

有线电视管理系统动态编码与基于数据的树形实现

陈争航 (成都西南交通大学电气工程学院 610031)

摘要: 本文介绍有线电视管理系统的一种新的实现方法, 该方法实现用户地址动态编码, 使得地址编码可以根据需要进行不同设置, 并引入树形来实现用户的注册与管理, 使用户管理更加方便。

关键词: 动态编码 基于数据库的树形

1 项目背景

在有线电视管理系统中, 通过控制箱实现对用户的开、关、加扰控制。在管理系统中, 为每一个控制箱分配一个唯一的16位地址编码, 每一个地址编码对应唯一的实际存在的地理地址。例如: 控制编码0000212121212121对应地理地址“成都市成华区建设路23号1单元”, 一个地理地址可以对应多个控制箱。在用户注册时, 根据用户所在地址为用户指定一控制箱, 并分配一个两位的端口(一个控制箱下有最多24个端口), 这样使用16为地址编码和2位端口号组成和用户唯一对应的18位地址编码。通过18位地址编码的前16位确定控制箱, 然后再通过端口号确定用户, 从而实现对用户的控制和管理。

在实际操作中, 将地址编码分为若干段, 不同段地址对应不同级地理地址。例如, 可以将16位地址分为5段, 第一段地址对应“市”级, 第二段对应“区”级, 第三段对应“路”级, 第四段对应“楼”级, 第五段对应“单元”级。这样做便将地址编码进行分区, 使同一地理地址的用户的18位地址中的前面部分相同, 从而将用户根据地理地址

进行分区, 方便管理。目前的有线电视管理系统所采用的方法存在下面两点不足:

(1) 地址编码不具有灵活性。目前所采用的方法一般是将16位地址编码分为固定的段数, 例如上面提到的5段, 而不能根据用户的需要进行灵活的分段。这样为某些地区的使用造成不便。例如, 在某些地区, 需要将地址编码分为多于5段, 而另外一些地区则需要将地址编码分为少于5段, 而有些地区在用户注册时并不需要使用全部的18位编码, 只需在控制时将编码补零, 使编码为18位。

(2) 用户注册不简便。目前的管理系统在用户进行注册时, 需要操作员输入用户地理地址并根据地理地址选择编码, 这样给操作造成不便。

2 方案的优点

本方案要实现的动态编码方法克服了上述的缺点, 主要优点有下面三点:

(1) 地址编码可以根据用户实际情况分段。本方案中, 地址编码可以分为不大于16的任意多段, 可分为1段, 也可以分为16段, 其

中最后一段为控制箱段(一个地理地址下可有多个控制箱)。

(2) 用户编码使用位数可以根据需要来定。地址编码可以根据需要使用不大于16的任意位, 加上2位端口号构成用户编码, 而在实现控制时将不足18位的用户编码补足18位即可。

(3) 用户注册更加方便。本方案中, 将地址编码信息映射为树形目录, 在用户注册时只需选取用户所在地理地址对应的目录, 选择控制箱, 然后为用户选取可用端口号即可。

3 具体设计方案

下面介绍具体设计方案, 开发工具采用C++Builder5, 数据库采用SQL Server2000。

3.1 数据库设计

数据库设计要达到以下几点要求:

(1) 应能保存地址编码的全局信息。要保存各段分别为几位以及各段位数和。

(2) 应能保存地址编码信息的各级编码所对应的地理名称及编码, 并有利于程序中实现树形目录。

根据要达到的要求, 数据库中

id	name	len
1	市	2
2	区	2
3	路	3
.....

表1 表 global-Info 示意图

设计了两个表来保存地址信息。表1 global-Info 保存地址编码的全局信息, 表2 detail-Info 保存各级编码所对应的地理名称及编码。

表1 global-Info 如图1所示。在该表中, 一条记录对应一级地址编码的信息, 各条记录按其对应的地理范围由大至小排列。该表中包含3个字段, id, name, len, 字段id为种子为1, 间隔为1的自增加整型字段, 它是表的索引, 它的作用在于使各条数据按顺序排列。字段name类型为char型, 表示该级编码对应的地理范围的类别, 该字段并没有实际意义, 可有可无, 只是为了在操作时更直观。字段len为int型, 存储该级地址编码为多少位。因为每一条记录对应地址编码的一级, 由记录条数可以得到地址编码分为多少段, 而将各条记录的len字段的值相加便是地址编码使用的总位数。

表2 detail-Info 如图2所示。在该表中, 一条记录对应一条地址编码的信息, 在程序中由该表产生地

name	code
成都市	01
金牛区	0101
交大路	0101023
交大西路	0101024
23号	0101023012
.....

表2 detail-Info

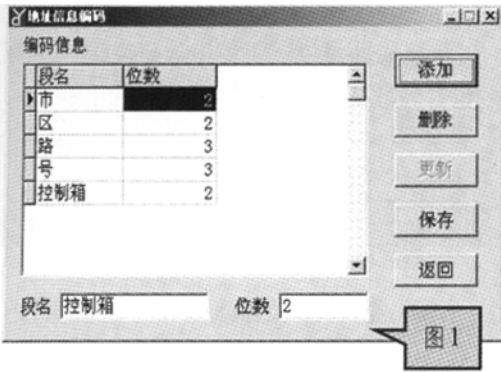


图1

址编码树形目录。表2code所对应的本级地理地址的名称，也是对应的树形目录中的目录名称。由图2可以看出，name字段只存储了当前级地理地址，而不包括其上级地理地址，而code字段则存储完整的编码，包括了其所有上级地理地址的编码。这样做是为了便于生成树形目录。

表2detail-Info中，code字段的值在表中是唯一的，由code值可以确定地理地址，而name字段的值在表中并不唯一，因为不同的父目录下可以有相同名称的子目录，而因为其父目录不同，所以它们的名称相同，编码却不会相同。

下面我们以图2中的数据作为编码信息，并以图1的数据作为其全局信息来说明如何应用这两个表。由图1中数据可知，第一级编码长度为2，第二级编码长度为2，第三级编码的长度为3。图2中，第3条记录的name字段值为“交大路”，code字段值为“0101023”，我们来通过其编码确定其地理地址。因为编码长度为7=2+2+3，由其全局信息可知，该编码为第三级编码。再由全局信息可知，编码中的前两为“01”表示当前地理地址所属的1级地址，则由图2可得到1级目录“成都”。由全局信息可知，前两级编码的总长度为2+2=4，所

以由编码中的前四为“0101”来确定其两级地理地址为“金牛区”，这样便却定了编码“0101023”对应的地理地址为“成都市金牛区交大路”。

在实际应用中，对更多的分段采用同样的方法。

3.2 程序设计

在了解了数据库设计后，程序设计便不难实现了。在程序设计中要完成以下几项任务：

- (1) 地址编码的全局信息的录入与修改。
 - (2) 地址编码信息的录入，通过树形目录实现。
 - (3) 将编码信息转化为树形目录。
- 全局信息录入与修改很容易实现，只需要设计一个录入界面来录入每一级地址的长度和对应的范围名。要注意的是限制录入记录个数不要超过16条，因为地址最多只能分为16段。具体实现在这里就不论述了。窗体设计大体如图3所示，其中段名表示范围名，位数为该级地址的长度。

地址编码信息录入要完成的任务是对地理地址进行编码，并实现树形化。这里的关键问题是将数据库数据以树形的转换为树形。对应于图2中的注册信息，转换为树形后应如图4所示。

下面以图2和图3、图4中所示的注册信息来说明注册信息的录入

和树形的实现。地址信息的录入采用添加子级的方法实现，C++Builder中实现代码如下：

```
TreeView1->Items->AddChild(name);
```

树形图中的“某有线电视台”文件夹是根目录，它的名称根据数据库中的电视台名称改变。在最初进行地址信息录入时，树形图中只有该默认文件夹，当选择根目录时，在名称栏中输入名称，并选择编号（下拉列表中的编号由程序产生），则通过添加按钮实现添加，并向表2detail-Info中写入数据，这样实现第一级目录“成都市”的添加。再选择“成都市”目录来添加它的子目录“金牛区”，以此类推，在录入地址信息时，子目录的地址编码由其父级的地址和本级地址合并组成，所以在添加子目录时，要获取当前目录的编码，同一目录的所有子目录的名称不能相同，可以通

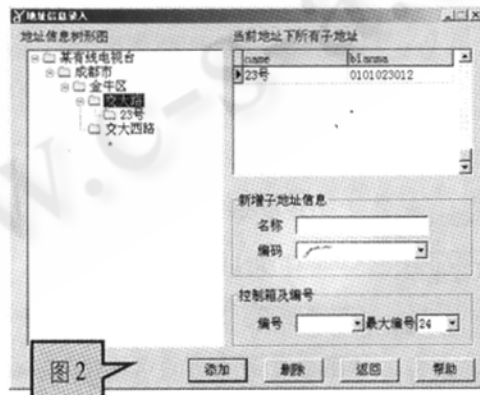


图2

过这一点获取当前目录的编码。另外，还应知道当前目录的级别，从而获知要添加的目录的级别，由此由表1global-Info来决定该级编码的长度，并判断名称是否重复。

在读取编码信息时，采用分级读取的方法。再打开地址信息录入窗口时，在树形中只添加第一级目录，当用户单击选择第一

级某一目时，添加该目录的子目录即二级目录，然后当用户单击选择二级目录时，再添加被选择目录的子目录，以此类推。第一级目录添加方法如下：

首先由表1global-Info读取第一级编码的位数，在图2中，第一级地址编码的长度为2。接着在表2detail-Info中查找编码长度为2（一级目录的编码长度）的所有记录，也就是所有一级目录，然后将对应的目录名添加为根目录的子目录，并展开第一级子目录。当选择第一级目录时，获取被选择目录的编码（记为b1），由表1global-Info中获取其子级即二级目录的编码长度，图2中所示为4，然后在表2detail-Info中查找编码的前2位（当前编码的长度）为b1且长度为4（子级的编码长度）的记录，找到的即为当前目录的所有子目录，然后将子目录添加到当前目录下。同理可以添加各级目录的子目录。这样树形便得以实现。

4 总结

该方案的优点在于：地址编码的总长度、段数，每一段的位数都可以按实际需要自由设置，这使得该方案具有极大的灵活性，可以适应于不同地区，不同规模的应用。另外，编码的树形实现，使得操作方便，更易于管理和维护。

方案的设计重点在于数据库的设计，该数据库的设计使编码的动态性成为可能，并极大的简化了树形目录的程序实现。在树形实现中，采用分级的方法简化了程序设计。 ■