



(12)发明专利

(10)授权公告号 CN 105786526 B

(45)授权公告日 2019.08.27

(21)申请号 201610173835.6

CN 102622215 A,2012.08.01,

(22)申请日 2016.03.24

CN 102186114 A,2011.09.14,

(65)同一申请的已公布的文献号

审查员 张伊璇

申请公布号 CN 105786526 A

(43)申请公布日 2016.07.20

(73)专利权人 江苏大学

地址 212013 江苏省镇江市京口区学府路
301号

(72)发明人 杨洋 唐伟 梅傑 邱成健

(51)Int.Cl.

G06F 8/38(2018.01)

G06F 8/30(2018.01)

(56)对比文件

US 2015081685 A1,2015.03.19,

US 2009055721 A1,2009.02.26,

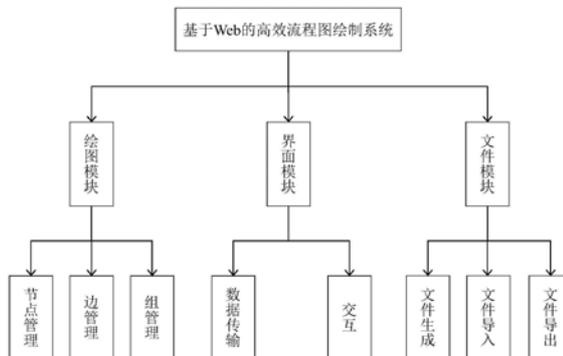
权利要求书2页 说明书6页 附图1页

(54)发明名称

一种基于Web的高效流程图绘制系统与方法

(57)摘要

本发明公开了一种基于Web的高效流程图绘制系统与方法,包括绘图模块、界面模块、文件模块;绘图模块包括节点管理模块、边管理模块以及组管理模块,负责接受传入的流程图的参数、封装成Image类对象,调用一个开源的算法库完成流程图自动布局和绘制工作;界面模块包括交互模块、数据传输模块,采用Bootstrap框架实现页面特效;数据传输模块负责前台、后台数据的异步传输工作;交互模块用于确定出用户点击的图元并将其名称传输给后台,后台检索该图元信息后再通过数据传输模块返回到用户界面;文件模块其一负责接受绘图模块发送过来的SVG文件,生成对应的PDF文件,导出MySQL数据库备份文件;其二负责导入文件、上传文件,以及解析上传的文件获得绘图参数的工作。



1. 一种基于Web的高效流程图绘制系统,其特征在于,包括绘图模块、界面模块、文件模块;

所述绘图模块包括节点管理模块、边管理模块以及组管理模块,负责接受传入的流程图的参数,将这些参数封装成Image类数据对象,并调用一个开源的算法库GraphViz的API完成流程图自动布局和绘制工作;

所述界面模块包括交互模块以及数据传输模块,属于系统用户页面主体部分,采用Bootstrap框架实现页面及页面特效;所述数据传输模块负责前台、后台数据的异步传输工作;所述交互模块一方面负责操作用户界面预览区中SVG图形的Dom接口,确定出用户点击的图元并将其名称传输给后台,后台检索该图元信息后再通过数据传输模块返回到用户界面;所述交互模块另一方面负责对SVG添加动画特效,以使用户进行个性化微调;

所述文件模块包括文件生成模块、文件导入模块和文件导出模块;所述文件模块一方面负责接受所述绘图模块发送过来的SVG文件,并根据该SVG文件生成对应的PDF文件,同时导出MySQL数据库备份文件;另一方面负责导入文件、上传文件,以及解析上传的文件获得绘图参数的工作;

所述绘图模块采用PHP语言编写实现;所述传入的流程图的参数包括:节点信息、边信息、组信息;

所述界面模块采用Html语言与Javascript语言编写;

所述数据传输模块采用Ajax技术实现,所述Ajax技术包括:异步Javascript和Xml技术;

所述交互模块采用两个Javascript脚本实现;

所述文件模块采用PHP和Python语言编写实现。

2. 一种基于Web的高效流程图绘制方法,其特征在于,包括如下步骤:

1) 用户提出绘图请求,发送绘图数据,所述绘图数据分为:输入参数绘图和导入文件绘图;

2) 如果用户通过导入文件绘图,则转步骤3),否则转步骤5);

3) 导入文件名被数据传输模块封装成XmlHttpRequest对象,并传递给文件导入模块;

4) 文件导入模块对XmlHttpRequest对象进行解析获取导入文件名,读取并上传文件,在后台解析文件获得绘图参数,之后将解析得到的绘图参数传递给绘图模块;

5) 界面模块中的数据传输模块将参数封装为XmlHttpRequest对象并发送给绘图模块;

6) 绘图模块将绘图参数封装成Node类、Edge类以及Group类的对象,并将这些信息存储到数据库中;所述的Node类包括:名称、标签、形状、边框色、填充色、字体色;所述的Edge类包括:始点、终点、始位置、终位置、标签、箭头形状、边的形状、颜色;所述的Group类包括名称、标签、边框色、填充色、字体色;

7) 绘图模块从数据库中读取当前流程图的参数信息,并将其封装成Image类对象,并将该对象传递给GraphViz引擎;其中Image类的数据成员包括布局方向、Node类对象集合、Edge类对象集合以及Group类对象集合;

8) 绘图模块调用GraphViz的API对传入的Image对象进行布局和绘图,生成该Image对象的SVG格式文件,并将该文件传递给文件生成模块;

9) 文件生成模块将SVG格式文件转换成PDF格式文件,并使用PHP执行外部命令

mysqldump的方式生成MySQL数据库备份文件,即SQL文件,将这些文件传递给了文件导出模块;

- 10) 文件导出模块将这些文件封装成二进制流的形式后传递给数据传输模块;
- 11) 数据传输模块将二进制流封装为XmlHttpRequest对象并将其传递给交互模块;
- 12) 交互模块对XmlHttpRequest对象进行解析,获取导出文件;
- 13) 用户从图形界面中下载流程图文件;

所述步骤13)还包括:用户还可以通过图形化界面对现有流程图进行编辑,所述编辑包括:对流程图布局方式、节点、边、组的属性进行修改和对流程图中节点位置进行微调;

所述对流程图布局方式、节点、边、组的属性进行修改的具体实现方法包括如下步骤:

1-1) 根据Javascript的冒泡事件流确定用户具体点击的是哪个图元,所述图元包括节点、边、组,取该图元的名称,将名称发送给数据传输模块,并将图元用其他颜色突出显示,之后利用睡眠函数在1秒内将图元颜色还原;

1-2) 数据传输模块将图元的名称封装成XmlHttpRequest对象并发送给后台PHP数据库查询脚本;

1-3) PHP数据库查询脚本将图元的名字解析后,根据图元的名字在数据库中检索图元对应的属性,并将其传递给数据传输模块;

1-4) 数据传输模块将图元对应属性封装成XmlHttpRequest对象并传递给图形界面;

1-5) 图形界面将图元对应属性解析后展示给用户;

所述对流程图中节点位置进行微调的功能具体实现包括如下步骤:

2-1) 用户通过图形界面拖拽节点至新位置,图形界面将该节点名称及位置传递给JQuery.GraphViz,之后脚本调用SVG格式提供的Dom接口,用户可以微调预览区的SVG格式流程图的节点位置;其中JQuery.GraphViz是一个开源代码,其授权为GPL协议;

2-2) 图形界面将其展示给用户;用户可以点击保存按钮,通过文件模块再次生成PDF文件,并提供最新的PDF文件下载。

一种基于Web的高效流程图绘制系统及方法

技术领域

[0001] 本发明属于计算机软件技术领域,涉及一种流程图绘制系统,尤其涉及一种基于Web技术且应用布局技术自动根据传入参数进行布局的流程图绘制系统。

背景技术

[0002] 在流程图的绘制过程中,现有方法应用最广泛的是利用Microsoft提供的Office系列软件如Word,PowerPoint或者Visio等进行绘制,这种方法费时费力。用户首先需要选择所画图类型,之后选择节点形状、然后拖拽节点到指定位置,再然后指定节点内容,最终连线并绘制成流程图。对于小型图,即节点以及连线数量比较少或者一次成型不需要过多修改的情况比较方便。然而,这种方法也存在一些难以克服的缺点。例如:1对于大型或者超大型流程图的绘制,这种方法会变得非常繁琐,需要绘图者手动去布局,拖拽的操作不仅费时费力,并且在图连线或者调整的过程中,还有可能导致原有的布局紊乱。2主流的流程图绘制软件都是以Windows平台为主,即使其他平台上有同类的软件,各种用户体验也很难做到一致,用户需要适应不同平台各个软件之间的差异性,增加了使用者的学习成本。

[0003] 鉴于上述方法在绘制流程图时的种种不便,AT&T实验室研发了一套开源作图工具软件GraphViz,通过该软件可以实现对流程图进行自动布局。用户可以根据GraphViz提供的Dot语言来编写绘图脚本,之后通过调用GraphViz的绘图命令对脚本进行解释从而生成各种图形。GraphViz在一定意义上解决了传统方法的问题,然而也带来了一些新的问题,例如,1虽然Dot语言是简单的脚本语言,然而并不是所有人都会用的,用户必须具有一定的编程基础,使用门槛较高。2由于系统自动对绘制出的流程图进行布局,并不允许用户对生成流程图的布局进行个性化的微调,尽管自动布局的流程图在大体上满足用户需求,然而在细节上并不一定能够满足用户的个性化需求,所以会在一定程度上影响用户的绘图体验。

发明内容

[0004] 本发明为了克服上述现有技术中绘制流程图的种种缺陷,提出了一种高效的流程图绘制系统及方法,实现本发明的技术方案如下:

[0005] 一种基于Web的高效流程图绘制系统,包括绘图模块、界面模块、文件模块;

[0006] 所述绘图模块包括节点管理模块、边管理模块以及组管理模块,负责接受传入的流程图的参数,将这些参数封装成Image类数据对象,并调用一个开源的算法库GraphViz的API完成流程图自动布局 and 绘制工作;

[0007] 所述界面模块包括交互模块以及数据传输模块,属于系统用户页面主体部分,采用Bootstrap框架实现页面及页面特效;所述数据传输模块负责前台、后台数据的异步传输工作;所述交互模块一方面负责操作用户界面预览区中SVG图形的Dom接口,确定出用户点击的图元并将其名称传输给后台,后台检索该图元信息后再通过数据传输模块返回到用户界面;所述交互模块另一方面负责对SVG添加动画特效,以使用户进行个性化微调;

[0008] 所述文件模块包括文件生成模块、文件导入模块和文件导出模块;所述文件模块

一方面负责接受所述绘图模块发送过来的SVG文件,并根据该SVG文件生成对应的PDF文件,同时导出MySQL数据库备份文件;另一方面负责导入文件、上传文件,以及解析上传的文件获得绘图参数的工作。

[0009] 进一步优选方案,所述绘图模块采用PHP语言编写实现;所述传入的流程图的参数包括:节点信息、边信息、组信息。

[0010] 进一步优选方案,所述界面模块采用Html语言与Javascript语言编写。

[0011] 进一步优选方案,所述数据传输模块采用Ajax技术实现,所述Ajax技术包括:异步Javascript和Xml技术。

[0012] 进一步优选方案,所述交互模块采用两个Javascript脚本实现。

[0013] 进一步优选方案,所述文件模块采用PHP和Python语言编写实现。

[0014] 基于上述高效流程图绘制系统,本发明还提出了一种基于Web的高效流程图绘制方法,包括如下步骤:

[0015] 1) 用户提出绘图请求,发送绘图数据,所述绘图数据分为:输入参数绘图和导入文件绘图;

[0016] 2) 如果用户通过导入文件绘图,则转步骤3),否则转步骤5);

[0017] 3) 导入文件名被数据传输子模块封装成XmlHttpRequest对象,并传递给文件导入模块;

[0018] 4) 文件导入模块对XmlHttpRequest对象进行解析获取导入文件名,读取并上传文件,在后台解析文件获得绘图参数,之后将解析得到的绘图参数传递给绘图模块;

[0019] 5) 界面模块中的数据传输子模块将参数封装为XmlHttpRequest对象并发送给绘图模块;

[0020] 6) 绘图模块将绘图参数封装成Node类、Edge类以及Group类的对象,并将这些信息存储到数据库中;所述的Node类包括:名称、标签、形状、边框色、填充色、字体色;所述的Edge类包括:始点、终点、始位置、终位置、标签、箭头形状、边的形状、颜色;所述的Group类包括名称、标签、边框色、填充色、字体色;

[0021] 7) 绘图模块从数据库中读取当前流程图的参数信息,并将其封装成Image类对象,并将该对象传递给GraphViz引擎;其中Image类的数据成员包括布局方向、Node类对象集合、Edge类对象集合以及Group类对象集合;

[0022] 8) 绘图模块调用GraphViz的API对传入的Image对象进行布局 and 绘图,生成该Image对象的SVG格式文件,并将该文件传递给文件生成模块;

[0023] 9) 文件生成模块将SVG格式文件转换成PDF格式文件,并使用PHP执行外部命令mysqldump的方式生成MySQL数据库备份文件,即SQL文件,将这些文件传递给了文件导出模块;

[0024] 10) 文件导出模块将这些文件封装成二进制流的形式后传递给数据传输模块;

[0025] 11) 数据传输模块将二进制流封装为XmlHttpRequest对象并将其传递给交互模块;

[0026] 12) 交互模块对XmlHttpRequest对象进行解析,获取导出文件;

[0027] 13) 用户从图形界面中下载流程图文件。

[0028] 进一步优选方案,所述步骤13)还包括:用户还可以通过图形化界面对现有流程图

进行编辑,所述编辑包括:对流程图布局方式、节点、边、组的属性进行修改和对流程图中节点位置进行微调。

[0029] 进一步优选方案,所述对流程图布局方式、节点、边、组的属性进行修改的具体实现方法包括如下步骤:

[0030] 1-1) 根据Javascript的冒泡事件流确定用户具体点击的是哪个元素,所述元素包括节点、边、组,读取该图元的名称,将名称发送给数据传输模块,并将该图元用其他颜色突出显示,之后利用睡眠函数在1秒内将该图元颜色还原;

[0031] 1-2) 数据传输模块将该图元的名称封装成XmlHttpRequest对象并发送给后台PHP数据库查询脚本;

[0032] 1-3) PHP数据库查询脚本将该元素的名称解析后,根据该元素的名称在数据库中检索该图元对应的属性,并将其传递给数据传输模块;

[0033] 1-4) 数据传输模块将该图元对应属性封装成XmlHttpRequest对象并传递给图形界面;

[0034] 1-5) 图形界面将该图元对应属性解析后展示给用户。

[0035] 进一步优选方案,所述对流程图中节点位置进行微调的功能具体实现包括如下步骤:

[0036] 2-1) 用户通过图形界面拖拽节点至新位置,图形界面将该节点名称及位置传递给JQuery.GraphViz,之后该脚本调用SVG格式提供的Dom接口,用户可以微调预览区的SVG格式流程图的节点位置;其中JQuery.GraphViz是一个开源代码,其授权为GPL协议;

[0037] 2-2) 图形界面将其展示给用户;用户可以点击保存按钮,通过文件模块再次生成PDF文件,并提供最新的PDF文件下载。

[0038] 与现有技术相比,本发明的有益效果:

[0039] 1、提供了基于Web的图形化用户界面,用户无需了解底层GraphViz的Dot语言,便可以利用GraphViz高效地绘制流程图;用户可以通过该界面指定流程图的一些属性,如布局方向、节点名称、节点颜色、线条连接的两个节点名称,线条的颜色样式等等,系统即能够调用GraphViz底层自动布局算法,实时绘制流程图。这样一方面既降低了用户的使用门槛,与此同时,基于Web的绘图工具能够规避各个平台用户体验的不一致性,减少了学习成本;另一方面用户无需费时费力地进行手动布局,从而可以快速绘制流程图。

[0040] 2、提供了对实时绘制出的流程图进行布局微调的功能。自动布局的流程图一般能够大体上满足用户需求,然而在细节上并不一定能够满足用户个性化的需求。所以,本发明实现了个性化微调的功能,允许用户动态地对已经自动布局生成流程图中的节点位置进行调整,用户可以通过按住鼠标左键拖拽的方式改变节点及其相连接的边的位置以生成满足自身需求的流程图。

[0041] 3、实现了GraphViz的图形化用户界面,方便用户使用,降低了用户的使用门槛。用户可以通过添加、删除按钮对节点、边以及组进行添加和删除,也可以通过点击节点、边以及组选中之后对其各个属性进行编辑。

附图说明

[0042] 图1是基于Web的高效流程图绘制系统的系统结构示意图。

[0043] 图2是本发明绘图方法实现的流程示意图。

具体实施方式

[0044] 下面结合附图和具体实施例对本发明作进一步详细描述。

[0045] 如附图1所示,将软件分割为三个模块,分别是绘图模块、界面模块、文件模块。其中,绘图模块包括节点管理、边管理以及组管理;界面模块包括交互以及数据传输;文件模块包括文件生成、文件导入和文件导出。

[0046] 绘图模块由PHP语言编写,它接受传入的流程图的各个参数(节点信息、边信息、组信息),将这些参数封装成Image类数据对象,并调用一个开源的算法库GraphViz的API完成流程图自动布局和绘制工作。

[0047] 界面模块是系统用户页面主体部分,由Html语言与Javascript语言编写,具体实现包括:(1)使用Bootstrap框架(包含jQuery库)完成了简洁美观的页面及页面特效;(2)对于数据传输子模块使用Ajax技术(异步Javascript和Xml技术)完成,主要进行前台、后台数据的异步传输工作;(3)对于交互子模块使用了两个Javascript脚本,一个负责操作用户界面预览区中SVG图形的Dom接口,确定出用户点击的图元并将其名称传输给后台,后台检索该图元信息后再通过数据传输模块返回到用户界面;另一个负责对SVG添加动画特效,以便用户进行个性化微调。

[0048] 文件模块主要由PHP和Python语言编写,它的主要任务是接受绘图模块发送过来的SVG文件,并根据该SVG文件生成对应的PDF文件,同时导出MySQL数据库备份文件;另外,它还负责导入文件,文件的上传工作,以及解析上传的文件获得绘图参数的工作。

[0049] 上述绘图模块、界面模块以及文件模块之间都是通过数据传递来实现衔接的,现对模块间的数据传递衔接作出详细说明:

[0050] 1) 用户采用输入参数绘图:界面模块接收到用户输入的参数之后,将参数封装为XmlHttpRequest对象,并发送给绘图模块。绘图模块解析该对象,将绘图参数封装为节点(Node)、边(Edge)、组(Group),并按指定格式存储在服务器数据库中,之后将数据库中该用户所对应的表的所有信息读出,封装为Image对象,并传递给GraphViz引擎。GraphViz引擎实时绘制SVG格式的流程图文件,并将其传递给文件生成模块。文件生成模块根据SVG流程图文件生成对应的PDF流程图文件,并使用PHP执行外部命令mysqldump的方式生成MySQL数据库备份文件,即SQL文件,将这些文件传递给了文件导出模块。文件导出模块将这些文件封装为二进制流的形式传递给数据传输模块。数据传输模块将二进制流封装为XmlHttpRequest对象,并将其传递给交互模块。交互模块对XmlHttpRequest对象进行解析,获取导出文件。

[0051] 2) 用户采用导入文件绘图:导入文件名被数据传输模块封装为XmlHttpRequest对象,并传递给文件导入模块。文件导入模块对XmlHttpRequest对象进行解析获取导入文件名,读取并上传文件,在后台解析文件获得绘图参数之后将解析得到的绘图参数传递给绘图模块。之后的所有数据传递与1)中所述的对应步骤的数据传递一致。

[0052] 系统绘图方法的具体流程如附图2所示,具体包含以下步骤:

[0053] 1) 用户提出绘图请求,发送绘图数据,这些数据包括:输入参数绘图和导入文件绘图。

- [0054] 2) 如果用户通过导入文件绘图,则转3),否则转5)。
- [0055] 3) 导入文件名被数据传输子模块封装成XmlHttpRequest对象,并传递给文件导入模块。
- [0056] 4) 文件导入模块对XmlHttpRequest对象进行解析获取导入文件名,读取并上传文件,在后台解析文件获得绘图参数,之后将解析得到的绘图参数传递给绘图模块。
- [0057] 5) 界面模块中的数据传输子模块将参数封装为XmlHttpRequest对象并发送给绘图模块。
- [0058] 6) 绘图模块将绘图参数封装成Node(节点)、Edge(边)以及Group(组)类的对象,并将这些信息存储到数据库中。其中Node(点)类包括名称、标签、形状、边框色、填充色、字体色;Edge(边)类包括始点、终点、始位置、终位置、标签、箭头形状、边的形状、颜色;Group(组)类包括名称、标签、边框色、填充色、字体色。
- [0059] 7) 绘图模块从数据库中读取当前流程图的参数信息,并将其封装成Image类对象,并将该对象传递给GraphViz,其中Image类的数据成员包括布局方向、Node类对象集合、Edge类对象集合以及Group类对象集合。
- [0060] 8) 绘图模块调用GraphViz的API对传入的Image对象进行布局和绘图,生成该Image对象的SVG格式文件,并将该文件传递给文件生成模块。
- [0061] 9) 文件生成模块将SVG格式文件转换成PDF格式文件,并使用PHP执行外部命令mysqldump的方式生成MySQL数据库备份文件,即SQL文件,将这些文件传递给了文件导出模块。
- [0062] 10) 文件导出模块将这些文件封装成二进制流的形式后传递给数据传输模块。
- [0063] 11) 数据传输模块将二进制流封装为XmlHttpRequest对象并将其传递给交互模块。
- [0064] 12) 交互模块对XmlHttpRequest对象进行解析,获取导出文件。
- [0065] 13) 用户从图形界面中下载流程图文件。
- [0066] 完成上述工作后用户即可在用户界面实时预览生成的图形,或者下载生成的PDF文件以及SQL文件。此外,用户还可以通过图形化界面对现有流程图进行编辑,这里编辑功能包含两方面的内容:1、对流程图布局方式、节点、边、组的属性进行修改;2、对流程图中节点位置进行微调。这两个方面的内容可以通过在预览区所在Div中分别绑定了两个用于交互的Javascript脚本来实现,第一个脚本是JQuery.Operate,第二个脚本是JQuery.GraphViz。下面对这两个功能的实现作详细介绍。
- [0067] 对于上述功能1,即对流程图布局方式、节点、边、组的属性进行修改的功能的具体实现方式如下:
- [0068] 1) 根据Javascript的冒泡事件流确定用户具体点击的是哪个元素(节点、边、组),读取该图元的名称(在图形中唯一标识该元素),将名称发送给数据传输模块,并将该图元用其他颜色突出显示,之后利用睡眠函数在1秒内将该图元颜色还原。
- [0069] 2) 数据传输模块将该图元的名称封装成XmlHttpRequest对象并发送给后台PHP数据库查询脚本。
- [0070] 3) PHP数据库查询脚本将该元素的名称解析后,根据该元素的名称在数据库中检索该图元对应的属性,并将其传递给数据传输模块。

[0071] 4) 数据传输模块将该图元对应属性封装成XmlHttpRequest对象并传递给图形界面。

[0072] 5) 图形界面将该图元对应属性解析后展示给用户。

[0073] 对于上述功能2,即对流程图中节点位置进行微调的功能的具体实现方式如下:

[0074] 1) 用户通过图形界面拖拽节点至新位置,图形界面将该节点名称及位置传递给JQuery.GraphViz,之后该脚本调用SVG格式提供的Dom接口,用户可以微调预览区的SVG格式流程图的节点位置。其中JQuery.GraphViz是一个开源代码,其授权为GPL协议。

[0075] 2) 图形界面将其展示给用户。用户可以点击保存按钮,通过文件模块再次生成PDF文件,并提供最新的PDF文件下载。

[0076] 以上所述仅为本发明的较佳实例而已,并不用于限制本发明,凡在本发明的原则和精神之类所做的任何修改、等同替换和改进等,均就包含在本发明的保护范围之内。在实际实施的过程中,用户界面、数据传输方式、交互方式、存储方式等,都可能随着技术的变革或相关领域技术人员的实施方法而改变,但上述形式的改变都不会从根本上改变本发明的流程图绘制、修改和跨平台统一方式,它们被认为是出于本发明权利要求书定义的范围之内。

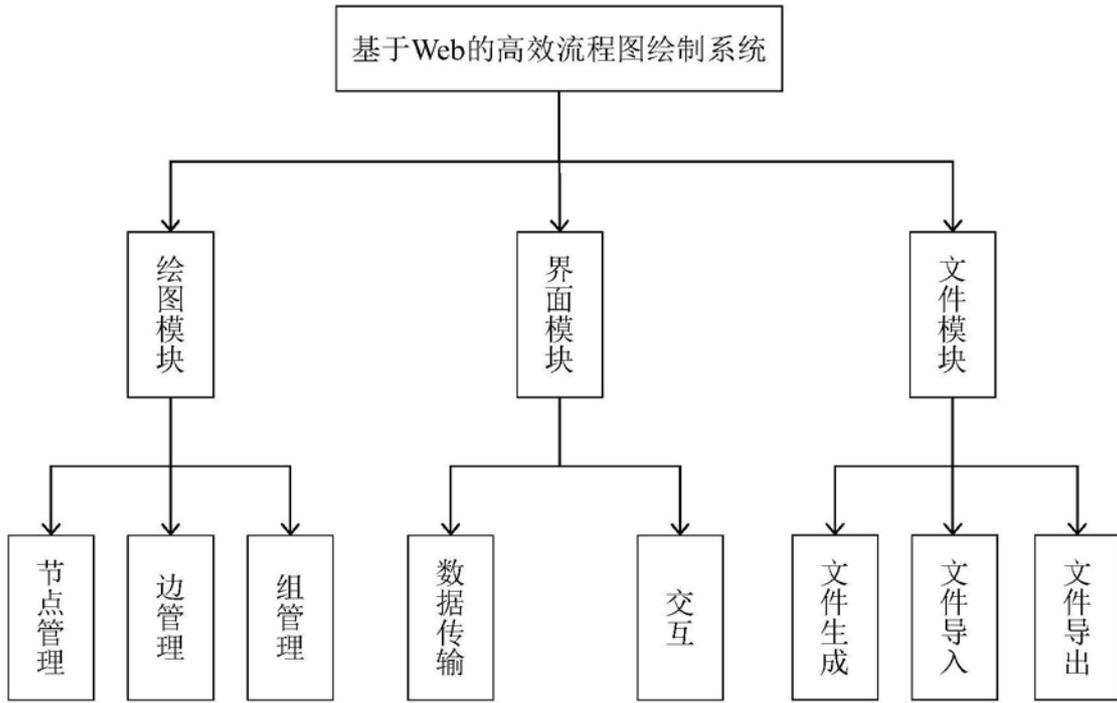


图1

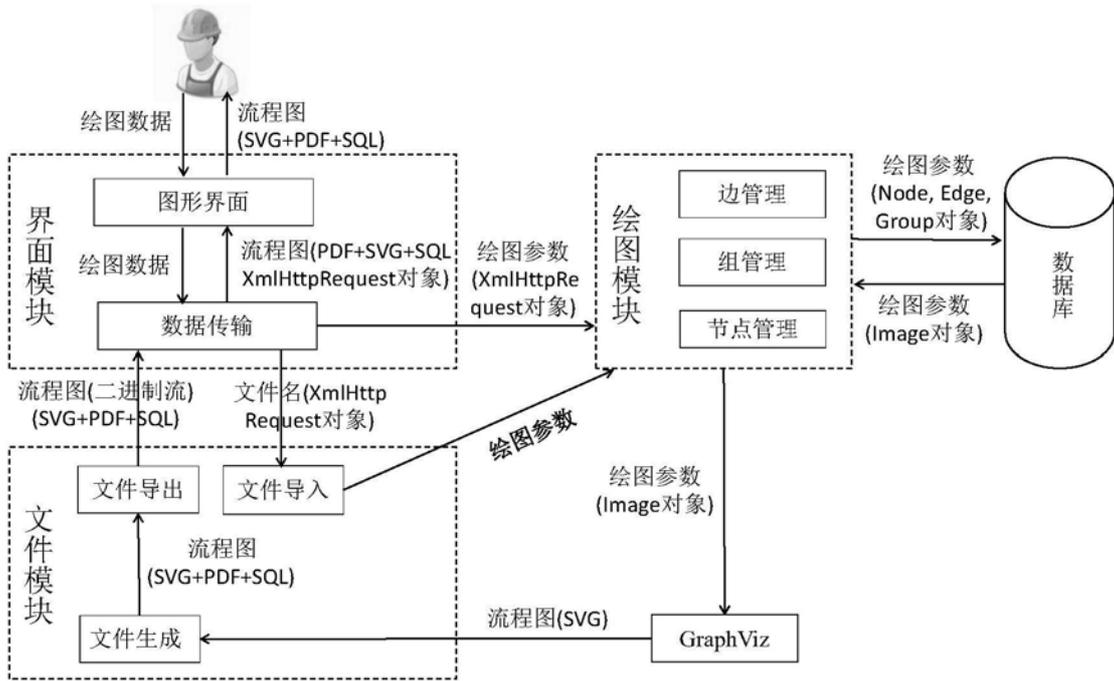


图2