

· 计算机软件理论、技术与应用 ·

基于 D3 的层次边聚合图设计与实现

刘旭

(SAP 中国研究院商务智能部 上海 201203)

摘要: 首先分析层次边聚合图(hierarchical edge bundles)使用扇环和样条曲线进行可视化布局的特点,阐述使用样条曲线聚合连接线以减少视觉复杂度的设计细节;接着在分析基于 SVG 的 D3(data-driven documents)可视化库特点的基础上,重点描述使用贝塞尔曲线分段绘制样条曲线的过程,并通过 D3 实现具有高亮、旋转等交互性的层次边聚合图。在案例分析中,应用基于 D3 的层次边聚合图对 jEdit 和 Linux 的代码重复检测结果进行可视化分析。其分析结果表明:基于 D3 实现层次边聚合图具有可行性;将层次边聚合图技术应用到软件可视化分析中具有可行性。

关键词: 可视化; 层次边聚合图; D3; JavaScript; SVG

中图分类号: TP391; TP311.1 **文献标志码:** A **文章编号:** 1673-159X(2017)04-0027-7

doi: 10.3969/j.issn.1673-159X.2017.04.005

Design and Implementation of Hierarchical Edge Bundles Based on D3

LIU Xu

(Department of Business Intelligence of SAP Labs China, Shanghai 201203 China)

Abstract: In this paper, the sector ring and spline visualization layout characteristic of hierarchical edge bundles are analyzed in detail, and design details of using spline curves to render bundles to reduce the visual complexity are expounded. On the basis of comparative analysis of the characteristics of D3 visualization library based on SVG, the paper mainly analyzes the process to render spline curves by using Bezier curve segments, and implements hierarchical edge bundles with highlight, rotation and more interactions using D3. The results of visualization experiments using code duplication detection data prove the technical feasibility of hierarchical edge bundles based on D3, and show the feasibility of hierarchical edge bundles in software visualization analysis.

Keywords: visualization; hierarchical edge bundles; D3; JavaScript; SVG

在信息可视化中,层次数据和网络数据的可视化都有很多方式,其中最常见的是结点连接图,即使用几何形状或文字代表数据结点,使用线段代表数据之间的关系。层次数据的结点连接图即为树形图,网络数据的结点连接图即为网络图。树形图的节点之间有层次关系,而在网络图中,每一个数据结点都可能和其他数据结点有关系,这种关系没有层次之分。

结点连接图可以清楚地表示结点之间的关系,其缺点是当线段较多时,线段和结点很容易交叠在一起,导致可视化分析的困难^[1]。图1是一个 SAP

Lumira 软件生成的网络结点连接图。可以看到,结点密集的地方已经不易看清结点的关系。如何使数据结点之间的关系变得清晰、易读是可视化需要解决的关键问题。对于层次结构的数据可以改用 Treemap(树图),而对于网络数据,常用的方式是使用力导向算法对结点和连线重新布局^[2],或将满足一定条件的连接线聚合成为线束。

在信息可视化中,数据结点常见的是一种更复杂的数据结构,例如在代码重复可视化的检测结果中,数据结点之间有层次关系,同时数据的叶子结点之间有非层次的网络关系。层次边聚合图(hiera-

收稿日期:2016-11-22

作者简介:刘旭(1982—),男,硕士,主要研究方向为商务智能、信息可视化及教育技术。E-mail: liuxuhere@hotmail.com

引用格式:刘旭.基于 D3 的层次边聚合图设计与实现[J].西华大学学报(自然科学版)2017,36(4):27.

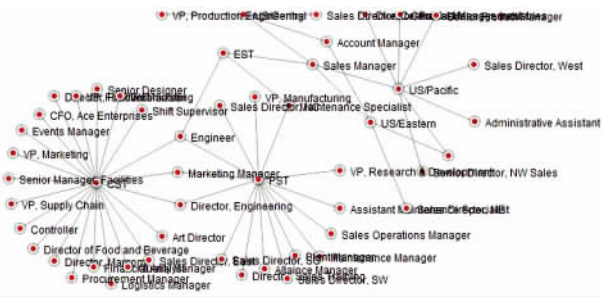


图1 SAP Lumira 软件生成的网络图

层次边聚合图(hierarchical edge bundles)是专门为了表示叶子结点之间存在网络关系的层次数据而设计的,这是一种较新颖的可视化形式^[3]。在不同的中文文献中,它往往使用不同的中文名,本文使用的中文名为层次边聚合图。层次边聚合图用于显示的数据类型在信息可视化中具有代表性,但在当前的各类可视化库

中,很难找到满足完整功能的层次边聚合图以应用于系统开发。本文探讨使用 D3 在 Web 平台上设计并实现可交互的层次边聚合图的细节(包括绘制和交互),并在代码重复检测结果的可视化应用中以实例研究证明层次边聚合图的作用。

1 层次边聚合图的布局

层次数据的 Treemap 可视化使用具有一定面积的块来表示数据结点,数据结点之间的层次关系使用块之间的包含关系表示^[4]。图 2 示出一个使用 SAP Lumira 软件生成的 Treemap。可以看到, Treemap 能够充分利用屏幕空间,可以通过块的位置表现数据结点的分布关系,还可以通过颜色填充方式表示数据结点的不同属性^[5]。

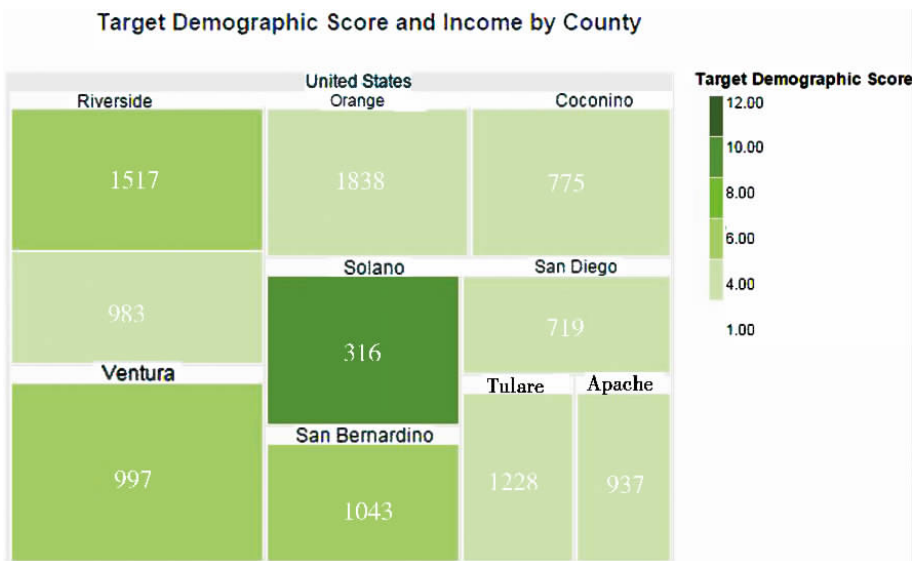
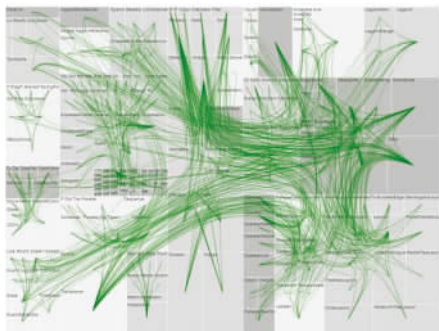
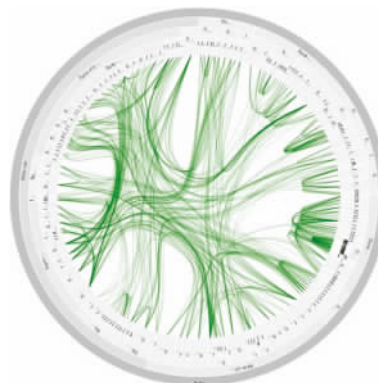


图2 SAP Lumira 软件生成的 Treemap

如果使用 Treemap 表示层次数据,在 Treemap 的矩形块之间添加连接线表示叶子结点之间的网络关系;那么这些连线叠加在矩形块上,不但结点之间的关系难以辨识,而且由于有的结点被密集的连接线覆盖,不便于进行鼠标点击等交互操作,如图 3(a) 所示。



(a) 连接线与矩形块重叠



(b) 连接线不与扇环重叠

图3 层次边聚合图对结点连线效果的改善

为避免连线与结点的重叠,叶子结点的布局方式需要为连线留出足够空间。层次边聚合图如图 3(b) 所示,使用圆环来表示数据^[6],根据层次数据

的层次数把圆环分成若干层,在最内层的是叶子结点,由内而外是越来越上层的结点,中间的圆形空间为连线保留。每一层的圆环被分成多个扇环表示各个数据结点^[7],扇环的角度根据每个结点的值占该层总值的比重而定,内层结点之间用连线表示网络关系。图 4 示出了一个由结点连接图生成层次边聚合图的过程。

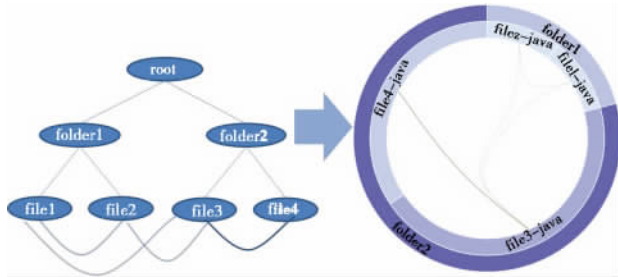
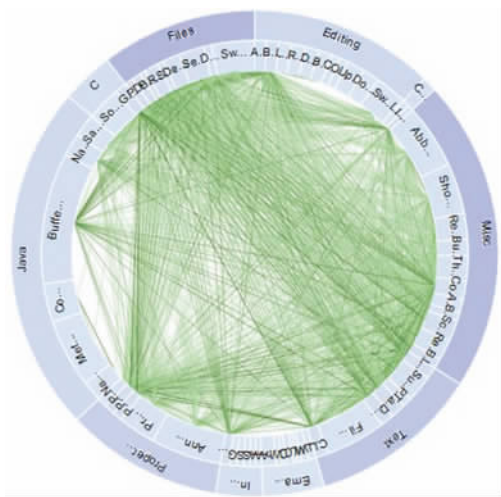


图 4 生成层次边聚合图的示意图

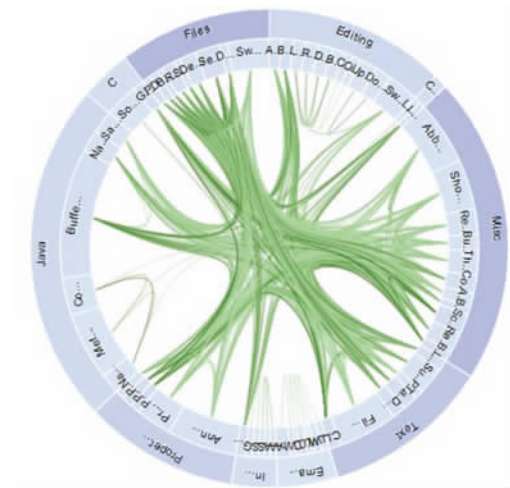
层次边聚合图不需要在各层次之间画连线,节省了空间,减少了视觉上的复杂性,还可以通过结点的颜色、材质等表示更多维度的数据。由于表示网络关系的连线都画在环的内部,不会与表示数据块的扇环交叠,因此,它改善了视觉上的复杂性。层次边聚合图需要将环内的空间留作连线的显示,层次过多则环内空间变小;因此,它不适合完整显示层次过多的数据。

2 连接线的聚合设计

对于层次边聚合图而言,如果使用直线作为连接线,效果类似于图 5(a),连接线之间的关系难以看清。如果通过适当的弯曲结点之间的连接线,使相似的连接线聚在一起构成线束,可以降低连线的视觉复杂度^[8],如图 5(b)所示。



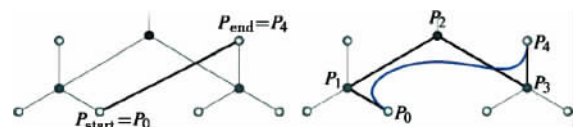
(a) 直线导致视觉复杂度高



(b) 聚合可显示线束的特点

图 5 使用直线与曲线作为连接线的不同效果

层次边聚合图使用的弯曲连接线是利用叶子结点的上层结点作为控制点生成的样条曲线。图 6(a) 示出一个在层次数据的结点连接图中,2 个叶子结点 P_{start} 和 P_{end} 有连接线的情况。可以看到,这 2 个叶子结点可以看作经由 3 个上层结点 P_1 、 P_2 、 P_3 连接,将 P_{start} 作为 P_0 , P_{end} 作为 P_4 。图 6(b) 示出了利用这 5 个点之间的连接线段作为控制多边形生成连接 P_0 与 P_4 的样条曲线的情况。



(a) 叶子结点及上层结点 (b) 控制多边形及样条曲线

图 6 利用上层结点作为控制点生成样条曲线

将层次数据的叶子结点放在最内层,实际上是将初始位置的树形结构的结点连接图做了变换。图 7 示出了这个变换的过程^[8]。由于连接线画在环的内部,如果要使用样条曲线作为结点连接线,样条曲线的控制点仍然应该使用环内部的初始位置的上层结点。

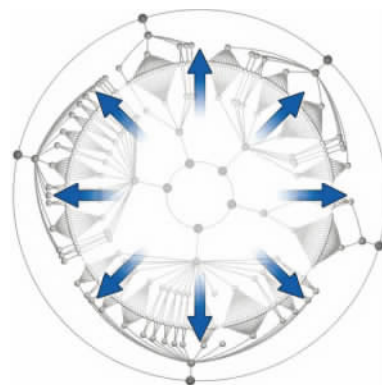


图 7 层次边聚合图使用的控制结点示意图
弯曲的连接线反映了 2 个叶子结点的上层结点

的信息,上层结点相似的叶子结点,连接线就会聚合在一起成为线束。当2条连接线对应的叶子结点具有相同的控制点时,2条样条曲线的某些部分会非常接近,以致难以分辨曲线的端点。图8(a)中, P_0 、 P_4 、 P_5 、 P_6 结点有共同的上层结点,图8(b)显示了它们之间可能的4条连接线,从图8(a)中难以分清图中的连接线是 (P_0, P_4) 与 (P_5, P_6) 2条线,还是 (P_0, P_6) 与 (P_5, P_4) 2条线。

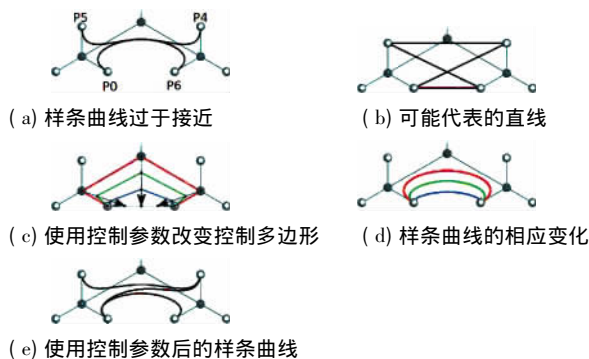


图8 使用控制参数减少曲线重合

为解决这个问题,可以引入一个值为0到1之间的控制参数^[8]。图8(c)和图8(d)示出了控制参数改变控制多边形的形状,进而改变样条曲线的弯曲程度的过程。图8(e)示出控制参数设为0.8的情况,可以清楚地分辨出图中的3条曲线。

3 D3 可视化技术

可视化的实现技术是指具体用什么技术在软件中绘制出信息数据的可视化表示,它基本上是指图形绘制的实现问题。考虑到软件的通用性和生命周期,现在的应用软件系统一般都有跨平台的趋势,这也就要求可视化实现跨平台。常见的跨平台可视化技术包括Java和HTML5。HTML5开发速度快,易于与Web应用整合,所以使用越来越广泛。

HTML5将Web作为绘图的平台,JavaScript是HTML5使用的程序设计语言^[9]。由于HTML5是基于W3C标准的,所以支持HTML5标准的浏览器即可运行使用HTML5技术的可视化Web程序。HTML5下常用的绘图技术有Canvas和SVG^[10]。Canvas提供类似传统的2D绘图的API,可以进行像素级控制。Canvas中的WebGL提供与OpenGL类似的高性能3D绘图API,速度快,而且WebGL可以使用硬件加速;然而,Web应用需要跨浏览器,WebGL在不同浏览器上的支持差别较大,而且信息可视化应用绘制的是抽象信息的隐喻,一般并不要求像素级别的控制,Canvas的优势难以发挥。

SVG是基于XML的二维矢量图格式,使用

SVG不仅便于绘图,还可以利用Web程序中操作DOM的方式操作SVG,便于实现交互和动画。SVG元素基于文本对象模型(document object model),可以直接使用JavaScript添加事件处理函数。如果使用Canvas进行绘制,则在需要进行交互性处理时,对于图形上的鼠标点击等事件,必须按照事件发生的坐标进行判断和发送,这实际上需要程序员自己去建立可以接受事件的对象,在进行事件处理时编写较为复杂的程序。由于SVG是基于DOM的,可以很方便地使用CSS去改变各SVG元素的属性,让浏览器自动重绘相应的元素。Canvas只是绘制像素,在需要重绘时必须由程序员自行判断要重绘的部分并进行重绘。这些因素都使得Canvas不适合用于开发可交互的并且易于与网页整合的信息可视化,而SVG成为首选。

使用JavaScript配合SVG实现的可视化技术已经比较成熟^[11],有大量的软件库可以利用。D3就是一个比较流行的基于数据的JavaScript库,全称是data-driven documents。D3是在Protovis的基础上开发的数据驱动的可视化库,添加了scale、layout等模块。D3的设计思想是直接将数据绑定到SVG元素上,尽可能高效操纵基于数据的文档对象,其具有2倍Protovis的时间性能。D3使用独特的链式语法,提供了操作各种复杂数据集的方法^[12],同时还为事件处理等交互性的实现提供了很好的支持。在可视化方面,D3有很多内建的基于SVG的布局函数;但是D3本身并不提供整体的完整开发框架,开发人员需要使用D3提供的功能自行设计并实现实用的可视化实例。

4 层次边聚合图的实现

如果使用D3进行层次边聚合图的实现,关键的部分是对于结点扇环的绘制和对于连接线的绘制。扇环的布局实现相对比较简单,可以使用D3中的d3.layout.partition函数,而单个扇环的绘制可以借助D3中的d3.svg.arc函数进行,其本质上是使用SVG画出相应的Path元素。

在连接线的绘制方面,需要解决的难点是样条曲线的绘制问题。图9中的圆点示意了一个层次边聚合图中用于绘制样条曲线的上层结点。包括SVG在内的很多图形绘制环境都提供了对二次和三次贝塞尔曲线绘制的原生支持。如果使用三次贝塞尔曲线来分段绘制一个区间的三次样条曲线,其标准的方法是使用三次样条插值方法生成区间中

每一段三次贝塞尔曲线的控制点, 然后根据控制点绘制出所有贝塞尔曲线, 这样即可得到一条二阶连续的三次样条曲线。

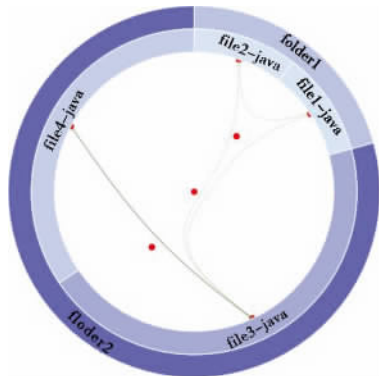


图 9 用于绘制样条曲线的上层结点

然而, 三次样条插值函数 $S(x)$ 是分段三次多项式。如果有 $n + 1$ 个结点, 则在给定区间上共划分为 n 个小区间, 每个小区间 $[x_k, x_{k+1}]$ 上的三次多项式可以写成

$$S(x) = a_i x^3 + b_i x^2 + c_i x + d_i \quad i = 0, 1, 2, \dots, n-1.$$

其中 a_i, b_i, c_i 和 d_i 都是待定系数, 每个小区间上的多项式都有 4 个待定系数, 则 $S(x)$ 共有 $4n$ 个待定系数。因为三次样条曲线在区间上的二阶导数连续, 所以在区间中每一个分段三次多项式的连接点 x_i ($i = 1, 2, \dots, n-1$) 都应满足左右两侧函数值和一阶、二阶导数的左右极限值相等, 即

$$S(k)(x_i - 0) = S(k)(x_i + 0) \quad k = 0, 1, 2.$$

这样一共可列出 $3(n - 1)$ 个方程, 再加上 $n + 1$ 个插值条件和在端点添加的 2 个边界条件, 共需要求解 $4n$ 个方程才能确定所有参数的值, 计算量较大, 而且获得的三次多项式不便于使用 SVG 中的 Path 直接进行绘制。在程序的具体实现中, 可以采用一些简化的方式来绘制样条曲线, 只需要保证曲线内部一阶导数连续即可, 这不但可以增强曲线内部的局部控制性, 而且有很好的视觉效果。

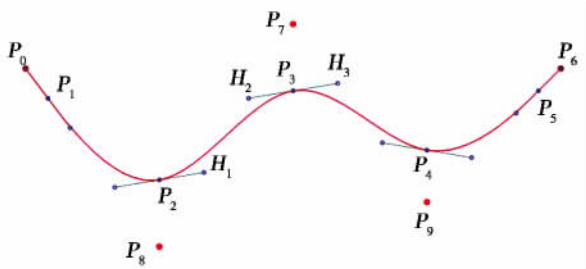


图 10 d3_svg_lineBasis 函数实现的样条曲线绘制

图 10 示出一种在 SVG 中容易绘制的算法, 图中显示了所有的控制点。图中 5 个点 P_0, P_8, P_7, P_9, P_6 是整个样条曲线的控制结点, 其他的点都是贝塞

尔曲线的控制点, 共绘制了 4 段三次贝塞尔曲线 $(P_1, P_2), (P_2, P_3), (P_3, P_4), (P_4, P_5)$ 。 (P_0, P_1) 和 (P_5, P_6) 是 2 条端点处的直线。每一段贝塞尔曲线的控制点位置由对应区间前后的 4 个样条曲线控制结点坐标通过线性变换来确定。例如, 贝塞尔曲线 (P_2, P_3) 中的 4 个控制点 P_2, H_1, H_2, P_3 仅由 P_0, P_8, P_7, P_9 的坐标确定, 与 P_6 无关, 这体现了曲线的局部控制性。每段贝塞尔曲线起止控制点两侧的控制点相对于起止控制点中心对称, 例如 H_2, P_3 和 H_3, P_3 点共线, 这样可以保证 2 段贝塞尔曲线在连接点的一阶导数连续 (因为两侧的控制点连线斜率即为一阶导数), 从而保证了曲线的平滑性。D3 已经在 `d3_svg_lineBasis` 函数中完整实现了这个绘制算法。

实用的可视化形式一般都需要良好的交互性。为解决层次边聚合图在连接线较多的情况下难以看清每一根线的起点和终点位置的问题, 可以在鼠标位于结点时, 上方高亮显示所有此结点的连接线和结点的文字描述。图 11 示出了高亮某个结点的所有连接线的状态。可以看到高亮结点之间的连接线变得更加清晰。

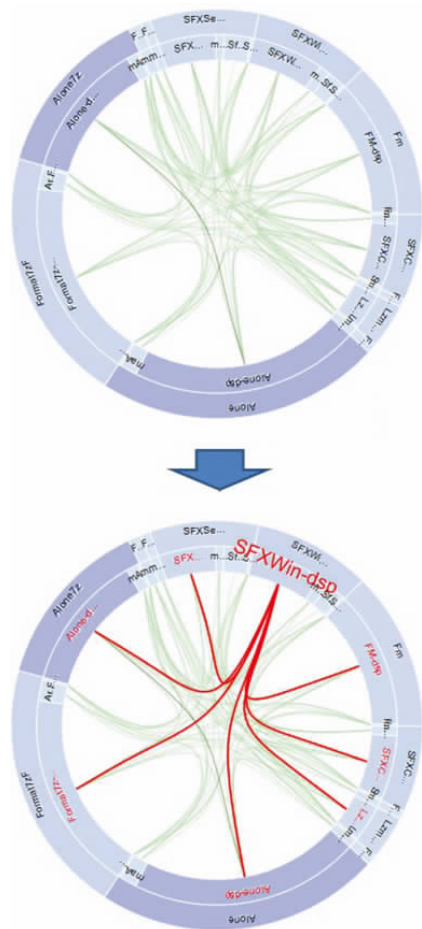


图 11 高亮显示连接线的情况
结点代表的对象名需要以文字标签的形式标

注在扇环上。如果文字太长或结点空间太小,则不同节点的字符之间可能发生重叠,影响文本的可阅读性。改进的做法是当没有足够的空间可以显示全部文字时,仅显示文本字符串的前几个字符并加上省略号,当鼠标移动到标签上方时再显示全部文字。图12示出了动态显示带省略号的标签上全部文字的例子。

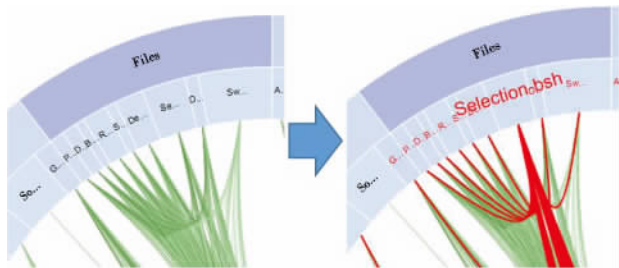


图12 动态显示带省略号的标签文字

由于层次边聚合图用扇环显示结点,很容易因为结点的文字描述有旋转角度而不便阅读;因此,为其加上自定义旋转的功能可以方便读取结点上的文字,旋转功能可以通过鼠标的拖放来实现。图13示出了旋转的效果。可以看出,有些文本在旋转前由于在环的底部,不便于阅读,旋转后显示在环的上方,更易于识别。

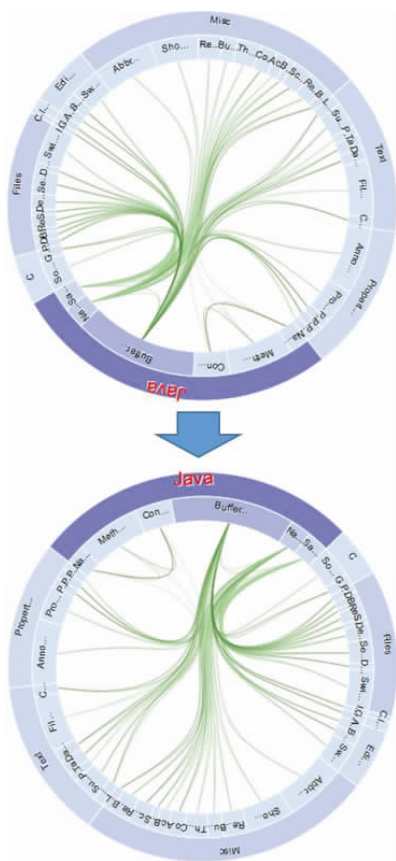


图13 旋转可以改善文本的可读性

5 可视化实验

在软件度量中,对于源程序的静态分析是一个重要的方面^[13]。源代码文件结构具有层次关系,而重复的代码文件之间具有网络关系,对代码重复的检测结果很适合用层次边聚合图进行可视化。

jEdit 是一个开源的代码编辑器。图14示出使用层次边聚合图对 jEdit 5.2.0 版源程序的 macros 文件夹进行代码重复可视化的结果。可以看到,某些比较复杂的 BeanShell 脚本文件代码重复度很高,因为很多脚本文件都有与其类似的操作。利用鼠标可以高亮与该文件有重复关系的所有文件,从而为重构这些重复代码提供便利。

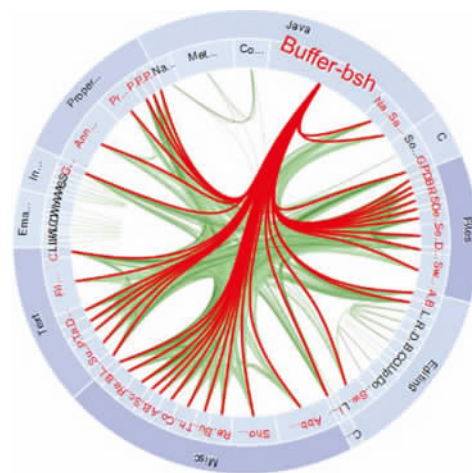


图14 对 jEdit 的 macros 文件夹重复代码可视化

Linux 内核由进程调度系统、内存管理系统、文件系统等多个模块组成^[14],开发周期较长,参与人数众多,代码重复是难以避免的问题。对 Linux 内核 2.6.32.67 版的网络模块代码重复检测结果如图15所示。从图中可以看出,虽然重复对的个数比较多,但是图中仅有一条明显的深色连接线,而其他连接线的颜色都较浅。深色的连接线代表了高重复度的代码文件对,这对文件是 \ipv4\netfilter\ip_tables.c 和 ipv6\netfilter\ip6_tables.c。iptables 是 Linux 内核用来设置、维护和检查 IP 包过滤规则的部分。由于 ipv6 和 ipv4 下的包过滤功能类似,ipv6 中的实现很显然是大量复制了 ipv4 中的代码,再略加修改完成的,这就造成了明显的代码重复。

Linux 内核支持的文件系统种类繁多,代码量庞大。文件系统的实现代码在 fs/目录下,对于不同文件系统的实现代码放在不同的子目录中。Linux 特有的文件系统是 ext2,而 ext3、ext4 都是 ext2 的改进版本。如果对这3种文件系统的代码重复进行可视化分析,将鼠标移到代表 super.c 文件

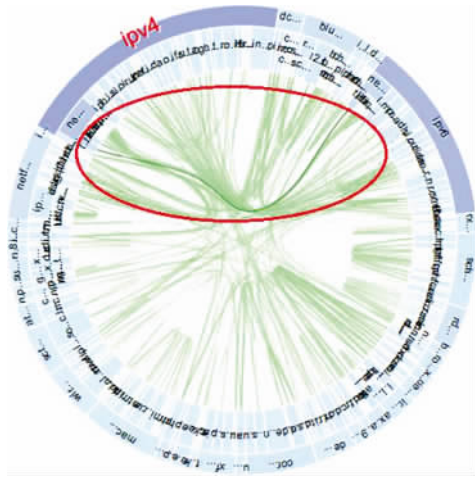
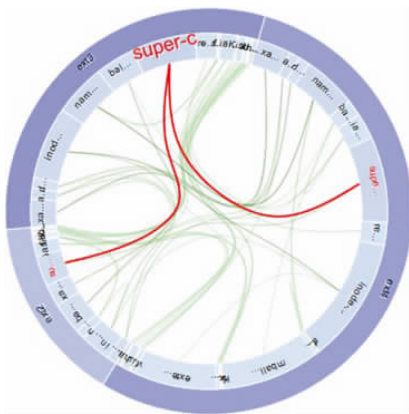
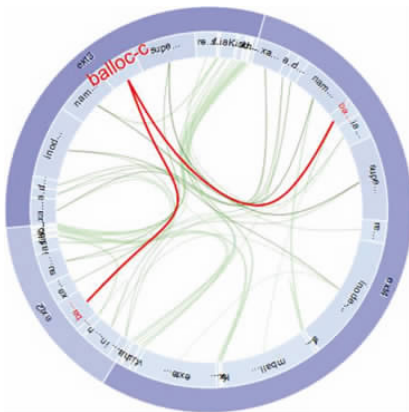


图 15 Linux 内核网络模块的代码重复

的节点上,可以看到恰好出现了 2 条高亮的连接线,分别连到了代表 ext2/super.c 的节点和代表 ext4/super.c 的节点上,如图 16(a) 所示。在图上移动鼠标可以发现 ext3 下的多个文件都有类似的情况,例如图 16(b) 所示的 balloc.c、ext3/namei.c 及 ext3/inode.c 等。这样的可视化结果说明 Linux 内核下的 ext2、ext3 和 ext4 的代码重复不是偶然现象,ext3 和 ext4 的代码很可能是通过复制 ext2 的代码再加以改进得到的。



(a) super.c 文件



(b) balloc.c 文件

图 16 层次边聚合图中显示的重复文件对

通过使用层次边聚合图进行的可视化分析易知, Linux 内核的很多代码重复都是由于比较底层的 C 语言限制了面向对象设计的应用,而为了在已有代码的基础上实现新的功能,往往直接复制了代码块再加以修改而造成的。如果可以使用较为现代的程序设计语言和模式,例如 ip6_tables 类可以继承 ip_tables 类, ext3 类可以重载 ext2 类,那么相应的代码重复将得到很大改善。如果不使用可视化工具,直接观察 Linux 的代码重复对数据列表,很难通过数据迅速发掘出深层次的问题。

6 结论

在信息可视化中,针对叶子结点之间具有网络关系的数据结构,层次边聚合图使用多层扇环表示数据结点之间的层次关系,使用连接线表示叶子结点之间的网络关系,并应用样条曲线聚合连接以减少视觉复杂度,这是一个相当高效的可视化形式。在目前几种常用的可视化技术中,将基于 SVG 的 D3 库的方式用于实现层次边聚合图比较适应 Web 可视化平台,易于产生高亮、旋转等交互性效果,其中样条曲线可以通过分段绘制贝塞尔曲线的方法来实现。在案例分析中,应用基于 D3 的层次边聚合图对 jEdit 和 Linux 的代码重复检测结果进行可视化分析,其结果验证了基于 D3 实现层次边聚合图这一可视化形式的技术可行性和该可视化形式在软件可视化领域中应用的可行性。

参 考 文 献

[1] FRUCHTERMAN T M, REINGOLD E M. Graph drawing by force - directed placement [J]. Softw ,Pract Exper ,1991 ,21 (11) : 1129.

[2] TAMASSIA R. Handbook of graph drawing and visualization [M]. [S. l.]: CRC Press ,2013: 383 - 408.

[3] MAEDA K. Visualization of code clone detection results and the implementation with structured data [J]. World Academy of Science , Engineering and Technology ,2011 ,51 (1) : 1200.

[4] 张昕,袁晓如. 树图可视化 [J]. 计算机辅助设计与图形学学报,2012 ,24 (9) : 1113.

[5] BRULS M , HUIZING K , VAN Wijk J J. Squarified treemaps [M]. [S. l.]: Springer ,2000: 2 - 3.

[6] CORNELISSEN B , ZAIMAN A , HOLTEN D , et al. Execution trace analysis through massive sequence and circular bundle views [J]. Journal of Systems and Software ,2008 ,81 (12) : 2252.

[7] CORNELISSEN B , HOLTEN D , ZAIMAN A , et al. Understanding execution traces using massive sequence and circular bundle views [J]. IEEE International Conference on Program Comprehension , 2007 ,81(12) : 49.

(下转第 50 页)

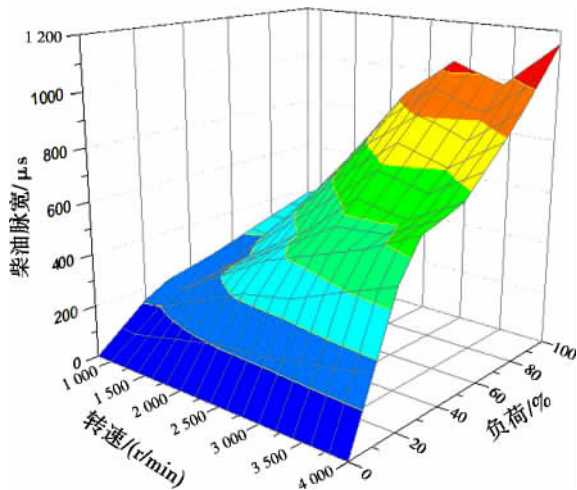


图 11 柴油喷射脉宽 MAP 图

4 结论

本文基于改造设计的柴油/甲醇双燃料发动机,设计了柴油/甲醇双燃料发动机标定系统,经过标定实验,得到如下结论。

1) 本标定系统目前已用于由 DK4A 型柴油发动机改造设计的双燃料发动机台架实验中,运行良好,可满足实验过程标定要求。

2) 本标定系统在保证发动机正常运转的条件下,可以实时改动发动机控制参数,同时实时监测发动机运行状况。

参 考 文 献

- [1]张军昌,师帅兵,张娟利. 柴油机燃用甲醇/柴油双燃料的试验研究[J]. 拖拉机与农用运输车, 2005(4):45.
- [2]刘亚哲,吴健,徐斌,等. 甲醇—柴油混合燃料在共轨发动机上的燃烧和排放特性研究[J]. 车用发动机, 2012(3):37.
- [3]KREMER F G, FACHETTI A. Alcohol as automotive fuel - Brazilian experience[J]. SAE Paper, 2000, 1: 1965.
- [4]姚春德,程传辉,王银山,等. 发动机采用柴油/甲醇组合燃烧的性能研究[J]. 工程热物理学报, 2007(1):169.
- [5]姚春德,王银山,李云强,等. 进气预混甲醇柴油机电控系统[J]. 天津大学学报, 2005(2):133.
- [6]嵇翠华,孔峰,赵不贻,等. 基于 CAN 总线的电控发动机标定系统的开发[J]. 内燃机工程, 2006(5):24.
- [7]任亮,李进,杨福源,等. 电控柴油机在线标定系统设计[J]. 内燃机工程, 2005(2):5.
- [8]张贵强,张付军,刘波澜,等. ECU 的多任务、多数据结构标定系统的研究开发[J]. 车辆与动力技术, 2004(2):22.
- [9]袁银南,朱磊,俞卫芳,等. 基于 CAN 总线的电控双燃料发动机标定系统的开发[J]. 小型内燃机与摩托车, 2006(2):8.
- [10]SOJA R, BANNOURA M. MPC5554/5553 微处理器解密[M]. 龚光华,等.译.北京:北京航空航天大学出版社, 2010:1-6
- [11]胡春明,刘娜,于勇,等. 基于 LabVIEW 的电控缸内直喷发动机标定系统的开发[J]. 拖拉机与农用运输车, 2009(2):92.

(编校:夏书林)

(上接第 33 页)

[8]HOLTEN D. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data [J]. Visualization and Computer Graphics, IEEE Transactions on, 2006, 12(5):741.

[9]WILLIAMS J L. Learning html5 game programming: A hands-on guide to building online games using Canvas, SVG, and WebGL [M]. [S. l.]: Addison-Wesley Professional, 2012:31-50.

[10]FLANAGAN D. JavaScript: The definitive guide: Activate your web pages [M]. [S. l.]: O'Reilly Media, 2011:622-666.

[11]刘旭. Chrome V8 引擎中的 JavaScript 数组实现分析与性能优化 [J]. 计算机与现代化, 2014(10):66.

能优化 [J]. 计算机与现代化, 2014(10):66.

[12]BOSTOCK M, OGIEVETSKY V, HEER J. D3 data-driven documents [J]. Visualization and Computer Graphics, IEEE Transactions on, 2011, 17(12):2301.

[13]SMITH R, HORWITZ S. Detecting and measuring similarity in code clones [C]// Proceedings of the Proceedings of the International Workshop on Software Clones (IWSC). [S. l.]: [s. n.], 2009:1-2.

[14]LOVE R. Linux Kernel Development [M]. [S. l.]: Novell Press, 2005:181-208.

(编校:饶莉)