面向服务领域软件系统的模型驱动建模方法*)

蒋哲远　蒋建国

(合肥工业大学计算机与信息学院　合肥 230009)

摘　要　面向服务体系结构(SOA)的工程化和建模对现有的建模技术和方法提出了新的挑战。提出了一种基于Web服务的领域服务原型系统的快速模型驱动建模框架。从服务构件的概念和标准统一建模语言(UML)2.0的建模构造出发,给出了一个综合的服务软件建模过程。在此基础上,讨论了模型驱动的Web服务的特性描述,重点是介绍一种基于UML扩充机制的面向Web服务描述语言(WSDL)的建模技术。通过一个流通领域的面向服务企业资源计划(ERP)系统的实际建模,展示了所提方法是切实可行的。

关键词　面向服务体系结构,建模,模型驱动,UML profile,企业资源计划

Model-driven Modeling Methodology for Service-oriented Domian Software Systems

JIANG Zhe-yuan　JIANG Jian-guo

(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)

Abstract　Reuse engineering and modeling in Service Oriented Architecture (SOA) mount new challenges to existing modeling technologies and approaches. A rapid model-driven modeling framework for service-oriented domian software prototyping systems is presented. From the perspective of service component and standard Unified Modeling Language (UML) 2.0 construction, a comprehensive modeling process for service software is given. Furthermore, a model-driven Web Service Profile is discussed, which stresses on the Web Services Description Language (WSDL) modeling technologies by using extended UML. The experimental results of modeling a practical service-oriented Enterprise Resource Planning(ERP) system in the circulation industry indicate the proposed method is feasible.

Keywords　Service oriented architecture, Modeling, Model driven, UML profile, Enterprise resource planning

1　引言

研究基于Web服务(Web services)的领域服务软件体系结构的首要问题是如何表示体系结构,即如何对软件体系结构建模。根据建模的侧重点不同,若考虑采用ACME, Wright或Rapide等传统的形式化体系结构描述语言[1]进行某种扩展,可实现对Web服务体系结构的精确描述。但实践中,往往需要那些有很少或几乎没有形式化知识的领域专家参与并验证开发工作。考虑到Web服务继承了许多对象和构件的特性,但也使用了一些工作流和业务流程元素。因此,Web服务的软件体系结构建模应该包括这些不同的方面。使用统一建模语言(Unified Modeling Language, UML)作为Web服务体系结构建模的标准表示法是一个自然选择[2]。

UML是一种通用的可视化建模语言,它采用了Kruchten提出的"4+1"模型(逻辑视图、进程视图、开发视图、物理视图和场景视图)描述软件体系结构[3],可用于对软件密集型(software-intensive)系统的制品(artifact)进行描述、可视化处理、构造和建立软件系统的文档。UML通过给建模软件提供高级抽象和自动化方法,支持模型驱动开发方法(model-driven development)。UML原专注于面向对象系统的建模[4],但能被很容易地扩充去支持其它概念的建模,例如业务活动、用户接口流和数据模式等。另一方面,在逻辑层UML 1.x版本一直没有提供对构件和服务的表达机制,最新的2.0版本为处理应用逻辑构造块引入了新的概念和机制。系统化的建模和开发过程在现有文献中很难发现。

为此,本文将从服务构件的概念和标准UML 2.0建模构造出发,分析SOA的工程化和建模对现有的建模技术和方法所提出的挑战,提出一种综合的面向服务领域软件系统的模型驱动建模技术和方法,并应用于面向服务ERP系统的实际开发建模。

2　面向Web服务建模的需求

2.1　面向服务体系结构的建模需求分析

从本质上看,SOA并不是革命性的方法,它只是对原有技术的革新。SOA的基础是把现实世界中对终端用户的一切有意义的业务活动的功能性表示都看作服务的概念,并在软件方案中给予封装。SOA提供了松耦合服务,能被合成为支持业务目标的业务流。类似的创新过去也曾经提出,例如CORBA服务和Microsoft的DCOM。SOA的新颖性是不同用户间的互操作仅依赖一些广泛接受的协议标准,如XML, SOAP和UDDI等。最重要的是抽象层次被进一步提高了,以至SOA的主要构造块是现实世界业务活动,它被封装在为消费者提供商业价值的服务中。由于面向服务方案的复杂性和可利用技术平台的多样性,考虑使用对象管理组织(Object Management Group, OMG)提出的模型驱动体系结构(Model-Driven Architecture, MDA)设计基于Web服务的面向服务软件体系结构是一个自然的选择[5]。此外,使用一个与实现无关的建模构件和服务来代表商业和技术之间的一个抽象层。在规约层,可通过构件和服务来捕获业务目标、规则、概念和流程,进而映射到技术制品中,并凭借这种方式在业务和技术之间提供一个有效的双向可追踪能力。

SOA建模的逻辑起始点是基于构件、基于接口建模和标

*)教育部博士点基金项目(20060359004);合肥工业大学博士学位专项资助基金。**蒋哲远**　博士,副研究员,CCF高级会员,主要研究领域为面向服务的软件工程、软件体系结构;**蒋建国**　教授,博士生导师,主要研究领域为智能信息处理。

准的 UML。问题是扩展后的基于构件方法和 UML 能否完成上述需求，能否为 SOA 建模提供必要的概念和机制，例如构件化、低耦合、高内聚、基于接口设计、隐藏设计，以及对象和活动流。第一代基于构件的开发方法把构件看作实现二进制代码或源代码软件制品。这种看法是受 UML 1.3 的强烈影响，在该版本中构件被定义为实现图，并由显示物理实现方面的实现图来表达。这种从底向上的构件方法意味着实际中的面向对象分析和设计，并且在系统的生命周期的最后阶段才能打包高耦合的类/对象进入可部署的构件。这种思考方法必然会导致对象和 Web 服务之间的语义鸿沟，这是因为在这里的服务主要是业务驱动单元，而不是低级的编程构造。UML 的最新版本 UML 2.0[6] 在用构件代表设计级制品，以及更高级的控制和工作流概念和表示法等方面提出了进一步的改进。

一般而言，若使用标准 UML 为 SOA 建模至少还需要重视如下的一些考虑事项。首先，目前的 UML 方法还没有界定 SOA 的三个重要的元素：服务、流和实现服务的构件。这需要建模者明确地鉴别、制定和实现服务所需的技术和过程，它们的流程和组合，以及实现和确保所需服务质量的企业级构件。第二，需要进行范例的替换，以便更好地区分 SOA 的两个关键角色服务提供者和服务请求者之间的截然不同的需求。第三，在为一个企业或者某个业务构建应用程序时，应考虑将来有可能被提升到一个供应链中使用，并且向其合作伙伴公开，这些合作伙伴又可能组合、联合和封装该应用程序到一个新的应用程序中。这是服务生态系统或者服务价值网(value network)的概念。

与 UML 的演化相对应，高级 CBD(Component-based development)方法已经意识到在设计级制品中使用构件概念的好处。然而，在实体驱动方式中，构件的识别和规约主要是借助严格地匹配其基本业务实体来完成，例如客户、产品和订单。在这种方式中，构件被当作传统的业务对象而不是业务服务。SOA 要求的服务粒度等级应与业务构件工厂方法中的业务构件系统相符，或 Catalysis 方法中定义的大型系统构件，但这两个方法都没有强调这一点。在 CBD 方法中，一个构件的接口被定义为保护操作的前向条件和后向条件，以及操作基调(signature)等组成。该方法没有考虑扩充接口构造，在此基础上提供表达方式，例如协调参数、质量服务参数和配置参数来提供更为丰富的构件规约。

2.2 基于 Web 服务的企业服务建模原型系统

在分析了如何运用服务构件和标准的 UML 为 SOA 建模的需求后，针对 SOA 概念模型，采用 Web 服务技术框架，以流通领域的面向服务 ERP 集成[7] 为应用背景，实现了一个图 1 所示的基于 Web 服务的企业服务原型系统的体系结构。本文选择流通领域作为面向 Web 服务的领域软件体系结构的应用研究实例，一方面是考虑到流通企业天然地强调“服务”，信息系统特别适合使用 Web 服务技术；二是考虑到物流产业作为对全球经济体系产生革命性影响的新兴产业，将为国民经济在高起点提供基础动力。随着信息技术的进步，特别是电子商务技术日益成熟，现代物流的涵义和范围有了极大的拓展和延伸，已成为网络环境下制造企业发展的一个核心竞争力的重要体现。面向流通领域 ERP 系统正是充分体现现代流通企业的全球化和网络化特点，采用先进管理思想和当前计算机的最新应用发展水平，全面地集成了流通企业的所有资源，能够广泛适用于大型流通企业的批发、零售、分销、连锁、物流配送等多种应用场景和模式，从而可为流通企业提供全方位的经营决策、计划、控制与业绩评估的技术管理平台。

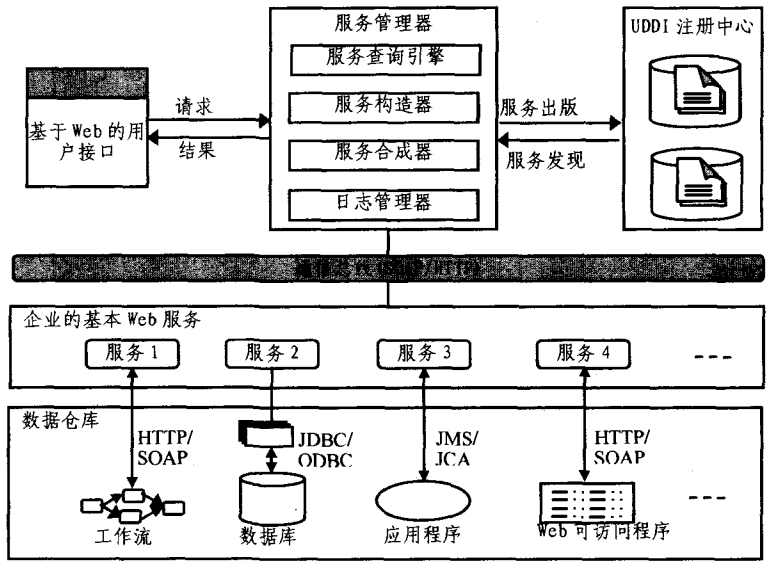

图 1 企业服务原型系统的体系结构

该系统主要包括通信总线、服务管理者以及构成 Web 服务体系结构的三类主体(角色)，即基于通用客户端的服务请求者、基于 UDDI 的服务注册中心和基于 Web 服务运行时环境的服务提供者。在服务管理器的管理和控制下，企业网络中的各种遗留系统(legacy systems)[9] 和服务都可以使用标准技术，例如 SOAP、HTTP、JDBC(Java Database Connectivity)、ODBC(Open Database Connectivity)、JMS(Java Messaging Service)和 JCA(Java Connector Architecture)，被打包作为基本的 Web 服务插入到通信总线中。通信总线主要使用异步的 SOAP 消息结构，可方便在不同平台上执行具有不同应用模型的程序之间的相互交互。这些模块已经使用 Java 和 IBM Web Services Toolkit (WSTK)实现。

面向服务的流通企业 ERP 的软件开发过程是首先完成了一个基于 Web 服务和软构件技术的通用开发平台 GERP-

ToolKit,可对领域的新服务构件和已有服务构件进行有效的查询、构造、组合、日志、仿真、分析和调式等管理,并在此基础上完成了整个 ERP 系统的开发[8]。

GERPToolKit 中的软构件遵循 COM、EJB 和 CCM(Corba Component Model)等构件标准,可分为以下两类。

(1)用户软构件:它又可分为通用处理软构件和专用处理软构件。其中用户引用前者时,通过给出适当的参数,系统将生成相应的功能构件。后者是用户自己使用的软构件,相当于固定功能构件。

(2)控制软构件:指标准的控制功能构件及界面构件等。

GERPToolKit 中的软构件目前由 26 个服务构件(service component)超类组成,其服务构件模型参见文献[10],存在形式包括原代码级和 EJB 控件两种形式,ERP 软件在集成时可根据需要灵活运用。GERPToolKit 的服务构件框图如图 2 所示。

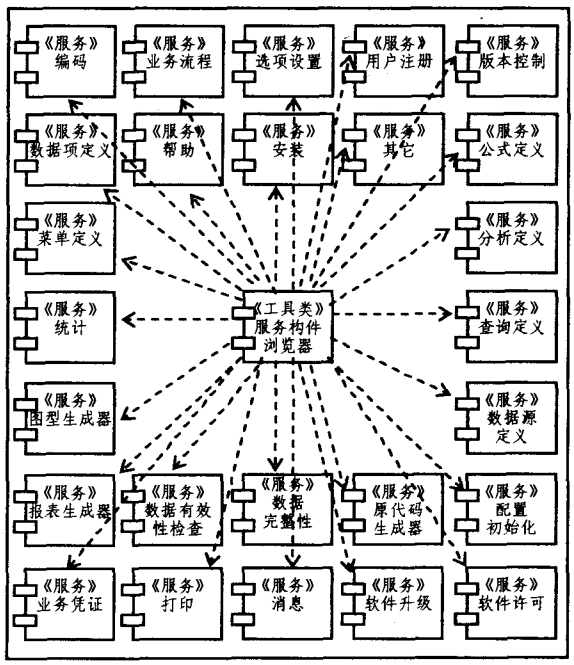

图 2 GERPToolkit 的服务构件图

3 面向服务体系结构建模

建模是一个贯穿软件体系结构开发生命周期的必要活动,它包括设计和需求度量、体系结构和软件系统的定义,以及为适合软件模型需要的各种建模类型进行分类。建模提供了一个有效方法来处理软件开发的需求和复杂性。

通过面向服务的分析和设计技术的应用,面向服务建模可使用有效的方法来鉴别 SOA 的基本方面。建模允许对 SOA 层的元素进行清晰的描述,并且提供一个环境对该层的元素进行可视化或者文本描述,以便为构造基于服务应用的体系结构提供一个重要的决策依据。

3.1 面向服务体系结构的建模过程和方法

基本上,面向服务体系结构的建模过程由三个步骤组成,它包括服务、构件与服务流的识别,以及规约和实现。其中,服务流也称为服务的编排(choreography)[11]。

服务识别(service identification)也称服务鉴别,主要通过业务域的分解、现有系统资源的分析,以及应用自顶向下或自底向上方法对需求功能的定义等过程来完成。

服务分类是服务规约(service specification)的必要活动,所有服务在被识别后都要被聚合(aggregate)到一个服务层。这个过程反映服务的粒度、使用范围、组合和分层。组成服务的构件在该阶段也应该被指定。

服务实现(service realization)包含面向服务应用中的服务识别。服务可能被应用于不同的目的,例如各种功能的集成,业务或数据服务,或者提供像安全和监控这样的特定功能。以适当的执行环境为目标,需要被鉴别使用这些服务的模块、构件和遗留软件系统。图 3 显示了上述描述的各种活动。

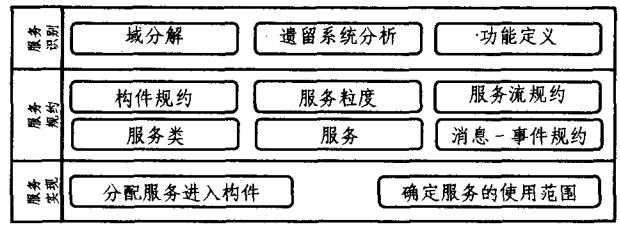

图 3 面向服务建模方法

上述的面向服务建模方法,为分析和设计活动中确定面向服务体系结构的基本方面提供了明确的方向。服务建模应该反映特定业务服务的创建,以及决策它们通过怎样的编排方式组合到应用中。

3.2 统一建模语言(UML)及其 UML Profile

UML 适用于各种软件开发方法、软件生命周期的各个阶段、各种应用领域以及各种开发工具。UML 包括语义概念、表示法和指导规范,提供了静态、动态、系统环境及组织结构的模型。它可被交互式的可视化建模工具所支持,这些工具提供代码生成和报表生成的功能。

UML 提供了几种扩充机制,包括约束(constraint)、构造型(stereotype)和标记值(tagged value),从而允许建模者在不改变底层建模语言的情况下做一些通用的扩充。

(1) 约束是对建模元素语义上的限制,可以关联到一个构造型定义,并用对象约束语言(Object Constraint Language,OCL)[12]来表达。OCL 基于一阶谓词逻辑,每一个 OCL 表达式都以 UML 模型元素为背景(由"self"引用),可使用该元素的属性和关系作为其项(term)。

OCL 表达式的语法用扩展巴科斯范式(Extended Backus-Naur Form,EBNF)定义。在 EBNF 中,"|"表示选择,"?"表示可选项,"*"表示零次或多次,"+"表示一次或多次。OCL 基本表达式的语法用 EBNF 定义如下:

PrimaryExpression :: = literalCollection | literal | pathname timeExpression ? featureCallParameters ? | "(" expression ")" | if expression
literal:: = ⟨string⟩|⟨number⟩ | "#" ⟨name⟩
timeExpression :: = "@" ⟨name⟩
featureCallParameters :: = "("(declarator)? (actualParameterList) ? ")")
ifExpression:: = "if" expression "then" expression "else" expression "endif"

从上面的定义可以看出,OCL 的基本表达式可以是一个 literal,literal 可以是一个字符串、数字或者是"#"后面跟一个模型元素或操作的名字;也可以是一个 literalCollection,literalCollection 代表 literal 的集合类型;还可以是一个包含选项的路径名,后面的可选项中包括时间表达式(timeExpression)、限定符(qualifier)或特征调用参数(featureCallParameters);还可以是一个条件表达式(ifExpression)。

(2) 构造型是 UML 中最重要的可扩充机制,提供了一

种在模型层中加入新的建模元素的方式，可以在构造型定义相关的约束和标记值以说明特定的语义和特征。所定义的构造型要求满足以下良构的规则，可用 OCL 表达式来描述。

① 构造型名不能与其基类(baseClass)重名：

Stereotype. oclAllInstances→forAll (st | st. baseClass <> self. name)；

② 构造型名不能与它所继承的构造型重名：

self. allSupertypes→forAll (st:Stereotype | st. name<> self. name)；

③ 构造型名不能与元类命名空间冲突，也不能和任何继承的构造型名或基类名冲突；

④ 基类名必须被提供，

self. baseClass<>''；

⑤ 构造型所定义的标记名不能与其基类元素的元属性命名空间冲突，也不能与它所继承的构造型的标记名冲突。

(3) 标记值是一种向模型中的建模元素附加新的非语义性属性的方法。标记表明了建模元素可扩展的特性的名称，值可以是任意的值，值的范围取决于用户或工具对标记值的解释。一个建模元素中的每个标记名，至多有一个标记值。

self. taggedValue→forAll(t1, t2:Taggedvalue| t1. tag = t2. tag implies t1= t2)

这些扩充能被组织为 UML profile，它是将基本元模型(metamodel)应用于一个特定平台或领域时，对一组有限的附加说明的定义[12]。现已存在众多的 UML profile，例如 CORBA Profile for UML 定义了使用 UML 为 CORBA 接口建模的特定方式；Java Profile for UML 定义了用 UML 为 Java 源代码建模的特定方式。

文献[14]给出了一个 UML 2.0 profile 例子。该例子用于演示服务、面向服务体系结构和面向服务解决方案的建模，其目标是提供一个描述企业级服务组合的公共语言。逻辑部分包含了服务规约，它代表了服务请求者和提供者，以及相关制品之间的约定。表 1 列出了用作 UML profile 构造型的 UML 2.0 元模型的元类(meta class)

表 1 SOA 的 UML 2.0 元模型元素

元类	构造型
类	消息，服务划分，服务提供者
分类器	服务消费者
协作	服务协作
连接器	服务信道
接口	服务规约
端口	服务，服务网关
属性	消息附件

UML 2.0 profile 归纳了每个构造型，其细节部分规定了它的元类、属性和应用限制。例如，一个构造型《消息》可看作为一个类，它为 WSDL 文档中所定义的概念表达提供语义，且含有一个"编码"的属性名。构造型《服务》也能被描述为它的扩充端口，为服务请求者和提供者之间交互的终端实现提供语义。

UML 2.0 profile 主要基于构造型来对模型的元素进行分类。使用构造型有助于模型的可读能力，同时可为模型转换工具的行为提供指南。UML 2.0 profile 适合定义一个特定的领域、技术或者建模实现。

3.3 模型驱动体系结构(MDA)

MDA 是 OMG 提出的一种基于平台无关的概念性框架，它允许对特定的业务抽象层进行建模，并为定义不同模型类型间相互转换、建模演化和模型驱动开发技术等提供自动化的支持工具。MDA 在开发中将软件功能与功能实现分离，支持创建类似系统时重用系统模型，使软件开发对新技术变化的适应能力更强，以解决由于新技术、新平台的出现，原结构设计不能再用的问题。

MDA 基于几个开放的 OMG 的建模标准，包括 UML、MOF、CWM 和 XML。MDA 依据这些标准为企业应用建立独立于实现技术的平台无关模型。其中，UML 用于识别应用开发模型和转换；MOF(Meta Object Facility)用于对元模型进行清晰的描述和定义，以便有能力对模型进行分析和自动转换；CWM(Common Warehouse Metamodel)跨越了设计、构建和管理数据仓库应用的整个生命周期，并支持对生命周期的管理；XMI(XML Metadata Interchange)使用 XML 标记语言在工具、数据库和应用之间共享和交换 UML 信息。

MDA 的核心思想以模型为中心，根据参考平台的不同，模型可以分为平台无关模型(Platform Independent Model, PIM)和平台特定模型(Platform Specific Model, PSM)。当开发者应用 MDA 来构建一个应用系统时，首先选择合适的 UML profile 来创建 PIM，并用 UML 或其它任何一个计算机能解释的定义良好的建模语言来描述。然后由平台专家遵循一定的转换规则把 PIM 映射到一个或多个 PSM，它包含了与选定平台和技术相关的实现细节。当在两个平台间建立标准化的映射规则之后，可以生成映射工具，自动完成 PIM 和 PSM 的相互转换，并进一步生成目标平台上的代码。

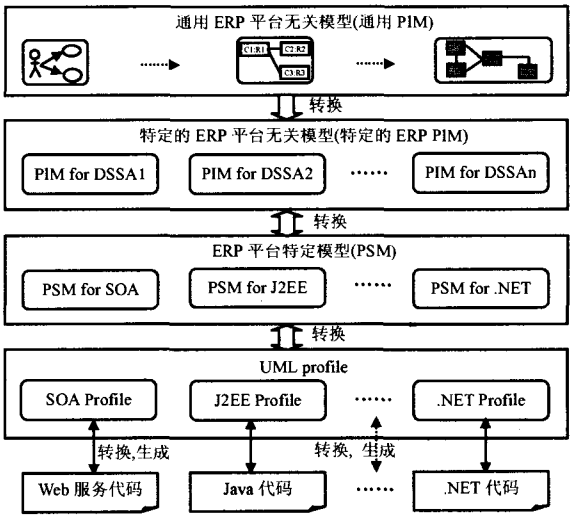

图 4 应用 MDA 到 ERP 系统开发过程

MDA 允许为良好的定义接口和基于 SOA 应用开发中的服务进行建模。图 4 显示的是应用 MDA 到面向服务流通企业的 ERP 系统开发过程实例。GERPToolKit 平台为每个特定领域的软件体系结构(Domain-Specific Software Architectures, DSSA)增加一个叫特定的 ERP PIM，并进入它的开发生命周期。所有这些特定的 PIM 又将被集成进入一个通用 PIM 中。使用各种转换规则，可分别实现从通用 PIM，特定的 ERP PIM、PSM 到 UML profile 的变换，进而可易如反掌地生成最终可部署代码。

4 模型驱动的 Web 服务的特性描述

模型驱动开发中，模型是用来描述一个系统的业务关注点、用户需求、活动、信息结构、构件和构件交互。这些模型贯穿系统的开发，并能转换为程序代码。就 ERP 软件中的 Web 服务构件开发来说，模型被转换为 Web 服务的描述语言。

UML是当前软件设计的建模标准，由于提供容易理解的图形化的表示法，因而成为一个理解Web服务描述的好选择。在使用UML为模型驱动的Web服务建模时，为了避开对Web服务中的一些关键概念进行有效检查，需要定义两类Web服务的特性描述，即基本特性描述Basic Profile和流程特性描述Process Profile。这两类特性描述主要是基于功能和有关团体的标准，例如OASIS(Organization for the Advancement of Structured Information Standards)、OMG、W3C和WS-I所提出的标准。图5展示了Web服务特性描述的构成示意图。

(1)Basic Profile遵循由WSDL、SOAP和UDDI三元组所描述的Web服务。目前，有许多各类Web服务可利用，重用现有的Web服务并进而创建组合的Web服务是领域ERP系统软件开发的一个关键。由于UDDI工具开发仍然稀少，而SOAP建模有诉求限制(由于访问和部署到SOAP引擎的问题)，因此Basic Profile的支配标准倾向是WSDL。

(2) Process Profile代表扩展的Web服务接口。这些扩展接口是基于现有的各种标准，例如BPEL4WS、OASIS的ebXML业务流程规范模式(ebXML Business Process Specification Schema，ebBPSS)以及W3C的Web服务编排接口(Web Services Choreography Interface，WSCI)。

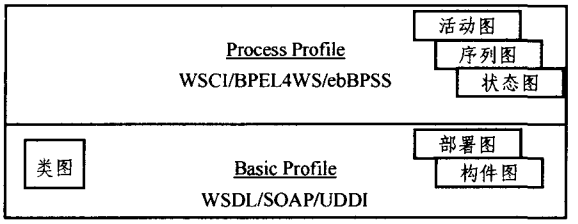

图5 Web服务特性描述

BPEL4WS是规定基于Web服务的业务流程行为的一个表示法。BPEL4WS的流程导出和导入功能使用Web服务接口。借助BPEL4WS，能建模较早描述的可执行的和抽象的业务流程行为，以及为业务交互协议的形式化规约提供一种语言。它也扩充Web服务交互并使它能支持业务事务。

ebBPSS提供业务流程规范的一个标准框架。它同ebXML的Collaboration Protocol Profile (CPP)和Collaboration Protocol Agreement (CPA)规范一起，在业务流程建模和使用ebXML的电子商务软件之间架起一座桥梁。

WSCI是一个编排标准，原为Sun Microsystems、Intalio和BEA等公司定义，现正提交给W3C做更广泛的修改和评估。

Web服务的目标是方便平台无关的和松耦合的应用系统的交互和集成。WSDL是目前描述和定位Web服务的标准方式，SOAP是用来处理跨平台的交互，而BPEL4WS是规定基于Web服务的业务流程行为。考虑到WSDL、SOAP绑定和BPEL4WS等的特性描述有相同的结构，并由于简洁的缘故，本文仅限对WSDL的特性描述进行讨论。

5 一种面向WSDL的UML Profile

5.1 WSDL建模的关键问题

Web服务的应用体验清楚地表明Web服务时常受交互问题、安全的支持不足，以及其它限制所困扰[15]。这些问题部分是由于标准的演化，允许自由地使用它们，并缺少使用指南。许多研究人员正在从事克服这些问题的研究。

作为Web服务的接口描述语言(Interface Description Language，IDL)的WSDL是体现Web服务交互性的一个关键[16]。许多Web服务工具包允许通过现有的准备暴露给Web服务客户的接口，实现自动生成WSDL文档。然而，这些方法存在一个服务接口的假定，且不支持Web服务描述的设计，也不是Web服务的客户视角。此外，自动地生成服务描述仅能反映现有的接口(例如Java)。即使接口很容易地开发和定义，开发人员也往往失去对WSDL构造的一些控制。这种方便来自灵活性的代价：服务接口的变化往往要求WSDL文档变化，相应地要求客户端调整。另一个方法是考虑从服务描述的构造作为出发点，如允许从WSDL生产部分的Java接口。这些方法对服务设计者提出了更多的要求，如WSDL语法的知识和Basic Profile的推荐。

针对上述问题的不足，并考虑到目前的面向服务ERP系统集成中开始存在大量的基于Web服务的遗留系统现状，本文提出了如下的一种基于UML的WSDL建模思路：

(1) 转换WSDL文档到UML表示法的支持工具；

(2) 基于UML的特性描述来捕获WSDL文档的结构规则，其中包括SOAP绑定，以及WS-I Basic Profile规则和WSDL的有关推荐；

(3) 可定制工具在设计UML模型或逆向工程时，应支持和运行各种有效性检查。

表2 两阶段WSDL建模所需的UML视图

阶段	UML视图
平台无关	类视图
平台特定	类视图，部署视图，构件视图

5.2 WSDL的层次结构

一个WSDL文档包含三部分描述：Web服务使用的数据类型、服务执行的操作和到一个通信协议的绑定。WSDL 1.1为SOAP，HTTP GET/POST和MIME定义了绑定扩充。既然SOAP绑定是最常用的，WS-I Basic Profile限制WSDL绑定到SOAP，本文也只考虑SOAP绑定。

为了捕获WSDL有关的限制，使用了四个特性描述：SOAP绑定、WSDL、WS-I和XML Schema。每个特性描述，除了WS-I Profile之外，都包含构造型定义和限制定义这两个部分。前者定义结构特性描述，以及实际的UML模型和视图中使用的构造型。后者规定视图所必须遵循的限制和规则。WS-I Profile被分成正在SOAP绑定、正在Web服务定义和正在XML Schema三部分，且大部分规则都使用OCL建模。对结构特性描述进行分层，有利于开发者在对服务进行描述时，灵活和有效地选择执行什么样的有效性检查，即他们能自由地挑选一个特性描述或者多个特性描述的联合对某个WSDL文档进行有效性检查。例如，WSDL和SOAP绑定的特性描述能被挑选用来评估一个包含SOAP绑定的WSDL文档的正确性。

鉴于WSDL、SOAP绑定和XML Schema的结构特性描述有同样的结构，本文仅描述WSDL和WS-I的结构特性描述。

5.3 WSDL的建模过程

受MDA开发方法的启发，WSDL的建模可划分为如下两个阶段：

(1) PIM阶段。该阶段的建模视角为WSDL的抽象部分，其中包括Definitions(定义)、Types(类型)、Message(消息)、PortType(端口类型)、Operation(操作)、Part(部件)和Fault(异常)；

(2) PSM阶段。该阶段完成WSDL的绑定部分，主要建模元素包括Service(服务)、Port(端口)和Binding(绑定)。

这种划分阶段的优点是它的抽象能力，每个阶段只关注

相关问题。PIM 阶段要求交互式的建模，同时一旦定义好部署平台，大部分的 PSM 实体能被自动地生成。表 2 给出了 WSDL 的两阶段建模所需的相应 UML 视图。

5.4 WSDL 的特性描述元素

(1)构造型。图 6 使用 UML 类图形式给出了一个两阶段的 WSDL 构造型定义，同时展示了所有新的建模元素都来自对 UML 元类"类"的扩充，它们代表了 WSDL 规约中的关键元素。图 6 中类有两种构造型：《元类》和《构造型》，前者是指标准的 UML 元模型中的构件元素，后者构造型类是对元模型元素的一个扩充，且该构造型必须被应用到构件上，即与构件元素是一个依赖(dependency)关系。

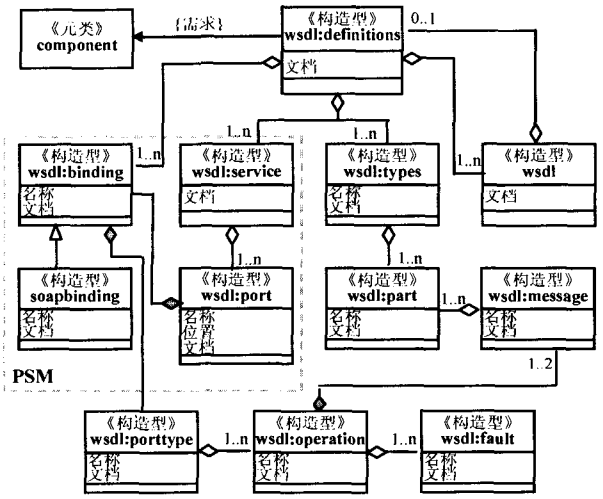

图 6 WSDL 构造型

(2) 限制。WSDL Profile 的限制主要是基于 WSDL 规约，且确保和 WS-I Basic Profile 一致。一个可能的限制例子是"禁止编码规则"，使用 OCL 表达如下：

context bndgSOAP: SOAPBinding inv:

bndgSOAP. encoding = "Literal"

(3) 标记值。PSM 阶段要求一些潜在环境的知识，这里包括 SOAP 引擎(engine)和构件容器(container)。从而一对有意义的标记定义可能是标记 SOAP 引擎的标记值是 JAXRPC 或者 Axis；标记容器的标记值可能是 J2EE 和. NET。

5.5 WS-I 规则的结构特性描述

WS-I(Web Service Interoperability)组织当前发布的 Basic Profile 1.0 的基本前提是确保 Web 服务能够进行通信。该 Basic Profile 并没有定义任何应用语义，相反地，它确保了消息可以在任何请求者和提供者之间进行交换。Basic Profile 为 WSDL 描述了最佳的使用情况，以此来描述可互操作的信息。为此，需要转换 WS-I Basic Profile 1.0 规则(关于 WSDL 和 SOAP 绑定)到 UML，更精确地说，到一个结构特性描述。例如，Basic Profile 1.0 的名称空间(namespace)规则 R2304 如下："一个描述中的每个 wsdl:portType 都至少包含一个名字属性具有唯一值的操作"。转换到 WS-I Basic Profile 的结构特性描述中，R2304 能用下面的 OCL 格式表达，它和构造型" wsdl:operation"的类操作相关联。

context operation inv:

operation. allinstances->forAll(c1, c2 | c1<>c2 implies c1. name <> c2. name)。

对少数不方便使用 OCL 表达的 WS-I 规则，可通过对应的操作实现。例如，Basic Profile 的规则 R4004："描述必须使用 W3C 推荐的 XML 1.0 版本"。当使用某个 XML 解析器分析一个独立描述所从属的 WSDL 文档时，不管该独立描述是否是一个有效的 XML 1.0 文档，它都需要被检查。

结束语 本文探讨了利用 UML 进行面向服务体系结构的建模和表达。UML 是一种功能强大的、面向对象的可视化建模语言，它采用了一整套成熟的建模技术，广泛地适用于各种应用领域。但由于 UML 的标准表示法不能完全满足面向服务软件体系结构建模的需要，因此有必要用 UML 的扩展机制对其表示法进行扩充，为具有 Web 服务体系结构意义的模型元素添加新的语义，构造出新模型元素，用以说明面向服务软件体系结构的特点。

考虑到重用现有的 Web 服务并进而创建组合的 Web 服务是面向服务领域 ERP 系统软件开发的一个关键，其基础是使用 UML 建模 WSDL。本文讨论了模型驱动的 Web 服务的特性描述，在此基础上提出了一类 UML profile 来定义 WSDL 文档的结构规则和 WSDL 描述的 Basic Profile 推荐。使用这些 UML profile 可帮助正确地设计符合 Basic Profile 的 WSDL 文档，同时可以用来检查现有 WSDL 描述的有效性。该建模方法的主要优势如下：(1)抽象层次高，有个可重用的通用 PIM；(2)可扩展性好，出现新的特定应用域，开发焦点转移到面向该特定领域的 PIM 而获得；(3)适应性强，移植到另一种实现技术，仅改变转换规则，并使用新技术重新实现系统的一些特性描述；(4)生产效率高，借助工具，能自动完成 PIM 和 PSM 的相互转换，并进一步生成目标平台上的代码。

面向服务软件体系结构是软件工程中的一个新兴领域，其本身还在发展完善中。UML 本身以及用于体系结构建模 UML 扩展机制目前仍处于发展过程之中，仍需对其进一步研究，以便在不久的将来，使 UML 成为面向服务软件体系结构的可靠的建模工具。

参 考 文 献

[1] Medvidovic N, Taylor R N. A classification and comparison framework for software architecture description languages. IEEE Transactions on Software Engineering, 2000, 26(1):70-93

[2] Stojanovic Z, Dahanayake A, Sol H. Modeling and design of service-oriented architecture// Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. New York, NY, USA: IEEE Press, 2004:4147-4152

[3] Kruchten P B. The 4 + 1 view model of architecture. IEEE Software, 1995, 12(6):42-50

[4] Kobryn C. Modeling enterprise software architectures using UML//Proceedings of the second international Conference on enterprise distributed object computing workshop. New York, NY, USA: IEEE Press, 1998:25-34

[5] Gronmo R, Skogan D, Solheim I, et al. Model-driven Webservices development//Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service. Los Alamitos, CA, USA: IEEE Computer Society Press, 2004:42-45

[6] OMG. UML superstructure specification version 2.0. http://www.omg.org/cgi-bin/doc? formal/05-07-04.pdf, 2004

[7] 蒋哲远，韩江洪，王钊. 面向 Web 服务的 ERP 协同集成框架研究. 计算机工程与应用，2005，41(16)：24-28

[8] Jiang Z Y. Service-oriented ERP system modeling using UML// Proceedings of the 2005 International Symposium on Computer Science and Technology. Marietta, GA, USA: The American Scholars Press, 2005:451-455

[9] 李相育，钱宇，李秀喜. 基于 Web 服务的流程工业遗留系统的信息集成. 计算机集成制造系统，2005，11(10)：1387-1391

[10] 蒋哲远，韩江洪. 面向服务领域软件的参考模型. 计算机科学，2007，34(8)：261-265

[11] Peltz C. Web services orchestration and choreography. Computer, 2003, 36(10):46-52

[12] OMG. UML 2.0 OCL Specification. http://www.omg.org/docs/ptc/03-10.14.pdf, 2003

[13] Rumbaugh J, Jacobson I, Booch G. The Unified Modeling Language Reference Manual. Second Edition. Boston, MA, USA: Addison-Wesley, 2004

[14] Johnston S. UML 2.0 profile for software Services. http://www-128.ibm.com/developerworks/rational/library/05/419_soa/, 2005

[15] Wang H B, Huang J Z, Qu Y Z, et al. Web services: Problems and future directions. Web Semantics: Science, Services and Agents on the World Wide Web, 2004, 1(3):309-320

[16] Jiang J, Systa T. UML-based modeling and validity checking of Web service descriptions// Proceedings of the IEEE International Conference on Web Services. Los Alamitos, CA, USA: IEEE Computer Society Press, 2005:460-467