



■ WebGIS系统的组成与分类

- 客户端、网络服务器、GIS引擎、数据库服务器
- 基于服务器、基于客户端

■ Web GIS的实现方式

- CGI、Server API、Java Servlet、ActiveX、Plug-in





本章章节

1

SVG在Web GIS中的应用

2

GeoVRML在Web GIS中的应用

3

海量遥感影像数据的Web发布

4

基于Web Service的开放式GIS

5

面向地图服务的OpenAPI

6

面向地学应用的Sensor Web服务



重点:

- SVG在Web GIS中的应用
- SVG实现GIS的方式
- 面向地图服务的OpenAPI

难点:

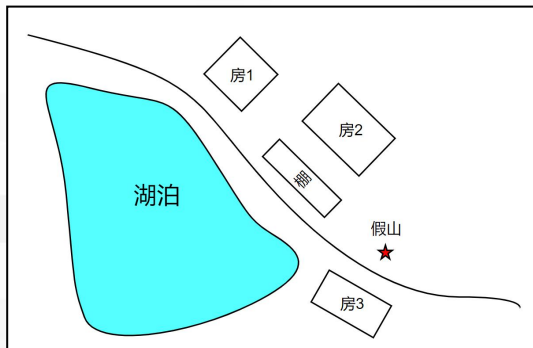
- SVG实现GIS的方式



4.1 SVG在Web GIS中的应用

19:49:06

- 4.1.1 SVG基础知识
- 4.1.2 SVG在Web应用中的优势
- 4.1.3 基于SVG的WebGIS

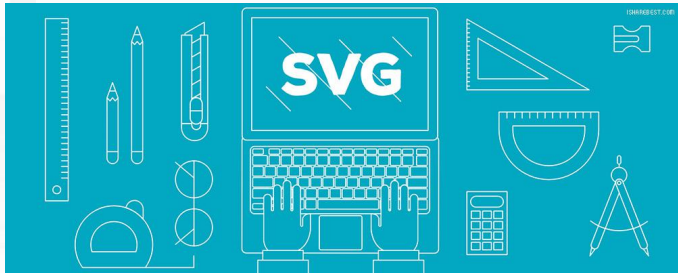


```
<svg width="1500" height="720" viewBox="0 0 100% 100%" xmlns="http://www.w3.org/2000/svg">
  <g>
    <title>Layer 1</title>
    <path id="svg_1" d="m56.5,58.05c178,45 166,58 237,141c71,83 145,130 211,130c66,0 63,11 62.5,10.95" stroke
    <path id="svg_2" d="m96.5,223.05c-9,-94 -54,-130 25,-110c79,20 79,-1 123,67c44,68 40,68 69,86c29,18 43,37
    <text xml:space="preserve" text-anchor="start" font-family="Helvetica, Arial, sans-serif" font-size="24"
    <rect transform="rotate(29.772916793823242 392.29730224609364,336.2162170410157)" id="svg_4" height="37.
    <rect stroke="#000" transform="rotate(44.42681884765625 341.1254577636719,206.36476135253906)" id="svg_5
    <rect transform="rotate(-45.31549835205078 277.77026367187494,98.71620941162111)" id="svg_6" height="54.
    <rect transform="rotate(-45.99977493286133 389.5946350097656,156.14865112304685)" id="svg_7" height="81.
    <path id="svg_8" d="m420,64151,280.8991815.41982,011.67477,-5.4198111.67477,5.4198115.41982,01-4.38472,3.
    <text xml:space="preserve" text-anchor="start" font-family="Helvetica, Arial, sans-serif" font-size="16"
    <text style="cursor: move;" xml:space="preserve" text-anchor="start" font-family="Helvetica, Arial, sans-
    <text style="cursor: move;" xml:space="preserve" text-anchor="start" font-family="Helvetica, Arial, sans-
    <text transform="rotate(-46.721187591552734 341.51583862304693,207.5703125)" xml:space="preserve" text="a
    <text xml:space="preserve" text-anchor="start" font-family="Helvetica, Arial, sans-serif" font-size="16"
  </g>
</svg>
```

4.1.1 SVG是什么

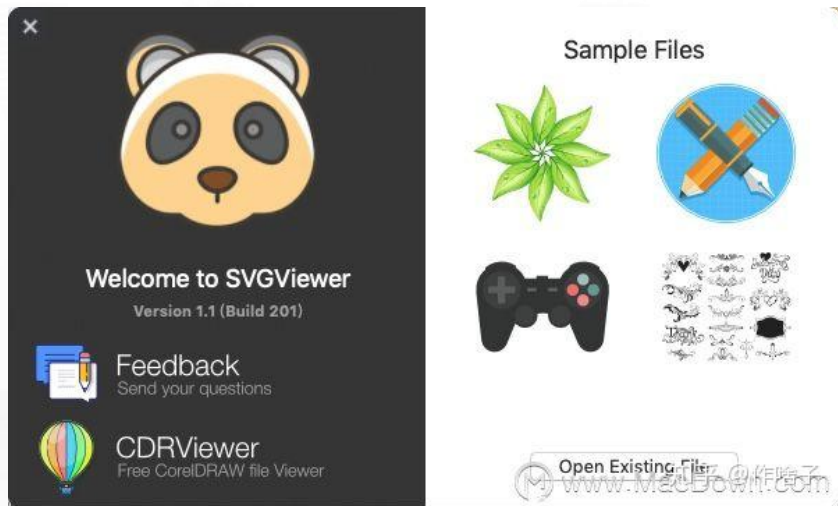
19:49:06

- SVG是Scalable Vector Graphics的英文缩写，可缩放矢量图形，是一种基于XML标准的图形影像描述语言。SVG的产生有其客观必然性。
- 进入20世纪90年代末，因特网的迅速发展使网上传统的栅格图像如（GIF，JPEG和PNG）已经很难满足人们的需要，为此，各知名厂商都纷纷推出了各自的Web图像解决方案。



- 4.1.1.1 SVG浏览器插件（目前已不需要）
- 4.1.1.2 在网页中嵌入SVG
- 4.1.1.3 SVG图形对象
- 4.1.1.4 对动画的支持
- 4.1.1.5 SVG事件简介
- 4.1.1.6 空间数据绑定

- 要实现SVG图像的显示，客户端卡可以安装SVG浏览器，Adobe研发的Svg Viewer功能强大，显示效果好，是网络上使用最多的。



4.1.1.2 在网页中嵌入SVG

19:49:07

■ 直接使用 `<svg>` 标签插入内容到网页中，成为DOM的一部分，然后可以使用CSS和JS进行控制。

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>SVG-GIS</title>
9   <!-- 可以用CSS控制SVG的样式，但属性与普通的网页元素不同 -->
10  <style type="text/css">
11    #testSvg {
12      border: 1px solid ■ #ccc;
13    }
14    #testSvg circle {
15      fill: ■ red;
16      stroke: ■ blue;
17      stroke-width: 3;
18    }
19  </style>
20 </head>
21
22 <body>
23   <svg width="400" height="300" id="testSvg">
24     <circle cx="100" cy="100" r="50" fill="red" stroke="black" stroke-width="2" id="testCircle"></circle>
25   </svg>
26 </body>
27
28 </html>
```

4.1.1.2 在网页中嵌入SVG

19:49:07

■ 可以使用 `<img>` `<embed>` `<object>` `<iframe>` 等标签把SVG文件插入到网页中。

```
//使用<img>标签

//或者SVG的base64编码


//使用<embed>标签
<embed id="embedSvg" type="image/svg+xml" src="test.svg"></embed>
//获取SVG DOM
var embedSvg = document.getElementById("embedSvg").getSVGDocument();
console.log("SVG DOM: ", embedSvg);

//使用<object>标签
<object id="objectSvg" type="image/svg+xml" data="test.svg"></object>
//获取SVG DOM
var objectSvg = document.getElementById("objectSvg").getSVGDocument();
console.log("SVG DOM: ", objectSvg);

//使用<iframe>标签
<iframe id="iframeSvg" src="test.svg"></iframe>
//获取SVG DOM
var iframeSvg = document.getElementById("iframeSvg").contentDocument;
console.log("SVG DOM: ", iframeSvg);
```

4.1.1.2 在网页中嵌入SVG

19:49:07

■ 读取SVG源码

```
<body>
  <style type="text/css">
    .svg-div {
      width: 400px;
      height: 300px;
      background: url("test.svg") no-repeat center / 50%;
      border: 1px solid ■ #ccc;
    }
  </style>

  <div class="svg-div"></div>

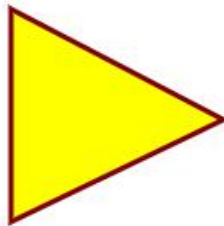
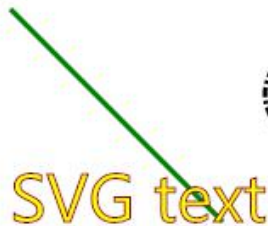
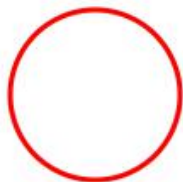
  <script>
    var svgStr = new XMLSerializer().serializeToString(document.getElementById("testSvg"));
    console.log(svgStr);
  </script>
</body>

</html>
```

4.1.1.3 SVG图形对象

19:49:07

- 矢量图形是用点和线来描述的，可以大大减少文件长度，提高传输效率。更重要的是，它将对图形效果的显示由服务器端移到客户端，可以充分利用客户端的资源，减轻服务器端的负担。



4.1.1.4 对动画的支持

19:49:07

- 目前，Web上播放的动画多为GIF格式，它也有着与网上传输图像格式相同的问题。即修改在服务器端实现，而不是客户端实现。SVG中提供了多门动画元素，可以叙述一个图形图像元素的实时变化。
- SVG中用标记<animateMotion>描述元素的缩放、旋转、偏斜等变换效果，用<animateColor>描述元素颜色的改变，还可以用<animate>描述元素淡入淡出的效果。
- 演示网站：<https://www.html5tricks.com/demo/html5-svg-dancing-flower/index.html>



4.1.1.5 SVG事件简介

19:49:07

- SVG提供了丰富的消息触发及事件响应事件，获取用户操作，如：在地图上移动、点击鼠标等。事件的影响能定义到整个文件对象中，也能定义在单个图像对象上。
- SVG有四种常用的鼠标响应的脚本写法(SMIL方式、Attributes方式、JavaScript+SMIL方式、EventListener方式)，根据不同的需要让大家有更多的选择。



➤ SMIL方式

```
<svg>
<rect x="15" y="15" width="40" height="40" fill="red">
<set attributeName="fill" to="blue" begin="click"/>
</rect>
</svg>
```

➤ Attributes方式

```
<svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:a3="http://ns.adobe.com/AdobeSVGViewerExtensions/3.0/"
a3:scriptImplementation="Adobe">
<script type="text/ecmascript" a3:scriptImplementation="Adobe">
<![CDATA[
function changeColor(evt)
{
var rect = evt.target;
rect.setAttributeNS(null, "fill", "blue")
}
]]></script>
<rect x="5" y="5" width="40" height="40"
fill="red"
onclick= "changeColor(evt)"/> u
</svg>
```

➤ JavaScript+SMIL方式

```
<svg onload="makeShape(evt)">
<script><![CDATA[
var svgns = "http://www.w3.org/2000/svg";u
function makeShape(evt) {
svgDoc = evt.target.ownerDocument;
var rect = svgDoc.createElementNS(svgns, "rect");v
rect.setAttributeNS(null, "x", "5");
rect.setAttributeNS(null, "y", "5");
rect.setAttributeNS(null, "width", "40");
rect.setAttributeNS(null, "height", "40");
rect.setAttributeNS(null, "fill", "red");
var set = svgDoc.createElementNS(svgns, "set");
set.setAttributeNS(null, "attributeName", "fill");
set.setAttributeNS(null, "to", "blue");
set.setAttributeNS(null, "begin", "click");
rect.appendChild(set);
svgDoc.rootElement.appendChild(rect);
}
]]></script>
</svg>
```



4.1.1.5 SVG事件简介

19:49:07

➤ EventListener方式

```
<svg onload="makeShape(evt)">
<script><![CDATA[
var svgnss = "http://www.w3.org/2000/svg";
function makeShape(evt) {
if ( window.svgDocument == null )
svgDoc = evt.target.ownerDocument;
var rect = svgDoc.createElementNS(svgnss, "rect");
rect.setAttributeNS(null, "x", 15);
rect.setAttributeNS(null, "y", 15);
rect.setAttributeNS(null, "width", 40);
rect.setAttributeNS(null, "height", 40);
rect.setAttributeNS(null, "fill", "red");
rect.addEventListener("click", changeColor, false); u
svgDoc.documentElement.appendChild(rect);
}
function changeColor(evt) {
var target = evt.target;
target.setAttributeNS(null, "fill", "blue");
}
]}</script>
</svg>
```

■ 外联法

■ 内嵌法

```
1 <script type="text/javascript" src="/map/xxx.json"></script>
2 <!-- 就可以在script标签里面使用jsonData -->
```

```
<g id="FA01">
  <rect x="7" y="7" width="50" height="50" style="fill: purple;"/>
  <text x="1" y="1" width="40" height="40" style="fill: purple;">
    桂林理工大学行知楼
  </text>
  <GeoAttribute>
    <面积>...</面积>
    <层数>...</层数>
  </GeoAttribute>
</g>
```

4.1.2 SVG在Web应用中的优势

- 1.基于XML格式，易于Web发布，传输，跨平台
- 2.交互性强，支持各种先进的网页交互技术
- 3.SVG是一种文本描述格式，易于查询和升级
- 4.数据量少，传输效率高
- 5.能获得更广泛的硬件支持
- 6.具有准确的颜色描述。

名称	修改日期	类型	大小
 例图.PNG	03/06/2021 13:10	PNG 图片文件	60 KB
 例图.svg	03/06/2021 12:59	Microsoft Edge ...	7 KB

4.1.3 空间数据在SVG文档中的组织

19:49:07

- 地理空间数据是用来描述地理空间现象的，一般可分为**空间几何数据**和**属性数据**。
- 由于空间几何数据比较复杂，在传统的GIS中，空间几何数据以文件的形式进行储存，且采用**拓扑数据模型**或**空间实体模型**进行组织。
- 两种模型的区别主要在于对**复杂空间对象**以及**空间对象之间关系的描述**

- 在拓扑数据模型中，其**优点**是多个不同的对象可以共用相同的结点，从而可以节省存储空间，可方便地进行各种空间分析，这种数据模型的最大**缺点**是维护拓扑关系的开销较大；
- 在实体模型中，空间对象的结点是直接存储在实体对象内部的，虽然会因公共结点的重复存储而产生数据冗余，但它无须维护实体间的拓扑关系



4.1.3 基于SVG的空间数据表达

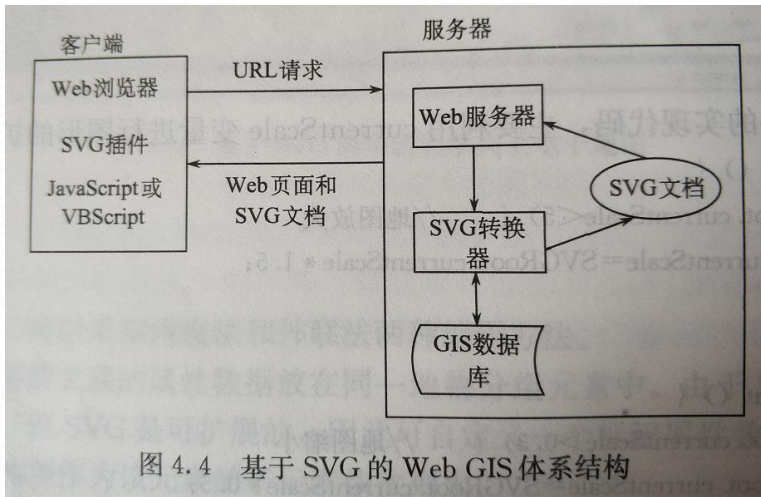
19:49:07

- 在空间数据处理中，一幅地图别分解为若干图块，每个图块包含若干层图，看作是要素（Feature）的集合（FeatureSet），并描述这些Feature的共同特征。每个Feature对应着一个空间实体，包含有几何属性（Geometry）和非几何属性（属性数据，Attribute）。几何属性又分为点、线、面与复杂几何对象，这几何对象都可以用SVG规范定义的标记来表示。



4.1.3 基于SVG的WEBGIS体系结构

19:49:07



在这个体系结构下，有一种简单的应用模式，即事先利用一些工具把GIS数据转换成SVG文档，供客户调用和显示。目前，ArcGIS9.0已提供了将地图数据直接转换成SVG文档的功能。



4.2. GeoVRML在Web GIS中的应用

19:49:07

- [4.2.1 VRML概述](#)
- [4.2.2 GeoVRML的特点](#)
- [4.2.3 基于GeoVRML的Web VRGIS](#)

4.2.1 VRML概述

1.VRML(Virtual Reality Modeling Language), **即虚拟现实建模语言**, 是20世纪90年代中期的一门新兴语言, 通过它可以创建一个虚拟的、可交互的场景。

- 存在感
- 交互性
- 立体感的视觉效果
- 立体感的听觉效果
- 动态显示与网络无关
- 脚本功能
- 全球资讯网参考点
- 超链接功能



4.2.2 GeoVRML的特点

19:49:07

- 1.支持多种坐标系统和参考椭球
- 2.全面提高数据精度
- 3.三维建模型功能进一步加强
- 4.浏览模式增强





4.2.3 基于GeoVRML的Web VRGIS

19:49:07

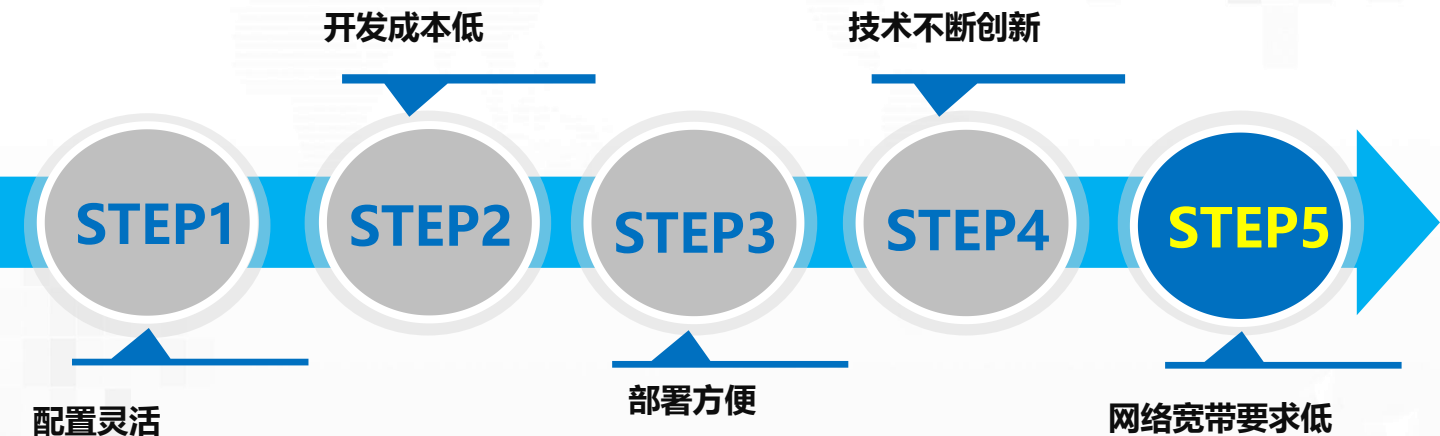
1.系统特点

2.地理要素的GeoVRML模拟

3.体系结构

4.2.3.1 系统特点

19:49:07



4.2.3.2地理要素的GeoVRML模拟

- 在GIS中，用点、线、面、体来表示基本的地理要素，而在VRML中定义了多种类型的几何节点，可以利用这些几何节点来模拟基本的GIS地理要素，达到对地理现象的三维模拟。

- 1) PointSet节点和Sphere节点。
- 2) IndexLineSet节点和Extrusion节点
- 3) IndexFaceSet节点
- 4) Box节点、Cone节点、Cylinder节点和Sphere节点
- 5) ElevationGrid节点
- 6) LOD节点

4.2.3.3 体系结构

基于VRML的WebGIS的核心就是在服务器端的VRML文件通过Web下载到Web 浏览器,再通过Web浏览器调用VRML浏览器进行三维场景的可视化 VRML。文件的产生有两种方式:

■ 静态生成

即预先将空间数据地理信息生成若干VRML文件,再利用INline 节点将这些文件串起来,根据用户在客户端的操作进行调用。

■ 动态生成

在服务器端根据客户端的请求,动态生成VRML文件,传送给浏览器。



4.3海量遥感影像数据的Web发布

19:49:08

4.3.1影像分块技术

4.3.2.空间索引技术

4.3.3.影响金字塔的建立

4.4.4.数据缓存技术

4.3.1 影像分块技术

- 由于在海量影像数据库中，每次调度和使用的图像数据只是数据库中的一小部分，**要高效地组织和管理数据，数据分块**是关键。
- 通过数据分块可以**减少数据的网络传输数据量**，方便数据压缩和有利于在计算机的内存中对图像数据进行运算处理，因此对存储的影像数据采用影像分块和重构方法来组织数据
- 在数据分块时，数据块的大小对影像调度效率有重要影响。

4.3.2 空间索引技术

- 常见的空间索引一般是自顶向下，逐级划分空间的各种数据结构空间索引，比较有代表性的包括BSP树，KDB树，R树，CELL树，四叉树和格网空间索引等。而格网索引对于影象数据库来说特别合适。
- 由于在影象数据库中数据按照分块的方式存储，而且数据块的划分是非常规则的并且彼此之间没有重叠，格网索引的建立和维护相当方便。
- 按照格网对图象进行编号然后通过编号和数据块的对应关系索引图象块。

4.3.3 影像金字塔的建立

19:49:08

海量影像数据管理模型对海量影像数据库管理系统所能管理的数据量，系统管理效率和可伸缩性是至关重要的。

- 对于影像库而言，一方面，数据所能表现的视觉效果和数据的分辨率呈正比关系。
- 另一方面，因为显示设备的限制，在同一时间内所能观察的范围和分辨率有限，过高的分辨率对于最终用户的视觉感受是没什么贡献的。

4.3.3 影像金字塔的建立

19:49:08

影像金字塔的建立可以采用以下**两种方式**：

- 如果生产的源数据本身就是多分辨率的，则可以直接建立不同的图像数据工程，自动构建金字塔式的影像数据库。
- 对于只生产了基础层的影像数据，为了提高调度效率，上层可以根据相应的比例尺从下层抽取数据来构建金字塔，这是目前多数系统所采用的方式。

4.3.4 数据缓存技术

对影像数据进行分块组织后，在显示某一区域的影像时，需要同时调入多个影像块进行拼接，显示的效率和调入影像块数以及每块的大小密切相关。在每块大小固定的情况下，调入的块数越多，且数据库的IO次数越多，花费的时间就越长。



4.4 基于Web Service的开放式GIS

19:49:08

4.4.1 Web Service概述

4.4.2 基于Web Service的开放式GIS

4.4.3 OGC的几个开放式GIS服务标准

什么是Web Service?

Web Service是指使用标准技术实现的，公布并运行在因特网上的一些业务流程。

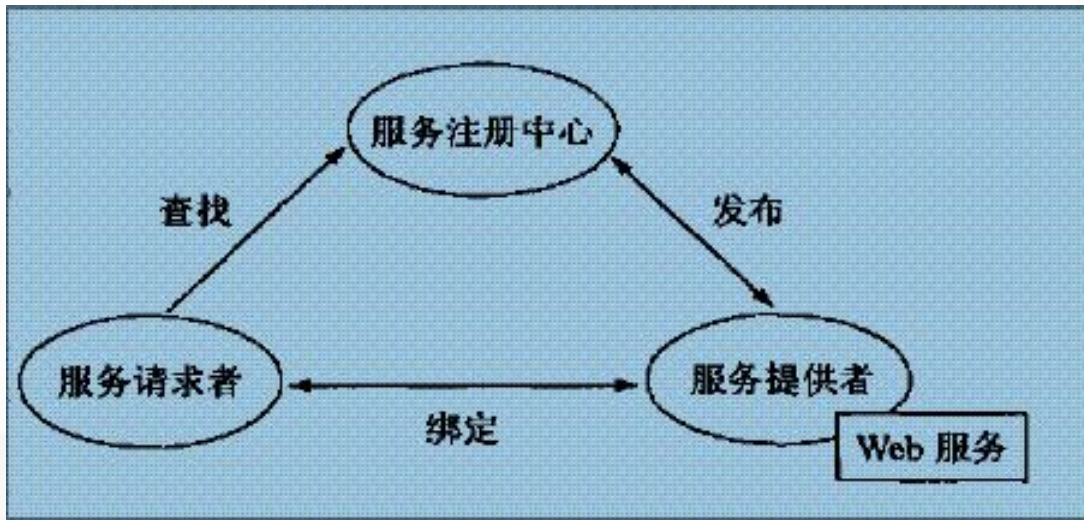
应用Web Service的公司可以轻松地通过标准网络协议使用XML格式把应用程序连接到任何用户端

Web Service还可以轻松的把来自完全不同硬件平台（如大型机、应用服务器和Web服务器）的应用程序互相联结起来。

<http://lbsyun.baidu.com/index.php?title=webapi>

Web Service 体系结构

WebService体系结构基于三种角色（服务提供者，服务注册中心，服务请求者）之间的交互。



4.4.2 基于Web Service的开放式GIS

19:49:08

所谓开放式GIS就是在网络环境中对不同种类的地理数据和地理处理方法的透明访问。开放式GIS的目的是提供一套具有开放界面规范的通用组件，这些组件可以实现不同种类地理数据和地理处理方法的透明访问。

开放式GIS的特点：

可互操作

1

普适性

2

3

可扩展

4

5

6

兼容性

7

支持信息团体

可移植性

可靠性

- ▣ 地理数据服务 (Data Service) :提供对空间数据的服务, 主要有WFS (Web Feature Service , 矢量数据服务) , WCS (Web Coverage Service , 栅格数据服务) , 地理数据服务返回的结果通常是带有空间参照系的数据。
- ▣ 地图表现服务 (Portrayal Service) :提供对空间数据的表现, 主要有WMS (Web Map Service , 地图服务) , 其中地图可以由多个图层组合起来, 可以用SLD (Styled Layer Descriptor) 对地图进行描述。地图服务的返回结果通常是矢量图形或栅格图形。

- ▣ 过程处理服务 (Processing Service) : 提供地理数据的查找、索引等服务, 主要有Geocoder (地学编码服务)、Gazetteer (地名索引服务)、Coordinate Transfer Service (坐标转换服务等)。
- ▣ 发布注册服务 (Registry) : 提供对各种服务的注册服务, 以便于服务的发现。其中包括: 数据类型、数据实例、服务类型、服务实例的注册服务。注册服务提供了各个注册项的登记、更新及查找服务。
- ▣ 客户端应用 (Client Application) : 即客户端的基本应用, 如地图的显示、地图浏览以及其他一些增值服务。



- 1.WMS:将具有地理空间信息的数据制作成地图提供给用户
- 2.WFS:为 浏览器提供经过GML格式封装的地理空间数据，支持对地理要素数据的插入、更新、删除、查询、和发现
- 3.WCS:支持网络化的地理空间数据的相互交换。此时地理空间数据作为包含地理位置或特征的“覆盖”



参考: <http://blog.csdn.net/fengyao1995/article/details/53244>

① 迎春小区



4.5 面向地图服务的OpenAPI

19:49:08

■ 4.5.1 OpenAPI的基本概念

■ 4.5.2 Google Map开放地图 API

4.5.1 OpenAPI的基本概念

19:49:08

在互联网时代，把网站的服务封装成一系列计算机易识别的数据接口开放出去，供第三方使用，所开发的API就被称为OpenAPI.

按照提供的服务内容进行分类，可分为：

搜索类

注册类

多媒体类

文字资讯类

地理信息类

用户及关系类

电子商务类

4.5.2 Google Maps开放地图 API

Google MapsAPI是目前应用最为成功和广泛的Open API, 开放地图API, 在交通、旅游、市场运销、广告等很多行业得到广泛的应用。



4.5.2 如何在网站中引入百度地图

<http://api.map.baidu.com/lbsapi/creatmap/index.html>





4.6 面向地学应用的Sensor Web服务

19:49:08

■ 4.6.1 Sensor Web的基本概念

■ 4.6.2 Sensor Web的观测服务体系

4.6.1 Sensor Web的基本概念

Sensor Web: 传感器网络，由几十、几百甚至几千个微型智能传感器节点通过无线通信方式构成的网络，具有自组织、自配置、自修复、低功耗、监测密度大等特点，形成覆盖面积较大的（准）实时探测区域。

环境监测、农业灌溉、空间探测、资源调查、
灾害预警、工业（家庭）自动化等领域

4.6.1 Sensor Web的基本概念

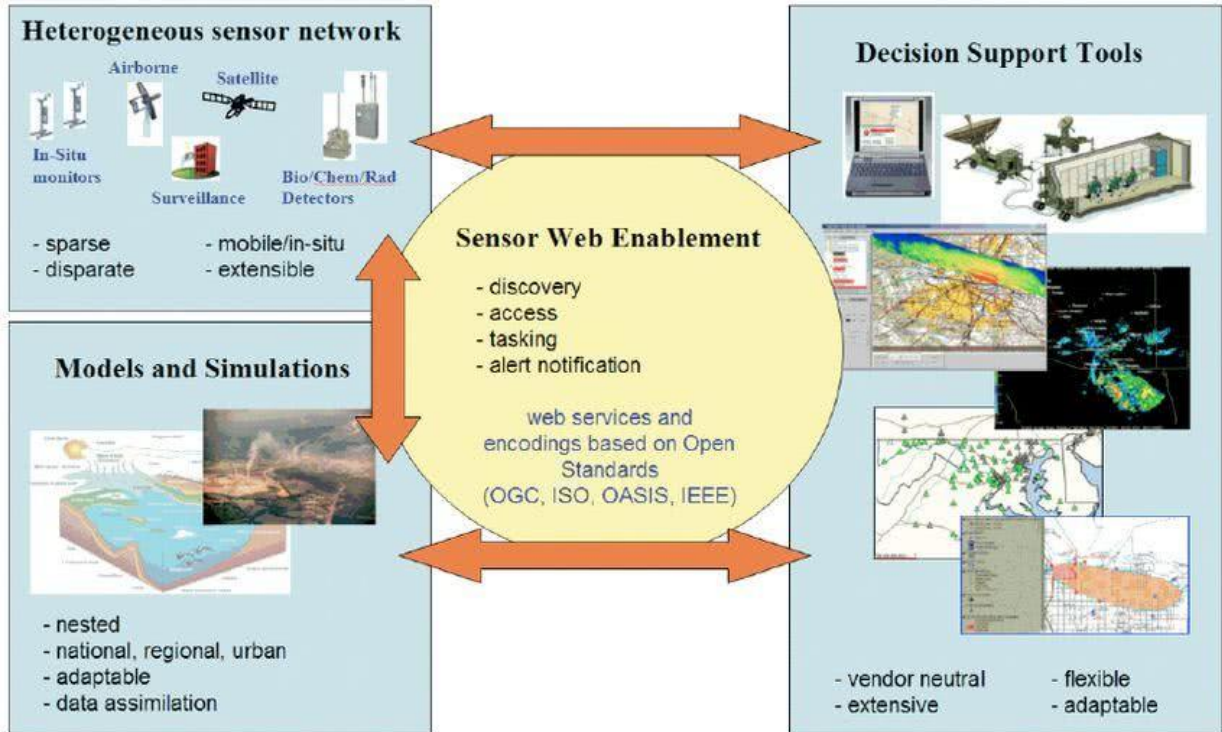
2003年 OGC和国际标准化组织制定了Sensor Web 标准和协议，制定处七个传感器网络标准，包含三个信息模型和四个信息服务实现规范

- 传感器描述语言编码标准
- 观测和测量编码标准
- Transducer模型语言

4.6.1 Sensor Web的基本概念

- 传感器规划服务
- Web通知服务
- 传感器预警服务
- 传感器观测服务

4.6.1 Sensor Web服务体系结构



4.6.2 Sensor Web的观测服务体系

Sensor web 服务体系中，SOS是其中最重要的组成部分，通过标准接口，提供符合标准信息模型的观测和测量数据。

- 资源层：提供SOS的资源
- 业务层：业务层SOS的核心部分
- 表现层：提供用户与SOS交互操作的界面



- 1、Web Server概念?
- 2、什么是SVG，SVG如何进行空间数据表达?
- 3、基于Web server的OpenGIS提供哪些服务?
- 4、如何使用地图服务API?
- 5、Sensor web 服务的七个传感器网络标准?