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 | 对徐江河的版本进行简单修改 |
| 2016.03.02 | v0.3 | 徐家逸 | 对概述的补充整合 |
| 2016.03.03 | V1.0 | 王子安 | 添加GitMining的层次划分和职责分配，具体阐述系统分层和职责的内容和过程 |
| 2016.03.05 | V1.1 | 王子安 | 修改网络部分的设计错误(依赖错误)  修改模块设计的图片 |

1. 层次划分和职责分配

层次划分：

GitMining系统的主要需求包括从搜索github项目和用户，展示github项目和用户信息。考虑到后期迭代的大数据挖掘、数据的可视化展示，系统需要提供大量的统计计算功能和多种图表的绘制，同时大量数据的保存和更新需要文件系统或者数据库的支持，数据通过GitMining提供的api获取并经过格式化和过滤产生系统可使用的数据。

简要分析以上需求，发现系统存在明显的数据可视化展示（界面）、存取（数据）、计算搜索（逻辑）和网络获取数据（网络）的划分，使用分层系统得以隔离四大职责，降低完成任务的复杂度。

职责分配：

界面层的职责包括图表的展示，接受搜索并展示搜索结果，展示项目和用户的详细信息，接收用户的统计请求，请求更新本地系统数据。所有职责集中到一个模块完成。项目和用户的详细信息来源于逻辑层的数据模块、统计计算请求逻辑层的统计模块完成、图表由逻辑层的图表生成模块提供、搜索通过请求逻辑层的数据整理模块使用多种搜索策略完成。

逻辑层包括3个模块：数据整理模块、图表生成模块、数据计算模块。3个模块完成界面层请求的4类任务（查询排序和搜索、提供图表、统计计算、数据更新）。

其中数据整理模块负责完成界面展示层的数据查询和搜索请求、逻辑层的图表生成模块和数据计算模块的数据请求、对查得的数据进行排序的请求，离线数据请求通过数据层的数据管理模块由数据层的数据存取模块取得，在线数据请求同通过数据层的数据管理模块委托网络层的数据处理模块获得。并且，数据处理模块向数据层请求数据的更新。搜索和排序借助搜索策略和排序策略实现。

图表生成模块通过第三方图表接口完成图表的生成，生成的图表可以交给界面层直接进行展示或者由界面层进行一定的包装后进行展示，用于生成图表的数据来源于逻辑层的数据整理模块。

数据计算模块的职责包括对使用数学公式对大量数据的统计计算，数据同样来源于数据整理模块，对数据处理、计算的核心部分集中于此。

数据层包括2个模块：数据管理模块、数据存取模块。

数据管理管理模块为逻辑层获取本地离线数据或者网络在线数据提供了接口上的选择，使用哪种类型的数据交由逻辑层进行判断。此模块控制数据从本地文件（或数据库）系统和网络向逻辑层的流入，同时控制数据从网络向本地文件（或数据库）系统的流入，即数据进行更新的要求。此模块控制数据的版本号等信息以提供对于数据更新的支持，数据更新的请求由向网络层的数据处理模块委托来得到。数据管理模块可以通过逻辑层提供的搜索策略为其提供较小量数据以避免系统在性能上的缺陷。

数据存取模块完成对于数据的格式化和解格式化及数据的存取。数据存取重视接口和实现尽可能分离，以此应对之后数据系统更换的风险。

网络层分为3个模块：数据处理模块、链接模块、API管理模块

数据处理模块借助API管理模块提供的api想链接模块请求网络数据、对从网络获取的数据进行格式化和过滤。一方面，过滤和格式化后的数据提供给数据层的数据管理模块并转交给逻辑层的数据整理模块以用于实时数据的处理和展示，另一方面，数据处理模块接受数据层的委托，从链接模块获取新数据流入数据存取模块用于数据的更新。

链接模块的职责包括与网络进行连接，过程中使用数据处理模块提供的API获取网络服务器的源数据，并将此数据交给数据处理模块进行格式化和过滤。

API管理模块向数据处理模块提供获取数据所用的API，供信息更新使用。

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. 层次与层间的职责与协作
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：拥有多种搜索策略和排序策略，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，对请求得到的数据进行排序；
        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：无
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：从network.api中获取向network.connection请求数据所需的API、对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：无；
        3. network.data：network.connection模块，network.api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。

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）