LAND COVER CLASSIFICATIONS: AN OBSOLETE PARADIGM

G. Villa[a, *], N. Valcarcel[a], A.Arozarena[a], L. Garcia-Asensio[a], M. E. Caballero[a], A. Porcuna[b], E. Domenech[a], J.J. Peces[a]

[a]National Geographic Institute of Spain. Cartographic Production Department –
(gmvilla, nvalcarcel, aarozarena, lgasensio, mecaballero, edomenech, jjpeces)@fomento.es
[b]Tragsatec. Spain – apfm@tragsatec.es

Commission IV, WG IV/4

KEY WORDS: Land Cover, Data Models, Object Orientation, Parametric, Classification

ABSTRACT:

Land Cover information is a critical aspect of geographical and environmental information. The actual Paradigm, the "classification" concept used in Anderson's, Corine Land Cover and many other land cover databases up to now, has shown important limitations. Using Parametric Object Oriented Data Model technology it is possible to overcome these limitations and obtain much more useful Land Cover databases. Classification Land Cover databases fall in an error that is frequent in cartographers when they begin using GIS technology: using the database to describe a map, not the reality. Land Cover, as any other geographic information theme, can and should be modeled in UML using Parametric Object Oriented technology. It is imperative to change all this paradigm if we want geographic information to achieve the goals that we expect of it in the 21st century.

1. INTRODUCTION: EXISTING LAND COVER DATABASES

Everybody agrees that Land Cover information is a **critical aspect** of geographical and environmental information. In 1985, European Commission launched Corine Program and, within it, **Corine Land Cover** (CLC) Project, to produce **land cover** (LC from now on) databases for the whole Europe. For this project, a 3 level 44 classes hierarchical Nomenclature was developed. CLC´s **Minimum Mapping Unit** (MMU from now on) is **25 Ha**. Two versions of Corine database have been produced: 1990 (CLC90) and 2000 (CLC00), and another one CLC2006 is beginning production phase. MURBANDY and MOLAND Projects were initiated in 1998 with the aim to monitor the developments of urban areas and identify trends at the European scale. For this projects, a 4 level 97 classes Nomenclature, obtained as an extension of CLC Nomenclature, was developed. In other countries, other LC Nomenclatures have been developed and used. *E.g: 1976 USA's USGS Anderson´s Nomenclature, etc.*

2. CLASSIFICATION CONCEPTS

All LC databases mentioned above are based on the concept of "classification". We can say that classifications are the present "paradigm" for land cover information. In this paper, we will try to demonstrate that it is an obsolete paradigm. Classifications are based on the following basic principles:

a) An "a priori" "Nomenclature" or "Legend" with a finite number of "classes" is defined before beginning production of the database. Each "class" is a different "category" of LC.

b) In most Nomenclatures, classes are organized "hierarchically": one 1st level class is divided in several 2nd level classes, and so on.

c) In the Nomenclature description, each class is defined in a "class definition": a piece of text that fixes "rules" to decide if a certain polygon should be assigned to it.

d) Class definitions usually make reference to some "parameters": physical variables that can take different values in the real world. E.g: tree density. Most class definitions include "threshold values" for different parameters, in order to decide if a polygon belongs to a particular class:

E.g.: "Residential structures cover more than 80% of the total surface. More than 50% of the buildings have three or more stories"

In this sense, it is said that "a classification is a partition of the attribute space"

2.1 Minimum Mapping Unit (MMU). Consequences in the Nomenclature

In order to obtain a database of a reasonable and "usable" size, and also to limit the production budget, it is obliged to define a MMU. MMU is the minimum surface permitted for any polygon. The MMU forces the photointerpreter, in many cases, to draw polygons that contain areas with different LC. In spite of this, he must assign the polygon to one, and only one, class of the Nomenclature. This has two consequences in the Nomenclature:

a) Need for "flexible" class definitions: In some cases we decide to assign the polygon to the "dominant" class among those present in the polygon: normally the class that occupies the greatest percentage of the polygon's surface). For this reason, class definitions must be somewhat "flexible". E.g. CLC 3.1.1. Broad-leaved forest: Vegetated formation composed principally of trees, including shrub and bush under storeys, where broad-leaved species predominate (more than 75% of the formation)

b) Need for Mixed classes: In cases in which there is no clear dominance of any class, we cannot apply the solution a). So we must include in the Nomenclature some "mixed classes", to

[*] Corresponding author

which we can assign these mixed polygons. E.g: CLC 3.1.3. Mixed Forest: Vegetated formation composed principally of trees, including shrub and bush understoreys, where neither broad-leaved nor coniferous species predominate.

2.2 Production principles

a) The working area is divided in a set of “polygons”, each enclosing a (theoretically) “uniform” area.
b) Each polygon is assigned to one, and only one, thematic “class” of the Nomenclature.
c) In the alphanumerical database, each polygon is associated to one registry with only one field that contains the class label. E.g.: “1.1.2.4”

2.3 The “class explosion problem”

Suppose we need to build a database of “people”. If we want to consider 3 characteristics: gender, height and weight, each with this possible values:

Characteristic	Possible values	Number of values
gender	- man - woman	2
height	- tall - medium height - small	3
weight	- fat - medium weigh - thin	3

Table 1: Characteristics and values of a database of people

A “classification” of people would then have this 2*3*3=18 classes:

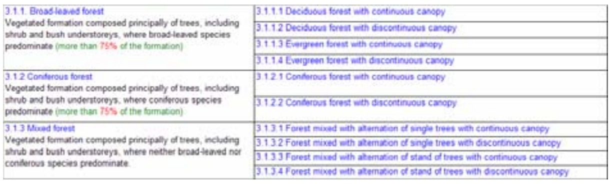
1. Men
    1.1. Tall men
        1.1.1. Tall and fat men
        1.1.2. Tall and medium-weight men
        1.1.3. Tall and thin men
    1.2. Medium height men
        1.2.1. Medium height and fat men
        1.2.2. Medium height and medium-weight men
        1.2.3. Medium height and thin men
    1.3. Small men
        1.3.1. Small and fat men
        1.3.2. Small and medium-weight men
        1.3.3. Small and thin men
2. Women
    2.1.Tall women
     …. etc…

But there are many other possible characteristics to be considered: nationality, age, study level, work, residence, eyes color, hair color, diseases, marital status, number of sons, hobbies, religion, etc, etc, etc…What would be the number of classes needed to store all this information?

250 * 100 * 4 * 100 * 250 * 5 * 4 * 20 * 4 * 10 * 20 * 10 =
= 8,000,000,000,000,000 = 8 * 10^{15} classes!!
Would these classes be useful in practice? Or would it be possible to implement them in an information system? not at all !! This is called by computer engineers the ‘class explosion’ problem. So this is clearly not the way to go…

3. PROBLEMS AND SHORTCOMINGS EXPERIENCED IN THE PRODUCTION AND USE OF HIERARCHICAL CLASSIFICATION LC DATABASES

Experience of many years in the production and use of CLC90, CLC2000, Murbandy and other LC databases, has shown several shortcomings and limitations of the Classification concept:
1) Proliferation of “threshold classes”, due to differences in the values of a particular parameter. E.g.: “1.1.1.1. Residential continuous dense urban fabric; 1.1.1.2. Residential continuous medium dense urban fabric”
2) Proliferation of “mixed classes”, due to the need to assign a class to polygons that contain several different LC types. E.g: “Mixed forest”, “Complex cultivation patterns”,…
3) “Class explosion”, due to the multiple “crossings” of several classification criteria.
E.g:



3.1.1. Broad-leaved forest Vegetated formation composed principally of trees, including shrub and bush understoreys, where broad-leaved species predominate (more than 75% of the formation)	3.1.1.1 Deciduous forest with continuous canopy
	3.1.1.2 Deciduous forest with discontinuous canopy
	3.1.1.3 Evergreen forest with continuous canopy
	3.1.1.4 Evergreen forest with discontinuous canopy
3.1.2 Coniferous forest Vegetated formation composed principally of trees, including shrub and bush understoreys, where coniferous species predominate (more than 75% of the formation)	3.1.2.1 Coniferous forest with continuous canopy
	3.1.2.2 Coniferous forest with discontinuous canopy
3.1.3 Mixed forest Vegetated formation composed principally of trees, including shrub and bush understoreys, where neither broad-leaved nor coniferous species predominate	3.1.3.1 Forest mixed with alternation of single trees with continuous canopy
	3.1.3.2 Forest mixed with alternation of single trees with discontinuous canopy
	3.1.3.3 Forest mixed with alternation of stand of trees with continuous canopy
	3.1.3.4 Forest mixed with alternation of stand of trees with discontinuous canopy

Figure 1: Class proliferation in the Moland Legend due to multiple criteria crossings

4) Mixed classes provide little information to user
- E.g.: “Complex cultivation patterns”,…
5) Mixed classes make the database “incoherent” to the hierarchical principle, as they are classes that contain other classes of the same level. E.g.: CLC “3.1.3. Mixed forest” contains “3.1.1. Broad-leaved forest” + “3.1.2. Coniferous forest”.
6) Mixed classes can easily lead to erroneous conclusions in the use of the database. E.g.: If one wants to know how many vineyards there are in a certain region, he will search for the class CLC 2.2.1: “Vineyards”. But there can also be plenty of vineyards “hidden” in other classes as: CLC 2.2.3: “Olive groves” (which includes the association of olives trees and vines); CLC 2.4.2: “Complex cultivation patterns”, etc…
7) Complexity of class definitions makes it difficult to produce, quality control, update and use the database:
- E.g.: CLC 1.1.2. “Discontinuous urban fabric: Most of the land is covered by structures. Buildings, roads and artificially surface areas are associated with vegetated areas and bare soil, which occupy discontinuous but significant surfaces. Between 10% and 80% of the land is covered by residential structures”.
This complexity makes these classes difficult to assign by the photointerpreter (increasing photointerpretation errors) and, what is worse, difficult to understand by non-expert users. To understand that a particular polygon is assigned to a certain class, it is necessary to have access to the definitions and “interpretation keys” of the Nomenclature, and keep them in mind all the time.
8) Complex definitions, in which several “classification rules” (parameters values intervals) apply simultaneously, increase the risk of incoherencies: overlapping definitions (some polygons can be assigned to more than one class) or under definitions (some polygons cannot be assigned to any of the classes).
- E.g.: Moland 1.1.1.1. Residential continuous dense urban fabric: “Most of the land is covered by structures and transport network. Buildings, roads and artificially surface areas cover

more than 80% of the total surface. Non-linear areas of vegetation and bare soil are exceptional. Residential structures cover more than 80% of the total surface. More than 50% of the buildings have three or more stories." For this reason, many of the HC nomenclatures are not rigorously correct from a theoretical point of view.

9) Important (and available) information lost: Information stored in the database is much less than information acquired by the photointerpreter. E.g.: In a Corine database the photointerpreter evaluates a certain polygon's broad leaved trees percentage as 85 %, and in consequence he labels it as 3.1.1("Broad-leaved forest")… But the user only receives the information that trees are "more than 30 %"

10) Important spatial variations in certain parameter values do not appear in the database. Polygons with very different percentage of buildings, or trees, cannot be differentiated in the information system if these variations do not "cross" the "threshold line".

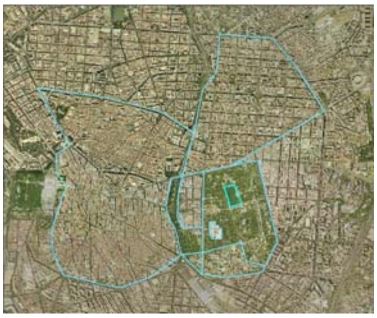


Figure 2: Different urban areas that wont be differentiated in a Land Cover classification"

E.g.: Urban areas with very different levels of building densities (as 10 % and 50 %) have to be assigned to the same Moland class (1.1.2.2. Residential discontinuous sparse urban fabric.)

11) When there are multiple LC types in a single polygon (because of the MMU size limitation) we must assign this polygon to a single class. This introduces "erroneous" information in the database and also in the statistics derived from it. E.g.: Vineyards in Galicia Region (Spain) have great social, cultural and economic importance, but they are present always in small parcels, so they are almost never "dominant" in a polygon, and almost "disappear" from Corine Land Cover database.

12) It is not possible to make calculation of derived variables and Indicators based on database's parameter values. There are many very important "indicators" that could be calculated from the values of the parameters appearing in class definitions (sometimes "crossing" them with exogenous information such as population, etc…). Eg:

- building density (m^3/m^2)in an area
- m^2 of building per person in an area
- Average height of buildings in a town
- % of impervious surface in an area
- % of trees in a forest
- m^2 of green areas per person in an area
- land take by transport infrastructures in a city
- etc.

What is more important, these parameters and indicators are the necessary input to very important "process models" such as, for example, climate change models.

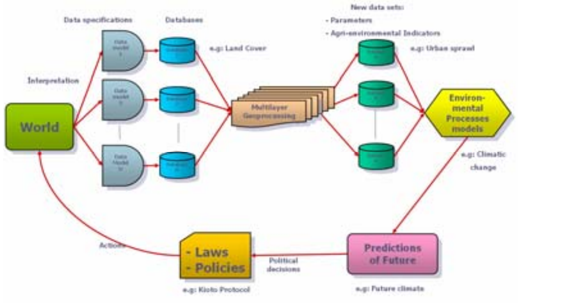


Figure 3: The use of biophysical parameters as input to environmental process models

HC databases do not allow calculating these parameters, because de actual values of the different parameters are not stored in them. E.g: You can not divide "Artificially surfaced areas are more than 80 %" by a surface.

13) Important temporal changes not detected: Temporal changes obtained from two dates of a HC LC database are not realistic. Important changes in certain parameters do not appear in the changes database, because: a) These changes do not "cross" the "definition rule" threshold. E.g.: If the building density of a polygon has increased from 11% to 79 % in the time between revisions of the database, this polygon is labeled as Corine´s 1.1.2. "Discontinuous urban fabric", in both databases, and so no change is registered. and/or: b)These changes are "hidden" in polygons assigned to dominant classes or to mixed classes.

14) Mosaicking different databases not possible: It is impossible to compare or "translate" two HC databases built with different Nomenclatures. E.g.: When we have a 3.1.1. "Broad leaved forest" polygon in Corine (defined as having more than 30 % of canopy closure), there is no way to know if it should be labeled as "forest" in a database with a different Nomenclature where "forests" are defined as "more than 50 % of canopy closure".

15) It is almost impossible to add external information from a specialized field to an existent HC database,. E.g.: Agriculture, Forestry…

16) It is not possible to make automatic generalizations of the database: If one wants to derive a database with smaller scale (by a generalization process), one has to aggregate polygons into bigger ones. In this case there is no way to automatically derive the class of the resultant polygon. E.g.: A polygon classified as Corine´s 1.1.1 ("Continuous urban fabric") aggregated with a polygon classified as 1.1.2 ("Discontinuous urban fabric") could result in an aggregated polygon that should be classified as 1.1.1 or 1.1.2. There is no way to know, without repeating the photointerpretation. This is a great obstacle if we are trying to implement INSPIRE´s Directive philosophy.

4. SOLUTION PROPOSED BY FAO´S LCCS

Land Cover Classification System (LLCS) has been developed by FAO and UNEP in an effort to solve the problems associated with existing HC LC databases. LCCS tries to establish a standardized comprehensive system that allows one to build, "a priori" classification "nomenclatures" and "legends" in a rigorous and coherent way. LCCS proposes to use a set of attributes called "classifiers", whose "threshold values" are used to systematically define a finite set of classes. An "ad-hoc" software package (LCCS Version 2) has been developed to permit the creation of Hierarchical Classification Nomenclatures with this philosophy. In respect to polygons that have various LC simultaneously, LCCS proposes the same solution as traditional HC LC databases: the creation of "mixed classes". This has the problems described in points 3), 4) and 5) of point 3 of this document. In our opinion, LCCS correctly exposes some of the problems of HC LC databases mentioned in point 3. of this document. In particular, it addresses problems 4) and 7). But the rest of the problems mentioned remain unsolved with LCCS approach:

- Plenty of information lost
- Mixed classes problems
- Spatial variations not registered
- Temporal variations not detected
- Parameters and Indicators calculations not possible
- Database mosaicking not possible
- Automatic generalization not possible
- Doesn't fulfill ISO 19100 standards
- Not object oriented
- Not UML
- Attributes values not stored in the database: it is still only a Classification Method!

From our point of view, LCCS makes the same mistake than traditional LC nomenclatures: it tries to classify each polygon in one and only one class, using the values of the "classifiers" to put sequentially the polygon to one side or the other of the "classification rules". It is important to note that the values that this classifiers have in a particular polygon are not stored in the database. Only the polygon's resultant class label is stored in the database.

5. PARAMETRIC OBJECT ORIENTED DATA MODEL SOLUTION

Most of the problems of LC databases cited above come from the attempt to "classify" the infinite variety of landscapes in a limited number of "closed" classes. The solution we propose to solve most of these problems is to use a Parametric Object Oriented Data Model. A data model is the description of what we are storing in a database and how. It is the "link" between reality and the Database. "Object Orientation" is the standard in Computer Science today. Parametric Object Oriented Data Models (POODM) are used extensively in every type of databases and Information Systems (airports, hospitals, production facilities…, and in some GIS databases. International standards as those from OGC and ISO are based on this technology. POODM allow an unprecedented flexibility and capability in de design and use of very complex Information Systems. POODM is the best technology to address the great complexity of Geographic information:

- Represent the "systemic" nature of the world.
- Define "objects" of different scales (Covers, Elements,…) and the relations between them.
- Assign attributes values to each one of these "objects" in a structured and organized way.

UML (Universal Modeling Language) lets us express, store, modify AND extend this structure easily and make it understandable by anybody. Land Cover has been traditionally modeled (Corine, Moland Andersons…) using classifications, legends, nomenclatures….They are, in our opinion, obsolete techniques. Up to now (to our knowledge) POODM have not been used for Land Cover Information

5.1 Basic principles

The use of Object Oriented Data Models for LC information is based on these basic principles:

1) The working area must be divided in a set of closed polygons, each one containing a surface that is as homogeneous as possible. Our aim is not to classify each polygon but to "describe" each one as well as possible. These descriptions are made associating "Land Covers", "Land Cover Elements" and "attributes" for them to each polygon.

2) "Land Covers" are thematic categories. They are defined with conceptual definitions of biophysical or socio-economic criteria (morphology, structure, relation with other land cover entities, etc….) "Land Cover Elements" are the objects found in the terrain that make up "Land Covers". Land cover elements (e.g.: buildings, trees, rock, sand, etc…) are the basic components of land cover. E.g.: trees are present in woodlands, urban fabric, agricultural areas,…But they are always Vegetation - Woody – Trees.

3) Each "Land Cover" (LC) and "Land Cover Element" (LCE) has its own attributes. Attributes are observable characteristics (biophysical or socio-economic) that describe LC or LCE in more detail. These attributes take different values in each "instance" (appearance of the LC or LCE). Some attributes are simple variables of the adequate type (e.g.: number of floors: integer). Other attributes are "Controlled lists" (e.g.: "vegetation distribution geometry"). Controlled lists are defined as "enumerations" in UML. Other attributes are more complex (e.g.: "vegetation state") and are represented as "UML classes" (white rectangles in the UML diagram).

4) Homogeneous polygons (at the scale of the database) have one Land Cover. When homogeneous surfaces have an area smaller than the Minimum Mapping Unit, the photointerpreter must draw a non-homogeneous polygon that encloses areas with different characteristics. In this case, the photointerpreter must measure (or estimate), and store in the database, the percentage of surface in which each "Land Cover" is present in the polygon. The sum of all percentages of each polygon must be 100 %.

5) For each "Land Cover" found in a polygon, the photointerpreter must study its "inner composition", and measure and store in the database:

- The average values for each of the attributes in this "Land Cover"
- The "Land Cover Elements" present in this "Land Cover", and the percentage of the surface that each occupies.
- The average values for each of the attributes in each "Land Cover Element". (e.g.: trunk diameter=0.40m; Number of floors = 4)

6) All the information of each polygon (percentage of surface of each LC present, average values of each parameter affecting each LC or LCE,…) is stored in an alphanumerical relational database (RDB). This RDB has been designed with two objectives in mind:

- Materialize as exactly as possible the Parametric Object Oriented Data Model represented in the UML diagram.
- Allow for unlimited future extensions of the model, making it as easy as possible to add more classes, parameters, conditions, etc…

If we look into the class definitions of any LC classification, Eg: Corine 1.1.2. Discontinuous urban fabric: "Most of the land is covered by structures. Buildings, roads and artificially surface areas are associated with vegetated areas and bare soil, which occupy discontinuous but significant surfaces. Between 10%

and 80% of the land is covered by residential structures. We can see they are expressed in terms of:
- Simple components: buildings, roads, bare soil,…
- Density thresholds: discontinuous
- Attributes: residential
- Percentages of polygon occupation: 10%, 80 %

The problem is that:
- Simple components are not explicit, not structured, incomplete, and not extensible
- Density thresholds are somewhat “arbitrary” and induce class explosion
- Attributes are not explicit, not structured, incomplete, not extensible, and actual values are not stored in the database
- Percentages of polygon occupation are not explicit, sometimes expressed vaguely, and the actual values are not stored in the database

On the other side, in the POODM solution:
- Simple components are explicit, structured, complete, and extensible
- Density thresholds are not neccesary
- Attributes are explicit, structured, complete, extensible, and actual values are stored in the database
- Percentages of polygon occupation are explicit, expressed rigorously, and the actual values are measured and stored in the database

From the “Conceptual Model” in an UML diagram an expert in Relational Databases can easily derive a “Physical Implementation” in any standard RDBMS. This RDB stores all the objects and attributes in a structured and robust way, and allows to interact with all these data through:
- SQL queries
- Different databases “crossing”
- Interaction with an specific “application”

The use of POODM for Land Cover Information has been developed, tested and is working in the Spanish SIOSE Project, which is in advanced production phase (finishing by end of 2008). A “physical implementation” (Relational Database and an Application to fill it) according with data model specifications have been developed and are in use in SIOSE production.

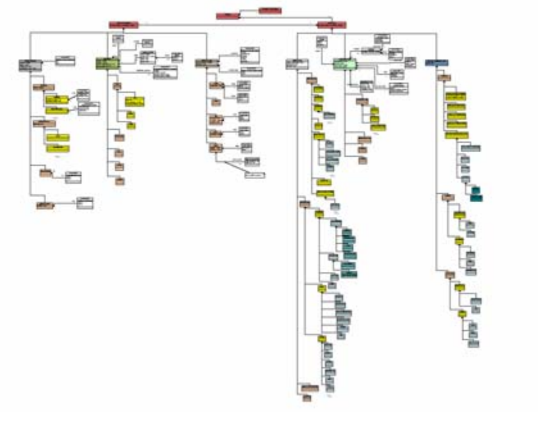


Figure 4: UML conceptual schema of a Parametric Object Oriented Land Cover Data Model

5.2 Examples

In order to make these concepts more clear, here are two examples of real landscape situations, and its codification in a Parametric Object Oriented Data Mode.

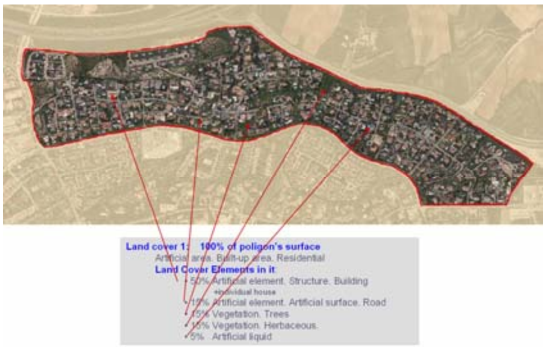

Land cover 1: 100% of poligon's surface
Artificial area. Built-up area. Residential
Land Cover Elements in it:
50% Artificial element. Structure. Building
+individual house
15% Artificial element. Artificial surface. Road
15% Vegetation. Trees
15% Vegetation. Herbaceous.
5% Artificial liquid

Figure 5: POODM description of a LC Polygon of the class Corine 1.1.2. “Discontinuous urban fabric”

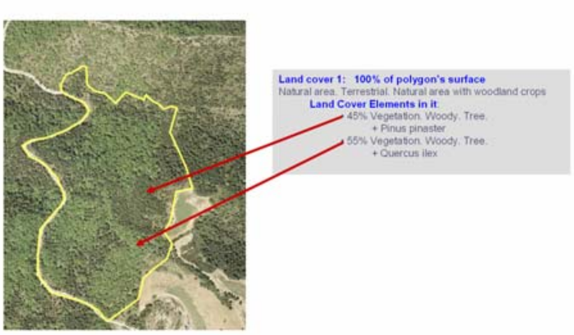

Land cover 1: 100% of polygon's surface
Natural area. Terrestrial. Natural area with woodland crops
Land Cover Elements in it:
45% Vegetation. Woody. Tree.
+ Pinus pinaster
55% Vegetation. Woody. Tree.
+ Quercus ilex

Figure 6: Polygon of the class Corine 3.1.3. “Mixed forest”

From a robust and well designed Parametric Object Oriented Land Cover Database, land cover classifications and nomenclatures can be derived by making appropriate SQL queries to the database.

5.3 Exploitation of the POODM database

5.3.1 Obtaining traditional thematic maps

From the POODM LC database, it is possible derive as many “views” as necessary. One view can be made assigning “classes” to certain combinations of covers and parameters values. Each “class” could then have a visual representation to make a “thematic map”. E.g: Standard Corine or Moland Nomenclatures could be “predefined views” of the POODM database.

5.4 Extensions of the Data Model

Once we have a database made with the POODM concept described above, it is very likely that some other person or institution has the need to input additional information of a specialized field. E.g.: agricultural, forestry, infrastructures, etc. In these cases, it is possible to “extend” the Data Model, using one or more of these techniques:

- Subdivide one UML class into multiple classes
- Add more UML classes

- Add values to controlled list text parameters
- Add more parameters to some classes.
- Add "conditions" that parameters should comply.
- Add types of "objects" (concepts) that can be associated to each polygon (at the same level as "cover")
- etc.

E.g: A forestry specialist could add some of these parameters about the trees in a polygon:

- height
- diameter of trunks
- health
- distance between each 2 trees
- species
- age
- etc…

5.5 Use of bio-physical parameters derived by Remote Sensing and/or field work

There are many bio-physical parameters as NDVI (Normalized Difference Vegetation Indexes), LAI (Leaf Area Index), FAPAR (Fraction of Absorbed Photo-synthetically Active Radiation), etc…that can be semi-automatically derived from multispectral images by remote sensing techniques, and/or by field work. The mean of this continuous value variables for each polygon´s area can then be input and stored in the POODM database as a parameter that qualifies each polygon. This approach improves greatly the "synergies" between Remote Sensing and GIS technologies in Land Use/Land Cover information.

6. SIOSE PROJECT IN SPAIN

Instituto Geográfico Nacional (IGN) Spain is coordinating SIOSE Project (Spanish Land Use and Land Cover Information System). It has the objective of building a LC information system at 1:25.000 nominal scale. This project is beginning the production phase. An Object Oriented Data Model (SIOSE data model) has been designed and implemented, in collaboration with more than 25 Spanish institutions interested in LC, organized in "Thematic Working Groups". From the UML diagram, a specialized team of IT engineers has developed 3 products:

- Data Model description document.
- Relational Database structure (in Access)
- A software package that allows to correctly input the information of each polygon in the DB (in Java).

There are several documents describing this project. Most interesting here is: "Modelo Conceptual del Proyecto SIOSE. Versión 1.11" All the products mentioned above have been built keeping in mind all the time the strict fulfilment of all related ISO and OGC International Standards as: Application Model Application Schema, ISO 19101 "Geographic Information – Reference Model", etc…One of the premises of SIOSE Data Model is that it must be possible to obtain automatically a standard Corine Nomenclature, and also a standard Moland Nomenclature from the POODM database. Corine or Moland maps (or databases) will be automatic "views" of SIOSE database. This assures backward compatibility and comparability with preexistent databases, as Corine 90, Corine 2000, Murbandy, Corine 2006, etc.

7. CLASSIFICATIONS "HIDDEN" IN LEGENDS OF OTHER GEOGRAPHIC INFORMATION THEMES

The ideas and problems presented here for the Land Cover case are applicable to the development of many other geographic information Themes Data Models:

Road networks:

- Main roads
- Secondary roads
- Local roads
- …

Hidrology:

- First order rivers
- Second order rivers
- …

Most of present GI databases are obsolete, because they were designed only to store information in paper maps. And, what is worse, many of new data models being built today are also obsolete, because they are "contaminated" ("poluted") by "map centric" "out of date" way of thinking. (E.g: the use of classifications, legends or nomenclatures)¡ Some of this obsolete "map centric" data models are even being proposed as international standards (ISO, INSPIRE,…). ISO, INSPIRE, and other international bodies should be careful not to adopt obsolete data models as standards. As INSPIRE is designed mainly to permit the interaction between existing databases, and as most existing G.I. databases are "map based", INSPIRE is in great danger of implementing obsolete data models (Corine Land Cover, LCCS,…)

8. CONCLUSIONS

In our opinion, Classification Land Cover databases fall in an error that is very frequent in cartographers when they begin using GIS technology: using the database to store attributes that describe a map, not the reality. It would be the same case as using a GIS database to store, for each city, the size of the letters that you should use to label it in the map (with a rule that relates population intervals and letter size) instead of storing the number of habitants of that city. Using the principles described briefly in this document, we hope that it will be possible to produce more "coherent" and much more "useful" LC databases. IGN Spain is proposing international institutions with responsibility in LC information (EEA, ISO, Inspire, GMES…) to adopt a similar philosophy for future LC databases: FAO, Corine, GMES, etc… and offers its collaboration for building a standard LC POODM. Geographic Information should not be modelled by classifications / legends / nomenclatures, as they imply a great decrease in the quantity and usefulness of information stored. FAO's LCCS is not an acceptable solution either. Land Cover, as any other G.I. theme can and should be modelled in UML using Parametric Object Oriented technology,…). It is imperative to change all this data models and databases if we want G.I. to achieve the goals that we expect of it in the 21st century. We need a new 'paradigm' in 21st century geographic information!

REFERENCES

CORINE Land Cover. Technical Guide. European Commission European Environment Agency.

SIOSE Project Technical Guide. IGN Spain