Final report

Urban Morphological Zones version F2v0

Definition and procedural steps

Prepared by:
Alejandro Simon, Jaume Fons, Roger Milego

28.12.2010

Project manager:
Birgit Georgi

European Topic Centre
Land Use and Spatial Information

European Environment Agency

Universitat Autònoma de Barcelona
Edifici C – Torre C5 4ª planta
08193 Bellaterra (Barcelona)
Spain, EU
Contact: etclusi@uab.cat

Table of Contents

List of Figures

1. INTRODUCTION

This document describes the methodology to create the Urban Morphological Zones (UMZ) and it is an update of the report published in 2007 (Milego, 2007)[1]. A UMZ describes the urban tissue of an area and is based on Land Cover classification, in this case, Corine Land Cover.

2. UMZ DEFINITION

An Urban Morphological Zone can be defined as "A set of urban areas laying less than 200m apart". Those urban areas are defined from land cover classes contributing to the urban tissue and function.

The Corine Land Cover classes (Figure 1) used to build the Urban Morphological Zone dataset are the following ones:

- **Core Classes**
  - 111 – Continuous urban fabric
  - 112 – Discontinuous urban fabric
  - 121 – Industrial or commercial units
  - 141 – Green urban areas
- **Enlarged core classes**: 123 (Port areas), 124 (Airports) and 142 (Sport and leisure facilities), are also considered if they are neighbours to the core classes or to one of them touching the core classes.
- 122 (Road and rail networks) and 511 (Water courses), when neighbours to the **enlarged core classes**, cut by 300m buffer.
- Forests & scrub (311,312,313,322,323,324), when they are **completely within** the **core classes**.

Figure 1. Land Cover (Barcelona Area)

Figure 2. Legend

[1] http://www.eea.europa.eu/data-and-maps/data/urban-morphological-zones-2000-umz2000-f1v0/

3. PROCEDURAL STEPS

3.1 Overview

In order to have a consistent procedure to easily reproduce the final UMZ vector layer, a model has been developed in ArcGIS. The model has 60 steps and the complete workflow is provided in Annex I. Next sections provide a description of the most relevant steps. It should be noted that some of the steps in the workflow are designed to overcome computer's capacity of processing due to the large size of features.

3.2 CLC Reclassification

The starting point is the **reclassification** of Corine Land Cover raster data (*Input data: CLC in raster format*).

CLC is reclassified according to the definition described above to obtain the following set of land cover classes:

- *Core classes* (Figure 3)
- *Ports, Airports and sport facilities* (Figure 4)
- *Transportation and rivers* (Figure 5)
- *Forests & scrub* (Figure 6)

Each set of land cover classes in raster format is **vectorised** in order to obtain the corresponding vector layer. The reason to proceed in this way is that CLC06 is available in a seamless vector format version but not previous years, 1990 – 2000. Then, keeping the same methodology for the different versions allows the temporal analysis of the urban areas.

The steps required are described below:

- *Core classes.* Directly vectorised. ***Output: Core Classes feature class*** *(Core_Classes.shp)*
- *Ports, Airports and sport facilities*
  Starting from the Core classes described above, the first step consists on adding those ports, airports and sport facilities which are neighbours to the core classes. This step is done iteratively until all neighbouring polygons are selected. ***Output: Enlarge Core Classes*** *(Enlarge10.shp)*[2]
- *Transportation and rivers.* Directly vectorised. ***Output: Linear feature class*** (Linear.shp)
- Forests & scrub. Directly vectorised. ***Output: Forests & scrub feature class*** *(Forestscrub.shp)*

[2] Due to the expertise applied, 10 iterative steps are more than enough to include all the enlarge objects to the core classes

Figure 3. Core classes

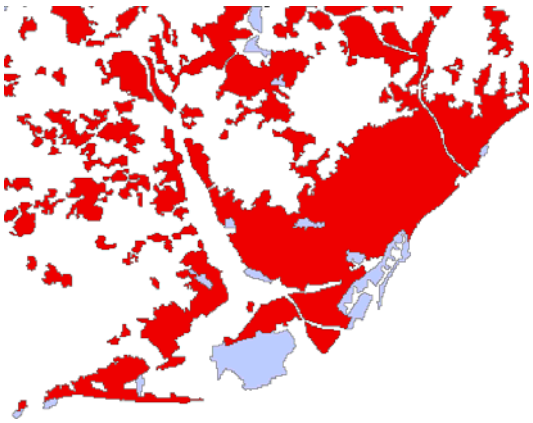


Figure 4. Enlarge Core classes

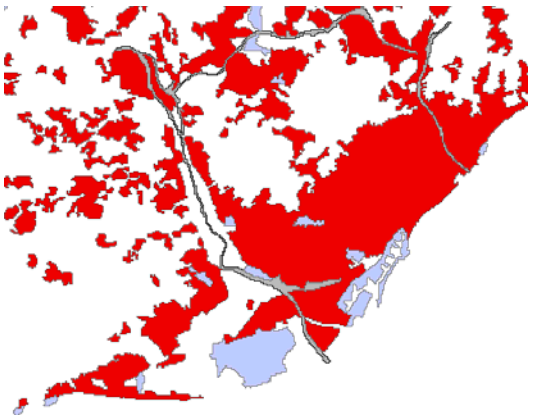


Figure 5. Enlarge Core Classes + Linear classes

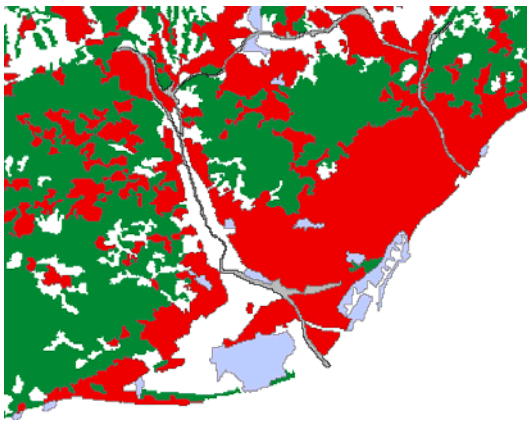


Figure 6. Enlarge Core Classes + Linear classes Forest & Scrub classes

3.3 Finding Gaps in the Core Classes

For the final delineation of the UMZ, it is needed to identify the gaps inside the UMZs. These can be forest areas and therefore they should be included (Figure 7). **Output: Gaps in the Core Classes** (Gaps_Core_Classes.shp)

The methodology to find the gaps only requires 4 steps:

- Core classes feature class (Core_Classes.shp) is converted from polygon to lines --> *Core_Classes_Lines.shp*
- Core_Classes_Lines.shp is converted again to a polygons feature class --> *core_classes_POLYGONtotal.shp*
- Finally, the core_classes_POLYGONtotal feature class is erased with the Core_Classes feature class --> *Gaps_Core_Classes.shp*

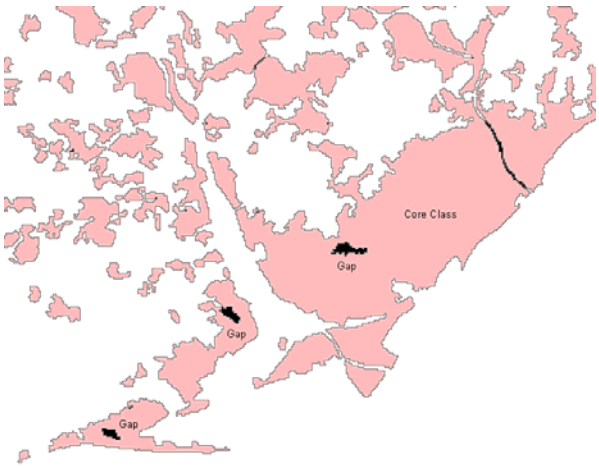


Figure 7. Gaps in the Core classes

3.4 Build-up UMZ

This last step is a list of processes to include all the features (previously explaned) that will participate in a UMZ

3.4.1 Addition of Forest to the Core Classes

All the forest geometries which are inside the UMZ will be added. Therefore all the forest contained by the “Gaps_core_classes” will belong to the UMZ (Figure 8). ***Output: Forest in Gaps of the Core Classes (***Forest_selected.shp***)***

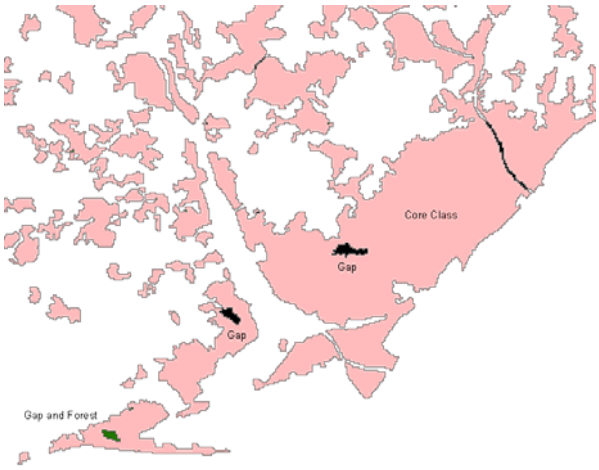


Figure 8. Forest classes in the Gaps of the Core classes

3.4.2 Addition of roads, railways and rivers to the enlarge core classes

Those roads, railway and river polygons neighbours to the enlarged core classes are selected, then clipped by a 300m buffer around the enlarged core classes, and finally merged to them. Since in this step mainly linear features are added, buffering is required in order to capture the actual area affected by the infrastructure. Clipping at 300 m has been found as the optimal distance after several trials in different European cities with predominant linear features (e.g. big rivers). The clip prevents having all UMZ merged together via linear features such as motorways or wide rivers (Figure 9 - Figure 12)

Output: Enlarge Core Classes + Linear classes less than 300 m away *(Enlarge_CoreClasses_plus_lines.shp)*

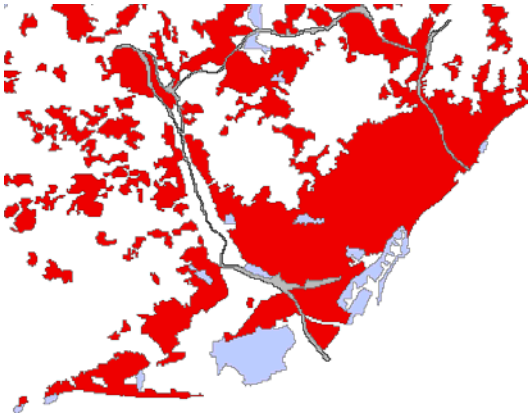


Figure 9. Enlarge Core Classes + Linear classes

Figure 10. Linear selected (Selection by neighbourhood Linear – Enlarge Core Classes)

Figure 11. Linear Selected cut by 300m buffer

Figure 12. Enlarge Core Classes + additional of linear features (roads, railways and rivers)

3.4.3 Final Built-up UMZ

The outputs from the steps 3.4.1 and 3.4.2 are merged in one feature class, class → ***Pre UMZ feature class*** *(PreUMZ.shp).* With this step a pre-UMZ layer is obtained, which is rasterised (100m resolution) to undertake the so-called UMZ built-up, explained in the next paragraph and Figure 13 - Figure 16.

Once all the land cover classes to become part of the UMZ have been merged, it is time to apply the definition that tells us a UMZ is a set of urban areas laying less than 200m apart. The best way to put in practice this definition is through an expansion and shrinking process of the rasterised pre-UMZ by 1 pixel (i.e. 100m). Doing so, in the expansion process some areas will be in touch and will remain in touch after the shrinking process as well, while areas not touching after the expansion process will come back to their original "place" after the shrinking process. In the end, urban areas laying less than 200 m apart will be connected without any modification of their original shapes (except for the connection zones). ***Output: UMZ Final feature class*** *(UMZ_Final.shp)*

Figure 13. Pre-UMZ

Figure 14. Expanded UMZ

Figure 15. Shrinked UMZ

Figure 16. UMZ 2006 v13

After this process, the final raster is vectorised. Two new fields are calculated

- "IDXX" = [FID] + 1
- "Area_ha" = Area in hectares

All the polygons below 25 ha (CLC limit) are removed, obtaining the UMZ2000 v.13 dataset (Figure 17). ***Output: UMZ Final feature class + New attributes*** (**Error! Reference source not found.**)

FID	Shape	Area_ha	IDXX
0	Polygon	32	1
1	Polygon	70	2
2	Polygon	99	3
3	Polygon	46	4
4	Polygon	87	5
5	Polygon	56	6
6	Polygon	205	7
7	Polygon	62	8
8	Polygon	135	9

Figure 17. Attributes of the UMZ 2006

3.4.4 Cleaning UMZ_final.shp

Finally optional steps can be made manually:

- The field IDXX can change the name to another more comprehensive. For example: UMZ2000_ID
- The name “UMZ_Final.shp” can be changed to a more comprehensive one. For example. UMZ2000.shp
- The shapefile can be exported to a geodatabase

4. OTHER UMZ GEOMETRIES

4.1 Changes 1990 – 2000

UMZ Changes (Figure 18) have been calculated as well, by unioning both UMZ90 and UMZ00 and looking for those polygons existing only in one year (i.e. 1990 or 2000). Most changes are Positive changes, understood as areas of urban sprawl (i.e. new UMZ areas between 1990 and 2000), while negative changes describe the reduction of a certain UMZ between 1990 and 2000 (warning: some negative changes might be due to different interpretations between 1990-2000). Only countries with CLC in both year have changes in UMZ

- Records with only UMZ 90 → Change “-”. This means Reduction of Urban Areas
- Records with only UMZ 00 → Change “+”. This means Urban Sprawl

Steps to follow:

1. Create a file geodatabase to keep the geometries
2. Union process between UMZ 1990 and UMZ 2000 feature class. *Output: UMZ90_00_Full*
3. Select ("FID_UMZ1990" = -1 OR "FID_UMZ2000" = -1)
4. Export the selection. *Output: UMZ90_00_FULL_select.shp*
5. Selection of the real changes. Erasing
    a. Select the countries with **NO** CLC in both years (90-00 and 00-06) from a NUTS0 Layer (as accurate as possible)
    b. Erase process between *UMZ90_00_FULL_select.shp* and the previous selection → Output: *Changes_90_00.shp*
    c. Manually work to remove the changes belonging to the selected countries. Small feature classes are left in these countries due to not perfect overlay among layers (NUTS0 and CLC) mainly in the coastal areas
6. Add and Calculate the Field “Area_ha”
7. Add a new field name “Change” type = Text with length = 1 to Change_*90_00.shp*
8. Delete unnecessary fields
9. Select “UMZ90_ID” = 0 and Calculate Value for “Change” = + . This means Urban Sprawl
10. Select “UMZ00_ID” = - 1 and Calculate Value for “Change” = - . This means Reduction of Urban Areas

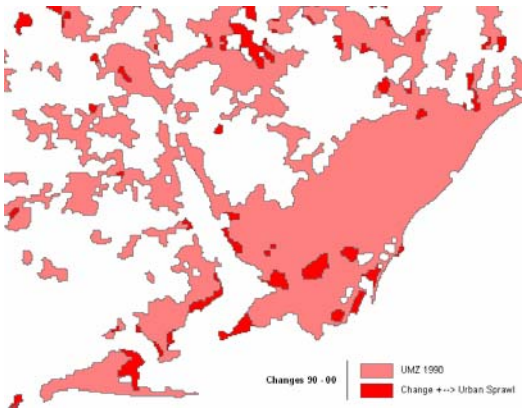


Figure 18. UMZ Changes 1990 - 2000

4.2 Changes 2000 – 2006

Same methodology that in the step 4.1

5. RELATIONSHIPS BETWEEN UMZ90, 00, 06

The methodology is described in the Annex II. These intermediate related tables allows linking UMZ 1990, 2000 and 2006 through the IDs

6. ATTRIBUTES

6.1 Population Figures

Intersecting the UMZ polygons with JRC's 2001 Population density grid[3] (Gallego, J., 2010) we can obtain a population figure within every single UMZ (Figure 19).
In the case no population figures are available; the value -9999 has been recorded instead.

The tool used is Zonal Statistics of the UMZ vector layer above the Population GRID v5.0 to calculate the basic statistics per UMZ (Fields created: count, area, min, max, range, sum, mean, and std fields, Majority, minority, median, and variety (This 4 only if the input contain integer data (Raster or Vector))

Steps to follow:

- Zonal Statistics between the UMZ 2000 vector layer and the Population Grid
- Identified the UMZ90_ID in the output table as Value_
- Sum field keeps the population for each UMZ. Notice that the real value is the number divided by 100. For further information see the Downscaling methodology for the population GRID[4].

[3] http://www.eea.europa.eu/data-and-maps/data/population-density-disaggregated-with-corine-land-cover-2000-2

[4] http://dataservice.eea.europa.eu/dataservice/metadetails.asp?id=1110

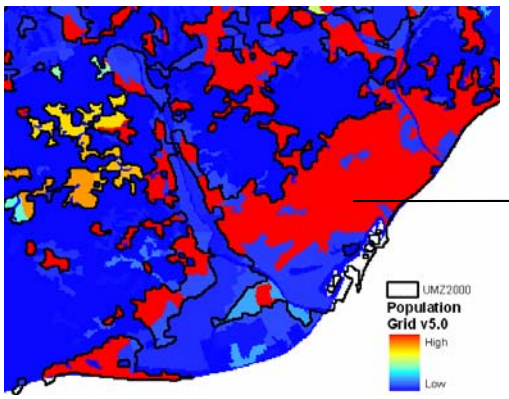


Figure 19. Population distribution in the UMZ

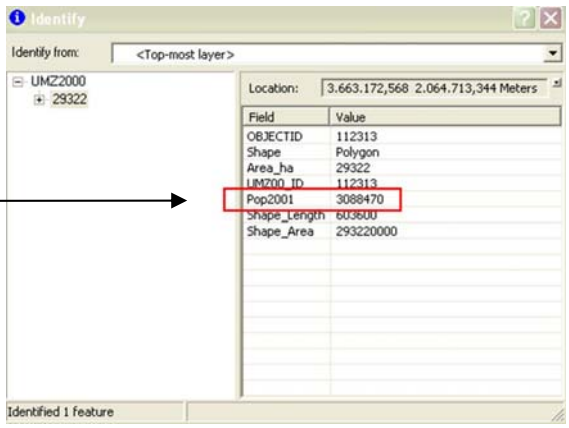


Figure 20. Population as an attribute

6.2 Commune codes

Using the EuroBoundaryMap Jan'06 dataset (other any other source with Commune boundaries could be used), commune attributes can be added to each UMZ. A relationship class between both datasets has been created in ArcGIS in the way that once you identify a UMZ you get the list and attributes of all communes or parts of commune being part of it (**Error! Reference source not found.**, **Error! Reference source not found.**). The whole process can be checked in Annex III.

Figure 21. UMZ 2006 Barcelona and the Communes 2006

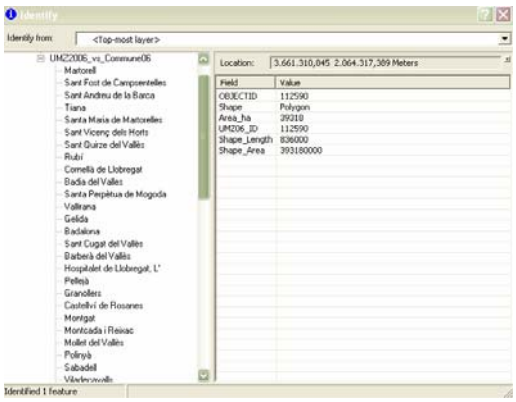


Figure 22. Attributes of the UMZ 2006

This step is redone with each UMZ feature class (1990, 2000, 2006)

6.3 Available attributes

The attributes available for each UMZ dataset are the following ones:

- UMZ90
  - UMZ90_ID: Unique identifier of each UMZ90
  - Area_ha: Area in hectares

- UMZ2000
  - UMZ00_ID: Unique identifier of each UMZ00
  - POP01: Population number of each UMZ00
  - Area_ha: Area in hectares

- UMZ2006
  - UMZ06_ID: Unique identifier of each UMZ00
  - Area_ha: Area in hectares

- UMZ CHANGES 1990-2000
  - UMZ90_ID: Unique identifier of each UMZ90
  - UMZ00_ID: Unique identifier of each UMZ00
  - CHANGES: + (Urban sprawl between 1990 and 2000) or – (UMZ reduction)
  - Area_ha: Area in hectares

- UMZ CHANGES 2000-2006
  - UMZ00_ID: Unique identifier of each UMZ90
  - UMZ06_ID: Unique identifier of each UMZ00
  - CHANGES: + (Urban sprawl between 2000 and 2006) or – (UMZ reduction)
  - Area_ha: Area in hectares

- UMZ90 vs Commune06
  - UMZ90_ID: Unique identifier of each UMZ90
  - Euroboundary attributes: Commune_ID, Commune name, Country Code and NUTS Code

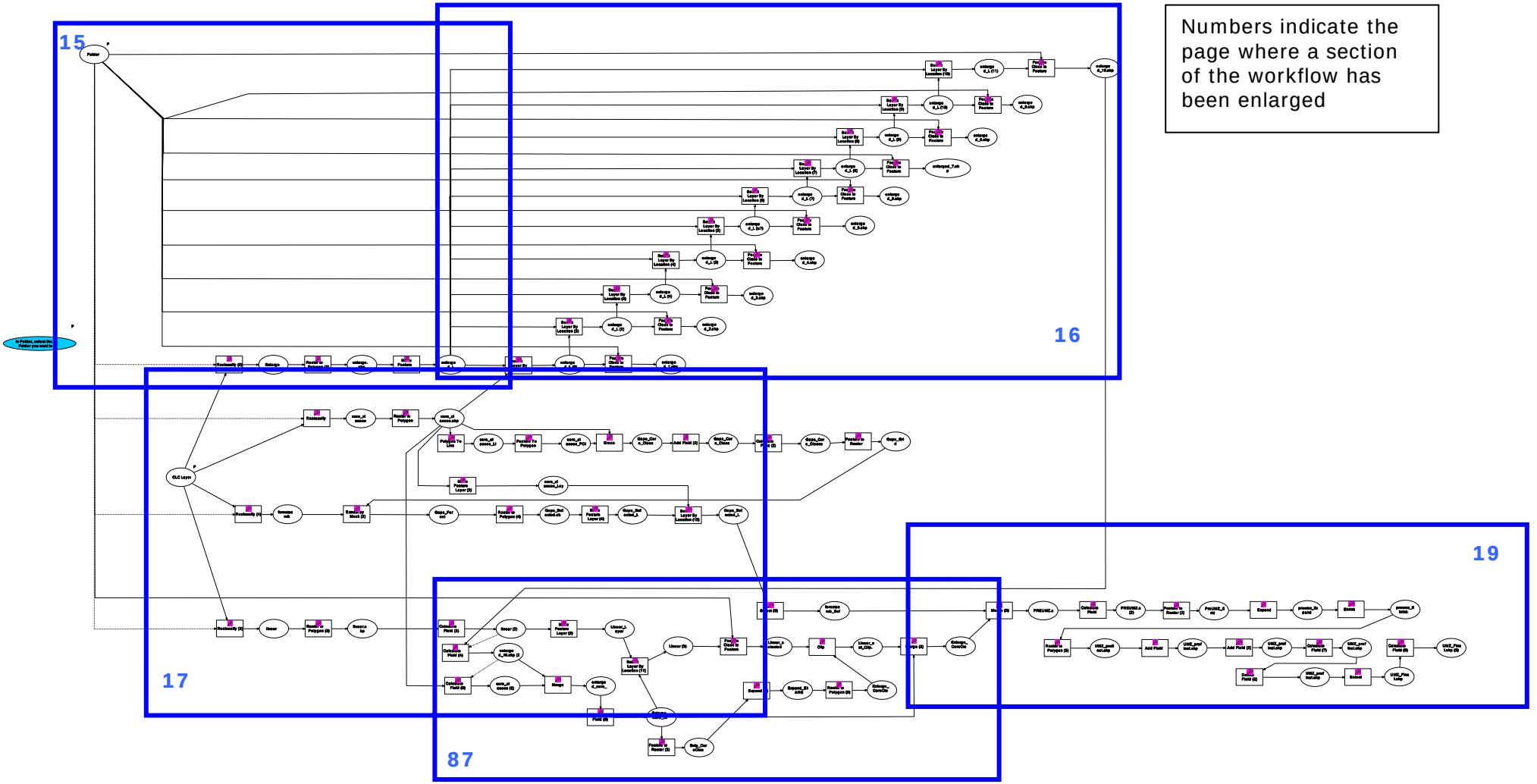
- UMZ00 vs Commune06
  - UMZ00_ID: Unique identifier of each UMZ00
  - Euroboundary attributes: Commune_ID, Commune name, Country Code and NUTS Code

- UMZ06 vs Commune06
  - UMZ06_ID: Unique identifier of each UMZ06
  - Euroboundary attributes: Commune_ID, Commune name, Country Code and NUTS Code

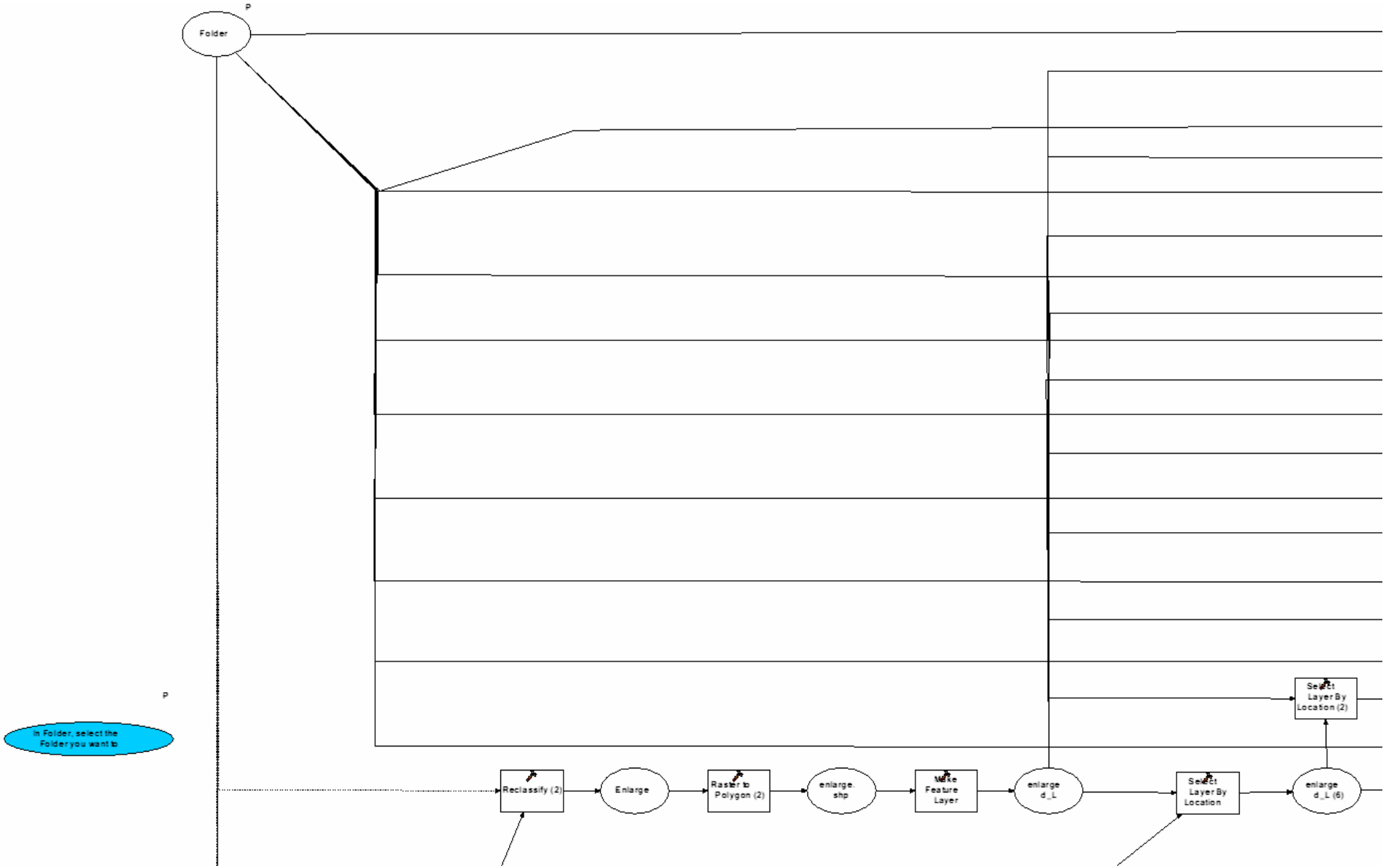
7. REFERENCES

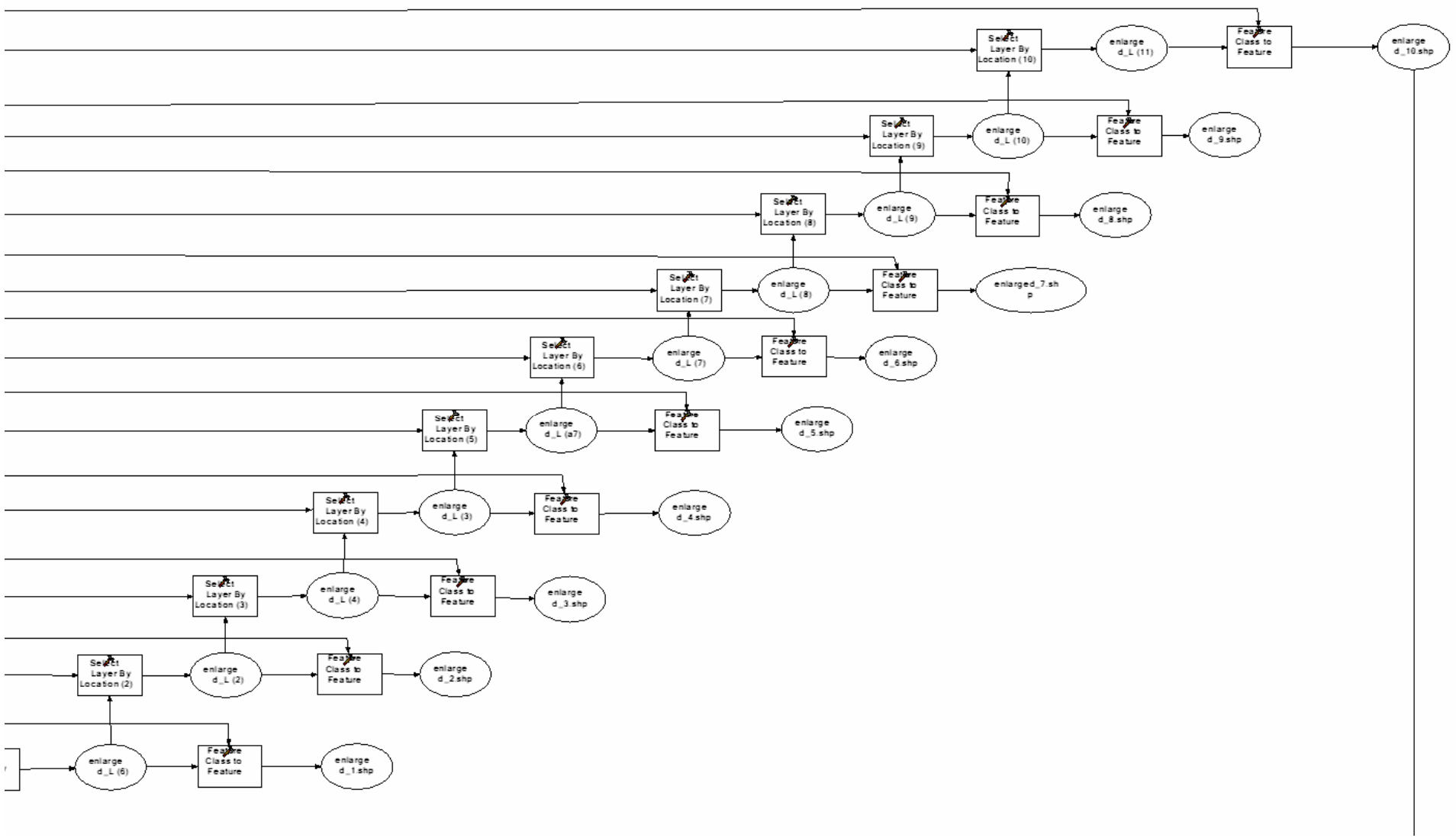
The general methodology description is described in Milego, 2007 (Definition and procedural steps. UMZ 2000). This paper can be downloaded here: http://dataservice.eea.europa.eu/download.asp?id=17335&filetype=.pdf

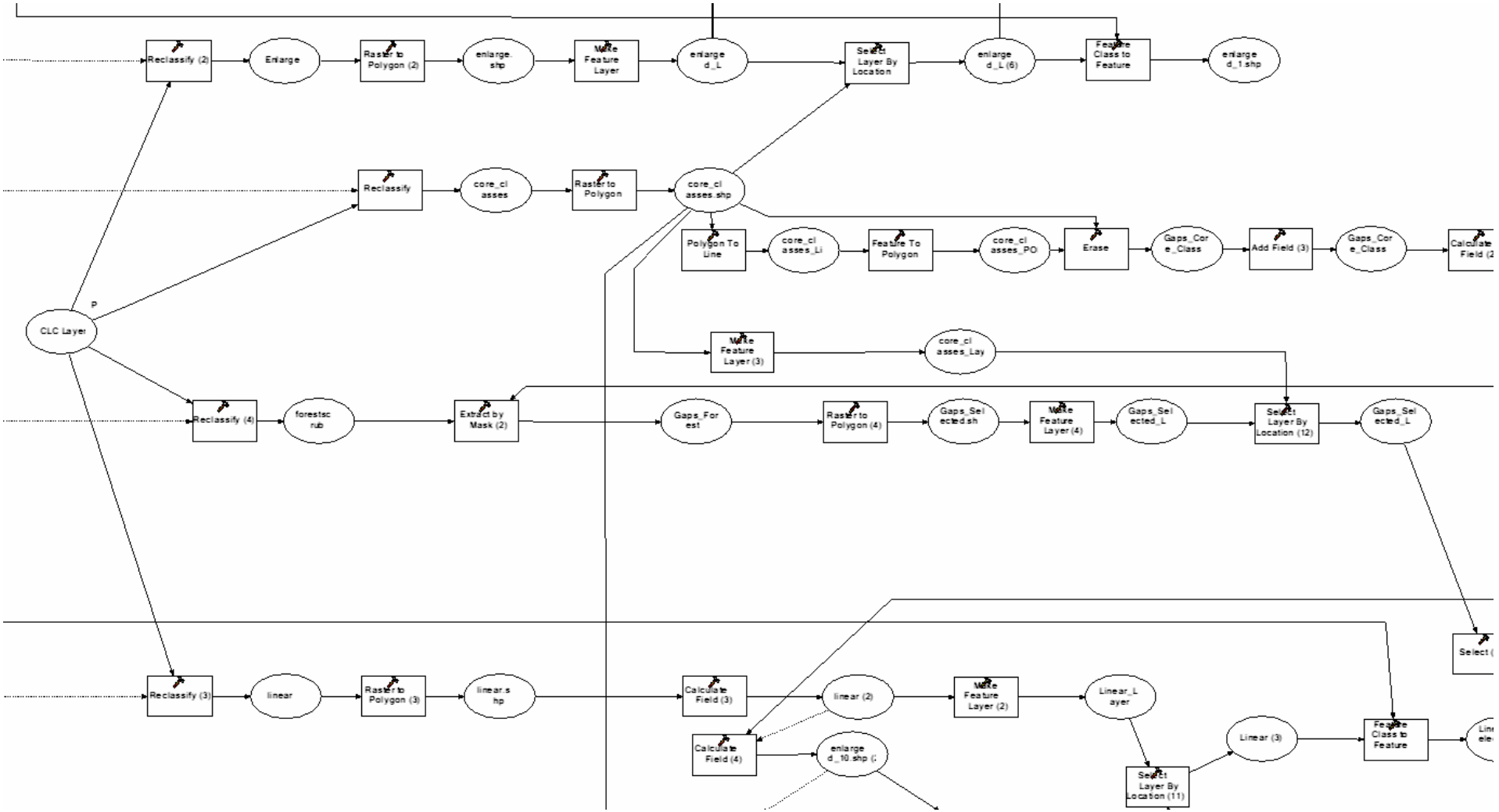
ANNEX I. WORKFLOW TO OBTAIN THE UMZ VECTOR LAYER.

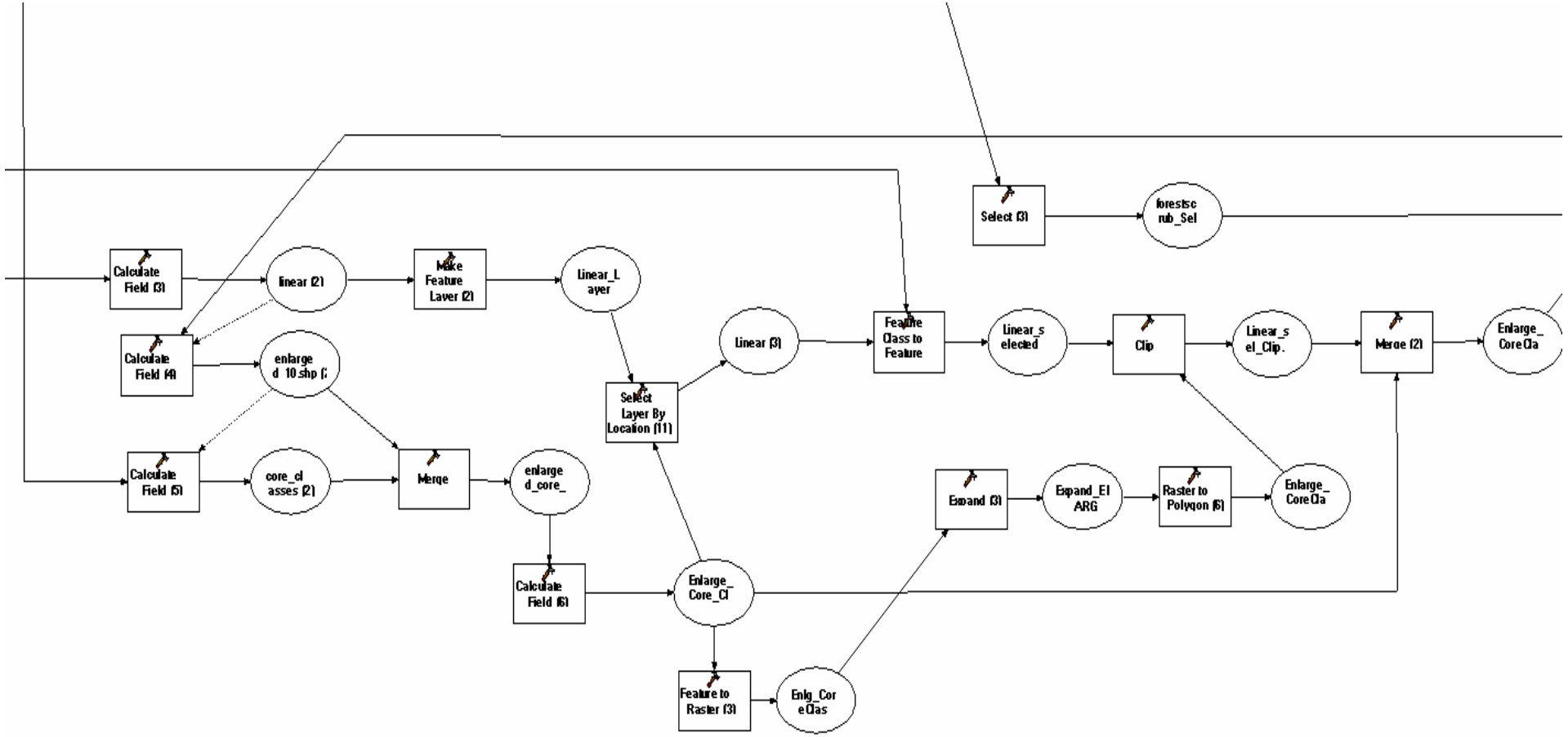


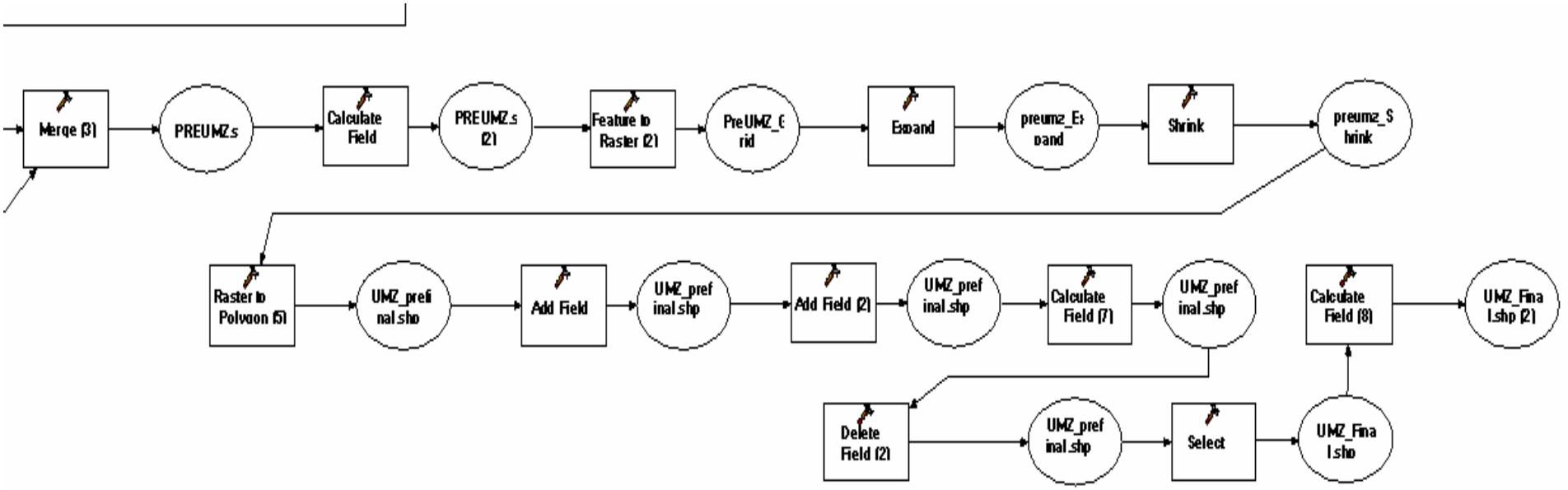
Numbers indicate the page where a section of the workflow has been enlarged


P
Folder
P
In Folder, select the Folder you want to
Reclassify (2)
Enlarge
Raster to Polygon (2)
enlarge. shp
Make Feature Layer
enlarge d_L
Select Layer By Location
enlarge d_L (6)
Select Layer By Location (2)


Select Layer By Location (10)
enlarge d_L (11)
Feature Class to Feature
enlarge d_10.shp
Select Layer By Location (9)
enlarge d_L (10)
Feature Class to Feature
enlarge d_9.shp
Select Layer By Location (8)
enlarge d_L (9)
Feature Class to Feature
enlarge d_8.shp
Select Layer By Location (7)
enlarge d_L (8)
Feature Class to Feature
enlarged_7.shp
Select Layer By Location (6)
enlarge d_L (7)
Feature Class to Feature
enlarge d_6.shp
Select Layer By Location (5)
enlarge d_L (a7)
Feature Class to Feature
enlarge d_5.shp
Select Layer By Location (4)
enlarge d_L (3)
Feature Class to Feature
enlarge d_4.shp
Select Layer By Location (3)
enlarge d_L (4)
Feature Class to Feature
enlarge d_3.shp
Select Layer By Location (2)
enlarge d_L (2)
Feature Class to Feature
enlarge d_2.shp
enlarge d_L (6)
Feature Class to Feature
enlarge d_1.shp


CLC Layer
Reclassify (2)
Enlarge
Raster to Polygon (2)
enlarge.shp
Make Feature Layer
enlarged_L
Select Layer By Location
enlarged_L (6)
Feature Class to Feature
enlarged_1.shp
Reclassify
core_classes
Raster to Polygon
core_classes.shp
Polygon To Line
core_classes_Li
Feature To Polygon
core_classes_PO
Erase
Gaps_Core_Class
Add Field (3)
Gaps_Core_Class
Make Feature Layer (3)
core_classes_Lay
Reclassify (4)
forestscrub
Extract by Mask (2)
Gaps_Forest
Raster to Polygon (4)
Gaps_Selected.sh
Make Feature Layer (4)
Gaps_Selected_L
Select Layer By Location (12)
Gaps_Selected_L
Reclassify (3)
linear
Raster to Polygon (3)
linear.shp
Calculate Field (3)
linear (2)
Make Feature Layer (2)
Linear_Layer
Calculate Field (4)
Select Layer By Location (11)
Linear (3)
Feature Class to Feature


Calculate Field (3)
linear (2)
Make Feature Layer (2)
Linear_L ayer
Select (3)
forestsc rub_Sel
Calculate Field (4)
enlarge d_10.shp (;
Linear (3)
Feature Class to Feature
Linear_s elected
Clip
Linear_s el_Clip.
Merge (2)
Enlarge_ CoreCla
Select Layer By Location (11)
Calculate Field (5)
core_cl asses (2)
Merge
enlarge d_core_
Expand (3)
Expand_El ARG
Raster to Polygon (6)
Enlarge_ CoreCla
Calculate Field (6)
Enlarge_ Core_Cl
Feature to Raster (3)
Enlg_Cor eClas


Merge (3)
PREUMZ.s
Calculate Field
PREUMZ.s (2)
Feature to Raster (2)
PreUMZ_G rid
Expand
preumz_Ex pand
Shrink
preumz_S hrink
Raster to Polygon (5)
UMZ_prelimi nal.shp
Add Field
UMZ_pref inal.shp
Add Field (2)
UMZ_pref inal.shp
Calculate Field (7)
UMZ_pref inal.shp
Calculate Field (8)
UMZ_Fina l.shp (2)
Delete Field (2)
UMZ_pref inal.shp
Select
UMZ_Fina l.shp

ANNEX II. HOW TO CREATE THE RELATED TABLES BETWEEN UMZ 1990, 2000, 2006

TABLE OF CONTENTS

1. INTRODUCTION

This document describes how to create the related table between the following feature classes: UMZ90, UMZ00, UMZ06. The output tables will be:

- UMZ1990_UMZ2000_relationship
- UMZ2000_UMZ2006_relationship

2. METHODOLOGY

2.1 Input: Feature Class

Shapefiles and Geodatabase are accepted to work with.

- UMZ1990. Last version with 98959 features
- UMZ2000. Last version with 112168 features
- UMZ2006. Last version with 81603 features

Shapefile

Create a specific folder to keep the UMZ feature classes**.**

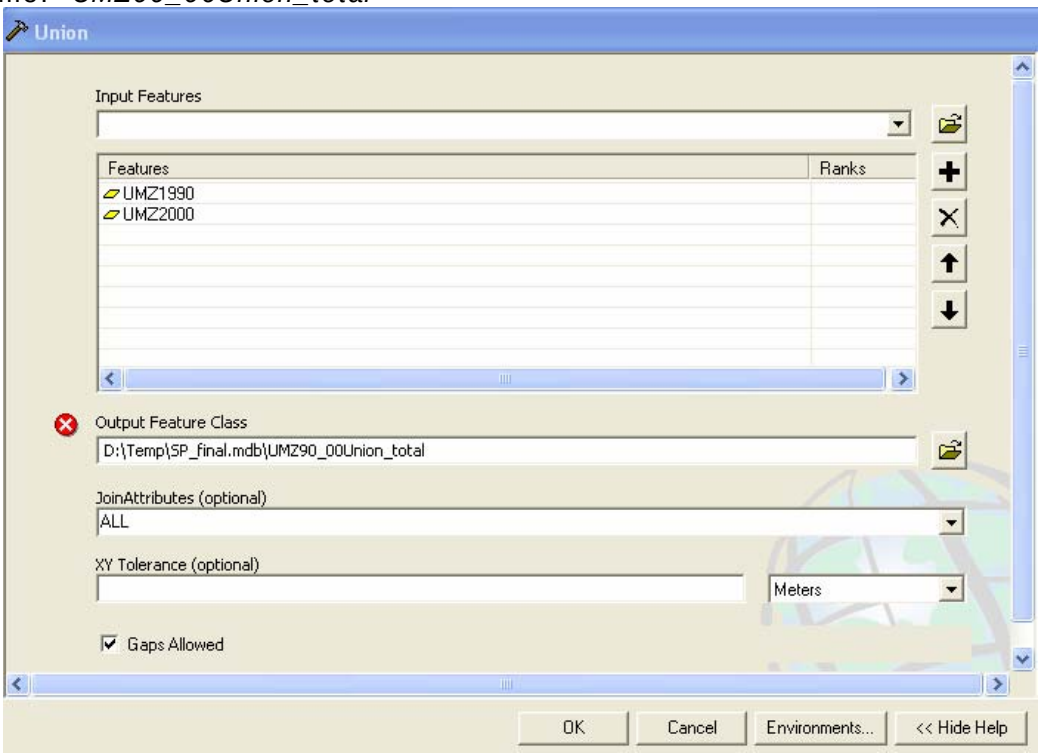
Geodatabase

Create a database. It can be done in the ArcCatalog. It has to include all the input feature classes.

Name: *"UMZs_join.mdb"*

2.2 ArcGIS

- Load UMZ1990, UMZ2000, UMZ2006
- Create a new feature class: Union UMZ1990 – UMZ2000
  Name: "*UMZ90_00Union_total"*
- Create a new feature class: Union UMZ2000 – UMZ2006
  Name: "*UMZ00_06Union_total"*



2.3 MS ACCESS

2.3.1 Inputs

- *UMZs_Relations_BASE.mdb.* Last location in:
  O:\2009_subvention\2_7_2_Urban\Tasks\UMZ\UMZ_table_relation
  This database will be used as template to create the final tables.
- *UMZ90_00Union_total* feature class
- *UMZ00_06Union_total.*

Notice two issues:

✓ This database and the respective queries have been developed to create the relationship table between UMZ1990 – UMZ2000 and UMZ2000 – UMZ2006. To create new tables these queries have to re-created or updated.

✓ Changing the names of the tables or the queries, makes the DB not to work

2.3.2 Methodology

- Open the database: "*UMZs_Relations_BASE.mdb*"

- Save as “UMZs_Relations_1990_2000_2006.mdb”
- Import the next tables:
    - UMZ90_00Union_total
      Name: “*tbl_UMZ90_00Union_total”.*
    - UMZ00_06Union_total
      Name: “*tbl_UMZ00_06Union_total”*
    - UMZ1990
      Name: “*tbl_UMZ1990”.* Required fields OBJECTID, and UMZ90_ID (at least).
    - UMZ2000
      Name: “*tbl_UMZ2000”.* Required fields OBJECTID, and UMZ00_ID (at least).
    - UMZ2006
      Name: *“tbl_UMZ2006”.* Required fields OBJECTID, and UMZ06_ID (at least).
    - Delete unnecessary fields to save space.

  These tables come from shapefiles or geotadabases. ArcCatalog can be used to export, or access, or any other tool. The field names should be respected to those mentioned above
- Run the queries according to the respective order. The description of how to use each of them is the picture below or in the description inside the “*UMZs_Relations_BASE.mdb”*

Two are the tables that related the IDs of the UMZ 1990, 2000, 2006.

- Relation IDs UMZ1990 - UMZ2000
- Relation IDs UMZ2000 - UMZ2006

Example of the methodology with UMZ1990 – UMZ2000.

2.3.3 Steps

GIS Software

In a GIS software, union UMZ1990 and UMZ2000 --> UMZ90_00Union_Total. This geometry contains all the attributes from both layers.

DB Software

In a DB software serveral queries are developed to obtain the final result.

1. Query to create a table named "tbl_UMZ90_00Union_total"" with the records with ID90 and ID00 > 0 from "UMZ90_00Union_Total" table --> "qry_01_Union_with_ID90_ID00"

2. Query to create an identical table (only the structure of the table) to "tbl_UMZunion_with_ID90_ID00" with the name "UMZ1990_UMZ2000_Relation_all". It will be used to add new records which don´t fit the sentence: ID90 > 0 AND ID00 > 0 --> "qry_02_create_final_UMZ1990_UMZ2000_Relation_all"

3. Query to identify the records (feature classes) from UMZ1990 which are not in tbl_UMZunion_with_ID90_ID00. These records only exist in UMZ1990 and not in UMZ2000 --> "qry_UMZ1990_not_coincident_tbl_UMZunion_with_ID90_ID00"

4. Query to identify the records (feature classes) from UMZ2000 which are not in tbl_UMZunion_with_ID90_ID00. These records only exist in UMZ2000 and not in UMZ1990 --> "qry_UMZ2000_not_coincident_tbl_UMZunion_with_ID90_ID00"

5. Add the selected records in "qry_UMZ1990_not_coincident_tbl_UMZunion_with_ID90_ID00" to the table "tbl_UMZ1990_UMZ2000_Relation_all" --> "qry_03_add_missing_UMZ1990_to_UMZ1990_UMZ2000_relation_all"

6. Add the selected records in "qry_UMZ2000_not_coincident_tbl_UMZunion_with_ID90_ID00" to the table "tbl_UMZ1990_UMZ2000_Relation_all" --> "qry_04_add_missing_UMZ2000_to_UMZ1990_UMZ2000_relation_all"

7. Query to create an identical table to "tbl_UMZ1990_UMZ2000_Relation_all" named "Relation_IDs_UMZ1990_UMZ2000" which will store the final relationship table UMZ1990 - UMZ2000 --> "qry_05_create_final_UMZ1990_UMZ2000_relationship_IDs"

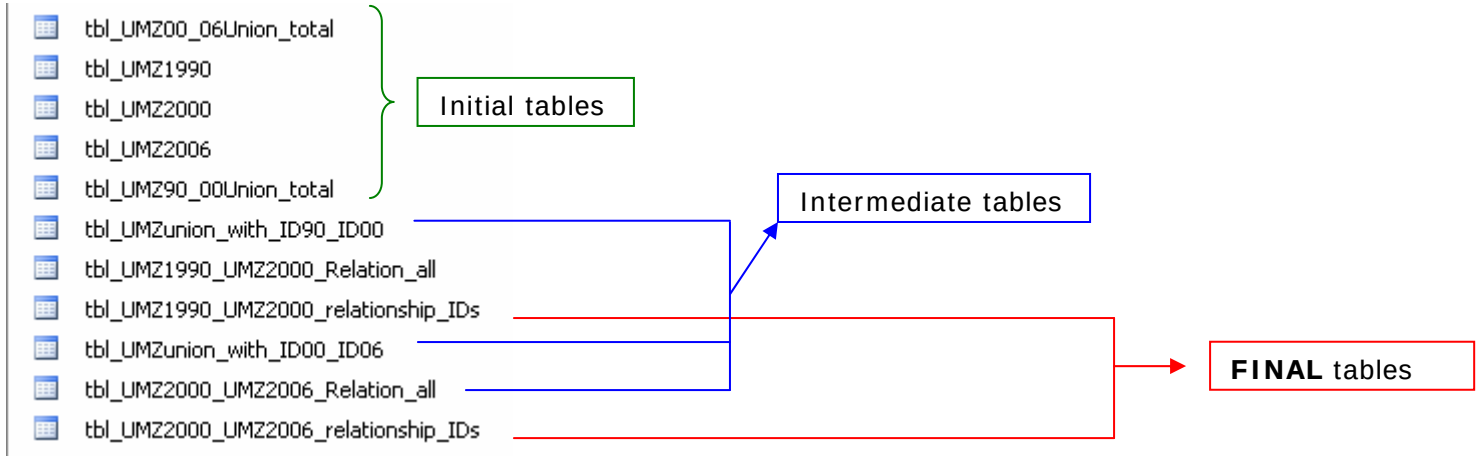
8. Query to update "tbl_UMZ1990_UMZ2000_relationship_IDs". Records with value 0 for ID90 update to -9999 --> "qry_06_update_final_UMZ1990_UMZ2000_relationship_ID90"

9. Query to update tbl_UMZ1990_UMZ2000_relationship_IDs. Records with value 0 for ID00 update to -9999 --> "qry_07_update_final_UMZ1990_UMZ2000_relationship_ID00"

qry_01_Union_with_ID90_ID00	Create a table with the records with ID90 and ID00 > 0 in "tbl_UMZ90_00Union_total"
qry_02_create_final_UMZ1990_UMZ2000_Relation_all	Create an identical table to "tbl_UMZunion_with_ID90_ID00". It will be used to add new records which don´t fit the sentence: ID90 > 0 AND ID00 > 0
qry_03_add_missing_UMZ1990_to_UMZ1990_UMZ2000_relation_all	Add the selected records in "qry_UMZ1990_not_coincident_tbl_UMZunion_with_ID90_ID00" to the table "tbl_UMZ1990_UMZ2000_Relation_all"
qry_04_add_missing_UMZ2000_to_UMZ1990_UMZ2000_relation_all	Add the selected records in "qry_UMZ2000_not_coincident_tbl_UMZunion_with_ID90_ID00" to the table "tbl_UMZ1990_UMZ2000_Relation_all"
qry_05_create_final_UMZ1990_UMZ2000_relationship_IDs	To create an identical table to "tbl_UMZ1990_UMZ2000_Relation_all" named "tbl_UMZ1990_UMZ2000_relationship_IDs" which will store the final relationship table UMZ1990 - UMZ2000
qry_06_update_final_UMZ1990_UMZ2000_relationship_ID90	To update tbl_UMZ1990_UMZ2000_relationship_IDs. Records with value 0 for ID90 update to -9999
qry_07_update_final_UMZ1990_UMZ2000_relationship_ID00	To update tbl_UMZ1990_UMZ2000_relationship_IDs. Records with value 0 for ID00 update to -9999
qry_08_Check_all_UMZ1990_NOTin_UMZ1990_UMZ2000_relationship_IDs	To check if any UMZ1990 is not in tbl_UMZ1990_UMZ2000_relationship_IDs
qry_09_Check_all_UMZ2000_NOTin_UMZ1990_UMZ2000_relationship_IDs	To check if any UMZ2000 is not in tbl_UMZ1990_UMZ2000_relationship_IDs
qry_11_Union_with_ID00_ID06	Create a table with the records with ID00 and ID06 > 0 in "tbl_UMZ00_06Union_total"
qry_12_create_final_UMZ2000_UMZ2006_Relation_all	Create an identical table to "tbl_UMZunion_with_ID00_ID06". It will be used to add new records which don´t fit the sentence: ID00 > 0 AND ID06 > 0
qry_13_add_missing_UMZ2000_to_UMZ2000_UMZ2006_relation_all	Add the selected records in "qry_UMZ2000_not_coincident_tbl_UMZunion_with_ID00_ID06" to the table "tbl_UMZ2000_UMZ2006_Relation_all"
qry_14_add_missing_UMZ2006_to_UMZ2000_UMZ2006_relation_all	Add the selected records in "qry_UMZ2006_not_coincident_tbl_UMZunion_with_ID00_ID06" to the table "tbl_UMZ2000_UMZ2006_Relation_all"
qry_15_create_final_UMZ2000_UMZ2006_relationship_IDs	To create an identical table to "tbl_UMZ2000_UMZ2006_Relation_all" named "tbl_UMZ2000_UMZ2006_relationship_IDs" which will store the final relationship table UMZ2000 - UMZ2006
qry_16_update_final_UMZ1990_UMZ2000_relationship_ID00	To update tbl_UMZ2000_UMZ2006_relationship_IDs. Records with value 0 for ID00 update to -9999
qry_17_update_final_UMZ1990_UMZ2000_relationship_ID06	To update tbl_UMZ2000_UMZ2006_relationship_IDs. Records with value 0 for ID06 update to -9999
qry_UMZ1990_not_coincident_tbl_UMZunion_with_ID90_ID00	To identify the records (feature classes) from UMZ1990 which are not in tbl_UMZunion_with_ID90_ID00. These records only exist in UMZ1990 and not in UMZ2000
qry_UMZ2000_not_coincident_tbl_UMZunion_with_ID00_ID06	To identify the records (feature classes) from UMZ2000 which are not in tbl_UMZunion_with_ID00_ID06. These records only exist in UMZ2000 and not in UMZ2006
qry_UMZ2000_not_coincident_tbl_UMZunion_with_ID90_ID00	To identify the records (feature classes) from UMZ2000 which are not in tbl_UMZunion_with_ID90_ID00. These records only exist in UMZ2000 and not in UMZ1990
qry_UMZ2006_not_coincident_tbl_UMZunion_with_ID00_ID06	To identify the records (feature classes) from UMZ2006 which are not in tbl_UMZunion_with_ID00_ID06. These records only exist in UMZ2006 and not in UMZ2000

2.3.4 Final Step.

The final tables in our database will be:



This database can be quite big in size, 1GB. To reduce the size, just the important tables can be extracted to other database, excel sheets, dbf, etc. or delete unnecessary fields and tables

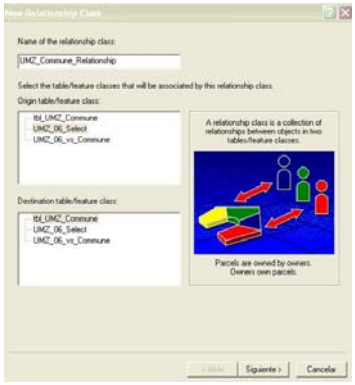
- tbl_UMZ1990_UMZ2000_relationship_IDs - UMZ1990_UMZ2000_relationship
- tbl_UMZ2000_UMZ2006_relationship_IDs - UMZ2000_UMZ2006_relationship

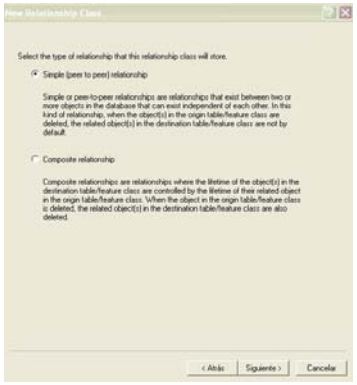
For example, a new database named *"UMZs_Relations_90_00_06_Final.mdb"* can be created and the previous tables imported.

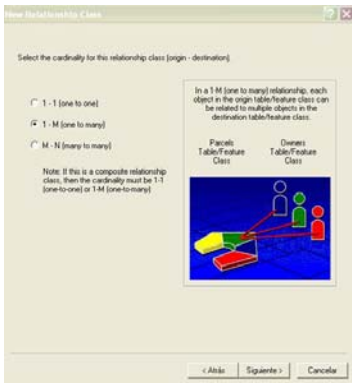
ANNEX III. DESCRIPTION OF THE RELATIONSHIP BETWEEN UMZ AND COMMUNES (LAU2) AT EU LEVEL

Steps to follow (Figure 23):

1. Make sure that the UMZs feature classes are in a geodatabase
2. Intersect the UMZ with the commune boundaries: “UMZ2000_vs_Communes”
3. Remove unnecessary fields
4. Export the table of this new feature class. Name: “Communes_UMZ2000”
5. Create the relationship in ArcCatalog







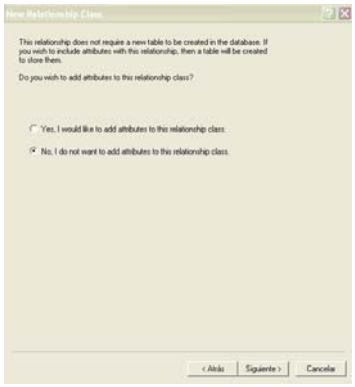


Figure 23. Procedure to create the relationship UMZ – Commune 2006