

# WEB 动态报表实现中的参数化过滤技术<sup>①</sup>

## Parametric Filtration Technique in Web Dynamic Reports Implementation

马瑞敏 王成良 (重庆大学 软件学院 重庆 400030)

**摘要:** 报表作为信息组织和分析的一种有利手段,是企业管理信息系统的重要组成部分。目前基于 B/S 结构的 Web 报表开发已成为主流,出现了多种 Web 报表生成工具,如水晶报表、ActiveReport、QuickReport 等,但都存在着开发效率和质量不甚满意的情况。本文通过在 .NET 平台下,对 Report Viewer Control 客户端报表工具和 Report Viewer 控件的研究,提出了一种通过参数化过滤技术来生成动态 Web 报表的方法,此方法减少了网络数据传输量,提高了报表生成效率,提升了系统性能,为实现复杂的 Web 动态报表提供了参考的解决方案。

**关键词:** WEB 报表 数据过滤器 参数化 report viewer control rdlc

随着 Internet 的迅速发展,企业管理信息系统逐步从传统的 C/S 模式向 B/S 模式转变,而报表作为信息浏览、分析和打印的有利工具,是企业管理信息系统中必不可少的重要环节,Web 报表的开发已经成为企业信息化的一项重要内容。

### 1 当前报表开发研究

目前已出现很多 Web 报表开发工具,国外著名的报表工具软件主要有 Business Objects 公司的水晶报表、Data Dynamics 公司的 ActiveReport、Qusoft 公司的 QuickReport 等,其中又以水晶报表为代表。它是报表软件中较为专业、功能较强的一种报表工具,它除了具有强大的报表核心功能之外,最大的优势是提供了接口,实现了与绝大多数流行开发工具的集成。自从早期集成到微软公司的 .NET 开发平台之后,水晶报表几乎成为开发人员在制作报表时候的主要选择。然而 Web 版的水晶报表也存在着许多不尽如人意的地方,它虽然支持复杂报表结构的设计,但是本身的设置和使用等也相当复杂;自定义性较差,数据处理比较麻烦;此外还存在版权问题,费用比较昂贵。ActiveReport 与 QuickReport 也都存在着使用复杂、设计繁琐等缺点。

Report Viewer Control 是微软在 Visual

Studio.NET(以下简称 VS.NET) 2005 版本中新加入的报表开发工具。Report Viewer Control 这一客户端报表工具与 Report Viewer 控件、报表定义语言(RDL)高度集成在一起,既提供可视化开发的工具和组件,又有核心语言的支持。这种报表文件的后缀名规定为 RDLC,其中 RDL 是报表定义语言即 Report Definition Language 的缩写,而 C 代表客户端处理功能即 Client-side Processing,强调了客户端的报表处理能力。RDLC 报表与水晶报表相比能更好地和 VS.NET 集成在一起,更加简单易用,数据处理功能也更强大。

传统的报表设计是静态的,即报表的内容和布局是不能改变的,如果用户以稍微不同的方式查看报表,报表设计者就需要为每个特定需求重新建立一份报表,这加大了系统的开发和维护成本。而在 B/S 模式下,Web 报表的实现会遇到更多问题,例如数据访问的速度、报表生成的效率、报表数据的动态性、报表能否传递参数等。因此克服 B/S 模式下报表设计的诸多局限性,建立灵活的报表以满足更大范围内的用户需求,使静态的报表尽可能数据动态化和形式动态化,不仅可以提高用户满意度和报表实际使用价值,而且也可提高报表在不同客户中的重用性,降低开发量,提高开发效率和降低维护成本。为此,本文通过在 .NET 平台下对 Report Viewer Control 客户端报

<sup>①</sup> 收稿时间:2008-08-21

表工具和 Report Viewer 控件的研究, 提出了一种通过参数化数据过滤器技术来生成动态 Web 报表的方法, 克服了 B/S 模式下报表设计的局限性, 为实现复杂的 Web 动态报表提供了参考的解决方案。

## 2 WEB报表的实现原理

基于 B/S 结构的 Web 应用一般采用三层体系结构, 即客户端/Web 服务器/数据库服务器。第一层客户端为表示层, 用于界面引导, 接受用户输入并向 Web 服务器发送请求, 显示处理结果; 第二层 Web 服务器为业务逻辑层, 该层处理应用中的所有业务逻辑, 包括对数据库的访问等工作; 第三层为数据中心层, 主要由数据库系统组成, 执行数据逻辑, 运行 SQL 语句或存储过程。数据由数据库管理系统在后台统一管理, 客户端在浏览器提交服务请求后, 经过 Web Server 响应, 并通过 DBMS 处理对数据的请求, 处理结果由 Web 服务器编译成 HTML 传给客户端, 而 HTML 则是所有客户端浏览器都能识别的文档。基于以上 web 应用运行机制的特点, 本文所提出的 Web 动态报表解决方案如图 1 所示:

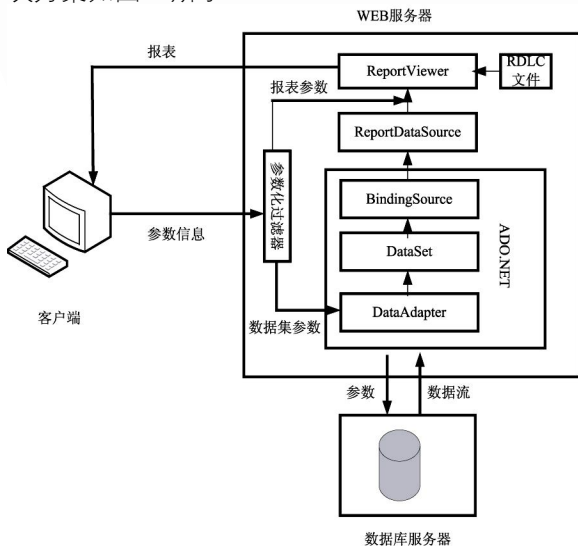


图 1 WEB 动态报表解决方案

用户在客户端向 Web 服务器递交报表请求并发送参数信息, Web 服务器中通过参数化过滤器来识别是报表参数还是数据集参数, 数据集参数提供在数据库服务器端进行数据过滤的条件, 报表参数实现对报表级别数据的过滤, 通过用户输入的参数值来实现报

表内容的动态显示, 最终组合报表定义和报表数据, 为用户返回一个格式化的输出。

### 2.1 参数化数据过滤技术

在几乎所有情况下, 报表都是以某种数据源为基础。RDLC 报表可以使用 ADO.NET 技术从 .NET Framework 支持的任何标准数据提供程序处获得数据。除了 SQL Server, 还包括 Oracle、Access、Excel、Informix、DB2 和其他通过 OLE DB 提供程序或者 ODBC 驱动可以访问的数据源。每个报表至少有一个数据源, 报表可通过单一数据集的数据源来获取数据, 也可通过多个数据集的多个数据源来获取数据。数据源提供了连接到数据库的能力, 数据集包括了一个为报表产生数据的查询表达式。

很多数据库都含有大量数据, 因此, 高效快速的获取报表所需要的恰当数据量, 无需每次生成报表时都重新查询数据库是很重要的, 这就需要通过参数实现。RDLC 报表设计过程中可以声明两种不同类型的参数: 数据集参数和报表参数。数据集参数源于数据库对象, 比如存储过程和用户定义函数, 报表参数应用于报表级别的数据。

数据集参数提供了数据在数据库服务器端进行过滤的条件, 经过过滤后只有少量数据传递到 Web 服务器端。通过传递数据库对象级别的过滤标准参数, 网络传输量可以大大降低, 报表显示也会更加高效。但是在参数可选值很多的情况下, 有可能在很短时间内返回了同一个报表的几个版本, 那么将会重复查询数据库, 也许会导致很长的等待时间和相同数据多次在网络中传输。如果在数据库服务器端没有对数据进行过滤, 将会返回大量的数据, 其中也许有很多不需要的数据, 为了获得报表所需要的数据要使用报表参数在 Web 服务器端进行过滤, 加重了 Web 服务器的负担。

基于以上考虑, 结合特定的报表需求, 本文提出参数化数据过滤器技术, 对数据进行双层过滤, 如图 2 所示。第一层过滤是通过数据集参数在数据库服务器端进行, 返回报表所需要的适当的数据量而无需每次生成报表时都重新查询数据库; 第二层过滤是通过报表参数在 Web 服务器端进行, 因为第一次过滤后得到的所有的数据都被缓存在 Web 服务器中, 通过报表参数过滤得到报表最终要显示的数据。这样既可以降

低网络传输量，报表显示也会更加高效。

数据库服务器上使用参数化的存储过程一般是筛选数据最为有效的途径，通过使用存储过程可以减轻

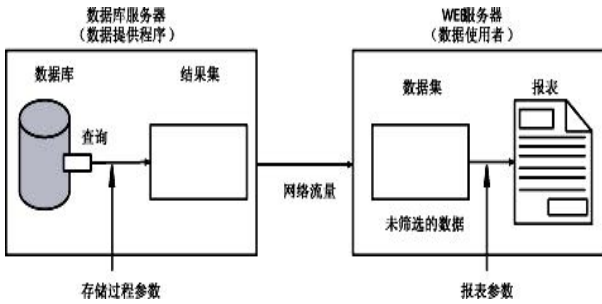


图 2 参数化数据过滤器结构图

网络通信量，提高数据检索速度，保护数据库的安全。参数的处理影响性能、效率和报表解决方案的灵活性，做出正确的选择和找到这些选项的最佳组合可以降低系统负荷，提高系统性能和报表的灵活性。

### 2.2 实现步骤

假设存在数据库 MyDb，包含表 Category，ProductSubCategory，Product，表间关系如图 3 所示，要求实现按子类别查看产品详细信息报表。我们将为报表使用产品类别参数和子类别参数。类别参数实现在数据库中筛选数据，得到一个范围较大的结果集；子类别参数实现在作为报表数据源的数据集中筛选数据（在 Web 服务器上），当子类别参数的值发生改变时不用重新访问数据库，从而得到报表最终将要显示的信息。

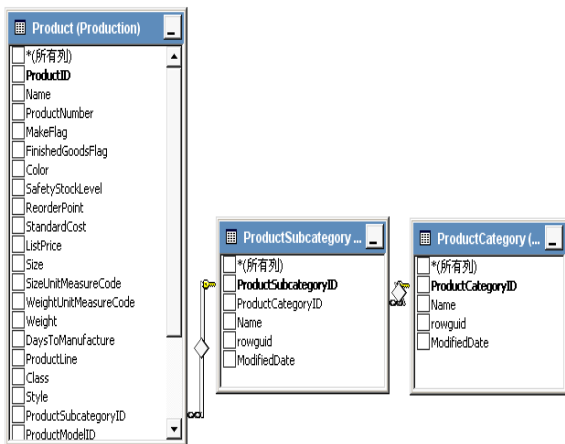


图 3 表间关系

### 2.2.1 WEB 界面设计

在 VS.NET 2005 中新建解决方案，并创建如图 4 所示的 Web 页面，用于接受用户输入参数值，实现按产品类别和子类别显示产品信息报表的功能，当用户的选择发生变化时，动态的显示报表内容。

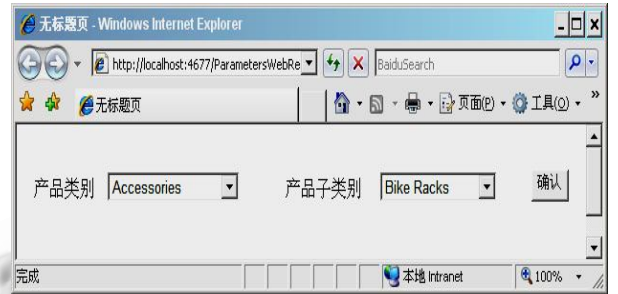


图 4 WEB 界面设计

页面包含两个 DropDownList 控件、一个 Button 控件，一个 ReportViewer 控件，其中 ReportViewer 的 Visible 属性为 False。为下拉列表框控件绑定数据源，分别显示类别列表和子类别列表，设置当产品类别发生改变时相对应的子类别也随之改变。

### 2.2.2 RDLC 报表设计

(1) 在数据库中创建带参数的存储过程，参数的可选值为产品类别值，该存储过程实现查询同一类别产品信息的功能。

(2) 在解决方案资源管理器中，添加新项，新建 RDLC 报表文件。

(3) 新建数据集文件，设置数据源将要使用的数据库连接。选择使用刚刚新建的存储过程获得数据库中的数据，建好后这个数据集就可以作为报表的数据的来源。

(4) 将数据源窗口中刚刚建好的数据集的字段拖动到在报表设计器中，如产品名称、颜色、类别、子类别等，设计报表。

(5) 创建报表参数对数据集中的数据进行过滤，先为报表添加子类别报表参数，要注意报表参数的数据类型要与数据库中该字段的数据类型相同，然后在报表项 table 的属性中设置过滤表达式，使得子类别字段的值等于报表参数值。

(6) 报表完成以后，回到 Web 页面，将新建的本

地报表分配给 ReportViewer 控件。

### 2.2.3 主要功能代码设计

在 Web 窗体的源文件中引入命名空间 System.Data.SqlClient 和 Microsoft.Reporting.WebForms。单击“确定”按钮时，触发响应事件，其具体过程描述如图 5 所示：

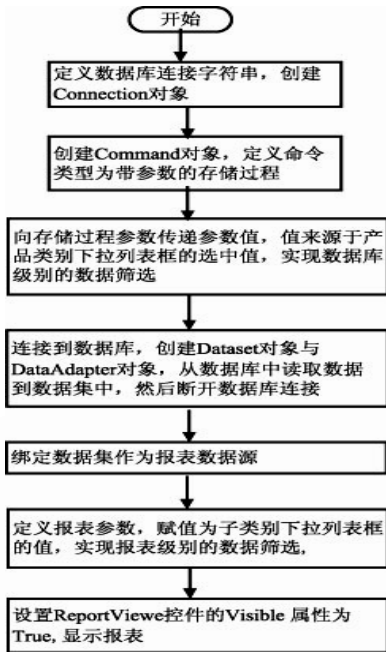


图 5 实现过程

编译该解决方案，运行结果如图 6 所示。

### 3 结束语

本文介绍了在 Visual Studio .NET 环境下，利用 Report Viewer Control 客户端报表工具和 Report Viewer 控件，通过参数化数据过滤来实现 Web 报表的动态生成，数据访问速度快、安全性高，降低了系统负荷，提高了系统性能和报表的灵活性，克服了 B/S 模式下报表设计的诸多局限性。同时，这种设计方法极

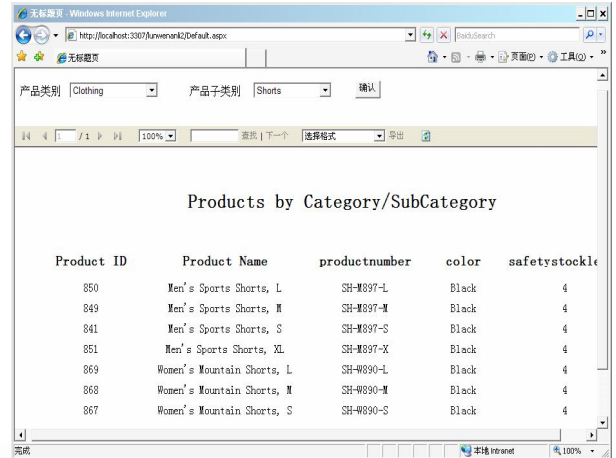


图 6 运行结果

大地提高了开发效率，使得开发人员可以从一些底层繁琐的编程任务中解脱出来，专注于实现用户特定的要求，是一种较好的动态 Web 报表解决方案，为更多复杂的报表需求提供了参考。

### 参考文献

- 1 Ullman C, Goode C. Beginning ASP.NET Using C#[M], Wrox Press Ltd, 2001: 383 - 446.
- 2 Turley P, Bryant T, Counihan J, DuVarney D. Professional SQL Server 2005 Reporting Services. Wiley Publishing, Inc, 2006: 110 - 137.
- 3 Gunderloy M, et al. 曲利君等译. SQL Server 2005 从入门到精通. 北京: 电子工业出版社, 2006.
- 4 飞思科技产品研发中心. NET 平台下 Crystal Reports 设计与开发. 北京: 电子工业出版社, 2003.
- 5 张能力. WEB 动态报表的实现. 计算机应用与软件, 2004, 21(4): 110 - 112.
- 6 Larson B. 用 ReportViewer 控制你的报表. Windows IT Magazine 国际中文版, 2007, 7: 77 - 79.