Class 7 – Vector Data Manipulation

1 Vector Edition

*The edition of vector data using SPRING is done over thematic, cadastral, network and digital terrain modeling (DTM) maps. The vector representation of these maps is the more precise way of representing a geographic object, using basic entities such as **points, lines and areas** (or polygons) to define thematic classes, geographic objects and numerical samples (contour lines and spot-heights)*

*When the user is editing vectors, especially from cadastral maps, it is necessary to go through the steps of **Digitalization, Adjustments and Generation of Polygons**. If he is editing a numerical IL, he needs to go through the Digitalization, along with a few adjustments.*

Vector data can be inserted in the system using import routines.
Check below some examples where ASCII files are imported.

1.1Importing ASCII files

Next you will see how to make up a file for the drainage and road maps. Note that there is no category and thematic classes to import the road map, therefore the user will have to define it:

***Note**: Always use the same syntax presented here to create names for categories and classes because they are case sensitive.*

Defining the thematic model for the roads map:

⇒ Creating a thematic category:

*Windows: **#Start – Spring<version><Language><system> – Spring<version> <Language>***

Linux: # Command to be typed on the Console (Shell) – # s_spring

*MAC: #Dock – Launchpad – Spring <version> < **language** >*

SPRING

Load the database **Course*

– [File] [Data Model...] or button 

Data Model – categories tab

– {Categories – Name: Roads} – (Model ⇔ Thematic) – (Create)

Data Model – thematic tab

– {Thematic Classes – Name: Principal} – (Create)

– {Thematic Classes – Name: Secondary} – (Create)

– {Thematic Classes – Name: Urban} – (Create)

– (Apply) – to save classes and categories.

Defining the visual parameters of the thematic classes:

– {Thematic Classes – Principal}

– (Visual ...)

Setting Visual Parameters **In* ***Lines*** *choose*

– (Width: 1)

– (Lines – Color...)

Color

** Choose a color*

– (Apply)

– (Close)

** Repeat the same steps for the other thematic classes*

Data Model

– (Close)

⇒ *Importing thematic data for roads:*

SPRING

** Load the project* ***Brasilia***

– [File][Import][Import Vectorial and Matricial data...]

Import

– Data tab (Directory...: C:|Tutor_10classses|Data) – Windows

~/Tutor_10classes/Data – Linux

~/Tutor_10classes/Data – MAC

– (Format ⇔ ASCII-SPRING: Mavias_L2D.spr)

– (Entity ⇔ Line with adjustment), (Unity ⇔ m), {Scale: 1/: 25000}

** Projection and bound box – not necessary*

– Output Tab

** Project – not necessary, the active project is taken*

– (Category ...)

Categories List

- (Categories: Roads) the one created above

- (Apply)

- {IL: Road_map)

- (Aply)

⇒ ***Importing labels:***

SPRING

- [File][Import][Import vectorial and matricial data]

Import

Data tab (File)... *C:|Tutor_10classes|Data - Windows*

~/Tutor_10classes/Data - Linux

~/Tutor_10classes/Data - MAC

**Follow the same steps above, except:*

- (Format ⇔ ASCII-SPRING: Mavias_LAB.spr)

- (Entity ⇔ Labels)

- (Execute)

Output tab

(Category..)

Categories list

{IL: Roads_map}

-(Apply)

(Close)

Visualizing the imported thematic data on the main screen:

Control Panel

- (Categories | Roads)

- (Infolayers | Road_map)

- (Lines), (Classes)

- (Select ...)

Data selection

**Select a specific class or All*

**Now, repeat the process above to create the rivers map. This time, it is not necessary to define*

the category to receive the rivers map because it is already defined in the database Course.

⇒ ***Importing thematic data for drainage:***

Windows: #Start - Spring<version><Language><system> - Spring<version> <Language>

Linux: # Command to be typed on the Console (Shell) - # s_spring

MAC: #Dock - Launchpad - Spring <version> < language >

SPRING

Load the database **Course*

Load the project **Brasilia*

- [File] [Impor][Import Vectorial and Matricial data...]

Import

- Data tab (Directory...: C:|Tutor_10classses|Data) - Windows

~/Tutor_10classes/Data - Linux

- (Format ⇔ ASCII-SPRING: Drenagem_L2D.spr)

- (Entity ⇔ Line with topology), (Unity ⇔ m), {Scale: 1/: 25000}

** Projection and bound box - not necessary*

** Project - not necessary, the active project is taken*

- Output tab

- (Category ...)

Categories List

- (Category: Drainage)

- (Apply)

- (IL: Rivers_map)

- (Execute)

⇒ Importing labels:

SPRING

- [File][Import][Import vetorial and matricial data ...]

Import

**Follow the same steps above, except:*

- (Format ⇔ ASCII-SPRING: Drenagem_LAB.spr)

- (Entity ⇔ Labels)

Output tab

- (Category ...)

Categories List

- (Category: Drainage)

- (Apply)

- (IL: Rivers_map)

- *(Execute)*

- *(Close)*

⇒ ***Visualizing the imported thematic data on the main screen:***

Control Panel

- *(Categories | Drainage)*
- *(Infolayers | Rivers_map)*
- *(Lines), (Classes)*
- *(Select ...)*

Data selection

**Select a specific class or All*

2 Graphical edition

*To perform a graphical edition, you need to open the "**Topological Edition..**" toolbar (image below) and select the parameters needed in order to get the traces you want.*

a) ***Operation**: Graphical Editor;*

b) ***Edit**: Lines or Points;*

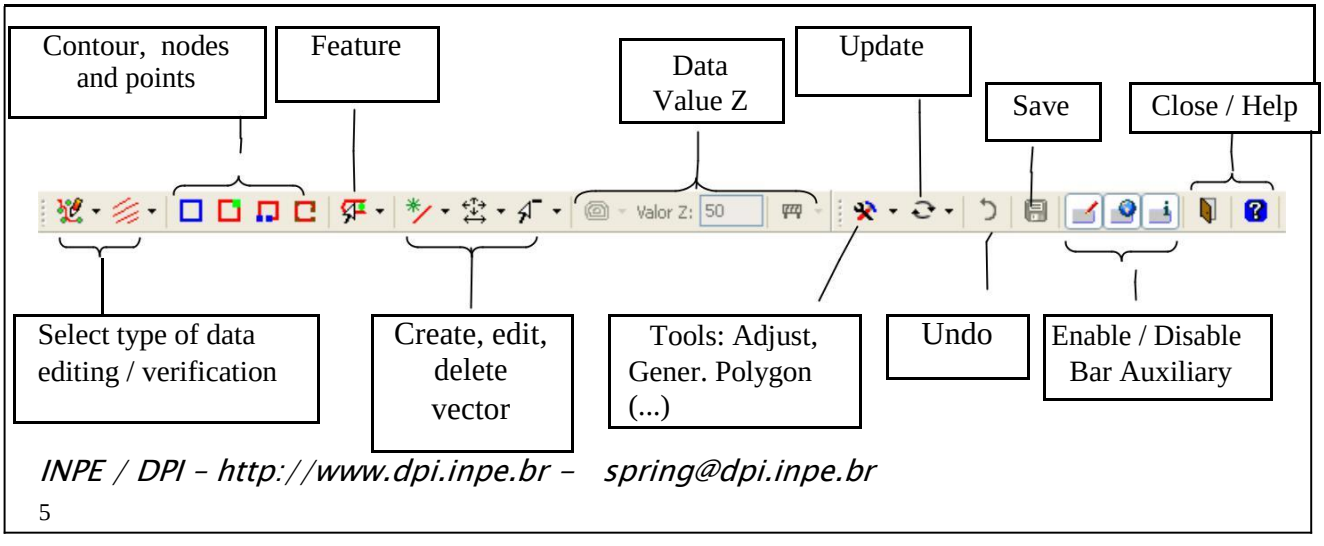
*c) **Mode**: Continuous or Step;*

*d) **Topology**: Manual or Automatic;*

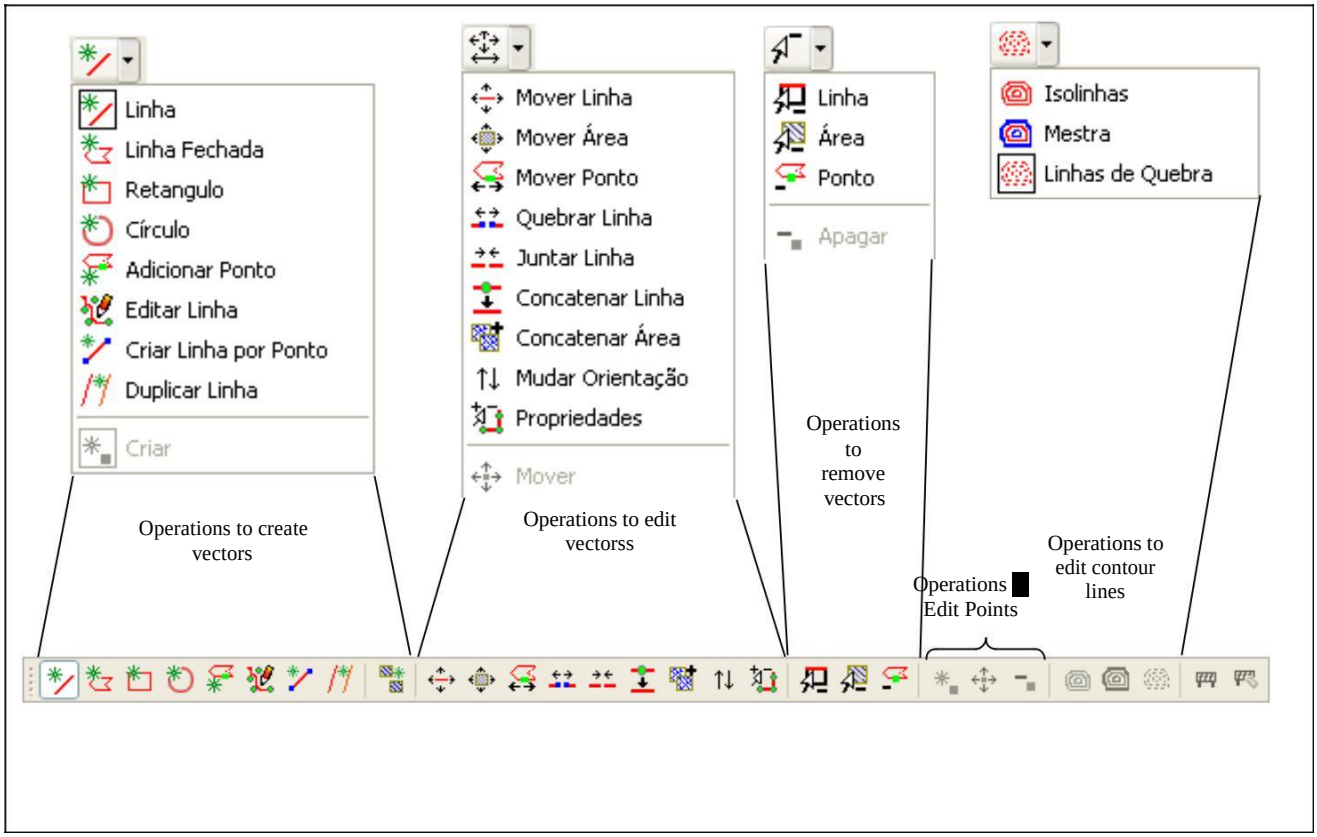
*e) **Digitalization step (mm)**: 0.00 to 2.00;*

f) ***Operation**: Described below.*

TOPOLOGYC EDITION TOOLBAR



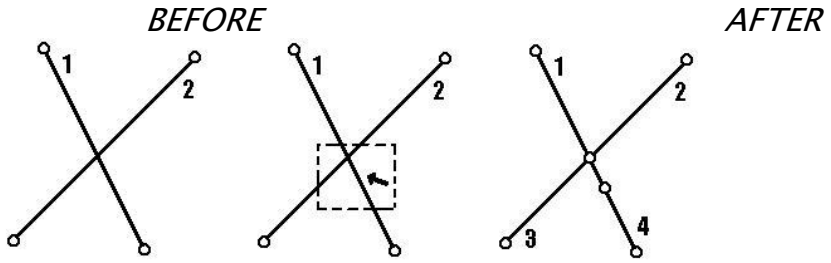
TOPOLOGIC EDITION AND CREATION TOOLBAR



When selecting an editing tool, check the tip at the main page bottom, because it will give information on how to use the tool. For example, when selecting the tool "Create Line"  *key, the following tip will be shown:*

When the user needs to solve the problem of creating a node where two arcs cross, he should do as follows.

In the figure below (manual topology situation) we have two arcs crossing and we want to break at their intersection. To do that, move close to the point where you want to break the lines, for the system looks for the closest line (arc1) to break. Besides, it automatically searches for the closest lines (arc2) that lie inside the ***Digitalization Step*** *to be broken. However, the arcs will be broken in two points, arc 1 breaks into arcs 1 and 4 and arc 2 into arcs 2 and 3.*



2.1Editing the Land Use Map

In the following exercise, the user will perform the polygonalization adjustment since not all polygons are already adjusted. The topology will be defined based on the map described on the ANNEX 1. To compose the map, the user needs to import some files. First, it is necessary to edit an ASCII file that will be the frame (boundary) of the IL in use. Then, the user should import this file. Next, import the water limits (file: Agua_L2D.spr) and urban area limits already established (file Urbano_L2d.spr). The other areas of the IL can be used for urban expansion.

The Land Use Map must have the following themes: Savannah, Water, and Urban (see ANNEX 1).

Note: On purpose, the user will have to manually adjust one of the arcs, following the instructions given above.

Next, the frame file will be shown (for instance, Lim_uso.spr). Choose any text editor (Notepad, Wordpad, vi, gedit, kwrite etc.) and edit the lines below, saving the file on the directory "C:|Tutor_10classes|Data":

LINES
INFO
*|| **ASCII file generated by SPRING***
*|| **Project Brasilia - Information layer Molde***
INFO_END

s 15 52 30 w 47 57 30
s 15 41 55 w 47 57 30
s 15 41 55 w 47 47 00

s 15 52 30 w 47 47 00
s 15 52 30 w 47 57 30
END
END

⇒ *Importing thematic data into the Land Use Map*

Windows: #Start - Spring<version><Language><system> - Spring<version> <Language>

Linux: # Command to be typed on the Console (Shell) - # s_spring

MAC: #Dock - Launchpad - Spring <version> < language >

SPRING

** Load the database **Course***

** Load the project **Brasilia***

- [File][Import...][Import vetorial and matricial data]

Import

- (Directory...: C:|Tutor_10classses|Data) -Windows
~/Tutor_10classes/Data - Linux

- (Format ⇔ ASCII-SPRING : Lim_uso.spr)

- (Entity ⇔ Line without adjust), (Unit ⇔ m), {Scale 1/: 25000}

** Projection and bound box - not necessary*

** Project - not necessary, the active project is taken*

- Output Tab

- (Category ...)

Categories List

- (Categories: Land_Use)

- (Apply)

- (IL: Use_map)

- (Execute)

** Repeat the same steps above for the files water limits and urban, but do not forget to click on **Mosaic**.*

Import

***-Data tab (File)** C:|Tutor_10aulas|Dados - Windows ou ~/Tutor_10aulas/Dados - Linux*

- (Format ⇔ ASCII-SPRING : Urbano_L2D.spr)

- (Entity ⇔ Line without adjust), (Unit ⇔ m), {Scale 1/: 25000} Output tab

- (Category ...)

Categories List

- *(Categories: Land_Use)*
- *(Apply)*

- *(PI: Use_map)*
- *(Mosaic)*
- *(Execute)*
- *(Close)*

⇒ *Visualizing the Land Use Map on the main screen:*

Control Panel

- *(Categories | Land_Use)*
- *(Infolayers | Use_map)*
- *(Lines)*

⇒ *Adjusting the lines and generating polygons at the Land Use Map:*

**Load the thematic plan Land_Use generated on the import above.*

- *[Edit][Vector...] or [Thematic][Vector Editing...]*

Vector Edit Tools

- *(Graphical Editor)*
- *(Show Adjusted Nodes)* 
- *(Show Unadjusted Nodes)* 
- *(Snap (mm) ⇔ 0.5)*
- *(Adjust)* 

** Verify the adjustments results on the footnote of the Main Window screen*

- *(Verification)*
- *(Verify ⇔ Nodes)*

SPRING

Vector Edit Tools

- *(Graphical Editor)*

** Use the tools (Delete Line), (Break Line), (Join Lines), (Add Line), (Delete Point) and (Move Point) to correct nodes*

**Use the facility in [View][Zoom Cursor] or*  *to locate the non-adjusted nodes.*

NOTE: After doing that, release the zoom Cursor to restart editing.

- *(Adjust)* 

Make sure there are no errors on the footnote of the Main Window screen. In case there are still nodes without adjustments, use the edition tools again to fix them. The user must proceed to the generation of polygons only after all lines are adjusted.

- (Gener. Polygons) 

Note**: During this phase of line edition, or correction of arcs that have not connected to each other to form polygons, the user can click on the option **Show Adjusted Nodes  *and **Show Unadjusted Nodes***  *on the **Topology Editor Toolbar**. After that, click on [Execute] [Draw] or button*  *on the active display where the edited arcs are. At the end of each arc (node) there will be a blue asterisk (*) for unconnected arcs and a green circle for nodes with more than one arc (the adjustment of two or more arcs is correct). It is possible to change the size of the cross or square by altering the "digitalization step".*

⇒ Associating thematic classes to vectors: Vector Edit Tools

- (Classes...) 

Thematic Classes Editor

- (Classes | Water)

- (Operation ⇔ Associate)

- (Entity ⇔ Polygon)

** Select the vector corresponding to the class Water on the screen.*

** Repeat the operation for the other polygons.*

** Use the Operation "Dissociate" in case you make any wrong association.*

** Repeat the operation for the other classes as shown in the Annex 1 map.*

- Close

**Visualize on the active screen the Land Use Map*

2.2Soil Map Edition

*The creation of the soil map will be based on the map in **Annex 2**. The user may import only the frame that defines the border of the Information Layer that will be created. The boundary between the classes of soil should be drawn using the digitizing table or the mouse. A thematic category and a set of classes should also be defined.*

⇒ *Defining the thematic model for the soil map:*

Windows: #Start – Spring<version><Language><system> – Spring<version> <Language>

Linux: # Command to be typed on the Console (Shell) – # s_spring

MAC: #Dock – Launchpad – Spring <version> < language >

SPRING

Load database **Course*

Load project **Brasilia*

– [File] [Data Model...] or button 

Data Model – Categories Tab

– {Categories – Name: Soil} – (Model ⇔ Thematic) – (Create)

– (Execute)

– {Thematic Classes – Name: LEd1} – (Create)

– {Thematic Classes – Name: LVd1} – (Create)

– {Thematic Classes – Name: Cd12} – (Create)

– {Thematic Classes – Name: Cd1} – (Create)

– {Thematic Classes – Name: Cd17} – (Create)

– {Thematic Classes – Name: Cd14} – (Create) Execute

Defining the visual of the thematic category

– {Categories – Soils}

– {Thematic Classes – LEd1}

– (Visual ...)

– (Apply) – To create the classes and category

Setting Visual Parameters Area Tab

– (Areas SOLID) – or any other pattern

– {Areas – Color...)

** Choose a color*

– (Apply)

** Repeat the same steps for the other thematic classes*

– (Close)

Data Model

– (Close)

⇒ *Importing the boundaries of the Soil Map:*

– [File][Import][Import Vectorial and Matricial data...]

Import

C:|Tutor_10classses|Data) -Windows

~/Tutor_10classes/Data - Linux

- (Format ⇔ ASCII-SPRING: Molde_L2D.spr)

- (Entity ⇔ Line without adjustment), (Unity ⇔ m), {Scale: 1/: 25000}

** Projection and bounding box - not necessary, the one of the active project is used*

** Project - not necessary, the active project is used*

Output Tab

- (Category ...)

Categories List

- (Categories: Soils) the one created above

- (IL: Soil_map)

- (Execute)

Vector Edition of the Soil Map

⇒ Calibrating the digitizing table (in case the table is connected):

SPRING

- [Tools] [Digitizer Calibration ...]

Digitizer Calibration

- (Coordinates ⇔ Planes)

- (Point 1)

** Select the point 1 on the map over the table*

- {X(m):XXXX}, {Y(m): YYYY} - Note: XXXX,YYYY correspond to the point coordinates

**Repeat for points 2, 3 and 4.*

- (Apply)

- (Test)

**Select on the map a point whose coordinate is known and compare to the presented coordinates.*

Preparing for on-screen edition (in case there is no digitizing table)

*Digitize the lines as shown in the figure in **Annex 2** or use a satellite image as a background:*

*- Visualize the image **Comp_3B_4R_5G***

⇒ *Edit vectors:*

SPRING

Load the information layer **Soil_map from the thematic category **Soil.***

- [Edit] [Vector...]

Edit vector Toolbar

- (Operation ⇔ Graphical Editor) 

- (Edit ⇔ Lines) 

Edit vectors in the continuous mode:

Editing vector Toolbar

- (Mode ⇔ Continuous)

- (Topology ⇔ Automatic)

- (Digit. Step(mm) ⇔ 0.50)

- (Operation ⇔ Create Line  *or Create Closed Line*  *)*

*- **Control Panel** ("Table cursor") - Note: Only when using a table for editing*

** Digitize the contour of the shapes of interest (table or screen).*

Adjusting lines and generating polygons

Edit vector Toolbar

- (Snap(mm) ⇔ 0.5)

- (Show Nodes)

- (Adjust) 

- Verify the adjustments on the bottom (footnote) of the Main Window.

- (Verification) 

- (Verify ⇔ Nodes) 

SPRING

- [Execute] [Draw] or button 

Editing vector Toolbar

- (Operation ⇔ Graphical Editor)

** To fix the nodes use the tools (Delete Line), (Break Line), (Join Lines), (Add Point), (Delete Point)*

and (Move Point).

** Use the Zoom feature at [View] [Zoom Cursor] or*  *to locate non-adjusted nodes. NOTE: After*

doing that, release the zoom cursor to restart editing.

- (Adjust) 

** Make sure there are no errors at the footnote of the* ***Main Window****. In case there are still non-*

adjusted nodes use the tools again to fix them. Go on with generating the polygons after all lines are

adjusted.

- (Gener. Polygons) 

⇒ *Associating thematic classes to the polygons*

Edit Vector Toolbar

- (Classes...) 

Thematic Classes Editor

- (Classes | LEd1)

- (Operation ⇔ Associate)

- (Entity ⇔ Polygon)

** Select the corresponding polygon on the active screen.*

** Repeat the operation for the other polygons (see ANNEX 2)*

** Use the operation "Dissociate" in case you make any wrong association.*

- Close

** Visualize the Land Use Map on the active screen*

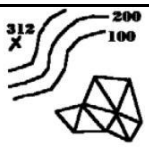
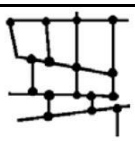
3Conversion of Formats

Vector and raster formats are usually supported for different data models. However, SPRING only supports the conversion between raster and vector formats of thematic data models.

Vector Format

The vector representation of an object tries to represent it as exactly as possible, precisely defining all positions, lengths and dimensions of its geometric entities.

Check on the table below the categories of different data models that can be represented on the vector format by SPRING:

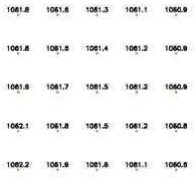
Category / Model	*Vector Representation*	*Example*
Thematic	*Points, Lines and Polygons*	
Numeric	*Samples (contour lines and spot-heights) and TIN (triangular grid)*	
Cadastral	*Points, Lines and Polygons*	
Network	*Points and Lines*	

Raster Format

The raster format can be defined as a set of cells located in contiguous coordinates, implemented as a 2D matrix. Each cell, also called matrix element, picture element or "pixel", is referenced by line and column indexes and contains a number that represents the type or attribute value that is mapped.

The vector and raster representations are not equivalent for the same data. There is usually a loss of precision when we transform the vector format to a raster format because continuous borders change to discrete according to the output image resolution. This loss can be compensated by the more efficient geographic analysis performed on the raster domain.

The following table presents different data models that can be represented in the raster format:

Category / Model	*Vector Representation*	*Example*
Thematic	*Thematic Image* • *One pixel – one point* • *Lined up pixels – one line* • *Grouped pixels – polygons*	
Numerical	*Rectangular grid* • *Real values associated to each point of the matrix*	
Image	*Monochromatic Image* • *Gray level pixels* *Synthetic Image (coded)* • *Pixels associated with a color table* *Classified Image* • *Group of pixels with the same color*	

Vector-Raster Conversion

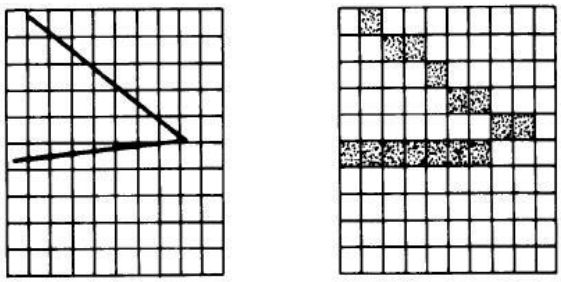
The vector-raster conversion creates the Thematic Image representation from the classes that are present on the Information Layers. One IL can contain just one Thematic Image. If there are any

changes on the vectors and classes, the conversion must be performed again.

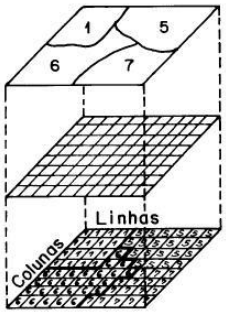
When we have a linear element, the conversion must be done overlapping a vector or linear element on a raster matrix. This conversion identifies which raster elements are crossing the line and generates codes with attributes or values of classes associated with the line.

Example:

Vector Format *Raster Format*



For polygonal elements, it is necessary to initially define the area to be converted (usually the bounding box of the set of polygons) and the "pixel" size. This information defines the grid that is laid over the original polygon map. Each "pixel" must be associated to a class (or some attribute value).



Original Map
Vector Format

Overlay grid

Map in raster format

⇒ *Converting a thematic map to raster:*

Windows: #Start - Spring<version><Language><system> - Spring<version> <Language>

Linux: # Command to be typed on the Console (Shell) - # s_spring

MAC: #Dock - Launchpad - Spring <version> < language >

SPRING

Load the database **Course*

Load the project **Brasilia*

Load the information layer **Land_Use*

SPRING

- [Thematic] [Vector->Raster…] Vector -> Raster

- {Horizontal: 30}, {Vertical: 30} -{8 bits no signal (0…255)}

- (Apply)

**Visualize the thematic image generated*

**Repeat the same steps for the Soil Map:*

⇒ *Converting a thematic map to vector:*

Control Panel

Load the thematic plan obtained from the classification of a satellite image such as: **tm345sub-isoseg-tematica*

SPRING

- [Thematic] [Raster ->Vector …]

Raster ->Vector

- (Arc Smoothing ⇔ Yes)

- (Apply)

**Visualize the vectors generated.*