

QT的模型视图结构

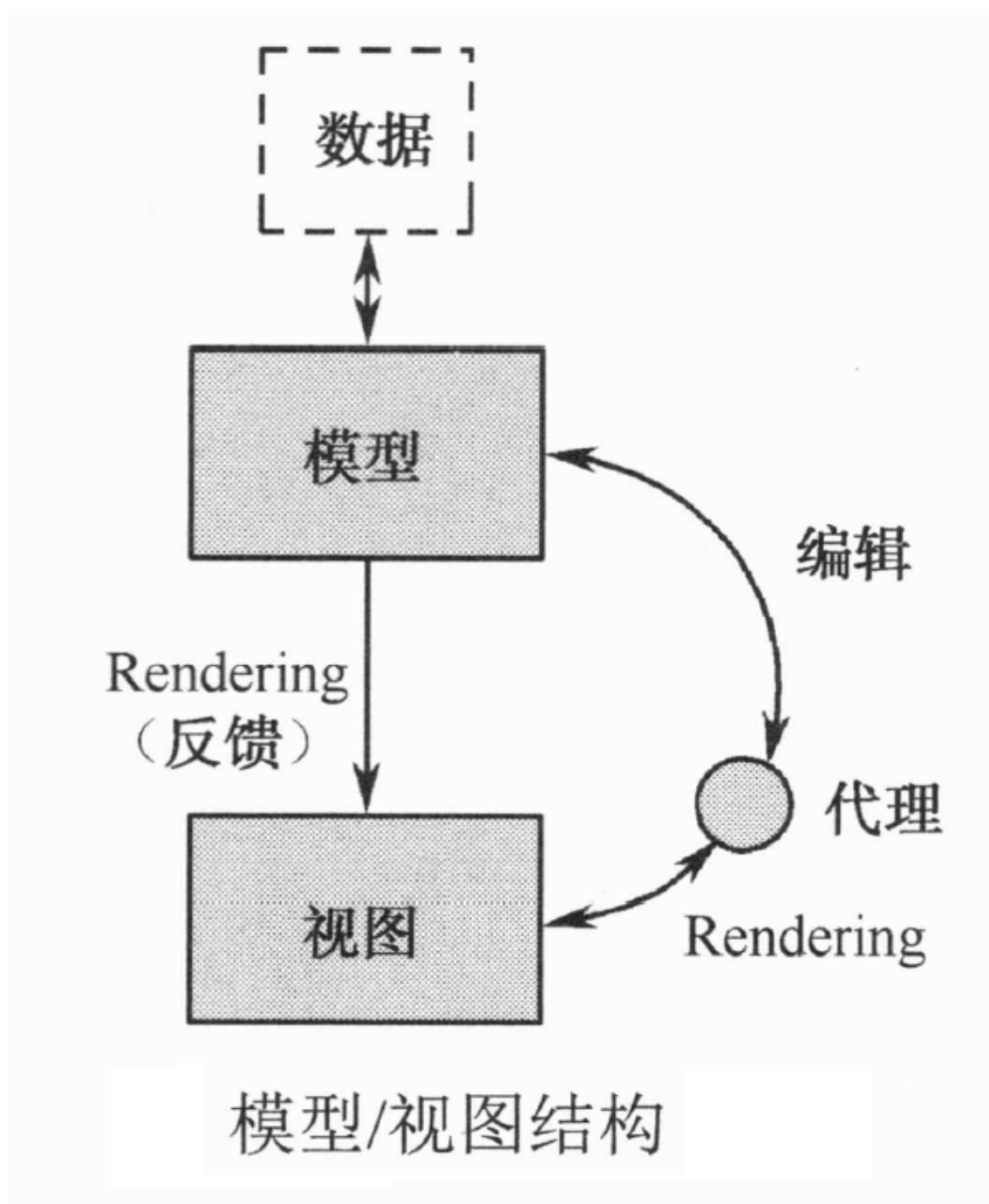
MVC 设计模式是起源于Smalltalk 的一种与用户界面相关的设计模式。通过使用此模式，可以有效地分离数据和用户界面。

MVC 设计模式包括三个元素：表示数据的**模型** (Model)、表示用户界面的**视图** (View) 和定义了用户在界面上操作的**控制器** (Controller)。

与MVC 设计模式类似，Qt 引入了模型/视图结构用于完成数据与界面的分离，即Interview框架。但不同的是，Qt 的Interview 框架把视图和控制器部件结合在一起，使得框架更为简洁。为了灵活地处理用户输入，Interview 框架引入了代理 (Delegate)。通过使用代理，能够自定义数据条目 (Item) 的显示和编辑方式。

Qt 的模型/视图结构分为三部分：模型 (Model)、视图 (View)和代理 (Delegate)。其中，模型与数据源通信，并为其他部件提供接口；而视图从模型中获得用来引用数据条目的模型索引 (Model Index)。在视图中，代理负责绘制数据条目，当编辑条目时，代理和模型直接进行通信。模型/视图/代理之间通过信号和槽进行通信，如图 所示。它们之间的关系如下。

- 数据发生改变时，模型发出信号通知视图。
- 用户对界面进行操作，视图发出信号。
- 代理发出信号告知模型和视图编辑器目前的状态



1 概述

简要地介绍Qt Interview 框架中模型、视图和代理的基本概念

1.1 基本概念

1 模型

Interview 框架中的所有模型都基于抽象基类 `QAbstractItemModel`, 此类由 `QProxyModel`、`QAbstractListModel`、`QAbstractTableModel`、`QAbstractProxyModel`、`QDirModel`、`QFileSystemModel`、`QHelpContentModel` 和 `QStandardItemModel` 类继承。其中, `QAbstractListModel` 类和 `QAbstractTableModel` 类是列表和表格模型的抽象基类, 如果实现列表或表格模型, 则应从这两个类继承。完成 `QStringList` 存储的 `QStringListModel` 类继承自 `QAbstractListModel` 类; 与数据库有关的 `QSqlQueryModel` 类继承自 `QAbstractTableModel` 类; `QAbstractProxyModel` 类是代理模型的抽象类; `QDirModel` 和 `QFileSystemModel` 类是文件和目录的存储模型。

2 视图

Interview 框架中的所有视图都基于抽象基类QAbstractItemView, 此类由QColumnView、QHeaderView、QListMew、QTableMew 和QTreeView 类继承。其中, QListView 类由QUndoView类和QListWidget 类继承; QTableView 类由QTableWidget 类继承; QTreeView 类由QTreeWidget 类继承。而QListWidget 类、QTableWidget 类和QTreeWidget 类实际上已经包含了数据, 是模型/视图集成在一起的类。

3 代理

Interview 框架中的所有代理都基于抽象基类QAbstractItemDelegate, 此类由QItemDelegate和QStyledItemDelegate 类继承。其中, QItemDelegate 类由表示数据库中关系代理的QSqlRelationalDelegate 类继承。

1.2 模型类/视图类

Interview 框架提供了一些可以直接使用的模型类和视图类, 如QStandardItemModel 类、QFileSystemModel 类、QStringListModel 类, 以及QColumnView 类、QHeaderView 类、QListView 类、QTableView 类和QTreeView 类等。

2 模型 (Model)

实现自定义模型可以通过QAbstractItemModel 类继承, 也可以通过QAbstractListModel 和QAbstractTableModel 类继承实现列表模型或表格模型。

在数据库中, 通常需要首先将一些重复的文字字段使用数值代码保存, 然后通过外键关联操作来查实的含义, 这一方法是为了避免冗余。

3 视图 (View)

实现自定义的View, 可继承自QAbstractItemView 类, 对所需的纯虚函数进行重定义与实

现, 对于QAbstractItemView 类中的纯虚函数, 在子类中必须进行重定义, 但不一定要实现, 可根据需要选择实现。

4 代理(Delegate)

在表格中嵌入各种不同控件, 通过表格中的控件对编辑的内容进行限定。通常情况下, 采用这种在表格中插入控件的方式, 控件始终显示。当表格中控件数目较多时, 将影响表格的美观。此时, 可利用Delegate 的方式实现同样的效果, 控件只有在需要编辑数据项时才会显示, 从而解决了所遇到的上述问题。