Component Design of Tree-structure Model

Zong-pu LI
School of Information, Linyi University, Linyi, Shandong, China

Keywords: Tree-structure, Component Design, Relational Database.

Abstract: This article bring up structure and design scheme of general tree-structure component based on the necessary of the tree-structure processing component in relational database. It use re-design components of the data services layer to design components in tree-structure. There are TreeManage component, cls TreeManage component and TreeServer component designed in the topic. It can improve the system’s development efficiency and system’s stability.

Introduction

There are large numbers table and tree-structure in some management information systems. The information of the table is described directly by relational database. The information of the tree-structure is described indirectly only by relational database. Since the data structure has changed, it is clearly different from processing information of table and information of tree-structure.

Data is organized by the table structure in relational database. It is convenient and flexible to forms handling, and is easy to learn and use, and therefore widely is used. Data of the table is linear structure in relational database. Each row of the table corresponds to a data element, called a record. It is linear between record and record.

Tree-structure is different from Table. There are branches and hierarchy relational in tree-structure, it is very similar to the tree of the nature. It is a nonlinear data structure. There are large numbers tree-structure in the objective world, such as genealogy, administrative organizations, etc. They are organized by tree-structure. tree-structure has been applied widely in the computer field. For example, it is used to indicate the source of the grammatical structure in compiling programs and is used to organized information in the database system and is used to describe the process of the implementation in the analysis algorithm behavior.

At present, there are a lot of relational database software products, but which can support tree-structured query is rarely. It is essential that design tree-structure component in management information system. These tree-structure components are specially used to process and save information of tree-structure in relational database. It’s main function is that add information, delete information, search information, edit information, save information, output selected information, handle data of tree-structure and define data structure. The common tree-structure data representation and component design scheme is as followed.

Representation of Tree-Structure

Generally, there are three methods of representation tree-structure in management information system: field notation, code notation and static pointer notation.

Field notation is to use several fields represents nodes to hierarchical relationships in a tree. Management information system is relational database basically. Each record is stored one node information in relational database. Suppose you need n fields to storage nodes information. If the tree height is M, so that add M-1 fields in database. Starting from root, each layer corresponds one new field. In this way, each record contains N+M-1 fields in the database. In addition to store node itself information, also store the path which is from the root to this node. This path clearly expresses node hierarchy relation in the tree.

Code notation is to use one code represents nodes hierarchical relationships in a tree. In addition to store node itself information of N fields, only add one field to store hierarchy information in the
This field is called Code field. The code width is not less than the number of the level of the tree. Starting from the roots, each layer corresponds to one bit in code. It is root corresponds first bit in code,……., the number N layer corresponds number N bit in code.

Static pointer notation is to use a pointer to indicate nodes hierarchical relationships in the tree. There are pointer type in some high-level programming languages. But database languages have not pointer type. We can apply the "pointer" concept in the database. Suppose there are N fields store node itself information, add one field as a pointer field. This field is a numeric field and pointer using, is called static pointer. It is indicating the location of node parents stored in the database. Because each record of database corresponds to a record number, so the record number may express storage location.

Through the above comparison, the field notation is better than code notation and static pointer notation to describe hierarchical relationships in the tree. But it has a large number of redundant. Static pointer notation can solve the redundancy problem. But it can not describe intuitively hierarchical relationships in the tree and worse than field notation in this respect. Code notation is neutral, whether redundancy or intuitive.

**Component Design Scheme of Tree-Structure**

There are three component design scheme of tree-structure. The first is to use the design scheme of components in outside-process of server. This scheme is to achieve the management of tree-structure which has been designed in data services layer. The second is to re-design components of the data services layer, and look it upon as a special case. When designing alone a data services layer, component of tree-structure is designed directly in the data services layer (Figure 1). The third is to use control to design components of tree-structure.

If using the first scheme, it is added a layer between user interface and the database system. This user interface is used to manage data of the tree-structure. So, efficiency will be lower. If using the third scheme, generally, it will require a separate designed interface for each instance. Thus, the reusability of the code is not very high.

Here, using the second scheme. User components are provided to the application program by outside-process of server model. The advantage is following:

- It can be completely separated with the application program, so that it is conducive to system debugging.
- It is not necessary to design application program for each tree-structure. When using it, alone call outside-process of server.
- It may be packaged with other different components. Retain an interface to let the users set. But, this has increased the complexity of the component design.

![Figure 1. Components and structure design.](image-url)
Figure 1 shows that the application system is a multi-layered system in the logical structure. The component is divided into three parts: TreeManage component, clsTreeManage component, and TreeServer component.

The process is as follows:
First, to create a project that it is ActiveX.dll, to develop clsTree and clsConnect, and compile them, to produce TreeServer.
Then, to create a project that is ActiveX.Exe, to add TreeServer.dll, to develop clsTreeManage and frmTreeManage, and compile them, to produce TreeManage.Exe.
When using TreeManage, only create clsTreeManage, and set their attribute.
TreeServer.dll is used by other systems.

ClsTreeManage of Tree-Structure

FrmTreeManage
User interface include:
TreeView. It is Tree Controls. To realize data display and data drag-drop function in the tree-structure.
ToolBar. It is Toolbar Control. To realize data add, delete and update function in the tree-structure.
MSFlexGrid. It is Grid Control. In addition to parent node, keyword, title and level information in each node, but also may contain the node number, full path title and full path number, etc. Each node information is expressed by several fields. In order to adapt to this uncertain field, using the grid control to display and edit the node information.
Menu. It can provide a floating menu to achieve to add node and delete node operation.

ClsTreeManage
This is a tree structure management class. Its attributes are as follows:
ClsTree. Call tree structure table class. Using to receive an instance object in the data set.
Model. Call application of categorical variable. Using to set interface and application mode in the user Interface initialization process. This application mode is including edit, query, etc.
Selvalue. Using to set field operating rules in tree-structure data set. For example, when editing node, saving fields, or querying fields, according to node address or full path.
Initialize. Using to initialize the user interface.

Treeserver
This is a tree-structure service component. It includes two parts which is named clsTree and clsConnect. clsTree is called tree-structured table class, and clsConnect is called database connection management class.

ClsTree
Using ADO to realize data’s add, delete, save, query and other operations.
Node data structure
Public Type Node
Parent As Long The value of the parent node.
Key As Long Node key value.
Level As Integer Node number of layers.
Caption As String Node title.
End Type
Main attributes: the table name, the parent field name, keyword, field name, node title field name, the database connection objects, data result sets, keyword values, root node, etc.
Main methods: class initialization, add nodes, delete nodes, update the node, mobile node, query node points, etc.

ClsConnect
There provides an efficient model approach of processing data records set by ODBC in ADO and RDO. But ADO’s efficiency is less than RDO in visiting database. In order to improve system operating efficiency and reducing network load, it is necessary to combine ADO and RDO. Therefore, two types of access methods are provided for the selection of database operations in the database connection management class.

Main attributes: ADO Connection object, RDO connection object, user name, user password, data source name, database system name, server name and so on.

Main methods: Open ADO Connection to open the RDO connection.

Conclusion
There are not only relational model but also tree-structure model in management information system. But, Some information which expressed by tree-structure components is less designed in the management information system. This article’s innovative is that can easily complete it by specialized component. Their Advantage is that can complete organization function and classification function. And it can improve the system’s development efficiency and system’s stability.

Acknowledgement
This research was financially supported by 2013 Shandong data structure Quality Course, Number 2013BK110

References