# 面向服务架构(SOA)的基本概念和方法

## SOA的概念

### 什么是**SOA**

面向服务架构（Service-Oriented Architecture，SOA）是Gartner于20世纪90年代中期提出的一种软件组建与开发的方式，它以“服务”为基本元素来组建企业IT架构。技术上，它是一种抽象的、松耦合的软件架构；业务上，它强调“重用”和“互操作”，将企业的IT资源整合成可操作的、基于标准的服务，从而能够被重新组合和应用。

需要强调的是，SOA是架构风格（架构设计的方法和原则），而不是具体的实现技术（如Web Servic，企业服务总线等）。在SOA架构风格中，“服务”是最核心的抽象手段，业务被划分为一系列粗粒度的业务服务和业务流程。业务服务相对独立、自包含、可重用，由一个或者多个分布的系统所实现，而业务流程由服务组装而来。[3]

一个“服务”定义了一个与业务功能或业务数据相关的接口，以及约束这个接口的契约，如业务规则、安全性要求、法律法规的遵循、服务质量要求、关键性能指标等。接口和契约采用中立、基于标准的方式进行定义，它独立于实现服务的特定技术（例如硬件平台、操作系统和编程语言等）。这使得构建在不同系统中的服务可以以一种统一的和通用的方式进行交互、相互理解。服务之间的交互是动态的，位置是透明的，使得服务的请求者和提供者之间高度解耦。这种松耦合的架构带来两点好处：一是适应变化的灵活性，二是当某个服务的内部结构和实现逐渐发生改变时，不影响其他服务。[3]

SOA架构是业务驱动IT，即IT和业务更加紧密地对齐。以粗粒度的业务服务为基础来对业务建模，会产生更加简洁的业务和系统视图；以服务为基础来实现的IT系统更灵活、更易于重用、更好（也更快）地应对变化；以服务为基础，通过显式地定义、描述、实现和管理业务层次的粗粒度服务（包括业务流程），提供了业务模型和相关IT实现之间更好的"可追溯性"，减小了它们之间的差距，使得业务的变化更容易传递到IT。[3]

概括起来，SOA的优点主要体现在三个方面：一是IT与业务对齐，支持业务的快速变化，能够更好更快地提供业务价值；其次是IT架构的灵活性；第三是IT资产的重用。

### **SOA**的主要特征

从前面的定义和分析可以看出，SOA具有以下主要特征：[3]

（1）**明确定义的接口**。服务接口必须是明确定义的，指明服务请求者所要求的绑定到服务提供者的细节, 减少使用者的不适当使用。使用者依赖服务协议来调用服务，所以服务定义必须长时间稳定，一旦公布，不随意更改。

（2）**自包含和模块化**。服务封装了那些在业务上稳定、重复出现的活动和组件，实现服务的功能实体是完全独立自主的，独立进行部署、版本控制、自我管理和恢复。

（3）**粗粒度**。服务数量不应该太大，依靠消息交互而不是远程过程调用（RPC），通常消息量比较大，但是服务之间的交互频度较低。

（4）**服务之间的松耦合性**。服务请求者到服务提供者的绑定与服务之间应该是松耦合的,服务使用者看到的是服务的接口，其位置、实现技术、当前状态等对使用者是不可见的，服务私有数据对服务使用者是不可见的。

（5）**无状态的服务设计**。服务应该是独立的，避免服务请求者依赖于服务提供者的状态。在实现时它不需要获取从一个请求到另一个请求的信息或状态，即服务不应该依赖于其他服务的上下文和状态。

（6）**单一实例**。避免功能冗余。

（7）**可重用**。服务应该是可以重用的，一个服务创建后能用于多个应用和业务流程。

## **SOA**的组成

### **SOA**的组成要素

SOA提供了一种构建分布式系统来将应用程序功能作为服务提供给终端用户应用程序或其他服务的方法。SOA的组成要素可以分为功能性和服务质量两类，这些要素及其相互关系如图 2‑1所示。[4]

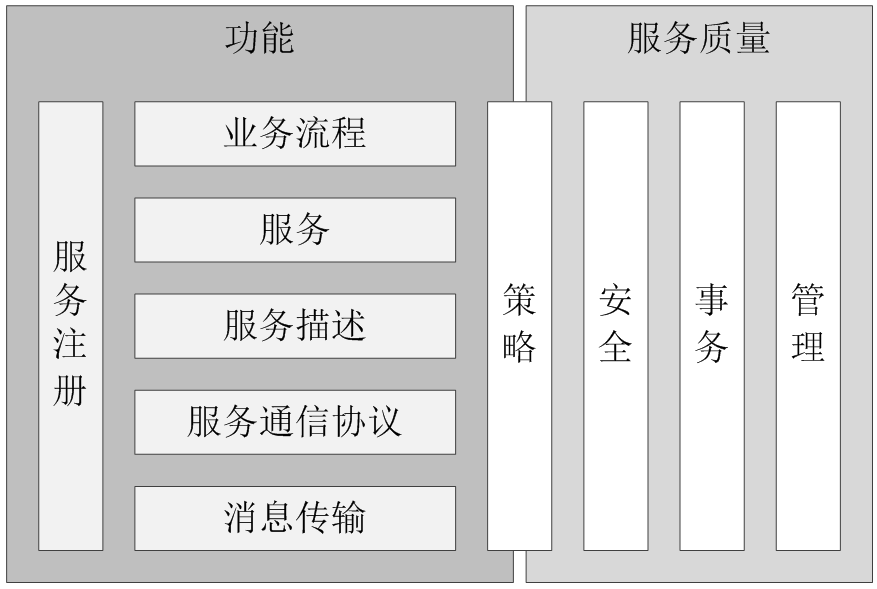


图 2‑1 面向服务体系架构的组成要素

Figure 2‑1 Elements of a service-oriented architecture

功能性方面的组成要素包括：

（1）消息传输（Transport）是一种机制，用于将来自服务使用者的服务请求传送给服务提供者，并且将来自服务提供者的响应传送给服务使用者。

（2）服务通讯协议（Service Communication Protocol）是一种经过协商的机制，通过这种机制，服务提供者和服务使用者可以就将要请求的内容和将要返回的内容进行沟通。

（3）服务描述（Service Description）用于描述服务是什么、应该如何调用服务以及成功地调用服务需要什么数据。

（4）服务（Service）描述实际可供使用的服务。

（5）业务流程（Business Process）是一个服务的集合，可以按照特定的顺序并使用一组特定的规则进行调用，以满足业务要求。业务流程本身可以被看作是服务，这样就产生了业务流程可以由不同粒度的服务组成的观念。

（6）服务注册中心（Service Registry）是一个服务和数据描述的存储库，服务提供者可以通过服务注册中心发布它们的服务，而服务使用者可以通过服务注册中心发现或查找可用的服务。服务注册中心可以给需要集中式存储库的服务提供其他的功能。

服务质量方面的组成要素包括：

（1）策略（Policy）是一组条件和规则，在这些条件和规则之下，服务提供者可以使服务可用于使用者。策略既有功能性方面，也有与服务质量有关的方面；因此，我们在功能和服务质量两个区中都有策略功能。

（2）安全（Security）是一组应用于调用服务的服务使用者的身份验证、授权和访问控制的规则集。

（3）事务（Transaction）是可用于一组服务以提供一致的结果的属性集。例如，假如要使用一组服务来完成一项业务功能，则所有的服务必须要么都完成，要么都不完成。

（4）管理（Management）是可用于管理提供的服务或使用的服务的属性集。

### SOA的角色与协作

面向服务架构中的主要角色包括服务使用者、服务提供者和服务注册中心，它们遵循“发现、绑定和调用”模式进行协作。图 2‑2展示了这种协作关系。

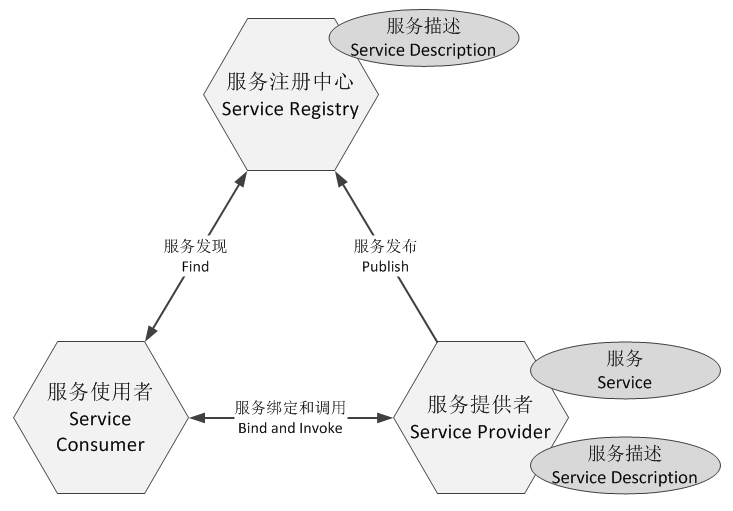


图 2‑2 面向服务架构中的角色与协作

Figure 2‑2 Collaborations in a service-oriented architecture

面向服务架构中的角色包括：

（1）服务使用者（Service Consumer）：服务使用者是一个应用程序、一个软件模块或需要一个服务的另一个服务。它发起对注册中心中的服务的查询，通过传输绑定服务，并且执行服务功能。服务使用者根据接口协议来执行服务。

（2）服务提供者（Service Provider）：服务提供者是一个可通过网络寻址的实体，它接受和执行来自使用者的请求。它将自己的服务和接口协议发布到服务注册中心，以便服务使用者可以发现和访问该服务。

（3）服务注册中心（Service Registry）：服务注册中心是服务发现的支持者。它包含一个可用服务的存储库，并答应感爱好的服务使用者查找服务提供者接口。

面向服务架构中的每个实体都扮演着服务提供者、使用者和注册中心这三种角色中的某一种（或多种）。

面向服务机构中的操作包括：

（1）发布（Publish）：为了使服务可访问，需要发布服务描述以使服务使用者可以发现和调用它。

（2）发现（Find）：服务请求者定位服务，方法是查询服务注册中心来找到满足其标准的服务。

（3）绑定和调用（Bind and invoke）：在检索完服务描述之后，服务使用者继续根据服务描述中的信息来调用服务。

### 企业服务总线

在面向服务架构的实现中，企业服务总线（Enterprise Service Bus，ESB）处于非常重要的位置。企业服务总线是实现面向服务架构的一种最佳实践（而不是特定的产品），如图 2‑3所示，它是位于服务使用者和服务提供者之间的具备消息转换和消息路由功能的企业范围的消息总线。

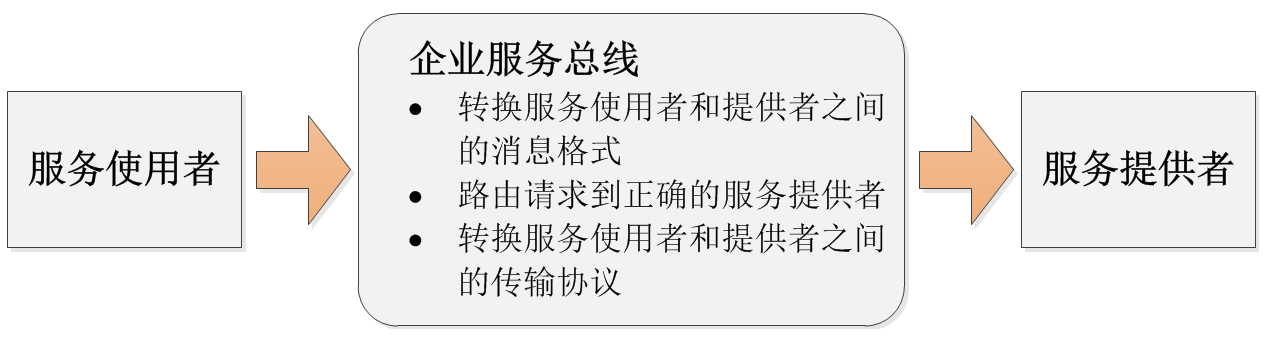


图 2‑3 企业服务总线

Figure 2‑3 Enterprise Service Bus

企业服务总线提供服务中介的能力，解耦了服务请求者和服务提供者，使得服务使用者能够以技术透明和位置透明的方式来访问服务。

ESB的基本特征和能力包括：描述服务的元数据和服务注册管理；在服务请求者和提供者之间传递数据及对这些数据进行转换的能力，并支持由实践中总结出来的一些模式如同步模式，异步模式等；发现、路由、匹配和选择的能力，以支持服务之间的动态交互，解耦服务请求者和服务提供者。高级一些的能力，包括对安全的支持、服务质量保证、可管理性和负载平衡等。

ESB所提供的基于标准的连接服务，将应用中实现的功能或者数据资源，转化为服务请求者能以标准的方式来访问的服务；当请求者来请求一个服务时，ESB中这种中介转化过程可能简单到什么也没有，也可能要很复杂的中介服务支持，包括动态地查找、选择一个服务，消息的传递、路由和转换、协议的转换。这种中介过程，是ESB借助于服务注册管理及问题域相关的知识（如业务方面的一些规则等）自动进行的，不需要服务请求者和提供者介入，从而实现了解耦服务请求者和提供者的技术基础。这使得服务请求者不需要关心服务提供者的位置和具体实现技术，双方在保持接口不变的情况下，各自可以独立地演变。

所以，ESB采用总线结构模式简化了应用之间的集成拓扑，通过源自实践的模式，提供了基于标准的通用连接服务，使得服务请求者和服务提供者之间可以以松散耦合、动态的方式交互，从而在不同层次上使得SOA解决方案是一个松散耦合、灵活的架构。

## **SOA**的方法论

如何贯彻SOA的原理和思想构建出符合SOA设计原则的IT系统呢？IBM提出的SOMA（Service Oriented Modeling and Architecture，面向服务的建模与架构）为面向服务的分析和设计提供了一种切实可行的方法。[3]

### SOMA概述

IBM的SOMA将面向服务的分析和设计分为服务发现、服务规约和服务实现三个阶段。服务的实现包括服务、组件和服务组装的实现。

在开始面向服务的分析和设计之前，需要明确业务需求和系统现状，它们是SOMA分析和设计过程的输入。具体包括：

（1）业务领域（Business Domain）和业务功能域（Business Function Area）。业务领域和业务功能域的划分勾勒了目标企业的业务结构，它一方面帮助我们从全局的角度来理解目标企业的业务，另一方面也是我们进行组织服务层次结构的重要依据。

（2）业务流程（Business Process）。业务流程，尤其是第一级的业务流程，对企业经营全局至关重要。通常，通过第一级的业务流程可以追溯到企业中最为重要的业务活动，因此第一级业务流程是我们进行服务分析和设计的主要入口点。

（3）业务目标（Business Goal）。组织和业务流程都是为业务目标服务的，为了完成业务目标，组织和业务流程都有可能进行适当的调整。分析业务目标在有些时候可以帮助我们发现一些通过业务流程分析遗漏的服务；与此同时，业务目标也是服务描述中一部分重要的内容。

（4）现有系统（Existing System）。现有系统是目前业务活动和业务流程的写照，通过分析现有系统模块和功能，能够帮助发现服务。与此同时，对于现有系统的分析和理解是进行服务实现设计的重要前提。

在掌握了业务领域划分、业务流程、业务目标和现有系统后，SOMA按照三个阶段来进行服务分析和设计——发现服务、描述服务和服务实现，如图 2‑4所示。

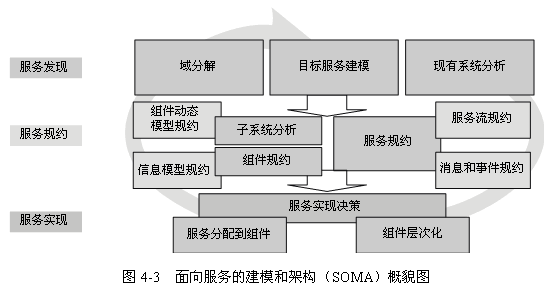


图 2‑4 **面向服务的建模和架构（SOMA）概貌图**

Figure 2‑4 Service Oriented Modeling and Architecture (SOMA)

在SOMA三个阶段的分析和设计过程中，分析和设计人员还需要借助于传统方法中的一些素材，如业务环境和业务用例、IT环境、当前应用或组件的模型和设计等，从而完成与现有业务和IT紧密结合的服务规范和服务设计。在运用SOMA的过程中，这三个阶段并不是一次性完成的，一般需要一个迭代的过程。另外，从企业范围而言，分析和确定的服务模型也有一个演化的过程，并逐渐地精化，越来越贴近业务。

### 服务发现

服务发现是SOMA进行服务分析和设计的第一步。服务发现的主要任务，是确定在一定范围内（通常是企业范围，或若干关键业务流程范围内）可能成为服务的候选者列表。

目前有三种方式发现服务的候选者，它们分别是自上而下的领域分解、自下而上的现有系统分析和中间对齐的业务目标建模。

**（1）自上而下（领域分解）方式**

自上而下的领域分解方式从业务着手进行分析，选择端到端的业务流程进行逐层分解至业务活动，并对其间涉及的业务活动和业务对象进行变化分析。

业务组件模型是业务领域分解的输入之一。业务组件模型是一种业务咨询和转型的工具，它根据业务职责、职责间的关系等因素，将业务细分为业务领域、业务执行层次和业务组件。由于企业内部和外部环境的不同，每个业务组件在成本、投资、竞争力等方面不尽相同，因此，每个业务组件在企业发展的过程中战略职责和演化的路径也是不同的，于是由于角度的不同，就形成了所谓的业务组件的"热点视图"。SOA是一种特别强调业务和IT互动的技术。对于面向服务的分析和设计，业务组件模型提供了进行服务划分的依据，而且这种划分的方法可以平滑地从业务视图细化到服务视图。

端到端的业务流程是业务领域分解的另一个输入。将业务流程分解成子流程或者业务活动，逐级进行，直到每个业务活动都是具备业务含义的最小单元。流程分解得到的业务活动树上的每一个节点，都是服务的候选者，构成了服务候选者组合。业务领域分解可以帮助发现主要的服务候选者，加上自下而上和中间对齐方式发现的新服务候选者，最终会构成一个服务候选者列表。在SOA的方法中，服务是业务组件间的契约，因此将服务候选者划分到业务组件，是服务分析中不可或缺的一步。服务候选者列表经过业务组件的划分，会最终形成层次化的服务目录。

变化分析的目的是将业务领域中易变的部分和稳定的部分区分开来，通过将易变的业务逻辑及相关的业务规则剥离出来，来保证未来的变化不会破坏现有设计，从而提升架构应对变化的能力。变化分析可能会从在未来需求的分析中发现一些新的服务候选者，这些服务候选者需要加入到服务候选者目录中。

**（2）自下而上（已有资产分析）方式**

自下而上的已有资产分析方式的目的是利用已有资产来实现服务，已有资产包括：已有系统、套装或定制应用、行业规范或业务模型等。

通过对已有资产的业务功能、技术平台、架构及实现方式的分析，除了能够验证服务候选者或者发现新的服务候选者，还能够通过分析已有系统、套装或定制应用的技术局限性，尽早验证服务实现决策的可行性，为服务实现决策提供重要的依据。

**（3）中间对齐（业务目标建模）方式**

中间对齐的业务目标建模方式的目的是帮助发现与业务对齐的服务，并确保关键的服务在流程分解和已有资产分析的过程中没有被遗漏。

业务目标建模将业务目标分解成子目标，然后分析哪些服务是用来实现这些子目标的。在这个过程中，为了可以度量这些服务的执行情况并进而评估业务目标，我们会发现关键业务指标、度量值和相关的业务事件。

结合这三种方式的分析，我们发现服务候选者组合，并按照业务范围划分为服务目录。同时为服务规约做好其他准备，如通过对已有资产分析进行的技术可行性评估，通过业务目标建模发现的业务事件等。

### 服务规约

经过服务发现阶段，服务目录基本形成，但是对于每个服务本身的属性信息依然零散。为了能够将服务作为业务和IT层面互动的契约，服务规约阶段是必不可少的。服务规约阶段的主要任务是规范性地描述服务各个方面的属性，其中既包括输入/输出消息等功能性属性，服务安全约束和响应时间等服务质量约束，以及服务在业务层面的诸多属性，如涉及的业务规则、业务事件、时间/人员消耗等。与此同时，规范描述服务相关方面的关系也很重要，如服务间依赖关系，服务和业务组件间关系，服务和IT组件间关系和服务消息间关系等。

进行服务暴露决策是服务规约的第一步。理论上所有的服务候选者都可以暴露为服务，但是一旦暴露为服务，该服务候选者就必须满足附加的安全性、性能等方面的要求。企业还必须为服务的规划、设计、开发、维护、监管支付额外的开支，因此，我们会根据一定的规则来决定将哪些服务候选者暴露为服务。

这些规则包含以下几个方面：

（1）业务对齐。该服务候选者可以支持相关的业务流程和业务目标。

（2）可组装。该服务候选者满足技术中立、自包含及无状态等特点，同时还满足复合应用的相关非功能性需求。

（3）可重用。该服务候选者可以在不同的应用、流程中重用，从而减少重复的功能实现，降低开发和维护的成本。

基于企业应用开发的经验，我们还可以有其他一些方面的考虑。

经过服务暴露决策后，层次化的服务目录基本形成。下一步是结合传统的方法学对服务各方面属性进行描述。这里说的传统的方法学是指企业架构，面向对象的分析和设计等。在企业架构方法学的产物中，企业数据模型有助于服务消息的定义，IT组件模型有助于服务和IT组件间关系的描述，企业安全架构有助于服务安全约束规约，企业基础设施架构有助于服务质量的描述。在面向对象分析和设计的产物中，业务用例和系统用例有助于服务消息、服务相关业务规则和业务事件等描述，组件静态模型和动态模型有助于描述服务间关系。

经过服务规约，服务组件（企业组件）和服务的各个方面的属性被规范下来，它会成为业务和IT层面互动的基础。以后，业务对IT的新需求会体现为服务层面的变更，IT层面的变化会尽量遵循服务规约。在SOA监管的配合下，任何对服务规约的变更都是可管理和可控制的。

### 服务实现

经过服务规约阶段，作为业务和IT互动的服务契约已经形成。但是服务契约和IT的现状还是有很大差距的，如和某个服务对应的业务逻辑分散于不同的应用中，分散在不同的地域，某些服务目前主要依靠人工完成，还没有IT层面的实现。

为了将服务契约落在实地，服务实现阶段通过差距分析，并结合传统方法学完成每个服务实现决策。这其中包括的内容甚多，其主要内容如下。

（1）现有系统分析：调研现有系统架构，了解架构风格、主要架构元素和能力和架构元素的基本特征；调研现有应用，了解应用主要功能和对外接口，技术实现特征等；如果应用构建已经遵循基于组件开发规范，编制应用已有组件目录；如果应用并没有组件化，将应用覆盖的业务功能和服务规约确定的企业组件进行映射，确定应用现有“组件”目录。

（2）确定服务分配：通过服务组件和现有系统分析确定的IT组件间差距分析，确定服务组件和IT组件间映射关系。例如，一个服务组件对应一个或多个IT组件，没有IT组件和服务组件对应，没有服务组件和IT组件对应，服务组件和IT组件对应时有能力缺失或不匹配等。经过差距分析，一些服务中介被确定下来去实现服务路由，或消息格式等不匹配现象，一些新的IT组件被确定下来，如实现某业务流程的组件或实现人工服务的组件等。最终，服务组件都被映射到IT组件上，从而完成服务分配。

（3）服务实现决策：服务分配仅仅确定了需要哪些组件来实现服务，但是并没有实现的策略和技术层面的决策。服务实现决策首先帮助确定服务实现策略，是在现有基础上进行服务包装，还是重新构建，如果重新构建，是采用已经包装好的应用，还是外包，或者自己构建；如果是服务包装的话，有哪些候选方案等。通常服务实现决策和传统的架构决策是关联在一起的。

（4）服务基础设施设计：服务的功能实现，非功能需求的满足都需要服务基础设施的支持。在进行服务实现决策后，需要根据具体的需求确定服务基础设施的能力，如用于支撑人工服务的人工服务容器，用于支撑服务编排的流程引擎等。

将服务实现阶段的各种产物和传统设计方法结合起来，就可以开始指导实际服务实现的实施。

## 本章小结

本章内容是课题研究设计的基础。首先说明SOA的基本概念，包括定义、优点及主要特征；接着介绍面向服务架构的组成要素，服务使用者、提供者和服务注册中心的协作，并对SOA实现中最关键的企业服务总线（ESB）的作用和特性做了说明；最后，以IBM的SOMA（面向服务的建模与架构）为例，解释了SOA方法学的主要内容。