

【软考达人】

软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题



微信扫一扫，立马获取



6W+ 免费题库



免费备考资料

PC版题库: ruankaodaren.com

高级系统架构设计师下午试题 (I) 模拟13

试题一

1、采用基于Web 2.0的Mashup技术架构Web应用系统，可以从不同的数据源抽取数据，聚合并转换数据，在不同的上下文使用，避免了复杂的集成过程，近年来受到了广泛的关注。Mashup核心活动包括数据输入、数据可视化、调度与监视、剪裁、转换与充实、动作，以及发布与推广等。可以通过使用搜索、语言翻译、工作流支持和其他改进技术来增强基础的Mashup。

请围绕“基于Mashup的Web应用系统设计与应用”论题，依次从以下3个方面进行论述。

1. 概要叙述你参与管理和实施的Web应用系统开发项目及你所承担的主要工作。
2. 简述常见的Mashup数据接口，详细论述你在所参与建设的Web应用系统项目中是如何应用Mashup技术，并分析应用这些技术之后对项目产生了哪些效果(或影响)。
3. 论述在你参与建设的Web应用系统项目实施过程中遇到的，与Mashup技术相关的问题及解决的办法，还有哪些需要进一步改进之处及如何进行改进。

试题二

2、软件的设计模式描述了在特定场景下解决一般设计问题的类和相互通信的对象。它是人们在长期的开发实践中良好经验的结晶，它提供了一个简单、统一的描述方法，使人们可以复用这些软件设计方法和过程管理经验。现在，设计模式已经广泛地应用在软件开发中。

请围绕“基于设计模式的软件架构设计与应用”论题，依次对以下3个方面进行论述。

1. 概要叙述你参与管理和开发的软件工程项目及你在其中所承担的主要工作。
2. 简要介绍设计模式的基本分类，详细说明在你所参与管理和开发的软件工程项目中应用了哪些设计模式、方法，以及选用它们的原因。
3. 分析并讨论使用设计模式的效果，并分析和评价设计模式对软件架构设计的影响。

试题三

阅读以下关于企业应用集成的叙述，根据要求回答问题。

[说明]

某大型企业随着信息化的进展，积累了许多异构的遗产信息系统，这些系统分别采用J2EE、.NET等技术进行开发，分布在不同的地理位置，采用不同的协议进行数据传输。企业要求集成后的系统能够实现功能整合，并在组织现有功能的基础上提供增值服务。FT软件开发公司承接了某大型企业应用系统集成任务。为了按时保质完成任务，选择合适的企业应用集成方法和架构非常重要。项目组在讨论方案时，提出了两种集成思路。

3魏工建议采用传统的应用集成方法，将应用集成分为多个层次，并采用消息代理中间件连接遗产系统。

4陈工建议采用基于SOA的方法进行应用集成，将现有遗产系统采用Web Service的方式进行包装，暴露统一格式的接口，并采用企业服务总线(ESB)进行连接。

项目组仔细分析比较了两种方案的优点和不足后，认为魏工和陈工的建议都合理，但是结合当前项目的实际情况，最后决定采用陈工的建议。

3、[问题1]

请分析比较两种集成方案的优点和不足，将表中的(1)~(6)空缺处的内容填写完整。

应用系统集成方案对比表

考虑因素	陈工推荐的集成方案	陈工建议的集成方案
拟采取的集成方法	涉及不同的集成层次，集成方法复杂多样	(1)
对企业集成需求的符合程度	(2)	(3)
集成系统体系结构	(4)	(5)
集成系统的可扩展性	遗产系统集成方法多样，系统耦合度高，可扩展性较差	(6)

4、[问题2]

企业数据的分布性和异构性是应用系统方便访问企业数据和在企业数据之上提供增值服务的主要障碍。基于SOA的企业集成通过信息服务提供集成数据的能力，针对该企业的集成实际情况，请用300字以内的文字列举3种基于SOA的企业集成中的“数据整合—信息服务”，并给出简要说明。

5、[问题3]

结合你的系统架构设计经验，请用300字以内的文字简要说明ET公司项目组在对该企业构建SOA架构时，除了注意原有系统架构中的集成需求之外，还需要在服务构建时重点关注哪些问题。

试题四

阅读以下关于UML软件系统建模的叙述，根据要求回答问题。

[说明]

车载GPS(Global Position System)终端是置于机动车内的实时定位装置，它的应用对象是需要定位和调度的车辆。车辆可以通过终端与GPS进行实时、准确的定位，并能够通过无线通信网络上报远程的车辆调度中心。中心可以通过终端远程监视车行轨迹，并可在特殊情况下通过终端控制车辆。同时，终端还装备车载电话，可以在出现特殊情况时及时地通知车辆调度中心。

图1所示为车载终端系统的用例图，对于车载GPS终端系统来说，主要的角色有两个，分别为车辆调度中心用户和车载终端用户。

图1

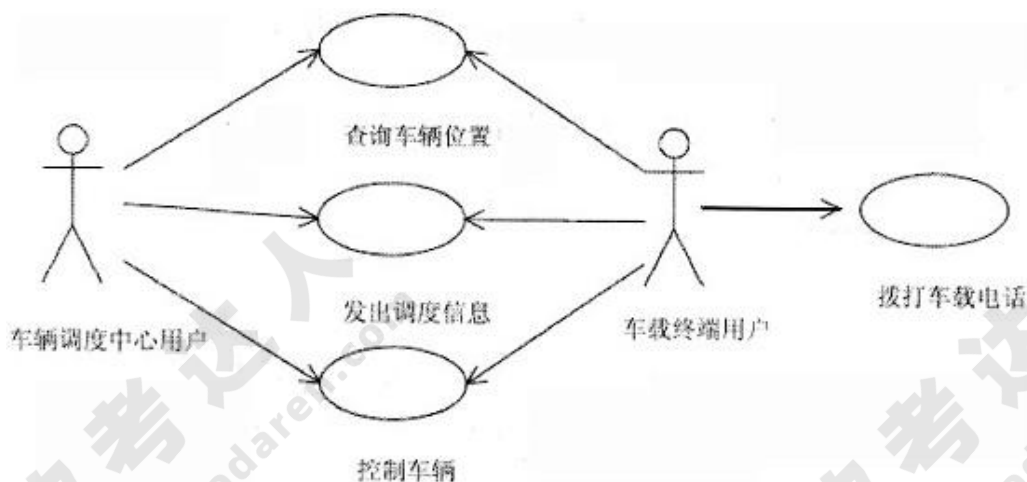
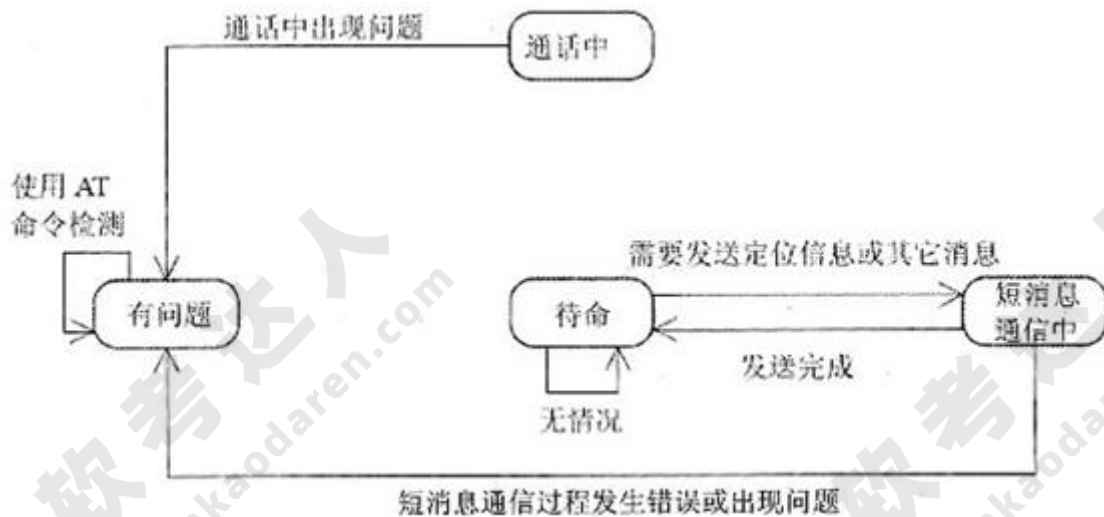


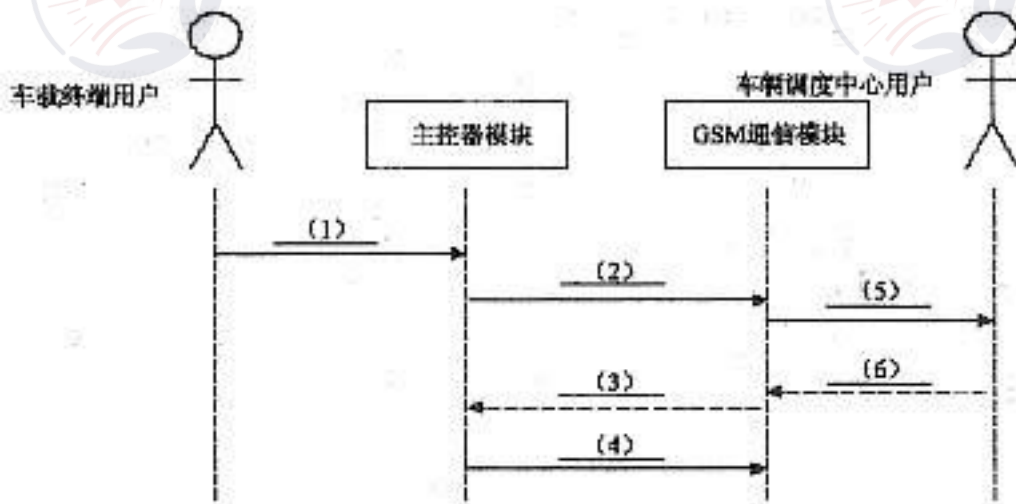
图2所示为车载终端系统中的GSM无线通信模块的部分状态图，用于与调度中心进行联系。GSM模块共有4个状态，分别为通话中、有问题、待命和短消息通信中。

图2



6、[问题1]

车载终端用户在遇到特殊情况下通过车载电话 (或按键) 与调度中心保持通信的处理过程顺序图如图3所示。



处理过程顺序图

结合你的系统架构经验，以及对GPS终端系统的理解，请将下列A~F标记的处理过程填入到图3中的(1)~(6)空缺处，并给出通过车载电话 (或按键) 与调度中心保持通信的正确处理顺序 (请用A~F表达，例如ABCDEF.)。

- A. 语音对话/按下按钮 B. 监听命令
C. 发送信息到通信模块 D. 要求监听
E. 通过GSM发送信息 F. 接收成功要求监听信息

7、[问题2]

建立顶层架构是基于UML对该车载GPS终端系统进行建模的步骤之一。顶层架构的主要目的是为后续的分析 and 设计活动建立一种结构和分划，以便开发人员在不同阶段，以及同一开发阶段的不同开发人员，能够聚集于系统的不同部分。结合你的系统架构经验，请简要说明在该车载GPS终端系统确立顶层架构的过程中需要综合考虑哪些因素？

试题五

阅读下列关于软件产品线方面的叙述，回答问题。

[说明]

A公司是一家中等规模的计算机企业，专门从事网络安全防护软件系统的开发。从最初仅开发基于Windows的个人防火墙产品开始，现在已经延伸到基Linux、Windows系列、Mac操作系统的个人防火墙、企业防火墙、入侵检测系统、病毒扫描系统，以及安全扫描系统等多种产品。公司原来的产品都是一个一个地开发，为每个软件对应地组织一个项目组。

为了适应快速变化的市场，降低开发成本，公司想引入产品线方法。然而，软件产品线方法涉及了一个软件开发企业的多个产品，所以，公司的王总决定在弄清楚以下3个问题之后再做决定：首先是本公司的业务范围是否适合使用产品线方法，其次是如何在原有产品的基础上建立产品线，最后是成功实施产品线的主要因素。

8、[问题1]

结合你的系统架构设计经验，请用200字以内的文字简要说明A公司是否适合采用产品线方法，并说明你的理由。

9、[问题2]

结合你的系统架构设计经验，请用400字以内的文字说明在原有产品的基础上建立软件产品线的方式，并进行简要评价。

10、[问题3]

结合你的系统架构设计经验，请用250字以内的文字说明成功实施产品线的主要因素。

答案：

试题一

1、1. 简要介绍你参与规划和设计的Web应用系统开发项目，尤其着重介绍Web应用系统的应用环境、总体需求(特别是质量属性需求)、Web应用系统的逻辑与物理拓扑结构及采用的技术等内容，简要说明自己在该项目中的角色、所承担的主要任务及开展的主要工作。论文叙述自己参与设计和实施的Web应用系统项目应有一定的规模，自己在该项目中担任的主要工作应有一定的分量。

2. Mashup程序从架构上是由客户机的Web浏览器、Mashup网站和API/内容提供者等3个不同的部分组成，它们在逻辑上和物理上都是相互脱离的(可能由网络和组织边界分隔)。Mashup的主要工作流程是，当移动用户向Mashup网站发起一个请求时，通过GPS系统附带自身的位置信息；Mashup网站接受请求，并将请求分解为对多个服务网站的数据请求调用，并为发起调用进行准备；针对不同服务网站的调用接口，采用不同的调用方式；最终在Mashup站点将请求信息进行数据内容聚合，并将用户所处位置的整合信息返回。目前，经常使用的3种Mashup接口如下。

(1) 聚合内容(Really Simple Syndication, RSS)式的Mashup接口。一种用于对网站内容进行描述和同步的格式，是目前使用最为广泛的Web资源发布方式，可以被称为资源共享模式的延伸。

(2) 表述性状态转移(Representational State Transfer, REST)式的Mashup接口。REST从资源的角度来看待整个网络，分布在各处的资源由统一资源标识符(Unified Resource Identifier, URI)确定，而客户端的应用通过URI来获取资源的表示。

(3) 基于简单对象访问协议(Simple Object Access Protocol, SOAP)的Web服务式Mashup接口。一种基于XML的数据格式定义，用来进行Web服务调用过程中的参数调用和返回。

采用基于Web 2.0的Mashup技术架构Web应用系统具有的优势：①仅需要使用现有Web应用程序(如Google Maps)公开的，基于Web的API(或Web服务)构建应用程序，集成过程相对简单；②直接使用Mashup技术集成两个或者更多的Web API，创建新的特性与功能；③使用Ajax技术调用基于Web的API，浏览器客户端不需要在每次与服务器通信时都重新加载整个页面，动态特性强；④使用SOA的思想集成底层系统，强调功能暴露与服务组合，以服务的形式集成并暴露现有系统的能力等。

在实现Mashup应用时，进行内容聚合的物理位置是一个十分重要的因素。目前很多Mashup站点都选择在客户端机器上进行内容聚合，构成所谓的胖因特网应用程序(Rich Internet Application, RIA)。这种在客户端进行内容聚合的优点主要表现在：①从Mashup服务器存储的角度来说，对服务器所产生的负载较轻，因为数据可以直接从内容提供者那里传送到客户端；②从网络传输的角度来说，在基于Ajax等技术和应用模型的基础上，客户端页面只请求需要更新的内容，而不用刷新整个页面，从而减少网络数据的通信量。

结合项目实践经验，说明你参与管理和开发的项目中，如何理解Mashup、门户、SOA、EAI/EII和SaaS之间的关系；能够全面和准确地描述Mashup各个核心活动(包括数据输入、数据可视化、调度与监视、剪裁、转换与充实、动作，以及发布与推广等)及其具体的实施内容；能够准确地描述如何管理Mashup开发，包括从计划和管理方法到集成、测试和部署的全过程；在优化安全性、隐私、可访问性、有用性和性能，通过使用搜索、语言翻译、工作流支持和其他改进来增强基础的Mashup，执行有效的负载和回归测试，避免造成企业Mashup故障的“反模式”等方面也要有一定的论述。

3. 有具体着眼点地论述在你参与建设的Web应用系统项目实施过程中遇到的，与Mashup技术相关的问题。针对具体的问题你采取了哪些解决技术、方法和措施，以及它们对该工程项目后期的工作产生了哪些积极(或消极)的影响(效果和存在的问题)。论文最后可以进一步讨论你在该工程项目中获得的与Mashup应用相关的体会，以及在今后的工作过程中，如果碰到类似的开发项目你将如何应用这些经验或教训。对需要进一步改进的地方，应有具体的着眼点，不能泛泛而谈。

4. 论文写作过程中值得关注的一些要点如下(全书同)。

(1) 整篇论文要结构合理、切中要害、陈述完整、言简意赅、语言流畅、字迹清楚，切忌对知识点的堆积、长篇大论、言之无物。

(2) 选择自己参与过的工程项目进行分析论述，所述项目切题真实，介绍清楚。

(3) 下午试卷Ⅱ是论述题目，问题中提到的中心内容在题目的说明中都有所涉及。在答题时首先要冷静并认真阅读题目，找出和问题相关的知识点，确定考题的关键考点，这是答题的前提。

(4) 摘要是全文概括，千万不要写成引言。

(5) 围绕论文主题，对所参与的项目进行科学叙述与评价，要有具体的着眼点，不能泛泛而谈，尽可能从字里行间让阅卷者体会到你的实际工作能力、业务水平和项目实践经验。

(6) 在考试过程中应注意技巧，让答题的思路最大限度地符合出题的思路，避免跑题，这样容易得到阅卷老师的共鸣。

(7) 根据考生对所参与的项目中针对本论文主题的相关叙述与评价，可确定他(她)有无参与信息系统项目开发过程的实践经验。

试题二

2、简要介绍你参与规划、设计和实施的大中型信息系统软件工程项目，简要说明自己在该项目中的角色、所承担的主要任务及开展的主要工作。论文叙述自己参与设计和实施的软件工程项目应有一定的规模，自己在该项目中担任的主要工作应有一定的分量。

2. 设计模式描述了在特定场景下解决一般设计问题的类和相互通信的对象。设计模式的核心在于提供了相关问题的解决方案。设计模式的基本组成元素有模式名称、问题或意图、情景、动机、解决方案、示例、结果情景、基本原理、相关模式和已知应用等。

设计模式主要用于得到简洁灵活的系统设计，GoF的书中共有23个设计模式，这些模式可以按两个准则来分类：一是按设计模式的目的划分，可分为创建型、结构型和行为型3种模式；二是按设计模式的范围划分，即根据设计模式是作用于类还是作用于对象来划分，可分为类设计模式和对象设计模式，如表所示。

设计模式空间				
		目的		
		创建型	结构型	行为型
范围	类	Factory Method	Adapter(类)	Interpreter Template Method
	对象	Abstract Factory Builder Prototype Singleton	Adapter(对象) Bridge Comoosite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

创建型模式是对对象实例化过程的抽象，它通过采用抽象类所定义的接口，封装了系统中对象如何创建、组合等信息。该模式允许在系统中创建对象，而不需要在代码中标识特定类的类型，这样

用户就不需要编写大量、复杂的代码来初始化对象。它是通过该类的子类来创建对象的。但是，这可能会限制在系统内创建对象的类型或数目。创建型模式主要有Factory Method(工厂方法)、Abstract Factory(抽象工厂)、Builder(构建器)、Prototype(原型)和Singleton(单独)等模式。

结构型模式主要涉及如何组合已有的类和对象以获得更大的结构，一般借鉴封装、代理和继承等概念将一个或多个类或对象进行组合、封装，以提供统一的外部视图或新的功能。该模式允许在不重写代码或自定义代码的情况下创建系统，从而使系统具有增强的重复使用性和应用性能。该模式控制了应用程序较大部分之间的关系，将以不同的方式影响应用程序。结构型模式主要有Adapter(适配器)、Bridge(桥接)、Composite(组成)、Decorator(装饰)、Facade(外观)、Flyweight(享元)和Proxy(代理)等。

行为型模式主要涉及算法和对象间职责及其提供的服务的分配，它不仅描述对象或类的模式，还描述它们之间的通信模式，特别是描述一组对等的对象怎样相互协作以完成其中任意一个对象都无法单独完成的任务。该模式可以影响一个系统的状态和行为流。通过优化状态和行为流转换和修改的方式，可以简化、优化并且提高应用程序的可维护性。行为型模式主要有Observer(观察者)、Strategy(策略)、Interpreter(解释器)、Template Method(模板方法)、Chain of Responsibility(职责链)、Command(命令)、Iterator(迭代器)、Mediator(中介者)、Memento(备忘录)、State(状态)和Visitor(访问者)等。

能够全面和准确地描述该软件工程项目的应用环境和需求，深入地阐述所选定的设计模式的主要组成元素，以及选用它们的原因，对于项目开发过程中采用的技术和方法要针对该软件工程项目的实际特点，具有一定的广度和深度。

3. 结合软件工程项目实际工作，举例说明设计模式的使用效果，并针对设计模式对软件开发的影响展开分析与讨论，着重阐述所采用的设计模式对该工程项目后期的工作产生了哪些积极(或消极)的影响(效果和存在的问题)。最后，进一步叙述你对设计模式在软件架构设计中的几点应用体会，以及在今后的工作过程中，如果碰到类似的开发项目你将如何应用这些经验或教训。对需要进一步改进的地方，应有具体的着眼点，不能泛泛而谈。

试题三

3、应用集成是指两个或多个应用系统根据业务逻辑的需要而进行的功能之间的相互调用和互操作。应用集成需要在数据集成的基础上完成。应用集成在底层的网络集成和数据集成的基础上实现异构应用系统之间语用层次上的互操作。它们共同构成了实现企业集成化运行最顶层会聚集成所需要的，技术层次上的基础支持。应用集成模式包括集成适配器、集成信使、集成面板和集成代理4种，每种应用集成模式都是对具有业务功能依赖关系的多个应用之间互操作实现方法的总结。在具体应用中，集成模式可能以某种变形(这是一种扩展集成模式的主要方式)的形式出现，这些变形可能不仅仅只是一种模式的实例化，也可能是一种具有广泛适用性的集成方式。

通常，应用集成主要采用点对点的紧耦合方式。这种集成方式虽然不需要对应用系统做较大的改动，但用这种方式集成的系统缺乏必要的柔性，不能适应业务系统快速重构的需求。随着应用软件系统设计和实现过程中标准化程度的不断提高，系统的开放性(可配置性、可扩展性)越来越好，组件化的系统实现及松散耦合的应用集成方式逐渐成为构建企业业务处理系统的主流。其中，松散耦合是实现系统柔性的基础。

在陈工建议的集成方案中，Web Service是一种可以接收从Internet或Intranet上传送的请求、轻量级的独立的通信技术，它允许网络上的所有系统相互间进行交互。Web Service的3个基本技术分别是UDDI、WSDL和SOAP，它们都是以XML为基础定义的。在Web服务体系中，使用WSDL来描述服务，UDDI来发布和查找服务，而SOAP用来执行服务调用。

面向服务架构体系结构(SOA)的几个关键特性是，它是一种粗粒度、松耦合的服务架构，服务之间通过简单、精确定义接口进行通信，不涉及底层编程接口和通信模型。SOA可以根据需求通过网络对松耦合、粗粒度的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交互的人为依赖性。

SOA技术将应用程序的不同功能单元通过一些良好定义的接口联系起来，并且接口采用中立的方式进行定义，它独立于实现服务的硬件平台、操作系统和编程语言。这样可以使得构建的服务以一种

统一和通用的方式进行交互。

企业服务总线(ESB)是对消息中间件的发展,采用了“总线”这样一种模式来管理和简化应用之间的集成拓扑结构,以广为接受的开放标准为基础来支持应用程序之间在消息、事件和服务的级别上动态地互连互通。ESB的基本核心功能包括:①提供位置透明性的消息路由和寻址服务;②提供服务注册和命名的管理功能;③支持多种消息传递范型;④传输协议转换;⑤支持多种数据格式及其相互转换;⑥提供日志和监控功能等。

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,也可能需要很复杂的中介服务支持,包括动态地查找和选择一个服务,以及消息的传递、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,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。

如果企业只是进行小规模或较浅层次的“战术”级应用集成,采用传统的集成方式即可较好的完成任务。但是如果企业想从集成系统的灵活性和可扩展角度考虑,能够持续地集成已有信息系统,并进行“战略”意义上的长远规划,那么面向服务的架构无疑是一种更加优越的方法。采用SOA架构思想进行应用集成,企业现有的应用系统都可以看做广义上的服务提供者,因此(1)空缺处应该填写“将现有系统看做抽象的服务提供者,集成方法统一明确”。

传统的应用集成方式往往采用分层的方式对待整个集成活动,因此在(2)空缺处应该填写“不同层次的集成方法关注点不同,功能组合方面能力较弱”。而采用SOA架构的应用集成方案中,(3)空缺处应该填写“强调功能的暴露与服务的组合,便于提供增值服务”。

对于(4)空缺处,传统的集成方式大多会采用中心辐射型的集成架构,系统的耦合程度较高。而对于(5)空缺处,SOA架构的ESB采用总线结构模式,简化了应用之间的集成拓扑,系统的耦合度较低。

采用SOA架构的应用集成方案通过源白实践的模式,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解决方案是一个松散耦合、灵活的架构。因此在(6)空缺处应该填写“集成系统具有模块化、松耦合的特点,可扩展性较强”。

整理以上分析内容,可得到两种集成方案各自的优点和不足,如下表所示。

应用系统集成方案对比表

考虑因素	手工建设的集成方案	手工建设的集成方案
拟采取的集成方法	涉及不同的集成层次,集成方法复杂多样	将现有系统看做抽象的服务的提供者,集成方法统一明确
对企业集成需求的符合程度	不同层次的集成方法关注点不同,功能组合方面能力较弱	强调功能的暴露与服务的组合,便于提供增值服务
集成系统体系结构	一般为中心辐射型,系统之间的耦合程度较高	基于总线结构的体系结构,系统的耦合度低
集成系统的可扩展性	遗产系统集成方法多样,系统耦合度高,可扩展性较差	集成系统具有模块化、松耦合的特点,可扩展性较强

4、企业数据的分布性和异构性是应用系统方便访问企业数据和在企业数据之上提供增值服务的主要障碍。数据集成和聚合技术在这种背景下诞生,用于提供对分布式数据和异构数据的透明访问。以服务为中心的企业集成通过信息服务提供集成数据的能力,目前主要包括如下集中信息服务。

(1)联邦服务(Federation Service):提供将各种类型的数据聚合的能力,它既支持关系型数据,也支持XML数据、文本数据和内容数据等非关系型数据。同时,所有的数据仍然按照自己本身的方式管理。

(2)复制服务(Replication Service):提供远程数据的本地访问能力,它通过自动的实时复制和数据转换,在本地维护一个数据源的副本。本地数据和数据源在技术实现上可以是独立的。

(3)转换服务(Transformation Service):用于数据源格式到目标格式的转换,可以是批量的或者是基于记录的。

(4) 搜索服务(Search Service)：提供对企业数据的查询和检索服务，既支持数据库等结构化数据，也支持如PDF等非结构化数据。

5、当基于SOA来构建一个企业级的系统架构时，一定要注意对原有系统架构中的集成需求进行细致的分析和整理。而关于系统中最重要元素，也就是SOA系统中服务的构建有两点需要特别注意的地方：①是对于服务粒度的控制；②是对于无状态服务的设计。

SOA系统中服务粒度的控制是一项十分重要的设计任务。通常来说，对于将暴露在整个系统外部的服务推荐使用粗粒度的接口，而相对较细粒度的服务接口通常用于企业系统架构的内部。从技术上讲，粗粒度的服务接口可能是一个特定服务的完整执行，而细粒度的服务接口可能是实现这个粗粒度服务接口的具体的内部操作。虽然细粒度的接口能为服务请求者提供更加细化和更多的灵活性，但同时也意味着引入较难控制的交互模式易变性，也就是说服务的交互模式可能随着不同的服务请求者而不同。如果暴露这些易于变化的服务接口给系统的外部用户，就可能造成外部服务请求者难于支持不断变化的服务提供者所暴露的细粒度服务接口；而粗粒度服务接口保证了服务请求者将以一致的方式使用系统中所暴露出的服务。虽然SOA并不强制要求一定要使用粗粒度的服务接口，但是建议使用它们作为外部集成的接口。通常，可以使用业务流程(BPEL)来创建由细粒度操作组成的业务流程的粗粒度的服务接口。

SOA系统架构中的具体服务应该都是独立的、自包含的请求，在实现这些服务的时候不需要前一个请求的状态，也就是说服务不应该依赖于其他服务的上下文和状态，即SOA架构中的服务应该是无状态的服务。当某一个服务需要依赖时，最好把它定义成具体的业务流程(BPEL)。在服务的具体实现机制上，可以通过使用EJB组件来实现粗粒度的服务。通常，可以利用无状态的Session Bean来实现具体的服务，如果基于Web Service技术，就可以将无状态的Session Bean暴露为外部用户可以调用到的Web服务，也就是把传统的Session Facade模型转化为EJB的Web服务端点。这样，就可以向Web服务客户提供粗粒度的服务。

如果要在J2EE的环境下(基于WebSphere)构建Web服务，Web服务客户可以通过两种方式访问J2EE应用程序。客户可以访问用JAX-RPC API创建的Web服务(使用Servlet来实现)；Web服务客户也可以通过EJB的服务端点接口访问无状态的Session Bean，但Web服务客户不能访问其他类型的企业Bean，如有状态的Session Bean、实体Bean和消息驱动Bean。对于后一种访问方式(公开无状态EJB组件作为Web服务)的优势在于，基于已有的EJB组件，可以利用现有的业务逻辑和流程。在许多企业中，现有的业务逻辑可能已经使用EJB组件编写，通过Web服务公开它可能是实现从外界访问这些服务的最佳选择。EJB端点是一种很好的选择，因为它使业务逻辑和端点位于同一层上。另外，EJB容器会自动提供对并发的支持，作为无状态Session Bean实现的EJB服务端点不必担心多线程访问，因为EJB容器必须串行化对无状态会话Bean任何特定实例的请求。由于EJB容器都会提供对于Security和Transaction的支持，因此Bean的开发人员可以无须编写安全代码及事务处理代码。性能问题对于Web服务来说一直都是一个问题，由于几乎所有EJB容器都提供了对无状态会话Bean群集的支持，以及对无状态Session Bean池与资源管理的支持，因此当负载增加时，可以向集群中增加服务器数目。Web服务请求可以定向到这些不同的服务器，同时由于无状态Session Bean池改进了资源利用和内存管理，使Web服务能够有效地响应多个客户请求。由此可以看到，通过把Web服务模型化为EJB端点，可以使服务具有更强的可伸缩性，并增强了系统整体的可靠性。

试题四

6、图2为车载终端用户通过车载电话(或按键)与调度中心保持通信的处理过程顺序图。其表达的处理过程如下：用户通过按下按钮或语音对话试图连接调度中心，主控器模块接收到连接请求后(或连接建立后)，就发送相应的信息给通信模块；通信模块负责处理并通过GSM发送消息，调度中心接收消息成功后要求监听信息，并把该要求返回给车载系统的GSM通信模块；GSM通信模块把该请求信息(即调度中心的要求监听信息)递交给主模块，主模块把监听命令递交给GSM通信模块，然后可由GSM通信模块递交给调度中心。

7、在初步的业务需求描述已经形成的前提下，基于UML的需求分析大致可分为以下几个步骤。

- 利用用例及用例图表示需求。从业务需求描述出发获取执行者和场景；对场景进行汇总、分类和抽象；形成用例；确定执行者与用例，用例与用例图之间的关系，生成用例图。

- 利用包图及类图表示目标软件系统的总体框架结构。根据领域知识、业务需求描述和既往经验

设计目标软件系统的顶层架构；从业务需求描述中提取“关键概念”，形成领域概念模型；从概念模型和用例出发，研究系统中主要的类之间的关系，生成类图。

上述两个步骤并没有时序关系，它们可以并行展开。其中，顶层架构的主要目的是为后续的分析活动建立一种结构和分划，以便开发人员在不同的开发阶段，以及同一开发阶段的不同开发人员，能够聚焦于系统的不同部分。顶层架构是分析和设计的阶段成果的承载体。随着开发过程的推进，框架中的内容不断丰富和翔实，最终演进为完整的面向对象的软件结构。UML包图是表示顶层架构的适当机制。

建立软件系统顶层架构的基本方法是，结合实际需求，从既往的架构设计经验模型中选取适当者，再进行微调或局部改造。目前有以下几种主要的架构模式。

(1) 流程处理模式。流程处理系统以算法和数据结构为中心，其系统功能由一系列的处理步骤构成，相邻的处理步骤之间以数据流通管道相互连接。该模式仅适合于采用批处理方式的软件系统，不适用于交互式系统。

(2) 客户/服务器模式。客户端负责用户输入和处理结果的呈现，服务器端则负责后台的业务逻辑处理。

(3) 模型—视图—控制器 (MVC) 模式。该模式将整个软件系统划分为模型、视图和控制器3个部分。模型负责维护并保存具有持久性的业务数据，实现业务处理功能，并将业务数据的变化情况及时通知视图；视图负责呈现模型的业务数据，响应模型变化通知，更新呈现形式，并向控制器传递用户的界面动作；控制器负责将用户的界面动作映射为模型中业务处理功能并实际调用之，然后根据模型返回的业务处理结果选择新的视图。MVC模式特别适合于分布式应用软件，尤其是Web应用系统。

(4) 分层模式。该模式将整个软件系统分为若干层次，最顶层直接面向用户提供软件系统的操作界面，其余各层为紧邻其上的层次提供服务。层次划分的主要原则是，较易变化的软件部分 (例如用户界面和与业务逻辑紧密相关的部件) 置于较高层次，较稳定的软件部分 (例如公共的技术服务部件) 则位于较低层次；每一层次尽量只访问其紧邻下层提供的服务，避免越级访问，尤其要避免逆向访问 (上层模块为下层模块提供服务)；在许多情况下，可以将目标软件系统的外部接口置于较低层次，目标软件系统其余部分对外部系统的访问或操作均通过这些外部接口所提供的公共服务来完成。分层模式可以有效地降低软件系统的耦合度，因此其应用十分普遍。

在全面了解软件架构样式的前提下，对于具体的应用需求而言，影响顶层架构选取的主要因素在于系统架构师的经验及他们对每种架构样式与当前软件项目之间匹配程度的判断。事实上，大型软件的顶层架构往往需要复合使用多种架构样式。例如，整个目标软件系统采用分层结构，在系统的不同层次内再分别使用适宜的其他种类的架构模式。

在确立顶层架构的过程中，需要综合考虑以下因素。

(1) 架构中包的数量。原则上，如果每个包中包含的软件元素 (例如类) 的数量过多，应考虑将其进一步细分；如果过少，则说明架构过早地陷入了细节，架构划分返工的可能性较大，同时也不合理地限制了后续分析和设计活动的自由空间。

(2) 架构中包之间的耦合度。包之间的依赖关系和连接关系应尽量简单、稀疏，例如，在分层结构中，通常要求某一层中的软件元素只与同层及相邻下一层的元素之间存在依赖关系。

(3) 软件元素的稳定性。要尽量抽取不稳定的软件元素之中相对稳定的部分，将不稳定的软件元素分类聚集于少数几个包中，以提高软件系统的可维护性。

(4) 软件元素的必然性。可以将可选功能和必须实现的功能分置于架构中不同的包或子包之中。

(5) 作为软件系统运行环境的物理网络拓扑。根据软件元素在分布环境的部署情况，区分顶层架构中的包，可以使包之间的消息传递与物理节点之间的通信相吻合，使后续的分析活动受益于顶层架构中明确定义的通信关系。

(6) 软件元素的安全和保密级别。根据安全访问的权限划分顶层架构中的包或者子包。

(7) 开发团队的技术专长。根据开发人员在问题领域和软件技术领域不同的专长划分顶层架构中的包，使每个包都能分配给最适合的开发人员进行后续的分析、设计、编码和测试等，从而有利于并行开发。

试题五

8、这是一道要求读者根据具体应用环境，分析软件产品线 (Software Product Line) 的适用性的综合理解题。本试题的解答思路如下。

(1) 软件产品线代表了一种强劲的软件开发范例，它可使软件生产在时间、成本和质量方面获得显著改善。它是一个十分适合专业的软件开发组织的软件开发方法，能够有效地提高软件生产率和质量，缩短开发时间，降低总开发成本。同时，它也是一个新兴的、多学科交叉的研究领域，研究内容和范围都相当广泛。

(2) 卡耐基梅隆大学软件工程研究所(CMU/SEI)对产品线 and 软件产品线的定义是，产品线是一个产品集合，这些产品共享一个公共的、可管理的特征集，这个特征集能满足选定的市场或任务领域的特定需求。这些系统遵循一个预描述的方式，在公共的核心资源(Core Assets)基础上开发的。这一定义体现了软件产品线的特征。

(3) 软件产品线开发有：①过程驱动；②特定领域；③技术支持；④以构架为中心等4个基本技术特点。与其他软件开发方法相比，软件开发组织选择软件产品线的宏观上的原因有：对产品线及其实现所需的专家知识领域的清楚界定，对产品线的长期远景进行了策略性规划。

(4) 题干信息说明了A公司的基本情况，其软件产品的特点是，多平台(Linux、Windows系列和Mac操作系统)、同类型(包括个人防火墙、企业防火墙、入侵检测系统、病毒扫描系统和安全扫描系统，这些都属于安全防护类产品)，并且还指出原来产品的开发方法是“一个个开发，每个软件一个项目组”。

(5) 由于该公司的产品都是在同一个领域——防火墙产品等，满足软件产品线建设的基本要求。而且该公司原有产品相对比较成熟，可以在原有产品的基础上建立产品线。因此A公司适合引入产品线方法。

9、这是一道要求读者指出基于现有产品集出发的“软件产品线”构建方法，并能够指出它们的区别与实施要点的综合分析题。本题所涉及的知识点如下。

(1) “软件产品线”方法可以从现有产品集出发，也可以重新构建产品线，但对于已有一些产品体系的公司来说，前一种方法更常用。

(2) 通常，将从现有产品集出发建立软件产品线的建设方式称为演化方式。它可在基于现有产品体系结构设计的基础上，将特定产品的构件逐步转化为产品线的共用构件，从基于产品的方法逐步转化为基于产品线的软件开发。该公司原来产品的开发方法的特点是：“一个个开发，每个软件一个项目组”，而且产品都是在防火墙这个领域，产品相对比较成熟，因此适合采用演化方式建立软件产品线。

(3) 将用新的软件产品线替代现有产品集的建设方式称为革命方式，即基本停止现有产品的开发，所有工作直接针对软件产品线的核心资源开发。遗留系统只有在符合体系结构和构件需求的情况下，才可以和新的构件协作。

(4) 演化方式主要优点是通过对投资回报周期的分解及对现有系统演化的维持使产品线方法的实施风险降到了最小。与革命方式相比，完成核心资源的总周期和总投资较大。

(5) 革命方式将因为重要需求的变化而导致的初始投资报废的风险加大。另外它与演化方式相比，完成核心资源的总周期和总投资较小。由于无法满足产品线方法对软硬件同步的需求，这种方法适用于软硬件结合密切且硬件需求差异大的现有产品集。

10、这是一道要求读者说明成功实施产品线的主要因素简答题。本题所涉及的知识点如下。

(1) 引入一种新的方法论，最重要是能够把握其实施要点，本问题就是要求读者从主要的成功保障因素出发，阐述实施中的关键要点。

(2) 由题干中给出的关键信息“为了适应快速变化的市场，降低开发成本，公司想引入产品线方法”，间接说明了软件产品线方法的主要功能和优点。同时，题干中也指出了在使用产品线方法之前要弄清楚的问题是：“①本公司的业务范围是否适合使用产品线方法；②如何在原有产品的基础上建立产品线；③成功实施产品线的主要因素”。

(3) 成功实施软件产品线的主要因素有：①对该领域的产品开发已具备长期积累的深入经验；②建立一个用于构建产品的好的核心资源库，以支持更高程度的过程规范；③创建稳定、可靠的产品线架构，以及将架构作为产品蓝图的能力；④在软件资源、人员组织和过程等方面能获得良好的管理支持。