

知识图谱定义及可视化系统 COIN

(A system for COnstructing and vlsualizing kNowledge graph)

需求规格说明文档

团队名称：nekoCoin

变更记录

修改人员	日期	原因	版本
冯波云	2021/3/20	初稿	ver1.0
李泳劭	2021/3/21	统一化风格	ver1.1

1. 引言

1.1 目的

- 本文档描述了知识图谱COIN的功能需求和非功能需求。开发小组的软件系统实现和验证工作都以此文档为依据。
- 除特殊说明外，本文档所包含的需求都是高优先级需求。

1.2 范围

- COIN项目能够读取预先定义的知识图谱并进行在线显示，可以通过在线以文本的形式编辑知识图谱，可以对知识图谱中要素和关系进行增加、修改、删除，并能够将图谱导出xml, 图片等文件。

1.3 参考文献

- 《COIN知识图谱定义及可视化系统》
- 《软件工程与计算(二) 软件开发的技术基础》
- 《软件工程与计算(三) 软件开发的团队基础》

2. 总体描述

2.1 背景与机遇

随着智能信息服务应用的不断发展，知识图谱已被广泛应用于智能搜索、智能问答、个性化推荐、情报分析、反欺诈等领域。知识图谱以其强大的语义处理能力与开放互联能力，可为万维网上的知识互联奠定扎实的基础，使 Web 3.0 提出的“知识之网”愿景成为了可能。知识图谱是将复杂的信息通过计算处理成能够结构化表示的知识，所表示的知识可以通过图形绘制而展现出来，为人们的学习提供有价值的

参考，为信息的检索提供便利。通常知识图谱通过对错综复杂的文档的数据进行有效的加工、处理、整合，转化为简单、清晰的“实体,关系,实体”的三元组，最后聚合大量知识，从而实现知识的快速响应和推理。然而，市面上大部分的知识图谱平台普遍存在手工操作复杂、可视化过于繁乱等问题。

2.2 用户特征

用户：用户使用本项目能够获得知识图谱，并对其进行增加、修改和删除等操作，以此获得结构化的信息，能更好地分析问题

管理员：主要维护知识图谱数据库信息与用户信息

2.3 项目功能

SF1: 允许用户增加知识图谱中实体-关系-实体

SF2: 允许用户修改知识图谱（以文本方式）

SF3: 允许用户删除知识图谱中实体和关系

SF4: 允许用户进行撤销删除操作

SF5: 允许用户从客户端上传指定格式的包含只是图谱实体与关系的文本文件

SF6: 允许用户进行知识图谱导出xml，图片等文件

2.4 约束

CON1: 系统将运行在网页浏览器上。

CON2: 用户、网站管理人员都通过web使用本系统。

2.5 假设与依赖

i. 网络畅通，无故障

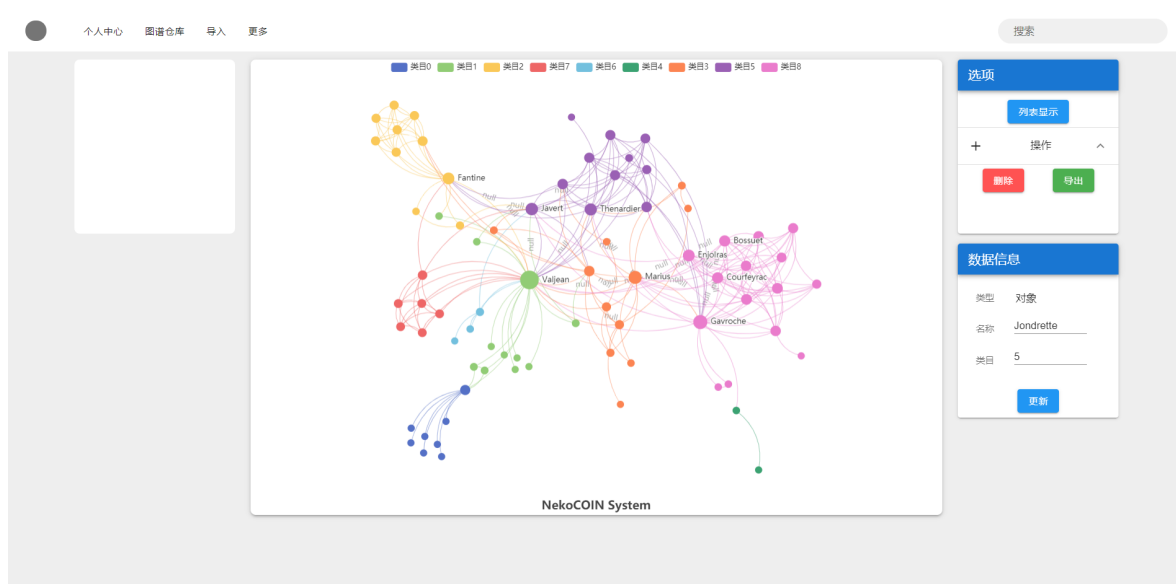
ii. 所有操作任务的完成时间不会超过 1 个小时，即使用暂停与中断不会超过 1 个小时

3.1 对外接口需求

3.1.1 用户界面

整体界面采取扁平化设计，界面最上方为导航栏；右侧有为用户提供的相关操作按钮：列表显示知识图谱，修改知识图谱(文本形式)，导出知识图谱和删除知识图谱。界面中间为知识图片的可视化区域。

点击知识图谱中的节点和关系时，浏览器右侧会出现矩形信息栏。



3.1.2 通信接口

本系统用户端与服务器采用http方式进行通信

3.2 功能需求

3.2.1 导出知识图谱

3.2.1.1特征描述

- 优先级：高
- 用户选择导出格式后，获得相应格式的文件

3.2.1.2 刺激/响应序列

- 刺激：用户选择“导出”。
- 响应：系统提示可选的导出格式，包括xml、json、图片等。
- 刺激：用户选择目标格式后并确认。
- 响应：系统进行后台计算、处理，完毕后用户浏览器会自动开始下载导入文件。

3.2.1.3 相关功能需求

GraphController.getAllNodesAndRelations	从后台获取所有节点和关系信息
NodeService.retrieveAll	调用nodeDao层方法从数据库获取数据
NodeDao.retrieveAll	从数据库查询所有节点和关系并返回

3.2.2 导入知识图谱

3.2.2.1 特征描述

- 优先级：高
- 用户上传相应格式的文件后，系统为其计算出相应的知识图谱并展示

3.2.2.2 刺激/响应序列

- 刺激：用户选择“导入”
- 响应：系统提示用户上传本地文件以及允许的格式(.CSV格式)

- 刺激：用户选择目标文件后并确认
- 响应：系统进行后台计算、处理，完毕后显示相应的知识图谱

3.2.2.3 相关功能需求

GraphController.importFile	接收用户传入的文件
GraphService.importGraphFromFile	将文件转化问Graph对象并持久化到数据库

3.2.3 在线编辑知识图谱

3.2.2.1 特征描述

- 优先级：高
- 用户对知识图谱进行在线操作

3.2.2.2 刺激/响应序列

- 刺激：用户点击修改
- 响应：系统提示可进行的操作，包括添加、删除节点、更改节点和关系。
- 刺激：用户选择更改节点(关系)，并输入旧的节点(关系)名称和新的节点(关系)名称。
- 响应：系统更新节点(关系)的属性。
- 刺激：用户选择添加节点，并输入要添加的节点名称
- 响应：系统新建节点并显示。
- 刺激：用户选择删除节点(关系)，并输入要删除节点(关系)的名称(开始节点，目标节点)。
- 响应：系统删除该节点(关系)，同时删除该节点相邻的所有关系连线。

3.2.2.3 相关功能需求

NodeService.insert	添加新节点
NodeService.deleteNodeByName	根据节点名称删除节点
NodeService.updateNode	修改节点名称
RelationshipService.buildRelation	添加关系
RelationshipService.deleteRelationByValue	根据关系名称删除关系
RelationshipService.deleteRelationByNodes	根据首尾节点名称删除关系
RelationshipService.updateRelation	更新关系

3.3 非功能需求

3.3.1 安全性

- Safety1：用户只能查看和维护自己的个人信息、知识图谱。
- Safety2：网站管理人员不能随意违规查看、修改用户的信息。
- Safety3：为防信息泄露，用户的账号、密码、姓名（名称）、联系方式等敏感数据必须密文存储。

3.3.2 可维护性

- Modifiability1: 关系和实体，将来可能会出现新的类型：
 - Modifiability1.1: 当关系和实体出现新类型时，系统应该能够在4个人一天内更新完毕

3.3.3 易用性

- Usability1: 客户初次使用时，能在十分钟内完成从知识图谱可视化到增加或删除节点的操作。

3.3.4 可靠性

- Reliability1:在客户端与服务器通信时，如果网络故障，系统不能出现故障。
 - Reliability1.1:客户端应该检测到故障，并尝试重新连接网络3次，每次15秒。
 - Reliability1.1.1:重新连接后，客户端应该继续之前的工作。
 - Reliability1.1.2:如果重新连接不成功，客户端应该等待5分钟后再次尝试重新连接。
 - Reliability1.1.2.1:重新连接后，客户端应该继续之前的工作。
 - Reliability1.1.2.2:如果重新连接仍然不成功，客户端报警。

3.3.5 约束

- IC1:系统要在网络上分布为一个服务器和多个客户端。

3.4 数据需求

3.4.1 数据定义

- DR1:系统需要存储单个用户的完成的增删改查记录。
- DR3:数据实体及其关系详情见下表:

对象类型	持有的数据
实体	实体类目 实体名称 实体大小 实体左边 实体右边 实体参与关系数
关系	关系出发节点 关系目标节点 关系值
图谱	组成图谱的节点 组成图谱的关系 图谱包含的类目
类目	类目名称

3.4.2 默认数据

- Default1:实体类目默认为0。
- Default2:类目名称默认为"类目0"。
- Default3:系统时间默认为当天。
- Default4:关系的连线类型默认为实线。

3.4.3 数据格式要求

- Format1:节点的数量必须为正整数。
- Format2: 坐标为双精度浮点数，可以是负值。
- Format3: 密码格式 长度 ≥ 6 的字符串。
- Format4: 日期格式 yyyy-MM-dd。
- Format5: 时间格式 HH:mm:ss。
- Format6: 日期时间，都采用北京时间，即“+8:00”。
- Format7: 数量，都是非负整数。