2011服务计算

1. 什么是面向服务的泛型，与面向结构、对象，基于构体相比的特点和优势。(PPT2)

答：服务：是自治、开放、自描述、与现实无关的网络构件。

面向服务的编程范型：指以服务的创建，服务的管理以及复用已有的服

务组装形成应用为基本活动的变成泛型。用过网络，使用标准方式互联。

命令式范型缺点：过程是对功能的抽象，而功能只能片面地反映问题空间事物的性质，这使得单纯用命令式范型开发大型程序时产生的代码不容易理解和维护。

基于构件VS面向对象：

|  |  |  |
| --- | --- | --- |
|  | **基于构件** | **面向对象** |
| 抽象视角 | 构件是对客观世界的实体或者实体联合能提供的功能和服务的建模；仅仅关注实体的功能和服务 | 对象是对客观世界基本实体的抽象，强调对实体的对应及对实体的建模；  涉及实体的静态属性特征 |
| 可复用程度和复用机制 | 以组合的方式实现复用 | 以继承的方式实现复用 |
| 粒度不同 | 大 | 小 |

面向服务VS面向对象：

|  |  |  |
| --- | --- | --- |
| **特点** | **面向对象的计算** | **面向服务计算** |
| 方法论 | 通过定义紧耦合的类来进行应用开发。应用架构为基于继承关系的层次式架构。  从构造函数——通过类或模型——到系统设计 | 通过定义松耦合的服务来进行应用开发，并将服务组装成可执行的应用。  从系统模型到服务模块，从服务抽象定义到服务实现绑定。通过搜索获得可用的服务实现 |
| 抽象和协作层次 | 往往由一个团队来负责应用的开发，并负责整个生命周期。开发者必须了解应用领域知识和编程。 | 开发任务由三个独立方承担：应用程序开发者，服务提供方和服务代理。其中，应用程序开发者需要了解应用逻辑，但不需要了解具体的服务是如何实现的。服务提供者需要编程能力，但不必了解使用服务的应用。 |
| 代码共享和复用 | 代码复用通过类成员的继承和库函数加以实现。其中库函数在编译时引入，且往往是平台相关的。 | 代码在服务层次复用。服务使用标准的结构，并发布在Internet库中。服务是平台无关的，且能够被查找并远程调用。服务代理支持系统的服务共享。 |
| 动态绑定和重新组合 | 在运行时将名称和方法进行关联。方法必须在应用部署前链接到可执行的代码。 | 在运行时将服务调用和服务进行绑定。可以在应用部署后，再进行服务选定。这一特色使得应用可以在运行时重组。 |

1. 与一般的软件测试相比，面向服务测试的特点，测试方法。（PPT14）

答：

测试特点：

* 基于规约的测试
  + 服务用户可能没看到或看不到源码
  + 服务提供者虽然看的见源码，可是不知道源码会被谁使用以及如何使用
  + 服务代理了解服务规约，但是依然不知道具体的使用方法和源码
* 动态性测试
* 协同测试
  + 协同验证与确认（CV&V）
  + 独立验证与确认（IV&V）

测试方法：

|  |  |  |
| --- | --- | --- |
| **测试方法** | **如何测试** | **应用范围** |
| 不测试 | 不做任何测试 | 一些对计算可信性要求不高的个人或小型应用 |
| “相信我，没错的！” | 用公司信誉作服务质量担保 | 1. 对日常的许多应用都是可以的，但不适用于对计算可信性较高的应用   2． 使用信誉或法律来保障服务软件的质量 |
| “统统告诉我‘代码’” | 测试一些可能的组合，仅使用那些通过测试的组合 | 1. 服务提供者愿意提供源代码 2. 评估比较客观 3. 真正进行测试以确保软件质量 4. 由于需要测试所有组合，且组合总数十分庞大，代价很昂贵 |
| CV&V | 用户选择进过排名的服务 | 1. SOA框架中的每一方（服务用户、服务中介和服务提供者）都需要参与到CV&V过程中来 2. 评估比较客观 3. 真正进行测试以确保软件的质量 4. 可以对新组合的应用程序进行运行时的测试 |

1. 服务，服务系统，举例。（PPT1）

答：服务举例：

* Traditional transportation and warehousing
* Information
* Finance, stock market
* Rental and leasing
* Scientific and technical services
* Administration
* Entertainments

服务系统举例：

1. 垂直服务、水平服务的区别和联系。（PPT3，PPT1）

答：五个水平层，四个垂直层



Horizontal layers implements functional requirements

Vertical layers provides system-support facilities and enablement

从消费者的角度来看,有许多服务可以被同时独立使用，所以称为垂直服务，分为纯粹的IT服务（pure）和IT服务。同时垂直服务可以由一些可重用的跨产业的通用服务建造，所以又称为水平服务。分为普通商务服务和IT服务。

**4个垂直层**:1integration layer:提供服务提供与请求者之间的协调,路由,服务要求的传输的关键支持2QoS layer:在多个方面提供了解决方案级别(不是服务级别)的QoS管理3data architecture layer:提供了与特殊领域数据架构集成的统一化的表示框架来推动价值链集成4governance layer:提供了能够保证SOA解决方案架构设计的指导**5水平层** 实现了功能需求:**一service provider tier:** (1operational system layer:包含了具有特定功能的现有的解决方案包,客户应用以及一流系统 2 service component layer:提供了在服务层定义了的服务实现3 service layer(包含在2个tier中):扩展SOA的三角模型为一个允许及推动服务注册,拆分,发现,绑定,接口聚合以及生命周期管理的综合性的逻辑层)**二service consumer tier:** (1 business process layer:管理所有被用作服务组合与拆分的业务逻辑2 consumer layer:为业务流程层,服务层等说明快速建立服务用户接口以满足客户需求.

1. SOA三角操作模型。技术，商业价值。（PPT3）

答：SOA三角操作模型：



服务提供者提供服务，并且把包含服务接口、服务访问地址等信息的服务描述以及服务的其他一些相关元信息发布到服务中心。

服务注册中心提供服务间寻址功能，是帮助服务消费者发现和定位合适服务的“中介服务”。服务注册中心剥离了服务消费者和服务提供者之间的服务寻址依赖，另外，在运行时通过约束条件在多个服务中选择了与条件最匹配的服务。

服务请求者可以通过服务注册中心间接获得服务描述，或者从服务提供者处直接获得服务描述，然后遵从服务描述的接口和地址约定实现和服务提供者所提供服务的交互。

（1）松耦合和复用性 （2）松耦合和灵活性

商业价值：（1）保护企业投资，提升现有IT资源的作用，促进IT资源复用 （2）提高企业的敏捷性 （3）支持企业外包管理模式

大题

1. WS—协议的作用。Web + service 建模，发布，查询，调用过程。（PPT5,6）

答：传输层提供了一系列定义了Web Service通信和交互的核心机制的协议。包括HTTP/HTTPS (超文本传输协议 /在安全套阶层之上的HTTP协议)、SMTP (简单邮件传输协议)、RMI/IIOP (远程方法调用/因特网互操作性协议)和 JMS (Java消息服务) 。RMI/IIOP/JMSRMI：使开发人员开发基于Java分布式技术的应用。使用RMI，远程java对象的方法可以被其他不同主机上的java客户调用。RMI使用对象序列化方法。IIOP是一个用于CORBA对象请求经纪人之间通信的传输协议。用于使不同语言编写的分布式程序通过互联网通信。JMS定义了可信赖的企业级消息的标准，该标准使基于J2EE的应用组件创建、发送、接受和读取消息。支持低耦合，可信赖和异步的分布式通信。

消息层（SOAP）的协议为编码Web Service消息定义了一种交互性机制。

Web服务事物管理定义了三个可被嵌入到服务协同框架事务协议，实现了跨现有事务管理的互操作性，长时间运行的补偿和异步商务过程流程。

Web Services：

建模：

1.define its interface in a standard way (WSDL & SOAP)

2.binding of WASL to SOAP

3.在注册表上发布Web Service

4.Stateful web service modeling：网络服务资源框架（WSRF）,粒度授权的状态管理

5.web services interoperability(互操作性) ：WSDL+SOAP+UDDI的结合:使用WSDL对web Service建模，把它绑定到SOAP，把服务发布到注册表上（如UDDI) 6.Modeling a Composite Web Service：多个网络服务需要彼此合作来建立一个综合的网络服务

7.三维网络服务建模：需要包括有关网络服务的三类信息：静态信息描述一个网络服务，动态信息描述网络服务的动态行为，关系信息描述网络服务和相关的服务提供者之间的关系。

发布： Web service是通过部署到连接到网络的应用服务器上实现的，之后将服务发布到网络上的服务注册表，因此所有用户都能够找到它。服务注册表提供<服务请求者需要的用来发现服务提供者和他们的web service的>信息，而不是具体实现。发布方式有两种，集中式发布UDDI和分布式发布WSIL。UDDI发布：1.UDDI 类似于一个集中式的仓库，其中保存有已经发布的网络服务的链接

2.并且还提供了对应的服务提供者的相关信息

3.还提供了SOAP接口使得基于消息的SOAP服务能够发布

4.根据访问控制政策的不同分为公共的和私有的。WSIL发布：Web servie 都是以XML文档的形式发布到一个定期的web server上的。WSIL是用来聚合从存在的服务描述文件的引用，聚合的方式可以是标准格式或者是特殊格式。在WSIL文档中，可以使用参考点来连接到一个发布到UDDI登记表中的服务或者连接到另外的WSIL文档中，从而形成了WSIL链

调用：SOAP主要支持两种交互模式：

1.远程过程调用模式：同步请求/响应交互方式，发送请求，并等待回应。

2.文件为导向的（Document-oriented）模式:异步交互方式，发出了一个完整的XML文件，并等待通知。结果将在处理后被送回。

1. 面向服务设计原则，举例。

答：

* 标准化服务合约
  + 实现一个标准化合约
* 服务松散耦合
  + 最小化依赖关系
* 服务抽象
  + 最小化元信息的可用性
* 服务可复用性
  + 实现通用的和可复用的逻辑与合约
* 服务自治
  + 实现独立的功能边界和运行时环境
* 服务无状态性
  + 实现可适应的，和状态管理无关的逻辑
* 服务可发现性
  + 实现可交流的元数据
* 服务可组合性

最大化可组合性