logo

**文件编号：**

**Visual-Vers——可视化版本控制工具**

体系结构设计报告

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文件标识： |  |
| 当前版本： | 1.0 |
| 作 者： | 汤英杰 顾启隽 叶扬 |
| 完成日期： | 2022-5-31 |

浙江工商大学计算机与信息工程学院

目 录

[1文档介绍 2](#_Toc438905773)

[1.1文档目的 2](#_Toc438905774)

[1.2文档范围 2](#_Toc438905775)

[1.3读者对象 3](#_Toc438905776)

[1.4参考文献 3](#_Toc438905777)

[1.5术语与缩写解释 3](#_Toc438905778)

[2系统概述 4](#_Toc438905779)

[3设计约束 4](#_Toc438905780)

[3.1软件、硬件环境的约束 4](#_Toc438905781)

[3.2用户界面约束 4](#_Toc438905782)

[3.3产品质量约束 4](#_Toc438905783)

[3.4隐含约束 5](#_Toc438905784)

[4设计策略 5](#_Toc438905785)

[4.1扩展策略 5](#_Toc438905786)

[4.2复用策略 5](#_Toc438905787)

[4.3折衷策略 5](#_Toc438905788)

[5系统总体结构 5](#_Toc438905789)

[6开发环境的配置 6](#_Toc438905790)

[7运行环境的配置 6](#_Toc438905791)

[8测试环境的配置 7](#_Toc438905792)

[9其他 7](#_Toc438905793)

# 1文档介绍

## 1.1文档目的

从该阶段开发正式进入软件的实际开发阶段，本阶段完成系统的大致设计并明确系统的数据结构与软件结构。在软件设计阶段主要是把一个软件需求转化为软件的表示过程，这种表示只是描绘出软件的总的概貌。本概要设计说明书的目的就是进一步细化软件设计阶段得出的软件总体概貌，把它加工成在程序细节上非常接近于源程序的软件表示。

## 1.2文档范围

本文档包括系统概述、设计约束、设计策略、系统总体结构、子系统的结构和功能、开发环境的配置、运行环境的配置、测试环境的配置。

## 1.3读者对象

本文档主要面向客户、架构设计人员、界面设计人员、编码人员及测试人员。

## 1.4参考文献

《软件工程》，陈明，中央广播电视大学出版社，2002.6

《软件工程导论第六版》， 张海藩，牟永敏，清华大学出版社，2013.8

《QtCreator快速入门》，霍亚飞，北京航空航天大学出版社

《Git从入门到精通》，高见龙，北京大学出版社

## 1.5术语与缩写解释

|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| Git | 一款分布式版本控制工具，目前为大众所使用 |
| VVS | 可视化版本库，Visual-Vers |
| DAG | 有向无环图，Directed Acyclic Graph |
| Diff | 差异，difference |
| Dagre | 一种有向无环图绘制算法 |
| DP | 动态规划，Dynamic Programming |

# 2系统概述

本系统是一款可视化的版本控制系统——Visual-Vers，目的在于补全git在可视化方面的不足，让版本控制工具大众化。它比git更简单，容易上手，更能用图形图像界面展示版本关系，能帮助新人更好的入门版本控制。目前百度上能找到的大多git可视化工具，都和git desktop差不多，只是显示了节点信息或者是做了差异对比，而没有显示各个版本节点、分支之间的继承推进关系。而本项目则完成了这一任务。

本系统具有的主要功能包括：

- 版本提交、记录

- 版本回滚

- 撤销变更

- 分支操作

- 版本合并

- 版本差异对比

- 文件变更对比

- 用户、时间记录

**文件系统**

本项目实际上一个文件系统，其中总的来看有两个树形结构。每个版本都是一棵文件树，文件树的根节点又构成了一棵版本继承树（因为版本合并操作的存在，版本继承树准确来说是一个DAG（有向无环图））

各个版本的文件树之间通过共享节点来节省空间。

**项目变更发现**

这里指在整个工作区中，找到所有用户修改过的文件（相对于上个版本）并罗列呈现，和下面的“文件变更发现”有所区别。

文件树中的每个节点都有一个哈希值，利用此哈希值可以快速判断两个节点是否相同。

git也是依赖哈希判断文件是否相同的，但它只关注文件内容，将文件名放置在树对象（即保存它的文件夹对象）中，而本项目则将文件内容和文件名称放在一起去获取哈希值，这样做的优势在于：git判断两个节点是否相同必须依赖上下文，而本项目具体存储的节点完全是“路径无关”的，可以做到散列存储。

**文件变更发现**

这里指的是对某个具体的文件，找出其修改内容（具体增删了哪些字符）

对应git当中的git diff功能，它的抽象问题是求两个字符串之间的最小编辑次数，将删除和插入视作一次编辑，修改视作删除+插入。

求变更方案的第一原则是编辑次数最小，它可以通过求最长公共子序列完成，复杂度为 O(n^2)，更优的解法是myers算法，同样为动态规划解法，复杂度为O(nd)，d为修改次数。本项目使用myers完成。

参考资料：<https://zhuanlan.zhihu.com/p/67024353>

**有向图绘制**

本项目的图形界面使用Qt制作，有向图绘制尝试了dagre布局算法

<https://www.yuque.com/antv/g6-blog/xxp5nl>

但我还没有完全理解这个算法，项目中目前使用的一个比较简陋的dp：

从根节点开始dfs布局，从底向上依次排版，每个点依据各个子节点需要宽度来布局，然后返回当前点所需要的宽度。

# 3设计约束

## 3.1软件、硬件环境的约束

|  |  |  |
| --- | --- | --- |
| **需求名称** | | **详细要求** |
| 运行平台 | 硬件需求 | 1G以上内存 |
| 操作系统 | Windows |
| 开发平台 | 操作系统 | Win All |
| 编程语言 | C++ |
| 开发框架 | Qt |
| 编译工具 | MinGW |

## 3.2用户界面约束

|  |  |
| --- | --- |
| **需求名称** | **详细要求** |
| 仓库的初始化 | 选择后，可以创建新的vvs仓库 |
| 打开已有的仓库 | 选择后，可以打开一个已有的仓库，进入软件主界面 |
| 打开最近仓库 | 可以快速打开最近打开过的仓库 |
| 撤销更改 | 点击后，可以撤销工作区内显示的所有修改 |
| 刷新工作区 | 点击后，在工作区内显示仓库的最新变动 |
| 提交更改 | 点击后，提交工作区内更改，形成新的节点分支并用图表示继承关系 |
| 对比此节点与工作区 | 点击后，对比选中节点与工作区的差异并显示差异 |
| 从此节点拉取/合并 | 点击后，将选中节点与当前节点进行合并，并形成新的节点。合并冲突后，查看目标文件差异时可直接“应用到工作区”，同时删除冲突记录 |
| 切换工作区到此节点 | 点击后，切换工作区至此节点 |
| 下班 | 点击后，退出主机面，关闭软件 |

## 3.3产品质量约束

|  |  |
| --- | --- |
| **主要质量属性** | **详细要求** |
| 正确性 | 系统必须保证节点间的对比和继承是正确的 |
| 健壮性 | 系统应能够保证无故障运行 |
| 可靠性 | 系统应保证工作区的切换以及节点的创建是可靠的 |
| 性能，效率 | 系统可以保证在短时间内完成节点差异对比，变更节点 |
| 易用性 | 用户界面要是友好的图形化界面，以用户的角度为思考 |
| 清晰性 | 界面清晰 |
| 安全性 | 共同协作时保证项目内容不泄露 |
| 可扩展性 | 避免需求变更大时大规模修改程序 |

## 3.4隐含约束

软件使用者可能对版本控制了解较浅，对他们提供的操作界面应直观易于理解，保证他们能够很快上手。

# 4设计策略

## 4.1扩展策略

本软件需要具备实现分布式仓库的可拓展性，具体体现在以下方面：

1. 本软件在本地管理项目，存储版本节点，并不依赖远程仓库；
2. 本软件在各个设备上存储逻辑相同，具有同一性。
3. 设计上节点的判重是路径无关的，版本并不依赖项目存在
4. 在物理结构上，本项目也按照两个树形结构存储，可以将一个版本的文件完整迁移到另一个项目。
5. 物理结构上，任一子目录的历史节点都可以拿出来作为一个独立的版本，容易实现不同仓库间的节点合并。

## 4.2复用策略

为了在以后的软件开发中可以复用本软件，软件要求要模块化，且模块独立性高，模块间接口简单，模块功能单一化、参数化程度高、可配置性好。

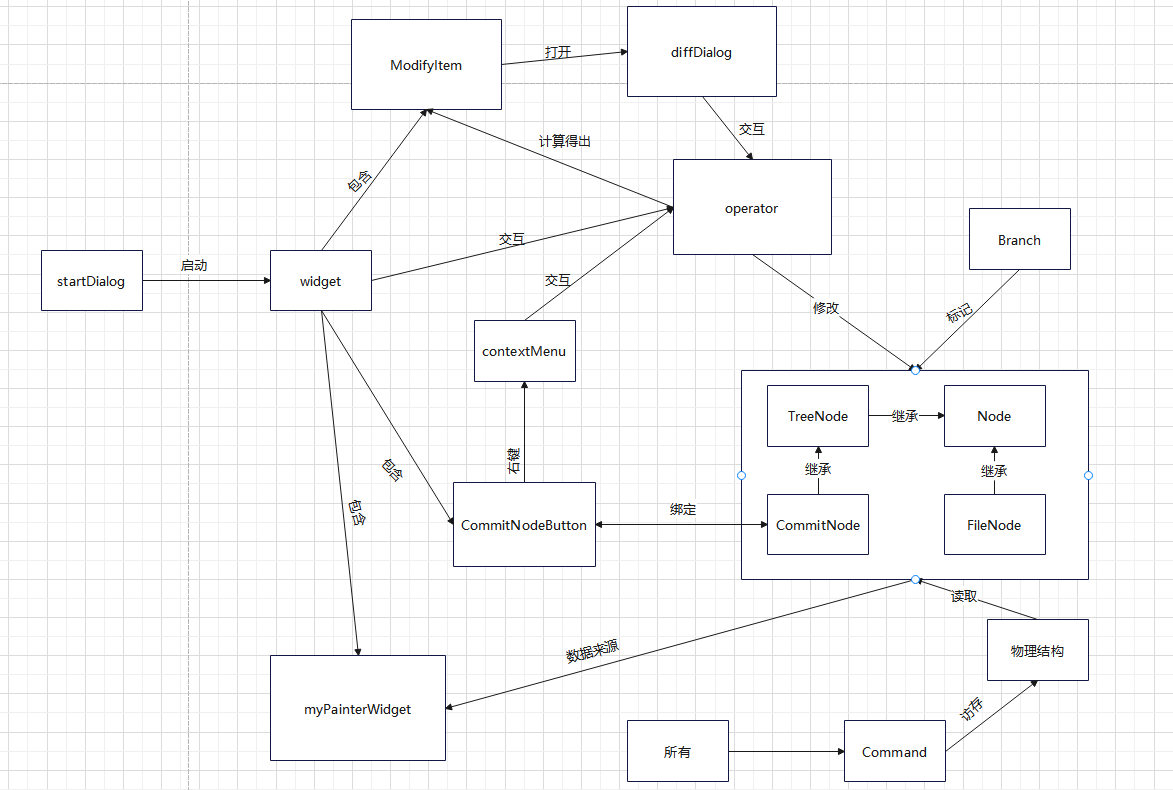
## 4.3折衷策略

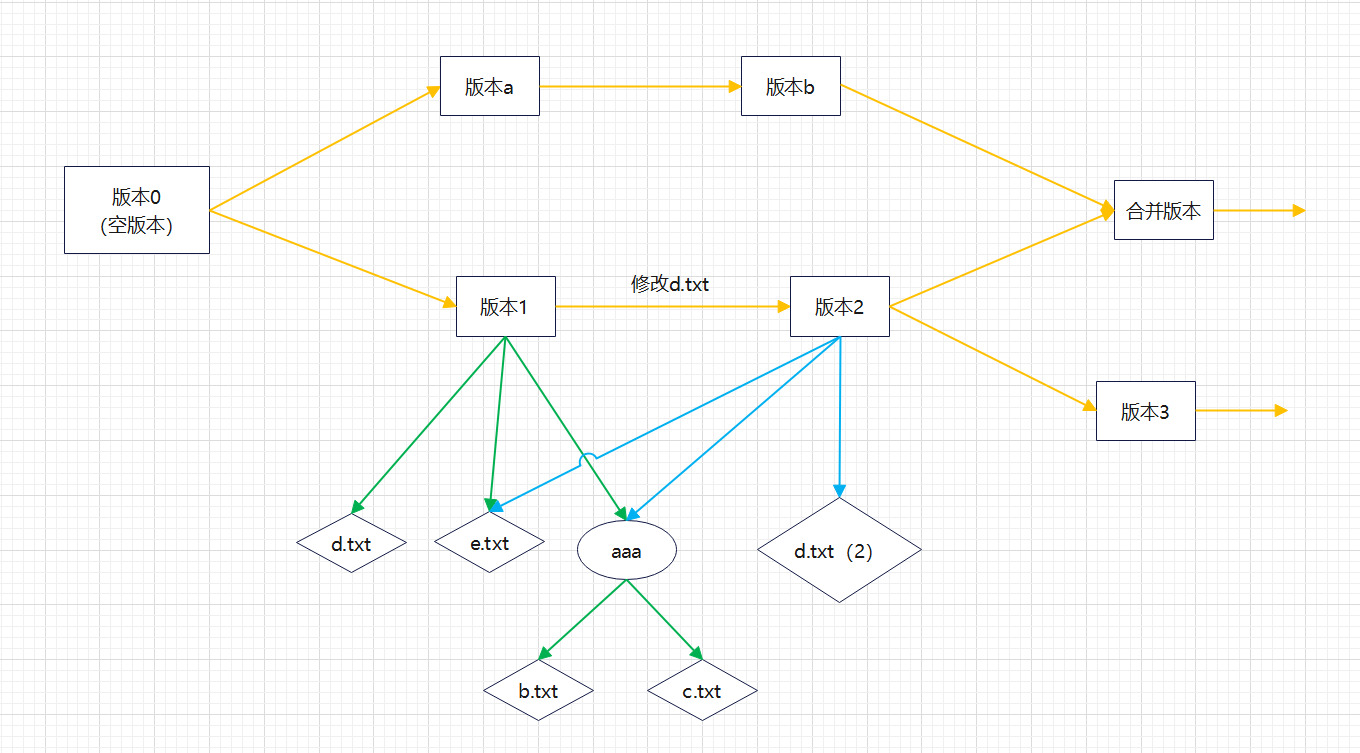
本系统主要是为了方便用户的使用的，应该以用户的方便为主旨，但是为了提高系统的性能，只能适当放弃方便性原则。

# 5系统总体结构

本系统是一款可视化的版本控制系统。主要是为在协同开发以及数据备份等应用场景存在需求的个人和团队提供服务，主要为使用者提供了仓库的创建，撤销更改刷新工作区，提交更改以及对于节点和工作区的管理。同时，本系统提供图像界面展示版本关系，显示各个版本节点、分支之间的继承推进关系。

模块总体结构：



版本控制结构：

# 6开发环境的配置

|  |  |  |
| --- | --- | --- |
| 类别 | 标准配置 | 最低配置 |
| 计算机硬件 | 奔腾3以上CPU，1G以上内存 | 奔腾3以上CPU，512M以上内存 |
| 软件 | Windows操作系统，Qt开发框架，C++编译器，MINGW | Windows操作系统，Qt开发框架，C++编译器，MINGW |
| 其他 |  |  |

# 7运行环境的配置

|  |  |  |
| --- | --- | --- |
| 类别 | 标准配置 | 最低配置 |
| 计算机硬件 | 奔腾3以上CPU，1G以上内存 | 奔腾3以上CPU，1G以上内存 |
| 软件 | Windows操作系统 | Windows操作系统 |
| 其他 |  |  |

# 8测试环境的配置

单元测试与集成测试的环境配置：计算机硬件要求奔腾3以上CPU，1G以上内存，10G以上剩余硬盘空间；软件要求Windows操作系统。

系统测试与验收测试环境配置：计算机硬件要求奔腾3以上CPU，1G以上内存，10G以上剩余硬盘空间；软件要求Windows操作系统。

# 9其他