

# SOA 研究进展<sup>\*</sup>)

邢少敏 周伯生

(北京航空航天大学计算机学院 北京 100083)

**摘要** 近年来,面向服务的体系结构(SOA)逐渐成为软件工程领域的研究热点,它在企业信息系统集成、分布式软件系统开发方面都具有明显的优势,也是应对企业灵活多变的业务需求挑战的关键技术。总结了面向服务的体系结构技术发展过程及其主要研究方向,介绍了当前面向服务的体系结构的研究进展,应用实践。并且在分析了典型的面向服务的体系结构之后,给出了面向服务的体系结构的定义。通过分析和总结面向服务的体系结构研究领域的主要研究方向和内容,提出了面向服务的体系结构研究的思路。探讨了当前面向服务的体系结构研究中的不足之处,并预测了面向服务的体系结构进一步的研究方向。

**关键词** 服务,面向服务的体系结构(SOA),体系结构,服务组合

## Progress of Research on Service-Oriented Architecture

XING Shao-min ZHOU Bo-sheng

(School of Computer Science and Technology, BeiHang University, Beijing 100083, China)

**Abstract** Recently, Service-oriented Architecture (SOA) is emerging as one of the primary research areas in software engineering and one of the key technologies to the integration of enterprise information system, development of distribute software system and tacking the mutative requirements. The history and main research aspects of SOA are summarized, and the recent research progress and application practices are introduced. A definition for SOA is brought up based on analyzing and comparing of several classical definitions about SOA. Based on analyzing and summarizing the main research aspects and content of SOA, directions of research on SOA are proposed and some disadvantages of current research on SOA are discussed. Finally, it is concluded with some significantly promising tendency about research on SOA.

**Keywords** Service, Service-oriented architecture, Architecture, Service composition

## 1 SOA 的起源与发展

### 1.1 SOA 的起源

近年来随着软件的快速发展,大多数企业都已经有了各种不同的不同时期、不同平台、不同技术体系的系统和应用程序。为了消除各个系统之间彼此孤立的问题,集成这些不同厂商、不同平台之间的系统的需求就越来越紧迫。这种异构系统的集成往往非常困难,然而一个企业又不可能单单使用一家厂商的技术,异构的问题是不可避免的。另外,激烈的竞争使得产品的周期大大缩短了,而产品的需求也变得更加多变,这就要求软件开发要快速响应需求变化,即随需应变。现有的面向对象语言、框架、设计模式尚不能完全解决软件开发所面临的异构和灵活多变的需求问题。现有的面向构件的开发方法也只能在功能、结构完全或近似匹配的环境下才能较好地复用,当构件模型不同,运行平台异构时,开发出来的软件很难具有适应性。Internet 环境下这种矛盾更加突出,要真正实现基于 COTS 构件的组装式开发还有许多问题需要解决<sup>[1]</sup>。传统技术由于其紧耦合的本质,注定不能从根本上解决这些问题。在这种情况下,SOA(Service-oriented architecture,面向服务的体系结构)应运而生。

### 1.2 SOA 的发展历程

服务的概念早在 CORBA 出现时就产生了。JINI<sup>[2]</sup> 的展

又进一步推动了 SOA 的早期概念体系的建立。Gartner 在 1996 年第 1 次阐述了 SOA 的概念,并认为未来 SOA 会成为占绝对优势的软件工程方法,从而结束已长达 40 多年的传统软件体系架构的主导地位<sup>[3]</sup>。但是由于当时技术发展的限制,始终没有广泛应用。近几年以来,随着 Web Service、XML 技术的发展,实现 SOA 时机逐渐成熟,并且由于 Microsoft、IBM 等厂商不遗余力的推崇,SOA 已经成为企业应用中的核心概念之一<sup>[4,5]</sup>,并成为软件工程技术发展的重要趋势<sup>[6]</sup>。

SOA 在刚开始的时候,它的关注点在企业应用集成和厂商相关的解决方案上,这个时期对于 SOA 的研究主要集中在软件系统集成和整合方面,紧接着 SOA 的发展逐步过渡到以 Web Service 以及 Web Service 组合为主要研究关注点,现在 SOA 的研究重点在业务流程管理以及业务流程的开放标准方面。

## 2 SOA 的研究现状

### 2.1 SOA 的有关定义

若干学者提出了自己的面向服务的体系结构的定义。

文献<sup>[7-9]</sup>分别从不同的侧面讲述了面向服务的体系结构的基本概念、技术、设计和应用。是系统地阐述面向服务的体系结构的著作。

<sup>\*</sup>) 本文得到北京市科技计划课题(y0105001040521)支持。邢少敏 博士研究生,研究方向为面向服务软件体系结构、过程建模;周伯生 博士生导师,主要研究方向为软件工程、软件过程改进。

文献[7]提到服务有八条基本原则,其中,自治性、松散耦合、抽象、以及需要正式契约被视为形成 SOA 根本基础的核心原则。文献[8]定义 SOA 是一种设计方式,它指导着业务服务在其生命周期中包括创建和使用的方方面面。SOA 也是一种定义和提供 IT 基础设施的方式,它允许不同应用相互交换数据、参与业务流程,无论它们各自背后使用的是何种操作系统或采用了何种编程语言。其中文献[9]定义 SOA 是一个软件架构,它包含四个关键概念:应用程序前端、服务、服务库和服务总线。一个服务包含一个合约、一个或多个接口和一个实现。

文献[10]对比了面向服务的体系结构和现存技术的不同和相同之处,用参照的方式陈述了面向服务的体系结构的概念。

IBM 的定义<sup>[4]</sup>:面向服务的体系结构是一个组件模型,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。这使得构建在各种各样的系统中的服务可以以统一和通用的方式进行交互。

其它定义中也提到,SOA 是一种架构模型,它可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。服务层是 SOA 的基础,可以直接被应用调用,从而有效控制系统中与软件代理互联网络的人为依赖性。SOA 的几个关键特性包括:一种粗粒度、松耦合服务架构,服务之间通过简单、精确定义适配器进行通讯,不涉及底层编程适配器和通讯模型。

根据以上定义和我们对于面向服务的体系结构研究和理解,我们认为:面向服务的体系结构是一种软件工程方法学,是一种新的软件开发范型,它以松散耦合的粗粒度服务为基本单元,通过开放标准的独立于硬件平台、操作系统和编程语言的接口发布服务,并通过开放式通信协议进行通信,服务可以某种方式组合起来,构成和跨平台、具有一定适应性的软件系统。

## 2.2 SOA 研究的主要内容与进展

目前,SOA 的研究主要集中在以下几个方面:

### 2.2.1 理论模型

#### a)参考模型

为了更好地促进未来系统间的互联互通,需要对基于 SOA 的企业应用系统的开发进行规范和指导。进行 SOA 参考模型的研究的意义正在于此。各个研究机构和各个公司提出了若干概念参考模型,作为面向服务的体系结构的指导方针。OASIS 在 2006 年初提出了 SOA 参考模型<sup>[11]</sup>,并在年底成为标准。OASIS 参考模型致力于 SOA 参考模型的标准化,其目标是定义最小的一组 SOA 核心概念,并确定它们之间的关系,用于提供构造具体 SOA 的共同语义,该参考模型处于概念定义与概念关联的层次,主要用于提供对 SOA 进行共同的理解,没有阐述如何运用这些概念深入地构造 SOA 的基本结构(如服务总线 and 组织机构等)。针对目前还没有公认的企业服务总线的状况,CBDI 提出了一个 SOA 参考模型,其中给出了企业服务总线的主要构成模块,但偏重于如何进行实现,并主张全面的支持 Web 服务<sup>[12]</sup>。各大公司也有自己的 SOA 参考模型<sup>[13,14]</sup>。国内也有北京大学麻志毅等提出了一个与具体实现技术无关的 SOA 参考模型<sup>[15]</sup>,该模型不直接受任何具体标准、技术或其它实现细节的约束,描述了应用服务的

环境设施以及其中的部件和部件之间的关系,是一个用以指导建立具体的 SOA 的抽象框架。

#### b)服务接口模型

在面向服务的软件体系结构中,服务提供者向服务注册机构发布服务的接口信息,该信息描述了这个服务对于外界环境的要求和它能够对外界提供的服务,如何保证接口信息的准确性对于面向服务系统的构建非常关键。形式化的接口描述是保证其准确性的有效手段,也是对其进行测试或验证的前提。因此,建立服务的形式化接口模型受到了关注。文献[16]提出了接口自动机描述接口之间的时序行为;文献[17]提出了时间接口,它对具备时间约束的接口时序行为进行了描述。以此为基础,文献[18,19]中提出了一种服务接口模型,对 3 个抽象层面进行接口信息描述,并在给出的每个层面接口的相容性和可替换性条件下提出了接口规范及其验证方法。

事务性是面向服务系统的一个非常重要的特点,面向服务的事务模型与传统的原子事务模型有着很大的区别<sup>[20]</sup>,前者往往在执行时间上比较长,而且很多操作不具备传统事务模型的原子性和可回滚性,对于长事务(long-running transactions)模型,需要提供补偿和错误处理机制对其发生的异常进行处理,因此,很多学者在事物补偿和错误处理机制方面展开了研究。文献[21]使用 Guarded 有限状态机对单个服务接口建模,然后将模型转换到 Promela 语言,并在 SPIN 上进行验证,以保证接口行为的正确性。这种方法在对接口层面的事务模型描述能力上存在不足,对 BPEL<sup>[22]</sup>中错误处理和补偿行为的描述不够精确,比如,它们不能处理 BPEL 中的多层作用域,同时也没有考虑 BPEL 中默认的错误处理和补偿机制。基于面向构件的接口理论,Butler<sup>[23]</sup>使用补偿通信顺序进程(compensating CSP,简称 cCSP)来描述长事务模型。补偿通信顺序进程是通信顺序进程的一种变体,它提供了补偿进程来描述补偿行为,同时使用中断进程、中断处理和产生中断进程对错误发生及错误处理行为进行描述,并且还提供了事务块进程来表示独立的子事务,但是并没有对并发进程中的分支进程发生错误后的进程同步策略进行讨论,而这一点在事务处理中非常重要。

#### c)面向服务的体系结构形式化描述

面向服务的体系结构本身也是一种软件体系结构,所以可以把软件体系结构研究方法和研究成果用在面向服务的体系结构的研究上。因此,很多对于面向服务的体系结构的研究也都是在体系结构这个层面上来做的。形式化描述可以保证面向服务的体系结构实施过程中的准确性,方便形式化验证。文献[24]在体系结构层次使用 Z 语言对 SOA 进行了形式化的描述,给出了服务组件和匹配和组合的形式化定义,分析了体系结构风格的可替换性。文献[25]提出了一个面向服务的轻量级形式化框架,用于捕获 SOA 中的主要组件,分析 SOA 的结构。这些研究对提高 SOA 的准确性和形式化验证能力上起到了推动作用。同时也属于软件体系结构研究的范畴。关于软件体系结构的研究,有专门的综述性文章,本文不再详细讨论。

### 2.2.2 服务建模、设计、发现、组合

面向服务源于软件工程理论的“关注点分离”的思想。也就是说将一个大的、复杂的软件问题分解为一系列单个关注点,单个组件,使得逻辑将需要解决的问题分解成更小的、相关片段的集合。每一段逻辑处理一个特定的关注点。这个思

想已经被不同的开发平台以不同的方式实现。例如,面向对象的编程与基于组件的编程方法,通过使用对象和组件而实现了关注点分离。面向服务能够被视作以截然不同的方式来实现关注点分离。这种不同就在于它不是以细粒度的对象和基于私有接口协议的组件为基础的,而是以粗粒度、基于开放标准协议的服务为基础的。尽管设计模式、程序框架大大地完善和发展了面向对象的理论,但是由于对象粒度大小,在当今互联网迅猛发展的情况下,所能提供的复用性、灵活性相当有限。而基于分布式组件的编程,比如 DCOM, COBRA 等都不是基于一个广泛认可的协议,导致使用这些组件模型开发出来的软件在互联互通上存在一定的难度,也造成了很多孤立系统的存在,产生了所谓的“信息孤岛”。而且组件之间耦合非常紧密,一旦需求发生变更,应对需求变更往往需要花费很大代价。而服务是粗粒度的,大到一个企业具体的一项业务都可以成为一个服务,这样软件复用的程度大大提高,而且服务是松耦合的,一旦需求变更,只需要花很小的代价就可以随需应变。

一般的认为服务的基本原则有以下八条<sup>[7]</sup>:

- 服务可复用 不管是否存在复用的机会,服务被设计为支持潜在可复用。

- 服务共享一个正式契约 为了与服务交互,只需要共享描述每个服务信息交换术语定义的正式契约。

- 服务是松散耦合的 服务被设计为无需紧密的、跨服务的依赖而交互。

- 服务是底层逻辑的抽象 只有经由服务契约所暴露的部分服务对于外部世界是可见的。契约之外所表达的底层逻辑是不可见的,且与服务请求者无关。

- 服务是可组合的 服务可能组合其他服务。这允许表示不同粒度的逻辑,并促进复用及抽象层的创建。

- 服务是自治的 逻辑由服务所控制,并位于一个清晰的边界内。服务已经在这个边界内被控制,并且不依赖于执行其控制的其他服务。

- 服务是无状态的 服务应当不需要管理状态信息,因此能够其维持松耦合性。服务应当尽可能设计成无状态的,即便这意味着要将状态管理移至别处。

- 服务是可发现的 服务应当允许其描述被发现,并被人工和可能会利用其逻辑的服务请求者所理解。

在这八条原则中,自治性、松散耦合、抽象、以及需要正式契约被视为形成 SOA 根本基础的核心原则。

#### a) 服务建模

SOA 是一种新的软件体系结构,包含有服务提供者提供的可帮助服务消费者实现各种功能的服务,由于 SOA 松耦合和粗粒度的本质特性,服务应该在独立于实现的较高抽象层次进行描述。目前对服务建模的研究主要有:利用 UML 进行服务建模<sup>[26,27]</sup>,利用自定义语言对服务行为建模<sup>[28]</sup>,利用体系结构描述语言进行服务建模<sup>[29]</sup>,研究 Web Service 的建模框架。Web 服务建模方面相对成功的模型是 OWL-S 和 WSMO。后面会有详细讨论。

#### b) 服务设计

服务的构件要考虑到当前大量使用面向对象或面向构件方法开发出来的大量遗留构件的复用问题,因此将现有的非面向服务的构件转化或者扩展为服务组件,为复用大量现有构件提供有效的途径。文献[30-32]都在这个方面作了一定的研究。除了要兼容以前的遗留构件外,服务的设计研究主

要应该是使用新的技术和标准来开发服务构件,在这个方面已经有了一些研究成果。比如 OSOA 联盟的 SCA、SDO 规范,Microsoft 的 WCF 以及 SUN 公司的 JBI 规范。这些规范为服务的设计和开发提供了方法、规范,对于面向服务的发展起到了实际的推动作用,是目前服务设计和开发方面的主流,也是整个面向服务的软件体系结构方面实质性的探索和研究。

#### c) 服务发现

Web 服务发现旨在实现从互联网中发现 Web 服务并且众多的 Web 服务中定位到满足要求的服务,服务发现的效果直接关系到服务调用的质量,影响服务组合的相容性和可替换性,关系到能否真正实现服务的即插即用。由于传统的 Web 服务发现机制查准率较低,Web 服务发现的研究者引入了语义网(Semantic Web)技术<sup>[33]</sup>加强服务描述信息的机器可理解性,支持用户需求和能力之间的逻辑推理匹配;利用 P2P 技术来处理服务元数据的交换,克服传统 UDDI 技术中服务元数据集中注册、集中存放对搜索广度带来的限制。值得一提的是,文献[34]是国内比较系统的总结和研究服务发现的文献,它系统地总结了服务发现的相关技术,并在此基础上提出了基于服务质量 QoS 的服务匹配算法,服务选择模型,提出了 Web 服务发现的体系结构,提出了 Web 服务发现的原型系统 starWSDS。服务发现是面向服务的体系结构的研究热点,也是难点,这方面需要解决的问题还很多,未来几年内仍然会有很多研究针对这个方面。

#### d) 服务组合

服务组合是面向服务的体系结构的研究中的热点和重点,面向服务的体系结构中的大量的研究都是关于服务组合的。大体上服务组合可以分为以下几大类:基于流程的方法、基于语义的方法、基于形式化方法、基于规划的方法、基于 AOP 的方法等等。

##### (1) 基于流程的方法

知名度比较高的研究项目有 HP 实验室的 eFlow、美国乔治亚大学的 METEORS 和 IBM 印度实验室的 Symphony。国内从事此项研究的单位有北京航空航天大学、国防科大、中科院软件所等,其中北航的 WebSASE 完成较早,影响较大。目前基于流程的研究大多基于 BPEL, BPEL 是最有影响力,最广泛接受的流程语言,并已经成为标准。此外 OWL-S, 是一个比较有名的基于语义的流程语言,在学术研究中占有重要地位。

##### (2) 基于语义的方法

为了实现服务的动态组合,一种方法是基于系统中已有的服务组合模板,它选择能够匹配该模板的具体服务<sup>[35-38]</sup>。这些方法缺点是服务模板需要人为干预,降低了组合自动化的程度。因此,一些学者提出了基于具体服务,直接生成服务组合的方法,该方法的好处是可以根据服务请求即时地进行服务组合而不需要预定义的组合模板,实现了服务组合的自动化。但是,现有的很多方法在组合过程中都是基于关键字的匹配<sup>[39-41]</sup>,没有考虑到服务的语义,具有很大的局限性。另有一些方法考虑了服务的语义,比如文献[42,43],它们的做法是首先对服务进行语义扩充,然后使用“最短路径”算法获得最优的服务组合,但是为了得到最优的组合,其算法效率受到了影响。文献[44]提出了一种基于领域本体的服务动态组合算法。该方法利用领域本体及其推理能力,生成一个优化的服务组合图,然后基于该图给出了一个高效的服务组合

算法。与已有的方法相比,该方法综合考虑了服务语义、服务组合的质量和效率等因素,可以自动地根据服务请求实现服务的动态组合,同时保证了服务组合的效率和效率和质量。

### (3) 基于形式化方法

形式化方法对于建模和验证软件系统是一种有效的方法,对于 Web 服务组合来说,保证其组合的正确性以实现其服务增值是十分必要的。文献[45]使用 Pi-演算对 Web 服务及其组合进行了形式化描述和建模,分析了 Pi-演算应用于 Web 服务组合需要解决的问题,讨论了 Pi-演算与 Web 服务协议栈的对应关系,说明了利用 Pi-演算建立 Web 服务组合模型的规则,指出了如何寻找代理和通道。最后给出了一个实际的模型,并利用形式化工具对建立的组合模型是否正确以及是否满足需求进行了验证。Nakajima 描述了如何使用 SPIN 模型检查工具来验证 Web 服务组合。其组合语言为 WSFL。为了使用 SPIN 工具来验证,首先要将 WSFL 描述的业务流程转换成 Promela 规范语言来描述,SPIN 提供模型检查功能,但不提供等价性的检查。Karamanolis, Giannakopou, Magee 和 Wheeler 将 Web 服务流程转换成 FSP 进程 (Finite State Process) 并使用 LTAS 工具 (the Labeled Transition System Analyzer) 进行模型检查<sup>[46]</sup>。LTAS 工具允许用户以确定性 FSP 进程来指定属性,在文献[47]中, Foster, Magee 等人描述了 LTAS 工具的 BPEL4WS 插件。他们把 BPEL4WS 程序转换成 FSP 进程并使用 LTAS 工具来验证 FSP 进程。Koshkina 和 vanBreugel 引入基于 BPEL4WS 的 BPE 演算<sup>[48]</sup>。使用 PAC 工具 (Process Algebra Compiler) 和 CWB-NC 工具 (Concurrency Work bench of the New Century) 来建模和验证 Web 服务协作。BPE 演算包含了 BPEL4WS 的控制流结构,基于结构化操作语义, BPE 演算作为 PAC 的输入,其输出作为 CWB 的输入,从而验证服务组合。此外,基于 Petri 网的 Web 服务组合建模方法也有相关研究在开展,文献[49]中提出的 WS-net 是一种可执行的体系结构描述语言,它支持着色 Petri 网的语义以及面向对象概念。在 WS-net 中,每个 Web 服务组件在三个层次被描述:接口网声明组件提供给其它组件的服务,互连接网指定组件要完成其任务需要使用到的服务;互操作层描述组件的内部操作行为。WS-net 实际是提出了一种体系结构模型,用于形式化体系结构的拓扑结构以及每个服务组件的行为和系统行为,但是 WS-net 并没有提供将 WSDL 转换成 WS-net 模型的自动转换工具。文献[50]也使用 Petri 网对服务交互进行分析和建模。

### (4) 基于规划的方法

文献[51]使用 Aiplanning 的方法进行服务的动态组合。SWORD<sup>[39]</sup>项目是基于 AI 规划的服务组合方法的典型代表,虽然它基于一个抽象的服务表示模型,但是它的思想对于使用 WSDL 的 Web 服务也是适用的,但是 SWORD 只能处理简单的服务组合问题。对于具有复杂业务语义的服务,要建立模型是一项艰巨的任务。

### (5) 其它

文献[52]使用 AOP 的思想将服务组合时的功能方面和非功能方面分开,并提出一种基于 AOP 的服务组合方法。文献[53]提出了自适应服务动态组合的方法和框架。文献[54]提出了基于属性图的 Web 服务组合模型和文献。

## 2.2.3 面向服务的全生命周期的软件开发方法

SOA 的提出势必引起软件开发过程的变革,但由于 SOA 尚处于起步阶段,目前尚不存在一套被广泛接受的开发方法学。所以面向服务的开发方法学也是亟待解决的问题。文献[55]系统阐述了面向服务的编程思想、相关特征、以及如何使用面向服务的方法来构建满足用户需求的软件系统。文献[56]提出了一个高适应性的面向服务的软件开发方法。首先给出了一个面向服务的支持业务端编程的适应性软件结构框架 CAFISE (convergent approach for information system evolution) Framework,然后给出了基于该框架的面向服务的适应性软件开发方法。CAFISE Framework 对异构资源的服务化、业务化及开放动态的面向服务软件体系结构提供了较好的支持。基于该结构框架的开发方法,从对影响软件的环境要素分析入手,通过质量属性驱动的体系结构分析,重点针对应用要适应的变化进行结构设计,最终通过以体系结构为基础的业务端编程,实现面向服务应用的即时开发与演化。文献[57]更是提出了一个过程嵌入的面向服务的体系结构,在面向服务的体系结构的研究中嵌入整个软件生命周期管理,用来适应软件需求等的变化。这些开发方法的出现为 SOA 在实践中的应用奠定了基础。

### 2.2.4 SOA 和 BPM 的结合研究

SOA 的一个重要原则是业务驱动服务、服务驱动技术。SOA 是以业务为核心的,而且 SOA 关注业务流程,为业务流程开发和组合应用,以服务组件的方式实现流程中的任务和活动,为业务流程模型带来了一种全新的结构,这种结构能够重新对流程进行建模以响应变更,从而能够极大地改善灵活性和敏捷程度。由此成为 BPM 的一个重要基础。基于“服务”的设计可以灵活更改流程,而且流程服务能够快速组合,从而编排成更大的端到端流程。所以, BPM 和 SOA 有密不可分的关系,两者相互结合相互促进,结合 BPM 和 SOA 研究软件适应性成为当前的一个研究热点<sup>[58]</sup>。

### 2.2.5 面向服务的体系结构的交叉研究

#### a) 结合语义 Web 技术的语义 SOA

语义 SOA 最初是从语义 Web Service 的研究开始的。将本体引入 Web Service 的描述,给 Web Service 赋予语义,能避免 Web Service 描述的二义性,方便了 Web Service 的发现、组合<sup>[59]</sup>;研究 Web Service 组合时将语义 Web 引入,使得组合过程中不同于传统的组合方式,更加准确和自动化<sup>[60]</sup>。

语义 SOA 中比较著名的一个例子是 WSMO。WSMO<sup>[61]</sup>是由欧盟 IST 支持的 SWWS 项目组制订的 Web 服务建模规范。WSMO 由 Ontologies, Goals, Web Service, Mediators 等 4 个部分组成。Ontologies 即本体,为其它各个部分提供统一的、机器可理解的词汇集,WSMO 大量使用了本体,对 Web 服务描述提供了强大的语义支持;Goals 用来描述用户需求;Web Service 用来描述 Web 服务;Mediators 用来解决异构和互操作问题。WSMO 中一个突出的特点是在 Ontologies, Goals, Web Service 等 3 个部分大量使用 Defined By Logical Expression 的形式,直接使用一阶谓词逻辑表达式对 Web 服务进行刻画,这种方法准确性高,便于机器理解,使 WSMO 的描述能力得以大大得高。另一个与 WSMO 类似的技术是 OWL-S。OWL-S 是由美国国防部信息开发办公室支持的 DAML 项目组在 DAML 基础上,针对 Web 服务的特点制订的关于如何描述 Web 描述的规范。OWL-S 从 Service Profile, Service Model, Service Grounding 三个层次描述一个 Web 服务。Service Profile 又叫服务能力广告,回答“服务需

要用户提供什么和能为用户提供什么”的问题; Service Model 回答“服务如何工作”的问题; Service Grounding 回答“如何使用该服务”的问题。OWL-S 使用 IOPEs (Inputs, Outputs, Pre-conditions, Effects) 来描述服务能力。

#### b) 结合 MDA 技术的模型驱动 SOA (MDSOA)

MDA (model driven architecture) 是 OMG (object management group) 提出出来的一种软件体系结构标准, 包括一种软件系统设计和实现的方法<sup>[62]</sup>。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。

为实现平台无关的 SOA, 应用 MDA 已有的成果来研究 SOA 是 SOA 研究中的一大趋势, 这方面以 IBM 的研究者为典型代表, 文献<sup>[63]</sup>使用模型驱动的方法贯穿整个软件开发生命周期, 基于把 MDA 的应用到业务集成和管理, 启发了 SOA 的模型驱动研究。文献<sup>[64]</sup>使用 UML Profile 为 Web Service 建模, 并使用 MDA 的方法, 实现了从业务分析模型到平台无关模型, 再到平台相关模型的转化, 是在 SOA 的研究中全面地使用 MDA 的一种方法。IBM 的商业软件 Software Architects 也是支持模型驱动的面向服务的开发方法的一个工具平台。

#### c) 结合 Agent 技术的 SOA (Agent SOA)

Agent 技术起源于分布式人工智能的研究, Agent 技术强调软件的分布性、自治性、智能性等特点, Agent 分为移动 Agent 和智能 Agent。这两种 Agent 各有特色, 由于 Agent 的智能性和移动性, 把 Agent 技术用于 SOA 的研究, 可以解决很多 SOA 研究中很多方面的问题, 并且能增强系统的动态性<sup>[65]</sup>。文献<sup>[66]</sup>针对复杂业务需求, 提出了多 Agent 协商的服务流程定制模型、算法和系统。并给出了一个原型系统, 该模型和系统能够针对复杂的业务需求进行建模和求解, 从而寻找到合适的服务。这是应用 Agent 的智能性解决复杂业务需求建模和求解的实例。

#### 2.2.6 面向服务的体系结构的模拟验证

对于 SOA 中的服务协作, 组合等进行形式化的模拟, 可以验证其正确性, 优化协作组合方法, 并为其提供理论支持。所以对于 SOA 中相关技术的模拟也是研究的一个方向。以 W. T. Tsai 为代表的一批研究人员, 在提出面向服务的过程定义和建模语言 PSML-S<sup>[67]</sup>基础上, 提出了 SOA 协作、建模、分析和模拟框架 PSML-C<sup>[57]</sup>, 并提出了为分布式系统快速开发的 SOA 建模和模拟框架<sup>[68]</sup>, 为了对服务协作进行模拟, 提出了 DSOCS 模拟框架<sup>[69]</sup>。W. T. Tsai 的这一系列文献为 SOA 的模拟、优化奠定了基础。

#### 2.2.7 面向服务的体系结构的评估和测试

评估和测试 SOA 是验证 SOA 的有效方法, 所以很多研究机构对 SOA 的评估和测试都有研究。目前评估方法包括给基于普适计算的评估<sup>[70]</sup>、基于场景的评估方法<sup>[71]</sup>, 测试方法包括渐进的组测试技术<sup>[72]</sup>、改变结构的测试技术<sup>[73]</sup>等。另外, 研究基于 SOA 的测试框架<sup>[74]</sup>和自动测试工具<sup>[75]</sup>也成为研究热点。

#### 2.2.8 面向服务的体系结构的应用

SOA 的应用是 SOA 研究中最多的, 尤其以国内的文献为主, 例如: 面向服务的工作流模型, 使用面向服务的方法将工作流控制和工作流模型解耦<sup>[76]</sup>; 将 SOA 应用到金融业务领域<sup>[77]</sup>; 构造面向服务的 Smart-Home<sup>[78]</sup>; 使用 SOA 集成软

件资源<sup>[79]</sup>。其他应用领域还包括电子商务、电子政务等;

#### 2.2.9 面向服务的体系结构的演化模型

##### a) 客户为中心的 SOA (CCSOA)

近年来, 以 W. T. Tsai 为代表的一批研究人员提出了不同于传统 SOA 服务模式的客户为中心的 SOA, 一般的 SOA 是以服务提供者为中心的服务模式, 用户在使用服务的时候, 需要在 Web 上找是不是有合适的服务可以供自己使用, 有合适的再绑定使用, 而以客户为中心的 SOA 强调用户的主动性, 首先由用户提出需要的服务需求, 用户需要按照指定的格式, 用特定的语言写出需求, 提交到服务注册服务器, 服务提供者一般不与用户直接沟通, 而是直接去注册服务器取用户需求, 根据用户需求开发相应的服务。这样使得服务可以最大限度地满足用户需求, 而不会开发出很多用户无法使用的服务。文献<sup>[80]</sup>扩展现有的 SOA 为客户为中心的 SOA, 系统的阐述了关于 CCSOA 的基本观点、概念、体系结构、技术。并给出了一个框架和开发过程, 最后给出了一个实例。文献<sup>[81]</sup>也基于 SOA 提出了一种新的面向服务的架构, 全局软件企业架构, 也就是以客户为中心的面向服务的体系结构。文献<sup>[82]</sup>提出了一个客户为中心的的服务的设计方法, 并将此方法在一个电子政务项目中作了实践。

##### b) REST SOA

REST 这个术语由 Roy Fielding 创造, 是和当前 SOAP 技术标准可以对比的 Web Service 的一种协议访问风格。2000 年, Fielding 在他的博士学位论文中第一次提出了 REST 概念<sup>[83]</sup>。Fielding 认为, 使用符合代表性状态传输 (REST) 设计约束的 Web 上部署的组件, 可以充分利用 Web 的有用特性, 从而使万维网达到最佳的工作效果。REST 强调资源 (名词) 有统一的接口, 且以此来对它们寻址, 使用四个 HTTP 标准的 GET、POST、PUT 以及 DELETE 动词来进行请求和响应, 使用资源寻址来处理其可变性。文献<sup>[84]</sup>对 REST 风格的 SOA 和传统 SOA 的方法进行了对比, 详细地分析 REST 的思想, 并指出了两者的不同和应用 REST 可以给 SOA 带来的好处。

#### 2.2.10 动态面向服务的体系结构

为了能让面向服务的体系结构能够适应灵活多变的需求, 一些学者在面向服务的体系结构的适应性方面作了研究: 文献<sup>[15]</sup>提出了一种面向服务的动态协同框架。该框架引入内置的运行时体系结构对象来解耦系统中的各个服务构件, 并通过该对象以体系结构的视角来重新解释服务部件之间的引用和交互。把体系结构这个抽象概念转换为可操作的对象, 从而利用面向对象语言的继承、多态等整套机制, 导出面向体系结构的动态演化技术。文献<sup>[85]</sup>提出了基于体系结构空间、支持动态演化的软件模型 SASM。SASM 是使用发射技术, 通过具有因果相联的基层和元层来构造。基层由可运行的服务形成; 元层是一个具有备树形层次、层间存在求精关系的体系结构空间, 该空间反映了需求的层次性, 提供给用户不同抽象级别的视图和管理手段。通过对体系结构空间的观察, 可获知系统的结构和行为信息。通过对体系结构空间的在线调整可实现对基层的修改而实现系统的非预设动态演化, 而空间中不同层次之间的求精关系保证了应用的完整性和演化的一致性。这一类研究的共同点都是借鉴了动态体系结构研究的一些技术, 方法和成果, 应用在 SOA 的动态演化中。

### 3 SOA 应用实践概述

SOA 引起广泛的注意,与工业厂商的大力推动有直接的关系,所以 SOA 一直是以工业实践为先导,结合学术研究在向前发展。目前,一个普遍的看法是:由于 SCA,SDO 两大 SOA 规范的制定和推动,SOA 的应用已经由概念宣传期走向实用期。这两个规范的制定也是 SOA 成熟的标志。SOA 在工业界的应用和推广体现在以下几个方面。

1) 工业标准(规范)的制定:各个 Web Service 相关的标准化组织在各个层面指定了相应的标准,这些标准为 SOA 的实施奠定了基础。比如 W3C 制定的 XML Schema, XSLT, SOAP, WSDL, OASIS 制定的 UDDI, WS-BPEL, WS-Security, WS-I 制定的一系列互操作规范,还有 SCA, SDO 规范的制定。

2) 实际产品的开发:IBM, Microsoft, BEA, ORACLE 等厂商各自开发了自己的 SOA 产品,都在不遗余力地宣传 SOA 的理念,宣传自己的产品。比较有代表性的有:IBM 的 WebSphere 系列、Microsoft 的 WCF(Indigo)、BEA 的 AquaLogic 系列、Oracle 的 Oracle SOA Suite、SAP 的 NetWeaver、Cape Clear & Sonic & IONA 的 ESB 产品、Mercury 的 SOA 测试产品。这些实际产品的开发,使人们不再怀疑 SOA 的实用性。这些工业厂商也成为推动 SOA 发展的主要力量。

3) 相关书籍的问世:比如前面提到的文献[7-9],类似的书籍还有很多,这些书籍的作者很多都是工业厂商的员工,他们把自己实施 SOA 的经验写成书,帮助大家实施 SOA,一定程度上说明了 SOA 实施的时机已经成熟。这些书籍一部分已经被引进到国内,供国内相关人员参考。

### 4 SOA 研究存在的不足

尽管近年来,面向服务的体系结构研究领域取得了上述的若干成果,但在应用方面,软件体系结构仍然不是很成熟。应该看到,SOA 还处于一个起步阶段。虽然在学术上作了很多的研究工作,但是这些成功的研究真正应用到实际中的不多,而各大厂商所极力推动的 SOA 对实际应用起到了什么积极作用,暂时还看不出。由此可见,应用 SOA 所带来的好处现在并没有很明显。在 SOA 的研究上,还有太多的问题需要解决。总结对 SOA 的研究,我们认为存在如下不足。

(1) 缺乏统一的 SOA 的概念,导致 SOA 的研究范畴模糊。各个厂商都有自己对 SOA 的概念定义和认识,概念的不统一导致设计人员交流上的困难,不利于支持工具的研制,也使得最终用户在选择解决方案时比较困惑,甚至有的用户认为 SOA 只是厂商炒作概念而已,这种概念上的模糊不利于 SOA 的推广和应用。

(2) 流程建模语言和方法繁多,BPEL, OWL-S 等等大量的流程建模语言使得 SOA 的研究陷入了重复定义流程建模语言的境地,WS-BPEL 的出现使得这一问题得到了一定程度上的解决。

(3) SOA 研究缺乏统一的理论模型支持。尽管提出了若干参考模型,但没有一个被广泛接受,各种各样的参考模型仍然在不断地提出。

(4) 虽然对于服务的互联和通信有 Web Service 这样的标准,但是对于服务的创建和开发,服务的组件模型,缺乏一个标准,随着 SCA 和 SDO 的标准化努力,这一问题有望得到缓解和解决。

(5) 缺乏一套成熟的面向服务的软件开发方法学,相对于面向对象的方法学,面向服务的方法学刚刚起步,现在,没有任何一个被广泛接受的面向服务的分析方法,设计方法等等。这一点可能不是一两年就能解决的问题,需要长期努力。

总的来说,SOA 的研究刚刚起步,所需要解决的问题应该不止上面提到的这几点,还有很多问题需要研究者去解决,同时也提供给研究者很多机遇。

**结束语** Gartner 的研究报告中指出:随着商业和技术的推动,SOA 将成为将来的发展趋势已经没有人怀疑了;2007 年,超过一半的公司将 SOA 作为 IT 战略来考虑;2008 年还没有将 SOA 引入到 IT 实现中的公司将成为一只恐龙,也就是说这个公司很落后了。

SOA 研究和应用已经取得了比较大的成果,但是,作为软件工程中一个正在兴起的研究领域,其面临的问题和挑战更多,现在,SOA 领域研究非常活跃。我们认为,在未来几年内,SOA 的研究将会围绕以下几个方面进行。

1) SOA 的理论模型,SOA 由于刚刚起步,其理论基础比较薄弱,所以完善 SOA 的理论体系和理论模型将会是一个研究重点。

2) 面向服务的软件开发方法学,面向服务的开发方法虽然有很多,但都没有形成体系,接下来,很多学者还会在开发方法学方面做深入研究,使得 SOA 形成一套体系化的方法学,推动其发展。

3) 语义 SOA: SOA 和语义 Web 的结合,是一个热点问题,在这个研究领域的论文层出不穷,但是遗留问题仍然很多,需要进一步研究。

4) 模型驱动的 SOA: MDA 的方法仍然会是构建 SOA 应用得比较自然的选择。

### 参考文献

- [1] 李刚. 适应性软件体系结构. 计算机科学, 2002
- [2] Arnold K e. The Jini specification. Addison-Wesley, 1999
- [3] Gartner. Service-Oriented Architecture Scenario[Z]. Technical report. 2003
- [4] IBM Patterns: Service-Oriented Architecture and Web Services. IBM RedBook, 2004
- [5] Microsoft. Service-Oriented Architecture Introduction. MSDN
- [6] 杨美清. 软件工程技术发展思索. 软件学报, 2005, 16(1)
- [7] Thomas Erl. Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR, 2005
- [8] Eric Newcomer G L. Understanding SOA with Web Services. Addison-Wesley, 2004
- [9] Dirk Krafzig D S. Enterprise SOA: Service-Oriented Architecture Best Practices. Prentice Hall PTR, 2004
- [10] Humberto Cervantes R S H. Technical Concepts of Service Orientation. Idea Group Inc., 2005
- [11] OASIS. Reference Model for Service Oriented Architecture 1.0. OASIS, 2006
- [12] CBDI. Modernizing Application Integration with Service-Oriented Architecture. CBDI, 2005
- [13] Bea. Domain Model For SOA: Realizing the Business Benefit of Service-Oriented Architecture. White paper, BEA, 2005
- [14] Microsoft. Service-Oriented Architecture. MSDN, 2004
- [15] 马晓星. 一种面向服务的动态协同架构及其支撑平台. 计算机学报, 2005, 28(4)
- [16] Alfaro L H T. Interface automata. ACM Press//Proc. of the 9th Annual ACM Symp. on the Foundations of Software Engineering (FSE 2001). 2001
- [17] Alfaro L HT, Stoelinga M. Timed interfaces//Sangiiovanni-Vin-

- centelli A, Sifakis J, eds. Proc. of the 2nd Intl Conf. on Embedded Software (EMSOFT 2002). Berlin: Springer-Verlag, 2002
- [18] Beyer D H T. An interface formalism for Web services // Proc. of the 1st Intl Workshop on Foundations of Interface Technologies (FIT 2005). 2005
- [19] Beyer D H T. Web service interfaces. ACM Press // Proc. of the 14th Intl World Wide Web Conf. (WWW 2005). 2005
- [20] Little M. Transactions and Web services. Communication of ACM, 2003, 46(10)
- [21] Fu X S. Analysis of interaction BPEL web service. www2004, ACM Press, 2004, 621-630
- [22] Bea M. Business process execution language for Web services. Version 1. 1. BEA, IBM, Microsoft, 2003
- [23] Butler M R. Executable semantic for compensation CSP // Proc. of the 2nd int'l workshop on webservice and Formal Methods, LNCS3670. 2005
- [24] Miao Huaikou X C. Formaling and analyzing Service-Oriented Architecture software architecture style // IEEE(EDOC'06). 2006
- [25] Aliaksei Yanchuk M M. A Lightweight Formal Framework for Service-Oriented Applications Design // Berlin Heidelberg 2005 (ICSOC 2005), LNCS 3826. Springer-Verlag, 2005
- [26] Tao Zhang S C, Jia X. A Modeling Framework for Service-Oriented Architecture // IEEE(QSIC'06). 2006, 0-7695-2718-3/06
- [27] Guadalupe Ortiz J H. Toward UML Profiles for Web Services and their Extra-Functional Properties // IEEE(ICWS'06). 2006, 0-7695-2669-1/06
- [28] Maria J, Zaha M D, A t H. Let's Dance: A Language for Service Behavior Modeling // OTM 2006. Berlin Heidelberg, LNCS 4275. Springer-Verlag, 2006
- [29] Ingolf H, Kruger R M. Systematic Development and Exploration of Service-Oriented Software Architecture // IEEE(WICSA04). 2004
- [30] Lee R, et al. A Framework for Dynamically Converting Components to Web Services // IEEE(SERA'05). 2005
- [31] Zhu H. Building Reusable Components with Service-Oriented Architectures. IEEE, 2005
- [32] Yih-Cheng, Lee S C C. A Service-Oriented Architecture for Design and Development of Middleware // IEEE (APSEC'05). 2005, 0-7695-2465-6/05
- [33] 陈德伟. 服务部署与发布绑定的基于P2P网络的Web服务发现机制. 计算机学报, 2005, 28(4)
- [34] 胡海洋. 反射中间件的研究与进展. 计算机学报, 2005, 28(9)
- [35] 李景山. 普适计算中基于接口语义描述的动态服务组合方法. 计算机研究与发展, 2004
- [36] Benatallah B S Q N A. Declarative composition and peer-to-peer provisioning of dynamic Web services // Proceedings of IEEE International Conference Data Engineering. 2002
- [37] Tomic V PB. On dynamic service composition and its applicability to e-business software systems // Proceedings of Workshop on Object-Oriented Business Solutions. 2001
- [38] Narayanan S M S. Simulation, verification and automated composition of Web services // Proceedings of international WWW Conference. 2002
- [39] SR Ponnekanti A F. SWORD: A Developer Toolkit For Web Service Composition // the 17th International W3C. 2002
- [40] Dynamically S T. composing Web services from on-line source // Proceedings of AAAI workshop on Intelligent Service Intergration. 2002
- [41] Benatallah B DM. The self-serv environment for Web services composition. IEEE Internet Computing, 2003
- [42] Mao Z, M B E K R. Fault tolerant, scalable, wide-area internet service composition. Berkeley U. C Technical Report. 2001
- [43] Zhang R A M B. Automatic composition of sematic Web services // Proceeding of International Conference on Web Service, 2003
- [44] 李曼. 基于领域本体的Web服务动态组合. 计算机学报, 2005, 28(4)
- [45] 廖军. 基于Pi-演算的Web服务组合的描述和验证. 计算机学报, 2005, 28(4)
- [46] Karamanolis C MJWS. Model checking of workflow schemas // Proceeding of the 4th International Enterprise Distributed Object Computing Conference. 2000
- [47] Foster H M J K J. Model based verification of Web Service compositions // Proceedings of the 18th IEEE International Conference on Automated Software Engineering. 2003
- [48] Koshkina M v B F. Modelling and verifying web service orchestration by means of the concurrency workbench. ACM SIGSOFT SEN, 2004
- [49] Zhang Jia C C K S. WS-Net: A petri-net based specification model for web services // Proceedings of the 2nd IEEE International Conference on Web Services. 2004
- [50] Lee W J. Modeling and analysis service Interactions In Service-Oriented Architecture // ISCS2003, Berlin Heidelberg, LNCS 2869. Springer-Verlag, 2003
- [51] Hilmar Schuschel MW. Automated Planning in a Service-Oriented Architecture // IEEE(WET ICE'04). 2004
- [52] Vargas-Solar G. Toward aspect oriented services coordination for building modern information systems // IEEE(ENC'04). 2004
- [53] Modafferi S A B. A Framework for Provisioning of Complex e-Services // IEEE(SCC'04). 2004
- [54] 任志宏. Web服务复合的若干关键问题研究. PhD Thesis. 中国科学院软件研究所, 2003
- [55] Francisco Curbera MN, Stockton ML. Toward a Programming Model for Service-Oriented Computing // ICSOC 2005, LNCS 3826. Berlin Heidelberg Springer-Verlag, 2005
- [56] 李刚. 基于CAFISE Framework的高适应性面向服务软件开发. 软件学报, 2006, 17(6)
- [57] W. T. Tsai Q H R P. SOA Collaboration Modeling, Analysis, Simulation in PSMLC // Proc. of the Second IEEE International Symposium on Service-Oriented Applications, Integration and Collaboration(SOAIC'06). 2006
- [58] Thomas W S G. BPM and SOA: Synergies and Challenges // WISE 2005, Berlin Heidelberg, LNCS 3806. Springer-Verlag, 2005
- [59] BingLi. Ontology and Ontology and Service Oriented Programming. PhD Thesis. Arizona state university, 2004
- [60] Brahim, Medjahed. Semantic Web enabled composition of Web services. PhD Thesis. Faculty of the Virginia Polytechnic Institute and State University, 2004
- [61] A. Polleres J B D. WSML-A Language Framework for Semantic Web Service // W3C Workshop on Rule Language for Interoperability. 2005
- [62] OMG Staff Strategy. Group. Model driven architecture White Paper. OMG White Paper, 2000
- [63] Zhu J, et al. Model-driven business process integration and management; A case study with the Bank SinoPac regional service platform. IBM J. RES. & DEV, 2004
- [64] Simon K, Johnson A W. A Model-Driven Development Approach to Creating Service-Oriented Solutions // ICSOC 2006, LNCS 4294. Berlin Heidelberg, Springer-Verlag, 2006; 624-636
- [65] Kangmin Zheng D K C H. A Service-Oriented Architecture in a Multi-Agency Environment; A Case Study in Enterprise Dynamics // IEEE(EDOCW'06). 2006, 0-7695-2743-4/06
- [66] 曹春. SCoAC: 一个面向服务计算的访问控制模型. 计算机学报, 2006, 29(7)
- [67] Tsai W T, et al. PSML-S: A Process Specification And Modeling Language For Service-Oriented Computing // the 9th IASTED International Conference on Software Engineering and Application. 2005
- [68] Tsai W T, et al. A Service-Oriented Modeling and Simulation Framework for Rapid Development of Distributed Applications. Simulation Modeling Practice and Theory, Elsevier, 2006

[69] Tsai W T, et al. Dynamic Collaboration Simulation in Service-Oriented Computing Paradigm1 // IEEE(ANSS'07). 2007,0-7695-2814-7/07

[70] Swaroop Kalasapur B S. Evaluating Service Oriented Architectures (SOA) in Pervasive Computing // IEEE(PERCOM'06). 2006,0-7695-2518-0/06

[71] Stephan Aier M S. Evaluating Integration Architectures-A Scenario-Based Evaluation of Integration Technologies

[72] Tsai W-T, et al. Testing Web Services Using Progressive Group Testing // AWCC 2004, LNCS 3309. Berlin Heidelberg, Springer-Verlag, 2004

[73] Chan W K, et al. Towards a Metamorphic Testing Methodology for Service-Oriented Software Applications // IEEE(QSIC'05). 2005

[74] Zhu H. A Framework for Service-Oriented Testing of Web Services // IEEE((COMPSAC'06). 2006,0-7695-2655-1/06

[75] Schahram Dustdar S H. Testing of Service-Oriented Architectures- A Practical Approach // NODeBerlin Heidelberg, LNCS 3263. Springer-Verlag, 2004

[76] Xu Wei. SOWAC: A Service-Oriented Workflow Access Control Model // IEEE(COMPSAC'04). 2004

[77] Fethi A, Rabhi F S. A Service-Oriented Architecture for financial business processes. Springer-Verlag, 2006

[78] Chao-Lin, Wu L C. Service-Oriented Smart-Home Architecture Based on OSGi and Mobile-Agent Technology. IEEE Transactions on Systems, man, and Cyberneticspart C: Applications and Reveys, 2007, 37(2)

[79] Cheng D, et al. A Service-Oriented Integrating Practice for Data Modeling, Analysis and Visualization // IEEE(ICEBE'06). 2006

[80] Tsai W T, et al. Consumer-Centric Service-Oriented Architecture: A New Approach // IEEE(SEUS-WCCIA'06). 2006

[81] Tsai W T, et al. Global Software Enterprise: A New Software Constructing Architecture // IEEE (CEC/EEE'06). 2006, 0-7695-2511-3/06

[82] Hong Cai K W. Service-Oriented Design Method and Practice For Constructing Citizen-Centric Public Services // ICEBE'06. 2006

[83] Fielding R T. Architectural Styles and the Design of Network-based Software Architectures. PhD Thesis. University of California, 2000

[84] Vinoski S. REST Eye for the SOA Guy. IEEE INternet Computing, 2007, 1089-7801/07

[85] 李长云. 一个面向服务的支持动态演化的软件模型. 计算机学报, 2006, 29(7)

(上接第 6 页)

分派最终目的就是要在组织结构中找到执行流程任务的人,即使任务分派是从流程实例入手的,那么最后也必须要从流程模型导航到组织模型才有意义。

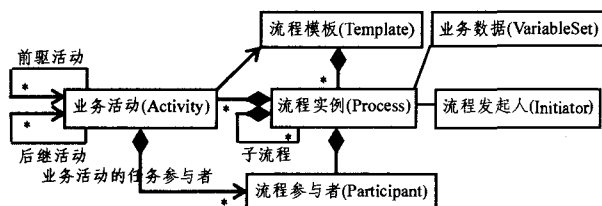


图 3 流程对象模型(ProcBOM)关系图

下面以一个实际的“费用报销流程”中“部门主管审批”步骤的任务分派需求为例,来展示基于规则技术的任务分派策略的基本描述形式:

• 任务分派需求

是申请人的部门主管;如果部门主管自己是申请人,那么启用代理人;如果审批人已经审批过,那么就跳过。

• 业务化分派规则

Rule1 分派的人员是流程发起人的部门的主管,分派模式是单一分派。

Rule2 如果分派的人员与流程发起人产生禁止性冲突那么启用分派的人员的代理人。

Rule3 如果分派的人员与以前的审批人产生重复性冲突那么跳过该结点。

• 技术性分派规则

Rule1 pset1 = assignSingleRandom (assignToPosition ( process. initiator. ownerDepartment. chiefPosition ))

Rule2 if forbiddenConflict ( pset1, process. initiator) pset1 = pset1. delegate

Rule3 if repeatedConflict ( pset1, process. allParticipants) skip

在规则技术中,面向业务人员的业务化规则要被系统自动转换为技术性规则,建立与底层实现的映射关系之后才能

被规则引擎所执行。

从以上例子可以看出,基于业务规则技术的工作流任务分派策略在描述形式上更加接近实际的业务环境,具有更好的易用性。

结束语 采用业务规则技术的工作流任务分派策略描述将是工作流任务分派技术发展的主要方向之一,它是提高任务分派策略表达能力、增强灵活性和易用性有效方法。增强任务分派准确性、动态性和自主性的任务分派控制算法也是任务分派策略模型的重要组成部分,可以被封装为底层的分派计算支撑。任务分派策略应该基于组织模型和流程模型导航来得到流程参与者,并且组织模型应该是与业务环境相匹配的,而不是简单的角色和员工。

参 考 文 献

[1] 陈传波,赵伟伟. 一种自主工作流任务分配策略. 华中科技大学学报(自然科学版), 2005: 20-22

[2] 段永强,曹健,张申生. 工作流系统中的动态任务调度. 中国机械工程, 2002, 13: 233-235, 241

[3] 汪良主,张申生,傅谦. 基于策略的适应性工作流资源动态分配技术. 上海交通大学学报, 2002, 36(4): 506-508

[4] 肖郑进,何钦铭,陈奇. 模糊环境中工作流任务分配的多级模型. 计算机研究与发展, 2007, 44(2): 302-309

[5] 怀进鹏,万良君. 基于能力的工作流任务组织化. 北京航空航天大学学报, 2003, 29(12): 1091-1096

[6] 张晓光,张申生,曹健. 策略约束面向角色和团队关系的工作流任务分配管理. 计算机研究与发展, 2002, 39(12): 1556-1563

[7] 万良君. 开放任务管理策略与工作流属性关系代数. 计算机集成制造系统, 2006, 12(12): 2082-2089

[8] 汪涛,黄力芹,吴耿锋. 工作流管理中的一种资源管理策略 // 第八届全国青年计算机学术会议(NCYCS'2000). 南京, 10/26-28, 2000

[9] 邹正武,谭庆平,徐建军. 基于规则引擎的柔性工作流管理系统. 微电子学与计算, 2006, 23(1): 127-129

[10] 李春芳,骆盈盈,谭庆平. 基于业务规则的工作流管理系统模型. 计算机工程与设计, 2006, 27(20): 3861-3863, 3886