

文章编号: 2095-2163(2019)04-0075-04

中图分类号: TP311.5

文献标志码: A

代码重构展示系统的设计与实现

石伟, 杨春花

(齐鲁工业大学(山东省科学院) 计算机科学与技术学院, 济南 250353)

摘要: 代码重构普遍存在于软件的开发维护过程中, 将重构从代码变更中检测并提取出来有利于对变更的理解。针对重构可视化研究方面的欠缺, 本文设计了一个代码重构展示系统, 实现了基于变更块的代码重构展示。通过该展示系统使代码变更中存在的重构模式变得更加简单、直观, 便于用户对代码变更的分析、理解。

关键词: 代码重构; 变更块; 展示系统

Design and implementation of code refactoring display system

SHI Wei, YANG Chunhua

(School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China)

[Abstract] Code refactoring is ubiquitous in the development and maintenance of software. Detecting and extracting refactoring from code changes facilitates understanding of the changes. Aiming at the lack of refactoring visualization research, this paper designs a code refactoring display system and realizes the display of code refactoring pattern based on hunks. Through the display system, the refactoring pattern existing in the code change becomes simpler and intuitive, which is convenient for users to analyze and understand code changes.

[Key words] code refactoring; hunk; display system

0 引言

随着软件项目的不断演化, 软件系统的维护也变得越来越困难。重构^[1]概念的提出, 为软件系统的开发维护提供了帮助。重构是通过对软件结构进行调整, 进而在软件特征功能不改变的前提下, 对软件的性能质量进行改善。代码变更每天都在产生, 由代码重构检测工具检测出的重构代码数量也是巨大的, 在面对大量的没有规律的复杂代码信息时, 将重构从代码变更中检测并提取出来有助于对代码变更的理解。然而正是由于软件规模的不断扩大, 这种理解也变得很困难。为了解决这个问题, Ware^[2]提出, 可视化方法是浏览大量数据并快速熟悉数据的首选方法。可视化技术为理解代码变更提供了便利。

现有的代码重构方法和可视化方法都有着自身的优缺点, 如何更好地运用这些可视化方法和模型来帮助代码重构分析人员观察代码重构检测结果、分析系统的重构关系及重构分布仍需要结合实际问题。现有的代码重构检测工具由于实现原理有所不同, 因此检测出的结果形式各不相同。代码重构

可视化相应的展示只侧重于代码重构数据的展示, 未对其中的重构模式进行展示, 这对分析人员查看结果并进行分析及可视化的工作都造成了困难。目前, 在重构检测及可视化方面的研究包括函数抽取重构的检测算法^[3]、基于变更相似性的跨语言克隆检测方法^[4]、基于 MVC 框架的系统重构方法研究^[5]、克隆代码可视化的研究^[6]、可视化在大数据分析实现的研究^[7]等。本文设计了一个代码重构展示系统, 基于变更块对一个源文件的 2 个连续版本中的代码变更数据进行分析处理, 实现了代码重构模式的相关展示, 使用户能够更好地分析、理解代码变更数据信息。

1 系统开发相关技术

1.1 MVC 设计模式

MVC 设计模式包括 3 个模块: 模型 (Model)、视图 (View) 和控制器 (Controller)。这 3 个模块都有各自的功能和特点, 彼此间相互独立, 改变一个模块不会影响到其它模块, 多个视图可以共享一个模型, 提高了代码的可重用性。模型用于封装应用程序的数据和用于控制、访问和修改数据的规则, 视图用于

作者简介: 石伟(1990-), 男, 硕士研究生, 主要研究方向: 软件应用技术; 杨春花(1974-), 女, 博士, 教授, 主要研究方向: 代码变更分析、软件演化等。

收稿日期: 2019-02-27

数据信息的可交互界面设计,控制器用于对请求进行响应,并根据请求进行相应的操作,完成用户请求。

1.2 GNU Diff 工具

GNU Diff 工具是一个文本差异化工具,处理输出的基本对象是一个变更块(hunk)。用于显示文件是否存在差异,并提供了一些抑制某些看起来不重要的差异的方法。最常见的差异是指文字或线条之间的空白量的变化。工具还提供了抑制字母表中的差异的方法,或者与匹配的正规表达式的行相匹配,将2个文档的内容进行比较,以使用户对文档内容信息进行分析 and 处理。

1.3 ChangeDistiller 工具

ChangeDistiller 工具是 Beat Fluri 等人^[8-9]编写的一个 Tree differ 算法。是一种基于细粒度的源代码抽象语法树差分算法。通过对代码变更前后的抽象语法树的对比,获取代码变更数据并对其进行分类。该算法将源代码变更数据定义为一个或一组基本抽象语法树的逻辑操作,通过比较2个抽象语法树节点之间的匹配和最小编辑脚本对变更数据进行提取。

1.4 XLoadTree 组件

XLoadTree 组件是基于 AJAX 和 XML 的动态加载 JS 树组件,通过对树节点的 source 属性定义指向一个 xml 文件,从而使数据进行载入,其可以包含嵌套的子节点以及指向其它 xml 文件的子节点,利用 DOM 进行转换。

1.5 Layout 布局

JQuery EasyUI 布局插件—Layout,布局是包含5个区域的容器。布局中心的区域面板是必不可少的,边缘区域面板是可供选择的,每个边缘区域面板都是可以通过拖拽边框调整边框尺寸的,并且还可以通过折叠触发器对面板区域进行折叠。布局是可以嵌套的,可以进行复杂的界面布局,通过布局将界面分为相应的区域,分别展示不同的信息。

2 系统分析与设计

基于对系统需求的综合分析,设计并实现了一个 Web 端的代码重构展示系统,将代码重构检测与可视化技术结合起来,应用到一个系统中。用户可以通过该系统对一个源文件的2个连续修改版本中的变更数据信息进行代码重构检测、变更块展示和重构模式展示等操作,使代码重构变得更加简单、直观,便于用户对存在重构的代码变更进行分析、理

解。

2.1 系统功能设计

在代码重构展示系统中,主要分为2个部分:前台和后台管理。在前台方面,未登录用户可以通过选择本地数据的修改版本数据进行文件信息浏览、变更块数据展示和重构模式展示;登录用户可以通过上传文件进行代码重构检测从而进行变更块数据展示和代码重构模式方面的相关展示。在后台管理方面,管理员可以对本地数据信息进行管理,并对用户的数据信息进行相应的管理。系统功能设计如下。

2.1.1 前台功能设计分析

(1)注册与登录:用户可以通过注册设置用户名和密码进行登录操作;

(2)修改个人信息:用户可以在登录系统之后对自己的个人信息进行修改;

(3)查询文件信息:用户可以根据源文件的修改版本和文件名对文件信息进行查询;

(4)上传文件数据:用户通过上传文件的方式,对文件进行代码重构检测,展示其变更块数据信息,如存在代码重构则进行重构模式展示;

(5)变更块展示:对存在代码变更的代码变更块数据进行相应模式的展示;

(6)重构模式展示:对代码变更中存在重构关系的重构模式进行相应的展示。

2.1.2 后台管理功能设计分析

(1)用户信息管理:管理员可以对用户信息进行管理;

(2)文件数据管理:管理员可以对数据信息进行添加、删除和修改等操作;

(3)文件数据查询:管理员可以根据源文件的修改版本和文件名对文件信息进行查询;

(4)变更块展示数据管理:管理员可以对变更块展示的数据进行管理;

(5)重构模式展示数据管理:管理员可以对重构模式展示的数据信息进行管理。

根据上述分析,可以得到系统的功能结构如图1所示。

2.2 数据库设计

在系统设计中,数据库设计是非常重要的一个环节。数据是系统设计的基础,数据库设计的合理、完善程度将直接决定系统的成败。在本系统中,数据库采用 MySQL 数据库。根据对系统功能需求的深入分析,并结合各实体之间的联系建立的。数据

库结构见表 1。

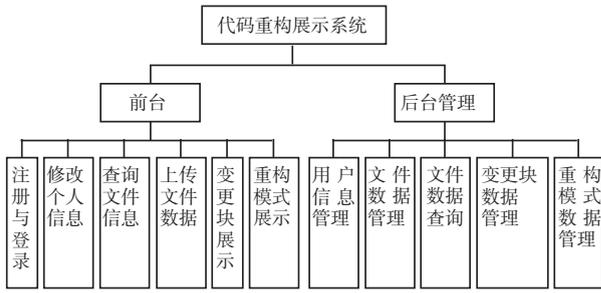


图 1 系统功能结构图

Fig. 1 Structure diagram of system function

表 1 数据库结构

Tab. 1 Database structure

表名	字段名
users	id, user_name, user_password, email
admin	id, admin_name, admin_password, email
revisions	id, author_id, commit_date, message
actions	id, revision, author_id, file_id, commit_date,
files	change_date, file_name, type
authors	id, file_name, file_path, type
contents	id, author, email
diffs	id, file_id, commit_id, file_content, type
hunks	id, file_id, commit_date, type, diff_content
	id, diff_id, add_line, reduce_line, hunk_content,

(1) 用户信息表 (users): 用于记录前台用户的基本数据信息;

(2) 管理员信息表 (admin): 用于记录管理员的基本数据信息;

(3) 版本信息表 (revisions): 用于记录版本库中版本数据的数据信息;

(4) 变更记录表 (actions): 用于记录版本信息的变更操作数据信息;

(5) 文件表 (files): 用于记录版本库中的所有文件数据信息;

(6) 开发者表 (authors): 用于记录项目版本开发者数据信息;

(7) 内容表 (contents): 用于记录在版本库中首次添加的文件内容数据信息;

(8) 变更内容表 (diffs): 用于记录一个源文件的 2 个连续版本之间的 diff 数据信息;

(9) 变更块数据表 (hunks): 用于记录代码变更的变更块及其行号范围等数据信息。

3 系统实现

该代码重构展示系统应用 B/S 架构, 采用 MVC 设计模式, 利用 JSP+JavaBean+Servlet 技术设计实现。系统整体由数据表现层、业务逻辑层和数据访问层 3 个功能模块组成, 使系统结构更加清晰, 运行效率更高, 降低了系统层与层之间的耦合性, 便于系统运行维护。其中, 数据表现层由 JSP 实现, 业务逻辑层由 JavaBean 实现, 数据访问层由 Servlet 实现。在系统实现过程中, 主要包括前台和后台 2 个管理模块。系统前台模块主要是接收用户的请求, 实现相应数据的展示功能。后台管理模块主要是用于对相关数据的管理。

3.1 前台功能实现

主要实现相关数据的展示和用户发送数据请求:

(1) 查询文件信息。用户通过源文件的修改版本和文件名对文件信息进行查询, 并可以查看文件的变更块展示及其是否存在重构, 如存在则可查看其重构模式展示;

(2) 上传文件数据。用户登录后可以通过上传文件数据, 利用 ChangeDistiller 工具、GNU Diff 工具和 Levenshtein 算法对文件数据进行重构检测。然后, 进行相应的代码变更块展示和重构模式展示;

(3) 变更块展示。利用 GNU Diff 工具对一个源文件的 2 个连续修改版本文件进行变更数据获取, 获取 hunk, 形成 hunk 集, 进行变更块展示;

(4) 重构模式展示。利用获取的 hunk 数据进行代码重构识别, 提取相关重构参数和变更块数据, 选取相应的重构模式展示模块, 进行重构模式展示。

3.2 后台管理功能实现

主要实现本地数据的获取、查询、分析、管理等功能:

(1) 文件数据管理。本地数据来源是利用 MiniGit 工具获取的开源项目数据, 并将数据进行分析、处理之后存储到本地数据库。该模块实现了本地数据的管理并对用户上传的文件信息进行管理, 可以对相关数据进行删除、修改和查询等功能;

(2) 变更数据管理。实现了对系统实现过程中的变更数据的管理。通过该模块可以实现数据的删除、修改和查询等功能, 以便更好地对变更数据进行管理。

系统主要功能运行界面如图 2、3 所示。

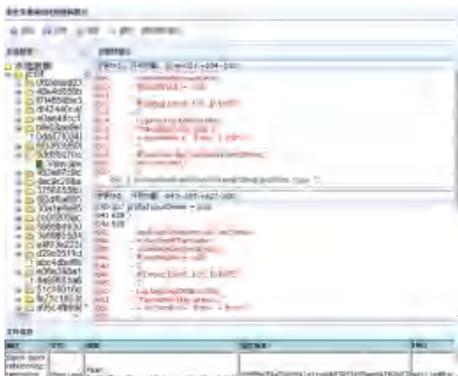


图2 变更块展示
Fig. 2 Hunk display

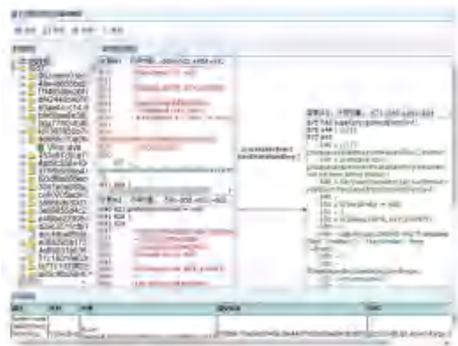


图3 重构模式展示
Fig. 3 Refactoring pattern display

4 结束语

代码重构展示系统实现了代码变更块和重构模

式的相关展示,实现了对一个源文件的2个连续修改版本的代码重构检测和重构模式展示。系统使用目前较成熟的框架和开发工具进行开发,为后期系统维护和后续开发提供了便利。

参考文献

- [1] FOWLER M, BECK K, BRANT J, et al. Refactoring: Improving the design of existing code[M]. Sebastopol, CA: Addison-Wesley Professional, 1999.
- [2] WARE C. Information visualization: Perception for design[M]. 2nd ed. San Francisco: Elsevier, 2004.
- [3] 刘阳,刘秋荣,刘辉. 函数抽取重构的自动检测方法[J]. 计算机科学, 2015, 42(12): 105-107.
- [4] 柳萌宇,钟浩,于海波. 基于变更相似性的跨语言克隆检测方法[J]. 计算机与现代化, 2016(4): 79-84, 99.
- [5] 李景焰, 芮文承, 胡伶俐, 等. 基于MVC框架的系统重构方法研究与实践[J]. 电脑知识与技术, 2015, 11(22): 63-66.
- [6] 何蕾. 克隆代码可视化系统的设计与实现[D]. 哈尔滨: 哈尔滨工业大学, 2015.
- [7] 王博, 沈方方. 可视化在大数据分析领域的实现要点[J]. 中国新通信, 2017, 19(14): 72.
- [8] FLURI B, GALL H C. Classifying change types for qualifying change couplings[C]//ICPC '06 Proceedings of the 14th IEEE International Conference on Program Comprehension. Washington, DC, USA: IEEE, 2006: 35-45.
- [9] FLURI B, WÜRSCH M, PINZGER M, et al. Change distilling: Tree differencing for fine-grained source code change extraction[J]. IEEE Transactions on Software Engineering, 2007, 33(11): 725-743.

(上接第74页)

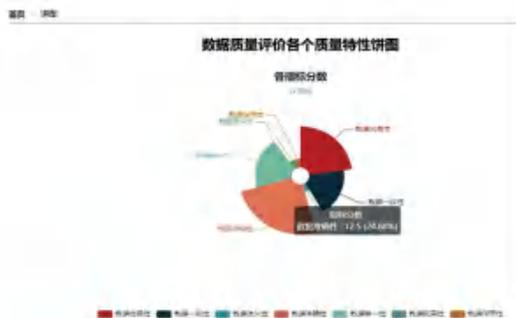


图4 指标分数玫瑰图
Fig. 4 Index Fractional Rose Map

5 结束语

本文研究了数据质量现状和方向,分析了目前数据质量研究存在的不足。针对存在的问题,首先利用GB/T来替代ISO标准,其次综合多种方法计算权重,最后设计并开发了系统。详细展示了数据质量评价流程和评价方法的使用,对评价结果进行了可视化图表展示,对系统存在的问题也进行了进

一步的优化,验证了方案和系统的可用性及高效性。

参考文献

- [1] Jaime Campos, Pankaj Sharma, Unai Gorostegui Gabiria, Erkki Jantunen, David Baglee. A Big Data Analytical Architecture for the Asset Management[J]. Procedia CIRP, 2017, 64.
- [2] 张元新. 网络开源数据可用性评价系统[D]. 哈尔滨工业大学, 2017.
- [3] 杨忠胜. 数据质量中精确性的研究[D]. 哈尔滨工业大学, 2013.
- [4] 延婉梅. 动车组大数据清洗关键技术研究与实现[D]. 北京交通大学, 2015.
- [5] Suvendu Chandan Nayak, Chitaranjan Tripathy. Deadline sensitive lease scheduling in cloud computing environment using AHP[J]. Journal of King Saud University - Computer and Information Sciences, 2018, 30(2).
- [6] 张炳江. 层次分析法及其应用案例[M]. 北京: 电子工业出版社, 2014.
- [7] Michael W. Grenn, Shahram Sarkani, Thomas Mazzuchi. The requirements entropy framework in systems engineering[J]. Systems Engineering, 2014, 17(4).
- [8] 汤永利, 徐国爱, 钮心忻, 等. 基于信息熵的信息安全风险模型[J]. 北京邮电大学学报, 2008(2): 50-53.
- [9] 文军. 基于熵权法的航空公司绩效评价研究[J]. 科学技术与工程, 2009, 9(22): 6939-6941.
- [10] 李大为. 基于B/S架构的绩效考核管理系统设计与实现[D]. 吉林大学, 2018.