

智能知识图谱平台

1. 目录

智能知识图谱平台

- 1. 目录
- 2. 引言
 - 2.1 目的
 - 2.2 范围
 - 2.3 参考文献
- 3. 总体描述
 - 3.1 项目前景
 - 3.2 项目功能
 - 3.3 用户特征
 - 3.4 约束
 - 3.5 假设与依赖
- 4. 详细需求描述
 - 4.1 对外接口需求
 - 4.1.1 用户界面
 - 4.1.2 硬件接口
 - 4.1.3 软件接口
 - 4.1.4 通信接口
 - 4.2 功能需求
 - 4.2.1 用户上传预定义格式的知识图谱
 - 4.2.2 用户查看知识图谱
 - 4.2.2.1 特征描述
 - 4.2.2.2 刺激/响应序列
 - 4.2.2.3 相关功能需求
 - 4.2.3 用户编辑知识图谱
 - 4.2.3.1 特征描述
 - 4.2.3.2 刺激/响应序列
 - 4.2.3.3 相关功能需求
 - 4.2.4 用户导出知识图谱为数据文件
 - 4.2.4.1 特征描述
 - 4.2.4.2 刺激/响应队列
 - 4.2.4.3 相关功能需求
 - 4.2.5 用户导出知识图谱为图片
 - 4.2.5.1 特征描述
 - 4.2.4.2 刺激/响应队列
 - 4.2.4.3 相关功能需求
 - 4.3 非功能需求
 - 4.3.1 安全性
 - 4.3.2 性能
 - 4.3.3 可靠性
 - 4.3.4 易用性
 - 4.3.5 可移植性
 - 4.3.6 约束
 - 4.4 数据需求
 - 4.4.1 数据定义
 - 4.4.2 默认数据
 - 4.4.3 数据格式要求
 - 4.5 其他

2. 引言

本文档详细描述了 nioCoin 知识图谱可视化系统的需求。该文档为需求及软件系统开发的根据。

本文档仅面向开发者，不将终端用户作为目标读者。

2.1 目的

该系统用于对给定格式知识图谱的可视化，可以帮助人类或机器对中规模数据中包含的特征进行发现。

2.2 范围

该文档包含 nioCoin 项目的总体描述，系统的对外接口，功能性需求与非功能性需求。

2.3 参考文献

1. IEEE 需求规格文档标准
2. [软件开发的技术基础](#)
3. [软件工程与计算\[三\]——团队与软件开发实践](#)

3. 总体描述

3.1 项目前景

同时随着社会的发展，越来越多的数据对公众可见，如何利用好这些数据，分析并得出有意义的指导结论，帮助使用者的各类决策，成为一个不可忽视的问题。本项目立足于数据分析中常用的知识图谱技术，提供了简单易用的可视化接口，帮助用户完成实体和实体关系的抽取，预览，存储，提供高效的图片查询乃至补全推理，助力社会公共知识体系的建设。

3.2 项目功能

当前系统包括了数据的导入导出，数据上传，数据可视化，简易的实体/关系编辑功能。

3.3 用户特征

用户是无（或不普遍具备）计算机背景知识，但期望从数据中获取直观知识的人，系统设计应满足简单易用，封装不必要的复杂选项。

3.4 约束

CON1：系统使用 Java 开发。**CON2**：系统使用迭代式开发。**CON3**：系统应该基于 Web。**CON4**：在每个迭代周期中，开发者要提交并维护

1. 计划文档
2. 软件需求规格说明文档
3. 设计描述文档
4. 测试报告

3.5 假设与依赖

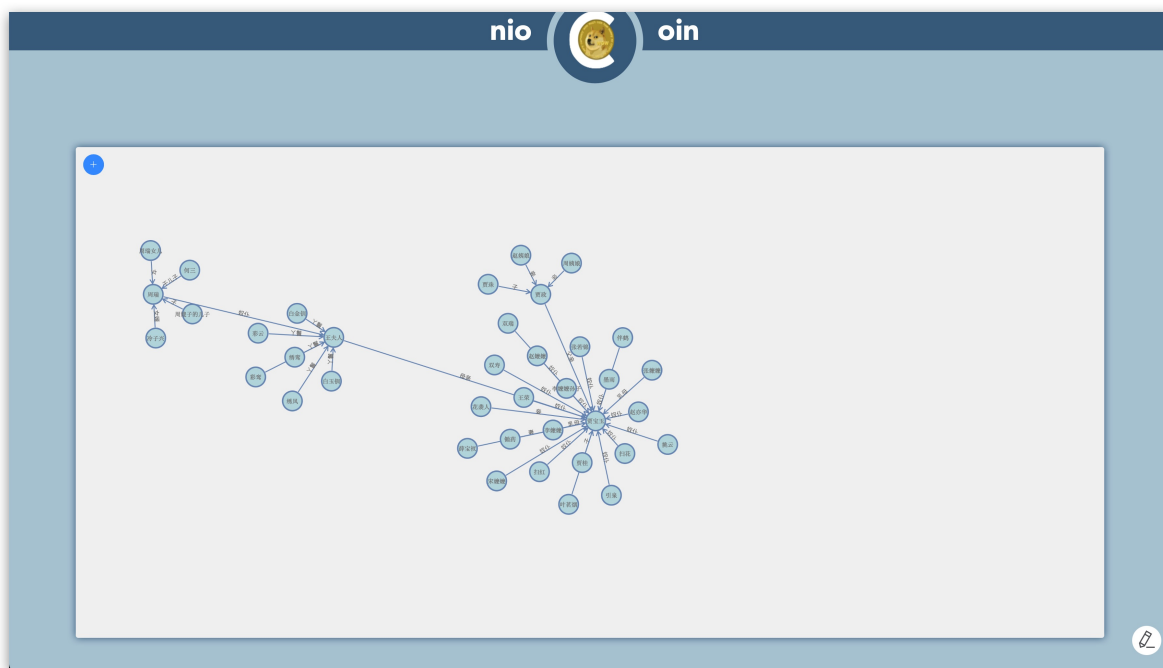
AS1：用户在流畅（ $BandWidth \geq 10M$ ）的网络环境下使用本系统。

AS2：用户不具有过大规模（ $|Nodes| \geq 10000$ ）的可视化请求。

4. 详细需求描述

4.1 对外接口需求

4.1.1 用户界面



4.1.2 硬件接口

内存: 4GB及以上

显存: 512MB及以上

4.1.3 软件接口

用户安装了主流浏览器(Chrome, Firefox, Edge, Safari)

4.1.4 通信接口

用户浏览器与服务器使用 HTTP 协议进行通信。

4.2 功能需求

4.2.1 用户上传预定义格式的知识图谱

4.2.2 用户查看知识图谱

4.2.2.1 特征描述

用户上传知识图谱后，获取知识图谱所有的节点和关系信息

优先级=高

4.2.2.2 刺激/响应序列

刺激：用户选择上传的知识图谱

响应：系统返回用户可视化的知识图谱

刺激：用户移动光标到具体的节点或关系

响应：系统返回用户节点或关系的相关信息

4.2.2.3 相关功能需求

编号	描述
kgbackend.transJSON	将Neo4j数据库中存储的节点和关系信息返回

4.2.3 用户编辑知识图谱

4.2.3.1 特征描述

用户获取到知识图谱后，可以通过文本编辑实现对知识图谱的增删改

优先级=高

4.2.3.2 刺激/响应序列

刺激：用户选择编辑知识图谱

响应：系统返回可编辑的文本形式的知识图谱界面

刺激：用户选择增加节点或关系

响应：系统增加数据库中的节点或关系并更新

刺激：用户选择删除节点或关系

响应：系统删除数据库中的节点或关系并更新

刺激：用户选择修改节点或关系

响应：系统修改数据库中的节点或关系并更新

4.2.3.3 相关功能需求

编号	描述
kgbackend.modify	修改Neo4j数据库中的节点或关系状态
kgbackend.transJSON	将Neo4j数据库中存储的节点和关系信息返回

4.2.4 用户导出知识图谱为数据文件

4.2.4.1特征描述

用户可以在操作界面将知识图谱的内容导出为xml格式文件

优先级=高

4.2.4.2刺激/响应队列

刺激：用户选择界面操作

响应：系统展示可进行的操作

刺激：用户选择导出数据文件

响应：系统自行导出下载xml文件供用户本地查看

4.2.4.3相关功能需求

编号	描述
kgbackend.getGraphData	从neo4j请求图谱数据，以便前端进行导出

4.2.5 用户导出知识图谱为图片

4.2.5.1特征描述

用户可以在操作界面将知识图谱的内容导出为png格式图片

优先级=高

4.2.4.2刺激/响应队列

刺激：用户选择界面操作

响应：系统展示可进行的操作

刺激：用户选择导出图片

响应：系统自行导出下载png图片供用户本地查看

4.2.4.3相关功能需求

编号	描述
kgbackend.getGraphData	从neo4j请求图谱数据，以便前端进行导出

4.3 非功能需求

4.3.1 安全性

Safety1：系统对用户应该只能开放节点的名称编辑

Safety2：系统对用户只能开放联系的名称的编辑

Safety3：系统应该只对有权限的用户开放

4.3.2 性能

Performance1：系统获取知识图谱的时间不能超过500ms

Performance2：系统导入知识图谱的响应时间不能超过3s

Performance3：系统的超时错误率不能超过0.1%

4.3.3 可靠性

Reliability1：如果在客户交互时，网络出现故障，系统不能出现故障。

Reliability2：数据库的数据实时备份，在丢失或者破损之后可以自动回复。

4.3.4 易用性

Usability1：无需用户使用手册或者专业培训即可直接使用系统

Usability2：用户编辑知识图谱或是导入知识图谱不会超过三个模态框。

4.3.5 可移植性

Portability1：系统或部件可以部署到Windows以及Linux服务器中

Portability2：数据库可以在Windows，Linux，MacOS中进行多平台移植。

4.3.6 约束

C1：系统采用分层模型开发

C2：系统使用neo4j数据库存储数据

C3：系统使用Java的Springboot+Vue3进行开发

4.4 数据需求

4.4.1 数据定义

DR1：系统需要存储来自不同数据源的知识图谱。

DR2：系统需要存储用户一定时间段内的操作记录。

DR3：系统删除的所有数据需要继续存储一段时间，以保证历史数据的完整性。

4.4.2 默认数据

在没有数据的时候默认在数据库中存入字符串null

4.4.3 数据格式要求

Format1：节点和关系的编辑数据统一使用json进行传输。

4.5 其他

系统安装部署的时候需要同时导入一些数据，保证系统在安装部署之后用户可以直接进行一定的试用体验。