

基于 SOA 的报表设计器^①

关宁^{1,2}, 赵杨^{1,2}, 王俊霖³

¹(中国科学院大学, 北京 100049)

²(中国科学院沈阳计算技术研究所, 沈阳 100168)

³(大连理工大学, 大连 116024)

摘要: 研究了面向服务的 Web 报表设计器, 通过提供服务的方式保障系统功能需求. 设计器作为一个粗粒度的服务, 由多个细粒度的子服务构成, 通过转换接口的方式为不同业务系统提供模板设计服务. 使用拖拽控件的方式, 降低开发难度, 便于使用和维护. 用 XML 格式保存和设计模板文件, 改变映射规则即可快捷地转化模板文件的格式以便导出. 解决了当下面向对象、面向集成、面向组件等紧耦合架构模式下报表设计器灵活性差、重用性低的问题, 具有易用、交互、并发、可扩展等特点.

关键词: SOA; 报表设计器; 报表模板; SOMA

Report Designer Based on Service Oriented Architecture

GUAN Ning^{1,2}, ZHAO Yang^{1,2}, WANG Jun-Lin³

¹(University of Chinese Academy of Sciences, Beijing 100049, China)

²(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

³(Dalian University of Technology, Dalian 116024, China)

Abstract: The report designer of facing service is based on web and meets the system requirements through providing service. Act as coarse-grained services, this report designer is composed of a plurality of fine-grained sub service, providing a templates-designed service for different business system by converting interface. It reduces development difficulty by dragging component to use. And it is convenient for using and maintain. The report designer uses XML to save and access the template files, and it is quick to change the format of template files through changing the mapping rules to provide convenience for exporting. The new designer solves the problem of inflexibility and non-reusability of the traditional designer which is based on Object-Oriented or Integration-Oriented or Component-Oriented. And the new designer also has the feature of accessibility, interactivity and expansibility.

Key words: SOA; report designer; report template; SOMA

在数据信息急速扩张的今天, 报表已经成为一种信息分析和组织的有力手段, 是进行信息浏览、分析、输出的有力工具, 是电子政务、企业信息系统重要组成部分^[1]. 然而, 由于报表数据的海量化、多样化、多变性等特点, 传统的基于 C/S 体系架构的报表开发工具显然已经无法满足当前信息的分类、浏览、分析、统计等工作, 其滞后的更新速度也成为数据飞速增长变化下报表统筹工作的掣肘之处. 因此, 发掘新概念下的报表设计器和引擎, 提高报表开发的效率和数据统

计的质量已经成为当前报表开发工作的重要内容之一. 常见的报表工具, 设计器往往只是内嵌于报表系统的一个功能组件, 用以完成一些简单的报表展示和模板设计工作, 本文开发设计的面向服务的 Web 报表设计器是独立于业务系统之外, 能够通过转换接口为不同的业务系统提供丰富报表模板设计服务的设计器. 该设计器结构上具有灵活性高、易扩展、可并发等特点, 功能上则有简洁易用、便捷高效、二次编码率低等特点, 可以有效提高报表设计的效率和质量.

① 基金项目: 国家水体污染控制与治理科技重大专项课题(2012ZX07505004)

收稿时间: 2016-05-16; 收到修改稿时间: 2016-06-16 [doi: 10.15888/j.cnki.csa.005590]

1 SOA介绍

面向服务的体系架构 (Service Oriented Architecture, SOA)是继面向过程、面向对象、面向框架之后出现的新的软件架构体系。基于紧耦合的系统架构在软件的开发过程中,系统模型与系统实现无法分离,这使得当系统需求发生变更时,这些模式无法快速变更结构以达到要求,灵活性较差。因此,在系统需求多样化的前提下,一种全新的、松耦合的体系架构成为大势所趋,SOA 也就此逐渐发展壮大起来。

SOA 使用 Web 服务来实现,包含了自制的、高服务质量、多样性的、易操作的、易发现的和潜在可复用的服务^[2]。

本文设计的基于 SOA 的报表设计器可以通过转变接口和协议的方式,有效的为不同的业务系统提供报表设计服务,从而可以适应不同的数据种类、不同数据量、不同报表模式需求的系统需要,既跳出传统内嵌于系统的报表设计模块的约束模式,灵活地更改服务,又避免了 C/S 模式下升级难,更新慢的问题。

2 系统设计

2.1 SOMA 的分析方法

SOMA(Service Oriented Modeling and Architecture, 面向服务的建模与体系结构),是 SOA 发展过程中的最新成果,是基于服务的建模以及针对服务的分析和设计的方法及理论^[3]。首先使用自顶向下的分析方式对业务系统进行领域和流程的分解,从而发现并鉴别服务,然后使用自底向上的分析对业务系统的现有资产进行分析和重划,将组件和应用程序分为变化的部分和不变的部分,变化的部分进行重构和新建,不变的部分作为候选方案从而减少重复开发,提高代码复用率。最后由中间向外,使用目标-服务建模方式发现和验证前两者未发现的服务。而后对服务进行分级和分类,划定服务的粒度和层次结构,并进行子系统分析,确定不同层次的子系统服务类型和相互之间的关系,同时,进行服务分配,将服务指派到目前已鉴别出来的子系统上去,最后根据相应的服务规范进行服务的实现。

2.2 报表设计器的 SOA 化

2.2.1 服务粒度的划分

服务粒度(Service Granularity)是服务的单位,是指一个服务所能完成的功能。一个服务可以由不同的细粒度服务组合而成,粒度适中的服务可以大大提高

服务的灵活性和程序的运行效率,同时也使得代码重复使用率较高。粒度过细,会造成功能散碎、开发成本高、后期维护困难、从而影响整体的容量和性能。粒度过粗,则会使得服务重用率变低、灵活性差、响应速度变慢。因此,划分一个粗细适中的服务粒度,是 SOA 的重要工作。

报表设计器本身可以作为一个粗粒度服务进行功能上的服务,但是鉴于报表样式和数据来源的多样化,将其划分为更细粒度的服务以提高设计器的复用性和灵活性。

2.2.2 主要服务及子服务提炼

在综合考虑服务粒度的基础上,本文的设计器由请求解析、模板设计、报表解析、报表输出四个主要的服务,其中,模板设计服务又可以划分为表头定义服务、格式定义服务、属性设置服务等子服务;报表解析服务可以划分为源码解析服务、编译服务、报表展示服务;报表输出服务可以细分为模板保存服务、格式转化服务、报表打印服务等子服务。数据服务器在预览之前为报表设计器提供了数据填充服务,数据填充服务可以大致划分为数据库连接服务和数据装载服务;各服务之间的关系如图 1 所示。

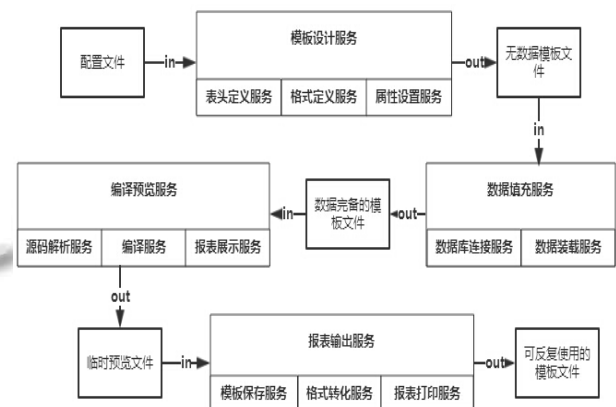


图 1 服务的子服务及各服务之间的关系

2.2.3 服务的规范及实现

服务的规范是指对服务的消息、组合、接口等相关信息做更详细的定义和描述,包括状态、输入、输出、消息、活动、与数据模型的关系等等,而服务实现的实质是决策,是整个 SOA 建模过程的验证。

2.3 整体架构

系统使用基于 SOA 的中间件 ESB 组合各服务模块。ESB,企业服务总线(Enterprise Service Bus),是中

件技术与 Web Service 等技术结合的产物, 联通服务使用者与服务提供者, 实现服务等级协议、服务管理监控、统一安全管理等功能, 起到服务中介的作用, 是 SOA 的基础设施. 所有的功能模块通过 ESB 调用其他相关模块以及功能, 实现系统的功能整合. 报表系统的整体框架如图 2 所示, 有背景颜色的为本文研究对象.

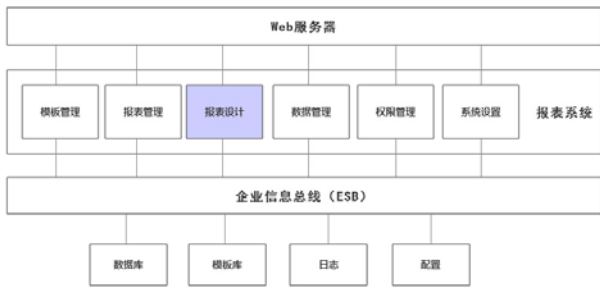


图 2 报表系统整体框架图

基于服务实现的考虑, 报表设计器也分为三层: 功能需求接口层、服务定义层、业务处理接口层. 其中功能需求接口层由前端 Web 页面构成, 是业务系统的主要部分, 主要负责用户验证、展示报表文件、发送报表请求、管理模板文件等; 服务定义层定义报表设计服务接口, 并通过接口为业务系统提供报表设计服务, 接受其他服务模块提供的服务等; 业务处理接口层主要是接收服务接口传递的数据并进行运算、处理, 包括通过 ESB 调用数据、日志, 将设计好的模板文件保存到数据库等, 并将操作结果返回到服务定义层. 报表设计器的结构图如图 3 所示.

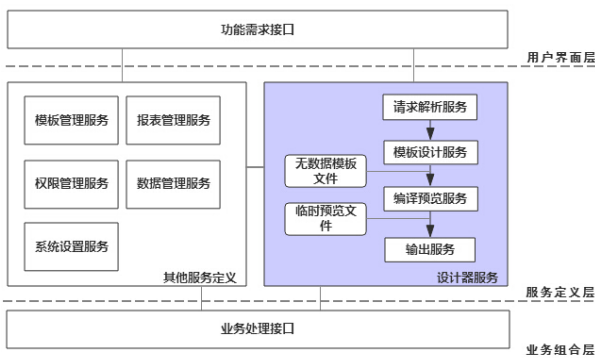


图 3 报表系统整体结构图

当业务系统发来报表请求后, 需要提炼该请求中的关键信息, 如报表种类, 模板名称, 用户的信息: 包括用户的模板库, 用户常用表格类型等, 当请求被解析之后, 若模板是已存在的, 则连接模板库进行提取,

若模板不存在, 则需要重新设计, 或者从模板库取出现存的模板进行再次编辑; 若需要设计/编辑模板, 则进行模板设计; 接下来, 设计完成, 在预览之前需要数据填充服务进行数据填充; 数据装载完毕之后, 就可以进行预览, 此时需要编译预览服务, 形成临时预览文件展示效果, 若设计者对其样式或数据有不满意的部分, 可以返回设计模块进行反复编辑直至满意; 若得到最终的满意成果, 可以进行输出, 这里的输出既可以作为后续使用的模板进行保存进模板库, 也可以进行各种格式的导出, 所以模板输出也封装为一个服务.

3 具体实现

3.1 模板设计

报表模板采用的是 XML 文件的 DTD 方法进行描述, 文档中规定了 XML 文档的整体结构、数据参数、数据字段、变量、面板及其内容和属性等, 报表结构 XML 代码如下:

```
<?XML version="1.0" encoding="UTF-8"?>
<!DOCTYPE Report Engines dtd/"report.dtd">
<Report name="report name" pageWidth="page_width"
pageHeight="page_height" leftMargin="left_margin"
rightMargin="right_margin" topMargin="top_margin"
bottomMargin="bottom_margin">
<parameter name="paraName" class="paraClass ">参数
描述</parameter>
<queryString><![CDATA[SQL 语句]]></queryString>
<field name="fieName" class="fieClass ">字段描
述</field>
变量描述
<variable name="varName" class="varClass "
resetType="Column" incrementType="Report"/>
面板定义
<background>背景 </background>
<pageHeader>页头 </pageHeader>
<tableHeader>表头</tableHeader>
<mainData>数据</mainData>
<tableFooter>表尾 </tableFooter>
<pageSummary>页内总结</pageSummary>
<pageFooter>页尾 </pageFooter>
</Report>
```

报表模板的布局则是由面板元素构成的, 整个页面呈水平排列状态, 每一部分都称为面板元素, 这些面板元素有自己的属性和内容, 在展示的时候, 报表设计器会根据他们的属性特征进行展示, 数据模型的元素不会显示, 页头、表头会在每一页进行展示, 页尾只出现在最后一页, 其余元素会根据他们的排列顺序、高度、宽度等属性进行展示. 默认情况下, 主要的报表面板元素有页头(pageHeader)、表头(tableHeader)、主数据域(mainData)、表尾(tableFooter)、页总结(pageSummary)、页尾(pageFooter)六种元素, 可根据需要进行高度的属性设置, 如果高度设置为 0, 该栏将不会出现在面板上但是在 XML 代码中它的节点仍然存在. 面板的上下左右边距也可以根据需要进行设置. 对于其他一些功能: 子表、交叉报表、图表等控件添加后, 会在 XML 中出现相应的节点元素, 同时在面板中也会展示出控件. 模板外观布局如图 4 所示.



图 4 模板布局图

3.2 控件拖拽技术

在使用报表设计器设计报表时, 本文的报表设计器对现有设计器功能进行优化, 实现了一部分常见控件的拖拽展示, 降低了设计的编码难度, 提高了交互性和易用性, 使设计过程更加简洁.

拖拽功能的实现是 DOM、JavaScript、HTML 的综合应用过程, 首先是在 HTML 文件中为所有可拖拽的控件定义拖拽类 Drag, 通过拖拽类为该控件定义鼠标事件: mousedown()、mousemove()、mouseup() 在 mouseup() 函数中响应报表展示器的监听函数, 监听函数检测到用户的拖拽控件后, 调用 createElement() 函数, 使用 DOM 对象创建拖入的组成元素并同时检测鼠标是否存在于面板区域, 若否, 则报错, 不允许在

非法区域创建控件. 若是, 则检测鼠标弹起时的当前坐标来确定该控件的所属面板元素, 并调用 endDrag() 方法结束控件所绑定的鼠标移动事件和鼠标弹起事件, 向该元素添加控件节点. 并将节点展示在 html 页面中的报表样式中, 报表展示器则继续监听. 若此时设计者不满意控件位置需要继续拖拽, 则返回起始处重新开始拖拽过程. 具体元素定义为:

```
<div class="drag_div" id="draggable"><div id="drag_p">
段落显示内容</div></div>
```

```
页面加载: <body onload="Drag()">
```

3.3 图表支持

本文采用开源产品 jFreeChart 作为图表的支持, 在图形报表绘制方面的功能十分强大. 一般情况下图表都出现在 pageSummary 面板区域内, 通过对数据字段的绑定来直观的显示数据. 以饼形图为例, 通过对字段的设置, 调用 PieDataSet 的 SetValue 方法向结果集中添加数据, 同时向所有注册的监听器发送一个 DatasetChangeEvent 事件, 然后使用 createPieChart() 方法创建一个饼图. SetValue 的原型声明为: public void setValue(java.lang.Comparable key, double value), 一个简单的饼图示例如图 5.



图 5 图表示例

3.4 报表解析

在模板设计结束后, 必须通过 HTML 对 XML 文件进行读取、操作、保存等操作. 这就需要提取面板中的 DOM 对象, 将所有 DOM 对象组合处理之后, 生成相应的报表文件, 并呈现为 XML 格式. 本文采用 DOM(Document Object Model)处理 XML 文档. DOM 的核心是树模型, 通过 org.w3c.DOM.Node 派生 XML 专用接口 Element、Document、Attr、和 Text, 使得遍

历各个节点算法都相对简单,如 `getParent()` 和 `getChildren()` 等方法^[4].在对 HTML 文件进行处理时候,首先是获取生成的 DOM 树,然后对该 DOM 树进行串行化处理.重点处理根节点和元素节点,根节点包含了 XML 的头部信息如版本号、编码方式等;处理完后头结点通过 `getChildren()` 方法获取到元素节点集合,逐一遍历该集合直至为空,对元素节点进行串行化.在这一过程中,需要注意的是,元素节点拥有不同的属性节点,所以同时也需要对这些节点的属性节点进行串行化.具体方式和处理元素节点大同小异:首先使用 `getAttributes()` 方法获取属性节点集合,再逐一遍历这些节点,用 `getNodeName()` 方法和 `getNodeValue()` 方法获取节点的属性名和属性值.串行化之后分别处理 DOM 对象的子节点转化为特定的 XML 输出.

3.5 报表导出

当模板已经编译完毕并预览满意后,需要对报表进行输出,此时的输出不仅是转化为各种格式的输出,也包括了模板的保存.模板的保存比较简单,数据库层有专门的模板库,将模板以 XML 格式存入模板库中,方便了再次的打开和编辑.以特定格式输出则需要报表导出服务中定义不同的映射规则,使用 XSL、iText 等技术对 XML 文档进行节点匹配、格式转换、页面适应等操作,可以将 XML 格式的报表文档转化成常见的格式进行导出,如 Excel、PDF、HTML 等.若有其他的格式需要转换,则需在报表输出服务的接口中添加相应的映射规则.

系统整体效果图如图 6 所示.

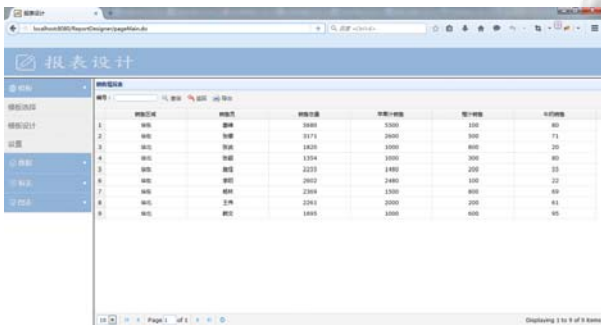


图 6 整体效果图

4 结语

本文使用面向服务的架构方法设计实现了一款基于 SOA 的 Web 报表设计器,由多种粒度的服务共同组成,实现了报表设计服务的解耦合,能够为不同的业务系统提供报表设计服务,具有灵活、易用、交互等优点.本设计器的整体效率很大程度取决于服务粒度的大小,如何寻找一个最优的服务粒度,既能够满足系统需要,又不会冗余堆积是设计器进一步优化需要探索的话题.

参考文献

- 1 胡加顺.Web 报表引擎[硕士学位论文].长春:吉林大学,2005.
- 2 李琦.基于 SOA 的社会保障信息系统架构研究[博士学位论文].南京:南京大学,2011.
- 3 王磊,张忠能.基于 SOA 的报表服务模型的研究与设计.微型电脑应用,2009,(3):4-7.
- 4 Mc Laughlin B.江文,唐俊,刘柯译.Java 与 XML.北京:中国电力出版社,2011.
- 5 Kadry S, Claver J. XML parser GUI using .NET technology. Information Engineering Research Institute Procedia, 2012, 2: 554-560.
- 6 白振龙,于放,李肖肖.基于 Struts 框架的报表编辑系统.计算机系统应用,2014,23(2):48-51.
- 7 张曦泽,王化文,王中秋.基于组件的报表自动生成系统的研究与实现.计算机工程与设计,2005,26(5):1358-1360.