

The Symbolization of Map Vector Data on Microstation

Liu Haiyan Sun Qun

Department of Cartography, Institute of Surveying and Mapping

No. 66 Longhai Middle Road,

Zhengzhou, Henan Province 450052

P.R. China

Email: lhy@371.net

Abstract

Now the mainly method of map producing is symbolization of vector data, which comes from map database. But the data usually includes the information of geo-attribute and location, not include the information of symbol (e.g. shape, size, color and so on). It means that there is a long distance between the map data and the symbols map. So this paper discusses the process of the map producing, which the data translates symbol on Microstation.

The geo-attribute information contains kind_code, type_code, describe_code. One kind_code and one type_code can express one element, and only match one symbol. The describe_code records the description information, which be expressed texts, symbol, or nothing on a map. So it is very important to analysis every element code in order to translate the symbol. In this part, we discuss the method of building the symbol library and the criterion of symbol size, symbol color, line weight on Microstation, which must be strictly obeyed the publishing requests. Defining the structure of symbolization control table is the emphasis in the paper. The tables control the symbolization process of all elements. And it is sample to maintain, if user want to add or delete one element, what he do is only add or delete one row. We expound the structures of the point, line, area element symbolization control table. In the point table, one kind_code, one type_code and describe_code indicate one symbol what is defined and its place_level, color and text. In the line table, we offer that the line symbol is composed of base line style and assistant line style, because the line style only has one color, but the line symbol have one color or two colors. The area symbol of the map is the most complex part. Through summarizing and analysis, we find all area symbols are made up of three forms: bound line, fill and pattern. So the area structures include those definitions. At last the symbolization program is finished with MDL on MicroStation. The order of the symbolization achieving is according to element kinds (e.g. water system, vegetation and so on) and types in one kind of elements, what is defined in the symbolization control table.

Key words : *Vector data, Attribute code, Entireness symbolization, Symbolization control table*

Introduction

It is very difficult to compile and modify map symbols because of map symbols' variety and complication. Before the method which we adopted is using simple symbols in compilation map, e.g. expressed a point symbol with a point or a circle, expressed a line symbol with difference color

line. These graphics only showed their location information, if you wanted to publish these maps, you should must replace these symbol according to publish requests after finished compilation and modification. So this method had some disadvantages:

- 1) This method gain a lower a efficiency to produce a map for a worker. In the process of complication map, he only distinguish the map element with difference colors simple point or line symbol. If he wants to know the symbol's attribute, he must often query information.
- 2) It is very difficult to process correctly the geo-relation between map elements, because in the computer screen there are not map symbolization graphics, e.g. you don't judge the distance between the circle symbol of residential area and its toponym; when complication map line elements which don't express their actual publish width, It take place the across and cover of symbols.

So It is a mainly problem of the map symbolization of all element. The mean of symbolization have two points: one is to realize symbolization map, another is that the whole process is symbolization, but not a part. The size, shape, color of symbols in screen is same as in publish, like "what see is what gain". The Microstation is a good software which fits the map's complication and modification, at same time we can solve the map publish problem, and gain the process color film. After the map database have been built, it is mainly path to use the vector data of map database for map producing and publishing. But the vector data usually don't include the information of shape, size, color and so on. It means that the map data is not the "map" which is made of various symbols. The data include the information of attribute and locate. So the map producing is the process which the data translate to symbol.

The attribute code match the map symbol

The map data is composed of the attribute information and coordinate. The attribute information expresses all kinds of element in map. Usually three class codes apply on attributes data:

- 1) class_code, which is coded according to different classes in map, e.g. traffic, boundary. There are many types of elements in one class.
- 2) type_code, which is coded according to different types in one class, the type is unique, the user can distinguish the element by it, e.g. , highway, main road.
- 3) describe_code, which is coded according to different descriptions in one type, e.g. the material of the road, the "building" or "built" road. One element may have several describe_codes.

One class_code and one type_code express one type elements, and match only one symbol. The describe_code records the description information, which be expressed with texts, with symbol, or nothing. So the code translating to the symbol must separate the three conditions:

- 1) Be expressed with symbol, e.g. the electric railway, the road with dykes and so on.
- 2) Be expressed with text, e.g. the material of the road.
- 3) Be expressed with nothing, e.g. the city's administration grade.

On the base of analysis, we must build the relation between attribute code and map symbol, and between describe_code and explaining information.

In addition, one element has usually several parameters, e.g. it's height, width and so on. These parameters must be expressed in the process of symbolization.

Build the symbol warehouse

The symbol warehouse is necessary for realization the symbolization. When making the symbol, the definition of the size, color, weight must be strictly obeyed the publishing request . In this case, we use the MicroStation to build the symbol.

1) define weights

There are 32 (0-31) line weight in MicroStation. These weights control the show of lines in screen , not the publishing weight. When publishing process, it links the “pen_table”, in which is defined “wt=1” to “0.1mm” by user.

2) define colors

The map is published with CYMK process color or special colors., but in MicroStation the symbol only use color_code (0-255). The color_code must be defined to match the publish color. So the symbol color must be defined before making.

3) make symbol

The point symbol and the area symbol save in the cell library. Using the “PLACE CELL” tool, you can place the point symbol, Using “PATTERN AREA” tool, you can fill the area. The custom line style help user make the complex line symbol.

Build the control relation table

It is difficult to design programme for each type element, because there many more type elements in map. It is sample to use the control relation table. The control table accord the point, line, area to setup. If user want to add or delete one element, what he do is add or delete one row, and don't modify the program. A whole map need build several tables to control.

1) The struct of the point symbol symbolization control table

```
typedef struct
{
    char Note[50];
    int Class_code;
    int Type_code;
    int Decscrib_code[6];
    int level;
    char CellName[7];
    int Text_level;
    int Text_color;
    long Text_font;
    int Parameter_level;
    int Parameter_color;
    long Parameter_font;
}CtlPoint;
```

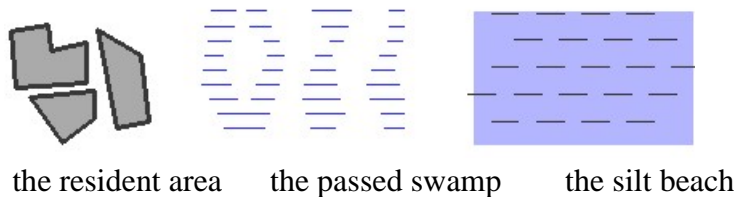
One Class_code, one Type_code and Decscrib_codes indicate one symbol what is defined its level, color and text.

2) The struct of the point symbol symbolization control table

```
typedef struct
{
    char Note[50];
    int Class_code;
    int Type_code;
    int Decscrib_code[6];
    int level;
    char LineStyle_Name[10];
    int LineStyle_color;
    int LineStyle_weight;
    char Accessorial_LineStyle_Name[10];
    int Accessorial_LineStyle_level;
    int Accessorial_LineStyle_color;
    int Accessorial_LineStyle_weight;
    int Text_level;
    int Text_color;
    long Text_font;
    int Parameter_level;
    int Parameter_color;
    long Parameter_font;
}CtLine;
```

In this struct, we offer the *Accessorial_LineStyle* because the LineStyle is alone color. But in map, the line symbol have one color or two colors, e.g. the highway symbol. So we define the *Accessorial_LineStyle* express the other color symbol.

3) The struct of the area symbol symbolization control table



The area symbol of the map is the most complex part. There are three examples to explain the point.
 e.g1, the resident area, which is compose of area_fill and area_bound_line.
 e.g2, the passed swamp, which is a pattern.
 e.g3, the silt beach, which is compose of the pattern and area_fill.
 Through summarize and analysis ,we find the all area symbol are made up of three form: bound line, area fill and pattern. So the struct include their's definition.

```

typedef struct
{
    char Note[50];
    int Class_code;
    int Type_code;
    int Decscrib_code[6];
    int level;
    char PatternCell_Name[7];
    float RowSpacing;
    float ColumnSpacing;
    float Pattern_Angle;
    int Pattern_color;
    char BoundLineStyle_Name[10];
    int BoundLine_color;
    int BoundLine_weight;
    int BoundLine_level;
    int BottomFill_level;
    int BottomFill_color;
    int Text_level;
    int Text_color;
    long Text_font;
    int Parameter_level;
    int Parameter_color;
    long Parameter_font;
}CtlArea;

```

The pattern fill need the parameters of *level*, *PatternCell_Name*, *RowSpacing*, *ColumnSpacing*, *Pattern_Angle*, *Pattern_color*.

The bound line is defined by the parameters of *BoundLineStyle_Name*, *BoundLine_color*, *BoundLine_weight*, *BoundLine_level* and so on.

The area fill is controlled by the *BottomFill_level* and the *BottomFill_color*.

These control tables specify the relation of the symbols and the codes. The users will realize the symbolization of the vector maps if they define these tables that point to those maps.

The realization of symbolization

The process of the realization of symbolization is showed in the frame. And we have finished the risk with MDL in MicroStation.

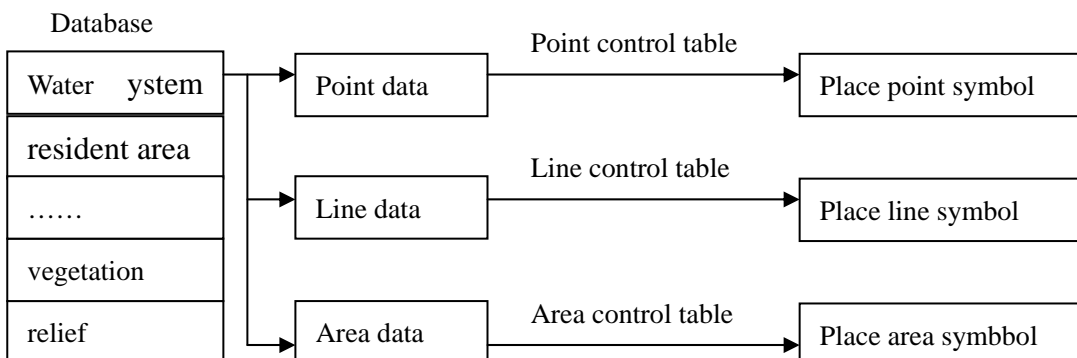




figure 1 map symbolization