污染源在线监控平台

概要设计



**编 制：**

**校 对：**

**审 核：**

**审 定：**

**批 准：**

**西安思坦环境科技有限公司**

目录

[1. 概述 1](#_Toc465236990)

[1.1. 项目背景 1](#_Toc465236991)

[1.2. 参考标准 1](#_Toc465236997)

[2. 设计原则 2](#_Toc465236998)

[3. 关键设计决策 3](#_Toc465236999)

[3.1. 分层架构模式 3](#_Toc465237000)

[3.2. 数据持久化与对象关系映射 4](#_Toc465237001)

[3.3. 面向服务体系结构SOA 5](#_Toc465237002)

[3.3.1. SCA 6](#_Toc465237003)

[3.3.2. ESB 6](#_Toc465237004)

[3.4. 并发服务器 7](#_Toc465237005)

[3.5. 大数据量处理与内存数据库 7](#_Toc465237006)

[4. 体系结构分析设计 8](#_Toc465237007)

[4.1. 可复用性功能分析 8](#_Toc465237008)

[4.2. 逻辑架构 8](#_Toc465237009)

[4.3. 基础功能分析 10](#_Toc465237010)

[4.3.1. 数据通信 10](#_Toc465237011)

[4.3.2. 数据持久化 12](#_Toc465237012)

[4.3.3. 界面交互 12](#_Toc465237013)

[4.4. 物理架构 14](#_Toc465237014)

[4.4.1. 基础通信 14](#_Toc465237015)

[4.4.2. 传输协议 14](#_Toc465237016)

[4.4.3. 数据通信 15](#_Toc465237017)

[4.4.4. 数据库系统 15](#_Toc465237018)

[4.4.5. 数据持久化 16](#_Toc465237019)

[4.4.6. SOA平台 16](#_Toc465237020)

[4.4.7. 平台\_现场机通信 17](#_Toc465237021)

[4.4.8. 平台\_平台通信 17](#_Toc465237022)

[4.4.9. 平台数据管理 18](#_Toc465237023)

[4.4.10. 数据缓存 18](#_Toc465237024)

[4.4.11. 数据采集 19](#_Toc465237025)

[4.4.12. 数据传输交换 19](#_Toc465237026)

[4.4.13. 数据统计分析 20](#_Toc465237027)

[4.4.14. 交互控件库 20](#_Toc465237028)

[4.4.15. 界面交互 20](#_Toc465237029)

[4.4.16. 平台界面交互 21](#_Toc465237030)

[4.4.17. 功能服务 21](#_Toc465237031)

[4.4.18. 应用服务器 21](#_Toc465237032)

[5. 数据库设计 22](#_Toc465237033)

[5.1. 数据库体系结构 22](#_Toc465237034)

[5.2. 商业数据库产品 23](#_Toc465237035)

# 概述

## 项目背景

随着我国经济的迅速发展，国民经济突飞猛进，城市规模不断扩大，工业企业不断增多，由此带来的问题就是污染源的数量大量增加以及分布的越来越复杂，已经严重的影响了经济的可持续发展，前些年，由于民众的环保意识普遍薄弱，而且还存在着丰厚的经济利益驱使，使得环保的监控管理工作变得日益复杂。如今，人们已经意识到环境对我们生存的重要性，环境保护已经成为我国的一项基本国策，越来越受到国家和广大群众的关注，我国的环境管理工作也在不断地深化，不断加大管理力度。

传统的环境监测方法需要环保部门将大量的时间、人力、物力投入到环境现场的数据采集工作之中。这种原有的监测方式，不但工作强度大、自动化程度低、数据完备性差、利用率低，很难真正的反应该排污口的实际排放量，不能很好地反映实际工况，而且随着污染源的增加，使得传统的监管方式已经远远不能达到保护环境、处罚排污企业的目的。

随着当今社会信息、网络技术的不断飞速发展，采用信息化管理已成为提高环保监管工作与决策水平的重要技术基础，利用现代化最先进的通讯技术、电子技术及软件开发技术，建立一套环境自动监测监控系统，实现监测监控系统的自动化，从更高的层次上对工业污染进行有效的管理，体现了环保作为一个新兴行业的发展方向；这样有利于加大对污染企业的监管力度，提高企业的自律意识，促进企业履行自己的环境义务，帮助企业提高环境治理的水平；不但可以大大降低环境污染提高企业的经济效益，同时，还提高了环保投资的社会效益。



## 参考标准

|  |  |
| --- | --- |
| 整体开发流程 | 《HJ511-2009环境信息化标准指南》  《HJ622-2011环境保护应用软件开发管理技术规范》 |
| 数据库规范 | 《HJ419-2007环境数据库设计与运行管理规范》 |
| 污染源代码 | 《GB16705-1996环境污染类别代码》  《GB16706-1996环境污染源类别代码》  《HJ417-2007环境信息分类与代码》 |
| 数据监测、技术规范 | 《HJ/T75-2007固定污染源烟气排放连续监测技术规范》  《HJ/T76-2007固定污染源烟气排放连续监测系统技术要求及检测方法》  《HJ/T355-2007水污染源在线监测系统运行与考核技术规范》  《HJ/T356-2007水污染源在线监测系统数据有效性技术判别规范》 |
| 数据采集传输 | 《HJ212-2005污染源在线自动监控（监测）系统数据传输标准》  《HJ352-2007环境污染源自动监控信息传输、交换技术规范》  《HJ660-2013环境监测信息传输技术规定》  《国控重点污染源自动监控信息传输与交换管理规定》 |
| 数据有效性标准 | 《“十二五”主要污染物总量减排考核办法》  《国家监控企业污染源自动监测数据有效性审核办法》 |

# 设计原则

本系统将根据国家环保总局关于在全国各省市实施《国家环境监控信息系统》的要求，充分考虑地方环保局及国家环保总局的要求，并结合环保部门的具体业务需求进行设计，具有明显的针对性。

该项目是环保信息化建设重要的组成部分，是一个基础性的建设工程，必须从环境保护的全局出发，将系统建成一个高起点，易于扩充、升级、管理和使用的网络化系统。在设计过程中，将遵循以下原则：

* 先进性与实用性相结合的原则

必须采用最先进的技术，使项目建成并投入使用后，可保持今后相当长一段时间内不落后，并且只需稍作扩展，就可满足今后发展的需要。同时在设计阶段，要根据实际情况和本项目的招标要求，保证该项目建成后具有很强的实用性和可靠性。

* 开放性与标准化原则

在总体设计上，应采用开放式的体系结构，使系统易于扩充，使相对独立的分系统易于进行组合调整。有适应外界环境变化的能力，即在外界环境改变时，系统可以不作修改或仅作小量修改就能在新环境下运行。网络选用的通信协议和设备要符合国际标准，将不同应用环境和不同结构优势有机地结合起来。也就是说，要使网络的硬件环境、通讯环境、软件环境、操作平台之间的相互依赖减至最小，发挥各自优势。同时，保证网络的互联，为信息的互通和应用创造有利的条件。

* 兼容性与可扩充性原则

必须兼顾数据的兼容性，要为今后数据的监管范围的扩大留有余地，如考虑与县市相应系统的数据兼容问题等。还要具有可扩充性，以便根据今后的应用需要，进一步扩充其他的应用功能。

* 可靠性与安全性原则

系统安全可靠运行是整个系统建设的基础。鉴于环保信息的重要性，网络系统要有很高的可靠性。另外，应确保系统的稳定、数据传输的正确性，防止异常情况的发生。

* 接入广泛性及接口标准化

为最大限度保护原有的投资、方便今后的升级、扩展，系统应符合标准化协议如工、等，采用标准化的设计和标准化的产品。在系统接入方面，采用开放的接口标准，使系统的接入更具有广泛性。在系统数据传输方面，采用PSTN、ADSL、无线通讯及手机通讯等方式，从而满足不同设备的需求。监控中心与软件系统有很好的开放性和扩展性，数据处理和使用允许多元化挖掘，并允许多种环境数据的接入可与任意数据采集仪结合。

# 关键设计决策

## 分层架构模式

分层描述的是这样一种架构设计过程：从最低级别的抽象开始，称为第1层。这是系统的基础。通过将第J层放置在第J-1层的上面逐步向上完成抽象阶梯，直到到达功能的最高级别，称为第N层。因而分层模式就可以定义为：将解决方案的组件分隔到不同的层中。每一层中的组件应保持内聚性，并且应大致在同一抽象级别。每一层都应与它下面的各层保持松散耦合。

分层模式的关键点在于确定依赖：即通过分层，可以限制子系统间的依赖关系，使系统以更松散的方式耦合，从而更易于维护。

## 数据持久化与对象关系映射

数据持久化就是将内存中的数据模型转换为存储模型，以及将存储模型转换为内存中的数据模型的统称. 数据模型可以是任何数据结构或对象模型，存储模型可以是关系模型、XML、二进制流等。

使用数据持久化有以下好处：

1、程序代码重用性强，即使更换数据库，只需要更改配置文件，不必重写程序代码。

2、业务逻辑代码可读性强，在代码中不会有大量的SQL语言，提高程序的可读性。

3、持久化技术可以自动优化，以减少对数据库的访问量，提高程序运行效率。

数据持久化对象的基本操作有：保存、更新、删除、查询等。

总之，数据持久化是解决程序与数据库之间交互访问的问题，使程序不直接访问数据库，而是直接访问Session会话，然后由Session会话与数据库“打交道”。只要确保，程序每次访问Session中的数据时，与数据库中的数据保持一致，就不会出现错误。

对象关系映射，是一种程序技术，用于实现面向对象编程语言里不同类型系统的数据之间的转换。从效果上说，它其实是创建了一个可在编程语言里使用的“虚拟对象数据库”。

面向对象是从软件工程基本原则（如耦合、聚合、封装）的基础上发展起来的，而关系数据库则是从数学理论发展而来的，两套理论存在显著的区别。为了解决这个不匹配的现象，对象关系映射技术应运而生。

对象关系映射（Object-Relational Mapping）提供了概念性的、易于理解的模型化数据的方法。ORM方法论基于三个核心原则： 简单：以最基本的形式建模数据。 传达性：数据库结构被任何人都能理解的语言文档化。 精确性：基于数据模型创建正确标准化的结构。 典型地，建模者通过收集来自那些熟悉应用程序但不熟练的数据建模者的人的信息开发信息模型。建模者必须能够用非技术企业专家可以理解的术语在概念层次上与数据结构进行通讯。建模者也必须能以简单的单元分析信息，对样本数据进行处理。ORM专门被设计为改进这种联系。

## 面向服务体系结构SOA

面向服务的体系结构（SOA）是一个组件模型，它将应用程序的不同功能单元（称为服务）通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种这样的系统中的服务可以以一种统一和通用的方式进行交互。

这种具有中立的接口定义（没有强制绑定到特定的实现上）的特征称为服务之间的松耦合。松耦合系统的好处有两点，一点是它的灵活性，另一点是，当组成整个应用程序的每个服务的内部结构和实现逐渐地发生改变时，它能够继续存在。而另一方面，紧耦合意味着应用程序的不同组件之间的接口与其功能和结构是紧密相连的，因而当需要对部分或整个应用程序进行某种形式的更改时，它们就显得非常脆弱。

基于SOA来构建的IT系统具备如下特点：

1.以业务为中心

SOA更多关注于用户业务，通过业务人员参与SOA系统的规划、设计和管理，使得IT系统能在对业务的深刻理解的基础上进行构建，实现IT系统与用户业务的密切结合。在具体实施中，通过把完成实际业务流程中的一项任务所需的IT资源组织为服务进行封装，从而达到以业务为核心，通过业务选择技术，避免技术制约业务的问题。

2.灵活适应变化

IT系统围绕用户业务构建，用户业务在实现层通过表现为一系列松散耦合的”服务“来实现，这些服务可以根据用户需求随需组合，使得IT系统对于业务的适应能力明显提高。

3.重用IT资源，提升开发效率

SOA强调对”服务“的重用，对原有IT资源的重用度提升是SOA带来的关键效果之一，大量具有高重用的服务资源，为快速构建新的业务功能和业务系统奠定基础，使得IT系统的开发和软件生产效率得到提升。同时，重用过程有利于保护用户前期的信息化投资和IT资产积累，节省IT系统开发成本，实现用户信息化的可持续性建设与发展。

4.更强调标准

SOA的实现强调基于统一的标准，SOA系统建立在大量的开放标准和协议之上，以实现系统及信息的互联互通和互操作。因此，SOA系统从规划到实施，标准都至关重要。

### SCA

服务组件架构（SCA）语言无关的编程模型，它提供了一种统一的面向服务构件的调用方式，使得客户可以把不同的软件模块通过服务构件的标准化而统一地封装起来和被调用访问。SCA描述了利用面向服务架构（SOA）来构建应用程序和系统的模型。SCA是基于开放标准（例如Web服务）构建的，它扩展和补充了先前的服务实现方法。

SCA的基本思想是将业务功能作为一系列服务来提供，这些服务组合到一起，以创建满足特定业务需要的解决方案。这些复合应用程序既可以包含专门为该应用程序创建的新服务，也可以包含来自现有系统和应用程序的业务功能（作为复合应用程序的一部分来重用）。SCA为服务组合和服务组件的创建（包括SCA复合应用程序内部现有应用程序功能的重用）提供了模型。

### ESB

ESB全称为Enterprise Service Bus，即企业服务总线。它是传统中间件技术与XML、Web服务等技术结合的产物。ESB提供了网络中最基本的连接中枢，是构筑企业神经系统的必要元素。ESB的出现改变了传统的软件架构，可以提供比传统中间件产品更为廉价的解决方案，同时它还可以消除不同应用之间的技术差异，让不同的应用服务器协调运作，实现了不同服务之间的通信与整合。从功能上看，ESB提供了事件驱动和文档导向的处理模式，以及分布式的运行管理机制，它支持基于内容的路由和过滤，具备了复杂数据的传输能力，并可以提供一系列的标准接口。

## 并发服务器

在网络程序里面，一般来说都是许多客户对应一个服务器，为了处理客户的请求， 对服务端的程序就提出了特殊的要求。目前最常用的服务器模型有：

• 循环服务器:服务器在同一个时刻只可以响应一个客户端的请求

• 并发服务器:服务器在同一个时刻可以响应多个客户端的请求

当服务器处理一个客户请求可能需要相当长一段时间的时候，使用迭代服务器模式（即整个服务器可能被耽搁客户长期占用）是会影响系统性能的。而处理并发连接的服务器可以称为并发服务器。

监控平台同时与多个现场机连接，存在并发通信的情况，而且通信数据具备实时性特征，必须在短时间内进行处理，一次，监控平台需要采用并发服务器与现场机之间进行通信。

## 大数据量处理与内存数据库

内存数据库，顾名思义就是将数据放在内存中直接操作的数据库。相对于磁盘，内存的数据读写速度要高出几个数量级，将数据保存在内存中相比从磁盘上访问能够极大地提高应用的性能。

内存数据库抛弃了磁盘数据管理的传统方式，基于全部数据都在内存中重新设计了体系结构，并且在数据缓存、快速算法、并行操作方面也进行了相应的改进，所以数据处理速度比传统数据库的数据处理速度要快很多，一般都在10倍以上。内存数据库的最大特点是其“主拷贝”或“工作版本”常驻内存，即活动事务只与实时内存数据库的内存拷贝打交道。

当监控平台通过并发服务器接收到大量现场数据时需要将数据快速存入数据库中，传统的关系型数据库将数据存储在硬盘中，其性能不足以支撑对其频繁大量的访问。因此选用内存数据库作为接收数据与关系数据库之间的缓存。

# 体系结构分析设计

## 可复用性功能分析



## 逻辑架构

通过对系统整体的需求分析，污染源在线监控平台系统逻辑架构如图所示。系统整体上采用了分层的体系结构，从下向上大致可划分为基础层、业务逻辑层和应用层。



## 基础功能分析

### 数据通信









### 数据持久化



### 界面交互







## 物理架构



### 基础通信

根据需求分析可知，系统平台与现场机都将接入环保专网或者基于互联网的vpn网络。因此不论现场机具体以何种物理方式接入网络，最终与平台之间将通过TCP/IP协议进行通信。建立网络通信连接至少要一对端口号(socket)。socket本质是编程接口(API)，对TCP/IP的封装，TCP/IP也要提供可供程序员做网络开发所用的接口，这就是Socket编程接口。主流的编程语言都已提供了对socket的支持。

### 传输协议

传输协议定义了系统之间数据交换的内容、格式，传输协议组件主要提供了依据外部数据生成协议格式数据和解析协议格式数据的抽象方法。为针对具体协议的协议组件提供基础功能。



### 数据通信

数据通信组件对基础通信和传输协议组件的功能进行了封装，并对通信模式进行了抽象，提供了基于通信模式的高级会话接口抽象。



### 数据库系统

数据库系统（Database System），是由数据库及其管理软件组成的系统。数据库系统是为适应数据处理的需要而发展起来的一种较为理想的数据处理系统，也是一个为实际可运行的存储、维护和应用系统提供数据的软件系统，是存储介质、处理对象和管理系统的集合体。

关系数据库，是建立在关系数据库模型基础上的数据库，借助于集合代数等概念和方法来处理数据库中的数据，同时也是一个被组织成一组拥有正式描述性的表格，该形式的表格作用的实质是装载着数据项的特殊收集体，这些表格中的数据能以许多不同的方式被存取或重新召集而不需要重新组织数据库表格。关系数据库的定义造成元数据的一张表格或造成表格、列、范围和约束的正式描述。每个表格（有时被称为一个关系）包含用列表示的一个或更多的数据种类。 每行包含一个唯一的数据实体，这些数据是被列定义的种类。当创造一个关系数据库的时候，你能定义数据列的可能值的范围和可能应用于那个数据值的进一步约束。而SQL语言是标准用户和应用程序到关系数据库的接口。其优势是容易扩充，且在最初的数据库创造之后，一个新的数据种类能被添加而不需要修改所有的现有应用软件。目前主流的关系数据库有oracle、db2、sqlserver、sybase、mysql等。

### 数据持久化

Hibernate是一个开放源代码的对象关系映射框架，它对JDBC进行了非常轻量级的对象封装，它将POJO与数据库表建立映射关系，是一个全自动的orm框架，hibernate可以自动生成SQL语句，自动执行，使得Java程序员可以随心所欲的使用对象编程思维来操纵数据库。 Hibernate可以应用在任何使用JDBC的场合，既可以在Java的客户端程序使用，也可以在Servlet/JSP的Web应用中使用，最具革命意义的是，Hibernate可以在应用EJB的J2EE架构中取代CMP，完成数据持久化的重任。

Hibernate框架中数据持久化机制：

在业务程序与数据库之间，Hibernate框架使用Session会话，来完成数据的提交、更新、删除、查询等等。

1、向数据库提交数据

在程序中保存对象时，会把数据保存到Session会话中，然后根据框架的配置文件，自动或手动决定什么时候把这种保存提交到数据库。

2、从数据库中查询数据

在查询数据之前，需要清理缓存（手动清理，或者通过配置文件框架自动清理）清理缓存的目的是为了使Session会话中的数据与数据库中的数据保持一致。然后程序只需要查询Session会话中的数据即可。

### SOA平台

Tuscany 是 Apache 的开源项目，它是 IBM、Oracle、SAP 等厂商联合成立的 SOA 标准化组织 -OSOA 支持下开发出的 SCA 框架，它既是开源界 SCA 的试金石，也是当前开源界最成熟的 SCA 框架之一。Tuscany是一套开源的sca框架模型，是做soa的基础架构。

Synapse 是一个简单的 XML 和 Web 服务管理与集成代理，可用于构成 SOA 和企业服务总线（ESB）的基础。Synapse是 Web 服务项目中一项成熟的 Apache 活动，并且是非常成功的 Apache Axis2 项目的一个分支。它提供了中介、管理、以及在各种不同的应用程序之间转换 XML 消息的能力。

### 平台\_现场机通信

平台与现场机之间的通信需要遵循《HJT 212-2005 污染源在线自动监控(监测)系统数据传输标准》。标准适用于污染源在线自动监控（监测）系统自动监控设备和监控中心之间的数据交换传输。规定了数据传输的过程及系统对参数命令、交互命令、数据命令和控制命令的数据格式和代码定义。



### 平台\_平台通信

平台之间的通信需要遵循《HJT 352 2007环境污染源自动监控信息传输、交换技术规范》和《HJ 660-2013环境监测信息传输技术规定》。

《HJT 352 2007环境污染源自动监控信息传输、交换技术规范》标准描述了国家级、省级之间的交换流程、交换模型，适用于国家级和省级之间的污染源自动监控信息交换活动；省级范围内的污染源自动监控信息交换可参照执行。描述了环境污染源自动监控系统信息的内容和报文格式，适用于各级环保部门。适用于各级环境保护部门之间的污染源自动监控信息交换活动。

《HJ 660-2013环境监测信息传输技术规定》标准规定了环境监测信息的传输模式、传输流程，传输的数据格式和代码定义。适用于国家各级环境监测站、各级自动监测站和有关单位之间环境监测信息的传输活动。



### 平台数据管理

监控平台所需要维护的持久化数据主要分为系统数据、监测数据、污染信息数据以及地理信息数据四类，系统数据主要包含用户信息权限等系统自身相关的数据；监测数据是现场机发送给平台的监测数据；污染信息是排污企业，污染源，污染因子，报警条件等信息；GIS数据信息是与GIS应用相关的地理信息。



### 数据缓存

redis是一个key-value存储系统。和Memcached类似，它支持存储的value类型相对更多，包括string(字符串)、list(链表)、set(集合)、zset(sorted set --有序集合)和hash（哈希类型）。这些数据类型都支持push/pop、add/remove及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。在此基础上，redis支持各种不同方式的排序。与memcached一样，为了保证效率，数据都是缓存在内存中。区别的是redis会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了master-slave(主从)同步。它提供了Java，C/C++，C#，PHP，JavaScript，Perl，Object-C，Python，Ruby，Erlang等客户端，使用很方便。

Redis支持主从同步。数据可以从主服务器向任意数量的从服务器上同步，从服务器可以是关联其他从服务器的主服务器。这使得Redis可执行单层树复制。存盘可以有意无意的对数据进行写操作。由于完全实现了发布/订阅机制，使得从数据库在任何地方同步树时，可订阅一个频道并接收主服务器完整的消息发布记录。同步对读取操作的可扩展性和数据冗余很有帮助。

### 数据采集

Netty是由JBOSS提供的一个java开源框架。Netty提供异步的、事件驱动的网络应用程序框架和工具，用以快速开发高性能、高可靠性的网络服务器和客户端程序。

也就是说，Netty 是一个基于NIO的客户，服务器端编程框架，使用Netty 可以确保你快速和简单的开发出一个网络应用，例如实现了某种协议的客户，服务端应用。Netty相当简化和流线化了网络应用的编程开发过程，例如，TCP和UDP的socket服务开发。

“快速”和“简单”并不意味着会让你的最终应用产生维护性或性能上的问题。Netty 是一个吸收了多种协议的实现经验，这些协议包括FTP，SMTP，HTTP，各种二进制，文本协议，并经过相当精心设计的项目，最终，Netty 成功的找到了一种方式，在保证易于开发的同时还保证了其应用的性能，稳定性和伸缩性。

### 数据传输交换

数据传输交换组件主要实现平台之间的数据传输交换。



### 数据统计分析

数据统计分析组件主要实现了对数据集的各种统计分析功能。



### 交互控件库

根据不同的客户端实现形式，可以选择不同平台上的交互控件库或者框架。控件库的种类繁多，不同平台的主流控件库或框架如下：

WPF（Windows Presentation Foundation）是微软推出的基于Windows Vista的用户界面框架，属于.NET Framework 3.0的一部分。它提供了统一的编程模型、语言和框架，真正做到了分离界面设计人员与开发人员的工作；同时它提供了全新的多媒体交互用户图形界面。

Bootstrap，来自 Twitter，是目前很受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。它由Twitter的设计师Mark Otto和Jacob Thornton合作开发，是一个CSS/HTML框架。Bootstrap提供了优雅的HTML和CSS规范，它即是由动态CSS语言Less写成。Bootstrap一经推出后颇受欢迎，一直是GitHub上的热门开源项目，包括NASA的MSNBC的Breaking News都使用了该项目。国内一些移动开发者较为熟悉的框架，如WeX5前端开源框架等，也是基于Bootstrap源码进行性能优化而来。

### 界面交互

由于系统所涉及的数据数量较多，数据之间也存在着逻辑联系，因此采用数据对象来对相关联数据进行封装，界面交互组件实现交互控件与数据对象之间的同步功能。



### 平台界面交互

平台界面交互组件主要实现具体功能的交互控件与数据对象之间的交互功能。

### 功能服务

功能服务以soa组件的形式对业务功能进行封装并对外发布，具体的服务封装粒度与具体的需求有关。

#### GIS服务

利用已有的地图服务（百度地图、高德地图）开放平台集成相关功能。

#### 视频监控

利用已有的视频监控解决方案提供的功能接口集成相关功能。

### 应用服务器

应用服务器引用功能服务，并根据不同的平台和客户端提供具体的功能。平台至少需要提供对web访问的支持。

Tomcat是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，由Apache、Sun 和其他一些公司及个人共同开发而成。由于有了Sun 的参与和支持，最新的Servlet 和JSP 规范总是能在Tomcat 中得到体现，Tomcat 5支持最新的Servlet 2.4 和JSP 2.0 规范。因为Tomcat 技术先进、性能稳定，而且免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。

Tomcat 服务器是一个免费的开放源代码的Web 应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP 程序的首选。对于一个初学者来说，可以这样认为，当在一台机器上配置好Apache 服务器，可利用它响应HTML（标准通用标记语言下的一个应用）页面的访问请求。实际上Tomcat 部分是Apache 服务器的扩展，但它是独立运行的，所以当你运行tomcat 时，它实际上作为一个与Apache 独立的进程单独运行的。

Spring是一个开源框架，Spring是于2003 年兴起的一个轻量级的Java 开发框架，由Rod Johnson创建。简单来说，Spring是一个分层的JavaSE/EEfull-stack(一站式) 轻量级开源框架。

Struts是采用Java Servlet/JavaServer Pages技术，开发Web应用程序的开放源码的framework。采用Struts能开发出基于MVC(Model-View-Controller)设计模式的应用构架，用于快速开发Java　Web应用。Struts实现的重点在C(Controller)，包括ActionServlet/RequestProcessor和我们定制的Action,也为V(View)提供了一系列定制标签（Custom Tag）。但Struts几乎没有涉及M(Model),所以Struts可以采用JAVA实现的任何形式的商业逻辑。

# 数据库设计

污染源监控有别于一般的工业监控，对于每个监控项目，并不仅仅简单的监控其上下限，而是必须符合相应的环保标准和标准级别。由于环保标准等级繁多，形式也十分多样，所以配置和维护每个监控项目将十分困难，对于数据库的要求也比较高。

## 数据库体系结构

本系统采用关系数据库进行数据库设计。能够更为简单的部署、管理和优化企业数据和分析应用程序。它还提供了额外的备份和恢复功能。在数据库平台的安全模型上有了显著的增强，由于提供了更为精确和灵活的控制，数据安全更为严格。

整个数据库系统包括基础地理数据库、环保信息数据库、监测数据库和用户数据库四大部分。如图一所示。

1、基础地理数据库由数字地图数据组成。

2、环保信息数据库，主要包括企业基本信息数据、废水数据等数据，其数据种类有:

企业基本信息数据:包括企业名称、监测点编号、监测点基本信息、排污口信息、危险化学品信息、安装监控设备信息等。

废水数据:排放的废水种类等。

3、监测数据库存放采集到的、未经修改的污染源水质在线监测数据，如COD、总磷、氨氮、流量、等。

4、用户数据库包括用户名、密码、权限设置等。

## 商业数据库产品

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 最流行的关系型数据库管理系统，在 WEB 应用方面MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。

MySQL是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，它分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。