1.区域拓扑结构设计

1.1. 信息来源。

1. KKS：土建
2. 定位：机构->地图->地图区域
3. IDC：
   1. 细致一点：省份->数据中心->大楼->楼层->机房->子机房
   2. 粗略一点：数据中心->机房

设备部分：->设备组->设备（/机柜->机柜内设备）

1.2.参考KKS的土建信息建立基本的拓扑树，在后台编辑，三维中显示。

建筑物(根节点)

-主机装置建筑物 （建筑分组）

主机房 （建筑/大楼）

主机房0层 （楼层）

#1汽机房区域 （区域）

...配电间 （机房）

主机房4.5m层

...

主机房11.0m层

#1燃气轮机房

-常规燃料供应建筑物

调压站控制室

-仪表和控制建筑物

集控楼

联合车间

-常规产热建筑物

#1锅炉房

-行政生活建筑物

行政楼

食堂

公寓

-定位区域1 （定位区域）

定位区域11 （定位区域）

定位区域22 （定位区域）

1.3.拓扑节点要关联KKS码

ID,KKS,Type（关联信息类型：区域节点、机房、设备、部件（子设备））

1.4拓扑节点信息:

ID,Name,PID,Number,Type（节点类型：区域、分组、大楼、楼层、机房）,Describe,Tag（分类用标签）

有空的话设计一下拓扑节点编辑界面，供Web后台后续界面优化使用。

1.5拓扑节点界面相关

1.界面上按名称查找节点。

2.三维内定位位置（单击节点/按钮/菜单项）

定位建筑时可以显示建筑的介绍信息：面积，高度，层数，图纸，照片。

这个信息也可以是在建筑物上三维中漂浮一个按钮，或者在节点界面中添加按钮/菜单项，点击按钮后出现信息界面。

3.打开子场景（双击节点/按钮/菜单项）。

1.打开子定位区域场景

2.打开机房管理场景

4.编辑节点对应地图区域的大小（菜单项/按钮）。

超级管理员(admin/root)或者运维人员才有编辑的权限，显示编辑开关按钮。

5.不同节点类型用不同的图标显示。

6.节点可以收缩、展开。

1.6区域节点(TopoNode)和地图区域(MapArea)的关系

一个区域节点可以管理一个或者0个三维地图区域

区域节点在后台增删改。

地图区域在三维中手动拖拽编辑。

同时三维中需要一个数值的编辑界面，以便能够精准编辑。

名称

中心坐标 X,Y,Z

旋转角度 X,Y,Z

缩放比例 X,Y,Z

这个编辑界面同时可用于设备的编辑。

1.7 关于编辑开关，是统一的一个编辑开关呢？还是选中一个物体/区域，该物体/区域上有个编辑按钮，点击编辑后开始编辑。

1.8 现在一个拓扑树有两种展示方式，物理拓扑树、定位拓扑树。

物理拓扑树：基于KKS码和实际物理位置创建的树。

定位拓扑树：由定位区域组成的树。

定位拓扑树是基于物理拓扑树的。

物理拓扑树上的节点，可以在三维区域中有一个对应的地图区域，也可以没有。

定位拓扑树就是展示出物理拓扑树的有三维区域的部分，没有关联地图区域的节点不用展示。

在展示定位拓扑树时还要把每个定位区域下的人员数量显示出来。父区域的人员数量包括子区域的人员以及子区域外父区域内的人员。

是否只显示有人员的区域，作为一个可以切换的配置项处理吧，默认是都显示的。

两种树在项目上线后应该不会发生改变，但是在上线前会有相应的调整。

1.9 物理拓扑树上的所有节点都能添加设备。

机房和区域实际上没区别。

机房可以打开独立的场景，区域可以可以打开独立的场景。

添加的设备的位置信息是相对于该物理节点的空间区域的坐标系的。

1.10 关于打开独立场景，给节点配置关联的模型。

存在关联模型的节点打开独立场景时显示该模型。

没有关联模型的打开独立场景时显示空的场景，或者只有一个空的地板，可以在地板上编辑机房，添加设备。

1.11 物理拓扑树上要显示相应的子设备，变成定位拓扑树后，则显示相应的子人员。

1.12 拓扑树工作分工：

1.楼建峰：数据结构、表结构、网站后台、WCF接口

2.刘嘉鹏：界面设计、交互设计

3.蔡旭璐：界面搭建：拓扑视图界面、建筑信息界面、区域编辑界面（优化）

4.王锴：建筑定位

5.叶宗雷：区域编辑功能，区域编辑界面（简单）

**2.人员拓扑树设计**

**2.1.区域人员拓扑树**

物理拓扑树通过进入定位模式变成定位拓扑树，在定位区域中展示定位的人员，这个是动态的，会相应的变化。

要实时（1s/2s/5s）获取人员所在区域信息，或者说是区域包括哪些人员的信息。

**2.2.组织人员拓扑树**

节点类型：

简单：部门->人员

复杂一点：部门->班组->人员

最复杂：机构(/公司)->机构(/公司)->部门->部门->班组->人员

文件夹节点（组织架构/所属）：机构->机构->部门->部门->班组

文件节点：人员

用最复杂的方式做，具体显示时可以创建的层级少一点。

Web后台编辑，3D前台展示。

编辑部分一开始不需要专门限制文件夹节点类型的层级关系。

**2.3.人员搜索界面**

人员查找界面：姓名，所属（机构->部门），工作（角色/职务/岗位），（定位）标签，当前位置（区域），位置（区域）路径，状态。

人员角色：职工、外委人员、来访人员

操作：定位人员、显示人员详细信息。

**搜索的结果同时在三维中高亮出来，搜索完后还能最小化成只有人员名称的简易界面，以便切换到不同的人员。**

**2.4人员信息界面（三维中飘在人员头上的）**

1.姓名

2.姓名，工号，标签号

3.照片（图片框），姓名，工号，标签号。

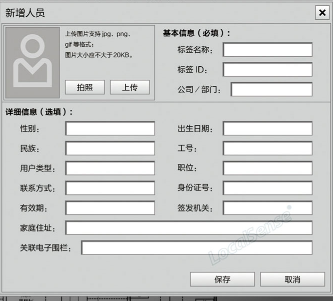
4.照片（图片框），姓名，工号，标签号，电话，邮箱。

做成摄像头拉近后显示比较详细信息的方式，从比较详细的信息中还能打开最详细的信息界面。

信息界面中可以给人员发送短信或者邮箱告警。

（短信和邮箱的发送功能是否支持要看电厂那边是否支持外网连接功能，这个有待确认。）

详细信息界面：



基本信息：

**姓名**、性别、照片、出生日期，民族，住址

定位信息：**标签编号，标签名称**

工作信息：**所属**（公司->部门，显示上为面包屑导航的方式，下拉编辑为机构树），**工号**、**职位**（下拉框选择），**电话**（短信提醒用）、**邮箱**（邮箱提醒用）。

通过所属信息区分出人员类型是电厂职工和外委人员以及访客。

Web后台编辑，3D前台展示。

3.区域权限

3.1 基本权限类型

1. 不能进入。
2. 不能出去。
3. 不能呆太久。
4. 不能出去太久。
5. 不能靠近边界。（围栏功能）

3.2 使用场景

1. 告警围栏

不具有权限的人都不能进入，进入则发送告警。

具体场景：制氢站不能进去，废水处理间不能进去，危险化学物品房间不能进去，发电机或者某一个大型设备范围不能靠近。

告警围栏也可能分等级。

根据靠近的程度，可以先发送轻微提示；具体靠近到一定程度后，发送警告，要求离开。

1. 工作区域划定

在一定时间段中必须在某一个区域内；或者多个区域内，区域之间移动时可以有一定的离开时间。离开该区域（超过误差距离，超过误差时间）时，提示告警。

具体：不能离开工作场所太久；执行检修某些设备时必须在设备所在的区域。

1. 限制区域进入时间

从进入开始，在某一个时间范围内不离开该区域，则告警。

具体：休息区、吸烟区等不能停留太久。

1. 巡检人员监控

人员必须在某一个时间范围到达一个地点，在一个时间范围内不能从一个区域出去。进去后就不能出去。

创建巡检路线区域，巡检区域告警：在某一个时间内，巡检任务时间，离开了巡检区域。

巡检路线关键点告警：没有在一个时间点+误差范围内，到达某一个关键点。

1. 危险物资（人物，如监狱环境下）监视

从进入开始，永远不能（直到移除了这个规则）离开某一个区域，离开便告警。

有些场景可以不用发告警，但是要记录下可疑行为，做后续的数据分析。

对于定位引导这种功能，和巡检有点像，从一个点到下一个点，指引人移动，感觉需要移动端App的支持；告警也是，短信、邮箱、电话以外直接发送App消息推送也是一个方式。（不用在意这个，随便想想的）

3.3 创建权限（AreaAuthority,区域权限）

1. 规则组（AccessRuleGroup）

整理规则以及扩展用，可以一开始有一个默认的组，所有的新增规则默认都数据该组。

属性:Id,Name,Describe,Pid(父的组，默认为空),Rules(规则列表)

一个规则只能属于某一个规则组。

界面操作相关：可以在创建权限时选择一个组，就添加了组内的所有规则。

1. 进出规则（AccessRule）

设计思路：

时间段

开始时间(StartTime)、结束时间(EndTime)

时间长度(TimeSpan)

时间长度，进入时间作为开始时间，计算出开始结束，超过结束时间则告警。

时间重复

是否重复，一次性的，每天，每周（周一.周二.周三.周四）

时间误差范围

开始时间-开始误差时间(StartErrorRange)

结束时间+结束误差时间(EndErrorRange)

进出限制类型：

不能进入，

不能出去，

可以进入，

可以出去：在某一个时间范围内可以出去，除此之外，必须在区域内。

不能呆太久：在某一个时间范围内可以进入，除此之外不能进入。

属性信息:

基本：

Id,Name,Describe,

时间：

StartTime,EndTime,TimeSpan,StartErrorRange,EndErrorRange

重复：

AccessType:NoEnter,NoLeave,CanEnter,CanLeave,CanStay,NoStay,

RepeatType:Day（每天）,Once（只一次）,WorkDay（周一到周五），Custom（自定义）,Temporary(临时的)

CustomDay:自定义的星期几，具体保存格式待定：1\_2\_3\_4\_5\_6\_7，0:0:0:0:0:0:0（某一个位置是1，代表有这一天），0000000。

其他:

**Areas:关联区域列表，列表结构**。区域和规则是多对多的关系，不过操作上是创建规则时选择区域。

Enable:是否激活

Tags:标签，分类用

GroupID:规则组，整理用

记录：

CreateTime:创建时间

ModifyTime:修改时间

DeleteTime:删除时间

1. 权限组（AreaAuthorityGroup）

定义：多个不同的权限的集合，分组用。

一个权限只能是一个分组，默认一个权限组。

属性:Id,Name,Describe,Pid(父的组，默认为空),Authoritys(权限列表)

界面操作相关：

1. 绑定：绑定权限时可以选择一个权限组就绑定了组内的全部权限。
2. 自动生成描述：最好能够根据规则的内容自动生成一段描述，如“根据规则1，可以进入区域1；根据规则2，不能进入区域2”。
3. 权限（AreaAuthority）

定义：多个不同的区域进出规则的集合。

属性:Id,Name,Describe,GroupID(组),Authoritys(权限列表)

一个权限可以属于多个不同的规则，一个规则可以属于不同的权限。

最终用于绑定的是权限。

如：

权限1：规则1(可以进入区域1)+规则2(不能进入区域2)+规则3(不能离开区域3)

权限2：规则1(可以进入区域1)+规则4(可以进入区域2)

判断时，根据权限获取规则列表，判断规则，发送告警。

可能问题1：不同规则间相互矛盾，如：

规则A可以进入区域A，规则B不能进入区域A。

处理方案1：

1. 智能判断并限制用户选择。最理想是能够判断出来相互矛盾的规则并指出来，不让同时包含在一个权限中。规则中有个时间段的概念在里面，要判断的话要在同一个时间段中，对于同一个区域，有相互矛盾的规定。时间段可能重叠，区域也可能重叠，比较复杂。
2. **不用管**。因为告警的判断是相互独立的，可以在互相不影响的情况下发送自己规则的告警。

3.4 绑定权限

1. 权限是绑定到人上还是绑定到卡上？

与人相关的可以是角色、岗位、部门等，与卡相关的是卡组。绑定到卡上，将卡发给某个人，某个人就具有了该卡的权限。卡上要标示出其编号，卡组，作用。绑定到人上，将卡发给某个人，该卡就具有了该人的权限。卡仅仅用于定位用，权限无关。

绑定到人员上的话，相关的东西比较多，设计起来比较复杂，另外应该说最终绑到人上，还是到卡上都能实现相同的效果，哪个简单哪个来。

**决定绑定到卡上吧**。

2. 直接绑定到卡上，还是卡组上(TagGroup)，或者是卡角色上(TagRole)？

如何实现：一次将一个权限赋予一批人（一个班组：1号路线巡逻组）？

下面的功能相当于对卡管理的扩充。

卡角色(TagRole)，

卡组(TagGroup)，一个卡可以属于多个卡组，因为不同卡组不同权限，如果

一般场景：

特殊场景：只有一定部门/职务（以上）的人，才能进入某一个区域的话，部门也要能够绑定权限了。

3.5 触发告警

3.6 告警处理