

基于 Web 的多刻面交互式特征定位工具 MFIE

彭 鑫 王金水 付 焜 赵文耘

(复旦大学计算机科学技术学院 上海 200433) (上海市数据科学重点实验室(复旦大学) 上海 200433)

摘 要 在执行软件维护任务中,开发人员经常需要在软件代码中寻找并理解与给定的功能性特征相关的程序元素(如类或方法),这一过程称为特征定位或概念定位。相关的经验研究表明,特征定位是一个以人为中心、信息密集型的探索和认知过程,包含交互式的信息探索、反馈和策略调整。基于这一思想,提出了一种多刻面、交互式的特征定位方法,并开发了基于 Web 的支持工具 MFIE(Multi-faceted Interactive Explorer)。介绍了 MFIE 所实现的多刻面、交互式特征定位方法,MFIE 的多刻面界面设计以及所提供的主要功能。在此基础上,还通过一个案例介绍了 MFIE 所支持的特征定位过程。

关键词 特征定位,多刻面,交互,Web,工具

中图分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.09.014

MFIE: A Web-based Multi-faceted Interactive Feature Localization Tool

PENG Xin WANG Jin-shui FU Kun ZHAO Wen-yun

(School of Computer Science, Fudan University, Shanghai 200433, China)

(Shanghai Key Laboratory of Data Science, Fudan University, Shanghai 200433, China)

Abstract In performing software maintenance tasks, developers often need to find and understand program elements (e. g. classes or methods) that are relevant to a given feature (called feature location or concept location). Some empirical studies have shown that feature location is a human-centric and information-intensive process with interactive exploration, feedback and strategy adjustment. Based on this idea, we proposed a multi-faceted and interactive features localization approach, and developed a Web-based supporting tool MFIE (Multi-Faceted Interactive Explorer). This paper introduced the features localization approach supported by MFIE, its multi-faceted interface design and other characteristics. Furthermore, the paper also described the usage of MFIE with a use case of feature location.

Keywords Feature location, Multi-facet, Interactive, Web-based, Tool

1 引言

在执行软件维护任务时,开发人员经常需要在软件代码中寻找并理解与给定的功能性特征相关的程序元素(如类或方法),这一过程一般被称为特征定位(feature location)或概念定位(concept location)^[1]。由于软件系统的规模和复杂性日益增长,在大量的源代码中进行特征定位变得十分困难且耗时。为此,研究者们提出了一系列基于程序的文本信息、结构依赖、动态运行轨迹等信息的特征定位方法和工具。

我们此前关于特征定位的经验研究^[2]表明,特征定位是一个以人为中心、信息密集型的探索和认知过程。在此过程中开发人员以一种交互式的方式查找并浏览一个多维度的信息空间,同时根据每一步的反馈不断调整自身的探索策略(例如所使用的关键字、所依循的依赖关系路径等)。然而,现有的特征定位方法(例如文献[3,4])一般都是非交互式的,遵循

“查询→结果”的一次性(one-shot)过程,即根据开发人员的输入一次性地产生经过排序的候选结果列表。在此过程中,开发人员经常会感到难以提供一个合适的特征查询(如特征描述或关键字),同时难以判断所返回结果的相关性,特别是在返回结果数量特别多的时候。

针对以上这些问题,我们提出了一种多刻面、交互式的特征定位方法,并开发了基于 Web 的支持工具 MFIE(Multi-Faceted Interactive Explorer)。该方法允许开发人员通过一种多刻面的智能化搜索界面对可能相关的程序元素进行交互式的分组、排序和过滤,从而辅助他们探索并理解相关的程序元素。该方法开始于一个开发人员给定的初始的特征查询,然后以一种交互式以及迭代的过程实现特征定位目标。在迭代过程的每一步中,该方法通过程序分析以及数据挖掘技术从当前返回的候选程序元素中抽取或挖掘多个语法及语义刻面,包括包结构、继承结构、使用依赖以及内涵等。这些多维

到稿日期:2013-11-10 返修日期:2014-01-28 本文受教育部高校博士点基金(20100071110031),国家高技术研究发展计划 863 项目(2012AA011202)资助。

彭 鑫(1979—),男,博士,副教授,主要研究方向为需求工程、软件维护、自适应软件系统,E-mail: pengxin@fudan.edu.cn; **王金水**(1981—),男,博士生,主要研究方向为软件维护; **付 焜**(1989—),男,硕士生,主要研究方向为软件维护; **赵文耘**(1964—),男,硕士,教授,主要研究方向为软件工程、电子商务。

度的信息以一种多侧面信息检索的用户界面^[5]呈现,允许开发人员快速、准确地判断所返回结果的相关性并决定是否通过刻画选取、关键字调整等方式继续精化特征查询方案。该方法的详细介绍、实验分析与相关问题讨论可以参考文献[6]。

本文的后续章节将首先概述多侧面、交互式的特征定位方法,然后介绍工具的界面布局 and 主要功能,并通过一个使用过程示例说明工具的使用过程。

2 方法概述

MFIE 所支持的多侧面、交互式的特征定位方法如图 1 所示^[6],其中灰色矩形代表的 4 个步骤表示用户交互的步骤。目前,MFIE 所实现的特征定位过程以方法为基本的程序单元。

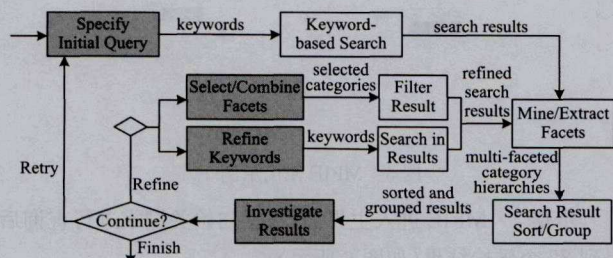


图 1 多侧面交互式特征定位方法概览^[6]

整个特征定位过程开始于开发人员所提供的初始查询关键字。基于该初始查询,MFIE 通过基于关键字的查找自动返回一个候选结果集(方法集合)。基于候选结果集中的方法,MFIE 通过程序分析和数据挖掘产生一个多侧面的分类层次,并基于此分类层次对所有候选结果进行分组和排序。开发人员通过浏览经分组和排序后的候选结果决定下一步的选择,即结束特征定位过程、使用新的关键字重新开始查询、继续精化原有查询。开发人员可以通过两种方式精化原有特征查询:通过多侧面分类结构组合选取侧面分类;精化原有的查询关键字。通过这两种方式之一精化原有查询后,MFIE 在当前候选结果基础上通过查询获得精化后的查询结果。基于精化后的查询结果集合,MFIE 再次抽取并挖掘多侧面分类结构,并进入下一轮迭代过程。

目前,MFIE 能够提供 3 个方面共 5 种不同的侧面,即 2 个结构侧面(包结构、继承层次)、1 个内涵侧面和 2 个依赖侧面(使用、被使用)。下面介绍这些侧面的含义。

1)包结构(Package Structure)侧面:包结构侧面是从候选元素中抽取包结构,反映了候选程序元素是如何分布在各个包里面。

2)继承层次(Inheritance Hierarchy)侧面:继承层次侧面反映了候选程序元素在类型层次中的继承关系。

3)内涵(Intent)侧面:内涵侧面是通过对候选程序元素的代码文本进行语义聚类而构建起来的,反映了与候选程序元素相关的主题(topic)或关注点(concern)。

4)使用(Use)侧面:使用侧面按照候选程序元素所依赖的模块对候选结果进行分类,其中的模块是通过对候选程序元素的依赖上下文进行语义聚类之后得到的。

5)被使用(UsedBy)侧面:被使用侧面按照依赖于候选程

序元素的模块对候选结果进行分类,其中的模块是通过对候选程序元素的依赖上下文进行语义聚类之后得到的。

3 工具简介

MFIE 是一个基于 Web 的代码浏览和特征定位工具,具有典型的多侧面信息检索界面。本章将介绍 MFIE 的界面布局以及所提供的主要功能。

3.1 界面布局

MFIE 的主界面如图 2 所示^[6],是一种典型的多侧面查询界面布局。整个界面可以划分为 4 个主要的区域,分别是查询面板(Search Panel)、查询结果面板(Result Panel)、侧面面板(Facet Panel)和历史浏览器(History Explorer)。

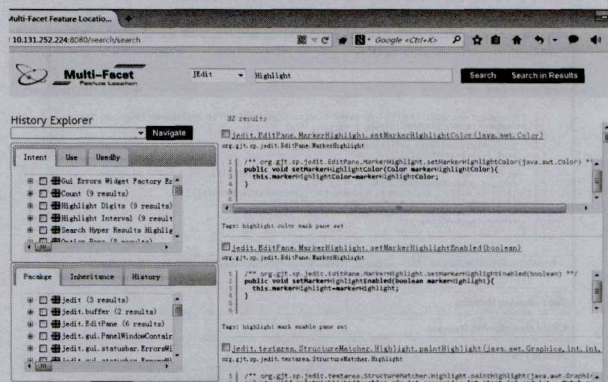


图 2 MFIE 界面布局^[6]

其中,查询面板支持关键字查询。其中的“Search”和“Search in Results”按钮分别表示重新开始一次特征定位过程(即图 1 中的 Retry)和精化现有查询关键字(即图 1 中的 Refine Keywords)。查询结果面板列出当前的候选结果,包括方法名、代码摘要、自动生成的代码标签等,点击方法名还可以查看完整的方法代码。侧面面板展示了 MFIE 当前所支持的 5 个侧面,点击某个侧面下的一个分类,则查询结果面板相应更新为与此分类相关的候选结果。通过侧面面板还可以进行侧面分类选取,点击“Update”按钮后可以实现候选结果的精化查询。历史浏览器中记录了当前特征定位过程的整个历史信息,通过选择历史浏览器中的过程结点可以回退到此前的任意一个步骤上继续探索过程。

3.2 结果浏览和侧面选取

在整个迭代式的特征定位过程中,根据相关度排序后的候选方法列表都显示在 MFIE 的查询结果面板中。由于完整的方法体可能占用过多的空间,同时妨碍开发人员定位关键词所在的位置,因此 MFIE 从每个方法的代码中截取了一个代码片断作为摘要。该代码片断是以方法代码中首个与关键词匹配的位置为中心,截取前后大约 250 个单词而生成的。除了对代码片断中的关键词进行高亮外,MFIE 对代码进行语法标识。通过不同的颜色,用户可以很清晰地识别代码中的各个语法单元,如关键词、数字、注释等。为了帮助开发人员快速地对查询结果有个大概的了解,MFIE 通过 TF/IDF (Term Frequency-Inverse Document Frequency)度量从各个候选方法中抽取 5 个最高权重的单词作为该方法的标签。

在 MFIE 结果列表中点击一个方法名链接就可以查看该方法的详细信息视图(见图 3)。开发人员可以通过该视图查

看程序元素的详细信息,例如完整的源代码、调用该方法以及被该方法调用的其他方法、所访问的成员变量等。

为了帮助开发人员更好地分析或优化查询结果,MFIE 提供了即时查询预览的功能。当开发人员选中某个刻面上的分类结点时,MFIE 会立即显示所有与选中分类相关的查询结果。通过这种方式,开发人员可以方便地尝试不同刻面的不同类别的各种组合,从不同维度对查询结果进行分析。需要说明的是,服务器端并没有响应即时查询的操作,因此当开发人员决定选用某些刻面的类别对查询结果进行优化时,他需要点击“Update”按钮,将选择的类别信息提交给服务器。之后,服务器端才会根据新的查询结果为各个刻面生成新的分类树,并将该探索步骤记录在历史探索器中。

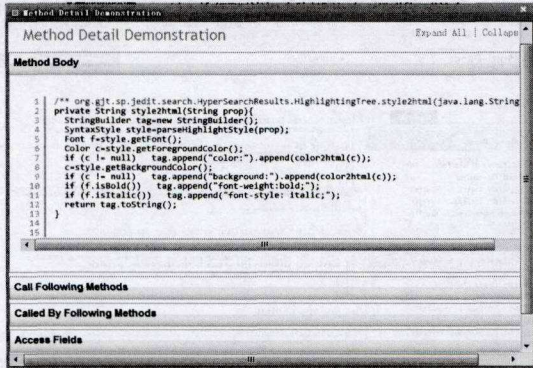


图3 MFIE 的方法详细信息视图

3.3 历史浏览器

为了支持开发人员在特征定位过程中的不同探索步骤之间进行跳转,MFIE 了提供如图 4 所示的历史浏览器。历史浏览器以一种树状结构记录开发人员在当前特征定位过程中的所有探索步骤。其中的每个结点对应一个探索步骤,并记录与该步骤相关的信息,例如所选取的刻面或所使用的关键词以及返回的结果数。树的根结点表示当前特征定位过程的起点。父结点和子结点之间的关系表示特征定位过程中的先后关系,而从属于统一父结点的并列结点则表示多个不同的探索方向。

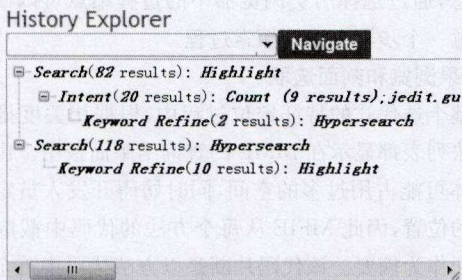


图4 MFIE 的历史浏览器

开发人员通过历史浏览器可以了解整个迭代式的特征查询精化过程。此外,开发人员如果发现当前的探索过程已经偏离了预想的方向(例如发现所期望的结果已经不在当前候选结果之中),那么可以通过点击历史浏览器中的某一个步骤结点回退到此前的一个步骤上。回退后,MFIE 将恢复此步骤上的查询上下文信息(例如候选结果列表),从而使得开发人员可以尝试其他策略(例如选取不同的刻面分类、使用不同

的关键词)。

4 使用过程示例

本章通过一个示例展示基于 MFIE 的特征定位过程。该示例针对的是一个开源的文本编辑器软件 JEdit。在此示例中,开发人员希望借助于 MFIE 找到与 JEdit 中高亮显示功能相关的 Java 方法。

步骤 1 开发人员打开 MFIE,初始页面如图 5 所示。在此页面上的项目下拉框中选择“JEdit”,输入关键字“highlight”,然后点击“Search”按钮。

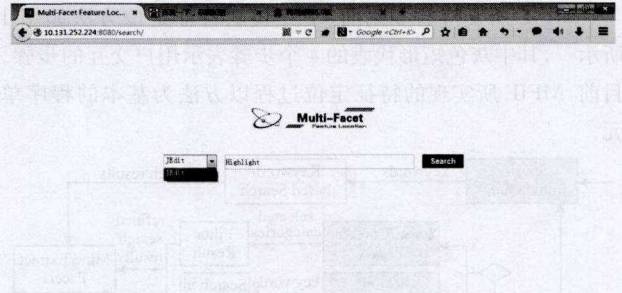


图5 MFIE 使用示例-1

步骤 2 MFIE 进入主界面,根据初始关键字进行查询后得到 82 个候选结果(如图 6 所示)。

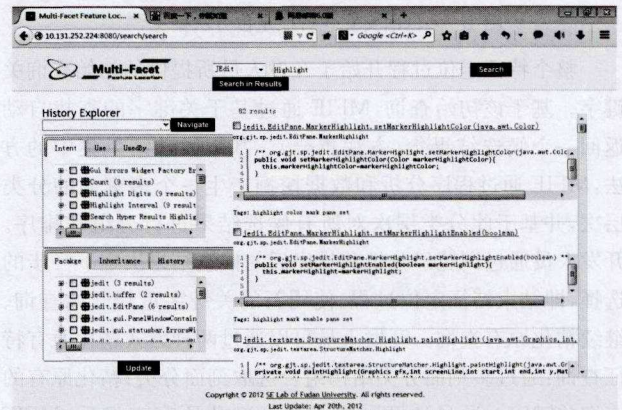


图6 MFIE 使用示例-2

步骤 3 通过 MFIE 结果面板中的候选方法列表进行查看,对于可能相关的方法还可以点击方法名查看详细信息(如图 7 所示)。

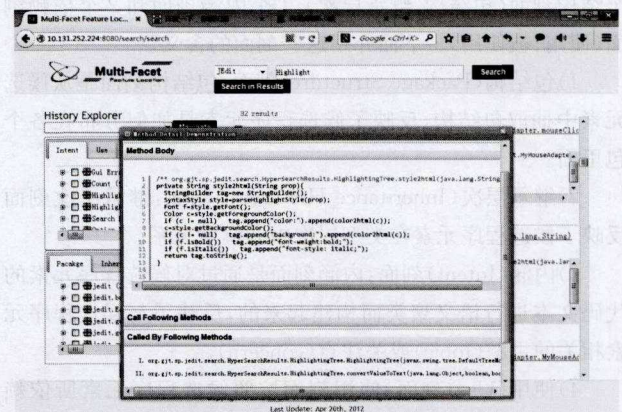


图7 MFIE 使用示例-3

步骤 4 由于高亮显示功能应该会使用与 Color 和 Tex-

area 相关的功能,因此通过 MFIE 刻画面板中的 Use 刻画选取了与此相关的两个分类,点击“Update”按钮进行更新后得到精化后的 18 个候选结果(如图 8 所示)。

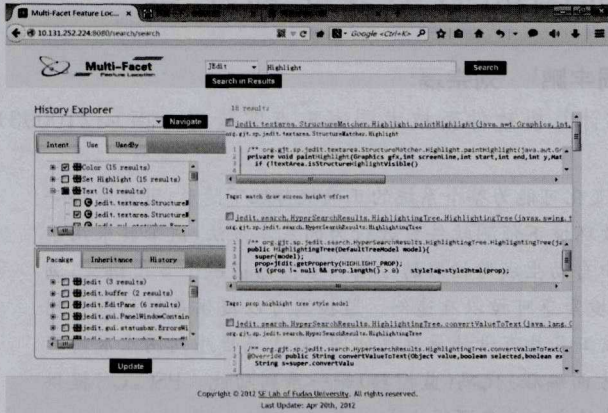


图 8 MFIE 使用示例-4

步骤 5 浏览 MFIE 结果面板,发现候选结果与预期相差较大,因此使用历史浏览器,回退到上一步,即使用 Use 刻画之前(如图 9 所示)。

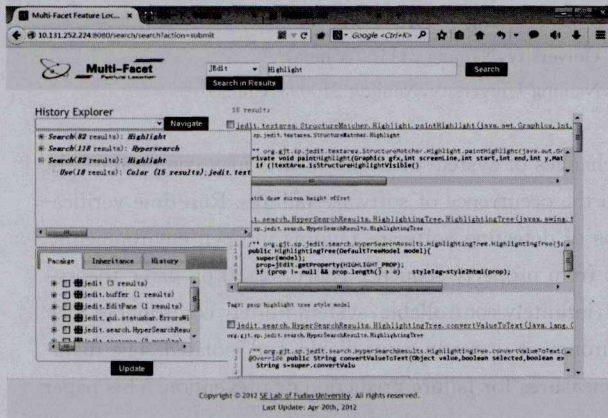


图 9 MFIE 使用示例-5

步骤 6 用户重新进行刻画精化,这次使用的是 Intent 刻画,选择了标签为“Highlight Digits”和“Highlight Interval”的两个分类,点击“Update”按钮进行更新后得到精化后的 18 个候选结果。

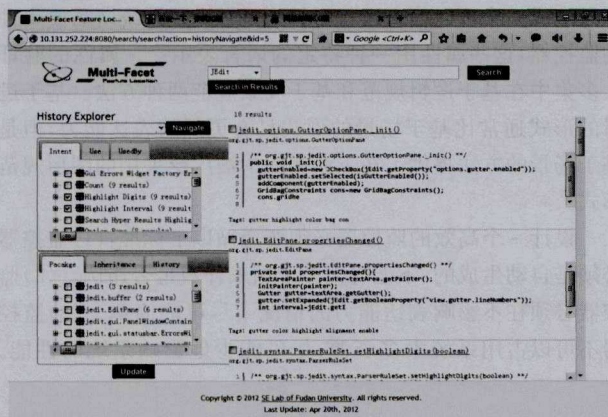


图 10 MFIE 使用示例-6

步骤 7 浏览 MFIE 结果面板,确认所期望的与高亮显示相关的实现方法都在此列表中,记录结果后当前特征定位过程结束。

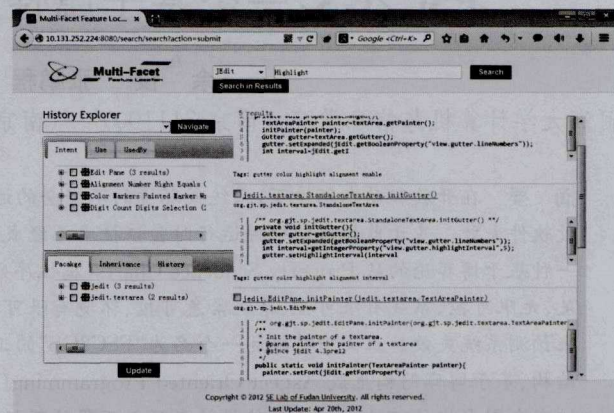


图 11 MFIE 使用示例-7

结束语 MFIE 是一个基于 Web 的代码浏览和特征定位工具,支持多刻画、交互式的特征定位过程。基于 MFIE 的特征定位过程具有以下几个方面的特点:开发人员不需要提供非常准确的初始查询,而只需要提供粗略的初始关键字,支持增量、迭代的查询精化;通过自动抽取和挖掘多刻画分类结构以及代码摘要和标签等手段,为开发人员提供关于候选结果的概貌,帮助其快速、准确地判断候选结果的相关性;开发人员可以通过多刻面的分类结构选取相关的分类,从而精化原有的特征查询;通过历史浏览器提供关于特征定位过程的完整记录,支持开发人员进行回退和重试。

参考文献

- [1] Dit B, Revelle M, Gethers M, et al. Feature location in source code: A taxonomy and survey[J]. J. Softw. : Evol. and Process, 2013, 25(1): 53-95
- [2] Wang J, Peng X, Xing Z, et al. An exploratory study of feature location process: Distinct phases, recurring patterns, and elementary actions[C]//ICSM. 2011: 213-222
- [3] Zhao W, Zhang L, Liu Y, et al. SNI AFL: Towards a static non-interactive approach to feature location[J]. ACM Trans. Softw. Eng. Methodol. , 2006, 15(2): 195-226
- [4] Poshvanyk D, Gu'eh'eneuc Y G, Marcus A, et al. Feature location using probabilistic ranking of methods based on execution scenarios and information retrieval[J]. IEEE Trans. Softw. Eng. , 2007, 33(6): 420-432
- [5] Hearst M A. Clustering versus faceted categories for information exploration[J]. Commun. ACM, 2006, 49(4): 59-61
- [6] Wang J, Peng X, Xing Z, et al. Improving feature location practice with multi-faceted interactive exploration[C]//ICSE. 2013: 762-771