项目设计文档

团队名称：ipoooo6

更新记录

日期	作者	变更说明	版本号
2021.03.10	吴雨晴	添加1-4.22内容	v1.0
2021.03.10	张嘉玥	添加4.2.3-5内容	v1.1
2021.03.11	吴雨晴	添加物理包图	v1.2
2021.03.16	冯泊涓	修改接口设计和架构	v1.3
2021.03.20	冯泊涓	修改接口设计和架构	v1.4
2021.03.28	冯泊涓	迭代二更新	v2.0
2021.04.18	冯泊涓	迭代二终版	v2.1

1.引言

1.1编制目的

本文档描述了COIN知识图谱定义及可视化系统的概要设计，说明详细设计和开发的目的，同时实现和测试人员及用户的沟通。开发小组的软件系统实现与验证工作都以此文档为依据。

1.2词汇表

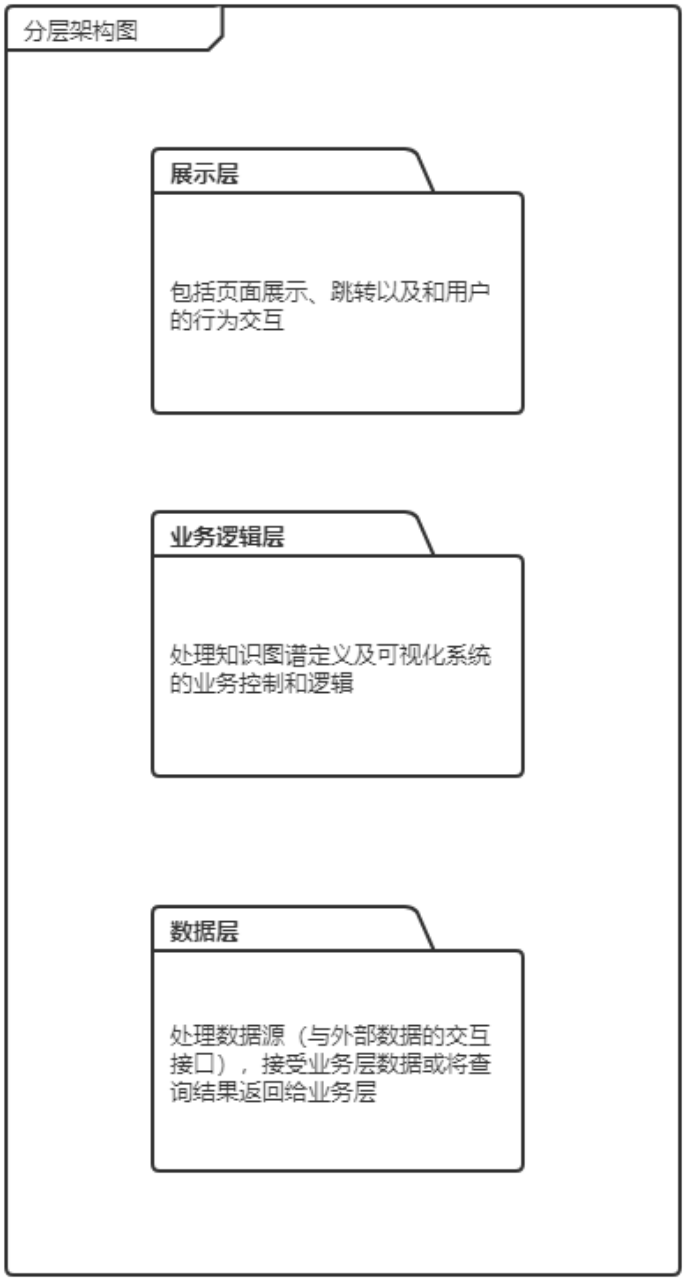
词汇名称	词汇含义	备注
COIN	知识图谱定义及可视化系统	

1.3参考资料

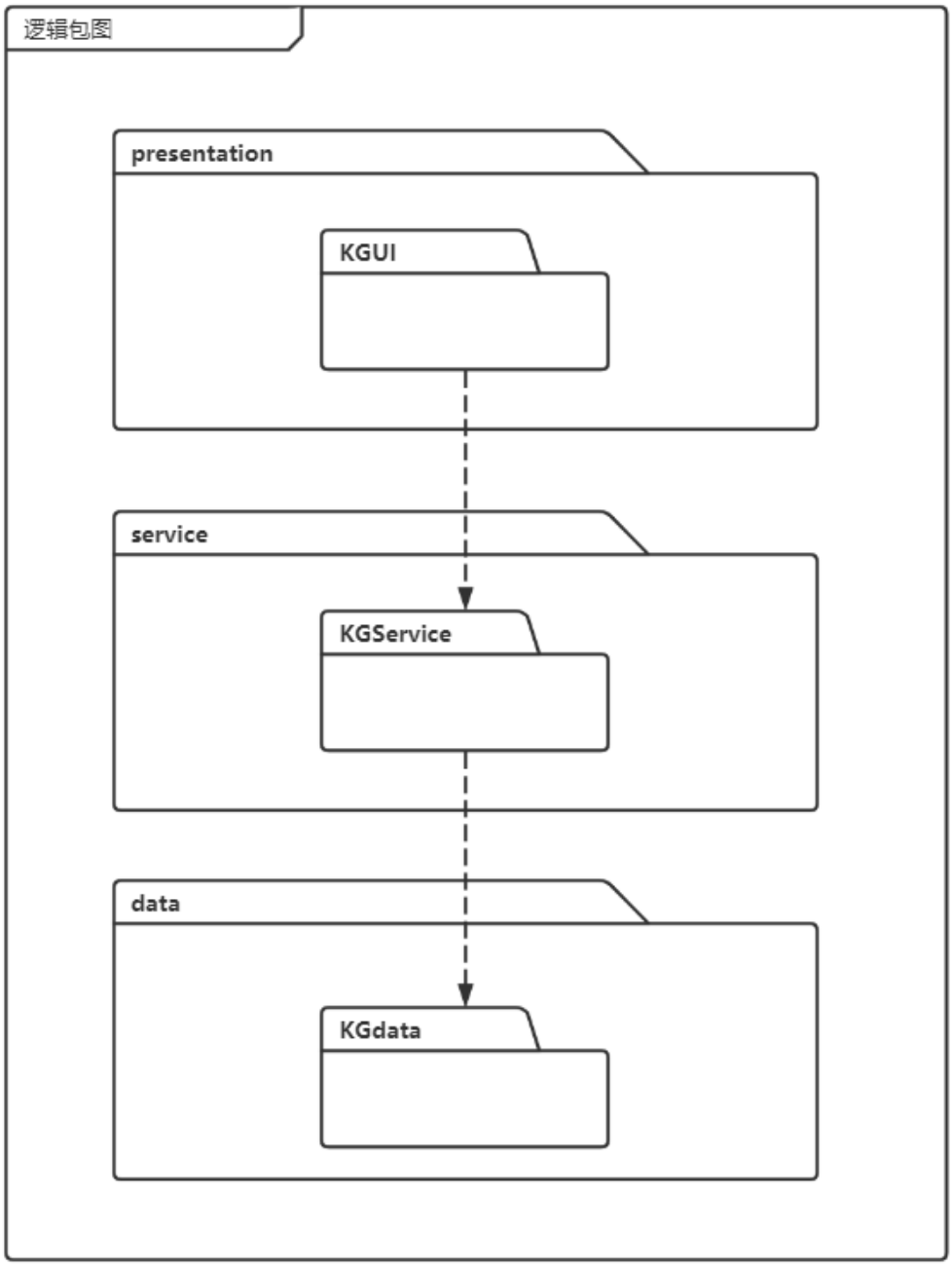
1. 软件工程与计算(卷二)——软件开发的技术基础/骆斌主编.-北京：机械工业出版社,2012.12（2018.4重印）
2. COIN需求规格说明文档v1.0

2.逻辑视角

2.1分层架构图



2.2逻辑包图



3.组合视角

3.1物理包的划分

开发（物理）包	依赖的其他开发包
KGController	VO，KGService
KGService	VO，PO，KGData，util
KGData	util，PO
VO	
PO	
util	
conf	util
view	

3.2物理包图

4.接口视角

4.1模块的职责

4.2模块的接口规范

4.2.1用户界面模块的分解

模块	职责
KGUI	页面的显示与跳转

需要的服务（需接口）	
服务名	服务
KGService.importJSON	上传JSON格式知识图谱
KGService.getAll	获取所有节点与关系
KGService.addEntity	添加指定名称的实体
KGService.addRelation	添加指定两个实体间的关系
KGService.updateEntityName	更新实体名称
KGService.updateRelationName	更新关系名称
KGService.reverseRelationDir	更新关系方向
KGService.deleteEntity	删除指定实体
KGService.deleteRelation	删除指定关系
KGService.getRelativeRelation	获取当前实体相关关系
KGService.getByLabel	从数据库获取某标签的知识图谱
KGService.searchEntityByName	根据节点名搜索节点
KGService.searchRelationByName	根据关系名搜索关系
KGService.updateLocation	保存布局
KGService.updateEntityProperty	更新实体属性
KGService.updateRelationProperty	更新关系属性
KGService.getDistinctLabels	获得不同实体标签及其统计数
KGService.getDistinctRelationNames	获得不同关系名及其统计数
KGService.getTree	根据实体id获得实体指定深度的树结构关系图谱

4.2.2业务逻辑模块的分解

模块	**职责**
KGService	负责知识图谱可视化相关逻辑

KGService.importJSON	语法	public ResponseVO importJSON(String data);
	前置条件	data符合JSON格式
	后置条件	数据库添加导入的知识图谱
KGService.getAll	语法	public ResponseVO getAll();
	前置条件	数据库中节点关系格式符合知识图谱定义
	后置条件	返回数据库中知识图谱所有节点和关系
KGService.addEntity	语法	public ResponseVO addEntity(EntityVO entityVO);
	前置条件	传入数据格式合法
	后置条件	添加实体并返回自动生成的实体id
KGService.addRelation	语法	public ResponseVO addRelation(RelationVO relationVO);
	前置条件	relationVO中source,target,relationName均不为空
	后置条件	添加关系并返回自动生成的关系id

KGService.updateEntityName	语法	public ResponseVO updateEntityName(EntityVO entityVO);
	前置条件	entityVO中id,newName不为空
	后置条件	将该实体名称更改为newName
KGService.updateRelationName	语法	public ResponseVO updateRelationName(RelationVO relationVO);
	前置条件	relationVO中id合法
	后置条件	将该id关系改名为newName
KGService.reverseRelationDir	语法	public ResponseVO reverseRelationDir(RelationVO relationVO);
	前置条件	relationVO中id合法
	后置条件	将该id关系反向并返回新id
KGService.deleEntity	语法	public ResponseVO deleteEntity(EntityVO entityVO);
	前置条件	entityVO中id有效
	后置条件	将实体和该实体的所有关系删除

KGService.deleRelation	语法	public ResponseVO deleteRelation(RelationVO relationVO);
	前置条件	relationVO中id有效
	后置条件	将该id关系删除
KGService.getRelativeRelation	语法	public ResponseVO getRelativeRelation(long id);
	前置条件	实体id有效
	后置条件	返回所有与该实体有关的关系
KGService.getByLabel	语法	public ResponseVO getByLabel(List< String> labels);
	前置条件	labels不为空
	后置条件	返回数据库中符合label的所有节点及其关系
KGService.searchEntityByName	语法	public ResponseVO searchEntityByName(String name);
	前置条件	name不为空
	后置条件	返回数据库中模糊查询获得的所有节点

KGService.searchRelationByName	语法	public ResponseVO searchRelationByName(String name);
	前置条件	name不为空
	后置条件	返回数据库中查询获得的所有同类关系
KGService.updateLocation	语法	public ResponseVO updateLocation(List < EntityVO > entityVOList);
	前置条件	每个entity中id不为空且存在合法properties(需要__D3_PROPS__x,__D3_PROPS__y,__D3_PROPS__fixed属性字段)
	后置条件	更新数据库对应实体的属性
KGService.updateEntityProperty	语法	public ResponseVO updateEntityProperty(EntityVO entityVO);
	前置条件	entityVO中id，properties不为空
	后置条件	更新数据库中对应实体的属性
KGService.updateRelationProperty	语法	public ResponseVO updateRelationProperty(RelationVO relationVO);
	前置条件	relationVO中id，properties不为空
	后置条件	更新数据库中对应关系的属性

KGService.getDistinctLabels	语法	public ResponseVO getDistinctLabels();
	前置条件	无
	后置条件	返回数据库中所有不同label列表及其统计数
KGService.getDistinctRelationNames	语法	public ResponseVO getDistinctRelationNames();
	前置条件	无
	后置条件	返回数据库中所有不同关系名列表及其统计数
KGService.getTree	语法	public ResponseVO getTree(long id, int depth);
	前置条件	实体id有效
	后置条件	返回该实体为树根，深度为depth的关系树，depth默认为3

需要的服务（需接口）	
服务名	服务
KGdata.getAllRelations	获取知识图谱所有关系
KGdata.getAllEntities	获取知识图谱所有节点
KGdata.getRelativeRelation	根据id获取指定实体相关关系
KGdata.insertEntity	插入新的entity实体
KGdata.insertRelation	插入新的实体间关系
KGdata.searchEntity	根据id获取实体
KGdata.updateEntityName	更新实体名称
KGdata.getChildEntity	获取实体的子实体
KGdata.updateRelationName	更新关系名称
KGdata.updateRelationDir	更新关系方向
KGdata.delEntity	删除实体
KGdata.delRelation	删除实体间关系
KGdata.getByLabels	根据label查询实体
KGdata.getEntityByName	根据实体name查询实体
KGData.getRelationByName	根据关系名查询关系
KGdata.updateLocation	保存一批节点的布局
KGdata.updateEntityProperty	更新实体属性
KGdata.updateRelationProperty	更新关系属性
KGdata.getDistinctLabels	获得不同label列表及其统计数
KGdata.getDistinctRelationName	获得不同关系名列表及其统计数

4.2.3数据模块的分解

模块	**职责**
KGdata	负责知识图谱数据的处理

提供的服务（供接口）		
getAllEntities	语法	public List< Entity > getAllEntities();
	前置条件	数据库知识图谱格式合法
	后置条件	返回知识图谱所有节点
getAllRelations	语法	public List< Relation > getAllRelations();
	前置条件	数据库知识图谱格式合法
	后置条件	返回知识图谱所有关系
getAllRelativeRelation	语法	public List< Relation > getAllRelativeRelation(long id);
	前置条件	实体id合法
	后置条件	返回对应实体所有相关关系
insertEntity	语法	public long insertEntity(Map< String, Object> map);
	前置条件	map中properties含有name字段且字段合法不为空
	后置条件	添加实体并返回自动生成的实体id
insertRelation	语法	public long insertRelation(Map< String, Object> map);
	前置条件	map中source，target，name合法且不为空
	后置条件	添加从source到target名为name的实体关系并返回关系id
searchEntity	语法	public Entity searchEntityByName(long id);
	前置条件	id合法
	后置条件	根据id查询实体
updateEntityName	语法	public void updateEntityName(long id, String newName);

	前置条件	id合法，newName不为空
	后置条件	根据id修改对应实体名称为newName
updateRelationName	语法	public void updateRelationName(long id,String newName);
	前置条件	id,newName均合法
	后置条件	修改该id关系名称为newName并返回新的关系id
updateRelationDir	语法	public void updateRelationDir(long id);
	前置条件	id合法
	后置条件	修改该id关系方向，返回新关系id
delEntity	语法	public void delEntity(long id);
	前置条件	id合法
	后置条件	删除id对应的实体
delRelation	语法	public void delRelation(long id);
	前置条件	id合法
	后置条件	删除该id关系
getByLabels	语法	public List< Long> getByLabels(List< String> labels);
	前置条件	labels合法
	后置条件	根据labels查询实体，返回实体id列表
getEntityByName	语法	public List< Long> getEntityByName(String name);
	前置条件	name合法
	后置条件	根据name查询实体，返回实体id列表

getRelationByName	语法	public List< Long> Relation getRelationByName(String name);
	前置条件	name合法
	后置条件	根据name查询关系，返回关系id列表
updateLocation	语法	public void updateLocation(long id, double x, double y, boolean fixed);
	前置条件	参数合法
	后置条件	更新对应实体的属性
updateEntityProperty	语法	public Entity updateEntityProperty(long id,Map< String,Object> properties);
	前置条件	properties合法
	后置条件	更新对应实体的属性
updateRelationProperty	语法	public Relation updateRelationProperty(long id,Map< String,Object> properties);
	前置条件	properties合法
	后置条件	更新对应关系的属性
getDistinctLabels	语法	public List< Map< String,Object> > getDistinctLabels();
	前置条件	无
	后置条件	返回所有label:count
getDistinctRelationNames	语法	public List< Map< String,Object> > getDistinctRelationNames();
	前置条件	无
	后置条件	返回所有name:count

5.信息视角

5.1VO定义

ResponseVO

含义	属性	字段
调用成功标识符	boolean	success
提示信息	String	message
内容	Object	content

EntityVO

含义	属性	字段
实体id	Long	id
实体属性中的名称，用于搜索实体	String	name
实体属性中的名称，用于实体更名	String	newName
实体标签	List	labels
实体属性	Map<String,Object>	properties

RelationVO

含义	属性	字段
关系id	Long	id
关系名	String	name
新关系名	String	newName
关系的起始实体id	Long	source
关系的终点实体id	Long	target
关系属性	Map<String,Object>	properties

TreeVO

含义	属性	字段
本树节点id	Long	id
本树节点名称	String	name
本树节点子节点	ArrayList	children
上级节点到本节点的关系名	String	relation
本节点距离root的深度	Integer	depth