**服务计算**

PPT1.3，课本1234，整理：灰色字体和斜体为课本内容，黑色字体来自PPT+课本，红色字体为贝佳复习内容问题。

问题：（from beijia recorded by fanxin）

Chapter1 什么是服务，为什么要有服务计算

P14 lifecycle, 有一般的什么不同（技术上和过程上的差异）

P18 自治系统{垂直服务、水平服务} 搞清楚是什么，为什么要拆分成。。。

Chapter2 XML不考

Chapter3 建模协议 WSDL、SOAP等，不要写定义，要知道它们做什么，主要的结构，可以画张图描述

诸如P11的具体的语言不考，但可以用这种方式举例子，当描述不清的时候

P18 P21 P24 好像SOAP挺重要的

UDDI做什么 P31 UDDI包括。。。？ 地位？ 关系？

P34 P35 P42 BPEL做什么 结构、内容

注意：可以写一段XML帮助描述

P59 自己发挥（书上）

UDDI WSIL 发布、查询

USML AUSE 做什么

P88记住最主要的部分 解释一下这个结构

**PPT1服务计算（1。2章）**

* 服务计算是什么？为什么存在

**服务定义**:服务代表了一种基于至少一个服务供应商和一个服务消费者关系的交互（活动）,以达到某种特定的商业目标或解决方案的目的。在服务计算生命周期中，服务提供商承诺实现任务并提供给服务消费者价值，二者共享基于有效和有价值服务的健康、长期互信的共同目标。IT商务服务的**特点**:1服务运作模型(定义了服务怎样被送达)服务可以由不同方式实现，分别代表不同服务运作模型，如端到端模型说明服务提供者可以直接发送服务到特定的服务对象，其他如服务托管模型，业务流程外包，数据中心外包和网上经纪代理服务；2服务收费模型(关注服务怎样收费)免费的收费模式 收费模式 政府服务模式

**服务系统定义**:通过IT软件系统实现服务的系统，它可以被看做是自我限制的封装的调用服务给外部世界的系统，其实一个有反馈的系统，它处理内部控制并对周围环境作出响应。其包含**6个元素**:输入 输出 目标 变化 组件(服务人员 服务合作伙伴 服务信息 服务激活 服务基础设施资源) 传感器。服务系统生命周期中的**5种基本组成：**服务人，服务协作商，服务信息，服务活动，服务架构资源。**服务系统4个方面**:1模型(服务运作模型 服务提供模型 服务能力模型)2技术(web服务 soa 网格 自动化计算)3结构( 业务结构 数据结构 应用架构 性能架构)4 优化运筹学 复杂系统 系统工程。为了支持复杂全面的服务系统，从跨学科主题调用技术和方法是必须的。

**服务计算技术的出现**：面对未来商务动态性，服务计算技术提出，其目的在于再一个良好定义的架构下建立、操作、管理和优化这些服务来满足更高的灵活性。因此我们探究服务内部来建立一个服务科学技术的基础。此外，服务计算引入传统工业界促进了工业特定应用的内部和外部集成，SOA作为服务计算的一种架构在此种背景下行程，通过面向服务架构和WEB SERVICE技术，更多的企业可通过过一个在平台独立方式良好定义的借口发布自己的商务应用来提高和商务伙伴的应用的交互，这样使得整个商务体系形成一条合作链。IT架构模型转移到面向服务架构，同时商务模型也发展到基于组件来实现敏捷和被需求的业务。IT工业为了符合着这些潮流，一些常识性的工作形成。最早的服务计算规则最为第一次尝试在2003年提出。

服务计算覆盖了商务和IT 服务的各个方面。对于商务服务，服务计算覆盖了面向服务的商务咨询方法，公共设施，商务处理模型，转移，集成，商务性能管理和工业解决模式。对于IT服务，服务计算覆盖了架构服务和IT级别自制系统服务。

总体来说,服务计算作为一种新出现的跨学科的理论，覆盖了有效创建和分级计算到模式、创建、操作和管理商务服务的科学技术。核心技术组件包括SOA和网络服务，商务处理集成和商务性能管理、服务调用方法。

**服务计算的目标（PPT）**1复杂的服务体系2灵活的服务体系3专业化和外包模型4计算环境的演变4IT专家和商业专家的距5新的增值和创新性职能**商务服务的组件**:1 策略基础设施和产品 2与日常活动相关的服务运作 3企业管理

**服务计算：（PPT from 课本1章）**

1目的于建立、运行、管理和优化在一个良好定义框架下的进程，来提供更高的灵活性，满足商务多变性

2商务应用被通过练好定义的借口以独立于平台的方式发布来提高协作

3核心技术

SOA和WEB SERVICES

商务过程组装和商务性能管理

**服务计算：（课本2章）**跨学科领域正在快速出现：服务计算。现代服务产业已经现代服务信息社会加以关注和重视，然而当服务产业提出越来越多需求，如何改变现在的人力来提供符合要求的服务仍然是一个挑战，尽管找到各个领域的工作市场专家很容易，但是找到足够好的合格的面向服务的上午咨询师和面向服务解决方案的构建者仍然很难。一个新的面向对象的学科是必要的，我们称之为**服务计算**：“服务计算以填平商务服务和IT服务差距和继续加强商务服务为任务，覆盖了将计算和信息改变到建模、创建、操作和管理商务服务的科学技术。”服务计算需要从多种专家和规则到以计算和信息技术为核心的无缝协作。 服务计算作为现代服务科学的核心技术基础，描述了一种由新型SOA概念、咨询方式、发布平台和工具激发的新的领域。服务计算暗示了一个比SOA和WEBSERVICE更加广阔的领域，终极目标是研究如何以高效和有效的方式来利用IT和计算技术来创建、操纵和管理商务服务。

* 服务生命周期：及其和SDLC系统构建生命周期的比较

**典型服务生命周期包括6个阶段：**:1 咨询和策略计划 2 服务接触 3服务提供 4服务运作 5 服务收费 6 服务管理**服务生命周期的关键因素**1数据信息2过程3人(服务消费者 服务顾问)4 财政因素5 知识技能 6创新和科技

咨询和策略计划：邀请第三方咨询公司来进行策略计划

服务接触：可能性验证，RFP，服务需求商议，建立工作状态（SOW statement of work），和交流

服务提供：建立服务发送团队和管理项目组，解决方案创建，工作统计分析，微设计、发展和实现，测试和部署。

服务运作：呼叫中心管理，选择服务运作工具，商务和IT 性能监控，变更管理和问题管理，配置管理等其他IT服务管理功能

服务收费：服务提供商依据和客户的交流来收集支付

服务管理

* 服务和制造模型的差别

1.最大的差别在于输出的产品不同，服务模型输出服务，制造模型输出商品。服务和商品在性质上有不同定义，商品是一个可触知的物理实体或者可以被创建并运送的商品。服务时可被触知和易消亡的，是一个被创建和使用同步或者几乎同步的过程或事件。在服务被产生后消费者无法拥有服务本体，但可以拥有并维持服务效果。

传统制造模型中，制造活动的目标是创建商品来满足通用的目的。商品必须满足预先定义的规格，然而商品本身不直接解决用户问题，商品作为服务提供商和端用户的连接。

服务模型中，服务提供者和服务消费者在服务环境中有紧密的关系，服务提供者的环境被分为服务前阶段和后阶段两部分，在服务提供者和服务消费者见的交互通常发生在服务的前阶段，服务提供商目的在于提供良好的服务体验给消费者来提高消费者满意程度和吸引更多消费者，同时保持当前用户。服务前端利用了服务平台来收集服务需求或者其他包含在提前设计的服务表单中的内容来给消费者提供服务。服务后端，服务提供商利用工厂模型来实现子服务到高效生产率的无缝集成，通常的行为包括：为服务操作环境配置资源准备来实现服务消费者需求和防止欺骗，收费消费者，运行商务智能分析来进一步改进服务。

2.从生产生命周期的角度来看，服务消费者被涉及到服务的所有阶段（生产前，生产中和生产后），然而一个制造商品的消费者主要涉及到了生产后期阶段。商品消费者几乎与商品提供者在整个生命周期内无明显交互，然而服务消费者通常参与服务实现阶段，这种不同导致在改进制造或者服务系统时需采用不同的方式。

* 服务自制系统：垂直、水平。为何拆分成这些？
* From the view of consumers
  + There are many services can be utilized simultaneously and independently
  + So they are called vertical services
  + Divided into pure IT services and IT-enabled services
* Vertical services can be constructed with some reusable cross-industry common services
  + So they are called horizontal services
  + Divided into common business services and IT services

垂直服务：

从消费者的角度来说，有很多服务是被同时和独立使用的，他们被成文垂直服务，分为纯IT服务和IT使能服务。

水平服务：

垂直服务可以被一些可复用的跨工业的公共服务来构建，被称为水平服务，分为公共商务服务和IT服务。



**PPT3 建模 发布和搜索 （3.4章）**

**WebService 定义（以下用WS代替）:**网络服务是一个可编程独立于实现的模块,提供了标准的接口，通过标准的通信协议描述提供通用访问，实现与语言和平台无关的服务。

1. WS的功能可以用不同语言在不同平台上实现；
2. 可以被组合来建立具体领域的应用和解决方案；
3. 可以被视为OO设计中对象概念的延展(包含了商务罗辑 可以在网络上查找到 可以接入网络或局域网)，和类一样都封装了实现细节并且只被特定的接口访问。WEB SERVICE通过一种标准语言发布并只能通过特定协议访问。

1封装了功能实现细节并且只能通过开放的接口访问；

2通常有完整全面的商务逻辑；

3可以通过WEB查询；

4可以被通过Internet和Intranet访问；

WEB SERVICE模型（Paragram of WS）：

正在将互联网从WEB内容的存储库到服务的存储库的改变，通过以下三种方式

1 每个组织以服务方式公开商务应用到Internet和Intranet，使这些服务通过标准可编程的接口，WS模型为跨越企业界限的B2B合作提供了便捷

2 WS技术为分布式计算提供了统一的松散的框架，为分布式计算和资源网络共享增加了跨平台跨语言的互用性

3通过集成独立发布的WS组件为新的商务过程，WS模式开启了新的工程软件有效消耗方式来快速构建和部署WEB应用。

## 3章：WS网络服务建模

WS建模的方式：WSDL和SOAP做什么的？其结构、内容、操作方式？说不清楚可以用XML来列举。数据结构不考

WS Modeling方式：

1. **WSDL**

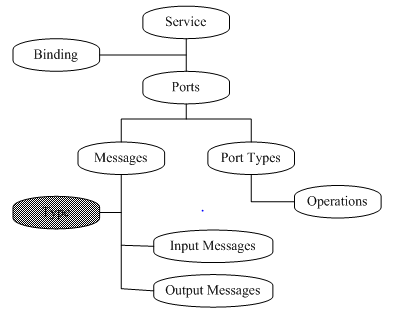
“一种XML格式,该格式将网络服务描述为一组在包含面向文档(异步)或面向过程(同步)的信息的端操作。”(WS：Web Service的简称)

WSDL定义了一个web服务的公共接口，包括功能点和怎样调用他们。它同时定义了与其他web服务交互的信息格式和协议邦定，信息格式定义了怎样翻译信息中的数据格式，协议邦定定义了怎样把信息映射到一个具体的网络传输端口，为同步（面向文档）和异步（面向过程）交互模式设计。

总之，WSDL回答了以下WHAT WHERE HOW的问题：服务关于*什么*？服务寄存于*哪里*？服务*如何*调用？。通常数据结构以XMLSchema被嵌入WSDL文件，WSDL用来连接SOAP和SMLSchema来定义Internet上的WS。用户程序阅读WSDL文档来理解WS做了什么，然后利用SOAP来确实调用在WSDL文档中被被列举的方法。这个过程是自动化的。

组成：

下图是WSDL的基本元素组成，WS被定义为一组服务接入点(端口)，端口有一系列的利用通用的绑定机制绑定到网络地址的端口类型，每个端口类型是可以通过信息访问的发布的操作，并被分类到包含参数的输入信息和包含结果的输出信息中。每个信息包含数据元素，每个数据元素属于一种数据类型。这种类型是XML Schema Definition(XSD)简化类型或者XSD复杂类型。XSD是WSDL经典的类型系统，然而，WSDL也允许其他的数据类型系统，如CORBA Interface Definition Language(IDL)数据类型。



**4种操作类型**:（从服务提供者的视角提供）

单向：单向操作接受信息但是不反馈信息

响应请求:（最常用）接受请求并反馈信息

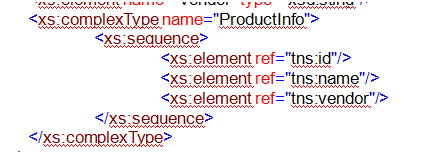
征求响应：（Solicit Response）：发送请求以得到反馈

通知：发送信息但是不等待响应

**数据类型：**

用作服务调用的信息，允许各种数据类型，主要基于XMLSchema，允许其他数据类型系统如(CORBA)，可以再分开的或者同一个文件内定义

XML Schema复杂类型，例如：



1. **SOAP**

定义：在WSDL为WS的借口建模后，WS对其他WS或者程序的访问和交互应该通过一种标准的交互协议。SOAP（Simple Object Access Protocal）正是这样一个WS交流的标准协议。

1. SOAP是一种简单而轻量级的协议，用于在WS之间交换结构化和规范化的信息，*它定义了WS交互的信息的模式.*
2. SOAP是基于XML的,并独立于任何操作系统、编程语言或分布式计算平台
3. SOAP需要绑定到现有的互联网协议,如的HTTP,SMTP，作为基础通信协议。

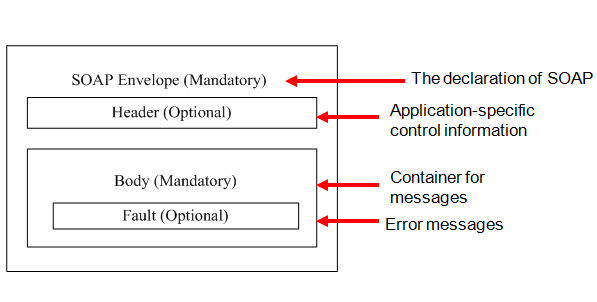
*总体来说，SOAP提供了一种运行在不同系统和通过不同技术和编程语言实现的WS之间的通信方式。*

支持的两种交互**模式**:远程过程调用 和面向文件的模式

远程过程调用（Remote Procedure Call RPC）是一种同步的请求/反馈的交互方式。请求者发送包含输入参数和输出变量的SOAP请求信息，这些数据被具体映射为一种单服务的操作，并且等待回应。收到服务请求后，服务调用相应的服务操作并回馈SOAP相应信息。

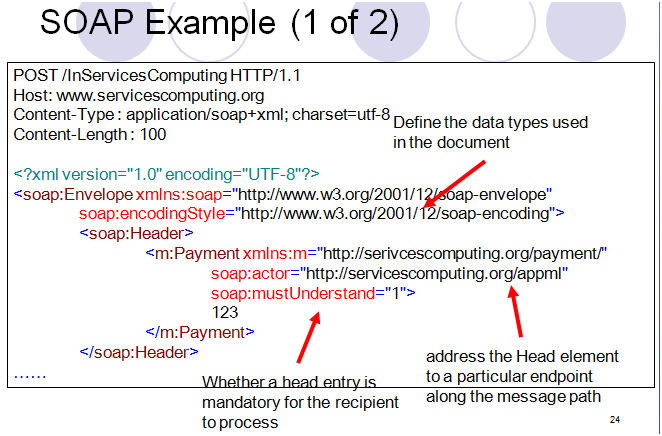
面向文件模式）（Document-oriented pattern）是一种异步的交互方式，一个服务请求者按照完整的XML文档发送SOAP信息请求完全处理，收到服务请求后，服务立刻回应确认信息，信息将会被异步处理，在整个进程结束时，结果会被返回到服务请求者。

SOAP结构：



基于SOAP的通信可以通过SOAP信息的方式实现，SOAP信息通常是XML格式的文本字符串文件，其包含四种元素：一个请求信封元素，代表了根元素和标示XML文档为SOAP信息；一个可选的头元素，包含关于特定SOAP信息的具体应用的控制信息；一个请求体元素，包含目标为终端端点的实际SOAP信息；一个可选的错误元素，作为SOAP体的一部分，标示错误信息。

下为实例：





* UDDI是做什么的？内容。2个东西在整体中地位？与上下层的关系？PPT30

**UDDI**

背景：在WS由WSDL建模后，需要对其他用户可用。通常，实现WS的第一步是布置应用到一个可以通过互联网访问的应用，然后，发布给网络上的服务登记这样它就可以被任何用户发现。服务登记不包含实际的WS实现，然而它提供了服务请求者发现WS服务商和WS所需要的信息，包括服务、服务提供商实现程序和相关描述服务的WSDL文件。实际WS实现程序和相关的WSDL文件通常定位在相应的服务提供者的服务器上并由服务提供者维护。

定义：Universal Description, Discovery, and Integration是一种通过大量查询海量元数据来发布和定位web服务的元服务的规格说明。

UDDI类似黄页清单，UDDI定义了一种保存注册的WS的XML信息格式的描述。UDDI记录了很多关于服务的信息来帮助服务请求者决定WHO WHAT WHERE 和HOW的答案。（UDDI定义一种机制来存储用XML信息注册的Web服务的描述 记录了wwwh）

包括四个关键元素:

商业实体，通过描述服务的信息来代表物理商务

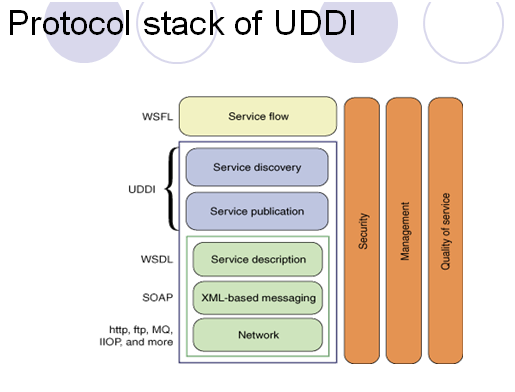
商业服务，代表商务提供的服务

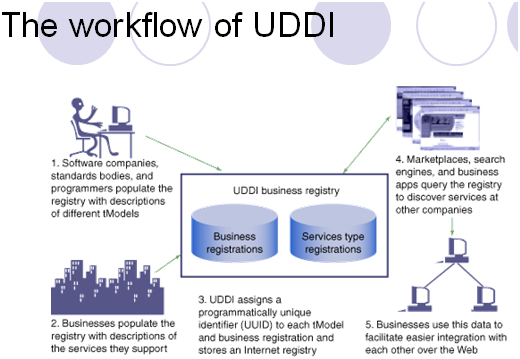
邦定模版，标示如何调用服务

技术模型，代表唯一的概念或者构造

使用XML schema描述其数据结构

课堂上补充：有可能考





* 带状态的服务：WSRF框架

**WSRF** web service resource framework 是一种基于XML的描述资源的方法.定义了一个使用web服务管理和接入稳定资源的系统的规格系统。

**包括四组规格**:使得通过WS接口访问资源的内部状态

1 WS资源性质 *WS-ResourceProperties*

2资源生命期 *WS-ResourceLifetime*

3基本错误 *WS-BaseFault*

4服务组*WS-ServiceGroup*

另外WSRF支持动态创建资源属性和关联值，WSRF描述如何通过WS接口访问WS资源的状态，定义了WS资源组织和定位的相关机制。

* BPEL核心元素

**BPEL**

Business Process Execution Language定义多种业务的协调互动,以实现一个共同的业务目标.精确地界定基本服务行为的跨企业的商业协议关注描述一个商务过程的行为,这个过程基于过程和合作伙伴的互动**建立商务过程模型的两种方式**1可执行的过程模型 为商务交互中参与者交互行为建模2抽象模型 显式描述没有显示其内在行为的参与者的行为 **用BPEL建立商务过程的两个阶段**:创建服务描述 创建商务过程

**BPEL的9个关键元素**

1Partners定义了商务过程中的关系

2partner link types 定义了两个WS间的对话关系，关于服务对话和提供的端口类型PortType（功能）中的角色。在伙伴连接类型中的角色标示了一个准确的WSDL端口模型。

3partner links 在商务过程中交互的伙伴服务，每个伙伴连接都以伙伴连接类型为特征，这个概念定义了在商务活动中关系的静态类型。

4business partners 两个商务伙伴之间的关系 为了给需要不止一个对话关系的商务伙伴间的关系建模，BPEL利用伙伴元素来标示在商务伙伴请求的容量，它允许伙伴联系的组织给予预期的商务企业关系，这一概念由商务连接的自己定义。伙伴定义不能被重载，意味着伙伴联系不能在多于一个伙伴定义中出现。

5endpoint references 动态邦定服务端口数据的机制：一个端点指使得BPEL动态选择一类特服务的提供商并调用其服务称为可能，实际服务到伙伴关系的绑定被端点引用来描述，在配置活动或者在活动中动态调用时，在BPEL活动实例中的每个伙伴角色被分配给唯一的端点引用。

6activities分为基础活动和结构化活动。基础活动是不可分的，如接受 回应 调用；机构化活动是按照预先定义顺序来组织商务过程的一组活动，三种结构：顺序（SEQUENCE WHILE SWITCH）同步（FLOW）随机选择（PICK）

7data handling 商务过程的状态 商务活动通常需要为有状态的交互建模，，包括收发的信息，其他相关数据。BPEL定义了三种对于有状态交互的处理方式：状态变量，表达式和作业ASSIGNMENT。变量被用来保持状态变量中商务过程的状态，表达式被用来从状态中提取数据并将其综合来控制过程的行为。作业可以被用来更新状态。BPEL为三种特性提供了XML的数据结构和WSDL得数据结构

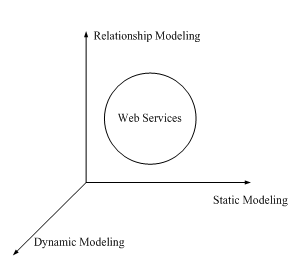
8Correlation关联 由处理在商务过程间长期有状态会话产生。信息关系被用来对应返回长时间商务过程给客户。当请求被伙伴解决，则有必要验证是否新的商务过程必须被实例化或者是否请求应该被导向现存的过程实例。

9Scope范围每个活动的上下文

* WSDL+BEPL？三维结构？做什么？结构？内容？

WSDL和BPEL能被用作单独WS和复合WS的建模。然而，WSDL和BPEL都是只关注描述WS的静态信息。WSDL描述了WS的抽象接口，和特殊信息形式和协议的绑定，以及服务的地址。BPEL描述了调用在一个商务过程中的WS组件间的关系。总之，他们提供了WS的静态建模

动态信息和关系建模： 一种三维模型被用来覆盖关于WS的三种类型信息，静态信息来描述WS，动态信息描述WS的动态行为，包括对于WS的历史调用和一些QOS的测算（在现实上在特定时间框架和成功的访问速度），关系信息表述了WS和其相应服务提供商的关系。这个三维描述模型的主要目的是：1帮助从WS登记中抽取或交换信息；12自动分析，聚集和索引WS。三维模型作为一个基础概念不仅表示了单独WS的语义信息，而其说明了WS的关系。另外，一个商务服务的数据库必须被建立作为一个可信的资源来承载目录信息和WS重复和推荐比例，这样来帮助服务请求者快速的访问和利用WS。



## 4章：WS网络服务发布和搜索

**Web服务的发布**:

UDDI

1UDDI的可被视为一个传统的黄页目录,分类,并组织出版的Web服务的各种组织形式;

2 UDDI的还提供相应的信息服务提供商,UDDI注册提供了一个SOAP接口,使SOAP消息为基础的服务发布被接入控制划分为公共和私有UDDI;

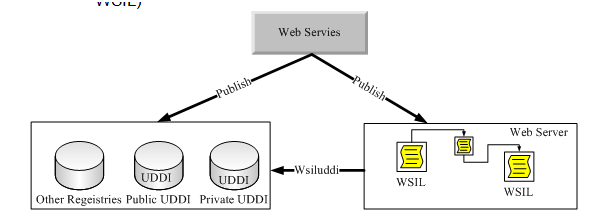
3集中式服务注册

WSIL

1 WSIL通过常规的ws发现部署调用Web服务,而不需要复杂的服务注册基础设施

2Web服务作为一个普通的XML文件发布到Web服务器上.Web服务检查语言( WSIL )是用来聚合现有的服务描述文件参考.参考指针可以用来连接一个发表在UDDI注册的服务或连接到另一个文件WSIL,所以WSIL链的形成

3分布式服务注册



* UDDI 和WSIL的发布查询？如何？
* Web Service 的发布和搜索

**发布有两种方式**

1 集中式服务发布：UDDI

2 分布式服务发布 WSIL

* 公共/私有的UDDI发布

1 UDDI 是一个集中式登记仓库，维护一组发布的WS的链接

2 并且还提供了对应的服务提供者的相关信息，如公司描述和项目建设组描述

3 还提供了一个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：公有UUDI允许公共访问，私有UDDI只允许特定用户有访问控制，服务提供商总是发布一个WS给现存的共有UDDI登记仓库，然而如果WS需要被内部用户秘密使用，最好的方式是建立私有UUDI

* WSIL 发布（web service inspection language ）

1Web servie 都是以XML文档的形式发布到一个定期的web server上的

2WSIL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

3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

* UDDIvsWSIL发布

主要区别在于花费和复杂度：选择UDDI和WSIL就像在黄页和四处查询间选择。

1 UDDI类似于传统的电话黄页查询，给不同组织发布的WS进行分类和组织；组织可以分享和利用UDDI登记库在不同的目录下维持一组WS，因此UDDI被认为是一个核心的构建块，它可能让组织间通过他们偏好的应用快速的容易的动态的定位和交互。

2 WSIL是一个组织间共享信息的更加廉价的解决方案。WSIL类似四处查询，基于XML文档，是通过web服务器来进行WS的搜索，部署和请求，并没有广泛的和复杂的登记设施。

* **简单的WS Discovery搜索/发现？**

WSIL通过链接连到其他的WSIL或者UDDI上；因此WSIL的搜索可以进行扩展

* UDDI 搜索

通常使用一个UDDI客户在UDDI登记处查找WS（UDDI for java 就是这样的一个客户，是一个IBM最早提出的开源项目，UDDI4J是一个JAVA类库提供给一个应用程序接口（API）以用来和UDDI登记存储库的交互，UDDI4J支持UDDI规格，SOAP传递，debug调试日志和配置容量）

有以下三个方向：

1. 搜索商务（允许根据一些标准寻找商务实体，可基于关键字，标识符，定位器、服务类型以及搜索的URL来搜索）

1 基于商务名称搜索

2 基于标识符搜索（唯一区别各实体的信息）

3 基于种类搜索 （UDDI将服务分成各类）

4 基于发现网址的搜索

1. 搜索服务;必须指定商务名称

1 基于服务名称的搜索

2 基于服务类型搜索

1. 搜索服务类型

1 基于服务类型名称

2 基于类型搜索

* WSIL搜索

WSIL文档提供了一种简单方便的方式，允许商务伙伴和提供者通过灵活的到其他发布服务的链接来发布自己的服务到网络上，是基于WSIL文档链而进行一个迭代的搜索过程，包括以下五个步骤：

S1 确定开始的WSIL文档的位置

S2 对指定的WSIL文档进行查找

S3 显示出在WSIL文档中包含的链接

S4 选择一个链接，加载选定WSIL文件的内容，如果被加载文件中还有其它链接，继续寻找更深的文档

5 重复3到4遍历所有相关链接直到相关信息被找到

显然为了发现服务WSIL通过手动引用所有的服务文档链，这样比较复杂。因此WSIL Crawling通常采用寻找工具在WSIL层的上层，例如java api的WS Inspection Language（WSIL4J）提供了可悲用来解析现存WSIL文档和程序化创建新WSIL文档的JAVA接口，大多数WSIL4J类表示在WSIL文档中出现的元素，一个WSIL代理类通常被用用于访问WSIL文档中特定类型的信息，一个人可以利用代理接口来浏览WSIL文档，然后直接访问感兴趣的UDDI商务服务。UDDI的操作提供给用户他们自己的WEB浏览器接口来允许用户从他们自己的UDDI登记中指定搜索标准。手动搜索比较费时，所以采用建立在WSIL的分析器上的搜索工具来进行

* 简单UDDI/WSIL搜索的讨论

利用UDDI，我们可以定位那些身份比较知名的商务服务，可以找到提供了什么商务和如何访问服务。然而简单的搜索机制有内在的问题：如效率、精确度、复杂度、内部交互。下为各自的缺点：

UDDI

1一个公共的UDDI登记包含众多的不同的实体，所以简单搜索不可能产生一组可控尺寸的结果集

2 简单UDDI需要服务请求者事先知道搜索的方面（商务、服务、服务类型）

3 创建者必须手动的书写复杂的搜索代码当查询请求是为了包含多元搜索标准或者跨越多重UDDI登记存储库；

4 基本的查询只支持特定类型的登记处查询

5 一个基础的UDDI查询只能基于一个方面进行查询

WSIL

1 当在链接文档中进行查找时，不能对已发现的服务进行聚合

2 对于来自不同数据源的搜索结果进行聚合，没有统一的发现机制

* 为什么存在USML?

USML(UDDI Search Makeup Language)UDDI搜索标记语言

USML是一种全面的基于XML的描述语言，目的是规范化UDDI搜索请求并使得复杂的UDDI搜索请求通过UDDI注册来得以使用。一个基于USML的搜索请求通常和多种搜索请求、UDDI源和集成操作交互，这样的请求根据多种搜索标准通过单UDDI或者多重UDDI注册来导航，搜索标准包括关键字、标示符、和分类。索索结果在返回到搜索请求方之前被集成。

UDDI规格中定义了三种可被请求的核心数据结构：商务、服务和服务类型（tModel）因此，USML的索索原则通常包括这三者。搜索类型被FindBy定义。

一个USML标记是一个包含了一组搜索请求的XML文档，么个搜索请求被何预先定义的标准关联，为了这些UMSL标记能够被服务登记存储库自动解析和处理，他们和描述每个请求和标准的数据结构的SCHEMA文档相关联。相关的XML Schema文件描述了WSML标记如何被解释。它定义了一个USML 寻找请求能包含的元素，如属性、值等。有效的USML文档必须遵守相关的XML Schema文档。

USML搜索标记和响应标记都被XML文档表示，因此为了USML响应能被自动和正确的解释，其必须和XMLSchema文档想关联。USML响应结构用XML文档的模式描述了UDDI搜索响应的结构，USML响应包括一组事务的列表，每个包含找到的商务信息

* AUSE的架构？



* **AUSE**

USML-based Advanced UDDI Search Engine

* 通过使用USML，请求者可以设计复杂的USML请求来对多个UDDI登记处基于特定搜索条件进行搜索。
* UDDI登记存储库使用AUSE来对USML进行解释和处理
* AUSE的目的是为了解释进来的USML请求，给相应的UDDI登记处分配和处理查询请求；然后将来自不同UDDI登记处的结果进行聚合；最后以USML的形式将结果返回给调用者
* AUSE的结构：

AUSE包括：

一个数据库元素（本地UDDI数据库）

作为本地快速服务的缓存，通常本地UDDI数据库上的商店和识别发布UDDI分类信息基于挂载的知识和自动更新机制

五个执行元素：

通知代理商（Notification Broker）

USML解析器 （USML Parser）

搜索命令构造器（Search Command Constructor）

UDDI源分发器 （UDDI Source Dispatcher）

信息集成器（Information Aggregator）

通知代理商允许对于搜索请求者的匿名搜索模式，当搜索请求者提交一个USML请求，现在通知代理商登记，确认信息立即被发送回服务请求者，请求者可能并不在线等待搜索结果。 在信息集成器完成来自不同UDDI登记的搜索结果的集成，它发送一个（RAN:Results Available Notice结果可用通知）给通知代理商，通知代理商通知搜索请求者。此后，应用中的接受者可以安排在方便的时间接受来自AUSE的结果。

接受到USML搜索请求后，通知代理商转向USML解析器，USML解析器根据需USML结构文档解释搜索标记，如果搜索标记是有效的XML文档，则转发到索索命令构造器。

索索命令构造器根据来自USML的结果集成单独的UDDI搜索命令并转发给UDDI源分发器。

接受到搜索请求后，UDDI源分发器首先搜索本地UDDI数据库，如果信息被找到，结果会不通过加载远程UDDI登记而很快送回，同时当搜索命令执行时本地UDDI数据库即时更新。成功的搜索结果通常被存储于到本地UDDI数据库里方便日后使用。如果数据没有找到，则UDDI资源分发器根据从本地UDDI数据库得到的UDDI分类信息，分发搜索请求到相应的UDDI登记存储库。本地UDDI数据库也通过自己的更新机制在编程时间内更新，自动发送对UDDI登记存储库有效的搜索命令并组织结果以一个良好定义的格式返回。UDDI资源分发器是一个只能组件，他能够动态分发构造的UDDI搜索命令给之前选择的USML请求。如果USML请求中未定义源信息，则UDDI源分配器自动分配UDDI搜索命令道一个已知的基于知识基础和智能的UDDI登记库。

来自多个UDDI登记库的搜索结果被信息集成器集成，基于集成操作和预先定义的规则解析并识别反复回结果，最后结果由基于XML的USML响应来标示。一个瀑布式（级）搜索机制被用来在不同粒度上优化搜索结果，过滤和集成机制被用于搜索结果从不同UDDI登记库的返回。

* 基于AUSE的搜索过程：12

S1：服务请求方产生一个USML请求，发送到AUSE ENGINE引擎，请求被在通知代理商注册；

S2：通知代理商返回确认信息到服务请求方；

S3：通知代理商发送索索请求道USML解析器来解释请求；

S4：USML解释器根据对应的USML结构激活搜索标记，将搜索标记转发给搜索命令构造器；

S5：搜索命令构造器产生一个或多个UDDI搜索请求，然后转发给UDDI搜索分发器；

S6：UDDI搜索分配器首先查看本地UDDI数据库，如果信息未找到，则擦哈讯请求UDDI登记的登记库和访问信息是否可用，搜索请求被信息集成器保存到数据库为了日后访问。

S7：UDDI搜索分发器分发搜索请求道对应的UDDI登记库；

S8：信息分发器接受来自本地UDDI数据库的信息，基于USML中定义的集成操作来侦听和组合来自对应UDDI登记库德搜索结果，然后组合集成搜索结果成为一个基于XML的响应；

S9：信息集成器发送结果可用通报（RAN: Result Available Notice）信息到通知代理商，告知对应搜索结果的可用性；

S10：通知代理商发送同通知信息到服务请求商

S11：服务请求商试图从信息集成器中提取搜索结果；

S12：信息集成器以基于XML的USML响应的格式发送搜索结果回搜索请求方

* AUSE提高效率的四个方式:

1 AUSE机制通过利用仅有的基于USML的查询请求和基于XML的对服务请求者的响应来自动减少来自服务请求者观点的网络拥塞

2通过避免掌握UDDI对不同注册表的搜索编程技巧来简化开发过程

3级串连的搜索机制从不同粒度上精简结果

4本地UDDI据库被用于提供快速服务的缓存

* DSDF



DSDF：（Dynamic Service Discovery Framework）

一种面向WSIL的先进搜索技术。WSIL文档通常连接在一起并嵌套多级到WSIL链中，通过WSIL链惊醒自动搜索显然耗时而且易错。因此面向WSIL的动态服务搜索框架（DSDF）被引入来提供一种在WSIL链中自动搜索WS，集成在每个WSIL文档中找到的WS和返回给请求方以及时反馈的机制。如图，一个服务请求方，通过程序或者WEB浏览器发送给DSDF一个请求，DSDF在网络上寻找合适的来自特定WEB连接的WSIL链以找到合适可用的服务，DSDF还提供了一种集成机制来组合所有的在每个WSIL文档中发现的WS，一次性返回组织好的结果给搜索请求方。因此，服务请求者从自动跟踪到候选发布的WS的WSIL文档中发布。

DSDF的结构

1 服务描述文件的搜索引擎 ：提供自动和深度搜索关联的服务描述文档，引擎采用了深度优先算法来遍历WSIL文档以找到合格的服务。

2 服务容器 ：存储缓存的每个服务描述文档链和链中WS的内容，通过链交换查看引擎发现和使用，至少服务名称和来源被定义为在合适的服务容器中的WS。容器也保存缓存的最近最常访问的服务描述文档链和联众特定WS的数据和元数据，因此减少频繁链访问。

包含了

1 有关服务描述文件链和包含在链中的WS的缓存信息

两种信息：类别的名称和类别的描述

相关的元数据：创建时间，文档大小 名称 位置

2 从外部组件获取请求的功能

3 值链变化侦测引擎：

提供了可以利用变量属性自动侦测服务描述文件中变化的缓存容量，查询基于时间初始的基础而非在服务容器中做内容缓存来查看服务描述文档链

4 控制参数

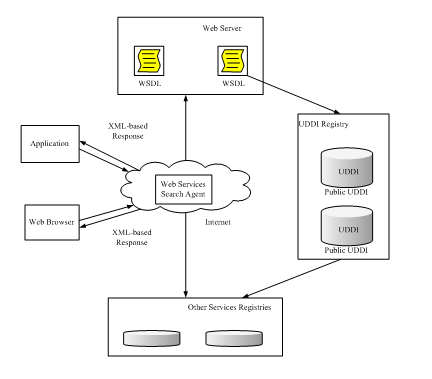
用来建立服务描述侦测引擎的初始配置数据集

* 实现

例如IBM的WSIL查询工具：WSIL Explorer 自动从文档连的跟开始遍历整个WSIL文档链，深入到所有链接的WSIL文件不管他们位置在本地还是远程，集成所有结果到一个XML文档作为返回

* PPT99 将UDDI WSDL放在一起怎样？ 联邦网络搜索框架
* 联合的搜索框架federated web service discovery framework

能够对基于UDDI和WSIL 的搜索建立统一接口，并且隐藏UDDI 和WSIL的复杂性和不同，这样基于UDDI登记和WSIL链的搜索对开发者透明，因此应用开发者可以对任何服务的搜索使用相同的接口并且在统一文档中书写查询语句；联合查询可以在将结果返回给用户前将结果进行存放和聚合。架构：



Federated WS查询代理实现了综合结果的聚合机制，和多重UDDI登记和WSIL文档交互。当服务请求方查找WS时，搜索代理采用一高中或者多种来自UDDI登记库或者WSIL文档链的基础数据结构回应。最终回应给搜索请求方的可能是新的XML机构或者现存的XML结构如WSIL，标准响应信息允许缓存搜索数据来加强搜索效率，同时，WS搜索代理可能自动调用选定的WS来获得确实的结果或者只是访问WS的容量。

联合搜索框架和网络搜索引擎的对比：

|  |  |  |
| --- | --- | --- |
|  | Federated Search framework | Web search engine |
| Usage | Unified search engine for Web services discovery | Unified search engine for Web content discovery |
| Search Targets | Multiple services registries | Multiple Web sites |
| Support interfaces | Webs services/HTTP | HTTP/Web services |
| Output | XML | HTML/XML |
| Search criteria | Combinations of three types of search options | Keywords (not sensitive to Web services) |

PPT 4567 课本5678整理：P多少指的是那一章的PPT的页码, Chapter也都是PPT的

Chapter4 SOA为什么存在 2级——>3级 好处、优势

Meta。。不考

P17 SOA到底怎么做的 要求 Business/IT

P18 做什么

P27 每一层做什么

Chapter5

服务的关系为什么需要Model

基于接口的服务发现时面向功能的，但在现实世界的商业中服务发现还要考虑商业功能之外的因素，其中商业关系就是一个很重要的因素。服务的关系在商业服务发现和整合上有重要作用。通过描述商业关系，服务注册支持更精确地服务发现。在基于SOA的方案中，服务和服务关系会变化，需要使用关系模型。

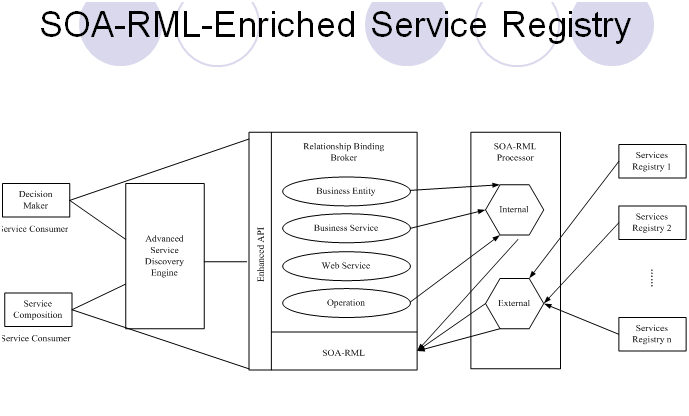
如何有效的服务发现

服务关系可以提供充分的信息使服务发现的判断更加精确。另外，服务关系模型可以提高服务注册的性能，使服务注册获得更加综合的信息并且根据服务的特征和服务之间的关系更好的把已注册的服务分类。

层次式SRM 关系 P28

|  |  |  |  |
| --- | --- | --- | --- |
| **Relationship**  **categories** | **Relationship sub-categories** | | **Comments** |
| **Business** | **IT** |
| B-B-R | partnership, parent-child, alliance | Exclusion |  |
| BS-BS-R |  | exclusion, parent-child, binding, community | intra/inter |
| WS-WS-R |  | parent-child, exclusion, binding, community |  |
| O-O-R |  | constraint, exclusion, cluster, parent-child |  |
| B-BS-R |  | containment, consumption, none |  |
| B-WS-R | Ownership | None |  |
| B-O-R | Ownership | None |  |
| BS-WS-R |  | implementation, none |  |
| WS-O-R |  | Inclusion, none |  |

P29



这张图表述了基于SOA-RML的服务注册。

三部分：1. RBB提供用于收集服务和发现服务消费者的请求。

2. 内部SOA-RML Processor在注册内部使用RBB收集的商业实体、商业服务、网络服务和操作的信息，建立关系的SOA-RML表示。外部SOA-RML Processor从外部服务注册获取信息用SOA-RML建立关系的表示.

3. ASDE(例如AUSE)提供给应用开发者从一个或多个服务注册搜索商业和服务信息的标准接口。

Chapter6 服务层次的QoS

P6 几个东西怎么拼在一起



WS-Security是一个协议族，用来增强通信技术，解决三个关于网络服务的质量保护的基本问题：对用户的身份验证和授权、消息整合、消息加密。

基于这些协议产生了六个模型，WS-Trust定义方法来请求和发出安全标志，建立信用关系。WS-Privacy表达隐私声明。WS-Authorization定义了网络服务如何管理授权数据和策略。WS-SecureConversation为安全通信定义了一个安全性上下文。WS-Federation定义联合验证、授权、账户等不同信用领域的机制。WS-Policy用来详细说明网络服务的终端策略。包括四个元素WS-Security Policy,WS-Policy Framework, WS-Policy-Attachment和WS-Policy-Assertions.

服务层面怎么保证QoS

QoS在服务的层面提出了四类标准，这四类是security，transaction, reliable messaging, resource lifetime management，这些标准帮助开发者和服务提供者在消息层面和交易层面增强网络服务的质量。这些标准关注部分QoS的属性例如安全性和可靠性，没有包括全面的QoS的属性，例如没有包括性能、可。这一层的QoS没有为不同粒度水平提供整合的QoS控制。

Chapter7 Requirement Driven Service Composition是个什么过程 用什么手段

服务计算的目标之一就是满足用户的需求。RDSC就是发现已有的商业服务，把它们组织起来、重新配置，组成新的商业过程来满足用户的需求。

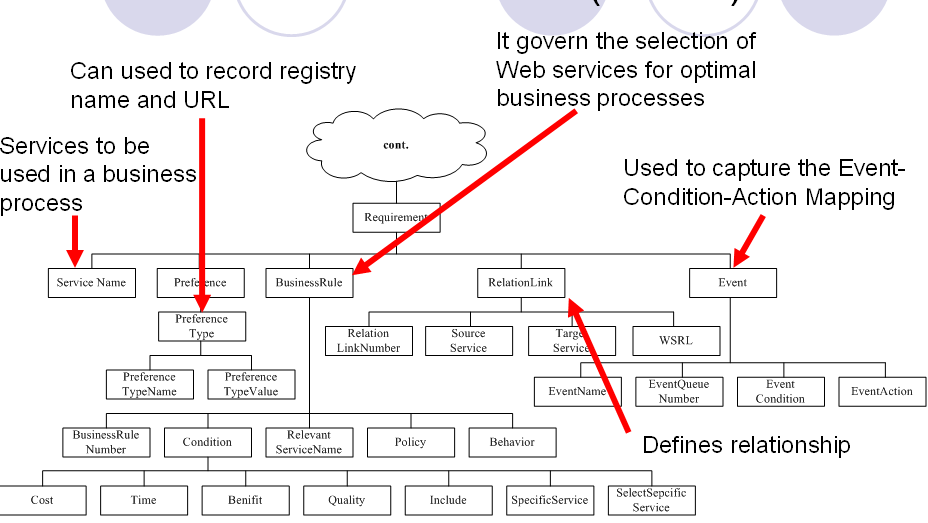
RDSC的过程：

1. 商业需求建模：包括目标组件和环境（用户接口、数据、功能/服务组件、事件和消息、环境），资产生命周期管理，项目管理，财政管理



1. 商业需求模型的表示：基于XML的BPOL（Business Process Outsourcing Language）下图是BPOL的数据结构，





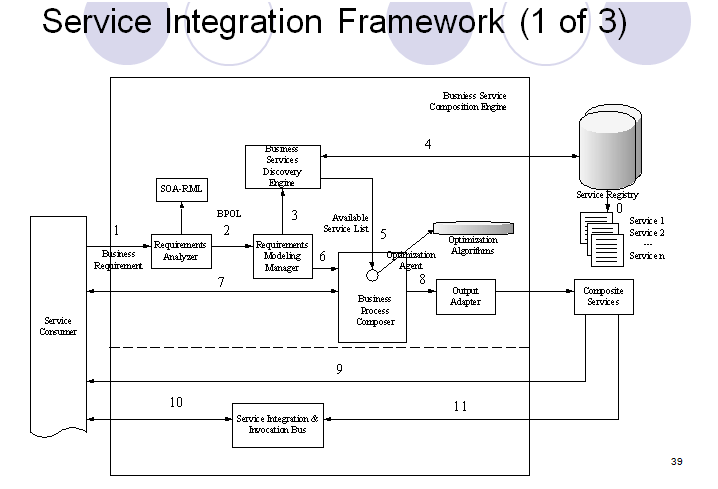
1. 需求驱动的服务发现：使用USML(UDDI Search Markup Language)。
2. 最优化商业服务组合，包括形式化服务组合，应用最优化算法

USML

用于定义多个注册的多个查询并把结果汇集起来.

为了使自动服务发现变得容易，很有必要把商业需求规格说明文档转化成USML查询。商业需求模型中的商业规则可以帮助产生更加精确的USML脚本。

P39



* Step 0: Business services are published to centralized service repositories, i.e., services registers.
* Step 1: Business requirements are inputted, either in a user interactive way or in the form of XML documents, to the Requirement Analyzer. Flow rules, references, and business relationships are described
* Step 2: the Requirement Analyzer creates business requirements documents based on the services composition requirement and SOA-RML annotations
* Step 3: The Requirement Modeling Manager parses the generated business requirements documents, automatically generates XML-based search scripts in USML, and passes the scripts to the Business Services Discovery Engine
* Step 4: The Business Services Discovery Engine conducts search processes across multiple service registries, based on the search criteria specified in the USML scripts created. The aggregated result is a list of available services that meet the search criteria
* Step 5: The available service list is passed to the Business Process Composer for service composition
* Step 6: The Optimization Agent in Business Process Composer use SOA-RML documents as input to extract the business flow rules, and drafts a business process using the selected services from the candidate lists
* Step 7: Users can interact with the Business Process Composer via Web Browsers or GUIs to tune the service selections and composition process
* Step 8: The result of Business Process Composer is formatted in target composite services via the Output Adaptor
* Step 9: The resulting composite services are returned to the Service Consumer
* Step 10: When the users want to access and use the newly composed business process, a Service Integration and Invocation Bus dynamically invokes the respective services, which are part of the newly created business process
* Step 11: The Service Integration and Invocation Bus accesses composite services

Chapter8

现代企业通常需要和服务提供者、伙伴和客户在价值链上为了一个共同目标而合作。

P36

(帮助理解)WS-Collab协作原语：任何商业协作都是在协作信息的交换中进行的，例如添加新的参与者、启动协作任务。为了使协作者之间的通信更加容易。WS-Collab创造了一个基于统一消息的通信方式。

(帮助理解)WS-Collab消息：WS-Collab定义了消息的结构的模板。消息通常包括几个部分，<transaction>和<task>是必须要有的部分。

(帮助理解)WS-Collab构造：用BPEL描述

(帮助理解)eBC ontology: 提供对商业合作过程中的信息的理解和解释的基础。定义共享知识。

解释协议栈

CxP：基于eBC本体，CxP定义了一个基本消息和复合消息的集合，这些消息可能在协作商业活动中的多个服务参与者之间被交换。CxP是一个面向商业目标的协议，支持各种商业构造和多种消息组合，以适应一个写作商业过程生命周期里的变化。

基于协作本体（collaboration ontology）、协作原语(collaboration primitive)和协作构造(collaboration construct)，WS-Collab建立了一个协作交换协议栈用来注解商业过程。

协作交换协议允许协作进程之间对等的内部行为，这一点很重要。协作交换协议还建立起IT传输层和商业场景之间的桥梁。协作交换协议还可被用于传播语义表示，控制信息交换流程，用动态的方式监控正在进行的活动。

协议栈有哪些东西：协作原语、协作消息、协作构造。协作构造用BPEL描述，协作消息用WSRF描述。

Chapter9 非重点（应该不考）

Chapter10

为什么网格计算要单独存在

为什么有了服务计算还要有网格计算

（理解）OGSA的概念：是一个用WSRF表示的，面向网格服务的分布式的交互和计算体系结构。主要目标坏死确保异性环境的互操作性，这样不同类型的系统就可以通信、共享资源。OGSA最核心的思想是使用网络服务的概念描述网格服务。

OGSA做什么？

1. 描述和定义一个基于网络服务的网格体系结构，使异性环境中分布式资源的共享和访问容易一些。
2. OGSA依赖用WSDL和WSRF描述的网格服务的定义。
3. OGSA由一系列网格服务组成，每一个网格服务都被标准接口表示。
4. 具有标准接口的网格服务可以在异性环境中被实现和部署，并且可以彼此之间通过交换基于SOAP协议的消息进行交互。

OGSA由网格服务组成，OGSA定义了一个网格服务实例的语义。包括服务功能、服务版本等。对于一个网格服务，每一次网格客户的调用都会产生一个运行时服务实例。通常，一个新的网格服务的实例被创造的同时会在宿主环境中产生一个新的进程，进程的职责是确保服务服从被定义的网格服务的语义。（OGSA含有类，类的接口定义良好，客户调用类是，OGSA生成类的对象，赋予这个对象一定的属性，对象的生命周期被一个进程监控）

OGSA还可以通过服务的版本信息管理网格服务的兼容性。

OGSA提供捕获网格服务实例的操作失败的状态信息的机制，这样操作失败时实例自动消亡然后释放资源。

OGSA部署实例



（理解）OGSI（Open Grid Services Infrastructure）:一个服务只有遵循OGSI的规范才是网格服务。

P18 区别和联系

区别：网络服务是基于上下文的，网格服务每一次别调用时都湖产生一个服务实例。

联系：

P31 架构 什么？

P39 数据流的走向有什么不同

有网格计算在有什么好处

为什么网格计算需要服务的知识

工作流