

基于 Web 的信息可视化系统的设计与实现^①

Design and Implementation of a Web-Based Information Visualization System

张 驰 罗铁坚 王相根 (中国科学院 研究生院 信息科学与工程学院 北京 100049)

摘 要: 可视化技术的发展能够帮助我们加深对大规模复杂数据和信息间相互关系的理解, 因此人们针对不同领域开发出各种功能的可视化系统。以高等级生物安全实验室协同工作平台 CBL 项目为背景, 根据生物安全实验室环境监控和生物安全知识库的可视化需求, 设计并开发了 CBL 信息可视化系统。该系统采用基于 Web 的 B/S 三层架构, 利用 MVC 设计模式, 根据需求分别采用合适的可视化模型开发可视化组件, 并根据可视化界面与用户交互的不同方式采用相应的 Web 客户端技术, 方便用户更加高效地获取和理解信息。

关键词: 可视化模型 组件 Web 可视化 可视化系统 MVC

1 引言

随着浏览器/服务器架构(B/S)的出现, 服务器端和客户端技术的不断成熟, 使得对基于 Web 的信息可视化系统的研究和开发变为可能^[1,2]。

2003 年, 国家科技部提出了“国家科技基础条件平台”战略计划。作为该战略计划的重要组成部分, 高等级生物安全实验室协同工作条件平台 CBL (Collaboration for BSL-3 Labs) 构建了一个为 BSL-3 实验室协作研究、组织管理和评价评估的综合信息化平台^[3]。本文针对 CBL 项目中环境监控和生物安全知识库两个可视化需求, 设计并开发了 CBL 信息可视化系统 CBLViz。该系统基于 Web 的 B/S 架构, 利用 MVC(Model-View-Controller)设计模式, 根据需求采用合适的可视化模型开发可视化服务组件。用户通过浏览器访问系统, 通过友好的界面和简单的交互查看所需要的信息, 方便并且高效。

2 系统结构

CBL 信息可视化系统采用了基于 Web 的三层 B/S 架构。从系统结构图(图 1)中可以看到其包括客户层、应用服务器层和数据服务器层。

2.1 客户层

这里的客户层主要由浏览器可以解释执行的页面组成, 也即系统与用户交互的界面。用户可以在客户层输入数据, 提交请求, 获得可视化图形结果, 并与之交互, 比如进行拖拽, 缩放, 检索等操作。根据不同需求本文采取了不同的技术, 比如 Applet 或者 RIA(Rich Internet Application)技术, 将在第 3 节重点介绍。

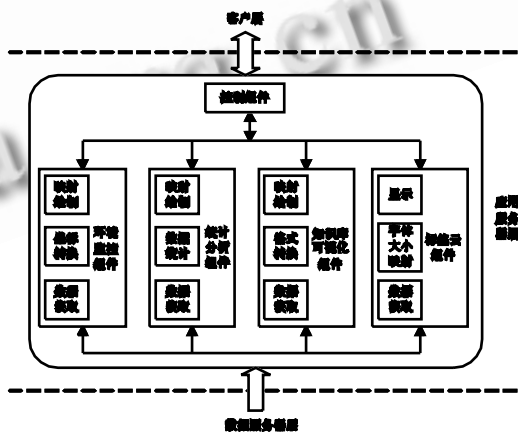


图 1 系统结构图

2.2 应用服务器层

应用服务器层的主要功能是接受来自客户层的请

^① 基金项目: 国家科技基础条件平台项目(2005DKA64100, 2005DKA10201)

收稿时间: 2008-03-12

求，从数据库层获取所需的信息数据，采用一定的可视化模型，将数据转换为图像传递给客户层。如图 2 所示，可视化模型是指将数据源通过过滤、映射、绘制处理流程后转变为图像的过程，即可视化流水线 (Visualization Pipeline)^[4]。由于可视化需求的不同，在模型的实现上采取了不同的方式，将在第三章中介绍。应用服务器主要由控制组件和可视化服务组件构成。我们采用了 Tomcat 作为系统的应用服务器。

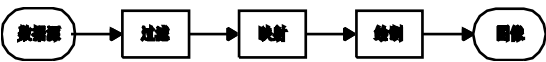


图 2 可视化模型图

2.3 数据服务器层

数据服务器层负责存储和提供应用服务器层所需的数据信息。主要包括各个生物安全实验室的环境数据和生物安全知识库中的文章、资料等信息。这些数据主要源自三个方面：1)各个生物安全实验室的传感器收集实验室的温度、压力等环境数据。2)生物安全知识库的专家和用户向知识库添加文章、评论等数据。3)CBL 平台中的网络爬虫自动抓取生物安全领域的新闻、博客等资讯。系统采用了 MySQL 作为数据库管理系统。

整个系统结构采用了 MVC 的设计模式，其中 Model 层由可视化服务组件充当，负责完成应用逻辑，提供可视化服务。View 层为表现层，在系统结构中为客户层，而 Controller 由控制组件实现，负责接受用户请求，将请求转发到相应可视化服务组件，并选择合适的可视化视图返回给用户。这样的系统结构有三个优点：1)应用逻辑与表现层完全脱离，便于独立开发和修改；2)采用可视化服务组件方式利于系统扩展，如果有新的可视化需求只需开发新的组件，原先的系统不受影响；3)用户无需安装客户端软件，利用浏览器即可访问，同时也可以进行一定的交互操作。

3 功能组件

CBL 信息可视化系统主要分为五个功能组件：控制组件、环境监控组件、统计分析组件、知识库可视化组件、标签云组件。

3.1 控制组件

控制组件在整个系统中充当 Controller 层。它的功能接受来自客户层的 Http 请求，根据请求参数调用

相应的可视化服务组件，这里的请求参数还包括可视化服务组件需要的配置信息，比如需要显示的图形元素数量，大小等等。当可视化服务组件处理产生可视化图像后，控制组件选择相应的视图返回给用户。如果访问的服务不存在或者提交的参数有误，控制组件会调用预先定义好的错误处理页面返回给用户。控制组件利用 Java Servlet 技术实现。图 3 反映了控制组件、客户层和可视化组件的交互过程。

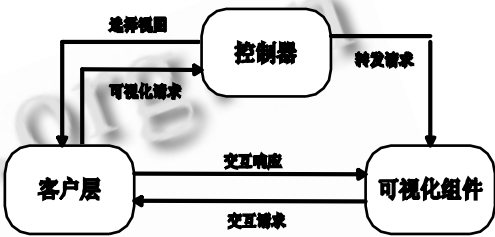


图 3 控制组件交互图

3.2 环境监控组件

环境监控组件的功能是让用户能够实时查看各个生物安全实验室的环境状况，以帮助他们进行实验计划和安排。由于需要实时查看环境数据，传统的静态页面视图已经不能够满足这个需求，因此我们采用了图 4 所示的基于 Web 的可视化参考模型，可视化 Web 服务器经过过滤、映射后，输出一定格式的可视化执行脚本，返回给用户。客户端利用支持相应脚本的浏览器来绘制、操纵可视化图形^[5]。该模型的优点在于客户端执行了一部分可视化任务，减轻了服务器压力，同时使可视化界面具备一定交互功能。

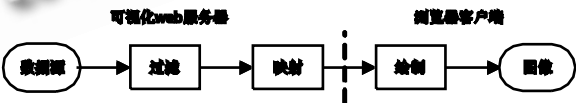


图 4 环境监控组件可视化模型图

该组件主要由三个模块构成：数据获取模块、数据转换模块和绘制模块。数据获取模块每隔三秒向数据库获取环境数据，数据转换模块根据坐标轴在可视化页面中的几何位置对其进行计算，将环境数据值转换为相应的坐标值。绘制模块负责整个可视化界面的绘制。我们采用了 OpenLaszlo，一种富 Internet 应用平台作为实现该组件的平台。用户在浏览器中安装 Adobe Flash Player 插件即可访问环境监控页面，图 5 为环境监控可视化页面的效果图。坐标横轴表示时

间，纵轴表示环境数据值，三条压力曲线从上而下分别表示实验室核心区、半污染区、清洁区的实时压力。

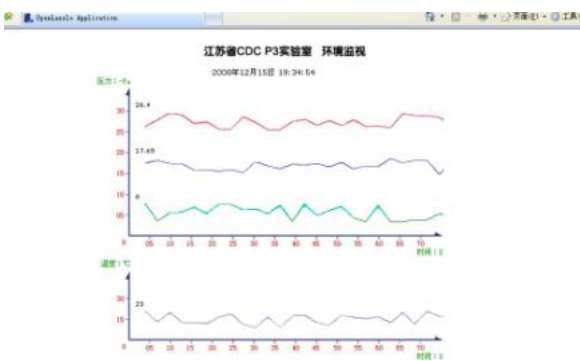


图 5 环境监测可视化效果图

3.3 统计分析组件

统计分析组件的作用是对某个实验室一段时间内的环境监测数据做统计，用折线图和面积图的方式表现某个实验室一段时间内异常环境数据发生的时间总和以及比例分布，以便实验室人员及时的发现实验室出现的异常状况。

由于实验室的传感器每三秒采集一次环境数据，每个实验室同一种环境数据一年大概有一千万条数据，对如此量级数据的计算需要强大的计算资源。同时组件产生的可视化结果是静态的图表，不需要交互，因此选择图 6 所示的基于 Web 的可视化参考模型。该模型中服务器端执行可视化流水线上的所有步骤，只返回给用户图像或视频，客户端用浏览器显示，用户与可视化过程没有交互[5]。

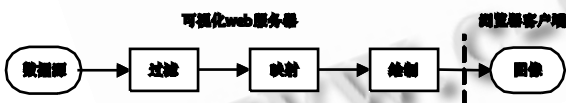


图 6 统计分析组件可视化模型图

该组件有两个核心模块：统计模块和绘制模块。统计模块从用户接受统计分析配置参数，包括实验室名称、统计数据类型(压力或者温度)、时间范围(月份、季度或者年度)，并从数据库中查询异常数据的条数从而计算发生时间。这里出现的一个问题是数据表中的环境数据条目是千万级，查询所需的时间远远超过用户等待服务器响应所能忍受的时间，因此我们对数据库设计做了优化。通过调研得到用户主要对每个月的数据进行统计，所以将原先设计的一张数据表格按照

月份进行拆分，这样表的数据降了一个量级，使得大部分查询操作的数据量减少。然后对环境数据字段加上索引，这是因为统计时对表的查询操作主要基于对环境数据字段的比较。优化后的效果十分明显，原先统计一个月的数据需要一分钟，优化后只需一秒钟。绘制模块获取统计模块生成的时间数据后，根据用户选择的图表类型(折线图或面积图)生成相应的图表。我们采用基于 Java 的第三方绘图类库 JFreeChart 实现。图 7 和图 8 分别显示了 2008 年 8 月江苏省 CDC P3 实验室温度监控时间统计的可视化结果。



图 7 温度监控时间折线统计图

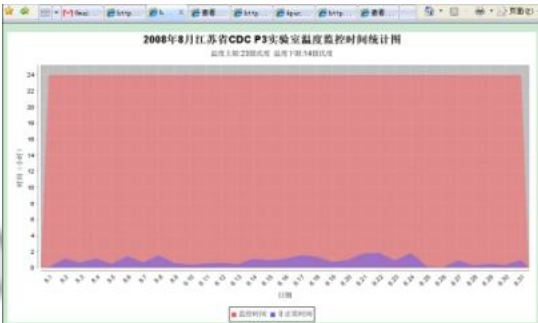


图 8 温度监控时间面积统计图

3.4 知识库可视化组件

开发知识库可视化组件的目的是能够让用户通过可视化的形式快速了解知识库中知识的组成结构，文章的主题分布以及高效地进行文章的查找。根据这个需求我们设计和开发了如图 9 所示的知识库可视化界面。它的功能主要有三个：1)用户可以通过点击某个文章目录结点查看其子目录结点或者该目录下的文章结点，当文章结点被选中时，该文章的类别路径将高亮显示。2)用户可以搜索标题包含指定关键字的目录或者文章，搜索结果中的结点将被自动展开并高亮显

示。3)由于页面大小有限,组件增加了缩放、平移的交互操作,同时每次用户操作后组件都会重新调整布局,将用户关注的部分放置到界面中心。

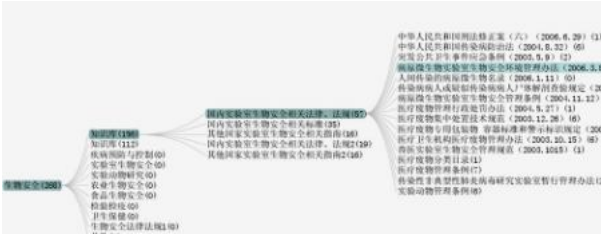


图 9 知识库可视化界面

由于该组件提供较多的用户交互操作,如果每次操作都和服务器通信,对网络通信要求太高,同时用户也不能忍受过长的响应时间。因此我们采用了图 10 所示的可视化模型。用户下载数据,在客户端执行可视化流水线[5]。



图 10 知识库可视化组件可视化模型图

组件以 Applet 形式实现,主要有两个模块:数据转换模块和绘图模块。数据转换由 DataTrans.java 类实现,它基于开源类库 Dom4j 将数据库中的目录和文章标题按照层次关系转换为 XML 格式的数据形式(如图 11)。其中 branch 元素表示目录结点,leaf 元素表示文章结点。绘图模块由 TreeView.java 类实现。它接收 DataTrans.java 生成的转换数据,利用 TreeView 的布局算法,将目录和文章结点组织成为树形结构展示,同时支持一定的用户交互。该类利用了基于 Java 的开源图形类库包 Prefuse 实现,并融合了开源文本搜索引擎 Lucene 实现了搜索功能,以 Applet 的方式嵌入在网页中发送给用户,用户下载代码后在客户端运行。图 12 用 UML 描述了 TreeView 类与其他辅助类的关联图。

```
<tree>
<branch>
  <attribute name="branchname" value="branchvalue"/>
  <leaf>
    <attribute name="leafname" value="leafvalue"/>
  </leaf>
</branch>
</tree>
```

图 11 知识库可视化数据格式

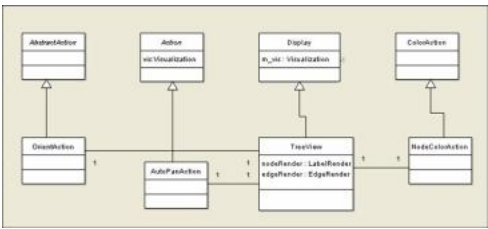


图 12 TreeView 类的关联图

3.5 标签云组件

标签云是一组拥有权重的标签集合,通过字体的大小或者其他可视化形式来表现权重的不同,同时提供一定的用户交互,用户可以通过点击标签的超链接获得数据。我们将其也应用在生物安全知识库中。生物安全知识库中的标签是由用户和专家建立,文章的标签反映了该文章的主题、类别等信息,用户通过标签可以找到主题相关的文章和资料。标签云组件以标签的访问次数作为权重设置标签的字体大小,以反映近期倍受关注的热门文章类别。

组件用标准 Java Bean 格式的 TagBean.java 类存储标签名、建立时间、标签链接等信息,同时利用 Java 内置的数据类型 HashMap 存储标签标题和其对应的 TagBean 对象。利用 HashMap 的优点包括:1)将标签和其属性信息相关联,方便通过标签查找与标签主题相关的信息。2)HashMap 的查找十分高效,当标签数量很大时,组件也能够在用户能够忍受的时间范围内返回用户请求的标签信息。实现标签云组件的关键是如何将标签被点击的次数合理地映射为标签字体大小上。通过对知识库中所有标签的点击次数统计后发现,点击次数并不是均匀分布,类似于正态分布。我们采用如下的线性化算法,将点击次数通过线性化计算后均匀地分布在一定的字体大小区间内。算法描述如下:

$$Sd = \sqrt{\frac{\sum_{i=1,n} (hitNum_i - meanNum)^2}{n}}$$
 (1)

$$Slope = \frac{(maxSize - minSize)}{4 \times Sd}$$
 (2)

$$FontSize_i = Slope \times (hitNum_i - meanNum) + midSize$$
 (3)

其中, hitNum_i 表示标签 i 的点击次数, n 表示知识库中标签的总数, meanNum 表示所有标签的点击

平均数, `maxSize` 和 `minSize` 分别表示标签字体大小区间的最大值和最小值, `midSize` 表示字体大小的中间值。算法首先计算所有标签点击次数的平均差 `Sd`。通过统计得到所有标签点击次数的偏离误差在 $-2 * sd$ 与 $2 * sd$ 之间, 因此根据公式(2)计算线性斜率 `Slope`, 最后根据点击平均数、字体中值以及斜率得到的线性方程计算标签 `i` 的字体大小值 `FontSizei`。

由于页面大小有限, 不能显示知识库内所有的文章标签, 标签云组件可以根据用户选择的文章创建时间参数和显示数量参数来显示标签云结果, 图 13 显示了标签云可视化页面, 用户可以通过点击标签查看相应主题的文章和资讯。



图 13 生物安全知识库标签云可视化界面

4 结语

CBLViz 信息可视化系统的结构良好, 利于扩展和维护, 已经投入使用。我们下一步的工作是在 CBL 协

作平台中发现更多的可视化需求, 为系统设计和开发相应的可视化服务组件, 增加更多的功能。同时系统的负载承受能力有限, 随着用户增多和数据量不断增大, 性能会下降。因此我们考虑利用负载均衡技术和数据库优化技术对系统进行优化以提高系统的性能。

参考文献

- 1 张文, 李晓梅. 基于 Web 的可视化研究与实现. 计算机工程与科学, 2002, 24(3): 25 – 27.
- 2 Bender M, Klein R, Disch A, Ebert A. A functional framework for Web-based information visualization systems. Visualization and Computer Graphics, 2000, 6(1): 8 – 23.
- 3 Nathan H, Burkhard W, Ewan T. A framework for interactive Web-based visualization. Proc. of the 7th Australasian User Interface Conference. Hobart, Australia, 2006. 137 – 144.
- 4 Wood J, Brodlie K, Wright H. Visualization over the world wide Web and its application to environmental data. Proc. of the 7th Conference on Visualization. San Francisco, California, United States, 1996. 81 – 86.
- 5 刘俊, 迟学斌, 单桂华. 一种基于 Web 的远程可视化系统的设计与实现. 中国科学院研究生院学报, 2006, 23(6): 782 – 786.