**深究什么是SOA**

**1.     背景**

IT行业就是术语和缩写流行的行业，各大厂商都喜欢隔三差五地推出一些新概念。为了不落人后，大家都喜欢争先恐后地跟进。有深入研究、务实研发的供应商，能够将概念落地，不断推出创新的产品和服务，赢得竞争优势。但“贴标签”的也大有人在，而且趋势是越贴越多，跟风炒作，“鱼目混珠，泥沙俱下”，以至于“混绕视听”了。

SOA就是这俱多“三字母”缩写的概念之中的最流行和热门的一个。但目前，SOA概念和解决方案，话语权方面基本上被国外巨头所控制，特别是大的中间件厂商。

但是真正能够完整实现SOA的落地解决方案和案例很少，刻意包装的成分比较多，特别是应用架构方面。重技术，轻方法论，造成企业实施SOA缺乏足够的架构方法、SOA治理、SOA实施运维方面的最佳实践，因此企业实施SOA缺乏系统的指导。

另一方面，国内的不少软件企业，由于不能提供完整意义上的SOA解决方案，只能提供部分的组件，小部分特性符合SOA思想，所以就任意曲解SOA的含义，随意解析SOA的概念。以至于国内没有一家软件企业不宣传SOA，不宣称其产品符合SOA架构的。由此造成，许多企业和客户对SOA是非常茫然的，对SOA的价值也转向怀疑和抵触。

这种厂商之间的无序竞争，不利于国内企业的自主创新，也不利于企业导入和实施有效的SOA，实现SOA的商业价值。

本文试图就SOA的来龙去脉，外延内涵和前世今生，来一个全面的阐释。一家之言，权作业界参考，希望带动大家做一些更深入的思考。文章比较长，如果兴趣不够，也可以就此打住。

**2.     为什么需要SOA**

SOA的出现不仅仅是厂商炒作的结果，本质上是两种力量驱动的结果：需求拉动、技术推动。业务需求的拉动，希望解决业务应用的问题；技术发展的推动，使得SOA具备了技术上的可行性，软件技术的发展推动了IT创新的商业价值。

**2.1.    需求拉动**

需求拉动方面，主要来自于两种信息化的困境。一个是“信息孤岛”造成基于系统之间互联互通的整合需求；另一个是业务的变化所导致对IT灵活性，以适应变化的需求。

目前国内外基本情况类似，经过30年的信息化建设，许多企业和政府部门都在不同时期、应用不同技术、与不同的厂商合作，建设了不同规模的应用系统，造成了信息化不是没有系统，而是信息孤岛太多的问题，而且不是没有数据，而是信息不一致，难以整合。因此，互连互通是当前信息化中的核心问题和核心需求。顺便说一句，那些认为中国企业的信息化起步晚，历史负担少，可以快速部署全新SOA，可以运用推倒重来的策略是不了解中国企业信息化，自我想当然的结果。事实上，我国信息化无论是金融、电信、电力等大行业，还是中小工商企业，“孤岛现象”还是非常严重，遗留系统的整合不一定就少，而且我国软件供应商的系统普遍架构能力比较弱，整合难度一点也不低。

这种互联互通需求，既包括企业内的各种应用系统之间的集成，也包括集团企业总部与下属企业、企业与上下游伙伴之间的业务协同。





 企业内互联互通的需求

另一方面，激烈的竞争和产业变革，需要企业不断调整其组织、流程和商业模式，以获得竞争优势，造成业务的不断变化，而且随着经济全球化，这种变革的步伐在不断加快。但僵化的IT基础设施难以迅速响应这种变化，造成IT与业务的不平衡和不匹配。因此，IT的灵活性以适应业务变革的需求，也是当前信息化建设过程中所面临的最大挑战。而且这方面的变革速度和变革幅度比国外许多企业都要大得多，毕竟我们的企业还是在快速成长，快速成熟的过程之中。





业务灵活性的需求需要一种新的架构技术来支撑企业实现其快速的灵活应变的业务战略。传统的信息化方法和软件研发方法是基于业务需求的直接映射。这种需求驱动的信息系统最大的缺陷就是对变化的适应性差，这也是传统软件工程造成的“软件危机”最直接的表现。如果要满足业务需求的柔性，就需要按照架构驱动，对业务进行适当的抽象，通过服务的表达和业务过程的原子化，来满足系统是按照企业架构来构造，这种架构是动态重构技术来支撑的，我们今天知道了，这种架构就是SOA。

**2.2.    技术推动**

软件出现最早是用于科学计算，然后是计算机辅助设计、辅助制造等等工业应用。在企业管理领域大规模应用后，业务需求不断的变化、系统不断增加、流程更复杂、系统越来越不堪重负，出现了需求交付方面的重大挑战，以至于人们用“软件危机”来描述软件工业所面临的困境。

软件技术发展过程中，一直在寻求解决四个基本问题的方法：质量问题、效率问题、互操作问题、柔性构造问题。这些问题今天依然困扰着软件行业。

造成这个局面的原因是异构性和标准规范的滞后。

* **屏蔽异构性**

异构性表现在计算机的软硬件之间的异构性，包括硬件（CPU和指令集、硬件结构、驱动程序等），操作系统（不同操作系统的API和开发环境）、数据库（不同的存储和访问格式）等等。长期以来，高级语言依赖于特定的编译器和操作系统API来编程，而他们是不兼容的，因此软件必须依赖于开发和运行的环境。

造成异构的原因源自市场竞争、技术升级以及保护投资等因素。希望屏蔽异构平台的差异性问题是促成中间件发展的驱动力之一。而支持SOA架构的中间件平台，已经在很大程度上屏蔽了系统环境的差异性，提供了一致的计算环境。

* **实现互操作**

因为异构性，产生的结果是软件依赖于计算环境，使得各种不同软件之间在不同平台之间不能移植，或者移植非常困难。而且，因为网络协议和通信机制的不同，这些系统之间还不能有效地相互集成。

造成互操作性不好的原因，主要是标准的滞后。解决软件之间的互操作性问题也是促成中间件发展的驱动力之一。而SOA技术从一开始就强调了标准的重要性，包括中间件平台的实现上都是基于全球共同的标准来实现。

* **共性凝练和复用**

软件应用领域越来越多，相同领域的应用系统之间许多基础功能和结构是有相似性的，每次开发系统都从零开始绝对不是一种好的方法，也是对质量和效率的很大的伤害。

尽可能多地凝练共性并复用以提高软件开发效率和质量，通过中间件通过提供简单、一致、集成的开发和运行环境，简化分布式系统的设计、编程和管理，这也是SOA发展的重要推动力。

软件技术发展内容，包括更好的程序设计语言、更好的平台和软件开发技术，如面向对象、组件开发、面向服务等等。而这方面，在技术上逐渐发展的成果大部分都凝聚在今天的SOA解决方案之中。





程序设计语言的发展

而这些技术推动因素，从本质上是通过复用、松耦合、互操作(标准)等机制来提高软件质量、加快软件研发效率、使研发出来的产品能够相互集成并灵活适应变化。

这些技术因素逐渐推动了SOA架构的形成和发展。





SOA架构的发展

**3.     如何准确理解SOA**

我并不打算介绍SOA的定义，事实上到现在为止，还没有一个权威的SOA标准定义，因为从不同角度，不同厂商和学术团队会有不同的答案。争论定义本身，不是目的。

OASIS（一个SOA标准组织）给予出的SOA定义“SOA是一个范式，用于组织和利用可能处于不同所有权范围控制下的分布式系统。”

维基百科给出的SOA定义“面向服务的体系结构（Service-oriented architecture）是构造分布式系统的应用程序的方法。它将应用程序功能作为服务发送给最终用户或者其他服务。它采用开放标准、与软件资源进行交互并采用表示的标准方式。”。

这些定义本身，一般人员要准确理解是非常困难的，既便是专业人士，未必能够深刻理解其内涵。如何更加形象理解SOA？怎么通俗化解析SOA的核心含义？

**3.1.    如何形象理解SOA**

事实上，SOA的思想我国很早就有了，印刷术的发展过程其思想就完整体现了SOA的核心含义。

印刷的内容――文字，在秦始皇统一六国之前，各国的文字是不统一的，据说许多常用的文字有十几种写法和读音，妨碍了各国之间的文化交流，就象SOA之前，各种软件平台、各种开发工具和各种接口的组件之间，没有统一的标准，对软件系统之间的整合造成巨大的困难。

因此，伟大的始皇帝统一了六国文字，“书同文、车同轨”就是通过标准解决“复用”和“互操作”等问题。这也为大规模的印刷和文明发展提供了一个良好的基础，这种“统一封装”的文字，对文化交流起到了一个“互操作”的标准作用。





SOA的形象解析

在没有印刷术之前，书籍要依赖于手工抄写，这样效率当然是非常低下，而且质量也不能获得一致性的保证，也就是书籍还无法“复用”。中国人首先发明了刻版印刷术，就是将书籍刻成一块一块的凸字版，然后就可以大规模进行印刷了，当印刷出来的书籍脱销时，下次还可以继续使用，大大提高了效率，这就是“复用”，软件通过组件的封装，也可以达到重复和在不同场合多次使用的“复用”效果。

刻版印刷术有个很大的问题就是文字之间是紧耦合的，同样一个字，在另一部书之中是不能“复用”的，必须重新雕刻，也就是说刻版印刷是没有“编排”特性的。就如软件技术中微软VB开发的Com+组件就只能在Windows环境之中使用，它不能与Java开发的EJB组件进行复用和编排，因为他们与开发环境和运行环境是紧耦合的，要在UNIX环境下使用，必须重新开发（相当于重新“刻版”）。活字印刷就是通过文字与版面之间的松耦合，通过“排版”来实现一部书的印刷版面的，这种松耦合就大大提高了文字的字模之间的复用和编排效率。我们标准封装的“服务”就类似一个一个的字模，通过服务编排（“排版”）来实现业务流程。

统一文字和活字印刷促进了人类文明进步，而SOA促进全球IT架构和应用的革命。

**3.2.    SOA的核心要素**

要准确全面理解SOA，首先必须理解SOA的核心要素：





SOA的核心要素

SOA的目标就是实现灵活可变的IT系统。要达到灵活性，通过三个途径来解决：标准化封装、复用、松耦合可编排。

互操作(标准化封装)、复用、松耦合等SOA技术的内在机制，也是中间件技术和产品的本质特征。

* **标准化封装（互操作性）**

传统软件架构，因为封装的技术和平台依赖性，一直没有彻底解决互操作问题。互联网前所未有的开放性意味着各节点可能采用不同的组件、平台技术，对技术细节进行了私有化的约束，构件模型和架构没有统一标准，从而导致架构平台自身在组件描述、发布、发现、调用、互操作协议及数据传输等方面呈现出巨大的异构性。各种不良技术约束的结果是软件系统跨互联网进行交互变得困难重重，最终导致了跨企业/部门的业务集成和重组难以灵活快速的进行。

在软件的互操作方面，传统中间件只是实现了访问互操作，即通过标准化的API实现了同类系统之间的调用互操作，而连接互操作还是依赖于特定的访问协议，如JAVA使用RMI，CORBA使用IIOP等。而SOA通过标准的、支持Internet、与操作系统无关的SOAP协议实现了连接互操作。而且，服务的封装是采用XML协议，具有自解析和自定义的特性，这样，基于SOA的中间件还可以实现语义互操作。

SOA要实现互操作，就是通过一系列的标准族，来实现访问、连接和语义等各种层面的互操作。

* **软件复用**

软件复用，即软件的重用，也叫再用，是指同一事物不作修改或稍加改动就多次重复使用。从软件复用技术的发展来看，就是不断提升抽象级别，扩大复用范围。最早的复用技术是子程序，人们发明子程序，就可以在不同系统之间进行复用了。但是，子程序是最原始的复用，因为这种复用范围是一个可执行程序内复用，静态开发期复用，如果子程序修改，意味着所有调用这个子程序的系统必须重新编译、测试和发布。





SOA的复用

为了解决这个问题，人们发明了组件（或者叫控件），如MS操作系统下的DLL组件。组件将复用提升了一个层次，因为组件可以在一个系统内复用（同一种操作系统），而且是动态、运行期复用。这样组件可以单独发展，组件与组件调用者之间的耦合度降低。

为解决分布式网络计算之间的组件复用，人们发明了企业对象组件，如（Com+,.NET,EJB等)，或者叫分布式组件。通过远程对象代理，来实现企业网络内复用，不同系统之间复用。

传统架构的核心是组件对象的管理。但分布式组件也是严重依赖其计算环境，由于构件实现和运行支撑技术之间存在着较大的异构性，不同技术设计和实现的构件之间无法直接组装式复用。

而现代SOA的重要特征就是以服务为核心，如WebService,SCA/SDO等。通过服务，或者服务组件来实现更高层次的复用、解耦和互操作，即SOA架构中间件。

因为服务是通过标准封装，服务组件之间的组装、编排和重组，来实现服务的复用。而且这种复用，可以在不同企业之间，全球复用，达到复用的最高级别，并且是动态可配置的复用。

* **耦合关系**

SOA架构在松耦合解耦过程也发展到了最后的境界。传统软件将软件之中核心三部分网络连接、数据转换、业务逻辑全部耦合在一个整体之中，形成“铁板一块”的软件，“牵一发而动全身”，软件就难以适应变化。分布式对象技术将连接逻辑进行分离，消息中间件将连接逻辑进行异步处理，增加了更大的灵活性。消息代理和一些分布式对象中间件将数据转换也进行了分离。而SOA架构，通过服务的封装，实现了业务逻辑与网络连接、数据转换等进行完全的解耦。





SOA不断解耦的过程

总之，从科学哲学的角度来看，SOA是一个不断解构的过程，传统软件强调系统性，耦合度过高，所以需要松耦合（解耦）；SOA也是一个组件粒度的平衡，集成电路趋势是集成度越来越高，软件发展的趋势是相反的过程；SOA是架构，更是方法，反映了人们对哲学思想的追求的原动力。

按照这个特性，SOA基本上来说与WebService并不是同一个概念，SOA并不一定需要WebService实现，理论上可以在其他技术体系下，实现SOA。但事实上，到目前为止，能够实现SOA架构风格的技术就是WebService，因为它的特性和厂商的支持力度，使得WebService成为了实现SOA实现技术的事实标准。也正因为WebService技术的成熟，才使得已经提出10多年了的SOA思想和概念，得以能够实现落地，成为一种可以使用的技术。这也就是回答了SOA和WebService的关系。

**3.3.    SOA的架构框架(Framework)**

SOA的核心主体是服务。所谓“服务（Service）” ，从业务角度而言，服务是一个可重复的经过标准封装的任务，例如: 检查帐号余额;开新帐户 等等…。SOA的目标是通过服务的流程化来实现业务的灵活性，所谓流程（Process）是由一系列相互关联的任务所组成，实现一个具体的业务功能。一个流程可以由一系列服务来实现。





SOA治理

服务就像一堆“元器件”，这些元器件通过封装形成标准服务，他们有相同的接口和语义表达规则。但服务要组装成一个流程和应用，还需要有效的“管理”，包括如何注册服务、如何发现服务、如何包装服务的安全性和可靠性，这些就是SOA治理。SOA治理乃是将SOA这一堆元器件，进行有效组装，形成一个“产品”的关键，否则它永远是一堆器件，而无法形成一个有机整体。

SOA治理的方法和体系，就是区别于一般组件开发的技术的重要区别和特征。

一个正确的框架，是指导我们开发和实施SOA架构的基础。由IBM提案，国际开放群组(The Open Group)提出了一个SOA架构的参考模型，这个架构框架目前是产业界最权威和严谨的SOA架构标准。The Open Group是一个非营利标准化组织，是一个厂商中立和技术中立的机构，致力于提出各种技术框架和理论结构，致力于促进全球市场的业务效率。The Open Group已有超过20年的标准制定与推广历史。在1996年，由X/Open与Open Software Foundation合并组成。The Open Group最有名是作为UNIX商标的认证机构。在过去，协会最出名的是其出版的Single UNIX Specification，它扩充了POSIX标准而且是UNIX的官方定义，其成员包括IT用户、供应商以及政府机构。The Open Group在中国的创始会员为金蝶集团，金蝶集团负责成立了中国分会。TOG在1993年提出的The Open Group Architecture Framework (TOGAF) 架构框架，是一套行之有效的企业架构。历经15年9个版本发展，支持开放、标准的SOA参考架构，已被80％的福布斯( Forbes)全球排名前50的公司使用。这个SOA参考模型为：





SOA标准模型

根据这个模型，完整的SOA架构由五大部分组成，分别是：基础设施服务、企业服务总线、关键服务组件、开发工具、管理工具等。

SOA基础实施是为整个SOA组件和框架提供一个可靠的运行环境，以及服务组件容器，它的核心组件是应用服务器等基础软件支撑设施，提供运行期完整、可靠的软件支撑。

企业服务总线是指由中间件基础设施产品技术实现的、通过事件驱动和基于XML消息引擎，为SOA提供的软件架构的构造物。企业服务总线ESB提供可靠消息传输、服务接入、协议转换、数据格式转换、基于内容的路由等功能，屏蔽了服务的物理位置，协议和数据格式。在SOA基础实现的方案上，应用的业务功能能够被发布、封装和提升（Promote）成为业务服务（Business Service）；业务服务的序列可以编排成为BPM的流程，而流程也可以被发布和提升为复合服务（Composited Service），业务服务还可以被外部的SOA系统再次编排和组合。ESB是实现SOA治理的重要支撑平台，是SOA解决方案的核心，从某种意义上说，如果没有ESB，就不能算作严格意义上的SOA。

关键服务实现，是SOA在各种业务服务组件的分类。一般来说，一个企业级的SOA架构通常包括：交互服务、流程服务、信息服务、伙伴服务、企业应用服务和接入服务。这些服务可能是一些服务组件，也可能是企业应用系统（如ERP）所暴露的服务接口等等。这些服务都可以接入ESB，进行集中统一管理。

开发工具和管理工具：提供完善的、可视化的服务开发和流程编排工具，涵盖服务的设计、开发、配置、部署、监控、重构等完整的SOA项目开发生命周期。

按照这个模型，许多SOA解决方案是只提供部分实现。这个行业中，许多国内的企业为了搭上SOA的便车，经常以偏概全，混绕概念。应该说真正按照SOA的思想和模型来构建整个企业的IT架构的案例是非常之少的。许多国外厂商的宣传案例，基本上是停留在部署应用服务器，开发了部分WebService组件，可以实现部分数据集成，这个层次而已，而这些WebService是部署在ESB平台之上的，就已经很不错了。实现了服务流程重组，实现SOA治理的案例就更是很少见到了。

国内有许多软件企业开发的系统，宣传是SOA架构的。基本上有几种情况，其一，有些开发组件和开发平台厂商，他们也自称中间件企业，基本上是提供一个工作流平台，许多还不支持BPEL的业务流程管理，只是传统的XPDL/WfMC工作流平台（Workflow不同于支持服务流程的Business Process），最常见的案例是OA办公审批，或者服务组件开发工具，而所谓的ESB产品大部分都是EAI的升级，可以与Webservice进行接口而已，就宣称这是ESB产品了，基本的服务注册、服务编排和安全管理都不具备。这些解决方案只是提供了许多WebService开发的组件，而不提供SOA治理的核心架构，相当于造了许多元器件，但还不能提供整机产品。

其二，许多宣称SOA架构的应用软件，基本上可以说是“支持”SOA，而不能称为“基于SOA”架构。因为支持SOA一般是指可以将其某些功能，封装为服务（WebService），可以在SOA架构之中进行管理，这比较容易达到。而“基于SOA”是指应用系统的业务功能都是封装为服务，通过ESB进行集中管理，业务实现是通过BPEL业务流程管理进行编排，用户交互是通过交互服务（如门户）进行管理，整个解决方案可以达到标准服务封装、服务复用、松耦合、服务编排与重组，并且基本符合TOG-SOA的架构模型。

按照这个标准，IT用户就可以了解到真正的SOA架构的框架模型，就可以识别是否是企业所需要的架构。

讲到这里，我们已经很清楚了，对于SOA的理解，有些学者或者咨询公司强调SOA不是一种技术，也不是软件，而是一种思想，一种架构风格。我认为这也是不完全准确的，这种观点认为SOA仅仅是思想和方法，将使得SOA成为一种不可知论，飘在空中，很难落地。

**4.     SOA如何落地**

自从1996年Garnter的分析师首次提出SOA的概念，到今天，已经过去10多年了，在前几年这个概念并没有引起多大的注意，只是到这几年在技术上WebService的成熟，以及互联网的普及，SOA作为企业架构的主要技术，才逐渐占据了产业的主流。

因为商业策略的不同，有些企业提出SOA已经过时，有些企业还认为SOA还没有达到成熟的阶段等等这些论点。从商业概念炒作周期来看，我认为SOA正当喧嚣之后的理性期，是业界和IT用户真正抛弃“炒作”，踏踏实实发挥SOA的商业价值的时候了，SOA需要落地。

SOA将来真正推广到企业中应用，要落地，就不能离开几个基本的东西：支撑SOA的基础中间件平台、符合SOA架构的应用系统（如ERP等）、构建SOA的方法论。





SOA落地途径

**4.1.    架构方法论**

方法和工具构成了工程技术域，要构建SOA架构的企业信息系统，确保业务和IT的真正匹配，首先必须从方法论入手。

许多企业的IT系统“孤岛”现象严重，本质上是缺乏足够有效的整体规划或者架构规划造成的。形象地说，构建企业IT大厦如同我们盖房子是一样的道理。我们许多企业建设信息系统时就采用了盖乡村民宅的做法。盖乡村民宅不需要严谨的规划，也没有复杂的地下设施建设（如自来水供水、排水、供气、地下停车场等），也没有需要建设污水处理、雨水收集等复杂的配套设施。而事实上，企业IT系统建设应该如城市建设，首先需要城市总体规划，然后根据功能区规划，设计和建设小区配套设施，“三通一平”实质就是构建建筑之间的公共基础设施，确保每栋建筑之间不是“孤岛”，然后每栋建筑还需详细的设计和工程施工。如果要消除信息孤岛，实现IT与业务的一致性，也需要有效的企业架构规划和设计。





为什么需要架构规划

透过现象看本质，SOA代表着一种面向服务的IT架构风格，SOA的技术本质和出发点，在于IT架构。而IT架构，是组织的企业架构的重要组成部分，它和组织的战略架构、业务架构一起，形成一个自上而下、紧密联系、相辅相成的有机整体。SOA代表着一种正在蓬勃兴起的革命性IT架构理念，和传统技术体系区别的关键特征之一就在于SOA是战略导向和业务驱动的。而国际和国内的各方面经验都告诉我们，对于一个组织而言，捕获战略、梳理业务和IT的最有效的措施就是架构。

企业架构(Enterprise Architecture，EA)，是从多个角度对组织的构件层次描述的规划蓝图，从各个层面反映组织的愿景、战略、业务、服务、人员、技术和产品及其相互之间的关系，辅以其管控和演进的规则。

一个企业架构内容包括业务架构（Business Architecture）、应用架构（Application Architecture）、信息架构（Information Architecture）、技术架构（Technology Architecture）等。

真正可以落地的SOA建设，必须且只能从架构出发。没有架构，"SOA"将变成一盘无法真正解决各种运营问题的技术和产品的大杂烩。优良的架构填补了业务需求与实际信息系统以及基础设施设计之间难以逾越的鸿沟。

在所有的架构开发方法（ADM- Architecture Development Methods）之中，开放群组TOG的TOGAF是目前最权威和最有影响力的一种。The Open Group于1993年开始应客户要求制定系统架构的标准，在1995年发表The Open Group Architecture Framework (TOGAF) 架构框架。TOGAF的基础是美国国防部的信息管理技术架构(Technical Architecture for Information Management: TAFIM)。TOAGF是一个架构框架，简而言之，TOGAF是一种协助开发、验收、运行、使用和维护架构的工具，它是基于一个迭代(Iterative)的过程模型，支持最佳实践和一套可重用的现有架构资产。它可设计、评估并建立组织的正确架构。TOGAF的关键是架构开发方法ADM：一个可靠的，行之有效的方法，以发展能够满足商务需求的企业架构。而2008年发布的TOGAF 9.0是符合SOA架构开发的最新版本。TOGAF所提出的“无边界信息流（Boundaryless Information Flow）”理念和愿景，是解决目前企业信息化孤岛问题的最有效方式。





TOGAF架构内容

**4.2.    基于SOA的应用系统**

基于SOA的应用系统构建方法与传统软件架构方法有所不同。

首先基于SOA的应用系统建模和管理的组件层次是服务：





面向服务的工程

基于服务的应用系统的本质特征是松耦合，以基本业务功能（服务封装）为系统的基本实现单元，然后通过服务编排（流程管理）来“组装”业务应用系统。相对于以往的应用系统，是面向技术组件，由系统程序实现业务流程，在复用、耦合方面都存在灵活性问题。





软件工程和系统设计的演进过程

基于SOA的应用系统构建过程是：





基于SOA的应用构建过程

服务建模是第一步，也就是服务识别和颗粒度确定。服务识别是方法论的第一步，服务识别的主要任务，是确定在一定范围内（通常是企业范围，或若干业务场景范围内）可能成为服务的候选者列表，并确定服务的颗粒度，以及标识服务的接口。服务建模也就确定了应用系统架构的耦合程度。

服务封装阶段的主要任务是对服务进行规范性的描述，其中包括输入/输出消息等功能性属性，以及服务在业务层面的诸多属性。并决定服务以何种形式向外提供服务。服务可能是新开发的业务功能和业务对象的封装，也可能是遗留系统的服务封装，将遗留系统的软件资产以服务的形式进行封装，在新的架构上利用已有的资产。

服务治理就是将已经封装好的服务进行集中统一有效的管理。通过ESB基础设施，提供服务注册、存储、安全控制和版本管理等。服务注册阶段的主要任务是将服务注册到服务库。此时需要决定服务的命名、安全、性能、时间特性。

服务编排就是根据业务流程的需求，对服务进行组合和组装。服务组装是以实现业务流程为目的，通过对业务服务的组合和组装，实现更粗粒度的业务服务，实现最终的业务需求。

应用交付阶段主要任务是完成业务系统服务化组装和服务部署，实现业务按需交付。

基于SOA的应用系统是SOA架构的重要组成部分，也是SOA落地的地基。

**4.3.    支撑SOA的中间件平台**

SOA方法论和基于SOA的应用系统要落地的支撑工具和技术基础就是中间件平台。这个在3.3.SOA的架构框架(Framework)之中已经阐述清楚了。

根据TOG-SOA模型，完整的SOA架构五大部分中，基础设施服务、企业服务总线、开发工具、管理工具等，都是中间件的基础平台。

交付服务之中的门户，也是需要支持JSR168和JSR286标准的Portlet容器和个性化交互以及终端适配的支撑平台。

业务流程管理需要支持BPEL规范的流程引擎和流程建模的工具，这个中间件平台用来支持服务的组合和服务流程编排，以满足业务重组的需求，来实现业务的灵活性。

SOA要落地的最后支撑平台就是满足SOA规范的中间件技术。

**5.     金蝶Ready SOA解决方案**

金蝶作为全球领先的SOA解决方案供应商，一直以来坚持脚踏实地，自主创新，与北京大学等高校合作，承担振兴国家基础软件的责任和使命，将国际领先的技术与方法，结合中国企业的实际需求，探索国内信息化的最佳实践，提供SOA的完整解决方案。通过多年的努力，金蝶成为中国唯一入选Gartner全球有能力提供SOA服务的十九家软件厂商之一。

金蝶让SOA落地的解决方案品牌为”readySOA”,意为可以落地、可以实施的SOA。

金蝶readySOA的核心内涵包括三个方面：

* 结合TOGAF而形成的SOA实施方法论；
* 拥有中国唯一全球第四通过Java EE 5.0认证的SOA基础设施，中国唯一完整实现TOG-SOA标准模型的金蝶Apusic中间件平台；
* 国内第一套基于SOA实现的企业应用软件金蝶EAS。

我们有这几方面的综合经验，所有金蝶ERP产品都是通过SOA架构去优化的，实现了标准的服务封装，通过金蝶中间件SOA的完整平台来支撑，并可以通过BPEL流程来进行编排和重组，而TOGAF是SOA架构最权威的方法论体系，可以指导企业如何导入、部署和运营SOA架构。

**5.1.    金蝶readySOA实施方法**

金蝶readySOA实施方法包括结合TOGAF形成的SOA实施方法、SOA架构成熟度模型等。

TOGAF架构开发方法ADM提供了可灵活利用的组织企业架构的开发和治理的过程。一个成功的SOA落地项目的建设并非一蹴而就，而是分阶段逐步实现的，其生命周期过程主要可以分为初步阶段、架构阶段、实施阶段、变更管理阶段四大环节完成。通常要从组织的某个独立的业务单元开始，之后再由小及大，逐渐在跨组织范围的整体业务中扩散，逐步完善整个组织的SOA 平台，最终实现随需应变的企业IT架构。针对SOA项目的指导，TOGAF定义了其架构开发方法ADM各阶段和SOA项目全生命周期的各阶段之间的映射关系，形成了具有良好操作性的架构开发方法体系。





    基于TOGAF的SOA实施方法论更详细的金蝶中间件readySOA实施方法论参加金蝶发布相关白皮书和技术资料出版物。

**5.2.    金蝶readySOA中间件平台**

金蝶中间件按照TOG-SOA的架构模型，开发了一套完整实现其模型的中间件平台产品系列：





金蝶Apusic中间件SOA完整实现产品系列

金蝶Apusic应用服务器（AAS）为企业应用提供稳定、高效、安全的开发平台与运行引擎，是所有组件和服务的容器与计算环境。

金蝶Apusic消息中间件(AMQ)提供消息传输服务的基础系统软件，保障数据在复杂的网络中高效、稳定、安全、可靠的传输，并确保传输的数据不错、不重、不漏、不丢，是实现SOA分布式计算的集成通信平台。

金蝶Apusic 企业服务总线（AESB）以面向服务的方式，实现异构、分布式系统之间集成共享、互联互通的基础软件平台，是SOA治理（服务注册、服务存储、服务路由、安全控制和版本管理）的基础设施。

金蝶Apusic业务流程管理（ABPM）是一款面向业务的、具有高度扩展性和强大整合能力的流程中间件，支持BPEL协议，完整并有力支撑了业务流程管理的全生命周期，实现服务的组合和流程编排。

金蝶Apusic数据交换和管理平台AEI（Apusic Exchange & Integrator），能够针对分布式的异构数据源，对数据进行分析、采集、转换、清洗、影射、持久等操作，提供分布式数据源之间数据集成和交换的解决方案，实现SOA的信息服务。

金蝶Apusic门户平台（APS-Apusic Portal Server）使组织的应用、人员、信息、流程有机聚合，使用户能够通过单一渠道访问所需的个性化信息，它支持标准的JSR Portlet的交互和个性化编排。

金蝶Apusic OperaMasks开发平台，是构建SOA服务和集成化的开发平台。AOM AppFrame是开放高效的基础应用运行平台，为企业应用开发提供构件化的编程模型与运行期支撑；AOM Studio是全开发生命周期支持的集成开发环境，为企业应用开发提供从代码开发、展现设计，到配置管理、协同开发的一站式支持；而 AOM BizModeler以直观快速的图形化方式地对业务逻辑进行分析建模、服务封装、流程编排、部署发布。

金蝶Apusic Universal Manager是金蝶中间件2.0产品的统一管控平台，依托于MaaS理念，基于浏览器的、完全可视化的、拥有插件体系可扩展的管控平台。

金蝶中间件Ready SOA支撑平台产品完全按照TOG－SOA参考架构进行设计，传承了金蝶十余年从事大规模关键性企业级应用开发以及核心基础设施建设之丰富经验。在SOA项目的全生命周期的建设过程中，金蝶Ready SOA支撑平台，能够有效地简化新项目开发和已有项目资产集成的过程，提升效率，降低运维成本。对期望通过建设SOA来改善其业务和IT基础设施的灵活性，提升业务敏捷性的组织而言，金蝶Ready SOA支撑平台是最佳选择。

**6.     结束语**

事实上，SOA这个话题内容是如此丰富，要阐述清楚SOA各方面的内容，哪怕写成一套丛书，也是有许多议题无法讲述清楚的。SOA引起业界的广泛关注，最近金蝶中间件等企业参与支持的中国SOA标准工作组正式启动，可以预期SOA将在中国得到更大的普及，让更多的企业能否利用SOA的强大优势，来增强IT竞争力。

SOA不是过时了，也不是空中楼阁，它需要落地，期望更多的务实的中国企业和业界人士，更多做些脚踏实地的研发，共同为SOA在中国的落地生根而努力。