上游追踪是从一个结点开始按流向相反的方向，根据一定的条件搜索此结点能够达到的网络集合。例如，水域某处发生污染，通过上游追踪分析可以寻找污染物可能来自哪些水域。本次以该实际情况为背景，利用iServer restapi以及iClient工具实现上游追踪功能。

**总体步骤**：

1. 利用iDesktop桌面软件制作水域网络数据集。
2. iServer发布网络分析服务。
3. 通过restapi+iClient for Classic工具实现上游追踪功能。

**数据及服务准备：**

1.数据制作

(1)制作水域网络数据集及设置流向(一般来说只用设置一个源点和一个汇点)，并制作地图，具体方式可参考iDesktop .net帮助文档

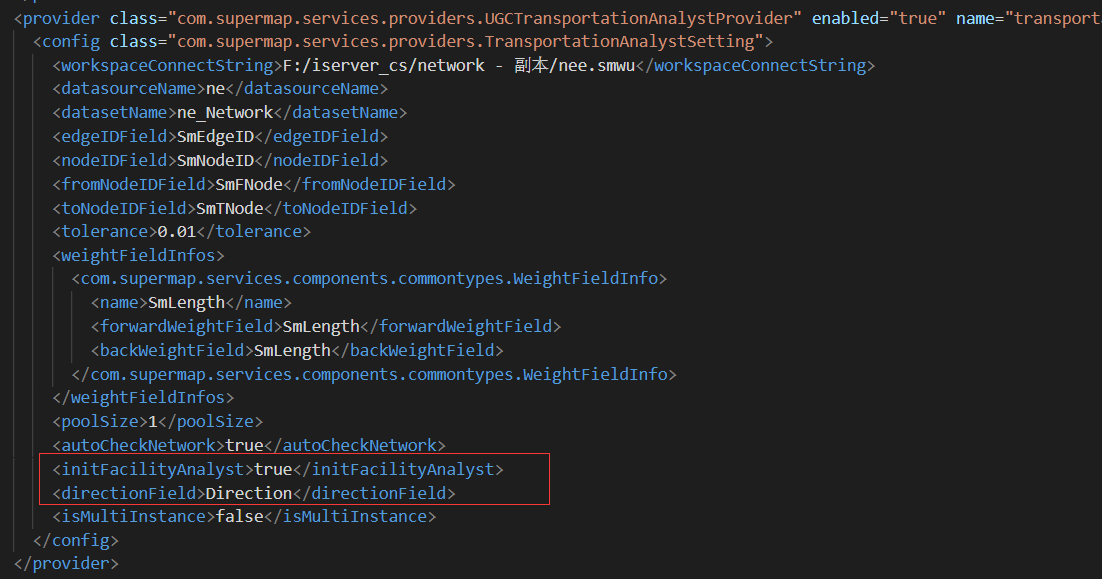
<http://support.supermap.com.cn/DataWarehouse/WebDocHelp/SuperMap-iDesktop-Net3/index.html>

2.发布服务

(1)发布网络分析服务以及地图服务具体方式可参考前文博客，以及帮助文档

<http://support.supermap.com.cn/DataWarehouse/WebDocHelp/iServer/index.htm>

(2)同时在webapps\iserver\WEB-INF\iserver-services.xml文件中找到对应服务提供者节点，并修改如下图所示节点：



**开发准备**：

上游追踪分析呈现与实现，需要rest-api以及SuperMap iClient for Classic

1. rest-api具体可参考iServer在线帮助文档地址：

<http://support.supermap.com.cn/DataWarehouse/WebDocHelp/iServer/index.htm>

1. SuperMap iClient for Classic产品开发指南 地址：

<http://iclient.supermap.io/web/introduction/classicDevelop.html>

**基于iClient for Classic实现上游追踪功能：**

为代码简洁实现利用CDN在线引入Classic文件，进行单网页的开发。

具体步骤如下：

1.引入js库

<script src="http://iclient.supermap.io/web/libs/iclient8c/libs/SuperMap.Include.js"></script>

<script src="http://iclient.supermap.io/dist/classic/iclient-classic.js"></script>

<script src="https://apps.bdimg.com/libs/jquery/2.1.4/jquery.min.js"></script>

2.加载地图

1. //初始化地图
2. map = new SuperMap.Map("map", {
3. controls: [
4. new SuperMap.Control.Navigation(),
5. new SuperMap.Control.Zoom()
6. ]
7. });
8. map.addControl(new SuperMap.Control.MousePosition());
9. //初始化图层
10. layer = new SuperMap.Layer.TiledDynamicRESTLayer("China", "http://localhost:8090/iserver/services/map-nee/rest/maps/connn", null, {maxResolution: "auto"});
11. //监听图层信息加载完成事件
12. layer.events.on({"layerInitialized": addLayer});
13. function addLayer() {
14. map.addLayer(layer);
15. //显示地图范围
16. map.setCenter(new SuperMap.LonLat(810 , 400), 0);
17. draw()
18. }

3.添加交互画点控件，并利用距离查询功能，查询到距离所绘制的点(及检测到污染物的点) 的弧段id(EdgeID)，供上游追踪分析使用

vectorLayer = new SuperMap.Layer.Vector("Vector Layer");

//定义画图控件

drawPolygon2 = new SuperMap.Control.DrawFeature(vectorLayer, SuperMap.Handler.Point);

drawPolygon2.events.on({"featureadded": function(e){

map.addLayer( vectorLayer);

drawPolygon2.deactivate();

//获取绘制出点

var geo=e.feature.geometry

console.log(geo)

//距离查询

var queryByDistanceParams = new SuperMap.REST.QueryByDistanceParameters({

queryParams: new Array(new SuperMap.REST.FilterParameter({name: "ne\_Network@ne"})),

returnContent: true,

distance: 300,

geometry: geo, //传入绘制的点几何对象

isNearest:true

});

var queryByDistanceService = new SuperMap.REST.QueryByDistanceService("http://localhost:8090/iserver/services/map-nee/rest/maps/connn");

queryByDistanceService.events.on({

"processCompleted": function(e){

console.log(e.result)

console.log(e.result.recordsets[0].features[0].attributes.SmEdgeID)

//获取到绘制点的最近弧段id

var Edge=parseInt(e.result.recordsets[0].features[0].attributes.SmEdgeID)

console.log(Edge,"Edge")

console.log("http://localhost:8090/iserver/services/transportationAnalyst-nee/rest/networkanalyst/ne\_Network@ne/traceup.json?edgeID="+Edge+"&isUncertainDirectionValid=true&returnFeatures=true&weightName=SMLENGTH")

qingqiu(Edge)

}

});

queryByDistanceService.processAsync(queryByDistanceParams);

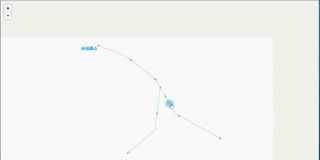
}});

map.addControl(drawPolygon2);

drawPolygon2.activate();

1. 将获得的EdgeID并利用rest-api -traceUp 资源执行 GET 请求，可查找给定弧段或节点的上游弧段和结点。同时将成功返回的要素构建成线对象（表示污染物可能来自于哪些水域）
2. $.ajax({
3. url:"http://localhost:8090/iserver/services/transportationAnalyst-nee/rest/networkanalyst/ne\_Network@ne/traceup.json?edgeID="+Edge+"&isUncertainDirectionValid=true&returnFeatures=true&weightName=SMLENGTH",
4. username:"你的用户名",
5. password:"你的密码",
6. type:"GET",
7. success:function(res){
8. console.log(res)
9. var jiegoulayer=new SuperMap.Layer.Vector("Vector Layer");
10. for(var i=0;i<res.edgesFeatures.length;i++)
11. {
12. var points=[]
13. // roadLine = new SuperMap.Geometry.LineString(points);
14. console.log(res.edgesFeatures[i].geometry.points)
15. for(var k=0;k<res.edgesFeatures[i].geometry.points.length;k++)
16. {
17. var l=res.edgesFeatures[i].geometry.points;
18. var geopoint=new SuperMap.Geometry.Point(l[k].x,l[k].y);
19. points.push(geopoint);
20. }
21. //构造为线对象，显示于
22. var roadLine = new SuperMap.Geometry.LineString(points);
23. var fea=new SuperMap.Feature.Vector(roadLine);
24. jiegoulayer.addFeatures(fea)
25. map.addLayer(jiegoulayer)
26. }
27. }
28. })

**效果展示：**



望大家多多指正。