目 录

[1 总体概述 1](#_Toc22632304)

[1.1 背景 1](#_Toc22632305)

[1.2 总体目标 1](#_Toc22632306)

[1.3 需求 1](#_Toc22632307)

[1.3.1 业务需求 1](#_Toc22632308)

[1.3.2 系统需求 1](#_Toc22632309)

[1.4 术语 1](#_Toc22632310)

[2 知识数据来源 2](#_Toc22632311)

[2.1 概述 2](#_Toc22632312)

[2.2 维修服务 3](#_Toc22632313)

[2.2.1 维修服务系统关系数据库数据 3](#_Toc22632314)

[2.2.2 线下excel数据 4](#_Toc22632315)

[2.3 故障诊断 5](#_Toc22632316)

[2.3.1 概述 5](#_Toc22632317)

[2.3.2 故障代码类故障 5](#_Toc22632318)

[2.3.3 E类故障（电气系统） 5](#_Toc22632319)

[2.3.4 H类故障（液压和机械系统） 6](#_Toc22632320)

[2.3.5 S类故障（发动机系统） 7](#_Toc22632321)

[2.4 产品结构 8](#_Toc22632322)

[3 知识图谱模式设计 10](#_Toc22632323)

[3.1 知识图谱模型设计 10](#_Toc22632324)

[3.2 知识图谱模型存储结果 11](#_Toc22632325)

[4 系统开发说明 13](#_Toc22632326)

[4.1 概述 13](#_Toc22632327)

[4.2 开发环境 14](#_Toc22632328)

[4.3 将半结构化数据转为结构化数据 14](#_Toc22632329)

[4.3.1 概述 14](#_Toc22632330)

[4.3.2 输入 14](#_Toc22632331)

[4.3.3 输出 14](#_Toc22632332)

[4.3.4 处理逻辑 15](#_Toc22632333)

[4.3.5 代码结构 16](#_Toc22632334)

[4.4 将结构化数据导入neo4j中 16](#_Toc22632335)

[4.4.1 概述 16](#_Toc22632336)

[4.4.2 输入 17](#_Toc22632337)

[4.4.3 输出 17](#_Toc22632338)

[4.4.4 增量抽取/全量抽取 17](#_Toc22632339)

[4.4.5 读取结构化数据 18](#_Toc22632340)

[4.4.6 读取配置表 18](#_Toc22632341)

[4.4.7 导入Neo4j 18](#_Toc22632342)

[4.4.8 代码结构 20](#_Toc22632343)

[4.5 处理流程 20](#_Toc22632344)

[4.5.1 维修服务 20](#_Toc22632345)

[4.5.2 故障诊断 20](#_Toc22632346)

[4.5.3 产品结构 20](#_Toc22632347)

[5 知识库结果（展示当前数据库中的已有结果） 21](#_Toc22632348)

[5.1 概述 21](#_Toc22632349)

[5.2 维修服务 21](#_Toc22632350)

[5.3 故障诊断 23](#_Toc22632351)

[5.4 产品结构 26](#_Toc22632352)

**XX智能服务知识抽取与存储设计文档**

2019年9月30日

1. 总体概述
   1. 背景

在“XX”的大背景下，XX的知识图谱模型设计是智能服务知识图谱应用任务下的一个子任务。

* 1. 总体目标

根据设计阶段结果中的配置文件和各种数据源的数据，抽取实体和关系并存储在Neo4j图数据库中。

* 1. 需求
     1. 业务需求

1. 从多种数据源抽取知识，包括如下数据源：
   1. 维修服务系统目前的业务数据模型、以及多个线下excel格式的业务知识，其中包含9272条客户记录，28个服务人员，13989条设备信息，1832条工单记录等数据。
   2. “小松挖掘机装修手册”中的挖掘机“故障诊断”知识。
   3. “挖掘机零件数据表”和“小松挖掘机装修手册”中的挖掘机组成结构知识。
2. 把知识存储在Neo4j图数据库中。
   * 1. 系统需求

设计一种通用的可配置的知识抽取工具，包括以下特性：

1. 知识抽取工具：能够将数据源的知识转换为图数据库中的知识。
2. 通用性：不止是针对某种数据源某种知识，而是具有通用性，可应用于所有数据源。
3. 可配置性：可以通过配置文件修改数据源的知识和图数据库中节点和边的对应关系。
4. 增量式：允许数据和概念图扩充。
   1. 术语
   2. **模式图：**又称概念图。是知识图谱的核心，在数据图之上，以图的形式存储经过提炼和抽象的知识。
   3. **数据图：**又称实例图。按照模式图的组织结构，以图的形式存储真实的数据。
   4. **知识模型：**是知识图谱的核心，以模型的方式对知识进行抽象化表达
   5. **知识抽取模型：**是知识模型与数据源之间对接的桥梁，描述知识模型与数据源的映射关系。
   6. **知识：**所有表达信息的、存储在知识图谱中的数据
   7. **概念知识：**较为抽象的、描述了某一类事物的知识
   8. **实例知识：**非常具体的、描述事实的知识
   9. **元知识：**比概念知识更加抽象，可用来产生概念知识的一种知识
   10. **概念/概念节点：**知识模型的核心元素之一，可理解为提炼出的知识中的名词。如“挖掘机”。
   11. **实例/实例节点****：**实例知识的核心元素之一，是“概念”的实例化，每个实例都会对应到模式图中的某个概念。如“挖掘机TD20190709100020419”是一个实例（后面的编号为某台具体的挖掘机的机号），它对应的概念是“挖掘机”。下面提到的XX概念（除了“抽象概念”）都有其对应的XX实例，二者关系与“概念”和“实例”的关系相似，便不再赘述。
   12. **概念关系/概念关系边：** 知识模型的核心元素之一，包含一个谓语动词,且首尾各自连接模式图中的一个概念（名词），形成{主}-<谓>-{宾}句式。如“{工单}-<属于>-{作业形态}”就是一条概念关系边，其中“工单”和“作业形态”都是概念。
   13. **实例关系/实例关系边：**实例知识的核心元素之一，是“概念关系”的实例化，每个实例关系都会对应到模式图中的某个概念关系。如“{工单TW20160509175822577}-<属于>-{故障维修}”是一条实例关系边，其中“故障维修”是“作业形态”的实例。它对应的概念关系边是“{工单}-<属于>-{作业形态}”。
   14. **内部属性：**以Key-Value的形式存储在实例节点/概念节点/概念关系/实例关系中，用来进一步描述详细信息。如：实例“工单TW20160509175822577”，有内部属性{是否停机：否}、{故障发生时间：2019/6/19 11:18:03}等。
   15. **主键：**与关系数据库中的主键概念类似，是对储存的数据对象予以唯一和完整标识的内部属性。一个概念只能有一个主键，且主键不能为空值。
   16. **直接概念：**其主键可直接映射到关系数据库表的单个字段（列）的概念。
   17. **合成概念**：与直接概念相反，其主键不能直接映射到关系数据库表的单个字段的概念。其对应的合成实例的取值，需要关系数据库表中的多个字段甚至联立多表的多个字段，进行一定的逻辑运算而产生。如现场服务人员的“故障一次解决率”。
   18. **抽象概念：**不会被实例化的，仅存在于模式图中的概念。
   19. **静态关系：**较为稳定，一般不会产生变化的关系边。如：“{挖掘机}-<产生>-{工单}”，“{挖掘机}-<属于>-{机型}”。
   20. **动态关系：**更新较为频繁的关系边。如:“{挖掘机}-<属于>-{质保状态}”

1. 知识数据来源
   1. 概述

知识来源多种多样，从结构化角度来看包括结构化数据、半结构化数据和非结构化数据，从格式来看包括mysql,pg,csv,excel,pdf,word等数据，且未来有可能新增其他格式的数据。接下来将逐个介绍不同的数据源。

* 1. 维修服务
     1. 维修服务系统关系数据库数据

XX具有智能服务系统，该系统具有一个PG数据库，该数据库中包含上百张表，但我们真正可以用到的只有如下13张数据表，数据范围如下所示：

* 13张数据库表，共57653行。
* 其中包含9272条客户记录，28个服务人员，13989条设备信息，1832条工单记录等数据。
* 时间范围：2019.3.21-2019.9.20

下表详细描述了13张表的信息，共有三列，分别是表名、该表拥有的行数、该表拥有的列数，其中表名包含英文表名和中文含义。

|  |  |  |
| --- | --- | --- |
| **表名** | **行** | **列** |
| plt\_org\_user （用户） | 51 | 14 |
| plt\_intesrv\_tysrvperson （现场服务人员） | 28 | 19 |
| plt\_intesrv\_tycustomer （客户） | 9727 | 34 |
| plt\_intesrv\_tydevice （设备） | 13898 | 55 |
| plt\_intesrv\_tywork （工单） | 1832 | 93 |
| plt\_intesrv\_tyworkdispatch （派工单） | 1550 | 27 |
| plt\_intesrv\_typartdeliveryinfo （出库单） | 1618 | 46 |
| plt\_intesrv\_tylinkman （联系人） | 9767 | 13 |
| plt\_intesrv\_tyworkmileage （服务路途） | 1421 | 21 |
| plt\_intesrv\_typartdelivery （零件） | 6220 | 41 |
| plt\_intesrv\_tyjob （方案） | 9137 | 26 |
| plt\_intesrv\_tytodorepair （终端报警） | 2404 | 22 |

下图以“plt\_intesrv\_tylinkman （联系人）”为例展示该数据库中的表。



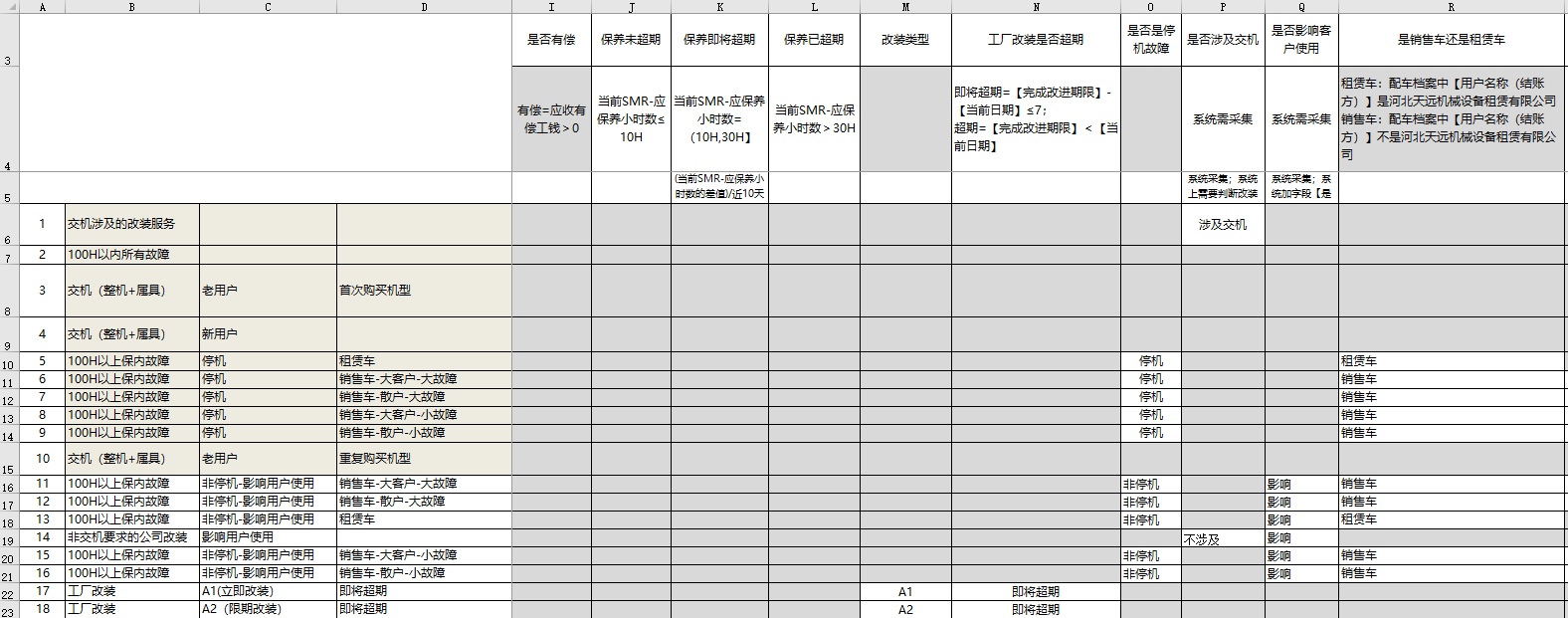
* + 1. 线下excel数据

XX的部分数据只存在于线下的excel表中，包括如下6张表：

|  |  |  |
| --- | --- | --- |
| **表名** | **行** | **列** |
| 190909\_人车绑定 | 82 | 7 |
| 服务人员培训记录 | 178 | 6 |
| 服务人员与区域对照表 | 69 | 4 |
| 服务人员驻点表 | 55 | 4 |
| 技术等级表 | 57 | 2 |
| 工单优先级 | 39 | 21 |

其中前五张表的结构和关系数据库类似，在此不再赘述。

工单优先级是XX的某个业务规则的表述，格式如下所示：



但工单优先级这种规则形式的知识的知识模型和导入过程还在开发中，尚未完成。

* 1. 故障诊断
     1. 概述

XX的维修人员需要使用小松挖掘机装修手册，是一个PDF版本的文件，共1246页，192MB。其中故障诊断主要包含四种模式：故障代码类故障、E类故障（电气系统）、H类故障（液压和机械系统）和S类故障（发动机系统）。其中，共有128个故障代码型故障，70个E/H型故障，18个S型故障。

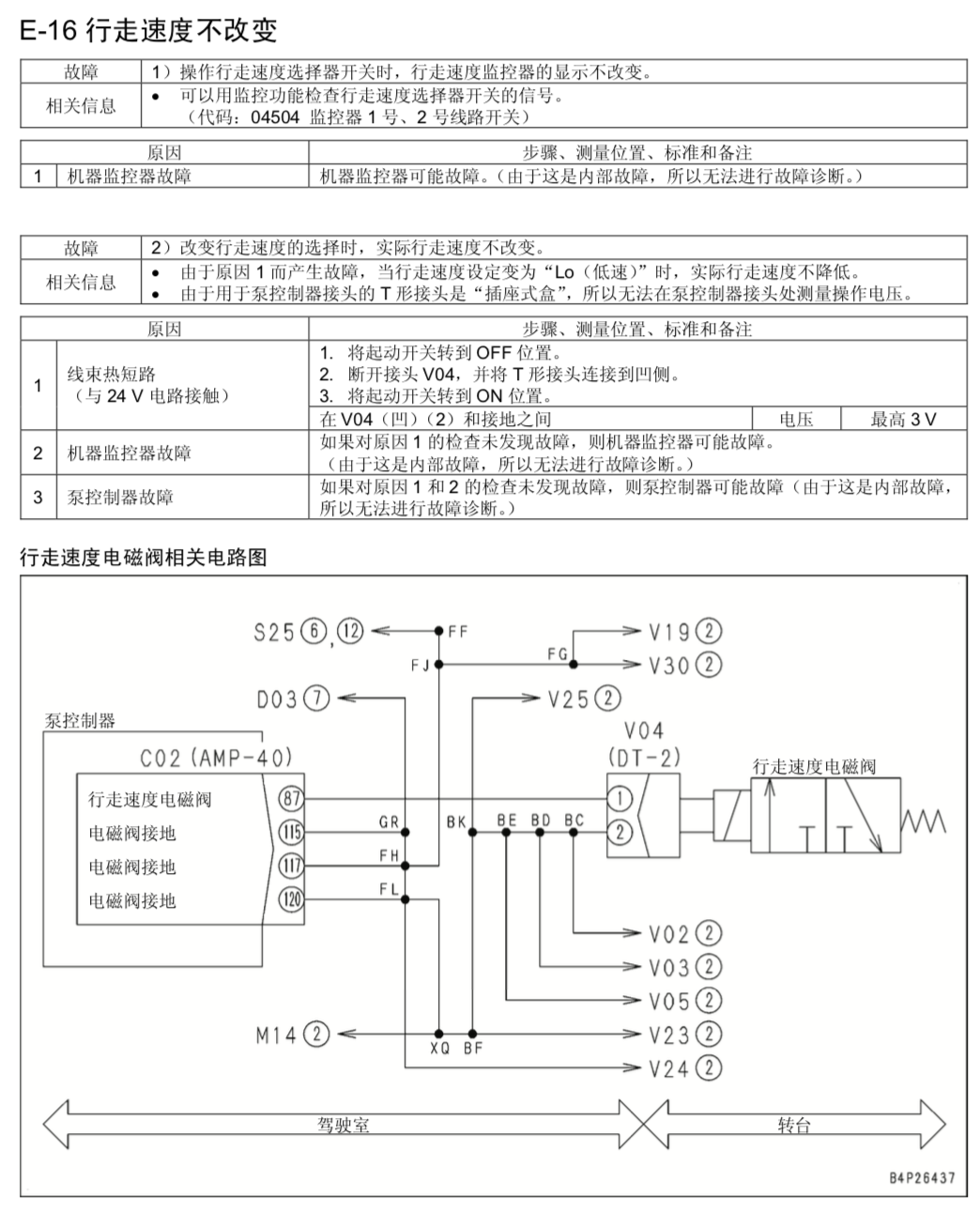
* + 1. 故障代码类故障

实例如下所示：



* + 1. E类故障（电气系统）

实例如下所示：



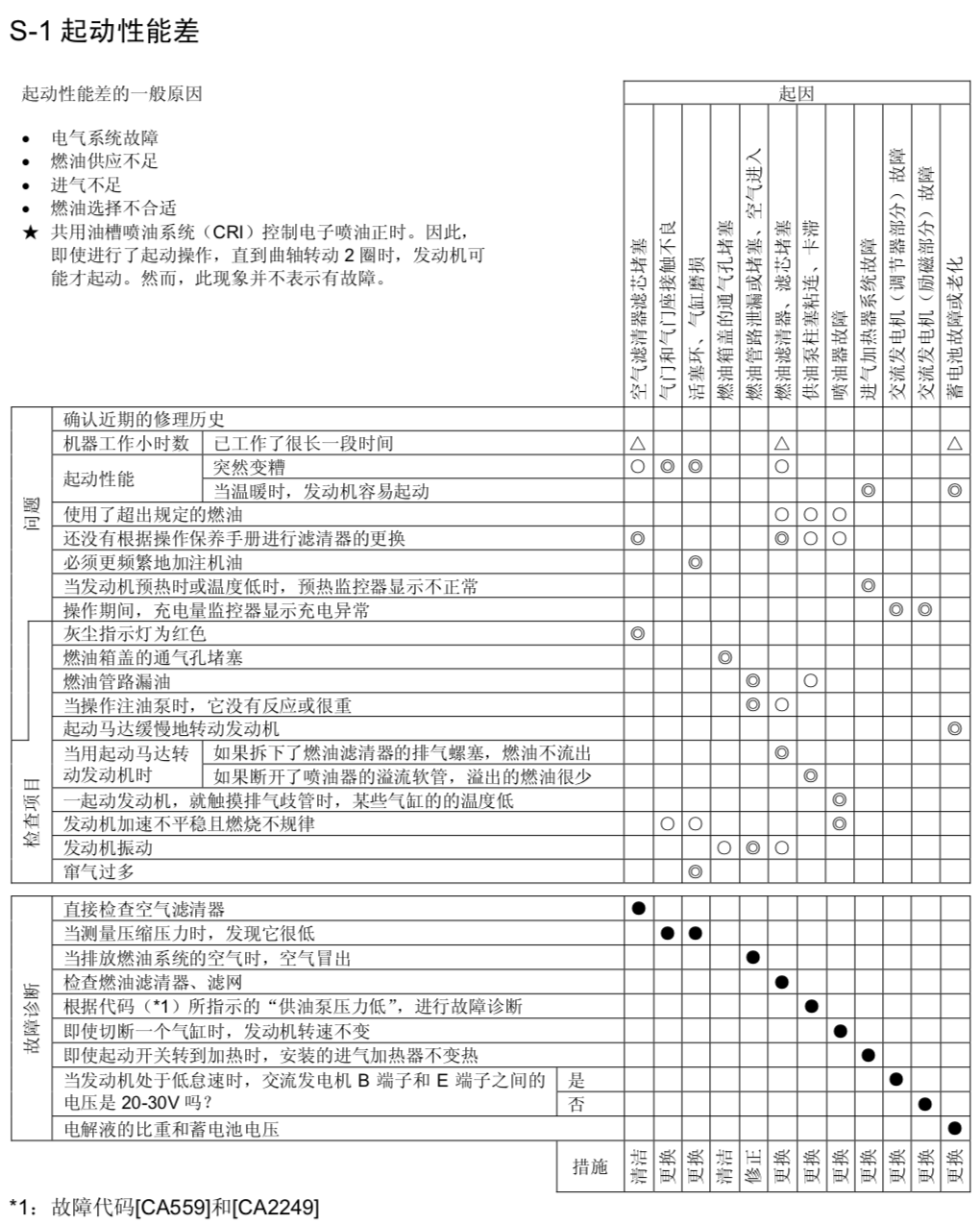
* + 1. H类故障（液压和机械系统）

实例如下所示：



* + 1. S类故障（发动机系统）

实例如下所示：



* 1. 产品结构

该数据源只包含一个excel文件，300行，示例如下所示：





其中，第一列是ID，第二列是当前部件，第三列是当前部件的父部件ID。

1. 知识图谱模式设计
   1. 知识图谱模型设计

在知识图谱模型设计中，具有概念节点和实例节点两类节点，具体含义见术语表。下面以工单示例如下所示：

配置表中的工单概念节点：

|  |  |
| --- | --- |
| **模式图中概念** | 工单 |
| **显示名** | 工单号 |
| **数据库表/属性** | plt\_intesrv\_TyWork.oid |
| **类型** | 直接 |
| **内部属性** | 工单号, 名称， 是否停机, 工单注册时间， 挖机的工作小时数， 故障发生时间， 工作开始时间， 工作结束时间 |
| **内部属性对应数据库属性** | id, workName， isShutDown, createTime, workHours， faultHappenTime， workStartTime，workFinishTime |

数据库中的工单实例节点：

|  |  |
| --- | --- |
| **plt\_oid** | 63E4FDFFE1DC5849999EF0171C25F56E |
| **plt\_id** | TW20190527161808607 |
| **plt\_workname** | 8000小时保养 |
| **plt\_isshutdown** | 1 |
| **plt\_createtime** | 2019-05-27 16:18:08 |
| **plt\_workhours** | 110 |
| **plt\_workstarttime** | 2019-05-27 16:18:09 |
| **plt\_workfinishtime** | 2019-05-30 13:42:34 |

* 1. 知识图谱模型存储结果

本工具在用Neo4j作为图数据库。在Neo4j中，会将概念节点们存为一个一个节点，并标记label为“概念节点”，同时属性中会记录该概念节点的相关配置信息，如所在知识域、所属数据库表、使用哪些列作为属性等，以工单为例，图数据库概念节点结果如下所示：



详细信息如下所示：

"labels": [

"工单",

"概念节点",

"维修服务"

],

"properties": {

"col": "plt\_oid",

"properties\_col\_ch": "['工单号', '名称', '是否停机', '工单注册时间', '挖机的工作小时数', '故障发生时间', '工作开始时间', '工作结束时间']",

"内部属性对应数据库属性": "id,workname，isshutdown,createtime,workhours，faulthappentime，workstarttime，workfinishtime",

"display\_name": "工单",

"type": "工单",

"properties\_col": "['plt\_id', 'plt\_workname', 'plt\_isshutdown', 'plt\_createtime', 'plt\_workhours', 'plt\_faulthappentime', 'plt\_workstarttime', 'plt\_workfinishtime']",

"逻辑代码连接": "nan",

"domain\_name": "维修服务",

"逻辑代码所需数据库表": "nan",

"内部属性": "工单号,名称，是否停机,工单注册时间，挖机的工作小时数，故障发生时间，工作开始时间，工作结束时间",

"compound\_table": "",

"模式图中概念": "工单",

"所属域": "维修服务",

"value": "工单",

"table": "plt\_intesrv\_tywork",

"数据库表/属性": "plt\_intesrv\_tywork.oid",

"显示名": "工单号",

"类型": "直接"

}

在Neo4j中，会将实例节点们存为一个一个节点，并标记label为“实例节点”，同时该节点的属性包括基本属性和配置属性，基本属性包括domain,display\_name等，配置属性则是根据配置表中指定哪些列作为属性。以工单为例，图数据库实例节点结果如下所示：



详细信息如下所示：

"labels": [

"实例节点",

"工单",

"维修服务"

],

"properties": {

"工作结束时间": "2019-05-30 13:42:34.000000",

"domain\_name": "维修服务",

"工作开始时间": "2019-05-27 16:18:08.699912",

"工单注册时间": "2019-05-27 16:18:08.000000",

"名称": "8000小时保养",

"故障发生时间": "2019-06-19 11:18:02.980667",

"type": "工单",

"是否停机": 1,

"display\_name": "TW20190527161808607",

"工单号": "TW20190527161808607",

"value": "63E4FDFFE1DC5849999EF0171C25F56E",

"挖机的工作小时数": 110

}

1. 系统开发说明
   1. 概述

本系统包括两部分，一部分是将结构化的关系型数据按照配置文件导入至Neo4j中，另一部分是将非结构化的数据转为结构化数据，以便使用第一部分的程序将非结构化数据导入至Neo4j中。两部分的关系如下所示：



* 1. 开发环境

部署图数据库所需软件：neo4j-community-3.5.9

开发语言：Python3.7

需使用库函数：pandas，numpy，tqdm，xlrd，py2neo

建议使用Anaconda，可较快安装上述库，且建议使用Jupyter Notebook进行编辑（也是当前代码格式），方便快速debug以保证结果正确性。

* 1. 将半结构化数据转为结构化数据
     1. 概述

XX提供的数据中包含半结构化数据，不方便直接在该数据上做很多操作。为了方便数据的进一步处理，需要将半结构化数据提取为结构化数据。

* + 1. 输入

包含半结构化数据的装修手册PDF文件。

* + 1. 输出

包含结构化数据的csv/excel文件。

* + 1. 处理逻辑

逻辑如下图所示：

详细逻辑如下所示：

1. 将PDF手动转为Word,再手动复制粘贴所需部分至excel中
2. 读取excel，分割故障，得到每个故障对应的行范围
   1. 读取每行的第一列的单元值
   2. 用正则表达式匹配看是否是故障标题
   3. 如果是的话就记下来
3. 逐个读取出每个实体/属性的值
   1. 读取每行的第一列的单元值，如果是实体/属性关键词的话，依据以下规则读出其内容
      1. 碰到“报警级别”
         1. 用其下方单元格作为值
         2. 向右读，找“故障代码”和“故障”
            1. 分别读出对应值
            2. 和故障标题得到的两个值进行对比

如果一样则得到该值

如果不一样则输出Warning，并以故障标题中的为准

* + 1. 碰到“故障内容”或“控制器的反应”或“机器发生的故障”
       1. 向右读出其值，但右方可能有多行值
          1. 根据黑线判断截止行数
          2. 判断右方第一列是否为特殊符号或空，若是的话则取右二为值，否则为右一
    2. 碰到“起因”
       1. 向下读出其值，但下方可能有多个“起因”，同时每个“起因”有多行
          1. 根据黑线判断多个“起因 ”之间的间隔
          2. 逐个读取每个“起因”的多行，合并字符串作为“诊断操作”
  1. 存为字典格式

1. 将故障值从字典形式转化为关系表形式
   1. 将具有关系的实体放在同一行
      1. 将标量值直接和主键关联
      2. 将列表值逐个取出，并和与之关联的实体放在同一行
   2. 存为csv
      1. 代码结构

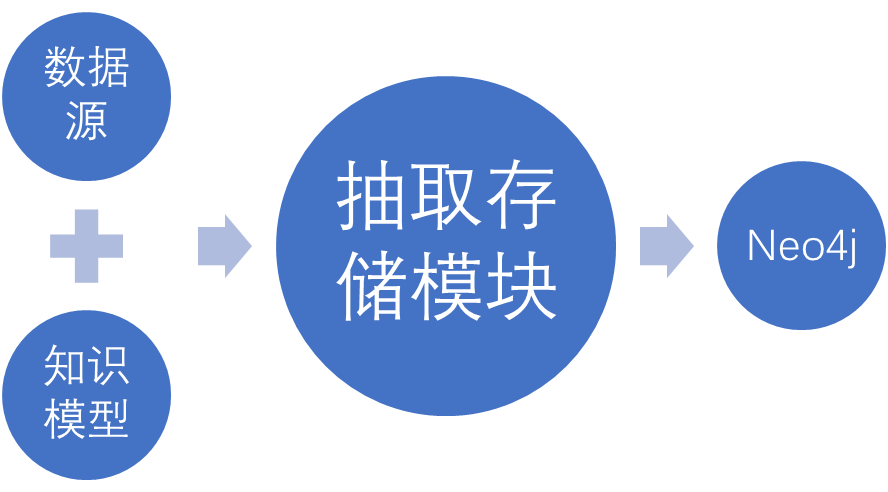
readManual.ipynb读取excel格式的维修手册并从中读取中故障代码类故障并将结果存为csv格式。

readManual-E+H.ipynb读取excel格式的维修手册并从中读取中E和H类故障并将结果存为csv格式。

readManual-S.ipynb读取excel格式的维修手册并从中读取中S类故障并将结果存为csv格式。

* 1. 将结构化数据导入neo4j中
     1. 概述

该模块将数据源和知识模型输入抽取存储模块中，该模块根据知识模型将数据源抽取并存储至Neo4j中，结构如下图所示：



* + 1. 输入

1. 知识图谱抽取模型配置表
2. 源数据（关系型数据格式）
   * 1. 输出

将源数据按照配置表的描述导入至Neo4j图数据库中。

* + 1. 增量抽取/全量抽取

全量抽取表示当前数据库为空且将所有数据导入，增量抽取表示当前数据库已有上次数据只需要更新最近的数据。

实现不同的抽取模式可以通过修改数据源实现，全量抽取即数据源为所有数据，增量抽取即数据源仅为最近更新的数据。

不过由于该程序具有自动去重功能，所以连续多次给全量数据源也不会重复导入数据，也就是实现了数据源每次都是所有数据但依然实现了增量抽取的效果。但同时由于需要将前面的所有数据去图数据库中搜索是否已存在，所以速度会较慢。各个模式对比如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模式 | 之前数据库状态 | 数据源 | 效果 | 速度 |
| 全量抽取 | 空 | 所有数据 | 全量抽取 | 慢 |
| Lazy增量抽取 | 已有数据 | 所有数据（包括更新数据） | 增量抽取 | 慢 |
| Fast增量抽取 | 已有数据 | 仅更新数据 | 增量抽取 | 快 |

* + 1. 读取结构化数据

本系统支持读取多种输入，包括如下：

* Mysql数据库
* pg数据库
* csv
* excel

不过由于本地读取csv最快，而读取远程数据库时间消耗较长，为了debug方便，一般会先将其他种类数据源转为csv格式。

readMySQL.ipynb可读取Mysql/pg/excel格式的数据并转换为csv格式。

* + 1. 读取配置表

逐个读取Excel格式的配置表中的三个sheet，逐行读入。

* + 1. 导入Neo4j

算法大致流程如下所示：（下图中，从左至右是顺序关系，从上至下是包含关系）

算法详细过程如下所示：

1. 连接数据库
2. 导入模式图
   1. 生成概念节点
      1. 添加基本内容，包括labels。
      2. 添加属性，包括概念表中的各列内容
   2. 生成概念节点间的关系
      1. 取出主语实体和宾语实体
      2. 增加属性
   3. 添加至Neo4j中
3. 导入数据图
   1. 对于每个数据库表/csv表
      1. 获得该表相关的表
         1. 获得属于该表的实体数据表
         2. 从概念表中抽取属于该表对应的概念集合Concept\_In\_this\_table
         3. 从关系表中抽取属于该表对应的关系集合Relationship\_in\_this\_table
      2. 对于该实体数据表的每行
         1. 生成该行的所有实体，对于每一个实体
            1. 检测是否为空，若不为空，进入下一步骤
            2. 检测该节点是否已存在

若存在，将该节点加入到待处理节点列表list中

若不存在，创建节点

初始化该节点

添加该节点的基本内容，包括labels、value等内容

添加属于该节点的属性值

读取属性值内容

将key\_value对添加至节点属性中

添加display\_name属性

根据中文列名找到对应数据库列名

根据数据库列名将该列内容写入display\_name中

将display\_name添加至节点属性中

将该节点加入到待处理节点列表list中

* + - 1. 根据关系集合Relationship\_in\_this\_table，生成该行实体间的所有关系
         1. 对于每个关系

取出主语实体

若已经指定主语列，且已被记录，则直接取出该实体

若实体已存在，则直接取出该实体

若实体不存在，则认为是外键，去图中寻找该实体

根据主语概念取出主语列，如果已被记录，则取出该实体

如果不属于属于本表的待处理节点列表list中，则说明该键是外键

1. 取出主语概念
2. 如果已被记录，则取出该实体
3. 如果不属于属于本表的待处理节点列表list中，则说明该键是外键
   1. 根据外键中文名从概念表中寻找主键的数据库表和列名
   2. 根据主键的数据库表和列名以及当前数据库表得到外键列名
   3. 根据外键列名读取该外键实体内容
   4. 根据外键实体内容在Neo4j中查找该节点
   5. 如果实体存在，记录下来该实体，否则不生成本关系（理论上外键实体一定有主键的，但是测试数据很坑，存在假外键/外键为空的情况）
4. 取出宾语实体
   1. 同主语实体
5. 建立关系
   1. 将记录的主宾实体建立关系
   2. 增加关系的属性，比如关系ID
6. 将本行生成的实体和关系添加至Neo4j中
   * 1. 代码结构

由于mysql/pg中的数据读取较慢和本地的csv文件读取较快，为了本地debug方便，会利用readMySQL.ipynb从Mysql/pg中读取数据并存入至本地csv。

transferDataToNeo4j-DataProvenance-v2.5.ipynb是Neo4j导入程序，运行时需要制定config文件路径和数据源路径。

* 1. 处理流程
     1. 维修服务

只需编写好知识模型配置文件，准备好数据源文件，即可使用导入程序将知识导入至Neo4j中。

* + 1. 故障诊断

需要先人工将装修手册PDF转为Word，再手动复制所需部分至excel中。再利用4.3中所描述程序将Excel文件中的半结构化数据转化为对应的csv格式的结构化数据，然后再将知识模型配置文件和结构化数据输入至导入程序中，将知识导入至Neo4j中。

* + 1. 产品结构

同4.5.1。

1. 知识库结果（展示当前数据库中的已有结果）
   1. 概述

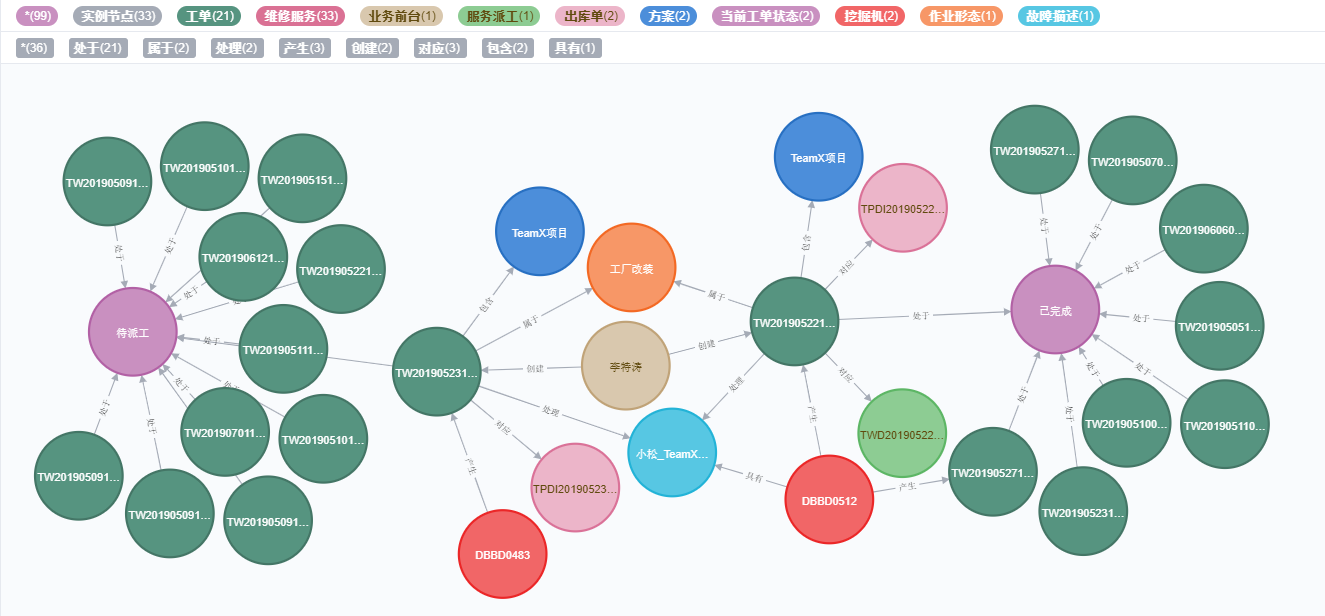
知识库中当前具有68315个节点，124862条边，详情见下表：

|  |  |  |
| --- | --- | --- |
| **知识域** | **节点数** | **边数** |
| 所有节点 | 68315 | 124862 |
| 概念节点 | 71 | 92 |
| 实例节点 | 68244 | 124770 |
| "维修服务" | 65431 | 120404 |
| "故障诊断" | 1814 | 3315 |
| "产品结构" | 1058 | 1143 |

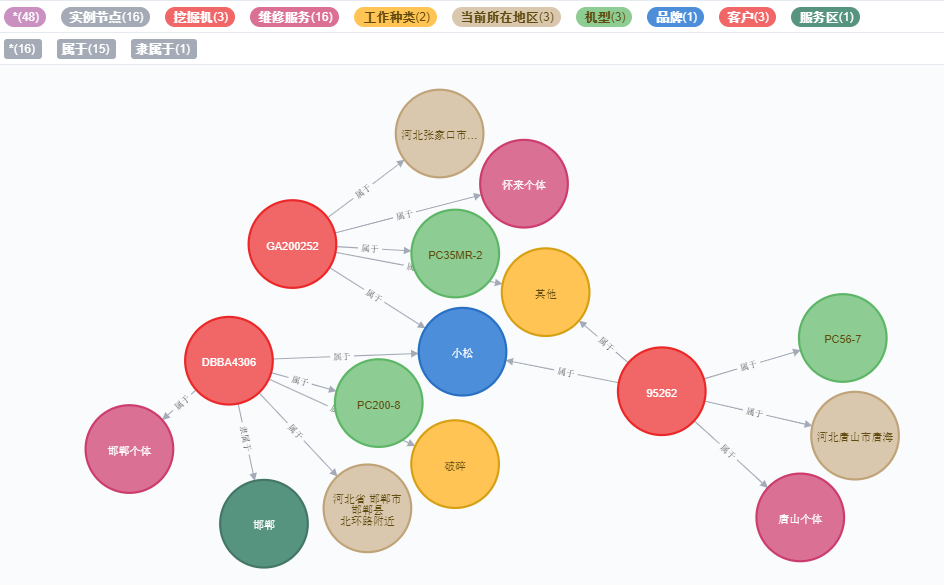
* 1. 维修服务

目前Neo4j中共49816个节点，92974条关系。每个概念及其数量可见下图：

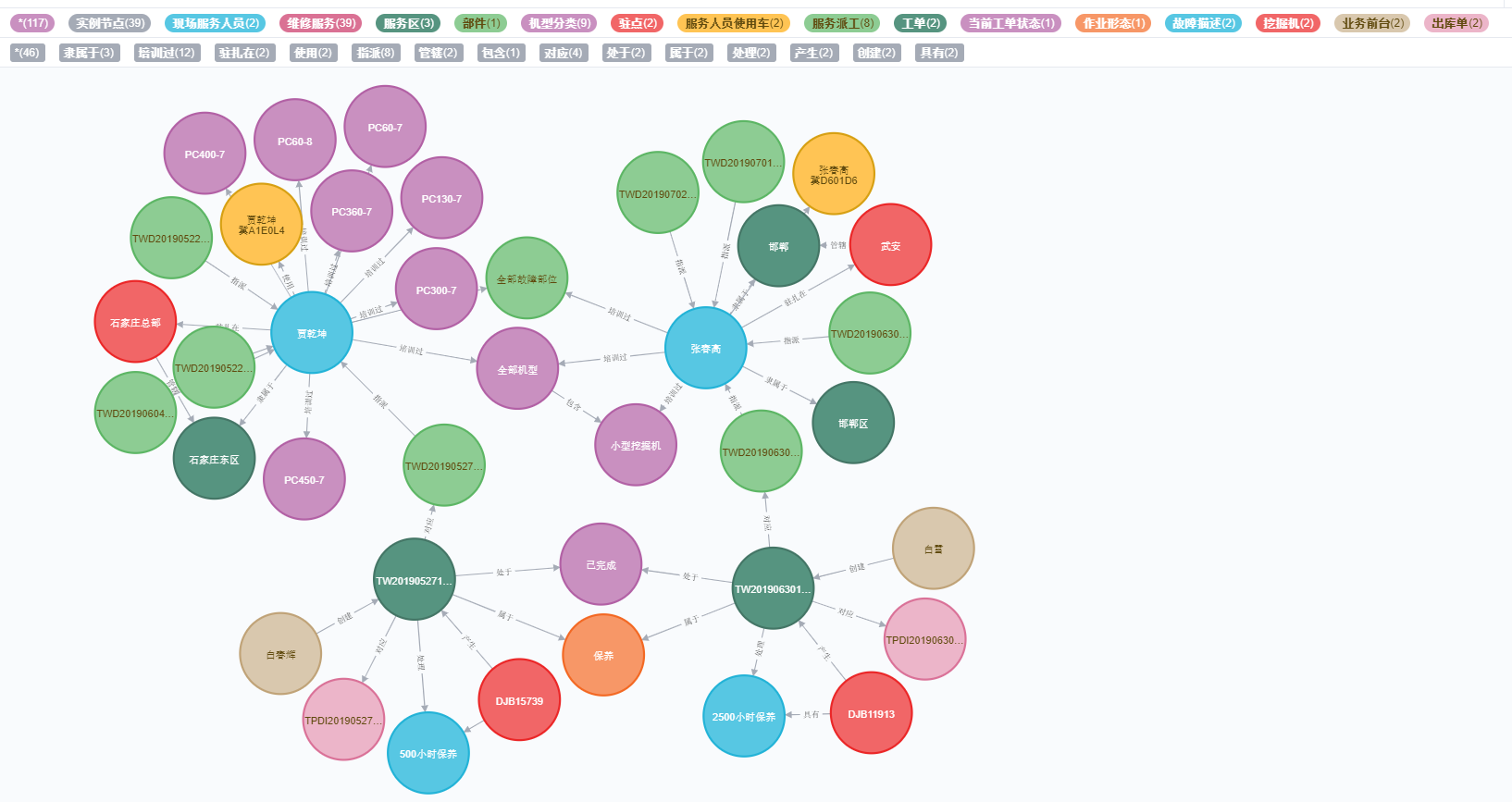
工单相关知识图谱示例如下：



挖掘机相关知识图谱示例如下：



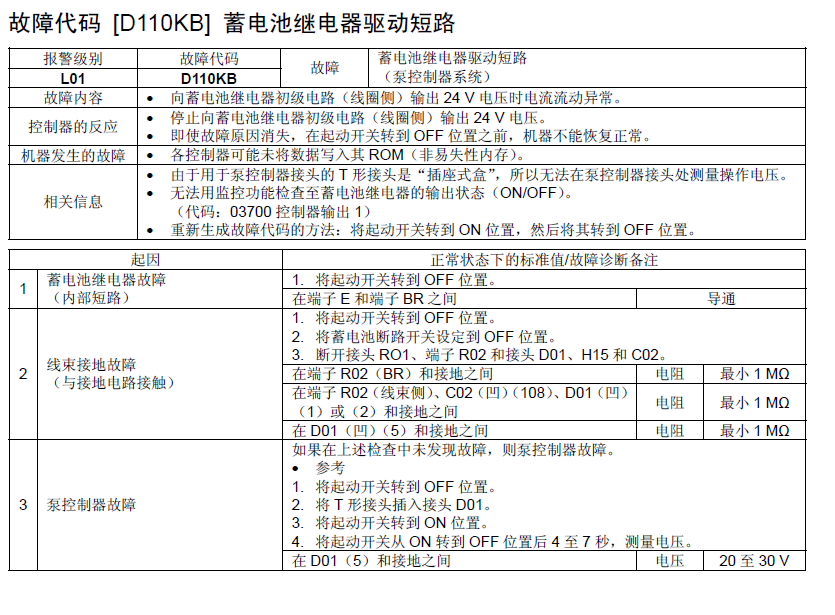
服务人员相关知识图谱示例如下：

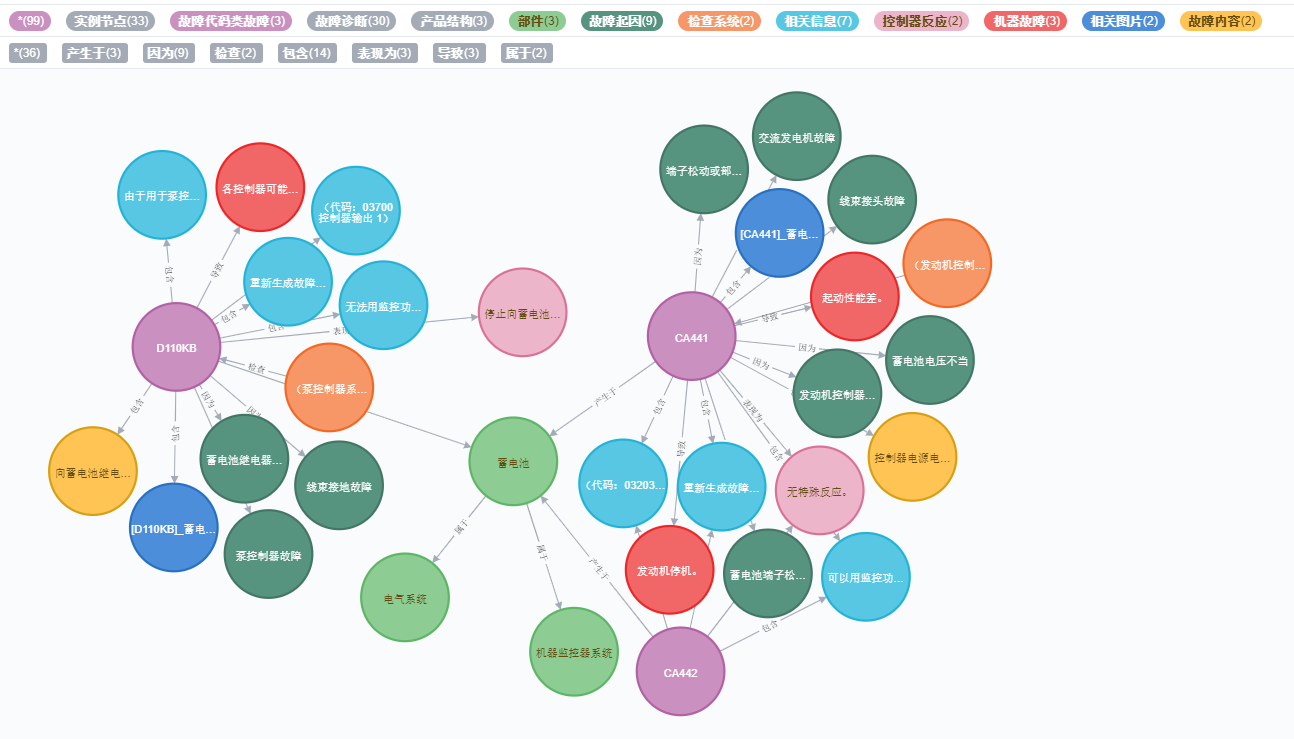


* 1. 故障诊断

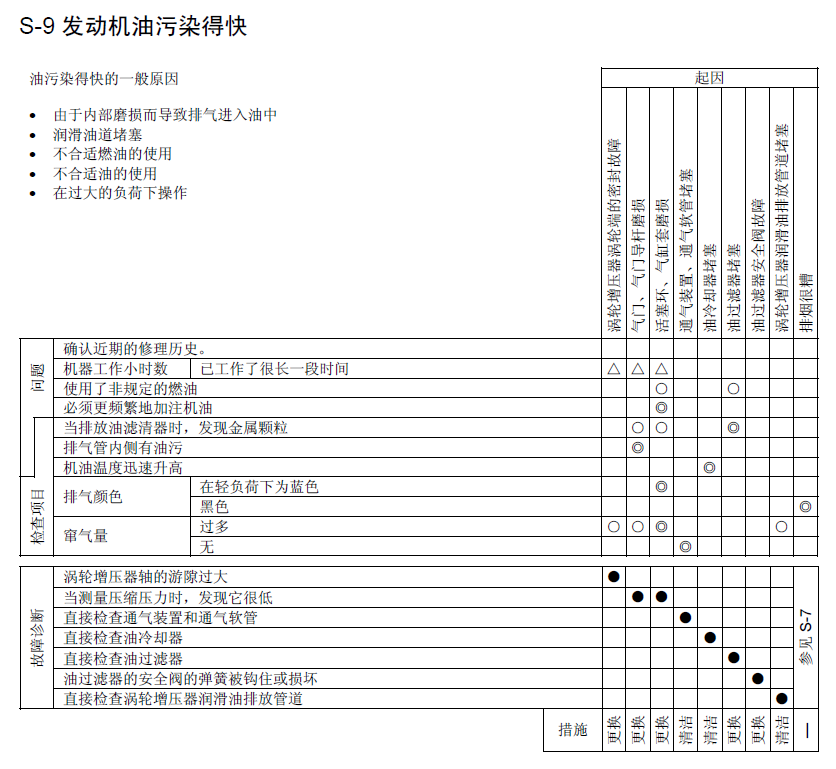
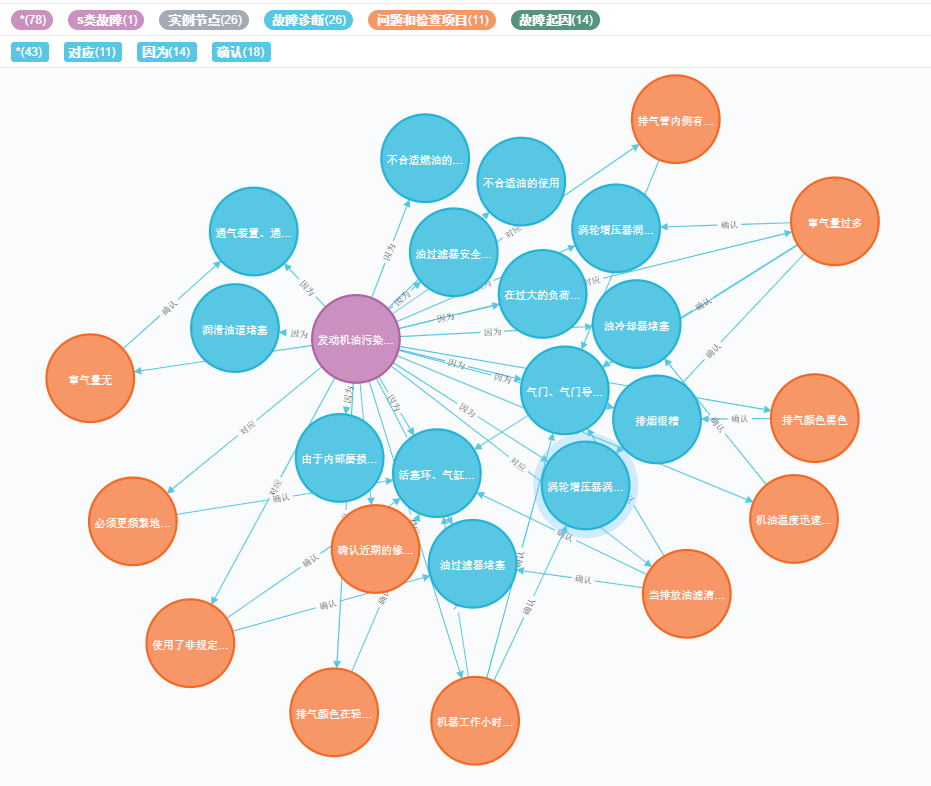
共1830个节点，3313条关系。每个概念及其数量可见下图：

故障代码类故障的源数据及其Neo4j示例如下所示：





S型故障的源数据及其Neo4j示例如下所示：

* 1. 产品结构

共1058个节点，1143条关系。

源数据及Neo4j结果如下所示：