**面向实时协同Web前端开发环境的代码重构机制与工具**

摘 要

实时协同编程环境是解决多个程序员协同开发同一项目的方案之一，相比现在主流的版本控制方案，实时协同环境的操作传送与集成都是由底层系统自动完成，对程序员更加友好。但是，实时协作在提供一系列好处的同时，也带来一个严重问题：每当一位程序员对源代码的某个代码域（code region）作了符合语法规则的更改但由于其他部分的源代码对该代码域有依赖而产生语法错误时，其他程序员无法开展正常的编译与调试工作，给编程工作的质量与效率带来负面影响。

为解决上述问题，本项毕业设计考虑在实时协同环境下、以web前端开发环境为运用场景，分析主要包括代码调用重构和文件引用重构两个模块的代码重构机制，以保证每当程序员对源代码的某个代码域作了符合语法规则的更改时，协同工作系统自动地对那些依赖该代码域的源代码执行关联更新。最后在此重构机制基础上以前端开发编辑器atom插件的形式实现这个重构工具。

**关键词：**代码重构，web前端，实时协同，atom

**Code Refactor Mechanism and Tools for**

**Real - time Collaborative Web Front - end Development Environment**

**ABSTRACT**

Real-time collaborative programming environment is to solve a number of programmers to develop the same project together. It is more friendly compared to git because the operation of the transmission and integration are automatically completed by the underlying system. However, there is also a serious problem when real-time collaboration provides a series of benefits: Whenever a programmer made changes to the source code in accordance with grammatical rules, but there are some syntax errors in other parts of the source code because of dependence, so bring a negative impact to the quality and efficiency of programming work.

To solve the above problems, this project is considered in the real-time collaborative environment and using the web front-end development environment as scene, then analyzes the code refactoring mechanism of two modules, including the code call refactoring and the file refactoring, to ensure that the collaborative system automatically performs an association update for source code whenever a programmer makes a change, finally implement this refactoring tool using atom plugin.

**Key words:** code refactor, web front-end, Real-time collaborative, atom

目 录

[1 引 言 1](#_Toc484354947)

[1.1 项目环境 1](#_Toc484354948)

[1.2 项目目标与用途 1](#_Toc484354949)

[1.3 论文组织结构 2](#_Toc484354950)

[2 项目背景及选型 3](#_Toc484354951)

[2.1 实时协同编程环境 3](#_Toc484354952)

[2.2 web前端开发环境 3](#_Toc484354953)

[2.3 代码重构 4](#_Toc484354954)

[2.4 项目选项 4](#_Toc484354955)

[2.5 atom插件开发 5](#_Toc484354956)

[2.5.1 atom开发语言简介 5](#_Toc484354957)

[2.5.2 atom插件安装与使用 5](#_Toc484354958)

[2.5.3 atom插件开发框架 5](#_Toc484354959)

[3 实验部分 8](#_Toc484354960)

[3.1 仪器和试剂 8](#_Toc484354961)

[3.1.1 仪器 8](#_Toc484354962)

[3.1.2 试剂 8](#_Toc484354963)

[3.2 溶液的配制及标定 8](#_Toc484354964)

[3.2.1 NaOH标准溶液的配制及标定 8](#_Toc484354965)

[8](#_Toc484354966)

[3.2.2 氯化钾离子强度调节剂的配制 8](#_Toc484354967)

[3.3 实验步骤 8](#_Toc484354968)

[4 结果和讨论 10](#_Toc484354969)

[4.1 多元酸体系的结果和讨论 10](#_Toc484354970)

[4.1.1 直接计算法 10](#_Toc484354971)

[4.1.2 半整数法 10](#_Toc484354972)

[4.1.3 分段拟合法 11](#_Toc484354973)

[4.2 氨基酸合铜体系的结果和讨论 11](#_Toc484354974)

[4.3 关于计算方法的讨论 11](#_Toc484354975)

[4.4 关于其他问题的讨论 11](#_Toc484354976)

[5 结论和展望 12](#_Toc484354977)

[5.1 结论 12](#_Toc484354978)

[5.2 展望 12](#_Toc484354979)

[12](#_Toc484354980)

[参考文献 13](#_Toc484354981)

[谢 辞 14](#_Toc484354982)

# 1 引 言

## 1.1 项目环境

协同编程是软件开发的精髓，随着业界对大规模复杂软件系统的需求持续增长,协作编程在软件开发中也处于越来越重要的地位。统计数据显示，软件工程师70%的时间都是用于软件项目的交流与协作，因此团队开发成员间如何更好地协同工作以确保软件开发的质量显得至关重要，团队间良好的协作能大幅提高软件开发的质量和效率。

目前业界主流的软件开发协作方式主要包括非实时协同编程环境和实时协同编程环境两种。

非实时协同编程环境是一种松耦合的协同，允许一组程序员访问共享的程序设计元素集合(源代码目录及文件)、独立完成程序设计任务,并于特定的时刻合并、同步他们对代码的修改，是一种“复制-修改-合并”的模型。实现这种协同工作的是一系列多样的、复杂的版本控制系统,如 SVN、 Git等，目前已被业界广泛采用。

实时协同编程环境是协同编程的另外一个方案，相比上述以版本控制为代表的非实时方案，实时协同环境更加紧密耦合，它的操作传送与集成都是由底层系统自动完成，无需程序员手动执行相关操作，也不会发生版本控制方案里多个程序员在一次提交中同时编辑了同一文件而导致需要手动解决冲突的额外开销。这种新的方法与技术使程序员只关注开发相关事项不用关注代码合并等繁琐事项，能显著提高软件项目的生产效率与质量。作为一种新兴的协同程序设计范例,实时协同编程环境有着广泛的应用场景,包括:1)某个编程模块工作量太大,以至于单个人无法按时完成,但由于该模块的相互依赖,导致其无法再继续被分解为更小的模块。在这种情况下,需要多名程序员以紧密耦合 的方式工作于同一模块。2)在敏捷软件开发过程中,结对编程是一种可以协助程序员产出更高质量的代码。实时协同环境自然支持结对编程,两位在不同地点的程序员展开虚拟结对编程。3)在教育领域中,实时协同程序可以在一些程序设计课程的实践环节中起到作用,以便学生与讲师一起查看与撰写源代码。

实时协同编程是一种新的方法与技术，具备一系列益处，但是同时也带来一个严重问题：每当一位程序员对源代码的某个代码域（code region）作了符合语法规则的更改但由于其他部分的源代码对该代码域有依赖而产生语法错误时，其他程序员无法开展正常的编译与调试工作，只能持续地编辑源代码，无法尽早通过编译与调试发现问题，背离了程序员应有的工作方式，给编程工作的质量与效率带来负面影响。因此本项毕业设计的主要研究内容是：在实时协同编程环境下、以web前端开发环境为运用场景，分析主要包括代码调用重构和文件引用重构两个模块的代码重构机制，以保证每当程序员对源代码的某个代码域作了符合语法规则的更改时，协同编程环境自动且实时地对那些依赖该代码域的源代码执行关联更新，不影响团队其他程序员正常的编译与调试工作。最后在此重构机制基础上以atom插件的形式实现这个重构工具。

## 1.2 项目目标与用途

Web 2.0时代后，用户对页面的美观与交互易用、开发者对前端项目的架构理解与模式设计都发生了翻天覆地的变化，前端技术与新框架层出不穷。在这种技术快速迭代的趋势下，前端开发受到了越来越多开发者的青睐，然而技术的日新月异也对保持良好的代码规范提出了巨大挑战，为了后续开发和维护的可持续性，很多大型项目不得不选择经常进行重构以适应业界的最佳实践。

同时，业界推出了如atom、subline等很多优秀的前端开发IDE，相比后端开发IDE的“笨重”，前端IDE响应迅速更适应于对网络延时要求高的实时协同编程环境。

鉴于前端IDE对实时协同编程环境更好的支持性以及前端开发对重构的巨大需求，本项毕业设计提供的实时协同编程环境下的这个重构工具可以显著提升前端项目的开发效率与质量。以一个多人小组协同开发以angularJS为基础框架的前端项目为例：

项目组所有成员通过实时协同编程环境，能以紧耦合的方式一起开发该前端项目，在实时协同环境下代码的提交、合并等操作传送与集成都是由底层系统自动完成，因此每个成员可以更专注于自己模块的业务逻辑代码编写，而不用花费大量时间用于项目代码的更新与冲突解决。更重要的是，当angularJS框架发布新版本或者项目结构越来越复杂，项目的架构师a不得不对项目进行重构的时候，使用本项毕业设计开发的重构工具，协同编程环境就会自动地对那些依赖该代码域的源代码执行关联更新，并实时地将改动同步到其它项目组成员的工作空间，使团队其他程序员的编译与调试工作能正常进行，保证了前端项目开发的效率与质量。

## 1.3 论文组织结构

本论文共有六章：

第一章：

# 2 项目背景及选型

## 2.1 实时协同编程环境

实时协同编程环境是一种新兴的协同编程解决方案,这种方案可以帮助程序员实现细粒度的、 即时的交互,以及对并发修改的合并与同步。这种编程环境允许一组程序员以紧密耦合的方式并行编辑同一组源代码文件与目录，且多名程序员对代码的修改可以被即时传送与合并；而且，这些实时的操作传送与集成是由底层系统自动完成的，无需程序员手动执行相关操作。在实时协同编程过程中，多名程序员可以在同一时刻并行访问与编辑共享源代码副本的任意部分，甚至并行编辑同一个源代码文件的内容。如图1所示,实时协同编程环境下,多名程序员可以在同一时刻并行访问与编辑共享源代码副本的任意部分,甚至并行编辑同一个源代码文件的内容。

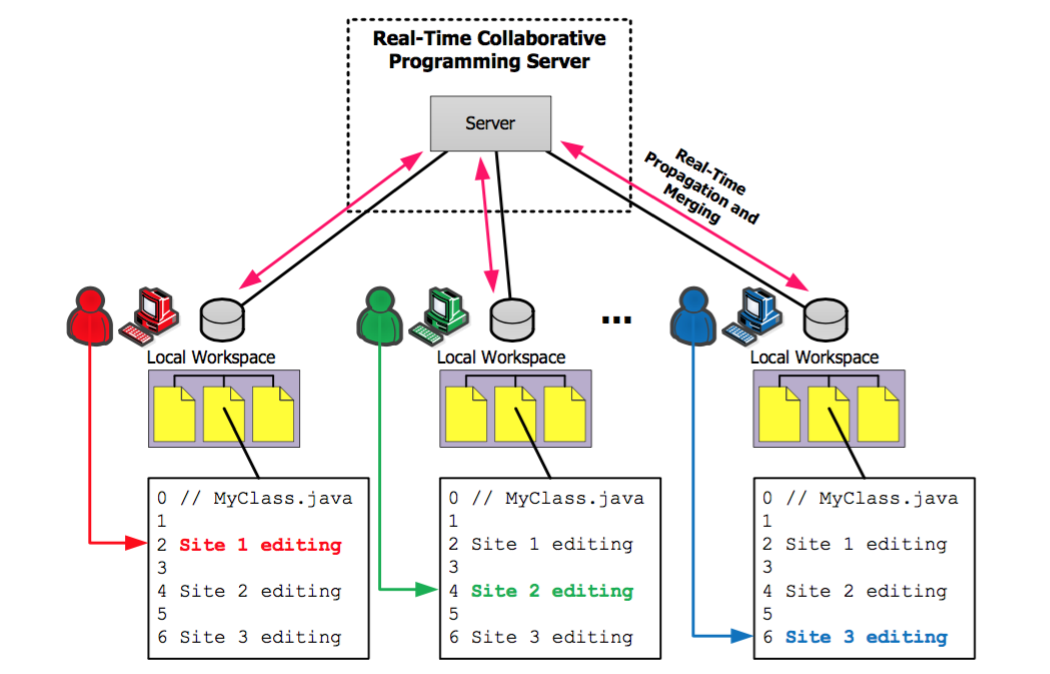


图1 实时协同编程环境下编辑同一个文件

## 2.2 web前端开发环境

web（World Wide Web）即全球广域网也称万维网，它是一种基于超文本和HTTP的、全球性的、动态交互的、跨平台的分布式图形信息系统[1]。随着移动终端的普及与web2.0的强大特性，web应用因其跨平台的特性越来越受到欢迎，不论是windows系统还是Mac系统、不论是dell个人电脑还是iPhone同一个web应用通过浏览器都可以完美运行，这极大减轻了开发者的负担。

Web开发从视图与数据的角度来看也分为前端和后端，前端负责视图的呈现和用户交互，后端负责业务逻辑处理和与数据库交互。随着前端项目的复杂化以及用户终端硬件的变革，开发者对于前端开发设计模式的思考也在经历着快速变革，前端代码的模块化、框架化、按需加载、依赖管理等等都在规范化，前端开发的功能与效率都在不断提升，因此Web前端越来越受到开发者的喜爱和重视。

Web前端项目开发的日益复杂化迫使人们不断提出架构和设计模式的改变，从传统mvc架构到angularJS框架为代表的mvvc、React框架为代表的Virtual DOM等等，而这些框架的出现一方面促进了前端开发的发展提升了前端开发的效率，另一方面也使前端工程像后端工程一样走向了模块化、工程化。相应地，前端项目对于IDE的需求也在逐渐变得更高，这种发展趋势让前端项目也有了代码重构的需求。

## 2.3 代码重构

为了保证软件系统的可维护性以及后续开发的可扩展，代码的可读性与优雅性一直是软件开发中强调的标准和目标。编程不仅仅是编写代码，一次就能写出完美的代码是几乎不太可能的，随着时间的推移与项目复杂度的增加，很多情况下开发者必须重构已编写的代码来修复错误和更改功能。而像代码重构这样大的改动，我们不可能也没必要手动去一一更改涉及到的代码，因此一个好用的代码重构工具对于提升软件开发的效率必不可少。

代码重构机制的核心是自动关联更新（automatic linked update）：每当程序员对源代码的某个代码域作了符合语法规则的更改，项目工程中其它依赖该代码域的源代码自动执行关联更新，避免发生代码依赖错误。

在web前端开发中，代码重构主要有两类运用场景：一是当某位程序员改动了其正在编辑的方法的名称，源代码中（无论是同一源代码文件还是其他源代码文件）所有依赖该方法的代码元素（例如调用该方法的语句）都被自动更新，以保证该项更改不会导致源代码产生语法错误。二是当某位程序员需要重构项目结构而对某个文件路径进行改变时(更改文件名或更改文件位置)，整个项目所有源代码中对该文件的引用都被自动更新，以保证该文件名字或者位置的更改不会导致源代码产生引用错误。

## 2.4 项目选项

完成实时协同Web前端开发环境场景下的源代码智能重构机制的分析与设计后，以此为基础，本项毕业设计将以代码编辑器插件的具体形式实现一个软件工具，以此帮助前端开发者更好地在实时协同环境下进行代码重构。

我们选择了4种现在前端开发主流的IDE：Sublime Text[2]、Visual Studio Code[3]、Adobe Brackets[4]和Atom[5]作为该软件工具候选的代码编辑器，主要从编辑器是否开源、提供的API与文档完善度、开源社区活跃性、编辑器性能等几个方面分析比较如表1：

表1 目前主流前端编辑器比较



Sublime Text优点是性能很好运行速度快，但是不开源且收费，对插件开发者不友好，不利用本项毕业设计。VSCode是微软基于atom改进的编辑器，其设计理念是提供用户一个性能强大、功能完备、质量保证的IDE，而不是鼓励用户自己编写插件扩展所需功能，官方很晚才提供插件开发API，文档与社区支持性都一般，所以VSCode从单纯用户使用角度是很有竞争力的，但是定位与本毕业设计的插件开发也不特别相符。Adobe Brackets 是Adobe公司推出的一款开源编辑器，对Adobe系列软件支持很好很适合偏设计的前端开发人员使用，如可以用Brackets 直接打开一个Psd文件，然后进行颜色和宽度的选择等。Brackets本身提供了丰富的API和详细的文档供插件开发者使用，但是社区活跃性一般不如atom，因此最终本项毕业设计选择Github公司开源的编辑器atom作为插件开发平台。

Atom 原先是Github公司内部使用的编辑器，现在全部开源且鼓励用户按需使用或编写自己所需插件以获得更适合自己的功能。Atom被官方自称为“为 21 世纪创造的可配置的编辑器”，其最显著也是最重要的特点就是插件化架构，对 Atom 而言最重要的是灵活而又完备的插件 API，多次官方新版本发布都在不断改进API与相应文档的支持。atom开发维护团队不仅提供了强大易用的API与使用文档，同时因为与Github深度整合让开源的力量发展迅速，开源社区成长速度极快且十分活跃，包括中文社区都是同类编辑器中最活跃的。社区驱动的方式让atom搭乘上了开源的快车，用户不仅可以共享每年数以万计发布的各类功能的插件，而且还可以按自己喜好改进甚至很方便地开发自己需要的插件。综上，基于atom对Web平台开发者友好对前端更友好，Hackable的显著特色让任何人都能简单贡献自己的插件，最终选择在atom平台下完成实时协同Web前端环境下的源代码重构插件开发。

## 2.5 atom插件开发

### 2.5.1 atom开发语言简介

Atom 文本编辑器是基于 WEB 技术（Chromium+Node.js）开发的，可以理解为编辑器本身就是一个跑在本地的网页。因此我们将完全使用 JavaScript 的 EcmaScript 6 规范[6]来制作插件，至于插件实现功能所需要的样式(如弹出框尺寸、光标选中某代码域后改变该代码域颜色等)可用css或less实现。

### 2.5.2 atom插件安装与使用

在开发插件之前，我们首先得明白atom上插件是如何被安装关联到编辑器。下载好atom后，我们还需要同时安装上apm(atom package manager)才能对atom插件包进行管理。

然后在自己开发或者其他开发者开发的插件代码工程目录下使用apm link命令，即可将此插件link到atom的插件库中(~/.atom/packages)，从而将插件关联到编辑器，实现atom的功能扩展。

安装成功后，在atom上打开manage packages的setting页，即可看到installed packages的数量相应增加，同时在community packages目录下可以看到安装的非atom自带插件的详细信息与插件管理设置功能。

最后启动插件，atom提供4种方式用于启动插件：一是快捷键启动，二是atom控制面板输入相应启动命令启动，三是通过导航栏菜单选择相应插件启动，四是通过右击菜单选择相应插件启动。也可在插件管理页面将插件设置为随编辑器自启动，但是要注意不同插件之间启动命令不能相互冲突

### 2.5.3 atom插件开发框架

atom是如此地鼓励社区进行插件开发，因此对于插件开发者不仅提供完备易用的API与文档，甚至于已经把插件开发框架集成到atom自身，所以我们可以很方便地使用 Atom 提供的工具创建一个新的 package（软件包）来开始插件的开发[7]。

启动atom编辑器后，打开控制面板(command+shift+p)，输入“Package Generator: Generate Package”并点击列表中正确的条目，然后在输入提示框中输入该插件软件包的名称，即可生成插件开发的框架代码。另外，在atom插件开发中需要注意的一点是，每当需要测试我们对软件包的改动的时候，都要重新加载当前窗口以确保 Atom 执行的是我们最新的源代码。atom强大的控制面板允许我们很方便地重载编辑器，在控制面板输入window:reload即可完成。

通过atom自带的package generate生成的插件工程基本结构如图2所示：

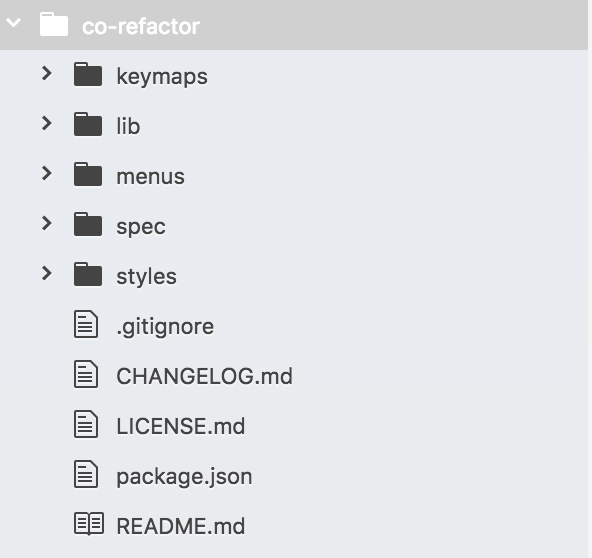


图2 atom插件骨架代码结构

工程主要包括5个目录和package.json文件，用途如下：

1. package.json文件

用于定义主文件位置，主文件是 Atom 软件包的入口文件，Atom 通过设置 package.json 里的main属性来找到主文件的位置，如图3所示：

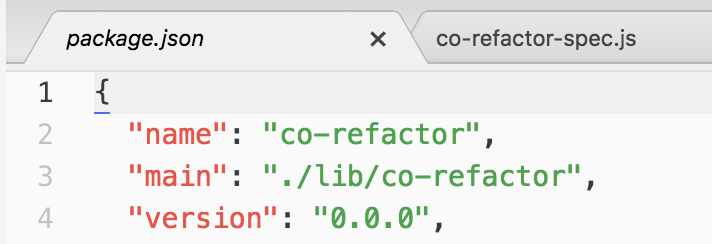


图3 package.json文件

1. keymaps文件夹

用于配置插件快捷键，在keymaps文件夹下自动生成了一个json配置文件co-refactor.json，然后在该json文件指定快捷键与对应指令即可，如图4以下配置可以让用户通过 Ctrl+Alt+O（Windows/Linux） 或 Cmd+Alt+O（MacOS） 来触发指令"co-refactor:toggle"：

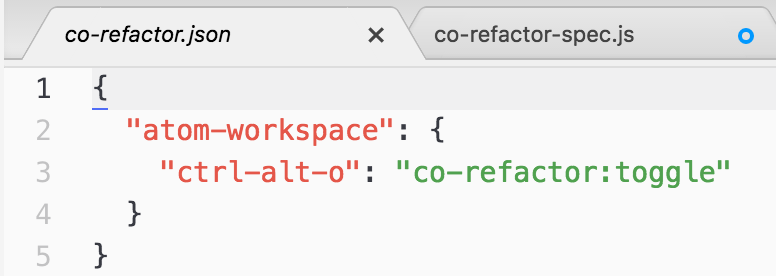


图4 co-refactor.json文件

1. menus文件夹

用于配置右击菜单和导航栏菜单，在menus文件夹下自动生成了一个json配置文件co-refactor.json，其中主要有两项配置内容：一是“menu”  对象用来定义插件的自定义应用菜单，通过 label 属性和submenu属性定义该自定义应用菜单层级，最后通过label 属性和command属性定义点击该菜单后执行的命令，如图5所示：

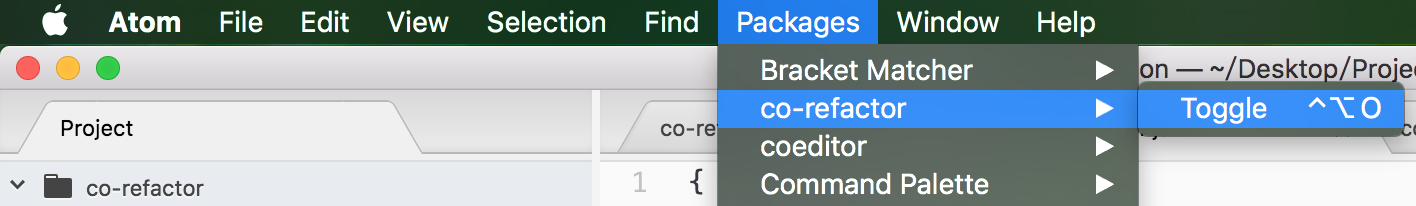


图5 atom导航栏菜单自定义

另外一个是“context-menu” 对象可以让我们定义右击菜单的一些新条目，每一个条目都是通过一个显示在菜单的 label 属性和一个点击后执行的命令的 command 属性来定义的，如图6所示定义右击菜单的“Toggle co-refactor”条目对应指令"co-refactor:toggle"：

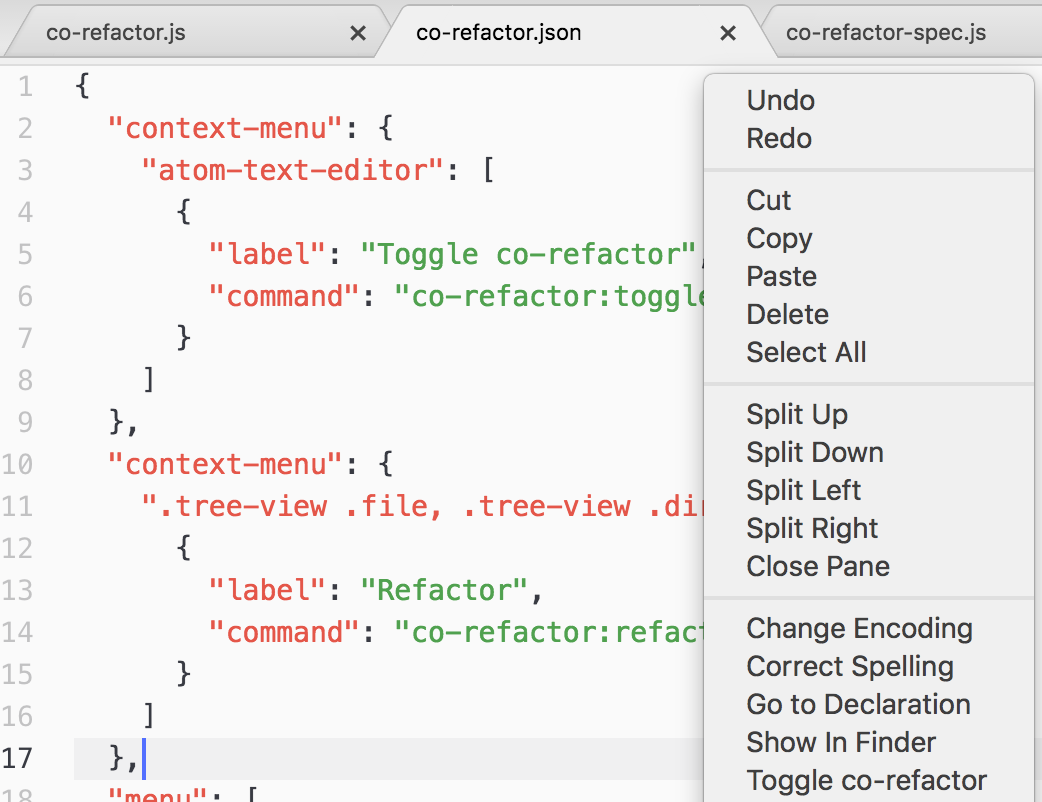


图6 atom右击菜单自定义

1. libs文件夹

主要包括插件主文件和处理业务逻辑的js文件，一般而言在项目package.json文件中通过main属性定义的插件主文件都在该文件夹下，如该generate框架项目就是在package.json文件中通过 "main": "./lib/co-refactor" 定义该插件主文件为lib文件夹下的co-refactor.js。

主文件co-refactor.js导出一个带有生命周期函数（Atom 在特定的事件发生时调用的处理函数）的对象，主要包括4个模块：

1. activate模块

在 Atom 初次加载软件包的时候调用，用来初始化一些诸如软件包所需的用户界面元素的对象，以及订阅软件包命令的处理函数。如图7定义'co-refactor:toggle'命令对应'toggle'函数：



图7 atom定义指令对应函数

1. deactivate模块

在软件包停用的时候调用，例如，当用户关闭或者刷新编辑器的时候。

1. serialize模块

Atom 会在使用软件包的过程中不断调用它以保存软件包的当前状态，然后将返回值在 Atom 下一次加载软件包的时候作为一个参数传递给上述activate模块。

1. toggle模块

为activate模块中定义的'co-refactor:toggle'指令对应函数，也就是插件启动入口函数，后续插件功能扩展由此开始。

# 3 项目需求分析与架构设计

## 3.1 功能需求分析

为了使用户更高效便捷地实现互相协助下的代码重构,我们首先从用户的角度展开系统的功能需求分析。在前端开发场景，最常见的重构主要有两类：代码调用重构和文件引用重构，因此我们的重构系统主要针对这两类场景提供强大的重构功能和友好的使用方法：

1. 代码调用重构:

该功能主要有两方面的需求：函数调用重构和变量重构，即用户选择某个函数或变量进行重构时，该文件中所有对这个函数或变量的调用都会自动关联更新。考虑用户使用的便捷性，用户在选择代码重构时，还应该可视化地告知用户这次重构操作会影响多少处代码，受影响的代码的位置等详细信息。

1. 文件引用重构:

该功能主要也包括两方面的需求：重构单个文件和重构一个文件夹，即用户可以在工程树状导航目录上选择一个文件或一个文件夹，重构这个文件或文件夹的命名、位置，则此工程项目所有引用了该文件的代码都会相应被关联更新。为了对用户使用更加友好，重构完成后也应该可视化地告知用户这次重构操作影响了多少文件。

重构功能完成后，我们再集成实时协同编程环境，将用户对工程的重构操作实时同步到其他参与协同编程的用户终端，实现实时协同编程环境下的代码重构。

## 3.2 用户界面需求分析

### 3.2.1 代码调用重构界面

考虑用户更改代码中函数名或变量名时的使用场景，该功能设计为用户光标选中重构对象，快捷键启动代码调用重构功能后，生成一个重构构件用于接收和显示重构信息。如图8该构件包括最基本的两个label显示重构对象当前命名和目标命名，为了增强这个工具的可视化，我们还增加一个list控件实时显示当前重构对象及其上下文:

### 3.2.2 文件引用重构界面

……。

## 3.3 项目架构设计

……。

# 4 结果和讨论

## 4.1 多元酸体系的结果和讨论

### 4.1.1 直接计算法

三级以下单独占行标题采用A.、B.、C.和a.、b.、c.…或（1）、（2）、（3）和①、②、③…。黑体五号。标题序号缩进2个汉字符。序号和标题之间空一格。

五号，黑体（英文Times New Roman），行距18磅，段前0行，段后0行。

……。

### 4.1.2 半整数法

A. 半整数法的求解过程

a. 可以直接得到半整数的情况

从表4.1可见，当为0.5和1.5时的pH分别是5.32和2.74，即……。

b. 无法直接得到半整数的情况

在用半整数生成函数法直接求解时，会遇到这样的问题……。表4.4列出了滴定过程中乙二酸溶液的各主要物理量的部分数据。从表4.4可见，……。

图表可横版

表序写在表题左方不加标点，空一格写表题，表题末尾不加标点，表格逐章编序，表序必须连续，如表4.4表示第四章的第四个表。表题：小五，黑体（英文Times New Roman），居中置于表上方，行距18磅，段前0行，段后0行。

……

（1）表格上下与正文之间各空一行；（2）采用三线表，两端与页面对齐；

（3）表中文字：小五，宋体 （英文Times New Roman），行距18磅，段前0行，段后0行。

……。

表4.4 草酸的部分数据列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *V*/mL | *E*/mV | pH | [H+]/（mol·L-1） |  |
| …… | …… | …… | …… | …… |
|  |  |  |  |  |
| 1.00 | 2.62×102 | 2.22 | 6.07×10-3 | 1.14 |
| 1.50 | 2.60×102 | 2.27 | 5.42×10-3 | 1.10 |
|  |  |  |  |  |
| …… | …… | …… | …… | …… |

续表4.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *V*/mL | *E*/mV | pH | [H+]/（mol·L-1） |  |
| …… | …… | …… | …… | …… |
| …… | …… | …… | …… | …… |
| …… | …… | …… | …… | …… |

（1）若表格分页，则该表第2页的表题省略，但表头（即“*V*/mL┄”所在行）应重写，并在表右上方加注“续表X.X”；

（2）“续表X.X”的格式：小五，黑体（英文Times New Roman），行距18磅，段前0行，段后0行，右空2格。



（1）图居中，上下与正文之间各空一行；

（2）图中文字：小五，宋体（英文Times New Roman），行距1倍，段前0行，段后0行。

图4.1 乙二酸 ≥1.15数据段曲线及其拟合曲线（实线--实际曲线，虚线--拟合曲线）

针对这种情况，可以采用多项式拟合的方法求解。以乙二酸为例，在Excel中，选取1.28<<1.15之间的数据，以为横坐标，pH为纵坐标，做-pH曲线（见图4.1），并添加趋势线，选择相关系数*R*2最接近1的多项式作为拟合方程，……。

B. 半整数法的计算结果

利用半整数生成函数法，对各种多元酸三次平行实验数据分别进行处理，并求平均值，结果见……。

（1）插图应有图序和图题，全文插图以章分组编序号，图序必须连续，不得重复或跳缺。如图4.1表示第四章的第一幅图。

（2）图题：小五，黑体（英文Times New Roman），居中置于图下方，行距18磅，段前0行，段后0行。

C. 半整数法计算结果的讨论

……。

### 4.1.3 分段拟合法

……。

## 4.2 氨基酸合铜体系的结果和讨论

……。

## 4.3 关于计算方法的讨论

## 4.4 关于其他问题的讨论

# 5 结论和展望

段落号：（1）…。

（2）…。

（3）…。

## 5.1 结论

（1）生成函数法可以分为直接计算生成函数法、分段拟合生成函数法及半整数生成函数法。这三种方法有如下特点：①……；②……；③……。

（2）本文运用三种不同生成函数法，测定了多元酸和氨基酸合铜配合物的稳定常数，得到了……。

（3）三种生成函数法中无论哪一种方法，对待测酸或配合物稳定常数的大小均有一定的要求，如……。

……。

## 5.2 展望

（1）生成函数法理论可靠，计算方便，但……。

（2）在生成函数法的应用中，还有以下问题有待研究和解决：①……；②……。

……。

段内层次号 ：①…；②…。

# 

注：期刊若只有期，没有卷，则可以省略卷号，如参考文献[1]示例。若只有卷，没有（或不分）期，则可以省略期号，如参考文献[2]示例。

# 参考文献

期刊中析出的文献

1. 李炳穆.理想的图书馆员和信息专家的素质与形象[J].图书情报工作，2000(2):5-8.
2. DES MARAIS D J, STRAUSS H, et al. Carbon isotope evidence for the stepwise oxidation of the Proterozoic environment[J]. Nature, 1992, 359: 605-609.
3. 陈桂娥，樊行雪，许振良.线性滴定中稳定常数测定方法比较[J].华东理工大学学报，1996，22(5): 620-625.

普通图书

1. 蒋有绪，郭泉水，马娟，等.中国森林群落分类及其群落学特征[M].北京：科学出版社，1998.
2. International Federation of Library Association and Institutions. Names of persons: national usages for entry in catalogues[M]. 3rd ed. London: IFLA International Office for UBC, 1977.

论文集，会议录

1. 雷光春.综合湿地管理:综合湿地管理国际研讨会论文集[C].北京:海洋出版社，2012.
2. BABU B V, NAGAR A K, DEEP K, et al. Proceedings of the Second International Conference on Soft Computing for Problem Solving, December 28-30, 2012[C]. New Delhi: Springer, 2014.
3. 孔宪京，邹德高，徐斌，等.台山核电厂海水库护岸抗震分析与安全性评价研究报告[R].大连:大连理工大学工程抗震研究所，2009.

报 告

1. World Health Organization. Factors regulating the immune response: report of WHO Scientific Group[R]. Geneva: W H O, 1970.

学位论文

1. 王燕.氨基酸–金属离子体系的测定[D].上海:同济大学，2009.
2. CALMS R B. Infrared spectroscopic studies on solid oxygen[D]. Berkeley: University of California, 1965.

专利文献

1. 刘加林.多功能一次性压舌板:中国，92214985.2[P]. 1993-04-01.

专著中析出的文献

1. 白书农.植物开花研究[M]//李承森.植物科学进展.北京:高等教育出版社，1998:146-163.
2. 钟文发.非线性规划在可燃毒物配置中的应用[C]//赵玮.运筹学的理论与应用:中国运筹学会第五届大会论文集.西安:西安电子科技大学出版社，1996:468-471.
3. 张田勤.罪犯DNA库与生命伦理学计划[N].大众科技报，2000-11-12(7).

标准文献

1. 全国信息与文献标准化技术委员会.文献著录:第4部分 非书资料: GB/T 3792.4-2009[S].北京:中国标准出版社，2010:3.
2. 萧钮.出版业信息化迈人快车道[EB/OL].(2001-12-19)[2002-04-15].http://www.creader. com/news/20011219/200112190019.html.

电子文献

1. Dublin core metadata element set: version 1.1[EB/OL].(2012-06-14)[2014-06-11].http:// dublincore.org/documents/dces/.

（1）所有引用的期刊需写出完整刊名。按论文中参考文献出现的次序，用阿拉伯数字自然编号，序码加方括号，顶格书写。

（2）五号，宋体（英文Times New Roman），行距18磅，段前0行，段后0行。

（3）参考文献不少于10篇，其中外文文献不少于2篇 （这是最低要求。各学院可以根据本学院情况制定数量要求）。

# 谢 辞

正文内容

出自内心，有感而发。正文：五号，宋体（英文Times New Roman），两端对齐，段落首行左缩进2个汉字符，行距18磅，段前0行，段后0行。