GitMining分层模型概略设计文档V0.3

1. 更新记录

|  |  |  |  |
| --- | --- | --- | --- |
| 2016.02.28 | v0.1a，v0.1b，v0.1c | 王子安，徐江河，徐家逸 | 分别进行初步架构 |
| 2016.03.01 | v0.2 | 王子安 | 设计统一后概述 |
| 2016.03.02 | v0.3 | 徐家逸 | 对概述的补充整合 |
|  |  |  |  |

1. 逻辑视角

GitMining系统选择分层体系结构，系统共分为4层，分别是展示层（Presentation）、逻辑计算层（Logic）、数据存储层（Data）、网络层（network）。展示层用于展现GUI界面，逻辑层用于数据的获取和筛选、图表的生成、大数据的统计计算，数据层用于数据的调度和存储，网络层用于数据结构的转化、数据的过滤、网络的连接、API的调用。

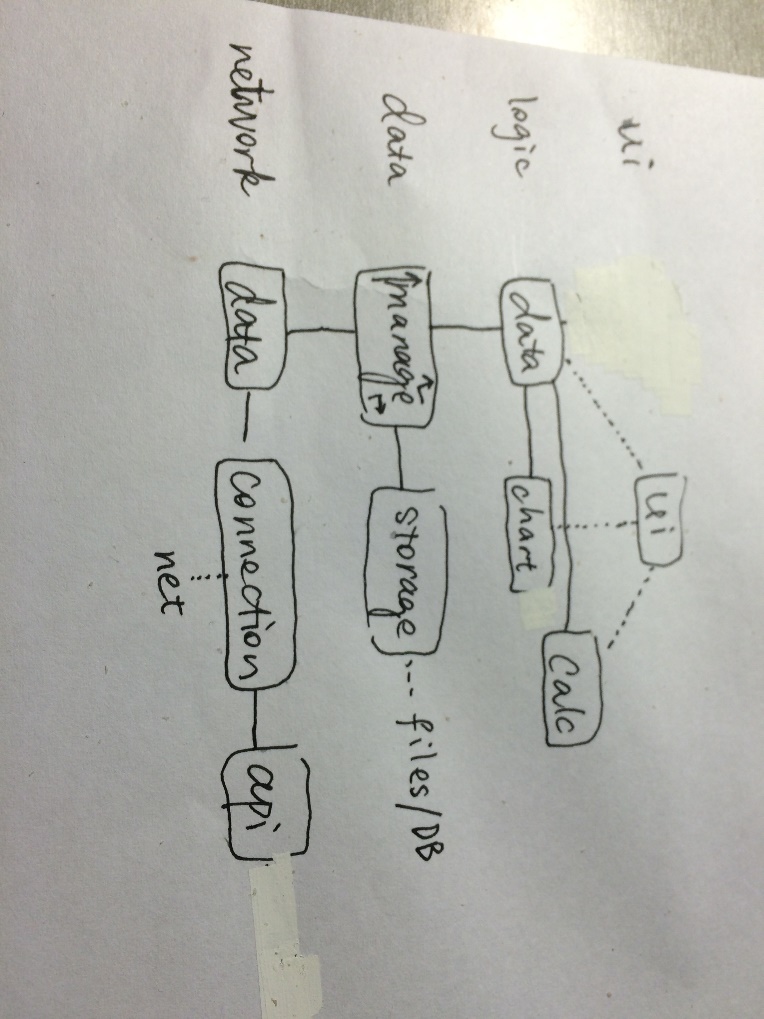
1. 层与层之间的分工与协作
2. 展示层Presentation
   * 1. 内部模块：单一模块presentation.ui。
     2. 职责：构建用户界面的窗体、列表、按钮、图表等以展示数据；接受用户界面的搜索、详细查看、更新数据等输入请求，从Logic获取相关统计数据、图表对象与计算结果。
     3. 协作者：Logic层各模块。
3. 逻辑层Logic
   * 1. 内部模块logic.data数据整理模块，logic.chart图表生成模块，logic.calc数据计算模块。
     2. 职责：
        1. logic.data：拥有多种搜索策略，并使用特定搜索策略从Data层获取Repository、User等信息，请求Data层向网络层委托以获得实时数据，请求Data层更新数据；
        2. logic.chart：利用logic.data的数据接口与（第三方）图表生成类制作图表对象；
        3. logic.calc：利用logic.data的数据接口与数学计算公式计算所需统计数据。
     3. 协作者：
        1. logic.data：Data层的data.manage模块；
        2. logic.chart：logic.data模块、logic.calc模块；
        3. logic.calc：logic.data模块。
4. 数据层Data
   * 1. 内部模块：data.manage数据管理模块，data.storage数据存取模块
     2. 职责：
        1. data.manage：向Network层请求新数据，回复Logic对网络实时数据的委托请求，与data.storage交互以存储或读取Git相关数据，对数据进行管理、控制数据的版本号等信息、使用Logic的搜索策略进行筛选；
        2. data.storage：对来自data.manage模块的Git相关数据进行格式化与解格式并进行存储、读取。
     3. 协作者：
        1. data.manage：Network层的network.data模块，data.storage模块；
        2. data.storage：~~data.manage~~模块
5. 网络层Network
   * 1. 内部模块：network.api API管理模块，network.connection 链接模块，network.data数据处理模块
     2. 职责：
        1. network.api：保存信息更新所需的API数据；
        2. network.connection：利用network.api提供的API与服务器链接取得数据；
        3. network.data：将从network.connection获取的数据进行处理，格式化为系统使用的Java对象。
     3. 协作者：
        1. network.api：无；
        2. network.connection：network.api模块；
        3. network.data：network.connection模块。
6. 项目架构风险预估及解决方案
   1. （老师上课威胁的）未来网站API提供的信息内可能有噪音：

在network.data中两个Pipeline中间添加Filter进行筛选，滤除错误信息。

* 1. （迭代三会发生的）数据存储方式由自定义文件系统变为数据库：

保持data.storage模块的数据存取接口不变，为数据库系统实现新的接口实现类。

* 1. 需要替换第三方图表绘制JAR包：

构建时为每种图表建立Wrapper类，为每套第三方JAR包的对象设计一套子类，利用方法委托隐藏内部实现，以方便地替换logic.chart模块实现。

临时放张图 方便理解.

1. 组合视角
2. 开发包的定义：

presentation\_ui：界面展示层包

presentation\_control：展示层与逻辑层对接包

logic\_chart：逻辑层负责绘制图表的包

logic\_calcul：逻辑层负责计算与数据处理的包

logic\_datum：逻辑层负责对于大量数据排序传递返回数据的包

data\_manage：数据层负责数据流动的包

data\_storage：数据层负责数据读写的包

net\_datum：网络层负责数据转换的包

1. 接口的定义：（详情见接口说明文档）