

数据整理脚本语义的可视化

罗中粟^{1,3)}, 熊凯^{2,3)}, 傅四维³⁾, 王永恒^{3)*}, 鲍虎军^{2,3)}, 巫英才^{2,3)}

¹⁾ (浙江工业大学计算机科学与技术学院 杭州 310023)

²⁾ (浙江大学 CAD&CG 国家重点实验室 杭州 310058)

³⁾ (之江实验室 杭州 311121)

(wangyh@zhejianglab.com)

摘要: 理解数据整理脚本的语义是数据工作者的常见需求。然而, 数据整理操作的类型及其代码的实现方式复杂多样, 使得数据工作者在理解脚本语义时费时费力。该工作通过收集数据工作者在理解数据整理脚本语义上的具体需求, 设计并实现了 1 个基于概览和细节模式的交互式可视分析系统 ChangeVis, 以帮助数据工作者理解表格在数据整理过程中的变化。ChangeVis 包含 4 个视图, 分别可视化代码块中的表结构变化、行列信息变化、单元格数据变化以及执行的数据转换操作的语义。通过案例分析和用户实验, 验证了 ChangeVis 在帮助数据工作者理解数据整理脚本语义的可用性和有效性。

关键词: 数据整理; 程序可视化; 可视化设计; 表格数据可视化

中图法分类号: TP391.41 DOI: 10.3724/SP.J.1089.2022.19901

Visualizing the Semantics of Data Wrangling Scripts

Luo Zhongsu^{1,3)}, Xiong Kai^{2,3)}, Fu Siwei³⁾, Wang Yongheng^{3)*}, Bao Hujun^{2,3)}, and Wu Yingcai^{2,3)}

¹⁾ (College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023)

²⁾ (State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310058)

³⁾ (Zhejiang Laboratory, Hangzhou 311121)

Abstract: Understanding the semantics of data wrangling scripts is a common requirement for data workers. However, due to the complexity and diversity of the types of data transformations and their parameters, understanding the semantics of wrangling scripts is time-consuming and laborious. We worked with four data workers and developed an interactive visualization system, called ChangeVis, to help data workers understand the changes of tables in the process of data wrangling. Based on the Focus+Context technique, ChangeVis scales well and illustrates the changes of table structure, the semantics of data transformations, the changes of column information, and the changes of cell data on a selected code snippet. We used two use cases and a semi-structured lab study to demonstrate the usability and effectiveness of ChangeVis.

Key words: data wrangling; program visualization; visualization design; tabular data visualization

基金项目: 国家自然科学基金青年项目(62002331); 国家自然科学基金面上项目(62072400); 之江实验室开放项目(2021KE0AC02)。罗中粟(1997—), 男, 硕士研究生, 主要研究方向为可视分析、数据整理; 熊凯(1997—), 男, 博士研究生, 主要研究方向为可视分析、数据整理; 傅四维(1990—), 男, 博士, 副研究员, 主要研究方向为人机交互、可视分析、多模态交互; 王永恒(1973—), 男, 博士, 论文通讯作者, 副研究员, 主要研究方向为大数据分析、智能决策、人机交互; 鲍虎军(1966—), 男, 博士, 教授, CCF 常务理事, 主要研究方向为计算机图形学、虚拟现实、计算机视觉; 巫英才(1983—), 男, 博士, 教授, 主要研究方向为可视分析、信息可视化、人机交互;

数据整理是为了满足数据分析等任务要求对数据进行转换、整合及错误纠正的数据预处理过程^[1]。在工作中,数据工作者常使用脚本语言(如 R, Python 等)完成数据整理工作,数据整理脚本通常由多个代码块组成,每个代码块是一段独立的数据处理逻辑,通常包含多步数据转换操作。数据工作者在众多场景中都需要阅读并理解这些代码块中数据整理的语义^[2],如对代码进行调试、检查、复用等。

然而,理解代码的语义要求数据工作者熟悉相应的脚本语言及其使用的函数包,且理解代码的过程费时费力、容易出错^[3-5]。对于数据整理的代码而言,实现数据转换操作的编程方式繁多,同一个数据转换操作,可以有不同的实现方式,如 Python 语言中,使用 `fillna` 和 `replace` 都可以完成对缺失值的替换。并且,同一个函数在使用不同参数时,可能会执行不同的数据转换操作,如 R 语言中的 `select` 函数,既可以用作对列重新排序操作,也可以用作删除列操作。这些都使得数据工作者理解数据整理代码的语义更加困难。

近年来,许多工作致力于协助数据工作者理解数据整理脚本。一些工作,如 `WrangleDoc`^[2],利用文字概述的方式来描述数据表在数据整理前后发生的变化,以帮助数据工作者理解代码块对数据表的影响。然而,文本描述的方式并不直观。因此,近年来,有不少工作擅长展示单步数据转换操作的可视化,但它们通常可扩展性较差,不适用于展示一整段代码块的语义。如 `SOMNUS`^[6]利用可视化展示数据整理脚本中的数据转换操作的语义。

为了解决以上问题,本文开发了 1 个基于概览和细节模式的交互式可视化系统 `ChangeVis`^①,以帮助数据工作者理解数据整理代码块的语义。该系统包含利用 4 个不同细节层次的视图,即概览视图、语义视图、统计视图和数据视图。概览视图用来展示数据整理脚本中整个数据转换过程的表结构变化概况,并支持数据工作者选择不同的代码块;语义视图阐明所选代码块中数据转换操作的具体语义;统计视图可视化所选代码块中初始表与最终表的列统计信息,同时呈现该数据整理过程中的行列变化信息;数据视图为数据工作者提供某一步数据转换操作所涉及的具体表格数据。同时,该系统支持合并具有相同数据转换语义的重复操作,以简化可视化元素,提高数据工作者的

理解效率。

最后,本文提供了 2 个案例,分别展示 `ChangeVis` 系统在帮助数据工作者理解脚本语义和检查数据整理脚本方面的应用,并通过用户实验验证系统的可用性和有效性。

1 相关工作

1.1 数据整理

近年来,研究者设计开发了用于辅助完成数据整理任务的各类系统^[1,7-12]。`Wrangler` 系统^[1,7]基于用户的交互行为推荐合适的的数据转换操作。`Wrex`^[8]和 `foofah`^[9]使用了样例编程的方法协助用户完成数据整理任务。除此之外,还有研究者设计了能够对网络结构进行数据整理的工具,如 `Origraph`^[10]等。另外,有一部分研究使数据工作者能直接从不同的网站获取数据进行数据整理任务,提升了数据整理中数据获取阶段的效率,如 `Dataxformer`^[11], `WebRelate`^[12]等。这些工作使数据工作者完成数据整理任务更加方便,但对于解释数据转换操作的语义却十分有限。

为了帮助数据工作者理解数据整理脚本,一些工具通过文本概述的方法描述脚本的数据转换语义。`WrangleDoc`^[2]通过归纳数据表经过代码块操作后的变化来帮助用户理解数据整理脚本,发现脚本中存在的细小问题。`Unravel`^[13]通过拖放和切换交互来帮助数据工作者探索和理解链式结构代码块,并提供一段自然语言文本来解释每步数据转换操作的语义。这些工作通过文本概述的方法描述代码块,缺少直观的可视化表达。

还有一些工具基于规则解析数据脚本代码,并可视化不同数据转换操作的语义。例如,`SOMNUS`^[6]通过基于图形图符的节点链接图可视化数据整理脚本的数据转换过程。`Datamations`^[14]和 `Data Tweening`^[15]则是利用动画的形式解释每一步的数据转换的语义。然而,这些工作均是针对单步数据转换操作的可视化,对于多步操作的语义可视化仅是单步可视化的拼接。这种拼接使得数据工作者需要逐一阅读,阻碍了理解速度。因此,本文设计了一种多步数据转换语义的可视化以提高数据工作者理解代码块的效率。

1.2 表格数据可视化

表格数据是一种被广泛应用于各个领域的数据结构^[16],由于表格数据分析在各领域起到重要

①<https://changevis.github.io/>

作用, 很多可视化的工作服务于表格数据^[17].

Furmanova 等^[17]将表格数据可视化分为三种类型: (1) 基于概览技术的可视化^[18-22], 基于表格数据中的数值进行可视化; (2) 基于投影技术的可视化^[23], 常用于可视化高维数据; (3) 基于表格式技术的可视化^[24-26], 会基于数据和类型进行相应的可视化, 并保留原有的表格数据结构. 由于每种技术各有其优缺点, 因此也有混合使用这几类技术的可视化工作^[17,27-29]来消除单一技术的不足.

以上的工作着重于对表格数据内容的可视化, 缺乏表格数据变化的展示. TACO^[16]通过使用时间轴概览、差异直方图和热力图可视化表格数据随时间发生的演变, 但是这种演变并不适用于可视化数据整理任务中的数据转换操作的语义. 因此, 本文设计了一种混合技术的表格数据可视化, 展现经过数据转换操作后表格数据发生的变化, 以此来揭示数据整理脚本中代码块的语义.

1.3 代码可视化

近年来不少工作利用可视化展示代码和数据的细节以增强用户对计算机程序的理解^[30], 从而帮助用户学习^[31-35]、调试^[36-41]代码. 部分可视化工具能够帮助用户学习编程语言及代码库. Online Python Tutor^[31]通过可视化代码运行时的状态、数据结构等信息, 帮助新手用户学习 Python 语言. Khaloo 等^[32]使用 VR 技术将代码库中的信息进行可视化, 相比于阅读传统文档, 可视化的方式更具有吸引力. 有的可视化工具^[33,34]通过展现每步代码的行为帮助用户学习算法. CodeMetropolis^[35]则是将代码中的类、变量等信息可视化为三维实体, 更为形象地展示代码结构等信息.

还有一些可视化工作旨在帮助用户对代码进行调试. 根据代码自动生成流程图的可视化工具^[36,37], 能帮助用户了解代码的执行逻辑, 从而对代码进行调试. Moseler^[38]等工作致力于可视化并发式的 Java 代码, 通过可视化各线程中的执行情况, 协助用户对代码进行调试及优化. 一些工作通过可视化代码中的数据结构^[39]或代码结构^[40,41]帮助用户调试代码.

以上工作致力于对代码逻辑、数据结构等信息的可视化, 以帮助用户学习和调试代码. 然而这类可视化工作不能解释数据整理代码的语义. 数据整理代码包含的数据转换操作繁多, 实现方式也复杂多样, 且代码中的变量为二维数据表, 用以上工作的可视化设计难以展示. 因此本文针对数据

整理代码的特性及数据工作者的需求, 设计并开发了 1 个能够帮助数据工作者理解数据整理代码语义的可视化系统.

2 设计需求

本文的设计目标是帮助数据工作者理解脚本中代码块的语义. 为了了解数据工作者的需求, 本文邀请了 4 位数据工作者, 2 位是在国家实验室工作的数据分析师, 2 位为计算机专业的学生, 他们都拥有三年以上的数据整理经验. 本文通过访谈获取他们在理解数据整理脚本语义时所采用的方法及面临的问题, 最后收集并整理出他们在完成该任务上的需求.

R1. 展示数据整理脚本中代码块的语义. 数据工作者需要理解代码块对数据表操作的语义, 即对表中哪些行/列做了何种数据转换操作. 通过展示操作的语义可以帮助数据工作者更好地完成代码检查、复用等工作.

R2. 提供能够展示数据转换操作语义的数据. 数据工作者常常会输出经过一段代码块的前后表, 通过对比表中数据来推断其中包含的数据转换操作. 但数据表常包含许多与操作无关的数据. 通过展示只与数据转换操作相关的数据示例, 能够提升数据工作者的推断效率.

R3. 展示数据整理过程中数据表结构的变化. 数据表的结构(即表的行列)变化可以揭示数据转换操作的类型, 如新增列、删除行等, 从而为数据工作者提供当前脚本内数据转换操作的概览, 以帮助他们选择感兴趣的代码块.

R4. 提供数据表在代码块执行前后各列数据的统计信息. 数据工作者需要对比代码块的输入表与输出表中列数据发生的变化, 以辅助数据工作者对操作语义的理解. 列数据的变化通常可由统计信息来描述, 其中数据分布、极值、离群值等是数据工作者常关注的重点.

R5. 展示数据转换操作后数据表中发生变化的数据及其占总数据的比例. 相同的代码对于不同的数据表可能会影响不同的数据. 数据工作者需要了解代码实际改变了数据表中哪些数据及多少数据, 以帮助他们掌握数据的变化.

3 ChangeVis 系统

为了满足以上需求, 本文开发了如图 1 所示的交互式系统 ChangeVis. ChangeVis 包含了 4 个视图: 概览视图(A 部分)展示数据表在数据整理代码块中的表结构变化; 语义视图(B 部分)可视化所选代码块内的数据转换操作, 帮助用户理解代码块的语义; 统计视图(C 部分)呈现代码块中初始表和最终表中各列的统计信息, 同时展示该段代码块中数据表的行列变化; 数据视图(D 部分)用于展示经过数据转换操作前后的详细单元格数据.

ChangeVis 采用了 SOMNUS^[6]中程序适配器(Program Adaptor)的设计来解析数据整理脚本中的代码. 该程序适配器分为三步: (1) 执行数据整理脚本以获取每行数据整理代码的输入表及输出表; (2) 解析每行数据整理代码以获取函数名、函

数参数、输入表及输出表的变量名; (3) 根据函数名、函数参数以及输入输出表的内容推断该行代码所执行的数据转换操作的类别. 基于 Guo 等^[7]定义的不同数据转换操作以及 Kasica 等^[42]总结的数据转换操作的类型空间, ChangeVis 组合并筛选了其中的 23 种常见的数据转换操作进行解析和可视化.

数据转换操作对表中的行列及单元格数据可造成 4 种类型的变化, 即数据的新增、删除、修改和不变. 考虑一致性, 系统在不同视图中统一使用绿色(新增)、红色(删除)、蓝色(修改)和灰色(不变)来分别编码这 4 种变化. 除此之外, 为了提升用户理解数据整理脚本语义的效率, ChangeVis 使用一套合并规则(见第 3.5 节)对代码块中重复的数据转换操作在可视化时进行合并来精简可视化的内容, 减少用户的阅读负担.

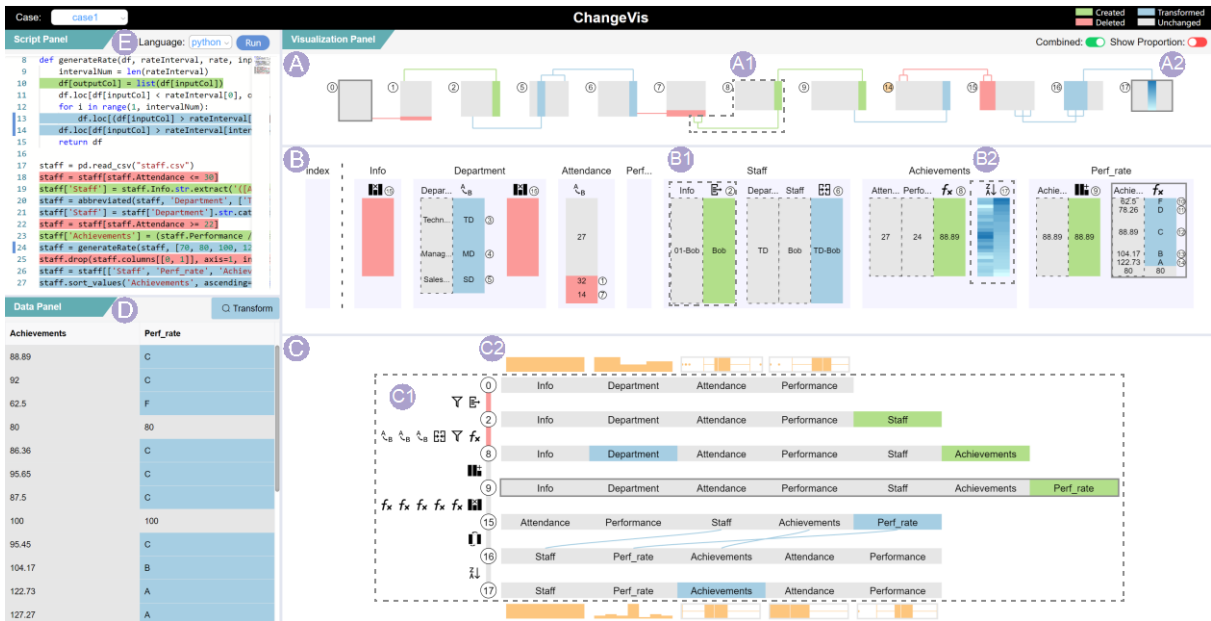


图 1 ChangeVis 系统界面

3.1 概览视图

概览视图利用一条表结构的变化链路为数据工作者提供当前脚本内数据转换操作的概览(R3), 如图 1 中 A 部分所示. 链路中的节点展示某步数据转换操作后数据表的结构变化, 节点使用表示数据变化类型的颜色对相应数据进行填充; 连接节点的边表示完成该数据转换操作涉及的行/列数据. 如图 1 中 A1 部分所示, 边的起始点连接该步操作输入表中的 2 列, 终止点连接输出表中的最后一列, 同时该列及边的颜色为绿色, 表示该操作根据输入表中的 2 列新增了最后 1 列.

由于排序操作会改变整张数据表中行数据的位置, 而对整个节点都进行颜色填充是无意义的, 因此 ChangeVis 针对排序操作只对排序列使用颜色的深浅渐变表示数据的排序情况. 如图 1 中 A2 部分, 填充的颜色从上至下由深变浅, 代表该步为降序操作.

此外, 用户可以点击节点进行交互, 选择感兴趣代码块的起始和终止节点, 脚本面板(图 1 中 E 部分)中会将对应操作的代码根据表数据变化的类型分别用不同的颜色进行高亮, 同时下方的语义视图和统计视图将只展示选中代码块内的信息.

由于程序所执行的数据转换操作的顺序与脚本中代码的实际编写顺序可能不一致(如自定义函数的调用会导致代码实际执行的顺序发生跳跃), 当用户选择一段代码块时, 其对应的数据转换操作可能并不连续, 无法构成一条完整的数据转换链路. 因此 ChangeVis 暂时不支持通过选择脚本面板中的代码进行探索.

3.2 语义视图

语义视图用来展示所选代码块中数据表的每一列所执行的数据转换操作的具体语义(R1). 视图中的紫色块表示数据表中的列, 紫色块中的操作子图表示一步数据转换操作, 如图 1 中的 B 部分所示, Staff 列中共包含 2 个操作子图, 表示 Staff 列共经历了 2 步数据转换操作.

下文将以图 1 中 B1 部分为例, 详细阐述每个操作子图需展示的 6 部分信息及其可视化方式: (1) 该步数据转换操作执行的主体列, 即实际发生数据变化的列, 用 1 个竖直矩形框表示, 放置在操作子图中的最右侧(如图 1 中 B1 部分的 Staff 列); (2) 完成数据转换操作的依赖列, 即影响主体列发生变化所依赖的数据列, 用带虚线的灰色竖直矩形框表示, 以区分主体列, 放置在操作子图的左侧. 同时, 虚线框的上方标注依赖列的列名(如图 1 中 B1 部分的 Info 列); (3) 该数据转换操作在所选代码块中的步骤序号, 将该步骤序号标注在主体列的右上方(如图 1 中 B1 部分的数字 2); (4) 该数据转换操作的类型, 用 1 个能表达该操作的语义的图标表示, 放置在主体列的正上方(如图 1 中 B1 部分的“抽取”图标). ChangeVis 为每一种数据转换操作类型匹配了 1 个语义图标(23 种数据转换操作的类型、定义描述、及其语义图标详见补充材料); (5) 数据转换操作对主体列中实际新增、修改、删除及不变的数据占比(R5), 在主体列中用不同的区域块展示, 每个区域块根据其发生数据变化的类型用对应的颜色进行填充(如图 1 中 B1 部分的绿色块). 此外, 由于缺失值是数据工作者的 1 个重要观察信息, 因此对于单元格数据为缺失值的区域块用白色背景单独编码; (6) 展示主体列中不同区域块实际变化的 1 个数据示例(如图 1 中 B1 部分的“Bob”), 同时在依赖列中相应的位置展示该数据示例对应的依赖列数据(如图 1 中 B1 部分的“01-Bob”). 主体列与依赖列中的数据示例一定程度上揭示了操作的语义(R2). 因此, 结合图 1 中 B1 部分的该 6 部分信息, 可以推出该数据转换操作的语义为“抽

取”Info 列的子字符串为新列 Staff, 且该操作填充了新列 Staff 中的所有单元格数据.

对于不涉及任何主体列的数据转换操作, 即操作对象为数据表中的行, 如删除表中的重复行等, ChangeVis 引入索引列(Index)作为此类操作的主体列, 并放置于语义视图的最左侧, 用 1 条虚线将右侧数据表中的实际列隔开. 若该类行操作有指定具体的行号, 将会在索引列的右侧使用带有具体行号标注的三角形符号展示该步操作所涉及的行号, 如图 2 中 a 部分. 若对行进行了新增或删除操作, 此步操作后的所有操作子图中竖直矩形框的高度也将发生相应的改变.

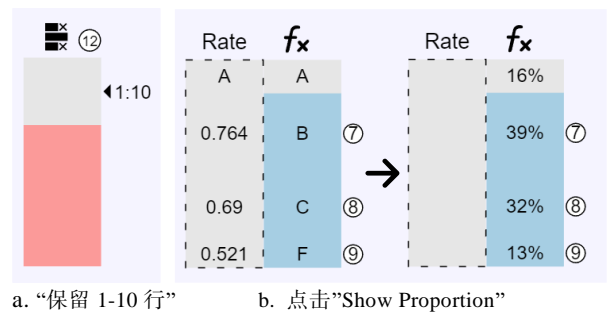


图 2 操作子图

由于排序操作不会对表内的值造成改变, 而是修改数据的位置信息. 为了展示排序操作的语义, ChangeVis 在语义视图中利用颜色的深浅变化来展示排序操作后数据位置发生的变化. ChangeVis 将主体列拆成左右两部分, 分别表示排序操作前后的单元格数据, 并用颜色的深浅编码数值的大小. 如图 1 中 B2 部分所示, 主体列(即 Achievements 列)的左半部分的颜色分布是混乱的, 表示 Achievements 列在操作前是无序状态, 而主体列的右半部分的颜色分布为由深到浅, 表示 Achievements 列在操作后为降序状态. 这种可视化设计不仅能展示排序操作的语义, 还能揭示该列数据的分布情况.

为了更直接地展示数据转换操作后数据表发生变化的数据占比, ChangeVis 支持用户点击图 1 中可视化面板右上角的“Show Proportion”按钮, 以查看操作子图中各个数据区域块的具体比例(R5), 如图 2 中 b 部分所示.

3.3 统计视图

统计视图包含操作时间轴线图(图 1 中 C1 部分)和列数据统计图(图 1 中 C2 部分)2 个部分, 分别用来展示代码块内的行列变化信息(R3)和初始表与最终表的列统计信息(R4).

操作时间轴线图的起始点和终止点分别表示该段代码块的输入表和输出表. 起始点与终止点间的节点用来展示当前代码块内引起表中的列发生变化(即列的新增、列的删除、列名的修改或列位置的重新排布)的数据转换操作. 节点内的数字表示该数据转换操作的步骤序号. 节点的右侧展示该步数据转换操作后数据表的列, 用列的背景颜色编码上一个节点到该节点中列数据的变化. 为了展示相邻节点的列排布位置的变化, 使用连接线将位置发生变化的列连接起来. 节点之间轴线的填充颜色, 编码这 2 个节点所代表的操作后数据表中行数目的变化, 而节点之间的左侧操作图标表示这 2 张数据表中发生的数据转换操作类型.

列数据统计图位于操作时间轴的首尾两端, 分别用来展示该段代码块的初始表与最终表中各列的统计信息. 针对不同数据类型的列, Change-Vis 使用不同类型的统计图进行展示: (1) 利用箱线图可视化数值类型的数据. 箱线图可以展示 1 组数值的分布区间、极值、离群值等; (2) 利用条带图可视化时间类型的数据. 条带图可展示时间的

连续分布情况; (3) 利用柱状图展示字符串类型的数据. 柱状图可以展示各个类别数据重复出现的个数, 以此观察是否包含重复值. 鼠标悬浮在统计图上可详细查看该列的统计信息, 如缺失值的个数、具体的离群值等.

3.4 数据视图

当用户点击语义视图中的操作子图, 数据视图, 如图 1 中 D 部分所示. 将展示该步数据转换操作所涉及的所有单元格数据(R2), 以更详细地揭示数据的变化. 数据视图中呈现的列为操作子图中所有的依赖列及主体列. 若存在某依赖列为主体列的情况(即该操作对主体列做的转换, 依赖于该列本身), 数据视图将这 2 列的列名分别添加“old”和“new”加以区分. 若该操作的主体列为索引列, 则数据视图将显示数据表中所有的列. 数据视图中单元格的背景颜色编码数据变化的类型. 此外, 视图的右上方会根据数据变化的类型提供对应颜色的筛选按钮, 用户可点击该按钮以筛选出符合此变化类型的单元格数据.

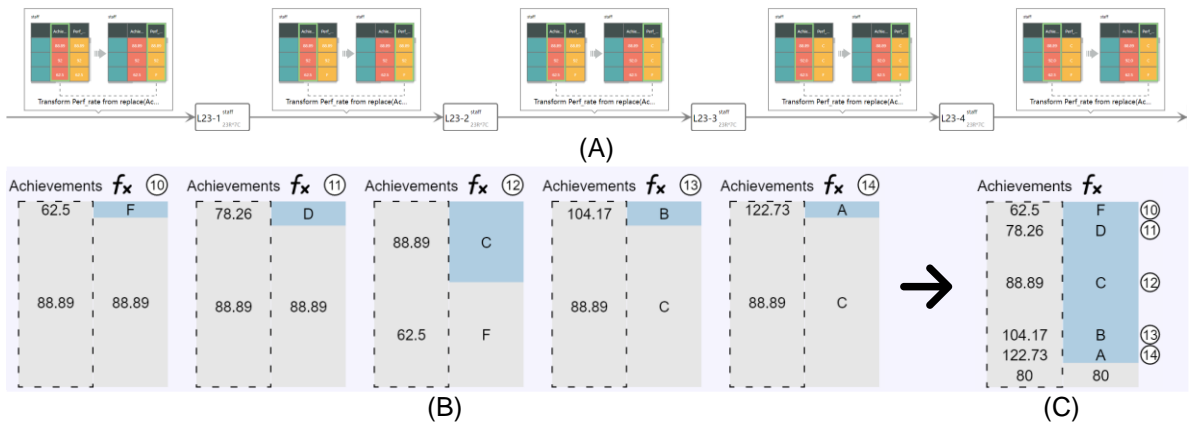


图 3 语义视图的设计迭代 (A)SOMNUS 图符, (B)ChangeVis 操作子图, (C)ChangeVis 合并后的操作子图

3.5 设计迭代

设计 ChangeVis 时本文与 4 位数据工作者反复沟通, 迭代多轮. 例如, 在设计语义视图时, 首先参考 SOMNUS^[6]使用独立的图符来可视化数据转换操作的语义(如图 3 中 A 部分所示, 每个图符包含 6 列数据), 但这种长链路的可视化表达占用了较多空间, 使得在有限的空间内图符难以清楚展示. 为了缩减展示内容, ChangeVis 使用的操作子图仅展示与数据转换操作相关的列数据(如图 3 中 B 部分所示, 每个操作子图包含 2 列数据), 从而使可视化效果更加精简. 但即使如此, ChangeVis 仍面临着当操作步骤较多时, 操作子图过长的问题. 考虑到数据整理脚本中常包含重复操作, 数据工

作者表示在阅读这些操作时理解一步操作的语义即可推广至其他重复操作. 因此, 为了缓解子图过长的问题, 本文提出了 1 种将重复操作合并的优化策略, 该策略包含以下 4 个条件: (1) 操作的主体列为同一列; (2) 操作的依赖列相同; (3) 操作类型相同; (4) 操作间没有出现改变依赖列或主体列数据的操作. 点击 ChangeVis 右上方的“Combined”按钮即可将满足条件的操作进行合并. 合并后概览视图中重复操作对应的节点将会合成 1 个节点, 并且语义视图中重复操作对应的操作子图将生成 1 个新的合并操作子图, 操作子图中原本位于右上方的步骤序号将移至相应区域块的右侧(如图 3 中 C 部分所示). 通过合并操作使可视化更紧凑, 进

一步减少用户的阅读时间, 加快用户对语义理解的效率。

4 系统评估

为了验证 ChangeVis 系统对数据工作者理解数据整理脚本语义的帮助, 本文通过 2 个数据整理任务案例和 1 个用户实验来探究 ChangeVis 的有效性和可用性。

4.1 案例 1——复用数据整理脚本

本案例为数据工作者在复用代码时需要理解脚本中代码块语义的场景。某公司每月需要对各部门员工的出勤情况和工作绩效进行评估, 数据工作者需要对考勤系统中的数据进行数据整理以完成之后的数据可视分析工作。

目前有 1 份使用 Python 语言对类似数据进行数据整理的脚本, 脚本中包含了 2 个自定义函数。然而, 该数据工作者熟悉的脚本语言为 R, 理解不熟悉的脚本语言编写的代码存在困难。数据工作者通过 ChangeVis 系统对该段数据整理脚本进行可视化, 以理解脚本中代码块的语义, 如图 1 所示。

数据工作者在概览视图中通过点击 2 个节点选择了 1 段代码块(图 1 中 A 部分)进行探索, 通过概览视图数据工作者发现在这段代码块内包含了新增列、转换列、删除列以及对数据表进行排序的操作(R3)。数据工作者通过观察下方的统计视图(图 1 中 C 部分)发现在第 14 步时删除了 Info 和 Department 列, 并在第 15 步进行了列重排操作, 将 Staff 和 Perf_rate 列移至表的最左侧(R3)。数据工作者阅读语义视图(图 1 中 B 部分), 发现在这段代码块中存在合并操作, 但通过阅读脚本并未发现重复的函数, 因此数据工作者点击了语义视图中 Perf_rate 列的操作子图, ChangeVis 的脚本面板中(图 1 中 E 部分)标注出相应的数据转换操作代码, 发现合并的操作为代码中的自定义函数代码块; 数据工作者通过点击 ChangeVis 右上角“Show Proportion”按钮查看每步数据转换操作实际影响数据的比例(R5)。数据工作者通过脚本面板(图 1 中 E 部分)中的代码以及 Perf_rate 列的操作子图中的示例数据(R2), 推断出此步操作为根据设定的

分类区间及 Achievements 列内的数据在 Perf_rate 中生成相应的标签(R1)。数据工作者点击了第 10 步操作对应的颜色块, 在数据视图内(图 1 中 D 部分)看到了详细的数据, 通过点击视图右上角的蓝色按钮(图 1 中 D1 部分), 筛选出经过代码块后改变的数据检查是否符合预期的结果(R2)。

数据工作者在阅读合并后的语义视图时发现, 在 Perf_rate 列中, 有部分的数据并没有发生改变, 不符合操作的预期, 因此数据工作者点击合并后的操作子图在脚本面板和数据面板中进行检查。发现在自定义函数 generateRate 中数据筛选条件的两端都设置为闭区间, 导致了等于筛选条件的数据并未能被成功转换。通过这一操作展现了 ChangeVis 可以帮助用户检查脚本中的隐藏问题。

4.2 案例 2——检查数据整理脚本

本案例为数据工作者在编写数据整理脚本时需要理解脚本语义并检查是否完成预期中的数据整理工作。数据工作者现在有 1 份二手车的成交数据, 他需要对不同车企、型号、生产年份的二手车保值率进行分析。

在完成 1 段数据整理脚本后, 数据工作者使用 ChangeVis 系统对脚本进行检查, 如图 4 所示。数据工作者点击“Combined”按钮将视图精简。在快速浏览了语义视图及各操作的数据示例后, 判断当前脚本的数据转换操作符合预期(R1, R3)。但当他阅读统计视图时发现了问题, Launch_Year 列中出现了离群值(R4)如图 4 中 C 部分, 而汽车的生产年份不应该存在较大的差距。同时数据工作者发现在初始表中 Launch_Date 列的数据类型就是字符型, (图 4 中 B 部分)并未被正确识别为日期型。数据工作者检查语义视图中的数据示例, 发现提取操作并没有问题(R1, R2), 他通过点击第 4 步(图 4 中 A 部分)的操作子图, 使用数据视图查看详细数据(R2), 发现在数据表中日期的格式是不同的(图 4 中 D 部分), 有的使用了年/月/日的格式, 而有少量数据却使用了月/日/年的格式, 由于操作的提取规则为提取第 1 个 ‘/’ 符号前的数据, 从而导致部分数据被提取的是月份而不是年份。



图 4 案例 2——检查数据整理脚本

4.3 用户实验

(1) 实验数据. 本文为实验准备了 3 份数据整理案例, 其中 1 份作为训练案例, 用于介绍系统及让参与者熟悉任务; 另外 2 份作为实验案例, 其脚本和数据表格分别来自 4.1 和 4.2 节的 2 个案例.

表 1 用户探索任务

任务编号	任务描述
T1	描述指定代码块中数据转换操作的语义
T2	描述指定代码块中数据表行列的变化
T3	检查指定代码块中输入表和输出表的列是否存在离群值或重复值
T4	找出指定代码块中只修改了列的部分数据的操作, 并描述修改数据占总数据的比例

(2) 参与者. 共有 12 名数据工作者(8 男, 4 女)参与了本次实验, 其中 5 名为拥有两至三年工作经验的数据分析师, 7 名为日常工作中需要完成数据整理任务的计算机专业学生; 实验的参与者均拥有数据整理的基础知识.

(3) 任务设计及实验流程. 在实验开始前, 本文首先向参与者描述了实验目的, 并收集参与者的个人基本信息. 然后使用训练案例向参与者详细介绍 ChangeVis 系统, 并让参与者使用 ChangeVis 完成 1 组探索任务, 如表 1 所示. 在熟悉系统及任务后, 参与者需要对 2 个实验案例分别完成 1 组探索任务. 接着, 参与者需要填写 1 份调查问卷, 如表 2 所示. 参与者根据问卷中的描述使用 7 级李克特量表对 ChangeVis 系统进行评分; 1

分代表对描述非常不认同, 7 分代表对描述非常认同. 最后, 参与者提出对 ChangeVis 系统的使用反馈及改进建议. 整个实验过程大约为 40 分钟, 实验结束后, 每位参与者获得了 20 元的报酬.

表 2 用户调查问卷

序号	问卷内容
Q1	系统设计美观
Q2	系统能展示单步数据转换操作的语义
Q3	系统能帮助你高效的理解一段代码块中数据整理的语义
Q4	从系统中你能获取数据表变化的信息
Q5	系统能帮助你获得当前数据转换操作导致数据的改变及其比例
Q6	系统能帮助你完成日常工作中理解数据整理脚本的需求

(4) 结果分析. 图 5 展示参与者根据调查问卷对 ChangeVis 系统的评分结果, 其中误差条表示 95% 的置信区间. 参与者认为可以快速准确的获得表变化信息和实际操作数据的比例(Q4, Q5), 这些信息对理解数据整理语义及检查操作起到积极作用. 在展示单步的数据转换操作时, ChangeVis 获得了相对较低的评分(Q2), 是由于 ChangeVis 注重展示的是一段代码块内的操作, 而对于单步操作的展示较为简单. 此外, 参与者认为本系统和日常工作使用的系统不兼容, 因此很难在工作中使用, 导致了 Q6 的评分相对较低. 从用户实验结果中可以得出, ChangeVis 能够高效地帮助数据工作者理解数据整理脚本中代码块的语义.

(5) 实验反馈. 参与者对 ChangeVis 系统提出了以下反馈及建议: a. 系统的学习成本较高. 由于 ChangeVis 中的视图较多且语义视图和统计视图包含的信息量较大, 需要花费较多时间进行学习; 在未来工作中应该增加系统中对视图的教学引导, 以帮助用户缩短学习成本. b. 希望加强视图间的联系. 用户在理解数据转换操作时需要同时阅读多个视图中的内容, 但这些内容在不同视图中难以对应. c. 展示操作的语义图标不够直观. 参与者认为部分图标难以清楚地表达数据转换操作的语义, 如抽取操作的图标, 不能直观体现出抽取子字符串操作的语义. 未来可以通过悬浮弹窗, 使用文字描述操作的语义解决这一问题. d. 提供语义视图中数据转换操作执行顺序的引导. 通过操作子图中右上方的步数来确定执行顺序的方式并不直观. 为了解决该问题, 在未来工作中可加强语义视图内各操作子图与概览视图中对应节点之间的关系, 促进用户对操作执行顺序的理解.

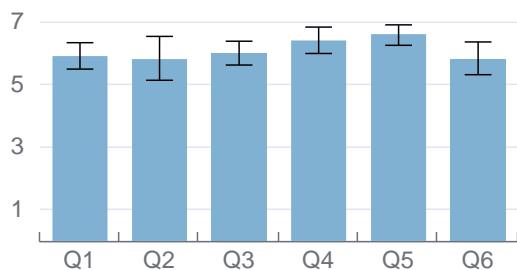


图5 用户实验结果

5 局限性与未来工作

本文工作的局限性有以下4点:

(1) 不支持包含多张输入表或输出表操作的可视化. 由于目前该系统是通过单张数据表中各列内的操作子图来展示数据转换操作的语义, 因此对于包含多表的操作, 如左连接等合并表的操作, 无法用本文的可视化方法来展示该操作的语义. 在未来的工作中将设计1种能够展示操作多张输入输出表的语义可视化方法.

(2) 不支持规则外的函数及数据转换操作的解析. 本文系统中目前只支持23种常用数据转换操作及其部分函数代码的解析, 对于规则外的函数或操作将会识别失败. 为了增强代码解析器的鲁棒性, 未来可以通过优化解析脚本内代码的方式, 并结合机器学习方法以适配更丰富的函数和数据转换操作.

(3) 表格行列过多时会导致视图宽高超限. 本

文系统的设计会展示数据表格中每1列数据转换操作的语义, 且操作子图的高度编码了表格的行数量. 因此, 当行列数量过多时, 系统会出现视图的宽高超限的问题. 在未来工作中计划利用 collapse-to-zoom^[43] 技术解决该问题.

(4) 缺乏与现有工作的对比. 本文通过用户实验对 ChangeVis 进行评估, 但未能与 SOMNUS^[6] 等系统进行量化对比实验. 未来考虑增加与相关系统进行对比实验, 进一步验证 ChangeVis 在帮助用户理解代码块语义上的有效性.

6 结语

本文通过收集数据工作者在理解数据整理脚本语义上的具体需求, 设计并实现了1个基于概览和细节模式的交互式可视分析系统. 本文通过4个视图展示数据整理脚本中执行的操作和对数据表造成的影响, 此外本文设计了1种合并规则帮助用户进一步提升阅读效率. 最后通过案例分析和用户实验验证系统的有效性和可用性. 未来考虑继续优化设计, 丰富代码解析器支持的操作及函数, 以支持对更多操作的展示.

参考文献(References):

- [1] Kandel S, Paepcke A, Hellerstein J, *et al.* Wrangler: interactive visual specification of data transformation scripts[C] //Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York: ACM Press, 2011: 3363–3372.
- [2] Yang C, Zhou S, Guo J L C, *et al.* Subtle Bugs Everywhere: Generating Documentation for Data Wrangling Code[C] //Proceedings of the 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE). Los Alamitos: IEEE Computer Society Press, 2021: 304–316.
- [3] Olson G M, Sheppard S, Soloway E. Can principles of cognition lower the barriers to programming[M]. Empirical Studies of Programmers: Second Workshop. Intellect Books, 1987: 248–263.
- [4] Qian Y, Lehman J. Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review[J]. ACM Transactions on Computing Education, 2017, 18(1): 1–24.
- [5] Sorva J, Karavirta V, Malmi L. A Review of Generic Program Visualization Systems for Introductory Programming Education[J]. ACM Transactions on Computing Education, 2013, 13(4): 1–64.
- [6] Xiong K, Fu S, Ding G, *et al.* Visualizing the Scripts of Data Wrangling with SOMNUS[J]. IEEE Transactions on Visualization and Computer Graphics, IEEE Computer Society: To Appear.
- [7] Guo P J, Kandel S, Hellerstein J M, *et al.* Proactive wrangling: mixed-initiative end-user programming of data transformation scripts[C] //Proceedings of the 24th annual ACM symposium on User interface software and technology. New York: ACM Press, 2011: 65–74.
- [8] Drosos I, Barik T, Guo P J, *et al.* Wrex: A Unified Programming-by-Example Interaction for Synthesizing Readable Code for Data Scientists[C] //Proceedings of the

- 2020 CHI Conference on Human Factors in Computing Systems. New York: ACM Press, 2020: 1–12.
- [9] Jin Z, Anderson M R, Cafarella M, *et al.* Foofah: Transforming Data By Example[C] //Proceedings of the 2017 ACM International Conference on Management of Data. New York: ACM Press, 2017: 683–698.
- [10] Bigelow A, Nobre C, Meyer M, *et al.* Origraph: Interactive Network Wrangling[C] //Proceedings of the 2019 IEEE Conference on Visual Analytics Science and Technology (VAST). Los Alamitos: IEEE Computer Society Press, 2019: 81–92.
- [11] Abedjan Z, Morcos J, Ilyas I F, *et al.* DataXFormer: A robust transformation discovery system[C] //Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering (ICDE). Los Alamitos: IEEE Computer Society Press, 2016: 1134–1145.
- [12] Inala J P, Singh R. WebRelate: integrating web data with spreadsheets using examples[C] //Proceedings of the ACM on Programming Languages. New York: ACM Press, 2017, 2: 1–28.
- [13] Shrestha N, Barik T, Parnin C. Unravel: A Fluent Code Explorer for Data Wrangling[C] //Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology. New York: ACM Press, 2021: 198–207.
- [14] Pu X, Kross S, Hofman J M, *et al.* Datamations: Animated Explanations of Data Analysis Pipelines[C] //Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. New York: ACM Press, 2021: 1–14.
- [15] Khan M, Xu L, Nandi A, *et al.* Data tweening: incremental visualization of data transforms[C] //Proceedings of the VLDB Endowment. New York: ACM Press, 2017, 10: 661–672.
- [16] Niederer C, Stitz H, Hourieh R, *et al.* TACO: Visualizing Changes in Tables Over Time[J]. IEEE Transactions on Visualization and Computer Graphics, 2018, 24(1): 677–686.
- [17] Furmanova K, Gratzl S, Stitz H, *et al.* Taggle: Combining overview and details in tabular data visualizations[J]. Information Visualization, SAGE Publications, 2020, 19(2): 114–136.
- [18] Claessen J H T, van Wijk J J. Flexible Linked Axes for Multivariate Data Visualization[J]. IEEE Transactions on Visualization and Computer Graphics, 2011, 17(12): 2310–2316.
- [19] Fua Y-H, Ward M O, Rundensteiner E A. Hierarchical parallel coordinates for exploration of large datasets[C] //Proceedings of the Visualization '99. Los Alamitos: IEEE Computer Society Press, 1999: 43–508.
- [20] Yalçın M A, Elmqvist N, Bederson B B. Keshif: Rapid and Expressive Tabular Data Exploration for Novices[J]. IEEE Transactions on Visualization and Computer Graphics, 2018, 24(8): 2339–2352.
- [21] Luo X, Yuan Y, Zhang K, *et al.* Enhancing statistical charts: toward better data visualization and analysis[J]. Journal of Visualization, 2019, 22(4): 819–832.
- [22] Wei Y, Mei H, Huang W, *et al.* An evolutionary model for operation-driven visualization design[J]. Journal of Visualization, 2022, 25(1): 95–110.
- [23] Liu S, Maljovec D, Wang B, *et al.* Visualizing High-Dimensional Data: Advances in the Past Decade[J]. IEEE Transactions on Visualization and Computer Graphics, 2017, 23(3): 1249–1268.
- [24] Gratzl S, Lex A, Gehlenborg N, *et al.* LineUp: Visual Analysis of Multi-Attribute Rankings[J]. IEEE Transactions on Visualization and Computer Graphics, 2013, 19(12): 2277–2286.
- [25] Pajer S, Streit M, Torsney-Weir T, *et al.* WeightLifter: Visual Weight Space Exploration for Multi-Criteria Decision Making[J]. IEEE Transactions on Visualization and Computer Graphics, 2017, 23(1): 611–620.
- [26] Zhang C, Chen Y, Yang J, *et al.* An association rule based approach to reducing visual clutter in parallel sets[J]. Visual Informatics, 2019, 3(1): 48–57.
- [27] Lex A, Schulz H-J, Streit M, *et al.* VisBricks: Multiform Visualization of Large, Inhomogeneous Data[J]. IEEE Transactions on Visualization and Computer Graphics, 2011, 17(12): 2291–2300.
- [28] Lex A, Streit M, Schulz H-J, *et al.* StratomeX: Visual Analysis of Large-Scale Heterogeneous Genomics Data for Cancer Subtype Characterization[J]. Computer Graphics Forum, 2012, 31(3pt3): 1175–1184.
- [29] Stahnke J, Dörk M, Müller B, *et al.* Probing Projections: Interaction Techniques for Interpreting Arrangements and Errors of Dimensionality Reductions[J]. IEEE Transactions on Visualization and Computer Graphics, 2016, 22(1): 629–638.
- [30] Price B A, Baecker R M, Small I S. A Principled Taxonomy of Software Visualization[J]. Journal of Visual Languages & Computing, 1993, 4(3): 211–266.
- [31] Guo P J. Online python tutor: embeddable web-based program visualization for cs education[C] //Proceedings of the 44th ACM technical symposium on Computer science education. New York: ACM Press, 2013: 579–584.
- [32] Khaloo P, Maghouthi M, Taranta E, *et al.* Code Park: A New 3D Code Visualization Tool[C] //Proceedings of the 2017 IEEE Working Conference on Software Visualization (VISSOFT). Los Alamitos: IEEE Computer Society Press, 2017: 43–53.
- [33] Hansen S, Narayanan N H, Hegarty M. Designing Educationally Effective Algorithm Visualizations[J]. Journal of Visual Languages & Computing, 2002, 13(3): 291–317.
- [34] Demetrescu C, Finocchi I, Stasko J T. Specifying Algorithm Visualizations: Interesting Events or State Mapping?[M]. Software Visualization. Berlin, Heidelberg: Springer, 2002: 16–30.
- [35] Balogh G, Beszédes Á. CodeMetropolis — A minecraft based collaboration tool for developers[C] //Proceedings of the 2013 First IEEE Working Conference on Software Visualization (VISSOFT). Los Alamitos: IEEE Computer Society Press, 2013: 1–4.
- [36] Kosower D A, Lopez-Villarejo J J, Roubtsov S. Flowgen: Flowchart-Based Documentation Framework for C++[C] //Proceedings of the 2014 IEEE 14th International Working Conference on Source Code Analysis and Manipulation. Los Alamitos: IEEE Computer Society Press, 2014: 59–64.
- [37] Cheon J, Kang D, Woo G. VizMe: An Annotation-Based Program Visualization System Generating a Compact Visualization[C] //Proceedings of the International Conference on Data Engineering 2015 (DaEng-2015). Singapore: Springer, 2019: 433–441.
- [38] Moseler O, Kreber L, Diehl S. The ThreadRadar visualization for debugging concurrent Java programs[J]. Journal of Visualization, 2022: 1–23.
- [39] Kumar N S, Revanth Babu P N, Sai Eashwar K S, *et al.* Code-Viz: Data Structure Specific Visualization and Animation Tool For User-Provided Code[C] //Proceedings of the 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON). Los Alamitos: IEEE Computer Society Press, 2021: 1–8.
- [40] Jbara A, Agbaria M, Adoni A, *et al.* ICSD: Interactive Visual Support for Understanding Code Control Structure[C] //Proceedings of the 2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER). Los Alamitos: IEEE Computer Society Press, 2019: 644–648.
- [41] Hori A, Kawakami M, Ichii M. CodeHouse: VR Code Visualization Tool[C] //Proceedings of the 2019 Working Conference on Software Visualization (VISSOFT). Los Alamitos: IEEE Computer Society Press, 2019: 83–87.
- [42] Kasica S, Berret C, Munzner T. Table Scraps: An Actionable Framework for Multi-Table Data Wrangling From An Artifact Study of Computational Journalism[J]. IEEE Transactions on Visualization and Computer Graphics, 2021, 27(2): 957–966.
- [43] Baudisch P, Xie X, Wang C, *et al.* Collapse-to-zoom: viewing web pages on small screen devices by interactively removing irrelevant content[C] //Proceedings of the 17th annual ACM symposium on User interface software and technology. New York: ACM Press, 2004: 91–94.

作者姓名 (按论文署名 顺序填写)	单位	工作邮箱 (域名为工作单位的邮箱)	特殊情况说明 (包括单位邮箱涉密不可 公开、非员工无单位邮箱等 情况请说明)
罗中粟	浙江工业大学 计算 机科学与技术学院	2112012227@zjut.edu.cn	
熊凯	浙江大学 CAD&CG 国家重点实验室	kaixiong@zju.edu.cn	
傅四维	之江实验室	siwei.fu@zhejianglab.com	
王永恒	之江实验室	wangyh@zhejianglab.com	
鲍虎军	浙江大学 CAD&CG 国家重点实验室	bao@cad.zju.edu.cn	
巫英才	浙江大学 CAD&CG 国家重点实验室	ycwu@zju.edu.cn	