

面向企业间集成的 Web 服务规范化设计方法^{*}

王忠杰 徐晓飞 战德臣

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘要 面向服务的计算被认为是电子商务环境下一种基于 Web 的、用以解决多企业间复杂业务交互的有效手段,但在现实中并没有达到预期的效果。目前对该方面的研究主要集中于技术层面,缺乏有效的方法来支持企业进行服务的识别与设计。本文首先分析了 Web 服务与企业内/企业间业务过程的关系,讨论了适合于 ERP 与电子商务系统集成的服务应满足何种性质,进而提出了服务的规范化模式与相应的规范化方法,重点讨论了如何从 ERP 与电子商务系统的业务模型以及基于构件的软件系统出发,识别规范化服务的途径。该方法对协助企业识别和设计有价值的 Web 服务具有重要意义,并为建立企业内 ERP 系统与企业间电子商务系统之间的集成提供了有效支持。

关键词 Web 服务,服务的识别与设计,规范化模式,企业间集成

Inter-enterprise Integration Oriented Normalized Design Method for Web Services

WANG Zhong-Jie XU Xiao-Fei ZHAN De-Chen

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

Abstract Service-oriented computing (SOC) is thought to be an effective way to tackle complex Web-Based business collaborations between enterprises in E-Business environment, but in practice it didn't reach the expectation. Current researches focus mainly on technological aspects, such as service publication, discovery, binding, and composition, etc. while lack in effective ways to support service identification and design before they are to be published and discovered by other enterprises. This paper analyzes the relationships between Web services and inter-/intra-enterprise business processes, discusses what kind of properties a "good" service should have to fit for the integration between ERP and E-Business systems, and accordingly presents a Service Normal Form (SNF) and the corresponding service normalization method. In this method, the way of how to identify normalized services from business process models, and how to create close mapping between services and legacy component-based information systems, are emphatically discussed. This method provides great significance to help enterprises identifying and designing valuable business services.

Keywords Web services, Service identification and design, Normal form, Inter-enterprise integration

1 引言

利用 Web 服务作为基本的构造单元来开发分布式的、基于 Web 的应用系统,以实现企业间频繁的业务协作与交互需求—即面向服务的计算^[1,2],已经成为目前电子商务环境下一种流行的计算模式。

Web 服务的整个生命周期可被分为五个大的阶段^[3],即:(1)识别与设计;(2)实现与封装;(3)描述与发布;(4)发现、绑定与组合;(5)管理与协调。目前的研究主要集中于后三个阶段,或称为“技术阶段”,并且已经形成了相对完整的体系架构^[1]与技术栈^[4],包含了一组成熟的技术与标准,例如用于服务描述的 WSDL、用于服务发布的 UDDI^[6]、用于消息传递的 SOAP^[7]、用于服务组合的 BPEL4WS^[8],以及其他诸如 WS-Transaction, WS-Security 等相关技术^[1,4]。归纳起来, SOC 已经为企业间基于电子商务的交互提供了技术方面的充分支持。针对阶段(2),因为 Web 服务是一种编程语言无关的软件实体,只要能够按照特定的技术标准进行封装^[9],任何语言都可用来开展 Web 服务,如 J2EE、Java WSDP、VS. Net 等。实际上, SOC 已经被认为是面向对象的编程(OOP)与基于构件的开发(CBD)的下一个发展阶段,可复用软件实体从细粒度的对象到一般粒度的构件,逐步发展到今天的大粒度的 Web 服务^[10,11],如图 1 所示。如果我们换一个视角,

一个服务也可被看作是一组软件构件的组合^[12],如图 2 所示。

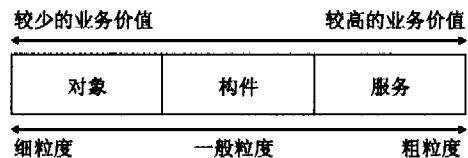


图 1 从对象、构件到服务:粒度逐渐增大

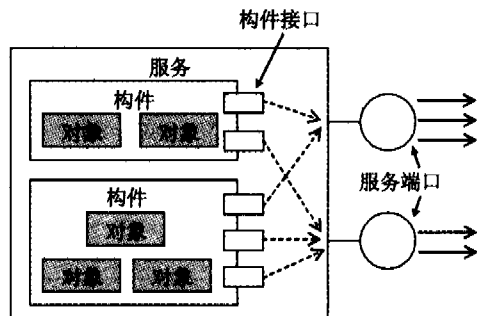


图 2 服务、构件与对象之间的组合关系

针对阶段(1),目前的研究工作中似乎被严重地忽略了。对服务的提供者来说,例如一个企业,很难找到确切的答案,

^{*}基金项目:国家自然科学基金项目资助项目(60573086)、十五国家 863 资助项目(2003AA4Z3210)、高等学校博士学科点专项科研基金资助课题(20030213027)。王忠杰 博士研究生,主要研究方向:可重构 ERP 系统、软件构件、重用技术;徐晓飞 博士,教授,博士生导师,主要研究方向:企业智能计算、管理与决策信息系统、数据库等;战德臣 博士,教授,博士生导师,主要研究方向:软件重构与复用、企业资源计划系统与现代集成制造系统等。

以回答诸如“什么样的服务是好的服务?”或者“如何有效地识别好的服务?”这样的问题,因此也就无法指导服务的提供者构造有价值的、可被其伙伴企业复用的服务。目前只有很少的文献简要地给出了若干设计原则,例如高内聚-低耦合^[3]、不要依赖于标准化^[13]、反映现实世界的业务文档等^[13],但这些原则非常分散,很难对服务提供者给予有效的帮助。

实际上,由于构件与服务在语义和结构方面都存在较大的相似性,构件识别与设计的一些原则与方法对服务来说是很好的参考。传统的构件设计方法^[14]通常从企业业务模型出发,对其进行语义分析,以得到高质量的构件。与构件相比,一个服务也描述了企业的某项业务,但具有更大的粒度。因此,服务设计也应该从业务模型(主要是业务过程模型)出发,通过某种手段,将过程模型映射为一组服务。这种服务识别方式也解决了目前存在于 SOC 中的另一个问题,即面向服务的系统(如企业间电子商务系统)很难紧密地与企业内信息系统(如 ERP)相结合的问题。

总而言之,目前的研究将注意力重点放在技术方面而忽视了业务方面,还缺乏好的规范化模式与系统化的设计原则,这将在某些方面导致 SOC 的进一步发展。

本文要解决的主要问题是服务的识别与规范化设计方法。我们从分析企业内/企业间业务过程出发,归纳提出一个好的服务应具备的性质。基于此,重点提出服务的规范化模式(SNF)以及相应的由业务过程模型出发进行规范化服务识别与设计的方法。本文的结果将为企业识别有价值的服务提供有效手段,在理论上对 SOC 的理论架构是一个有价值的补充。

2 服务与企业信息系统之间的映射

如前所述,企业发布的服务应该与企业内运行的信息系统(如 ERP、CRM 等)紧密关联,从而形成一个集成的、有效

的宏观系统,以支持企业内与企业间的业务。因此,服务识别的任务应该也从对现有信息系统的分析入手。这里我们首先讨论 Web 服务与信息系统的关系。

信息系统可分为两个层次:业务层与技术层。前者刻画了企业的管理模式、业务功能与流程,体现为一组业务模型;后者由一组软件实体(通常是软构件)构成,以实现业务层的业务模型所描述的业务功能。对服务来说,为了实现与信息系统的集成,也应该包含着两个层次。

图 3 给出了服务与信息系统之间的映射关系,共存在 4 种映射:

- M_1 : 业务模型与构件模型之间的映射,属于领域工程的研究范畴;
- M_2 : 业务模型与 Web 服务业务模型之间的映射;
- M_3 : 构件模型与 Web 服务的软件模型之间的映射;
- M_4 : Web 服务的业务模型与技术模型之间的映射。

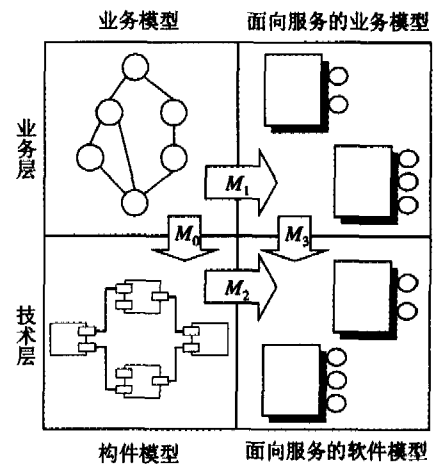


图 3 面向服务的企业信息系统的映射

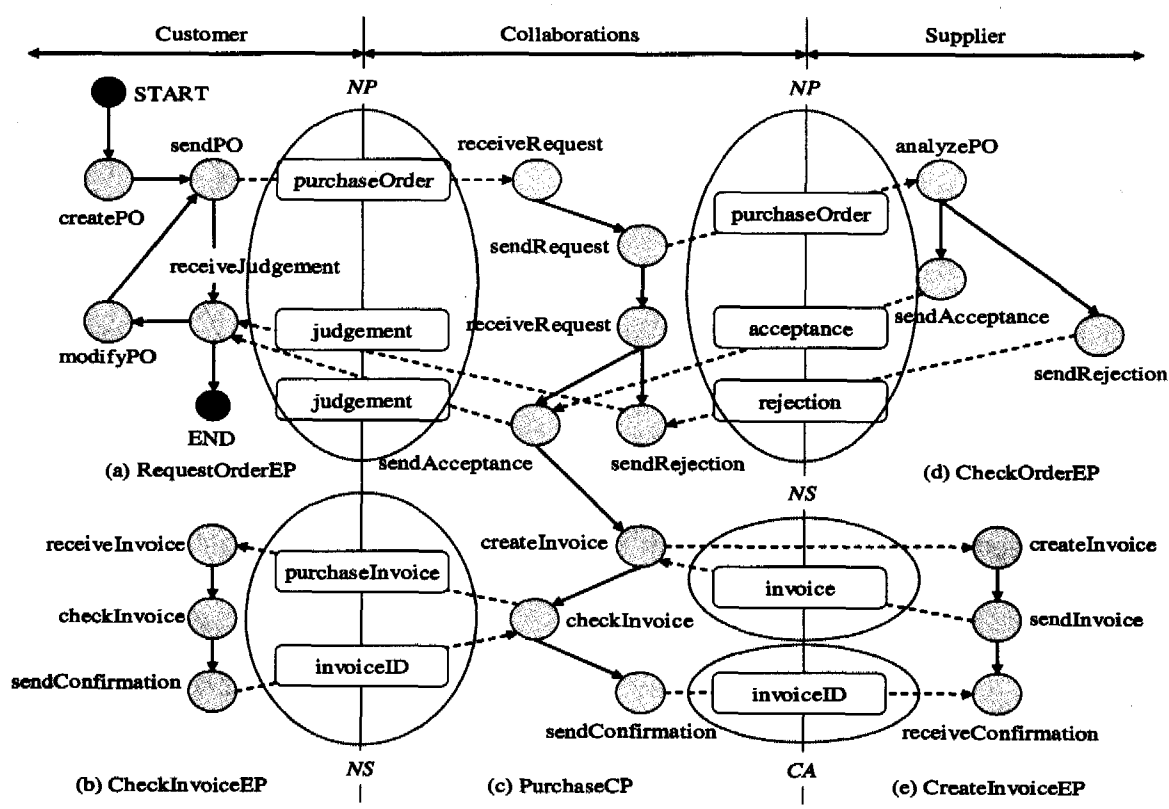


图 4 企业内与企业间业务过程的示例(引自文[15])

本文重点探讨 M_2 与 M_3 两种映射。

2.1 面向服务的企业内与企业间业务模型

在当今电子商务时代,一个完全封闭的企业是不存在的,企业或多或少与其他企业发生协作。目前已经存在多种企业间管理模式,如虚拟企业、B2B 电子商务、供应链等。

企业范围内的基本业务功能可表达为一组业务过程,称为可执行过程(EP)。企业间的交互活动也可表达为一组业务过程,称为契约过程(CP)^[15],一个 CP 由多个 EP 构成。CP 与 EP 具有相同的语义,二者之间的区别在于,构成 EP 的业务活动分由不同的企业执行。图 4 给出了 EP 与 CP 的一个实例。

任何特定的 CP 由一组原子交互模式构成,并具有 6 种不同的类型^[16],分别为 CS、CA、NS、ND、NP、PS,这些模式描述了构成 CP 的 EP 之间进行数据交换的方式。这里我们不再讨论每一种模式的细节,有兴趣者可参阅文[15]。

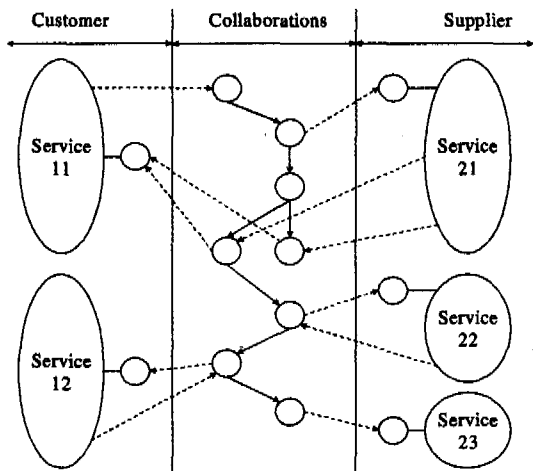


图 5 面向服务的企业间业务过程

在面向服务的计算环境下,企业间的 CP 可表达为服务之间的互操作^[16]。每个企业构造并发布自己的 Web 服务,CP 即可由这些 Web 服务的组合加以实现,并使用诸如 BPEL4WS、XLANG 等服务组合语言进行描述。图 5 给出了一个面向服务的企业间业务过程的示例,该示例由图 4 中 PurchaseCP 过程转化过来,通过将图 4 中各企业内过程构造为服务得到。

为了从 EP/CP 识别服务,必须深入探讨服务与 EP/CP 之间的映射(即图 3 中的 M_1)。一般地说, M_1 包含了如下的子映射:

(1) 服务与业务活动之间的关系,即一个服务应包含哪些活动;

(2) 业务活动与服务端口之间的映射,即服务的每一个端口应该向其他服务提供哪些业务活动所实现的功能;

(3) 活动的输入与服务端口之间的文档映射,即一个服务端口应从其他服务接收哪些业务文档(如订单、发票等)并将其传递给服务内的哪些活动。

2.2 面向服务的软件体系结构

作为构成企业间信息系统的一种基本软件实体,Web 服务也应表达为两个方面。业务层面已经在 2.1 节中进行了阐述。在技术层面,一个服务应由一组软构件构成(如图 2 所示)。本节我们讨论在技术层面服务与软构件的映射,即图 3 中的 M_2 ,并分解为如下的子映射:

(1) Web 服务与软构件之间的关系,即一个服务应包含哪些软构件;

(2) 构件接口与服务端口之间的功能映射,即每个服务端

口应包含哪些软构件的哪些接口所提供的哪些功能;

(3) 构件接口与服务端口之间的消息映射,即一个服务端口应从其他服务接收哪些消息并将其转发给那些软构件的哪些接口。

图 6 给出了 M_1 和 M_2 两种映射,从中可以看出 CP 与构件如何被映射为服务。

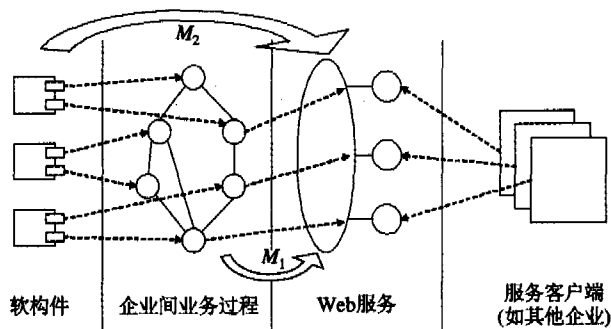


图 6 软构件与 Web 服务之间的关系

3 SNF: 服务规范化模式

从上文分析可以看出,Web 服务的设计应分为两个方面:业务规约设计,即将业务过程模型映射为服务的语义模型;技术设计,即将基于软构件的软件体系结构映射为服务的实现模式。一个好的服务应在业务与技术方面均提供良好的性能。本节我们将讨论好的服务应具备何种性质,并提出服务的规范化模式。

3.1 SNF: 业务层面

如上讨论,服务与业务过程模型之间存在映射,一个服务可看作企业业务过程向外界暴露的一个“通道”。通过服务,企业从其伙伴企业获取必要的业务数据,并返回相应的结果。因此,服务规约设计可归纳为划分企业间/企业内业务过程的过程,即将业务过程分解为一组子过程,并将每一个子过程封装为服务,同时定义相应的端口以及通过端口交换的文档数据。

定义 1 一个服务被定义为 $S_{spec} = (act_set, port_set, doc_set, \omega, \nu, \rho)$,其中,

(1) act_set : 服务所包含的业务活动集合,表示为 $\{a_1, a_2, \dots, a_m\}$;

(2) $port_set$: 服务的端口集合,表示为 $\{p_1, p_2, \dots, p_n\}$;

(3) doc_set : 服务通过其端口从客户端接收的业务文档的集合,表示为 $\{d_1, d_2, \dots, d_m\}$;

(4) ω : 活动集合与服务端口之间的映射,即每一个端口应包含哪些活动,表示为 $w: \{a_1, a_2, \dots, a_i\} \rightarrow p_j$;

(5) ν : 文档集合与端口之间的映射,即服务的每一个端口都能够接收哪些文档,表示为 $\nu: \{d_1, d_2, \dots, d_j\} \rightarrow p_k$;

(6) ρ : 文档集合与业务活动之间的映射,即每一个活动应从服务的特定端口接收哪些文档,表示为 $\rho: \{d_1, d_2, \dots, d_j\} \rightarrow a_i$ 。

除了基本定义,一个好的服务还应遵循以下原则:

(1) 事务完整性: 服务通过端口提供的所有功能应包含在同一个事务范围内,即一个服务应是一个原子事务单元;

(2) 文档完整性: 服务从任何接口所接收的任何文档,应反映现实中真实的业务文档^[13],如订单、合同、发票等;这些文档应是完整的,即不允许接收一个文档片断;另外,如果可能,每个端口应仅接收一个文档;

(3) 事务补偿性: 一个服务相应的补偿服务, 如果可能, 也应该在同一个服务中加以实现。也就是说, 服务应对其客户端提供补偿机制, 以保证在服务失败时的事务完整性;

(4) 服务依赖性: 服务应提供一种调用规则, 以描述相关服务之间的语义依赖关系与调用次序, 以保证客户端在使用服务时的方便性。

3.2 SNF: 技术方面

定义 2 技术方面的服务规范化模式可定义为 $S_{impl} = (comp_set, port_set, msg_set, \sigma, \xi)$, 其中:

(1) $comp_set$: 服务所包含的软构件的集合, 表示为 $\{c_1, c_2, \dots, c_m\}$ 。对 $\forall c \in comp_set$, 存在接口集合 $inf_set(c)$;

(2) $port_set$: 服务端口集合, 表示为 $\{p_1, p_2, \dots, p_n\}$;

(3) msg_set : 服务通过端口所能够接收的消息集合, 表示为 $\{m_1, m_2, \dots, m_m\}$;

(4) σ : 消息集合与端口之间的映射, 即每一个端口能够从外界接收哪些消息, 表示为 $(\{m_1, m_2, \dots, m_j\} \rightarrow p_k)$ 。 σ 是定义 1 中的映射 ν 的技术表示;

(5) ξ : 消息集合与软构件接口集合之间的映射, 即如何将每一条消息分配给构件接口, 表示为 $\nu: msg_set \rightarrow \cup inf_set(c)$ 。 ξ 是一个复杂映射, 可被分解为三种子映射:

- 映射(MM): 直接将服务的一个输入消息 m 映射为某一软构件的某一接口 $inf(c)$ 的输入消息, 表示为 $m \rightarrow inf(c)$;
- 分解(MD): 将服务的输入消息 m 分解为一组子消息 $\{m_1, m_2, \dots, m_k\}$, 并将每一个子消息映射到某一构件的接口, 表示为 $\{m_1, m_2, \dots, m_k\} = MD(m), mi \rightarrow inf(c), i \in [1, m]$;
- 综合(MS): 将服务的一组输入消息 $\{m_1, m_2, \dots, m_k\}$ 合并起来并形成某一构件接口的输入消息, 表示为 $m = MS(m_1, m_2, \dots, m_k), m \rightarrow inf(c)$ 。

4 规范化服务的识别与设计方法

目前有关 SOC 的研究中, 存在多种方法和语言可用于服务的组合, 形成面向服务的业务过程(即图 6 中的 CP)。但实际上, 应首先肯定 CP 的规约(即企业之间的交互需求)。服

务设计者根据这些规约以识别一组服务并实现之, 然后使用这些可执行的服务来构造可执行的 CP。

在第 2 节中我们已经讨论了业务过程与服务的业务规约之间的映射, 以及基于构件的信息系统与服务的技术规约之间的映射。这里我们将根据这两类映射, 给出一种规范化服务的识别与设计方法。该方面分为三步: (1) 从业务过程模型识别规范化服务并设计服务与业务活动之间的包含关系; (2) 设计业务层面的服务, 构造服务端口, 以及端口与活动之间的映射; (3) 设计技术层面的服务, 设计服务与软构件之间的包含关系, 并构造服务端口与构件接口之间的映射。

4.1 从业务过程模型识别规范化服务

假设 CP 是多个企业间的交互过程, $EP_set(E_i)$ 是企业 E_i 范围内的一组可执行业务过程, 以支持 CP 的执行。那么, 从 CP 与 $EP_set(E_i)$ 出发, 针对 E_i 而识别规范化服务的方法需要遵循以下步骤:

Step 1.1: 根据 CP 包含的原子交互模式, 将 CP 分解为一组基本单元 $\{u_1, u_2, \dots, u_k\}$, 每一个 u_i 包含了 CP 的一组业务活动。不同的原子交互模式具有不同的分解原则, 限于篇幅, 这里不再详述;

Step 1.2: 提取每一个 u_i 中包含的业务活动之间所交换的文档, 形成文档集合 $D_i = \{d_{i1}, d_{i2}, \dots, d_{ip}\}$;

Step 1.3: 对属于同一企业的任意 D_i 与 D_j , 如果 $D_i \subseteq D_j$, 那么合并 u_i 与 u_j , 形成一个新单元 D_{ij} ;

Step 1.4: 将每一个 u_i 映射为一个服务 s_i , u_i 中的服务集合构成 s_i 的活动集合 act_set , 文档集合构成 s_i 的 doc_set 。

根据上述过程对图 4 中的 PurchaseCP 过程进行分解, 可得到表 1 所示的 5 个服务基本单元。

4.2 设计服务端口: 业务层面

在将一个 CP 分解为一组服务之后, 我们从业务层面设计服务的端口。对每一个服务 s_i , 遵循以下步骤:

Step 2.1: 对 $act_set(s_i)$ 中包含的每一个活动 a_{ij} , 构造一个端口 p_{ij} 。如果 a_{ij} 被其他 EP 中的活动所调用, 那么构造一个映射 $\omega: a_{ij} \rightarrow p_{ij}$;

表 1 从图 4 分解所得到的基本单元

单元	活动集合	文档	所属角色
u_1	{“sendPO”, “receiveJudgement”}	purchaseOrder (new) purchaseOrder (with judgement)	Supplier
u_2	{“receiveInvoice”, “checkInvoice”, “sendConfirmation”}	purchaseInvoice (unconfirmed) purchaseInvoice (confirmed)	
u_3	{“analyzePO”, “sendAcceptance”, “sendRejection”}	purchaseOrder (new) purchaseOrder (with judgement)	Customer
u_4	{“createInvoice”, “sendInvoice”}	purchaseInvoice (unconfirmed)	
u_5	{“receiveConfirmation”}	purchaseInvoice (confirmed)	

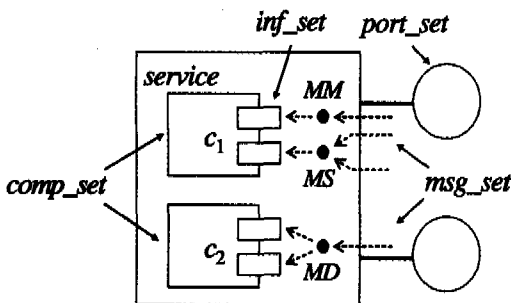


图 7 消息的三种映射机制

Step 2.2: 对 $doc_set(s_i)$ 中的每一个文档 d_{ij} , 在 d_{ij} 与每一个使用 d_{ij} 作为其输入的活动 a_{ij} 之间构造一个映射 $\rho: d_{ij} \rightarrow a_{ij}$;

Step 2.3: 对每一个文档 d_{ij} , 在 d_{ij} 与每一个满足 $\rho: d_{ij} \rightarrow a_{ij}$ 与 $\omega: a_{ij} \rightarrow p_{ij}$ 的端口 p_{ij} 之间构造一个映射 $\nu: d_{ij} \rightarrow p_{ij}$ 。

4.3 设计服务端口: 技术层面

技术层面的设计过程与业务层面的设计相似。对每一个服务 s_i , 遵循以下步骤:

Step 3.1: 在企业目前运行的信息系统中, 寻找所有支持 s_i 中包含的业务活动运行的软构件集合 $comp_set(s_i)$, 构成 s_i 的 $comp_set$;

Step 3.2: 对包含在 $doc_set(s_i)$ 中的每一个文档 d_{ij} , 为其构造一个消息 m_{ij} ;

Step 3.3: 在每一个消息 m_{ij} 与每一个从外界接收 d_{ij} 的服务端口 p_{ij} 之间构造一个映射 σ ;

Step 3.4: 在消息与构件接口之间建立映射 ξ 。如果一个接口 inf 接收某一单独的消息 m_{ij} , 那么使用 MM 的映射方式; 如果某一消息的一部分信息即可满足接口 inf 的需求, 那么使用 MD 映射方式; 如果接口 inf 需要多条消息的复合, 那么使用 MS 的映射方式。图 7 种给出了 MM、MS 和 MD 的几个示例;

Step 3.5: 使用特定的程序语言来实现 s_i , 例如 VS. Net、J2EE 等;

Step 3.6: 使用 UDDI 技术封装与描述 s_i , 并为 $msg_set(s_i)$ 中的每一个消息书写 SOAP 格式。

结论 本文分析了目前在面向服务的计算领域存在的一个关键问题, 即不存在成熟的服务识别与设计方法。为解决该问题, 讨论了业务层面的服务与企业内/企业间业务过程之间的映射, 以及技术层面的服务与基于软构件的软件架构之间的映射。基于这两类映射, 提出了一种服务的规范化模式 SNF, 建立了好的服务应遵循的标准, 以及规范化服务的识别与设计方法。该方法将企业管理模式, 企业内信息系统与服务识别、设计紧密结合在一起, 为企业发现有价值的 Web 服务提供了有效的手段, 并支持企业内信息系统与企业间电子商务系统的有效集成。

参考文献

- 1 Papazoglou M P, Georgakopoulos D. Service-Oriented Computing. Communications of the ACM, 2003, 46(10): 25~28
- 2 Yang J. Web Service Componentization. Communications of the ACM, 2003, 46(10): 35~40

- 3 Papazoglou M P, Yang J. Design Methodology for Web Services and Business Processes. In: Proceedings of the Third International Workshop on Technologies for E-Services, Lecture Notes in Computer Science, London, UK, 2002, 2444: 54~64
- 4 Curbera F, Khalaf R, Mukhi N, et al. The Next Step In Web Services. Communications of the ACM, 2003, 46(10): 35~40
- 5 W3C. Web Services Description Language (WSDL) 1.1. Mar. 15, 2001. <http://www.w3.org/TR/wsdl>
- 6 UDDI specification, version 3.0. <http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm#uddiv3>
- 7 W3C. SOAP specification, version 1.2. Jun. 24, 2003. <http://www.w3.org/TR/soap12/>
- 8 Curbera F, Golland Y, Klein J, et al. Business Process Execution Language for Web Services. Jul 31, 2002. <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel>
- 9 Lee R, Kim H K, Yang H S. An Architecture Model for Dynamically Converting Components into Web Services. In: Proceedings of 11th Asia-Pacific Software Engineering Conference (APSEC'04), Busan, Korea, 2004. 648~654
- 10 Hanson J. Coarse-grained interfaces enable service composition in SOA. <http://builder.com.com/5100-6386-14-5064520.html>
- 11 Anexinet Corp. Enabling Service-Oriented Architecture: Aligning the enterprise for agility and competitive advantage. [Technical White Paper], 2004
- 12 Sanders R T, Bræk R, von Bochmann G, et al. Service Discovery and Component Reuse with Semantic Interfaces. <http://beethoven.site.uottawa.ca/dsrg/PublicDocuments/Publications/Sand05a.pdf>
- 13 Cape Clear Software Inc. Principles of Service-Oriented Architecture (SOA) Design. [Technical Report], CPV-DOC-303, 2004
- 14 Keith L, Mili A. A goal-driven approach to enterprise component identification and specification. Communications of the ACM, 2002, 45(10): 45~52
- 15 Jung J Y, Hur W, Kang S H, et al. Business Process Choreography for B2B Collaboration. IEEE Internet Computing, 2004, 8(1): 37~45
- 16 Lambros P, Schmidt M T, Zentner C. Combine business process management technology and business services to implement complex Web services. [White Paper], IBM Corp, 2001

(上接第 130 页)

上支付; 最后将车票或机票以及行程安排返回给客户。从上述过程可以看到, Schedule Service 是由 Train Query Service (TQS)、Airline Query Service (AQS)、Train Ticket Service (TTS)、Airline Ticket Service (ATS) 和 PayByCMB Service (PBCS) 组合而成。用 Petri 网对 Schedule Service 建模如图 3 所示, 其中用到了平行、选择、顺序和调用组合结构。针对图 3 的 Petri 网, 我们可以构造其可达树或者可达图以研究其活性, 从而验证 Schedule Service 的正确性。

4 相关工作

文[2]提出了一种 Web 服务的 Petri 网模型, 对 Web 服务网做了定义, 并在此基础上提出了 Web 服务代数, 但是与我们的模型相比较, 文[2]中的模型过于复杂, 在实际应用中不便使用, 并且没有定义 Web 服务网的变迁规则以及活性。文[3,4]对现有 Web 服务和 Web 服务组合技术做了总结, 它们指出除 Petri 网外, 还可以使用 π 演算和有限状态机等手段对服务组合进行形式化建模, 并且各种方法各有优劣。一些文献使用 Petri 网为 WSCI、WSFL 和 BPEL4WS 等服务组合语言建模^[7], 具有一定的参考价值。除此以外, 还有文献描述了如何使用 Petri 网对 workflow 进行建模和分析。

结束语 Web 服务组合是 Web 服务应用的重要手段, 需要对 Web 服务组合流程进行建模, 以实现可靠的组合服务。

文中首先根据 Petri 网理论定义了 Web 服务及其服务网 (Service Net), 并给出了五种基本组合结构的 Petri 网模型; 然后定义了服务网的变迁规则, 从而进一步分析其可达性和活性, 以验证组合服务的正确性, 以及组合服务是否能在有限步内结束; 最后举例说明此方法的应用。Petri 网提供了一种有效的手段去模拟、分析和验证 Web 服务组合, 但如何使用 Petri 网对语义 Web 服务以及自动化的 Web 服务组合进行建模则是我们进一步的研究课题。

参考文献

- 1 袁崇义. Petri 网原理. 电子工业出版社, 1993
- 2 Hamadi R, Benatallah B. A Petri Net-based Model for Web Service Composition. ADC, 2003, 2003, 17
- 3 Milanovic N, Malek M. Current Solutions for Web Service Composition. IEEE Internet Computing, NOVEMBER • DECEMBER 2004
- 4 Papazoglou M P, Dubray J. A Survey of Web Service Technologies. [Technical Report], June 2004
- 5 Zhang J, Chang C K, Chung J Y, Kim S W. WS-Net: A Petri-net Based Specification Model for Web Services. ICWS'04, 2004
- 6 蒋昌俊. Petri 网的行为理论及其应用. 高等教育出版社, 2003
- 7 孙健, 陶晓峰. 基于 Petri 网的 Web 服务 BPEL4WS 建模与分析. 计算机工程, 2004, 30(22)
- 8 Cardoso J, Sheth A. Introduction to Semantic Web Services and Web Process Composition. SWSWPC 2004, LNCS 3387, 2005