass relations, narrower terms into subclass relations and synonym terms into equivalence relations. Values in datasets can be related to classes that can be understood by the community, although vocabularies themselves are not directly referred to in datasets. Their role is to provide a kind of interchange language for thematic knowledge. As we will see below, this is very related to one of the semantic services that we have identified, which translates a dataset thematic structure to terms in a known vocabulary.

1.2.4 Modelled themes and Description Logic definitions

Expressing the values of the datasets in a repository by means of an ontology, with classes and relations, does not solve the problem that arises when two producers have different definitions of a theme (epistemic semantic heterogeneity). The previous examples considered thematic concepts having unambiguous meanings. For instance, every member of the geosciences community will in principle agree that “pine forest” is a particular type of “forest” where pine trees predominate. However, different producers (and consumers) will usually have different ideas on what a “land suitable for building” means, since the definition of an area as suitable for building may depend on the urban planning model that has been considered or may be affected by different local regulations.

Our representation model deals with this kind of themes that we have denominated *modelled* themes. Each modelled theme such as “land suitable for building” may have different definitions. The definition of a modelled theme can be formally expressed by means of Description Logic (DL) axioms and may include references to other themes (quantitative, qualitative or even modelled). Note that “land suitable for building” expresses a quality and in consequence can also be considered as a qualitative theme.

Quantitative themes are sufficiently defined by providing the boundaries of their intervals; and while qualitative themes in general could be also modelled through DL definitions, we only use modelled themes when different definitions may appear. Thus, the set of modelled themes is a subset of the set of qualitative ones in our representation. When one or more models are assigned to a qualitative theme, its meaning is not obtained from its name but from its definitions. The decision whether a theme is modelled or is kept as primitive depends usually on the required level of detail of the ontology.

The representation of quantitative and qualitative datasets, vocabularies and modelled themes in the ontology is described in Chapter 4.

1.2.5 Merging

As we will discuss in Chapter 5, several tools for merging ontologies, which rely on semi-automatic methods mainly based on lexical similarities among the names of classes and properties, have been proposed. However, merging geographic dataset ontologies presents some significant particularities that are not well covered by these methods and tools. Dataset ontologies are usually structured in simple, almost flat, hierarchies of themes. In addition, the spatial distribution of values in datasets also provides an indication of possible relations. For instance, if the union of all the spatial units having the value “pine forest” in a dataset is contained by the union of the spatial units having the value “forest” in the other datasets in the repository, it can be entailed that “pine forest” is probably a subclass of “forest”. In our case, we are interested in merging the application ontology of a dataset into the higher level ontology of the whole repository, which is dynamically built in this process.

We propose two merging methods to deal with these dataset ontologies. The first one is a manual method that enables a domain expert to manually determine mappings with some guidance. The second one is a semi-automatic method that generates a list of suggested mappings that can be confirmed or modified by the expert. The key element of this method is the algorithm (referred to as *mapping algorithm*) that generates the list of suggestions. We have developed and tested three different mapping algorithms that focus on different aspects. The first one is based on lexical similarities between class names as well as on the structure of the ontologies being merged. The second one introduces the use of a terminological base in order to consider synonymy, hypernymy and hyponymy. And the third one is based on the spatial distribution of dataset values. All of them have been implemented in the OntoGIS tool.

Our merging methods and mapping algorithms are described in more depth in Chapters 5 to 8.

1.2.6 Semantic queries

Our approach aims at building an ontology representing the thematic information of a repository of datasets. This ontology, in the framework of the Semantic Web, makes it possible to define semantic services or queries that enable agents to find and integrate thematic information. We use the term agent here in a wide sense, including not only autonomous software agents, but also humans or other applications.

We have identified three main types of semantic queries that we briefly discuss now. It is important to remark that these three operations are not provided by current catalogues. From a semantic point of view, current catalogues only provide a simple keyword-based service for finding datasets.

The first type of semantic query is *finding*: it enables agents to find datasets and dataset values containing information on a particular theme. The agent may also decide whether subclasses of the selected theme are of interest too.

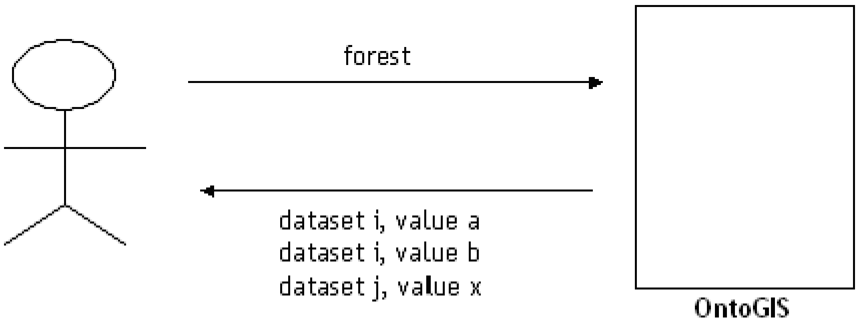
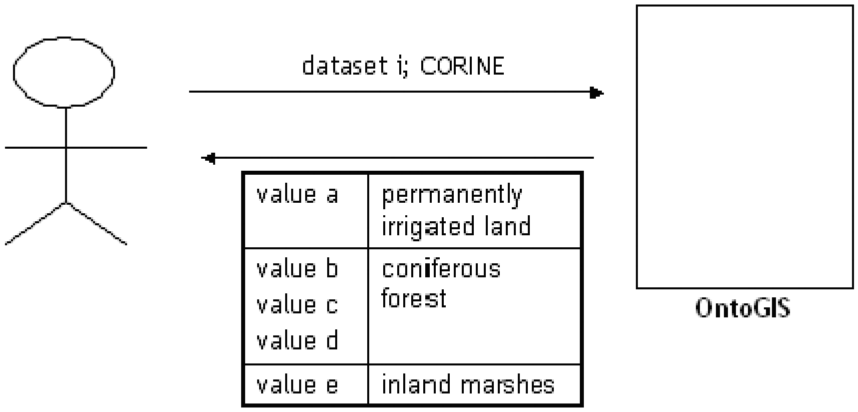


Figure 2. First type of semantic query, finding

The second type of semantic query is *translation*: if a dataset is expressed using a set of themes that an agent does not understand, the agent can ask the system to translate from the dataset organization to a specific vocabulary that it understands, as for instance the CORINE land cover vocabulary (Bossard et al. 2000).



value a	permanently irrigated land
value b value c value d	coniferous forest
value e	inland marshes

Figure 3. Second type of semantic query, translation

Finally, the third type of semantic query is *integration*, the most complex one: it integrates data from different datasets that contain information on a particular theme in the same spatial area and generates a new dataset. There are two main approaches to perform this operation. One looks for the maximum consensus among the information coming from the different datasets. The other one looks for the most specific information that can be said about an area without being contradictory with the source datasets. Our approach combines both. A particular application of this semantic query is to find areas that contain semantic contradictions among different datasets. This provides a measure of the agreement between two or more datasets, and can be used as a measure of quality if one is considered as the reference.

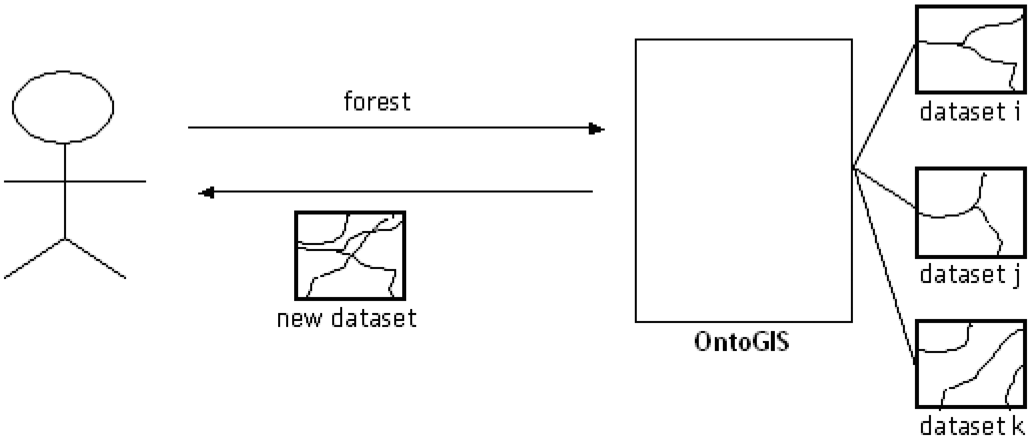


Figure 4. Third type of semantic query, integration

The three types of semantic queries, as well as some relevant variations, will be further described in Chapter 10. A formal model of how the responses to each one are calculated based on DL will also be presented in that chapter.

1.3 Objectives of the Thesis

The main objective of this thesis is to define a formal framework to solve the problem of semantic interoperability in the geographic domain, particularly overcoming terminological, metaphysical and epistemic discrepancies.

Other partial objectives refer to the three specific components of the semantic framework:

- An ontology has to be defined to represent the thematic concepts and relations in a repository of datasets and known vocabularies. Furthermore, it has to support the definition of models to obtain new thematic concepts from others, through Description Logic axioms.
- A semi-automatic merging method has to be defined to integrate the application ontologies of datasets into the repository ontology, taking into consideration the particularities of geographic information. The key point of the method is the mapping algorithm that should generate a list of suggested mappings.
- The three types of semantic services that have been identified have to be formally defined in terms of Description Logic. In particular, these semantic services have to support themes defined in terms of Description Logic.

As a final objective, it is important that the semantic services provided by this framework can be used by external applications in the context of the Semantic Web. In particular, we will test that these services support the use of the thematic information in the repository to index and retrieve geo-referenced videos according to their geothematic content.

1.4 Structure of the Thesis

Chapter 2 presents some basic notions on ontologies, Description Logic and the OWL language for representing ontologies, that are necessary to understand the rest of the chapters.

Chapter 3 provides an overview of the most relevant work in the area of Geographic Information Interoperability. It describes the main metadata standards for GI, the most widely used thesauri and normalized vocabularies in this field, and discusses several different approaches focusing on semantics mainly based on ontologies.

Chapters 4 to 10 describe the three elements of our semantic framework: conceptual model (ontology), merging methods and semantic queries.

Chapter 4 discusses our formal conceptual model, which is the core of the semantic framework. It consists of an ontology that represents thematic concepts in the repository and their semantic relations, as well as logical definition of concepts (modelled themes). The ontology is expressed in the OWL language, namely according to the DL profile, which permit Description Logic reasoners as FaCT or Racer to deal with it.

Chapter 5 focuses on merging methods. First, it presents related work on merging/alignment, mainly based on the definition of similarity functions. Then it describes a manual method that enables a domain expert to establish the relations between thematic concepts. Then it introduces the semi-automatic merging method that relies on three different algorithms for the generation of suggested mappings (mapping algorithms) that are discussed in the subsequent chapters. Finally, a specific method for quantitative datasets is presented.

Chapter 6 describes a mapping algorithm based on lexical similarities among the names of classes and on the structure of the ontologies being merged. The algorithm is based on an asymmetric similarity measure between class names. The structure of the ontologies also influence the algorithm through the mechanisms of mapping restrictions and structural rules.

Chapter 7 introduces a second mapping algorithm that uses a terminological base to find similarities considering synonyms, hyponyms and hypernyms terms. In our implementation we have considered the WordNet lexical base and the GEMET thesaurus of environmental terms. The algorithm is based on a score measure that has been defined as well as the so-called term mapping restrictions. It also considers the structure of the ontologies through mapping restrictions and structural rules.

Chapter 8 discusses a third mapping algorithm that obtains semantic relations from the overlapping of the spatial distribution of dataset values. Compared to other approaches based on the spatial information, it supports many-to-many equivalences and can be computed in real time.

Chapter 9 evaluates the mapping algorithms that have been presented in the previous chapters. To do this, some evaluation experiments with real land cover/land use datasets have been conducted. A relaxed definition for precision and recall measures specific for ontology merging/alignment is also provided in this chapter.

Chapter 10 formally defines the three types of semantic services or queries, and some relevant variations, in terms of Description Logic. Especially relevant is the integration service, which eliminates some restrictions of other approaches and supports the use of modelled themes in the process.

Chapter 11 discusses the use of our semantic framework in a multimedia context, namely in indexing and retrieving geo-referenced still images and videos by their geothematic content. An algorithm for segmenting and indexing geo-referenced videos, a semantic model for describing still images and videos and its representation through MPEG-7 are also described in this chapter.

Chapter 12 summarizes the main results obtained in this thesis, and describes further directions of work.

Appendix A describes the OntoGIS tool that has been developed to implement the semantic framework defined in this thesis, including the ontology, merging methods and semantic queries. It has been developed in Java using the HP Jena API.

The remaining appendixes cover specific issues related to different chapters. Appendix B presents the complete OWL document defining our ontology. Appendix C includes the CORINE and Anderson vocabularies for land cover/land use that are used in one of the evaluation experiments in Chapter 9. The details of the experiment can be found in Appendix D. Likewise, Appendix E includes the vocabularies for land cover/land use from the USGS Earth Land Cover Maps that have been used in another evaluation experiment that is detailed in Appendix F. Finally, Appendix G presents the XML Schema document for the image and video metadata defined in Chapter 11.

2 Some basic notions on ontologies and Description Logic

This chapter briefly presents some basic concepts on ontologies, Description Logic and the W3C language OWL for representing ontologies in the Semantic Web. These concepts are needed to understand the rest of the chapters.

2.1 Definitions of ontology

The term ontology is used in Philosophy since Aristotle. It comes from the Greek and means the science of the being (*onto*). The Webster Dictionary defines it as “the branch of metaphysics that studies the nature of existence or being as such, as distinct from material existence, spiritual existence, etc.”

Gruber (Gruber 1993) was one of the first authors that introduced this term in Computer Science. He provided the most quoted definition of ontology as an “explicit specification of a conceptualization”, where by conceptualization we understand a shared abstract and simplified vision of the world that is intended to be represented. Thus, any knowledge-based system is committed to (is logically consistent with) a conceptualization, in an either explicit or implicit form. An ontology in Artificial Intelligence is typically used to provide a common vocabulary for a set of agents. An ontology consists of a set of names of entities (including classes, relationships and functions), and their definitions in a way readable by humans, and may also include a set of axioms constraining the possible interpretations. According to Gruber, although ontologies are often equated with taxonomic hierarchies of classes, with class definitions and subsumption relation, they do not have to be limited to these forms. Instead, an ontology is the statement of a logical theory.

(Guarino and Giaretta 1995) arguments that Grubers’s conceptualization, which in fact is based on the notion of (Genesereth and Nilsson 1987), is a set of extensional relations describing a particular state of affairs. Instead, they propose an intensional approximation: a conceptualization is “an intensional semantic structure which encodes the implicit rules constraining the structure of a piece of reality”. Guarino (Guarino 1998) provides a more flexible definition of ontology as “a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualization of the world”. In this context, an ontology can only specify a conceptualization in a weak way. He affirms that an ontology *O* commits to a conceptualization *C* if *O* has been designed with the purpose of characterizing *C* and *O* approximates *C*. This permits different ontologies to commit to the same conceptualization in different ways. This way, one ontology may be closer to the conceptualization that another one. An ontology gets closer to the conceptualization by adding either more axioms or more concepts and relations. As a result, he makes the distinction between coarse-grained and fine-grained ontologies. Typically, fine-grained ontologies (very detailed) will be used as references while coarse-grained ontologies (more generic) can be shared more easily. And according to this level of generality, he distinguishes between four kinds of ontologies: top-level, domain, tasks and application, as Figure 5 shows.

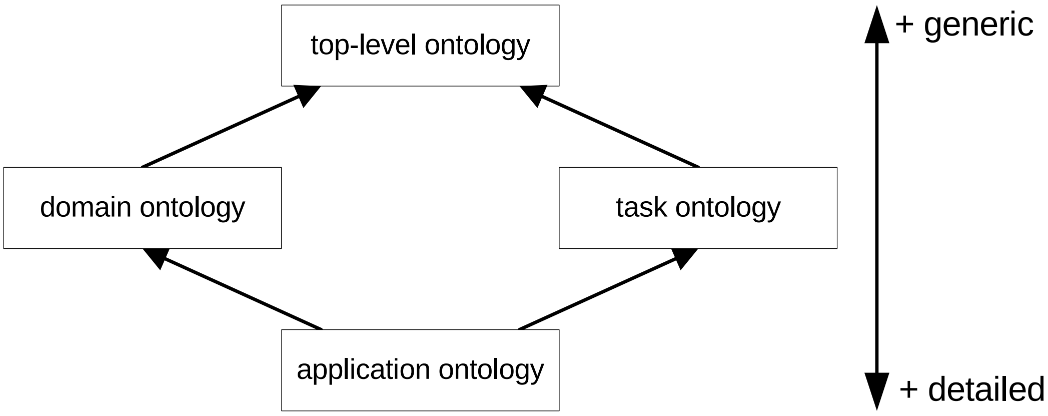


Figure 5. The four kinds of ontologies according to (Guarino 1998)

Our approach is based on the notions of Guarino. We assume that each map may have its own application ontology. These application ontologies are merged in a higher level ontology describing the repository. The repository ontology is not exactly a domain ontology, since this term is usually only used to refer to the ontology for a complete generic domain (like medicine, or geoscience in our case).

This repository ontology does not describe all the semantics in all the subfields of geography, but instead, describes in a precise and formal way the contents of the repository that may contain knowledge from several of these subfields.

Apart from Gruber’s and Guarino’s definitions of ontology, which are the most widely recognized, many other definitions exist. However, although there may be some discrepancies on how ontologies are defined, there is a high consensus on their usage (Gómez-Pérez et al. 2004). They are mainly used to specify a common vocabulary, through concepts (entities), usually documented with a text-based definition and structured forming a hierarchy, roles (properties of these entities), relations between concepts, individuals (instances of entities) and logical axioms constraining the possible interpretations.

2.2 Description Logic

We can observe that logic plays an important role in the previous definitions of ontology. It is well acknowledged that logic provides a formal and precise way for representing semantics and reasoning with it. This was the main limitation of other approaches to knowledge representation based on *ad hoc* data structures and reasoning procedures like semantic networks and frame systems.

Distinct logic formalisms differ in terms of their representation power and computational complexity of inference. The more restricted the representational power, the faster the inference in general. *Propositional Logic* provides too little expressive power. *First-order Logic* (FOL) is very expressive but inference is not only expensive, but may not terminate.

Description Logic (DL) appeared as a new logical formalism for conceptual modelling, providing an expressive language with terminable reasoning algorithms. Other previous names of DL were *terminological systems* and *conceptual languages*, both emphasizing the (hierarchical) structure of concepts that represent a domain. We include in this section some of the basic notions of DL formalism. More details on Description Logic can be found in the book (Baader et al. 2002) as well as in the on-line courses (Franconi 2002) and (Lutz and Sattler 2002).

A DL knowledge base (KB) comprises two components: *TBox* and *ABox*. The TBox (also often called taxonomy) contains intensional knowledge that introduces the terminology (hence the 'T' from TBox), i.e. the vocabulary of a domain, through declarations that describe general properties of concepts. The ABox contains extensional (also called assertional, hence the 'A' from ABox) knowledge specific to the individuals (instances) of the domain in terms of the vocabulary introduced in the TBox. In other words, the TBox contains the definitions of concepts and roles (binary relations between concepts that can be identified to properties), while the ABox contains definitions of individuals.

DL semantics is based on a set-theoretic interpretation. A concept denotes a set of individuals. A role is a set of pairs of related individuals. In addition to atomic concepts and roles, the TBox can contain complex descriptions of other concepts and roles. We will briefly discuss below the different description languages that contain different constructors. These constructors can also be defined in terms of sets. For instance, intersection of concepts, which is denoted by $C \sqcap D$, is used to restrict the set of individuals under consideration to those that belong to both concept C and concept D. Similarly, the interpretation of a value restriction, written $\forall R.C$, is the set of individuals that are related through R to an individual belonging to the set denoted by the concept C. Note that in DL we can restrict the concept for the second individual of a role, but not for the first one. As a consequence, roles are not specific for any particular concept (which is common in databases, frame logic systems and some DL implementations like OWL).

Regarding the TBox, there are two types of declarations. The basic one is a *concept definition* that provides a logical equivalence through necessary and sufficient conditions. For example, we can define a woman as the set of individuals that are both person and female:

$$Woman \equiv Person \sqcap Female$$

The other type of declarations are *inclusion axioms*, where only a logical inclusion is provided. For instance:

$$Dog \sqsubseteq Animal$$

Although inclusion axioms have less definitorial impact, they are common in the construction of taxonomies (hierarchical structures of concepts).

The ABox contains assertions on individuals, usually called *membership assertions*. For instance:

$$Person \sqcap Female(MARIA)$$

$$hasChild(MARIA,ANNA)$$

The first assertion indicates that the individual Maria belongs to the concepts *Person* and *Female*, and consequently to *Woman*, while the second assertion indicates that Maria has Anna as a child (*hasChild* is an atomic role). The first is usually called a *concept assertion*, while the second a *role assertion*.

A DL knowledge base not only stores terminologies and assertions, but also offers services that reason about them. The basic inference service in Description Logics is *subsumption*, typically written as $C \sqsubseteq D$. Determining subsumption is the problem of checking whether the concept denoted by *D* (the *subsumer*) is considered more general than the one denoted by C (the *subsumee*). In other words, subsumption checks whether the first concept always denotes a subset of the set denoted by the second one. For instance, one may be interested in knowing if it can be deduced from the declarations in the TBox that $Mother \sqsubseteq Woman$. Other typical reasoning service is *concept satisfiability*, which is the problem of checking whether a concept expression does not necessarily denote the empty concept. In fact this is a particular case of subsumption. Another significant reasoning service is *instance checking*, which determines whether a given individual belongs to a certain concept. Other common reasoning services are *retrieval* and *realization*, which obtain respectively the individuals that belong to a given concept, and the concept to which an individual belongs. Systems based on DL usually implement other services derived from these ones. *Classification* (constructing the subsumption hierarchy between all concepts defined) is the most common of these services.

An important remark has to be done concerning the analogy between databases and DL knowledge bases; the DB schema can be compared to the TBox, while the instances (rows) in the DB can be compared to the ABox. However they follow different semantic approaches. The database represents exactly one interpretation, where the classes and relations in the schema are interpreted by the instances in the DB. In addition, an ABox represents many different interpretations, namely all its models. Consequently, absence of information in a database is interpreted as negative information, while absence of information in an ABox only indicates lack of knowledge. For example if we only have one assertion related to Peter, *hasChild*(*PETER*,*HARRY*), in the database semantics we determine that Peter only has one child. But the ABox deals with incomplete knowledge and we can only affirm that Harry is a child of Peter, but we do not know whether or not

Peter has more children. In fact we cannot deduce either that all Peter's children are male. The semantics of ABoxes is often characterized as an "open-world" semantics, while the semantics of databases is characterized as a "closed-world" semantics.

2.2.1 DL constructors and languages

There are a considerable number of DL constructors. The choice and combination of different constructors define different languages. The most constructors a language has, the most expressive it is, but also the most inefficient it is. This way, a compromise solution has to be found for each particular case between expressive power and efficiency.

The simplest language is $\mathcal{AL}$ (attribute language). It has the following constructors, where we use the letters A and B for atomic concepts, the letter R for atomic roles, and the letters C and D for concept descriptions:

$$\begin{array}{lll} C, D \rightarrow & A\ | & \text{(atomic concept)} \\ & \top\ | & \text{(universal concept)} \\ & \bot\ | & \text{(bottom concept)} \\ & \neg A\ | & \text{(atomic negation)} \\ & C \sqcap D\ | & \text{(intersection)} \\ & \forall R.C\ | & \text{(value restriction)} \\ & \exists R.\top & \text{(limited existential quantification)} \end{array}$$

Note that the existential quantification $\exists R.C$ represents the set of individuals related to another one that belongs to concept C. For instance $\exists hasChild.Female$ is the concept of "individuals having a female child". However, $\mathcal{AL}$ language only provides a limited existential quantification that does not permit to restrict the concept of the second individual of the pair. Following the example, we can only obtain the individuals having a child, but we cannot precise whether they are female.

It is worth including here a remark about notation related to the existential quantification. The so-called "fills" constructor (:)

$$R : a$$

stands for the set of individuals that are related to the individual a through the role R (we say that they have a as a filler of the role R). In a language with full existential quantification and singleton sets (not $\mathcal{AL}$ language as we have seen), this constructor does not add anything new and can be expressed as:

$$\exists R.\{a\}$$

We will use the "fills" constructor in several places in this thesis.

For historical reasons, the sublanguage of $\mathcal{AL}$ language obtained by disallowing atomic negation is called $\mathcal{FL}^{-}$ and the sublanguage of $\mathcal{FL}^{-}$ obtained by disallowing limited existential quantification is called $\mathcal{FL}_0$.

Other relevant constructors in DL contained in other languages are:

- Union ($C \sqcup D$) which represent the set of individuals that belong to concept C or concept D (or both)
- Negation ($\neg C$) which is the negation of a no necessarily atomic concept
- Cardinality restrictions ($\geq n\ R$ and $\leq n\ R$) which indicate the set of those individuals related to at least/at most n individuals through role R
- Qualified cardinality restrictions ($\geq n\ R.C$ and $\leq n\ R.C$) which indicate the set of those individuals related to at least/at most n individuals of concept C through role R
- Enumeration ($\{a_1,...,a_n\}$) that permits to extensively define a concept through a set of individuals
- Role hierarchy ($R \sqsubseteq S$) which permits inclusion axioms between roles (R and S are roles)
- Inverse role (R^{-}) that identifies a role which is inverse to R. For instance, the concept $\exists child^{-}.Doctor$ refers to those individuals having a parent who is a doctor.
- Transitive role expresses that $R(a,b)$ and $R(b,c)$ mean $R(a,c)$

Each of these constructors is identified by a symbol, and languages are named according to the symbols of the constructors it supports. For instance, negation has the symbol $\mathcal{C}$ and the language that contains all the constructors of $\mathcal{AL}$ languages plus negations is identified as $\mathcal{ALC}$. Table 1 shows the main DL constructors and the symbol that identify them.

Constructor	Symbol
Negation	$\mathcal{C}$
Union	$\mathcal{U}$
Existential quantification	$\mathcal{E}$
Cardinality restrictions	$\mathcal{N}$
Enumeration	$\mathcal{O}$
Transitive role	$\mathcal{R}^{+}$
Role hierarchy	$\mathcal{H}$
Inverse role	$\mathcal{I}$
Qualified cardinality restrictions	$\mathcal{Q}$

Table 1. Common DL constructors and their symbols

The languages $\mathcal{ALC}$ and $\mathcal{ALCUE}$ are equivalent, since union and existential quantification can be represented using negation and the constructors of $\mathcal{AL}$. The language $\mathcal{ALC}_{\mathcal{R}}{}^{+}$ ($\mathcal{ALC}$ with transitive roles) is usually identified as $\mathcal{S}$. This way, the language that supports all the constructors mentioned in Table 1 is called $\mathcal{SHIQ}$. Reasoning in $\mathcal{SHIQ}$ has been deeply investigated by (Horrocks et al. 2000), and is the basis of OWL and the two main DL reasoners: FaCT (Horrocks 1998) and Racer

(Haarslev and Möller 2001). DIG (DL Implementors Group) has implemented a common interface for DL reasoners[1]. DIG interface has been implemented by both FaCT and Racer, and permits external applications to access them in a standard way.

2.3 OWL: a language for representing ontologies in the Semantic Web

The first languages for representing ontologies were developed at the beginning of the 1990’s. Some of the most relevant were KIF (Knowledge Interchange Format) (Genesereth and Fikes 1992), based on First-order Logic (FOL); and CycL (Lenat and Guha 1990), used to build CYC ontology, Ontolingua (Farquhar et al. 1997), OCML (Operational Conceptual Modelling Language) (Motta 1999) and F-logic (Kifer et al. 1995), all based on frames combined with FOL. DL-based systems mainly used Lisp-like languages as in KL-ONE (Brachman and Schmolze 1985), CLASSIC (Borgida et al. 1989) or LOOM (MacGregor and Bates 1987). The variety of ontology languages inspired OKBC (Open Knowledge Base Connectivity) (Chaudhuri et al. 1998), which was developed as a protocol for interconnectivity between knowledge bases with different representation languages. More information on these and other languages can be found in (Gómez-Pérez et al. 2004), (Fensel et al. 2003) and (Baader et al. 2002).

The appearance of the Semantic Web initiative has motivated the development of new ontology languages based on XML in the last years, many of them promoted by the W3C. The first ontology language based on a markup language was SHOE (Simple HTML Ontology Extension) (Luke and Helfin 2000), although it is not XML-based but rather an extension of HTML. In 1999 the W3C published RDF (Resource Definition Framework) (W3C 2004a; b) as a basis for describing web resources through triples of type <resource, property, value>. RDFS (RDF Schema) (W3C 2004c) permits to build taxonomies that express classes of resources and their subclass relationships, and define properties and associate them with classes. RDFS is the base on top of which other ontology languages have been defined, evolving from DAML (McGuinness et al. 2003) and OIL (Fensel et al. 2002) to DAML+OIL (Connolly et al. 2001) and finally to OWL Web Ontology Language (W3C 2004d; e). OWL is an standard of the W3C and is widely used by the Semantic Web community. A variety of tools exist to deal with OWL at different levels (editors, parsers, reasoners,...).

While RDFS only provides simple representation constructors (mainly subclass), OWL supports the definition of complex ontologies through DL constructors. OWL is mainly based on a DL $\mathcal{SHIQ}(\mathcal{D})$ language, where $(\mathcal{D})$ stands for data types support. OWL also supports domains of properties. Table 2 shows some of the main constructors supported by OWL. We can observe that it also uses RDF(S) constructors. In the OWL terminology concepts are usually called classes and roles properties.

[1] http://sourceforge.net/projects/dig

Definition of classes (concepts)	
`owl:Class`	Used to define a concept
`owl:Thing`	The top concept
`rdfs:subClassOf`	Used to provide an inclusion axiom (subsumption)
`owl:equicalentClass`	Used to denote intersection equivalence between concepts (definition)
`owl:intersectionOf`	Used to denote intersection of concepts
`owl:unionOf`	Used to denote union of concepts
`owl:complementOf`	Used to denote the negation of a concept
`owl:disjointWith`	Used to denote that two concepts are disjoint (their intersection is forced to be empty)
`owl:oneOf`	Used to define a concept as an enumeration of individuals
Definition of properties (roles)	
`rdf:Property`	Used to define a generic role
`owl:DatatypeProperty`	Used to define a role with a data type range
`owl:ObjectProperty`	Used to define a role having a class as its range
`rdfs:domain`	Used to specify the domain of a role
`rdfs:range`	Used to specify the range of a role
`owl:SymmetricProperty`	A subclass of owl:ObjectProperty that denotes that a role is symmetric
`owl:TransitiveProperty`	A subclass of owl:ObjectProperty that denotes that a role is transitive
`owl:FunctionalProperty`	A subclass of rdf:Property that denotes that a role is functional (for each instance there is at most one value for the property)
`owl:InverseFunctionalProperty`	A subclass of rdf:Property that denotes that a role is both functional and inverse of another one
`owl:inverseOf`	Used to specify the inverse of a property
`rdf:type`	Used to specify the type of a property (symmetric, transitive,...)
Restrictions of properties	
`owl:Restriction`	Used to denote that a concept is defined as the restriction of a property
`owl:allValuesFrom`	Used to denote a value restriction ($\forall$): all values of a property must belong to a certain class
`owl:someValuesFrom`	Used to denote an existential quantification ($\exists$): at least one value of a property must belong to a certain class
`owl:hasValue`	Used to denote that a property has a specific value
`owl:minCardinality`	Used to denote minimum cardinality ($\geq n$)
`owl:maxCardinality`	Used to denote maximum cardinality ($\leq n$)
`owl:onProperty`	Used to specify the property a restrictions is assigned to

Table 2. Main OWL constructors

Since not all the systems have the same requirements, and the implementation of some of the constructors defined in OWL are very inefficient, the W3C has defined three different profiles: OWL Lite, OWL DL and OWL Full.

OWL Lite is a very reduced subset of OWL that makes the following restrictions:

1. *owl:minCardinality* and *owl:maxCardinality* cannot be used
2. *owl:cardinality* can only have values 0 or 1
3. *FunctionalProperty* and *InverseFunctionalProperty* can only be used with *ObjectProperty* (not with *DatatypeProperty*)
4. *owl:hasValue* cannot be used
5. *owl:unionOf*, *owl:disjointWith*, *owl:complementOf* and *owl:oneOf* cannot be used
6. *owl:cardinality* cannot be used with *TransitiveProperty*
7. metaclasses (classes as individuals of another classes) cannot be used

OWL DL profile removes the restrictions 1 to 4. This corresponds to a $\mathcal{SHIQ}(\mathcal{D})$ language and can be used with DL reasoners as FaCT and Racer. The ontology of our repository is expressed in OWL DL, as we will see in Chapter 4.

Finally OWL Full removes all the restrictions. It is more expressive, but there are no reasoning engines supporting it.

3 Related work on GI interoperability

In this chapter we provide an overview of existing approaches addressing the problem of geographic information interoperability.

In the first section of this chapter we will briefly describe the main metadata schemas for describing geographic information. Although they have not been designed to represent semantics, they provide a general overview of the content of a dataset, with other relevant information concerning to its use. This way, metadata is at the core of any interoperability solution, although other elements are also necessary to face up semantic interoperability.

A simple solution for the problem of semantic interoperability is that the community agrees on a shared (usually hierarchical) list of concepts, typically implemented through a thesaurus. This way, entities used in datasets must refer to terms appearing in the shared thesaurus. This can be a valid approach for a reduced domain, but certainly not for the complex geosciences domain. This solution has several drawbacks that are discussed in Section 3.2, which also presents some of the most widely used thesari in geosciences.

The use of ontologies for semantic interoperability has motivated several theoretical studies on how geographic concepts are represented through ontologies. An overview of the most relevant of them is presented in Section 3.3, which also covers other more practical approaches to the problem of semantic integration of GI from different datasets.

The last section of this chapter presents a set of approaches that address semantic interoperabilities although are not based on ontologies. We will see that these approaches are very related to our definition of integration (third type of semantic query) based on Description Logic.

3.1 Metadata schemas for GI description

Different national and international organizations regulate the development of standards for GI metadata schemas. The most extended in USA is CSDGM (Content Standards for Digital Geospatial Metadata) developed by the American FGDC (Federal Geographic Data Committee) (FGDC 1998). In Europe, Technical Committee 287 of CEN developed a voluntary norm ENV 12657 related to metadata (CEN 1998). Technical Committee 211 of ISO is also in the process of defining a set of specifications for geographic information interoperability, where ISO 19115 (ISO 2003a) is the approved standard concerning metadata. ISO standard covers the main elements of CEN norm (Gouveia et al. 2001), which has almost completely been replaced in Europe. Finally, the Open GeoSpatial Consortium has completely adopted ISO 19115 in their specifications.

Focusing on thematic information, these standards support the definition of a set of keywords defining the main topic of the dataset. However, the definition of each possible value of the dataset is even more relevant in the context of semantic integration. For instance, a dataset may have “land-cover” as a theme keyword, while its values are “forests”, “agricultural areas”, etc. The part of the metadata standards that deals with data schemas can be used to describe these values. In the case of ISO/TC211 specifications, data schemas are included in other document, 19109 (ISO 2005), called Rules for application schemas. ISO 19109 has been developed to cover the following scopes:

- conceptual modelling of features and their properties from a universe of discourse;
- definition of application schemas;
- use of the conceptual schema language for application schemas;
- transition from the concepts in the conceptual model to the data types in the application schema;
- integration of standardized schemas from other ISO geographic information standards with the application schema.

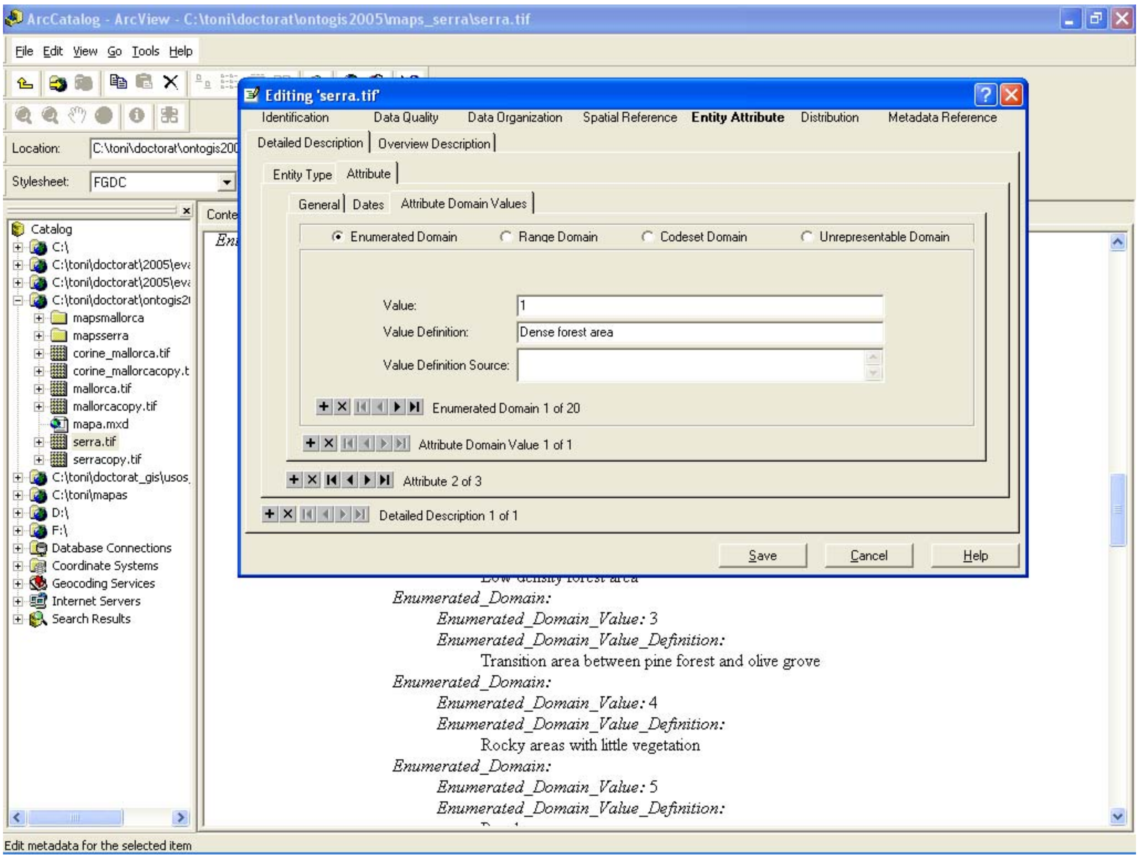
In the case of CSDGM the description of the application schema (entities, attributes and values) is covered in the own standard. As an example, Figure 6 shows a fragment of the metadata providing the schema description of the dataset “GIRAS Landuse/Landcover data for the Conterminous United States”, developed by the American Environmental Protection Agency's Office of Information Resources Management, and taken from the Geoscience Data Catalog of the United States Geological Survey.

```
Entity_and_Attribute_Information:
  Detailed_Description:
    Entity_Type:
      Entity_Type_Label: cover.PAT
      Entity_Type_Definition: Polygons in the coverage
      Entity_Type_Definition_Source: ESRI ARC/INFO
    Attribute:
      Attribute_Label: LUCODE
      Attribute_Definition:
        Anderson level 2 land use classification code number
        The first digit represents the level one value and the second
        digit (ones place) represents the subdivision of the level 1
        or level 2 value.
      Attribute_Definition_Source: GIRAS
      Attribute_Domain_Values:
        Enumerated_Domain:
          Enumerated_Domain_Value: 1
          Enumerated_Domain_Value_Definition: Urban or built-up land
      Attribute_Domain_Values:
        Enumerated_Domain:
          Enumerated_Domain_Value: 11
          Enumerated_Domain_Value_Definition: Residential
```

```
Attribute_Domain_Values:
  Enumerated_Domain:
    Enumerated_Domain_Value: 12
    Enumerated_Domain_Value_Definition: Commercial and services
Attribute_Domain_Values:
  Enumerated_Domain:
    Enumerated_Domain_Value: 13
    Enumerated_Domain_Value_Definition: Industrial
    ...
```

Figure 6. Fragment of metadata describing the schema application of GIRAS landuse/landcover dataset of USA

Due to the importance of metadata, there is a need for easy-to-use metadata editors. For instance, ESRI incorporated in the ArcGIS 8 toolkit, a tool called ArcCatalog that enables the user to access, organize and search his/her geographic data. This tool includes a metadata editor (see Figure 7) based on FGDC that supports exportation to ISO 19115.



ArcCatalog - ArcView - C:\toni\doctorat\ontogis2005\maps_serra\serra.tif

Location: C:\toni\doctorat\ontogis200

Stylesheet: FGDC

Editing 'serra.tif'

● Enumerated Domain ○ Range Domain ○ Codeset Domain ○ Unrepresentable Domain

Value: 1

Value Definition: Dense forest area

Value Definition Source:

Enumerated Domain 1 of 20

Attribute Domain Value 1 of 1

Attribute 2 of 3

Detailed Description 1 of 1

```
            Low density forest area
    Enumerated_Domain:
        Enumerated_Domain_Value: 3
        Enumerated_Domain_Value_Definition:
            Transition area between pine forest and olive grove
    Enumerated_Domain:
        Enumerated_Domain_Value: 4
        Enumerated_Domain_Value_Definition:
            Rocky areas with little vegetation
    Enumerated_Domain:
        Enumerated_Domain_Value: 5
        Enumerated_Domain_Value_Definition:
```

Edit metadata for the selected item

Figure 7. Metadata editor of ESRI ArcCatalog

However, due to the recent approval of ISO 19109, the majority of editors for ISO metadata do not support schemas descriptions. This is the main reason why our tool OntoGIS only supports CSDGM at this moment.

Nevertheless, these schema definitions provided by ISO and FGDC standards do not consider semantics. In particular, they do not take into account possible relationships between different values used by different dataset providers. For instance, two land cover datasets may be structured according to different thematic categories; how they are related is not covered by the standards. Referencing these values to normalized vocabularies is an attempt to solve this problem. However, we will discuss in the following section that this approach incorporates other problems and other tools are needed to achieve semantic interoperability, namely ontologies.

Furthermore, these two standards comprise more than 300 elements. Completing all these elements require a highly qualified person and quite a lot of time (Nogueras-Iso et al. 2005). ISO 19115 defines a profile called "Core metadata for geographic dataset" that includes only 22 elements mainly oriented to be used in catalogues. Other organizations have used Dublin Core (ISO 2003b; DCMI 2005) to describe their datasets. None of these approaches covers the application schema.

Metadata is made public by means of catalogues, which incorporate distributed searching services. OGC has developed specifications for defining catalogue services (OGC 2004; 2005). Figure 8 shows a general view of the architecture of OGC catalogue services.

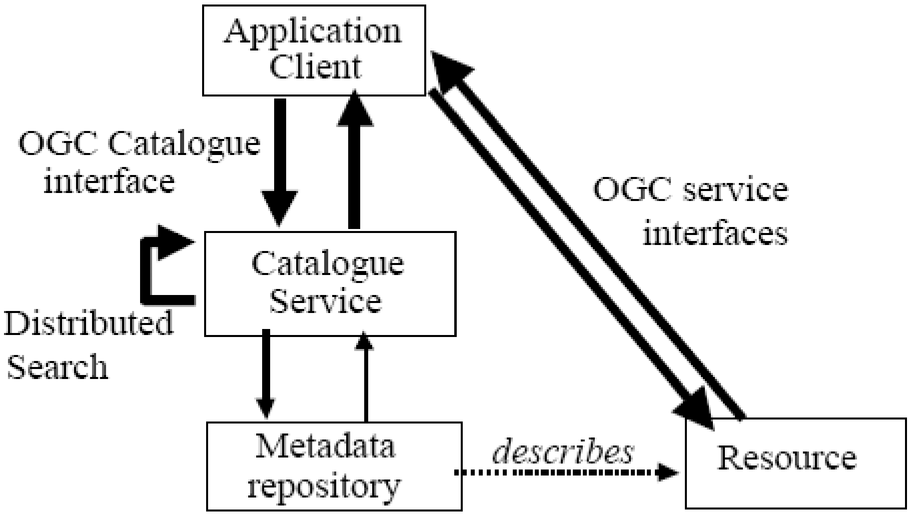


Figure 8. Role of the Catalogue Server in the OGC architecture. Image captured from http://www.opengeospatial.org

The notion of repository that we use in this thesis is very related to catalogues. However, catalogues give access to datasets metadata, but not to the datasets themselves which are accessed using other OGC specifications. Our repository contains references not only to metadata but also the dataset itself. This enables us to implement semantic services that access actual datasets. This way, in this thesis we will talk about repositories and not about catalogues.

3.2 Thesauri and normalized vocabularies

A solution sometimes proposed in other fields of knowledge is that the community agrees on a shared (hierarchical) list of concepts, typically implemented through a thesaurus. This way, every entity used in a dataset should refer to terms appearing in the shared vocabulary.

Let us now give a brief explanation of what a thesaurus is. A thesaurus is a hierarchically structured and controlled set of terms. Terms comprise one or more words and are organized through a set of standardized reciprocal relations among them. ISO 2788:1986 (ISO 1986) provides an international norm for the definition of mono-lingual thesauri, while ISO 5964:1985 (ISO 1985) covers multi-lingual thesauri, which are out of the scope of our work. Broader term (BT) and narrower term (NT) are two reciprocal relations between terms that structure the thesaurus in a hierarchical way, where the relation top term (TT) indicates the top of the broader terms for a given term in this hierarchy. Related terms (RT) is another reciprocal relation that indicates a relation different from the hierarchical. The other pair of reciprocal relations is use-instead (USE) and used-for (UF). These relations organize terms with a common meaning in two types: preferred and non-preferred. A non-preferred term contains its preferred term through the USE relation, while a preferred term contains its set of non-preferred terms through the UF relation. Only preferred terms participate in other relations. Note that these names of relations (BT, NT, TT, RT, USE and UF) are normalized by the ISO standard, and are present in all thesauri. Some include other relations, as DEF which provides a human-understandable textual definition for a term. For instance, the GEMET thesaurus for environmental terms (EEA 2001) also includes GROUP and THEME relations, which provide two classifications of terms in 30 groups and 40 general themes, to facilitate defining sets of the thesaurus according to particular interests. Figure 9 shows the term “wetland” and its relations in GEMET.

wetland
DEF Areas that are inundated by surface or ground water with frequency sufficient to support a prevalence of vegetative or aquatic life that requires saturated or seasonally saturated soil conditions for growth or reproduction. (Source: LANDY)
UF *humid zone*
THEME NATURAL AREAS, LANDSCAPE, ECOSYSTEMS
GROUP LAND (LANDSCAPE, GEOGRAPHY)
TT land
BT terrestrial area
NT bog; fen; marsh; moor; pond; pool; riparian zone; swamp

Figure 9. Definition of term “wetland” in GEMET

In the case of geographic information, there are no global shared vocabularies, but there are some initiatives that have defined different normalized hierarchical vocabularies of terms for specific domains. Land cover/land use is probably the most studied one.

Focusing on land cover/land use, in Europe we have to stress the CORINE project, which defines a vocabulary consisting of a hierarchy of three levels with 5, 15 and 44 items respectively (Bossard et al. 2000). It was used in the production of a dataset for land-cover in the EU at a scale of 1:100,000, considering areas of at least 25 hectares. The terms included in the vocabulary are oriented to that scale and minimum size of area, and do not fit properly other datasets at more detailed scales. In USA, the equivalent to CORINE is the land cover/land use classification used by the United States Geology Survey, which is an adaptation of the Anderson vocabulary (Anderson et al. 1976). Other well-known vocabularies have been defined by other organizations, as the vocabulary used in the framework of the International Geosphere Biosphere Programme (IGBP). The evaluation experiments for our mapping algorithms that will be described in Chapter 9 are based on merging these and other vocabularies.

Related to these vocabularies is the Land Cover Classification System (LCCS) (Di Gregorio and Jansen 1998; Di Gregorio 2005) developed by UN’s FAO. It permits the definition of land cover classes in terms of independent variables (or classifiers). LCCS has been used in the framework of developing land cover maps of several African countries. However, LCCS does not provide a normalized vocabulary of land cover classes, but a system to create new vocabularies, specifying how each vocabulary class is determined by different classifiers.

The Spatial Data Transfer Standard, SDTS (ANSI 1998), is a format for the exchange of geographical data standardized by ANSI. Its Spatial Features subset includes a vocabulary with 200 entities and 1200 alternative terms, focusing on topographic and hydrographic maps. However, it covers a very reduced domain, that is mainly oriented to vector maps, and does not contain concepts for thematic maps. Related initiatives are two big thesauri of geographical terms in the context of world-wide gazetteers: Getty thesaurus of geographic names[1] and the Gazetteer Content Standard[2] from the Alexandria Digital Library. As in the case of SDTS, these do not provide concepts typical from thematic maps.

Some thesauri on environmental information are also very relevant in our context. The most complete of them is the GEneral Multilingual Environmental Thesaurus, GEMET (EEA 2001), developed by the European Environment Agency. It is the result of merging the terms from a list of 8 previously existing thesauri in different languages. It contains around 6,500 terms that cover a wide spectrum of environmental concepts. As we have mentioned above, apart from the hierarchical structure, terms are also organized according to 30 groups and 40 general themes. Translation of terms to more than 15 languages is also provided. We have used GEMET in the context of the terminological mapping algorithm (see Chapter 7), and in particular, we have used it in the evaluation experiments of Chapter 9.

Nevertheless, none of these thesauri covers all the subfields of GI. Even GEMET, which is the widest, does not include, for instance, terms for geomorphology.

Furthermore, the use of thesauri as the basis of a semantic interoperability approach presents different problems. A first limitation is that thesauri as GEMET aim at defining thematic concepts, but its structure is not oriented to be used for organizing data sources. If we look at the following definition of the term “forest” in GEMET, we see that it has no broader term showing that forest is a specific type of vegetated area. This information is necessary for representing the semantic relations among values from different datasets, although not for defining terms.

[1] http://www.getty.edu/research/conducting_research/vocabularies/tgn/

[2] http://www.alexandria.ucsb.edu/gazetteer/

Forest
DEF A vegetation community dominated by trees and other woody shrubs, growing close enough together that the tree tops touch or overlap, creating various degrees of shade on the forest floor. It may produce benefits such as timber, recreation, wildlife habitat, etc. (Source: DUNSTE)
THEME FORESTRY; NATURAL AREAS, LANDSCAPE, ECOSYSTEMS
GROUP LAND (LANDSCAPE, GEOGRAPHY)
TT land
BT terrestrial area
RT forestry
NT coniferous forest; coppice; deciduous forest; forestry unit; indigenous forest; mixed forest; mountain forest; natural forest; primary forest; rain forest; temperate forest; timber forest; tropical forest; wood

Figure 10. Definition of the term "Forest" in GEMET

A second problem is that terms in a vocabulary or thesaurus have to be defined having a particular context in mind, in order to avoid ambiguities, which is the key issue for a good thesaurus. This way, for a thesaurus being clear and unambiguous, it has to be constrained to a specific context. For instance, CORINE focuses on a scale of 1:100,000 where the smallest considered spatial unit has an extension of at least 25 hectares. It is difficult that a general vocabulary or thesaurus contains themes related to very specific of local realities, as for example "*ferreret* habitats" (*ferreret* is a little amphibious endemic from Majorca) or "*ANEI*" (natural area of special interest). An approach strictly based on shared vocabularies/thesauri does not fit our needs of expressing the relations among thematic values from datasets from different contexts, since the thesaurus is itself biased by its own context.

A third problem is that thesaurus terms are defined through textual definitions. However, a reasoning engine cannot entail logical conclusions from these definitions. This limitation is particularly significant in the case of modelled themes, where the same term, for instance "land suitable for building" may have many different interpretations conditioned by local regulations on urban planning. Our representation of modelled themes by means of DL axioms enables us to use them with reasoning engines.

Summarizing these three problems, we can observe that an approach strictly based on normalized vocabularies/thesauri constraints the capability of dataset authors to describe their local real world, where they may need to capture specific concepts that have not been considered in the general vocabulary, or that may be based on different mental models, perhaps even conditioned by local regulations.

A fourth problem refers to the fact that a continous evolving domain as geosciences requires a continous evolving thesaurus, which presents problems of maintenance and of datasets dealing with different versions of the same thesaurus.

Finally, it has to be mentioned that a huge effort has been carried out in the field of medicine to build a big thesauri that tries to overcome these problems. In this context, the US National Library of Medicine has developed the Unified Medical Language System (UMLS[3]). It is based on a meta-thesaurus that contains over 1 million biomedical concepts and 5 million concept names from more than 100 controlled

[3] http://www.nlm.nih.gov/research/umls/

vocabularies and classifications for different uses of medicine data, some in multiple languages. In particular it includes SNOMED CT[4], an American standard with almost one million concept names in English and over 670,000 in Spanish. The meta-thesaurus also establishes relationships between terms from different source vocabularies.

We can observe that the thesauri available in the geographic domain are very far from UMLS and other solutions are necessary for semantic integration.

3.3 Ontologies for GI

As we have already mentioned, the integration of geographic information has a significant semantic component that cannot be addressed only with metadata standards and thesauri. A semantic framework representing the thematic component of geographic information is needed to facilitate sharing information. This semantic framework is usually expressed through ontologies since they provide explicit formal definitions of thematic concepts and their relations in datasets, and thus facilitates the definition of services for translating or integrating geographic information from different sources. This is in fact the approach of our work, where we define a formal ontology representing the thematic knowledge in a repository, a method for merging new knowledge from new datasets in the ontology, and finally services for finding, translating and integrating information.

The importance of ontologies as a tool for semantic interoperability has motivated a long list of research activities in the last years, from very different perspectives. Some of them focus on knowledge representation aspects, analyzing the problems of modelling geographic information through concepts and relations in ontologies. This often includes philosophical discussions on how space and geographic objects are perceived, and thus better represented. (Agarwal 2005) provides an exhaustive review of some of the different viewpoints and objectives of GI ontology modelling. This section firstly discusses some of the most relevant approaches on these issues, and then focuses on more practical ones. It has to be noted that, since our approach focuses on the thematic component of GI, those ontologies that have been specifically developed for representing space and topological relations in space are out of the scope of this work. And so are ontologies specifically designed for representing time and temporal relations.

Naïve Geography (Egenhofer and Mark 1995) studies the common-sense vision of the geographic world, that is how non-expert people think and reason about geographic space and time. This has motivated a line of research that tries to investigate ontological considerations in how people understand and categorize concepts (geographic kinds in their terminology) related to Geography (Smith and Mark 1998). In this work, they also discuss the particularities of the geographical domain that make it ontologically distinct from non-geographical domains. In (Smith and Mark 1999; Mark et al. 2001; Smith and Mark 2001) the authors conduct different experiments with non-expert subjects to observe how they classify different concepts, and how these concepts are related to top-level concepts as for instance “geographic features” or “something that could be

[4] http://www.snomed.org/

portrayed on a map". This work has little connection to our approach since it focuses on how high level concepts are perceived and represented, while we deal with more specific concepts.

Frank affirms in (Frank 1997) that no single geographic ontology can capture all the aspects of reality. Instead, small ontologies can be built for specific aspects of physical, cognitive, administrative or legal reality. These "small theories" have to be formally described in a way that makes it possible to combine them with other similar "small theories". This is in fact the approach that we have followed in our work, where each dataset and normalized vocabulary corresponds to a "small theory", while the repository ontology states the relations among them and provides a formal framework based on DL supporting the integration of data originally described through these "small theories".

Frank also suggests in (Frank 2001; 2003) that GI should be structured in a 5-tiers ontology. These five tiers are:

- Tier 0: human-independent reality (the physical world)
- Tier 1: observation of physical world
- Tier 2: objects and properties
- Tier 3: social reality
- Tier 4: subjective knowledge

Tier 0 describes the physical world through four-dimensional (3D space and time) continuous fields of attribute values, assuming that only one single physical world exists. Tier 1 contains the results of observing tier 0, assuming that observations of reality are limited. This tier provides measured values of properties at concrete locations in time and space. Tier 2 divides the observed world into objects with properties. Tier 3 obtains, from these objects representing the physical world, new objects according to social constructs. For instance, country borders, natural park or building zones are objects that are valid only within a social context. Finally, tier 4 is the subjective view of cognitive agents (persons and organizations). These agents have incomplete and partial knowledge of reality, but they use this knowledge to deduce other facts and take decisions. However, from our point of view this classification does not address the process of how people classify objects in entities or concepts at each level. And neither how different classifications (caused by different mental models) can be integrated. Furthermore, this theoretical construction can hardly be implemented because its complexity. For instance, representing in an ontology the physical world through fields cannot be achieved.

Kuhn defines the concept of Semantic Reference System (Kuhn 2003a). The spatial component of geographic data is represented through geographic coordinates, which are referred to a spatial reference system that specifies the geodetic datum, map projection and coordinate system. Methods for transforming data from one spatial reference system to another have been largely studied. Likewise, the temporal component of GI refers to a temporal reference system. Transformation methods between temporal reference systems also exist (for instance for transforming from one calendar to another). Semantic reference systems are conceived in a similar way as spatial and temporal reference systems. As he points out in (Kuhn 2003b) "users of geographic information should be able to refer thematic data to semantic reference systems, just as they refer geometric data to spatial reference systems". Furthermore, methods for projecting and translating data from different semantic reference systems are necessary. In (Kuhn 2003c) he implements a prototype of semantic reference system in Haskell functional language. This implementation supports the notions of semantic datum and semantic frame which are at the core of the reference system, as well as simple referencing, projection and transformation methods. It has to be noted that the idea of data referring to semantic reference systems is in fact very related to what we propose as our semantic framework, where thematic concepts used in datasets are referred to ontology classes, and translation and integration services are defined.

Special attention has received the representation of fields through ontologies, from both philosophical and practical points of view (Burrough and Frank 1996; Peuquet et al. 1998). The most relevant work in this area is (Kemp and Vckovsky 1998) that defined the concept of ontology of fields. They argued that the classical definition of a field as a function on a domain which is a subset of space-time is accurate, explicit and expressive, and provides access to the full set of mathematical tools for the characterization of fields. This way, an ontology of fields has to represent properties related to the following elements of the definition of a field:

- domain
- range
- association rule
- field as a whole

In particular, the thematic component is represented through the properties of the range. They characterize some properties that should be captured related to range: whether values are directly measured or derived, whether the scale of measurement is nominal or ordinal, and the dimension of the range (in case of ranges represented as vector spaces). However, we can observe that this corresponds to a very generic ontology, where these properties aim at describing the structure of the range, but not at representing the meaning of the values of a particular field.

Although not related to our work, it is worth mentioning here a particular type of geographic ontologies that focus on the interpretation of satellite or aerial images (Câmara et al. 2001). The authors point out the duality objects/field of an image: "while the domain scientist may believe she recognizes objects in a remotely sensed image, she is actually measuring fields". They propose a framework with three ontologies: one ontology, called physical ontology, that describes the image as a field in terms of low-level parameters; a second ontology, called structural ontology, that represents the concepts of the domain; and a third ontology which contains the interpretation algorithms that obtains objects of the structural ontology from the physical level.

Fonseca (Fonseca 2001; Fonseca et al. 2002) proposes the use of ontologies to conduct GIS developments and defines the concept of ODGIS (Ontology Driven GIS), following what Guarino defines as ODIS (Ontology Driven Information System) (Guarino 1998). ODGIS's are built using software components derived from various ontologies. These components are classes that can be used to develop new GIS applications. Being ontology-derived, these classes embed knowledge extracted from ontologies (Fonseca et al. 2000). This approach is oriented to the development of GIS and supposes the existence of a domain ontology which drives the process. But this is not useful in our case where we have to deal with existing datasets that have been created neither in the framework of the ODGIS nor following any domain ontologies.

Uitermark (Uitermark 2001; Uitermark et al. 2002) proposes a framework for semantic integration of datasets of the same domain. The framework comprises the application ontologies of the dataset and a domain ontology with general reference concepts. The framework also contains a set of surveying rules. Surveying rules are those that determine how a terrain is *transformed* into a geographic dataset, represented by means of a set of object instances. Finally, the domain ontology concepts are manually refined in order to reflect the concepts in the application ontologies, and all the semantic relations (equivalence, subclass/superclass, whole/part) between concepts are determined. Having this framework, the integration is carried out in two steps. In the first step pairs of overlapping objects from two datasets are obtained. In the second, these pairs are checked for consistency with surveying rules. This determines whether the objects represent the same physical object or not. All the objects that at the final of the process have not been matched indicate errors, either on surveying rules or on modelling rules. He has evaluated the framework with two datasets of land use with different scales structured according to different application ontologies, having around 700 and 300 objects each. The domain ontology is very reduced with six classes (building, road, railway, water, land, and 'otherland'). From our perspective, the most relevant contribution of this work is the consideration of surveying rules in the process of integration. However, the implementation of surveying rules is very dependant on the particular domain and can hardly be used in other types of thematic maps. And in our case, our repository may contain datasets for which we do not know their surveying rules. Furthermore, the process of checking consistency with surveying rules uses several matrix of *n*x*m* where *n* and *m* are the number of objects in both datasets. This cannot be used with bigger real datasets. For instance, the datasets that we have used in our evaluation (see Chapter 9) contain more that 10^8 spatial units. A final drawback of this approach is that the semantic relations between concepts have to be determined manually.

BUSTER (Visser et al. 2001) is a complete system developed by the University of Bremen to integrate and query heterogeneous information from different geospatial datasets. It supports queries of type concept@location and provides integration services at the syntactic, structural and semantic levels. Focusing on the semantic level, they define thematic and spatial knowledge in a dataset through what they call the Comprehensive Source Description, CSD (Visser and Schuster 2002; Visser et al. 2002a; Visser et al. 2002b). The spatial description consists of references to a gazzeteer, as Getty Thesaurus of Geographic Names. The thematic information is described through a terminological ontology based on references to some standardized vocabularies as WordNet, UpperCyc Ontology that includes about 3,000 terms of the most general concepts, GEMET that includes more than 5,000 terms on environmental disciplines, as well as other taxonomies on scientific knowledge, like for instance the Google Web Directory that comprises a classification of plants. This way, the merging phase (Schuster and Stuckenschmidt 2001) consists of an ontology expert that relates, in several steps, each thematic concept in a dataset with one or more terms of the standard vocabularies. As the usual result, a new class is created for the thematic concept in the terminological ontology, and is related to the standard taxonomies through axioms of type subclass or property value restriction. For example, the class “Coniferous-Forest” in CORINE is inserted in BUSTER in the following way:

```
class-def Coniferous-Forest
      subclass-of Geographical-Region
      slot-constraint vegetation value-type Coniferophyta
```

where “Geographical-Region” is a term that appears in UpperCyc, while “Coniferophyta” appears in the Google Web Directory. Once datasets have been described by means of CSDs and the terminological ontology is built, three types of queries are defined: spatial, terminological and combined spatio-terminological (concept@location).

The main difference from our approach is that in BUSTER all this work is done by the expert in a manual way, and there is no similarity or distance of any type that can guide the expert in this process. Furthermore, it assumes that all concepts can unambiguously be defined, and do not provide an equivalent to our modelled themes.

Similarity measures are often used to automate the merging process, as in our case. Similarity definitions are specifically addressed in the related work of Chapter 5, namely in Section 5.1. Particularly, some of the approaches discussed there focus on the geographic context: the asymmetric definition of similarity of Rodríguez and Egenhofer (Rodríguez 2000; Rodríguez and Egenhofer 2003; 2004); the work of Universidad de Zaragoza on semantic disambiguation in geothematic thesaurus in the context of spatial data infrastructures (SDIs) (Nogueras-Iso et al. 2004; Lacasta et al. 2005; Nogueras-Iso et al. 2005); and the definition of similarity based on contextual regions of (Schwering and Raubal 2005).

We now consider three different initiatives to merge dataset application ontologies. In principle they are not restricted to the geothematic context, but we include them here since the authors have conducted experiments with geothematic information. However, all of them require the intervention of an ontology expert to determine mappings between concepts.

(Cruz and Rajendran 2003) defines a simple model based on XML to represent one-to-one and one-to-many equivalence relations between classes in a hierarchical domain (as the geothematic one). A simple query mechanism that considers mappings to integrate data from two datasets is also provided based on XPath. However, mappings have to be determined in a manual way and only simple types of relations can be defined.

(Kavouras and Kokla 2002) defines a merging method based on formal concept analysis (FCA). FCA is based on a mathematical definition of concepts through lattices (more details can be found in (Ganter and Wille 1999)). The approach of Kavouras and Kokla defines a method that in 7 steps obtains an integrated lattice representing the thematic concepts in two datasets. However the first two steps have to be manually carried out. In step 1, the expert has to determine equivalences and overlappings between the classes in both datasets according to his/her knowledge of the domain. As a result, for each pair of overlapping classes s/he has to define a new class as subclass of both classes in the pair. In step 2, the expert also has to identify common attributes between both dataset classes.

The approach of Hakimpour (Hakimpour and Geppert 2001; Hakimpour and Timpf 2001; 2002) is based on specifying intensional definitions (through logical axioms) for thematic concepts in datasets. This way, merging two datasets can be seen as a process of matching these intensional definitions. Equivalence, specializations, overlapping and disjointness relations can be identified in this way. The requirement is that both dataset ontologies must refer to a shared higher level ontology. It has to be remarked that this is a theoretical approach that has not been implemented. Note that the task of providing intensional definitions cannot be conducted by any user but needs an expert on ontologies and logics. Furthermore, s/he has to manually determine mappings between concepts in dataset ontologies and concepts in the shared high level ontology.

A very different approach is proposed by (Wilson 2004), who defines the main elements that would be the basis for a logical framework describing spatial and thematic information. However, he focuses on how the information is spatially distributed in maps. The thematic component receives little attention and is considered as a flat set of values with no hierarchy. He defines a method for integrating maps, which is mainly based on statistical operators defined for the spatial component.

Finally, although not based on ontologies, we consider that two particular applied approaches related to land cover integration are of special interest. It is worth remarking that in Chapter 9, we will carry out some evaluation experiments with land cover datasets. (Comber et al. 2003; 2004) use statistical methods to characterize different types of discrepancies between a temporal series of land cover datasets. (Fritz and See 2005) tries to evaluate the discrepancy between two land cover raster maps with different legends (application ontologies). A group of experts determines through questionnaires how complex it is for them to differentiate between two classes from different datasets. These responses are used to define a fuzzy function that indicates the level of importance of a discrepancy of values in a cells. Furthermore, an expert determines what classes are allowed to overlap, and what overlappings are forbidden. All this information is used to generate a map that provides what they call “spatial agreement”. This map highlights the areas where significant discrepancies have been found and that may require further attention and possible re-mapping.

3.4 Lattices and other algebraic structures for GI integration

We discuss in this section three different approaches to integrate geographic information that are not implemented by means of ontologies. In the first subsection we analyze the approach of (Worboys and Duckham 2002) that defines an algebraic model. The second subsection presents an evolution of this work that relies on lattices (Duckham and Worboys 2005). Finally, the third subsection discusses the work carried out at the Université de Provence (Phan Luong et al. 2003; 2004) that is also based on lattices. The three approaches were originated in the framework of the European project Rev!GIS[5].

Although these approaches do not use ontologies, it has to be remarked that concepts in a TBox are usually thought of as having a lattice-like structure, according to the

[5] IST-1999-14189

subsumption relation. Consequently these constructions, lattice and ontology, are very related.

These approaches are very related to our third type of semantic query (integration). Due to that reason they are analyzed in further detail.

Before getting into the details, let us very briefly introduce here the definition of lattice, as well as other necessary concepts.

An *order set* is a pair $(M, \leq)$, where M is a set and $\leq$ an order relation on M. An *order relation* is a binary relation that fulfils reflectivity, antisymmetry and transitivity. If $\forall x,y \in M$, $x \leq y$ or $y \leq x$, then we say that the order set is *complete*; otherwise we say that it is a *partial order set*.

Let $(M, \leq)$ be an order set and A a subset of M. A *lower bound* of A is an element $s \in M$ such that $s \leq a$, $\forall a \in A$. Likewise, an *upper bound* of A is an element $t \in M$ such that $a \leq t$, $\forall a \in A$. If there exists a largest element in the set of all lower bounds of A, it is called the *infimum* of A, and is denoted by *inf A* or $\bigwedge A$. Likewise, if there exists a smallest element in the set of all upper bounds of A, it is called the *supremum* of A, and is denoted by *sup A* or $\bigvee A$. If $A=\{x,y\}$, we usualy write $x \wedge y$ to denote *inf A*, and $x \vee y$ to denote *sup A*. Infimum and supremum are also usually called *meet* and *join* respectively.

An order set $V = (M, \leq)$ is a $\wedge$-*semi-lattice* if $\forall x,y \in M$, $x \wedge y$ exists. V is a $\vee$-*semi-lattice* if $\forall x,y \in M$, $x \vee y$ exists. And finally, V is a *lattice* if $\forall x,y \in M$, both $x \wedge y$ and $x \vee y$ exist. V is called a *complete lattice* if $\bigwedge A$ and $\bigvee A$ exist for every A subset of M. Every complete lattice has a largest element (*top* or *unit element*) and a smallest element (*bottom* or *zero element*).

3.4.1 Worboys and Duckham 2002

(Worboys and Duckham 2002) proposes an algebraic model of the spatial and thematic information. A dataset is identified by means of a function that maps the space (S) to the thematic space (T).

$$f: S \rightarrow T$$

In this way, the dataset resulted of the integration of two datasets has its thematic space $T1 \otimes T2$ and its function $f1 \otimes f2$. Two projection functions $p1$ and $p2$ do the map from $T1 \otimes T2$ to $T1$ and $T2$ respectively.

$$\begin{tikzcd} & S \arrow[dl, "f_1"'] \arrow[d, "f_1 \otimes f_2"] \arrow[dr, "f_2"] & \\ T_1 & T_1 \otimes T_2 \arrow[l, "p_1"] \arrow[r, "p_2"'] & T_2 \end{tikzcd}$$

Figure 11. Integration of two datasets according to Worboys et al.

The computation of $T1 \otimes T2$ and $f1 \otimes f2$ will vary according to the structure of the thematic space of the original datasets. Three different possibilities are identified and described by the authors.

3.4.1.1 First case: no structure in the thematic spaces

In this case, the themes are atomic, with no hierarchy, and independent among them in the sense that there is no constraint related to their overlay. The resulting thematic space will be the Cartesian product:

$$T1 \otimes T2 = \{ (t1,t2) \mid t1 \in T1,\, t2 \in T2 \}$$

$$p1: (t1,t2) \mapsto t1$$

$$p2: (\mathrm{t1},t2) \mapsto t2$$

$$f1 \otimes f2: \mathrm{s} \mapsto (f1(s), f2(s))$$

3.4.1.2 Second case: partition structure in the thematic spaces

In this case, the two original datasets share the same set of thematic values, but each dataset uses a different partition of that set. Thus, the datasets map spatial units to sets of themes and not to atomic themes as in the first case. The solution here is based on the intersection of sets of themes:

$$T1 \otimes T2 = \{ t1 \cap t2 \mid t1 \in T1,\, t2 \in T2,\, t1 \cap t2 \neq \varnothing \}$$

According to this definition, the integration is only defined when the intersection is not empty. The projection and integration functions are defined in the following way:

$$p1: t1 \cap t2 \mapsto t1, \text{ where } t1 \in T1,\, t2 \in T2$$

$$p2: t1 \cap t2 \mapsto t2, \text{ where } t1 \in T1,\, t2 \in T2$$

$$f1 \otimes f2: s \mapsto f1(s) \cap f2(s)$$

3.4.1.3 Third case: hierarchical structure in the thematic spaces

In this case, each dataset has a hierarchy that structures its thematic space. The hierarchies of the two datasets are different, but they have in common the bottom level that is composed of a set of atomic values. Formally, the authors define a *U*-hierarchy as a subset of $\wp(U)$, where *U* is the set of all the atomic thematic classes containing all the singleton sets formed by the elements of *U*, and does not contain the empty set. Each of the datasets to be integrated conforms a different *U*-hierarchy, but they share the same set *U* of atomic themes.

Let *T* be the set of labels for the elements of the *U*-hierarchy *H*. The set of atoms in *U* labelled by *t*, where $t \in T$, is denoted by $t\alpha$. There is a partial order in *T* defined as:

$$t \leq t' \text{ iff } t\alpha \subseteq t'\alpha$$

Although *T* is not necessarily a lattice, since join and meet may not be closed, the meet (or infimum) operation can be defined in the following way:

$$t \wedge t' = t'' \text{ iff } t\alpha \cap t'\alpha = t''\alpha$$

Therefore, given two thematic hierarchies *T1* and *T2* on the same set *U* of atoms, the integrated thematic hierarchy is obtained in the following way:

$$T1 \otimes T2 = \{\ t1 \wedge t2 \mid t1 \in T1,\ t2 \in T2,\ t1 \wedge t2 \neq \bot\ \}$$

Note that a new element is added to the integrated hierarchy wherever the meet among two themes is not empty and does not exist yet in the hierarchy.

The integrated function can be defined in the following way, assuming that $f1(s) \wedge f2(s)$ will always be different from $\bot$:

$$f1 \otimes f2 : s \mapsto f1(s) \wedge f2(s)$$

However, as the authors point out, the projection functions

$$p1 : t1 \wedge t2 \mapsto t1, \text{ where } t1 \in T1,\ t2 \in T2$$

$$p2 : t1 \wedge t2 \mapsto t2, \text{ where } t1 \in T1,\ t2 \in T2$$

are not in general well-defined in the case multiple inheritance.

3.4.2 Duckham and Worboys 2005

(Duckham and Worboys 2005) is in fact a continuation of the previous section. The goal is again to define an algebraic model supporting the integration of two hierarchical datasets (third type of thematic space described above). However, the current approach is based on the extensional knowledge (the instances or individuals) in the datasets.

That means that the integrated thematic hierarchy is not built a priori according to the intensional knowledge, but can be obtained according to how the thematic values are distributed across the space of datasets. Thus, the constraint that determined that both *U*-hierarchies had to share the same set *U* of atomic terms is not necessary here.

The thematic hierarchy of each dataset is organized in a $\vee$-semi-lattice, having a top ($\top$) element. A dataset is modelled through what the authors call a classification, which is defined as a tuple:

$$C = (X, P, T, g)$$

where *X* is the spatial region, *P* is a partition of *X*, *T* is the thematic $\vee$-semi-lattice, and *g* is a function $g{:}P \rightarrow T$. Since the space *X* is usually considered fixed, the spatial component is characterized by *P*, and then, a classification is usually expressed with the tuple (*P*, *T*, *g*). Moreover, given two classifications *C1* and *C2*, the functions $\pi{:}P1 \rightarrow P2$ and $\tau{:}T2 \rightarrow T1$ can be defined, where τ has to preserve the lattice and it has to satisfy the statement $\tau\ g2\ \pi(p1) \leq g1(p1),\ \forall\ p1 \in P1$.

The extensional form of the thematic $\vee$-semi-lattice, *T'*, can be defined as the partial order set generated by the image of *g*:

$$T' = \langle Im(g) \rangle$$

If *g'* is the restriction of *g* to codomain *T'*, there is a morphism from *C*=(*P*, *T*, *g*) to *C'*=(*P*, *T'*, *g'*). Now the extension function is defined in the following way. Note that the definition in the article presents some errors, that are corrected here.

$$\begin{aligned} e: T' &\rightarrow \wp(X) \\ t &\mapsto \bigcup_{y \in \mathrm{Im}(g), y \leq t} g^{-1}(y), \text{ where } t = \bigvee_{y \in \mathrm{Im}(g), y \leq t} y \end{aligned}$$

Note that this means that given a theme $t \in T'$, the extensional function returns the set of areas that are mapped to the theme *t* (through function *g*):

$$e: t \mapsto \{\, x \in X \mid g(x) = t \,\}$$

The extensional classification is defined as *C** = (*P*, *T**, *g**), where *g**=*eg'* and *T**=*Im*(*e*). A classification is called regular if it is isomorphic to its extension.

If we have two regular classifications, *C1* and *C2,* that have to be integrated, we have the following diagram (assuming that *C1* and *C2* are already in its extensional form, all the * symbol have been dropped):

$$\begin{tikzcd}
T_1 \arrow[r, "\tau_1"] & T_1 \otimes T_2 & T_2 \arrow[l, "\tau_2"'] \\
P_1 \arrow[u, "g_1"] & P_1 \otimes P_2 \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[u, "g_1 \otimes g_2"'] & P_2 \arrow[u, "g_2"']
\end{tikzcd}$$

Figure 12. Integration of two regular extended classifications according to Duckham et al.

The partition of the integrated space and the integrated function are defined as:

$$P1\otimes P2 = \{\, p1 \cap p2 \mid p1 \in P1, p2 \in P2, p1 \cap p2 \neq \bot \,\}$$

$$g1\otimes g2\text{: } p1 \cap p2 \mapsto g1p1 \cap g2p2$$

The integrated taxonomy of themes, $T1\otimes T2$, is built in $Im(g1\otimes g2) \cup T1 \cup T2$. But in order for the result to be a $\vee$-semi-lattice, the Dedekind-MacNeille completion is used. This construction returns the smallest lattice L containing a partial order P as a subset, and it will be denoted as $L = DM(P)$.

$$T1\otimes T2 = DM(Im(g1\otimes g2) \cup T1 \cup T2)$$

Summing up, the integrated classification is defined as:

$$C1\otimes C2 = (\, P1\otimes P2, T1\otimes T2, g1\otimes g2 \,)$$

And in order to obtain the integrated classification from $C1$ and $C2$, the following steps are applied:

- Obtain $P1\otimes P2$, and the projections π1 and π2
- Obtain the extensional functions $e1$ (for each $t1 \in Im(g1)$) and $e2$ (for each $t2 \in Im(g2)$)
- Once having $e1$ and $e2$, obtain $T1^*$ (for each $t1 \in T1$), $T2^*$ (for each $t2 \in T2$), $g1^*$ (for each $p1 \in P1$) and $g2^*$ (for each $p2 \in P2$). We have in consequence $C1^*$ and $C2^*$. But note that since $C1$ and $C2$ are regular, $g1^*$ is equivalent to $g1$, $g2^*$ to $g2$, $T1^*$ to $T1$ and $T2^*$ to $T2$
- Once having $g1^*$ and $g2^*$, obtain the extensional form of $g1\otimes g2$ (for each $p \in P1\otimes P2$)
- Once having the extensional form of $g1\otimes g2$ (and in consequence its image), obtain the extensional thematic lattice $T1\otimes T2$

3.4.2.1 Main limitations

The main drawback of this approach is the fact of being based on a closed-world assumption. It assumes that if two datasets $D1$ and $D2$ are integrated and the set of all the areas for the class A in $D1$ is a subset of the set of all the areas for the class X in $D2$, then A is a subclass of X. However, assuming that this is true with these particular two datasets, it is not a universal truth. Enlarging the area of those datasets, or integrating the result with a third dataset can turn it to false.

A	*B*
B	*B*

Dataset D1

X	*X*
Y	*Y*

Dataset D2

Figure 13. Example of two raster datasets to be integrated

The previous figure shows two datasets of the same area, with the same four spatial units. In this case, A is set as a subclass of X, since the set of all the spatial units where A is present in $D1$ are a subset of the areas where X is present in $D2$. However, if we expand the area of the datasets as in the following figure, we see that A is no longer a subclass of X. This shows that this method based on the spatial component can generate false assumptions on the thematic space.

A	*B*	*A*
B	*B*	*B*

Dataset D1

X	*X*	*Y*
Y	*Y*	*Y*

Dataset D2

Figure 14. Example of two raster datasets to be integrated, with a larger area

And on the other way round, small cartographic errors or different sampling and generalizing methods can make that the set of areas for A is not a subset of X, although semantically A should be a clear subclass of X.

Other problem of this approach is related to the generation of the integrated thematic space. It only considers those classes that are used in the datasets (that have any area), but not their superclasses. Note that the partial order relation in the extensional form $T1 \otimes T2$ is ultimately generated from functions $e1$ and $e2$, which only consider the image of $g1$ and $g2$ respectively, and therefore do not take into account the other classes in the dataset hierarchies. This makes that the upper levels of the integrated lattice are often not precisely defined.

Finally, this work does not define the formalism to create the partial order relation on $Im(g1 \otimes g2) \cup T1 \cup T2$ before converting it to a $\vee$-semi-lattice by means of the Dedekind-MacNeille completion. Although it may be intuitively clear through the example, the algebraic construction is not complete enough to represent in a precise and formal way how the order relation in the integrated lattice is obtained.

3.4.3 Phan Luong et al.

The work of Phan Luong et al. (Phan Luong et al. 2003) is based on the use of lattices in order to integrate two thematic datasets. We describe here briefly the main constructions they use.

Let ***S*** be a non-empty set representing the space and its division, and ***I*** the set of the thematic classes. (***I***, ≤) is a partial order set, where they denote by ≤ the relation "less specific than": x ≤ y means that x is less specific or contains less information than y. Adding the ⊥ (top) and ⊤ (bottom) elements to ***I***, we have a complete lattice. Although it is more usual to consider the partial order relation in the other direction, being the top class the most generic, we will maintain here the notation used by the authors.

Two operations that will be used for integration are defined in the lattice, consensus (⊗) and aggregation (⊕). Let *I* and *J* be subsets of ***I***:

$$I \otimes J = max(\{x \wedge y \mid x \in I, y \in J\})$$
$$I \oplus J = max(\{x \vee y \mid x \in I, y \in J\})$$

On the one hand, consensus operation computes the information that *I* and *J* have in common. The result is more generic than *I* and *J*. On the other hand, aggregation computes the information that, being more specific than *I* and *J*, does not contain any conflicts among their elements. Thus, the result is more specific than *I* and *J*.

An information source ***D*** is defined as the triple (*P*(***S***), *C*(***I***), *R*) where *P*(***S***) is a finite covering of ***S***, *C*(***I***) is a finite collection of subsets of ***I*** and *R* is a binary relation between *P*(***S***) and *C*(***I***). The deduced function *f* is defined as follows:

$$f\colon S \rightarrow \wp(\boldsymbol{I})$$
$$x \mapsto \{i \in \boldsymbol{I} \mid \exists\ (X,I) \in R, x \in X, i \in I\ \}$$

This function maps the space units of the datasets to a set of the thematic classes defined in the lattice. This way, two data sources are equivalent if every spatial element is mapped to the same set of themes.

The process of integrating two data sources *D1* and *D2*, identified with their deduced functions *f1* and *f2*, will result into a new data source *D*=(*P*(***S***), *C*(***I***), *R*), with a deduced function that satisfies:

$$\forall p \in \boldsymbol{S}, f1(p) \otimes f2(p) \sqsubseteq f(p) \sqsubseteq f1(p) \oplus f2(p)$$

where $\sqsubseteq$ is defined as:

$$I \sqsubseteq J, \text{ if } \forall x \in I, \exists y \in J \text{ such that } x \leq y$$

where *I* and *J* are non-empty subsets of ***I***.

From the definition of the integration above, two integration functions can be easily obtained. A pessimistic function uses $f1(p) \otimes f2(p)$, while an optimistic function takes $f1(p) \oplus f2(p)$. Any intermediate possibility is of course also valid, but the authors do not provide a way to calculate it.

(Phan Luong et al. 2004) extends this construction in order to consider also a quality measure in the process. They present an example with the UK Land Cover Map of 2000, where this quality value is obtained from the set of thematic classes that are present in a spatial unit, in percentage terms. However their approach is specific to this case and hardly extensible to other situations.

4 Formal conceptual model: an ontology for representing Thematic GI

The core of the semantic framework is a formal ontology that represents, using the constructors of Description Logics, the thematic knowledge in a repository of datasets. This ontology, as argued previously, has to deal with geographic concepts or themes (either qualitative, quantitative or modelled), datasets and their values, vocabularies and their terms, as well as the relations among all of them.

As an implementation of the semantic framework, we have developed a tool that we have called OntoGIS (see Appendix A for a description of the tool). It covers the edition of themes, datasets and vocabularies, and goes beyond the management of the ontology by also supporting the merging process and the three types of semantic queries. The tool represents the ontology in OWL, more particularly using the DL profile. It has been programmed using the HP Jena toolkit, which is an API for processing RDF and OWL. Jena includes its own reasoning engine, which only supports some limited inference services, and can be connected to external DIG reasoners such as FaCT or Racer.

Our ontology contains concepts (or classes), roles (or properties or slots), logical axioms and individuals (instances), since it is based on Description Logic. In this chapter we discuss the organization of the main classes and properties of our ontology. We include their OWL definition, since OWL provides a formal way to represent them, more precise than UML diagrams, and at the same time easier to read than a notation consisting of logical axioms (especially in what respects to property types and domain and range of properties). In the following sections we describe the different parts of the ontology. The complete OWL document can be found in Appendix B, where an image of the structure of the classes seen in Protégé 3.0[1] is also shown. Our ontology was partially described in (Navarrete et al. 2004).

4.1 Datasets and values

The repository consists of a set of datasets. Two files are related to a dataset, the source data, and the metadata files. The metadata file could be in principle expressed in any of the standards described in Chapter 3, although our tool currently only supports FGDC CSDGM.

The metadata file contains all the information that we need from a dataset. We have selected the most relevant elements for our context, which will be added to the

[1] http://protege.stanford.edu

ontology. They are abstract, purpose, theme keyword, the UTM zone number (to simplify we assume that UTM is the horizontal coordinate system used), the bounding coordinates, and finally its internal entity-attribute schema, which contains the attributes and the values they may take. In fact, as we have already explained in Chapter 1, we assume that there is only one attribute with thematic information per dataset, and therefore we only store the values for this main thematic attribute. Furthermore, a property has to be added to the ontology to record the name of the thematic attribute of a dataset. Note that in the case of a dataset containing several thematic variables, it can be represented as different logical datasets, one for each thematic variable.

Datasets are represented through the ontology class *Dataset*. An individual of the class is created for each dataset in the repository. It has the following properties:

- *datasetTitle*: the title that identifies the dataset.
- *datasetURI*: contains the path for the source data file.
- *datasetMetadataURI*: contains the path for the metadata file. This file contains all the elements that describe the dataset, including those regarding the following properties
- *datasetAbstract*: contains the abstract read from the metadata file
- *datasetPurpose*: contains the purpose read from the metadata file
- *datasetThemeKeyword*: contains the keyword describing the main theme of the dataset. It is read from the metadata file
- *datasetUTMZone*: the number of the UTM Zone
- *datasetBoundingNorth*: contains the North in the bounding coordinates, read from the metadata file
- *datasetBoundingSouth*: contains the South in the bounding coordinates, read from the metadata file
- *datasetBoundingEast*: contains the East in the bounding coordinates, read from the metadata file
- *datasetBoundingWest*: contains the West in the bounding coordinates, read from the metadata file
- *datasetThematicAttribute*: contains the name of the attribute that contains the thematic information of the dataset

We have distinguished among quantitative and qualitative datasets. They are represented in the ontology through two subclasses of *Dataset*, *QuantitativeDataset* and *QualitativeDataset* respectively. However, they have the same properties, assigned to the class *Dataset* and inherited by both. But it is useful to make this distinction because they have different connection procedures. Qualitative values can only be connected to qualitative themes while quantitative values can only be connected to quantitative classes (from a classification of a quantitative theme).

The following fragment of OWL document provides a formal definition of classes and properties related to qualitative and quantitative datasets:

```
<owl:Class rdf:ID="Dataset"/>
<owl:Class rdf:ID="QualitativeDataset">
      <rdfs:subClassOf rdf:resource="#Dataset"/>
</owl:Class>
```

```
<owl:Class rdf:ID="QuantitativeDataset">
      <rdfs:subClassOf rdf:resource="#Dataset"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="datasetTitle">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetURI">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetMetadataURI">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetAbstract">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetPurpose">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetThemeKeyword">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetUTMZone">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#integer"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetBoundingNorth">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
```

```
... (the same for datasetBoundingSouth, datasetBoundingEast and
datasetBoundingWest)
<owl:DatatypeProperty rdf:ID="datasetThematicAttribute">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

The values for the main thematic attribute are also read from the metadata file, namely from the "entity and attribute" section. The ontology class *DatasetValue* is used to represent these attribute values. An individual of this class is created for each value. The class has the following properties:

- *valueName*: the actual value that is stored in the source data file, usually a numeric value. It is read from the metadata file of the dataset.
- *valueDefinition*: a textual definition for the value. It is read from the metadata file of the dataset.
- *valueDataset*: a reference to the dataset individual.

The following fragment of OWL document provides a formal definition of the classes and properties related to dataset values:

```
<owl:Class rdf:ID="DatasetValue"/>
<owl:DatatypeProperty rdf:ID="valueName">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#DatasetValue"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="valueDefinition">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#DatasetValue"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="valueDataset">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#DatasetValue"/>
      <rdfs:range rdf:resource="#Dataset"/>
</owl:ObjectProperty>
```

We also define a new class in order to represent qualitative values. A dataset value individual is a qualitative value if and only if its role *valueDataset* is a reference to a qualitative dataset. Thus, the *QualitativeDatasetValue* class is defined as:

```
      QualitativeDatasetValue ≡
                  DatasetValue ⊓ ∀valueDataset.QualitativeDataset
```

The previous DL expression can be written in OWL as:

```
<owl:Class rdf:ID="QualitativeDatasetValue">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#DatasetValue"/>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="valueDataset"/>
            </owl:onProperty>
            <owl:allValuesFrom>
              <owl:Class rdf:ID="QualitativeDataset"/>
            </owl:allValuesFrom>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
</owl:Class>
```

QuantitativeDatasetValue class is defined in a similar way:

```
      QuantitativeDatasetValue ≡
                  DatasetValue ⊓ ∀valueDataset.QuantitativeDataset
```

And in OWL:

```
<owl:Class rdf:ID="QuantitativeDatasetValue">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#DatasetValue"/>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="valueDataset"/>
            </owl:onProperty>
            <owl:allValuesFrom>
              <owl:Class rdf:ID="QuantitativeDataset"/>
            </owl:allValuesFrom>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
</owl:Class>
```

From a very strict point of view, it would be necessary to add another class, *DatasetArea*, in order to reflect the spatial units in a dataset, either cells in a raster dataset or geometrical features in a vector dataset. This class would have properties to represent the geographical location and the value associated to it in the dataset. However, a dataset may contain millions of spatial units, especially in the case of raster datasets. This would make the management of the ontology extremely inefficient. Thus, we keep this information in the dataset and read it from there when it is necessary during the processes of merging and query. We will see how geographic themes are connected to values in a dataset in the next sections. We will also see how this in fact represents not only a dataset value but all the areas in the dataset that have that value and thus are linked to a particular theme. Thus, a query can obtain the datasets values related to a theme, and reading from the dataset files, get the individual spatial units associated to that theme.

4.1.1 Abstract values

We have already discussed in 1.2.2 that the values of qualitative datasets can often be grouped building a hierarchy. This hierarchy is sometimes made explicit in the legend of the maps. But in fact, only its leaf nodes exist physically in the data file. Consequently, we refer to them as *physical* values, while *abstract* values are those that have been added in upper levels to group concepts, forming the hierarchy. In some particular cases, abstract values are also added below the physical ones (see aggregations and mixtures in Chapters 6 and 7). The top of the hierarchy corresponds to the main theme of the dataset, that has been read from the metadata file and identified with the property *datasetThemeKeyword* of the class *Dataset*. Note that it is also an abstract value since it does not appear in the data file.

It is worth noting here that hierarchies of values of a dataset are not supported by metadata standards, and therefore they are not read from the metadata file but are instead created in the OntoGIS tool.

This hierarchy of values can be very useful for merging the dataset into the ontology. The semi-automatic merging method (see Chapter 5) uses this hierarchy in order to check similarities in the structures of ontology and dataset. On the other hand, since quantitative datasets are not merged in the same way, as explained earlier, abstract values will not be defined for them. This way, the class *AbstractDatasetValue* is defined as a subclass of *QualitativeDatasetValue* to represent abstract values.

Moreover, a property *valueChildOf* is added to qualitative themes in order to build the hierarchy. This property contains a reference to the more generic value in this dataset (refered as parent). Note that the parent of a value is a generic *DatasetValue* , since although usually is an abstract value it may also be a physical one. The property is defined as transitive so that the reasoner is able to get all the ancestors of a given value, which is necessary for the merging process.

The following fragment of OWL document provides a formal definition of *AbastractDatasetValue* class and its properties:

```
<owl:Class rdf:ID="AbstractDatasetValue">
      <rdfs:subClassOf rdf:resource="#QualitativeDatasetValue"/>
</owl:Class>
<owl:TransitiveProperty rdf:ID="valueChildOf">
      <rdfs:domain rdf:resource="#QualitativeDatasetValue"/>
      <rdfs:range rdf:resource="#DatasetValue"/>
</owl:TransitiveProperty>
```

4.2 Geographic Themes

Each geographic theme is represented by means of an ontology class. There is a class called *Theme*, which subsumes every thematic class. This means that *Theme* is a superclass of any thematic class, or in other words, that it is the top of the taxonomy of classes.

A thematic class is identified by a name and uses the namespace of OntoGIS by default. For instance, the theme *Forest* is represented through the class *http://www.upf.edu/ontogis#Forest*. Since URIs cannot contain blank spaces, accents or other characters, a property *themeName* is assigned to thematic classes in order to store their name. For instance, the theme *Pine tree forest* is represented through the thematic class *http://www.upf.edu/ontogis#Pine_tree_forest*, whose property *themeName* has the value “Pine tree forest” (a string). This is formally defined in OWL in the following way:

```
<owl:Class rdf:ID="Theme"/>
<owl:DatatypeProperty rdf:ID="themeName">
      <rdfs:domain rdf:resource="#Theme"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

In addition, OWL classes also have a property *Documentation*, that we use to add a textual definition, understandable by humans, to every thematic class.

The *Theme* class has two direct subclasses, each representing the different types of themes that can be used in our context:

- *QuantitativeTheme* used to represent a quantitative theme
- *QualitativeTheme* used to represent a qualitative theme

This way, every thematic class is a descendant of (or it is subsumed by, in DL terms) one of these two classes, depending on its type.

4.2.1 Qualitative themes

A qualitative theme is represented by an ontology class that is a subclass of *QualitativeTheme*.

A qualitative thematic class can be related to other qualitative themes. The main relation is subclass, or subsumption in DL terms. A qualitative thematic class is a subclass of another one if it represents a more specific theme. This means, in DL terms, that any individual of the subclass is also an individual of the superclass. For instance, an individual of the class “pine forest” is also an individual of the class “forest”, while there may be other individuals of “forest” that are not individuals of “pine forest”. The subsumption relation conforms a taxonomy of the qualitative themes according to their genericity.

Any other DL relation can be set among different themes besides subsumption. For instance, two themes can be set to be equivalent or disjoint. On the other hand, more complex relations (such as intersection, union and complement) can also be set by means of modelled themes, as it was explained in Section 1.2.4. The formal representation of modelled themes is provided in 4.2.4.

Finally, a qualitative theme may also have property values *restrictions*. As an example, let us consider that the *Forest* class has a property *mainTypeOfTree*, and a subclass *Pine forest*. The property is inherited by *Pine forest*, which might include the restriction setting that its value is "*pine tree*" for each of its individuals. Note that if two classes restrict a functional property with different values, they will be necessarily disjoint. From a practical point of view, this permits the expert to define different coverings (or partitions, since the classes will be disjoint if the property is functional) of a class made of different subclasses. Following the previous example, the class *Forest* may have a partition according to the property *mainTypeOfTree*, comprinsing all its subclasses that restrict the value of *mainTypeOfTree* to a particular type of tree. On the other hand, *Forest* may have another partition according to the *density* property, comprising subclasses *High density forest*, which restricts the value of the *density* property to "*high*", and *Low density forest*, which restricts its value to "*low*".

An important issue to be considered relates to when qualitative thematic classes are realized. The realizations of the themes are in the datasets. From a conceptual point of view, each spatial unit in a dataset that refers to the theme could be considered an individual of the theme. However, as indicated previously, the number of spatial units in a dataset is too big to be managed inside the ontology. To solve this, we consider that an individual of a qualitative theme represents the realization of the theme through a value of the thematic variable of a dataset. This means that each time a thematic class is connected with a value in a dataset, a new individual of the thematic class is created.

An individual of a qualitative theme may be connected to a dataset value by means of a role (property in OWL terminology), called *qualitativeThemeConnection*. Its range is the class *QualitativeDatasetValue*, where its domain is the class *QualitativeTheme*, and consequently any of its subclasses. This way, an individual of a qualitative thematic class is related to a specific individual of a value in a qualitative dataset. Note that, since we have identified an individual of a thematic class as the realization of the theme through a dataset value, this property is functional, i.e., an individual of *QualitativeTheme* is connected to at most one dataset value.

The following fragment of OWL document provides a formal definition of classes and properties related to qualitative themes:

```
<owl:Class rdf:ID="QualitativeTheme">
      <rdfs:subClassOf rdf:resource="#Theme"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="qualitativeThemeConnection">
      <rdf:type rdf:resource=
          "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QualitativeTheme"/>
      <rdfs:range rdf:resource="#QualitativeDatasetValue"/>
</owl:ObjectProperty>
```

Connections can be set manually, value by value, or semi-automatically through the mapping algorithms, all the values set at the same time. Both methods are described in detail in Chapter 5, while the specific mapping algorithms are discussed in Chapters 6 to 8.

It is important to note that as soon as a theme is connected to a particular value, all the superclasses of the theme become connected too. Since the individual of the value of the theme taking part in the connection is also an individual of all its superclasses, then these superclasses are also connected in an indirect way. For instance, if we have the theme *Pine forest*, that is a subclass of *Forest*, when *Pine forest* is connected to a particular value *v1* in a dataset, a new individual (for instance called *pf1*) is created for the class *Pine forest*:

```
PineForest ⊑ Forest ⊑ ... ⊑ QualitativeTheme
QualitativeDatasetValue(v1)
PineForest(pf1)
qualitativeThemeConnection(pf1,v1)
```

and since *pf1* is also an individual of the class *Forest*, we can infer that *Forest* is connected with this dataset too.

However, users are usually interested in a distinction between “direct” and “inferred” connexions. This motivates variations in the semantic queries either considering inference or not, as we will further describe in Chapter 10. An special case appears when the connected value is abstract. Sometimes it will be interesting to get not just the abstract value, but all its descendant physical values. The reasoner can easily do get them because the property *childOf* is transitive.

4.2.2 Quantitative themes

Reasoning with quantitative themes is simpler than with qualitative ones, becasue quantitative themes are not semantically related to each other. An integration between two dataset of the same quantitative theme would be meaningless, since if a particular spatial unit has two different values in these datasets, no logical conclusion can be achieved. Consequently, we will see that the semantic query that integrates two or more datasets (third type of semantic query that will be formulated in Section 10.3) is only defined for qualitative datasets, and not for the quantitative ones. Nevertheless, we will also see that quantitative themes may be used in the definition of modelled themes, and an integration query focusing on a modelled theme may require the use of quantitative datasets.

From a semantic point of view, the most relevant issue concerning a quantitative theme is how it is classified in a set of intervals or classes. Reasoning with quantitative themes is in fact reasoning with the classes that are part of the classification. In our approach we only consider crisp classifications, where each interval is defined by its minimum and maximum values. Let us remark that a quantitative theme may have any number of different classifications and each dataset or each user may use its own.

Sometimes a quantitative class may be contained in another quantitative class of a different classification, and our model could reflect this by setting a subclass relation between them. However, the usual case is that each class in a classification overlaps with two or more classes of another one, and this approach would be absolutely insufficient. More complex constructions, probably based on fuzzy logic, would be needed to transform a dataset from one classification to another. This is out of the scope of Description Logic, and our work does not deal with either transforming a dataset to other classification or integrating two quantitative datasets of the same theme. Only when quantitative themes take part in the definition of a modelled theme through a logical axiom, they may have a semantic interest related to integration.

A quantitative theme is represented by an ontology class that is a subclass of *QuantitativeTheme*, while we represent its classification by means of the ontology class *QuantitativeClassification*. A quantitative classification has the following three properties:

- *quantitativeClassificationName*: each classification is given a name to identify it.
- *quantitativeClassificationUnit*: identifies the name of the units of measure that are used to define the different intervals of the classification (for instance Celsius degrees in temperature)
- *quantitativeClassificationTheme*: a reference to the quantitative theme class.

Note that this last property should relate a particular individual of a classification to the quantitative theme being classified. This means that the individual of classification should be related not with an individual of the theme but with the whole ontology class. In logical terms, the filler of the role should be a class. However this comes out of Description Logic, where the filler of a property has to be an individual, not a class.

It has to be remarked that the full version of OWL supports metaclasses. A metaclass is a class having another classes as its individuals. Particularly, any class is an individual of a class called *Class*. The use of metaclasses would enable us to set *Class* as the range of the property, and the value for a particular individual would be a particular ontology class. However, since this goes out of Description Logic, DL reasoners cannot deal with it. For instance, Racer directly discards a class having a property with classes as values. Therefore, we avoid the use of these features of OWL Full.

W3C has published a technical report (W3C 2005) describing several different possibilities in order to represent this type of situations in DL. None of them is perfect and all of them have different drawbacks.

In our case, there is not a taxonomy in the quantitative themes, because they do not subsume one another. This fact simplifies the requirements for reasoning, since the relation among a theme and a classification does not have to be inferred to other superclasses. However, we will see in 4.2.4 that a quantitative theme may be used in modelled themes, and in consequence, it may have other superclasses according to the logical definition of the model.

The solution that better fits our needs is the same that will be used for modelled themes: the classification is connected to an “ideal” individual of the theme. This may be a little confusing since this “ideal” individual does not represent the realization of the theme in a dataset, but a kind of idealized or archetypical representation of the theme. The drawback is that we will have to deal with both “real” individuals representing a realization in a dataset, and “ideal” individuals in order to represent the theme for classification. However it is not difficult to differentiate them, since “ideal” individuals are linked to quantitative classifications through the role *quantitativeThemeClassification*, while “real” individuals never will. On the other hand, we will see in the next section that a “real” individual may be linked to a dataset value through the role *quantitativeClassConnection*, while an “ideal” individual cannot be.

Furthermore, a role *quantitativeThemeClassification* is added to *QuantitativeTheme* to manage the relation between an “ideal” individual of *QuantitativeTheme* and its quantitative classification.

A quantitative classification is composed of several intervals or classes, where each interval is defined by its thresholds and is given a name to be identified. The ontology class *QuantitativeClassDescription* represents an interval or class in a quantitative classification. The number of intervals in a classification is not limited. In order to distinguish between a class in the ontology and a class in a classification, we will respectively use “ontology class” (or simply “class”) and “quantitative class” to denote them.

A quantitative class has the following properties:

- *quantitativeClassName*: each quantitative class is given a name in order to identify it. This name usually has a meaning for a human as ‘cold’ or ‘warm’ in the example of temperatures
- *quantitativeClassMinimumValue*: this identifies the lower threshold of the quantitative class. It is expressed in the units of measure defined in the quantitative classification
- *quantitativeClassMinimumOpen*: this represents whether the minimum value is included in the interval (closed interval) or not (open interval)
- *quantitativeClassMaximumValue*: this identifies the upper threshold of the quantitative class. It is also expressed in the units of measure defined in the quantitative classification
- *quantitativeClassMaximumOpen*: this represents whether the maximum value is included in the interval (closed interval) or not (open interval)
- *quantitativeClassClassification*: a reference to the quantitative classification

Since quantitative classes can be used to form logical expressions for modelled themes (see 4.2.4), each quantitative class has also to be modelled by means of an ontology class. Therefore, each time a quantitative class is added to a classification, a new individual of the class *QuantitativeClassDescription* is created, and a new ontology class is also created to represent this quantitative class. The new ontology class will be a subclass of both the related quantitative theme and the ontology class *QuantitativeClass*. For instance, when a quantitative class called *warm* is added to the classification *classification1* of the theme *Temperature*, a new ontology class is created as subclass of *Temperature* and *QuantitativeClass*. To assure the uniqueness of the URIs, we concatenate the names of the quantitative theme, classification and class in the URI of the new ontology class, as in *http://www.upf.edu/ontogis#Temperature-classification1-warm*. All the classes of the same classification are set as mutually disjoint.

Note that any subclass of *QuantitativeClass* will also be a subclass of *QuantitativeTheme*, since it is specializing its meaning. Therefore we can entail that *QuantitativeClass* is a subclass of *QuantitativeTheme*.

The class *QuantitativeClass* has a property *quantitativeClassDescription* that refers to an individual of *QuantitativeClassDescription* that contains the description of the quantitative class. The ontology class for the interval has a restriction setting that the value for this property is the corresponding individual of *QuantitativeClassDescription*.

The following fragment of OWL document provides a formal definition of classes and properties related to quantitative themes:

```
<owl:Class rdf:ID="QuantitativeTheme">
      <rdfs:subClassOf rdf:resource="#Theme"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="quantitativeThemeClassification">
      <rdf:type rdf:resource=
          "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeTheme"/>
      <rdfs:range rdf:resource="#QuantitativeClassification"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="QuantitativeClassification"/>
<owl:DatatypeProperty rdf:ID="quantitativeClassificationName">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassification"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="quantitativeClassificationUnit">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassification"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="quantitativeClassificationTheme">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassification"/>
      <rdfs:range rdf:resource="#QuantitativeTheme"/>
      <owl:inverseOf rdf:resource="#quantitativeThemeClassification"/>
</owl:ObjectProperty>
```

```
<owl:Class rdf:ID="QuantitativeClassDescription"/>
<owl:DatatypeProperty rdf:ID="quantitativeClassName">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassDescription"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="quantitativeClassMinval">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassDescription"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="quantitativeClassMaxval">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassDescription"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="quantitativeClassClassification">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassDescription"/>
      <rdfs:range rdf:resource=
            "#QuantitativeClassificationDescription"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="QuantitativeClass">
      <rdfs:subClassOf rdf:resource="#QuantitativeTheme"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="quantitativeClassDescription">
      <rdf:type rdf:resource=
          "http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClass"/>
      <rdfs:range rdf:resource="#QuantitativeClassDescription"/>
      <owl:inverseOf rdf:resource="#quantitativeClassClassification"/>
</owl:ObjectProperty>
```

We can observe that the semantics of the quantitive thematic content is encapsulated by means of quantitative classes or intervals and their thresholds. A quantitative dataset value is not represented by a quantitative theme, but by one of its quantiative classes. This way, a property called *quantitativeClassConnection* is used to connect quantiative dataset values to quantitative classes. As in the case of qualitative themes, storing an individual of the quantitative class for each related spatial unit in a dataset is not feasible. And again as in the case of qualitative themes, we consider that an “ideal” individual of a quantitative class represents the realization of the class through a related value of a dataset. This means that each time a quantitative class is connected to a dataset value, a new individual of the quantitative class is created.

The range of *quantitativeClassConnection* property is the class *QuantitativeDatasetValue*, while its domain is the class *QuantitativeClass*. As in the case of *qualitativeThemeConnection*, this property is functional as far as we have identified an individual of a quantitative class as the realization of the class through a particular dataset value. This means that an individual of *QuantitativeClass* may be connected to at most one individual of *QuantitativeDatasetValue*.

4.2.3 Connections

A role *themeConnection* can be defined in order to group all the connections, being either qualitative or quantitative. In a language with role constructors ($\mathcal{R}$) a connection can be defined as the union of *quantitativeClassConnection* and *qualitativeThemeConnection*:

```
      themeConnection ≡
            quantitativeClassConnection ⊔ qualitativeThemeConnection
```

However, neither OWL nor most DL reasoners provide this constructor. On the other hand, OWL supports RDFS capability of defining role hierarchies by means of the *subPropertyOf* construction, which in our case has a similar effect. Reasoners as FaCT or Racer do support role hierarchies.

It is also important to define an inverse property for connections that reflects the qualitative themes or quantitative classes connected to a given dataset value. *datasetValueConnection* is declared as the inverse of *themeConnection*. Note that this new property could also be specialized in two subproperties, *qualitativeDatasetValueConnection* and *quantitativeDatasetValueConnection*, that would be respectively the inverse of *qualitativeThemeConnection* and *quantitativeClassConnection*. However, they are unnecessary and are not included in the model: on the one hand, using *datasetValueConnection* is enough to get the themes connected to a particular value (either qualitative or quantitative); and on the other hand, the model is already restricted to the requirements by means of the range and domain of the existing properties and consequently it will not allow to have a quantitative dataset value connected to a qualitative theme or vice versa. Note also that *datasetValueConnection* is also a functional property since given a dataset value it can be connected to at most one theme or class.

According to these considerations, properties involved in connections are now formally defined in the following way:

```
<owl:ObjectProperty rdf:ID="themeConnection">
      <rdf:type rdf:resource=
          "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Theme"/>
      <rdfs:range rdf:resource="#DatasetValue"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="qualitativeThemeConnection">
      <rdfs:subPropertyOf rdf:resource="#themeConnection"/>
      <rdf:type rdf:resource=
          "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QualitativeTheme"/>
      <rdfs:range rdf:resource="#QualitativeDatasetValue"/>
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="quantitativeClassConnection">
      <rdfs:subPropertyOf rdf:resource="#themeConnection"/>
      <rdf:type rdf:resource=
          "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeTheme"/>
      <rdfs:range rdf:resource="#QualitativeDatasetValue"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="datasetValueConnection">
      <rdf:type rdf:resource=
          "http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
      <owl:inverseOf rdf:resource="#themeConnection"/>
      <rdfs:domain rdf:resource="#DatasetValue"/>
      <rdfs:range rdf:resource="#Theme"/>
</owl:ObjectProperty>
```

It has to be noted that these definitions do not take into account the restriction that all the values in a dataset should be connected to quantitative classes of the same classification. However, this restriction is programmed in the tool and checked at the time the connections are set.

4.2.4 A Description Logic perspective and modelled themes

We have seen in 4.2.1 that a qualitative theme can be related to other themes. These relationships can be subclass/superclass (subsumption), equivalence and disjointness. Restrictions of values for properties can also be defined and can be used in order to build partitions.

However defining a theme in a more precise way is needed in some cases. The solution is to provide a Description Logic (DL) definition. A DL definition consists of an axiom of equality where the left hand side is an atomic concept (the concept being defined). Thus, a definition provides a necessary and sufficient condition for the concept. Instead, when we set a class as a subclass of another, we are adding a logical axiom that is only necessary. These axioms of inclusion (not equality) are usually called specializations or usually terminological axioms in the DL literature. Obviously when we set a theme as equivalent to another it is in fact a definition, since it imposes a necessary and sufficient condition.

A terminology (TBox) in DL is a set of definitions, that usually does not include specialization axioms since they have less definitorial impact. However, our knowledge base contains not only definitions but also inclusion axioms, since it focuses on representing the specialization of the geographical themes.

In our particular case we are using a $\mathcal{SHIQ}$ DL language, extended with data types. We recall here that $\mathcal{SHIQ}$ is a language that has the following constructors:

- $\mathcal{S}$: represents the $\mathcal{ALC}_{\mathcal{R}}{}^{+}$ family, where $\mathcal{ALC}$ comes from:
    - $\mathcal{AL}$ (Attribute Language): atomic concepts, universal concept, bottom concept, atomic negation, intersection of concepts, value restriction and limited existential quantification
    - $\mathcal{C}$: Complement (negation) of concepts. This implicitly includes union of concepts and full existential quantification.
    - $\mathcal{R}^{+}$ means transitive roles (functional roles are in fact also included)
- $\mathcal{H}$: means hierarchies of roles
- $\mathcal{I}$: means inverse of roles
- $\mathcal{Q}$ means qualified number restrictions

The main DL reasoners, FaCT and Racer, can deal with our ontology since they support $\mathcal{SHIQ}$ language. OWL is also a $\mathcal{SHIQ}$ language and can consequently be used to represent our ontology.

We do not use other more expressive constructors of other DL languages such as enumerations, since they are not relevant in our case. On the other hand, we use domains of roles, that, although are not a feature of DL, are present in frame languages and in fact in the DL profile of OWL, and are supported by FaCT and Racer.

A qualitative theme may be modelled by means of a DL definition axiom. Note that there is no sense on adding a DL definition to a quantitative theme, since its meaning is already unambiguous. From now on we call *modelled theme* to the qualitative theme that has been given a DL definition. Modelled themes are mainly used in our context in order to represent those themes that are affected by a combination of different thematic variables (either other qualitative themes or quantitative classes). It is important to note here that different experts or dataset producers may have different definitions for the same theme, since again they model the reality in different ways. Therefore, a modelled theme may have different models, each one consisting of a DL definition. Each model will be given a name in order to be identified.

As a very simplified example, let us imagine that the fire brigade of Majorca defines an area at fire hazard as a pine forest where the average temperature in summer is above 25ºC and that have less than 50 ml of precipitations in summer (please note that they use more complex models for sure). Here the modelled theme is “area at fire hazard” and a model (definition) is created for it. The fire brigade of other regions or other organizations probably have other definitions for fire hazards. In order to differentiate these different definitions, they are given a name (for instance, “Majorca fire brigade definition”). We will see later how the logical definition is included in the ontology.

The ontology class *Model* is used to represent each of the different definitions for a modelled theme. Its properties are the name (*modelName*) and the reference to the qualitative theme that it is defining (*modelTheme*). As in the case of quantitative classifications, this property cannot have a class as its value since this would go out of DL. In consequence we follow the same approach and use “ideal” individuals of themes that express not the realization of the theme in a dataset but instead an idealized or archetypical representation of the theme. Again the drawback is that we will have to deal with both “real” and “ideal” individuals of themes. However it is not difficult to differentiate them, since an “ideal” individual has the role *themeModel*, which is defined as the inverse of role *modelTheme* and used to link it to its model. On the other hand, we will see that "real" individuals may be connected to dataset values through the role *qualitativeThemeConnection*.

According to these properties, a class *ModelledTheme* can be defined to represent the themes that are affected by a model in terms of "ideal" individuals, in the following way:

```
ModelledTheme ≡ QualitativeTheme ⊓ ∃themeModel.Model
```

Note that the class *ModelledTheme* only contains "ideal" individuals of the qualitative classes, but not the "real" individuals that connect a theme to a dataset.

When a model is added to a modelled theme, a new individual of the class *Model* is created. And a new class is also created for the model. This new class is a subclass of the modelled theme, since in fact it is specializing its meaning, and consequently *Model* will be a subclass of *QualitativeTheme*. An axiom is also added setting this new class to be equivalent to the logical expression that defines the theme. It has to be noted that the model should be logically equivalent to the modelled theme, since it is a definition. But since there may be more than one model for a modelled theme this could produce logical contradictions. To avoid this, models are inserted as subclasses of their modelled themes, and we will see below that when a query requiring inference is executed, only one of the models will be selected and set as equivalent to the modelled theme.

Following our example, a class called *Area_at_fire_hazard-Majorca_fire_brigade* is created as a subclass of *Area_at_fire_hazard*. According to the definition provided, we set the class *Area_at_fire_hazard-Majorca_fire_brigade* to be equivalent to the intersection of qualitative theme *Pine_forest* and quantitative classes *Temperature-clftem1-more25* and *Precipitations-clfpre1-less50*. Two "ideal" individuals will be also created, one for the modelled theme *Area_at_fire_hazard*, and another for the model *Area_at_fire_hazard-Majorca_fire_brigade*.

According to these considerations, the OWL definition of classes and properties related to models and modelled themes is the following:

```
<owl:Class rdf:ID="ModelledTheme">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#QualitativeTheme"/>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="themeModel"/>
            </owl:onProperty>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="Model"/>
            </owl:someValuesFrom>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
</owl:Class>
```

```
<owl:ObjectProperty rdf:ID="themeModel">
      <rdf:type rdf:resource=
          "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QualitativeTheme"/>
      <rdfs:range rdf:resource="#Model"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="Model">
      <rdfs:subClassOf rdf:resource="#QualitativeTheme"/>
</owl:Class>
<owl:DatatypeProperty rdf:ID="modelName">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Model"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="modelTheme">
      <rdf:type rdf:resource=
          "http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
      <rdfs:domain rdf:resource="#Model"/>
      <rdfs:range rdf:resource="#QualitativeTheme"/>
      <owl:inverseOf rdf:resource="#themeModel"/>
</owl:ObjectProperty>
```

The following fragment of OWL document shows the part of the ontology describing our example of model for fire hazard:

```
<owl:Class rdf:ID="Area_at_fire_hazard">
      <rdfs:subClassOf rdf:resource="#Risks"/>
</owl:Class>
<owl:Class rdf:ID="Area_at_fire_hazard-Majorca_fire_brigade">
      <rdfs:subClassOf rdf:resource="#Area_at_fire_hazard"/>
      <rdfs:subClassOf rdf:resource="#Model"/>
      <owl:intersectionOf rdf:parseType="Collection">
            <owl:Class rdf:about="#Pine_forest"/>
            <owl:Class rdf:about="#Temperature-clftem1-more25"/>
            <owl:Class rdf:about="#Precipitations-clfpre1-less50"/>
      </owl:intersectionOf>
</owl:Class>

<Area_at_fire_hazard rdf:ID=
            "modelledTheme-Area_at_fire_hazard-Majorca_fire_brigade"/>
      <themeModel rdf:resource=
            "#model-Area_at_fire_hazard-Majorca_fire_brigade"/>
</Area_at_fire_hazard>
<Area_at_fire_hazard-Majorca_fire_brigade rdf:ID=
            "model-Area_at_fire_hazard-Majorca_fire_brigade"/>
      <modelName>Majorca fire brigade definition</modelName>
      <modelTheme rdf:resource=
            "#modelledTheme-Area_at_fire_hazard-Majorca_fire_brigade"/>
</Area_at_fire_hazard-Majorca_fire_brigade>
```

Let us suppose now that we have three datasets of respectively vegetation, temperature and precipitations, and they are conveniently connected to the themes and quantitative classes. These datasets can be integrated through the third type of semantic query (see Chapter 10 for further details) to obtain the areas at fire hazard according to the definition.

As we have seen, concepts that participate on the right hand side of the definition can be either quantitative or qualitative themes. Since a modelled theme is in fact a qualitative theme, it may also include references to other modelled themes and models. This provides great flexibility. Furthermore, the reasoner can infer subsumption relations between the definitions of different models, even from different modelled themes.

As we have already mentioned, a modelled theme is always a qualitative theme. We can observe that it is often not clear when a qualitative theme should be defined by means of a model. There is no strict rule for this matter, which is left to the expert's decision. In some cases, the name of a qualitative theme may be clear enough and there is no need to define a model for it. A textual definition can be provided to help humans to understand the meaning of the class. But in other cases the expert may decide that a logical definition is needed to understand the meaning of the theme, not only by humans but also by intelligent agents or external applications.

Let us consider an example where the expert may decide that a class *Forest* has no need for a logical definition since there is no confusion on what a forest is. But in some contexts, it may be important to make explicit the percentage of surface of the spatial unit covered by trees in order to decide if it can be considered as a forest. Let us suppose that the threshold is a 70%. In this case, a quantitative theme *Trees surface percentage* should be added to the ontology with a classification with two quantitative classes, one over 70% (let us call this class *Trees_over70*) and another below this threshold. This way, the qualitative theme *Forest* can be given a model *Forest_model70* stating that it is equivalent to the quantitative class *Trees_over70*, and consequently *Forest* becomes a modelled theme. This example shows that the decision of giving a logical definition to a qualitative theme depends on the level of detail that the expert wants to provide.

Another related issue concerns to the connection to dataset values. There can be a dataset in the repository where the value *forests* is defined as in the model *Trees_over70*. In this case, dataset value *forests* would be directly connected to the model. But other dataset may be more generic and no definition, neither textual, has been provided for the value "forests". In this case, the value would be connected to the modelled theme.

Analogously, both modelled themes and models may have subclasses (we should recall that a model is subclass of its modelled theme). For instance, *Forest* may have two different models, but a class *Pine forest* should be a subclass of any model of forest and consequently is defined as a subclass of the modelled theme. But in other cases, there may be classes that are only subclasses of a particular model.

We can observe that this flexible way of defining themes also enables us to represent them through classifiers, as the FAO's Land Cover Classification System (LCCS) (Di Gregorio and Jansen 1998; Di Gregorio 2005) does. Let us recall from Chapter 3, that LCCS is a system that defines land cover classes in terms of independent classifiers. Three main binary classifiers are defined: *Presence of Vegetation*, with two possible values *primarily vegetated* and *non-primarily vegetated*; *Edaphic Condition*, with two possible values *terrestrial* and *aquatic or regularly flooded*; and *Artificiality of Cover*, with two possible values *artificial/managed* and *(semi)natural*. The combination of these three classifiers gives rise to eight different classes. For instance, the class *Bare Areas* is defined as a primarily non-vegetated, terrestrial and (semi)natural. Other specific classifiers can be defined for one or more of these classes. To represent the LCCS class *Bare Areas* in our approach, we have to define three qualitative themes, one for each classifier. Each of them has two disjoint subclasses, for instance the theme *Presence of Vegetation* has two disjoint subclasses *primarily vegetated* and *non-primarily vegetated*. Finally, the class *Bare Areas* is defined as the intersection of the qualitative themes *primarily vegetated*, *terrestrial* and *(semi)natural*. If we have three different datasets, one for each one of the three classifiers, we integrated them in order to obtain the *Bare areas*, through the third type of semantic query, as we will see in Chapter 10. These classifiers could also be quantitative. In this case the classifier would be represented as a quantitative theme, while each of its possible values as a quantitative class in a quantitative classification.

When a semantic query requiring inference is executed, if a modelled theme has more that one model, the user has to decide which model has to be considered before the query is executed. The selected model will be set as equivalent to its modelled theme (it was a subclass and now is set as a superclass too), and the rest of models are temporarily removed since they do not participate in the selected definition. This enables the reasoner to use the definition properly: a definition axiom is a necessary and sufficient condition. The reasoner can infer subsumption relations between modelled themes from their definitions. Once the query finishes, these modifications are undone, reflecting again the different model classes as subclasses of the modelled theme. In fact, to be more precise, each query creates a temporal copy of the ontology in the reasoning engine. This copy only contains the model selected by the user, but not the others. The copy is destroyed once the query finishes.

It has to be remarked that the ability of our semantic framework to define modelled themes in a flexible way, supporting reasoning about them, is the main contribution of this chapter and constitutes a significant improvement with respect to other approaches described in Chapter 3.

4.2.5 Mixtures of qualitative themes

We discuss in this section a particular case of qualitative themes. Some datasets include classes that correspond to a mixture of qualitative themes, as for instance “mixed pine and oak forest”. This class only considers those forest regions where both pines and oak coexist. Note that this class is not the intersection of qualitative classes “pine forests” and “oak forests”, since a mixed forest is neither a pine forest nor an oak forest: it is a different particular type of forest. We will model this type of qualitative classes by means of the *qualitativeThemeMixOf* property. Again, the natural way of defining this property would be having a class as it range (using metaclasses). But this is not covered by DL and not supported by Racer or FaCT, and consequently, as in the case of models and quantitative classes, we have to use “ideal” individuals to model this relation.

We can define a qualitative mixture theme, *QualitativeMixTheme*, as a qualitative theme having an “ideal” individual related to at least two qualitative themes through the property *qualitativeThemeMixOf*:

```
      QualitativeMixTheme ≡
            QualitativeTheme ⊓
            (≥2 qualitativeThemeMixOf) ⊓
            ∀qualitativeThemeMixOf.QualitativeTheme
```

The following fragment of OWL document provides a formal definition of classes and properties related to mixtures of qualitative themes:

```
<owl:ObjectProperty rdf:ID="qualitativeThemeMixOf">
      <rdfs:domain rdf:resource="#QualitativeTheme"/>
      <rdfs:range rdf:resource="#QualitativeTheme"/>
</owl:ObjectProperty>
<owl:Class rdf:ID="QualitativeMixTheme">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#QualitativeTheme"/>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="qualitativeThemeMixOf"/>
            </owl:onProperty>
            <owl:allValuesFrom>
              <owl:Class rdf:ID="QualitativeTheme"/>
            </owl:allValuesFrom>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="qualitativeThemeMixOf"/>
            </owl:onProperty>
            <owl:minCardinality rdf:datatype=
            "http://www.w3.org/2001/XMLSchema#nonNegativeInteger">2
            </owl:minCardinality>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
</owl:Class>
```

This definition permits a query on a given theme to be expanded with all the classes where the theme is mixed. For instance, following with the example of mixed forests, the following DL expression corresponds to all the classes where “forest” appears, including in mixings:

```
      Pine_Forest ⊔ ∃qualitativeThemeMixOf.Pine_Forest
```

The second part of the expression strictly refers to “ideal” individuals, and their classes should be obtained from them. Consequently, the accurate expression is:

```
      Pine_Forest ⊔
            { C | C(x), ∀x ∈ (∃qualitativeThemeMixOf.Pine_Forest) }
```

4.3 Vocabularies

Normalized vocabularies can be introduced into the ontology. This makes it possible to relate themes that are used in datasets to terms that can be understood by the community that has produced the vocabulary.

Vocabularies are represented through the ontology class *Vocabulary*. Each particular vocabulary corresponds to an individual of this class. The class has the following properties:

- *vocabularyName*: each vocabulary is given a name in order to identify it.
- *vocabularyDescription*: a textual description of the vocabulary, including its objectives and who maintains it.
- *vocabularyNamespace*: a namespace for the vocabulary.
- *vocabularyScale*: this value indicates if the vocabulary is focused on a particular scale.
- *vocabularyMinAreaSize*: this value indicates if the vocabulary is focused on a particular minimal area size.

A vocabulary consists obviously of a set of terms. Terms can be related among each other by means of the most typical thesauri connectors: broader term, narrower term and synonym term. We have not considered other thesauri connectors as related term. A vocabulary term is represented through the ontology class *VocabularyTerm*. Each particular term corresponds to an individual of this class. It has the following properties:

- *termName*: each term is given a name in order to identify it.
- *termDefinition*: a textual definition.
- *termBroader*: a reference to a (unique) broader term in this vocabulary.
- *termNarrower*: a list of references to its narrower terms in this vocabulary.
- *termSynonym*: a list of references to its synonym terms in this vocabulary.

The following fragment of OWL document provides a formal definition of classes and properties related to vocabularies and terms:

```
<owl:Class rdf:ID="Vocabulary"/>
<owl:FunctionalProperty rdf:ID="vocabularyName">
      <rdfs:domain rdf:resource="#Vocabulary"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="vocabularyDescription">
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
      <rdfs:domain rdf:resource="#Vocabulary"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
```

```
<owl:FunctionalProperty rdf:ID="vocabularyNamespace">
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
      <rdfs:domain rdf:resource="#Vocabulary"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="vocabularyScale">
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#float"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
      <rdfs:domain rdf:resource="#Vocabulary"/>
</owl:FunctionalProperty>
<owl:DatatypeProperty rdf:ID="vocabularyMinAreaSize">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Vocabulary"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>

<owl:Class rdf:ID="VocabularyTerm"/>
<owl:FunctionalProperty rdf:ID="termName">
      <rdfs:domain rdf:resource="#VocabularyTerm"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="termDefinition">
      <rdfs:domain rdf:resource="#VocabularyTerm"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="termBroader">
      <rdfs:domain rdf:resource="#VocabularyTerm"/>
      <rdfs:range rdf:resource="#VocabularyTerm"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#ObjectProperty"/>
      <owl:inverseOf rdf:resource="#termNarrower"/>
</owl:FunctionalProperty>
<owl:ObjectProperty rdf:ID="termNarrower">
      <rdfs:range rdf:resource="#VocabularyTerm"/>
      <rdfs:domain rdf:resource="#VocabularyTerm"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="termSynonym">
      <rdfs:range rdf:resource="#VocabularyTerm"/>
      <rdfs:domain rdf:resource="#VocabularyTerm"/>
</owl:ObjectProperty>
```

When a vocabulary is introduced into the ontology, individuals of *Vocabulary* and *VocabularyTerm* are created. But no new thematic classes are added. This is done when the vocabulary is integrated into the ontology. In this case, a new class is created for each term, and its broader term is added as superclass, its narrower terms are added as subclasses, while its synonyms are added as equivalent classes. The user that runs this process can set a new class as a top of the vocabulary, being a superclass of all the new classes. This integration operation is usually carried out when there are no classes related to the main theme in the ontology. Otherwise, the process should check if it produces inconsistencies or duplications, and therefore follow the same process described for dataset schemas in Chapter 5, integrating either term by term in a manual way or all the terms at the same time through the semi-automatic merging based on syntactical and structural similarities.

To reflect the connection between a vocabulary term and a thematic class, two new roles *themeTermConnection* and *termThemeConnection*, which are analogous to *themeConnection* and *datasetValueConnection* for connections between dataset values and thematic classes. Note that, although vocabulary terms usually refer to qualitative themes, they could evantually be connected to quantitative themes. Note also that *themeTermConnection* is a functional property, allowing a term to be connected to one thematic class at most. These properties are defined in OWL in the following way:

```
<owl:ObjectProperty rdf:ID="themeTermConnection">
      <rdf:type rdf:resource=
          "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Theme"/>
      <rdfs:range rdf:resource="#VocabularyTerm"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="termThemeConnection">
      <rdf:type rdf:resource=
          "http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
      <owl:inverseOf rdf:resource="#themeTermConnection"/>
      <rdfs:domain rdf:resource="#VocabularyTerm"/>
      <rdfs:range rdf:resource="#Theme"/>
</owl:ObjectProperty>
```

5 Merging methods

We have observed in previous chapters that different qualitative datasets present different application ontologies. The objective of this chapter is to describe how these ontologies are assembled in order to obtain a higher level ontology, the taxonomy of qualitative themes. Since we focus on merging qualitative themes, the taxonomy of qualitative themes will often be referred to as the *repository ontology* hereafter.

There are two main approaches, *merging* and *alignment*, to bring together different ontologies. While the process of merging two ontologies creates a new single coherent ontology, the alignment process preserves the two original ontologies and establishes links (usually called *mappings*) between them (Noy and Musen 1999). In this line, (KnowledgeWeb Consortium 2005), which aims at providing a formal framework for defining alignments, defines mapping, alignment and merging in the following way:

- Mapping: a formal expression that states the semantic relation between two or more entities belonging to different ontologies.
- Ontology Alignment: a set of correspondences between two or more (in case of multi-alignment) ontologies. These correspondences are expressed as mappings.
- Ontology Merging: the creation of a new ontology from two or more (possibly overlapping) source ontologies. This concept is closely related to that of integration in the database community.

According to these definitions, we can affirm that our repository ontology is obtained by merging, since it is created from the application ontologies. However, it is important to note here that our merging is done in a particular way, because the objective is not exactly to obtain a new ontology from two existing ones. Instead, what we have is a repository that grows as new datasets are added. This way, dataset ontologies are merged as they are inserted in the repository. Our merging process will always involve on the one hand the repository ontology, and on the other hand the new dataset being inserted.

Furthermore, as we have described in the representation model for datasets and values (Section 4.1), the application ontology of a qualitative dataset is obtained from the metadata file and represented in our model by means of an individual of the class *QualitativeDataset* and several individuals of the class *QualitativeDatasetValue*. Consequently, although our merging conceptually refers to a dataset ontology, it physically involves a set of individuals of *QualitativeDatasetValue*. Thus, the result of the process is not really a new ontology but the modification of the repository ontology, with new classes and mappings (connections) between these classes and dataset values (through properties *datasetValueConnection* and *qualitativeThemeConnection*). The solutions for merging that are propounded in this chapter will be described from a conceptual point of view. This way, we will describe them in terms of two ontologies *A* and *B*, where *B* is merged into *A*: *A* is modified, while *B* remains unchanged. The implementation in the OntoGIS tool needs a slight adaptation in order to deal with our taxonomy of qualitative themes (repository ontology) and with individuals of *QualitativeDataset* and *QualitativeDatasetValue* (dataset ontology). Nevertheless, since these adaptations are straightforward, we will not provide more details on that issue.

It is also worth clarifying that the majority of approaches on merging geographic datasets focus on merging just two of them. We can observe that our approach supports any number of datasets. In a particular case where just two datasets have to be integrated, the user simply should create a new repository including the required two datasets. The user can manage as many repositories as desired.

Apart from this difference between the implementation and conceptual points of view, the ontologies in our context present a significant specific trait: the thematic classes are organized through subclass/superclass relation (subsumption in terms of DL), forming a hierarchy (taxonomy) of classes. Other eventual relations between thematic classes are not considered. This fact has to be carefully considered for devising a valid solution to the merging problem. This way, classical relational schema integration techniques, that do not cover subsumption, cannot be applied in our context. The solution that we will propound can also be used to merge taxonomic ontologies in a different context than the geographic one. A clear example of the application in another context is the merging of ontologies used to classify resources in web directories such as Yahoo![1] or Google[2].

In this context, we propound two merging methods: a *manual method* where a domain expert determines the mappings among dataset values and qualitative thematic classes; and a *semi-automatic method*, where a list of suggested mappings is generated in an automatic way, and is presented to the expert, who may accept or modify the mappings in the list. It has to be noted that, although there may be a clear distinction between “method” and “methodology” in Software Engineering, these terms are used indiscriminately in the field of Ontological Engineering (de Hoog 1998). In fact, some authors use the expression “methodologies and methods” to avoid confusions, as in (Gómez-Pérez et al. 2004). In our work, we will use the term “method”, since “methodology” usually refers to the whole life cycle of a system, involving different phases. Ontology merging can be considered as one of these phases, and consequently a merging method describes how it is performed in a series of steps organized according to a certain workflow.

The key point in the semi-automatic method is the algorithm that generates the list of suggested mappings. To avoid the confusion with the term “merging algorithm”, that often is used in the literature to refer to merging methods or methodologies, we have denominated *mapping algorithm* to the algorithm for the generation of suggested mappings. A relevant aspect that has to be addrerssed by the mapping algorithm is the fact that dataset application ontologies usually have a very simple structure. They usually contain a small number of classes, organized in an almost flat hierarchy. They do not contain properties either. As we will see in Section 5.1, some existing algorithms can only be effectively applied to ontologies being bigger and denser than our dataset

[1] http://dir.yahoo.com
[2] http://www.google.com/dirhp

ontologies, where some information can be entailed from the depth of a class in the taxonomy or from its neighbours. Other algorithms that provide mappings between classes with common properties cannot be applied in our case either. In this context, we have developed three different mapping algorithms, each focusing on a particular aspect of the ontologies, that will be discussed in the following chapters. A mapping algorithm based on similarities among names of classes and dataset values is described in Chapter 6; a mapping algorithm that also uses a terminological base (for instance, a thesaurus) in order to consider synonymy, hypernymy and hyponymy, is presented in Chapter 7; finally, a mapping algorithm based on how values are spatially distributed in the datasets is explained in Chapter 8. In our implementation in OntoGIS the user can select the algorithm to use. Usually the terminological algorithm provides better results than the string-based. On the other hand, the algorithm based on spatial distribution provides good results when datasets contain a big number of spatial units for each thematic value. An evaluation of these mapping algorithms is carried out in Chapter 9.

This chapter is organized as follows. In the next section we discuss related work on merging and alignment. Section 5.2 describes our two merging methods, manual and semi-automatic for qualitative datasets. Finally, the case of merging quantitative datasets is discussed separately in Section 5.3.

5.1 Related work

Aligning classes or entities from different ontologies is a common problem in the disciplines of ontology merging and database schema integration. Figure 15, elaborated by (KnowledgeWeb Consortium 2004), shows a classification of the different alignment methods. Other similar classification with less detail can also be found in (Rahm and Bernstein 2001).

Focusing on the lower part of the diagram, we can observe that our work is included in the categories of terminological, structural and extensional (instance-based) methods. Terminological methods are divided between string-based and language-based. Particularly, our algorithm based on string similarities (Chapter 6) falls into the category of string-based methods, while our algorithm using a terminological base (Chapter 7) belongs to the category of language-based methods. Our algorithm based on the distribution of the values of spatial units (Chapter 8) is a particular type of extensional method. They three also consider structural issues. Internal structural aspects of a class refer to its properties with their data types, while its external aspects refer to its relation with other classes in the graph or taxonomy. Since our dataset application ontologies are usually simple, with no properties and types, structural similarity measures cannot be applied. Instead, structure is used to conduct the merging process.

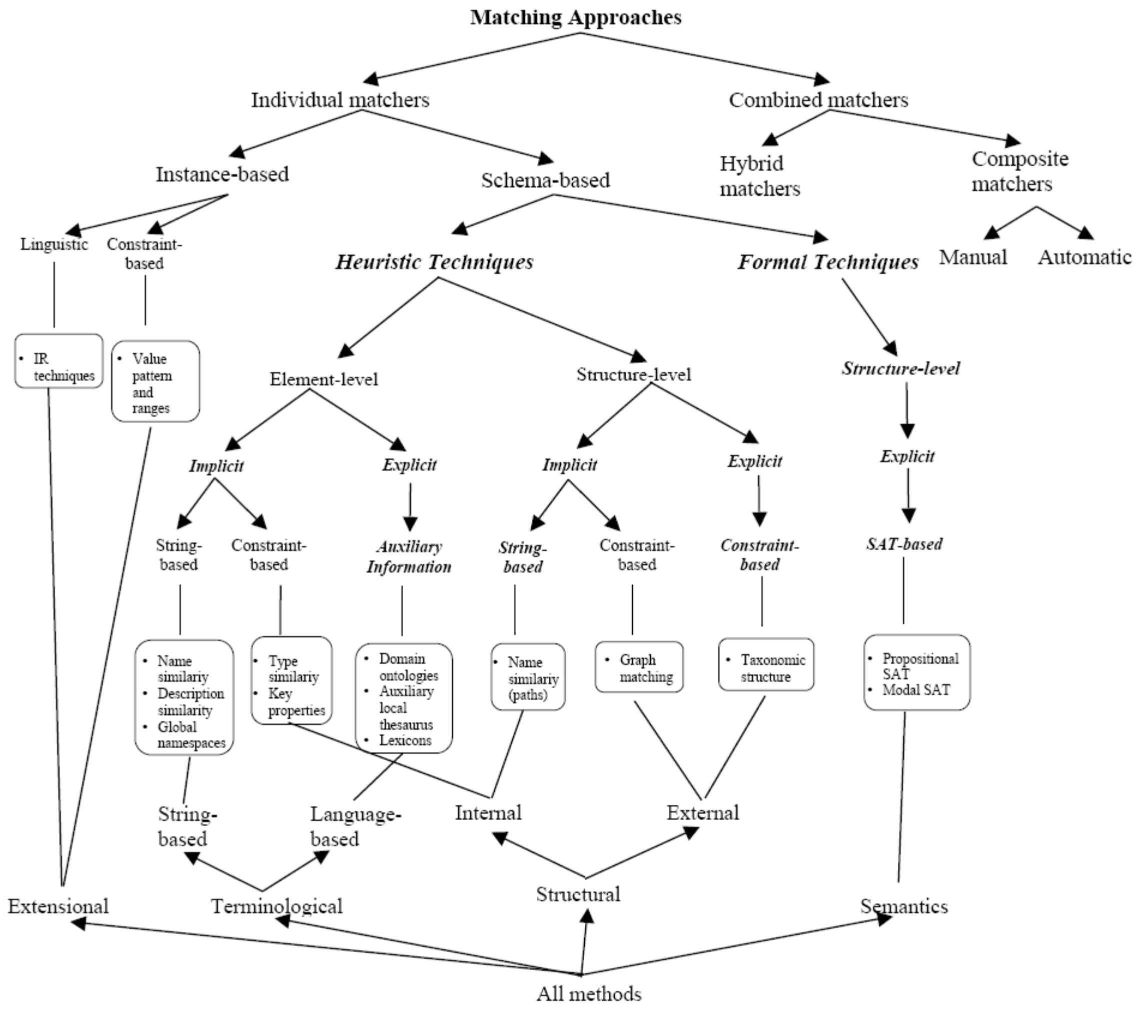


Figure 15. Classification of alignment methods according to (KnowledgeWeb Consortium 2005). Image captured from http://knowledgeweb.semanticweb.org/

In the following subsections we describe other approaches related to ours. Subsection 5.1.1 defines similarity and distance functions, as well as their properties. Subsection 5.1.2 presents several string-based similarity measures, related to our algorithm based on name similarities. Subsection 5.1.3 presents other measures for comparing entities in taxonomic and graph structures, and is related to our algorithm that uses a terminological base. In Subsection 5.1.4 we describe extensional methods in the context of geospatial information. Subsection 5.1.5 describes different ways of composing a global similarity or distance from partial measures. Subsection 5.1.6 presents some cognitive bases on asymmetric similarities and presents some approaches that have used them. Finally, Subsection 5.1.7 briefly describes the most relevant merging methods and systems in the literature.

5.1.1 Definition of similarity and distance functions

A *similarity measure* can be formally defined as a function

$$s: O \text{ x } O \rightarrow [0,1]$$

from a pair of objects to a real number (normalized in the interval [0,1]) that indicates how similar the objects are, satisfying the following two properties:

- Maximality: $\forall x \in O$, $s(x,x) = 1$
- Symmetry: $\forall x,y \in O$, $s(x,y) = s(x,y)$

On the other hand, a *dissimilarity measure*

$$d: O \times O \rightarrow [0,1]$$

is the opposite of the similarity and indicates how different two objects are:

$$d(x,y) = 1 - s\ (x,y)$$

A dissimilarity function satisfies symmetry and minimality ($\forall x \in O$, $d(x,x) = 0$). Furthermore, a distance is a dissimilarity function that satisfies the triangle inequality:

$$\forall x, y, z \in O, d(x,y) + d(y,z) \geq d(x,z)$$

Note that geometric distances also satisfy the property of definiteness (that in fact includes minimality):

$$\forall x, y \in O, d(x,y) = 0 \text{ iff } x = y$$

However, the fact of dealing with equivalent classes or synonyms makes us not to consider this property in the definition of a distance.

Furthermore, significant studies on Cognitive Science, originated by (Tversky 1977), affirm that the way people perceive similarities between concepts does not satisfy these three principles. Regarding minimality/maximality, two identical stimuli can be judged as different in different contexts. Regarding symmetry, judgements of the form “a is like b” are directional. For instance we say “an ellipse is like a circle” but not “a circle is like an ellipse”. As Tversky points out, “the direction of asymmetry is determined by the relative salience of the stimuli; the variant is more similar to the prototype than vice versa”. Regarding the triangle inequality, similarities are usually referred to a common feature between concepts, but the principle will not be satisfied if *A* and *B* are similar according to a feature *f1*, but *B* and *C* are similar according to a different feature *f2*. For instance, an *athletic field* is similar to a *stadium* (because both are sports facilities) and a *stadium* is similar to a *theatre* (because both are constructions where people go to attend events), but an *athletic field* is not similar to a *theatre* (Rodríguez 2000; Rodríguez and Egenhofer 2003).

These studies have influenced different ontology merging and alignment approaches based on asymmetric similarity measures in the last 10 years. More details on asymmetric measures can be found in 5.1.6. In fact, we will see that we will define asymmetric similarity measures for our algorithms.

5.1.2 String-based similarity measures

Several similarity and dissimilarity functions exist to compare two strings.

Hamming distance is a classical normalized dissimilarity function used in codification, defined in the following way, where *S* and *T* are two strings and |*S*| denotes the length of *S*:

$$dist(S,T)=\frac{\left(\sum_{i=1}^{\min(|S|,|T|)} S[i]\neq T[i]\right)+\left||S|-|T|\right|}{\max(|S|,|T|)}$$

Substring distance is a normalized distance function obtained from the longest common substring *X* between two strings *S* and *T*:

$$dist(S,T)=\frac{2\cdot|X|}{|S|+|T|}$$

Q-gram distance (Sutinen and Tarhio 1995) is also a distance function which counts the number of groups of *Q* contiguous characters (qgrams) that are common in both strings. The tri-gram distance is the most usual among them.

$$dist(S,T)=\frac{|qgrams(S)\cap qgrams(T)|}{q\cdot\max(|S|,|T|)}$$

Levenstein edit distance (Levenstein 1966) counts the number of operations needed to transform one string into another. The operations usually considered are insertion, deletion and substitution of a character. Some variants of this edit distance exist, often coming from the field of molecular and DNA sequence matching. The Needleman-Wunsch distance (Needleman and Wunsch 1970) assigns a different cost for each type of edit operations. The Smith-Waterman distance (Smith and Waterman 1981) additionally uses an alphabet mapping to costs. Gotoh (Gotoh 1981) and Monge-Elkan (Monge and Elkan 1996) variants use variable costs for gaps (inserts or deletion). In the last years, some variants have appeared that use a large number of arbitrarily defined parameters whose weights are learnt from training data. Some of these use probabilistic models based on Hidden Markov Models (Ristad and Yianilos 1997; Durbin et al. 1998; Bilenko and Mooney 2003), while (Bilenko and Mooney 2005) defines Alignment Conditional Random Fields (ACRFs), that are based on a probabilistic model in an undirected graph.

Another widespread measure, not based on edit distance, is the one defined by Jaro (Jaro 1989; 1995), usually used together with the Wrinkler variant (Winkler 1999), commonly referred to as the Jaro-Wrinkler distance function. It counts the common characters between two strings even if they are misplaced by a “short” distance. This way, given two strings *S* and *T*, *common*(*S*,*T*) is the number of characters in *S* that are “common with” *T*, where a character *s* in *S* is in common with *T* if the same character appears in a closer position in *T*, at a distance of less than half the length of the shorter string. On the other hand, *transp*(*S*,*T*) measures the number of transpositions of characters in *common*(*S*,*T*) relative to *common*(*T*,*S*). The normalized Jaro distance is then defined as follows:

$$dist(S,T) = \frac{1}{3} \cdot \left(\frac{common(S,T)}{|S|} + \frac{common(S,T)}{|T|} + \frac{transp(S,T)}{2 \cdot common(S,T)} \right)$$

The Winkler variant modifies this by slightly improving the weight of poorly matching pairs *S,T* that share a long common prefix. *P* is the length of the longest common prefix of *S* and *T* and *F* is a constant scaling factor for how much the score is adjusted upwards for having common prefixes. The Jaro-Winkler distance is defined as follows (where *Jaro* refers to the Jaro distance):

$$dist(S,T) = Jaro(S,T) + \frac{P}{F} \cdot (1 - Jaro(S,T))$$

(Stoilos et al. 2005) defines another distance which considers both the common and different parts between the two strings. It is defined in the interval [-1, 1] and combines some of the previous measures:

$$dist(S,T) = comm(S,T) – diff(S,T) + winkler(S,T)$$

where *comm*(*S*,*T*) refers to the common part and corresponds to the substring distance; *winkler*(*S*,*T*) corresponds to the Winkler variation of the edit distance; and *diff*(*S*,*T*) refers to the different part and considers the length of the unmatched substrings, and is obtained through the following expression:

$$diff(S,T) = \frac{uLen_S \cdot uLen_T}{p + (1-p) \cdot (uLen_S + uLen_T - uLen_S \cdot uLen_T)}$$

where $p \in [0, \infty)$ and $uLen_S$ represents the length of the unmatched substring from the initial string *S* divided by the length of *S*.

Token-based distances are based on a segmentation of the strings into tokens, usually terms (or words) separated by blank spaces. Several measures exist to deal with tokens (a further description can be found in (Salton and McGill 1983)). Jaccard measure is probably the most used. It is obtained from the number of common tokens in both strings. If strings *S* and *T* are respectively composed of tokens $\{s_1,...,s_n\}$and $\{t_1,...,t_m\}$, $S \cap T$ is the set of common tokens, while $S \cup T$ is the set of all tokens in both *S* and *T*. The Jaccard similarity is obtained through the expression:

$$sim(S,T) = \frac{|S \cap T|}{|S \cup T|}$$

Dice’s coefficient and cosine similarity are other two token-based measures very similar to Jaccard measure, with the same numerator but different denominators: in Dice’s coefficient it is the sum of the number of tokens in *S* and *T*, while in the cosine similarity it is the square root of the sum of the square number of tokens (each string is identified as a vector and the similarity is the cosine of the angle of two vectors). Also related is TF/IDF (Term Frequency/Inverse Document Frequency), where each token is given a weight depending on the frequency of the term in a corpus of documents. However, this measure is particularly oriented to Information Retrieval in the context of obtaining the relevance of a term in a document, and are not really useful for the alignment of ontologies in our context where there are no references to any text document corpus. A "soft" version of TF/IDF, SoftTFIDF, is defined in (Cohen et al. 2003b) with a similar focus. It combines string-based similarities with TF/IDF in order to consider tokens lexically similar (apart from those equal) in the set of common tokens. Jensen-Shanon (Dagan et al. 1999) is another token-based measure, based on probability distributions, that does not fit our situation.

Experiments of (Bilenko et al. 2003; Cohen et al. 2003a; b) aim at comparing different string and token distances in the context of identifying distinct records that refer to the same entity from a census dataset. They also provide a Java toolkit to support this task. The higher precision was obtained using the Levenstein edit-distance metric, modified by the Winkler method.

We will see that although our work is based on tokens, our approach is different from the typical problems of information retrieval or record linkage. We have used a variation of the substring method for terms (tokens) that discards substrings shorter than three characters. However, any other string-based distance could have been used instead. We will then define an asymmetric similarity that is obtained from the partial similarity of their terms.

5.1.3 Similarity measures in taxonomies and graphs

Terminological methods use external resources as lexical bases or thesaurus. WordNet (Miller 1990; Fellbaum 1998) (more details can be found in Chapter 7) has been the most widely used. In WordNet terms and their meanings are organized through sets of synonyms (synsets). For instance, the noun forest has two synsets, one for each of its two senses or meanings:

- forest, wood, woods -- (the trees and other plants in a large densely wooded area)
- forest, woodland, timberland, timber -- (land that is covered with trees and shrubs)

Hypernyms (more generic terms) and hyponyms (more specific terms) are provided for each synset, generating an is-a taxonomy. Terms are also provided with holonyms and meronyms, i.e. a whole-part relation. For instance, Figure 16 shows the hierarchical hypernyms of the first sense of forest.

```
forest, wood, woods -- (the trees and other plants in a large densely wooded area)
   => vegetation, flora, botany -- (all the plant life in a particular region or period; "Pleistocene
                    vegetation"; "the flora of southern California"; "the botany of China")
     => collection, aggregation, accumulation, assemblage -- (several things grouped together or
                    considered as a whole)
       => group, grouping -- (any number of entities (members) considered as a unit)
         => abstraction -- (a general concept formed by extracting common features from specific
                    examples)
           => abstract entity -- (an entity that exists only abstractly)
             => entity -- (that which is perceived or known or inferred to have its own distinct
                    existence (living or nonliving))
```

Figure 16. Example of hypernyms of one sense of forest in WordNet

Given two classes, terminological methods obtain in a first step their corresponding terms in WordNet. Distance or similarity between both classes will be consequently obtained from a semantic distance or similarity defined over the WordNet terms.

Semantic similarity in fact represents a special case of semantic relatedness. Therefore, the most usual way to evaluate this semantic similarity in a taxonomy is to measure the shortest path between the nodes (Rada et al. 1989). The shorter the path, the more similar they are. This is usually referred to as path distance or semantic distance.

However, a common problem of big terminological bases, usually elaborated from different sources, is that they often provide a heterogeneous structure of the hypernymy/hyponymy relation. Adjacent nodes are not necessarily equidistant and some parts of the hierarchy may be denser than others and the semantic distance is biased. For instance, we can see in Figure 17 that “pine” has the hypernym “conifer”, which has the hypernym “gymnospermous tree”, which has the hypernym “tree”. However, we can see as another trees present lower detail in their hypernym structure. For instance, “oak” has “tree” as its direct hypernym, and does not provide the information that an “oak” is an “angiospermous tree”, although this information exists for the “maple”.

In this context of heterogeneous structure, path distance does not provide a reliable indicator for how similar two terms are. As an example, we can observe from Figure 18 that the distance between “bottle” and “car” is 4, while the distance between two trees as “pine” and “eucalyptus” is higher, 5, although they are clearly perceived as more similar than a bottle and a car. Moreover, this example also shows us that intuitively unrelated terms may share a common hypernym, even at a short distance (2 in the example).

pine, pine tree, true pine -- (a coniferous tree)
=> conifer, coniferous tree -- (any gymnospermous tree or shrub bearing cones)
=> gymnospermous tree -- (any tree of the division Gymnospermophyta)
=> tree -- (a tall perennial woody plant having a main trunk and branches forming a distinct elevated crown; includes both gymnosperms and angiosperms)

oak, oak tree -- (a deciduous tree of the genus Quercus; has acorns and lobed leaves; "great oaks grow from little acorns")
=> tree -- (a tall perennial woody plant having a main trunk and branches forming a distinct elevated crown; includes both gymnosperms and angiosperms)

maple -- (any of numerous trees or shrubs of the genus Acer bearing winged seeds in pairs; north temperate zone)
=> angiospermous tree, flowering tree -- (any tree having seeds and ovules contained in the ovary)
=> tree -- (a tall perennial woody plant having a main trunk and branches forming a distinct elevated crown; includes both gymnosperms and angiosperms)

eucalyptus, eucalypt, eucalyptus tree -- (a tree of the genus Eucalyptus)
=> gum tree, gum -- (any of various trees of the genera Eucalyptus or Liquidambar or Nyssa that are sources of gum)
=> tree -- (a tall perennial woody plant having a main trunk and branches forming a distinct elevated crown; includes both gymnosperms and angiosperms)

Figure 17. Several examples of types of trees in WordNet

bottle -- (a glass or plastic vessel used for storing drinks or other liquids; typically cylindrical without handles and with a narrow neck that can be plugged or capped)
=> vessel -- (an object used as a container (especially for liquids))
=> **container** -- (any object that can be used to hold things (especially a large metal boxlike object of standardized dimensions that can be loaded from one form of transport to another))

car, railcar, railway car, railroad car -- (a wheeled vehicle adapted to the rails of railroad; "three cars had jumped the rails")
=> wheeled vehicle -- (a vehicle that moves on wheels and usually has a container for transporting things or people; "the oldest known wheeled vehicles were found in Sumer and Syria and date from around 3500 BC")
=> **container** -- (any object that can be used to hold things (especially a large metal boxlike object of standardized dimensions that can be loaded from one form of transport to another))

Figure 18. Examples of hypernyms for one of the senses of bottle and one of the senses of car in WordNet

Some modifications have been proposed to try to avoid these problems. (Lee et al. 1993), and others later, propose path distances with variable edge weights that are influenced by the local edge density and the node depth. Also related is (Wu and Palmer 1994), which considers path distances and depth, and defines similarity between classes *A* and *B* as a function of their respective distance to their common superclass *C* (N_A and N_B) and the depth of class *C* (distance between *C* and the top node, N_C):

$$sim(A,B) = \frac{2 \cdot N_C}{N_A + N_B + 2 \cdot N_C}$$

Other authors as (Sussna 1993; Richardson et al. 1994) also consider part-of relations in paths. Moreover, Sussna defines another weighted path distance that also considers the type of relation. Other authors have also used path distance in relational schemas, as in the case of the DIKE schema matcher (Palopoli et al. 1998).

Path distance has been widely used in implemented systems, usually combined with other measures. Some examples are (Valtchev and Euzenat 1997), (Guarino et al. 1999), (Su and Gulla 2003; Su 2004) or (Silva and Rocha 2003; Silva et al. 2004). However, the abovementioned modifications to the original path distance cannot be applied to our context, where the application ontologies have a simple, almost flat, structure.

A related measure for semantic comparison in taxonomies can be found in (Hotho et al. 2003). Their conceptual comparison is based on semantic cotopy, which is defined as the set of superclasses and subclasses of a class. Given two classes A and B, and their respective semantic cotopies SC_A and SC_B, they define the taxonomic overlapping as the quotient between the intersection of SC_A and SC_B, and their union. Their similarity measure is based on the upwards cotopy, which is the set of superclasses of a class (not subclasses here). Their similarity is then defined as the quotient between the intersection of upwards cotopies and their union.

A related but different approach is based on the notion of Information Content (Resnik 1995; 1999). This is based on the idea that the more information two concepts share, the more similar they are. The similarity between two classes depends on the information that contains the most specific class that subsumes them both. In this method, each class in the taxonomy is assigned a probability of finding an instance of that class. Let $p{:}O \rightarrow [0,1]$ be this function of probability, where O is the taxonomy of classes. p is monotonically nondecreasing: if C_1 is subclass of C_2, $p(C_1) \leq p(C_2)$. Moreover, the probability of a unique top class is 1. The Information Content (IC) of a class is defined in the following way:

$$IC(C) = -\ log(p(C))$$

This expression fulfils the intuition that the more abstract a concept (and higher its probability), the lower its information content. Note also that the information concept of a unique top class is 0.

Similarity between two classes, as stated above, is defined according to the information content of their most specific common superclass:

$$sim(C_1,C_2) = max_{\mathrm{C:\ C1} \sqsubseteq \mathrm{C,\ C2} \sqsubseteq \mathrm{C}}\ (\ IC(\mathrm{C})\ )$$

Other variant of the similarity based on Information Content is provided by (Lin 1998). But in any case, the problem of this approach is how the probabilities of concepts are obtained. Resnik used noun frequencies from the Brown Corpus of American English (Francis and Kucera 1982), “a large (1,000,000 word) collection of texts across genres ranging from news articles to science fiction”. Each time that a noun appears in the corpus is counted as an occurrence.

However this frequency has little relevance in our case where we do not deal with plain text but with dataset metadata. And it does not seem relevant either a probability obtained from the number of times a word appears in the metadata files of a repository.

In the geospatial context, the work of Rodríguez and Egenhofer (Rodríguez 2000; Rodríguez and Egenhofer 2003; 2004) has to be remarked. It is based on their Matching-Distance Similarity Measure. Since this is an asymmetric similarity measure, it will be discussed in depth in 5.1.6.

Finally, also focused on geospatial information but with a very different purpose is the work from Universidad de Zaragoza on semantic disambiguation in thesaurus in the context of Spatial Data Infrastructures (SDIs) (Nogueras-Iso et al. 2004; Lacasta et al. 2005; Nogueras-Iso et al. 2005). As it was discussed in Section 3.1, a catalogue server provides search services over datasets metadata. This metadata may contain keywords describing the content, and may also contain the reference to a thesaurus where each keyword is defined. However, when a user specifies a semantic query expressed through a set of terms, s/he is not aware of the several thesauri referred in the catalogue. An internal process of indexing is needed to relate the query to the keywords and thesauri of the catalogue. This process uses WordNet as a general reference: both terms in thesauri and queries are represented as collections of WordNet synsets.

The process of indexing a keyword (that refers to a thesaurus) in a metadata record, firstly gets all its broader terms in the referred thesaurus. Basically, this returns a branch in the tree of thesaurus terms. The method then retrieves all possible synsets related to the terms in the branch. Note that for a thesaurus term, its related synsets refer to different senses of the term. Finally, a voting algorithm decides the best synset (i.e. the right sense) among them. In this algorithm each of the synsets related to a thesaurus term votes for the synsets related to the rest of terms in the branch. A synset votes for another mainly according to their hypernyms. Each synset has a score at the final of the voting process, and the one with the highest is elected as the disambiguated synset. On the other hand, the query is not disambiguated and is represented through all the synstes related to the terms in the query.

Finally, a vector model is used to retrieve the metadata records from the query keywords, according to the related synsets. In the case of index entries for metadata keywords, their weights are obtained from the scores of the voting phase.

Nevertheless, it has to be noted that this approach only indexes keywords that describe the general content of the dataset, but not its internal schema. This way, the part of the metadata related to dataset values is not considered. Consequently, thematic dataset values will not be indexed. This approach is not an alternative for merging thematic dataset application ontologies.

5.1.4 Similarity measures based on geospatial instances

Extensional methods compare the set of instances (individuals) in order to obtain relations between their classes. These methods can be applied when the ontologies being merged share their instances.

The most widely used measure in this context is an adaptation of the token-based Jaccard string similarity discussed in 5.1.2. Given two classes from different ontologies, their similarity is obtained from the quotient the number of their shared instances and the union of all their instances. In terms of a probabilistic interpretation of the set of instances, *P*(*X*) refers to the probability of a random instance to be in set *X*. The Jaccard similarity between two classes *A* and *B* from different ontologies is then defined in the following way:

$$sim(A,B) = \frac{P(A \cap B)}{P(A \cup B)}$$

When the ontologies being merged do not share all their instances, machine learning methods are needed to obtain these probabilities. For instance, this is the case of GLUE (Doan 2002; Doan et al. 2004), which uses several learners to obtain a Jaccard similarity. These and other learning methods are out of the scope of our work.

In the geospatial context, datasets may share the same territory. Consequently, the objective here is to find relations between the spatial distribution of the spatial units in the datasets, either features in the case of vector datasets or cells in raster datasets. If the area covered by the spatial units of two particular themes in two datasets present a high spatial overlapping, it probably indicates a relation between these themes.

In Chapter 3 we have already analyzed the work of Uitermark (Uitermark 2001; Uitermark et al. 2002) and Duckham and Worboys (Worboys and Duckham 2002; Duckham and Worboys 2005) that provide two merging methods based on spatial instances. Uitermark's approach focuses on integrating two vector datasets with a plain structure of values. Duckham and Worboys consider a hierarchical thematic structure and define an algebraic method for both merging and integration. Our method also considers a hierarchical thematic structure and defines a formal model based on the spatial extent of dataset values, providing a higher flexibility than Duckham and Worboys. Our approach and a deep comparison with Duckam and Worboys are presented in Chapter 8.

5.1.5 Compound similarity measures

When a merging method uses different partial similarity or distance measures for particular aspects, a global measure is needed. The simplest compound distance between *A* and *B* is the average of the *n* partial distances:

$$dist(A,B) = \frac{\sum_{i=1}^{n} d_i(A,B)}{n}$$

where *n* is the number of dimensions being analyzed (string, path distance, cotopies,...), each of them having a different normalized partial distance d_i. Note that the expression for the average similarity would be analogous. Euclidean distance is sometimes used instead of the average distance.

$$dist(A,B) = \frac{\left(\sum_{i=1}^{n} d_i(A,B)^2\right)^{\frac{1}{2}}}{n}$$

Very often the partial distances are given a different weight to give more importance to some of the dimensions. A weighted average is used in this case. This is defined in the following way, where w_i is the weight for element i, and the sum of all w_i is 1.

$$dist(A,B) = \frac{\sum_{i=1}^{n} w_i \cdot d_i(A,B)}{n}$$

Again, the weighted average similarity is analogous. An example of system that uses a weighted average similarity is OLA (Euzenat and Valtchev 2003; Euzenat et al. 2004).

A different approach is the so-called similarity flooding algorithm (Melnik et al. 2001). This algorithm considers the two ontologies *O* and *O'* as directed labelled graphs and is based on the assumption that if there is a path from class *C1* to class *C2* through class *P* in ontology *O*, and another path from *C1'* to *C2'* also through class *P'* in ontology *O'*, where we already know that *P* is similar to *P'*, then *C1* is similar to *C1'* and *C2* to *C2'*. The algorithm is an iterative process that first computes an initial similarity usually based on strings, and then obtains new similarities in each step as described above, until similarity changes less than a threshold or it has run a certain number of steps.

Finally, asymmetric similarities based on Tversky also provide a global similarity which is obtained from different features. They are analyzed in the following subsection.

5.1.6 Asymmetric similarities

Tversky (Tversky 1977), as it was mentioned in 5.1.1, pointed out that people perceive similarities in a way that does not satisfy the properties of maximility, symmetry and triangle inequality. Instead, he defined the contrast model, which expresses the similarity between two objects as a linear combination (or a contrast) of the measures of their common and distinctive features. This similarity is defined through the following expression:

$$sim(A, B) = \theta \cdot f(A \cap B) - \alpha \cdot f(A \setminus B) - \beta \cdot f(B \setminus A)$$

for θ, α, $\beta \geq 0$ and $f(A \cap B)$ expresses the common features between A and B, $f(A \setminus B)$ represents the distinctive features of A not present in B, and $f(B \setminus A)$ the distinctive features of B not present in A. The three weights θ, α and β offer different relative salience for the common and distinctive features, providing an asymmetric measure.

Tversky similarity is usually expressed in the following normalized way:

$$sim(A,B) = \frac{f(A \cap B)}{f(A \cap B) + \alpha \cdot f(A \setminus B) + \beta \cdot f(B \setminus \mathrm{A})}$$

Tversky also affirmed that context influences how features may be given different weights. He presented an experiment, where the subjects had to determine the most similar country from a list to Austria. When the list was composed of Sweden, Hungary and Poland, the selected country was Sweden (note that the experiment was carried out during the cold war). But when Poland was substituted by Norway, the selected country was Hungary. This shows that context determines the most salient feature (non-communist in one case, and geographic proximity or history in the other).

Tversky’s feature-based similarity has influenced the work of other cognitive psychologists. (Krumhansl 1978) proposed a model assuming that within dense regions of a stimulus range, discriminations are finer than within relatively less dense subregions. (Medin et al. 1993; Goldstone 1994; Goldstone et al. 1997) affirm, based on several experiments, that a diagnostic process determines the weights for the different features of a particular set of objects under consideration. (Heit 1997) extends the contrast model to deal with problems of category-based induction (reasoning about instances and categories).

It can also be deduced from Tversky and Krumhansl that similarity and dissimilarity should not be considered as inverse measures. Although it is true that when one increases the other decreases, people may prioritize different features when measuring either similarity or dissimilarity between objects. However, this does not affect our work, since our measures only deal with similarities and not dissimilarities.

In the geographic field, Egenhofer and Mark (Egenhofer and Mark 1995) in their analysis of the way people think and reason about geographic space and time (naïve geography), also maintain that semantic distances between geographic entities are asymmetric. Even spatial distance is not always perceived as symmetric, since it is frequently seen as a measure for how long it takes to get from one place to another. For instance, terrain difficulties, street directions or traffic jams at rush hours may influence this perceived distance.

Especially relevant for merging geospatial ontologies is the Matching-Distance Similarity Measure of Rodríguez and Egenhofer (Rodríguez 2000; Rodríguez and Egenhofer 2003; 2004). They define a global similarity that considers three different aspects: word matching, features (parts, functions and attributes) and semantic neighbourhood. The semantic neighbourhood of radius *r* for a class *C* is defined as the set of classes that are up to a path distance of *r* from *C*, where the path considers relations of synonymy (equivalent classes), is-a and part-whole. A different similarity function is defined for each of these three aspects, while the global similarity is obtained through a weighted average of them three.

In this case, the α factor in the Tversky formula is calculated specifically for each pair of classes *A* and *B* (*A* from one ontology and *B* from the other), according to their depth in the hierarchies:

$$\alpha(A,B) = \frac{\min\left(depth(A), depth(B)\right)}{depth(A) + depth(B)}$$

They always consider β factor as 1-α, and the Tversky similarity becomes:

$$sim(A,B) = \frac{f(A \cap B)}{f(A \cap B) + \alpha(A,B) \cdot f(A \setminus B) + (1 - \alpha(A,B)) \cdot f(B \setminus \mathrm{A})}$$

This similarity is computed for each of the three different considered aspects: word matching (sim_w), features (sim_u), and semantic neighbourhood (sim_n). f function is defined in different ways for each case. This way, in the case of the word similarity sim_w, where they consider only whole words, $f(A \cap B)$ is the number of common words, and $f(A \setminus B)$ the number of words in A but not in B. In the case of the feature similarity sim_u, $f(A \cap B)$ is the number of common features of A and B, where a distinct weight is applied to parts, functions and attributes. Finally, in the case of the semantic-neighbourhood similarity sim_n for a particular radius r, $f_r(A \cap B)$ is defined as the number of common classes in the neighbourhoods of radius r for A and B. Each of these three partial similarities is given a weight (respectively w_w, w_u and w_n, such that w_w+w_u+w_n =1) to compute the global similarity.

They have also carried out some experiments of integrating datasets, testing different values for the different weights. They have integrated on the one hand a carefully selected subset of WordNet related to geospatial concepts, and on the other hand the Spatial Data Transfer Standard, SDTS (ANSI 1998). One of the most relevant results of these experiments shows that better results are obtained when features are not considered (w_u is set to 0). Regarding the other two aspects, they obtain a high recall when w_w is set to 1, although precision is below 50%. If the method combines names and semantic neighbourhood, with w_w=w_n=0.5, precision increases but recall is also below 50%. Nevertheless, these results are obtained in a context where the set of WordNet has been previously selected to have a high homogeneity with SDTS: these datasets present a considerable number of common concepts (with exactly the same name), and an analogous organization of the is-a relation, with a similar density: given a concept common to both ontologies, their respective sets of subclasses (and equivalently superclasses) usually have a high overlapping.

But this method would obtain much worse results in our context, where the dataset application ontologies that we are merging are simpler but more heterogeneous. In our context, datasets are almost flat (usually 2 levels and hardly ever more than 3) and do not provide features. Moreover, if a vocabulary like CORINE has been already merged, the density of the repository ontology is very different that the one of the dataset ontology being merged. Consequently, the definition of an α function based on depth is meaningless, as so is the semantic-neighbourhood.

In addition, we will observe that another significant advantage of our method is that it does not only focus on finding 1-to-1 equivalence mappings. In our context, it is usual that a common concept is specialized in different ways in both ontologies, and consequently other types of mappings for other relations have to be provided. For instance, our method can generate a mapping stating that a concept in one ontology is subclass of another concept in the other ontology, or that the union of two concepts in one ontology is equivalent to the union of other three concepts in the other ontology.

Finally, also in the geographic field but from a very different approach, it is worth mentioning the work of (Schwering and Raubal 2005). They measure semantic similarity from conceptual spaces. A conceptual space (Gärdenfors 2000) is a representation of a concept as a *n*-dimensional convex region in a vector space. Note that a concept can be seen as a *n*-dimensional vector, where each dimension corresponds to one feature, and the concept space is the region that is obtained by connecting the *n* components of the vector in a *n*-dimensional space. Similarity between two concepts is then calculated in terms of the distance between their concept spaces. Several distances between two concept spaces have been defined, but they propose an asymmetric measure obtained as the average of the minimum distance between each vector component in one concept space to the other concept space. However, this theoretical approach can be hardly put into practice, at least for geographic themes.

5.1.7 Existing merging systems

Different systems have been developed to help the ontology expert to carry out ontology merging. Several extensive surveys covering different aspects of merging methods and systems have also been written like (Rahm and Bernstein 2001), (Wache et al. 2001), (Do et al. 2002), (OntoWeb 2002), (Gómez-Pérez et al. 2004), and the most complete (SEKT 2003) and (KnowledgeWeb 2005). In this section we briefly describe some of the most representative ones. Let us recall that we have already analyzed some merging systems specific for geographic ontologies in 3.3.

The most related to our approach, in particular to our semi-automatic method, is PROMPT (Noy and Musen 2000), developed by the Stanford Medical Informatics Group at Stanford University and formerly called SMART (Noy and Musen 1999). PROMPT is the most popular ontology merging system since it has been implemented as a plug-in of the Protégé ontology editor[3]. PROMPT method is also based on a list of suggested operations that the system generates. PROMPT supports the following operations:

- “merge classes”, that identifies two classes from two ontologies as equivalent
- “merge properties”, that identifies two properties from two ontologies as equivalent
- “merge property bindings”, that identifies bindings between property and class as equivalent
- “shallow copy”, that copies a class from one ontology to the other one
- “deep copy”, that copies a class with all its subclasses from one ontology to the other one

The system also generates a list of the possible conflicts derived from these operations. Both lists are re-calculated when the user executes or modifies one operation from the

[3] http://protege.stanford.edu

list of suggestions. The algorithm that determines the list of suggested operations is mainly based on lexical similarities. However, other algorithms can be incorporated into the system, as it has been the case of the AnchorPROMPT algorithm (Noy and Musen 2001), which also considers structural aspects. It represents each ontology as a directed labelled graph, where classes are represented through nodes, and properties (relations) through arcs. An initial set of pairs of related terms in the two ontologies, called "anchors", is determined either manually or by means of lexical similarities. The algorithm takes each pair of anchors (n_1,n_1') and (n_2,n_2'), where n_1 and n_2 belong to one ontology and n_1' and n_2' to the other one, and analyzes those paths going from n_1 to n_2 and from n_1' to n_2' having the same length (shorter than a parameter). The algorithm is based on a cumulative score between nodes (classes) from both ontologies, that is incremented when two nodes appear in similar positions of these paths. Once all the pairs of anchors have been processed, the algorithm determines equivalences between pairs of classes according to the score. Is-a relation between classes could be considered in the same way as the other relations (properties). However, since it has a different meaning, AnchorPROMPT groups the classes related through is-a in the same node of the graph, which is called an "equivalence-group". A maximum size for equivalent-groups can be determined (in the case of maximum size 1, is-a relation is treated as the other relations). It has to be remarked that AnchorPROMPT only finds one-to-one equivalence mappings. Other types of relations or cardinalities are not considered. Note that anchors are also restricted to one-to-one equivalences.

Chimaera (McGuinness et al. 2000) is a well-known web-based environment for managing large ontologies that also implements a functionality for merging. Chimaera has been built on top of Ontolingua server by the Knowledge Systems Laboratory, also at Stanford University. Merging is mainly based on lexical similarities, involving not only class names, but also their definitions, and considering also derivatives and acronym and expanded forms of names.

Cupid (Madhavan et al. 2001) is a general database schema matcher that has been developed at Microsoft Research. It combines lexical, terminological (using an internal precompiled thesaurus) and structural methods (based on a context similarity in a tree structure). Other examples of database schema matchers are SemInt (Li and Clifton 2000), LSD (Doan et al. 2000; 2001), Artemis (Castano et al. 2000) or Coma (Do and Rahm 2002), among many others. Two surveys on database schema matching can be found in (Rahm and Bernstein 2001) and (Do et al. 2002).

FCA-Merge (Stumme and Maedche 2001) is based on Formal Concept Analysis (FCA). Let us recall from Chapter 3 that FCA is based on a mathematical definition of concepts through lattices (see (Ganter and Wille 1999) for more details). We have already described the merging method for geographic information of (Kavouras and Kokla 2002) that is also based on FCA. FCA-Merge method requires a collection of documents on the domain of the ontologies to be merged. It comprises three steps. The first step, instance extraction, obtains a table for each ontology relating its concepts to the documents that contain information about them. The second step, concept lattice computation, generates a lattice from the table containing all the concepts in both ontologies. This lattice is then pruned. The concepts from the pruned lattice are candidates to be concepts in the merged ontology. In the third step, generation of the final merged ontology, the ontology expert decides whether each candidate concept is finally included in the merged ontology. IF-Map (Kalfoglou and Schorlemmer 2003) is another FCA-based merging system. Glue (Doan 2002; Doan et al. 2004), an evolution of LSD, is another example of a merging system based on instances (although not in FCA), in this case applying statistical analysis and machine learning techniques.

5.2 Manual and semi-automatic methods

The objective of the merging process is to connect the values in a dataset with the thematic classes in the repository. This process usually requires the creation of new thematic classes that represent concepts that had not previously been considered.

The merging process is conducted according to one of the two merging methods that we have defined. The first one (see 5.2.1) is a manual method where a domain expert determines the relations between dataset values and thematic classes in the repository. The other method (see 5.2.2) automatically generates a list of suggested mapping actions between values and thematic classes. These mapping actions can be confirmed or rejected by the expert. Three different algorithms have been built for obtaining the list of suggestions. Both methods, as well as the three algorithms, have been implemented in the OntoGIS tool.

5.2.1 Manual method

The manual process of merging is carried out by a domain expert. This expert has to have a high familiarity with geography, particularly with the specific domain of the dataset being merged. This way, the expert will be able to conceptualize dataset values through thematic classes, and to identify relations between thematic classes. Ideally, this expert should not be required to have any knowledge about ontologies and logic. The tool should provide an intuitive way to establish relationships, hiding formal definitions.

This method comprises the following steps, which can be executed according the workflow presented in Figure 19:

1. Select a dataset (and see its properties)
2. Select a dataset value (and see its properties, particularly the definition)
3. Select the thematic class that closely describes the dataset value
4. Select the mapping action
5. Execute the mapping action

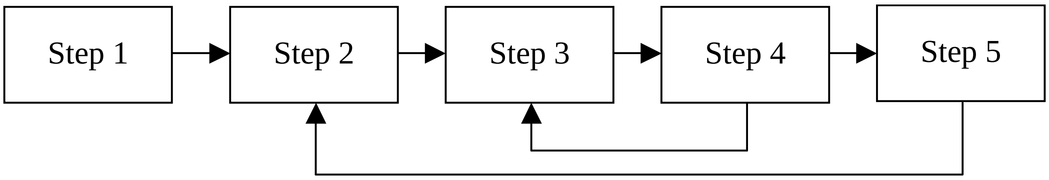
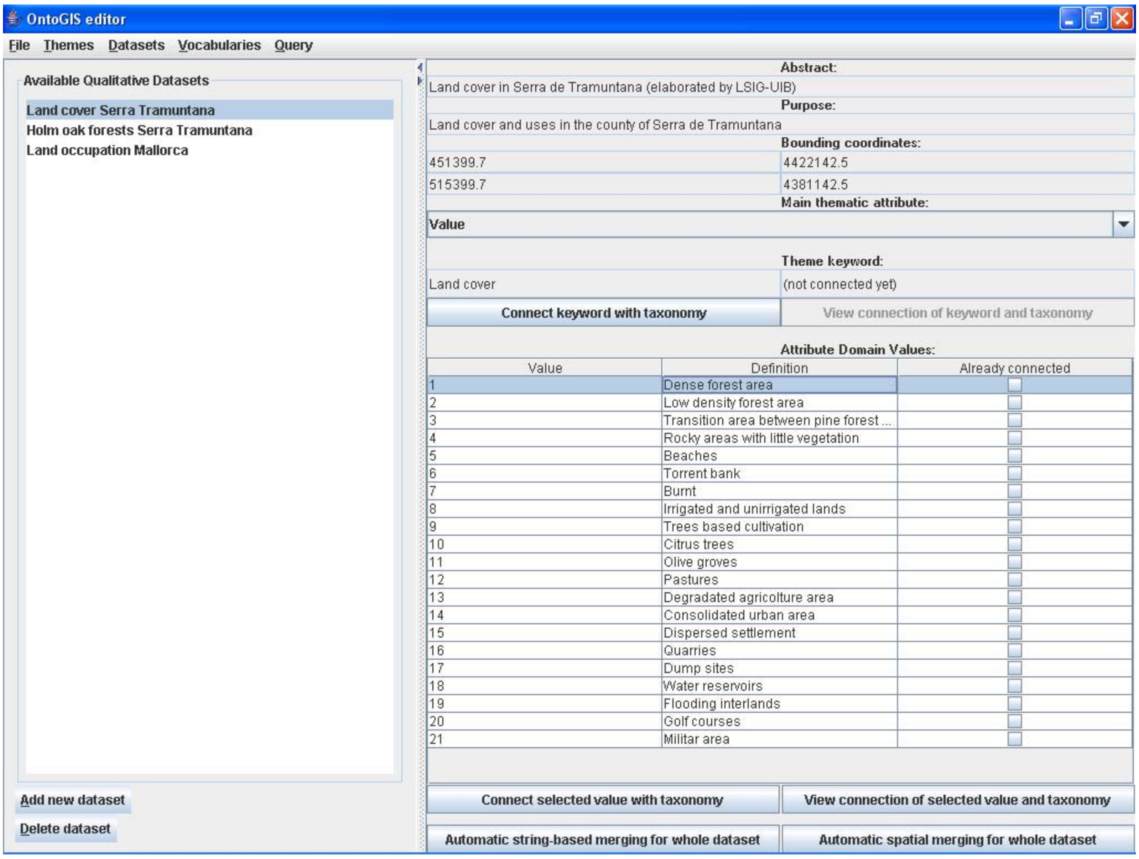


Figure 19. Workflow of the manual merging method

It may be worth recalling that the properties of datasets and dataset values are obtained from the metadata file of the dataset. Consequently, the method does not cover the creation of this metadata. It also has to be noted that the abstract value corresponding to the main theme of the dataset (read from the keyword element of the metadata file) can also be connected in exactly the same way as physical values.

Regarding steps 1 and 2, a graphical browsing of datasets and dataset values has to be provided to enable the user to easily visualize properties and to select datasets and values. Figure 20 shows an example of such capability in our implementation in OntoGIS. “Land cover Serra Tramuntana” has been selected among qualitative datasets (step 1). Value “1”, with definition “Dense forest area” has been selected among the values of the selected dataset (step 2).


OntoGIS editor
File
Themes
Datasets
Vocabularies
Query
Available Qualitative Datasets
Add new dataset
Delete dataset
Connect keyword with taxonomy
View connection of keyword and taxonomy
Connect selected value with taxonomy
View connection of selected value and taxonomy
Automatic string-based merging for whole dataset
Automatic spatial merging for whole dataset

Land cover Serra Tramuntana
Holm oak forests Serra Tramuntana
Land occupation Mallorca

Abstract:
Land cover in Serra de Tramuntana (elaborated by LSIG-UIB)

Purpose:
Land cover and uses in the county of Serra de Tramuntana

Bounding coordinates:

451399.7	4422142.5
515399.7	4381142.5

Main thematic attribute:
Value

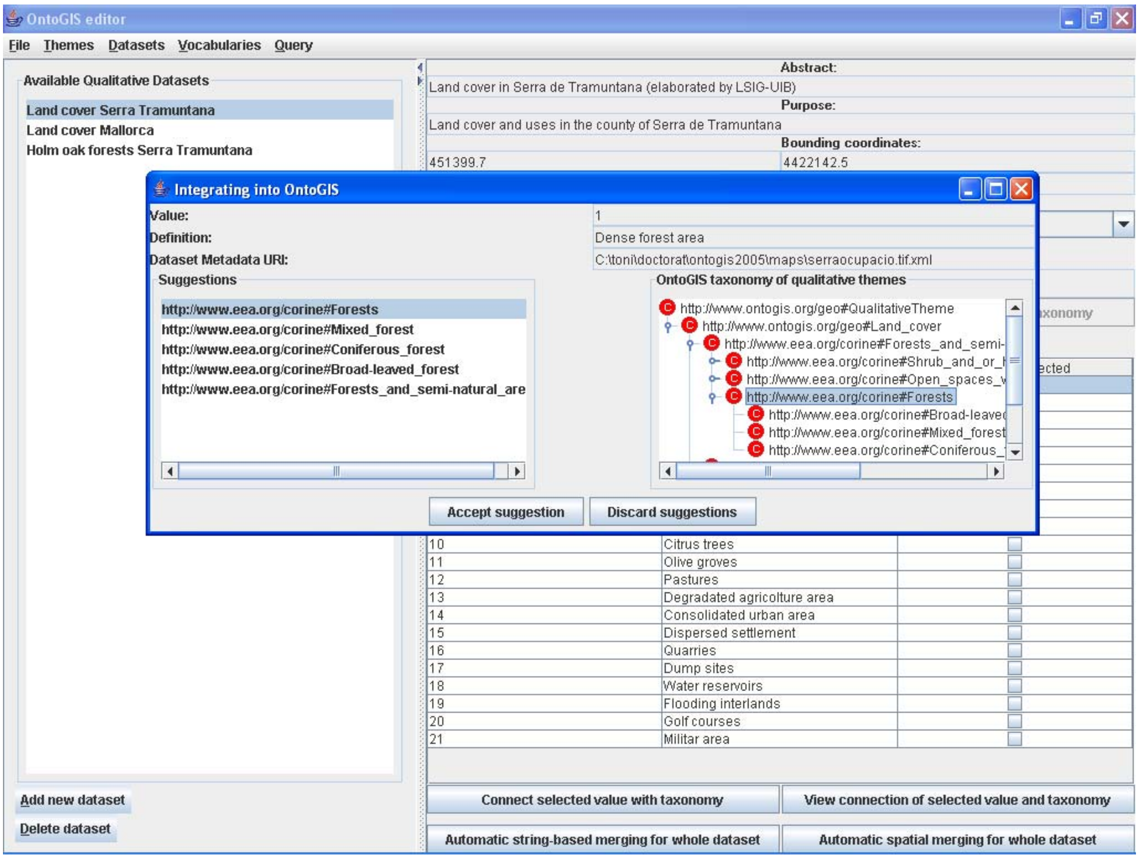
Theme keyword:
Land cover (not connected yet)

Attribute Domain Values:

Value	Definition	Already connected
1	Dense forest area	□
2	Low density forest area	□
3	Transition area between pine forest ...	□
4	Rocky areas with little vegetation	□
5	Beaches	□
6	Torrent bank	□
7	Burnt	□
8	Irrigated and unirrigated lands	□
9	Trees based cultivation	□
10	Citrus trees	□
11	Olive groves	□
12	Pastures	□
13	Degradated agricolture area	□
14	Consolidated urban area	□
15	Dispersed settlement	□
16	Quarries	□
17	Dump sites	□
18	Water reservoirs	□
19	Flooding interlands	□
20	Golf courses	□
21	Militar area	□

Figure 20. OntoGIS: selection of a dataset and a dataset value (steps 1 and 2 of the method)

Step 3 can be a hard task for the user, especially if the ontology comprises a big number of classes. To help the user to find the best class to be connected to the dataset value, a list of classes with similar names is generated. This is based on the assumption that similar names usually refer to similar concepts. The similarity measure based on class names that will be applied to the string-based mapping algorithm (see Chapter 6) is also used here. Figure 21 shows an example of the list of classes with similar names. In the example, the class “Forest” has been selected. Note that the namespace of the class indicates that it belongs to the CORINE vocabulary.


OntoGIS editor
File Themes Datasets Vocabularies Query
Integrating into OntoGIS
Accept suggestion
Discard suggestions
Add new dataset
Delete dataset
Connect selected value with taxonomy
View connection of selected value and taxonomy
Automatic string-based merging for whole dataset
Automatic spatial merging for whole dataset
xonomy
ected

Available Qualitative Datasets

- Land cover Serra Tramuntana
- Land cover Mallorca
- Holm oak forests Serra Tramuntana

Abstract:
Land cover in Serra de Tramuntana (elaborated by LSIG-UIB)

Purpose:
Land cover and uses in the county of Serra de Tramuntana

Bounding coordinates:
451399.7 4422142.5

Value: 1
Definition: Dense forest area
Dataset Metadata URI: C:\toni\doctorat\ontogis2005\maps\serraocupacio.tif.xml

Suggestions

- http://www.eea.org/corine#Forests
- http://www.eea.org/corine#Mixed_forest
- http://www.eea.org/corine#Coniferous_forest
- http://www.eea.org/corine#Broad-leaved_forest
- http://www.eea.org/corine#Forests_and_semi-natural_are

OntoGIS taxonomy of qualitative themes

- http://www.ontogis.org/geo#QualitativeTheme
  - http://www.ontogis.org/geo#Land_cover
    - http://www.eea.org/corine#Forests_and_semi-
      - http://www.eea.org/corine#Shrub_and_or_l
      - http://www.eea.org/corine#Open_spaces_v
      - http://www.eea.org/corine#Forests
        - http://www.eea.org/corine#Broad-leavec
        - http://www.eea.org/corine#Mixed_forest
        - http://www.eea.org/corine#Coniferous_

10	Citrus trees		□
11	Olive groves		□
12	Pastures		□
13	Degradated agricolture area		□
14	Consolidated urban area		□
15	Dispersed settlement		□
16	Quarries		□
17	Dump sites		□
18	Water reservoirs		□
19	Flooding interlands		□
20	Golf courses		□
21	Militar area		□

Figure 21. OntoGIS: selection of a class among those with a name similar to the selected value (steps 3 of the method)

Once one of the suggested classes has been selected, at step 4 the user has to select the mapping action, i.e. the relation between the value and the selected class. Note that in the usual case, this mapping action will involve the creation of a new class for the value. However, in the case of the class having the same or a very similar name, this will not be necessary. Note also that in any case, a connection will be created between the value and a class (either new or not). The following are the different possible mapping actions. Texts in italic correspond to the text presented to the user in our implementation, where "*V*" is the definition of the selected value and "*C*" the name of the selected class.

- *Add new class "V", equivalent to "C"*: through this operation a new thematic class is inserted in the ontology as equivalent to the selected class. The tool takes the definition of the value as the name for the new class and also creates the connection among the dataset value and the new class. This operation should be chosen if the value represents exactly the same meaning as the selected class, but they are lexically different: this is the case when value definition and class are synonyms.
- *No need to add class "V", represented through "C"*: this operation should be chosen when the definition of the value is so similar (or equal) to the name of the thematic class that there is no need to add it as a new class. This happens when both are expressed through the same or derivative words. It simply creates

a connection between the dataset value and the selected thematic class, and no new class will be added.

- *Add new class “V”, subclass of “C”*: in this case, a new thematic class is added as a subclass of the selected class, and the new class is connected to the dataset value. The name of the new class is taken from the definition of the value. This operation should be chosen when the value represents a specialization of the selected theme.
- *Add new class “V”, subclass of “C” with restriction of property value*: this operation creates the new class as a subclass of the selected theme, sets the connection with the dataset value, and also allows the expert to define a restriction of a property value for the new class. This way, the user can create different classifications for the selected theme, each one restricting a particular property. The name of the new class is again taken from the definition of the value. The user will select if the union of all the existing classes restricting the same property value is equivalent to their superclass. Note that in this case the subclasses form a partition of the superclass, since their intersection is empty and their union is the superclass.
- *Add new class “V”, superclass of “C”*: in this case, the new thematic class is added as a superclass of the selected class and is connected to the dataset value. The superclasses of the selected theme are assumed to be also superclasses of the new class (otherwise the user should discard the suggestion). Again, the name of the new class is taken from the definition of the value. This operation should be chosen when the value represents a generalization of the selected theme.

It has to be noted that restrictions of property values provide an integrated way for determining many-to-many relations. Let us consider an example where a theme *T* has been classified in two different ways in two datasets: for instance, *A* and *B* in one dataset form a partition according to property *p1*; and *X*, *Y* and *Z* in the other dataset form another partition according to property *p2*. Although there are usually not subclass relations among *A, B, X, Y* and *Z*, there exists a many-to-many equivalence relation among them, since the union of *A* and *B* has been determined to be equivalent to the union of *X, Y* and *Z*.

Once the user selects a mapping action, it is executed at step 5. The corresponding classes and connections are created. The dataset value is marked in the interface as already connected, since a dataset value can only be connected to at most one class. The process can be continued at step 2 by selecting more values, until all values are connected.

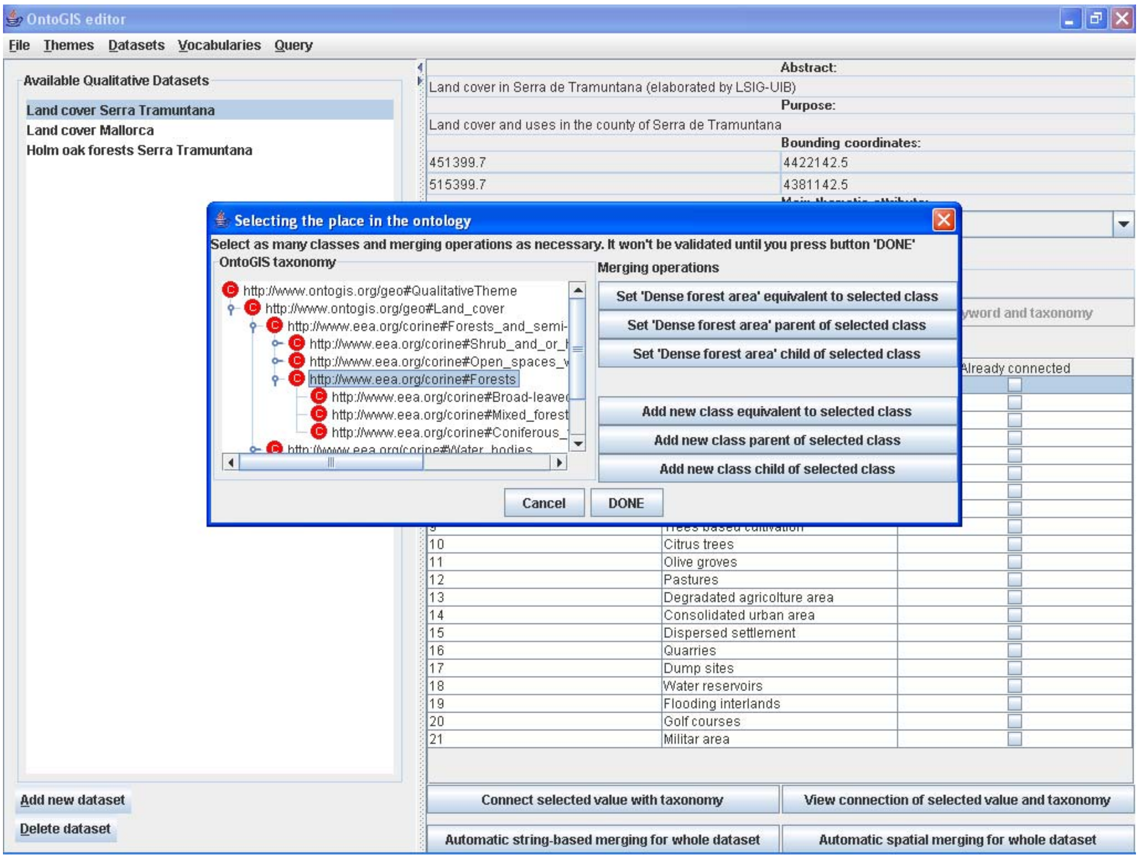
This workflow has an alternative scenario when one of the following conditions occurs at step 3: either there is no thematic classes with a name similar to the value definition, or the user discards the suggested thematic classes. In this case, a new class will be created to represent the value. Note that the option of creating just the connection is not present in this case, since classes with names similar to the value, if any, have been discarded. The user has to decide the place in the taxonomy where the new class will be created. Note that this may include several mapping actions between the new class and other existing classes. Consequently, several iterations of steps 3 and 4 are carried out.

This way, in each of those iterations the user selects a thematic class and a relation between this selected class and the new class (connected to the value).

Furthermore, in some occasions, especially when the repository ontology contains little information regarding the theme being merged, it may also be necessary to add other classes apart from the one connected to the value. This way, once the user has selected a class from the repository, s/he has the following merging actions available, where "*V*" is the definition of the selected value and consequently the name of the new class connected to the value:

- Set *V* as equivalent to selected class
- Set *V* as subclass of selected class
- Set *V* as superclass of selected class
- Add new class as equivalent to selected class
- Add new class as subclass of selected class
- Add new class as superclass of selected class

Figure 22 shows how this alternative scenario has been implemented in OntoGIS. Note that superclass is tagged as "parent" while subclass is tagged as "child".


OntoGIS editor
File
Themes
Datasets
Vocabularies
Query
Available Qualitative Datasets
Add new dataset
Delete dataset
Selecting the place in the ontology
OntoGIS taxonomy
Merging operations
Set 'Dense forest area' equivalent to selected class
Set 'Dense forest area' parent of selected class
Set 'Dense forest area' child of selected class
Add new class equivalent to selected class
Add new class parent of selected class
Add new class child of selected class
Cancel
DONE
yword and taxonomy
Connect selected value with taxonomy
View connection of selected value and taxonomy
Automatic string-based merging for whole dataset
Automatic spatial merging for whole dataset

Land cover Serra Tramuntana
Land cover Mallorca
Holm oak forests Serra Tramuntana

Abstract:
Land cover in Serra de Tramuntana (elaborated by LSIG-UIB)

Purpose:
Land cover and uses in the county of Serra de Tramuntana

Bounding coordinates:

451399.7	4422142.5
515399.7	4381142.5

Select as many classes and merging operations as necessary. It won't be validated until you press button 'DONE'

```
http://www.ontogis.org/geo#QualitativeTheme
  http://www.ontogis.org/geo#Land_cover
    http://www.eea.org/corine#Forests_and_semi-
      http://www.eea.org/corine#Shrub_and_or_
      http://www.eea.org/corine#Open_spaces_v
      http://www.eea.org/corine#Forests
        http://www.eea.org/corine#Broad-leaved
        http://www.eea.org/corine#Mixed_forest
        http://www.eea.org/corine#Coniferous_
    http://www.eea.org/corine#Water_bodies
```

			Already connected
			□
			□
			□
			□
			□
			□
			□
			□
9	Trees based cultivation		□
10	Citrus trees		□
11	Olive groves		□
12	Pastures		□
13	Degradated agricolture area		□
14	Consolidated urban area		□
15	Dispersed settlement		□
16	Quarries		□
17	Dump sites		□
18	Water reservoirs		□
19	Flooding interlands		□
20	Golf courses		□
21	Militar area		□

Figure 22. OntoGIS: selection of classes and relations (steps 3 and 4) in the alternative scenario

5.2.2 Semi-automatic method

Unlike the manual merging method that deals with isolated values, the semi-automatic method considers the whole dataset structure. It is based on a list of mapping actions that are automatically generated, at least one for each dataset value. The execution of these mapping actions is not automatic and requires the confirmation of an expert. This method usually permits the user to merge the dataset in a faster and easier way with respect to the manual method, especially in the case where the repository ontology contains a big number of classes. As in the case of the manual method, the user is expected to be an expert with a high knowledge in the specific domain of the dataset being merged.

This method comprises the following steps, which can be executed according the workflow shown in Figure 23:

1. Select a dataset (and see its properties and dataset values)
2. Define a hierarchy structure of dataset values (optional)
3. Automatic generation of suggested list of mapping actions
4. Confirm or modify mapping actions

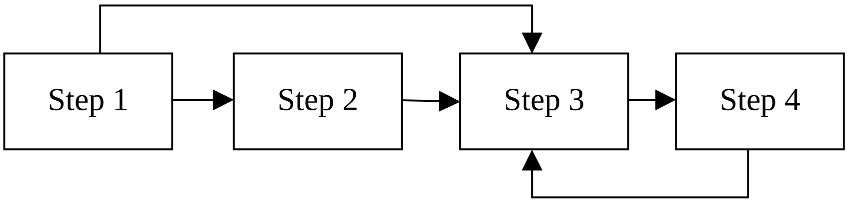
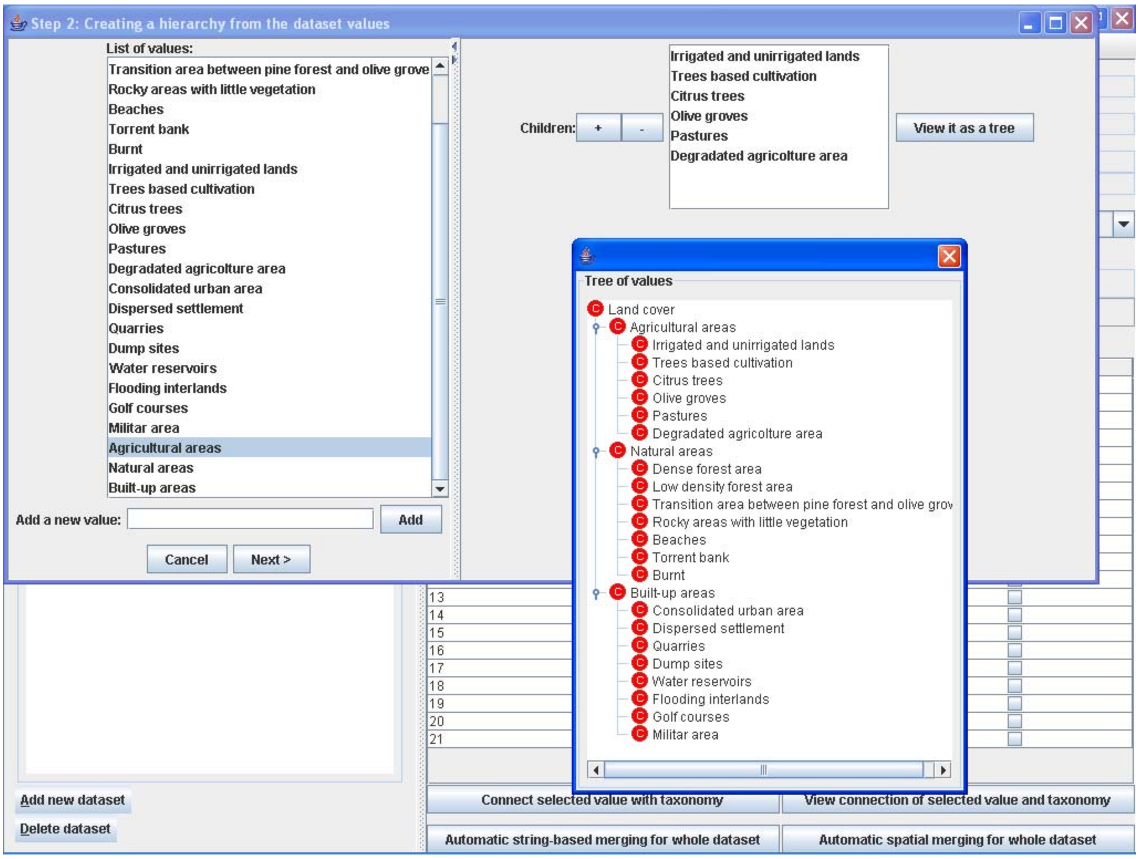


Figure 23. Workflow of the semi-automatic merging method

Regarding step 1, a graphical browsing of datasets and dataset values has to be provided as explained in the previous section (see also Figure 20). It has to be noted again that dataset and values properties are extracted from metadata files.

As we have already mentioned in previous chapters, values of qualitative datasets are often conceptually organized in a hierarchy. This hierarchy is sometimes made explicit in the legend of the maps, although only leaf nodes (physical values) in this hierarchy appear in data (and metadata). The hiearchy will comprise physical values as well as abstract values. The user at step 2 may add as many abstract values as necessary building the hierarchy. Figure 24 shows the implementation of this step in the OntoGIS tool. This is an optional task, since the algorithms do not need this hierarchy to work. However, they can obtain better results if the hierarchy is provided.

Step 3 is clearly the key step of this method. As we have already mentioned, we have developed and tested three different algorithms for the generation of the list of suggested mapping actions (mapping algorithms). The first one (Chapter 6) is based on the lexical information and graph structures. The second one (Chapter 7) also uses a thesaurus in order to find synonyms, hypernyms and hyponyms. Finally, the third one (Chapter 8) is based on the spatial distribution of values in datasets. All three have been implemented in OntoGIS, where the user can select the algorithm to be used.


Step 2: Creating a hierarchy from the dataset values
List of values:
Add a new value:
Add
Cancel
Next >
Children:
+
-
View it as a tree
Tree of values
Add new dataset
Delete dataset
Connect selected value with taxonomy
View connection of selected value and taxonomy
Automatic string-based merging for whole dataset
Automatic spatial merging for whole dataset
13
14
15
16
17
18
19
20
21

List of values:

- Transition area between pine forest and olive grove
- Rocky areas with little vegetation
- Beaches
- Torrent bank
- Burnt
- Irrigated and unirrigated lands
- Trees based cultivation
- Citrus trees
- Olive groves
- Pastures
- Degradated agricolture area
- Consolidated urban area
- Dispersed settlement
- Quarries
- Dump sites
- Water reservoirs
- Flooding interlands
- Golf courses
- Militar area
- Agricultural areas
- Natural areas
- Built-up areas

Children:

- Irrigated and unirrigated lands
- Trees based cultivation
- Citrus trees
- Olive groves
- Pastures
- Degradated agricolture area

Tree of values

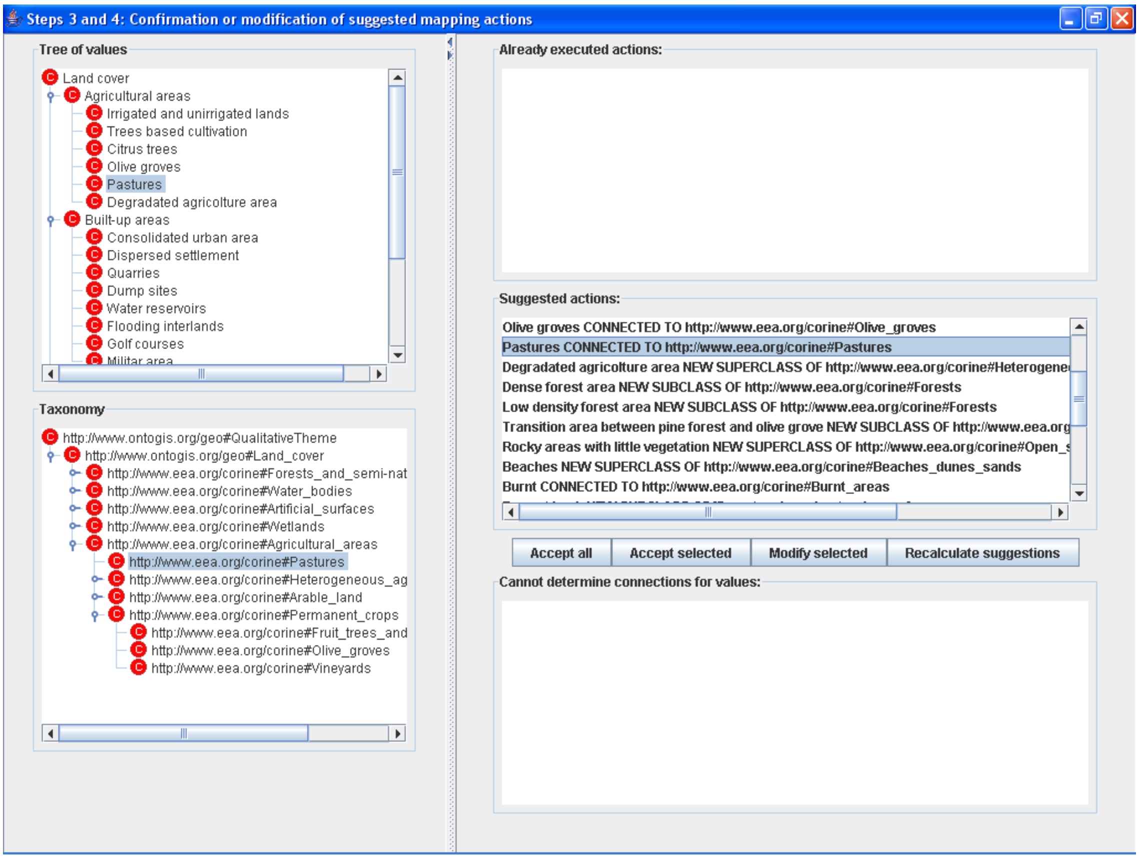
- Land cover
  - Agricultural areas
    - Irrigated and unirrigated lands
    - Trees based cultivation
    - Citrus trees
    - Olive groves
    - Pastures
    - Degradated agricolture area
  - Natural areas
    - Dense forest area
    - Low density forest area
    - Transition area between pine forest and olive grov
    - Rocky areas with little vegetation
    - Beaches
    - Torrent bank
    - Burnt
  - Built-up areas
    - Consolidated urban area
    - Dispersed settlement
    - Quarries
    - Dump sites
    - Water reservoirs
    - Flooding interlands
    - Golf courses
    - Militar area

Figure 24. OntoGIS: Definition of a hierarchy of dataset values (step 2)

Finally, at step 4, the user can confirm, and consequently execute, either all mapping actions or a selection of them. Alternatively, s/he can modify a selected mapping action. In this case, the corresponding value will be merged manually as described in the previous section. At any moment, after confirming or modifying any number of mapping actions, the user may determine to recalculate the list of suggestions. The method goes consequently back to step 3. Since the algorithm has more information from the actions that have already been confirmed or modified by the user, it may produce different results than in the previous iteration. Summarizing, the expert has the following options once the list of suggestion has been generated:

- Execute all the merging actions from the list at once
- Execute a particular merging action from the list
- Modify a particular merging action and execute it
- Recalculate the list of suggestions, taking into account the operations that have already been executed (eventually with modifications)

Figure 25 shows an example of a list of suggested actions (see list at the middle of the right panel). The selected action can be accepted or modified by the user. Note also that when the user selects an action, the involved value and repository class are highlighted in the left panel.


Steps 3 and 4: Confirmation or modification of suggested mapping actions
Tree of values
Land cover
Agricultural areas
Irrigated and unirrigated lands
Trees based cultivation
Citrus trees
Olive groves
Pastures
Degradated agricolture area
Built-up areas
Consolidated urban area
Dispersed settlement
Quarries
Dump sites
Water reservoirs
Flooding interlands
Golf courses
Militar area
Taxonomy
http://www.ontogis.org/geo#QualitativeTheme
http://www.ontogis.org/geo#Land_cover
http://www.eea.org/corine#Forests_and_semi-nat
http://www.eea.org/corine#Water_bodies
http://www.eea.org/corine#Artificial_surfaces
http://www.eea.org/corine#Wetlands
http://www.eea.org/corine#Agricultural_areas
http://www.eea.org/corine#Pastures
http://www.eea.org/corine#Heterogeneous_ag
http://www.eea.org/corine#Arable_land
http://www.eea.org/corine#Permanent_crops
http://www.eea.org/corine#Fruit_trees_and
http://www.eea.org/corine#Olive_groves
http://www.eea.org/corine#Vineyards
Already executed actions:
Suggested actions:
Accept all
Accept selected
Modify selected
Recalculate suggestions
Cannot determine connections for values:

Olive groves CONNECTED TO http://www.eea.org/corine#Olive_groves
Pastures CONNECTED TO http://www.eea.org/corine#Pastures
Degradated agricolture area NEW SUPERCLASS OF http://www.eea.org/corine#Heterogene
Dense forest area NEW SUBCLASS OF http://www.eea.org/corine#Forests
Low density forest area NEW SUBCLASS OF http://www.eea.org/corine#Forests
Transition area between pine forest and olive grove NEW SUBCLASS OF http://www.eea.org
Rocky areas with little vegetation NEW SUPERCLASS OF http://www.eea.org/corine#Open_s
Beaches NEW SUPERCLASS OF http://www.eea.org/corine#Beaches_dunes_sands
Burnt CONNECTED TO http://www.eea.org/corine#Burnt_areas

Figure 25. OntoGIS: Suggested mapping actions to be accepted or modified by the user (steps 3 and 4)

It has to be noted that the objective of the algorithm is to obtain a list that requires little modification by the user. The ideal situation takes place when the user directly confirms all the operations from the list at once, with no modifications.

5.3 A method for merging quantitative datasets

While the process of merging qualitative datasets is considerably complex, we have considered a simplified merging for the quantitative case.

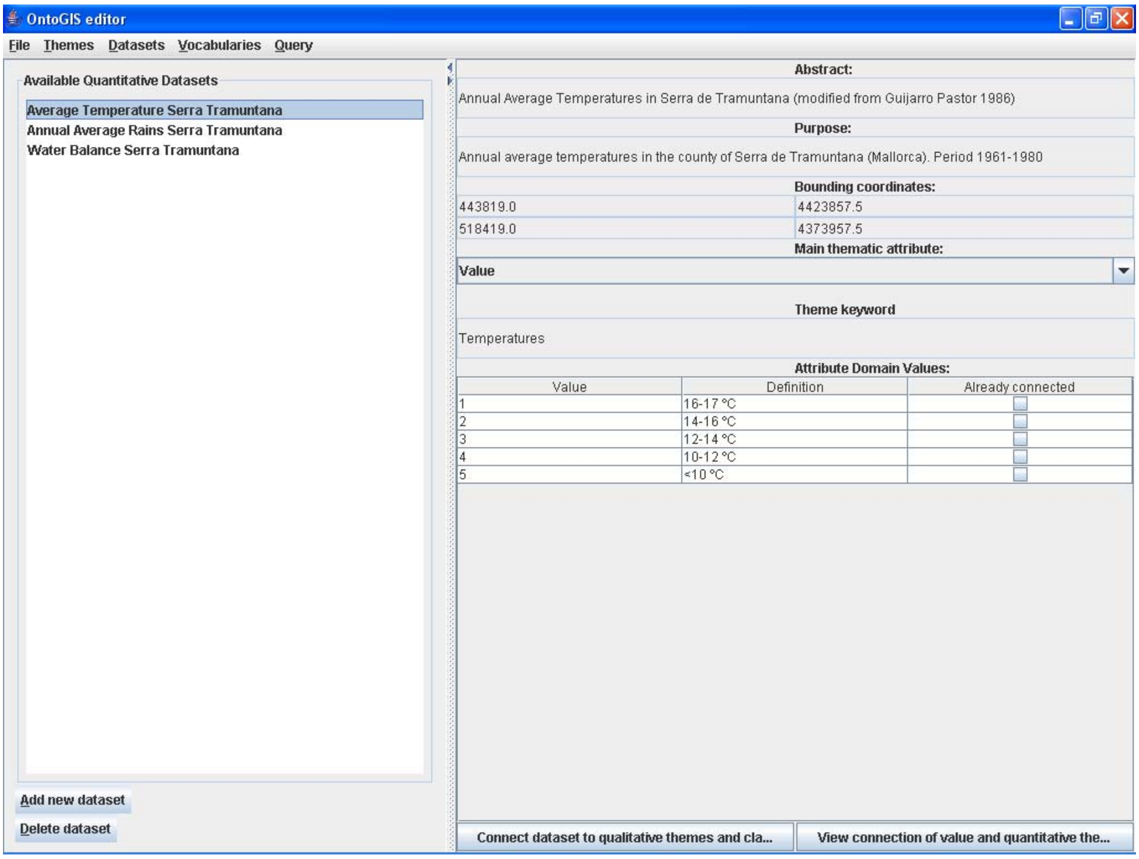
As we have already mentioned in Chapter 4, we consider that quantitative themes are not semantically related to each other. This way, the integration of two datasets of the same quantitative theme would be meaningless, since if a particular spatial unit has two different values in these datasets, no logical conclusion can be achieved.

Consequently, the merging method for quantitative datasets aims at providing a quantitative theme for the overall dataset, as well as a quantitative classification where each dataset value (an interval of usually numeric values) is mapped to a quantitative class that identify the thresholds of the interval. This is manually done by the expert.

The method is organized in the following steps:

1. Select a quantitative dataset (and see its properties and dataset values)
2. Set a quantitative theme for the dataset
3. Set an appropriate quantitative classification
4. Set a quantitative class for each dataset value

Step 1 permits the user to select a quantitative dataset from the repository and see its main properties, as well as its values (see Figure 26).


OntoGIS editor
File
Themes
Datasets
Vocabularies
Query
Add new dataset
Delete dataset
Connect dataset to qualitative themes and cla...
View connection of value and quantitative the...

Available Quantitative Datasets

- Average Temperature Serra Tramuntana
- Annual Average Rains Serra Tramuntana
- Water Balance Serra Tramuntana

Abstract:

Annual Average Temperatures in Serra de Tramuntana (modified from Guijarro Pastor 1986)

Purpose:

Annual average temperatures in the county of Serra de Tramuntana (Mallorca). Period 1961-1980

Bounding coordinates:

443819.0	4423857.5
518419.0	4373957.5

Main thematic attribute:

Value

Theme keyword

Temperatures

Attribute Domain Values:

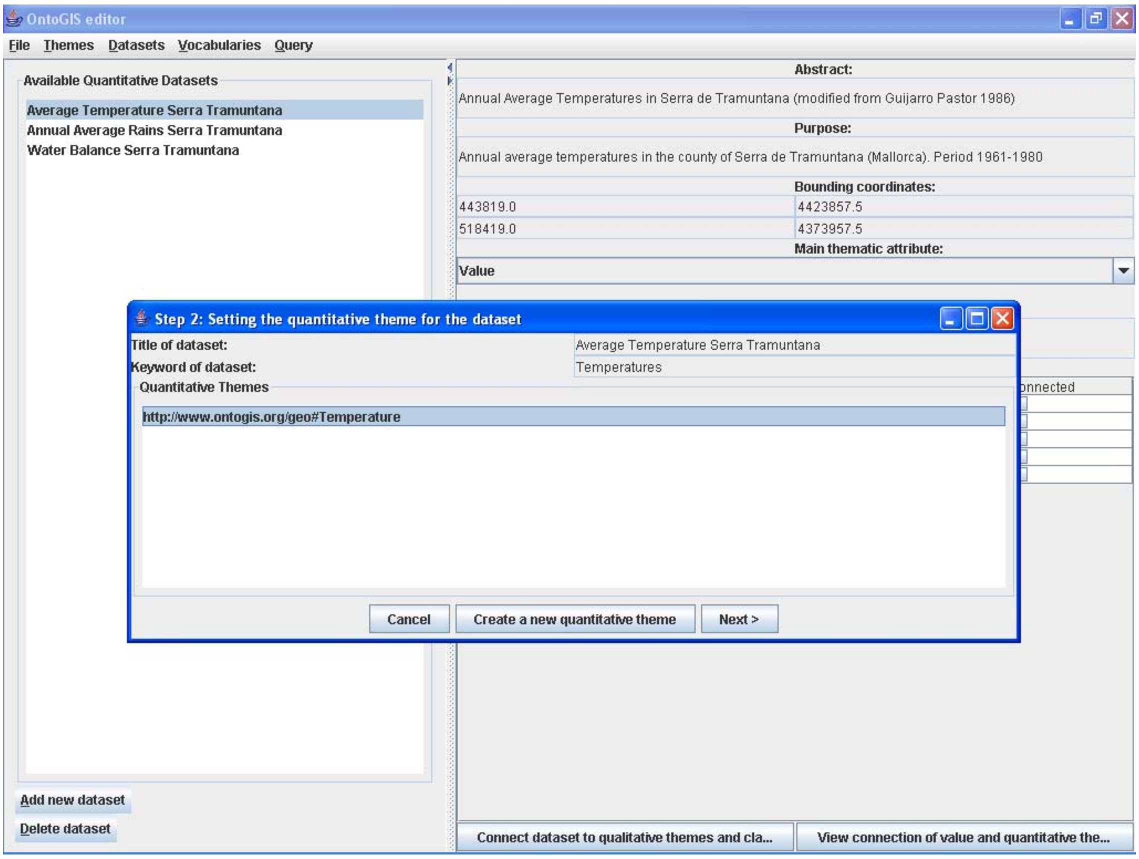
Value	Definition	Already connected
1	16-17 °C	□
2	14-16 °C	□
3	12-14 °C	□
4	10-12 °C	□
5	<10 °C	□

Figure 26. OntoGIS: Selection of a quantitative dataset (Average temperature Serra Tramuntana)

In Step 2 the user selects one of the available quantitative themes that better describes the theme of the dataset. A list of those with a similar name to the theme keyword of the dataset is presented to the user. Alternatively, the user may define a new quantitative theme (see Figure 27).

In Step 3 the user selects one of the quantitative classifications existing for the selected quantitative theme. Only those classifications with the same number of classes than the number of dataset values are considered. Alternatively the user may create a new classification.

Finally, in Step 4, the user assigns each quantitative class to its corresponding dataset value. In the case of a new classification, the user has to define the properties of the quantitative class by setting the thresholds of the interval. Figure 28 shows how the user has created a new quantitative classification in the left panel, and how in the right panel s/he defines the intervals (quantitative classes) for each dataset value.



OntoGIS editor
File Themes Datasets Vocabularies Query
Available Quantitative Datasets
Add new dataset
Delete dataset
Connect dataset to qualitative themes and cla...
View connection of value and quantitative the...
onnected

Average Temperature Serra Tramuntana
Annual Average Rains Serra Tramuntana
Water Balance Serra Tramuntana

Abstract:
Annual Average Temperatures in Serra de Tramuntana (modified from Guijarro Pastor 1986)

Purpose:
Annual average temperatures in the county of Serra de Tramuntana (Mallorca). Period 1961-1980

Bounding coordinates:

443819.0	4423857.5
518419.0	4373957.5

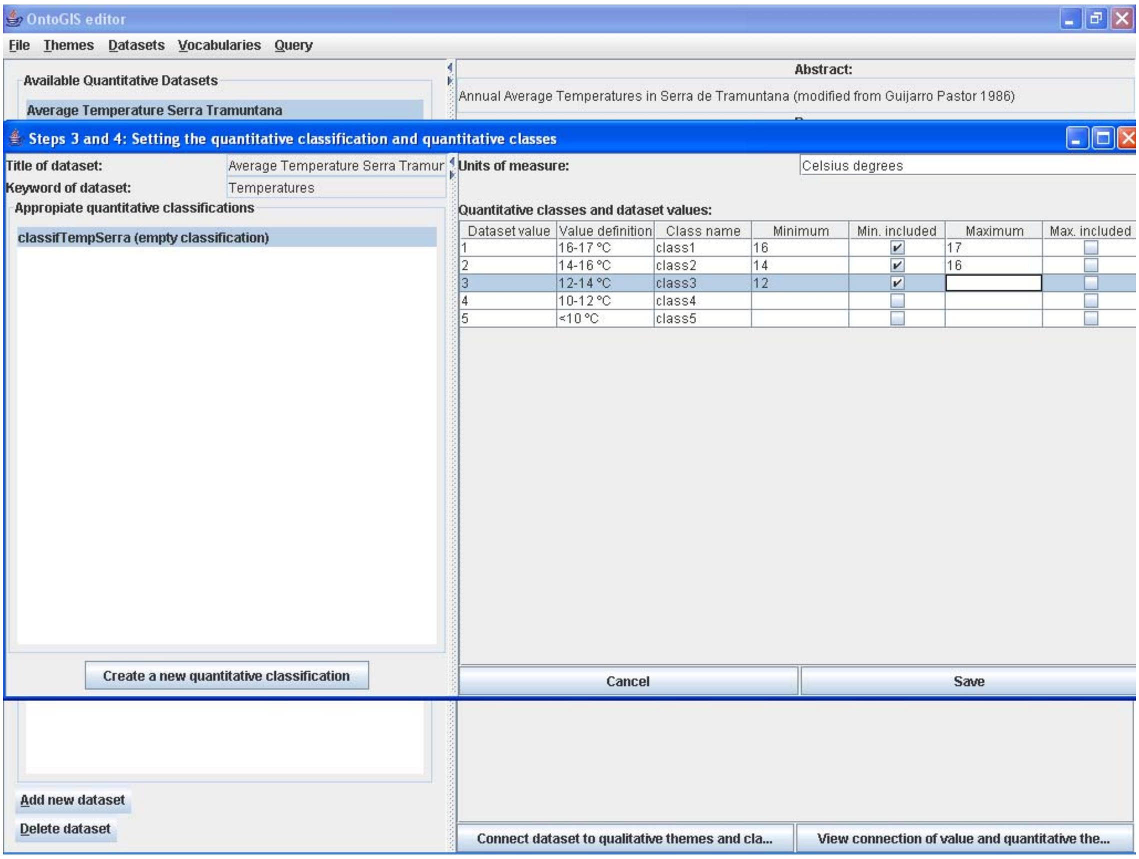
Main thematic attribute: Value

Step 2: Setting the quantitative theme for the dataset

Title of dataset: Average Temperature Serra Tramuntana
Keyword of dataset: Temperatures

Quantitative Themes
http://www.ontogis.org/geo#Temperature

Figure 27. OntoGIS: Setting a quantitative theme (Temperature) for the dataset



OntoGIS editor
File Themes Datasets Vocabularies Query
Available Quantitative Datasets
Create a new quantitative classification
Cancel
Save
Add new dataset
Delete dataset
Connect dataset to qualitative themes and cla...
View connection of value and quantitative the...

Average Temperature Serra Tramuntana

Abstract:
Annual Average Temperatures in Serra de Tramuntana (modified from Guijarro Pastor 1986)

Steps 3 and 4: Setting the quantitative classification and quantitative classes

Title of dataset: Average Temperature Serra Tramur
Keyword of dataset: Temperatures

Appropiate quantitative classifications
classifTempSerra (empty classification)

Units of measure: Celsius degrees

Quantitative classes and dataset values:

Dataset value	Value definition	Class name	Minimum	Min. included	Maximum	Max. included
1	16-17 °C	class1	16	☑	17	□
2	14-16 °C	class2	14	☑	16	□
3	12-14 °C	class3	12	☑		□
4	10-12 °C	class4		□		□
5	<10 °C	class5		□		□

Figure 28. OntoGIS: Setting a quantitative classification and a quantitative class for each value

6 String-based mapping algorithm

This algorithm is mainly based on the lexical similarities among names of classes in the two ontologies being merged. The heuristics of this algorithm is based on the fact that if two classes from different ontologies have the same name, they normally refer to the same concept; but even if they are not equal but share one or more terms, they probably refer to related concepts. However, a pure string matching algorithm skips relations among synonyms. For instance, there is a clear semantic relation between “pine forest” and “wood”, although string matching techniques will not find it. A terminological base can be used to detect this type of relations. We present in this section an algorithm that focuses on string matching, while Chapter 7 describes the terminological approach. The internal graph structures of the ontologies to be merged are also considered in both approaches.

In both cases, stop words (articles, determiners, prepositions and conjunctions) are removed from both strings before they are compared. And so are substantives referring to spatial units, such as area (for instance, in forest area), zone or land, among others, as well as their derivatives.

As it has been already discussed in Chapter 5, in our context, one of the two ontologies corresponds to the application ontology of a dataset, while the other refers to the higher level ontology of the repository. However, it is worth recalling that in our case, the application ontology is represented through a set of individuals of *QualitativeDataset* and *QualitativeDatasetValue* classes. We provide here a generic algorithm dealing with two ontologies; its implementation in the OntoGIS tool just requires a straightforward adaptation to deal with individuals of *QualitativeDatasetValue* instead of classes of the dataset ontology.

In Section 6.1 we describe the similarity measures that are used in the mapping algorithm, and how the type of mapping between two classes (the relation between them) is also obtained from these similarity measures. Section 6.2 presents the algorithm, which is driven by the structure through the mechanism of what we have called mapping restrictions. Structure also influences the mapping algorithm by means of what we call structural rules, that are discussed in 6.2.1. Section 6.3 analyzes some particular cases that deserve special attention and require a slight modification of the general algorithm. Finally, Section 6.4 presents the final algorithm considering all the special cases described in the previous section.

6.1 Similarity measures and mapping actions

The string-based mapping algorithm is based on a similarity measure between the names of each pair of classes. For each of these pairs, their names are split into terms which are compared between them. We have used in our implementation a variation of the substring method in order to compute the similarity between two terms, although any of the similarity functions discussed in Section 5.1 could be used instead. Our variation of the substring method considers only substrings starting at the first character of the term with a length of 3 or more characters. This way, a term and another derived from it by adding a prefix are not considered as related. Nevertheless, our experience shows us that prefixes are seldom used in our context, and by discarding them the roots of terms are given more importance. It is worth noting that in this context, term and word are used as synonyms, since terms comprising more than one word are not considered here. However, we will see in Chapter 7 that compound terms are considered in the thesaurus-based approach.

According to these premises, we define the term similarity function, *tsim*, between two terms *u* and *v* in the following way:

$$tsim(u,v) = \frac{2 \cdot length(x)}{length(u) + length(v)} \ ,$$

where *x* is the longest common substring between *u* and *v* beginning at the first character of both and containing at least 3 characters.

Considering that a complete string (a class name, in our context) is represented by a set of terms, our global similarity function is defined as the average of the best possible *tsim* for each term in the set. We define the global function of similarity, *sim*, between two complete strings *S*={s_1, ..., s_n} and *T* as follows:

$$sim(S,T) = \frac{\sum_{i=1}^{n} max_{t \in T}(tsim(s_i,t))}{n}$$

As we can observe our similarity measure satisfies the maximality principle since *sim*(*S*,*S*) is clearly always 1. But it does not satisfy symmetry and the triangle inequality. Let us consider the example where *A*={*a*, *b*, *c*}, *B*={*b*, *c*, *d*, *e*}, *C*={*d*, *e*}, and where *a*, *b*, *c*, *d* and *e* are terms. We can observe that *sim*(*A*,*B*) = 2/3 and *sim*(*B*,*A*)=1/2, which breaks the symmetric principle. Furthermore, *sim*(*A*,*B*)=2/3, *sim*(*B*,*C*)=1/2 and *sim*(A,*C*)=0, which breaks the triangle inequality, that in terms of similarities is expressed as *sim*(*A*,*B*) + *sim*(*B*,*C*) ≤ *sim*(*A*,*C*) + 1. Note that this expression is obtained from the typical triangle inequality with distances, *dist*(*A*,*B*) + *dist*(*B*,*C*) ≥ *dist*(*A*,*C*), where distance and similarity are opposite, *dist*(*A*,*B*) = 1 - *sim*(*A*,*B*). It has to be noted that since *tsim* function is symmetric, the similarity *sim* between strings comprising a single term is symmetric too, although does not satisfy the triangle inequality.

Our definition of the *sim* measure presents some similarities with the recursive field matching algorithm defined by (Monge and Elkan 1996), which instead of a term similarity function considers a *match* function that returns 1 in the case of one string being equal or abbreviating the other and 0 otherwise. They use it in the different context of record linkage: they compare text records in large databases that are written in different ways but are related, as for instance “Univ.” and “University”, “Dept.” and “Department” or “Caltech” and “California Institute of Technology”.

In our case, this asymmetric similarity allows us to represent not only if two strings are linked, but also to determine how one string is contained in the other. For instance, $sim(A,B)=1$ and $sim(B,A)=1/2$ mean that all the terms in A are contained in B, which adds new terms, usually providing a more specific information. In consequence, our algorithm can entail that B is a subclass of A, or in other words, A subsumes B.

Nevertheless, whenever we need an indicator of how similar two elements are in both directions, we use an average similarity, *avsim,* defined as follows:

$$avsim(S,T) = \frac{sim(S,T) + sim(T,S)}{2}$$

This average function *avsim* is clearly symmetric, although does not satisfy the triangle inequality.

Let us consider the example where $A=\{a, b\}$, $B=\{a, b\}$, $C=\{a, b, c, d\}$, and where a, b, c and d are terms. Observe that $sim(A,B)=sim(A,C)$, but obviously A is more similar to B (in fact equal) than to C. We can see that *avsim* is a better indicator since $avsim(A,B)=1$ and $avsim(A,C)=3/4$.

Hence, given a class in one ontology, *avsim* can be used to obtain its more similar class in the other ontology, and then *sim* can be used to entail which one of these two classes is subsumed by the other one.

This way, two names A and B are equal if $avsim(A,B)=1$. But often a term in one expression is a derivative of another term in the other, or there are small variations in long expressions of several terms. In order to consider these situations, we define a threshold, called *equivalence threshold*, λ. The algorithm will consider that A and B are equivalent if both $sim(A,B)$ and $sim(B,A)$ are greater than or equal to λ. Although the user may modify it, we have set 0.75 as the default value for λ. This fixes the threshold in a case where A and B comprise 4 words each and 3 of them match. Using the default value, A and B will be assumed as equivalent, but in the example would not if they had more than 4 words or less than 3 matched words.

Similarly, if $avsim(A,B)=0$, A and B can be considered as not related. But they are not related either in the case of a small similarity, and another threshold, *no-relation threshold*, μ, has been defined. If both $sim(A,B)$ and $sim(B,A)$ are smaller than or equal to μ they are considered as not related. Although the user may also modify it, the default value for μ is 0.25. It may be worth clarifying that although the default values satisfy the expression $\mu = 1- \lambda$, the user may set other values for λ and μ that do not satisfy it.

According to these similarity measures and thresholds, the mapping action between two classes C_D with name S_D and C_R with name S_R (where C_D is a class from the dataset ontology and C_R from the repository ontology) is determined through the following criteria:

1. If $sim(S_D,S_R) \geq \lambda$ and $sim(S_R,S_D) \geq \lambda$, both names are considered equivalent and the algorithm suggests the action of adding a connection between C_D and C_R. Note that no new class is added in the repository ontology, since there already exists a class with the same (or very similar) name
2. If $sim(S_D,S_R) \leq \mu$ and $sim(S_D,S_R) \leq \mu$, the algorithm deduces that there is no relation between C_D and C_R
3. Otherwise, the algorithm assumes that there exists a relation, although it is not an equivalence. These are the different possibilities:
   - 3.a. If $sim(S_D,S_R) \geq \lambda$, and in consequence $sim(S_R,S_D)<\lambda$, it is assumed that almost all the terms in S_D are contained in S_R, while the extra terms in S_R specialize the common information. Therefore, the algorithm suggests that C_R is a subclass of C_D, $C_R \sqsubseteq C_D$
   - 3.b. In the same way, if $sim(S_R,S_D) \geq \lambda$, and consequently $sim(S_D,S_R)<\lambda$, the algorithm suggests that $C_D \sqsubseteq C_R$
   - 3.c. If $sim(S_D,S_R)<\lambda$ and $sim(S_R,S_D)<\lambda$, it is assumed that they share information but specialize it in different ways. In consequence they have a common superclass C_X: $C_D \sqsubseteq C_X$ and $C_R \sqsubseteq C_X$. The name S_X of the new class C_X is extracted from the common part of S_D and S_R

It has to be noted that we would like to provide at least one superclass and one subclass relation for each dataset class C_D. In the case of condition (1), the subclass and superclass of C_D is the same repository class C_R (equivalent to C_D). But let us briefly discuss the case of condition 3. If C_D is set as superclass of C_R (3.a), we are also interested in determining whether another repository class being superclass of C_R is also superclass of C_D. The same can be said in the case of C_D subclass of C_R (3.b), where we are interested in checking if any subclass of C_R is also subclass of C_D. And finally, in the case of common superclass (3.c), we are interested in checking whether C_X is subclass of any superclass of C_R. In this last case, note that C_X may also be subclass of one (or more) of the superclasses of C_D.

6.2 Structure-driven algorithm

While the previous section has presented the logic applied to determine the mapping action for a given pair of classes (one from the dataset and the other from the repository), we now concentrate on the algorithm responsible for obtaining such pairs.

Given a class C_D from the dataset ontology, its most similar class C_R in the repository ontology will be the one maximizing $avsim(C_D,C_R)$. However, since our algorithm does not only consider equivalence mappings, we are not interested in mapping C_D to its most similar class, but to the class that provides the most reliable mapping. This way, instead of considering *avsim* to find the best repository class to map to C_D, we consider the repository class C_R that maximizes the expression $max(sim(C_D, C_R), sim(C_R, C_D))$. In the case of several C_R with the same $max(sim(C_D, C_R), sim(C_R, C_D))$, we consider the one among them that maximizes $min(sim(C_D, C_R), sim(C_R, C_D))$. Note that the obtained

C_R is very often the same as the one obtained through *avsim*. But this assures that a class C_R that produces either a subclass or superclass mapping will be always preferred to one that produces a common superclass mapping.

To simplify the notation, we define the functions *maxsim* and *minsim* as follows:

$$maxsim(C_D, C_R) = max(\ sim(C_D, C_R), sim(C_R, C_D)\ )$$
$$minsim(C_D, C_R) = min(\ sim(C_D, C_R), sim(C_R, C_D)\ )$$

However, there may be repository classes that should not be considered for a mapping. The structure of both ontologies and the previous mappings determine the set of permitted candidates to be mapped to a particular dataset class C_D. Note that the third step of the semi-automatic method can be run after some values have already been manually mapped (see 5.2.2, and particularly Figure 23). Consequently, the new mappings have to be consistent with the mappings set by the expert. But even in the case where no manual mappings have been set yet, a suggested mapping for the class C_D will determine the candidates to be mapped to the superclasses and subclasses of C_D.

Let us consider the following example. The repository ontology contains the following classes and axioms, among others: *Urban area*, *Dense settlement*, *Dispersed settlement*, *Vegetation*; *Dense settlement* ⊑ *Urban area*, *Dispersed settlement* ⊑ *Urban area*. The dataset ontology contains the classes and axioms: *Vegetation*, *Dense forest*, *Urban area*; *Dense forest* ⊑ *Vegetation*. If the structure is not considered, *Vegetation* will be mapped to *Vegetation*, *Urban area* will be mapped to *Urban area*, and finally *Dense forest* will be mapped to *Dense settlement*. However, it is clear that *Dense forest* is not related to *Dense settlement*. This way, when *Vegetation* is mapped to *Vegetation*, a mechanism is needed for preventing *Dense forest*, as well as the other subclasses of *Vegetation*, to be mapped to *Dense settlement*.

Formalizing this example, we are mainly interested in how the mapping between C_D and C_R influences the possible mappings for their subclasses and superclasses. Let us first consider the case where C_D has been mapped as equivalent to C_R. It is clear that subclasses of C_D will also be subclasses of C_R, and that superclasses of C_D will also be superclasses of C_R. This way, we are interested in how the subclasses of C_D may be related to subclasses of C_R, since they provide two different ways of specializing the same concept. In the same way, we are also interested in how superclasses of C_D may be related to superclasses of C_R.

Nevertheless, since classes with a common superclass are not necessarily disjoint in Description Logic, it is important to note that a subclass of C_D can also be related to any class C_S in the repository, such that $C_S \sqcap C_R \neq \emptyset$, but may happen that $C_S \not\sqsubseteq C_R$. Note that in any case, any subclass of C_D will still be a subclass of C_R. Forcing subclasses of C_D to be only mapped to subclasses of C_R may cause that some relations with other classes, as C_S, are lost. However, our experience indicates that this is not a frequent case since datasets almost always contain disjoint values. On the other hand, if these restrictions are not considered, false positives usually appear, as in the example of *Dense forest* and *Dense settlement*, and worse results are obtained. Hence, we force our algorithm to find mappings for subclasses of C_D only among subclasses of C_R. And the same is applied to superclasses.

This can be also generalized to the other types of relations between C_D and C_R, not only to equivalence. In the case of C_D being either subclass or superclass of C_R, we know that C_D and C_R share some information and we want the algorithm to concentrate on how this information is specialized or generalized in both ontologies. Using similar arguments to avoid false positives as in the case of equivalence, the algorithm restricts subclasses and superclasses of C_D to be mapped to respectively subclasses and superclasses of C_R. A particular case has to be differentiated. If C_D is mapped as superclass of C_R, a mapping may also be suggested between a subclass of C_D and C_R. But instead if C_D is mapped as subclass of C_R, we are not interested in direct mappings among subclasses of C_D and C_R, since they are indirectly related through C_D. Particularly, if a subclass of C_D was mapped as superclass of C_R, then a cyclic definition would appear and C_R, C_D and its subclass would all be set as equivalents. Consequently, each subclass of C_D, C_{Ds}, will be restricted to be mapped to subclasses of C_R different from C_R itself ($C_{Ds} \sqsubset C_R$). Note that this avoids possible loops in the relations. Likewise, if C_D is superclass of C_R, then each superclass of C_D, C_{DS}, should be restricted to be mapped to superclasses of C_R different from C_R ($C_{DS} \sqsupset C_R$).

Finally, let us consider the case of C_D and C_R having a common superclass C_X. We assume that subclasses of C_D will only be mapped to subclasses of C_R, different from C_R itself. Regarding superclasses of C_D, we cannot determine their restrictions until C_X is not inserted into the repository by setting one or more superclasses for it. We will see below that the algorithm assures that when a common superclass mapping is suggested, all the equivalence, subclass and superclass mappings have already been determined. This means that in fact C_D and C_R already have a common superclass, C_Y, determined by a previous suggested mapping (otherwise, at least they share *QualitativeTheme* as superclass). This way, the new mapping simply provides a new class C_X that better represents what C_D and C_R have in common, and consequently C_X is a subclass of C_Y, and perhaps even equivalent to it. Adding the suggestion "C_X subclass of C_Y" provides new restrictions for superclasses of C_D being also subclasses of C_Y. These will be restricted to be mapped to repository classes being subclasses of C_Y and superclasses of C_R. Furthermore, we are also interested in determining if other subclasses of C_Y subsume C_X. However, a specific but common case should be distinguished here: when the dataset structure is flat and consequently the C_Y is too general (C_Y is either *QualitativeTheme* or it subsumes the class at the top of the dataset hierarchy), our experience indicates that C_X is very frequently subclass of the superclass(es) of C_R, since otherwise C_D would have been mapped to another class different from C_R. Consequently, in this case, we set C_X as direct subclass of the direct superclass(es) of C_R, and the superclass of C_D, if has not already been mapped, will be restricted to be mapped to superclass(es) of C_R.

We call *mapping restrictions* of a class (in the dataset ontology in our case) to this type of axioms that restrict the set of classes in the other ontology (the repository in our case) that can be mapped to the given class. For instance, when C_D is equivalent to C_R, the mapping restriction "subclass of C_R" is assigned to all the subclasses of C_D. Likewise, the mapping restriction "superclass of C_R" is assigned to the superclasses of C_D. Moreover, in the case of an equivalence, no other dataset class can be mapped as equivalent to C_R. Consequently, another mapping restriction "distinct from C_R" is assigned to all the other classes in the dataset.

The use of mapping restrictions is a significant improvement to other merging methods based on lexical similarities that either do not contemplate the structure or that consider it through path distance, depth or information context. The drawbacks of applying these approaches both in general and in our specific context have already been discussed in 5.1.3.

Considering the graph structure by means of mapping restrictions means a non-deterministic algorithm: the suggested list of mapping actions may be different depending on the order in which classes are processed, and consequently on the order in which mapping restrictions are generated. The best merging would be the one maximizing the sum of the *maxsim* measure for each pair of mapped classes. However, analyzing all the possible orders to obtain the best set of mapping actions has an exponential computational cost.

Instead, we propose a greedy algorithm that firstly processes those classes of the dataset ontology having more reliable suggested mappings. This way, the algorithm starts setting the mapping restrictions for the mappings that have been previously confirmed by the user. Note that these mappings are not based only on assumptions. Afterwards, the algorithm will prioritize the classes of the dataset ontology with the highest *maxsim*/*minsim* similarities.

The algorithm uses a list of mapping restrictions for each class in the dataset ontology. It also uses three sets of classes from the dataset ontology:

- The set of those classes for which the user has already confirmed its mapping (*ConfirmedSet*)
- The set of those classes that have already been processed and the algorithm has generated a suggested mapping action (*SuggestedSet*)
- The set of those classes that have not been processed yet (*NotProcessedSet*)

Note that these three sets are a partition of the set of all the classes in the dataset ontology, that is, they are mutually disjoint and their union is the complete set.

The algorithm is an iterative process that in each step

(1) takes the best candidate from the classes in the dataset ontology, according to its *maxsim* and *minsim* with the classes in the repository ontology satisfying the mapping restrictions;
(2) entails the mapping action for the selected class;
(3) if the relation is subclass, determines the corresponding superclass relation, if it is superclass determines the corresponding subclass relation, and if it is common superclass, determines the superclass of the new class; and
(4) generates the corresponding mapping restrictions.

The mapping algorithm is included here. To simplify the notation we use indistinctly the name of the class and the class itself. Each function (their name is written in bold and italic) will be discussed below.

```
//remove stop words
removeStopWords

//process mapping restrictions for the confirmed mappings
for each class $CC \in ConfirmedSet$,
              $CC$ mapped to $CM$ through relation type $Rel$ do
    addMappingRestrictions($CC$, $CM$, $Rel$)
end for

//process values in $NotProcessedSet$
while $NotProcessedSet \neq \emptyset$ do
    for each class $CD_i \in NotProcessedSet$ do
        let $RSet_i$ = satisfyingRestrictions($CD_i$)
        let $CRmax_i = CR_j$ such that
            $CR_j \in RSet_i$ and
            $maxsim(CD_i,CR_j) \geq maxsim(CD_i,CR_k), \forall k$ and
(1)         $(\exists p\ maxsim(CD_i,CR_j)= maxsim(CD_i,CR_p) \Rightarrow$
                $minsim(CD_i,CR_j) \geq minsim(CD_i,CR_p)\ )$
    end for
    let $CDmax = CD_i$ , $CRmax = CRmax_i$ such that
        $maxsim(CD_i,CRmax_i) \geq maxsim(CD_j,CRmax_j), \forall j$ and
        $(\exists k\ maxsim(CD_i,CRmax_i)= maxsim(CD_k,CRmax_k) \Rightarrow$
            $minsim(CD_i,CRmax_i) \geq minsim(CD_k,CRmax_k)\ )$
    if $maxsim(CDmax,CRmax) > \mu$ then
(2)     $typeOfRelation$ = suggestMapping($CDmax$,$CRmax$)
        if $typeOfRelation \neq$ equivalence then
(3)        processRelatedMappings($CDmax$,$CRmax$,$typeOfRelation$)
        end if
(4)     addMappingRestrictions($CDmax$, $CRmax$, $typeOfRelation$)
        $NotProcessedSet = NotProcessedSet \setminus \{ CDmax \}$
        $SuggestedSet = SuggestedSet \cup \{ CDmax \}$
    else
        deepCopy( $NotProcessedSet$ )
    end if
end while
```

It is important to clarify that for efficiency reasons, the similarities with repository classes for each class C_{Di} are not computed in each iteration. Instead, this information is computed before the main loop starts, and it is stored in a three dimensional matrix. The first dimension of the matrix corresponds to the classes in the dataset, the second one to the classes in the repository, while the third one refers to one of the two partial *sim* measures. Likewise, the matrix also represents those repository classes that satisfy the mapping restrictions for each C_{Di}, $RSet_i$: the repository classes that do not belong to $RSet_i$ contain a –1 in the corresponding cells. The set $RSet_i$ is not recomputed each iteration either, but only when a new mapping restriction is added to C_{Di}. Note that in the worst case, this may happen at most $n\cdot(n+1)/2$ times, where n is the number of classes in the dataset ontology. This worst case would occur in the situation of the dataset values being organized in only one branch (value i is child of value $i+1$), since each new mapping makes all the other values to change their mapping restrictions. Note also that obviously no real dataset has this organization. Finally, the data structure also maintains the best candidate to be mapped to each C_{Di} at every moment (the one maximizing *maxsim*/*minsim*), that again is only modified when new mapping restrictions are added to C_{Di}.

It has also to be noted that when maximums are calculated in the algorithm, there may be more than one class with the same maximum value. For instance in

```
let $CRmax_i = CR_j$ such that
      $CR_j \in RSet_i$ and $maxsim(CD_i,CR_j) \geq maxsim(CD_i,CR_k), \forall k$ and
      $(\exists p\ maxsim(CD_i,CR_j)=maxsim(CD_i,CR_p) \Rightarrow$
                        $minsim(CD_i,CR_j) \geq minsim(CD_i,CR_p)\ )$
```

there may be two repository classes C_{Rx} and C_{Ry} that satisfy the condition for C_{Di}. In these situations, the class that has been firstly processed will be selected.

The function ***removeStopWords*** removes stop words from all the class names in both ontologies. We recall that stop words are those terms with a void meaning, like articles, determiners, prepositions and conjunctions. However, some of them are not removed yet, since they are necessary for detecting aggregations, mixtures or negations, like “and”, “or”, “with”, “without” or “no” (see Section 6.3). As it was already mentioned above, substantives referring to spatial units (area, region, land or zone, among others) are also removed in this process. Likewise, morphemes that correspond to these substantives, like “land” in “grassland”, are removed too. Finally, some usual endings like –ing, –ed, or –s (from plural) are also eliminated.

The function ***addMappingRestrictions***, given a class C_D in the dataset ontology and its mapped class C_R in the repository ontology, adds the mapping restrictions to the subclasses and superclasses of C_D:

```
function addMappingRestrictions (  $CD \in$ Dataset Ontology,
            $CR \in$ Repository Ontology, typeOfRelation $\in$ {subclass,
                  superclass, equivalence, commonSuperclass} )
      if typeOfRelation = equivalence then
            for each $CDs \in$ NotProcessedSet such that $CDs \sqsubset CD$ do
                  add mapping restriction "$\sqsubset CR$" to $CDs$
            end for
            for each $CDS \in$ NotProcessedSet such that $CD \sqsubset CDS$ do
                  add mapping restriction "$\sqsupset CR$" to $CDS$
            end for
            for each $CX \in$ NotProcessedSet such that $CX \not\equiv CD$ do
                  add mapping restriction "$\not\equiv CR$" to $CX$
            end for
      else if typeOfRelation = subclass then
            for each $CDs \in$ NotProcessedSet such that $CDs \sqsubset CD$ do
                  add mapping restriction "$\sqsubset CR$" to $CDs$
            end for
            for each $CDS \in$ NotProcessedSet such that $CD \sqsubset CDS$ do
                  add mapping restriction "$\sqsupseteq CR$" to $CDS$
            for each $CDS \in$ NotProcessedSet such that $CD \sqsubset CDS$ do
      else if typeOfRelation = superclass then
            for each $CDs \in$ NotProcessedSet such that $CDs \sqsubset CD$ do
                  add mapping restriction "$\sqsubseteq CR$" to $CDs$
            end for
            for each $CDS \in$ NotProcessedSet such that $CD \sqsubset CDS$ do
                  add mapping restriction "$\sqsupset CR$" to $CDS$
            end for
```

```
         else if typeOfRelation = commonSuperclass then
              for each CDs ∈ NotProcessedSet such that CDs ⊏ CD do
                   add mapping restriction "⊏ CR" to CDs
              end for
         end if  end if  end if  end if
    end function
```

The function ***satisfyingRestrictions*** has as parameter a class in the dataset ontology, and returns the set of classes in the repository ontology such that satisfy the mapping restrictions of the class. Note that in our implementation using the matrix with *sim*'s, this functions consists of returning those classes in the matrix that do not contain the value -1.

The function ***suggestMapping*** suggests the connection of a class of the dataset ontology to a class of the repository ontology, according to the rules discussed in Section 6.1. It returns the type of relation: subclass, superclass, equivalence or common superclass. The function is defined as follows:

```
    function suggestMapping ( CD ∈ Dataset Ontology,
                              CR ∈ Repository Ontology )
         returns {subclass,superclass,equivalence,commonSuperclass}
      if $sim(CD,CR) \geq \lambda$ and $sim(CR,CD) \geq \lambda$ then      // $CD \equiv CR$
        create suggestion "CD equivalent to CR"
        return equivalent
      else
        if $sim(CD,CR) \geq \lambda$ and $sim(CR,CD) \leq \lambda$ then     // $CD \sqsupseteq CR$
          create suggestion "CD superclass of CR"
          return superclass
        else
          if $sim(CD,CR) \leq \lambda$ and $sim(CR,CD) \geq \lambda$ then   // $CD \sqsubseteq CR$
            create suggestion "CD subclass of CR"
            return subclass
          else
            //common superclass ($sim(CD,CR) \geq \mu$ and $sim(CR,CD) \geq \mu$)
              let SX be the common part of the names of CD and CR
              create suggestion "CD and CR have common superclass SX"
              return commonSuperclass
          end if
        end if
      end if
    end function
```

It is important to note that some suggested mappings may be subsumed by anothers. For instance, the algorithm may suggest "C_D subclass of C_{R1}" and "C_D subclass of C_{R2}" where C_{R1} is subclass of C_{R2}. Although this is not incorrect, it may result confusing to the user. This way, before a new suggestion is added, although for simplicity is not explicitly written in the algorithm, it has to be checked whether the new suggestion is subsumed or subsumes a previous one.

The function ***processRelatedMappings*** carries out the third step of the algorithm when the relation of the suggested mapping is not an equivalence. If C_D is mapped as subclass of C_R, it determines whether C_D can be mapped as superclass of any subclass of C_R. Equivalently, if C_D is mapped as superclass of C_R, it determines whether C_D can be mapped as subclass of any superclass of C_R. Finally, if the mapping relation is “common superclass”, it determines a superclass for the resulting new class. Note that in any case, the mapping restrictions of C_D have to be preserved. *processRelatedMappings* uses function ***suggestMappingWithRestriction***, which returns a Boolean indicating whether two classes can be mapped through a certain type of relation, and if possible, maps them. They are defined in the following way:

```
function processRelatedMappings( CD $\in$ Dataset Ontology,
            CR $\in$ Repository Ontology , RSet $\subseteq$ Repository Ontology
            toR $\in$ {subclass, superclass, commonSuperclass} )
  if toR=subclass then
    for each $CRs \in RSet$ such that $CRs \sqsubseteq CR$ do
      if suggestMappingWithRestriction(CD,CRs,superclass) then
        addMappingRestrictions(CD,CRs,superclass)
    end for
  else
    if toR=superclass then
      for each $CRS \in RSet$ such that $CRS \sqsupseteq CR$ do
        if suggestMappingWithRestriction(CD,CRS,subclass) then
          addMappingRestrictions(CD,CRS,superclass)
      end for
    else //common superclass
      let SX be the common part of the names of CD and CR
      let CY be the class such that $CR \sqsubseteq CY$ and $CD \sqsubseteq CY$
      let CDtop be the top class in the dataset ontology
      if $CDtop \sqsubseteq CY$ then //CX subclass of superclasses of CR
        for each CRS such that $CR \sqsubseteq CRS \sqsubseteq CY$ and
            (CRS2 such that $CR \sqsubseteq CRS \sqsubseteq CY \Rightarrow CRS2 \not\sqsubseteq CRS$) do
          create suggestion "SX subclass of CRS"
          add mapping restriction "$\sqsupset$ CR" to CDs
        end for
      else //find superclass of CX
        if $\exists CZ$ such that $CR \sqsubseteq CZ \sqsubseteq CY$ or $CD \sqsubseteq CZ \sqsubseteq CY$ and $sim(SX,CZ) \geq \lambda$
            and (CA satisfying the same condition
                          $\Rightarrow sim(SX,CZ) \geq sim(SX,CZ)$) then
          if $sim(CZ,SX) \geq \lambda$ then //SX already exists: CZ
            remove suggestion
                  "CD and CR have common superclass SX"
          else
            create suggestion "SX subclass of CZ"
            for each CDS such that $CD \sqsubseteq CDS \sqsubseteq CY$ do
              add mapping restriction "$\sqsupset$ CR" to CDs
            end for
          end if
        else
          create suggestion "SX subclass of CY"
          for each CDS such that $CD \sqsubseteq CDS \sqsubseteq CY$ do
            add mapping restriction "$\sqsupset$ CR" to CDs
          end for
        end if
      end if
    end if
  end if
end function
```

```
function suggestMappingWithRestriction ( CD ∈ Dataset Ontology,
            CR ∈ Repository Ontology ,
            toR ∈ {subclass, superclass} )
      if toR=subclass then
            if sim(CD,CR) ≤ λ and sim(CR,CD)≥λ then
                  create suggestion "CD subclass of CR"
                  return true
            end if
      else
            if toR=superclass then
                  if sim(CD,CR)≥ λ and sim(CR,CD) ≤ λ then
                        create suggestion "CD superclass of CR"
                        return true
                  end if
            end if
      end if
      return false
end function
```

The function ***deepCopy*** has as parameter a set of classes of the dataset ontology for which the algorithm could not find lexical similarities. For each class C_D in the set, the function suggests the mapping action of adding C_D to the repository ontology, while its axioms of subclass in the dataset ontology are also inserted in the repository.

```
function deepCopy (CDSet ⊆ Dataset Ontology)
      for each CD∈ CDSet do
            for each CDS such that CD ⊑ CDS do
                  create suggestion "CD subclass of CDS"
            end for
            CDSet = CDSet \ {CD}
      end for
end function
```

Note that the functions *suggestMapping* and *deepCopy* only suggest connections, but do not execute them. If the user accepts a mapping, the following actions have to be executed, depending on the type of relation. Note also that when the mapping is executed, the relations among classes in the dataset have to be kept in the repository ontology (see function ***keepDsRelations*** below). To simplify the notation, we identify with the same symbol, for instance C_D, both the name of the class and the class itself.

```
"CD equivalent to CR":
    add connexion between CD and CR
    keepDsRelations(CD)

"CD subclass of CR":
      add class newCD
      add axiom "newCD ⊑ CR"
      add connexion between CD and newCD
      keepDsRelations(CD)

"CD superclass of CR":
      add class newCD
      add axiom "newCD ⊒ CR"
      add connexion between CD and newCD
      keepDsRelations(CD)
```

```
"CD and CR have a common superclass CX":
      add classes CX and newCD
      let CY be the superclass of CX
            //CY determined in processRelatedMappings
      add axiom "$CX \sqsubseteq CY$"
      add axioms "$newCD \sqsubseteq CX$" and "$CR \sqsubseteq CX$"
      add connexion between CD and newCD
      keepDsRelations(CD)
```

The function ***keepDsRelations*** checks if the class being inserted is related to other classes in the dataset that have previously been inserted in the repository:

```
function keepDsRelations ($CD \in$ Dataset Ontology)
      for each CDs in the dataset ontology such that $CDs \sqsubseteq CD$ and
        CDs is in the repository ontology do
            add axiom "$CDs \sqsubseteq CD$"
      end if
      for each CDS in the dataset ontology such that $CD \sqsubseteq CDS$ and
        CDS is in the repository ontology do
            add axiom "$CD \sqsubseteq CDS$"
      end if
end function
```

6.2.1 Structural rules

We have seen how structure and previous mappings influence new mappings through the mechanism of mapping restrictions. We introduce now what we call *structural rules*, that enable the algorithm to entail new mappings also according to the structure of the ontologies and previous mappings. They are based on the assumption that two classes from two ontologies that have equal contexts probably refer to the same concept. Let us consider an example where the dataset class C_D has a unique superclass C_{D1} and a unique subclass C_{D2}, while the repository class C_R has a unique superclass C_{R1} and a unique subclass C_{R2}. Let us also consider that the similarity between C_D and C_R is smaller than μ, and that the algorithm has already determined that C_{D1} is equivalent to C_{R1} and C_{D2} is equivalent to C_{R2}.

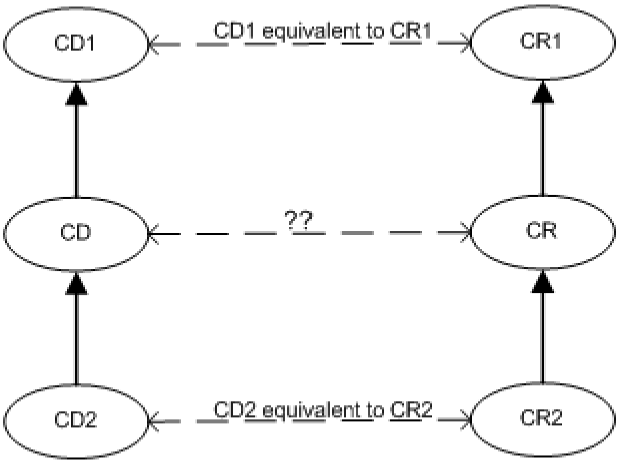


Figure 29. Example of structural rule

In this situation it is very likely that C_D will be equivalent to C_R, although their names are different. This assumption is based on a heuristics, since they may not be necessarily equivalent. This is an example of what we call a structural rule, a heuristics-based rule that determines how new mappings can be inferred from previous ones when no lexical similarities exist. Note that while mapping restrictions avoid new mappings to be inconsistent with the structure and previous mappings, structural rules infer new mappings from them both.

We can generalize this example to any number of subclasses of C_D and C_R, where each subclass of C_D has been mapped as equivalent to a different subclass of C_R. We call this example of structural rule “common parent and children”.

Another example of structural rule determines a new equivalence mapping when two classes from two ontologies have the same parent and “brothers”, as we can see in Figure 30.

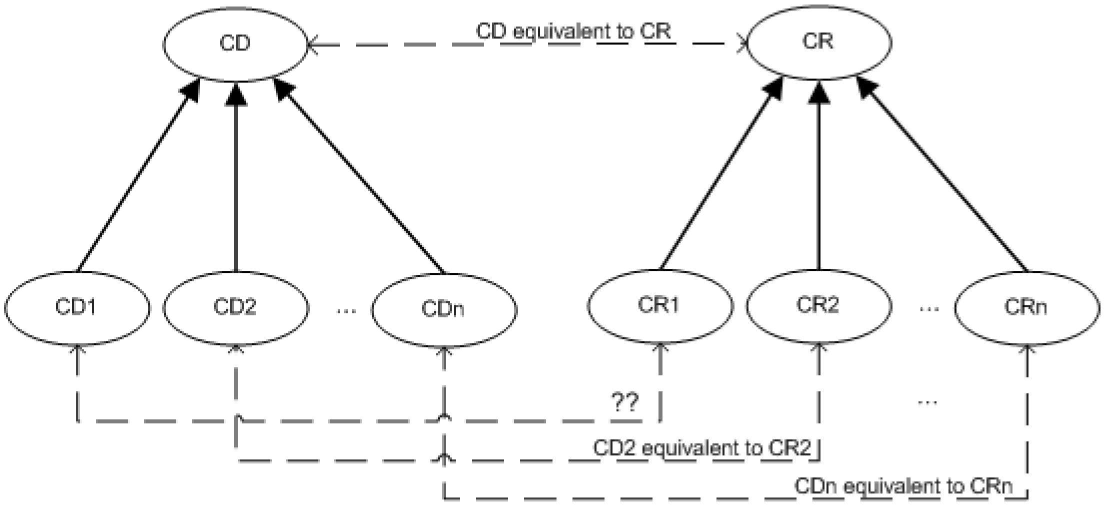


Figure 30. Example of structural rule “common parent and brothers”

In this case C_{D1} has exactly one class C_{R1} satisfying its mapping restrictions, and the only superclass of C_{D1} has been set as equivalent to the only superclass of C_{R1}, while all the “brothers” of C_{D1} and C_{R1} have also been related as equivalent between them. In this case, we also apply a structural rule based on the assumption that C_{D1} will probably be equivalent to C_{R1}. We call this structural rule “common parent and brothers”.

In general, structural rules can only be applied when one class from the dataset has exactly one class from the repository satisfying its mapping restrictions but no relation has been found between them (their *maxsim* is smaller than or equal to μ). The function *deepCopy* in the algorithm has to be replaced by ***checkStructuralRulesAndDeepCopy***, which for each dataset class in *NotProcessedSet* checks whether a structural rule can be applied for it. Since inferred mappings may influence other structural rules to be applied, this is an iterative process that finishes when no more inferred mappings are obtained. At that moment, all the classes that still belong to *NotProcessedSet* are copied to the repository through the function *deepCopy*.

```
function checkStructuralRulesAndDeepCopy($CDSet \subseteq DatasetOntology$)
  repeat
    let ruleFound be a Boolean variable initilized to false
    for each $CD \in CDSet$ do
      let RSet = satisfyingRestrictions(CD)
                  //RSet was previously calculated
      if $|RSet|=1$ then
        let CDS be the unique direct superclass of CD if exists
        let CR be the class in RSet
        let CRS be the unique direct superclass of CR if exists
        if there is a suggestion "CDS equivalent to CRS" then
          //check "common parent and children" structural rule
          if $\forall$CDi direct subclass of CD, $i=1,...,k$
                there is a suggestion "CDi equivalent to CRj",
                for some $j \in \{1,...,k\}$ where CRj is direct subclass of CR
            then
            create suggestion "CD equivalent to CR"
            $CDSet = CDSet \setminus \{CD\}$
            ruleFound = true
          else
            //check "common parent and brothers" structural rule
            if $\forall$CDi direct subclass of CDS, $i=1,...,k$, $CDi \neq CD$,
                there is a suggestion "CDi equivalent to CRj",
                for some $j \in \{1,...,k\}$ where CRj is direct subclass
                of CRS, $CRj \neq CR$ then
              create suggestion "CD equivalent to CR"
              $CDSet = CDSet \setminus \{CD\}$
              ruleFound = true
            end if
          end if
        end if
      end if
    end for
  until ruleFound=false
  deepCopy(CDSet)
end function
```

In our implementation we have only considered these two structural rules, but others could be identified and added to the function. A flexible way to add new structural rules would be useful.

In practical terms, the main utility of structural rules is that they enable the algorithm to deal with possible spelling mistakes in the names of classes. They also enable the algorithm to detect equivalences when different names are used for the same class.

6.3 Some significant special cases

We discuss in this section some special cases that require a slight modification of the general algorithm. These are relations between a dataset class and several repository classes (6.3.1), and complex definitions of a dataset class, including aggregations (6.3.2), mixtures (6.3.3) or negations of elements (6.3.4). The final algorithm that covers all these cases is included in Section 6.4.

6.3.1 One-to-many relations

The previous algorithm may connect one repository class to several dataset values. But given a dataset class C_D, it finds at most one repository class C_R to be mapped to (apart from the mappings obtained in *processRelatedMappings*). However, there is a particular and frequent case where a value may need to be mapped to more than one class. Let us consider an example where a dataset contains the value *Wetland*, while in the repository ontology two classes *Woody wetland* and *Herbaceous wetland* exist. The previous algorithm would map *Wetland* to one of the two classes, would remove it from *NotProcessedSet*, and would not be mapped to the other. It is worth mentioning that both *Woody wetland* and *Herbaceous wetland* have the same similarity measure with *Wetland*, and the algorithm would select the one processed firstly.

While in this example the relation was subclass, superclass relation could also be considered, as in the following example: dataset class *Urban forest* should be connected as subclass of the repository classes *Urban land* and *Forest*. But note that all the relations should be of the same type. On the other hand, this type of 1-to-many relations are not considered for equivalence and common superclass relations. In the case of equivalence relations, if a dataset value is set as equivalent to two repository classes, these become also equivalent. Consequently, this should only be done if the two repository classes are already equivalent, and in such case, adding the second mapping does not provide new information. Finally, our experience indicates that considering many “common superclass” relations typically causes that the value is mapped to repository classes that are really not semantically related. As an example, let us consider the repository classes *Dense forest* and *Dispersed settlement*, and the dataset class *Dense settlement*. Note that *Dense settlement* should not be related to *Dense forest*, since they do not have nothing in common. We will see that the terminological approach can distinguish that in *Dense settlement* the substantive “settlement” should be prioritized with respect to the adjective “dense”.

Generalizing these ideas, we are interested in the case where a class from the dataset ontology C_D can be mapped to several classes from the repository, $C_{R1},\ldots, C_{Rn}$, such that the type of relation is either always superclass ($C_D \sqsupseteq C_{Ri}$, $i=1,\ldots,n$), or always subclass ($C_D \sqsubseteq C_{Ri}$, $i=1,\ldots,n$). In the case of superclass relation, if the class C_R with the highest $maxsim(C_D,C_R)$ satisfies $sim(C_D,C_R) \geq\lambda$ and $sim(C_R,C_D)<\lambda$, then all the other classes C_{Ri} from the repository satisfying $sim(C_D,C_{Ri}) \geq\lambda$, and consequently also $sim(C_{Ri},C_D)<\lambda$, will also be mapped as subclasses of C_D. In the case of subclass relation, if the class C_R with the highest $maxsim(C_D,C_R)$ satisfies $sim(C_D,C_R)<\lambda$ and $sim(C_R,C_D) \geq\lambda$, then all the other classes C_{Ri} from the repository satisfying $sim(C_{Ri},C_D) \geq\lambda$, and consequently $sim(C_D,C_{Ri}) <\lambda$, will also be mapped as superclasses of C_D.

To that purpose, the following part of the original algorithm that determines the best class $CRmax_i$ from the repository to be mapped to the dataset class CD_i

```
for each class $CD_i \in NotProcessedSet$ do
      let $RSet_i = satisfyingRestrictions(CD_i)$
      let $CRmax_i = CR_j$ such that $CR_j \in RSet_i$ and
            $maxsim(CD_i,CR_j) \geq maxsim(CD_i,CR_k)$, $\forall k$ and
            $(\exists p\ maxsim(CD_i,CR_j)= maxsim(CD_i,CR_p) \Rightarrow$
                  $minsim(CD_i,CR_j) \geq minsim(CD_i,CR_p)\ )$
end for
```

has to be replaced, in order to deal now with a set of classes *CRmaxSet*$_i$, by:

```
for each class $CD_i \in NotProcessedSet$ do
      let $RSet_i = satisfyingRestrictions(CD_i)$
      let $CRmax_i = CR_j$ such that $CR_j \in RSet_i$ and
            $maxsim(CD_i,CR_j) \geq maxsim(CD_i,CR_k)$, $\forall k$ and
            $(\exists p\ maxsim(CD_i,CR_j)= maxsim(CD_i,CR_p) \Rightarrow$
                  $minsim(CD_i,CR_j) \geq minsim(CD_i,CR_p)\ )$
      if $sim(CD_i,CRmax_i) \geq \lambda$ and $sim(CRmax_i, CD_i) < \lambda$ then
            let $CRmaxSet_i = \{CR_k \in RSet_i \mid \mathrm{sim}(CD_i,CR_k) \geq \lambda\}$
      else
            if $\mathrm{sim}(CD_i,CR_j) < \lambda$ and $sim(CR_j, CD_i) \geq \lambda$ then
                  let $CRmaxSet_i = \{CR_k \in RSet_i \mid sim(CR_k, CD_i) \geq \lambda\}$
            else
                  let $CRmaxSet_i = \{CR_j\}$
            end if
      end if
end for
```

The modified general algorithm, including these modifications and the remaining special cases is presented in Section 6.4.

6.3.2 Aggregations of values

Thematic datasets often present aggregations of concepts in a single value. This is especially more frequent at small scales (continental or world-wide) datasets. For instance, the land cover map of the US Geology Survey (USGS) has values like "*Orchards, Groves, Vineyards, Nurseries, and Ornamental Horticultural Areas*", the CORINE land cover map has values like "*Industrial, commercial and transport units*" or "*Mine, dump and construction sites*", or the Simple Biosphere Model (SBM) has "*Agriculture or Grassland*". It has to be noted that the conjunction *and* in these cases does not indicate a logical *and* operator. On the contrary, it almost always refers to a logical *or* operator. For example, an area will be considered as "*Mine, dump and construction sites*" when the area is any (but of course not all) of the atomic values "*mine*", "*dump*" or "*construction sites*". We will see in the terminological approach that the use of a lexical base like WordNet, which indicates the lexical category (noun, adjective, verb or adverb) of a term, provides a better separation of the atomic values being aggregated. For instance, a class with name "*broadleaf and needleleaf trees*" is split in "*broadleaf*" and "*needleleaf trees*" in the string-based approach, but would obtain "*broadleaf trees*" and "*needleleaf trees*" using WordNet.

This way, whenever the algorithm finds a value containing an aggregation by means of *and*, *or*, *and/or*, commas or slashes, it is transformed in a hierarchy with a physical value and several abstract values linked through the property *valueChildOf*. From a conceptual point of view, the class is represented as the union of the classes that are part of the aggregation. Following the example above, the class "*Mine, dump and construction sites*" would be transformed in the structure composed by the new classes "*mine*", "*dump*" and "*construction sites*", with the axiom indicating that "*Mine, dump and construction sites*" is equivalent to the union of these three new classes.

Let us now discuss another example that will introduce other aspect of our heuristics. When the previous algorithm is used to merge USGS and the Biosphere Atmosphere Transfer (BAT) datasets, a relation is found between the values "*Dryland Cropland and Pasture*" from USGS and "*Crops, Mixed Farming*" from BAT. Note that, after removing stop words and morphemes, the USGS value is split in "*dry crop*" and "*pasture*", while the BAT value is split in "*crop*" and "*mix farm*", and that there is a subclass-of mapping action between "*dry crop*" and "*crop*". However, in principle, this mapping action does not indicate that "*Dryland Cropland and Pasture*" is a subclass of "*Crops, Mixed Farming*", since pasture is not a subclass of either "*crop*" or "*mix farm*". But in these cases, we apply the following heuristics: if "*pasture*" is not related to any value of the other dataset, it is also considered related to "*crop*". Note that this is based on the assumption that the values that form an aggregation are usually strongly related among them. Consequently, the class "*Dryland Cropland and Pasture*" would be mapped as a subclass of "*Crops, Mixed Farming*". Again, we will see in the terminological approach that this heuristics is refined with the information extracted from the terminological base (see 7.3.1).

The structure-based algorithm will also be slightly modified. Firstly, aggregations among the classes in the set *NotProcessedSet* have to be found and split, obtaining the new aggregating classes (see *findAndSplitAggregation* function below). After this, the set *NotProcessedSet* will only contain those values that are not in any aggregation, while two new sets *AggregationsSet* and *AggregatingSet* will contain respectively aggregations and aggregating classes. Now classes in *NotProcessedSet* are processed, and once the set becomes empty, classes in *AggregatingSet* are also processed in the usual way. Finally, the mapping for the aggregation classes are obtained from the mapping actions of their aggregating classes (see *mappingAggregation* function below). The final algorithm, which also incorporates other special cases, will be shown in Section 6.4.

This procedure has an exception: if there is a class in the repository with the same name of the aggregation class, it is not necessary to split it in aggregating classes, since the equivalence will be found.

We now define functions *findAndSplitAggregation* and *mappingAggregation* in order to formalize these ideas.

findAndSplitAggregation function determines if a class C_D, with name S_D, in the dataset ontology (in the set *NotProcessedSet*) is an aggregation and if so, splits it. It is defined in the following way:

```
function $findAndSplitAggregation( CD )$
      if $SD = SD_1$ $conj$ $SD_2$ ... $conj$ $SD_n$
        where $conj \in$ {"and", "or", "and/or", ",", "/"},
        $SD_1,...,SD_n$ are strings (with one or more words) and
        $\nexists CR$ with name $SR$ in the repository such that
                    ($sim(SD,SR) \geq \lambda$ and $sim(SR,SD) \geq \lambda$) then
            add new classes $CD_1,...,CD_n$ with respective names
            $SD_1,...,SD_n$
            add new axiom: $CD \equiv$ union of($CD_1,...,CD_n$)

            $NotProcessedSet = NotProcessedSet \setminus \{CD\}$
            $AggregationsSet = AggregationsSet \cup \{CD\}$
            $AggregatingSet = AggregatingSet \cup \{CD_1,...,CD_n\}$
      end if
end function
```

mappingAggregation function tries to entail a relation for an aggregation class C_D from the mapping actions obtained for its aggregating classes, $C_{D1},...,C_{Dn}$. It is defined as follows, where C_R and C_S are classes of the repository:

```
function $mappingAggregation( CD )$
   if $\exists CR$ such that $CR \sqsubseteq CD_i$, $i$=1,...,$n$ and
          $\forall C_S$ such that $CS \sqsubseteq CD_i$, $i$=1,...,$n$, $CS \sqsubseteq CR$ then
      create suggestion "$CD$ superclass of $CR$"
  else
    if $\exists CR$ such that $CR \sqsupseteq CD_i$, $i$=1,...,$n$ and
          $\forall CS$ such that $CS \sqsupseteq CD_i$, $i$=1,...,$n$, $CS \sqsupseteq CR$ then
      create suggestion "$CD$ subclass of $CR$"
    else
      if $\exists CR$ and $\exists i \in \{1,...,n\}$ such that $CD_i \sqsubseteq CR$ and
          $\forall CD_j$, $j \neq i$, ($CD_j \sqsubseteq CS \Rightarrow CS \sqsubseteq CR$) then
        create suggestion "$CD$ subclass of $CR$"
      end if
    end if
  end if
end function
```

Note that in the case of an equivalence between an aggregating class C_{Di} and a class C_R in the repository, C_R is also a subclass of C_{Di}, and consequently the suggestion "C_R subclass of C_D" will be added.

6.3.3 Mixtures of values

Another typical special case can be identified when a single value in a dataset combines different elements, like *Cropland/Grassland Mosaic* in USGS. This value requires that both cropland and grassland coexist in the same land region. Note that this does not exactly correspond to a logical intersection of classes, since this land region cannot strictly be considered as either cropland or grassland, but simply as a mixture of them. This fact drives us to force that this value can only be mapped to a mixture thematic class that combines both *cropland* and *grassland*. Mixtures of qualitative themes and how they are modelled was discussed in 4.2.5.

Apart from *mosaic*, other terms as *mixing* or *association* and their derivatives also suggest mixtures of elements, as for instance in *Annual and permanent crops association*. Complements with the word *with*, as in *Broadleaf Shrubs with Bare Soil*, also imply that both elements are needed.

An exception can be considered in values with a name of the form like *Shrub and/or herbaceous vegetation association* in CORINE. In this case, the use of *and/or* (also *or* would have the same effect) does explicitly not force both elements to coexist, and consequently, *shrub* or *herbaceous vegetation* can be considered as its subclasses. In fact, CORINE includes subclasses like *Natural grassland* that clearly only refer to one of them (*herbaceous vegetation* in this case).

As in the aggregation case, the structure-based algorithm has to be modified to deal with mixtures. A process (see *findAndSplitMixture* function bellow) determines if a class is a mixture of different elements and extracts them (referred to as *mixing classes* from now on). Two new sets of classes are defined, *MixturesSet* and *MixingSet*, the former for containing the mixture classes and the latter for the mixing ones. The classes in *MixingSet* are processed after the ones in *AggregatingSet*. Finally, the mappings for the mixture classes are obtained from the mapping actions of their respective mixing classes (see *mappingMixture* function below). As in the case of aggregations, mixtures are not considered if there exists a class in the repository with the same name. The algorithm with all the special cases is presented in Section 6.4.

We now define the functions *findAndSplitMixture* and *mappingMixture*, that will formalize all the intuitions discussed above.

findAndSplitMixture function determines if a class C_D, with name S_D, in the dataset ontology (in the set *NotProcessedSet*) is a mixture of classes $C_{D1},...,C_{Dn}$. It is defined as follows:

```
function findAndSplitMixture ( CD )
  if SD =   $SD_1$ conj $SD_2$ ... conj $SD_n$ mix |
            mix [of] $SD_1$ conj $SD_2$ ... conj $SD_n$ |
            $SD_1$ with $SD_2$
              where conj $\in$ {"and", ",", "/"},
              mix$\in$ {"mosaic","association","mixing","mixture"},
              $SD_1$,...,$SD_n$ are strings (with one or more words) and
            $\nexists$CR with name SR in the repository such that
                       (sim(SD,SR) $\geq\lambda$ and sim(SR,SD) $\geq\lambda$) then
      add new classes $CD_1$,...,$CD_n$ with names $SD_1$,...,$SD_n$
      add new axiom: CD $\equiv$ mixture of($CD_1$,...,$CD_n$) (*)

      NotProcessedSet = NotProcessedSet \ {CD}
      MixturesSet = MixturesSet $\cup$ {CD}
      MixingSet = MixingSet $\cup$ {$CD_1$,...,$CD_n$}
  end if
end function
```

(*) this is modelled through "ideal" individuals of qualitative classes and the property qualitativeThemeMixOf as explained in Chapter 4.

mappingMixture function tries to find relations for a given mixture class C_D, from the mapping actions obtained for its mixing classes C_{D1} ,..., C_{Dn}. It is defined as follows:

```
function mappingMixture ( CD )
  if $\exists CR$ in the repository ontology such that
                $CR \sqsubseteq$ QualitativeMixTheme and
                $\forall CR_M$ such that $CR.qualitativeThemeMixOf = CR_M$ ,
                     $CR_M \sqsubseteq CD_i$, for $i \in \{1,...,n\}$ then
      create suggestion "CD superclass of CR"
  end if
  if $\exists CR$ in the repository ontology such that
                $CR \sqsubseteq QualitativeMixTheme$ and
                $\forall CR_M$ such that $CR.qualitativeThemeMixOf = CR_M$ ,
                     $CR_M \sqsupseteq CD_i$, for $i \in \{1,...,n\}$ then
      create suggestion "CD subclass of CR"
  end if
end function
```

Note that if all the mixing classes in C_D had an equivalent mixing class in C_R, then C_D and C_R would be set as equivalent, since both axioms "C_D subclass of C_R" and "C_D superclass of C_R" would have been suggested. However, this will never happen since in that case, $sim(S_D,S_R)$ and $sim(S_R,S_D)$ would be both greater than λ. Nevertheless, this function will also be used for the terminological approach, where an equivalence between mixtures do not mean that they have exactly the same name.

Finally, a small modification is needed in the function *findAndSplitAggregation* to deal with mosaics with "and/or" or "or" conjunctions, since they are considered as aggregations. The conditional:

```
if $SD = SD_1$ conj $SD_2$ ... conj $SD_n$
            where conj $\in$ {"and", "or", "and/or", ",", "/"} and
            $SD_1,...,SD_n$ are strings (with one or more words) and
            $\nexists CR$ with name SR in the repository such that
                       ($sim(SD,SR) \geq \lambda$ and $sim(SR,SD) \geq \lambda$) then
```

has to be replaced by:

```
if $SD = SD_1$ conj1 $SD_2$ ... conj1 $SD_n$ |
      $SD_1$ conj2 $SD_2$ ... conj2 $SD_n$ mix |
      mix [of] $SD_1$ conj2 $SD_2$ ... conj2 $SD_n$
            where conj1 $\in$ {"and", "or", "and/or", ",", "/"} and
            conj2 $\in$ {"or", "and/or", ","} and
            $SD_1,...,SD_n$ are strings (with one or more words) and
            $\nexists CR$ with name SR in the repository such that
                       ($sim(SD,SR) \geq \lambda$ and $sim(SR,SD) \geq \lambda$) then
```

6.3.4 Negations of values

Finally, some names of dataset values include negative modifiers. For instance, let us consider the dataset class *Non dense forest*, and the repository class *Dense forest*. Our original algorithm would determine that that *Non dense forest* should be mapped as subclass of *Dense forest*, which is obviously incorrect. To avoid this, we include a very simple treatment of negations that is based on discarding the similarities for the negated element. In the example "*Non dense*" (the negative particle and its following term) is not considered and is replaced by a random string starting by the substring "negation". This string avoids *Non dense forest* to be mapped as superclass of *Dense forest*. This way, the algorithm will map *Non dense forest* as having a common superclass with *Dense forest*.

Although this is not a complete solution for the problem, it resolves the most usual case at a very low cost. On the other hand, note that a solution closer to Description Logic would add a new dataset class with the negation of value and an axiom stating that they are disjoint. In the example a new class *Dense forest* would be added to the dataset ontology, as well as an axiom indicating that dataset classes *Non dense forest* and *Dense forest* are disjoint. The algorithm would find that *Dense forest* in the repository is equivalent to *Dense forest* in the dataset, and consequently it would be mapped as the complement of *Non dense forest*. However, note that this solution simply finds the classes in the repository that do not share information with the value *Non dense forest*, but not the ones that are really related to it. Instead, we are interested in positive relations for the value, and our solution focuses on finding them.

This way, a new function *transformNegations*, which simply replaces the negative element of a class for a random string starting with "negation". This function has been added to the final algorithm, that is presented just below.

6.4 Final algorithm

We include here the final mapping algorithm that takes into consideration the special cases described in the previous section.

```
//remove stop words and transform negations
removeStopWords
transformNegations

//process mapping restrictions for the confirmed mappings
for each class CC ∈ ConfirmedSet,
                 CC mapped to CM through relation type Rel do
      addMappingRestrictions(CC, CM, Rel)
end for

//find and split mixtures and aggregations
for each class C ∈ NotProcessedSet do
      findAndSplitMixture(C)
      findAndSplitAggregation(C)
end for
```

```
//process values in NotProcessedSet
while $NotProcessedSet \neq \emptyset$ do
      for each class $CD_i \in NotProcessedSet$ do
            let $RSet_i$ = $satisfyingRestrictions(CD_i)$
            let $CRmax_i = CR_j$ such that $CR_j \in RSet_i$ and
                  $maxsim(CD_i,CR_j) \geq maxsim(CD_i,CR_k), \forall k$ and
                  $(\exists p\ maxsim(CD_i,CR_j)= maxsim(CD_i,CR_p) \Rightarrow$
                        $minsim(CD_i,CR_j) \geq minsim(CD_i,CR_p)\ )$
            if $sim(CD_i,CRmax_i) \geq \lambda$ and $sim(CRmax_i, CD_i) < \lambda$ then
                  let $CRmaxSet_i = \{CR_k \mid sim(CD_i,CR_k) \geq \lambda\}$
            else
                  if $sim(CD_i,CR_j) < \lambda$ and $sim(CR_j, CD_i) \geq \lambda$ then
                        let $CRmaxSet_i = \{CR_k \mid sim(CR_k, CD_i) \geq \lambda\}$
                  else
                        let $CRmaxSet_i = \{CR_j\}$
                  end if
            end if
      end for
      let $CDmax = CD_i$ , $CRmaxSet = CRmaxSet_i$ such that
            $maxsim(CD_i,CRmax_i) \geq maxsim(CD_j,CRmax_j), \forall j$ and
            $(\exists k\ maxsim(CD_i,CRmax_i)= maxsim(CD_k,CRmax_k) \Rightarrow$
                  $minsim(CD_i,CRmax_i) \geq minsim(CD_k,CRmax_k)\ ),$
            $CRmax_i \in CRmaxSet_i,\ CRmax_j \in CRmaxSet_j,\ CRmax_k \in CRmaxSet_k$
      for each $CRmax \in CRmaxSet$ do
            if $maxsim(CDmax,CRmax) > \mu$ then
                  if $typeOfRelation \neq$ equivalence then
                        $processRelatedMappings(CDmax,CRmax,$
                                                $typeOfRelation)$
                  end if
                  $addMappingRestrictions(CDmax, CRmax,$
                                                $typeOfRelation)$
                  $NotProcessedSet = NotProcessedSet \setminus \{ CDmax \}$
                  $SuggestedSet = SuggestedSet \cup \{ CDmax \}$
            else
                 $checkStructuralRulesAndDeepCopy(NotProcessedSet)$
      end if
end while

//map aggregations and mixtures classes
//    from the mapping actions of their aggregating and mixing
for each class $C \in AggregationsSet$ do
      $mappingAggregation$(C)
end for
for each class $C \in MixturesSet$ do
      $mappingMixture$(C)
end for
```

7 Terminological mapping algorithm

As it has already been mentioned, the algorithm described in the previous chapter has a significant limitation: synonyms are not considered. This way, although forest is a synonym of wood (both are defined as “the trees and other plants in a large densely wooded area” in WordNet), the *avsim* measure between “pine forest” and “wood” is zero. The heuristics of the algorithm presented in this section is based on a score measure that considers not just the terms in class names, but also their synonyms (as well as hypernyms and hyponyms) in the terminological base.

However, this simple example shows us one of the problems of using terminological bases: another terminological base like the GEMET thesaurus does not consider forest and wood as synonyms. GEMET uses the definition of wood that appears in the McGraw-Hill Zanichelli Dizionario, “a dense growth of trees more extensive than a grove and smaller than a forest”, and considers wood as a narrower term of forest. As it was discussed in Chapter 3, a thesaurus is structured according to a particular context. In this case, WordNet has a general orientation, while GEMET focuses on environmental disciplines, and consequently provides greater detail in their terms. This shows us that a mapping algorithm using different terminological bases will produce different results. In any case, by using either WordNet or GEMET, the algorithm is provided with more information than just using strings.

WordNet (Miller 1990; Fellbaum 1998) is a big lexical database developed at the Cognitive Science Laboratory at Princeton University. The origin of WordNet is a dictionary supporting conceptual searches besides the classical alphabetical ones. Concepts in WordNet are organized through logical groupings, called synsets (see also 5.1.3). Each synset is a set of synonymous words or collocations. A word or collocation may appear in different synsets, each providing a different sense for the word or collocation. Synsets are semantically related among them. These relations include hypernymy/hyponymy (more-generic/more-specific), meronymy/holonymy (part-of/whole-of), antonymy and entailment. Each synset has a lexical category: nouns, verbs, adjectives and adverbs. And each category has its own semantic network of relations. In the particular case of nouns, the most relevant category in our context, they are organized in a taxonomy based on the hypernymy/hyponymy relation between synsets. Nouns are also related through meronymy/holonymy and antonymy. Current version (2.1) of WordNet contains 117,097 nouns, organized in 81,426 synsets.

As it has been discussed in Chapter 3, a thesaurus is a controlled and hierarchically structured set of terms, organized through the relations broader/narrower term (BT/NT). Terms also have other standardized relations as related term (RT), use-instead (USE) or used-for (UF). It has to be noted that BT/NT relations in a thesaurus are in fact equivalent to hypernymy/hyponymy relations in WordNet. Likewise, UF/USE relations provide a kind of synonymy. As we have also already mentioned in Chapter 3, GEMET (EEA 2001), which focuses on environmental terms, is the most relevant thesaurus in the GI domain.

We characterize a terminological base, which is the core of this algorithm, as a set of terms. A term has a name, which is a string that usually contains only one word, although sometimes may comprise several (compound terms, also called collocations). Terms in the terminological base are related according to three different types of relationships: synonymy, hypernymy and hyponymy. We can observe that both lexical bases as WordNet and thesauri as GEMET fulfil these conditions. We have developed two implementations of the algorithm, one dealing with WordNet, and another one dealing with GEMET. However, any other lexical base or normalized thesaurus could be used instead. From now on, to simplify the notation, we will use indistinctly thesaurus and lexical base, and will point out specific situations where differences between them exist.

We will see in this section that the mapping algorithm using a terminological base is very similar to the one presented in the previous section, but significant changes are introduced in the definition of the similarity measure, and in how the mapping actions are determined according to this measure.

It has to be remarked that if two classes are lexically equivalent according to the algorithm in the previous section (their *minsim* is higher than or equal to λ), then there is no need to find synonyms or hypernyms/hyponyms by means of this terminological approach.

As in the previous section, we present here a generic mapping algorithm for two ontologies. It is worth recalling that in our particular context, one of the ontologies (the dataset application ontology) is represented through a set of individuals of *QualitativeDataset* and *QualitativeDatasetValue* classes, and a straightforward adaptation is needed to be implemented in the OntoGIS tool.

In principle, it can be expected that the terminological algorithm produces better results than the one based on strings only. However, in certain cases, the terminological approach may produce false positives. This happens principally when the terminological base does not cover in detail the area of knowledge of the dataset. Summing up, compared to the string-based approach, this algorithm usually provides better recall, but may obtain worse precision. Since this algorithm needs more space and time than the string-based one, the user should consider whether the terminological base provides a good support for the particular dataset being merged.

The structure of this chapter is also similar to Chapter 6. In Section 7.1 we describe the similarity (now called score) measures and how mapping actions are obtained from them. Section 7.2 presents the structure-driven algorithm. Some particular cases requiring slight algorithm changes are analyzed in Section 7.3. And finally, Section 7.4 presents the final algorithm considering all these modifications.

7.1 Score measures and mapping actions

As we have seen in Chapter 6, class names usually comprise several terms. The first step of the terminological approach consists in retrieving the most similar entry in the thesaurus for each term. This way, given a class with name *S* with the set of terms $\{t_1,...,t_n\}$, we obtain a list of their similar –often equal– thesaurus terms $\{th_1,...,th_m\}$. While the pure lexical similarity between two classes was obtained from the similarity between their terms $\{t_1,...,t_n\}$, the thesaurus-based similarity is obtained from $\{th_1,...,th_m\}$, taking also into consideration the synonyms, hypernyms and hyponyms of each th_i.

The correspondence between terms and thesaurus terms is not necessarily 1-to-1. Instead, given a term, we consider all the terms in the thesaurus that have a *tsim* higher than or equal to a threshold λ. This way, derivatives can also be retrieved. Let us also recall that frequent suffixes, like –ing, –s or –ed, are removed from both sides. It has also to be remarked that in the case of WordNet only nouns are considered. The algorithm looks for similarities in the index of both nouns and adjectives, but in the case of adjectives, their related nouns, if there is any, are used instead. For instance, the nouns “farming, agriculture, husbandry” are used for the adjective “agricultural”. Verbs or adverbs, that seldom appear are not considered. In the case of GEMET, since it does not provide lexical information, there is no way to distinguish between adjectives and nouns, and every term is considered.

Another relevant aspect to be considered is related to the fact that terminological bases contain compound terms. A compound term is a lexical unit that comprises more than one word, and that usually has the lexical function of a noun. This way, combinations of words from the class name are searched in the terminological base. If two or more words match a compound thesaurus term (their *minsim* is higher than or equal to λ), they are considered as a unit, and will not be used individually. For instance, if we have a class “rain forest” in the dataset, since “rain forest” is a noun in WordNet, they will be consider as only one term, with one associated thesaurus term; neither “rain” nor “forest” will be considered as terms.

Summing up, given a class name *S*, it will be divided in terms $\{t_1,...,t_n\}$, where each of these terms may comprise one or more words. Each term is associated with a set of thesaurus names:

$$thesaurusTerms(t_i) = \{\, th \in \text{Thesaurus such that } minsim(t_i,th) \geq \lambda \}$$

Note that, since both terms and thesaurus terms may have more than one word, *minsim* is used to measure similarities. Note also that in the case of single terms, *minsim* and *tsim* give the same value, since *tsim* is symmetric.

As it was discussed in 5.1.3, a common problem of big thesauri, usually elaborated from different sources, is that they often provide a heterogeneous structure of the hypernymy/hyponymy relation. In this context of a heterogeneous structure, path distance does not provide a reliable indicator for how similar or different two terms are. Intuitively unrelated terms may share a common hypernym at a short distance. This fact drives us to not consider as related two terms that share a hypernym. Instead, we will focus on finding if a term in one class is synonym, hypernym or hyponym of another term in another class, as described just below.

Let us consider two terms t_1 and t_2, and their respective sets of thesaurus terms $thSet_1$ and $thSet_2$ ($thSet_i$=$thesaurusTerms(t_i)$, i=1,2). The sets of synonyms of $thSet_1$ and $thSet_2$ are respectively $sSet_1=\{s_{11},..., s_{1m}\}$ and $sSet_2=\{s_{21},..., s_{2n}\}$. $hPOSet_1=(\{h_{11},..., h_{1p}\}, \leq)$ and $hPOSet_2=(\{h_{21},..., h_{2q}\}, \leq)$ are their respective partial order sets of hyponyms, and $HPOSet_1=(\{H_{11},..., H_{1r}\}, \leq)$ and $HPOSet_2=(\{H_{21},..., H_{2s}\}, \leq)$ their respective partial order set of hypernyms, where $\leq$ refer to the hypernym/hyponym relation in the thesaurus, and $hPOSet_1$, $hPOSet_2$, $HPOSet_1$, $HPOSet_2 \subset$ Thesaurus. To simplify the notation, the sets of hyponyms and hypernyms are expanded with their synonyms. In fact, WordNet automatically does it when retrieves hyponyms or hypernyms of a term, since it is organized by means of synsets. It is important to clarify that in the case of WordNet hyponyms and hypernyms are obtained through synsets and not through individual terms. For instance, the thesaurus term “rain forest” has the synset “forest, woodland, timberland, timber” as an hypernym. In order to obtain the following hypernyms, only the hypernyms of this synset are considered, which are two synsets: “land, dry land, earth, ground, solid ground, terra firma” and “biome”. This way, hypernyms of other meanings of “forest” like “vegetation, flora, botany” are not considered.

It has also to be noted here that a simplification is done in our implementation due to optimization purposes: partial order sets $hPOSet$ and $HPOSet$ for a given term th, are represented as arrays of nodes, where each of these nodes contains a term (hypernym or hyponym depending on the set) and its distance to th in the thesaurus. This way, if h_1 and h_2 are hyponyms of th, the implementation cannot know if h_1 is related to h_2. But this is not necessary in any case.

According to these considerations, we can identify three types of relations between two terms t_1 and t_2. These three types of relations are:

1. t_1 and t_2 are equivalent, which happens if one of the following occurs:
   1.1. $sim(t_1, t_2) \geq \lambda$ and $sim(t_2, t_1) \geq \lambda$
   1.2. $\exists\ th_{1i} \in thSet_1$ such that $th_{1i} \in thSet_2$
   1.3. $\exists\ th_{1i} \in thSet_1$ such that $th_{1i} \in sSet_2$
   1.4. $\exists\ s_{1i} \in sSet_1$ such that $s_{1i} \in thSet_2$
   1.5. $\exists\ s_{1i} \in sSet_1$ such that $s_{1i} \in sSet_2$
2. t_1 is more specific than t_2, which happens if one of the following occurs:
   2.1. $\exists\ th_{1i} \in thSet_1$, such that $th_{1i} \in hPOSet_2$
   2.2. $\exists\ H_{1i} \in HPOSet_1$, such that $H_{1i} \in thSet_2$
   2.3. $\exists\ s_{1i} \in sSet_1$, such that $s_{1i} \in hPOSet_2$
   2.4. $\exists\ H_{1i} \in HPOSet_1$, such that $H_{1i} \in sSet_2$
3. t_1 is more generic than t_2, which happens if one of the following occurs:
   3.1. $\exists\ th_{1i} \in thSet_1$, such that $th_{1i} \in HPOSet_2$
   3.2. $\exists\ h_{1i} \in hPOSet_1$, such that $h_{1i} \in thSet_2$
   3.3. $\exists\ s_{1i} \in sSet_1$, such that $s_{1i} \in HPOSet_2$
   3.4. $\exists\ h_{1i} \in hPOSet_1$, such that $h_{1i} \in sSet_2$

In fact, we can observe that, since we have expanded hyponyms and hypernyms sets with synonyms, conditions 2.3, 2.4, 3.3 and 3.4 are redundant.

It is worth mentioning here that we can assume that the thesaurus satisfies the following three properties, which is certainly the case of WordNet and GEMET:

- synonymy is a reflexive relation: *a* is a synonym of *a*
- synonymy is a symmetric relation: if *a* is a synonym of *b*, then *b* is also a synonym of *a*
- hyponym and hypernymy are in fact the same relation determined by a unique partial order: *a* is hyponym of *b* if and only if *b* is hypernym of *a*

These properties allow us to store the sets of synonyms, hypernyms and hyponyms of the terms of the class names of only one of the two ontologies being merged. In our case, we generate these sets for all the terms in the dataset ontology. This is motivated by the fact that the repository ontology is usually much bigger and the process of obtaining all the synonym, hypernym and hyponym sets for each term of its classes would heavily slow down the merging process. Therefore, for each term in the repository ontology, we only retrieve its thesaurus terms. And consequently, the previous conditions can be reduced to the following, where in our case t_1 is a term of a class in the dataset ontology and t_2 in the repository ontology:

1. t_1 and t_2 are equivalent if one of the following occurs:
   $sim(t_1, t_2) \geq \lambda$ and $sim(t_2, t_1) \geq \lambda$
   $\exists\, s_{1i} \in sSet_1$ such that $s_{1i} \in thSet_2$
2. t_1 is more specific than t_2 if the following occurs:
   $\exists\, H_{1i} \in HPOSet_1$, such that $H_{1i} \in thSet_2$
3. t_1 is more generic than t_2 if the following occurs:
   $\exists\, h_{1i} \in hPOSet_1$, such that $h_{1i} \in thSet_2$

Consequently, given any pair of terms t_1 and t_2 from two classes of different ontologies, the algorithm obtains the relation between them, that can be one of the following: equivalent (condition 1), t_1 more specific than t_2 (condition 2), or t_1 more generic than t_2 (condition 3). Then, given two related terms, we define a *term mapping relation*, or *tmr*, as a tuple <t_D , t_R, relation, *tscore*>, where *tscore* is a measure of the reliability of the *tmr* that is described just below.

Given a class C_D of one ontology (from dataset in our case) with terms $\{t_{D1},..., t_{Dm}\}$, and a class C_R in the other ontology (from repository in our case) with terms$\{t_{R1},..., t_{Rn}\}$, the algorithm has to decide for each t_{Di}, which t_{Rj} is the best to be mapped to. To do so, we define a score measure between two terms, *tscore*, that describes how reliable the relation between them can be considered. For t_{Di} the best related t_{Rj} will be the one maximizing the *tscore*.

tscore function is defined in the interval [0,1]. *tscore* between t_{Di} and t_{Rj} has to be zero if they are not related according to the three conditions mentioned above. In addition, if t_{Di} and t_{Rj} have a high string-based similarity, $minsim(t_{Di},t_{Rj}) \geq \lambda$, *tscore* has to be 1. In the remaining cases, *tscore* has to give priority to mappings with synonyms and in case of hypernyms or hyponyms, it has to prioritize the mapping with the closest terms in the hierarchy. In this particular case, we use as an indicator of closeness the “vertical” path distance in the hypernym/hyponym hierarchy. By “vertical” we mean that only paths from top to leaf nodes are considered. This way, if t_{Di} is not hyponym or hypernym of t_{Rj}, the “vertical” distance between them is infinite.

It is worth clarifying that we only use this path distance when two thesaurus terms are related according to the 3 above conditions. Given two thesaurus terms th_{Rj} and th_{Rk} either hypernyms or hyponyms of another thesaurus term th_{Di}, this path distance will allow us to prioritize the thesaurus term th_R (either th_{Rj} or th_{Rk}) closer to th_{Di}. But it is not used, as other merging systems do, as the global indicator of similarity between two classes.

According to these premises, the *tscore* function between two terms t_1 and t_2 is defined in the following way:

$$tscore(t_1,t_2)=\begin{cases} 1 & if\ sim(t_1,t_2)\geq\lambda\ and\ sim(t_2,t_1)\geq\lambda \\ max_{i,j}\left(\dfrac{threlated(th_{1i},th_{2j})}{1+distance(th_{1i},th_{2j})}\right) & otherwise \end{cases}$$

where $th_{1i}\in thesaurusTerms(t_1)$ and $th_{2j}\in thesaurusTerms(t_2)$; $threlated(th_{1i}, th_{2j})$ is a Boolean function which is 1 if t_1 and t_2 are related through the thesaurus terms th_{1i} and th_{2j} according to the 3 conditions above mentioned, and 0 otherwise; and $distance(th_{1i},th_{2j})$ is the number of arcs in the hyponym/hypernym relationship of the thesaurus between th_{1i} and th_{2j} in case they are related (note that the distance is 0 if th_{1i} and th_{2j} are synonyms); in case of two unrelated terms, their distance is defined as the number of terms in the thesaurus (instead of infinite), although this has no effect in *tscore* which would be 0, since the numerator would be 0 too.

We also need a measure that allow us to identify the best class to be mapped to C_D. This way, we define a score function, *score*, obtained from the average of the best *tscore* for each term of C_D, in an analogous way as *sim* similarity function has been defined in Chapter 6. The *score* function between two classes with names $C=\{t_1, ..., t_n\}$ and E is defined as follows:

$$score(C,E)=\frac{\sum_{i=1}^{n} max_{t\in E}(tscore(t_i,t))}{n}$$

Note that as *sim* function, this *score* function is also asymmetric since $score(C, E) \neq score(E, C)$. However, note that *tscore* is a symmetric function, and consequently $score(E, C)$ does not need to re-compute the *tscore* measures. As in the string-based algorithm, we define a symmetric function *avscore* and it will be used to find the best class to be mapped to a given class:

$$avscore(C,E) = \frac{score(C,E)+score(E,C)}{2}$$

Once the algorithm has found that the dataset class C_D has to be mapped to the repository class C_R (how this is mapping is found will be discussed in Section 7.2), then it has to determine the mapping action between C_D and the selected C_R. To do so, we define *tmrSet* as the set of the best term mapping relations between terms of C_D and C_R. This set initially contains, for each term t_{Di} of C_D, its best *tmr*, i.e. the one with the highest *tscore* with a related term in C_R:

```
for each $t_{Di} \in C_D$ do
      if $\exists t \in C_R$ such that $related(t_{Di},t)=1$ then
            let term $t_R \in C_R$ be such that
                  $\forall u_R \in C_R,\ tscore(t_{Di},\ t_R) \geq tscore(t_{Di},u_R)$
            let $r$ be the relation between $t_{Di}$ and $t_R$
            add tmr < $t_{Di}$, $t_R$ , $r$, $tscore(t_{Di},\ t_R)$ >
      end if
end for
```

where *related*(t_1, t_2) is a Boolean function returning 1 if t_1 and t_2 are related according to the 3 conditions mentioned above, and 0 otherwise.

In a second stage, new *tmr*'s are added to those terms in C_D that have no relations with terms in C_R:

```
for each $t_{Di} \in C_D$ do
      if $\forall t_R \in C_R$ , related($t_{Di}$,$t_R$)=0 then
            add tmr < $t_{Di}$, -, "noRelationD", 0 >
      end if
end for
```

Finally, new *tmr*'s are added for those terms in C_R that have no relations with terms in C_D:

```
for each $t_{Rj} \in C_R$ do
      if $\forall t_D \in C_D$ , related($t_D$,$t_{Rj}$)=0 then
            add tmr < - , $t_{Rj}$, "noRelationR", 0 >
      end if
end for
```

Note that two new types of relation between terms have been defined: *noRelationD* which indicates that a term in C_D has no relation, and *noRelationR* which indicates that a term in C_R has no relation.

The logic for determining the mapping action between two classes C_D and C_R is based on the relations in *tmrSet* and is funded on similar assumptions to those made for the string-based approach. C_D and C_R are considered equivalent if and only if all the *tmr* in *tmrSet* have "equivalent" relations. Let us consider now the case where we have one or more *tmr*'s with "more specific" relation: if we have any other *tmr* with "more generic" relation, we assume that C_D and C_R have common information that is specialized in different ways and consequently they can be considered as having a common superclass; otherwise, if there are no “more generic” *tmr*’s, C_D can be considered as subclass of C_R. The opposite is applied for determining if C_D is superclass of C_R. Let us consider the case when we have *tmr*’s with no-relations: we assume that if we have common information and an unrelated term, this term provides more specific information; this way, when there is at least an “equivalent” relation, the behaviour of a “noRelationD” is similar to a “more specific”, while a “noRelationR” is similar to a “more generic”. These criteria are summarised in the following way:

- C_D equivalent to C_R if and only if:
    - $\forall tmr \in tmrSet$, tmr.relation = "equivalent"
- C_D subclass of C_R if and only if occurs one of the following:
    - $\exists tmr_1 \in tmrSet$, tmr_1.relation = "more specific" and
      $\nexists tmr_2 \in tmrSet$, tmr_2.relation = "more generic" and
      $\nexists tmr_3 \in tmrSet$, tmr_3.relation = "noRelationR"
    - $\exists tmr_1 \in tmrSet$, tmr_1.relation = "equivalent" and
      $\exists tmr_2 \in tmrSet$, tmr_2.relation = "noRelationD" and
      $\nexists tmr_3 \in tmrSet$, tmr_3.relation = "more generic" and
      $\nexists tmr_4 \in tmrSet$, tmr_4.relation = "noRelationR"
- C_D superclass of C_R if and only if occurs one of the following:
    - $\exists tmr_1 \in tmrSet$, tmr_1.relation = "more generic" and
      $\nexists tmr_2 \in tmrSet$, tmr_2.relation = "more specific" and
      $\nexists tmr_3 \in tmrSet$, tmr_3.relation = "noRelationD"
    - $\exists tmr_1 \in tmrSet$, tmr_1.relation = "equivalent" and
      $\exists tmr_2 \in tmrSet$, tmr_2.relation = "noRelationR" and
      $\nexists tmr_3 \in tmrSet$, tmr_3.relation = "more specific" and
      $\nexists tmr_4 \in tmrSet$, tmr_4.relation = "noRelationD"
- C_D has a common superclass with C_R (create C_X: $C_D \sqsubseteq C_X$, $C_R \sqsubseteq C_X$) if and only if occurs one of the following:
    - $\exists tmr_1 \in tmrSet$, tmr_1.relation = "more specific" and
      $\exists tmr_2 \in tmrSet$, tmr_2.relation = "more generic"
    - $\exists tmr_1 \in tmrSet$, tmr_1.relation = "equivalent" and
      $\exists tmr_2 \in tmrSet$, tmr_2.relation = "noRelationR" and
      $\exists tmr_3 \in tmrSet$, tmr_3.relation = "noRelationD"
- Otherwise, there is no relation between C_D and C_R

As in the string-based algorithm, if the relation between C_D and C_R is not an equivalence, the algorithm will search the related mappings, in order to try to have at least one subclass and one superclass mappings for each dataset class.

7.2 Structure-driven algorithm

The algorithm that determines the order in how mappings are processed follows the same strategy as the string-based approach described in 6.2. It is also based on the graph structure of the ontologies being merged and on the mapping restrictions being generated during the process. Furthermore, it should execute firstly the most reliable mappings and should prioritize equivalence and subclass/superclass mappings with respect to common superclass. The algorithm also considers structural rules to infer new mappings in the case of terminologically unrelated classes, as in the case of the lexical algorithm.

However, unlike *sim* measures in the string-based approach, *score* does not provide an indication of the type of relation between two classes. On the other hand, *avscore* is a measure of how strongly related two classes are. The algorithm has to be adapted to use *score* (and *avscore*) with the same strategy followed by the string-based algorithm.

While we have pointed out that a matrix has been used to store all the *sim* measures between dataset classes and repository classes in the string-based algorithm, now a matrix *M* is defined to contain not only *avscore* measures but also the corresponding mapping action that can be deduced from the term mapping restrictions. *M* contains *m*x*n* cells, where *m* and *n* are the number of classes respectively in the dataset ontology and in the repository ontology. The cell *M*(*i*,*j*) has two fields, *avscore* and *typeOfRelation* that respectively contain *avscore*(C_{Di},C_{Dj}) and the type of relation between C_{Di} and C_{Rj} (equivalence, subclass, superclass, commonSuperclass or noRelation), where C_{Di} and C_{Rj} belong respectively to dataset and repository ontologies. This matrix is computed at the beginning of the algorithm and the values of a dataset class C_D are only modified when a new mapping restrictions is added to C_D. In this case, if C_{Di} cannot be mapped to C_{Rj} according to the mapping restrictions of C_{Di}, then the pair <-1,no relation> will be assigned to *M*(*i*,*j*).

From now on, wherever we use *avscore*(C_{Di},C_{Dj}) in the algorithms, we are referring to *M*(*i*,*j*).*avscore*; likewise, wherever we use *typeOfRelation*(C_{Di},C_{Dj}), we are referring to *M*(*i*,*j*).*typeOfRelation*.

The algorithm follows the following flow:

(1) It removes stop words
(2) It processes the mapping restrictions for the manually confirmed mappings
(3) It extracts the thesaurus terms for all the terms in the classes of both ontologies, and in the case of the dataset ontology, it also extracts the synonyms, hypernyms and hyponyms of these thesaurus terms
(4) It generates the matrix *M*
(5) It processes the possible equivalence mappings, ordered according to their *avscore*; for each of these, it generates the corresponding mapping restrictions
(6) It processes the possible subclass and superclass mappings in a similar way. Related mappings (see above) are also processed
(7) It processes the possible common superclass mappings, and related mappings in a similar way.
(8) It checks whether structural rules can be applied to the remaining dataset classes

Each time a mapping is suggested for a dataset class, it is removed from the set that contains the dataset classes not processed yet, *NotProcessedSet*. The process finishes when all the dataset classes have been processed (*NotProcessedSet* is empty).

It is worth noting here that no threshold μ is defined in this algorithm: while in the string-based algorithm only *maxsim* measures above a threshold μ were considered, we now use any relation obtained from the thesaurus, since it is more reliable than a simple common substring of three or more characters. Likewise, threshold λ is only defined to determine the thesaurus terms for the terms in the classes, but not to determine if two classes are equivalent.

We present now the detailed structure-driven algorithm according to the previous discussions:

```
//(1) remove stop words
removeStopWords

//(2) process mapping restrictions for the confirmed mappings
for each class $CC \in ConfirmedSet$,
                $CC$ mapped to $CM$ through relation type $Rel$ do
    addMappingRestrictions($CC$, $CM$, $Rel$)
end for

//(3) extract information from the thesasurus
for each class $CD_i \in DatasetOntology$ do
    obtain its set of terms: $termsSetD_i$ = { $t_{i1}$,...,$t_{in}$ }
    for each term $t_{ij} \in termsSetD_i$ do
        obtain $thSet_{ij}$ = $thesaurusTerms(t_{ij})$
        obtain $sSet_{ij}$, $hPOSet_{ij}$ and $HPOSet_{ij}$ for $thSet_{ij}$
    end for
end for
for each class $CR_k \in RepositoryOntology$ do
    obtain its set of terms: $termsSetR_k$ = { $t_{k1}$,...,$t_{kp}$ }
    for each term $t_{km} \in termsSetR_k$ do
        obtain $thSet_{km}$ = $thesaurusTerms(t_{km})$
    end for
end for

//(4) fill matrix $M$
fillMatrix

//(5) process equivalences for dataset classes in
//    NotProcessedSet
let $score0$ be a Boolean initialized to false
while $NotProcessedSet \neq \emptyset$ and $score0$=false do
    for each class $CD_i \in NotProcessedSet$ do
        let $RSet_i$ = satisfyingRestrictions($CD_i$)
        let $CRmax_i$ = $CR_j$ such that $CR_j \in RSet_i$ and
           $typeOfRelation(CD_i,CR_j)$=equivalence and
           ($avscore(CD_i,CR_j) \geq avscore(CD_i,CR_k)$,
           $\forall k$ such that $typeOfRelation(CD_i,CR_k)$=equivalence)
    end for
    Let $CDmax$ = $CD_i$ , $CRmax$ = $CRmax_i$ such that
        $avscore(CD_i,CRmax_i) \geq avscore(CD_j,CRmax_j)$, $\forall j$
    if $avscore(CDmax,CRmax) > 0$ then
        suggestMapping($CDmax$,$CRmax$)
        addMappingRestrictions($CDmax$, $CRmax$, equivalence)
        $NotProcessedSet = NotProcessedSet \setminus \{ CDmax \}$
        $SuggestedSet = SuggestedSet \cup \{ CDmax \}$
    else
        $score0$ = true
    end if
end while
```

```
//(6) process subclasses and superclasses for dataset classes in
//    NotProcessedSet
score0 = false
while $NotProcessedSet \neq \emptyset$ and score0=false do
      for each class $CD_i \in NotProcessedSet$ do
            let $RSet_i$ = satisfyingRestrictions($CD_i$)
            let $CRmax_i = CR_j$ such that $CR_j \in RSet_i$ and
               (typeOfRelation($CD_i$,$CR_j$)=subclass or
               typeOfRelation($CD_i$,$CR_j$)=superclass) and
               (avscore($CD_i$,$CR_j$) $\geq$ avscore($CD_i$,$CR_k$),
               $\forall k$ such that (typeOfRelation($CD_i$,$CR_k$)=subclass or
                  typeOfRelation($CD_i$,$CR_k$)=superclass)
      end for
      Let $CDmax = CD_i$ , $CRmax = CRmax_i$ such that
            avscore($CD_i$,$CRmax_i$) $\geq$ avscore($CD_j$,$CRmax_j$), $\forall j$
      if avscore(CDmax,CRmax) > 0 then
            suggestMapping(CDmax,CRmax)
            processRelatedMappings(CDmax,CRmax,
                                   typeOfRelation(CDmax,CRmax))
            addMappingRestrictions(CDmax,CRmax,
                                   typeOfRelation(CDmax,CRmax))
            NotProcessedSet = NotProcessedSet \ { CDmax }
            SuggestedSet = SuggestedSet $\cup$ { CDmax }
      else
            score0 = true
      end if
end while

//(7) process common superclasses for dataset classes in
//    NotProcessedSet
score0 = false
while $NotProcessedSet \neq \emptyset$ and score0=false do
      for each class $CD_i \in NotProcessedSet$ do
            let $RSet_i$ = satisfyingRestrictions($CD_i$)
            let $CRmax_i = CR_j$ such that $CR_j \in RSet_i$ and
                  typeOfRelation($CD_i$,$CR_j$)=commonSuperclass and
                  avscore($CD_i$,$CR_j$) $\geq$ avscore($CD_i$,$CR_k$), $\forall k$
      end for
      Let $CDmax = CD_i$ , $CRmax = CRmax_i$ such that
            avscore($CD_i$,$CRmax_i$) $\geq$ avscore($CD_j$,$CRmax_j$), $\forall j$
      if avscore(CDmax,CRmax) > 0 then
            suggestMapping(CDmax,CRmax)
            processRelatedMappings(CDmax,CRmax,commonSuperclass)
            addMappingRestrictions(CDmax,CRmax,commonSuperclass)
            NotProcessedSet = NotProcessedSet \ { CDmax }
            SuggestedSet = SuggestedSet $\cup$ { CDmax }
      else
            score0 = true
      end if
end while

//(8) check structural rules for remaining dataset classes
checkStructuralRulesAndDeepCopy ( NotProcessedSet )
```

Functions *removeStopWords, addMappingRestrictions*, *satisfyingRestrictions* and *checkStructuralRulesAndDeepCopy* are defined in the same way as for the string-based approach. On the other hand, there are significant differences in the function *suggestMapping*, which connects a class of the dataset ontology to a class of the repository ontology, according now to the type of relation stored in *M*. Likewise, *processRelatedMappings* is defined in a similar way as in the string-based approach, but now using matrix *M*.

Function ***fillMatrix*** contains the logic for obtaining the type of relation between classes from the tmr's (term mapping restrictions) according to the rules specified in Section 7.1. It is defined as follows:

```
function fillMatrix
  for each $CD_i \in$ Dataset Ontology do
    for each $CR_j \in$ Repository Ontology do
      M(i,j).avscore = avscore($CD_i$,$CR_j$)
      let tmrSet the set of tmr's for $CD_i$ and $CR_j$
      if $\forall tmr \in tmrSet$, tmr.relation = "equivalent" then    //$CD \equiv CR$
        M(i,j).typeOfRelation = equivalent
      else
        if ( $\exists tmr_1 \in tmrSet$ , $tmr_1$.relation = "more specific" and
                $\nexists tmr_2 \in tmrSet$ , $tmr_2$.relation = "more generic" and
                $\nexists tmr_3 \in tmrSet$ , $tmr_3$.relation = "noRelationR" ) or
           ( $\exists tmr_1 \in tmrSet$ , $tmr_1$.relation = "equivalent" and
                $\exists tmr_2 \in tmrSet$ , $tmr_2$.relation = "noRelationD" and
                $\nexists tmr_3 \in tmrSet$ , $tmr_3$.relation = "more generic" and
                $\nexists tmr_4 \in tmrSet$ , $tmr_4$.relation = "noRelationR" )
                then //$CD \sqsubseteq CR$
          M(i,j).typeOfRelation = subclass
        else
          if ( $\exists tmr_1 \in tmrSet$ , $tmr_1$.relation = "more generic" and
                $\nexists tmr_2 \in tmrSet$ , $tmr_2$.relation = "more specific" and
                $\nexists tmr_3 \in tmrSet$ , $tmr_3$.relation = "noRelationD" ) or
             ( $\exists tmr_1 \in tmrSet$ , $tmr_1$.relation = "equivalent" and
                $\exists tmr_2 \in tmrSet$ , $tmr_2$.relation = "noRelationR" and
                $\nexists tmr_3 \in tmrSet$ , $tmr_3$.relation = "more specific" and
                $\nexists tmr_4 \in tmrSet$ , $tmr_4$.relation = "noRelationD" )
                then // $CD \sqsupseteq CR$
          M(i,j).typeOfRelation = superclass
          else
            if  ( $\exists tmr_1 \in tmrSet$ , $tmr_1$.relation = "more specific" and
                  $\exists tmr_2 \in tmrSet$ , $tmr_2$.relation = "more generic" ) or
                ( $\exists tmr_1 \in tmrSet$ , $tmr_1$.relation = "equivalent" and
                  $\exists tmr_2 \in tmrSet$ , $tmr_2$.relation = "noRelationR" and
                  $\exists tmr_3 \in tmrSet$ , $tmr_3$.relation = "noRelationD" )
                then //common superclass
              M(i,j).typeOfRelation = commonSuperclass
            else
              M(i,j).typeOfRelation = noRelation
            end if
          end if
        end if
      end if
    end function
```

Mappings are executed in the same way as in the string-based approach, except in the case of equivalence. In the string approach, an equivalence between classes C_D and C_R means that they have the same (or very similar) name, and consequently no new class is added, just a connection between C_D and C_R. In the terminological approach, it has to be checked whether they have the same name or are synonyms. Therefore, the mapping action would be executed in the following way:

```
if $sim(CD,CR) \geq \lambda$ and $sim(CR,CD) \geq \lambda$ then
      add connexion between CD and CR
else
      add class newCD
      add axiom "$newCD \equiv CR$"
      add connexion between CD and newCD
end if
keepDsRelations(CD)
```

7.3 Some significant special cases

As in the case of the string-based algorithm (see 6.3), there are some special cases that require a slight modification of the general algorithm. They are discussed in this section. These special cases comprise those already presented in the string-based approach, with some adaptations. Three other situations are specific for the terminological approach. The final algorithm that covers all these cases is included in Section 7.4.

7.3.1 Redundancies in the dataset values structure

Let us consider an example from the USGS land cover vocabulary, which contains *pasture* and *grassland* among their physical values. However, one of the two senses (or synsets) of *pasture* in WordNet has the hypernym *grassland*. Since physical values in the dataset are assumed not to overlap, all grasslands are supposed to be identified by means of the value *grassland*. This drives the algorithm to discard this hypernym relation. Note that this provides a simple mechanism for disambiguation, since in fact the algorithm discards one of the senses of pasture (the one that makes it an hyponym of grassland).

Generalizing and formalizing this idea, let us consider two classes C_{D1} and C_{D2} from the dataset ontology, where C_{D1} has a unique thesaurus term th_1, while C_{D2} also has a unique thesaurus term th_2. If th_1 is hyponym of th_2, th_2 and its hypernyms are removed from the set of hypernyms of th_1. Likewise, th_1 and its hyponyms are removed from the set of hyponyms of th_2. Function ***removeRedundancies*** is defined to eliminate redundancies of this type, and will be included in the final general algorithm, which is presented in Section 7.4.

```
function removeRedundancies
   for each pair $C_{D1}$,$C_{D2}$ $\in$Dataset ontology, with names $S_{D1}$,$S_{D2}$,
            such that $C_{D1}$ = { $t_1$ } and $C_{D2}$ ={ $t_2$ }
             and $th_1\in thesaurusTerms(t_1)$ and |$thesaurusTerms(t_1)$|=1
             and $th_2\in thesaurusTerms(t_2)$ and |$thesaurusTerms(t_2)$|=1
            do
      if $th_1 \in hPOSet_2$ then
            for each $th \in HPOSet_1$ such that $th \in HPOSet_2$ do
                  $HPOSet_1 = HPOSet_1 \setminus \{ th \}$
            end for
            $HPOSet_1 = HPOSet_1 \setminus \{ th_2 \}$

            for each $th \in hPOSet_2$ such that $th \in hPOSet_1$ do
                  $hPOSet_2 = hPOSet_2 \setminus \{ th \}$
            end for
            $hPOSet_2 = hPOSet_2 \setminus \{ th_1 \}$
      end if
   end for
end function
```

This process of removing redundancies has to be also applied to aggregating or mixing classes.

7.3.2 Meronymy relation

Although it is not common, sometimes the meronym (part-of) of a term is used instead of the term itself in a class name. For instance, the Simple Biosphere Model includes a value *Evergreen Needleleaf Trees*. Since *tree* in WordNet is a meronym of the synset *forest, wood, woods* (a tree is part of a forest) we can assume that this value is equivalent to *Evergreen Needleleaf Forest* from the USGS land cover vocabulary.

This way, we can observe that in this case *trees* can be considered as a synonym of *forest*. And the hyponyms and hypernyms of the synset *forest, wood, woods* can also be respectively considered as hyponyms and hypernyms of *trees*.

However, in order to avoid unpredicted results, the use of meronyms in such a way is limited. Only meronyms or holonyms of the term are considered. But not those from its hyponyms and hypernyms. For instance, the synset *forest, wood, woods* has the hypernym *vegetation, flora, botany*. The meronyms and holonyms of this synset will not be considered.

A final aspect has to be mentioned concerning meronyms. Although a meronym is assumed to be a synonym of its holonyms, their distance is not considered as 0. Instead, our heuristics assumes that a meronym is less similar to its holonym than one synonym to another; but, on the other hand, it also assumes that they are more similar than a hyponym to its hypernym. This way, the distance between a meronym and its holonym (and vice versa) is 0.5.

Meronymy relation as itself is not considered in GEMET, where it is sometimes reflected through the RT relation, and sometimes included in the BT/NT hierarchy.

Consequently this case is not considered when dealing with GEMET (or other thesauri). Even in the case of using WordNet, our implementation allows the user to enable or disable the use of meronymy. By default, it is disabled, since its use is not frequent and, on the other hand, causes an overhead.

7.3.3 Lexical categories in *avscore*

WordNet provides the lexical category of a term. This information can be used to refine the *avscore* measure in order to prioritize nouns in the meaning of a class name. Let us now consider the example of the dataset class *Herbaceous wetland* and the repository classes *Permanent wetland* and *Herbaceous crop*, and let us also suppose that the dataset class has not any mapping restriction. Note that *avscore* measure is the same in both cases (0.5). We would like to slightly modify the measure in order to prioritize *wetland* with respect to *herbaceous*, and consequently to make the algorithm to select the mapping with *Permanent wetland*. This way, when *score* measure is computed, the *tscore* of two terms is multiplied by a certain factor ω, depending on whether both terms are nouns or not:

$$score(C,E) = \frac{\sum_{i=1}^{n} \max_{t_j \in E}(\omega_{ij} \cdot tscore(t_i, t_j))}{n}$$

Factor ω_{ij} will be smaller than 1 if any of the two terms is not a noun, and will be 1 otherwise.

Furthermore, it has also to be noted that often class names contain an adjective and a word denoting a spatial unit, as in *Urban area* or *Agricultural land*. The meaning of these classes is clearly embedded in the adjectives. The algorithm, after removing "area" or "land" since are considered as stop words, gives the adjective the lexical role of a noun.

7.3.4 One-to-many relations

As in the string-based approach (see 6.3.1), we are interested in detecting those situations where a dataset class is either the subclass or the superclass of several repository classes. We are interested in those repository classes C_R such that their relation with C_{Dmax} is the same as the one between C_{Dmax} and C_{Rmax}, and that satisfy $score(C_{Dmax},C_R) \geq score(C_{Dmax},C_{Rmax})$ in the case of a superclass relation or $score(C_R,C_{Dmax}) \geq score(C_{Rmax},C_{Dmax})$ in the case of a subclass relation. Note that for efficiency reasons, this requires to store both partial *score* measures, instead of *avscore*, in the matrix *M*.

To that purpose, the following part from the processing of subclasses or superclasses (6) in the original algorithm, that determines the best class $CRmax_i$ from the repository to be mapped to the dataset class CD_i

```
for each class $CD_i \in NotProcessedSet$ do
      let $RSet_i$ = $satisfyingRestrictions(CD_i)$
      let $CRmax_i$ = $CR_j$ such that $CR_j \in RSet_i$ and
            ($typeOfRelation(CD_i,CR_j)$=subclass or
            $typeOfRelation(CD_i,CR_j)$=superclass) and
            ($avscore(CD_i,CR_j) \geq avscore(CD_i,CR_k)$,
            $\forall k$ such that ($typeOfRelation(CD_i,CR_k)$=subclass or
            $typeOfRelation(CD_i,CR_k)$=superclass)
end for
```

has to be replaced, in order to deal now with a set of classes *CRmaxSet*$_i$, by:

```
for each class $CD_i \in NotProcessedSet$ do
      let $RSet_i$ = $satisfyingRestrictions(CD_i)$
      let $CRmax_i$ = $CR_j$ such that $CR_j \in RSet_i$ and
            ($typeOfRelation(CD_i,CR_j)$=subclass or
            $typeOfRelation(CD_i,CR_j)$=superclass) and
            ($avscore(CD_i,CR_j) \geq avscore(CD_i,CR_k)$,
            $\forall k$ such that ($typeOfRelation(CD_i,CR_k)$=subclass or
            $typeOfRelation(CD_i,CR_k)$=superclass)
      if $typeOfRelation(CD_i,CR_j)$=superclass then
            let $CRmaxSet_i$ = { $CR_k \in RSet_i$ |
                        $typeOfRelation(CD_i,CR_k)$=superclass and
                        $score(CD_i,CR_k) \geq score(CD_i,CR_j)$ }
      else //subclass
            let $CRmaxSet_i$ = { $CR_k \in RSet_i$ |
                        $typeOfRelation(CD_i,CR_k)$=subclass and
                        $score(CR_k,CD_i) \geq score(CR_j,CD_i)$ }
      end if
end for
```

7.3.5 Aggregations, mixtures and negations of values

Aggregations and mixtures are managed in the same way as in the string-based approach. Nevertheless, the use the lexical categories of terms provided by WordNet also offers a better way of separation of the atomic values being aggregated. For instance, a class with name *broadleaf and needleleaf trees* is split in *broadleaf trees* and *needleleaf trees*, instead of in *broadleaf* and *needleleaf trees* by the string-based approach. This way, a very simple grammar can be defined in order to provide better transformations of class names to aggregations. We have identified the following rules:

```
adjective1 conj adjective2 noun
      -> aggregation(adjective1 noun , adjective2 noun)

adjective noun1 conj noun2
      -> aggregation(adjective noun1 , adjective noun2)

noun1 conj noun2 noun3
      -> aggregation(noun1 noun3 , noun2 noun3)

where conj ∈ {"and", "or", "and/or", ",", "/"}
```

Function *findAndSplitAggregation*, that was defined in 6.3.2 with a simpler grammar with no lexical categories, has to be expanded according to these new transformations. Likewise, more complex grammars can also be defined for mixtures in function *findAndSplitMixture*.

Note that since *findAndSplitAggregation* and *findAndSplitMixture* may modify the number of classes that are involved in the process, this has necessarily to be done before the matrix *M* is computed.

In the case of negations, let us recall that the string-based algorithm replaces the negated part of the class name with a random string that starts by "negation". In the terminological algorithm, since WordNet also provides antonyms, the negated part can be now replaced by its antonyms, if it has any. If it has not antonyms, it is managed as in the string-based approach. This way, function *transformNegations* is refined to take antonyms into consideration.

Note that GEMET does not provide lexical categories and antonyms. This way, the functions for aggregations, mixtures and negations defined for the string-based approach have to be used when GEMET is the selected terminological base.

7.4 Final algorithm

We include here the final mapping algorithm that takes into consideration the special cases described in the previous section.

```
//remove stop words and redundancies and transform negations
removeStopWords
removeRedundancies
transformNegations

//process mapping restrictions for the confirmed mappings
for each class CC ∈ ConfirmedSet,
                 CC mapped to CM through relation type Rel do
      addMappingRestrictions(CC, CM, Rel)
end for

//find and split mixtures and aggregations
for each class C ∈ NotProcessedSet do
      findAndSplitMixture(C)
      findAndSplitAggregation(C)
end for

//extract information from the thesasurus
for each class $CD_i$ ∈ DatasetOntology do
      obtain its set of terms: $termsSetD_i$ = { $t_{i1}$,...,$t_{in}$ }
      for each term $t_{ij}$∈$termsSetD_i$ do
            obtain $thSet_{ij}$ = thesaurusTerms($t_{ij}$)
            obtain $sSet_{ij}$, $hPOSet_{ij}$ and $HPOSet_{ij}$ for $thSet_{ij}$
      end for
end for
```

```
for each class $CR_k \in RepositoryOntology$ do
      obtain its set of terms: $termsSetR_k$ = { $t_{k1}$,...,$t_{kp}$ }
      for each term $t_{km} \in termsSetR_k$ do
            obtain $thSet_{km}$ = $thesaurusTerms(t_{km})$
      end for
end for

//fill matrix M
fillMatrix

//process equivalences for dataset classes in
//    NotProcessedSet
let score0 be a Boolean initialized to false
while $NotProcessedSet \neq \emptyset$ and score0=false do
      for each class $CD_i \in NotProcessedSet$ do
            let $RSet_i$ = $satisfyingRestrictions(CD_i)$
            let $CRmax_i$ = $CR_j$ such that $CR_j \in RSet_i$ and
               $typeOfRelation(CD_i,CR_j)$=equivalence and
               ($avscore(CD_i,CR_j) \geq avscore(CD_i,CR_k)$,
               $\forall k$ such that $typeOfRelation(CD_i,CR_k)$=equivalence)
      end for
      Let CDmax = $CD_i$ , CRmax = $CRmax_i$ such that
            $avscore(CD_i,CRmax_i) \geq avscore(CD_j,CRmax_j)$, $\forall j$
      if avscore(CDmax,CRmax) > 0 then
            suggestMapping(CDmax,CRmax)
            addMappingRestrictions(CDmax, CRmax, equivalence)
            NotProcessedSet = NotProcessedSet \ { CDmax }
            SuggestedSet = SuggestedSet $\cup$ { CDmax }
      else
            score0 = true
      end if
end while

//process subclasses and superclasses for dataset classes in
//    NotProcessedSet
score0 = false
while $NotProcessedSet \neq \emptyset$ and score0=false do
      for each class $CD_i \in NotProcessedSet$ do
            let $RSet_i$ = $satisfyingRestrictions(CD_i)$
      let $CRmax_i$ = $CR_j$ such that $CR_j \in RSet_i$ and
            ($typeOfRelation(CD_i,CR_j)$=subclass or
                  $typeOfRelation(CD_i,CR_j)$=superclass) and
                  ($avscore(CD_i,CR_j) \geq avscore(CD_i,CR_k)$,
                  $\forall k$ such that ($typeOfRelation(CD_i,CR_k)$=subclass
                  or $typeOfRelation(CD_i,CR_k)$=superclass)
            if $typeOfRelation(CD_i,CR_j)$=superclass then
                  let $CRmaxSet_i$ = { $CR_k \in RSet_i$ |
                        $typeOfRelation(CD_i,CR_k)$=superclass and
                        $score(CD_i,CR_k) \geq score(CD_i,CR_j)$ }
            else //subclass
                  let $CRmaxSet_i$ = { $CR_k \in RSet_i$ |
                        $typeOfRelation(CD_i,CR_k)$=subclass and
                        $score(CR_k,CD_i) \geq score(CR_j,CD_i)$ }
            end if
      end for
```

```
    let $CDmax = CD_i$ , $CRmaxSet = CRmaxSet_i$ such that
        $avscore(CD_i,CRmax_i) \geq avscore(CD_j,CRmax_j)$,
        $CRmax_i \in CRmaxSet_i$, $CRmax_j \in CRmaxSet_j$
    for each $CRmax \in CRmaxSet$ do
        if $avscore$(CDmax,CRmax) > 0 then
            $suggestMapping(CDmax,CRmax)$
            $processRelatedMappings(CDmax,CRmax,$
                                  $typeOfRelation$(CDmax,CRmax))
            $addMappingRestrictions(CDmax,CRmax,$
                                  $typeOfRelation$(CDmax,CRmax))
            $NotProcessedSet = NotProcessedSet \setminus \{ CDmax \}$
            $SuggestedSet = SuggestedSet \cup \{ CDmax \}$
        else
            $score0$ = true
        end if
    end for
end while

//process common superclasses for dataset classes in
//    $NotProcessedSet$
$score0$ = false
while $NotProcessedSet \neq \emptyset$ and $score0$=false do
    for each class $CD_i \in NotProcessedSet$ do
        let $RSet_i = satisfyingRestrictions(CD_i)$
        let $CRmax_i = CR_j$ such that $CR_j \in RSet_i$ and
            $typeOfRelation(CD_i,CR_j)$=commonSuperclass and
            $avscore(CD_i,CR_j) \geq avscore(CD_i,CR_k)$, $\forall k$
    end for
    Let $CDmax = CD_i$ , $CRmax = CRmax_i$ such that
        $avscore(CD_i,CRmax_i) \geq avscore(CD_j,CRmax_j)$, $\forall j$
    if $avscore$(CDmax,CRmax) > 0 then
        $suggestMapping(CDmax,CRmax)$
        $processRelatedMappings(CDmax,CRmax,$commonSuperclass)
        $addMappingRestrictions(CDmax,CRmax,$commonSuperclass)
        $NotProcessedSet = NotProcessedSet \setminus \{ CDmax \}$
        $SuggestedSet = SuggestedSet \cup \{ CDmax \}$
    else
        $score0$ = true
    end if
end while

//process remaining dataset classes
    $checkStructuralRulesAndDeepCopy$ ( $NotProcessedSet$ )

//map aggregations and mixtures classes
//    from the mapping actions of their aggregating and mixing
for each class $C \in AggregationsSet$ do
    $mappingAggregation$(C)
end for
for each class $C \in MixturesSet$ do
    $mappingMixture$(C)
end for
```

8 Mapping algorithm based on the spatial distribution of dataset values

This algorithm is based on the level of overlapping among the spatial extents of sets of values from different datasets. We call *spatial extent* of a dataset value to the union of all the spatial units in the dataset such that their main thematic variable has the indicated value. A high overlapping between the spatial extents of two different values from different datasets means that they probably refer to equivalent themes. If the spatial extent of the first value is contained by the spatial extent of the second one (in a different dataset), it probably indicates that there is a subclass relation between their thematic classes. Furthermore, the algorithm does not only consider single values, but sets too. This is necessary for detecting the typical case when two themes are classified in different ways in two datasets. For instance, let us consider a land use dataset with values “broad-leaved forest” and “coniferous forests”, among others. Another dataset has the values “dense forest” and “sparse forest”, among others. Although the algorithm will not probably find any relation among individual values, it will deduce that the union of “broad-leaved forest” and “coniferous forests” is equivalent to the union of “dense forest” and “sparse forest”. The suggested action in this case will consist on defining a new class, which will have two different classifications: one comprising its subclasses “broad-leaved forest” and “coniferous forests”, while the other is made of its subclasses “dense forest” and “sparse forest”. Note that the sets do not necessarily comprise only two values, as it was in this example.

Our algorithm presents some similarities with the work of Duckham and Worboys (Duckham and Worboys 2005), analysed in depth in 3.4.2, who define an algebraic method for both merging and integration, based on the spatial distribution of values. In terms of Duckham and Worboys’s work, the value of a spatial unit is determined by the function g (its domain is the set of spatial units and its range is the set of values of the given thematic variable), while the extension function e (its domain is parts of the set of values of the thematic variable and its range is parts of the set of spatial units) is the equivalent to what we have called spatial extent. The merging approach from Duckham and Worboys is based on the spatial intersection of the spatial extents of the different values in both datasets. If the spatial extent of one value is contained by another’s, the first value is assumed to be a subclass of the second.

However, as we have already discussed, a significant limitation of this approach relates to its applicatino to real datasets, which have a big number of spatial units. In this case, it is extremely unlikely that one spatial extent is totally contained in another one, since different interpretations for a particular area, different generalizing methods or simply small cartographic errors usually exist. Consequently, Duckham’s method seldom finds any relation among two real datasets. Our approach, more flexible, uses a threshold for measuring the level of spatial containment among two spatial extents. Over the threshold, it is assumed that a subclass relation exists. This threshold has a default value defined in the implementation tool, but can be changed by the expert user.

Another significant advantage of our approach refers to the fact that our method considers different classifications of the same theme in different datasets, such as the previous example on forests. The approach of Duckham and Worboys only compares extents of individual values and cannot obtain these types of relations.

We will see that the first version of our algorithm presented in Section 8.1, with an exponential execution time of $\mathcal{O}(2^{m+n})$, where m and n are the number of values in each dataset. To improve this, we provide an optimized algorithm in Section 8.2 that can be run in real time, with a polynomial execution time.

This algorithm is based on an algebraic framework for describing datasets and its spatial units and values, that is presented in the following subsection. This way, unlike the two previous mapping algorithms, this one is specific for merging geographic dataset ontologies and can hardly be used in other contexts.

8.1 Formal definition

We define a dataset as the tuple:

$$D = < S, V, a >$$

where S is the set of spatial units, V the set of physical dataset values and a the function that assigns a value to a spatial unit:

$$a: S \rightarrow V$$

The spatial extent of a value is defined by means of the following function e:

$$\begin{aligned} e: \wp(V) &\rightarrow \wp(S) \\ U &\mapsto \{ s \in S \mid a(s) \in U \} \end{aligned}$$

If two datasets, $D1 = < S1, V1, a1 >$ and $D2 = < S2, V2, a2 >$, have to be merged, two similarity functions, $m1$ and $m2$, are defined in the following way, where $|e(U1)|$ indicates the area of the spatial extent of the set $U1$:

$$\begin{aligned} m1: \wp(V1) \text{ x } \wp(V2) &\rightarrow [0,1] \\ (U1,U2) &\mapsto |e(U1) \cap e(U2)| \,/\, |e(U1)| \end{aligned}$$

$$\begin{aligned} m2: \wp(V1) \text{ x } \wp(V2) &\rightarrow [0,1] \\ (U1,U2) &\mapsto |e(U1) \cap e(U2)| \,/\, |e(U2)| \end{aligned}$$

Given $U1 \in \wp(V1)$ and $U2 \in \wp(V2)$, $m1$ returns the quantity of the intersection of the extents of $U1$ and $U2$, $e(U1) \cap e(U2)$ that is contained in the extent of $U1$, $e(U1)$. On the other hand, *m2* returns the quantity of the intersection of extents contained in the extent of *U2*, *e*(*U2*).

Let us consider the following example of the merging of two simple datasets. Note that although these datasets use a raster representation, the method remains exactly the same for a polygon-based vector representation. In the case of rasters, a spatial unit corresponds to a cell, while in the case of vectors, it is a polygon.

s11 *A*	*s12* *B*
s13 *B*	*s14* *C*

Dataset D1

s21 *X*	*s22* *Y*
s23 *Z*	*s24* *Z*

Dataset D2

$D1 = < S1, V1, a1 >$
$S1 = \{ s11, s12, s13, s14 \}$
$V1 = \{ A, B, C\}$

$a1: S1 \rightarrow V1$
$a1(s11) = A$
$a1(s12) = B$
$a1(s13) = B$
$a1(s14) = C$

$e1: \wp(V1) \rightarrow \wp(S1)$
$e1(\{A\}) = \{s11\}$
$e1(\{B\}) = \{s12,s13\}$
$e1(\{C\}) = \{s14\}$
$e1(\{A,B\}) = \{s11,s12,s13\}$
$e1(\{A,C\}) = \{s11,s14\}$
$e1(\{B,C\}) = \{s12,s13,s14\}$
$e1(\{A,B,C\}) = S1$

$m1: \wp(V1) \text{ x } \wp(V2) \rightarrow [0,1]$
$m1(\{A\},\{X\}) = 1$
$m1(\{A\},\{Y\}) = 0$
$m1(\{B\},\{Y\}) = 0.5$
$m1(\{B\},\{Z\}) = 0.5$
$m1(\{A\},\{X,Y\}) = 1$
$m1(\{A,B\},\{X\}) = 0.33$
$m1(\{B,C\},\{Y,Z\}) = 1$
...

$D2 = < S2, V2, a2 >$
$S2 = \{ s21, s22, s23, s24 \}$
$V2 = \{ X, Y, Z\}$

$a2: S2 \rightarrow V2$
$a2(s21) = X$
$a2(s22) = Y$
$a2(s23) = Z$
$a2(s24) = Z$

$e2: \wp(V2) \rightarrow \wp(S2)$
$e2(\{X\}) = \{s21\}$
$e2(\{Y\}) = \{s22\}$
$e2(\{Z\}) = \{s23,s24\}$
$e2(\{X,Y\}) = \{s21,s22\}$
$e2(\{X,Z\}) = \{s21,s23,s24\}$
$e2(\{Y,Z\}) = \{s22,s23,s24\}$
$e2(\{X,Y,Z\}) = S2$

$m2: \wp(V1) \text{ x } \wp(V2) \rightarrow [0,1]$
$m2(\{A\},\{X\}) = 1$
$m2(\{A\},\{Y\}) = 0$
$m2(\{B\},\{Y\}) = 1$
$m2(\{B\},\{Z\}) = 0.5$
$m2(\{A\},\{X,Y\}) = 0.5$
$m2(\{A,B\},\{X\}) = 1$
$m2(\{B,C\},\{Y,Z\}) = 1$
...

If, for instance, we consider 0.9 as the threshold, we can get from *m1* and *m2* the following relations:

- *A* is equivalent to *X* (*A* is subclass of *X*, and *X* of *A*). Obviously, *A* is also subclass of {*X*,*Y*}, and *X* of {*A*,*B*}

- *Y* is subclass of *B*
- {*B*,*C*} is equivalent to {*Y*,*Z*}: they conform two different classifications of the same class
- ...

If the dataset being merged presents an homogenous thematic and spatial structure, a high threshold close to 1 can be chosen. For instance, in one of the evaluation experiments that will be described in Chapter 9, consisting on merging land cover datasets of Eurasia provided by the United States Geology Survey, we have chosen a threshold of 0.95. On the other hand, if datasets present very different thematic categorizations, a lower threshold may be needed to find relations. In other experiments we have chosen thresholds between 0.66 and 0.75 in those cases. Likewise, different scales (and resolutions) make datasets to comprise different spatial units, and a low threshold has also to be considered.

The mapping algorithm of two datasets *D1* and *D2* can be written in the following way:

```
for each $U1 \in \wp(V1)$ do
      for each $U2 \in \wp(V2)$ do
            if $m1(U1, U2)$ > threshold then
                  create suggestion "U1 subclass of U2"
            end if
            if $m2(U1, U2)$ > threshold then
                  create suggestion "U2 subclass of U1"
            end if
      end for
end for
```

To be more precise, subclass relations are not between values but between classes. This way, when a subclass relation is found, two new classes are created for both values (if they had not been previously created in the process) and the subclass relation is set among the new classes. For instance, in the case where we said that *Y* is subclass of *B*, two new classes should be added to the ontology, C_Y and C_B, and the relation "C_Y subclass of C_B" should also be added. In the case of sets different from singletons, such as {*A*,*B*}, a new class, $C_{A \cup B}$, is added and denotes the union of classes C_A and C_B.

The mapping algorithm is then modified in the following way:

```
for each $U1 \in \wp(V1)$ do
      for each $U2 \in \wp(V2)$ do
            if m1( U1, U2 ) > threshold then
                  Let $C_{U1}$ be the class connected to U1
                  Let $C_{U2}$ be the class connected to U2
                        //$C_{U1}$ and $C_{U2}$ are created if necessary
                  create suggestion "$C_{U1}$ subclass of $C_{U2}$"
            end if
```

```
        if m2( U1, U2 ) > threshold then
            Let $C_{U1}$ be the class connected to U1
            Let $C_{U2}$ be the class connected to U2
                //$C_{U1}$ and $C_{U2}$ are created if necessary
            create suggestion "$C_{U2}$ subclass of $C_{U1}$"
        end if
    end for
end for
```

In our particular case, the main objective of the merging method is to merge the dataset structure of values into the overall taxonomy of qualitative thematic classes. Consequently, the result of the merging process is the addition of new classes, connected to dataset values, to the taxonomy. The simplest way to do it is by selecting an already connected dataset from the repository and merge it with the new dataset. New classes will be added to the taxonomy, since they are related through the process to other classes connected to the selected existing dataset, that are already in the ontology. As an example, let us consider the following new dataset *D3*, that is merged in the ontology, once *D1* and *D2* were previously merged.

s31 *M*	*s32* *N*
s33 *M*	*s34* *N*

Dataset D3

If we select *D2* in order to merge *D3* into the taxonomy, we can see that, for instance, C_M would be added having C_X as its subclass, and C_N would be also added having C_Y as its subclass. Note that, since C_A is equivalent to C_X, C_A becomes also subclass of C_M.

Although the added relations are always consistent with the ontology, this merging process is not complete. In the example, no relation is set among C_C and C_N, although C_C should be subclass of C_N. Consequently, for a complete merging, the process should be repeated for any dataset in the repository that may have semantic relations with the new dataset. This way, we first ask the user to manually connect the main theme to an existing class, adding the new class if necessary. Then, any value in any dataset in the repository that is connected to a subclass of this main theme class, is considered for the merging. This assures us to use any semantic related dataset value, although its dataset has a different main theme, but not the unrelated ones. Note that datasets are compared one by one, and consequently, a set U_i of values will only comprise values from one dataset. For instance, following with the example of integrating *D3*, a set containing values *B* (from *D1*) and *Z* (from *D2*) would be meaningless. The process first looks for relations between values from *D3* and *D1* and then between values from *D3* and *D2*.

Hence, our complete mapping algorithm is then modified in the following way, where $D = < S, V, a >$ is the dataset to be merged, *DD* is the set of datasets in the repository having values connected to subclasses of the main theme of *D*, and V_i is the set of values of the *i*-th dataset in *DD* that is connected to a subclass of the main theme of *D*.

```
for each $U \in \wp(V)$ do
      for each $Di \in DD$ do
            for each $Ui \in \wp(Vi)$ do
                  if $m1( U, Ui )$ > threshold then
                        Let $C_U$ be the class connected to $U$
                        Let $C_{Ui}$ be the class connected to $Ui$
                              //$C_U$ and $C_{Ui}$ are created if necessary
                        create suggestion "$C_U$ subclass of $C_{Ui}$"
                  end if
                  if $m2( U, Ui )$ > threshold then
                        Let $C_U$ be the class connected to $U$
                        Let $C_{Ui}$ be the class connected to $Ui$
                              //$C_U$ and $C_{Ui}$ are created if necessary
                        create suggestion "$C_{Ui}$ subclass of $C_U$"
                  end if
            end for
      end for
end for
```

The quality of the results of this algorithm relies on the statistical value of the datasets being merged. They have to contain enough spatial units for each value. Otherwise, the mappings generated by the algorithm may not be semantically valid, and will produce inconsistencies when the mappings are applied to other datasets.

When two datasets have a common value (labelled with the same name), our experience shows that it is not uncommon the case where the value has very different spatial extent in the two datasets. Likewise, some values that a priori are not semantically related, may have high spatial overlapping. In these cases, although the algorithm finds valid spatial relations among values, their translation into mappings with class names may sound meaningless. Apart from possible cartographic errors, this is usually caused because both datasets use different models for the same value. The positive point is that the list of the mappings suggested by the spatial algorithm can provide a valuable help for the expert user to detect inconsistencies. The expert can use this information to define DL models for the involved themes, and to manually map them. In extreme situations where the names do not represent the reality of the dataset at all, the expert may even decide to modify the metadata file to assign them other more appropriate names. We will see in Chapter 10 that the third type of semantic query (integration) can be used to obtain an indication of the degree of inconsistencies between two or more datasets. It is also useful to see on a map the regions with more inconsistencies.

8.2 An optimized version of the algorithm

Comparing the spatial extents of two sets of values from different datasets may be a slow process. In the case of raster datasets, it requires a cell-by-cell comparison of the whole dataset. In the case of vectors, it may require to execute spatial operators between a great number of polygons. Furthermore, the process has to be repeated each time that a value belongs to a set being compared, and thus, it becomes extremely inefficient. As it was discussed in the previous section, 2^{m+n} comparisons of dataset values will have to be done in the worst case, where m and n are the number of values in each dataset. Since the number of spatial units in a dataset may be enormous (for instance some of the datasets that we have used for the evaluation contain more than 100 million spatial units) it is strictly necessary to reduce the number of this type of comparisons.

We propound here a variation of the previous algorithm that uses a *m*x*n* matrix *M*, where *m* and *n* are respectively the number of values in two datasets *D1* and *D2*. *M*(*i*,*j*) contains the area of the overlapping space (measured for instance in hectares) between the *i*-th value of dataset *D1*, and the *j*-th value of dataset *D2*. In the case of vector datasets, only atomic values have to be compared to fill *M*. This way, only *m*x*n* comparisons of spatial extents are required, instead of 2^{m+n} of the previous algorithm. In the case of raster datasets with a common tessellation, they have to be traversed only once to fill *M*, comparing cell by cell. If the dataset being inserted comprises *X* cells, this solution requires *X* cell comparisons, while the previous one required $X{\cdot}2^{m+n}$. Likewise, in the case of vectors, if the dataset being inserted contains *X* polygons, then *X* overlap operations will be needed.

In our tool we have only implemented support for raster datasets. But it has to be noted that there is no restriction on that sense. In fact, once *M* matrix has been filled, the algorithm works in the same way for vectors and rasters. It is even possible to integrate raster and vector datasets together in the repository if the corresponding topological functions are developed.

The following pseudo-code describes the process of filling *M* matrix in the case of two raster datasets *D1* = < *S1*, *V1*, *a1* > and *D2* = < *S2*, *V2*, *a2* >, where the overlapping area is measured in number of cells, and *M* is initialized with 0's, and again *m* and *n* are respectively the number of values in two datasets *D1* and *D2*:

```
for x=0 to m-1 do
      for y=0 to n-1 do
            value1 = D1.getValueAtCell(x, y)
            value2 = D2.getValueAtCell(x, y)
            v1 = dataset1.transformIndex(value1)
            v2 = dataset2.transformIndex(value2)
            M(v1,v2) = M(v1,v2) + 1
      end for
end for
```

where *transformIndex* transforms a value in its position in the list of values of its dataset, that is, an index between 0 and the number of values in the dataset minus 1.

It has to be noted that, since datasets may be big and filling *M* is the most expensive part of the algorithm, the tool permits the user to save and load *M* matrixes. Furthermore, it is also important to note that once *M* matrix has been filled, it makes it possible to compare whatever sets of values without accessing the datasets. This way, the similarity functions *m1* and *m2* are now obtained in the following way, for $U1 \in \wp(V1)$ and $U2 \in \wp(V2)$:

$$|U1| = \sum_{i,v1i \in U1} \sum_{j=0}^{n-1} M(i,j) \qquad\qquad |U2| = \sum_{j,v2j \in U2} \sum_{i=0}^{m-1} M(i,j)$$

$$|U1 \cap U2| = \sum_{i,v1i \in U1} \sum_{j,v2j \in U2} M(i,j)$$

$$m1(U1,U2) = \frac{|U1 \cap U2|}{|U1|} \qquad m2(U1,U2) = \frac{|U1 \cap U2|}{|U2|}$$

As in the previous algorithm if *m1*(*U1*,*U2*) is greater than the threshold, it suggests that *U1* is subclass of *U2*, while if *m2*(*U1*,*U2*) is greater than the threshold, it suggests that *U1* is superclass of *U2*. Consequently, if both functions are greater than the threshold, it suggests that *U1* and *U2* are equivalent.

Two new matrixes *M1* and *M2* are generated from *M*. They represent the ratio of the spatial extent of one value contained in another's. This way, *M1*(*i*,*j*) contains the similarity $m1(\{v_{1i}\},\{v_{2j}\})$, where v_{1i} is the *i*-th value of dataset *D1*, and v_{2j} is the the *j*-th value of dataset *D2*. On the other hand, *M2*(*i*,*j*) contains $m2(\{v_{1i}\},\{v_{2j}\})$. Note that this way, *M1*(*i*,*j*) contains the ratio of the spatial extent of v_{1i} that overlaps the spatial extent of v_{2j}, while *M2*(*i*,*j*) contains the ratio of the spatial extent of v_{2j} that overlaps the spatial extent of v_{1i}. Note also that the sum of the values in a row of *M1* is always 1, while the sum of the values in a column of *M2* is also always 1.

Besides the use of matrixes, the original algorithm is also modified in order to avoid the 2^{m+n} comparisons between every set of values in $\wp(V1)$ with every set in $\wp(V2)$. A greedy approach is proposed, which firstly processes those values having the highest similarities *m1* or *m2*. Furthermore, we consider that any type of these mapping relations (equivalence, subclass or superclass) between two individual values is relevant. However, we assume that only equivalence provides meaningful information for non-atomic sets of values. Note that in this case, an equivalence indicates that a common concept has been specialized in different ways in both ontologies.

The new algorithm first selects the highest value in *M1* and *M2*, and its position (*i*,*j*). The *i*-th value from *D1* (v_{1i}) and the *j*-th value from *D2* (v_{2j}) are considered as the best candidates to be mapped. If *M1*(*i*,*j*) is greater than or equal to the threshold, v_{1i} is suggested to be a subclass of v_{2j}. Likewise, if *M2*(*i*,*j*) is greater than or equal to the threshold, v_{1i} is suggested to be a superclass of v_{2j}. Consequently, if both values are greater than the threshold, an equivalence is suggested between them.

In the case of not obtaining an equivalence, the algorithm adds v_{1i} to *U1* and v_{2j} to *U2*, and it starts the process of searching an equivalence between sets. The maximum among the values in the *i*-th row of *M1* and *j*-th column of *M2* is selected as the best candidate. If the maximum is obtained from *M1* at position (*i*,*k*), then the *k*-th value from *D2* (v_{2k}) is added to *U2*. Otherwise, if the maximum is obtained from *M2* at position (*k*,*j*), the *k*-th value from *D1* (v_{1k}) is added to *U1*. The similarities *m1*(*U1*,*U2*) and *m2*(*U1*,*U2*) are obtained again, and an equivalence is suggested if they both are greater than the threshold. Otherwise, the process continues adding values to either *U1* or *U2* until either an equivalence is obtained or no more values can be added. It has also to be noted that once a value is involved in an equivalence, it is not considered in other sets to be mapped. However, it could be involved in other mappings of atomic values, where the relation will not be an equivalence.

However, a particular case has to be further analyzed. When *U1* and *U2* are suggested as equivalent in this way, the algorithm may miss mappings for values with small spatial extents. Let us consider that value v_{1i} of *D1* , $v_{1i} \notin U1$, has a small spatial extent. Even if there exists a value in *U2*, v_{2j}, such that $m1(\{v_{1i}\},\{v_{2j}\})=1$, note that $m2(\{v_{1i}\},\{v_{2j}\})$ is probably very small. But, perhaps the similarity between *U1* and *U2*, which is already greater that the threshold, would grow if v_{1i} was added.

Formalizing this idea, once an equivalence mapping is found between two sets *U1* and *U2*, the remaining values have to be analyzed. In particular, those values v_1 from *D1* such that $m1(\{v_1\},U2) >$ threshold and those values v_2 from *D2* such that $m2(U1,\{v_2\}) >$ threshold will be considered. This way, if value v_1 satisfies that $m1(U1\cup\{v_1\},U2) > m1(U1,U2)$ and $m2(U1\cup\{v_1\},U2) > m2(U1,U2)$, then v_1 is added to *U1*. Likewise, if value v_2 satisfies that $m1(U1,U2\cup\{v_2\}) > m1(U1,U2)$ and $m2(U1, U2\cup\{v_2\}) > m2(U1,U2)$, then v_2 is added to *U2*.

According to all these considerations, we present below the final algorithm for merging two datasets $D1 = < S1, V1, a1 >$ and $D2 = < S2, V2, a2 >$, where $V1=\{v_{11},...,v_{1m}\}$ and $V2=\{v_{21},...,v_{2n}\}$. It also uses *M* matrix as defined above, while *M1* and *M2* are implemented through a three dimensional matrix *M3*, where the third dimension indicates whether *m1* or *m2* similarity functions have been used (*m1* corresponds to 0 and *m2* to 1). It also uses the sets $EqMVSet_1$ and $EqMVSet_2$ that contain those values in *V1* and *V2* respectively that have already been mapped in an equivalence. Finally, it also uses a boolean matrix *Used* of size *m*x*n*, where *Used*(*i*,*j*) indicates whether v_{1i} and v_{2j} have been related by means of a 1-to-1 mapping, and that is initialized to false. The algorithm generates 1-to-1 mappings (with relations of subclass, superclass or equivalence), as well as M-to-N equivalence mappings, in the following way:

```
repeat
   let maxv be the maximum of M3 and i,j,k its position
            such that Used(i,j)=false
   if maxv > 0 then
      let U1,U2 be two sets: U1 = { v1i } and U2 = { v2j }
      if M3(i,j,0) ≥ threshold and M3(i,j,1) ≥ threshold then
         analyzeRemaining(U1,U2) //note that this may modify U1 and U2
         EqMVSet1 = EqMVSet1 ∪ U1
         EqMVSet2 = EqMVSet2 ∪ U2
         create suggestion "U1 equivalent to U2"
         if |U1|=1 and |U2|=1 then
            Used(i,j)=true
            else if |U1|>1 and |U2|=1 then
               for each v1k∈U1 do
                  Used(k,j)=true
                  create suggestion "v1k subclass of v2j"
               end for
            else
               if |U1|=1 and |U2|>1 then
                  for each v2k∈U2 do
                     Used(i,k)=true
                     create suggestion "v1i superclass of v2k"
                  end for
               end if
            end if
         end if
```

```
        else
            if $M3(i,j,0) \geq$ threshold then
                $Used(i,j)$=true
                create suggestion "$v_{1i}$ subclass of $v_{2j}$"
            else
                if $M3(i,j,1) \geq$ threshold then
                    $Used(i,j)$=true
                    create suggestion "$v_{1i}$ superclass of $v_{2j}$"
                end if
            end if
            if $v_{1i} \notin EqMVSet_1$ and $v_{2j} \notin EqMVSet_2$ then
                $compareSets(U_1,U_2)$
            end if
        end if
    end if
until $maxv$=0
```

The function *compareSets* recursively compares two sets of values and adds more values to the sets until an equivalence is found or no more values can be added:

```
function $compareSets$ ( $U_1 \in \wp(V1)$, $U_2 \in \wp(V2)$ )
        let $maxv0$ be the maximum of $M3$ and $i,j,k$ its position such that
                $k$=0, $v_{1i} \in U_1$, $v_{2j} \notin U_2$ and $v_{2j} \notin EqMVSet_2$
        let $maxv1$ be the maximum of $M3$ and $p,q,r$ its position such that
                $r$=1, $v_{2q} \in U_2$, $v_{1p} \notin U_1$ and $v_{1p} \notin EqMVSet_1$
        if $maxv0 \geq 0$ or $maxv1 \geq 0$ then
                if $maxv0 \leq maxv1$ then
                        $U_1 = U_1 \cup \{ v_{1p} \}$
                else
                        $U_2 = U_2 \cup \{ v_{2j} \}$
                end if
                if $m1(U_1,U_2) \geq$ threshold and $m2(U_1,U_2) \geq$ threshold then
                        $analyzeRemaining(U_1,U_2)$
                        create suggestion "$U_1$ equivalent to $U_2$"
                        $EqMVSet_1 = EqMVSet_1 \cup U_1$
                        $EqMVSet_2 = EqMVSet_2 \cup U_2$
                else
                        $compareSets(U_1,U_2)$
                end if
        end if
end function
```

Finally, *analyzeRemaining* function analyzes whether another value (typically having a small spatial extent) can be added to either U_1 or U_2 increasing both *m1* and *m2* similarities between them. In the case of existing more than one value satisfying this condition, the one that produces the highest similarity is chosen. It has to be noted that this function may modify its parameters U_1 and U_2.

```
function analyzeRemaining ( $U_1 \in \wp(V_1)$, $U_2 \in \wp(V_2)$ )
      let $W_1$ be the set containing all $v_{1i} \in V_1$ such that
            $v_{1i} \notin EqMVSet_1$ , $m1(\{v_{1i}\},U_2) \geq$ threshold,
            $m1(U_1 \cup \{v_{1i}\},U_2) \geq m1(U_1,U_2)$ and $m2(U_1 \cup \{v_{1i}\},U_2) \geq m2(U_1,U_2)$
      let $W_2$ be the set containing each $v_{2j} \in V_2$ such that
            $v_{2j} \in V_2$, $v_{2j} \notin EqMVSet_2$ , $m2(U_1,\{v_{2j}\}) \geq$ threshold,
            $m1(U_1,U_2 \cup \{v_{2j}\}) \geq m1(U_1,U_2)$ and $m2(U_1,U_2 \cup \{v_{2j}\}) \geq m2(U_1,U_2)$
      if $W_1 \neq \emptyset$ then
            let $w_1 \in W_1$ be such that $\forall v_{1k} \in W_1$:
            $m1(U_1 \cup \{v_{1k}\},U_2)+m2(U_1 \cup \{v_{1k}\},U_2) \leq m1(U_1 \cup \{w_1\},U_2)+m2(U_1 \cup \{w_1\},U_2)$
      end if
      if $W_2 \neq \emptyset$ then
            let $w_2 \in W_2$ such that $\forall v_{2k} \in W_2$:
            $m1(U_1,U_2 \cup \{v_{2k}\})+m1(U_1,U_2 \cup \{v_{2k}\}) \leq m1(U_1,U_2 \cup \{w_2\})+m1(U_1,U_2 \cup \{w_2\})$
      end if
      if $W_1 \neq \emptyset$ and ($W_2 = \emptyset$ or $m1(U_1 \cup \{w_1\},U_2) + m2(U_1 \cup \{w_1\},U_2) \geq$
                              $m1(U_1,U_2 \cup \{w_2\}) + m2(U_1,U_2 \cup \{w_2\})$ ) then
            $U_1 = U_1 \cup \{w_1\}$
            analyzeRemaining($U_1$, $U_2$)
      else
            if $W_2 \neq \emptyset$ and ($W_1 = \emptyset$ or $m1(U_1 \cup \{w_1\},U_2) + m2(U_1 \cup \{w_1\},U_2) <$
                              $m1(U_1,U_2 \cup \{w_2\}) + m2(U_1,U_2 \cup \{w_2\})$ ) then
                  $U_2 = U_2 \cup \{w_2\}$
                  analyzeRemaining($U_1$, $U_2$)
            end if
      end if
end function
```

To finish this section, it is worth remarking that this implementation of the spatial algorithm is faster than the terminological algorithm. Furthermore, if the datasets contain enough spatial units for each value to consider the proposed merging as statistically valid, then the spatial algorithm usually produces more reliable results than the terminological one. Nevertheless, the terminological approach, or even the string-based one, are necessary for situations where either we want to merge a normalized vocabulary and no dataset is available for it, or datasets do not provide a good statistical base.

9 Evaluation of mapping algorithms

The objective of this chapter is to evaluate the mapping algorithms that have been presented in Chapter 6 to 8. Since ontology merging/evaluation is still a recent research area, no consolidated framework or benchmark for the evaluation of ontology merging exists yet. As we have already discussed in previous chapters, our methods focus on hierarchical ontologies (taxonomies), and the evaluation initiatives for this case are even less developed than for ontologies with a structure closer to relational schemas. This means that unfortunately the results of our evaluation experiments cannot be compared to other methods. Furthermore, we will also observe that although precision and recall are common measures for evaluating information retrieval methods, they are not appropriate for evaluating ontology merging/alignment.

In the first section of this chapter we will discuss the most relevant initiatives for developing frameworks for ontology merging/alignment evaluation, as well as for providing relaxed definitions of precision and recall measures oriented to ontology merging/alignment. In Section 9.2 we will present a new definition of precision and recall that overcomes the limitations of the existing ones. This is one of the main contributions of this chapter.

Focusing on the specific case of geographic ontologies, the fact that there are no available frameworks for the evaluation of merging/alignment also means that we cannot test our algorithms with known ontologies that produce acknowledged alignments. Given this situation, a first possibility that we considered for the evaluation was asking domain experts to define the reference alignments for certain pairs of ontologies (particularly from datasets). However, defining an alignment between two geographic ontologies can be a difficult task for a human, that often introduces noise. Instead, we have developed other strategies for obtaining a reference merging/alignment to be compared to the results of our algorithms. Sections 9.3 to 9.5 present different experiments aiming at evaluating different aspects of the algorithms. Each experiment is based on a specific strategy for obtaining the reference merging/alignment.

In particular, we have chosen the domain of land cover and land use for these evaluations, since it is clearly qualitative and presents significant differences depending on the different providers, either local, continental or world-wide. These different approaches make the application of merging methods interesting in order to find the relations between the values in different datasets. It is worth clarifying here that land cover and land use are in fact two different domains, and can be defined in the following way (Di Gregorio and Jansen 1998):

- Land cover is the observed (bio)physical cover on the earth’s surface.
- Land use is characterized by the arrangements, activities and inputs people undertake in a certain land cover type to produce, change or maintain it.

This definition of land use establishes a direct link between land cover and the actions of people in their environment. As an example, while “grassland” is a cover term, “rangeland” or “tennis court” refer to the use of a grassland cover. Nevertheless, land cover and land use are often mixed and datasets contain classes from both perspectives. For instance in CORINE (Bossard et al. 2000), which in principle focuses on land cover, we can find classes that clearly refer to uses, as “industrial or commercial units” or “Sport and leisure facilities”.

There are several widely-used land cover/land use datasets, such as CORINE or those developed by the International Geosphere Biosphere Programme (IGBP) or by the United States Geology Survey (USGS). However, there are no available acknowledged translations between their values, and consequently, our merging process cannot be compared to an acknowledged result. In this context, two significant initiatives have to be mentioned. On the one hand, FAO has promoted the Land Cover Classification System (LCCS) (Di Gregorio and Jansen 1998; Di Gregorio 2005) (see also Chapter 3), which permits the definition of land cover classes in terms of independent variables (or classifiers). LCCS has been used in the framework of developing land cover maps of several African countries. It is worth noting that LCCS does not provide a normalized vocabulary of land cover classes, but a system to create new vocabularies, where it is specified how each vocabulary class is determined by different classifiers. On the other hand, the Global Observation of Forest and Land Cover Dynamics[1] (GOFC-GOLD), in conjunction with FAO among others, has an initiative aiming at the harmonization of land cover maps (Herold and Schmullius 2004). In the first stage, they have defined some vocabularies used in national or continental land cover maps (CORINE, USGS, IGBP,...) in LCCS.

In Section 9.3 we will check whether the mapping algorithms behave properly in the best possible condition, where an ontology *O* is merged with an exact copy of itself *O’*. It is worth noting that none of the systems evaluated in EON’2004 obtained both precision and recall of 1 in this test. Furthermore, we will introduce some modifications in the class names of *O’*, in order to check whether the right mappings can be inferred by applying structural rules.

In Section 9.4, we will split a well-known hierarchical ontology in two ontologies: on the one hand the leaf nodes of the tree, and on the other hand the upper levels. We will check if our string-based and thesaurus-based algorithms can reconstruct the original hierarchy from the two separated ontologies. This test will be carried out with the CORINE and the Anderson (Anderson et al. 1976) vocabularies.

In Section 9.5, we will evaluate the algorithms in the context of several land cover maps of Eurasia obtained from the Global Land Cover project[2] of the USGS. Each of these datasets is organized through a different well-known vocabulary (or application

[1] http://www.fao.org/gtos/gofc-gold

[2] http://edcsns17.cr.usgs.gov/glcc/glcc.html

ontology). As it was mentioned above there exist no acknowledged translations among them. We will consider the results of the algorithm based on the spatial distribution of dataset values (*spatial algorithm* hereafter) as the reference alignment, that will be compared to the results of the other two algorithms. It is important to note that these datasets are rasters with a cell size of 1km^2, and contain 169 million spatial units. Since the area of Eurasia contains a great number of units for every considered land cover class, it provides a good statistical base that enables us to consider the results of the spatial algorithm as a valid reference alignment for the evaluation.

It is important to mention that we have conducted other experiments, also with other datasets, to check whether the algorithms behave properly in certain specific cases. Although these experiments have been useful to tune the algorithms, they are not oriented to produce an objective assessment in terms of precision/recall, and consequently are not discussed in this chapter. We include in this chapter only those evaluation experiments that produce objective measures by comparing the results of the algorithms to a reference merging/alignment.

Finally, another clarification has to be made. This chapter evaluates mapping algorithms for the semi-automatic merging method. The manual method is implicitly evaluated since all the merging/alignment operations discussed in this chapter can be carried out in a manual way in the OntoGIS tool, according to the steps described in Chapter 5.

9.1 Related work

Although in the field of information retrieval there exist some well consolidated initiatives for providing a common infrastructure for evaluation, this is still not the case of ontology merging/alignment. In the case of information retrieval, the most significant initiative is the Text Retrieval Conference[3] (TREC), mainly organized by the American National Institute of Standards and Technology (NIST). For each TREC conference, a large set of documents is provided to the participants, as well as a set of queries (called topics in TREC). Their results are organized in “pools” and judged by human experts (more details on this method can be found in (Voorhees 2004) and (Sparck Jones and Rijsbergen 1975)), providing precision/recall measures for each query and for each participant. Furthermore, participants can train their systems with the judgements of previous editions. TREC have several tracks for particular focus areas. Some of them deal with data different from text, as it is the case of TREC Video track, which has become an independent evaluation called TRECVID[4] (more details can also be found in (Over et al. 2005)).

With a similar aim, in the last two years some initiatives have arisen in the field of ontology merging/alignment. They provide several pairs of ontologies that have to be aligned, and the result is compared to a reference alignment generated by experts. The most salient of these evaluation initiatives are the Information Interpretation and Integration Conference[5] (I3CON), organized by NIST, and the Ontology Alignment

[3] http://trec.nist.gov

[4] http://www.itl.nist.gov/iaui/894.02/projects/trecvid/

[5] http://atl.external.lmco.com/projectes/ontology/i3con.html

Contest at the Evaluation of Ontology-based Tools Workshop[6] (EON), held in the International Semantic Web Conference. Unfortunately all the ontologies in the benchmarks provided in these two conferences present a structure similar to a relational schema, with multiple relations between classes through roles, and none of them focus on a hierarchical structure (taxonomy) of classes. On the other hand, particularly related to our work is a panel on web directories alignment that was held in EON 2005. Its objective was to evaluate several tools in the context of aligning ontologies related to web directories of Yahoo![7], Google[8] and Looksmart[9]. These ontologies present a similar structure to our thematic geographic ones, since they consist on a hierarchy of classes with neither attributes nor other relations. The reference alignments do not contain all the possible mappings, but just some particular ones that are in principle difficult to find. These way, precision cannot be computed and the evaluation focuses on recall. More details can be found in (Euzenat et al. 2005), and particularly in (Avesani et al. 2005) where the methodology of the evaluation is discussed. However, the provided reference alignments are highly questionable and full of logical contradictions. As a clear example of this, let us observe the following case study, that is extracted from the merging of directories related to education of Google and Yahoo!. The following are two fragments of respectively Google and Yahoo! ontologies on education:

```
Top                                 Top
 Reference                           Education
  Education                           K-12 (55032)
   K through 12 (4244)                 Schools  (26177)
    Private Schools (330)               Democratic (16)
     Sudbury Valley Model (26)           Sudbury Model (11)
     ...                                ...
     Waldorf Schools (134)              Waldorf (35)
      Australia (13)
      Canada (8)
      Ireland (3)
      New Zealand (3)
      United Kingdom (16)
      United States (75)
```

Figure 31. Part of Google (left) and Yahoo! (right) directories for education. In brackets the number of documents of each category

The mappings that are provided in the reference alignment corresponding to the previous classes are:

```
      Private Schools equivalent to Schools
      Private Schools equivalent to Sudbury Model
      United States equivalent to Schools
      Sudbury Valley Model equivalent to Sudbury Model
```

We can see that *United States* is clearly not equivalent to *Schools*, since the latter contains 26,177 documents many of them from other countries, and the majority of them corresponding to other types of schools (different from *Waldorf schools*). Likewise, *Private schools* is not equivalent to *Sudbury Model*, since *Private schools*

[6] http://co4.inrialpes.fr/align/Contest

[7] http://dir.yahoo.com

[8] http://www.google.com/dirhp

[9] http://www.looksmart.com/r?country=uk

comprise other types of schools. In fact, *Private Schools* is not equivalent to *Schools* either, since there are other types of schools that are considered subclasses of *Schools* and that are subclasses of *K through 12* but not of *Private schools*. This situation drives us to dismiss the possibility of evaluating our algorithms through this benchmark. In fact, it may be worth noting that the systems evaluated in this panel obtained very low recall measures, all below 1/3.

Going back to I3CON and the main panel of EON, it is important to note that they both present two significant limitations. On the one hand, they only focus on 1-to-1 mappings where the relation is equivalence, and other types of mappings are not considered. On the other hand, evaluations are carried out in terms of precision/recall, which are good indicators for information retrieval but not for alignments. In ontology alignment, precision and recall measures provide a crisp evaluation that considers that each proposed mapping is either valid or not. However, it does not evaluate how close the proposed mapping is to the expected one. For instance, let us consider that a system provides a mapping "*A* equivalent to *X*", and the expected result is "*A* equivalent to *Y*". In the case of *X* being a subclass of *Y*, the obtained mapping can be considered as better than the one in the case where *X* and *Y* are not related.

To avoid this last problem, (Ehrig and Euzenat 2005) proposes a relaxed definition of precision and recall. While the standard precision (*P*) and recall (*R*) are computed through the following definitions, where *A* is a proposed alignment and *B* the reference one:

$$P(A,B) = \frac{|A \cap B|}{|A|} \qquad R(A,B) = \frac{|A \cap B|}{|B|}$$

the relaxed definitions depend on a certain function *w*, called overlapping proximity, in the following way:

$$P_w(A,B) = \frac{w(A,B)}{|A|} \qquad R_w(A,B) = \frac{w(A,B)}{|B|}$$

where *w* function is defined in terms of another function σ, called proximity function, that provides an indicator for each mapping in the alignment of how close it is to the reference:

$$w(A,B) = \sum_{(a,b)\in M(A,B)} \sigma(a,b)$$

where *M*(*A*,*B*) is the set of pairs (*a*,*b*) ∈ *A*x*B* such that *a* and *b* refer to the same mapping, and where both *a* and *b* can only appear in one pair. We can observe that *w* and *M* definitions depend on each other. They propose a method that generates all the possible combinations of pairs and obtains the one that maximizes *w*(*A*,*B*). However, this only considers the case of 1-to-1 mappings. If many-to-many mappings were considered, $2^{|A|}$x$2^{|B|}$ pairs of mappings would have to be compared to obtain *M*.

Let us then consider the case of 1-to-1 mappings, where *a* is the mapping in the obtained alignment "a_1 equivalent to a_2", and b is the mapping in the reference alignment "b_1 equivalent to b_2", where a_1 and b_1 are classes in the first ontology and a_2 and b_2 classes in the second one. The standard precision and recall are obtained if σ is defined as follows:

$\sigma(a,b)=1$ if $a_1=b_1$ and $a_2=b_2$
$\sigma(a,b)=0$ otherwise

Other definitions for the proximity function are provided. In the case of the so-called symmetric proximity, σ is defined as follows:

$\sigma(a,b)=1$ if $a_1=b_1$ and $a_2=b_2$
$\sigma(a,b)=0.5$ if ($a_1=b_1$ and a_2 is either subclass or superclass of b_2) or ($a_2=b_2$ and a_1 is either subclass or superclass of b_1)
$\sigma(a,b)=0$ otherwise

In another case, they define two different proximity functions, σ_p and σ_r, one for precision and another for recall, in the following way:

$\sigma_p(a,b)=1$ if ($a_1=b_1$ and $a_2=b_2$) or if ($a_1=b_1$ and a_2 is superclass of b_2) or ($a_2=b_2$ and a_1 is subclass of b_1)
$\sigma_p(a,b)=0.5$ if ($a_1=b_1$ and a_2 is subclass of b_2) or ($a_2=b_2$ and a_1 is superclass of b_1)
$\sigma_p(a,b)=0$ otherwise

$\sigma_r(a,b)=1$ if ($a_1=b_1$ and $a_2=b_2$) or if ($a_1=b_1$ and a_2 is subclass of b_2) or ($a_2=b_2$ and a_1 is superclass of b_1)
$\sigma_r(a,b)=0.5$ if ($a_1=b_1$ and a_2 is superclass of b_2) or ($a_2=b_2$ and a_1 is subclass of b_1)
$\sigma_r(a,b)=0$ otherwise

The different definitions of σ are limited to 1-to-1 equivalence mappings in both the proposed and the reference alignments. Instead, our merging algorithms usually produce mappings with other relations (subclass, superclass or common superclass), and a class in one ontology can be connected to more that one class in the other. Consequently these measures cannot be used. To solve this problem, in the next section we propound a new definition of precision and recall that considers all the types of relations.

9.2 Defining precision and recall for ontology merging/alignment

We have observed in the previous section that the traditional definitions of precision and recall only consider whether or not a given obtained mapping is exactly equal to one in the reference alignment. However they do not offer any indication about how close the proposed mapping is to the expected one. We have also discussed the relaxed definitions of (Ehrig and Euzenat 2005) which are restricted to 1-to-1 equivalence mappings. In this section we present a new relaxed definition of precision and recall for the context of ontology merging/alignment. It is based on providing a precision and a recall measures for each mapping considering all the axioms that can be inferred from both the obtained and the reference mapping. Let us firstly discuss how partial precision and recall for a given mapping are obtained and then we will focus on how the global measures are obtained from the partial ones.

9.2.1 Partial precision and recall for a 1-to-1 mapping

Let us consider a mapping *M'*=<*A'*, *B'*, *rel'*> from the reference alignment, and a mapping *M*=<*A*, *B*, *rel*> from the obtained alignment, where *A'* and *A* are classes from one ontology and *B'* and *B* from the other one. We consider that mappings *M* and *M'* are related when either *A*=*A'* or *B*=*B'* (or both). Let us firstly consider the case of *A*=*A'* (in this subsection we will denote by *A* both classes) and both *rel* and *rel'* are of type subclass. We discuss now how to obtain partial precision, *p*(*M*,*M'*), and recall, *r*(*M*,*M'*), in this case, and then we will generalize their definitions for the other cases.

Let us also consider that *B* has *m* superclasses, $B_{S1},...,B_{Sm}$, and *n* subclasses, $B_{s1},...,B_{sn}$, while *B'* has *m'* superclasses, $B'_{S1},...,B'_{Sm'}$, and *n'* subclasses $B'_{s1},...,B'_{sn'}$. Note that superclasses and subclasses of *B* comprise *B* itself, and that superclasses and subclasses of *B'* also comprise *B'* itself. From the mapping *M'* we can infer the following *m'* axioms: $A \sqsubseteq B'_{S1},..., A \sqsubseteq B'_{Sm'}$, while from the mapping *M* we can infer the following *m* axioms: $A \sqsubseteq B_{S1},..., A \sqsubseteq B_{Sm}$. Let us call these set of inferred axioms *X*(*M'*) and *X*(*M*) respectively. *p*(*M*,*M'*) and *r*(*M*,*M'*) are obtained in terms of the number of common inferred axioms in *X*(*M'*) and *X*(*M*) in the following way:

$$p(M,M') = \frac{|X(M) \cap X(M')|}{|X(M)|}, \qquad r(M,M') = \frac{|X(M) \cap X(M')|}{|X(M')|}$$

If *B* is equivalent to B', *p*(*M*,*M'*) and *r*(*M*,*M'*) are clearly 1. If *B* is subclass of *B'*, $X(M') \subseteq X(M)$ and consequently, *p*(*M*,*M'*)=*m'*/*m* and *r*(*M*,*M'*)=*m'*/*m'*=1. Likewise, if *B* is superclass of *B'*, $X(M) \subseteq X(M')$ and consequently, *p*(*M*,*M'*)=*m*/*m*=1 and *r*(*M*,*M'*)=*m*/*m'*.

Note that the case of *rel*=*rel'*=superclass is equivalent to this one. In the case of *rel*=*rel'*=equivalence, the mapping *A* equivalent to *B* infers axioms for both subclasses and superclasses of *B*: $A \sqsubseteq B_{S1},..., A \sqsubseteq B_{Sm}$ and $A \sqsupseteq B_{s1},..., A \sqsupseteq B_{sn}$, and similarly for *A* equivalent to *B'*. This way, if *B* is superclass of *B'*, *p*(*M*,*M'*)=(*m*+*n'*)/(*m*+*n*) and *r*(*M*,*M'*)= (*m*+*n'*)/(*m'*+*n'*), and likewise, if *B* is subclass of *B'*, *p*(*M*,*M'*)=(*m'*+*n*)/(*m*+*n*) and *r*(*M*,*M'*)= (*m'*+*n*)/(*m'*+*n'*). Observe that *m*+*n*=*m'*+*n'*, and consequently precision and recall are the same.

This can be generalized for any *rel* and *rel’* and for any relation between B and B', as the following table shows. Note that this is the case where M is related to M' because $A=A'$. In the case where $B=B'$, precision and recall would be determined analogously.

rel'	***rel***	***B* related to *B’***	**Partial Precision**	**Partial Recall**	**Note**
Subclass	Subclass	Subclass	m'/m	$m'/m'=1$	
Subclass	Subclass	Superclass	$m/m=1$	m/m'	
Subclass	Subclass	Equivalent	$m/m=1$	$m/m'=1$	$m=m', n=n'$
Subclass	Superclass	Subclass	$0/n=0$	$0/m'=0$	
Subclass	Superclass	Superclass	$0/n=0$	$0/m'=0$	
Subclass	Superclass	Equivalent	$0/n=0$	$0/m'=0$	$m=m', n=n'$
Subclass	Equivalent	Subclass	$m'/(m+n)$	$m'/m'=1$	
Subclass	Equivalent	Superclass	$m/(m+n)$	m/m'	
Subclass	Equivalent	Equivalent	$m/(m+n)$	$m/m'=1$	$m=m', n=n'$
Superclass	Subclass	Subclass	$0/m=0$	$0/n'=0$	
Superclass	Subclass	Superclass	$0/m=0$	$0/n'=0$	
Superclass	Subclass	Equivalent	$0/m=0$	$0/n'=0$	$m=m', n=n'$
Superclass	Superclass	Subclass	$n/n=1$	n/n'	
Superclass	Superclass	Superclass	n'/n	$n'/n'=1$	
Superclass	Superclass	Equivalent	$n/n=1$	$n/n'=1$	$m=m', n=n'$
Superclass	Equivalent	Subclass	$n/(m+n)$	n/n'	
Superclass	Equivalent	Superclass	$n'/(m+n)$	$n'/n'=1$	
Superclass	Equivalent	Equivalent	$n/(m+n)$	$n/n'=1$	$m=m', n=n'$
Equivalent	Subclass	Subclass	m'/m	$m'/(m'+n')$	
Equivalent	Subclass	Superclass	$m/m=1$	$m/(m'+n')$	
Equivalent	Subclass	Equivalent	$m/m=1$	$m/(m'+n')$	$m=m', n=n'$
Equivalent	Superclass	Subclass	$n/n=1$	$n/(m'+n')$	
Equivalent	Superclass	Superclass	n'/n	$n'/(m'+n')$	
Equivalent	Superclass	Equivalent	$n/n=1$	$n/(m'+n')$	$m=m', n=n'$
Equivalent	Equivalent	Subclass	$(m'+n)/(m+n)$	$(m'+n)/(m'+n')$	
Equivalent	Equivalent	Superclass	$(m+n')/(m+n)$	$(m+n')/(m'+n')$	
Equivalent	Equivalent	Equivalent	$(m+n)/(m+n)=1$	$(m+n)/(m'+n')=1$	$m=m', n=n'$

Table 3. Partial precision and recall for a mapping M=<A, B, rel> with respect to a reference mapping *M’=<A’,B’,rel’>*

Note that we have not included common superclass relations in this table. Let us recall that a mapping of type “A and B have common superclass C” has a mapping “C subclass of D” associated which determines where the class C is inserted in the repository taxonomy (note that C may be equal to D). This way, the common superclass mapping can in fact be evaluated through the mapping “A subclass of D”.

9.2.2 Global precision and recall for a merging/alignment

Let us discuss now how the global precision and recall for the whole merging/alignment are obtained. The first issue is, given an obtained mapping *M=<A,B,rel>*, which reference mapping *M'=<A',B',rel'>* is selected for it. This selection is carried out according to this algorithm:

```
let $S_A$ be the set of mappings in the reference alignment
                                                where $A'=A$
if $|S_A|=1$ then
      $M'$ = the mapping in $S_A$
else if $|S_A|>1$ then
      let $S_{Ar}$ be the set of mappings in the reference alignment
                                                where $A'=A$ and $rel=rel'$
      if $|S_{Ar}|=1$ then
            $M'$ = the mapping in $S_{Ar}$
      else
            $M'$ = the mapping in $S_{Ar}$ such that $B'$ is closer to $B$
      end if
else //$|S_A|=0$
      let $S_B$ be the set of mappings in the reference alignment
                                                where $B'=B$
      if $|S_B|=1$ then
            $M'$ = the mapping in $S_B$
      else if $|S_B|>1$ then
            let $S_{Br}$ be the set of mappings in the ref. align.
                                                where $B'=B$ and $rel=rel'$
            if $|S_{Br}|=1$ then
                  $M'$ = the mapping in $S_{Br}$
            else
                  $M'$ = the mapping in $S_{Br}$ s.t. $A'$ is closer to $A$
            end if
      else //$|S_B|=0$
            $M'$ = an empty mapping: $|X(M)\cap X(M')|=0$,
                                                $p(M,M')=0$, $r(M,M')=0$
      end if  end if
end if  end if
```

Let μ be the obtained alignment that comprises *k* mappings $M_1,...,M_k$. And let μ' be the reference alignment that contains *k'* mappings $M'_1,...,M'_{k'}$. Given an obtained mapping M_i, let us denote by M'_{ri} the reference mapping related to M_i. Global precision *P* and recall *R* are defined as follows:

$$P(\mu,\mu') = \frac{\left|\bigcup_{i=1}^{k}\left(X(M_i)\cap X(M'_{ri})\right)\right|}{\left|\bigcup_{i=1}^{k}\left(X(M_i)\right)\right|}, \qquad R(\mu,\mu') = \frac{\left|\bigcup_{i=1}^{k}\left(X(M_i)\cap X(M'_{ri})\right)\right|}{\left|\bigcup_{i=1}^{k'}\left(X(M'_i)\right)\right|}$$

Considering that both μ and μ' do not contain redundant mappings, the cardinal of the union is the sum of the cardinals:

$$P(\mu,\mu') = \frac{\sum_{i=1}^{k} \left| X(M_i) \cap X(M'_{ri}) \right|}{\sum_{i=1}^{k} \left| X(M_i) \right|}, \qquad R(\mu,\mu') = \frac{\sum_{i=1}^{k} \left| X(M_i) \cap X(M'_{ri}) \right|}{\sum_{i=1}^{k'} \left| X(M'_i) \right|}$$

We can observe that these measures of precision and recall present three relevant properties:

1. The same results are obtained whether equivalences are represented as a single mapping (*A equivalent to B*) or two subclass/superclass mappings (*A subclass of B* and *A superclass of B*).
2. In the case of *M=M'* (*A=A'*, *B=B'*, *rel=rel'*), partial precision and recall will be 1. Analogously, if μ=μ', global precision and recall will also be 1.
3. This method gives more importance to the mappings that affect a bigger number of classes, as in the cases of adding a subclass at the bottom of the hierarchy or a superclass at the top.

Note that property 3 is particularly relevant in the case of our string-based and terminological algorithms, since they are based on mapping restrictions. Those classes that generate a bigger number of mapping restrictions have a bigger impact in the merging process.

To clarify the definitions we provide a simple example. Let us consider the following two ontologies:

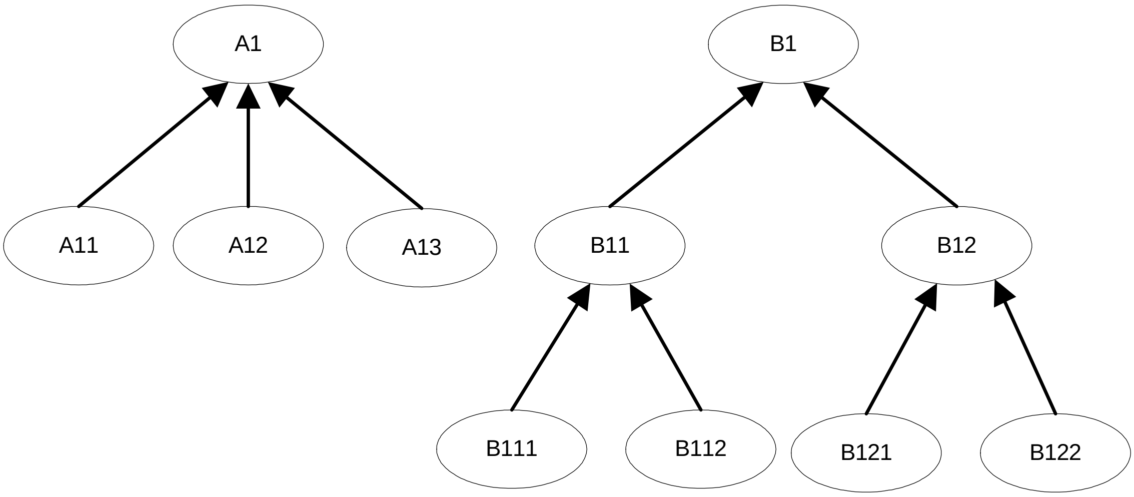


Figure 32. Example of two simple ontologies to be merged

Let us consider that the obtained and reference mergings (μ and μ') contain the following mappings:

μ

M_1: *A11* subclass of *B11*
M_2: *A13* equivalent to *B12*

μ'

M'_1: *A11* equivalent to *B111*
M'_2: *A12* equivalent to *B112*
M'_3: *A13* superclass of *B121*

We can observe that M_1 will be related to M'_1, while M_2 will be related to M'_3. We obtain:

$X(M_1) = \{ A11 \sqsubseteq B1, A11 \sqsubseteq B11 \}$

$X(M_2) = \{ A13 \sqsubseteq B1, A13 \sqsubseteq B12, A13 \sqsupseteq B12, A13 \sqsupseteq B121, A13 \sqsupseteq B122 \}$

$X(M'_1) = \{ A11 \sqsubseteq B1, A11 \sqsubseteq B11, A11 \sqsubseteq B111, A11 \sqsupseteq B111 \}$

$X(M'_2) = \{ A12 \sqsubseteq B1, A12 \sqsubseteq B11, A12 \sqsubseteq B112, A12 \sqsupseteq B112 \}$

$X(M'_3) = \{ A13 \sqsupseteq B121 \}$

$X(M_1) \cap X(\mathrm{M'}_1) = \{ A11 \sqsubseteq B1, A11 \sqsubseteq B11 \}$

$X(M_2) \cap X(\mathrm{M'}_3) = \{ A13 \sqsupseteq B121 \}$

$$P(\mu, \mu') = \frac{2+1}{2+5} = \frac{3}{7} \ , \qquad R(\mu, \mu') = \frac{2+1}{4+4+1} = \frac{1}{3}$$

9.2.3 Partial precision and recall for a many-to-many equivalence mapping

We are interested not only in 1-to-1 mappings but also in those mappings where the union of a set of classes in one ontology is equivalent to the union of another set of classes in the other ontology. Our relaxed definitions of precision and recall based on inferred axioms can be generalized to deal with these many-to-many equivalence mappings.

Let us consider the example of Figure 33 that contains a mapping reference *M'* indicating that the union of *A1* and *A2* is equivalent to the union of *B1*, *B2* and *B3*.

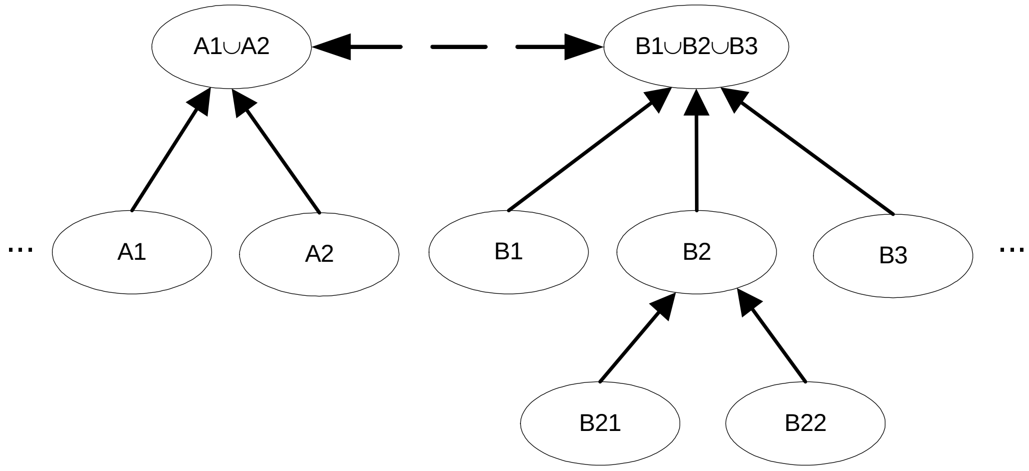


Figure 33. Example of a many-to-many equivalence mapping

Let us now consider that the obtained mapping *M* is *A1* equivalent to the union of *B1* and *B21*. Note that the union of *B1*, *B2* and *B3* has also as subclasses the union of *B1* and *B2*, the union of *B1* and *B3* and the union of *B2* and *B3*. Likewise, note that the union of *B1* and *B2* is also superclass of the union of *B1* and *B21*, the union of *B1* and *B22*, and both *B21* and *B22*.

This way, we can infer the following axioms, *X*(*M*), from the obtained mapping:

A1 superclass of Union(*B1*,*B21*)
A1 superclass of *B1*
A1 superclass of *B21*
Union(*A1*,*A2*) superclass of *B1*
Union(*A1*,*A2*) superclass of *B21*
A1 subclass of Union(*B1*,*B21*)
A1 subclass of Union(*B1*,*B2*)
A1 subclass of Union(*B1*,*B2*,*B3*)

On the other hand, from the reference mapping, we can infer the following axioms, *X*(*M’*) (in bold those common with *X*(*M*)):

Union(*A1*,*A2*) superclass of Union(*B1*,*B2*, *B3*)
Union(*A1*,*A2*) superclass of Union(*B1*,*B2*)
Union(*A1*,*A2*) superclass of Union(*B1*,*B3*)
Union(*A1*,*A2*) superclass of Union(*B2*,*B3*)
Union(*A1*,*A2*) superclass of Union(*B1*,*B21*)
Union(*A1*,*A2*) superclass of Union(*B1*,*B22*)
Union(*A1*,*A2*) superclass of Union(*B21*,*B3*)
Union(*A1*,*A2*) superclass of Union(*B22*,*B3*)
Union(*A1*,*A2*) superclass of *B1*
Union(*A1*,*A2*) superclass of *B2*
Union(*A1*,*A2*) superclass of *B3*
Union(*A1*,*A2*) superclass of *B21*
Union(*A1*,*A2*) superclass of *B22*
Union(*A1*,*A2*) subclass of Union(*B1*,*B2*, *B3*)
***A1* subclass of Union(*B1*,*B2*, *B3*)**
A2 subclass of Union(*B1*,*B2*, *B3*)

This way, we can see that $|X(M)|=8$, $|X(M')|=16$ and $|X(M\cap M')|=4$. Consequently, we can obtain partial precision and recall:

$p(M,M') = 4/8 = 1/2$, $r(M,M') = 4/16 = 1/4$

It is important to remark that when many-to-many relations are provided, the intersection of the inferred axioms from different mappings may not be empty. Consequently, global precision and recall are defined in the same way:

$$P(\mu,\mu') = \frac{\left|\bigcup_{i=1}^{k}\left(X(M_i)\cap X(M'_{ri})\right)\right|}{\left|\bigcup_{i=1}^{k}\left(X(M_i)\right)\right|}, \qquad R(\mu,\mu') = \frac{\left|\bigcup_{i=1}^{k}\left(X(M_i)\cap X(M'_{ri})\right)\right|}{\left|\bigcup_{i=1}^{k'}\left(X(M'_i)\right)\right|}$$

But observe that the cardinal of the union is not the sum of the cardinals (as it was in the previous section).

9.2.4 Contextual definition of precision and recall

As we have stated above, the definitions of precision and recall depend on where the classes involved in mappings are placed in the hierarchy. However, although we consider that this is a significant factor to be considered in the evaluation, particularly in the case of our string and terminological algorithms, an evaluation independent from the place of classes in the hierarchy may sometimes be useful.

For instance, let us consider two ontologies that share a subtree of classes, but where one of the ontologies provides higher detail for their superclasses, i.e. the common subtree is at different depth in each ontology. If we separately merge these two ontologies with a third one, precision and recall for the classes in the subtrees will probably be different since they have a different number of superclasses. The same can be said if one of the two ontologies provides greater detail through subclasses of the leaf nodes in the common subtree.

For a mapping <*A*,*B*,*rel*> compared to a reference mapping <*A*,*B'*,*rel'*>, we can define a new precision and recall that are also based on the inferred axioms, but now restricted to a context of *B* and *B'*. This context comprises *B*, *B'* and all the classes "between" them in the same branch of the hierarchy: classes being superclasses of *B* and subclasses of *B'*, or vice versa. Note that in fact, local precision (p_c) and recall (r_c) are defined in the same way but now $X_c(M)$ only comprises those axioms that affect classes in the context:

$$p_C(M,M') = \frac{|X_C(M) \cap X_C(M')|}{|X_C(M)|}, \qquad r_C(M,M') = \frac{|X_C(M) \cap X_C(M')|}{|X_C(M')|}$$

Since this approach aims at giving all the mappings the same weight, global measures, P_c and R_c, are now obtained as the average of the partial ones:

$$P_C(\mu,\mu') = \frac{\sum_{i=1}^{k} p_C(M_i, M'_{ri})}{k}, \qquad R_C(\mu,\mu') = \frac{\sum_{i=1}^{k} r_C(M_i, M'_{ri})}{k'}$$

If we calculate P_c and R_c for the example given in subsection 9.2.2, we obtain:

$X_c(M_1) = \{ A11 \sqsubseteq B11 \}$

$X_c(M_2) = \{ A13 \sqsubseteq B12, A13 \sqsupseteq B12, A13 \sqsupseteq B121 \}$

$X_c(M'_1) = \{ A11 \sqsubseteq B11, A11 \sqsubseteq B111, A11 \sqsupseteq B111 \}$

$X_c(M'_3) = \{ A13 \sqsupseteq B121 \}$

$X_c(M_1) \cap X(M'_1) = \{ A11 \sqsubseteq B1 \}$

$X_c(M_2) \cap X(M'_3) = \{ A13 \sqsupseteq B121 \}$

$p_c(M_1,M'_1)=1,\ r_c(M_1,M'_1)=1/3$

$p_c(M_2,M'_3)=1/3,\ r_c(M_2,M'_3)=1$

$P_c(\mu,\mu')=2/3,\ R_c(M_1,M'_1)=2/3$

9.3 Merging copies with and without modifications

In this section we will analyze the mappings obtained by the mapping algorithms in the ideal case of merging an ontology *O* with an exact copy of itself, *O'* (where *O'* is merged into *O*).

In the case of the string-based algorithm, it can easily be proved that for each pair of classes <*A', A*>, where $A'\in O'$ and $A\in O$, where the names of *A'* and *A* are equal, the algorithm suggests an equivalence mapping, with no need to create the new class (only connection). Mapping restrictions are propagated correctly, since subclasses of *A'* should only be mapped to subclasses of *A*, and superclasses of *A'* to superclasses of *A*.

In the case of the terminological algorithm, since the string-based similarity between *A'* and *A* is 1, the result is the same as the obtained by the string-based algorithm, with no need to search synonyms, hyponyms and hypernyms.

In the case of the spatial algorithm, since the spatial extent of *A'* will be exactly the same as the spatial extent of *A*, an equivalence mapping will also be suggested.

Consequently, we can observe that the three mapping algorithms obtain a precision and recall of 1. It has to be remarked that none of the systems evaluated in EON'2004 obtained both precision and recall of 1 in this test.

We will now introduce modifications in the names of some classes in the copy of the ontology. For instance, let us consider that the original ontology *O* is the CORINE land cover vocabulary. CORINE is structured in a three-levels hierarchy of themes. The complete structure of CORINE can be found in Appendix C. We now introduce a change in the name of a leaf node in *O'*, for instance the name of the class *Discontinuous urban fabric* (1.1.2) is replaced by *X*.

We can observe that both the string-based and terminological mapping algorithms will determine the equivalence mappings for the rest of the classes. Class *X* has only one class in *O* satisfying its mapping restrictions, *Discontinuous urban fabric*. Although their score is 0, the structural rule “common parent and brothers” can be applied, and consequently, a mapping “*X* equivalent to *Discontinuous urban fabric*” will be inferred.

Let us now consider that the change of the class name is not at the bottom layer. For instance, let us consider that the name of *Urban fabric* (1.1) is replaced by *Y*. In this case, all the other mappings will be correctly generated, and the structural rule “common parent and children” will be applied to *Y*. Note that “common parent and brothers” could also be applied here. Consequentlythe mapping “*Y* equivalent to *Urban fabric*” will be inferred.

Structural rules will also be applied if more than one change is introduced in *O'*. Only in the case of more than one change introduced among the direct subclasses of a given class the right mappings will not be inferred by the structural rule “common parent and brothers”. Even in this case, the rule “common parent and children” could be applied if the modifications have not been introduced in either top or leaf nodes.

We can observe that structural rules provide a good way to overcome possible spelling mistakes in the names of classes. And also to deal with synonyms that do not appear in the terminological base.

Finally, since class names do not affect the spatial algorithm, this would obtain the right equivalence mappings if any number of modifications is introduced in the names of classes.

9.4 Merging hierarchical land cover/land use vocabularies

In this section, we will split a well-known hierarchical ontology in two ontologies: on the one hand the leaf nodes of the tree; on the other hand the other nodes that conform a hierarchy of two or three levels. The objective of the experiment is to evaluate if our string-based and thesaurus-based algorithms can reconstruct the original hierarchy from the two separated ontologies. Consequently, the reference merging in this case obtains the original ontology.

This test have been carried out with two different hierarchical ontologies: the CORINE land cover vocabulary and the Anderson vocabulary for land cover and land use (Anderson et al. 1976). While CORINE is mainly used in Europe, the Anderson vocabulary is particularly widely used in North America, where the US Geology Survey uses a variation in its land cover maps. The complete structure of both the CORINE and the Anderson vocabularies can be found in Appendix C. CORINE is structured in three levels, while Anderson in two.

The following table presents the results of the experiment with the Anderson vocabulary. The list of detailed mappings can be found in Appendix D. Since all the mappings in both the obtained and the reference alignment are of type subclass, standard precision and recall are equal to our relaxed measures.

	string-based (α=0.75)	**WordNet without meronyms**	**WordNet with meronyms**	**GEMET**
Correct mappings	18	26	26	25
Wrong mappings	0	0	0	0
No relation mappings	19	11	11	12
Total mappings	37	37	37	37
Precision	**1.00**	**1.00**	**1.00**	**1.00**
Recall	**0.49**	**0.70**	**0.70**	**0.68**

Table 4. Precision and recall for the experiment of merging lower and upper levels of the Anderson vocabulary

In this test, precision is 1 in all the cases. But the use of the terminological algorithm provides better recall results. Note that GEMET and WordNET provide similar results. In the specific case of WordNet, the use of meronyms do not change the results. We have conducted the same test with PROMPT and no mapping has been found.

In the case of CORINE, it contains one more level than Anderson. Some of the obtained mappings relate classes from the bottom level directly to classes at level 1, skipping the relations to the classes at level 2. For instance, the class *Salt marshes*, is directly mapped in the terminological as subclass of *Wetlands*, skipping the relation with *Coastal wetlands*, which is a subclass of *Wetlands*. Furthermore, some relations are of type “common superclass”.

The following table shows the results of the experiment, and the obtained relaxed precision and recall according to our definitions. The list of detailed mappings can be found in Appendix D.

	string-based (α=0.75)	**WordNet without meronyms**	**WordNet with meronyms**	**GEMET**
Obtained mapping equal to reference mapping	12	15	15	13
Obtained mapping intersects reference mapping	6	9	9	13
Wrong mappings	2	6	7	3
No relation mappings	24	14	13	15
Total mappings	44	44	44	44
Relaxed precision	0.92	0.82	0.79	0.89
Relaxed recall	0.37	0.47	0.47	0.47
Contextual relaxed precision	0.78	0.73	0.71	0.78
Contextual relaxed recall	0.40	0.49	0.49	0.50

Table 5. Precision and recall for the experiment of merging lower and upper levels of the CORINE vocabulary

The best results are obtained using GEMET, with a relaxed precision very close to 90% and a recall around 50%. Note also that the fact of using WordNet increases recall with respect to the string-based approach, but decreases precision. As in the case of Anderson, the use of meronyms has little effect on the results. Finally, it is worth mentioning that contextual precision measure is lower than the generic one, since we discard many correct inferred axioms at level 1. The same experiment with PROMPT obtained only one (correct) mapping. Although precision is 1, recall is almost zero: 0.02.

9.5 Merging Eurasia land cover/land use datasets

We use in this case a set of land cover/land use datasets from the Global Land Cover project of the USGS[10]. More specifically, we use the Eurasia Land Cover Characteristics Data Base, which consists of different land cover/land use maps of Eurasia, each with a different thematic classification. All these datasets have a common tessellation ($169 \cdot 10^6$ cells), with the same resolution (1 pixel = 1 km^2) and projection (Lambert azimuthal equal area, optimized for Europe).

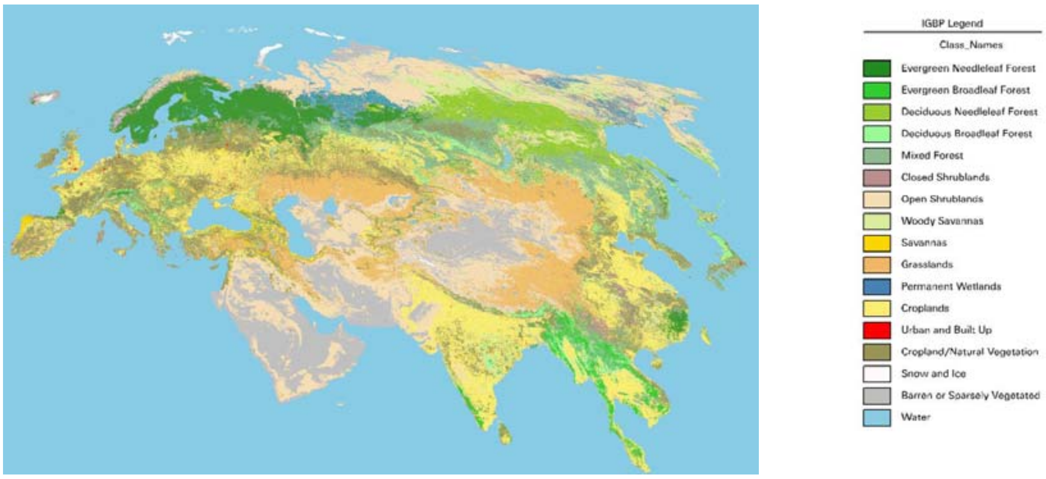

IGBP Legend
Class_Names
Evergreen Needleleaf Forest
Evergreen Broadleaf Forest
Deciduous Needleleaf Forest
Deciduous Broadleaf Forest
Mixed Forest
Closed Shrublands
Open Shrublands
Woody Savannas
Savannas
Grasslands
Permanent Wetlands
Croplands
Urban and Built Up
Cropland/Natural Vegetation
Snow and Ice
Barren or Sparsely Vegetated
Water

Figure 34. IGBP land cover map of Eurasia in the USGS Earth land cover map. Lambert Azimuthal Equal Area Projection (Optimized for Europe). Image obtained from http://edcsns17.cr.usgs.gov/glcc/glcc.html

We have selected the following datasets, since their structure is more homogeneous than the others:

- USGS, which is a modification of the Anderson vocabulary
- International Geosphere Biosphere Programme (IGBP)
- Biosphere Atmosphere Transfer (BAT)
- Simple Biosphere Model (SBM)
- Simple Biosphere Model (SBM2)

[10] http://edcsns17.cr.usgs.gov/glcc/glcc.html

The detailed vocabulary of each dataset can be seen in Appendix E. Only USGS presents a hierarchical taxonomy, with three levels.

In this experiment we have merged pairs of datasets using the mapping algorithm based on spatial distribution of dataset values. The results can be found in Appendix F. The result for a given pair of datasets could be considered as a reference merging/alignment to be compared to the result obtained by the terminological algorithm with WordNet. However, we can obtain a more accurate precision if we consider the number of spatial units that validate the obtained mappings in the datasets. For instance, let us consider that a mapping is obtained setting *A* subclass of *B* (where *A* and *B* belong to different ontologies). We can obtain the number of spatial units of value *A*, $|A|$, the number of spatial units of *B*, $|B|$, and finally the number of spatial units of their intersection, $|A \cap B|$. Partial spatial precision for this mapping can be defined as $|A \cap B|/|A|$, while partial spatial recall as $|A \cap B|/|B|$. This way, global spatial precision can be obtained as the average of the partial precision of each obtained mapping. Unfortunately this method does not give us the total number of reference mappings, and consequently recall cannot be computed in this way. Instead of recall, we will consider the percentage of the classes that have been mapped. Although this measure is not an accurate indicator, it gives us a hint on how well the obtained alignment would cover the reference one.

Let us note that $|A|$, $|B|$ and $|A \cap B|$ can be directly read from the matrix used in the spatial algorithm. Note also that the case of a common superclass mapping has to be computed in a slightly different way. If the obtained mapping is "*A* and *B* have a common superclass", the spatial precision is:

- 1, if *A* and *B* share spatial units and, if *A* also shares spatial units with *C*, then *A* and *B* share more spatial units than *A* and *C*.
- 0, otherwise

In this section we focus on evaluating several alignments/mergings of pairs of datasets in terms of the spatial precision.

The following table shows the results of merging IGBP and USGS datasets. WordNet without considering meronyms has been used. In this case, the algorithm was restricted to find only mappings for physical values of USGS (and not for the higher levels of its hierarchy).

Class(es) in IGBP	**Relation**	**Class(es) in USGS**	**Spatial precision**
Evergreen Needleleaf Forest (1)	equivalent	*Evergreen Needleleaf Forest* (14)	1.0
Evergreen Broadleaf Forest (2)	equivalent	*Evergreen Broadleaf Forest* (13)	1.0
Deciduous Needleleaf Forest (3)	equivalent	*Deciduous Needleleaf Forest* (12)	1.0
Deciduous Broadleaf Forest (4)	equivalent	*Deciduous Broadleaf Forest* (11)	1.0
Mixed Forest (5)	equivalent	*Mixed Forest* (15)	1.0
Closed Shrublands (6)	subclass	*Shrubland* (8)	0.84
Open Shrublands (7)	subclass	*Shrubland* (8)	0.68
Woody Savannas (8)	subclass	*Savanna* (10)	0.42
Nonwoody Savannas (9)	subclass	*Savanna* (10)	1.0
Grasslands (10)	equivalent	*Grassland* (7)	1.0

Permanent Wetlands (11)	common superclass “Wetland”	*Herbaceous Wetland* (17)	1.0
Permanent Wetlands (11)	common superclass “Wetland”	*Wooded Wetland* (18)	1.0
Croplands (12)	superclass	*Dryland Cropland and Pasture* (2)	1.0
Croplands (12)	superclass	*Irrigated Cropland and Pasture* (3)	0.94
Urban and Built-up (13)	equivalent	*Urban and Built-Up Land* (1)	1.0
Snow and Ice (15)	equivalent	*Snow or Ice* (24)	1.0
Barren or Sparsely Vegetated (16)	equivalent	*Barren or Sparsely Vegetated* (19)	0.94
Water Bodies (17)	equivalent	*Water Bodies* (16)	1.0
Cropland/Natural Vegetation Mosaic (14)	no relation		
	no relation	*Cropland/Grassland Mosaic* (5)	
	no relation	*Cropland/Woodland Mosaic* (6)	
	no relation	*Mixed Shrubland/Grassland* (9)	
	no relation	*Wooded Tundra* (21)	
	no relation	*Mixed Tundra* (22)	
	no relation	*Bare Ground Tundra* (23)	
Global Spatial Precision			**0.93**
Ratio of mapped IGBP classes			**0.94**

Table 6. Spatial precision for the merging of IGBP and USGS datasets (only physical values)

In this case, the same experiment carried out in PROMPT after removing the same stop words and suffixed, found 10 equivalence mappings. Global spatial precision is 0.99, although the ratio of IGBP classes mapped is 0.59, 0.36 below than our approach.

The following table shows the results of merging USGS and IGBP, but now permitting mappings for the abstract values of USGS. Note that in the case of this type of mapping, its precision is obtained from the union of the spatial extent of all the subclasses of the abstract value.

Class(es) in IGBP	Relation	Class(es) in USGS	Spatial precision
Evergreen Needleleaf Forest (1)	equivalent	*Evergreen Needleleaf Forest* (14)	1.0
Evergreen Broadleaf Forest (2)	equivalent	*Evergreen Broadleaf Forest* (13)	1.0
Deciduous Needleleaf Forest (3)	equivalent	*Deciduous Needleleaf Forest* (12)	1.0
Deciduous Broadleaf Forest (4)	equivalent	*Deciduous Broadleaf Forest* (11)	1.0
Mixed Forest (5)	equivalent	*Mixed Forest* (15)	1.0
Closed Shrublands (6)	subclass	*Shrubland* (8)	0.84
Open Shrublands (7)	subclass	*Shrubland* (8)	0.68
Woody Savannas (8)	subclass	*Savanna* (10)	0.42
Nonwoody Savannas (9)	subclass	*Savanna* (10)	1.0
Grasslands (10)	equivalent	*Grassland* (7)	1.0
Permanent Wetlands (11)	subclass	Wetland (abstract value)	1.0
Croplands (12)	equivalent	Cropland and Pasture (abstract value)	0.99
Urban and Built-up (13)	equivalent	*Urban and Built-Up Land* (1)	1.0

Snow and Ice (15)	equivalent	*Snow or Ice* (24)	1.0
Barren or Sparsely Vegetated (16)	equivalent	*Barren or Sparsely Vegetated* (19)	0.94
Water Bodies (17)	equivalent	*Water Bodies* (16)	1.0
Cropland/Natural Vegetation Mosaic (14)	no relation		
	no relation	*Mixed Shrubland/Grassland* (9)	
	no relation	*Wooded Tundra* (21)	
	no relation	*Mixed Tundra* (22)	
	no relation	*Bare Ground Tundra* (23)	
Global Spatial Precision			**0.93**
Ratio of mapped IGBP classes			**0.94**

Table 7. Spatial precision for the merging of IGBP and USGS datasets (physical and abstract values)

In this case, the obtained results by using PROMPT were the same as in the experiment without abstract values.

The following table shows the results of merging BAT and USGS. WordNet with meronyms has been used, and only physical values have been considered.

Class(es) in BAT	Relation	Class(es) in USGS	Spatial precision
Crops, Mixed Farming (1)	superclass	*Dryland Cropland and Pasture* (2)	1.00
Short Grass (2)	subclass	*Grassland* (7)	1.00
Evergreen Needleleaf Trees (3)	equivalent *	*Evergreen Needleleaf Forest* (14)	1.00
Deciduous Needleleaf Tree (4)	equivalent *	*Deciduous Needleleaf Forest* (12)	0.79
Deciduous Broadleaf Trees (5)	equivalent *	*Deciduous Broadleaf Forest* (11)	1.00
Evergreen Broadleaf Trees (6)	equivalent *	*Evergreen Broadleaf Forest* (13)	1.00
Tall Grass (7)	subclass	*Grassland* (7)	0.00
Tundra (9)	superclass	*Wooded Tundra* (21)	1.00
Tundra (9)	superclass	*Mixed Tundra* (22)	1.00
Tundra (9)	superclass	*Bare Ground Tundra* (23)	0.00
Irrigated Crops (10)	equivalent	*Irrigated Cropland and Pasture* (3)	1.00
Icecaps and Glaciers (12)	equivalent	*Snow or Ice* (24)	1.00
Bogs and Marshes (13)	common superclass “Wetland”	*Herbaceous Wetland* (17)	1.00
Bogs and Marshes (13)	common superclass “Wetland”	*Wooded Wetland* (18)	1.00
Inland Water (14)	subclass	*Water Bodies* (16)	1.00
Ocean (15)	subclass	*Water Bodies* (16)	1.00
Evergreen Shrubs (16)	subclass	*Shrubland* (8)	0.74
Deciduous Shrubs (17)	subclass	*Shrubland* (8)	1.00
Mixed Forest (18)	equivalent	*Mixed Forest* (15)	0.79
Interrupted Forest (19)	common superclass	*Barren or Sparsely Vegetated* (19)	0.00
Desert (8)	no relation		
Semidesert (11)	no relation		

	no relation	*Urban and Built-Up Land* (1)	
	no relation	*Cropland/Grassland Mosaic* (5)	
	no relation	*Cropland/Woodland Mosaic* (6)	
	no relation	*Mixed Shrubland/Grassland* (9)	
	no relation	*Savanna* (10)	
Global Spatial Precision			**0.82**
Ratio of mapped BAT classes			**0.89**

* since "tree" is meronym of "forest"

Table 8. Spatial precision for the merging of BAT and USGS datasets (only physical values)

This experiment with PROMPT obtained no mappings, since there are no common class names between the two ontologies.

The following table shows the results of merging USGS and BAT now permitting mappings for the abstract values of USGS.

Class(es) in BAT	**Relation**	**Class(es) in USGS**	**Spatial precision**
Crops, Mixed Farming (1)	superclass	*Dryland Cropland and Pasture* (2)	1.00
Short Grass (2)	subclass	*Grassland* (7)	1.00
Evergreen Needleleaf Trees (3)	equivalent	*Evergreen Needleleaf Forest* (14)	1.00
Deciduous Needleleaf Tree (4)	equivalent	*Deciduous Needleleaf Forest* (12)	0.79
Deciduous Broadleaf Trees (5)	equivalent	*Deciduous Broadleaf Forest* (11)	1.00
Evergreen Broadleaf Trees (6)	equivalent	*Evergreen Broadleaf Forest* (13)	1.00
Tall Grass (7)	subclass	*Grassland* (7)	0.00
Tundra (9)	equivalent	*Tundra* (abstract value)	0.99
Irrigated Crops (10)	subclass	*Irrigated Cropland and Pasture* (3)	1.00
Icecaps and Glaciers (12)	equivalent	*Snow or Ice* (24)	1.00
Bogs and Marshes (13)	subclass	Wetland (abstract value)	1.00
Inland Water (14)	subclass	*Water Bodies* (16)	1.00
Ocean (15)	subclass	*Water Bodies* (16)	1.00
Evergreen Shrubs (16)	subclass	*Shrubland* (8)	0.74
Deciduous Shrubs (17)	subclass	*Shrubland* (8)	1.00
Mixed Forest (18)	equivalent	*Mixed Forest* (15)	0.79
Interrupted Forest (19)	subclass	Forest land (abstract value)	0.12
Desert (8)	no relation		
Semidesert (11)	no relation		
	no relation	*Urban and Built-Up Land* (1)	
	no relation	*Cropland/Grassland Mosaic* (5)	
	no relation	*Cropland/Woodland Mosaic* (6)	
	no relation	*Mixed Shrubland/Grassland* (9)	
	no relation	*Savanna* (10)	
Global Spatial Precision			**0.85**
Ratio of mapped BAT classes			**0.89**

Table 9. Spatial precision for the merging of BAT and USGS datasets (physical and abstract values)

In this case, the experiment with PROMPT obtained 1 equivalent mappings, with a global spatial precision of 0.99, but a very low ratio of BAT classes mapped (0.06).

We will now compare SBM1, SBM2 and USGS two by two, with two aims: obtaining spatial precision, and checking that the obtained mappings are not contradictory among them.

The following table shows the results of merging SBM2 and USGS. WordNet with meronyms has been used, and only physical values have been considered.

Class(es) in SBM2	**Relation**	**Class(es) in USGS**	**Spatial precision**
Broadleaf Evergreen Trees (1)	equivalent *	*Evergreen Broadleaf Forest* (13)	1.00
Broadleaf Deciduous Trees (2)	equivalent *	*Deciduous Broadleaf Forest* (11)	0.93
Broadleaf and Needleleaf Trees (3)	common superclass *	*Mixed Forest* (15)	1.00
Needleleaf Evergreen Trees (4)	equivalent *	*Evergreen Needleleaf Forest* (14)	0.83
Needleleaf Deciduous Trees (5)	equivalent *	*Deciduous Needleleaf Forest* (12)	0.79
Short Vegetation (6)	common superclass	*Barren or Sparsely Vegetated* (19)	0.00
Dwarf Trees and Shrubs (8)	superclass	*Shrubland* (8)	0.03
Agriculture or Grassland (9)	superclass	*Grassland* (7)	0.95
Agriculture or Grassland (9)	superclass	*Dryland Cropland and Pasture* (2)	1.00
Agriculture or Grassland (9)	superclass	*Irrigated Cropland and Pasture* (3)	1.00
Ice Cap and Glacier (11)	equivalent	*Snow or Ice* (24)	1.00
Water, Wetlands (10)	superclass	*Water Bodies* (16)	1.00
Water, Wetlands (10)	superclass	*Herbaceous Wetland* (17)	1.00
Water, Wetlands (10)	superclass	*Wooded Wetland* (18)	0.00
Shrubs with Bare Soil (7)	no relation		
	no relation	*Urban and Built-Up Land* (1)	
	no relation	*Cropland/Grassland Mosaic* (5)	
	no relation	*Cropland/Woodland Mosaic* (6)	
	no relation	*Mixed Shrubland/Grassland* (9)	
	no relation	*Savanna* (10)	
	no relation	*Mixed Tundra* (22)	
	no relation	*Bare Ground Tundra* (23)	
Global Spatial Precision			**0.70**
Ratio of mapped SBM2 classes			**0.91**

* since “tree” is meronym of “forest”

Table 10. Spatial precision for the merging of SBM2 and USGS datasets (only physical values)

This experiment carried out with PROMPT obtained no mappings.

The following table shows the results of merging SBM1 and USGS. WordNet with meronyms has been used, and only physical values have been considered.

Class(es) in SBM1	Relation	Class(es) in USGS	Spatial precision
Evergreen Broadleaf Trees (1)	equivalent *	*Evergreen Broadleaf Forest* (13)	1.00
Broadleaf Deciduous Trees (2)	equivalent *	*Deciduous Broadleaf Forest* (11)	0.93
Deciduous and Evergreen Trees (3)	common superclass *	*Mixed Forest* (15)	1.00
Evergreen Needleleaf Trees (4)	equivalent *	*Evergreen Needleleaf Forest* (14)	0.83
Deciduous Needleleaf Trees (5)	equivalent *	*Deciduous Needleleaf Forest* (12)	0.79
Groundcover Only (7)	common superclass	*Barren or Sparsely Vegetated* (19)	0.00
Agriculture or Grassland (12)	superclass	*Grassland* (7)	0.35
Agriculture or Grassland (12)	superclass	*Dryland Cropland and Pasture* (2)	1.00
Agriculture or Grassland (12)	superclass	*Irrigated Cropland and Pasture* (3)	1.00
Persistent Wetland (17)	common superclass "Wetland"	*Herbaceous Wetland* (17)	1.00
Persistent Wetland (17)	common superclass "Wetland"	*Wooded Wetland* (18)	1.00
Water (19)	equivalent	*Water Bodies* (16)	1.00
Ice Cap and Glacier (20)	subclass	*Snow or Ice* (24)	1.00
Groundcover with Trees and Shrubs (6)	no relation		
Broadleaf Shrubs with Perennial Groundcover (8)	no relation		
Broadleaf Shrubs with Bare Soil (9)	no relation		
Groundcover with Dwarf Trees and Shrubs (10)	no relation		
Bare Soil (11)	no relation		
	no relation	*Urban and Built-Up Land* (1)	
	no relation	*Cropland/Grassland Mosaic* (5)	
	no relation	*Cropland/Woodland Mosaic* (6)	
	no relation	*Shrubland* (8)	
	no relation	*Mixed Shrubland/Grassland* (9)	
	no relation	*Savanna* (10)	
	no relation	*Wooded Tundra* (21)	
	no relation	*Mixed Tundra* (22)	
	no relation	*Bare Ground Tundra* (23)	
Global Spatial Precision			**0.84**
Ratio of mapped SBM1 classes			**0.67**

* since "tree" is meronym of "forest"

Table 11. Spatial precision for the merging of SBM1 and USGS datasets (only physical values)

This experiment with PROMPT obtained one equivalence mapping. This gives us a global spatial precision of 1, but a very low ratio of SBM1 classes mapped (0.07).

Finally, the following table shows the results of merging SBM2 and SBM1. WordNet with meronyms has been used.

Class(es) in SBM2	**Relation**	**Class(es) in SBM1**	**Spatial precision**
Broadleaf Evergreen Trees (1)	equivalent	*Evergreen Broadleaf Trees* (1)	1.00
Broadleaf Deciduous Trees (2)	equivalent	*Broadleaf Deciduous Trees* (2)	1.00
Broadleaf and Needleleaf Trees (3)	common superclass	*Deciduous and Evergreen Trees* (3)	1.00
Needleleaf Evergreen Trees (4)	equivalent	*Evergreen Needleleaf Trees* (4)	1.00
Needleleaf Deciduous Trees (5)	equivalent	*Deciduous Needleleaf Trees* (5)	1.00
Shrubs with Bare Soil (7)	superclass	*Broadleaf Shrubs with Bare Soil* (9)	1.00
Agriculture or Grassland (9)	equivalent	*Agriculture or Grassland* (12)	0.91
Water, Wetlands (10)	supreclass	*Water* (19)	1.00
Water, Wetlands (10)	superclass	*Persistent Wetland* (17)	0.06
Ice Cap and Glacier (11)	equivalent	*Ice Cap and Glacier* (20)	1.00
Short Vegetation (6)	common superclass	*Groundcover Only* (7)	0.00
Dwarf Trees and Shrubs (8)	no relation		
	no relation	*Groundcover with Trees and Shrubs* (6)	
	no relation	*Broadleaf Shrubs with Perennial Groundcover* (8)	
	no relation	*Groundcover with Dwarf Trees and Shrubs* (10)	
	no relation	*Bare Soil* (11)	
Global Spatial Precision			**0.82**
Ratio of mapped SBM2 classes			**0.91**

Table 12. Spatial precision for the merging of SBM2 and SBM1 datasets (only physical values)

This experiment carried out with PROMPT obtained three equivalence mapping. This gives us a global spatial precision of 0.97, but a low ratio of SBM2 classes mapped (0.27).

We have observed that these datasets offer a clear example of how the same reality can be classified in different ways, and how classes with equal or similar names in different ontologies may have been modelled in different ways, and consequently may have little overlapping. Even though, our terminological algorithm have obtained a precision ranging from 70% to 93%, with an average of 83%. Although recall cannot be properly computed, we have also seen that a mapping is suggested for more than the 85% of classes in average. Although the mappings obtained by PROMPT are usually precise, it generates very few mappings, often none. Compared to PROMPT, our approach always generates much more mappings, with a precision slightly below PROMPT when it returns mappings. In the best case for PROMPT, the ratio obtained by our approach was 36 points above the ratio obtained by PROMPT.

10 Semantic queries

In this chapter we focus on the semantic services that agents (in a wide sense, including humans and applications) need to find, translate and integrate thematic information in the context of an interoperable repository. These semantic services comprise 1) obtaining datasets and values for a selected theme, 2) translating a dataset or a dataset value to a different vocabulary, and 3) integrate different datasets into a new one depicting a particular theme. Other semantic services consisting in getting attribute values of ontology classes (dataset, dataset value, vocabulary,...) are not discussed in this chapter, since they are straightforward.

Each of these three semantic services or queries, as well as some variations that are also discussed in this chapter, are formally defined in the framework of our ontology in terms of Description Logic (DL). This way, queries can be implemented using a DL reasoner. Since our OntoGIS tool enables the connection to different reasoners, we have tested our implementation of queries with two of them: Racer and a simpler OWL-Lite reasoner included in the Jena API.

We will see in Chapter 11 how these types of semantic queries can be used by an external application to index images and videos and generate multimedia presentations according to the thematic content.

It is important to remark that these three semantic services are not available in current catalogues. Metadata standards only provide a very simple support for semantics through keywords. Consequently, the only “semantic” service that catalogues offer is a keyword-based service for finding datasets. However, this does not addresses the problem of semantic heterogeneity. We can observe that our semantic framework based on an ontology of the repository including DL definitions for themes enables us to define new functionalities that are of special importance for the integration of thematic information from different sources.

10.1 First type of semantic query: finding

This query retrieves the list of datasets and values that contain information related to a particular geographic theme. This query is useful for user’s searches. A user typically needs information on one geographic theme and does not know the details of the organization of the datasets in the repository. This query will give her/him the list of which dataset values (and their datasets) where s/he can find information related to the theme of interest.

The response to this query does not only consider the theme itself but also all the classes in the ontology that are subsumed by it. It is important to note that this includes all its subclasses (direct and inferred) and also all its equivalent classes, since an equivalent class is in fact also a subclass in DL.

The system uses the property *themeConnection* and its inverse *datasetValueConnection* of our representation model. We recall here that *themeConnection* has two subproperties, *qualitativeThemeConnection* for connecting qualitative themes and *quantitativeClassConnection* for connecting quantitative classes (see Section 4.2 for more details).

The process of obtaining the connected value given a qualitative theme or a quantitative class is similar. In fact, the user may be interested in a quantitative theme as temperature (and not a specific quantitative class) and the process will be similar too. To simplify the syntax in this section, we will simply use “thematic class” to denote an ontology class representing either a qualitative theme or a quantitative theme or class.

Given a thematic class, *SelectedTheme*, we can obtain the set of connected dataset values through the DL expression:

```
DatasetValue ⊓ (∃datasetValueConnection.SelectedTheme)
```

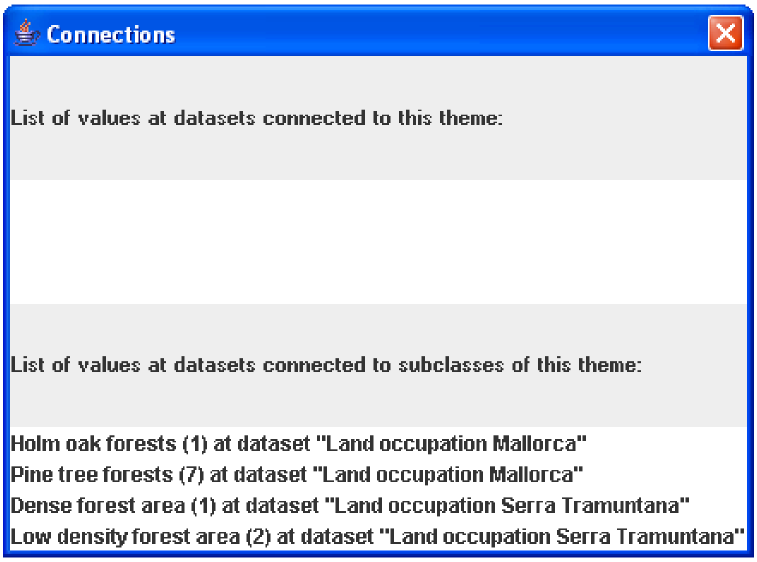
Note that this expression returns all the individuals of the class *DatasetValue* that are connected to an individual of the selected theme. Since we have constrained the domain of the role *datasetValueConnection* to *DatasetValue*, the first part of the intersection can in fact be removed:

```
∃datasetValueConnection.SelectedTheme
```

Note also that all the individuals of the subclasses of *SelectedTheme* are also individuals of *SelectedTheme*. Therefore, the reasoner will also return the values connected to the subclasses of *SelectedTheme*.

As we have already mentioned in previous sections, it is often important for a user to know the class that has been directly connected to a value, without considering its subclasses. In a purist Description Logic, given a particular individual, we cannot in principle distinguish which its “original” class (without considering inference) is. The only way to compute it would be checking among all the returned classes which one is subsumed by the others. However most APIs support this differentiation, which is the case of Jena. This way, we distinguish two variations of this first type of query: one using all the inferred individuals of *SelectedTheme* (which includes all its inferred subclasses), and the other using only direct individuals of the class, with no inference. From now on, we will denote this query by *query1Inference* in the first case and by *query1Direct* in the second case.

Figure 35 shows an example of the datasets and values connected to the theme *Forest*. It shows that the class *Forest* is not directly connected (*query1Direct*) to any dataset value, but it has subclasses that are connected to some values in the datasets *Land occupation Mallorca* and *Land occupation Serra Tramuntana* (using *query1Inference*).



List of values at datasets connected to this theme:

List of values at datasets connected to subclasses of this theme:

Holm oak forests (1) at dataset "Land occupation Mallorca"
Pine tree forests (7) at dataset "Land occupation Mallorca"
Dense forest area (1) at dataset "Land occupation Serra Tramuntana"
Low density forest area (2) at dataset "Land occupation Serra Tramuntana"

Figure 35. Example of first type of query: datasets and values connected to theme "Forest"

Furthermore, we will see in Chapter 11 that in some cases only the datasets connected to the selected theme, and not the values, are needed. We define a slight variation of *query1Inference* and *query1Direct* to be used in these cases, and will respectively call them *query1InferenceDs* and *query1DirectDs*:

```
(∃datasetValueConnection.SelectedTheme).valueDataset
```

A reciprocal for the first type query can be defined: given a value (*v*), return the connected thematic classes. The following expression

```
Theme ⊓ (themeConnection:v)
```

or its equivalent in languages with singleton sets:

```
Theme ⊓ (∃themeConnection.{v})
```

returns one individual of a thematic class connected to value *v*. The result of the reciprocal query is then the set of classes for the individual returned by the previous expression. Note that again, since the domain of the role *themeConnection* is already constrained to the *Theme* class, we can remove it from the expression:

```
{ C | C(themeConnection:v) }
```

Given an individual connected to value *v*, the query returns all the classes that have it as individual. This includes its own direct class and also all its superclasses. Again it is important to distinguish here among direct and inferred connections. In the case of inference, inferred superclasses will be included, while in the direct case will not. We denote by *query1Inference** and *query1Direct** the reciprocal of *query1Inference* and *query1Direct* respectively.

It is worth noting that, while a thematic class can be directly connected to different values (through different individuals of the theme), a value is directly connected with only one thematic class. Thus, considering only direct connections, we can affirm that, where C is a thematic class and v_i is an individual of *DatasetValue*:

$$\textbf{if } \mathtt{query1Direct(C)} = \{\mathtt{v}_\mathtt{i} \mid \mathtt{i} \in Z^{+}\} \textbf{ then } \mathtt{query1Direct}^{*}(\mathtt{v}_\mathtt{i}) = \mathtt{C}$$

Regarding the opposite of this statement, we can affirm that:

$$\textbf{if } \mathtt{query1Direct}^{*}(\mathtt{v}) = \mathtt{C} \textbf{ then } \mathtt{v} \in \mathtt{query1Direct(C)}$$

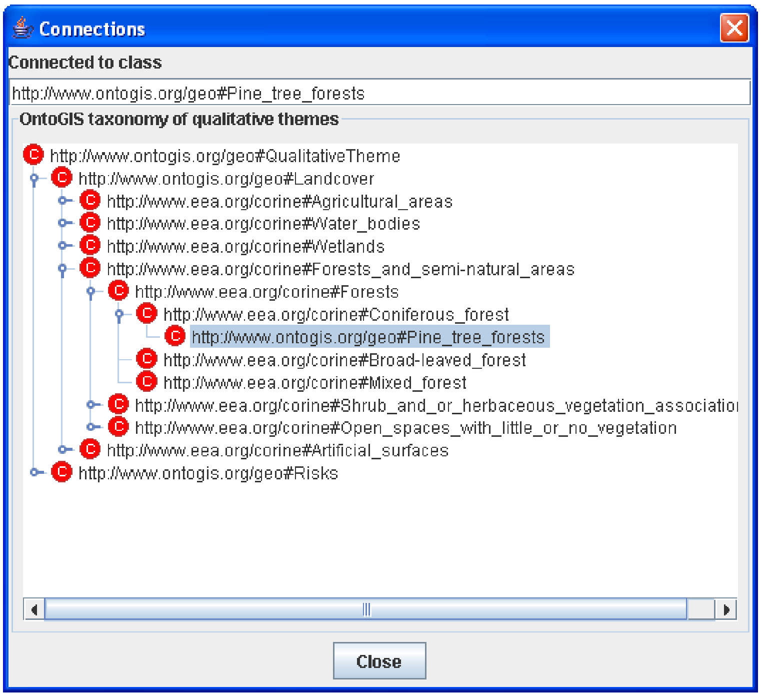
In the case of using inference, the corresponding expressions are weaker, since in this case a value can be connected to several classes:

$$\textbf{if } \mathtt{query1Inference(C)} = \{\mathtt{v}_\mathtt{i} \mid \mathtt{i} \in Z^{+}\} \textbf{ then } \mathtt{C} \in \mathtt{query1Inference}^{*}(\mathtt{v}_\mathtt{i})$$

And:

$$\textbf{if } \mathtt{C} \in \mathtt{query1Inference}^{*}(\mathtt{v}) \textbf{ then } \mathtt{v} \in \mathtt{query1Direct(C)}$$

Figure 36 shows an example of the results of *query1Direct** for the value 7, *Pine forest*, in dataset *Land occupation Mallorca*.



Connected to class

http://www.ontogis.org/geo#Pine_tree_forests

OntoGIS taxonomy of qualitative themes

```
http://www.ontogis.org/geo#QualitativeTheme
  http://www.ontogis.org/geo#Landcover
    http://www.eea.org/corine#Agricultural_areas
    http://www.eea.org/corine#Water_bodies
    http://www.eea.org/corine#Wetlands
    http://www.eea.org/corine#Forests_and_semi-natural_areas
      http://www.eea.org/corine#Forests
        http://www.eea.org/corine#Coniferous_forest
          http://www.ontogis.org/geo#Pine_tree_forests
        http://www.eea.org/corine#Broad-leaved_forest
        http://www.eea.org/corine#Mixed_forest
      http://www.eea.org/corine#Shrub_and_or_herbaceous_vegetation_associatio
      http://www.eea.org/corine#Open_spaces_with_little_or_no_vegetation
    http://www.eea.org/corine#Artificial_surfaces
  http://www.ontogis.org/geo#Risks
```

Figure 36. Example of inverted first type of query with no inference: class connected to value "Pine tree forest" (7) in dataset "Land occupation Mallorca"

In an analogous way, we also define the reciprocal of *query1InferenceDs* and *query1DirectDs*, and respectively call them *query1InferenceDs** and *query1DirectDs**. Given a dataset *d*, *query1InferenceDs** and *query1DirectDs** return the set of themes connected to any of the values of *d* with or without inference respectively:

```
{ C | C(x), ∀x ∈(themeConnection:valueDataset:d) }
```

10.2 Second type of semantic query: translation

This operation translates a dataset value to terms that are included in a particular normalized vocabulary. This operations enables an agent (either human or automatic) that understands only a particular vocabulary to read a dataset that uses its own non-normalized representation schema. The vocabulary has to have been previously integrated into the ontology and consequently, each term is represented through a thematic class in the ontology. Note that this query can only be applied to qualitative datasets.

Given a particular value of the selected dataset the objective is to return a term in the selected vocabulary that is not contradictory with the value. In particular, we are interested in the most specific one among them. This process will usually mean a loss of precision with respect of the original value.

We firstly obtain the thematic class in the ontology that is directly connected to the given value. The reciprocal of the direct first type of query (*query1Direct**) is used. Let *C* be the class returned by this operation. Let now *VSet* be the set of all the classes representing terms in the selected vocabulary. The query then has to return the most specific among those classes of *VSet* subsuming *C*:

```
min { V ∈VSet | C ⊑ V }
```

where the minimum is referred to the subsumption relation. It can be then expressed as:

```
V | V ∈VSet ∧ C ⊑ V ∧  (V ⊑ X, ∀X | X∈VSet ∧ C⊑X)
```

The following algorithm shows the process for obtaining the solution to this expression, where *Voc* is the selected vocabulary and *val* is a particular value in the selected dataset.

```
Let C be the class connected to value val:
      C=query1Direct$^*$(val)
Let VSet be the set of classes connected to terms in vocabulary Voc:
      for each v $\in$ Voc do
            VSet $\leftarrow$ VSet $\cup$ (termThemeConnection:v)
      end for
Let X be a variable initially containing the reference to Top class:
      X $\leftarrow$ $\top$
for each V $\in$ VSet do
      if C $\sqsubseteq$ V $\wedge$ V $\sqsubseteq$ X then

            X $\leftarrow$ V
      end if
end for
return X
```

If at the end of the algorithm *X* refers to the *Top* class, then no translation is possible. Otherwise, it contains the reference to a class that is the representation of a vocabulary term. Note that if the vocabulary has not been previously integrated into the ontology, and consequently its terms are not represented though classes, then the *VSet* set is empty and the operation will return the *Top* class.

We have also defined a variation of this query that translates all the values of a given dataset. We denote by *query2* and *query2Ds* respectively the translation of a value and the translation for a whole dataset. Given a dataset *ds*, *query2Ds* returns a set of classes connected to terms in vocabulary *Voc*, as follows:

```
Let S be a set of classes initially empty
Let VSet be the set of classes connected to terms in vocabulary Voc:
      for each v $\in$ Voc do
            VSet $\leftarrow$ VSet $\cup$ (termThemeConnection:v)
      end for
for each val $\in$ (valueDataset:ds) do
      Let C be the class connected to value val:
            C=query1Direct$^*$(val)
      X $\leftarrow$ $\top$
      for each V $\in$  VSet do
            if C $\sqsubseteq$ V $\wedge$ V $\sqsubseteq$ X then
                  X $\leftarrow$ V
            end if
      end for
      if $\top \not\sqsubseteq$ X then
            S $\leftarrow$ S $\cup$ {X}
      end if
end for
return S
```

10.3 Third type of semantic query: integration

This operation gets information from different datasets in order to integrate them in a new one. In the first subsection we will analyze the restrictions of some previous work on integration of hierarchical thematic datasets based on mathematical structures (order sets and lattices), that have already been described in Section 3.4. Subsection 10.3.2

describes in depth our algorithm for this operation. Subsection 10.3.3 discusses how modelled themes can be involved in the process of integration. Subsection 10.3.4 describes the so-called pessimistic integration. Finally, in subsection 10.3.5 we present two examples of integration of real datasets of the area of Serra de Tramuntana in Majorca.

It is important to recall here that our approach is not limited to work with only two datasets. Several datasets can be integrated at once. Another important issue is that the integration can be focused on a particular theme. For instance, having two datasets of land use with different thematic classifications, the user may be interested in extracting and integrating only the information related to agricultural areas. The use of modelled themes in the integration is another important element of our approach.

10.3.1 Related work

Phan-Luong, Pham and Jeansoulin (Phan Luong et al. 2003; 2004) use a complete lattice in order to define integration (see 3.4.3 for a detailed discussion). This structure is appropriate to represent an order set (a set with a partial order relation). In the case of a taxonomy, the subsumption relation is a partial order among the set of classes, and therefore is often modelled through a lattice. However, this representation on the one hand cannot model roles (properties), and on the other hand presents problems dealing with non-disjoint classifications, which is in fact the usual case of thematic geographic information. As an example that shows this problem, let us suppose that we have a class *Forest* and other class *Calcareous land* in the ontology. These classes are not disjoint, since there is no contradiction in having a land region being both at the same time. This is a typical situation, since the first is a type of land occupation and the second a type of geomorphologic land. However, in order to have a complete lattice, any pair of classes have to have an infimum, which is the element being the greatest lower bound of the pair in the partial order. Since they are not disjoint, the infimum should not be the class $\perp$ (*Bottom* or *Nothing*), and a new class has to be added, representing the intersection of both. It will be a subclass of both classes. The problem is that this process has to be done with any pair of non-disjoint classes in the ontology, and in fact, the obtained new class has to be intersected again with the remaining non-disjoint existing classes. Consequently, the process of adding new classes has to be done not only for any pair of classes, but for any subset of classes in the ontology. This makes the ontology to grow in an exponential way.

In consequence, the approach of Phan-Luong can only be applied in simple cases with small sets of classes that can be considered as mutually disjoint. Instead, our approach continues being based on DL, and supports bigger ontologies with non-disjoint classes. Furthermore, it also supports roles restrictions and DL definitions (models).

Regarding this last point, DL definitions, it constitutes a significant improvement compared to other existing approaches. Our approach makes it possible to integrate several datasets according to one particular theme (third type of query), where this theme may be modelled through a DL definition. In practical terms, this allows the system to check the land regions where the definition is satisfied in the particular scenario of the selected datasets in the repository. As an example, let us come back to our modelled theme *Area at fire hazard*, discussed in 4.2.4, as well as its DL definition. The system can integrate several datasets in order to produce a new one showing the areas under fire hazard according to the definition. If the selected modelled theme has more than one model, the user is prompted to select one of them. We will further discuss the issue of modelled themes involved in integration in 10.3.3.

The algebraic model of Worboys and Duckham (Worboys and Duckham 2002) (see 3.4.1 for a detailed discussion) presents also several significant restrictions. The integration of hierarchical thematic spaces presents the requirement that both hierarchies have to share a common set of atomic themes. However, these atomic themes are not used in the datasets, and in fact do not belong to their original thematic structure. Therefore, it is a human expert who should generate the set of atomic classes from the two original dataset hierarchies. This way, each class in the datasets has to be a superclass of at least an atomic class, while each atomic class has to be the subclass of at least one class in each dataset. This process is not trivial and often not possible unless the expert adds artificial classes only used to connect two thematic classes from different datasets. Consequently, the main theme portrayed in the two datasets has to be very similar if no exactly the same, since otherwise it will not be possible at all to find the set of atomic values, and the integration will not be possible to be carried out. We will see that our approach makes it possible to integrate datasets about very different themes. It is worth recalling here that in our case, the integrated thematic structure is obtained before the construction of the integrated dataset, during the merging process in a more flexible way.

Apart from this restriction, the optimistic integration based on lattices defined by Phan-Luong et al. is equivalent to this algebraic model (which is in fact previous). The work of Worboys and Duckham does not have the possibility of the pessimistic approach that Phan-Luong et al. adds. And finally, as in the case of Phan-Luong et al., it supports neither roles restrictions nor models.

Regarding (Duckham and Worboys 2005) (see 3.4.2 for a detailed discussion), it removes the restriction of the shared set of atomic classes, although it maintains the rest. But, as it was already discussed, it presents two significant limitations, which mainly refer to the merging phase.

On the contrary, our approach is mainly based on an open-world assumption typical of DL, where the knowledge (and its representation through the ontology) is not supposed to be complete. This open-world assumption is especially relevant in a context like ours where the ontology is built as new datasets are added to the repository, and consequently new knowledge is introduced. Furthermore, as it has already been mentioned, this open-world approach eliminates the restriction of classes organized in a lattice, where supremum and infimum have to exist for any pair of classes. Furthermore, DL provides richer semantic descriptions of classes as well as more semantic relations apart from subsumption. In particular, as it has already been said, our approach permits modelled themes and their DL definitions to participate in the integration, providing a especially relevant model checking capability, which allows users to find where a modelled theme is satisfied.

10.3.2 Description of our algorithm

The main issue in this operation is how the integrated value is obtained for a spatial unit that has values *v1* in dataset *D1* and *v2* in dataset *D2*, where *v1* is directly connected to the thematic class *X1* and *v2* is directly connected to class *X2*. Let us consider the example where *X1* is the class *Pine forest*, while *X2* is the class *Forest*, and where *Forest* subsumes *Pine forest*. There can be two different approaches:

- The integrated value is *Forest*, since we cannot assure whether the area labelled as *Forest* contains other types of forest different from pine forests
- The integrated value is *Pine forest*, since we can assume that the second dataset has a more precise thematic classification, and *Pine forest* does not have any conflict with *Forest*

Note that the first solution corresponds to a pessimistic approach in terms of the definition of Phan-Luon et al., while the second corresponds to the optimistic one. In the pessimistic approach we assure that the integrated value is more generic than or equivalent to all the original values. On the other hand, through the optimistic approach, the integrated value is more specific than or equivalent to all the original values. We always use the optimistic one when it is possible. But when it cannot be used because it returns a contradiction (*X1* and *X2* are disjoint), the pessimistic integration will be used instead.

The example shown above is the simplest case, where *X1* subsumes *X2* or vice versa. However we will often find that there is no subsumption relation among *X1* and *X2*. According to the optimistic approach, the integrated value should be the intersection of the concepts *X1* and *X2*, unless they are disjoint. However, returning something like "*X1* and *X2*" does not provide more information than a simple overlay operation. Therefore, the objective here is to define an algorithm that uses all the knowledge we have on the subclasses of *X1* and *X2* in order to return which information can be found that is common to *X1* and *X2*.

Let us suppose that we have a very simple ontology with only three classes *X1*, *X2* and *A*, where *A* is subclass of both *X1* and *X2*:

$$A \sqsubseteq X1,\ A \sqsubseteq X2$$

If we transform this ontology in a lattice, we will get that $X1 \wedge X2$ is *A*. However, this is not true in DL, since it deals with incomplete knowledge. There is no axiom that prevents that a new class *B*, different from *A*, is inserted as subclass of both *X1* and *X2*. Therefore, the only assertion that can be made is:

$$A \sqsubseteq X1 \sqcap X2$$

However, if the open-world assumption is not closed in a certain way, we will never obtain anything different from "*X1* and *X2*", except in the cases where *X1* subsumes *X2* or vice versa. This "world closing" has a drawback: if more datasets are inserted later, and in consequence more thematic classes are added to the ontology, the result of the integration datasets *D1* and *D2* before this addition may be different than after it.

However it is important to note that, although the first integration would be less precise than the second one, it does not contain any contradiction after the addition of new classes. That means that as far as the knowledge is becoming more complete, the integration process is also becoming more precise.

Let us firstly discuss the case where there is no subsumption relation among *X1* and *X2* and they do not have any common subclass. Note that in DL, this fact does not necessarily mean that *X1* and *X2* are disjoint. *X1* and *X2* are disjoint only if the following expression is true:

$$X1 \sqcap X2 \sqsubseteq \bot$$

But note that this is a logical comparison, not restricted to a particular interpretation. The reasoner has to check if according to the axioms in the ontology, it can be entailed that *X1* is disjoint with *X2*. Therefore, even if there is no class being subclass of both *X1* and *X2*, it does not mean that the expression is true.

If it can be entailed that *X1* and *X2* are disjoint, a contradiction has been found: according to the axioms in the ontology, a particular area should not be labelled as *X1* in a dataset and *X2* in another. In this case, it is not possible to integrate both values and the *bottom* class is returned to indicate that there is a contradiction. The system would try now a pessimistic integration (see 10.3.4).

In the other case, where *X1* and *X2* do not have common subclasses but are not disjoint, there is no option but returning "*X1* and *X2*", as in an overlay. To do this, a new class labelled as "*X1 and X2*" is added to the ontology, and so the following axiom is:

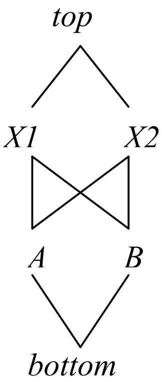
$$\text{"X1 and X2"} \equiv X1 \sqcap X2$$

Let us now analyze the case where there is no subsumption relation among *X1* and *X2*, but they do have common subclasses. The objective is to find the equivalent to the infimum in a lattice: the most generic class among those subsumed by both *X1* and *X2*. This could be defined in the following way:

```
Let S be the set of common subclasses to X1 and X2:
      $S = \{ C \mid C \sqsubseteq X1, C \sqsubseteq X2 \}$
Return $C \in S$ such that $D \sqsubseteq C, \forall D \in S$
```

Note that if there already exists a class that is equivalent to the intersection of *X1* and *X2*, this class will be returned.

A complete lattice has the restriction that given two elements they necessarily have an infimum (and also a supremum). This is again not equivalent in DL. The following structure is not a lattice, since *X1* and *X2* do not have an infimum (and *A* and *B* do not have a supremum). But it is a valid subsumption classification in DL.



In this case, where *S*, the set of common subclasses of *X1* and *X2*, comprises the classes *A* and *B*, we know that the union of *A* and *B* is a subset of the intersection of *X1* and *X2*:

```
A ⊔ B ⊑ X1 ⊓ X2
```

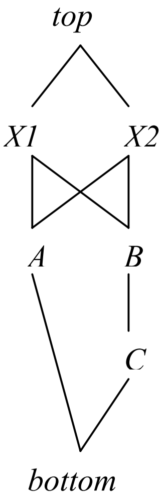
And therefore, the value returned is *A* ⊔ *B*. To do this, a new class is added "*A union B*" to the ontology, and a new axiom is also added to indicate that this new class is equivalent to the union of *A* and *B*:

```
"A union B" ≡A ⊔ B
```

Note that it can be entailed that the new class is subsumed by both *X1* and *X2*:

```
"A union B" ⊑ X1
"A union B" ⊑ X2
```

But since there can be subsumption relations among the elements in *S*, before adding the new class for the union of the elements of *S*, the classes that are subsumed by other classes in *S* can be removed from it. The following diagram shows an example.



In this case, *A*, *B* and *C* conform the set *S*, since they are subsumed by both *X1* and *X2*. But since *C* is subsumed by *B*, it can be removed from *S*, and the result of the integration is again "*A union B*".

In consequence, the previous definition of the most generic class among those subsumed by *X1* and *X2*, can be replaced by the following, assuming that they have common subclasses:

```
    Let S be the set of common subclasses to X1 and X2:
        S = { C | C ⊑ X1, C ⊑ X2 }
    if ∃C ∈S such that D ⊑ C, ∀D ∈S then
        return C
    else
        for each C ∈S do //simplify S
            if ∃D ∈S such that D ⊑ C and D≠C then
                S = S \ {D}
            end if
        end for
        add a new class X
        add a new axiom: $X \equiv S_1 \sqcup … \sqcup S_k$ , where $S_i \in S$
        return X
    end if
```

Thus, the complete algorithm for integration of two values for the same area *x* in different datasets *D1* and *D2* is:

```
Let X1 be the class connected to the value of area x at dataset D1
Let X2 be the class connected to the value of area x at dataset D2
    //both X1 and X2 are obtained through query1Direct*
if X1≡X2 then
    return X1
else
    if X1 ⊓ X2 ⊑ ⊥ then //X1 and X2 are disjoint
        return ⊥ //returns a contradiction
    else
        Let S be the set of common subclasses to X1 and X2:
            S = { C | C ⊑ X1, C ⊑ X2 }
        if S = ∅ then
            //S is empty, no common subclases
            add a new class X
            add a new axiom: X ≡ X1 ⊓ X2
            return X
        else
            if ∃C ∈S such that D ⊑ C, ∀D ∈S then
            //C subsumes all the elements in S
                return C
            else
                for each C ∈S do //simplify S
                    if ∃D ∈S such that D ⊑ C and D≠C then
                        S = S \ {D}
                    end
                end for
```

```
                        add a new class X
                        add a new axiom: $X \equiv S_1 \sqcup … \sqcup S_k$ , where $S_i \in S$
                        return X
                  end if
            end if
      end if
end if
```

This algorithm can be easily generalized for *n* datasets, where *n* is a natural number greater than 1:

```
Let Xi be the class connected to the
            value of area x at dataset i, i=1,...,n
            //all of them are obtained through query1Direct*
if $X1 \equiv ... \equiv Xn$ then
      return X1
else
      if $X1 \sqcap ... \sqcap Xn \sqsubseteq \bot$ then //X1,..., Xn are disjoint
            return $\bot$ //returns a contradiction
      else
            Let S be the set of common subclasses to X1 and X2:
                  $S = \{ C \mid C \sqsubseteq X1, ..., C \sqsubseteq Xn \}$
            if $S = \emptyset$ then
                  //S is empty, no common subclases
                  add a new class X
                  add a new axiom: $X \equiv X1 \sqcap ... \sqcap Xn$
                  return X
            else
                  if $\exists C \in S$ such that $D \sqsubseteq C$, $\forall D \in S$ then
                  //C subsumes all the elements in S
                        return C
                  else
                        for each $C \in S$ do //simplify S
                              if $\exists D \in S$ such that $D \sqsubseteq C$ and $D \neq C$ then
                                    $S = S \setminus \{D\}$
                              end
                        end for

                        add a new class X
                        add a new axiom: $X \equiv S_1 \sqcup … \sqcup S_k$ , where $S_i \in S$
                        return X
                  end if
            end if
      end if
end if
```

This algorithm is executed for each spatial unit of the destination integrated dataset. If the spatial units of the original datasets do not exactly overlap, their intersection is calculated. This way, the process for one spatial unit in the destination dataset will always involve a tuple of *n* values, where *n* is the number of datasets being integrated. Given a spatial unit, the query for its tuple of dataset values may return a class that is not connected to any value in the selected datasets. Note also that the values of the generated dataset will be connected to thematic classes in the ontology. In fact, the URI of the class is used for providing a definition to the value.

An important remark about an implementation aspect has to be made here. The computation of this query for the thousands or millions of spatial units in datasets means that a particular tuple of dataset values may be processed a great number of times. To avoid this, the system keeps a list of the tuples of dataset values that have already been processed, together with their corresponding resulted integrated class. Before calculating the query to integrate a new tuple of *n* values (where *n* is the number of different datasets) for a new spatial unit, the process checks in the list whether the tuple has already been calculated. Otherwise, the query is executed and the result is stored in the list.

A relevant variation of this third type of query is defined in order to filter the results through a particular theme. This is especially useful when having different datasets from different sources and the goal is an integration focusing on a particular aspect (a particular theme). For instance, let us suppose the case where we have different land occupation datasets from different sources, and the user may be interested in integrating them to analyze only forests. Only those classes that are subsumed by the selected filtering theme *T* (forests in the example) can be returned by the algorithm. Furthermore, only those values that are connected to *T* or its subclasses will be considered in the integration process. If none of the values of a spatial unit is connected to a subclass of forests, the algorithm will return the *Botton* class, meaning that these values cannot be integrated with this thematic filter. An exception is applied to this restriction when a subclass *C* of *T* has been explicitly defined as a DL intersection of other classes $C_1,...,C_n$. In this case, if a value is connected to one of these classes C_i, although C_i may not be a subclass of *T*, then the value will be considered in the process in order to check whether the definition of class *C* can be satisfied. We say that the definition of a class *C* is *satisfied* in a spatial unit if the reasoner can infer *C* from the classes connected to the different values of the spatial unit in different datasets. We will see in the next subsection that this exception is particularly useful for applying the integration query to obtain the spatial units where a modelled theme can be satisfied.

We denote by *query3Filtered* this variation of the query. We will also see in Chapter 11 that it will be used in the context of indexing images and video based on the thematic content.

Finally, the integration query can be used in the context of evaluating maps discrepancies or maps temporal changes, since the integrated map resulting from the query stresses the areas where inconsistencies between two or more datasets exist. This map can help dataset producers to detect possible areas where significant discrepancies between these datasets exist, requiring further attention and possible re-mapping. This can be especially useful in the case of having a dataset that can be considered as a valid reference, and others can be compared to it. It can also be used to highlight possible changes in a temporal series of datasets structured according to the same application ontology.

In this context, the percentage of spatial units that produce inconsistencies can also be used as a measure of *spatial agreement* (Fritz and See 2005) (see also 3.3) between different datasets. We have implemented a slight adaptation to *query3* in OntoGIS that enables the user to avoid the creation of new classes in the integration process. This way, if a spatial unit contains two values that are connected to classes *A* and *B* that are neither disjoint not one subclass of the other, it will be considered as a special type of contradiction, instead of adding the class "*A and B*". The user can also give different weights to the two types of contradictions in order to obtain a final spatial agreement, and to depict in the integrated map the areas of more serious conflicts or changes. Other measures of spatial agreement could also be defined taking into consideration the path distance between *A* and *B* in the case of one being subclass of the other.

10.3.3 Modelled themes involved in the integration

Modelled themes can be involved in the integration process. The usual way to do this is by checking whether a tuple of values satisfies one of its models. As we have mentioned above, we say that the definition of a class *C* is *satisfied* in a spatial unit if the reasoner can infer *C* from the classes connected to the different values of the spatial unit in different datasets. Let us recall our example of fire hazard discussed in 4.2.4, where the modelled theme *Area_at_fire_hazard* was given a model called *model-Area_at_fire_hazard-Majorca_fire_brigade,* defined as the intersection of the qualitative class *Pine_forest* and the quantitative classes *Temperature-clftem1-more25* (in classification *clftem1*) and *Precipitations-clfpre1-less50* (in classification *clfpre1*). In a query integrating the corresponding datasets focusing on areas at fire hazard, the algorithm has to check for each spatial unit whether its dataset values are connected (considering inference) to the classes involved in the definition of the model. If so, the definition of *Area_at_fire_hazard* is satisfied, and consequently the result of the query for that spatial unit will be *Area_at_fire_hazard*.

Figure 37 shows the results of the integration of three datasets: one for land occupation, one for temperatures (using classification *clftem1*) and one for precipitations (using classification *clfpre1*). The integration process has been filtered by the theme *Area_at_fire_hazard*. The areas in red are those satisfying the model, and consequently have the value *Area_at_fire_hazard*. The areas in black are classified as *Unknown*, since the result of the integration process is not a subclass of the filtering theme.

Note that if the user has not selected either a dataset of temperatures with the classification *clftem1* or a dataset of precipitations with the classification *clfpre1* to be integrated, the model cannot be satisfied and all the spatial units will have the value *Unknown*.

In this example the modelled theme *Area_at_fire_hazard* comprises only one model. As we have already discussed in 4.2.4, the model is set as equivalent to the modelled theme at the moment of executing a query. When a modelled theme has several models, the user has to decide which model s/he wants to consider. The selected model will be set as equivalent to the modelled theme, while the rest of the models will be temporarily removed (see again 4.2.4). This way, the inference engine has a necessary and sufficient definition for the modelled theme. Alternatively, the user may decide not to consider any model, and they all are temporarily removed, and the algorithm works with the modelled theme as a "normal" qualitative theme.

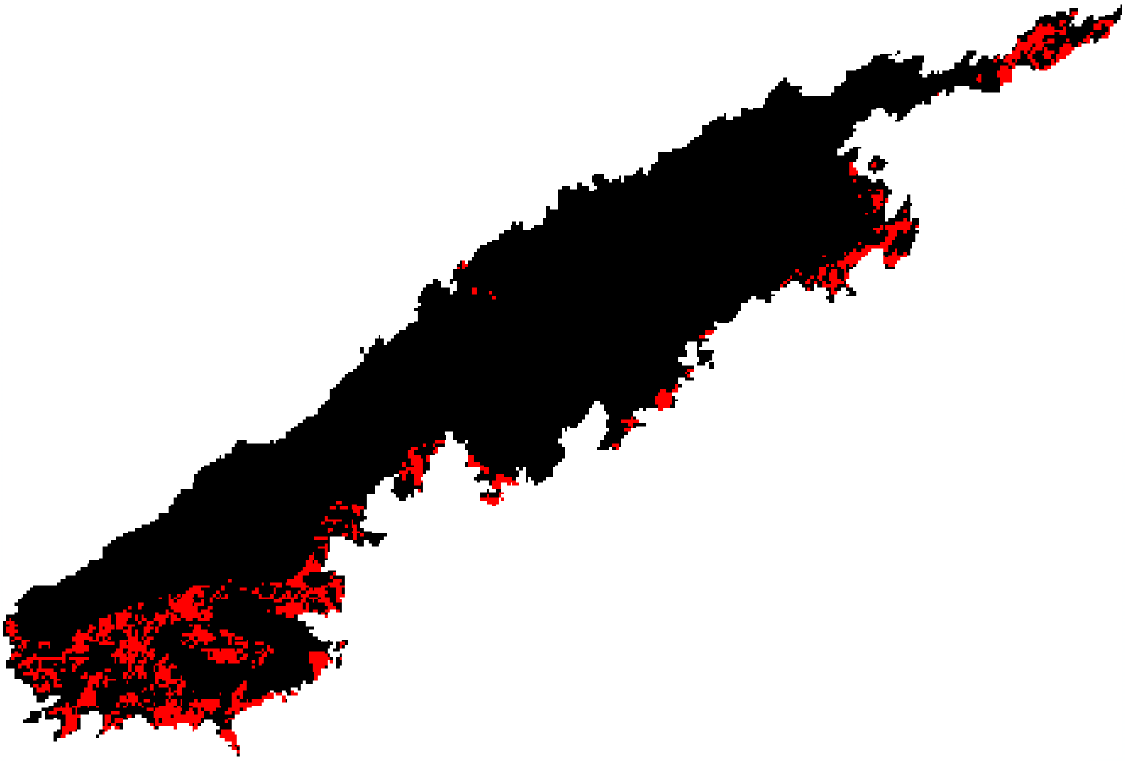


Figure 37. Example of integration involving a modelled theme

It is important to recall here that this type of query always work with a copy of the ontology that is sent to the inference engine. This copy only contains the models that have been selected, which are equivalent to their modelled themes. This way, the inference engine has at most one logical definition (necessary and sufficient) for each modelled theme. Once the query finishes, the copy is removed from the inference engine.

Finally, it is worth clarifying that modelled themes are not necessarily only be used to filter the query. Since a modelled theme is also a qualitative theme, once the algorithm knows whether it is satisfied in a particular spatial unit, it can be used as another qualitative theme. For instance, in subsection 10.3.5 we will see an example of an integration involving two modelled themes, where the filtering theme is a superclass of both.

10.3.4 The pessimistic approach

As we have already mentioned, when the optimistic approach returns a contradiction, the system should use the pessimistic approach instead. Given a tuple of dataset values, and the set of thematic classes directly connected to them, the pessimistic approach is based on returning the information that all of them have in common, removing the part that make them different. In other words, the pessimistic approach returns a thematic class that is superclass of all the connected thematic classes. If there are more than one (at least the class *Theme* will be superclass of all of them), the most specific is returned. Note that this is equivalent to the definition of a supremum.

It may seem a little contradictory to firstly get the directly connected thematic classes and then calculate their superclasses. But, although this could be directly computed using *query1Inference** instead of *query1Direct**, the pessimistic integration is only called once the directly connected classes have been obtained and it has been proved that they are disjoint.

The following algorithm shows the pessimistic approach to the integration:

```
Let Xi be the class connected to the
           value of area x at dataset i, i=1,...,n
           //all of them are obtained through query1Direct*
Let XSet be the set of their common superclasses:
      { X | Xi ⊑ X, ∀i=1,...,n }
Let S be the most specific among the classes in XSet
      S | S ∈ XSet ∧(S ⊑ X, ∀X ∈ XSet)
return S
```

If the algorithm returns one of the classes defined in our model of representation (*Theme*, *QualitativeTheme*, *QuantitativeTheme*, *QuantitativeClass*, *ModelledTheme*, *Model* or *QualitativeMixTheme*), it means that the classes do not have anything in common and cannot be integrated.

10.3.5 Two examples of datasets integration

In this subsection we will describe two examples of integration of datasets. The second example involves modelled themes, while the first one not. We have used in these two examples several real raster datasets of the area of Serra de Tramuntana in Majorca, with a cell size of 10 m^2 and an area of interest of around 5 million cells.

The repository contains two qualitative datasets of land use/land cover that are structured according to different application ontologies. It also contains several quantitative datasets: one for annual rains, one for slopes and another one for distance to roads, among others. The repository initially contained the CORINE vocabulary that has been merged into the taxonomy of themes.

The first example consists in a query integrating the qualitative datasets focusing on forests (class *http://www.eea.org/corine#Forests*). In the first dataset there are two values that have been connected as subclasses of this class: *Holm oak forests*, as a subclass of *http://www.eea.org/corine#Broad-leaved_forest*, and *Pine tree forests,* as a subclass of *http://www.eea.org/corine#Coniferous_forest*. In the second dataset, other two classes have been connected as direct subclasses of *http://www.eea.org/corine#Forests*: *Dense forest area* and *Low density forest area*. Since the four classes connected to the datasets have no subclasses in common, when a spatial unit has two different values in both datasets, then the integrated value will be their intersection. For instance, if a spatial unit has the value *Holm oak forests* in one dataset and *Dense forest area* in the other, the integrated value will be *Dense forest area AND Holm oak forests*, where this class is defined as the DL intersection of the corresponding two classes.

The dataset resulting from this query can be seen in Figure 38. The result of the pessimistic approach can be seen in Figure 39, where instead of intersections, the class *Forests* is returned when two different values are obtained.

In the second example all the datasets in the repository are integrated in order to obtain a new one for risks of forest fires. The repository contain the definition of two modelled themes: *High risk of forest fires* and *Moderate risk of forest fires*. These two classes are subclasses of the qualitative theme *Risk of forest fires*, which is the theme used to filter the query.

High risk of forest fires has one model that defines it as the intersection of the qualitative class *http://www.eea.org/corine#Forests*, and the quantitative classes corresponding to a distance smaller than 1 km to roads, to a slope greater than 25°, and to annual rains below 800 mm.

Moderate risk of forest fires has one model that defines it as the intersection of the qualitative class *http://www.eea.org/corine#Forests* and the quantitative classes corresponding to a distance smaller than 1 km to roads, and also to the union of the qualitative classes corresponding to a slope smaller than 25° and to annual rains above 800 mm.

The dataset resulting from this query can be seen in Figure 40. Note that the values of the integrated dataset correspond to the two subclasses of the filtering theme *Risk of forest fires*.

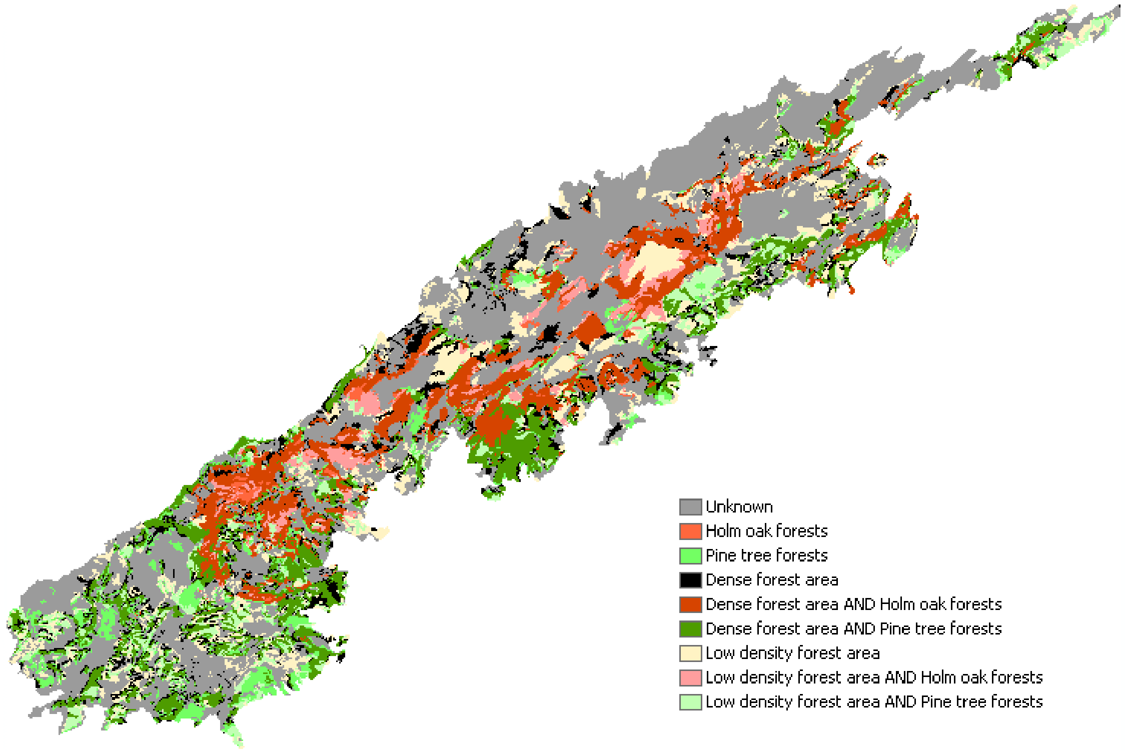

Unknown
Holm oak forests
Pine tree forests
Dense forest area
Dense forest area AND Holm oak forests
Dense forest area AND Pine tree forests
Low density forest area
Low density forest area AND Holm oak forests
Low density forest area AND Pine tree forests

Figure 38. Result of the integration of two land use/land cover datasets of Serra de Tramuntana (Majorca) to obtain a new one focusing on forests

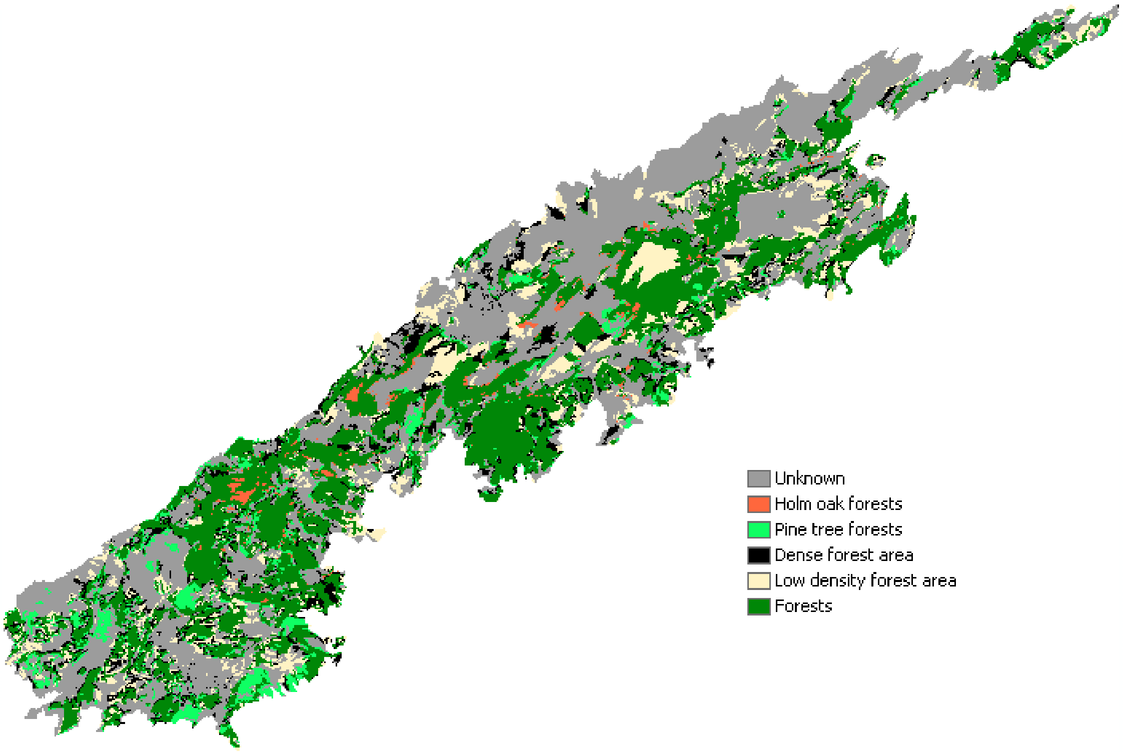


Figure 39. Result of the pessimistic integration of two land use/land cover datasets of Serra de Tramuntana (Majorca) to obtain a new one focusing on forests

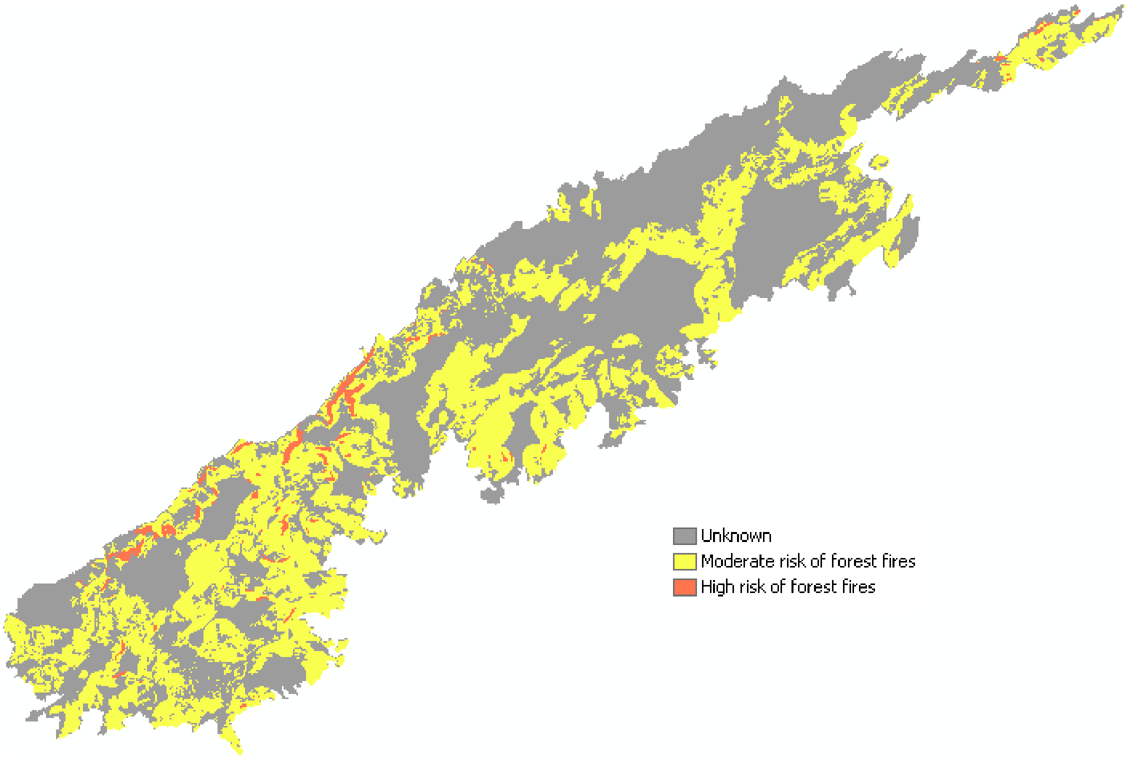


Figure 40. Result of the integration of several datasets of Serra de Tramuntana (Majorca) to obtain a new one focusing on risks of forest fires

11 VideoGIS: A multimedia context

In this chapter we will present an example of how the thematic geographic information in our ontology can be used in a different context in the framework of the semantic web. More specifically, we will describe the role of our semantic framework in a process of indexing and retrieving geo-referenced multimedia elements, and particularly video sequences, according to their thematic geographic content. We will not discuss architectural issues in this chapter, since they are out of the scope of this thesis. Instead, we will focus on how geo-referenced videos are modelled to support thematic-based indexing and retrieval, and how the semantic services defined in Chapter 10 are used in this context.

In a geo-referenced video sequence, some properties of the camera regarding its location are captured during the video recording. These properties include position (typically obtained from a GPS receiver), orientation with respect to the North (for instance captured from a digital compass), and often vertical tilt (the angle with respect to the horizon). Focal length and receiver (negative or CCD in a digital camera) length are also recorded in order to determine the angle of vision. All these properties make it possible to obtain the geographic area (*area of vision* from now on) that can be seen in each frame of the video sequence. Consequently, each frame can be associated with its corresponding area of vision and with the thematic information in the area of vision. This way, a video base containing a collection of video sequences can be queried according to their related themes. For example, the query asking “forests” would return all the fragments of videos in the video base containing forests or related themes (subclasses of forest).

The multi-layered structure of geographic information can provide a rich description for video sequences. Furthermore, the complex structure of video, with its temporal dimension, makes this integration challenging. We have been working during more than ten years in different aspects of hypermedia modelling, and this integration is especially interesting for us.

In Section 11.1 we give a brief overview of different approaches using geo-referenced video. We can classify them in two main groups: a first one that focuses on the enhancement of the video images with information of buildings or other elements; and a second one that is mainly oriented to provide visual information for geographic features in a GIS environment, especially in contexts like road management, fire decision support or land-cover maps validation. Instead, our proposal follows a different direction: it uses the thematic geographic information that can be extracted from a GIS or spatial database, once the video is geo-referenced, in order to index that video. This is the basis for a digital video library, and consequently our work is more related to video information retrieval than to the abovementioned systems. Our library will permit external agents (in a broader sense, including users or applications) to access the video collection to retrieve elements satisfying thematic criteria. The results of a query will be a collection of video fragments (segments) extracted from different video sources. We have called VideoGIS to the system that implements this type of integration of geo-referenced video and geographic information.

It is important to remark that spatial information is only processed at the time of indexing videos or still images. The library has to provide all the necessary thematic information to the clients through meta-information attached to video segments or still images. This way, clients do not need to have access to geographic datasets and, since they do not have to deal with spatial data, can be more generic.

After the discussion on different uses of geo-referenced video in Section 11.1, Section 11.2 presents a general background for image and video retrieval, including a general description of MPEG-7, the ISO standard for audiovisual meta-information. Section 11.3 describes two previous prototypes of VideoGIS, while Section 11.4 describes our semantic-based proposal for indexing and querying still images and video sequences based on thematic information, using the repository ontology, as well as its three types of queries, that was described in the previous chapters. Finally, in the last section we discuss some research possibilities for continuing this work.

11.1 Related work: using geo-referenced video

The use of geo-referenced video has become relatively frequent in the last years. This has been especially remarkable in organizations that maintain linear infrastructures as streets, roads and railroads. Special systems with video and location capturing devices are designed to be mounted on vehicles. This way, the vehicle can easily obtain geo-referenced videos for the whole network. Once processed and integrated in a GIS environment, these geo-referenced videos provide a rich and intuitive support for monitoring and decision making.

Several commercial systems exist to record geo-referenced videos and to add them to GIS environments through a simple post-processing procedure, which basically includes a synchronization of the video and the location data, and an interpolation to obtain a location for each frame (note that GPS receivers get measures slower than the video frame rate). Probably the most extended software tools are MediaMapper and GeoVideo, both developed by Red Hen Systems[1], which also provides different hardware systems for recording geo-referenced video that can be mounted on vehicles or on aircrafts, or that can be carried by pedestrians. These software tools enable the user to view the path of the camera on a map. They also show a cursor on that path that indicates the exact location of the camera during video playback. The user may control the video by moving this cursor on the path, as well as from a typical video control with play/stop/rewind/fast-forward buttons. An analogous commercial tool is CamNavMapper, by BlueGlen[2], that also covers acquisition, integration in a mapping environment and similar playback controls. A similar playback functionality is provided

[1] http://www.redhensystems.com
[2] http://www.blueglen.com/

by ImageCat's Views[3]. Other simpler software tools like GeoMovie by Magic Instinct Software[4] or VideoMapper[5] provide a postproduction process that superimposes the coordinates of the camera and other meta-information on the video image.

As an indication of the increasing importance of geo-referenced video, the Open GeoSpatial Consortium has proposed a geo-video service for its OWS-3 (OGC Web Services Phase 3) interoperability initiative. The aim of this initiative is to develop "a web service for access to video data including geo-location information". This service will provide an interface for requesting stream video, that can be controlled through play-back commands from a web service client. The service also will provide metadata in the video stream sufficient for a client to geo-locate the video.

Some examples of the use of geo-referenced video, apart from linear infrastructures management, are (Nobre and Câmara 2001) and (Wulder et al. 2005). The former has developed a forest fire decision support system. This system has a collection of geo-referenced aerial videos, and at the moment of a fire alarm it enables firemen to watch videos of the affected area. The latter also uses geo-referenced aerial videos in a system for the validation of land-cover maps of inaccessible areas of Canada. Also relevant is (Yoo et al. 2005), that has developed a system that combines geo-referenced video captured from a specially equipped vehicle called 4S-Van (Lee et al. 2003) and a 3D model of a city consisting in a 2D feature-based map with building heights and a digital elevation model. A method called VWM (Virtual World Mapping) (Kim et al. 2003a; Kim and Park 2004) enables them to link spatial segments in frames to buildings from the city model. This architecture supports visualizing geo-referenced videos enhanced with information of buildings in a GIS environment (Kim et al. 2003b), as well as developing other interfaces, as the Personal navigation system (Hwang et al. 2003) for portable devices. This is very related to the discipline of Augmented Reality, where geo-referenced video recorded from a camera carried by the user is processed in real-time in order to show her/him (usually through special devices) an augmented version of the image, presenting information on what s/he is seeing.

Although from a different approach, it is worth mentioning the Aspen Movie Map Project (Lippman 1980) developed at MIT in 1978, which is considered the first project that combined video and geographical information, and in fact is usually referred to as the birth of multimedia (Negroponte 1995). Using four cameras on a truck, all the straight segments of the streets of Aspen were filmed in both directions, as well as every turn (also in both directions), taking an image every three meters. The system consists of two videodiscs that enables users to "drive" through the city, deciding in each crossing which direction to follow. The user could stop in front of some of the major buildings of Aspen and walk inside. Interiors of several buildings were also filmed. A screen was used to show video, while another showed a street map of Aspen. The user could point to a spot on the map and jump directly to it, instead of finding the way through the city.

It has to be remarked that we do not consider here those approaches consisting in video clips with an overall spatial reference for the whole clip. Instead, we focus on geo-

[3] http://www.imagecatinc.com
[4] http://www.justmagic.com
[5] http://www.videomapper.com

references at the level of frames or segments. A non-exhaustive list of “classical” examples of the use video clips in geographical multimedia applications are BBC Doomsday Project (Openshaw et al. 1986), a video-disk-based map of Great Britain where the user can visualize videos and other multimedia elements from certain localities; (Shiffer 1992), which provides a collaborative hypermedia tool for urban planning; the CD-ROM of ParcBIT (Blat et al. 1995), a hypermedia application for supporting architects to develop a plan for a technologic park; and the hypermedia application for the North Norfolk coastal management discussed in (Raper 2001).

11.2 Some foundations on image and video retrieval

The aim of this section is to provide a general background on the field of image and video retrieval. The first subsection examines the process of indexing images and videos. More details on this subject can be found in our paper (Navarrete and Blat 2003). The second subsection discusses different video models. More details can be found in our previous publications (Navarrete and Vega 2003) and (Navarrete and Blat 2002a). The final subsection briefly describes MPEG-7 focusing on structural and semantic issues. MPEG-7 is the ISO specification for the description of audiovisual content, More details on MPEG-7 and other standards for the description of the audiovisual content can also be found in (Navarrete and Vega 2003).

11.2.1 Image and video indexing

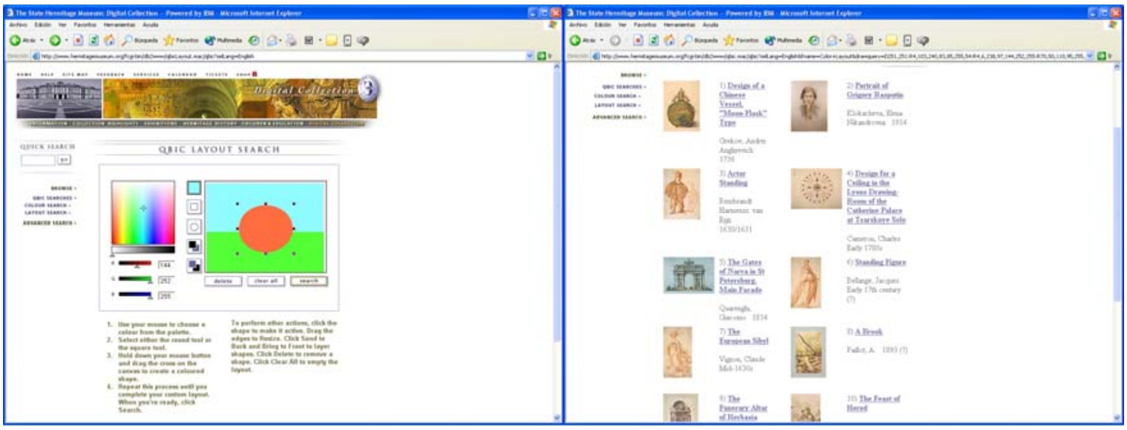
Indexing is the process of representing the contents of an image or a video sequence to support subsequent searches. Indexing may be carried out manually or using automatic techniques. In the case of video, indexing is usually preceded by a phase of segmentation where the original footage is split into segments, which can be considered the minimal unit of meaning. Segments are often identified with shots[6]. However, we will see in 11.2.2 that other approaches do not necessarily rely on shots. After the segmentation process, each segment has to be indexed.

In a manual indexing, an expert provides a set of terms describing an image, video, or video segment. Queries will also be expressed through terms, and consequently the search process will only involve text. Although this is a simple approach, it presents the main drawback of its high cost, especially in the case of big collections of data. Furthermore, other significant problem relates to the interpretation between different users, since two people will hardly describe an image or video in the same way.

Automatic techniques are focused on the extraction of a set of low-level parameters of the image related to colour, shapes, textures and layout. In such a way an image could be identified almost univocally. Instead of expressing a query through terms, an example of an image similar to what is being searched is usually provided by the user. This query paradigm is usually known as *query by example* or *query by image content*.

[6] Shot is defined as a unit of action photographed without interruption and constituting a single camera view (Webster dictionary).

Classical examples of systems using this paradigm are IBM's QBIC (Flickner et al. 1995), called CueVideo in a more recent version, which is the basis of the image searcher of Hermitage Museum; SWIM system (Zhang et al. 1995) of the National University of Singapore; and VideoQ project (Chang et al. 1997), and its evolution VisualSEEk, of Columbia University. In the case of video, the frames that best represent the segment are extracted (*key-frames*), and the image processing techniques will be applied to these key-frames.



1. Use your mouse to choose a colour from the palette.
2. Select either the round tool or the square tool.
3. Hold down your mouse button and drag the cross on the canvas to create a coloured shape.
4. Repeat this process until you complete your custom layout. When you're ready, click Search.

To perform other actions, click the shape to make it active. Drag the edges to Resize. Click Send to Back and Bring to Front to layer shapes. Click Delete to remove a shape. Click Clear All to empty the layout.

1) Design of a Chinese Vessel, "Moon Flask" Type
Orekov, Andrei Angkevich 1736

2) Portrait of Grigory Rasputin
Klokacheva, Elena Nikandrovna 1914

3) Actor Standing
Rembrandt Harmensz. van Rijn 1630/1631

4) Design for a Ceiling in the Lyons Drawing Room of the Catherine Palace at Tsarskoye Selo
Cameron, Charles Early 1780s

5) The Gates of Narva in St Petersburg. Main Facade
Quarenghi, Giacomo 1814

6) Standing Figure
Bellange, Jacques Early 17th century (?)

7) The European Sibyl
Vignon, Claude Mid-1630s

8) A Brook
Fallot, A. 1893 (?)

9) The Funerary Altar of Herbasia

10) The Feast of Herod

Figure 41. An example of QBIC layout query. Image captured from the web of Hermitage Museum (http://www.hermitagemuseum.org)

However, these techniques based on low-level image parameters are not precise enough, perhaps with the exception of some very concrete contexts, like in problem of face detection and recognition, or in the case of (Wang et al. 2002), which detects and identifies fishes that appear in video sequences. Even more important is that there is no semantics in the description of the image or video. Although this is appropriate for systems based on the query by example paradigm, it is not valid for more generic systems like a general purpose digital library or a web searcher for images or videos.

To solve the high cost and interpretation problems of manual methods and the lack of semantics of automatic algorithms based on pixel-level parameters, some projects have used other automatic techniques that are not directly applied to the image but to enclosed materials. In these cases queries will be expressed by means of terms. (Enser and Sandom 2002; Yang et al. 2004) refer to this approach as concept-based indexing, while content-based indexing would be the approach that only considers low-level image parameters. User tests made by (Christel and Conescu 2005) in the framework of the Informedia digital library, (Haupmann and Witbrock 1997; Hauptmann 2005), show that a concept-based orientation nearly always produces better results in terms of precision and user satisfaction.

The simplest concept-based technique is the used by Columbia Universisty's WebSEEk (Smith and Chang 1997), which assumes that file name and path partially describe the content of an image or video. However, this method is clearly too simplistic. More sophisticated are Google[7] and Yahoo![8], that provide a search for images and another for videos in the Internet. They are based on the assumption that an image in a web page is

[7] http://www.google.com
[8] http://search.yahoo.com

usually described by means of a text that is located near the image tag. This way, apart from the file name and path, the adjacent text is extracted and used to index images and, in the case of Yahoo!, also videos. In the case of videos in Google, this method is not applied. Instead, a manual metadata editor is used for indexing videos. Yahoo! also indexes metadata for videos created manually by users. This method based on adjacent text usually retrieves images (or videos) that do not objectively correspond to the searched terms, but in contrast, the user usually can find one or more appropriate images (or videos) for the query in the first page of results. Nevertheless, both Google and Yahoo! present two important drawbacks: on the one hand they can only retrieve resources linked from HTML pages and consequently this method cannot be applied to a bank of autonomous images or videos; on the other hand, they deal with atomic videos and do not consider their internal temporal structure.

Other approach of concept-based indexing is based on attaching synchronized production notes to the video. This is a usual approach in news archives, where the Japan Broadcasting Corporations, NHK, (Kim and Shibata 1996) is a classical reference, that also uses natural languages techniques. Using a synchronized script is an equivalent method for films or TV serials.

Other methods extract the audio from the video and obtains its textual transcription by means of speech processing techniques. This text is processed as in automatic text retrieval systems. An example of the use of audio is IBM's CueVideo, the evolution of QBIC. However, speed processing algorithms are not completely reliable and even less in situations where speakers have not trained the system.

Other source of information are the texts that are usually overprinted on news videos. Informedia digital library (Haupmann and Witbrock 1997; Hauptmann 2005) from Carnegie Mellon University combines its own speech recognition software Sphinx, and a VOCR (Video Optical Character Recognition) to extract the description attached to a news video. Especially relevant for a geographic information context is a specific project of Informedia (Christel and Olligschlaeger 1999), that extracts location names from both the overprinted text and the sound track. These locations are associated with the video, and consequently the system can retrieve the news videos for a given place. By combining this with a gazetteer and a simple mapping server, a new interface is provided to allow users to visualize a map of the region while the video news is played. This interface also permits users to click on places or regions on a map to retrieve their related video news.

In this context of providing an indexation of a collection of videos through enclosed material, our work follows an analogous approach. In our case, this enclosed information is the geo-reference, i.e. camera position, orientation, tilt and angle of vision. The indexing mechanisms use a GIS in order to extract the thematic information in the area of vision for selected frames. In the following subsection we will discuss the video model we have defined and how these representative frames are selected.

11.2.2 Video modelling

Although a video can be modelled in many different ways, depending on the concrete application, during the last 15 years there have been several attempts to develop a general model independent from the application. Several models of that kind have been published but usually they are only adaptations or variations of older ones. They can be grouped in two big families, clearly differentiated between each other. The first one, segmentation-based models, gives priority to the structure, while the second one, stratification-based models (or object-based models), prioritizes the conceptual aspects. We describe them, showing some of the most relevant variants.

11.2.2.1 Segmentation-based models

The simplest approach to model video is the use of shots as the basis of the description. Each shot in this model has attached meta-information describing it. This model is typically used when dealing with automatic segmentation. The result of the automatic process is the detection of the shot (or segment) boundaries, while some low-level properties can be attached to the shot as the average histogram, the most representative frame, et cetera. A good example of a digital video library using this approach is the Físchlár Digital Video System of Dublin City University (Lee et al. 2000).

Segmentation-based models do not imply necessarily automatic segmentation. They are based in a segmentation of the video in a set of shots that will embed meta-information to be described. On the one hand, the segmentation may be made manually or automatically. On the other hand, the description may consist of low-level visual aspects, but may also include semantic information manually generated. Shots are temporally ordered and it is possible to create a multilevel abstraction over them, grouping them into scenes and sequences.

Models based on segments present an important lack of flexibility. Segments have associated meta-information describing them, but it is not possible to represent the semantics of a set of segments. This is called the granularity problem: there is only one descriptive coarseness for all the video; while a coarseness can be useful for reflecting some aspects, it can be unsuitable for others. An important collateral effect is that users will not assume the same coarseness as the author did. Although this problem is not significant in some situations like in films, which have a clear shot/scene/sequence structure, in other contexts it constitutes an important restriction. More flexible mechanisms are needed in these cases.

(Hjelsvold and Midtstraum 1994) proposed a solution for the granularity problem maintaining a fixed hierarchy of composition units (shots, scenes and sequences). Each of these units are related to a “frame sequence”, but there can be other frame sequences independent from the structure. Since the meta-information is attached to the frame sequence and not to the composition unit, this model allows free queries, independent from the structure. An example could be the need to identify every frame sequence within Casablanca where Ingrid Bergman can be seen.

Other variation comes from (Bibiloni 1999). He considers a fixed set of shots as the bottom level of a dynamic hierarchy. He defines an algebra of operations and an edition language to program the hierarchy.

11.2.2.2 Stratification-based models (or object-based models)

To solve the granularity problem, the stratification model (Davenport et al. 1991; Aguierre Smith 1992) was proposed. The idea behind it is that while segmentation models make a partition of the footage in segments, the stratification model segments the contextual information. Instead of having a list of shots each one with a description, we will have a list of descriptions having pointers to frames. These descriptions are called strata, which give name to the model. This is a subtle difference but has significant consequences: it allows video units to overlap and encompass each other.

Each stratum is a single descriptive attribute which has been derived from the shooting environment. These attributes define the environmental context during the recording process: the “where”, “who”, “what”, “when”, “why” and “how”; a stratum is a contextual element. Each stratum has associated an starting and ending frame and any frame may have a variable number of strata associated with it or with part of it. Therefore, the description of any chunk of video is obtained by examining the different strata embedded in that chunk.

(Adali et al. 1996) and (Subramanian 1997) defined the object-centred model, that is a specialization of the stratification model. They use the objects and events in the video as strata. Therefore, a video base will be based on the following three components:

- Objects present in the video
- Properties of these objects, which can be frame-dependent or frame-independent
- Activities (events), actions done by the objects

Opposite to the segmentation approach, these three components are related to the semantics and not to the structure. However, a structural unit can be identified as the chunk of image with the same set of objects and events. This means that either the appearance or disappearance of a new object will produce a new structural unit, called segment. Either a Frame segment trees (FS-Tree) or a R-segment trees can be used to link objects with segments in an efficient way. However, there is no support for hierarchical structures (scenes and sequences).

Object-based models can be used for continued raw material, while segmentation models are inadequate since shots cannot be determined. An example is a continuous aerial video. On the other hand, one problem of the object-based approach is that it can frequently generate a micro-segmentation when many objects are involved.

Finally, an interesting addition to the stratification model is given by (Tran et al. 2000). Their model is also based on objects and events, but unlike the “traditional” model, they do not have to be related to time values. This can be useful for instance when extracting metadata from the script. Temporal relations (*before*, *during*, *overlaps*, *starts*, …) among objects and events may be defined. They define an algebra and a calculus to allow queries in such an environment. Nevertheless, from our point of view, the advantage of allowing concepts without time, it is not worth the complexity of the queries.

11.2.3 MPEG-7

MPEG-7 (ISO 2000), formally named “Multimedia Content Description Interface” is an ISO standard, ISO 15938, for describing the features of multimedia content so users can search, browse, and retrieve that content more efficiently and effectively. It is important to note that the standard only limits to the description itself, and neither to the description generation nor to the description consumption (applications that will use those descriptions).

To describe each element, MPEG-7 defines four normative elements: descriptors (D), description schemes (DS), a Description Definition Language (DDL) and coding schemes. A DS specifies the structure and semantics of the relationships between its components, that can be either D’s or other DS’s. DDL is a standardized language to define both D’s and DS’s, based on XML Schema.

The standard is divided in seven components:

- ISO 15938-1: MPEG-7 Systems
- ISO 15938-2: MPEG-7 Description Definition Language
- ISO 15938-3: MPEG-7 Visual
- ISO 15938-4: MPEG-7 Audio
- ISO 15938-5: MPEG-7 Multimedia DSs (MDS)
- ISO 15938-6: MPEG-7 Reference Software
- ISO 15938-7: MPEG-7 Conformance

We are especially concerned with part 5, MPEG-7 MDS, that provides a framework for generic descriptions of all kinds of multimedia, including audio, video and text, that are not considered isolated in parts 3 and 4. Its organization is shown in Figure 42.

We focus on the module of “Content description”, which is divided into:

- Structural aspects: the description of audiovisual content from the viewpoint of its structure, breaking it down into spatial, temporal and spatio-temporal components
- Semantic aspects: the description of audiovisual content from the viewpoint of its conceptual notions

This distinction does correspond to the two main types of video models that were presented in the previous section: models based on segments and models based on objects (or strata) respectively.

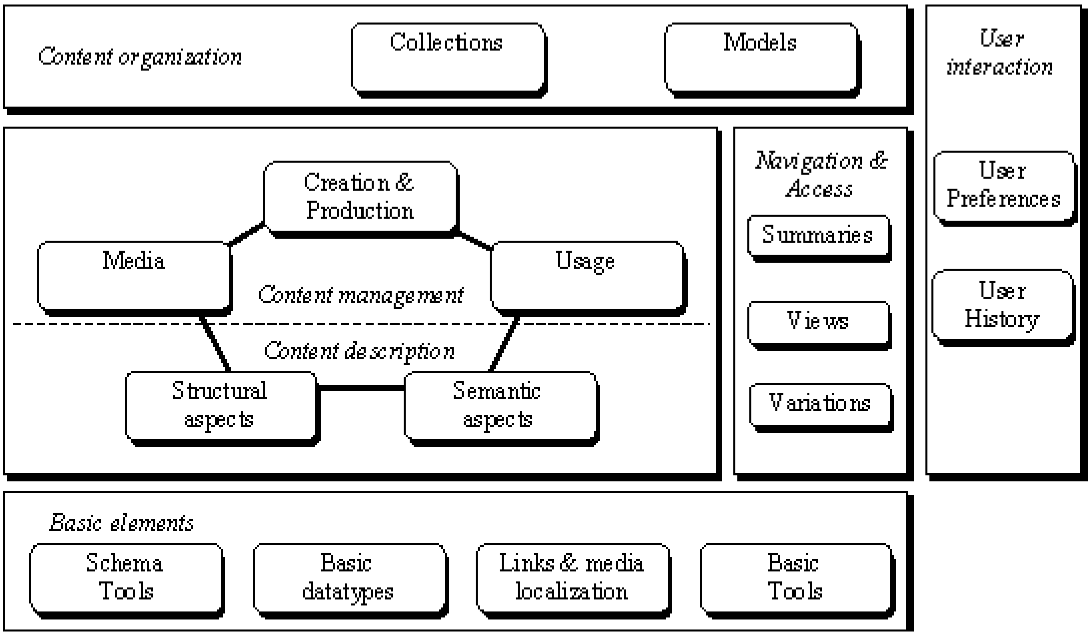


Figure 42. Overview of MPEG-7 MDS

The core part of the structural description is the Segment DS. The segment is defined as a section of an audio-visual content item. It describes the result of a spatial, temporal, or spatio-temporal partitioning of the AV content. In fact, Segment DS is an abstract class, with nine subclasses that may contain both spatial and temporal properties: Multimedia Segment DS, AudioVisual Region DS, AudioVisual Segment DS, Audio Segment DS, Still Region DS, Still Region 3D DS, Moving Region DS, Video Segment DS and Ink Segment DS.

The Segment DS is recursive, i.e., it may be subdivided into sub-segments, and thus may form a hierarchy (tree). The resulting segment tree is used to describe the media source, the temporal and / or spatial structure of the AV content. Nevertheless, a graph structure can be defines by means of the SegmentRelation DS, that reflects relationships between segments that cannot be represented through the segment tree. It can use relations as “is-close-to”, “is-composed-of”, et cetera.

We can see that the Segment DS obviously follows the segmentation-based modelling with the corresponding problems that we have enunciated above, mainly related to the granularity.

Regarding the semantic approach, the core element is now the Semantic DS, that is intended to encapsulate all the description of a narrative world. It is related to the SemanticBase DS that describes the semantic entities in the narrative world. It is an abstract class that can be extended by the following classes:

- Object DS, that represents the objects
- Event DS, that represents the actions
- SemanticPlace DS and SemanticTime DS, that describe respectively a place and a time in a narrative world

- Semantic State DS, that describes parametric attributes of a semantic entity at a given time or spatial location in the narrative world, or in a given location in the media (e.g., the piano's weight is 100 kg or the cloudiness of a day)
- Concept DS, that describes a semantic entity that cannot be described as a generalization or abstraction of a specific object, event, time place, or state

As in the case of the Segment DS, the conceptual description is recursive and can be organized as a tree. Finally, the Semantic Relation DS allows also to create a graph by representing relationships between semantic concepts or between semantic concepts and segments.

11.3 Two previous approaches

We describe in this section two previous prototypes of VideoGIS. This evolution will make it easier to understand the final prototype discussed in Section 11.4, which considers the semantics of the thematic information.

11.3.1 VideoGIS prototype 1

This first version of VideoGIS has been our first approach to a geo-referenced video retrieval based on geographic information. The indexing process is based on determining the features appearing in video segments. This way, information concerning features can be overprinted on the image when video is played. On the other hand, given a query on either geometric or alphanumeric attributes of features in the database, the collection of the resulting video segments can be dynamically concatenated in a single video sequence that is presented to the user. The development of this prototype has produced a segmentation algorithm based on the geographic features appearing in videos, an indexing structure using PostgreSQL database, a sub-system for queries, as well as some issues related to the presentation. The prototype and these contributions are further analyzed in (Navarrete and Blat 2002b). This first prototype of VideoGIS was mainly developed during a stay of three months at the Environmental Systems Analysis Group of the Universidade Nova de Lisboa in 2001.

Geo-referenced videos are supposed to be filmed from the air from a camera attached to an airplane. To simulate this we have built a tool that generates video sequences from the orthophoto of the metropolitan area of Barcelona, together with coordinates for each frame.

Videos are encoded in QuickTime format. Geo-reference of a video is provided through an XML file which includes general properties of the video (source file and frame rate) as well as the geographic coordinates for each frame in the video, as the following code shows:

```
<video src="v0.mov" fps="15">
      <frame>
            <number>0</number>
            <x1>347003</x1>
            <y1>4700506</y1>
            <x2>347403</x2>
            <y2>4700806</y2>
      </frame>
      <frame>
      ...
</video>
```

On the other hand, geographic information consists of vector datasets in GML 2.0, the XML-based language for encoding geographic information standardized by Open Geospatial Consortium, currently in version 3.1.1 (OGC 2004). These datasets contain vector features represented by means of the geometries supported by GML: Point, LineString, LinearRing, Polygon, MultiPoint, MultiLineString, MultiPolygon and MultiGeometry.

The prototype uses PostgreSQL, a RDBMS with spatial indexing capabilities by means of R-trees. Spatial data is imported from GML documents and features are inserted in the database, using one table for each type of feature. It also uses two dictionary tables, one describing the tables of features and other describing the columns of each table.

Segmentation is determined by two types of queries that the prototype has to support, which are related to the two abovementioned functionalities:

- Retrieve all the features in one segment; this query is needed to overprint the elements of interest on a segment
- Retrieve all the segment showing one (or a set of) feature(s) according to a spatial query; this query is needed to select the segments that will form part of a final video

In this context, a video segment is defined as a set of contiguous frames containing the same set of features. This way, the segmentation algorithm extracts the rectangle of vision for each frame and obtains the set of features that it contains. This set is compared with the set of the previous frame, and if they are different a new segment is started. The following algorithm describes this process:

```
start first segment
for every frame in the video do
      get the coordinates of the frame (a polygon)
      get the set of features in that polygon
      if this set "differs" from the previous one then
            finish segment
            start new segment
      end if
end for
finish last segment
```

Note that this approach does not differentiate between features of different types. We will observe in Section 11.4 that our semantic approach considers different segmentations, one for each semantic layer (where simplifying a semantic layer could be seen now as a type of feature).

Indexing is based on a variation of the frame-segment tree (Subramanian 1997). This type of tree combines a list of the features with a list of the segments of video. Each feature has a dynamic list with pointers to the segments where it is present, while each segment has a dynamic list with pointers to its features. Since these segments are stored in a PostgreSQL database, a typical B-tree is used to index them, instead of the segment tree proposed by Subramanian. On the other hand, geographic features are indexed through R-trees. Figure 43 shows the indexing structure, which makes it possible an efficient retrieval of both the segments containing a given feature, and the features contained in a given segment. More information on this structure as well as on other spatial and temporal indexing structures can be found in (Navarrete 2001).

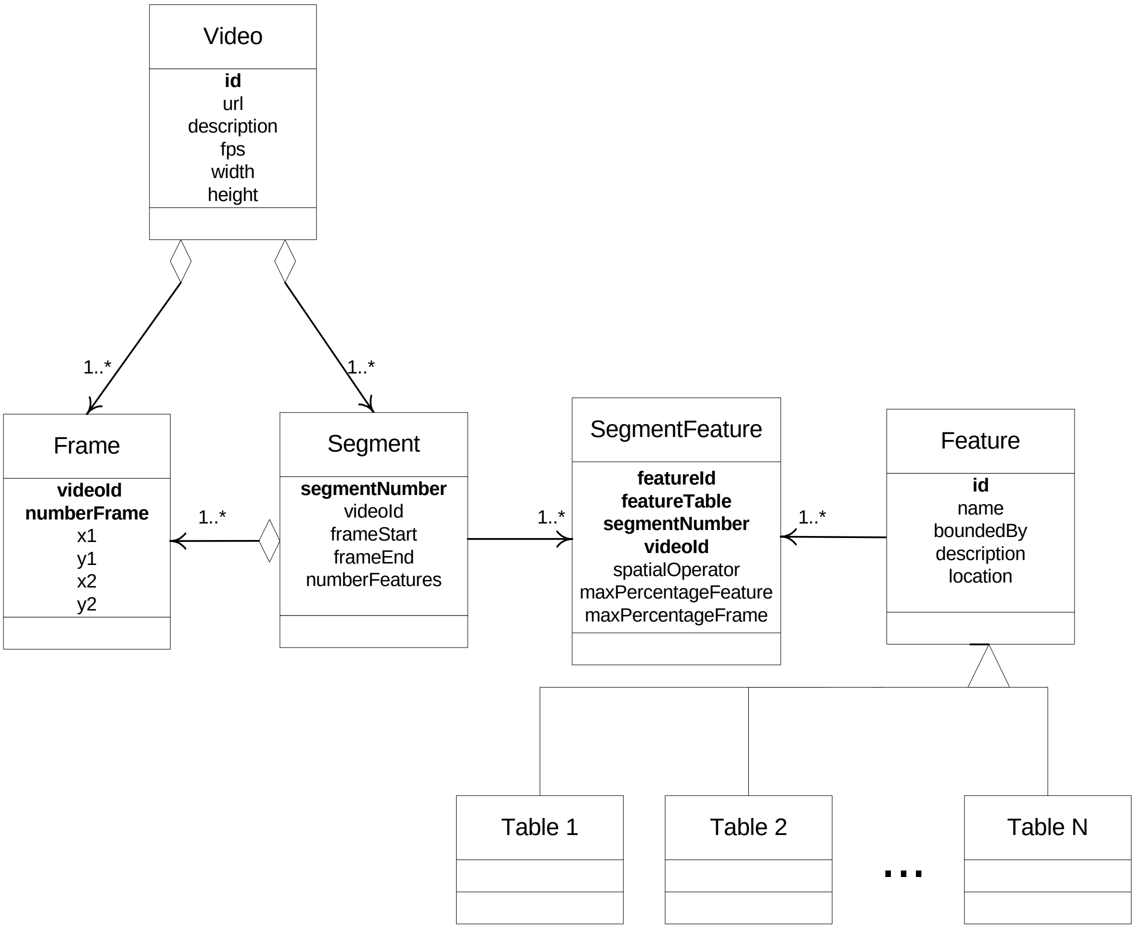
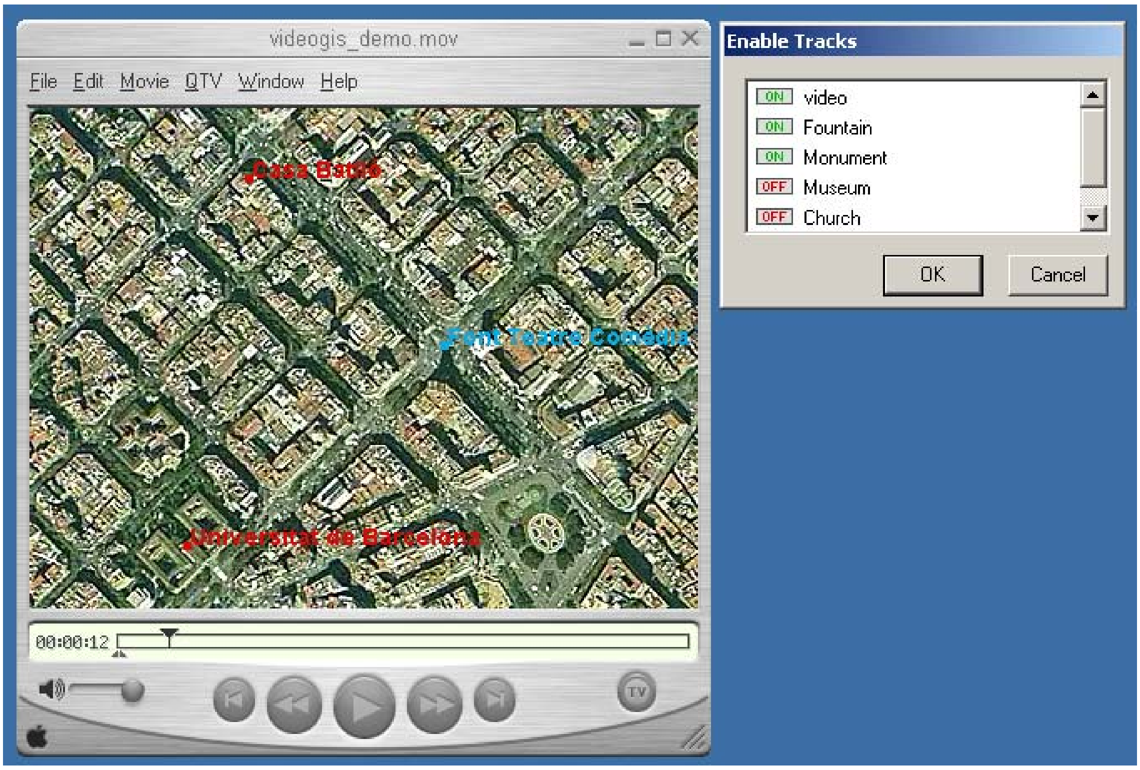


Figure 43. Indexing structure of VideoGIS prototype 1

The query system has been developed in Java, using QuickTime for Java, the Java API for QuickTime. A QuickTime movie is an object storing time-based data (audio, video, synchronized text) in QuickTime. It may be self-contained or can handle references to other sources of data. It also may contain more than one stream of data, each one called a track. When the movie is played in a QuickTime player, the user may hide or show these tracks (this can also be configured by the author). There is a special type of tracks

(HREF tracks) that allows hyper-linking. These links may be automatically activated or may need user clicks to traverse them.

The video resulting of a query is a QuickTime movie which contains a track with the video information and one track for every layer (type of feature) of the geographical information (see Figure 44). This way, the user (or the system) can hide or show layers (tracks) as in a GIS. An HREF track is also provided to link to the information of features.



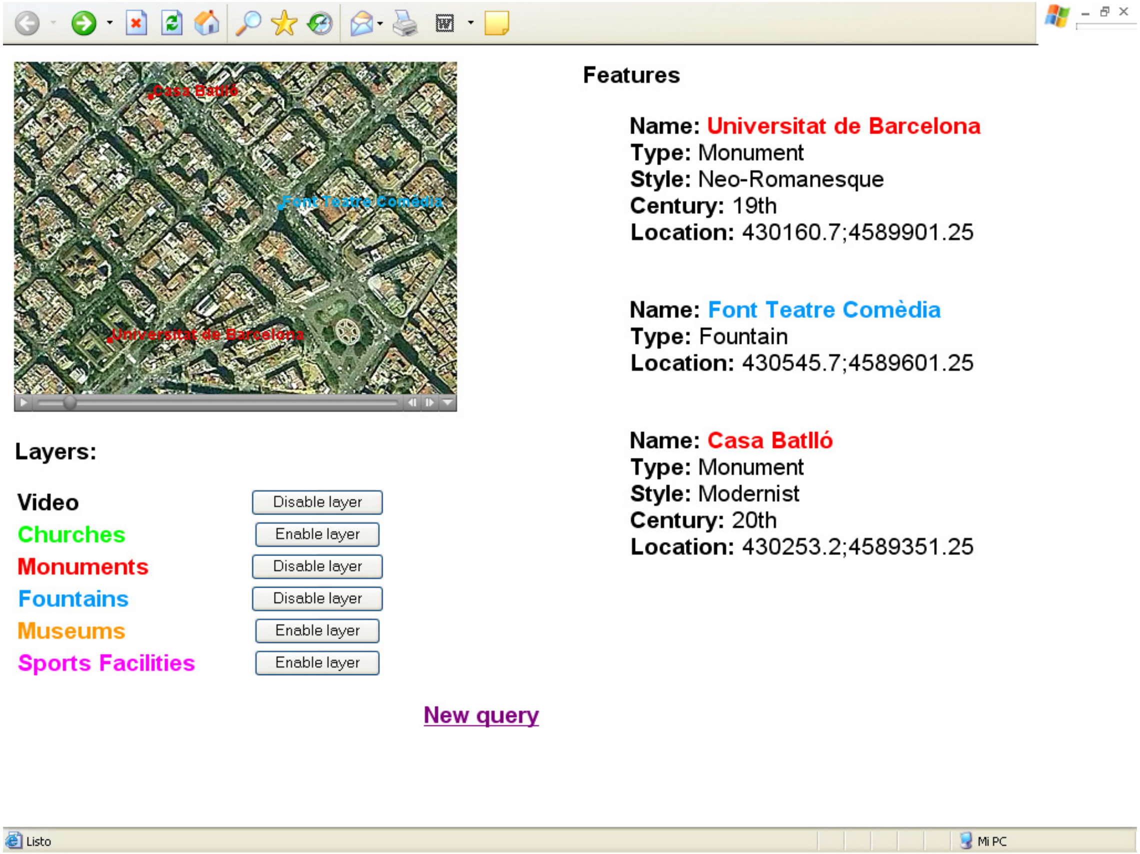
Enable Tracks

ON video
ON Fountain
ON Monument
OFF Museum
OFF Church

Figure 44. A VideoGIS video with several layers or layers played on QuickTime player

The prototype has been developed as a Java servlet, which is responsible of generating the final video and of embedding it an HTML page. A simple interface is provided to enable the user to control the video (through the QuickTime plug-in and JavaScript methods) and to show the information of the features as they are appearing in scene, as Figure 45 shows.

This prototype has been evaluated with geographic features corresponding to different types of points of interest (POIs) as monuments, museums, hotels or restaurants. This point-based information is not particularly dense and the feature-based segmentation algorithm produced good results. However, if other type of geographic information is considered, containing more complex and denser geometries, it would produce a micro-segmentation, i.e. a division in many too short segments, comprising very little frames (or even only one). In our particular case of thematic information, this approach would not be usable, and other type of video segmentation has to be designed.



Features

Name: Universitat de Barcelona
Type: Monument
Style: Neo-Romanesque
Century: 19th
Location: 430160.7;4589901.25

Name: Font Teatre Comèdia
Type: Fountain
Location: 430545.7;4589601.25

Name: Casa Batlló
Type: Monument
Style: Modernist
Century: 20th
Location: 430253.2;4589351.25

Layers:

Video	Disable layer
Churches	Enable layer
Monuments	Disable layer
Fountains	Disable layer
Museums	Enable layer
Sports Facilities	Enable layer

New query

Figure 45. VideoGIS web-based interface

11.3.2 VideoGIS prototype 2

This second prototype of VideoGIS aims at exploring how to deal with a collection of raster-based thematic datasets and needed a different approach to segmentation and indexing than the previous prototype based on features. In this case we also consider that videos may be filmed from the ground and each frame is geo-referenced in an XML file with some properties of the camera: location, orientation and angle of vision. This prototype was mainly developed during a stay of two months at the GIS service of Universitat de les Illes Balears in 2003.

Regarding segmentation, we have considered an approach where distance between the camera in the first frame of two consecutives segments is kept constant. This way, segments have not a fixed time length, but do correspond to a fixed spatial length. The distance between two segments is chosen by the expert and depends on the frame rate and on the size of raster cells.

Regarding indexing, each segment is described by means of a representative frame (or key-frame). In our implementation we have chosen the first frame of the segment as the representative one. For each key-frame, the system obtains its area of vision from its camera properties, up to a fixed distance. On the other hand, each raster dataset has several classes and the system counts how many cells of each class are contained in the area of vision that was obtained for the key-frame. This number of cells (in fact, the area measured in m^2) provides an indicator of the presence of this class in this frame.

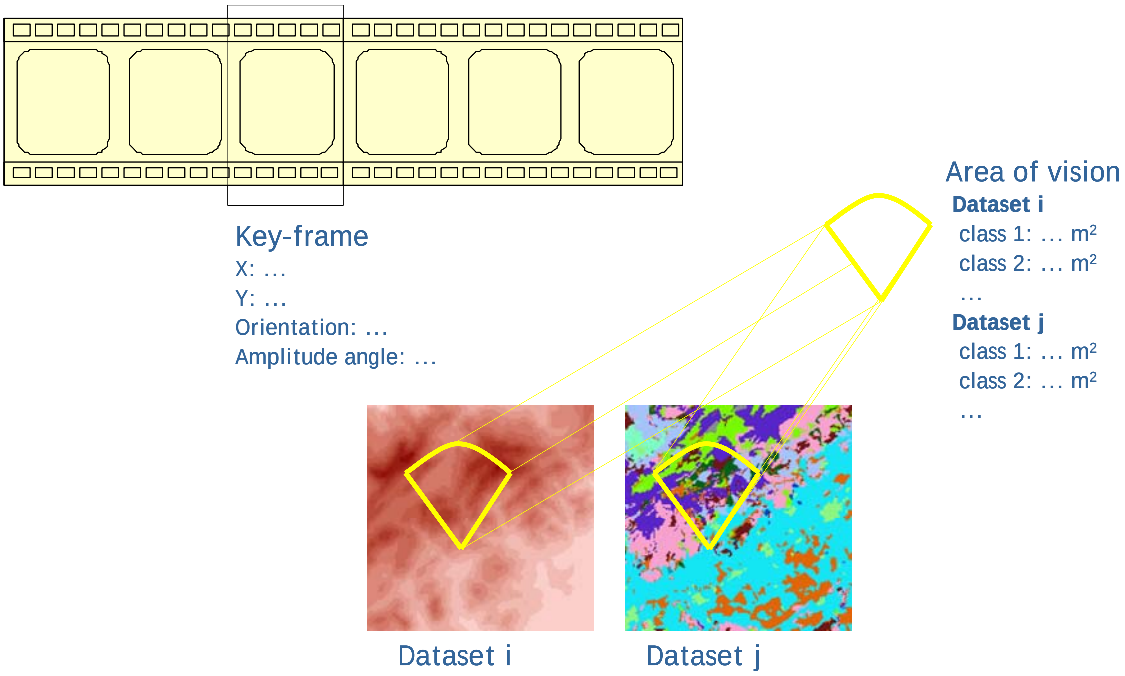


Figure 46. The process of indexing a segment in VideoGIS prototype 2

The segment is consequently indexed according to the dataset classes that are visible and to their area, like the following description file in XML shows. The system also uses a digital elevation model (*pathDEM*) and a viewshed analysis algorithm, and hence the distinction between the total area of vision (*totalArea*) and the area of vision discarding hidden zones (*totalVisibleArea*).

```
<video radius="1000" amplitudeAngle="90" framerate="25"
                                         pathDEM="ALTURES_STR2">
      <segment X="446182,416487004" Y="4381773,1264712"
            orientation="0" totalArea="79" totalVisibleArea="25"/>
            <dataset name="serra_boscos">
                  <class name="0" area="54" />
                  <class name="1" area="7" />
                  <class name="3" area="18" />
            </dataset>
            <dataset name="serra_ocupacio">
                  <class name="0" area="54" />
                  <class name="1" area="6" />
                  <class name="2" area="6" />
                  <class name="3" area="12" />
                  <class name="7" area="1" />
             </dataset>
      </segment>
      <segment X="457035,567861704" Y="4378491,94117187"
            orientation="0" totalArea="80" totalVisibleArea="36"/>
            ...
      </segment>
      ...
</video>
```

These segmentation and indexing algorithms have been developed as a tool running on ESRI's ArcGIS software and have been programmed in ArcObjects. To simulate geo-referenced videos we have also built a tool for ESRI ArcScene, where the user can draw a path and the system generates a video sequence from a 3D digital elevation model with the texture of the orthophoto of the Serra de Tramuntana (Majorca), as well as the coordinates file for the key-frames.

However, this new approach presents two main problems: on the one hand, the fixed spatial length segmentation misses areas that could be of interest in the middle of two key-frames. On the other hand, the indexing structure does not provide any semantic on the thematic information. For instance, this approach cannot reply questions like what is dataset "serra_boscos" about, what does class "3" in this dataset mean, or what relations between classes in different datasets exist. This way, a richer structure is needed to reflect the semantics of the thematic information in the datasets in the collection. In fact this was the motivation for this thesis on semantic representation of thematic geographic information and semantic interoperability issues. The following section describes the final prototype of VideoGIS, which uses the ontology of the repository and all the artefacts that were described in the previous chapters.

11.4 A semantic-based indexing and querying

In this section we describe how geo-referenced images and videos can be indexed and queried in the context of a digital library organized by geothematic content. The main type of query that the system has to support is, given a theme, retrieve the images or video segments depicting that theme.

Each image or video segment is indexed according to a set of themes from the ontology, $T_1,...,T_n$ (referred to as *indexing themes* from now on). They have to be previously selected by the user responsible of the indexing process. Each indexing theme gives rise to a different layer of meta-information of the image or video segment. This way, each layer of meta-information corresponds to a particular view of the thematic information of the image or video segment, focusing on one of the indexing themes.

The set of indexing themes may comprise qualitative themes usually with subclasses, quantitative themes with at least one quantitative classification, or modelled themes with at least one model. In the case of a quantitative theme having more than one classification, one classification has to be selected. Likewise, a model has to be selected for a modelled theme if it has more than one. It has to be noted that, although it is not compulsory, all the images and videos in the collection are usually indexed according to the same set of themes. However, there is no problem on indexing different images with different focus, i.e. using different sets of themes. For instance, an image can be indexed according to agricultural uses and soil salinity, while another according to forest uses and average temperature. Furthermore, $T_1,...,T_n$ are usually disjoint, or at least they do not have common subclasses in the thematic ontology. But again, this is not compulsory and the user may select themes with non-empty intersection.

It is also important to note that the indexing process is driven by this set of themes, and not by datasets. This way, a user previously selects the themes that s/he is interested in, but s/he must not know in which datasets this thematic information can be found. The system will be responsible for finding the involved datasets and dataset values.

11.4.1 Still image

Given a geo-referenced image I , the first step is to determine its area of vision A, which is obtained from the location and focal properties of the camera: camera position (X,Y,Z), orientation (ρ), tilt (θ), focal length (f) and receiver length (s) and width (w). Receiver length and width correspond to the size of the negative or CCD of the camera, measured in millimetres. The angle of vision (α) can be computed from f, s and w and will be used to obtain the area of vision.

We also use a Digital Elevation Model and a viewshed analysis algorithm to determine which areas can be seen from the camera position and which are hidden by elevations, up to a certain distance d from the camera. Consequently, the result of this process is a multi-polygon (or set of cells in a raster model) that corresponds to the area of vision A.

The thematic information of this area of vision A is used to index the image I, according to the indexing themes T_1,...,T_n. As it was mentioned above, the indexing process has n dimensions, one for each indexing theme. For each indexing theme T_i, the indexing process firstly obtains the involved dataset values, i.e. those connected to classes related to T_i. This is done by means of the first type of semantic query (see Section 10.1). Then, all the involved datasets are integrated according to T_i, through the filtered version of the third type of semantic query (see Section 10.3). The result of this operation is a new virtual dataset ds_i that comprises only the area of vision A and its values {C_{i1},...,C_{ip}} are direct references to classes of the ontology. The i-th dimension has p index entries. Each index entry stores the related theme and also the area of its spatial extent in ds_i, which provides an indication of the relevance of the value in the image. An index entry is consequently represented as the tuple:

$$< \mathtt{I},\ \mathtt{T}_{\mathtt{i}},\ \mathtt{C}_{\mathtt{ij}}\ ,\ \mathtt{area}_{\mathtt{ij}} >$$

where I is the image being indexed, identified through its URI, T_i is the indexing theme, C_{ij} is one of the themes of the virtual dataset ds_i comprising the area of vision, and $\mathrm{area}_{\mathrm{ij}}$ is the area of the spatial extent of C_{ij} in ds_i. Note that the set of visible themes at a given dimension may be empty. In this case, its corresponding index entry will not be created.

The following algorithm obtains the index entries for an image I with area of vision A. It has been simplified by assuming that datasets in the repository contain the same spatial units, as for instance in the case of raster datasets with the same tessellation.

```
for each $T_i \in \{T_1,...,T_n\}$ do
    $DsSet_i$ = query1InferenceDs($T_i$)
        //where $DsSet_i$ is a set of m datasets $\{ds_{i1,...,}\ ds_{im}\}$
    for each $ds_{ik} \in DsSet_i$ do
        $dsA_{ik}$ = region of dataset $ds_{ik}$ contained in A
    end for
    $ds_i$ = query3Filtered($dsA_{i1}$, ..., $dsA_{im}$, $T_i$)
        // where $ds_i$ is a virtual dataset with p values $\{C_{i1},...,C_{ip}\}$
    for each $C_{ij} \in \{C_{i1},...,C_{ijp}\}$ do
        add index entry $< I, T_i, C_{ij}, |e(C_{ij},ds_i)| >$
            // where $|e(C_{ij},ds_i)|$ is the area of
            //    the spatial extent of $C_{ij}$ in $ds_i$
    end for
end for
```

Note that given two indexing themes T_i and T_j, the total area of ds_i may be different to the total area of ds_j, since these datasets depend on the results of *query3*, which could even be empty if no subclass of the indexing theme is found in the area of vision.

Finally, if it is necessary, the thematic classes that have been used for indexing can be converted to any other vocabulary. This would be done through the second type of semantic query (see Section 10.2).

As it was mentioned above, the typical query in this digital library consists in obtaining the images that depict a given theme *T* in the ontology. The implementation of this query is as simple as retrieving those images having one index entry that contains a theme *C* subclass of *T* in its third element.

$$< \ldots, \ldots, C \sqsubseteq T , \ldots >$$

In case of several images to be returned, they are ordered according to a simple ranking algorithm that prioritizes those index entries $< I, T_i, C_{ij}, area_{ij} >$, where $C_{ij} \sqsubseteq T$, that maximize the ratio between $area_{ij}$ and the total area of vision of the image.

Finally, it is worth clarifying that this thematic-based approach does not exclude other information being used in parallel indexes. For instance, another layer could be used to include manual annotations, which could also be indexed in a separate structure permitting other non-thematic types of queries.

11.4.1.1 Image metadata and its relation with metadata standards

The images that are returned as the result of a query are provided with meta-information describing their thematic content. This way, clients can be developed to take profit of this meta-information.

A specific structure for this meta-information can be considered, having at most as many layers as selected indexing themes. Each layer contains a list of visible themes, related to the indexing theme, with the area of its spatial extent. Note that there may be less that *n* layers, since there may be indexing themes with no visible themes. Camera properties can also be inserted in the image meta-information. This will allow clients using a GIS to be able to extract further information. But also would allow other simpler clients to find spatial relations between images according to their camera location. The following XML code shows the meta-information of an image, while the schema can be found in Appendix G.

```
<Image uri="..." visibleArea="...">
      <CameraProperties>
            <CameraPosition>
                  <X>X</X>
                  <Y>Y</Y>
                  <Z>Z</Z>
            </CameraPosition>
            <CameraOrientation>ρ</CameraOrientation>
            <CameraTilt>θ</CameraTilt>
            <CameraAngleOfVision>α</CameraAngleOfVision>
      </CameraProperties>
      <Layer indexingTheme="T1">
            <VisibleTheme theme="C11" spatialExtentArea="area11" />
            <VisibleTheme theme="C12" spatialExtentArea="area12" />
            ...
            <VisibleTheme theme="C1p spatialExtentArea="area1p />
      </Layer>
      <Layer indexingTheme="T2">
            <VisibleTheme theme="C21" spatialExtentArea="area21" />
            ...
      </Layer>
      ...
      <Layer indexingTheme="Tn">
            ...
      </Layer>
</Image>
```

Note that this meta-information contains both camera properties and the thematic information.

Although specific clients can be developed according to the metadata structure explained above, we now analyze how it can be related to clients relying on metadata standards. Apart from MPEG-7, we have also considered Dublin Core (ISO 2003b; DCMI 2005), which provides a set of elements for a generic description of a resource.

In the case of Dublin Core, it provides a keyword element for describing the resource. In our case, we can use this keyword to include all the themes in all the layers:

```
      dc:keyword="C11, C12, ..., C1p, C21,..."
```

Note that the area of the spatial extent of these themes cannot be provided, and consequently no indicator on the relevance of each theme in the image is given.

In the case of MPEG-7, in the Semantic part of the image description, a *SemanticBase* descriptor can be used to include each geographic theme appearing in the image. A theme could be seen as either a concept or an object. Although some thematic information can be seen in the image (land cover is a clear example), many of them cannot be visually appreciated (temperature, average income,...). Consequently, we have chosen to represent themes as concepts.

Furthermore, an image has at most as many semantic descriptions (semantic worlds in the terminology of MPEG-7) as indexing themes. Note that we say “at most” since there may be layers with no visible themes, and consequently no semantic description is provided at those layers. Each semantic world is related to its indexing theme through the normative relation *embodiedIn*, which according to the standard indicates that the semantic base (the indexing theme) symbolizes in some sense the semantic world (the image semantics).

```
<Image>
      <Semantic id="layer_1">
            <!-- Semantic concept and relation for indexing theme -->
            <SemanticBase xsi:type="ConceptType" id="c1">
                  <Label><Name>T1</Name></Label>
            </SemanticBase>
            <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:
                              2001:embodiedIn" target="#c1"/>

            <!-- Semantic concepts for p visible themes C1k -->
            <SemanticBase xsi:type="ConceptType" id="...">
                  <Label><Name>C11</Name></Label>
            </SemanticBase>
            ...
            <SemanticBase xsi:type="ConceptType" id="...">
                  <Label><Name>C1p</Name></Label>
            </SemanticBase>
      </Semantic>
      ...
      <Semantic id="layer_n">
            ...
      </Semantic>
</Image>
```

Note that C_{11} or C_{12} represent classes in the repository ontology, and their URI would be used in this semantic description.

The global area of vision of the image is described by means of a *SemanticBase* of type *SemanticPlaceType* and through its *Extent* element.

```
      <SemanticBase xsi:type="SemanticPlaceType" id="visibleArea">
            <Label> <Name>Visible Area</Name> </Label>
            <SemanticPlaceInterval>
                  <Extent measurementType="area"
                              unit="hectares" value="10.5" />
            </SemanticPlaceInterval>
      </SemanticBase>
```

As we have already mentioned, the area of the spatial extent of a theme in the image provides a significant indicator of its relevance. It is described here through a *SemanticBase* element of the type *SemanticStateType*, and its *AttributeValuePair* element.

```
      <SemanticBase xsi:type="SemanticStateType" id="c11sea">
            <Label> <Name>SpatialExtentArea</Name> </Label>
            <AttributeValuePair>
                  <Attribute><Name>spatialExtentArea</Name></Attribute>
                  <Unit><Name>hectares</Name></Unit>
                  <FloatValue>3.5</FloatValue>
            </AttributeValuePair>
      </SemanticBase>
```

This state is related to its thematic concept through a *Relation* element in the *Graph* of relations of a semantic world of the image.

```
<Semantic id="...">
      <!-- Semantic concept and relation for indexing theme -->
      <SemanticBase xsi:type="ConceptType" id="c1">...</SemanticBase>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:
                              2001:embodiedIn" target="#c1"/>

      <!-- Semantic place for visible area -->
      <SemanticBase xsi:type="SemanticPlaceType" id="visibleArea">
            ...
      </SemanticBase>

      <!-- Semantic concepts for themes -->
      <SemanticBase xsi:type="ConceptType" id="c11">...</SemanticBase>
      <SemanticBase xsi:type="ConceptType" id="c12">...</SemanticBase>
      ...

      <!-- Semantic states for spatial extents -->
      <SemanticBase xsi:type="SemanticStateType" id="c11sea">
            ...
      </SemanticBase>
      <SemanticBase xsi:type="SemanticStateType" id="c12sea">
            ...
      </SemanticBase>
      ...

      <!-- Semantic relations -->
      <Graph>
            <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:
              2001:stateOf" source="#c11sea" target="#c11"/>
            <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:
              2001:stateOf" source="#c12sea" target="#c12"/>
      </Graph>
</Semantic>
```

It has to be noted that every semantic world of the image will contain both the semantic place element for the visible area and semantic states for spatial extent areas.

Finally, the camera properties can also be specified in the *CreationInformation* part of the image description. Note that the visible area of the image was considered as part of the narrative world and included in the semantic description, but since these camera properties refer to the physical world are included in the *CreationInformation* part.

```
<CreationInformation>
  <Creation>
    <CreationCoordinates>
      <Location>
        <GeographicPosition datum="...">
          <Point longitude="X" latitude="Y" altitude="Z" />
        </GeographicPosition>
      </Location>
    </CreationCoordinates>
  </Creation>
</CreationInformation>
```

MPEG-7 makes it possible the definition of new descriptors through DDL. This way, a *VideoGISCreationCoordinates* can be defined by extending *CreationCoordinates*, containing the camera location as well as the other parameters. Nevertheless, since these descriptors gets off the standard, they cannot be used by off-the-shelf clients.

```
<CreationInformation>
  <Creation>
    <VideoGISCreationCoordinates>
      <Location>
        <GeographicPosition datum="...">
          <Point longitude="X" latitude="Y" altitude="Z" />
        </GeographicPosition>
      </Location>
      <VideoGISCameraOrientation>ρ</VideoGISCameraOrientation >
      <VideoGISCameraTilt>θ</VideoGISCameraTilt>
      <VideoGISCameraAngleOfVision>α</VideoGISCameraAngleOfVision>
    </VideoGISCreationCoordinates>
  </Creation>
</CreationInformation>
```

11.4.2 Video

The process of indexing geo-referenced videos follows a similar approach than for indexing geo-referenced still images. Videos are also indexed according to a set of indexing themes, $T_1,...,T_n$, that as in the case of still images, have to be previously selected from the ontology by the user in charge of the process. Our video model is based on objects or strata, where each selected indexing theme corresponds to a stratum. We will see that the video is firstly segmented according to the thematic information, and then, each segment will be indexed in a similar way as described for a still image.

As in the case of still images, each frame of a geo-referenced video V has several properties concerning the camera: camera position (X,Y,Z), orientation (ρ), tilt (θ) and focal length (f). Note that the value of these properties may change frame-by-frame. On the other hand, two other properties of the camera, receiver length (s) and width (w), are constant along the whole video. The angle of vision (α) can be computed from f, s and w. These properties make it possible to obtain the area of vision A_j for a given frame f_j. Furthermore, a new property is needed to state the frame rate (fps), which is also constant along the video.

Segmentation is carried out according to the thematic information in the area of vision of each frame of the video. The process is equivalent to the feature-based segmentation described in 11.3.1, where each segment represents a sequence of frames containing the same features or themes in this case. However, this process is done for each indexing theme, and consequently a parallel structure of *n* segmentations is generated. From now on, we denote by *layer* each of these different *n* segmentations. This way, each frame in the video belongs to at most *n* different segments, one for each layer or indexing theme. As in the case of still images, we say "at most" since segments with no visible themes are not considered. In the following segmentation algorithm, index *i* is used for indexing themes and *j* for frames:

```
for each $T_i \in \{T_1,...,T_n\}$ do
      $DsSet_i$ = query1InferenceDs($T_i$)
            //where $DsSet_i$ is a set of m datasets $\{ds_{i1,...,} ds_{im}\}$
      for each frame $f_j$ in V do
            for each $ds_{iu} \in DsSet_i$ do
                  $dsA_{iuj}$ = region of $ds_{iu}$ contained in $A_j$
            end for
            $ds_{ij}$ = query3Filtered($dsA_{i1j}$, ..., $dsA_{imj}$, $T_i$)
                  // where $ds_{ij}$ has p values $\{C_{ij1},...,C_{ijp}\}$
            if $\{C_{ij1},...,C_{ijp}\} \neq \{C_{i(j-1)1},...,C_{i(j-1)q}\}$ then
                  if $j \neq 0$ then
                        finish segment at layer i
                  end if
                  start new segment at layer i
            end if
      end for
      finish segment at layer i
end for
```

Each segment obtained through the abovementioned process is indexed in a similar way to how a still image was. There are only two slight differences. On the one hand, the index entry has now six elements since a starting and ending frame are needed for each segment, apart from the reference to the video. On the other hand, the area of the spatial extent is now obtained from the average of every frame in the segment. An index entry for a video segment is consequently represented as the tuple:

$$< V,\ T_i,\ sf_i,\ ef_i,\ C_{ik}\ ,\ area_{ik} >$$

where *V* is the video (identified through its URI), T_i is the indexing theme, sf_i and ef_i are respectively the starting and ending frame of the segment being indexed at layer *i*, C_{ik} is one of the visible themes at the segment at layer *i*, and $area_{ik}$ is the average of the areas of the spatial extent of C_{ij} at every frame in the segment at layer *i*. Note that the set of visible themes at a given layer may be empty. In this case, the index entry would not be created.

Each frame *j* at layer *i* has a set of *p* visible themes $\{C_{ij1},...,C_{ijp}\}$. If this set of visible themes is different from the set of the previous frame $\{C_{i(j-1)1},...,C_{i(j-1)q}\}$ (note that they do not necessarily have the same number of visible themes *p* and *q*), it indicates a change of segment. The following algorithm includes both segmentation and indexing. We use index *i* for indexing themes, index *j* for frames, and index *k* for visible themes in a frame or segment. The variable $areaC_{ik}$ is used to store the sum of the areas of the spatial extent of C_{ijk} at every frame *j* along the current segment at layer *i*, while the variable sf_i stores the frame where the current segment at layer *i* started. Note that segments with an empty set of visible themes are not indexed.

```
for each $T_i \in \{T_1,...,T_n\}$ do
  $DsSet_i$ = query1InferenceDs($T_i$)
  for each frame $f_j$ in V do
    for each $ds_{iu} \in DsSet_i$ do
      $dsA_{iuj}$ = region of $ds_{iu}$ contained in $A_j$
    end for
    $ds_{ij}$ = query3Filtered($dsA_{i1j}$, ..., $dsA_{imj}$, $T_i$)
            // where $ds_{ij}$ has p values $\{C_{ij1},...,C_{ijp}\}$
    if $\{C_{ij1},...,C_{ijp}\} \neq \{C_{i(j-1)1},...,C_{i(j-1)q}\}$ then
      if $j \neq 0$ then // finish segment at layer i:
        for each $C_{i(j-1)k} \in \{C_{i(j-1)1},...,C_{i(j-1)q}\}$ do
          add index entry <V, $T_i$, $sf_i$, j-1, $C_{i(j-1)k}$, $areaC_{ik}/(j-sf_i)$>
        end for
      end if
      // start new segment at layer i:
      $sf_i$ = j
      for each $C_{ijk} \in \{C_{ij1},...,C_{ijp}\}$ do
        $areaC_{ik}$ = $|e(C_{ijk},ds_{ij})|$
      end for
    else // continue segment at layer i:
      for each $C_{ijk} \in \{C_{ij1},...,C_{ijp}\}$ do
        $areaC_{ik}$ = $areaC_{ik}$ + $|e(C_{ijk},ds_{ij})|$
      end for
    end if
  end for
  //finish segment at layer i:
  for each $C_{ijk} \in \{C_{ij1},...,C_{ijp}\}$ do
    add index entry <V, $T_i$, $sf_i$, j, $C_{ijk}$, $areaC_{ik}/(j-sf_i+1)$>
  end for
end for
```

Since two consecutive frames usually have the same thematic information, a simple optimization can be done to this algorithm by only checking sets of themes each *F* frames. *F* is a parameter that the user may modify, and its default value is the same as the property *fps*. This indicates that, by default, a comparison is made for each second of video. If the comparison between frames f_a and f_{a+F} results in different sets of themes, f_a will be then compared with $f_{a+F/2}$. The distance between frames being compared is successively divided by two until the exact frame for the end of segment is found.

There are two types of queries for videos in this digital library. The simplest one is the retrieval of a complete video in its original order. The other, analogously to the typical query for still images, is the retrieval of the video segments that depict a given theme *T* in the ontology. The implementation of this query is also similar to the one for images: it will retrieve those video segments with one index entry containing a theme *C* subclass of *T* in its fifth element.

$$< ..., ..., ..., ..., C \sqsubseteq T , ... >$$

In case of several segments to be returned, they are also ordered according to a simple ranking algorithm that prioritizes those index entries that maximize the ratio between $areaC_{ik}$ and the average area of vision of the segment.

Finally, as in the case of still images, it is also worth clarifying that this thematic-based approach does not exclude other information being used in parallel segmentations and indexes. For instance, another layer could be used to include manual annotations, which would determine a separate segmentation layer with its own indexing structure permitting non-thematic types of queries.

11.4.2.1 Video metadata and its relation with metadata standards

The complete video or the set of video segments that are returned as the result of one of the two types of queries, has to be provided with meta-information describing its thematic content. This way, clients can be developed to take profit of this meta-information.

As in the case of images, we firstly describe a specific structure for this meta-information, and later we discuss how Dublin Core and MPEG-7 can be used for that purpose. A video segment comprises only a layer of meta-information, containing its indexing theme and a list of visible themes (related to the indexing theme) with their spatial extent area. The segment meta-information also includes the camera properties of a representative frame, which will enable clients to find spatial relations between video segments according to their camera location. The following XML code shows the meta-information of a video segment, while the schema can be found in Appendix G.

```
<VideoSegment video="..." indexingTheme="T_i"
            startFrame="s" endFrame="e" averageVisibleArea="...">
      <CameraPropertiesAtFrame frameNumber="...">
            <CameraPosition>
                  <X>X</X>
                  <Y>Y</Y>
                  <Z>Z</Z>
            </CameraPosition>
            <CameraOrientation>ρ</CameraOrientation>
            <CameraTilt>θ</cameraTilt>
            <CameraAngleOfVision>α</CameraAngleOfVision>
      </CameraPropertiesAtFrame>
      <VisibleTheme theme="C_1e1" spatialExtentArea="area_11" />
      <VisibleTheme theme="C_1e2" spatialExtentArea="area_12" />
      ...
      <VisibleTheme theme="C_1ep spatialExtentArea="area_1p />
</VideoSegment >
```

In the case of visualizing a complete video, the *Video* element contains a list of *VideoSegment* elements, as in the following fragment of XML code (the schema can be found in Appendix G):

```
<Video uri="..." fps="fps">
      <VideoSegment video="..." indexingTheme="T1" startFrame="s11"
                              endFrame="e11" averageVisibleArea="...">
            ...
      </VideoSegment >
      <VideoSegment video="..." indexingTheme="T1" startFrame="s12"
                              endFrame="e12" averageVisibleArea="...">
            ...
      </VideoSegment >
      ...
      <VideoSegment video="..." indexingTheme="T2" startFrame="s21"
                              endFrame="e21"  averageVisibleArea="...">
            ...
      </VideoSegment >
      ...
</Video>
```

Since Dublin Core is a generic schema, it is not appropriate to represent the structure of parallel segments with different visible themes each. However, as in the case of images, a keyword-based description can be provided for each segment, containing only those visible themes related to one indexing theme.

In the case of MPEG-7, as it was mentioned in the description of the standard (see 11.2.3), it supports both segmentation and stratification approaches, and can be used to implement our model. This way, a *VideoSegment* element is associated to every segment obtained through our thematic-based segmentation algorithm. It comprises a *MediaTime* element that contains the time reference to the original sequence, and also a semantic description (semantic world). A *SemanticBase* of type *ConceptType* is used to describe each theme appearing in a video segment. Furthermore, as in the case of a still image, the indexing theme for a segment is represented through a concept related to the segment through the normative relation *embodiedIn*. However, it is worth recalling here that a video segment only corresponds to one layer, while a still image contains at most *n* layers, one for each of the *n* indexing themes. This way, a video segment only has one semantic world, but each frame of the segment will belong to at most *n* segments (one for every layer). Finally, a *SemanticPlace* element is used to represent the average area of vision of the video segment.

```
<VideoSegment id="vs11">
      <MediaTime>
            <MediaTimePoint>...</MediaTimePoint>
            <MediaDuration>...</MediaDuration>
      </MediaTime>
      <Semantic id="vs11sem">
            <!-- Semantic concept and relation for indexing theme -->
            <SemanticBase xsi:type="ConceptType" id="c1">
                  <Label><Name>T1</Name></Label>
            </SemanticBase>
            <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:
                  2001:embodiedIn" target="#c1"/>
            <!-- Semantic place for the visible area -->
            <SemanticBase xsi:type="SemanticPlaceType" id=" vs11VArea">
                  <Label> <Name>Visible Area</Name> </Label>
                  <SemanticPlaceInterval>
                        <Extent measurementType="area"
                                    unit="hectares" value="10.5" />
                  </SemanticPlaceInterval>
            </SemanticBase>

            <!-- Semantic concepts for visible themes -->
            <SemanticBase xsi:type="ConceptType" id="c11">
                  <Label><Name>C11</Name></Label>
            </SemanticBase>
            ...
            <SemanticBase xsi:type="ConceptType" id="c1p">
                  <Label><Name>C1p</Name></Label>
            </SemanticBase>

            <!-- Semantic states for spatial extent areas of themes -->
            <SemanticBase xsi:type="SemanticStateType" id="c11sea">
                  <Label> <Name>SpatialExtentArea</Name> </Label>
                  <AttributeValuePair>
                        <Attribute>
                              <Name>spatialExtentArea</Name>
                        </Attribute>
                        <Unit><Name>hectares</Name></Unit>
                        <FloatValue>3.5</FloatValue>
                  </AttributeValuePair>
            </SemanticBase>
            ...
            <SemanticBase xsi:type="SemanticStateType" id="c1psea">
                  ...
            </SemanticBase>

            <!-- Semantic relations between extent areas and themes -->
            <Graph>
                  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:
                    2001:stateOf" source="#c11sea" target="#c11"/>
                  ...
                  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:
                    2001:stateOf" source="#c1psea" target="#c1p"/>
            </Graph>
      </Semantic>
</VideoSegment>
```

And in the case of a complete video, concepts related to themes are defined in the semantic world of the video. The video has several temporal decompositions, one for each layer (indexing theme). Each temporal decomposition contains several non-overlapping video segments, each of them related to the concept of its indexing theme, through the normative relation *embodiesIn,* and to its visible themes through the normative relation *mediaReferenceOf* (that according to the standard is used when the semantic base is not seen in the video). Furthermore, each video segment contains a semantic description that includes a semantic state with the area of the spatial extent of each semantic concept in the segment.

```
<Video>
      <Semantic>
            <!-- Semantic concepts for indexing themes -->
            <SemanticBase xsi:type="ConceptType" id="c1">
                  <Label><Name>T1</Name></Label>
            </SemanticBase>
            <SemanticBase xsi:type="ConceptType" id="c2">
                  <Label><Name>T2</Name></Label>
            </SemanticBase>
            ...
            <!-- Semantic concepts for visible themes -->
            <SemanticBase xsi:type="ConceptType" id="c11">
                  <Label><Name>C11</Name></Label>
            </SemanticBase>
            <SemanticBase xsi:type="ConceptType" id="c12">
                  <Label><Name>C12</Name></Label>
            </SemanticBase>
            ...
            <Graph>
               <!-- Semantic relations between each video segment
                        and its indexing theme -->
               <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:
                  2001:embodiedIn" source="#vs11sem" target="#c1"/>
               <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:
                  2001:embodiedIn" source="#vs12sem" target="#c1"/>
               <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:
                  2001:embodiedIn" source="#vs21sem" target="#c2"/>
               ...

               <!-- Semantic relations between each video segment
                        and its visible themes -->
               <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:
                  2001:mediaReferenceOf" source="#c11" target="#vs11"/>
               <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:
                  2001:mediaReferenceOf" source="#c12" target="#vs11"/>
               <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:
                  2001:mediaReferenceOf" source="#c11" target="#vs21"/>
               ...
            </Graph>
      </Semantic>

      <!-- Temporal decomposition, one for indexing theme (layer) -->
      <TemporalDecomposition id="td1">
            <!-- Video segments -->
            <VideoSegment id="vs11">
                  <MediaTime>...</MediaTime>
                  <Semantic id="vs11sem">
                        <!-- Semantic place for the visible area -->
                        <SemanticBase xsi:type="SemanticPlaceType"
                                    id="vs11VArea">
                              ...
                        </SemanticBase>
                        <!--Semantic state for spatial extent areas and
                            its relations with thematic concepts-->
                        <SemanticBase xsi:type="SemanticStateType"
                                          id="c11sea11">
                              <Label>
                                    <Name>SpatialExtentArea</Name>
                              </Label>
                              <AttributeValuePair>
                                    <Attribute>
```

```
                                        <Name>spatialExtentArea</Name>
                                    </Attribute>
                                    <Unit><Name>hectares</Name></Unit>
                                    <FloatValue>3.5</FloatValue>
                              </AttributeValuePair>
                        </SemanticBase>
                        <SemanticBase xsi:type="SemanticStateType"
                                          id="c12sea11">
                              ...
                        </SemanticBase>
                        ...
                        <Graph>
                              <Relation type="urn:mpeg:mpeg7:cs:
                                SemanticRelationCS:2001:stateOf"
                                source="#c11sea11" target="#c11"/>
                              <Relation type="urn:mpeg:mpeg7:cs:
                                SemanticRelationCS:2001:stateOf"
                                source="#c12sea11" target="#c12"/>
                              ...
                        </Graph>
                  </Semantic>
            </VideoSegment>
            <VideoSegment id="vs12">
                  <MediaTime>...</MediaTime>
                  <Semantic id="vs12sem">...</Semantic>
            </VideoSegment>
            ...
      </TemporalDecomposition>

      <TemporalDecomposition id="td2>
            <VideoSegment id="vs21">
                  <MediaTime>...</MediaTime>
                  <Semantic id="vs21sem">...</Semantic>
            </VideoSegment>
            ...
      </TemporalDecomposition>
      ...
</Video>
```

Finally, as in the case of still images, the camera properties can be captured in the *CreationInformation* part. In this case, all the properties (camera position, camera orientation ρ, camera tilt θ and camera angle of vision α) vary along the video. Consequently, these properties are captured for a representative frame at each segment and the *CreationInformation* is associated to each segment.

11.5 Further research directions

The previous section has described a digital library that retrieves video segments satisfying a thematic query. Nevertheless, other significant issues arise regarding presentation and navigation of the resulting video segments on the one hand, and their spatial reference on the other. Although they are out of the scope of this PhD work, we point out some topics that we would like to address in the future.

Storyboard is the most widely used metaphor for showing the results of searches. Following this metaphor, a video is represented by means of a set of temporarily ordered shot's key-frames that are simultaneously presented on screen as thumbnails.

This type of interface avoids downloading big amounts of video data and provides a quick general view of the video. In the case of queries returning several video segments, each one is represented by means of a thumbnail of one of its frames. Storyboards are used in the majority of video retrieval systems. For instance, (Christel 2006) describes this type of interface in Informedia, and (Smeaton and Lee 2002) in the Físchlár Digital Video Library. These interfaces also provide more elements as a relevance indicator or a title. However, storyboards may be too long and thus, become useless. Furthermore, they do not provide relations between the returned segments.

To partially avoid these problems, another type of interface was developed, called video collage. It aims at visualizing news video segments that results from a query, together with their respective contexts in Informedia (Christel et al. 2002; Ng et al. 2003). A video collage contains a rectangular panel where news video segments are placed by means of thumbnails, as well as text lists showing the “who,” “what,” “when,” and “where” information of the selected video segments. The panel may be organized either spatially or temporarily. In the spatial distribution, segments are placed on a map according to the location names that they contain. In the temporary distribution, video segments are organized according to two axes: the vertical axe is the relevance of the news video segment in respect to the query, while the horizontal axe corresponds to the date when the news were broadcasted. The user may change the granularity of the panel in both cases in order to properly visualize crowded spatial regions or time periods.

For our system, we would like to explore a different approach that goes beyond a retrieval system aiming at a hypermedia presentation generator. This way, given a thematic query, the system returns not just a list of resulting video segments, but a hypermedia presentation dynamically generated. This presentation contains two main parts: on the one hand, a video sequence dynamically generated from a selection among the resulting video segments, and on the other hand, a textual information describing the thematic information attached to video segments and suggesting links to other information or segments. We call *video-itinerary* to this dynamic presentation, since the result of searching a certain theme is a kind of visual itinerary along the locations where this theme is present. How video-itineraries are dynamically generated and how they can be navigated to build a hypervideo network are the two key issues here.

Regarding the first issue, the generation of the video-itinerary, the most relevant research topic is how the composition of segments can provide a narrative thread to the final sequence. Different montage (also called editing, mainly in USA) theories have been elaborated along the history of cinema that could be used in the context of a narrative-aware composition. Realism filmmaking movement considered that a film should concentrate on reflecting the reality, giving a minimum role to the film director. According to this principle, the semantics of a film is entirely provided by the *mise-en-scène*, i.e. what can be seen in a shot. On the other hand, Expressionism and Formalism focused on the communication between the filmmaker and the audience. Particularly Soviet formalist filmmakers and theorists Pudovkin and Einsestein gave a prominent semantic role to montage, especially important in silent films. Pudovkin defined the montage as the method which controls the “psychological guidance” of the spectator. Einsestein considered that montage means a collision between shots rather than a linkage, and focused his attention on how new meanings arise when shots are put together, and on their psychological impact on the spectator. This dichotomy is nowadays clearly overcome, and it is widely recognized that both montage and *mise-en-scène* are important elements in the film narrative. Nevertheless, some modern filmmaking movements as Dogma 95 led by Lars Von Trier again refuses the use of montage for modifying the film semantics, among other rules (known as the Vow of Chastity) that search what they consider the purity of films.

In our case, *mise-en-scène* is described through the thematic information associated to segments, but the way they are selected and temporarily organized may provide extra narrative elements to the final sequence. Montage treatises contain different techniques that have to be considered for giving continuity to sequences and providing them with the desired narrative (for instance temporal or spatial change). Although techniques based on sound and on characters (for instance in a dialogue sequence) cannot be applied in our context, other techniques could be considered. Particularly relevant for our context is documentary montage theory. In our case, montage may be influenced by different factors, including the relevance of the searched theme in the obtained video segments, or the location of the camera in video segments in order to group them according to spatial criteria providing the feeling of an itinerary. In this context, the user may also be interested in specifying an expected duration for this video and a type of rhythm, factors that would also have influence on the selection and composition of segments. Detection of zoom operations or other typical camera movements could also be considered.

The other relevant research issue is navigation, which is determined according to two main axes: thematic information and spatial location. This way, links can be automatically created from a segment to other segments containing related geographic themes or to segments that belong to the same spatial area. Other spatial relations between segments could also determine links, like segments closer to a certain distance or even those with a location that can be reached following a certain direction (North, South, East or West) from the current segment. The result is a hypervideo network that is dynamically built.

Finally, we would like to briefly discuss here an issue that has not been considered in the current implementation of VideoGIS. We have focused on providing thematic information to segments since interoperability of thematic information is the main topic of this PhD. This way, we have deliberately not considered the spatial reference of segments. Nevertheless, storing the spatial extent of each theme that appears in a video segment would be a minor extension to our model. This approach has two main drawbacks: on the one hand it requires a considerable amount of space to store all these polygons, and on the other hand only specific clients capable to deal with topologic information may take profit of it. The main advantage of this approach is that it would make it possible to define queries with spatial relations (for instance “road infrastructures touching forests” or “water lands closer than 5 km to dumping sites”). It also would provide refined spatial relations for hypervideo linking.

Another step ahead in this direction would be the mapping between these spatial extents and moving regions in video segments. This would probably need a calibration operation considering known elements in both the image and datasets. This way, the screen coordinates for the moving region, related to the spatial extent of a theme, could be stored. This would make it possible to overprint information on moving regions (for instance painting them with different colours). Furthermore, anchors for links could be defined on specific regions of the screen. Topological relations between moving regions could also be considered (for instance “forests above wetlands”).

12 Conclusions and future research directions

This chapter summarizes the main results that have been obtained in this work, and also mentions the main directions of future research work.

12.1 Conclusions and main contributions

We have presented in Chapter 4 a formal conceptual model that addresses the main problem that motivates this thesis. Geographic datasets are structured in different ways depending on the particular needs of the dataset author. Thematic concepts can also be defined in different ways by different authors (modelled themes). In this context, integrating geographic information from different datasets presents a significant challenge from the semantic point of view. One of the elements of a new semantic framework for the integration is our conceptual model which is specified by means of an ontology based on the constructors that Description Logic offers. Particularly, it is expressed in the DL profile of OWL, which enables standard reasoners as Racer or FaCT to deal with it. Our ontology provides a semantic framework for a repository of datasets and known vocabularies, representing their thematic concepts as well as the semantic relations among them. Furthermore, our ontology supports definitions of thematic concepts in terms of other thematic concepts through DL axioms.

This ontology is built by adding new datasets or vocabularies. This involves a merging process (Chapter 5) that relates the application ontology of the new dataset or vocabulary to the ontology representing the overall repository. This process usually involves an almost flat and small ontology (from the dataset) and a bigger and denser one (from the repository). We have developed a manual method that enables a domain expert to determine the mappings between classes in both ontologies. The manual method presents some differences in the cases of qualitative and quantitative datasets. Furthermore, a semi-automatic merging method has also been defined, which automatically generates a list of suggested mappings that can be confirmed or modified by the domain expert. This list of suggestions is generated by the so-called mapping algorithm. The main contribution related to merging is the definition of three different mapping algorithms. These mapping algorithms address the specificities of our context, but are generic for merging datasets presenting a hierarchical structure of themes (taxonomy).

The string-based mapping algorithm (Chapter 6) considers the similarities between the names of classes, as well as the structure of the ontologies. Three significant contributions presented in this chapter have to be mentioned: the definition of an asymmetric similarity measure between two class names; the mechanism of mapping restrictions that drives the algorithm by constraining the classes that can be mapped to each dataset class; and finally, the mechanism of structural rules that enables the algorithm to infer new mappings according to the structure of the ontologies being mapped.

The terminological mapping algorithm (Chapter 7) uses a terminological base, such as the WordNet lexical base or the GEMET thesaurus of environmental terms, to extend terms in class names with their synonyms, hyponyms and hypernyms (and in some cases also with meronyms and holonyms). Two contributions presented in this chapter have to be mentioned: on the one hand, the definition of an asymmetric score measure which is based on synonyms, hyponyms and hypernyms; and on the other hand, the mechanism of the so-called term mapping restrictions which makes it possible for the algorithm to determine the type of relation between two classes. This algorithm also uses the mapping restrictions and structural rules mechanisms defined in Chapter 6.

The mapping algorithm based on the spatial distribution of dataset values (Chapter 8), also called spatial mapping algorithm, determines the relations among classes from the level of overlapping of their respective spatial units. This mapping algorithm presents significant advantages with respect to other existing approaches on this issue: it is more flexible, it supports many-to-many equivalence relations and it can be computed in real time, even for big real datasets.

Chapter 9 stresses the lack of a widely-used framework for evaluating ontology merging/alignment. This lack is particularly remarkable in the case of ontologies with a hierarchical structure like ours, where no benchmark exists. Furthermore, precision and recall are not good indicators of the quality of a merging/alignment, since they do not consider how close an obtained mapping is to the corresponding reference mapping. A key contribution of this chapter is the definition of a relaxed precision and recall specific for ontology merging/alignment. These relaxed measures are based on the intersection of the sets of axioms that can be inferred from the mappings. We have conducted some significant experiments to evaluate the string-based and terminological mapping algorithm with real datasets. We have observed that the use of the terminological algorithm always increments recall with respect to the string-based approach, but sometimes makes precision to slightly decrease. Relaxed precision ranges between 70% and 100% depending on the experiment. In the experiments described in Section 9.4, relaxed recall ranges from 40% (with the string-based mapping algorithm) to 70% (with the terminological mapping algorithm). In the experiments described in Section 9.5, where relaxed recall has not been computed, one mapping (or more) is provided for more than 85% of the classes. Comparing our approach to other merging tool such as PROMPT, ours obtains a similar precision, but with a significantly higher recall.

Our ontology defined in Chapter 4 provides a semantic framework supporting the definition of semantic services (usually referred to as semantic queries). We have identified three semantic queries (Chapter 10), each with some variants, and we have defined them in terms of DL. Especially relevant is the third type of query that makes it possible to integrate data from different datasets to create a new one. Our solution is mainly based on an open-world assumption typical of Description Logic, where the knowledge (and its representation through the ontology) is not supposed to be complete. It also presents significant advantages with respect to other existing formal solutions for geographic datasets integration: on the one hand, it removes the restriction of classes being artificially organized in a lattice; on the other hand, the use of DL permits richer definitions of concepts. In particular, it supports an integration involving modelled themes and their DL definitions. This provides a model checking capability, which enables users to find where a modelled theme is satisfied.

Finally, Chapter 11 presents an example of how our ontology can be used in a different context in the framework of the semantic web, namely in a process of indexing and retrieving geo-referenced multimedia elements (still images and video sequences), according to their thematic geographic content. The indexing process uses the three types of semantic services defined in Chapter 10. One of the main contributions of this chapter is the definition of a semantic model for describing still image and video in terms of thematic information. The representation of this model by means of MPEG-7 is relevant too. Furthermore, another significant contribution is the algorithm for segmenting and indexing geo-referenced video according to its thematic content.

12.2 Future research directions

Although the objectives of this thesis have been achieved, this work provides a base to new directions of future research. We briefly discuss here our research plans.

The first research line we would like to explore is related to the representation of quantitative themes. We can observe that our approach mainly focuses on qualitative themes, while the main role of quantitative themes is its participation in the definition of models. Reasoning with quantitative themes is very limited, mainly because Description Logic only provides "qualitative" reasoning. As an example, given a set of disjoint intervals, a DL reasoner cannot obtain those intervals greater than a particular threshold. DL can only reason with concepts but not with numbers and arithmetical operations or relations like "<" or ">". A combined use of DL with other logic formalisms would enable us to reason on numeric intervals. Furthermore, it would enable us to deal with pure quantitative datasets (with raw unclassified data) that could be dynamically classified by the system according to a particular classification. It would also support translations from datasets expressed in one unit of measure (for instance Celsius degrees) to another one (for instance Fahrenheit degrees). We would also like to explore the possibility of representing relations between classes from different classifications through fuzzy logic.

This enhanced reasoning system would enable us to deal with more complex models. Geographers often use decision models that consist of arithmetical expressions combining different (often quantitative) variables that are given different weights. As a very simplified and fictitious example, one could define "flooding risks" through the sum of 1 over the distance in meters to a torrent, multiplied by 5, and the average rains in autumn measured in mm, multiplied by 3. The result of this definition is a quantitative range of values, that can be later classified in intervals. We can observe that a non-modelled quantitative theme is unambiguously defined by its unit of measure. For instance a temperature value expressed in Celsius degrees is clearly unambiguous. However, this may not be true in modelled quantitative themes as in the example of flooding risks. Note also that different decision models may generate different units of measure for the same “flooding risks” theme. A way to represent the meaning of these units of measure in these models is needed.

A second research line that we would like to follow refers to how the spatial and temporal components of GI can be better represented in the semantic framework. This involves the introduction of gazetteers and calendars in the framework. This would enable us to define not only thematic queries, but also queries of type “concepts in a space at a certain time”. In the case of space, an important line of research refers to the representation and reasoning on the spatial relations between concepts. In the example of the risks of forest fires in 10.3.5 we introduced a variable “distance to roads” in the definition of the model. This assumes that a dataset providing these distances will be inserted in the repository. But defining more complex models requiring that the reasoning system itself determines whether a certain spatial relation is satisfied is much more complex. Some initial work on extending a DL language to support some spatial reasoning has been carried out in (Wessel 2002; 2003). But this approach is based on manually representing the topological relations between spatial units, which is something not feasible for real datasets.

Space also has an influence on the thematic component. Boundaries between different thematic categories are usually fuzzy. For instance, where a pine forests finishes and garrigue starts cannot be easily depicted with a line. The translation to this reality with fuzzy boundaries to crisp datasets often causes discrepancies between different datasets. We would like to analyze how this fuzziness can be taken into consideration in the dataset integration (third type of semantic service) and in the mapping algorithm based on the spatial distribution of dataset values.

Some models are also based on complex algorithms that combine thematic, spatial and temporal elements. We would also like to study how the semantic framework could be extended to deal with task ontologies referring to these algorithms.

The third main line of research derived from our thesis has a more technical focus: we would like to explore how our semantic framework can be used in the context of existing Spatial Data Infrastructures (SDIs) in order to enhance them with semantic services. Since several of these SDIs support different languages, a particular aspect that we would also like to study in the future is how multilinguality would affect our work. Another related research line also with an important technical focus, although with a significant semantic component, relates to how different repositories represented through our semantic framework could interoperate in order to build a network of repositories.

Finally, some research lines are directly related to VideoGIS and what we have called *video-itineraries*. These lines have already been discussed in 11.5. The first research line in this context relates to an “intelligent” composition of video segments aiming at providing a narrative thread to the final sequence. A second line is related to navigation. Links have to be automatically generated according to two main axes: thematic information and segment spatial locations. The result is a dynamically built hypervideo network. A final line in the context of VideoGIS arises from considering spatial extents for the themes in video segments, as well as the links between them and moving regions in the video.

References

Adali, S., K. S. Candan, S.-S. Chen, K. Erol and V. S. Subrahmanian (1996). Advanced Video Information System: Data Structures and Query Processing. *ACM-Springer Multimedia Systems Journal*.

Agarwal, P. (2005). Ontological Considerations in GIScience. *International Journal of Geographic Information Science* **19**(5): 501-536.

Aguierre Smith, T. S. (1992). *If you could see what I mean*. MS Thesis. Massachusetts Institute of Technology. Cambridge, Massachusetts, USA.

Anderson, J. R., E. E. Hardy, J. T. Roach and R. E. Witmer (1976). *A Land Use And Land Cover Classification System For Use With Remote Sensor Data*. Geological Survey Professional Paper 964, Washington, DC, USA.

ANSI (1998). *Spatial Data Transfer Standard*. ANSI Standard. ANSI NCITS 320-1998. American National Standards Institute (ANSI).

Avesani, P., F. Giunchiglia and M. Yatskevich (2005). A Large Scale Taxonomy Mapping Evaluation. *International Semantic Web Conference*.

Baader, F., D. L. McGuinness, D. Nardi and P. F. Patel-Schneider, Eds. (2002). *The Description Logic Handbook: Theory, implementation, and applications*, Cambridge University Press.

Berners-Lee, T., J. Hendler and O. Lassila (2001). The Semantic Web. *Scientific American*(May 2001).

Bibiloni, A. (1999). *Un sistema de edición video por contenido aplicado a la creación de entornos hipervídeo dinámicos*. PhD Thesis. Departament de Matemàtiques i Informàtica, Universitat de les Illes Balears. Palma de Mallorca, Spain.

Bilenko, M. and R. Mooney (2003). Adaptive duplicate detection using learnable string similarity measures. *SIGKDD*.

Bilenko, M., R. Mooney, W. W. Cohen, P. Ravikumar and S. E. Fienberg (2003). Adaptive Name Matching in Information Integration. *IEEE Intelligent Systems* **18**(5): 2-9.

Bilenko, M. and R. Mooney (2005). Alignments and String Similarity in Information Integration: A Random Field Approach.

Bishr, Y. (1998). Overcoming the semantic and other barriers to GIS interoperability. *International Journal of Geographic Information Science* **12**(4): 299-314.

Blat, J., A. Delgado, M. Ruiz and J. M. Seguí (1995). Designing multimedia GIS for territorial planning: the ParcBIT case. *Environment and Planning B: Planning and Design* **22**: 665-678.

Borgida, A., R. J. Brachman, D. L. McGuinness and L. A. Resnick (1989). CLASSIC: A structural data model for objects. *ACM SIGMOD International Conference on Management of Data*.

Bossard, M., J. Feranec and J. Otahel (2000). *CORINE land cover technical guide – Addendum 2000*. European Environment Agency (EEA), Copenhagen, Denmark.

Brachman, R. J. and J. G. Schmolze (1985). An overview of the KL-ONE knowledge representation system. *Cognitive Science* **9**(2): 171-216.

Burrough, P. A. and A. U. Frank, Eds. (1996). *Geographic Objects with Indeterminate Boundaries*. GISDATA II. London, UK, Taylor and Francis.

Câmara, G., M. J. Egenhofer, F. Fonseca and A. M. V. Monteiro (2001). What's In An Image? *Spatial information theory: Foundations of Geographic Information Science. COSIT 2001*, Springer.

Castano, S., V. D. Antonellis and S. D. C. d. Vimercati (2000). Global viewing of heterogeneous data sources. *IEEE Transactions on Knowledge and Data Engineering* **13**(2): 277–297.

CEN (1998). *Geographic Information European Prestandards, Euro-norme volutaire for Geographic Information - Data description - Metadata*. ENV 12657. European Committee for Standardization (CEN) - TC 287.

Chang, S., W. Chen, H. Meng, H. Sundaram and D. Zhong (1997). VideoQ: an automated content based video search system using visual cues. *ACM International Conference on Multimedia*.

Chaudhuri, K., A. Farquhar, R. E. Fikes, P. D. Karp and J. Rice (1998). *Open Knowledge Base Connectivity*. Technical Report.

Christel, M., T. D. Ng, H. Wactlar and A. Hauptmann (2002). Collages as Dynamic Summaries for News Video. *ACM Multimedia*, Juan-les-Pins, France.

Christel, M. and R. Conescu (2005). Addressing the Challenge of Visual Information Access from Digital Image and Video Libraries. *Joint Conference on Digital Libraries*, Denver, Colorado, USA.

Christel, M. (2006). Evaluation and User Studies with Respect to Video Summarization and Browsing. *Multimedia Content Analysis, Management and Retrieval 2006, part of the IS&T/SPIE Symposium on Electronic Imaging 2006*, San Jose, California, USA.

Christel, M. G. and A. M. Olligschlaeger (1999). Interactive Maps for Digital Video Library. *IEEE International Conference on Multimedia Computing and Systems*.

Cohen, W. W., P. Ravikumar and S. E. Fienberg (2003a). A Comparison of String Metrics for Matching Names and Records. *American Association for Artificial Intelligence*.

Cohen, W. W., P. Ravikumar and S. E. Fienberg (2003b). A Comparison of String Distance Metrics for Name-Matching Tasks. *IJCAI Workshop on Information Integration on the Web*.

Comber, A., P. Fisher and R. Wadsworth (2003). A semantic statistical approach for identifying change from ontologically divers land cover data. *6th AGILE Conference*, Lyon, France.

Comber, A., P. Fisher and R. Wadsworth (2004). Integrating land-cover dataa with different ontologies: identifying change from inconsistency. *International Journal of Geographic Information Science* **18**(7): 691-708.

Connolly, D., F. v. Harmelen, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider and L. A. Stein (2001). *DAML+OIL Reference Description*. W3C Note. http://www.w3.org/TR/daml+oil-reference. W3C.

Cruz, I. F. and A. Rajendran (2003). Semantic Data Integration in Hierarchical Domains. *IEEE Intelligent Systems* **March-April**: 66-73.

Dagan, I., L. Lee and F. Pereira (1999). Similarity-based models of word cooccurrence probabilities. *Machine Learning 34* **34**(1-3).

Davenport, G., T. S. Aguierre Smith and N. Pincever (1991). Cinematic primitives for multimedia. *IEEE Computer Graphics & Applications*.

DCMI (2005). *DCMI Metadata Terms*. Dublin Core Metadata Initiative. http://dublincore.org/.

de Hoog, R. (1998). Methodologies for Building Knowledge Based Systems: Achievements and Prospects. *Handbook of Expert Systems*. J. Liebowitz. Boca Raton, Florida, USA, CRC Press.

Di Gregorio, A. and L. J. M. Jansen (1998). *Land Cover Classification System (LCCS): Classification Concepts and User Manual*. Food and Agriculture Organization of the United Nations (FAO), Rome, Italy.

Di Gregorio, A. (2005). *Land Cover Classification System (LCCS), version 2: Classification Concepts and User Manual*. Food and Agriculture Organization of the United Nations (FAO), Rome, Italy.

Do, H.-H., S. Melnik and E. Rahm (2002). Comparison of schema matching evaluations. *GI-Workshop Web and Databases*, Erfurt, Germany.

Do, H.-H. and E. Rahm (2002). Coma – a system for flexible combination
of schema matching approaches. *VLDB*.

Doan, A., P. Domingos and A. Halevy (2000). Learning source descriptions for data integration. *WebDB Workshop*.

Doan, A., P. Domingos and A. Halevy (2001). Reconciling schemas of disparate data sources: a machine-learning approach. *ACM SIGMOD Conference*.

Doan, A. (2002). *Learning to Map between Structured Representations of Data*. PhD thesis. University of Washington. Seattle, Washington, USA.

Doan, A., J. Madhavan, P. Domingos and A. Halevy (2004). Ontollogy matching: a machine learning approach. *Handbook of ontologies, International handbooks on information systems*. S. Staab and R. Studer. Berlin, Germany, Springer Verlag**:** 385–404.

Duckham, M. and M. F. Worboys (2005). An algebraic approach to automated information fusion. *International Journal of Geographic Information Science* **19**(5): 537-557.

Durbin, R., S. Eddy, A. Krogh and G. Mitchison (1998). *Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.*, Cambridge University Press.

EEA (2001). *GEMET version 2001 (GEneral Multilingual Environmental Thesaurus)*. http://www.mu.niedersachsen.de/system/cds. European Environment Agency, European Topic Centre on Catalogue of Data Sources (ETC/CDS).

Egenhofer, M. J. and D. M. Mark (1995). Naive Geography. *Spatial Information Theory-A Theoretical Basis for GIS, International Conference COSIT'95*, Semmering, Austria, Springer-Verlag.

Egenhofer, M. J. (2002). Toward the semantic geospatial web. *10th ACM international symposium on Advances in geographic information systems*, McLean, Virginia, USA, ACM Press.

Ehrig, M. and J. Euzenat (2005). Relaxed Precision and Recall for Ontology Matching. *K-Cap Workshop on Integrating ontology*, Banff, Canada.

Enser, P. G. B. and C. J. Sandom (2002). Retrieval of Archival Moving Imagery - CBIR Outside the Frame? *Conference on Image and Video Retrieval*.

Euzenat, J. and P. Valtchev (2003). An integrative proximity measure for ontology alignment. *1st International Workshop on Semantic Integration*, Sanibel Island, Florida, USA.

Euzenat, J., D. Loup, M. Touzani and P. Valtchev (2004). Ontology Alignment with OLA. *3rd EON Workshop, 3rd International Semantic Web Conference*, Hiroshima, Japan.

Euzenat, J., H. Stuckenschmidt and M. Yatskevich (2005). Introduction to the Ontology Alignment Evaluation 2005. *K-Cap Workshop on Integrating ontology*, Banff, Canada.

Farquhar, A., R. E. Fikes and J. Rice (1997). The Ontolingua Server: A Tool for Collaborative Ontology Construction. *International Journal of Human Computer Studies* **46**(6): 707-727.

Fellbaum, C., Ed. (1998). *WordNet. An Electronic Lexical Database*, MIT Press.

Fensel, D., F. van Harmelen and I. Horrocks (2002). OIL and DAML+OIL: Ontology Languages for the Semantic Web. *Towards the Semantic Web*. J. Davies, D. Fensel and F. van Harmelen. Chichester, UK, Wiley.

Fensel, D., J. Hendler, H. Lieberman and W. Wahlster, Eds. (2003). *Spinning the Semantic Web*. Cambridge, Massachusetts, USA, MIT Press.

FGDC (1998). *Content Standard for Digital Geospatial Metadata, v2.0*. FGDC standard. FGDC-STD-001-1998. Federal Geographic Data Committee, Metadata Ad Hoc Working Group.

Flickner, M., H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker (1995). Query by image content: the QBIC system. *IEEE Computer*: 23-31.

Fonseca, F. T., M. J. Egenhofer and C. Davis (2000). Ontology-Driven Information Integration. *AAAI Workshop on Spatial and Temporal Granularity*, Austin, Texas, USA.

Fonseca, F. T. (2001). *Ontology-Driven Geographic Information Systems*. PhD thesis. University of Maine

Fonseca, F. T., M. J. Egenhofer, P. Agouris and G. Câmara (2002). Using Ontologies for Integrated Geographic Information Systems. *Transactions in GIS* **6**(3): 2002.

Francis, W. N. and H. Kucera (1982). *Frequency Analysis of English Usage: Lexicon and Grammar*. Boston, Massachusetts, USA, Houghton Mifflin.

Franconi, E. (2002). Course on Description Logics.

Frank, A. U. (1997). Spatial Ontologies. A Geographical Point of View. *Spatial and Temporal Reasoning*. O. Stock. Dordrecht, The Netherlands, Kluwer Academic Publishers.

Frank, A. U. (2001). Tiers of ontology and consistency constraints in geographical information systems. *International Journal of Geographic Information Science* **15**(7): 667-678.

Frank, A. U. (2003). Ontology for Spatio-Temporal Databases. *Spatiotemporal Databases: The Chorochronos Approach*. M. e. a. Koubarakis, Springer-Verlag.

Fritz, S. and L. See (2005). Comparison of land cover maps using fuzzy agreement. *International Journal of Geographic Information Science* **19**(7): 787-807.

Ganter, B. and R. Wille (1999). *Formal Concept Analysis*. Berlin-Heidelberg, Germany, Springer.

Gärdenfors, P. (2000). *Conceptual Spaces: the Geometry of Thought*. Cambridge, Massachusetts, USA, MIT Press.

Genesereth, M. R. and N. J. Nilsson (1987). *Logical Foundation of Artificial Intelligence*. Los Altos, California, USA, Morgan Kaufmann.

Genesereth, M. R. and R. E. Fikes (1992). *Knowledge Interchange Format, version 3.0 reference manual*. Technical Report Logic-92-1. Stanford University.

Goldstone, R. L. (1994). Similarity, Interactive Activation, and Mapping. *Journal of Experimental Psychology: Learning, Memory, and Cognition* **20**(1): 3-28.

Goldstone, R. L., D. L. Medin and J. Halberstadt (1997). Similarity in Context. *Memory and Cognition* **25**(2): 237-255.

Gómez-Pérez, A., M. Fernández-López and O. Corcho (2004). *Ontological Engineering*. London, UK, Springer-Verlag.

Goodchild, M., M. J. Egenhofer, R. Fegeas and C. Kottman, Eds. (1999). *Interoperating Geographic Information Systems*. The International Series in Engineering and Computer Science, Kluwer.

Gotoh, O. (1981). An Improved Algorithm for Matching Biological Sequences. *Journal of Molecular Biology* **162**: 705-708.

Gouveia, C., P. Henriques, R. Nicolau, J. Rocha and M. Santos (2001). Moving from CEN TC 287 to ISO/TC 211 - The approach of the Portuguese National Geographic Information Infrastructure. *4th AGILE Conference on Geographic Information Science*, Brno, Czech Republic.

Gruber, T. R. (1993). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *Formal Ontology in conceptual Analysis and Knowledge Representation*. N. Guarino and R. Poli, Kluwer Academic Publishers.

Guarino, N. and P. Giaretta (1995). Ontologies and Knowledge Bases: Towards a Terminological Clarification. *Towards Very Large Knowledge Bases*. N. Mars. Amsterdam, IOS Press.

Guarino, N. (1998). Formal Ontology and Information Systems. *Formal Ontology in Information Systems*, Trento, Italy, IOS Press.

Guarino, N., C. Masolo and G. Verete (1999). OntoSeek: Content-Based Access to the Web. *IEEE Intelligent Systems.* **14**(3): 70-80.

Haarslev, V. and R. Möller (2001). RACER system description. *Int. Joint Conf. on Automated Reasoning*, Springer.

Hakimpour, F. and A. Geppert (2001). Resolving Semantic Heterogeneity in Schema Integration: an Ontology Based Approach. *Formal Ontology in Information Systems*, Ogunquit, Maine, USA, ACM.

Hakimpour, F. and S. Timpf (2001). Using Ontologies for resolution of Semantic Heterogeneity in GIS. *4th AGILE Conference on Geographic Information Science*, Brno, Czech Republic.

Hakimpour, F. and S. Timpf (2002). A Step towards Geodata Integration using Formal Ontologies. *5th AGILE Conference on Geographic Information Science*, Palma de Mallorca, Spain.

Haupmann, A. G. and M. J. Witbrock (1997). Informedia: News-on-demand multimedia information acquisition and retrieval. *Intelligent Multimedia Information Retrieval*. M. T. Bradbury. Cambridge, Massachusetts, USA, MIT Press**:** 213-239.

Hauptmann, A. (2005). Lessons for the Future from a Decade of Informedia Video Analysis Research. *International Conference on Image and Video Retrieval*, Singapore, Springer-Verlag.

Heit, E. (1997). Features of Similarity and Category-Based Induction. *Interdisciplinary Workshop on Categorization and Similarity*, Edinburgh, UK.

Herold, M. and C. Schmullius (2004). *Report on the Harmonization of Global and Regional Land Cover Products Meeting*. Global Observation of Forest and Land Cover Dynamics (GOFC-GOLD), Rome, Italy.

Hjelsvold, R. and R. Midtstraum (1994). Modelling and querying data. *20th International Conference on Very Large Data Bases*, Santiago, Chile.

Horrocks, I. (1998). Using an expressive description logic: FaCT or fiction? *6th International Conferences on Principles of Knowledge Representation and Reasoning*.

Horrocks, I., U. Sattler and S. Tobies (2000). Practical reasoning for very expressive description logics. *Journal of the Interest Group in Pure and Applied Logic* **8**(3): 239–264.

Hotho, A., A. Mädche, S. Staab and V. Zacharias (2003). On Knowledgeable Unsupervised Text Mining. *Text Mining. Theoretical Aspects and Applications*. G. N. J. Franke, I. Renz, Physica-Verlag/Springer**:** 131-152.

Hwang, T.-H., K.-H. Choi, I.-H. Joo and J.-H. Lee (2003). MPEG-7 metadata for video-based GIS applications. *IEEE International Geoscience and Remote Sensing Symposium*, IEEE.

ISO (1985). *Guidelines for the establishment and development of multilingual thesauri*. ISO 5694:1985. International Organization for Standardization.

ISO (1986). *Guidelines for the establishment and development of monolingual thesauri*. ISO 2788:1986. International Organization for Standardization.

ISO (2000). *Multimedia Content Description Interface*. ISO International Standard. ISO 15938. International Organization for Standardization JTC1/SC29/WG11.

ISO (2003a). *Geographic Information - Metadata*. ISO standard. ISO/TC211 19115. International Organization for Standardization, TC 211.

ISO (2003b). *Dublin Core Metadata Element Set, v.1.1*. ISO Standard. ISO 15836. International Organization for Standardization.

ISO (2005). *Geographic Information - Rules for application schemas*. ISO International Standard. ISO/TC211 19109. International Organization for Standardization, TC 211.

Jaro, M. A. (1989). Advances in Record-Linkage Methodology as Applied to Matching the 1985

Census of Tampa, Florida. *Journal of the American Statistical Association* **89**: 414-420.

Jaro, M. A. (1995). Probabilistic linkage of large public health data files. *Statistics in Medicine* **14**: 491–498.

Kalfoglou, Y. and M. Schorlemmer (2003). If-map: an ontology mapping method based on information flow theory. *Journal of data semantics* **1**: 98–127.

Kavouras, M. and M. Kokla (2002). A method for the formalization and integration of geographical categorizations. *International Journal of Geographic Information Science* **16**(5): 439-453.

Kemp, K. K. and A. Vckovsky (1998). Towards an ontology of fields. *GeoComputation*, Bristol UK.

Kifer, M., G. Lausen and J. Wu (1995). Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM* **42**(4): 741-843.

Kim, S.-S., K.-H. Kim and K.-O. Kim (2003a). Web-Based Media GIS Architecture Using the Virtual World Mapping Technique. *Korean Journal of Remote Sensing* **19**(1): 71-80.

Kim, S.-S., S.-H. Lee, K.-H. Kim and J.-H. Lee (2003b). A unified visualization framework for spatial and temporal analysis in 4D GIS. *IEEE International Geoscience and Remote Sensing Symposium*, IEEE.

Kim, S.-S. and J.-H. Park (2004). Geographic hypermedia using search space transformation. *International Conference on Pattern Recognition*.

Kim, Y. B. and M. Shibata (1996). Content-Based Video Indexing and Retrieval - A Natural Language Approach. *IEICE Transactions on Information and Systems* **E79-D**(6): 695-705.

KnowledgeWeb (2005). *Specification of a common framework for characterizing alignment*. Project Deliverable. KWEB/2004/D2.2.1/v2.0. KnowledgeWeb Consortium (IST Project IST-2004-507482).

KnowledgeWeb Consortium (2004). *State of the art on ontology alignment*.

KnowledgeWeb Consortium (2005). *Specification of a common framework for characterizing alignment*.

Krumhansl, C. (1978). Concerning the Applicability of Geometric Models to Similarity Data: The Interrelationship Between Similarity and Spatial Density. *Psychological Review* **85**(5): 445-463.

Kuhn, W. (2003a). Semantic reference systems. *International Journal of Geographic Information Science* **17**(5): 405-409.

Kuhn, W. (2003b). Semantic reference systems. Seminar at Universitat Pompeu Fabra, Barcelona, Spain.

Kuhn, W. (2003c). Implementing Semantic Reference Systems. *6th AGILE Conference*, Lyon, France.

Lacasta, J., P. R. Muro-Medrano, J. Nogueras-Iso and F. J. Zarazaga-Soria (2005). Web Ontology Service, a key component of a Spatial Data Infrastructure. *11th EC GI & GIS Workshop: ESDI Setting the Framework*, Sardinia, Italy.

Lee, H., A. F. Smeaton, C. O'toole, N. Murphy, S. Marlow and N. E. O'connor (2000). The Físchlár Digital Video Recording, Analysis and Browsing System, Paris, France.

Lee, J. H., M. H. Kim and Y. J. Lee (1993). Information Retrieval Based on Conceptual Distance in IS-A Hierarchies. *Journal of Documentation* **49**(2): 188-207.

Lee, S., S. Kim, J. Choi and J. Lee (2003). 4SVan: A Prototype Mobile Mapping System for GIS. *Korean Journal of Remote Sensing* **19**(1).

Lenat, D. B. and R. V. Guha (1990). *Building Large Knowledge-based Systems: Representation and Inference in the Cyc Project*. Boston, Massachusetts, USA, Addison-Wesley.

Levenstein, I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Cybernetics and Control Theory*.

Li, W. and C. Clifton (2000). SemInt: a tool for identifying attribute correspondences in heterogeneous databases using neural network. *Data Knowledge Engineering* **33**(1).

Lin, D. (1998). An information-theoretic definition of similarity. *15th International Conference on Machine Learning*, Madison, Wisconsin, USA.

Lippman, A. (1980). Movie maps: An application of the optical videodisc to computer graphics. *Computer Graphics (SIGGRAPH)*.

Luke, S. and J. D. Helfin (2000). *SHOE 1.01 Proposed Specification*. Technical Report. University of Maryland. Department of Computer Science.

Lutz, C. and U. Sattler (2002). ESSLLI Course on Description Logics.

Lutz, M. (2005). *Ontology-based Discovery and Composition of Geographic Information Services*. PhD Thesis. Universität Münster, Dublin, Ireland.

Lutz, M. and E. Klien (2006). Ontology-based retrieval of geographic information. *International Journal of Geographic Information Science* **20**(3): 233-260.

MacGregor, R. and R. Bates (1987). *The Loom knowledge representation language*. Technical Report ISI/RS-87-188. University of Southern California, Information Science Institute, Marina del Rey, CA, USA.

Madhavan, J., P. A. Bernstein and E. Rahm (2001). Generic Schema Matching with Cupid. *27th VLDB Conference*, Rome, Italy.

Mark, D. M., A. Skupin and B. Smith (2001). Features, Objects, and other Things: Ontological Distinctions in the Geographic Domain. *Spatial information theory: Foundations of Geographic Information Science. COSIT 2001*, Springer.

McGuinness, D. L., R. E. Fikes, J. Rice and S. Wilder (2000). An Environment for Merging and Testing Large Ontologies. *7th International Conference on Principles of Knowledge Representation and Reasoning*, Breckenridge, USA.

McGuinness, D. L., R. E. Fikes, L. A. Stein and J. Hendler (2003). DAML-ONT: An Ontology Language for the Semantic Web. *Spinning the Semantic Web*. D. Fensel, J. Hendler, H. Lieberman and W. Wahlster. Cambridge, Massachusetts, USA, MIT Press.

Medin, D. L., R. L. Goldstone and D. Genter (1993). Similarity in Context. *Similarity in Context* **100**(2): Similarity in Context.

Melnik, S., H. Garcia-Molina and E. Rahm (2001). *Similarity Flooding: A Versatile Graph Matching Algorithm*. Extended Technical Report. Stanford InfoLab Publication Server.

Miller, G. A. (1990). WordNet: An on-line lexical database. *International Journal of Lexicography* **4**(3).

Monge, A. and C. Elkan (1996). The field matching problem: algorithms and application. *2nd International Conference on Knowledge Discovery and Data Mining*.

Motta, E. (1999). *Reusable Component for Knowledge Modelling: Principles and Case Studies in Parametric Design*. Amsterdam, The Netherlands, IOS Press.

Navarrete, T. (2001). *VideoGIS: Combining Video and Geographical Information*. DEA (Master) thesis. Departament de Tecnologia, Universitat Pompeu Fabra. Barcelona.

Navarrete, T. and J. Blat (2002a). VideoGIS: segmentación y modelado de video basado en información geográfica. *3er Congreso Interacción Persona-Ordenador*, Leganés, Madrid, Spain.

Navarrete, T. and J. Blat (2002b). VideoGIS: Segmenting and indexing video based on geographic information. *5th AGILE Conference on Geographic Information Science*, Palma de Mallorca, Spain.

Navarrete, T. and J. Blat (2003). Indización automática de vídeo. *El profesional de la Información* **12**(6): 430-442.

Navarrete, T. and J. Vega (2003). *Speed-FX Metadata Report*. Deliverable. Deliverable 4.1.1. Speed-FX Project (IST-2001-34337).

Navarrete, T., J. Blat and M. Ruiz (2004). Semantic interoperability of field-based thematic geographic information. *15th International Workshop on Database and Expert Systems and Applications*, Zaragoza, Spain, IEEE Press.

Nebert, D. (2001). *Developing Spatial Data Infrastructures: The SDI Cookbook v.1.1.* Global Spatial Data Infrastructure.

Needleman, S. B. and C. D. Wunsch (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Molecular Biology* **48**: 444–453.

Negroponte, N. (1995). *Being digital*. p. 65-67. London, UK, Hoder and Stoughton.

Ng, T. D., H. Wactlar, A. Hauptmann and M. Christel (2003). Collages as Dynamic Summaries of Mined Video Content for Intelligent Multimedia Knowledge Management. *AAAI Spring Symposium Series on Intelligent Multimedia Knowledge Management*, Palo Alto, California, USA.

Nobre, E. M. N. and A. S. Câmara (2001). Spatial Video: exploring space using multiple digital videos. *6th Eurographics Workshop on Multimedia*, Manchestere, UK, Springer-Verlag.

Nogueras-Iso, J., J. Lacasta, J. Á. Bañares, P. R. Muro-Medrano and F. J. Zarazaga-Soria (2004). Exploiting disambiguated thesauri for information retrieval in metadata catalogs. *X Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2003)*, San Sebastián, Spain, Springer-Verlag.

Nogueras-Iso, J., F. J. Zarazaga-Soria and P. R. Muro-Medrano (2005). *Geographic Information Metadata for Spatial Data Infrastructures*. Berlin-Heidelberg, Germany, Springer.

Noy, N. F. and M. A. Musen (1999). SMART: Automated Support for Ontology Merging and Alignment. *12th Banff Workshop on Knowledge Acquisition, Modeling and Management*, Banff, Alberta, Canada.

Noy, N. F. and M. A. Musen (2000). PROMPT: Algorithm and Tool for Automated Ontology Merging. *17th National Conference on Artificial Intelligence (AAAI-2000)*, Austin, Texas, USA.

Noy, N. F. and M. A. Musen (2001). Anchor-PROMPT: Using non-local context for semantic matching. *17th International Joint Conference on Artificial Intelligence, Workshop on Ontologies and Information Sharing*, Seattle, WA, USA.

OGC (2004). *OpenGIS Geography Markup Language (GML) Encoding Specification, v. 3.1.1*. OGC 03-105r1. OpenGIS Consortium. Editors: S. Cox, P. Daisey, R. Lake, C. Portele, A. Whiteside.

OGC (2005). *Catalogue Services Specification, 2.0.0*. OGC 04-021r3. Open Geospatial Consortium. Editors: D. Nebert, A. Whiteside.

OntoWeb (2002). *A survey on ontology tools*. Project Deliverable. Deliverable 1.3. OntoWeb Consortium (IST-2000-29243).

Openshaw, S., C. Wymer and M. Charlton (1986). A geographical information and mapping system for the BBC Domesday optical disks. *Transactions of the Institute of British Geographers* **11**: 296-304.

Over, P., T. Ianeva, W. Kraaij and A. F. Smeaton (2005). TRECVID 2005 An Overview. *TREC Video Retrieval Evaluation (TRECVID)*.

Palopoli, L., D. Sacca and D. Ursino (1998). An automatic technique for detecting type conflicts in database schemas. *7th International Conference on Information and Knowledge Management*.

Peuquet, D., B. Smith and B. Brogaard (1998). *The Ontology of Fields*. Report of a Specialist Meeting Held under the Auspices of the Varenius Project.

Phan Luong, V., T. T. Pham and R. Jeansoulin (2003). Integrating information under Lattice structure. *BDA*.

Phan Luong, V., T. T. Pham and R. Jeansoulin (2004). Data Quality Based Fusion: Application to Land Cover. *7th International Conference on Information Fusion*, Stockholm, Sweden.

Rada, R., H. Mili, E. Bicknell and M. Blettner (1989). Development and Application of a Metric on Semantic Nets. *IEEE Transactions on System, Man, and Cybernetics* **19**(1): 17-30.

Rahm, E. and P. A. Bernstein (2001). A survey of approaches to automatic schema matching. *VLDB Journal* **10**(4): 334-350.

Raper, J. (2001). *Multidimensional Geographic Information Science*. London, UK, Taylor & Francis.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *14th International Joint Conference on Artificial Intelligence*.

Resnik, P. (1999). Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Articial Intelligence Research* **11**: 95-130.

Richardson, R., A. F. Smeaton and J. Murphy (1994). *Using WordNet as a knowledge base for measuring semantic similarity between words.* Working paper CA-1294. Dublin City University, School of Computer Applications, Dublin, Ireland.

Ristad, E. S. and P. N. Yianilos (1997). Learning string edit distance. *ICML*.

Rodríguez, M. A. (2000). *Assessing Semantic Similarity Among Spatial Entity Classes*. Spatial Information Science and Engineering, University of Maine

Rodríguez, M. A. and M. J. Egenhofer (2003). Determining Semantic Similarity Among Entity Classes from Different Ontologies. *IEEE Transactions on Knowledge and Data Engineering* **15**(2): 442-456.

Rodríguez, M. A. and M. J. Egenhofer (2004). Comparing Geospatial Entity Classes: An Asymmetric and Context-Dependent Similarity Measure. *International Journal of Geographical Information Science* **18**(3): 229-256.

Salton, g. and M. J. McGill (1983). *An Introduction to Modern Information Retrieval*. New York, USA, McGraw-Hill.

Schuster, G. and H. Stuckenschmidt (2001). Building shared terminologies for ontology integration. *Künstliche Intelligenz (KI)*, Vienna, Austria.

Schwering, A. and M. Raubal (2005). Measuring Semantic Similarity between Geospatial Conceptual Regions. *1st International Conference on GeoSpatial Semantics*, Mexico City, Mexico, Springer.

SEKT (2003). *State-of-the-art survey on Ontology Merging and Aligning V1*. Project Deliverable. D4.2.1. Semantically Enabled Knowledge Technologies, SEKT, Consortium (IST-2003-506826).

Shiffer, M. J. (1992). Towards a Collaborative Planning System. *Environment and Planning B: Planning and Design* **19**: 709-722.

Silva, N. and J. Rocha (2003). An ontology mapping framework for the semantic web. *6th International Conference on Business Information Systems*, Colorado Springs, USA.

Silva, N., J. Santos and J. Rocha (2004). Proposal for the combination of ontology assemble and ontology mapping processes. *International Conference on Knowledge Engineering and Decision Support*, Porto, Portugal.

Smeaton, A. F. and H. Lee (2002). Designing the User Interface for the Físchlár Digital Video Library. *Journal of Digital Information* **2**(4).

Smith, B. and D. M. Mark (1998). Ontology and Geographic Kinds. *International Symposium on Spatial Data Handling*, Vancouver, Canada.

Smith, B. and D. M. Mark (1999). Ontology with human subjects testing: An Empirical Investigation of Geographic Categories. *American Journal of Economics and Sociology* **58**: 245-272.

Smith, B. and D. M. Mark (2001). Geographical categories: an ontological investigation. *International Journal of Geographic Information Science* **15**(7): 591-612.

Smith, J. and S. Chang (1997). Visually Searching the Web for Content. *IEEE Multimedia* **4**(3): 12-20.

Smith, T. F. and M. S. Waterman (1981). Identification of common molecular subsequences. *Journal of Molecular Biology* **147**: 195–197.

Sparck Jones, K. and C. v. Rijsbergen (1975). *Report on the need for and provision of an "ideal" information retrieval test collection*. British Library Research and Development Report 5266. Computer Laboratory, University of Cambridge.

Stoilos, G., G. Stamou and S. Kollias (2005). A String Metric for Ontology Alignment. *4th Internation Semantic Web Conference*, Springer-Verlag.

Stumme, G. and A. Maedche (2001). FCA-MERGE: Bottom-Up Merging of Ontologies. *17th International Joint Conference on Artificial Intelligence*, Seattle, WA, USA, Morgan Kaufmann.

Su, X. and J. A. Gulla (2003). Semantic enrichment for ontology mapping. *9th International Conference on Applications of Natural Language to Information Systems*, Manchester, UK.

Su, X. (2004). *Semantic Enrichment for Ontology Mapping*. Department of Computer and Information Science, Norwegian University of Science and Technology. Trondheim, Norway.

Subramanian, V. S. (1997). *Principles of Multimedia Database Systems*. San Francisco, California, USA, Morgan Kaufman Publishers.

Sussna, M. (1993). Word sense disambiguation for free-text indexing using a massive semantic network. *2nd International Conference on Information and Knowledge Management*, Arlington, Virginia, USA.

Sutinen, E. and J. Tarhio (1995). On using q-gram locations in approximate string matching. *3rd Annual European Symposium on Algorithms (ESA95),* Springer-Verlag.

Tran, D. A., K. A. Hua and K. Vu (2000). Semantic Reasoning based Video Database Systems. *11th International Conference on Databases and Expert Systems Applications*, London, UK.

Tversky, A. (1977). Features of Similarity. *Psychological revies* **84**(4): 327-352.

Uitermark, H. (2001). *Ontology-based geographic data set integration*. PhD Thesis. Universiteit Twente. Deventer, The Netherlands.

Uitermark, H., P. van Oosterom, N. Mars and M. Molenaar (2002). Ontology-Based Geographic Data Set Integration. *5th AGILE Conference on Geographic Information Science,* Palma de Mallorca, Spain.

Valtchev, P. and J. Euzenat (1997). Dissimilarity measure for collections of objects and values. *2nd Symposium on Intelligent Data Analysis*.

Visser, U., H. Stuckenschmidt, H. Wache and T. Vögele (2001). Using Environmental Information Efficiently: Sharing Data and Knowledge from Heterogeneous Sources. *Environmental Information Systems in Industry and Public Administration*. C. Rautenstrauch and S. Patig. Hershey, USA & London, UK, IDEA Group**:** 41-73.

Visser, U. and G. Schuster (2002). Finding and Integration of Information - A Practical Solution for the Semantic Web -. *ECAI Workshop on Ontologies and Semantic Interoperability,* Lyon, France.

Visser, U., H. Stuckenschmidt, G. Schuster and T. Vögele (2002a). Ontologies for Geographic Information Processing. *Computers & Geosciences* **28**(1): 103-118.

Visser, U., T. Vögele and C. Schlieder (2002b). Spatio-Terminological Information Retrieval using the BUSTER System. *EnviroInfo*, Vienna, Austria.

Voorhees, E. M. (2004). Overview of TREC 2004. *13th NIST Text Retrieval Conference (TREC)*, Gaithersburg, MD, USA.

W3C (2004a). *RDF/XML Syntax Specification (Revised)*. W3C Recommendation. http://www.w3.org/TR/rdf-syntax-grammar/. W3C. Editor: D. Beckett.

W3C (2004b). *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C Recommendation. http://www.w3.org/TR/rdf-concepts/. W3C. Editors: G. Klyne, J.J. Carroll.

W3C (2004c). *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation. http://www.w3.org/TR/rdf-schema/. W3C. Editors: D. Brickley, R.V. Guha.

W3C (2004d). *OWL Web Ontology Language Semantics and Abstract Syntax*. W3C Recommendation. http://www.w3.org/TR/owl-semantics/. W3C. Editors: P.F. Patel-Schneider, P. Hayes, I. Horrocks.

W3C (2004e). *OWL Web Ontology Language Reference*. W3C Recommendation. http://www.w3.org/TR/owl-ref/. W3C. Editors: M. Dean, G. Schreiber. Authors: S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, L.A. Stein.

W3C (2005). *Representing Classes As Property Values on the Semantic Web*. Working Group Note. W3C Ontology Engineering and Patterns Task Force.

Wache, H., T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hübner (2001). Ontology-Based Integration of Information - A Survey of Existing Approaches. *International Joint Conference on Artificial Intelligence, Workshop on Ontologies and Information Sharing*, Seattle, WA, USA.

Wang, C. H., H. C. Lin, C. C. Shih, H. R. Tyan, C. F. Lin and H. Y. Mark Liao (2002). Querying Image Database by Video Content. *Advances in Multimedia Information Processing (PCM 2002). 3rd IEEE Pacific Rim Conference on Multimedia*.

Wessel, M. (2002). On Spatial Reasoning with Description Logics - Position Paper. *International Workshop in Description Logics*, Touluse, France.

Wessel, M. (2003). Some Practical Issues in Building a Hybrid Deductive Geographic Information System with a DL-Component. *10th International Workshop on Knowledge Representation meets Databases*, Hamburg, Germany.

Wilson, N. (2004). The beginnings of a logical semantics framework for the integration of thematic map data. *International Journal of Geographic Information Science* **18**(4): 389-415.

Winkler, W. E. (1999). *The state record linkage and current research problems*. Statistics of Income Division, Internal Revenue Service Publication.

Winter, S. (1998). Bridging Vector and Raster Representation in GIS. *6th ACM international symposium on Advances in geographic information systems*, Washington, DC, USA.

Worboys, M. F. (1995). *GIS: A Computing Perspective*. London, UK, Taylor & Francis.

Worboys, M. F. and M. Duckham (2002). Integrating spatio-thematic information. *Geographic Information Science, GIScience*, Springer.

Wu, Z. and M. Palmer (1994). Verb semantics and lexical selection. *32nd Annual Meeting of the Associations for Computational Linguistics*, Las Cruces, New Mexico, USA.

Wulder, M., J. White and S. Mcdonald (2005). Truth in the Air. Geolocated Video Validates Satellite Land Cover. *GeoWorld*.

Yang, M., B. Wildemuth and G. Marchionini (2004). The relative effectiveness of concept-based versus content-based video retrieval. *ACM Multimedia*.

Yoo, J., J. Choi, K. Sung and J. Kim (2005). Vehicular Image Based Geographic Information System for Telematics Environments. Integrating Map World into Real World. *IEEE International Geoscience and Remote Sensing Symposium*, IEEE.

Zhang, H. J., C. Y. Low and S. W. Smoliar (1995). Video parsing, retrieval and browsing: an integrated and content-based solution. *ACM Multimedia*: 15-24.

Appendix A OntoGIS: an application for managing geo-ontologies

Our semantic framework has been implemented in a prototype tool called OntoGIS. It enables the user to manage the ontology of thematic concepts. It also enables the user to add new datasets and vocabularies to the repository. In the case of dataset, the user can visualize their metadata, while in the case of vocabularies s/he can see their internal structure of narrower/broader/synonym terms. It implements the three merging methods: manual method for qualitative themes, semi-automatic method for qualitative themes and manual method for quantitative themes. In the particular case of the semi-automatic method, it also implements the three mapping algorithms. Finally, it also implements the three types of semantic queries.

The tool has been implemented in Java, namely version 1.5, and uses HP Jena, namely version 2.1. Jena is a Java framework for building Semantic Web applications, that provides an API for processing RDF, RDFS and OWL ontologies. It also provides an API for reasoning and includes a rule-based inference engine. The reasoning API can also connect to DIG reasoners as FaCT or Racer.

Our prototype implements a class called OntoManager that encapsulates all the operations related to the management of the ontology through Jena. This way, if a change of API is necessary, only this class has to be modified. In fact, the change from Jena 1 to Jena 2 forced us to completely modify this class, since the way how Jena manages ontology models suffered great changes from version 1 to 2. The prototype stores each repository ontology by means of an OWL document. This document fits the OWL DL profile and can be opened in any compliant OWL editor, as for instance Protégé. It is worth remarking that at the time of starting the development of OntoGIS there was no OWL plugin for Protégé, and this fact decided us to develop an autonomous tool.

The prototype deals with datasets in GeoTIFF formats. This is a common raster format in the GIS community, which is basically a TIFF image with a header describing the coordinates and cell properties of the raster. This header can be either embedded into the image file or in a separate file. The prototype supports metadata compliant with the standard FGDC CSDGM (see Chapter 3).

Regarding the interface, we have tried to avoid the use of specific vocabulary related to ontologies, in order to facilitate the use to users with little or no knowledge on ontologies. In principle, the user should be an expert on the GI domain, and not necessarily an expert on ontologies.

We include here some screenshots to show the main elements and functionalities of the OntoGIS prototype.

Figure 47 shows the starting screen of OntoGIS. We can see the six menus: “File” to open and close repository documents and to quit the application; “Themes” for managing the ontology of qualitative, quantitative and modelled themes; “Datasets” to manage qualitative and quantitative datasets; “Vocabularies” to manage normalized vocabularies of terms with narrower and broader terms and synonyms; “Query” to execute the three types of semantic queries; and finally “Help”, that provides a description of the system and functionalities.

Figure 48 shows the main screen to manage the taxonomy of qualitative themes. We can see in the left panel the tree of classes (concepts), in this case containing all CORINE elements. The user may add or delete new classes, and can also view those datasets that are connected to the selected class (first type of semantic query). In the right panel, the user can edit the documentation of the selected class, as well as all its semantic relations: superclasses, subclasses, equivalent classes, disjoint classes and property restrictions. The user can also create a new model (and add definitions to it) to the selected model from this screen. Figure 48 shows the screen for the management of the list of quantitative themes, and their classifications and classes. Figure 49 shows the screen for the management of modelled themes, where the user can add new models and their corresponding DL definitions.

Figure 51 shows the main screen to manage qualitative datasets. In the left panel we can see the list of the qualitative datasets that are currently in the repository, and the user can add new ones or delete an existing one. In the right panel we can see in the upper part the main metadata elements describing the abstract, purpose, bounding coordinates, main thematic attribute and the theme keyword. The lower part shows the application schema, with the list of permitted values for the thematic attribute, as well as their definitions. The user can connect a selected value with the taxonomy (manual merging), can view the theme connected to a selected value (first type of semantic query) and can also merge the complete dataset using the semi-automatic method (with the three different mapping algorithms). Figure 52 shows the screen for the management of quantitative dataset.

Figure 53 shows the window for the management of normalized vocabularies of terms. In the example, Corine is the only vocabulary in the repository. We can see in the middle panel the list of panels, where “Arable land” has been selected. In the right panel we can see its definition, and broader, narrower and synonym terms. The user can connect a particular term (manual merging method) or merge the whole vocabulary (semi-automatic merging method) from this screen.

How the different merging methods have been implemented in the prototype can be seen in Chapter 5.

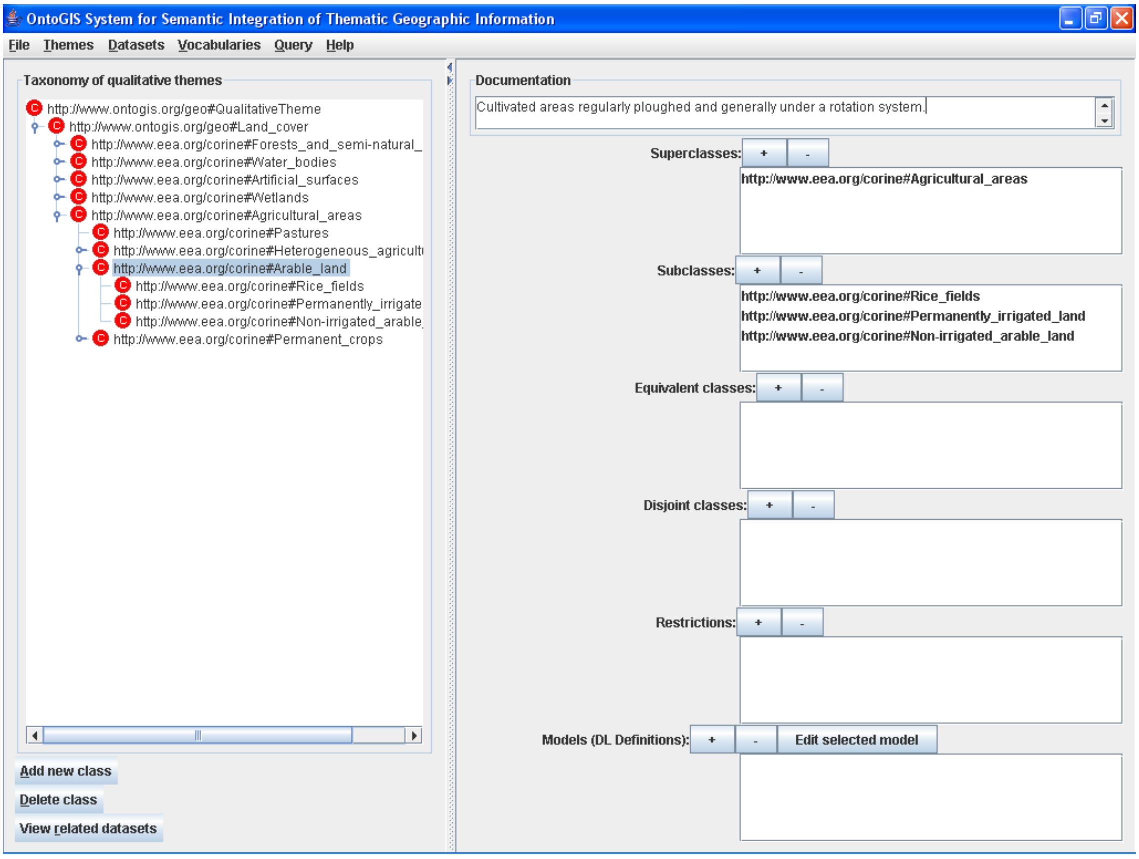
Figure 54 shows an example of the result of the first type of semantic query. In this case the user was trying to find dataset values connected to the theme “Forest and semi natural areas”. No dataset value is directly connected to it, but the value 7 “Pine tree forests” from dataset “Land cover Majorca” is connected to one of its subclasses.

Figure 55 shows an example of the result of the second type of semantic query. In this case the user has asked the system to translate the dataset “Land cover Majorca” to the only available vocabulary Corine. Only one value from the dataset has been connected to the taxonomy of qualitative classes, and consequently, only this value can be translated. The system generates a new dataset, with one value (1) whose definition is a reference to the Corine term “Coniferous forest”. Figure 56 shows how this new dataset is seen in a GIS environment as ESRI ArcMap.

Figure 57 shows an example of the result of the third type of semantic query. In this case two land cover datasets have been integrated focusing on “forests and semi-natural areas”. Finally, Figure 58 shows how the integrated dataset is seen in ESRI ArcMap.

OntoGIS System for Semantic Integration of Thematic Geographic Information
File
Themes
Datasets
Vocabularies
Query
Help
OntoGIS System

Figure 47. OntoGIS: starting screen



Taxonomy of qualitative themes

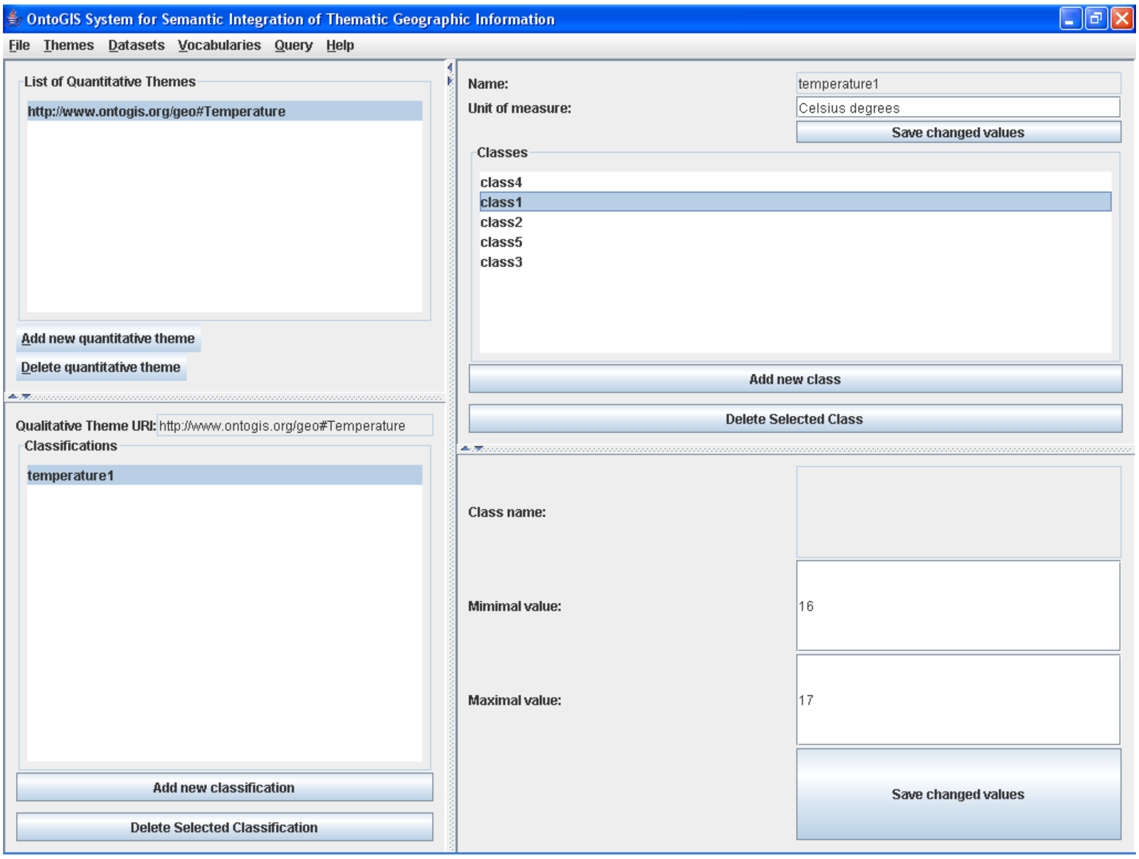
- http://www.ontogis.org/geo#QualitativeTheme
- http://www.ontogis.org/geo#Land_cover
  - http://www.eea.org/corine#Forests_and_semi-natural_
  - http://www.eea.org/corine#Water_bodies
  - http://www.eea.org/corine#Artificial_surfaces
  - http://www.eea.org/corine#Wetlands
  - http://www.eea.org/corine#Agricultural_areas
    - http://www.eea.org/corine#Pastures
    - http://www.eea.org/corine#Heterogeneous_agricult
    - http://www.eea.org/corine#Arable_land
      - http://www.eea.org/corine#Rice_fields
      - http://www.eea.org/corine#Permanently_irrigate
      - http://www.eea.org/corine#Non-irrigated_arable
    - http://www.eea.org/corine#Permanent_crops

Documentation: Cultivated areas regularly ploughed and generally under a rotation system.

Superclasses:
http://www.eea.org/corine#Agricultural_areas

Subclasses:
http://www.eea.org/corine#Rice_fields
http://www.eea.org/corine#Permanently_irrigated_land
http://www.eea.org/corine#Non-irrigated_arable_land

Figure 48. OntoGIS: management of the taxonomy of qualitative themes


OntoGIS System for Semantic Integration of Thematic Geographic Information
File
Themes
Datasets
Vocabularies
Query
Help
List of Quantitative Themes
Add new quantitative theme
Delete quantitative theme
Classifications
Add new classification
Delete Selected Classification
Classes
Save changed values
Add new class
Delete Selected Class
Save changed values

http://www.ontogis.org/geo#Temperature

Qualitative Theme URI: http://www.ontogis.org/geo#Temperature

temperature1

Name: temperature1

Unit of measure: Celsius degrees

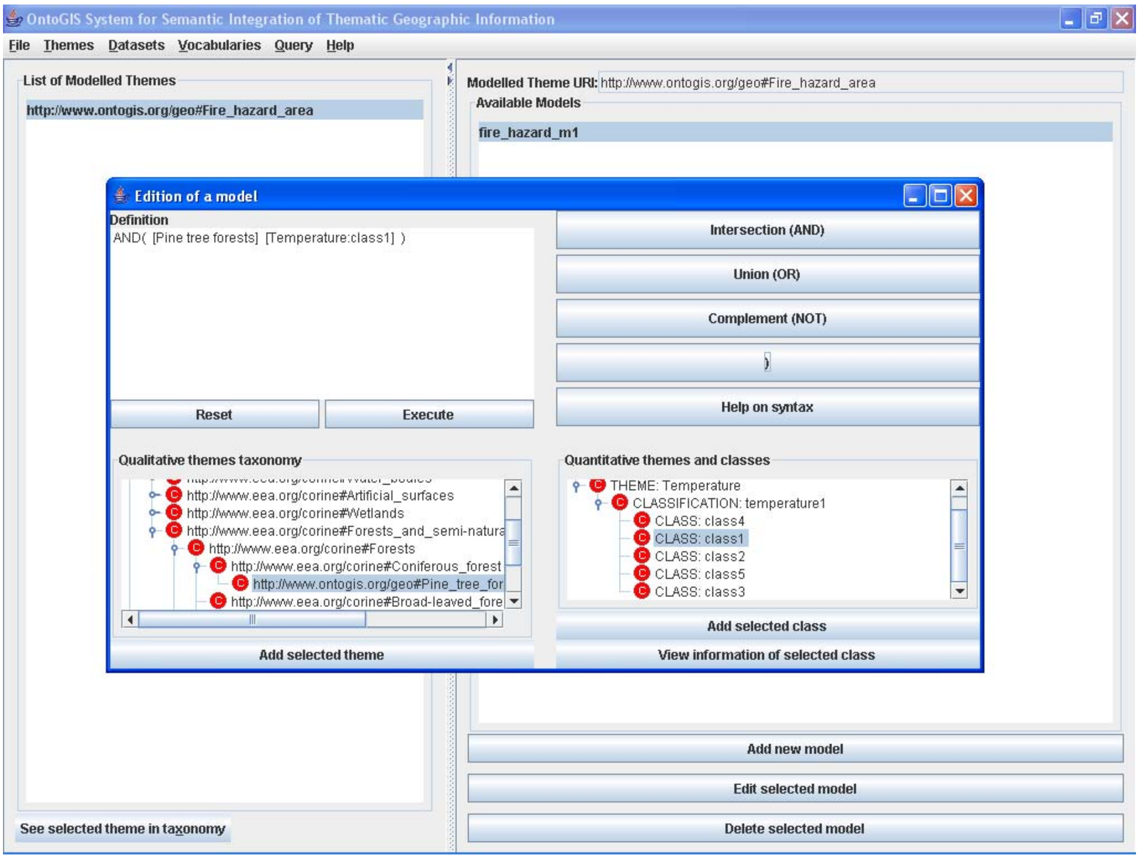
class4
class1
class2
class5
class3

Class name:

Mimimal value: 16

Maximal value: 17

Figure 49. OntoGIS: management of the list of quantitative themes



http://www.ontogis.org/geo#Fire_hazard_area

Modelled Theme URI: http://www.ontogis.org/geo#Fire_hazard_area

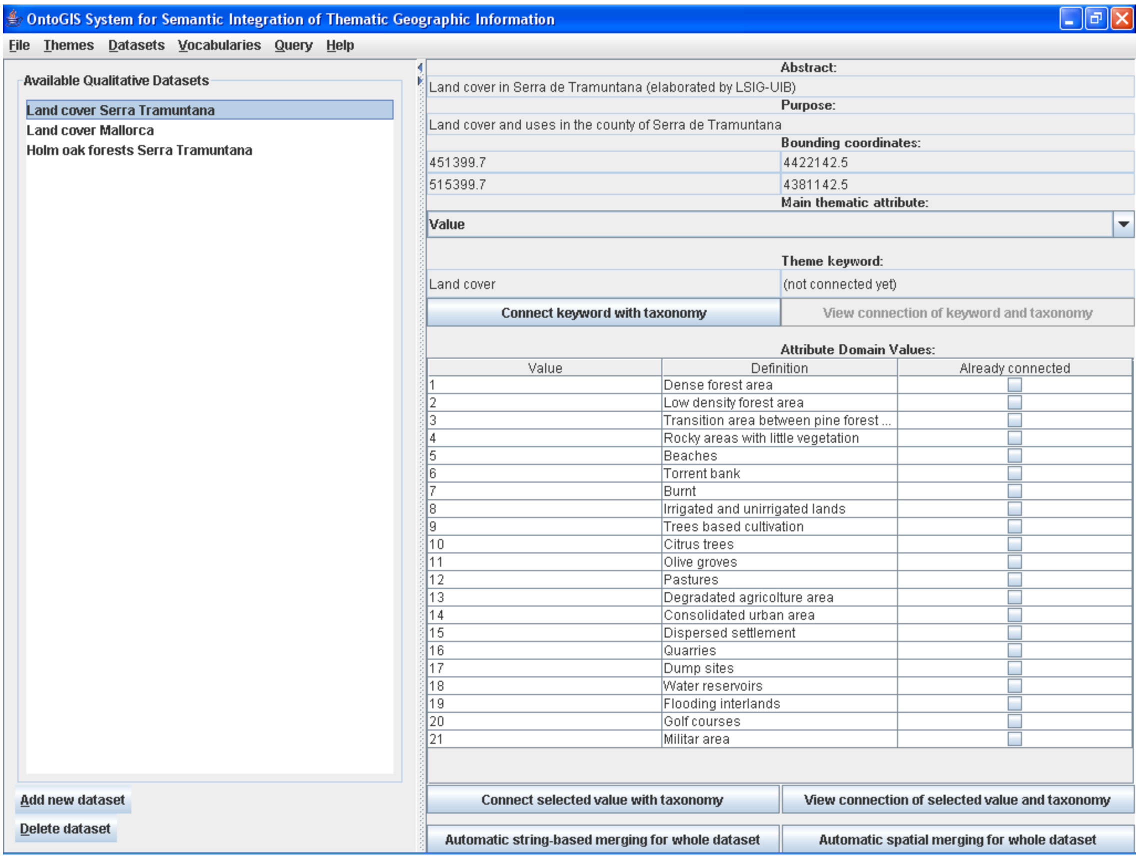
fire_hazard_m1

AND([Pine tree forests] [Temperature:class1])

http://www.eea.org/corine#Artificial_surfaces
http://www.eea.org/corine#Wetlands
http://www.eea.org/corine#Forests_and_semi-natura
http://www.eea.org/corine#Forests
http://www.eea.org/corine#Coniferous_forest
http://www.ontogis.org/geo#Pine_tree_for
http://www.eea.org/corine#Broad-leaved_fore

THEME: Temperature
CLASSIFICATION: temperature1
CLASS: class4
CLASS: class1
CLASS: class2
CLASS: class5
CLASS: class3

Figure 50. OntoGIS: management of the list of modelled themes, and their models


OntoGIS System for Semantic Integration of Thematic Geographic Information
File Themes Datasets Vocabularies Query Help
Available Qualitative Datasets
Add new dataset
Delete dataset
Connect keyword with taxonomy
View connection of keyword and taxonomy
Connect selected value with taxonomy
View connection of selected value and taxonomy
Automatic string-based merging for whole dataset
Automatic spatial merging for whole dataset

Land cover Serra Tramuntana
Land cover Mallorca
Holm oak forests Serra Tramuntana

Abstract:
Land cover in Serra de Tramuntana (elaborated by LSIG-UIB)

Purpose:
Land cover and uses in the county of Serra de Tramuntana

Bounding coordinates:

451399.7	4422142.5
515399.7	4381142.5

Main thematic attribute:
Value

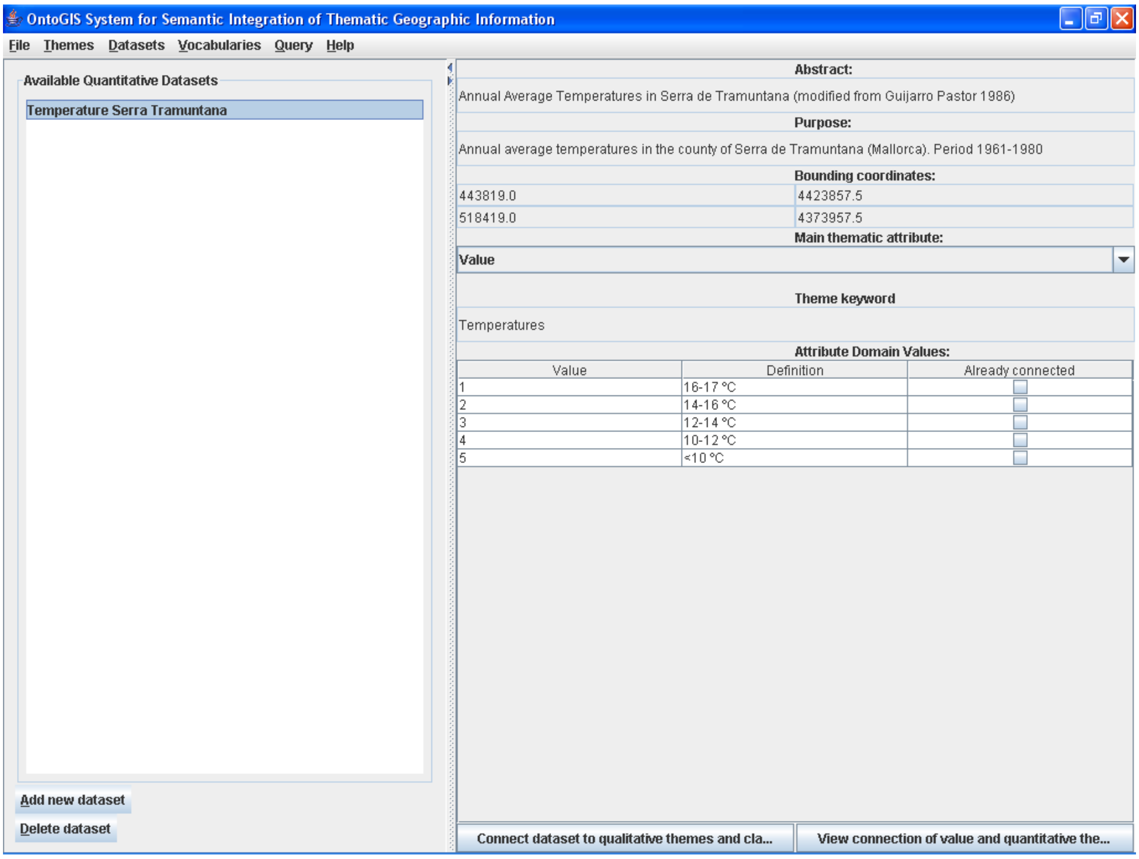
Theme keyword:

Land cover	(not connected yet)

Attribute Domain Values:

Value	Definition	Already connected
1	Dense forest area	□
2	Low density forest area	□
3	Transition area between pine forest ...	□
4	Rocky areas with little vegetation	□
5	Beaches	□
6	Torrent bank	□
7	Burnt	□
8	Irrigated and unirrigated lands	□
9	Trees based cultivation	□
10	Citrus trees	□
11	Olive groves	□
12	Pastures	□
13	Degradated agricolture area	□
14	Consolidated urban area	□
15	Dispersed settlement	□
16	Quarries	□
17	Dump sites	□
18	Water reservoirs	□
19	Flooding interlands	□
20	Golf courses	□
21	Militar area	□

Figure 51. OntoGIS: management of qualitative datasets


OntoGIS System for Semantic Integration of Thematic Geographic Information
File Themes Datasets Vocabularies Query Help
Available Quantitative Datasets
Add new dataset
Delete dataset
Connect dataset to qualitative themes and cla...
View connection of value and quantitative the...

Temperature Serra Tramuntana

Abstract:
Annual Average Temperatures in Serra de Tramuntana (modified from Guijarro Pastor 1986)

Purpose:
Annual average temperatures in the county of Serra de Tramuntana (Mallorca). Period 1961-1980

Bounding coordinates:

443819.0	4423857.5
518419.0	4373957.5

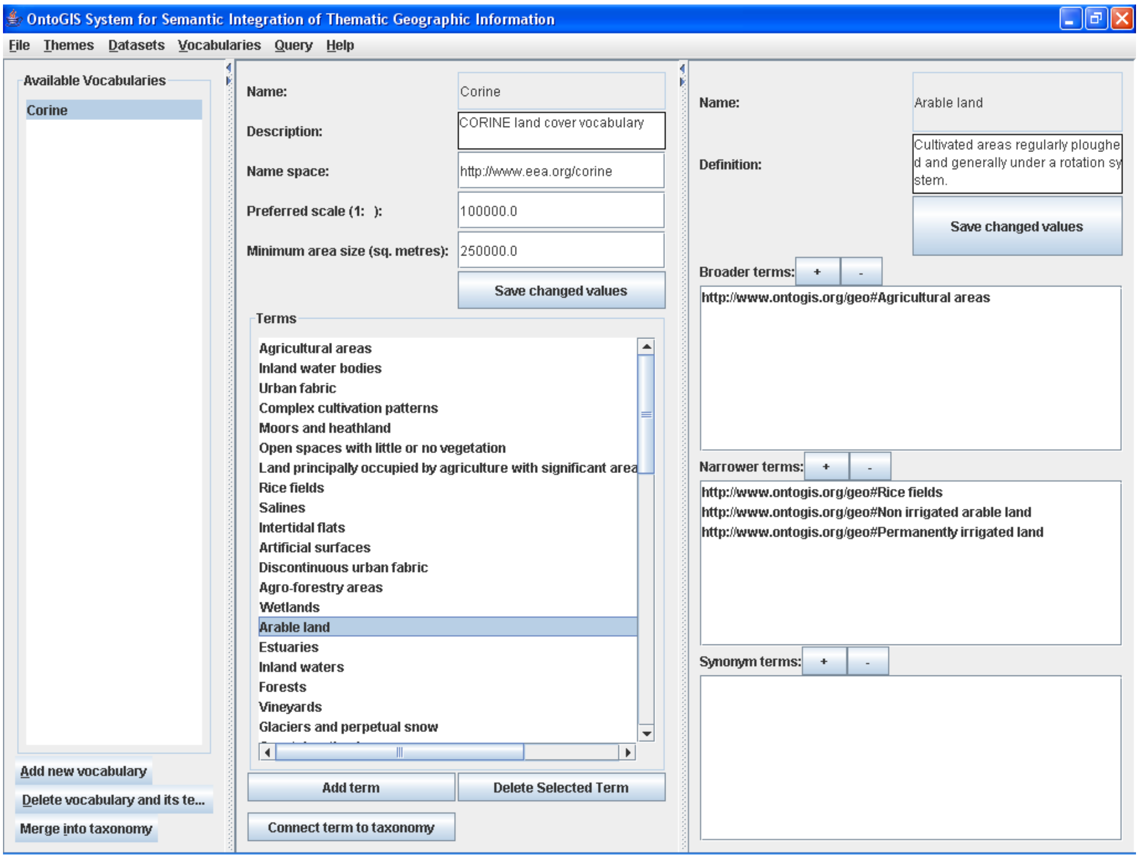
Main thematic attribute:
Value

Theme keyword
Temperatures

Attribute Domain Values:

Value	Definition	Already connected
1	16-17 °C	□
2	14-16 °C	□
3	12-14 °C	□
4	10-12 °C	□
5	<10 °C	□

Figure 52. OntoGIS: management of quantitative datasets



Corine

Name: Corine

Description: CORINE land cover vocabulary

Name space: http://www.eea.org/corine

Preferred scale (1:): 100000.0

Minimum area size (sq. metres): 250000.0

Agricultural areas
Inland water bodies
Urban fabric
Complex cultivation patterns
Moors and heathland
Open spaces with little or no vegetation
Land principally occupied by agriculture with significant area
Rice fields
Salines
Intertidal flats
Artificial surfaces
Discontinuous urban fabric
Agro-forestry areas
Wetlands
Arable land
Estuaries
Inland waters
Forests
Vineyards
Glaciers and perpetual snow

Name: Arable land

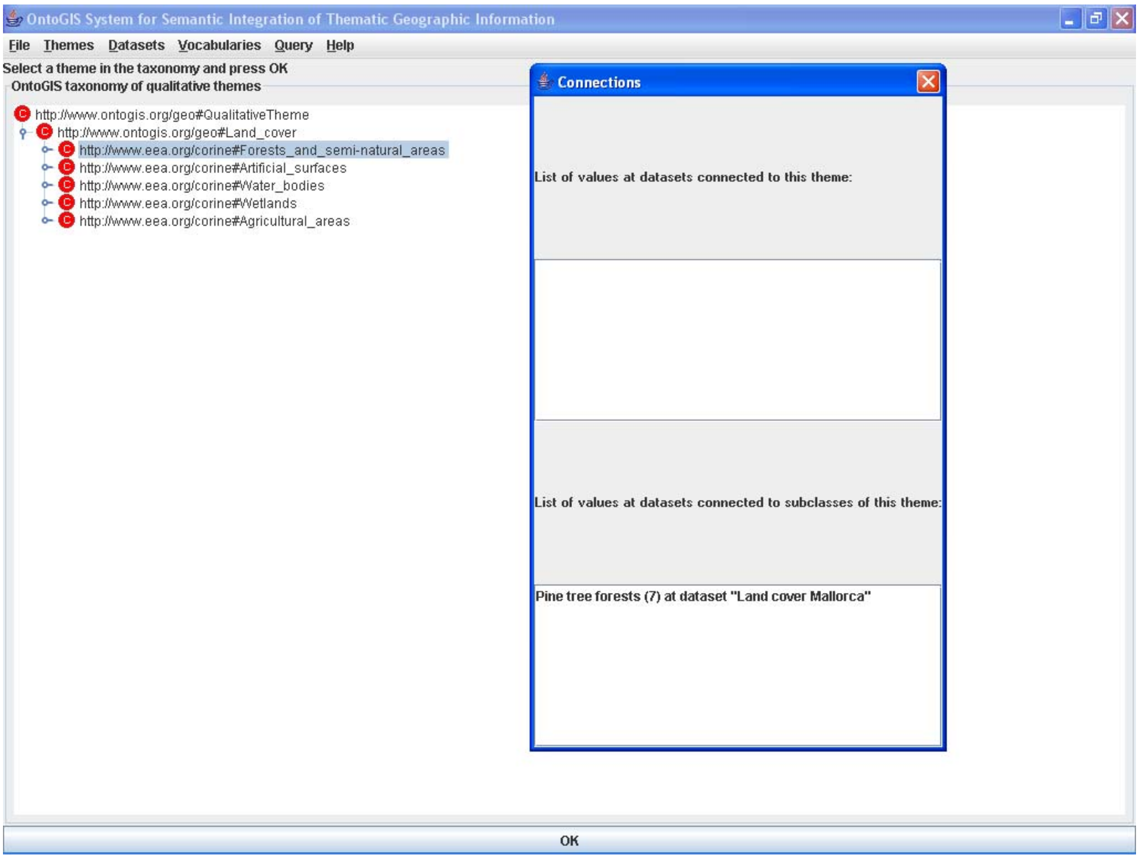
Definition: Cultivated areas regularly ploughe d and generally under a rotation sy stem.

Broader terms:
http://www.ontogis.org/geo#Agricultural areas

Narrower terms:
http://www.ontogis.org/geo#Rice fields
http://www.ontogis.org/geo#Non irrigated arable land
http://www.ontogis.org/geo#Permanently irrigated land

Synonym terms:

Figure 53. OntoGIS: management of normalized vocabularies of terms


OntoGIS System for Semantic Integration of Thematic Geographic Information
File
Themes
Datasets
Vocabularies
Query
Help
Connections
OK

Select a theme in the taxonomy and press OK

OntoGIS taxonomy of qualitative themes

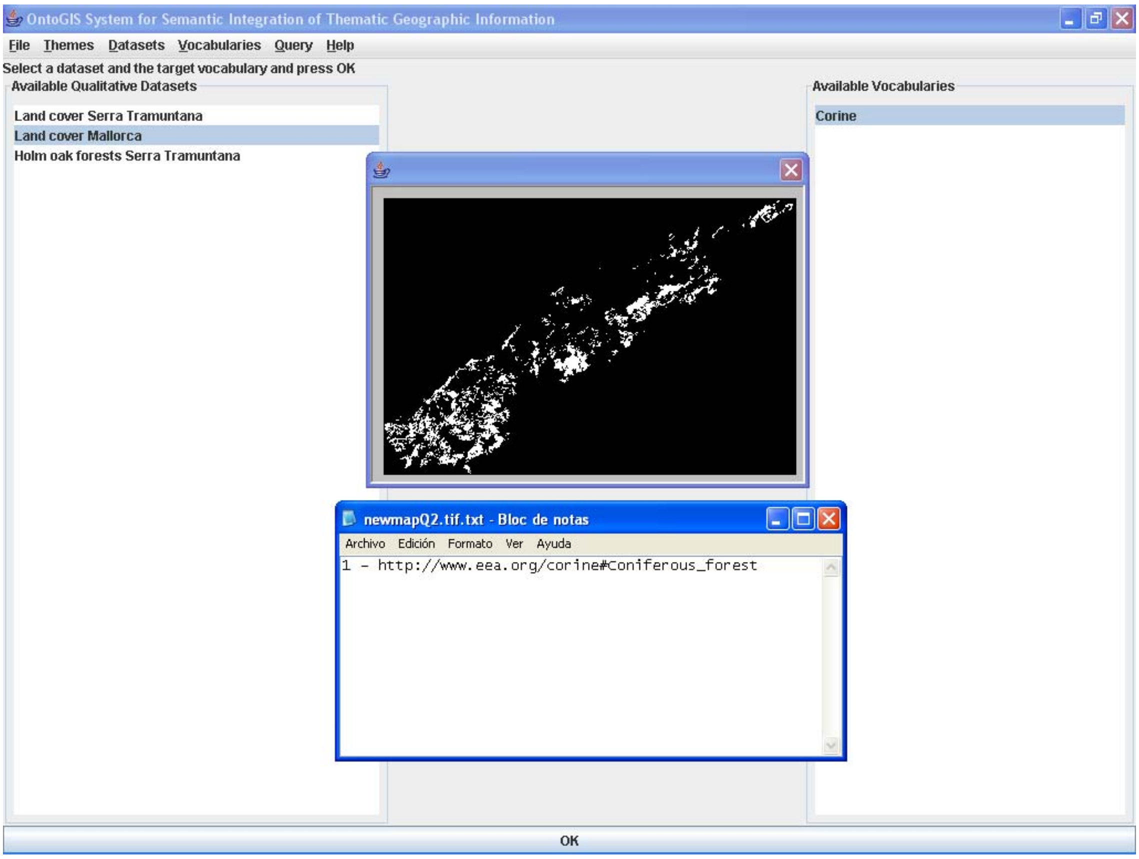
http://www.ontogis.org/geo#QualitativeTheme
http://www.ontogis.org/geo#Land_cover
http://www.eea.org/corine#Forests_and_semi-natural_areas
http://www.eea.org/corine#Artificial_surfaces
http://www.eea.org/corine#Water_bodies
http://www.eea.org/corine#Wetlands
http://www.eea.org/corine#Agricultural_areas

List of values at datasets connected to this theme:

List of values at datasets connected to subclasses of this theme:

Pine tree forests (7) at dataset "Land cover Mallorca"

Figure 54. OntoGIS: first type of semantic query


OntoGIS System for Semantic Integration of Thematic Geographic Information
File Themes Datasets Vocabularies Query Help
newmapQ2.tif.txt - Bloc de notas
Archivo Edición Formato Ver Ayuda
OK

Select a dataset and the target vocabulary and press OK

Available Qualitative Datasets

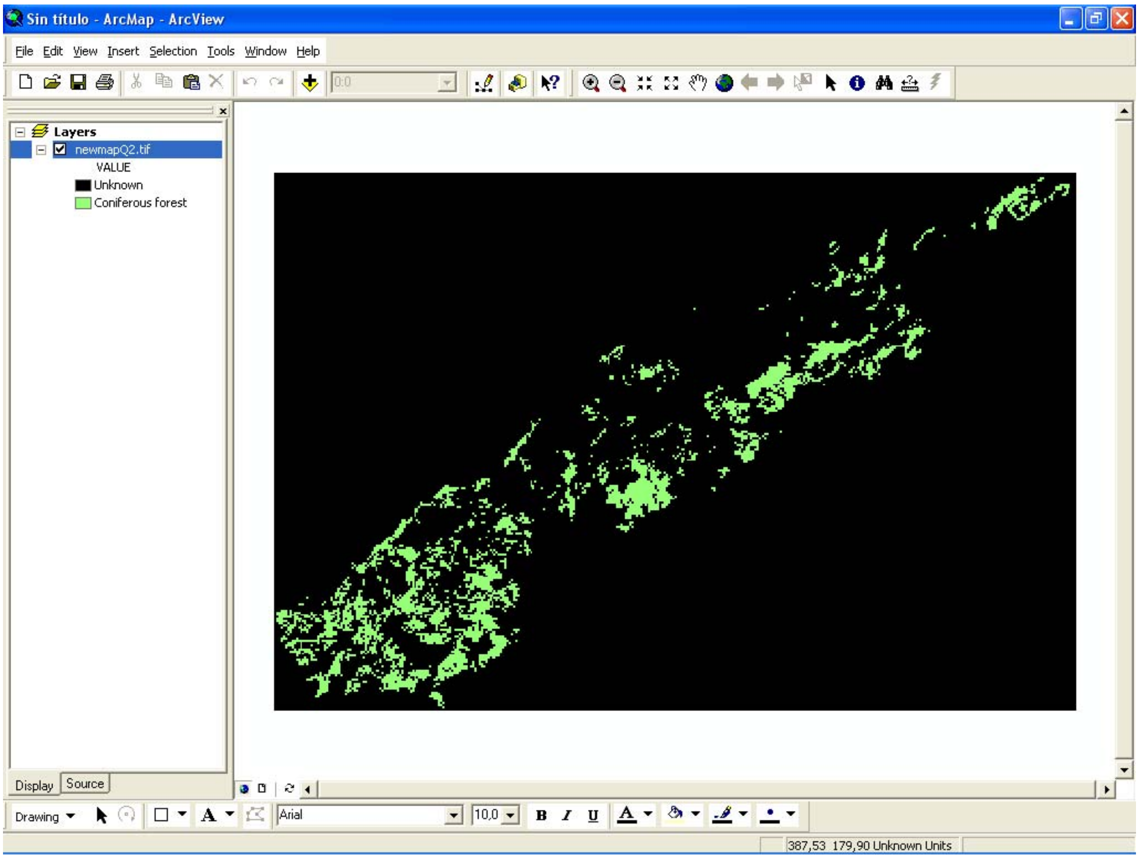
- Land cover Serra Tramuntana
- Land cover Mallorca
- Holm oak forests Serra Tramuntana

Available Vocabularies

- Corine

```
1 - http://www.eea.org/corine#Coniferous_forest
```

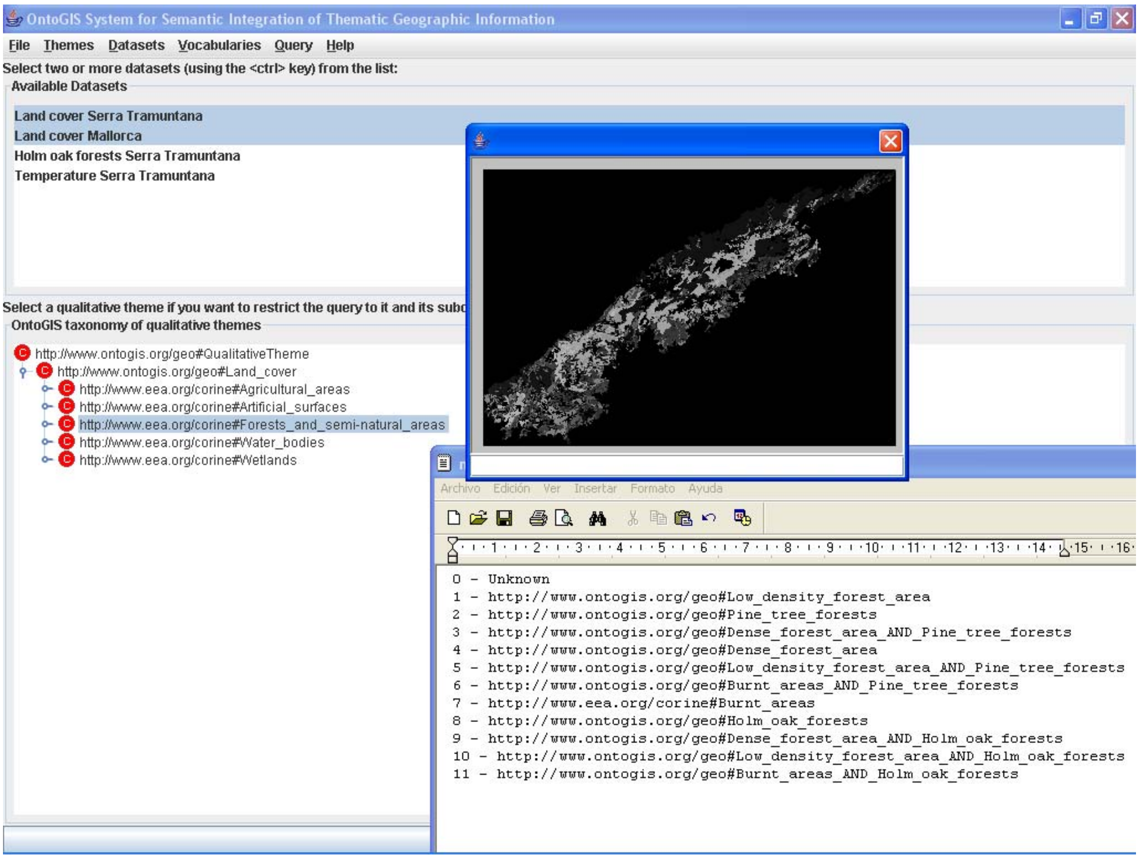
Figure 55. OntoGIS: second type of semantic query


Sin título - ArcMap - ArcView
File Edit View Insert Selection Tools Window Help
Display Source
Drawing
Arial
10,0

Layers
newmapQ2.tif
VALUE
■ Unknown
🟩 Coniferous forest

387,53 179,90 Unknown Units

Figure 56. Results of the second type of semantic query in ESRI ArcMap


OntoGIS System for Semantic Integration of Thematic Geographic Information
File Themes Datasets Vocabularies Query Help
Archivo Edición Ver Insertar Formato Ayuda

Select two or more datasets (using the <ctrl> key) from the list:

Available Datasets

- Land cover Serra Tramuntana
- Land cover Mallorca
- Holm oak forests Serra Tramuntana
- Temperature Serra Tramuntana

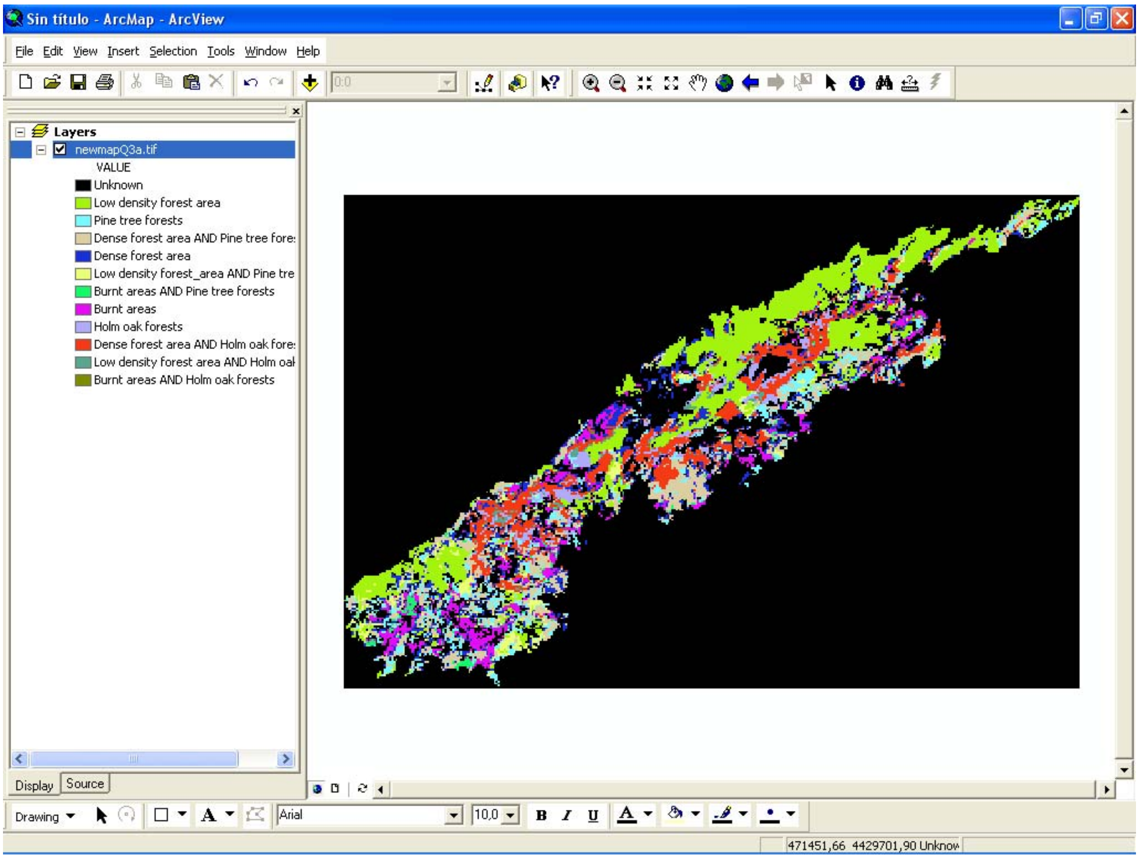
Select a qualitative theme if you want to restrict the query to it and its subc

OntoGIS taxonomy of qualitative themes

- http://www.ontogis.org/geo#QualitativeTheme
- http://www.ontogis.org/geo#Land_cover
  - http://www.eea.org/corine#Agricultural_areas
  - http://www.eea.org/corine#Artificial_surfaces
  - http://www.eea.org/corine#Forests_and_semi-natural_areas
  - http://www.eea.org/corine#Water_bodies
  - http://www.eea.org/corine#Wetlands

```
0 - Unknown
1 - http://www.ontogis.org/geo#Low_density_forest_area
2 - http://www.ontogis.org/geo#Pine_tree_forests
3 - http://www.ontogis.org/geo#Dense_forest_area_AND_Pine_tree_forests
4 - http://www.ontogis.org/geo#Dense_forest_area
5 - http://www.ontogis.org/geo#Low_density_forest_area_AND_Pine_tree_forests
6 - http://www.ontogis.org/geo#Burnt_areas_AND_Pine_tree_forests
7 - http://www.eea.org/corine#Burnt_areas
8 - http://www.ontogis.org/geo#Holm_oak_forests
9 - http://www.ontogis.org/geo#Dense_forest_area_AND_Holm_oak_forests
10 - http://www.ontogis.org/geo#Low_density_forest_area_AND_Holm_oak_forests
11 - http://www.ontogis.org/geo#Burnt_areas_AND_Holm_oak_forests
```

Figure 57. OntoGIS: third type of semantic query


Sin título - ArcMap - ArcView
File Edit View Insert Selection Tools Window Help
Display Source
Drawing
Arial
10,0
471451,66 4429701,90 Unknow

Layers

newmapQ3a.tif

VALUE

- ■ Unknown
- 🟩 Low density forest area
- 🟦 Pine tree forests
- ■ Dense forest area AND Pine tree fore:
- 🟦 Dense forest area
- 🟨 Low density forest_area AND Pine tre
- 🟩 Burnt areas AND Pine tree forests
- 🟪 Burnt areas
- 🟪 Holm oak forests
- 🟥 Dense forest area AND Holm oak fore:
- 🟩 Low density forest area AND Holm oal
- 🟫 Burnt areas AND Holm oak forests

Figure 58. Results of the third type of semantic query in ESRI ArcMap

Appendix B OWL document for the complete ontology

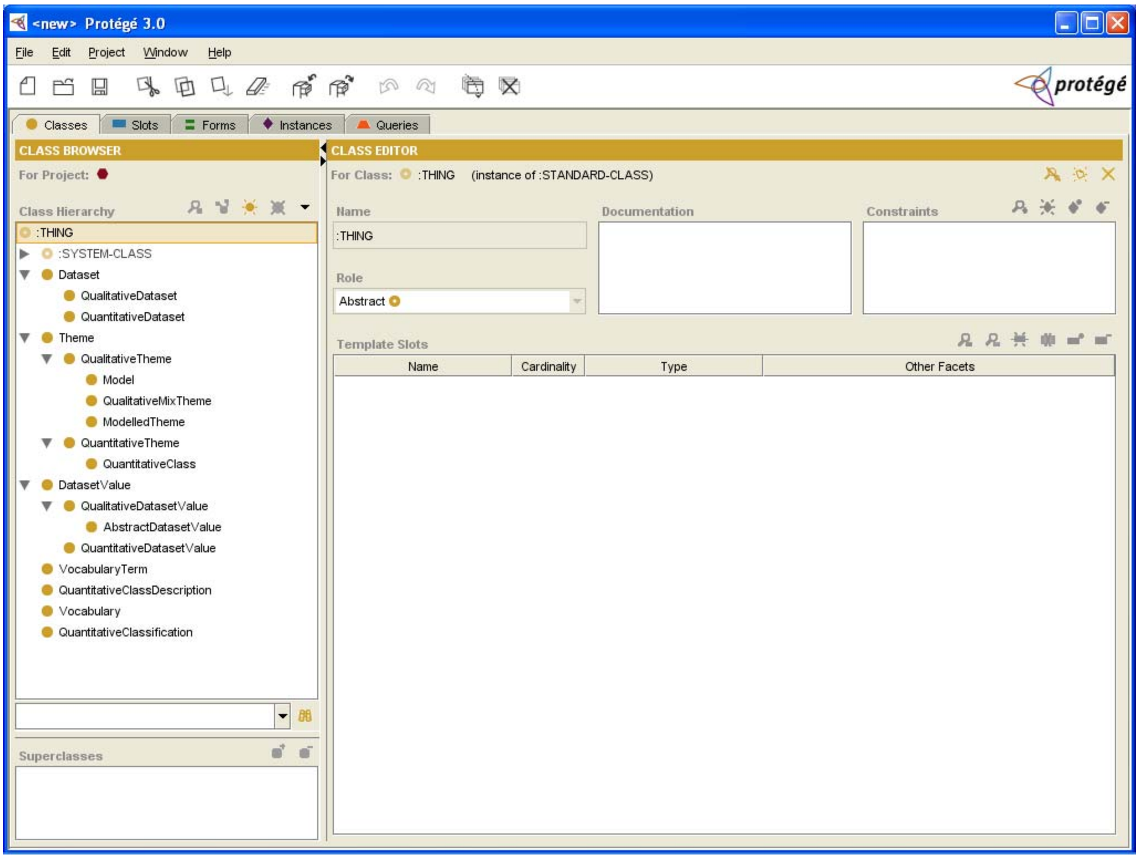
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.upf.edu/ontogis#"
    xml:base="http://www.upf.edu/ontogis">
<owl:Ontology rdf:about=""/>
<owl:Class rdf:ID="Dataset"/>
<owl:Class rdf:ID="QualitativeDataset">
      <rdfs:subClassOf rdf:resource="#Dataset"/>
</owl:Class>
<owl:Class rdf:ID="QuantitativeDataset">
      <rdfs:subClassOf rdf:resource="#Dataset"/>
</owl:Class>
<owl:Class rdf:ID="DatasetValue"/>
<owl:Class rdf:ID="QualitativeDatasetValue">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#DatasetValue"/>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="valueDataset"/>
            </owl:onProperty>
            <owl:allValuesFrom>
              <owl:Class rdf:ID="QualitativeDataset"/>
            </owl:allValuesFrom>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="QuantitativeDatasetValue">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#DatasetValue"/>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="valueDataset"/>
            </owl:onProperty>
            <owl:allValuesFrom>
              <owl:Class rdf:ID="QuantitativeDataset"/>
            </owl:allValuesFrom>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="AbstractDatasetValue">
      <rdfs:subClassOf rdf:resource="#QualitativeDatasetValue"/>
</owl:Class>
<owl:Class rdf:ID="Theme"/>
<owl:Class rdf:ID="QualitativeTheme">
      <rdfs:subClassOf rdf:resource="#Theme"/>
</owl:Class>
<owl:Class rdf:ID="QuantitativeTheme">
      <rdfs:subClassOf rdf:resource="#Theme"/>
</owl:Class>
<owl:Class rdf:ID="QuantitativeClassification"/>
<owl:Class rdf:ID="QuantitativeClassDescription"/>
<owl:Class rdf:ID="QuantitativeClass">
      <rdfs:subClassOf rdf:resource="#QuantitativeTheme"/>
</owl:Class>
<owl:Class rdf:ID="ModelledTheme">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#QualitativeTheme"/>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="themeModel"/>
            </owl:onProperty>
            <owl:someValuesFrom>
              <owl:Class rdf:ID="Model"/>
            </owl:someValuesFrom>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="Model">
    <rdfs:subClassOf rdf:resource="#QualitativeTheme"/>
</owl:Class>
<owl:Class rdf:ID="QualitativeMixTheme">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#QualitativeTheme"/>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="qualitativeThemeMixOf"/>
            </owl:onProperty>
            <owl:allValuesFrom>
              <owl:Class rdf:ID="QualitativeTheme"/>
            </owl:allValuesFrom>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="qualitativeThemeMixOf"/>
            </owl:onProperty>
              <owl:minCardinality rdf:datatype=
            "http://www.w3.org/2001/XMLSchema#nonNegativeInteger">2
              </owl:minCardinality>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="Vocabulary"/>
<owl:Class rdf:ID="VocabularyTerm"/>

<owl:DatatypeProperty rdf:ID="datasetTitle">
      <rdf:type rdf:resource="
http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetURI">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetMetadataURI">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetAbstract">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetPurpose">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetThemeKeyword">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetUTMZone">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#integer"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetBoundingNorth">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#float"/>
```

```
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetBoundingSouth">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetBoundingEast">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetBoundingWest">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="datasetThematicAttribute">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Dataset"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="valueName">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#DatasetValue"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="valueDefinition">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#DatasetValue"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="valueDataset">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#DatasetValue"/>
      <rdfs:range rdf:resource="#Dataset"/>
</owl:ObjectProperty>
<owl:TransitiveProperty rdf:ID="valueChildOf">
      <rdfs:domain rdf:resource="#QualitativeDatasetValue"/>
      <rdfs:range rdf:resource="#DatasetValue"/>
</owl:TransitiveProperty>
<owl:DatatypeProperty rdf:ID="themeName">
      <rdfs:domain rdf:resource="#Theme"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="qualitativeThemeConnection">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QualitativeTheme"/>
      <rdfs:range rdf:resource="#QualitativeDatasetValue"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="quantitativeThemeClassification">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeTheme"/>
      <rdfs:range rdf:resource="#QuantitativeClassification"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="quantitativeClassificationName">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassification"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="quantitativeClassificationUnit">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassification"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="quantitativeClassificationTheme">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassification"/>
      <rdfs:range rdf:resource="#QuantitativeTheme"/>
      <owl:inverseOf rdf:resource="#quantitativeThemeClassification"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="quantitativeClassName">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassDescription"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="quantitativeClassMinval">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassDescription"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="quantitativeClassMaxval">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassDescription"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="quantitativeClassClassification">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClassDescription"/>
      <rdfs:range rdf:resource=
            "#QuantitativeClassificationDescription"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="quantitativeClassDescription">
```

```
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeClass"/>
      <rdfs:range rdf:resource="#QuantitativeClassDescription"/>
      <owl:inverseOf rdf:resource="#quantitativeClassClassification"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="themeConnection">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Theme"/>
      <rdfs:range rdf:resource="#DatasetValue"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="qualitativeThemeConnection">
      <rdfs:subPropertyOf rdf:resource="#themeConnection"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QualitativeTheme"/>
      <rdfs:range rdf:resource="#QualitativeDatasetValue"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="quantitativeClassConnection">
      <rdfs:subPropertyOf rdf:resource="#themeConnection"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QuantitativeTheme"/>
      <rdfs:range rdf:resource="#QualitativeDatasetValue"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="datasetValueConnection">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
      <owl:inverseOf rdf:resource="#themeConnection"/>
      <rdfs:domain rdf:resource="#DatasetValue"/>
      <rdfs:range rdf:resource="#Theme"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="themeModel">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#QualitativeTheme"/>
      <rdfs:range rdf:resource="#Model"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="modelName">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Model"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="modelTheme">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
      <rdfs:domain rdf:resource="#Model"/>
      <rdfs:range rdf:resource="#QualitativeTheme"/>
      <owl:inverseOf rdf:resource="#themeModel"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="qualitativeThemeMixOf">
      <rdfs:domain rdf:resource="#QualitativeTheme"/>
      <rdfs:range rdf:resource="#QualitativeTheme"/>
</owl:ObjectProperty>
<owl:FunctionalProperty rdf:ID="vocabularyName">
      <rdfs:domain rdf:resource="#Vocabulary"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="vocabularyDescription">
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
      <rdfs:domain rdf:resource="#Vocabulary"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="vocabularyNamespace">
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
      <rdfs:domain rdf:resource="#Vocabulary"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="vocabularyScale">
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#float"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
      <rdfs:domain rdf:resource="#Vocabulary"/>
</owl:FunctionalProperty>
<owl:DatatypeProperty rdf:ID="vocabularyMinAreaSize">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Vocabulary"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:FunctionalProperty rdf:ID="termName">
      <rdfs:domain rdf:resource="#VocabularyTerm"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="termDefinition">
      <rdfs:domain rdf:resource="#VocabularyTerm"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
      <rdfs:range rdf:resource=
            "http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="termBroader">
      <rdfs:domain rdf:resource="#VocabularyTerm"/>
      <rdfs:range rdf:resource="#VocabularyTerm"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#ObjectProperty"/>
      <owl:inverseOf rdf:resource="#termNarrower"/>
</owl:FunctionalProperty>
<owl:ObjectProperty rdf:ID="termNarrower">
      <rdfs:range rdf:resource="#VocabularyTerm"/>
      <rdfs:domain rdf:resource="#VocabularyTerm"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="termSynonym">
```

```
      <rdfs:range rdf:resource="#VocabularyTerm"/>
      <rdfs:domain rdf:resource="#VocabularyTerm"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="themeTermConnection">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
      <rdfs:domain rdf:resource="#Theme"/>
      <rdfs:range rdf:resource="#VocabularyTerm"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="termThemeConnection">
      <rdf:type rdf:resource=
            "http://www.w3.org/2002/07/owl#InverseFunctionalProperty"/>
      <owl:inverseOf rdf:resource="#themeTermConnection"/>
      <rdfs:domain rdf:resource="#VocabularyTerm"/>
      <rdfs:range rdf:resource="#Theme"/>
</owl:ObjectProperty>
</rdf:RDF>
```



For Class: :THING (instance of :STANDARD-CLASS)

Name: :THING

Role: Abstract

Name	Cardinality	Type	Other Facets

```
:THING
  :SYSTEM-CLASS
  Dataset
    QualitativeDataset
    QuantitativeDataset
  Theme
    QualitativeTheme
      Model
      QualitativeMixTheme
      ModelledTheme
    QuantitativeTheme
      QuantitativeClass
  DatasetValue
    QualitativeDatasetValue
      AbstractDatasetValue
    QuantitativeDatasetValue
  VocabularyTerm
  QuantitativeClassDescription
  Vocabulary
  QuantitativeClassification
```

Figure 59. Ontology classes seen in Protégé 3.0

Appendix C CORINE and Anderson Land cover/Land use classifications

CORINE land cover classification (Bossard et al. 2000):

1. Artificial surfaces
    - 1.1. Urban fabric
        - 1.1.1. Continuous urban fabric
        - 1.1.2. Discontinuous urban fabric
    - 1.2. Industrial, commercial and transport units
        - 1.2.1. Industrial or commercial units
        - 1.2.2. Road and rail networks and associated land
        - 1.2.3. Port areas
        - 1.2.4. Airports
    - 1.3. Mine, dump and construction sites
        - 1.3.1. Mineral extraction sites
        - 1.3.2. Dump sites
        - 1.3.3. Construction sites
    - 1.4. Artificial non-agricultural vegetated areas
        - 1.4.1. Green urban areas
        - 1.4.2. Sport and leisure facilities

2. Agricultural areas
    - 2.1.Arable land
        - 2.1.1. Non-irrigated arable land
        - 2.1.2. Permanently irrigated land
        - 2.1.3. Rice fields
    - 2.2. Permanent crops
        - 2.2.1. Vineyards
        - 2.2.2. Fruit trees and berry plantations
        - 2.2.3. Olive groves
    - 2.3. Pastures
        - 2.3.1. Pastures
    - 2.4. Heterogeneous agricultural areas
        - 2.4.1. Annual crops associated with permanent crops
        - 2.4.2. Complex cultivation
        - 2.4.3. Land principally occupied by agriculture, with significant areas of natural vegetation
        - 2.4.4. Agro-forestry areas

3. Forests and semi-natural areas
    - 3.1. Forests
        - 3.1.1. Broad-leaved forest
        - 3.1.2. Coniferous forest
        - 3.1.3. Mixed forest
    - 3.2. Shrub and/or herbaceous vegetation association
        - 3.2.1. Natural grassland
        - 3.2.2. Moors and heathland
        - 3.2.3. Sclerophyllous vegetation
        - 3.2.4. Transitional woodland shrub
    - 3.3. Open spaces with little or no vegetation
        - 3.3.1. Beaches, dunes, and sand plains
        - 3.3.2. Bare rock
        - 3.3.3. Sparsely vegetated areas
        - 3.3.4. Burnt areas
        - 3.3.5. Glaciers and perpetual snow

4. Wetlands
    - 4.1. Inland wetlands
        - 4.1.1. Inland marshes
        - 4.1.2.Peat bogs
    - 4.2. Coastal wetlands
        - 4.2.1. Salt marshes
        - 4.2.2. Salines
        - 4.2.3. Intertidal flats

5. Water bodies
    - 5.1. Inland waters
        - 5.1. 1. Water courses
        - 5.1.2. Water bodies
    - 5.2. Marine waters
        - 5.2.1. Coastal lagoons
        - 5.2.2. Estuaries
        - 5.2.3. Sea and ocean

Anderson land cover and land use classification (Anderson et al. 1976):

1. Urban or Built-up Land
    - 1.1. Residential
    - 1.2. Commercial and Services
    - 1.3. Industrial
    - 1.4. Transportation, Communications, and Utilities
    - 1.5. Industrial and Commercial Complexes
    - 1.6. Mixed Urban or Built-up Land
    - 1.7. Other Urban or Built-up Land

2. Agricultural Land
    - 2.1. Cropland and Pasture
    - 2.2. Orchards, Groves, Vineyards, Nurseries, and Ornamental Horticultural Areas
    - 2.3. Confined Feeding Operations
    - 2.4. Other Agricultural Land

3. Rangeland
    - 3.1. Herbaceous Rangeland
    - 3.2. Shrub and Brush Rangeland
    - 3.3. Mixed Rangeland

4. Forest Land
    - 4.1. Deciduous Forest Land
    - 4.2. Evergreen Forest Land
    - 4.3. Mixed Forest Land

5. Water
    - 5.1. Streams and Canals
    - 5.2. Lakes
    - 5.3. Reservoirs
    - 5.4. Bays and Estuaries

6. Wetland
    - 6.1. Forested Wetland
    - 6.2. Nonforested Wetland

7. Barren Land
    - 7.1. Dry Salt Flats.
    - 7.2. Beaches
    - 7.3. Sandy Areas other than Beaches
    - 7.4. Bare Exposed Rock
    - 7.5. Strip Mines Quarries, and Gravel Pits
    - 7.6. Transitional Areas
    - 7.7. Mixed Barren Land

8. Tundra
    - 8.1. Shrub and Brush Tundra
    - 8.2. Herbaceous Tundra
    - 8.3. Bare Ground Tundra
    - 8.4. Wet Tundra
    - 8.5. Mixed Tundra

9. Perennial Snow or Ice
    - 9.1. Perennial Snowfields
    - 9.2. Glaciers

Appendix D Detailed hierarchical merging of CORINE and Anderson

The following table presents the results of merging level 2 of the Anderson classification into level 1:

	Term-based lexical (α=0,75)	**WordNet without meronyms**	**WordNet with meronyms**	**GEMET**
1 Urban or Built-up Land				
11 Residential	No relation	Subclass of Urban or Built-up Land	Subclass of Urban or Built-up Land	Subclass of Urban or Built-up Land
12 Commercial and Services	No relation	No relation	No relation	No relation
13 Industrial	No relation	No relation	No relation	Subclass of Urban or Built-up Land
14 Transportation, Communications, and Utilities	No relation	No relation	No relation	No relation
15 Industrial and Commercial Complexes	No relation	No relation	No relation	Subclass of Urban or Built-up Land
16 Mixed Urban or Built-up Land	Subclass of Urban or Built-up Land	Subclass of Urban or Built-up Land	Subclass of Urban or Built-up Land	Subclass of Urban or Built-up Land
17 Other Urban or Built-up Land	Subclass of Urban or Built-up Land	Subclass of Urban or Built-up Land	Subclass of Urban or Built-up Land	Subclass of Urban or Built-up Land
2 Agricultural Land				
21 Cropland and Pasture	No relation	No relation	No relation	Subclass of Agricultural Land
22 Orchards, Groves, Vineyards, Nurseries, and Ornamental Horticultural Areas	No relation	Subclass of Agricultural Land	Subclass of Agricultural Land	Subclass of Agricultural Land
23 Confined Feeding Operations	No relation	No relation	No relation	No relation
24 Other Agricultural Land	Subclass of Agricultural Land	Subclass of Agricultural Land	Subclass of Agricultural Land	Subclass of Agricultural Land
3 Rangeland				
31 Herbaceous Rangeland	Subclass of Rangeland	Subclass of Rangeland	Subclass of Rangeland	Subclass of Rangeland
32 Shrub and Brush	Subclass of	Subclass of	Subclass of	Subclass of

Rangeland	Rangeland	Rangeland	Rangeland	Rangeland
33 Mixed Rangeland	Subclass of Rangeland	Subclass of Rangeland	Subclass of Rangeland	Subclass of Rangeland
4 Forest Land				
41 Deciduous Forest Land	Subclass of Forest Land	Subclass of Forest Land	Subclass of Forest Land	Subclass of Forest Land
42 Evergreen Forest Land	Subclass of Forest Land	Subclass of Forest Land	Subclass of Forest Land	Subclass of Forest Land
43 Mixed Forest Land	Subclass of Forest Land	Subclass of Forest Land	Subclass of Forest Land	Subclass of Forest Land
5 Water				
51 Streams and Canals	No relation	Subclass of Water	Subclass of Water	No relation
52 Lakes	No relation	Subclass of Water	Subclass of Water	No relation
53 Reservoirs	No relation	Subclass of Water	Subclass of Water	No relation
54 Bays and Estuaries	No relation	Subclass of Water	Subclass of Water	Subclass of Water
6 Wetland				
61 Forested Wetland	Subclass of Wetland	Subclass of Wetland	Subclass of Wetland	Subclass of Wetland
62 Nonforested Wetland	Subclass of Wetland	Subclass of Wetland	Subclass of Wetland	Subclass of Wetland
7 Barren Land				
71 Dry Salt Flats	No relation	No relation	No relation	No relation
72 Beaches	No relation	No relation	No relation	No relation
73 Sandy Areas other than Beaches	No relation	No relation	No relation	No relation
74 Bare Exposed Rock	No relation		No relation	
75 Strip Mines Quarries, and Gravel Pits	No relation	No relation	No relation	No relation
76 Transitional Areas	No relation	No relation	No relation	No relation
77 Mixed Barren Land	Subclass of Barren Land	Subclass of Barren Land	Subclass of Barren Land	Subclass of Barren Land
8 Tundra				
81 Shrub and Brush Tundra	Subclass of Tundra	Subclass of Tundra	Subclass of Tundra	Subclass of Tundra
82 Herbaceous Tundra	Subclass of Tundra	Subclass of Tundra	Subclass of Tundra	Subclass of Tundra
83 Bare Ground Tundra	Subclass of Tundra	Subclass of Tundra	Subclass of Tundra	Subclass of Tundra
84 Wet Tundra	Subclass of Tundra	Subclass of Tundra	Subclass of Tundra	Subclass of Tundra
85 Mixed Tundra	Subclass of Tundra	Subclass of Tundra	Subclass of Tundra	Subclass of Tundra
9 Perennial Snow or Ice				
91 Perennial Snowfields	Subclass of Perennial Snow or Ice	Subclass of Perennial Snow or Ice	Subclass of Perennial Snow or Ice	Subclass of Perennial Snow or Ice
92 Glaciers	No relation	Subclass of Perennial Snow or Ice	Subclass of Perennial Snow or Ice	No relation

Table 13. Results of merging level 2 of the Anderson classification into level 1

The following table presents the results of merging level 3 of the CORINE classification into levels 1 and 2:

	Term-based lexical (α=0,75)	**WordNet without meronyms**	**WordNet with meronyms**	**GEMET**
1. Artificial surfaces				
1.1. Urban fabric				
1.1.1. Continuous urban fabric	Subclass of Urban fabric	Subclass of Urban fabric	Subclass of Urban fabric	Subclass of Urban fabric
1.1.2. Discontinuous urban fabric	Subclass of Urban fabric	Subclass of Urban fabric	Subclass of Urban fabric	Subclass of Urban fabric
1.2. Industrial, commercial or transport units				
1.2.1. Industrial or commercial units	Subclass of Industrial, commercial and transport units	Subclass of Industrial, commercial and transport units	Subclass of Industrial, commercial and transport units	Subclass of Industrial, commercial and transport units
1.2.2. Road and rail networks and associated land	No relation	No relation	No relation	No relation
1.2.3. Port areas	No relation	No relation	No relation	No relation
1.2.4. Airports	No relation	No relation	No relation	No relation
1.3. Mine, dump and construction sites				
1.3.1. Mineral extraction sites	Subclass of Mine, dump and construction sites	Subclass of Mine, dump and construction sites	Subclass of Mine, dump and construction sites	Subclass of Mine, dump and construction sites
1.3.2. Dump sites	Subclass of Mine, dump and construction sites	Subclass of Mine, dump and construction sites	Subclass of Mine, dump and construction sites	Subclass of Mine, dump and construction sites
1.3.3. Construction sites	Subclass of Mine, dump and construction sites	Subclass of Mine, dump and construction sites	Subclass of Mine, dump and construction sites	Subclass of Mine, dump and construction sites
1.4. Artificial non-agricultural vegetated areas				
1.4.1. Green urban areas	Subclass of Urban fabric	Subclass of Urban fabric	Subclass of Urban fabric	Subclass of Urban fabric
1.4.2. Sport and leisure facilities	No relation	No relation	No relation	No relation
2. Agricultural areas				
2.1.Arable land				
2.1.1. Non-irrigated arable land	Subclass of Arable land	Subclass of Arable land	Subclass of Arable land	Subclass of Arable land
2.1.2. Permanently irrigated land	“Brother” of Permanent crops	“Brother” of Permanent crops	“Brother” of Permanent crops	“Brother” of Permanent crops
2.1.3. Rice fields	No relation	Subclass of Shrub and/or herbaceous vegetation	Subclass of Shrub and/or herbaceous vegetation association	No relation

		association		
2.2. Permanent crops				
2.2.1. Vineyards	No relation	No relation	No relation	No relation
2.2.2. Fruit trees and berry plantations	No relation	No relation	Subclass of Forests	No relation
2.2.3. Olive groves	No relation	Subclass of Forests	Subclass of Forests	No relation
2.3. Pastures				
2.3.1. Pastures	Equivalent to Pastures	Equivalent to Pastures	Equivalent to Pastures	Equivalent to Pastures
2.4. Heterogeneous agricultural areas				
2.4.1. Annual crops associated with permanent crops	No relation	No relation	No relation	No relation
2.4.2. Complex cultivation	No relation	Subclass of Industrial, commercial or transport units	Subclass of Industrial, commercial or transport units	Subclass of Agricultural areas
2.4.3. Land principally occupied by agriculture, with significant areas of natural vegetation	No relation	No relation	No relation	No relation
2.4.4. Agro-forestry areas	No relation	No relation	No relation	No relation
3. Forests and semi-natural areas				
3.1. Forests				
3.1.1. Broad-leaved forest	Subclass of Forests	Subclass of Forests	Subclass of Forests	Subclass of Forests
3.1.2. Coniferous forest	Subclass of Forests	Subclass of Forests	Subclass of Forests	Subclass of Forests
3.1.3. Mixed forest	Subclass of Forests	Subclass of Forests	Subclass of Forests	Subclass of Forests
3.2. Shrub and/or herbaceous vegetation association				
3.2.1. Natural grassland	No relation	Superclass of Pastures	Superclass of Pastures	Subclass of Forests
3.2.2. Moors and heathland	No relation	Subclass of Shrub and/or herbaceous vegetation association	Subclass of Shrub and/or herbaceous vegetation association	Subclass of Wetlands
3.2.3. Sclerophyllous vegetation	“Brother” of Shrub and/or herbaceous vegetation association	“Brother” of Shrub and/or herbaceous vegetation association	“Brother” of Shrub and/or herbaceous vegetation association	“Brother” of Shrub and/or herbaceous vegetation association
3.2.4. Transitional woodland shrub	Subclass of Shrub and/or herbaceous vegetation association	Subclass of Shrub and/or herbaceous vegetation association	Subclass of Shrub and/or herbaceous vegetation association	Subclass of Shrub and/or herbaceous vegetation association
3.3. Open spaces				

with little or no vegetation				
3.3.1. Beaches, dunes, and sand plains	No relation	No relation	No relation	No relation
3.3.2. Bare rock	No relation	No relation	No relation	No relation
3.3.3. Sparsely vegetated areas	“Brother” of Shrub and/or herbaceous vegetation association	“Brother” of Shrub and/or herbaceous vegetation association	“Brother” of Shrub and/or herbaceous vegetation association	“Brother” of Shrub and/or herbaceous vegetation association
3.3.4. Burnt areas	No relation	No relation	No relation	No relation
3.3.5. Glaciers and perpetual snow	No relation	No relation	No relation	No relation
4. Wetlands				
4.1. Inland wetlands				
4.1.1. Inland marshes	“Brother” of Inland wetlands	Subclass of Inland wetlands	Subclass of Inland wetlands	Subclass of Inland wetlands
4.1.2. Peat bogs	No relation	Subclass of Wetlands	Subclass of Wetlands	Subclass of Wetlands
4.2. Coastal wetlands				
4.2.1. Salt marshes	No relation	Subclass of Wetlands	Subclass of Wetlands	Subclass of Wetlands
4.2.2. Salines	No relation	No relation	No relation	“Brother” of Coastal wetlands
4.2.3. Intertidal flats	No relation	Subclass of Wetlands	Subclass of Wetlands	“Brother” of Coastal wetlands
5. Water bodies				
5.1. Inland waters				
5.1.1. Water courses	Subclass of Water bodies	Subclass of Water bodies	Subclass of Water bodies	Subclass of Water bodies
5.1.2. Inland Water bodies	Equivalent to Inland waters	Equivalent to Inland waters	Equivalent to Inland waters	Equivalent to Inland waters
5.2. Marine waters				
5.2.1. Coastal lagoons	“Brother” of Coastal wetlands	“Brother” of Coastal wetlands	“Brother” of Coastal wetlands	Subclass of Coastal wetlands
5.2.2. Estuaries	No relation	Subclass of Water bodies	Subclass of Water bodies	Subclass of Water bodies
5.2.3. Sea and ocean	No relation	Subclass Marine waters	Subclass Marine waters	Subclass of Water bodies

Table 14. Results of merging level 3 of the CORINE classification into levels 1 and 2

Appendix E Classifications of datasets in USGS Earth Land Cover Maps

United States Geological Survey classification (USGS), based on Anderson, where physical values in italic and abstract values in plane:

- *Urban and Built-Up Land* (1)
- Agricultural Land
  - Cropland and Pasture
    - *Dryland Cropland and Pasture* (2)
    - *Irrigated Cropland and Pasture* (3)
  - *Cropland/Grassland Mosaic* (5)
  - *Cropland/Woodland Mosaic* (6)
- Rangeland
  - Herbaceous Rangeland
    - *Grassland* (7)
  - Shrub and Brush Rangeland
    - *Shrubland* (8)
  - Mixed Rangeland
    - *Mixed Shrubland/Grassland* (9)
    - *Savanna* (10)
- Forest Land
  - Deciduous Forest Land
    - *Deciduous Broadleaf Forest* (11)
    - *Deciduous Needleleaf Forest* (12)
  - Evergreen Forest Land
    - *Evergreen Broadleaf Forest* (13)
    - *Evergreen Needleleaf Forest* (14)
  - Mixed Forest Land
    - *Mixed Forest* (15)
- *Water Bodies* (16)
- Wetland
  - *Wooded Wetland* (18)
  - *Herbaceous Wetland* (17)
- Barren Land
  - *Barren or Sparsely Vegetated* (19)
- Tundra
  - *Wooded Tundra* (21)
  - *Mixed Tundra* (22)
  - *Bare Ground Tundra* (23)
- *Snow or Ice* (24)

International Geosphere Biosphere Programme (IGBP) classification:

Evergreen Needleleaf Forest (1)
Evergreen Broadleaf Forest (2)
Deciduous Needleleaf Forest (3)
Deciduous Broadleaf Forest (4)
Mixed Forest (5)
Closed Shrublands (6)
Open Shrublands (7)
Woody Savannas (8)
Nonwoody Savannas (9)
Grasslands (10)
Permanent Wetlands (11)
Croplands (12)
Urban and Built-Up (13)
Cropland/Natural Vegetation Mosaic (14)
Snow and Ice (15)
Barren or Sparsely Vegetated (16)
Water Bodies (17)

Biosphere-Atmosphere Transfer Scheme (BAT) classification:

Crops, Mixed Farming (1)
Short Grass (2)
Evergreen Needleleaf Trees (3)
Deciduous Needleleaf Tree (4)
Deciduous Broadleaf Trees (5)
Evergreen Broadleaf Trees (6)
Tall Grass (7)
Desert (8)
Tundra (9)
Irrigated Crops (10)
Semidesert (11)
Icecaps and Glaciers (12)
Bogs and Marshes (13)
Inland Water (14)
Ocean (15)
Evergreen Shrubs (16)
Deciduous Shrubs (17)
Mixed Forest (18)
Interrupted Forest (19)

Simple Biosphere Model (SBM1) classification:

Evergreen Broadleaf Trees (1)
Broadleaf Deciduous Trees (2)
Deciduous and Evergreen Trees (3)
Evergreen Needleleaf Trees (4)
Deciduous Needleleaf Trees (5)
Groundcover with Trees and Shrubs (6)
Groundcover Only (7)
Broadleaf Shrubs with Perennial Groundcover (8)
Broadleaf Shrubs with Bare Soil (9)
Groundcover with Dwarf Trees and Shrubs (10)
Bare Soil (11)
Agriculture or Grassland (12)
Persistent Wetland (17)
Water (19)
Ice Cap and Glacier (20)

Simple Biosphere 2 Model (SBM2) classification:

Broadleaf Evergreen Trees (1)
Broadleaf Deciduous Trees (2)
Broadleaf and Needleleaf Trees (3)
Needleleaf Evergreen Trees (4)
Needleleaf Deciduous Trees (5)
Short Vegetation (6)
Shrubs with Bare Soil (7)
Dwarf Trees and Shrubs (8)
Agriculture or Grassland (9)
Water, Wetlands (10)
Ice Cap and Glacier (11)

Appendix F Spatial merging of USGS Earth Land Cover Maps

Class(es) in IGBP	Relation	Class(es) in USGS
Evergreen Needleleaf Forest (1)	equivalent	*Evergreen Needleleaf Forest* (14)
Evergreen Broadleaf Forest (2)	equivalent	*Evergreen Broadleaf Forest* (13)
Deciduous Needleleaf Forest (3)	equivalent	*Deciduous Needleleaf Forest* (12)
Deciduous Broadleaf Forest (4)	equivalent	*Deciduous Broadleaf Forest* (11)
Mixed Forest (5)	equivalent	*Mixed Forest* (15)
Union of - *Closed Shrublands* (6) - *Open Shrublands* (7) - *Woody Savannas* (8) - *Nonwoody Savannas* (9)	equivalent *	Union of - *Shrubland* (8) - *Mixed Shrubland/Grassland* (9) - *Savanna* (10) - *Wooded Tundra* (21)
Grasslands (10)	equivalent	*Grassland* (7)
Permanent Wetlands (11)	equivalent	Union of - *Herbaceous Wetland* (17) - *Wooded Wetland* (18)
Croplands (12)	equivalent	Union of: - *Dryland Cropland and Pasture* (2) - *Irrigated Cropland and Pasture* (3)
Urban and Built-up (13)	equivalent	*Urban and Built-Up Land* (1)
Cropland/Natural Vegetation Mosaic (14)	equivalent	Union of - *Cropland/Grassland Mosaic* (5) - *Cropland/Woodland Mosaic* (6)
Snow and Ice (15)	equivalent	*Snow or Ice* (24)
Barren or Sparsely Vegetated (16)	equivalent	Union of - *Barren or Sparsely Vegetated* (19) - *Mixed Tundra* (22) - *Bare Ground Tundra* (23)
Water Bodies (17)	equivalent	*Water Bodies* (16)

* *Nonwoody Savannas* (9) is a subclass of *Savanna* (10)

Table 15. Results of merging IGBP and USGS datasets using the algorithm based on the spatial distribution of dataset values with a threshold of 0.95

Class(es) in BAT	Relation	Class(es) in USGS
Union of - *Crops, Mixed Farming* (1) - *Deciduous Needleleaf Tree* (4) - *Tall Grass* (7) - *Mixed Forest* (18) - *Interrupted Forest* (19)	equivalent *	Union of - *Urban and Built-Up Land* (1) - *Dryland Cropland and Pasture* (2) - *Cropland/Grassland Mosaic* (5) - *Cropland/Woodland Mosaic* (6) - *Savanna* (10) - *Deciduous Needleleaf Forest* (12) - *Mixed Forest* (15)
Short Grass (2)	equivalent	*Grassland* (7)
Evergreen Needleleaf Trees (3)	equivalent	*Evergreen Needleleaf Forest* (14)
Deciduous Broadleaf Trees (5)	equivalent	*Deciduous Broadleaf Forest* (11)

Evergreen Broadleaf Trees (6)	equivalent	*Evergreen Broadleaf Forest* (13)
Tall Grass (7)	subclass	*Savanna* (10)
Desert (8)	equivalent	Union of - *Barren or Sparsely Vegetated* (19) - *Bare Ground Tundra* (23)
Tundra (9)	equivalent	Union of - *Wooded Tundra* (21) - *Mixed Tundra* (22)
Irrigated Crops (10)	equivalent	*Irrigated Cropland and Pasture* (3)
Union of - *Semidesert* (11) - *Evergreen Shrubs* (16) - *Deciduous Shrubs* (17)	equivalent **	Union of - *Shrubland* (8) - *Mixed Shrubland/Grassland* (9)
Icecaps and Glaciers (12)	equivalent	*Snow or Ice* (24)
Bogs and Marshes (13)	equivalent	Union of - *Wooded Wetland* (18) - *Herbaceous Wetland* (17)
Union of - *Inland Water* (14) - *Ocean* (15)	equivalent	*Water Bodies* (16)

* *Crops, Mixed Farming* (1) is a superclass of *Urban and Built-Up Land* (1) and *Dryland Cropland and Pasture* (2); *Deciduous Needleleaf Tree* (4) is a superclass of *Deciduous Needleleaf Forest* (12); Mixed Forest (18) is a subclass of Mixed Forest (15); and *Interrupted Forest* (19) is a superclass of *Cropland/Woodland Mosaic* (6)

** Deciduous Shrubs (17) is a subclass of Shrubland (8)

Table 16. Results of merging BAT and USGS datasets using the algorithm based on the spatial distribution of dataset values with a threshold of 0.95

Class(es) in SBM2	Relation	Class(es) in USGS
Broadleaf Evergreen Trees (1)	equivalent	*Evergreen Broadleaf Forest* (13)
Union of - *Broadleaf Deciduous Trees* (2) - *Broadleaf and Needleleaf Trees* (3) - *Needleleaf Evergreen Trees* (4) - *Needleleaf Deciduous Trees* (5)	equivalent *	Union of - *Deciduous Broadleaf Forest* (11) - *Deciduous Needleleaf Forest* (12) - *Evergreen Needleleaf Forest* (14) - *Mixed Forest* (15)
Short Vegetation (6)	equivalent	*Savanna* (10)
Shrubs with Bare Soil (7)	equivalent	Union of - *Shrubland* (8) - *Mixed Shrubland/Grassland* (9) - *Barren or Sparsely Vegetated* (19) - *Bare Ground Tundra* (23)
Dwarf Trees and Shrubs (8)	superclass	*Wooded Tundra* (21)
Dwarf Trees and Shrubs (8)	superclass	*Mixed Tundra* (22)
Dwarf Trees and Shrubs (8)	superclass	*Wooded Wetland* (18)
Agriculture or Grassland (9)	equivalent	Union of - *Urban and Built-Up Land* (1) - *Dryland Cropland and Pasture* (2) - *Irrigated Cropland and Pasture* (3) - *Cropland/Grassland Mosaic* (5) - *Cropland/Woodland Mosaic* (6) - *Grassland* (7)
Water, Wetlands (10)	equivalent	Union of - *Water Bodies* (16) - *Herbaceous Wetland* (17)
Ice Cap and Glacier (11)	equivalent	*Snow or Ice* (24)

* Broadleaf Deciduous Trees (2) Subclass of Deciduous Broadleaf Forest (11); Needleleaf Deciduous Trees (5) superclass of Deciduous Needleleaf Forest (12); and Needleleaf Evergreen Trees (4) superclass of Evergreen Needleleaf Forest (14)

Table 17. Results of merging SBM2 and USGS datasets using the algorithm based on the spatial distribution of dataset values with a threshold of 0.95

Class(es) in SBM1	Relation	Class(es) in USGS
Evergreen Broadleaf Trees (1)	equivalent	*Evergreen Broadleaf Forest* (13)
Union of - *Broadleaf Deciduous Trees* (2) - *Deciduous and Evergreen Trees* (3) - *Evergreen Needleleaf Trees* (4) - *Deciduous Needleleaf Trees* (5)	equivalent *	Union of - *Deciduous Broadleaf Forest* (11) - *Deciduous Needleleaf Forest* (12) - *Evergreen Needleleaf Forest* (14) - *Mixed Forest* (15)
Union of - *Agriculture or Grassland* (12) - *Groundcover with Trees and Shrubs* (6) - *Groundcover Only* (7)	equivalent **	Union of - *Urban and Built-Up Land* (1) - *Dryland Cropland and Pasture* (2) - *Irrigated Cropland and Pasture* (3) - *Cropland/Grassland Mosaic* (5) - *Grassland* (7) - *Cropland/Woodland Mosaic* (6) - *Savanna* (10)
Groundcover with Dwarf Trees and Shrubs (10)	equivalent	Union of - *Wooded Tundra* (21) - *Mixed Tundra* (22)
Bare Soil (11)	equivalent	Union of - *Barren or Sparsely Vegetated* (19) - *Bare Ground Tundra* (23)
Persistent Wetland (17)	equivalent	Union of - *Herbaceous Wetland* (17) - *Wooded Wetland* (18)
Water (19)	equivalent	*Water Bodies* (16)
Ice Cap and Glacier (20)	equivalent	*Snow or Ice* (24)
Union of - *Broadleaf Shrubs with Perennial Groundcover* (8) - *Broadleaf Shrubs with Bare Soil* (9)	equivalent	Union of - *Shrubland* (8) - *Mixed Shrubland/Grassland* (9)

* *Broadleaf Deciduous Trees* (2) is subclass of *Deciduous Broadleaf Forest* (11); *Deciduous Needleleaf Trees* (5) superclass of *Deciduous Needleleaf Forest* (12) and *Evergreen Needleleaf Trees* (4) is superclass of *Evergreen Needleleaf Forest* (14)

** *Agriculture or Grassland* (12) is superclass of *Urban and Built-Up Land* (1), *Dryland Cropland and Pasture* (2), *Irrigated Cropland and Pasture* (3) and *Cropland/Grassland Mosaic* (5); G*roundcover with Trees and Shrubs* (6) is superclass of *Savanna* (10); and *Groundcover Only* (7) is subclass of *Grassland* (7)

Table 18. Results of merging SBM1 and USGS datasets using the algorithm based on the spatial distribution of dataset values with a threshold of 0.95

Class(es) in SBM2	Relation	Class(es) in SBM1
Broadleaf Evergreen Trees (1)	equivalent	*Evergreen Broadleaf Trees* (1)
Broadleaf Deciduous Trees (2)	equivalent	*Broadleaf Deciduous Trees* (2)
Needleleaf Evergreen Trees (4)	equivalent	*Evergreen Needleleaf Trees* (4)
Needleleaf Deciduous Trees (5)	equivalent	*Deciduous Needleleaf Trees* (5)
Union of - *Broadleaf and Needleleaf Trees* (3) - *Short Vegetation* (6)	equivalent *	Union of - *Deciduous and Evergreen Trees* (3) - *Groundcover with Trees and Shrubs* (6)
Shrubs with Bare Soil (7)	equivalent	Union of

		- *Broadleaf Shrubs with Bare Soil* (9) - *Bare Soil (*11) - *Broadleaf Shrubs with Perennial Groundcover* (8)
Dwarf Trees and Shrubs (8)	equivalent	Union of - *Groundcover with Dwarf Trees and Shrubs* (10) - *Persistent Wetland* (17)
Agriculture or Grassland (9)	equivalent	Union of - *Agriculture or Grassland* (12) - *Groundcover Only* (7)
Water, Wetlands (10)	equivalent	*Water* (19)
Ice Cap and Glacier (11)	equivalent	*Ice Cap and Glacier* (20)

* *Short Vegetation/Grassland* (6) subclass of *Groundcover with Trees and Shrubs* (6), and *Broadleaf and Needleleaf Trees* (3) superclass of *Deciduous and Evergreen Trees* (3)

Table 19. Results of merging SBM2 and SBM1 datasets using the algorithm based on the spatial distribution of dataset values with a threshold of 0.95

Appendix G Schemas for meta-information of images and video

XML Schema for the meta-information of a still image (see 11.4.1.1):

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://www.upf.edu/videogis"
            xmlns="http://www.upf.edu/videogis"
            elementFormDefault="qualified">
  <xsd:element name="Image">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="CameraProperties"
                  minOccurs="1" maxOccurs="1"/>
         <xsd:element ref="Layer"
                  minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="CameraProperties">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="CameraPosition"
                  minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="CameraOrientation"
                  minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="CameraTilt" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="CameraAngleOfVision"
                  minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="CameraPosition">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="X" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Y" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Z" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="CameraOrientation" type="xsd:float"/>
  <xsd:element name="CameraTilt" type="xsd:float"/>
  <xsd:element name="CameraAngleOfVision" type="xsd:float"/>
  <xsd:element name="X" type="xsd:float"/>
  <xsd:element name="Y" type="xsd:float"/>
  <xsd:element name="Z" type="xsd:float"/>

  <xsd:element name="Layer">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="VisibleTheme"
                  minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="indexingTheme"
                  type="xsd:string" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="VisibleTheme">
    <xsd:complexType>
      <xsd:attribute name="theme" type="xsd:string" use="required"/>
      <xsd:attribute name="spatialExtentArea"
                  type="xsd:float" use="required"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

XML Schema for the meta-information of a video (see 11.4.2.1):

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://www.upf.edu/videogis"
            xmlns="http://www.upf.edu/videogis"
            elementFormDefault="qualified">
  <xsd:element name="Video">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="VideoSegment"
                  minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="uri" type="xsd:string" use="required"/>
      <xsd:attribute name="fps" type="xsd:float" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="VideoSegment">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="CameraPropertiesAtFrame"
                  minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="VisibleTheme"
                  minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="indexingTheme"
                  type="xsd:string" use="required"/>
      <xsd:attribute name="startFrame"
                  type="xsd:integer" use="required"/>
      <xsd:attribute name="endFrame"
                  type="xsd:integer" use="required"/>
      <xsd:attribute name="averageVisibleArea"
                  type="xsd:float" use="required"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="CameraPropertiesAtFrame">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="CameraPosition"
                  minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="CameraOrientation"
                  minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="CameraTilt" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="CameraAngleOfVision"
                  minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="frameNumber"
                  type="xsd:integer" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="CameraPosition">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="X" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Y" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Z" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="CameraOrientation" type="xsd:float"/>
  <xsd:element name="CameraTilt" type="xsd:float"/>
  <xsd:element name="CameraAngleOfVision" type="xsd:float"/>
  <xsd:element name="X" type="xsd:float"/>
  <xsd:element name="Y" type="xsd:float"/>
  <xsd:element name="Z" type="xsd:float"/>

  <xsd:element name="VisibleTheme">
    <xsd:complexType>
      <xsd:attribute name="theme" type="xsd:string" use="required"/>
      <xsd:attribute name="spatialExtentArea"
                  type="xsd:float" use="required"/>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```