***学术型硕士博士（打印时删除）***

****

隶书，二号，加粗居中，单倍行距，段前0.5行

选填：硕士或者博士

校标尺寸(cm)：

2.6×10.0，居中

**硕/博士学位论文**

黑体，二号，加粗居中，单倍行距

**中文论文题目**

仿宋，三号，单倍行距，缩进4.5字符

姓 名：

学 号：

所在院系：

学科门类：

学科专业：

指导教师：

副指导教师：（如有副导师或校外导师，可在此填写）

二〇一〇年五月

宋体，三号，单倍行距，居中

***职业型硕士博士（打印时删除）***

****

隶书，二号，加粗居中，单倍行距，段前0.5行

选填：硕士或者博士

校标尺寸(cm)：

2.6×10.0，居中

**硕/博士学位论文**

**（专业学位）**

隶书，四号，加粗居中，单倍行距

黑体，二号，加粗居中，单倍行距

**中文论文题目**

仿宋，三号，单倍行距，缩进4.5字符

姓 名：

学 号：

所在院系：

职业类型：

专业领域：

指导教师：

副指导教师：（如有副导师或校外导师，可在此填写）

二〇一〇年五月

宋体，三号，单倍行距，居中

***同等学力硕士博士（打印时删除）***

****

隶书，二号，加粗居中，单倍行距，段前0.5行

选填：硕士或者博士

校标尺寸(cm)：

2.6×10.0，居中

**硕/博士学位论文**

**（同等学力）**

隶书，四号，加粗居中，单倍行距

黑体，二号，加粗居中，单倍行距

**中文论文题目**

姓 名：

学 号：

所在院系：

学科门类：

学科专业：

指导教师：

副指导教师：（如有副导师或校外导师，可在此填写）

二〇一〇年五月

仿宋，三号，单倍行距，缩进4.5字符

宋体，三号，单倍行距，居中

校标尺寸(cm)：

2.6×10.0，居中

****

A dissertation submitted to

Tongji University in conformity with the requirements for

the degree of Doctor of Philosophy

Times New Roman体，四号，居中，单倍行距，段前0磅，段后0磅

Arial体，小二，加粗居中，单倍行距

**Title**

Candidate:

Student Number:

School/Department:

Discipline:

Major:

Supervisor:

Times New Roman体，三号，单倍行距，缩进4.5字符

Times New Roman体,三号，居中

May, 2010

|  |
| --- |
| 5cm左右  **中**  **文**  **题**  仿宋，四号，加粗，行距16磅，段前段后0磅  **目**  **姓**  **名**  **同济大学**  5cm左右 |

此页可下载

**学位论文版权使用授权书**

本人完全了解同济大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版本；学校有权保存学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构送交论文的复印件和电子版；在不以赢利为目的的前提下，学校可以适当复制论文的部分或全部内容用于学术活动。

学位论文作者签名：

年 月 日

**同济大学学位论文原创性声明**

此页可下载

本人郑重声明：所呈交的学位论文，是本人在导师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年 月 日

**摘要**

宋体，五号，Abstract部分用Times New Roman ，五号

标题：黑体，三号，加粗居中，单倍行距，段前24磅，段后18磅

内容：宋体，小四，行距20磅，段前段后0磅，首行缩进2字符

在实际工程结构的服役过程中，由于非线性与随机性的耦合作用，工程结构特别是混凝土结构的非线性反应具有不可精确预测的性质。因此，从概率密度演化的角度考察工程结构的非线性性状是准确把握结构非线性性能的必由之路。本文基于随机结构反应概率密度演化的思想对于随机结构分析理论进行了深入的探讨，初步建立了随机结构反应概率密度演化的基本图景。

结构静力非线性分析是评价结构抗震性能的重要手段。对于具有双线型广义随机本构关系材料的结构，其塑性截面分布状态的演化过程即非线性损伤构形状态转移过程反映了结构内力演化的性质。无记忆特性结构的非线性损伤构形状态转移过程具有马尔可夫性，通过结构的力学分析可建立风险率函数与状态转移速率之间的关系，进一步考虑状态之间的逻辑关系，即可得到概率转移速率矩阵。对于有记忆特性结构及力-状态联合演化过程，可通过引入相应的记忆变量构造向量马尔可夫过程，并采用次序分析方法建立其确定性的概率密度演化方程。关于简单结构的情况进行了解析求解，并据以探讨了结构非线性构形状态演化的若干特征，发现了在实际应用中可能具有重要意义的稳定构形现象。讨论了力-状态的解耦问题。基于非线性构形状态本身的性质以及演化过程的规律，初步研究了可能的简化与近似方法。

……

最后，关于进一步工作的方向进行了简要的讨论。

宋体，小四，行距20磅，段前段后0磅，“关键词”三字加粗

**关键词**：随机结构，马尔可夫过程，非线性构形状态，差分方法

**ABSTRACT**

标题：Arial ，三号，加粗居中，单倍行距，段前24磅，段后18磅

内容：Times New Roman ，小四，行距20磅，段前段后0磅

In practical engineering, the structures usually exhibits strong nonlinearity coupled with randomness of the involved parameters. This makes it almost impossible to exactly predict nonlinear response of the structures, particularly for the concrete structures. To tackle the difficulty, it is necessary to capture the nonlinear performance of the structures in the sense of probability, instead of purely deterministic standpoint. The present thesis is the result of the efforts devoted to developing the probability density evolution method for analysis of nonlinear stochastic structures.

……

In the finality, the problems requiring further studies are discussed.

**Key Words:** stochastic structure, Markov process, nonlinear configuration state, difference method

Times New Roman 小四，行距20磅，段前段后0磅，“Key Words”两词加粗

**目录**

标题：黑体，三号，加粗居中，单倍行距，段前24磅，段后18磅

标题：宋体，小四号，行距18磅,，段前段后0磅，两端对齐，页码右对齐

[第1章 引言.........................................................1](#_Toc93734158)

[1.1 概述..........................................................1](#_Toc93734159)

[1.2 随机结构分析现状..............................................1](#_Toc93734160)

[1.2.1 线性随机结构分析.........................................1](#_Toc93734161)

......

[第3章 结构非线性损伤构形状态的随机演化分析.........................3](#_Toc93734162)

......

[3.2 结构非线性构形状态转移过程及其演化方程........................3](#_Toc93734163)

[3.2.1 结构非线性构形状态转移过程分析............................3](#_Toc93734164)

......

[第7章 结论与展望...................................................5](#_Toc93734165)

[7.1 结论..........................................................5](#_Toc93734166)

[7.2 进一步工作的方向..............................................5](#_Toc93734167)

[致谢................................................................6](#_Toc93734168)

[参考文献............................................................7](#_Toc93734169)

[附录A ××××......................................................8](#_Toc93734170)

[个人简历、在读期间发表的学术论文与研究成果..........................9](#_Toc93734171)

第1章 引言

1.1 概述

j0205582随着互联网技术的迅猛发展，越来越多的企业使用web服务作为企业应用的开放接口，越来越多的web服务被开发并发布。企业内部使用web服务作为各个功能组件间沟通的桥梁，以根据需求通过网络对松散耦合的粗粒度应用组件进行分布式部署、组合和使用。同时，大量互联网企业将自身功能通过开放API暴露在互联网上，用户通过使用这些基于SOAP或REST的API来使用这些企业提供的服务。终端用户通过组合这些服务来构成一个新的应用已经成为一件比较容易的工作，如下图所示，通过使用GitHub（提供代码托管服务）和travis-ci（提供代码构建服务）提供的API，终端用户可以自己完成一个持续发布（Continuous Delivery）的平台。

用户通过使用GitHub的RESTful的API将代码提交到GitHub的库里，同时使用API为该代码打上特定标签，之后调用travis-ci的API对GitHub内的代码进行构建，验证Test case，构建成功之后travis-ci将调用GitHub的API将构建好的代码即发布包上传到GitHub的发布目录下，至此，用户通过简单的API调用为自己创造了一个持续发布的系统。随着云计算时代的来临，服务更是成为其核心概念，web服务更是成为其中不可缺少的一部分。纵观现在所有的云计算系统，开放的web服务都成为其标配组件，如IaaS（Infrastructure as a Service）的行业先驱Amazon的EC2，他的AWS（Amazon web serivce）能够提供一整套的管理虚拟机的web服务。因此，通过将不同企业提供的web服务组合成流程来构建一个全新的应用程序成为一种新的应用模式。

现在，BPEL，BPMN，BPML 等服务流程的建模语言发展越来越成熟，业务人员的建模水平不断提高，业务流程的验证手段也不断发展，由于业务流程逻辑设计不当而产生的流程失效情况越来越少。然而，由于各种因素，如地域、价格或其他因素使得某个服务不可用或达不到应用的要求导致整个流程失败也是这样的构建方式的缺点。在当前互联网上，某些种类的服务通常有很多的服务提供商，他们提供服务相同或类似，不同的是服务的QoS（Quality of service，服务质量）。如上文提到的GitHub，相类似的代码托管服务商在全世界范围内有很多家，但是在中国境内能使用及价格上比较有竞争力的只有区区几家。如果能在组合服务的时候选择更好的服务，或者如何能在服务运行过程中动态换掉不能达到服务质量要求的服务，能极大的增强整个流程的可用性。

1.2 国内外研究现状

二级标题：黑体，四号，顶左，单倍行距，段前12磅，段后6磅，序号与题名间空一个字符

1.2.1 web服务的QoS评估

由于现有web服务实现技术的制约以及受到运行环境等多种因素的影响，在实际使用中，web服务可能无法提供其所声明的服务质量，形成复合服务的执行瓶颈。因此需要使用QoS评估技术对服务提供的服务质量进行评估以及时发现瓶颈．动态选择替换满足指定QoS需求的服务参与协作。

目前，在web服务QoS评估方面已经有很多研究，一个典型的Web服务QoS评估过程如下：输入为一组QoS约束和备选服务集，根据服务的历史QoS数据，预测备选服务是否能满足指定的QoS需求，输出为满足QoS需求的最优服务。

在文献[[1]](#endnote-1)中提出了一种基于贝叶斯网络的Web服务QoS动态评估方法，通过为候选成员服务建立基于贝叶斯网络的服务质量评估模型，选择能够满足指定多种QoS需求的最为合适的服务参与协作；并且基于监控信息引入模糊规则推理服务在实际执行中提供的服务能力，支持评估模型的动态更新，提高了评估结果的准确性。该方法不仅提出了QoS的评估方法，而且引入了推理机制，可以动态的更新评估模型，对实行分级策略的web服务可以进行评估。

1.2.2 服务替换技术

在服务替换方面，针对在BPEL（Business Process Execution Language，业务过程执行语言）中使用WSDL描述的服务进行替换已经出现很多方法。一般过程如下：对备选服务集进行语义标注，使其操作方法，输入和输出在语法及语义上一致；根据需求进行替换。如在文献[[2]](#endnote-2)中提出的使用AOP（Aspect Oriented Programming,面向切面编程）技术在流程执行过程中动态替换服务。其基本过程是这样的：使用SAWSDL（Semantic Annotations for WSDL and XML Schema）声明备选服务集，并对备选服务的进行语义标注，对所有服务的操作名称，输入和输出参数进行映射，使他们在语法上一致；在流程执行过程中，使用AOP技术进行替换。该研究使用SAWSDL使备选服务集的接口一致，所以在替换时不需要考虑服务之间的差异性。这种方法的缺点是：当有新服务加入的时候，需要处理所有备选服务的映射关系，因而不能动态的扩展备选服务集。

在文献[[3]](#endnote-3)中提出在BPEL中使用RESTful web服务，并且把组合的流程包装成RESTful web服务进行发布。通过扩展BPEL标准语言，增加BPEL标签<put>、<get>、<delete>和<post>，该研究将面向资源的RESTful web服务引入到BPEL中。流程实例直接对RESTful web服务进行操作，而不是使用BPEL中的伙伴关系链接（Partner Link）的方式。这样的一种处理，虽然使得在BPEL中使用RESTful web服务成为可能，但是将BPEL中角色关系丢失了。基于对这个问题的改进，在文献[[4]](#endnote-4)中进行了更进一步的扩展，通过增加<Role>标签，使RESTful web服务与流程及流程中其他服务之间的关系更加明确。但是这种扩展并没有处理异构服务之间的差异性，基于这种扩展不能很好的进行服务替换。

同时，国外对于使用RESTful web服务替换SOAP服务也有一些研究，在文献[[5]](#endnote-5)中提出使用RESTful web服务替换基于SOAP的web服务。该研究通过SAWSDL和MicroWSMO[[6]](#endnote-6)处理异构服务的差异，使其语法和语义一致，但是该研究的缺点是没有提出如何处理异构服务在流程执行过程的调用，因异构服务所基于的协议的不同，在流程执行过程是有差别的，因此不同协议的服务需要不同的适配器。

1.3 主要工作

本文的要就旨在提出一套方案解决服务组合中全局QoS约束下的异构服务选择与替换问题。从服务请求者的角度设计一个服务编排及运行的框架。基于此框架设计与运行的服务能在运行时按照用户的需求自动替换失效的服务，使流程能够按照用户的需求正常的执行下去。

首行（不含标题）正文段前0.7行

本文重点研究服务组合中的服务选择算法及原子服务的替换技术，其中包括：

1. 在用户给定全局的QoS约束下，通过服务选择算法选择出满足用户全局QoS约束的服务构成流程；
2. 在流程运行时，动态替换失效的服务，包括SOAP和REST服务。

1.4 论文组织结构

第一章 绪论 主要介绍本文的研究背景和服务替换的国内外研究现状。

第二章 相关技术的研究与分析 主要介绍web服务选择及替换的相关技术及算法。

第三章 基于skyline的全局QoS约束下的web服务选择 从流程的服务选择角度，在满足用户提出的全局QoS约束下，实例化用户构造的抽象的组合服务。对其中的算法进行研究。

第四章 异构服务替换技术的研究 主要对基于SOAP的服务和基于REST服务的替换技术的研究，以及流程语言BPEL在其中的应用的研究。

第五章 系统实现及实验 主要对第三章及第四章提出的研究结果进行应用，对实现的系统进行描述，以及对相关算法的实验验证，说明其有效性和精确性。

第六章 结论与展望 总结了论文的主要研究内容和结论，列举了待继续研究的问题，展望未来的研究工作。

**第2章 相关技术的研究与分析**

本章将对后面章节中所需要的各种理论进行介绍。首先，对Web服务及Web服务组合先关概念进行简要的概述，然后对Web服务替换及服务质量QoS进行阐述，同时，对服务选择算法中使用的skyline理论进行说明。本章的目的是为后面章节提供理论铺垫，是后面章节的问题阐述更为简明。

2.1 Web服务

2.1.1 Web服务简介

传统的分布式计算面临着两大难题：一是平台之间的互操作问题；二是客户端与服务端之间的紧耦合问题。尽管DCOM和CORBA等分布式计算技术能够解决分布式应用于互操作问题，但Web服务因其技术的简单性、兼容性及成本的低廉而越来越受到关注。Web服务做面向服务的体系架构（SOA）和面服务的计算（Service-Oriented Computing, SOC）的主要实现技术得到广泛的引用。随着Web服务标准的持续完善和支持Web服务的企业级软件平台的不断成熟，同时，云计算时代的到来，越来越多的企业将其业务功能和组件封转成标准的Web服务发布出去，从而实现快速便捷地寻求合作伙伴、挖掘潜在呵护和打到业务增值的目的。使用Web服务主要有以下益处：

互操作性：Web服务使用通用的数据描述语言来传输数据（如XML），因此，任何平台上引用程序与其他网络服务的应用程序都可以通信。

易用性：只要开发人员遵守Web服务标准，他们可以使用任意的语言、平台和架构来实现他们的引用程序。

无处不在的计算：因为Web服务是基于Internet的，因为在任何地方都可以访问和消费Web服务。一个额外的好处是，Web服务已经发展到遵守现有的基于互联网的安全性措施，如防火墙。

Web服务是基于XML等一系列标准上的Web应用程序，他将程序进一步封装，将其中业务功能暴露出来，使用XML数据来调用或消费该功能。基于XML的Web服务的主要标准和技术如下：

WSDL：Web Services Description Language(WSDL)[[7]](#endnote-7) 是为描述Web服务发布的XML格式。WSDL描述Web服务的公共接口。这是一个基于XML的关于如何与Web服务通讯和使用的服务描述；也就是描述与目录中列出的Web服务进行交互时需要绑定的协议和信息格式。通常采用抽象语言描述该服务支持的操作和信息，使用的时候再将实际的网络协议和信息格式绑定给该服务。

SOAP：Simple Object Access Protocol(SOAP)[[8]](#endnote-8)是交换数据的一种协议规范，使用在计算机网络Web服务（web service）中，交换带结构信息。SOAP为了简化网页服务器（Web Server）从XML数据库中提取数据时，节省去格式化页面时间，以及不同应用程序之间按照HTTP通信协议，遵从XML格式执行资料互换，使其抽象于语言实现、平台和硬件。

j0205582UDDI：Universal Description, Discovery, and Integration(UDDI)[[9]](#endnote-9) 它是一个基于XML的跨平台的描述规范，可以使世界范围内的企业在互联网上发布自己所提供的服务。

图2-1解释了Web服务的如何运作。其中有三个参与者。

服务提供者（Web服务）将其提供的Web服务的描述信息（WSDL文件）注册到目录服务器上。

服务消费者（如，软件代理，程序或人）通过使用UDDI协议的目录服务器找到满足其需求的服务提供者。然后获取描述Web服务详细信息的WSDL。

通过WSDL文件内描述的API和数据结构，服务消费者通过标准的消息协议SOAP向服务提供者发送请求，服务提供者处理请求返回SOAP消息。

随着Web2.0的出现，Web服务也出现了新的类型。RESTful web服务，它是遵守REST[[10]](#endnote-10)架构原则来实现的web服务。REST(Representational State Transfer)是一种轻量级的Web service架构风格，其实现和操作比SOAP和XML-RPC更为简洁，可以完全通过HTTP协议实现，还可以利用缓存Cache来提高响应速度，性能、效率和易用性都由于SOAP协议。REST架构是针对Web应用而设计的，其目的是为了降低开发的复杂性，提高系统的可伸缩性。REST提出了如下的设计准则：

* 资源是由URI来指定；
* 对资源的操作包括获取、修改、创建和删除资源，这些操作正好对应HTTP协议提供的GET、POST、PUT和DELETE方法；
* 通过操作资源的表现形式来操作资源；
* 资源的表现形式则是XML或者HTML，取决于读者是机器还是人，是消费web服务的客户软件还是web浏览器。当然也可以是任何其他的格式。

Restful Web服务从以下几个方面定义一个Web服务：

* URI，比如：http://example.com/resources/。
* Web服务接受与返回的互联网媒体类型，比如：JSON，XML ，YAML 等。
* Web服务在该资源上所支持的一系列请求方法（比如：POST，GET，PUT或DELETE）。

2.1.2 Web服务描述语言

WSDL 1.x

对于基于SOAP的Web服务，WSDL占有绝对优势的地位，几乎所有的SOAP服务都用WSDL作为描述语言。WSDL是一个用来描述Web服务和说明如何与Web服务通信的XML语言。他为用户提供详细的接口说明书。一个WSDL文档包含了服务的四个重要信息。即接口信息、数据信息、绑定信息和地址信息。

WSDL文件包含以下元素：

Type：XML定义的数据类型，一般的，这里使用XML Schema来定义数据类型；

Message：定义一个操作的输入或者输出；

Part：消息参数；

Operation：服务支持的操作的抽象描述；

Port Type：定义一个Web服务支持的所有操作（包括输出，输出和错误）；

Service：相关端点的集合，包括其关联的借口、接口和消息等。

WSDL支持4中消息交换方式：

单向（One-way）：服务端接受消息；

请求响应（Request-response）：服务端点接收请求消息，然后发送响应消息；

要求应答（Solicit-response）：服务访问端发送要求消息，然后接收应答消息；

通知（Notification）：服务访问端点发送通知消息。

WSDL 1.1致力于创建低耦合性，并最大限度地重用每一个可能的片段。例如，可以定义一个消息，既用作输入或输出，或者被多个其他端口使用。然而，这样的结构导致难以阅读，所以在WSDL 2.0对这一点进行了改进，将<message>标签删掉了，而是用定义元素的方法代替。

WSDL 2.0

WSDL 2.0也是一种用作描述Web服务的接口说明的文档。它不仅可以描述基于SOAP的Web服务，它也可以描述其他类型的Web服务。WSDL 2.0增强了对RESTful服务的支持。WSDL 2.0改进了HTTP绑定，每一个操作可以独立的绑定一个HTTP操作原语（如，GET, PUT或者POST），而在WSDL 1.1规范中在同一个WSDL中只能有一种HTTP操作。

WADL

WADL(Web Application Description Language)用来描述基于HTTP的应用，如那些遵循REST架构风格开发的Web应用。WADL是一种以资源为中心的描述语言，一个WADL文档通常由一组资源描述组成。而WSDL则是以接口为中心。

2.2 Web服务组合及替换

Web服务组合源于软件重用，他的基本思想是使用系统中已有的Web服务，通过一定结构的组合，创造出新的或质量更高的服务用以满足用户不断增长的需求。Web服务组合具有这样一些特点：多层次性和可扩展性；动态性与自适应行；业务流程的自动化；较高的软件生产率。

Web服务组合旨在实现不同功能Web服务的有机合成以形成功能更为强大的服务。目前，工业界和学术界分别从不同角度对Web服务组合进行了大量研究，并提出了多种Web服务组合的方法。总的说来，工业界方法侧重提出组合描述语言、开发相关编辑工具和执行引擎；而学术界则侧重于语义、智能规划等方面的研究。当前，有几种常见Web服务组合方法[[11]](#endnote-11)，基于BPEL4WS的Web服务组合方法，基于语义的Web服务组合方法，基于组件的Web服务组合方法，基于Petri网的Web服务组合方，基于进程代数的Web服务组合方法。本文中将主要使用基于BPEL的Web服务组合方法，简单介绍一下基于BPEL4WS的组合方法。

BPEL4WS（Business Process Execution Language for Web Services），是由IBM、微软和BEA公司提出的一个Web服务组合描述语言，他是一种基于XML的，用来描写业务过程的语言，被描写的业务过程的每一个单一步骤则由Web服务来实现。它可以将Web服务调用、数据操作、故障抛出或终止一个流程等不同活动连接起来，从而创造更复杂的流程。BPEL4WS流程是一个流程图，用来表达特定业务的处理逻辑和算法，流程每一步叫做一个活动。这些或可以嵌套在结构化活动中，而结构化活动顶一个了活动的运行方式和控制流程。BPEL4WS主要是利用了WSDL是的服务的动态绑定成为可能，但它并不支持应用运行时的流程模型的调整。

2.2.1 BPEL语言

BPEL语言是一种流程建模语言，它依赖以与一系列的XML的开放技术标准，例如WSDL，XML Schema和XPath等。下图给出了BPEL与它涉及的标准的关系。

XPath是用来定位XML文档中的节点和节点集。XML Schema是一个遵循XML标准的用来定义其他基于XML标准的标准。类似于想SOAP、WSDL和BPEL都是用XML Schema来定义的。SOAP是Web服务传递的消息的封装标准。WSDL是对BPEL影响比较大的一个标准，WSDL语言为BPEL流程与其他参与者之间的交互接口提供了描述的标准。BPEL流程最终也将做一个Web服务提供出来，并且使用WSDL来描述。BPEL是一个无关平台的标准，流程发的发布、查询和调用都是通过Web服务的执行平台来实现。BPEL本身关注的是流程的逻辑、事件处理、异常处理等问题。

2.2.3 BPEL流程结构

要组成一个BPEL流程，需包含以下这些基本的元素：伙伴链接（Partner link）、变量（variable）、活动（activity）。伙伴是指与业务流程进行交互的Web服务，而Web服务于BPEL流程之间的交互关系就被定义为伙伴链接。变量用来保存流程状态。活动是BPEL流程的基本的单位，用来定义一个流程的基本行为。除了这些以外，还有补偿处理程序（compensation handler）、异常处理程序（fault handler）和事件处理程序（event handler）等。流程对于撤销某一事务行为并将其对应的数据恢复到执行前的状态的这个操作的定义是在补偿处理程序结构中定义的。异常处理程序的作用是捕获并且处理业务流程执行过程中出现的异常，而事件处理程序的作用则是用来并行处理流程执行过程中产生的之间。

伙伴链接（partner link）

BPEL流程主要是用来描述企业之间或者不同部门之间的业务交互的，每次业务流程之间的交互都是通过Web服务接口来实现的。业务流程与这些Web服务是对等关系的，在服务层次上他们也是相互依赖的。这就是说一个Web服务可以调用BPEL流程中定义的服务，同时这个服务也很有可能是BPEL中顶一个的某一服务的提供者。伙伴链接就是用来在BPEL中形容这种关系的。如图2-2所示，一个伙伴链接是一个伙伴链接类型的实例。伙伴链接类型使用定义服务的角色来描述他们之间的关系，使用portType来描述交互式产生的消息的接收。图2-3给出了BPEL和Web服务之间的关系的示例。

变量（variable）

变量是BPEL用来保存业务逻辑等发送给伙伴链接所产生的中间数据信息。在BPEL中的变量可以声明为WSDL消息类型，也可以是XML Schema类型或者XML Schema元素。声明在全局的流程作用域中的变量叫做全局变量，而声明在<scope>中的变量则被称为局部变量。变量只有在声明它的作用域范围内或其嵌套域中才是可见的。

相关集（correlation set）

相关集是将会话和消息关联在一起的一组属性的集合。BPEL公国属性来指定消息的上下文。同时把它作为区别其他的唯一的标识。在WSDL中，属性通过不同的属性别名来映射到消息的不同的作用域中去，以实现不同消息的关联。在一个业务流程的声明周期内，它将会涉及到许多的不同的回话，涉及的Web服务也有很多个，这就必须保证特定的消息和回话能匹配到正确的流程实例中去，而相关集就是BPEL提供用来实现这一功能的。

活动（activity）

活动是组成BPEL流程结构的基本元素。BPEL中一个语句或者一个步骤的执行就是一个活动。活动又分为基本活动和结构化活动两类。基本活动执行某种功能，他是具体行为的执行者，是不可分的。而结构化活动描述的是对流程逻辑的控制，它规定了基本活动的执行顺序。像流程中的数据流、异常处理及时间处理等都是在结构化活动中得到说明的。

基本活动有以下几种：

receive/reply，提供一个同步或异步调用一个Web服务的接口。

invoke，调用活动，调用一个Web服务。

assign，赋值活动，用于更新变量。

throw，抛出活动，用来抛出异常。

wait，等待活动允许业务流程等待某一给定时段或者直到某一时刻为止。

terminate，终止活动用于立即终止某一运行中的业务流程实例。

Empty，空活动是不定义任何操作的活动。

结构化活动：

Sequence，顺序活动，用来指定一个顺序执行的活动集合。

While，循环活动，指定一个循环执行的活动集。

forEach，多分枝处理结构是用来指定从一组活动中选择一个特别活动。

Flow，并发活动，表示活动之间的执行是并发的。

Pick，时间选择活动会等待一组相互排斥事件中的一个事件的发生，然后执行与发生事件相关联的活动。

作用域（scope）

Scope活动可以将流程定义的部分逻辑细分到使用这个活动的范围中。在scope中可以顶一个不同的异常处理程序。作用域为嵌套在其中的活动提供上下文，并且也是定义异常处理程序和补偿处理程序的地方。因此，可以将作用域看做是一个可补偿的、可恢复的工作单元的封装。每个活动的执行上下文由scope来提供。Scope可以提供异常处理程序、事件处理程序、补偿处理程序、数据变量和相关集。Compensate活动命名了执行补偿所在的作用域，即将要执行的补偿处理程序的作用域。为了让补偿处理程序通过名称来调用，所有直接嵌套在作用域和活动的名称必须是唯一的。Compensation handler活动只能用于业务流程中的异常处理程序或补偿处理程序中，这个异常处理程序或补偿处理程序包括了补偿将执行的作用域的上一级作用域的。

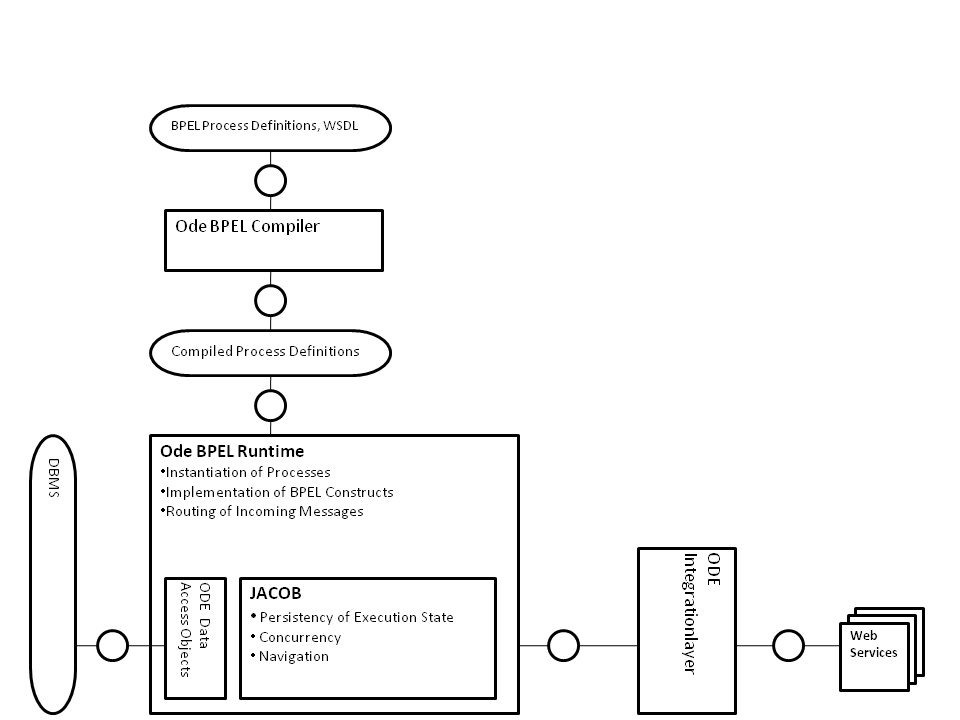
2.2.3 BPEL流程引擎

BPEL仅是一个定义流程的标准，它的具体实现还要依赖于流程引擎的具体实现。当前，比较有影响力的开源BPEL引擎有，ActiveBPEL，JbossBPEL以及apache ODE。ActiveBPEL出现的较早而且对BPEL标准的支持和实现都较好，因此一直以来都占据主要地位。但是，它的开源社区近几年都不太活跃，版本更新也比较慢。而Apache ODE出现后，软件本身的优势以及它的开源社区的活跃，并且由于是托管于Apache基金会，因此，近几年逐渐占有一席之地。下面介绍一下Apache ODE。

Apache ODE的开发目标是建立一个可靠的、轻量级的、可嵌入的能管理长时间运行流程的组件，该流程由BPEL流程描述语言定义。当前ODE版本支持BPEL2.0标准。Apache ODE系统架构的关键模块包括

ODE BPEL编译器：编译器服务将BPEL源文件（BPEL流程描述文件、WSDL文件以及Schemas）转化成一个编译好的、适合于流程引擎执行的流程表示形式。

ODE BPEL运行时环境：提供对于执行编译好的BPEL流程所需的功能。运行时模块通过提供对于各种不同BPEL结构的实现，来执行流程运行的工作。

ODE Integeration（ILs）：ODE BPEL运行时通过集成层与外界Web服务进行交互。目前ODE提供对于Axis2和JBI的服务集成层。

2.3 Web服务质量

从计算机系统诞生开始，人们就孜孜不倦的致力于提高系统的服务性能和服务质量。通常意义上的QoS指带宽、延迟、抖动、丢失率等，主要是网络流量参数的客观值。IETF在[[12]](#endnote-12)中将QoS描述为：“QoS是网络在传输数据流时需要满足的一系列服务（指数据包经过若干网络节点所接受的传输服务）要求。”

服务质量QoS是网络的一种安全机制，是用来解决网络延迟和阻塞等问题的一种技术。QoS支持管理员有效地使用现有资源，并在无需扩充网络使它更快或者超量提供资源的前提下，保证关键的应用程序接受的服务是搞质量的。配置QoS意味着，网络管理员可以更好地控制网络，减少成本，或提高顾客的满意度。QoS的提供增加了网络的性能可预知性，网络资源可以有效地分配，提高网络的吞吐量，减少网络丢弃数据包尤其是较重要的数据包的概率。

QoS的提出成功解决了计算机网络中的相关问题，得到了很好的应用。而在SOA领域，人们也越来越关注服务的QoS问题。

能够提供满足用户需求的高质量服务是至关重要的。但是由于业务应用本身的动态性和复杂性，网络负载的不可预测性与动态性等因素，是服务质量成为服务计算领域中重要的研究问题。如果没有服务质量管理，企业应用的性能可能会降低，甚至导致功能性错误或者效益损失[[13]](#endnote-13)。

为了提供有质量保证的Web服务，目前，研究者们在服务发现、服务选择、服务组合中都引入了QoS来进行考虑。QoS可以作为SOA领域衡量Web服务的一项重要指标。QoS可以体现用户对Web服务的满意程度，通过对Web服务的各个QoS属性的度量，可以反映出Web服务的质量；第二，QoS可以反映用户对服务质量的个性化需求，只需将用户的个性化需求映射为对某个或某些质量因素的特定需求；第三，利用QoS进行Web服务选择、组合在现实中是可行的。

2.4 Skyline服务

**第3章 Qos驱动的异构Web服务替换**

在RESTful服务越来越流行的今天，越来越多的服务提供商提供了RESTful的服务，或者使用RESTful服务来代替原来的SOAP服务。如Google，它使用RESTful的搜索API取代了之前工作良好的基于SOAP的搜索API。尽管如此，SOAP仍然有相当的基础，它的标准以及基于它产生的标准保证它在短时间内不会全部被RESTful服务替代[[14]](#endnote-14)。在Web 2.0中的服务组合被叫做Mashup[[15]](#endnote-15)[[16]](#endnote-16)[[17]](#endnote-17)，它将网络上其他的服务或者信息组合到一起呈现给用户。这种简单的组合是原有的服务产生出新的价值，普通用户通过组合几个服务创造出一个新的应用成为了可能。但是Mashup却没能在企业级应用中迅速普及，主要原因是他没有一个标准的描述语言。在SOA中，mashup应用可以被看做是服务组合，而BPEL是描述Web服务组合的一个已经被认可的标准。但是，当前BPEL标准不能集成RESTful服务，BPEL基于WSDL的，它通过WSDL来操作Web服务。本章将讨论，在BPEL流程运行时使用REST服务或者SOAP服务来替换BPEL中的服务。

3.1 REST服务和SOAP服务

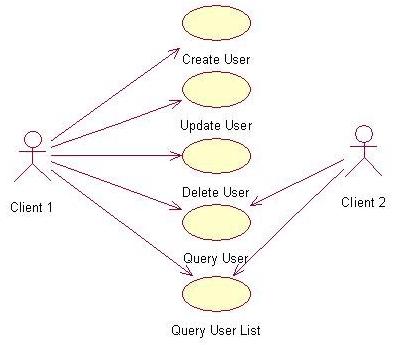
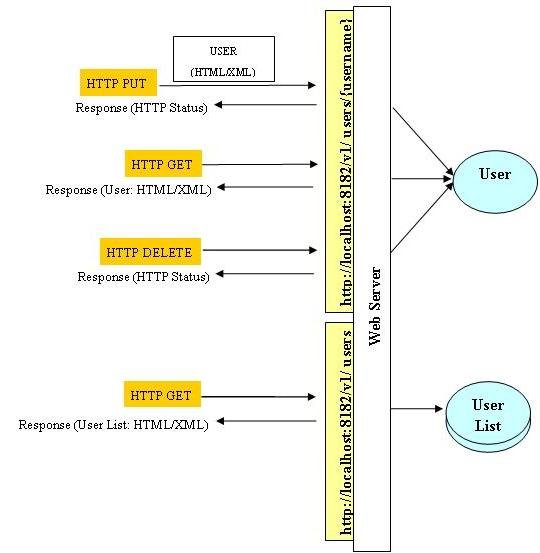
为了解释两种服务的架构的区别，我们这里使用一个在线用户管理的场景对比两种类型的服务。如图3.1所示在这场景中，客户端（Client1）可以添加，删除，更新和查询用户信息。

图3.1 用户管理用例图

3.1.1 REST服务的架构

图3.2 RESTful服务的架构

使用REST风格服务实现的架构图如图3.2所示。每一个资源（Resource）都有一个唯一URI地址，客户端可以通过这个URI访问到这个资源，每一个资源有4种操作GET, POST, PUT和DELETE，这个4个操作都是http协议中定义的基本操作。通过这四种方法，RESTful服务实现了类似数据库的CRUD（Create, Read, Update, Delete）操作：

GET: 类似于Read，获取URI所指向的资源的信息；

POST: 类似于Create，向URI提交数据创建新的子资源；

PUT: 类似于Update，提交信息更新URI所指向的资源的信息；

DELETE: 删除URI所指向的资源。

如图3.2，URI=http://localhost:8080/users对应于用户列表这个资源，它支持两个操作GET, POST.

GET /users 对这个URI使用GET操作，会返回用户列表；

POST /users 使用POST提交数据，会根据提供的信息在列表中创建新的用户。

用户列表并不支持DELETE和PUT方法，如果把用户列表删除那么所有的用户信息都将丢失，同时用户列表也不能被整体更新，只能更新用户列表内某个用户的信息。

如图3.2， URI=http://localhost:8080/users/123对应于Id=123的用户资源。它支持3个操作GET, PUT和DELETE

GET /users/123 获取id=123的用户信息；

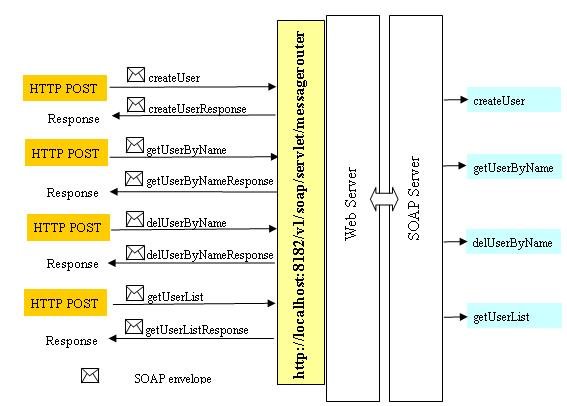
PUT /users/123 更新id=123的用户信息；

DELETE /users/123 删除id=123的用户；

用户资源不支持POST操作，这是因为创建用户的操作是定义在users资源上的，跟id=123的用户或者其他某个特定的用户是没有关系的。

REST服务无论在服务定位、数据还是接口依赖方面，耦合度都是很低的14，每一个操作对应的URI总是固定的，如前文中id=123的用户的URI总是固定的/users/123。

3.1.2 SOAP服务的架构

图3.3 SOAP服务的架构

如图3.3所示，与 REST 架构相比，SOAP 架构图明显不同的是：所有的 SOAP 消息发送都使用 HTTP POST 方法，并且所有 SOAP 消息的 URI 都是一样的，这是基于 SOAP 的 Web 服务的基本实践特征。

<?xml version=**"1.0"** encoding=**"UTF-8"** standalone=**"no"**?>

<soap:Envelope xmlns:soap=**"http://schemas.xmlsoap.org/soap/envelope/"**>

<soap:Body>

<p:getUserList xmlns:p=**"http://www.exmaple.com"**/>

</soap:Body>

</soap:Envelope>

图3.4 getUserList SOAP消息

如图3.4所示，这是获取用户列表的SOAP消息，这个消息会被发送到这个SOAP服务的服务地址http://localhost:8080/soap/messagerouter上，根据这个消息的<Body>标签中的<getUserList>标签，该消息会被Web service服务器路由到getUserList的处理器上。RESTful Web 服务使用标准的 HTTP 方法 (GET/PUT/POST/DELETE) 来抽象所有 Web 系统的服务能力，而不同的是，SOAP 应用都通过定义自己个性化的接口方法来抽象 Web 服务。例如本例中的 getUserList方法。

3.2 QoS驱动的异构Web服务替换的研究

为了讨论了异构Web服务替换的技术，这里先说明一下本文的预设的应用场景，终端用户通过使用BPEL组合服务，并且指定全局QoS服务质量约束整个流程的执行质量，在BPEL流程运行过程中，系统动态的将不满足QoS约束的Web服务替换成满足约束的服务，直到执行结束。替换中使用备选Web服务包括SOAP服务和REST服务。

在BPEL流程中对SOAP服务进行替换，已有很多研究，如文献提出了QoS驱动的服务适配方法，通过使用SAWSDL（Semantic Annotations for WSDL）[[18]](#endnote-18)标记web服务的WSDL文件，在BPEL流程执行时动态替换服务。而在BPEL中替换或者使用REST却是相当困难，这是因为BPEL最初制定的时候是以WSDL为基础，而使用WSDL很难描述REST服务。使得直接在BPEL中使用RESTful服务比较困难。文献3中给出的解决方案是对BPEL进行扩展，为其添加<get>, <post>, <put>和<delete>标签，直接给BPEL操作REST服务的能力，这样在BPEL语言中有两种不同的activity用来操作Web服务，<invoke>活动可以操作基于WSDL描述的服务，即SOAP服务；<get>等活动用来操作REST服务。但是这样的扩展却不利于替换操作，两种服务需要使用五种活动进行交互，使得替换的复杂性极大的提升。

在BPEL中进行异构服务替换需要解决一下几个问题：

Web服务操作的映射：如何映射不同的Web服务的操作，对于SOAP是自定义的操作，如上一节中提到的getUserList；而对于REST服务是（URI, HTTP Method）二元组；

请求参数的映射：对SOAP服务是封装在SOAP信封的中数据，而REST服务一般使用XML, JSON或文本直接发送；

响应参数的映射：SOAP服务把数据都封装在SOAP信封中返回给客户端，客户端需要解开封装才能知道结果，但是REST服务在返回数据的同时，通常使用HTTP Status Code[[19]](#endnote-19)指示这个次服务请求的结果，如201代表创建成功，401代表认证失败等。

本文对于在BPEL中替换异构Web服务提出这样的方法，将两种服务都转化成一种作为中介的服务，通过这种中介服务联系起两种服务。本文称该中介服务为Mock服务。

3.2.1 Mock服务

Mock是作为软件测试中的概念被定义的，在《The Art of Unit Testing》[[20]](#endnote-20)一书中对mock的定义是：Mock对象是一个假的对象，这个对象通过验证一个对象上的交互操作发生与否来帮助判定一个测试用例是否通过。使用Mock对象，测试人员可以避开不必要的复杂程序逻辑而判断程序的行为正常与否。

在本文中，BPEL的中需要被替换的服务全部用Mock服务代替，在BPEL流程执行前，系统会根据用户指定的QoS约束选择出真实的Web服务跟Mock服务绑定，同时BPEL过程中，如果跟Mock服务绑定的Web服务达不到声明的QoS服务质量，则系统会重新选择满足QoS约束的Web服务跟Mock服务绑定。在BPEL与Web服务交互时，Mock服务会接受BPEL传来的请求信息，系统通过预先定义的规则将请求消息转化成与Mock服务绑定的真实Web服务能接受的请求信息，整个过程如图3.5所示。

这里并没有使用抽象服务而是使用了Mock对象这个软件测试中的概念，是因为Mock对象在不需要时可以随时去掉用真实的Web服务代替，在本文中，Mock服务可以是一个抽象的Web服务，也可以是一个具体的真实的Web服务。当BPEL中的Mock服务是具体的真实的Web服务时，该BPEL流程可以在任何支持BPEL标准的流程引擎中正常运行。同时，无论该BPEL流程的Mock服务为抽象或是具体的Web服务，在本文所提出的框架下，它都支持异构Web服务的替换。使用Mock服务保证了BPEL语言的结构不被破坏，它可以运行在任何标准的BPEL引擎中。

画图，从上到下BPEL -> mock->真实web服务

**第4章 基于skyline的全局QoS约束下的web服务选择**

面向服务的计算能够无缝地把现存的各种商业应用服务组合起来，形成新的增值服务满足用户的需求。然而随着web2.0的出现，越来越多的web服务以及开放API出现，满足相同功能需求的服务越来越多，在选择服务时不仅要考虑的满足功能需求，还需要考虑服务质量的需求。如何能从具有不同QoS属性的高度动态化的web服务中，以有效的方式在较短的时间内选择出满足用户对于全局QoS需求的服务，已经是一个web服务组合中一个比较重要的问题。

然而，在日益增多的Web服务组件中，具有相同功能和不同QoS的服务组件会不可避免地大量出现。这些服务组件会组合出成千上万的具有相同功能不同QoS的组合服务，使得基于QoS的服务选择问题成为NP难题。其研究难点在于解空间过于庞大，无法在多项式时间完成搜索[[21]](#endnote-21)，导致基于QoS的服务选择时间开销过大。

目前，在基于全局QoS约束的研究工作中，较常使用线性规方法。文献[[22]](#endnote-22)[[23]](#endnote-23)将基于全局QoS的服务选择问题映射为多维背包问题，并提出使用混合整数规划（Mixed Integer Programming, MIP）方法寻找最优组合。当服务数量较少时，线性规划方法是很有效的，但当服务数量增加时，其时间复杂性将呈指数增加[[24]](#endnote-24)，以至于无法在对时间花费敏感的组合服务中加以应用。

文献[[25]](#endnote-25)提出了一种基于Skyline的web服务选择，首先通过skyline查询区分出skyline服务，缩小搜索空间，之后再通过层次聚类构造QoS level Tree，之后使用MIP方法在对每一层的服务进行求解，找出最优。该方法通过skyline查询和层次聚类将备选服务的数量减少到很小的数量，之后使用MIP的效率会比较高。本文在其基础上进行改进，进一步提高算法的效率。

为了对全局约束下的服务选择进行介绍，本文先给出服务组合中的相关概念的形式化说明，假定S为一组抽象服务类的集合，每一个抽象服务类， 表示一组功能性属性相同，但是非功能性属性不同的服务的集合。上述服务类的信息均由服务注册中心管理（例如，区域医疗协同平台）。一个抽象的组合服务，它表示由一组抽象服务类组成的组合服务；一个具体的组合服务CS，可以被定义为是一个抽象的组合服务的实例化结果。通过为每一个抽象服务类绑定具体服务被实例化为具体的组合服务。

4.1 QoS效用函数

Web服务的非功能属性可以用来描述Web服务的质量，一般包括响应时间、可用性、价格以及吞吐量等，这些属性能够通过定量计算得到相应的QoS属性值。一个Web服务s的服务质量可以用向量来表示，其中表示Web服务s的第i个QoS属性。组合服务的QoS属性值可以通过组件服务的QoS属性值和相应的组合类型（顺序、概率、循环等）聚合获得，如组合服务CS的QoS属性向量，表示CS的第i个属性值的聚合函数。如响应时间的聚合函数，可用性的聚合函数，吞吐量的聚合函数。虽然，服务的组合类型不同（顺序、概率、循环等），但是均可通过文献[[26]](#endnote-26)中提到的方法将其转化为顺序类型。

在Web服务组合中，由于候选服务具有多QoS属性，每一个QoS属性值的单位和范围不尽相同，不利于从全局QoS最优的角度对候选服务的QoS属性值进行计算或评估。因此，QoS效用函数被提出来了，它用来将服务类中的每个候选服务的QoS属性向量映射到一个实数值，通过该值对每一个候选服务进行排序或分类，便于选择到满足QoS约束的服务组件。本文中的效用函数与文献[[27]](#endnote-27)中的方法相似（既简单加权方法），其构造方法是，将Web服务中的候选服务（或服务类）与其对应的最大值或最小值进行比较，从而将多个QoS属性值进行归化处理（范围在0~1），使其转化到一个综合衡量的实数值（独立于每个具体属性的单位或范围），其中，加权值（范围在0-1）代表用户的优先级或个人偏好。例如，在顺序聚合模型中，候选服务和组合服务CS的QoS效用函数U(s)和U(CS)可以通过下式计算得到：

且

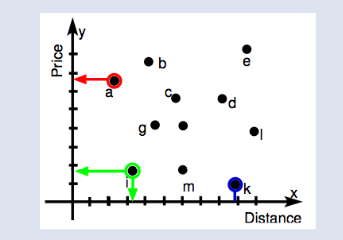
上式中：代表用户偏好，并且；表示服务类中所有候选服务中第k个属性值的最小值；表示所有服务类中第k个属性的最小值；类似地，和为最大值。

基于全局QoS约束的服务选择是一个最优化问题，该问题的重点是从所有可能的服务组合中选择一个QoS效用函数值最大且满足全局QoS约束的组合服务。例如，如果全局QoS约束，求得的组合服务是最优组合服务，则必须满足以下两个条件：

组合服务中所有服务类的QoS效用函数值U(CS)最大；

组合服务的QoS聚合值。

解决这类优化问题最直接的方法是穷举所有可能的组合，并选出最优的组合。例如，对于包含n个服务类的组合服务，每个服务类有l个备选服务，那么将有高达个候选服务被选择。使用穷举搜索将会导致时间复杂度过高，耗费昂贵的时间资源，对于实时性较强或者在Web服务运行时才进行服务选择的应用是不合适的。在本文的方法中，先使用skyline查询找出服务类中的skyline服务，以缩小搜索空间。

4.2 Skyline服务

Skyline查询通常也被称为Pareto最优查询，最早作为最大化向量问题研究[[28]](#endnote-28)[[29]](#endnote-29)[[30]](#endnote-30)[[31]](#endnote-31)，是典型的多目标优化难题，目前在数据库研究领域引起了广泛的关注。Skyline查询在许多现实应用如多标准决策，城市导航系统，数据挖掘与可视化，用户偏好查询，市场分析，环境监视等领域或应用中发挥着重要的作用。一个Web服务的QoS有很多属性，不失一般性，假设现在仅考虑Web服务的响应时间和价格，如图4-1所示，以x轴代表响应时间，y轴代表价格。若干Web服务的分布如图。从图中可以看出，并不是所有的Web服务都具有参考价值，Web服务a具有最小的响应时间，Web服务k具有最低的价格，而Web服务i相对于其他服务不是价格低就是响应时间短。在这个服务集合中，这个三个Web服务没有被其他Web服务所支配。

Web服务的QoS具有多维属性，可以使用Skyline查询来找出其中有参考价值的Web服务集合，再从其中选出合适Web服务。

定义1：支配关系，在一个服务类S中, 有两个服务 x, y ∈ S. x 支配 y, 记做 x ≺ y, 当且仅当x在QoS的所有维度都不比y服务的差，并且至少有一维要优于y。

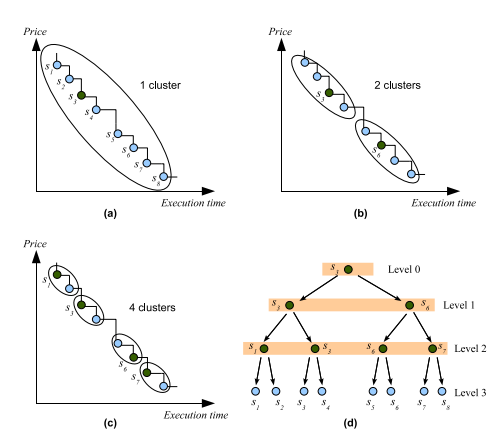
定义2：Skyline服务，在一个服务类中，所有没有被其他服务支配的服务都是Skyline服务。

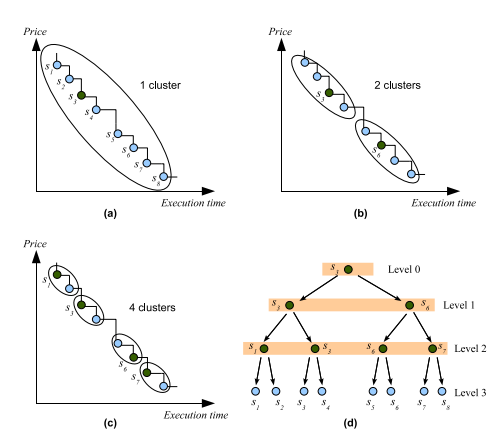
判断一个服务是不是skyline服务，需要把所有的服务进行两两比较。当前已经有很多比较成熟的算法[[32]](#endnote-32)来进行skyline查询。每一个服务类中skyline服务由服务的提供者来维护，在本研究中，skyline服务类表由一个云计算信息服务平台维护，在每次服务选择请求来之前，该平台已经维护好了一份每个服务类的skyline服务列表，第5章中会对该平台做详细介绍。

当一个服务类中的skyline筛选出来时，它们可以直接被插入到MIP中进行求解，来得出最优的服务组合。但是，当服务类的skyline服务数量很大的时候，直接进行MIP是相当低效的。因此，需要对skyline服务进行进一步的处理，使搜索空间进一步缩小。

3.3 基于skyline的服务选择算法

在文献18中，使用层次聚类处理每一个服务类的skyline服务。本研究中也使用同样的方法进行处理。首先，将每一个服务类的skyline服务聚类到k簇中，k=2,4,8,…,K,这里的K为skyline服务的数量。同时，从聚类后的各个簇中，选取一个QoS效能值最大的服务作为这个簇的代表，然后，将这些作为代表的服务构造成一个二叉树。树的叶子节点对应于skyline服务集合的每一个服务，根节点和中间的节点对应于聚类后产生的集合的代表服务。如，根节点对应于skyline服务集合中效能值最大的服务，根节点的后继节点对应于k=2聚类后产生的两个簇的代表服务，后面依次类推。每一个服务类都有一个相对应的二叉树。如图4-2所示，8个skyline服务分别被聚类成1,2,4,8个簇，在聚类成2个簇时，和分别是两个簇中QoS效能值最大的服务，所以它们被插入到代表服务二叉树的第二层。

当一个服务选择的请求到达时，整个算法的搜索从每一个的服务类的代表服务二叉树的根节点开始进行计算。如，在最开始搜索的时候，每一个的服务类的二叉树的第一层服务，即根节点的服务会被插入到MIP中进行最优化求解。如果找到解则算法结束。如果没有找到解，则根节点的下一层的服务会被插入到MIP中求解。整个过程将一直重复，直到找到解，或者到达最底层，即将所有的skyline服务都插入MIP中求解。如果在第一层就找到解，那么，整个组合服务的聚合QoS效能值将最大。越在低的层次找到解，则效能值越低。但是，如果每次都从第一层开始的话，是比较浪费时间的。使用MIP是比较耗时的，对一些本来不存在解的层次使用MIP将会浪费较多的时间，如果从第一层一直执行到最后一层的话，消耗的时间必定超过直接将skyline插入到MIP中进行求解，因为当算法执行最后一层时，搜索空间就是所有的skyline服务。如何能在求解前定位有解的层次，或者排除掉无解的层次，可以极大的提高算法的效率。本文使用虚拟最优服务来排除没有包含解的层次。

虚拟最优服务：对于一个服务类的代表服务二叉树的每一层，构造一个虚拟的服务，这个服务的QoS的每一维的属性是该层所有代表服务中各个维度最好的值。

如图4-3，在a中所有skyline被聚类成一个簇，是这个簇中QoS效能值最大的服务，同时也是代表服务二叉树的根节点，由于根节点只有一个服务，所以对应于根节点的虚拟最优服务既是根节点所代表的服务本身。在b中有2个簇，和分别是各自簇中效能值最好的服务，它们被插入到二叉树的第二层，这一层的虚拟最优服务是这个2个服务的QoS的各维度最好值构造出的服务，即b中的，而在c中则是。每一层的服务都可以用一个虚拟最优服务作为代表。为每一个服务类的每一层都构造一个虚拟最优服务，当服务组合的请求到达时，算法从第一层开始，首先计算组合服务中，由所有服务类第一层的虚拟最优服务组合而成的组合服务的QoS，如果这个组合服务的QoS满足全局QoS约束的要求，则该层可能有解，从这一层开始MIP运算。反之，如果该组合服务的QoS不满足全局QoS约束，则进入到下一层继续计算QoS聚合值，直到找到由虚拟最优服务组合而成的组合服务的QoS满足全局QoS约束，则从该层开始进行MIP运算求解最优解。

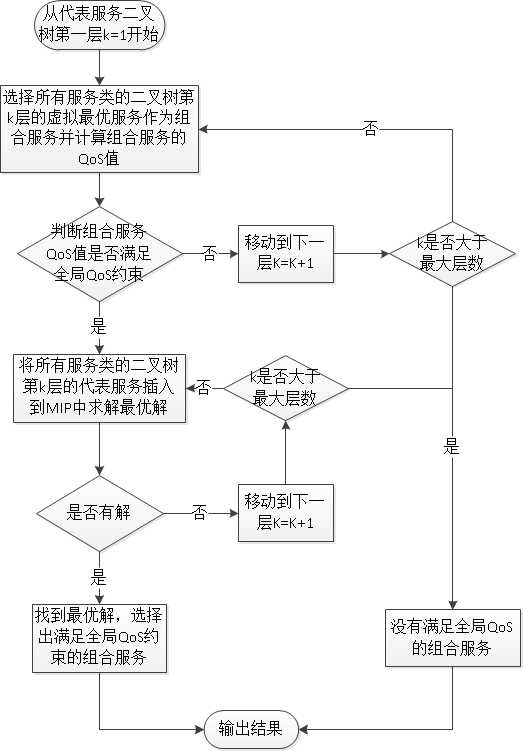


图4-4显示了本文所提出的服务选择方法的流程，首先，通过组合虚拟最优服务并计算组合服务的QoS，来定位有解的层次，然后，从有解的层次开始进行MIP运算求解最优解。MIP模型如下。

目标函数：

xij代表服务类Sj中的服务yij是否被选择，如果yij被选择，则xij = 1否则 xij = 0。

约束条件：

这个MIP模型求解组合服务中各个原子服务的效能值最大并且满足全局QoS约束的组合，由于MIP的输入是来自于服务类的代表服务二叉树的每一层，如果得出解的层次越高，则选择出的组合服务将拥有越高的效能值。这是因为在代表服务二叉树中，从顶层（根节点）到最底层，搜索范围在逐渐扩大，每往下一层，Web服务的数量会翻倍，而新加入的Web服务的QoS效能值会比上一层的Web服务的效能值低，如在代表服务二叉树中，根节点的所代表的服务是这个服务类的skyline服务集合中效能值最大的服务，如果以每一个服务类的代表服务二叉树的根节点组合服务，则这个组合服务的效能值将最大。通过使用虚拟最优服务排除掉没有解的层次，避免不必要的运算，可以极大的提高算法的效率。

3.4 实验与讨论

为了验证本章所提出的服务选择的方法，本文将在不同数量的备选Web服务的和不同的数量的服务类情况下对比以下方法：

ExactSkyline: 基于Skyline的标准的全局约束下求解最优解的方法。

SkylineRep: 文献18中所提出的方法。

SkylineRep-VOP: 本章所提出的服务选择方法。

备选服务的QoS使用QWS数据集，该数据集包含2500个真实世界的web服务的9个QoS属性。这些服务来源于网络，包括UDDI注册中心，服务搜索引擎和服务门户网站，数据集中的QoS数据是通过专业的基准测试工具测量得出的。关于该数据集的详细内容可以参考文献[[33]](#endnote-33)[[34]](#endnote-34)。所有的算法使用java实现，对于混合整数规划使用lpsolver5.5提供的java库来解决。所有实验使用一台intel双核2.4GHz处理器和4GB内存，windows7系统的机器完成。

本实验设计两种场景进行实验对比：

服务类的数量固定，服务类中的备选服务变化：服务类的数量固定为10个，每个备选服务类的中的Web服务数量从100增加到1000。

服务类的数量变化，服务类中的备选服务固定：服务类的数量从10增长到50个，每个服务类中的Web服务数量固定为100个。

上图4-5显示了当备选服务数量变化时，3个种方法的实验结果，从图中可以看出，ExactSkyline的曲线直线上升，而其他两种方法的曲线一直是平滑的。这是因为，ExactSkyline方法是直接把每个服务类的skyline服务集插入到MIP中进行求解的，随着备选服务的增加，skyline服务的数量也必定会增加，导致搜索空间增大，运算时间变长。而其他两种方法由于对skyline服务进行聚类，并且选出一些代表服务作为搜索输入，影响这两种方法的运算时间的因素是处理的服务类的代表服务二叉树的层次数，处理的层次数越多时间越长。从图上可以看出SkylineRep和SkylineRep-VOP的计算时间并没有随着服务数量的变化而出现大的变化。通过对skyline服务进行聚类后，选出代表性的Web服务并且构造代表服务二叉树，每一层的服务数量是固定的，因此，在服务类的数量以及用户提出的QoS约束不变的情况下，搜索空间并不会变大，计算时间不会有太大的变化。同时，从图中可以看出本文提出的改进方法SkylineRep-VOP在时间上有一定的优势，这是因为使用虚拟最优服务排除掉了一些无解的层次之后，直接从可能存在解的层次开始进行MIP，因此耗时比直接从第一层开始的SkylineRep要少。

上图4-6显示了服务类数量变化的时候，3个方法的运算时间，可以看出3条曲线都在上升，由于备选服务类的数量在逐渐增多，无论对于那种方法搜索空间都在逐渐增大。但是使用聚类的两种方法SkylineRep和SkylineRep-VOP仍然占有优势，同时使用本文提出的改进方法在时间上仍然占优。

同时，我们通过以下公式计算了解的最优度：

u代表Skyline-VOP方法计算出的最优效用值，uexact值代表使用纯粹的MIP方法计算出的最优效用值，结果表明SkylineRep-VOP的最优度大于95%。

3.5 结论

本文提出了一种基于Skyline的在全局QoS约束下的web服务选择方法。通过选出skyline服务并在其基础上进行层次聚类，从而缩小所有空间。同时，增加虚拟最优点用来定位有解的层次，进一步缩小搜索范围。通过将本文提出的方法同标准的MIP解法，以及文献18进行对比后，本文所提出的方法在性能上优于其他两种方法。

**第4章 结论与展望**

**第7章 结论与展望**

7.1 结论

本文的研究工作初步探讨了随机结构反应的概率密度演化问题，对于具有不同类型本构关系的随机结构反应分析问题提出了两种分析方法，初步建立了随机结构非线性反应的基本图景，给出了具有一定普遍意义的分析方法。

……

7.2 进一步工作的方向

本文的研究虽然取得了初步的成功，但依然任重道远，尚有许多有待进一步深入进行的研究工作，这里择其要者简要讨论如下：

……

**致谢**

标题：黑体，三号，加粗居中，单倍行距，段前24磅，段后18磅；正文部分仿宋，小四，行距16磅，段前段后0磅，首行缩进2个汉字符

逾尺的札记和研究纪录凝聚成这么薄薄的一本，高兴和欣慰之余，不禁感慨系之。记得鲁迅在一篇文章里写道：“人类的奋战前行的历史，正如煤的形成，当时用大量的木材，结果却只是一小块”。倘若这一小块有点意义的话，则是我读书生活的最好纪念，也令我对于即将迈入的新生活更加充满信心。

回想读书生活，已经整整二十个年头，到同济求学将近五年，攻读博士学位也已三年了。进入同济大学以来，深深醉心于一流学府的大家风范。名师巨擘，各具特点；中西融合，文质相顾。处如此佳境以陶铸自我，实乃人生幸事。

……

2010年5月

**参考文献**

标题：黑体，三号，加粗居中，单倍行距，段前24磅，段后18磅；内容宋体，五号，（英文用Times New Roman体，五号），行距16磅，段前段后0磅

[1] Elishakoff I., Ren Y. J. & Shinozuka M, Variational principles developed for and applied to analysis of stochastic beams. Journal of Engineering Mechanics, 1996,Vol.122 (6): 559～565

[2] 吕西林，金国芳，吴晓涵.钢筋混凝土结构非线性有限元理论与应用.上海：同济大学出版社，1997

[3] 陈建军，车建文，陈勇.具有频率和振型概率约束的工程结构动力优化设计. 计算力学学报，2001，Vol.18（1）:74-80

[4] 陈景润.组合数学. 郑州：河南科学技术出版社，1985

[5] 丁光莹.钢筋混凝土框架结构非线性反应分析的随机模拟分析:[博士学位论文] .上海：同济大学土木工程学院，2001

[6] 丁义明,方福康,范文涛.离散动力系统的密度演化方法. 见：许国志主编.系统科学与工程研究. 上海：上海科技教育出版社,2000:62-77

……

**附录A ×××××**

标题要求同各章标题，正文部分：宋体，小四，（英文用Times New Roman体，小四），两端对齐书写，段落首行左缩进2个汉字符。行距20磅（段落中有数学表达式时，可根据表达需要设置该段的行距），段前0磅，段后0磅。

……

**把restful服务转换成soap对应起来，uri/resource path/对应SOAP服务的endpoint, http\_method+服务名字对应SOAP服务的操作。**

**每一个服务（即一个操作）有一个列表，列出他相同功能的服务列表，参照adapt service（aop拦截替换服务的论文）里面的说明**

**Rest服务中的地址怎么处理？地址中参数baseuri/{id}/details**

**3.1 rest服务和SOAP服务的架构的不同导致的区别，**

**参考ibm文章**

**3.2 如何处理服务替换，如何标记替换的服务，同类服务，输入输出的转换。Rest服务的url地址的转换。**

**1. 以wadl来描述rest服务。Rest服务与soap服务的一一对应关系**

**2. 有中间结构来来处理两者的转换，如使用抽象服务或标准服务。**

**3.3 替换的时机，确定什么时候替换，替换谁，选谁来替换，使用英文文献**

**Adaptive qos-aware service process**

**个人简历、在读期间发表的学术论文与研究成果**

**个人简历：**

陈建兵，男，1975年8月生。

1997年7月毕业于东北大学 矿山建设（工业与民用建筑）专业 获学士学位。

1997年9月入同济大学读硕士研究生，1999年3月提前攻读博士学位。

**已发表论文：**

[1] 陈建兵，李杰.结构设备体系动力相互作用研究.地震工程与工程振动，2001，Vol.21 （3）:70-74

标题要求同各章标题，正文部分：宋体，五号（英文用Times New Roman体，五号），行距16磅，段前段后0磅，学术论文书写格式同参考文献,“个人简历”、“已发表论文”等字加粗

……

**待发表论文：**

[1] 陈建兵，李杰.随机结构分析中的均值参数反应与反应均值.力学季刊（已接收）

……

**研究报告：**

[1] 李杰,赵昕,陈隽,陈建兵,章萍.福建泉州层间隔震结构房屋振动台试验研究. 2001

……

[7] Pat. P. W. Chan and Michael R. Lyu. Dynamic Web Service Composition: A New Approach in Building Reliable Web Service. 22nd International Conference on Advanced Information Networking and Applications, 2008

Times New Roman体，四号，居中，单倍行距，段前0磅，段后0磅，选填Master、Doctor

1. 吴国全.基于模式的复合服务监管关键技术研究[D].合肥：中国科学技术大学，2009. [↑](#endnote-ref-1)
2. Vikas Agarwal, Pankaj Jalote. From Specification to Adaptation: An Integrated QoS-driven Approach for Dynamic Adaptation of Web Service Compositions. IEEE International Conference on Web Services, 2010. [↑](#endnote-ref-2)
3. Cesare Pautasso. RESTful Web service composition with BPEL for REST. Data & Knowledge Engineering, 68 (2009) 851–866. [↑](#endnote-ref-3)
4. Hao Yu, Cheng Zhu, Hongming Cai, Boyi Xu. Role-Centric RESTful Services Description And Composition For E-Business Applications. IEEE International Conference on e-Business Engineering, 2009. [↑](#endnote-ref-4)
5. Teodoro De Giorgio, Gianluca Ripa, and Maurilio Zuccal`. An Approach to Enable Replacement of SOAP Services and REST Services in Lightweight Processes. Current Trends in Web Engineering, 2010. [↑](#endnote-ref-5)
6. Kopecky, J., Gomadam, K., Vitvar, T.: hRESTS: an HTML Microformat for Describing RESTful Web Services. In: Web Intelligence. IEEE, Los Alamitos (2008). [↑](#endnote-ref-6)
7. http://www.w3.org/TR/wsdl [↑](#endnote-ref-7)
8. www.w3.org/TR/soap/ [↑](#endnote-ref-8)
9. http://uddi.xml.org/ [↑](#endnote-ref-9)
10. Representational State Transfer (REST), Roy Thomas Fielding, Author. 2000. Available

    at http://www.ics.uci.edu/~fielding/pubs/dissertation/rest\_arch\_style.htm [↑](#endnote-ref-10)
11. N. Milanovic, M. Malek. Current solutions for Web Services Composition[J]. IEEE Internet Computing, 2004:pp.-59 [↑](#endnote-ref-11)
12. E. Crawley, R. Nair, B. Rajagopalan. A framework for qos-based routing in the internet[S]. IETF RFC 2386, 1998. [↑](#endnote-ref-12)
13. 廖渊， 唐磊， 李明树. 一种基于QoS的服务构建组合方法[J]. 计算机学报. 2005, 84(4): 627-634. [↑](#endnote-ref-13)
14. 王建斌,胡小生,李康君,赵靓. REST风格和基于SOAP的Web Services的比较与结合[J]. 计算机应用于软件. 2010, 27(9):297-300 [↑](#endnote-ref-14)
15. Alessandro Bozzon, Marco Brambilla, Federico Michele Facca, A Conceptual Modeling Approach to Business Service Mashup Development, 2009 IEEE International Conference on Web Services, 2009, pp. 751-758. [↑](#endnote-ref-15)
16. Xuanzhe Liu, Yi Hui, Wei Sun, Haiqi Liang, Towards Service Composition Based on Mashup, 2007 IEEE Congress on Services. [↑](#endnote-ref-16)
17. Javier Soriano, David Lizcano, Juan J. Hierro, Marcos Reyes, Christoph Schroth, and et al, Enhancing User-Service Interaction through a Global User-Centric Approach to SOA, Fourth International Conference on Networking and Services, pp. 194-203. [↑](#endnote-ref-17)
18. World Wide Web Consortium (W3C). Semantic Annotations for WSDL and XML Schema, http://www.w3.org/TR/sawsdl/, 2007 [↑](#endnote-ref-18)
19. http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html [↑](#endnote-ref-19)
20. Osherove, Roy (2009). "Interaction testing with mock objects et seq". The art of unit testing. Manning. ISBN 978-1-933988-27-6. [↑](#endnote-ref-20)
21. Yu T, Zhang Y, Lin KJ. Efficient algorithms for Web services selection with end-to-end QoS constraints. ACM Trans. on the Web, 2007, 1(1):1-26. [↑](#endnote-ref-21)
22. Zeng LZ, Benatallah B, Ngu AHH, Dumas M, Kalagnanam J, Chang H. QoS-Aware middleware for Web services composition. IEEE Trans. on Software Engineering, 2004, 30(5):311-327. [↑](#endnote-ref-22)
23. Zeng LZ, Benatallah B, Dumas M, Kalagnanam J, Sheng QZ. Quality driven Web services composition. Proc. 12th Int"l Conf. World Wide Web (WWW), May 2003. [↑](#endnote-ref-23)
24. Alrifai M, Risse T. Combining global optimization with local selection for efficient QoS-aware service composition. Proc. of the 18th Int’l Conf. on World Wide Web. 2009. [↑](#endnote-ref-24)
25. Skyline文献 [↑](#endnote-ref-25)
26. J. Cardoso, A. P. Sheth, J. A. Miller, J. Arnold, and K. Kochut. Quality of service for workﬂows and web service processes. J. Web Sem., 1(3):281–308, 2004. [↑](#endnote-ref-26)
27. K. . P. Yoon and C.-L. Hwang. Multiple Attribute Decision Making: An Introduction (Quantitative Applications in the Social Sciences). Sage Publications, 1995. [↑](#endnote-ref-27)
28. Kung H T, Luccio F, Preparata F P. On finding the maxima of a set of vectors[J]. Journal of the ACM, 1975, 22(4): 469-476. [↑](#endnote-ref-28)
29. Bentley J L, Kung H T, Schkolnick M, et al. On the average number of maxima in a set of vectors and applications[J]. Journal of the ACM, 1978, 25(4): 536-543. [↑](#endnote-ref-29)
30. Bentley J L, Clarkson K L, Levine D B. Fast linear expected-time algorithms for computing maxima and convex hulls[C]. In: SODA, 1991. [↑](#endnote-ref-30)
31. Balke W T, Guntzer U. Multi-Objective query processing for database systems[C]. In: the international conference on Very Large Data Bases, 2004: 936-947. [↑](#endnote-ref-31)
32. D. Papadias, Y. Tao, G. Fu, and B. Seeger. Progressive skyline computation in database systems. ACM Trans. on Database Systems, 30(1):41–82, 2005. [↑](#endnote-ref-32)
33. Al-Masri, E., and Mahmoud, Q. H.. Discovering the best web service, (poster) 16th International Conference on World Wide Web (WWW), 2007, pp. 1257-1258. [↑](#endnote-ref-33)
34. Al-Masri, E., and Mahmoud, Q. H.. QoS-based Discovery and Ranking of Web Services, IEEE 16th International Conference on Computer Communications and Networks (ICCCN), 2007, pp. 529-534 [↑](#endnote-ref-34)