高级系统架构设计师下午试题(Ⅱ)模拟19

试题一

论基于Mashup的Web应用系统设计与应用

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

论数据仓库与数据挖掘在企业信息化中的应用

企事业单位的决策越来越需要建立在对历史数据和相关数据的科学分析的理性基础上。数据仓库已经成为数据分析和联机分析处理中日趋重要的平台。然而，数据仓库的设计与实现过程面临许多技术上的挑战，例如，多个异种数据源的集成带来的困难等。

请围绕"数据仓库与数据挖掘在企业信息化中的应用"论题，依次对以下3个方面进行论述。

4、概要叙述你参与管理和开发的管理信息系统项目及你所担任的主要工作。

5、简要讨论你在从事数据仓库的设计时是如何进行规划和分析的，详细描述数据仓库设计、数据集成和测试，以及部署数据仓库的过程。

6、分析并讨论你在数据仓库设计与实现过程中遇到的主要问题及其解决办法，以及你进一步应用数据仓库技术的有关设想。

试题三

论软件架构风格及其应用

软件架构设计的一个核心问题是如何有效地使用重复的体系结构模式，即达到软件体系结构级的软件重用。软件架构风格(Software Architecture style)是描述软件系统组织方式的常用模式，在实践中已经被多次应用。按照Shaw和Garlan的说法，"一种体系结构风格定义了构件类型和连接件类型的词汇表，以及它们如何组合的约束条件"。软件架构风格通常分为数据流(Data Flow)风格、调用/返回(Call/Return)风格、独立构件(Independent Components)风格、虚拟机(Virtual Machines)风格和仓库(Repositories)风格5大类。在实际应用中，随着软件系统规模的扩大和复杂，一个系统往往会同时使用多类体系结构风格，这些风格可以交叉组合，彼此重叠。

请围绕"软件架构风格及其应用"论题，依次从以下3个方面进行论述。

7、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担任的主要工作。

8、请说明以上软件架构风格分类中每一类有哪些经典的软件体系结构设计风格，并就其中至少两类论述其具体的软件架构风格的构件、连接件类型和组合约束要求等结构特征及其应用特点。

9、具体阐述你参与管理和开发的项目中在体系结构设计时选择使用软件架构风格的情况，包括选择的依据、多个风格组合使用的情况和最终实际效果等，还有哪些需要进一步改进之处及如何进行改进。

试题四

论SOA在企业集成架构设计中的应用

企业应用集成(Enterprise Application Integration，EAI)是每个企业都必须面对的实际问题。面向服务的企业应用集成是一种基于面向服务体系结构(Service-Oriented Architecture，SOA.的新型企业应用集成技术，强调将企业和组织内部的资源和业务功能暴露为

服务，实现资源共享和系统之间的互操作性，并支持快速地将新的应用以服务的形式加入到已有的集成环境中，增强企业IT环境的灵活性。

请围绕"SOA在企业集成架构设计中的应用"论题，依次从以下3个方面进行论述。

10、概要叙述你参与管理和实施的企业应用集成项目及你在其中所担任的主要工作。

11、具体论述SOA架构的内容、特点，以及你熟悉的工具和环境对SOA的支持，在应用中重点解决了哪些问题。

12、通过你的切身实践详细论述SOA在企业应用集成中发挥的作用和优势。

答案：

试题一

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、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。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网站接收请求，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，RJA)。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/ElI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。

试题二

4、介绍你在论文中准备列举的、含有数据仓库与数据挖掘技术的管理信息系统的项目背景、项目投资和项目周期等基本情况。在第一段的末尾，尽量用一两句话简要说明在该项目中你所担任的角色和所承担的主要任务。参与设计和实施的信息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。

5、论文的第二部分是体现你在数据仓库与数据挖掘技术应用方面相关理论知识和实践思想的精华所在。在叙述本部分内容时，应注重理论与自身实践经历的结合。论文要点如下。

①数据仓库是一个用于更好地支持企业或组织管理决策的、面向主题的、集成的、相对稳定的、反映历史变化的数据集合。数据仓库用于支持决策，面向分析型数据处理，它不同于企业现有的操作型数据库；此外，数据仓库是对多个异构的数据源有效集成，集成后按照主题进行了重组，并包含历史数据，而且存放在数据仓库中的数据一般不再修改。

数据仓库系统需要高性能数据库服务器、并行数据库技术、数据库互操作技术、决策支持查询优化技术，以及支持多维分析的查询模式等核心技术的支持。

数据挖掘的核心技术包括两类，分别为预言和描述。预言技术用历史预测未来，如分类等：描述技术发现数据中潜在的规律，如关联分析、序列模式、聚集和异常检测等。

数据仓库的设计开发不同于一般意义上的原型法，数据仓库的设计是数据驱动的。这是因为数据仓库是在现存数据库系统基础上进行开发的，它着眼于有效地抽取、综合、集成和挖掘已有数据库的数据资源，服务于企业高层领导管理决策分析的需要。但需要说明的是，数据仓库系统开发是一个经过不断循环、反馈而使系统不断增长与完善的过程。数据仓库的设计与开发大体可以按照以下步骤进行：概念模型设计一技术准备工作一逻辑模型设计一物理模型设计一数据仓库生成一数据仓库运行与维护。注意，这里所讨论的数据仓库系统的开发步骤并不是绝对的顺序。在数据仓库开发的整个过程中，自始至终要求决策人员和开发团队的共同参与和密切协作，要求保持灵活的头脑，不做或尽量少做无效工作或重复工作。

②在详细讨论你如何规划和分析该数据仓库项目的设计时，应注意以下数据仓库系统的设计原则。

·通用性原则。企业各地分支机构在组织构架、业务划分与侧重、其所运行的OLTP系统所依赖的RDBMS，以及数据综合分析与决策支持系统所需要的数据源的类型与格式等不尽相同，这些都在企业数据综合分析与决策支持系统通用化设计的考虑范围之内。

·可扩展性原则。随着业务内容的变化，业务系统的信息范围会发生变化，而对于作为统一信息服务平台应设计性能良好的体系结构，保证系统灵活的功能可扩展性，即在保持系统架构与原业务分析逻辑的前提下，系统能实现简洁的分析主题与功能性扩充。

·技术开放性原则。为保护用户投资，通过透明访问技术，要保证系统能够独立于具体平台工具，对用户形成统一的功能和界面。在工具和平台的选择上给用户提供自由选择的最大余地。

·兼容性原则。企业在信息化建设过程中所积累的信息资源是企业最为宝贵的财富，新建的经营决策分析系统应有效地兼容原系统，尤其兼容原系统的数据资源。

③详细描述你所参与的应用项目的数据仓库设计、数据集成和测试，以及部署数据仓库的过程。能给出相应的系统总体结构框架。例如，系统的实现是否基于元数据的全程管理，是否涉及数据获取层、数据整合层，以及数据展现层的全过程，是否支持分析模型的维护以及数据源的结构性变化，是否提供包括展现报表定制、元数据维护、门户定制及统一安全管理等管理服务。

④分析并讨论你在数据仓库设计与实现过程中遇到过的主要问题及所采取的解决办法。例如，异种数据源的企业应用集成接口实现异种数据源的透明访问，要支持各种关系数据库、平面文件和XML文件等形式。根据企业的分析应用需求，通过设计与实现操作数据存储(ODS)层来达到面向应用的企业级数据视图，系统也支持通过异种数据源的企业应用集成接口直接实施数据仓库的ETL过程。对数据源实现元数据级的管理，数据源的连接类型(ODBC、OLEDB、JDBC、Native)、连接字符串，

以及该数据源的数据结构信息都以技术元数据的形式存储于元数据库中，通过控制台对其进行业务语义定义，使用户对整个企业的信息系统能够较全面的掌控。

6、论文的第三部分应结合自己在项目管理和开发过程中的实际情况，查找曾经遇到过哪些问题，以及针对这些具体问题的应对策略。对需要进一步改进的地方，应有具体的着眼点，不可脱离实际提出过高的要求。论文的结尾部分，建议采用提纲的方式介绍自己在该工程项目中获得的经验体会。论文的最后一句可延伸说明，在今后的工作过程中，如果碰到类似的IT项目，你将如何应用这些经验或教训。

试题三

7、简要介绍你参与规划、设计和实施的大中型软件工程项目的基本情况，尤其有针对性地介绍与软件架构风格或软件重用方面的需求和应用环境，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。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。

8、结合你的项目实践经验，介绍以下软件架构风格方面的知识点。

①Garlan和Shaw将软件架构风格分为5大类

·数据流风格：包括批处理序列架构风格(Batch Sequential)和管道/过滤器架构风格(Pipes/Filters)。

·调用/返回风格：包括主程序/子程序架构风格(Main Program and Subroutine)、数据抽象和面向对象架构风格(Data Abstraction and Object-Oriented)及层次结构架构风格(Hierarchical Layers)。

·独立构件风格：包括进程通信架构风格(Communicating Processes)和事件驱动架构风格(Event systems)。

·虚拟机风格：包括解释器架构风格(Interpreters)和基于规则的系统(Rule-based Systems)架构风格。

·仓库风格：包括数据库架构风格(Databases)和黑板架构风格(Blackboards)。

其他架构风格包括：特定领域软件体系结构(Domain-specific Software Architectures)架构风格、状态转移(State Transition System)架构风格、分布式处理(Distributed Processes)架构风格和REST(REpresentational State Transfer)混合架构风格等。

论文中5类软件架构风格，每一类给出两个具体的体系结构风格即可，给出的体系结构风格属于"其他"中的某一类者也可。

②每一种具体的架构风格的模型

数据流风格包括批处理序列架构风格和管道/"过滤器架构风格。

·批处理序列架构风格：组件为一系列固定顺序的计算单元，组件间只通过数据传递交互。每个处理步骤是一个独立的程序，每一步必须在前一步结束后才能开始，数据必须是完整的，以整体的方式传递。

·管道/过滤器架构风格：每个构件都有一组输入和输出，构件读取输入的数据流，经过内部处理，然后产生输出数据流。这个过程通常通过对输入流的变换及增量计算来完成，包括通过计算和增加信息丰富数据、通过浓缩和删除精炼数据、通过改变记录方式转化数据、递增地转化数据等。在输入被完全消费之前，输出便产生了。这里构件被称为过滤器，连接件就是数据流传输的管道，将一个过滤器的输出传到另一个过滤器的输入。

调用/返回风格包括主程序/子程序架构风格、数据抽象和面向对象架构风格、层次结构架构风格。

·主程序/子程序架构风格：单线程控制，把问题划分为若干处理步骤，构件即为主程序和子程序。子程序通常可合成为模块。过程调用作为交互机制，即充当连接件。调用关系具有层次性，其语义逻辑表现为子程序的正确性，取决于它调用的子程序的正确性。

·数据抽象和面向对象架构风格：这种风格的构件是对象。对象是抽象数据类型的实例。在抽象数据类型中，数据的表示和它们的相应操作被封装起来。对象的行为体现在其接受和请求的动作。连接件即是对象间交互的方式，对象是通过函数和过程的调用来交互的。对象具有封装性，一个对象的改变不会影响其他对象。对象拥有状态和操作，也有责任维护状态。这种结构风格中包含有封装、交

互、多态、集成和重用等特征。

·层次结构架构风格：层次系统组织成一个层次结构。构件在一些层实现了虚拟机。连接件通过决定层间如何交互的协议来定义，拓扑约束包括对相邻层间交互的约束。这个风格的特点是每层为上一层提供服务，使用下一层的服务，只能见到与自己邻接的层。大的问题分解为若干个渐进的小问题，逐步解决，隐藏了很多复杂度。修改一层，最多影响两层，而通常只能影响上层。上层必须知道下层的身份，不能调整层次之间的顺序。独立构件风格包括进程通信架构风格和事件驱动架构风格。

·进程通信架构风格：构件是独立的过程，连接件是消息传递。这种风格的特点是构件通常是命名过程，消息传递的方式可以是点对点、异步和同步方式及远过程调用等。

·事件驱动架构风格：构件不直接调用一个过程，而是触发或广播一个或多个事件。系统中其他构件中的过程在一个或多个事件中注册，当一个事件被触发，系统自动调用在这个事件中注册的所有过程。一个事件的触发就导致了另一个模块中过程的调用。这种风格中的构件是非命名的过程，它们之间交互的连接件往往是以过程之间的隐式调用(Implicit Invocation)来实现的。基于事件的隐式调用风格的主要优点是为软件重用提供了强大的支持，为构件的维护和演化带来了方便；其缺点是构件放弃了对系统计算的控制。虚拟机风格包括解释器架构风格和基于规则的系统。

·解释器架构风格：一个解释器通常包括完成解释工作的解释引擎、一个包含将被解释的代码的存储区、一个记录解释引擎当前工作状态的数据结构，以及一个记录源代码被解释执行的进度的数据结构。具有解释器风格的软件中含有一个虚拟机，可以仿真硬件的执行过程和一些关键应用；其缺点是执行效率较低。

·基于规则的系统：基于规则的系统包括规则集、规则解释器、规则/数据选择器及工作内存。仓库风格包括数据库架构风格和黑板架构风格。

·数据库架构风格：数据库架构是库风格最常见的形式。构件主要有两大类，一个是中央共享数据源，保存当前系统的数据状态；另一个是多个独立处理元素，处理元素对数据元素进行操作。

·黑板架构风格：黑板架构包括知识源、黑板和控制3部分。知识源包括若干独立计算的不同单元，提供解决问题的知识，知识源响应黑板上的变化，也只修改黑板。黑板是一个全局数据库，包含解域的全部状态，是知识源互相作用的唯一媒介。知识源响应是通过黑板状态的变化来控制的。黑板通常应用在对于解决问题没有确定性算法的系统中，例如信号处理、问题规划及编译器优化等软件系统的设计中。

论文中给出每一类软件架构风格中一个具体的体系结构风格即可。

9、结合项目的实际状况，指出在体系结构设计时选择使用软件架构风格的情况，包括选择的依据及多个风格组合使用的情况等，要给出实际的效果及分析。论文最后可以进一步讨论你在该工程项目中获得的、与软件架构风格应用方面相关的体会，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。

试题四

10、简要介绍你参与管理和实施的大中型企业应用集成项目的基本情况，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。参与设计和实施的系统集成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。

11、SOA架构。

·SOA架构的内容：SOA架构的基本元素是服务，SOA指定一组实体(服务提供者、服务消费者、服务注册表、服务条款、服务代理和服务契约)，这些实体详细说明了如何提供和消费服务。SOA中的服务是自包含、无状态的实体，可以由多个组件组成，通过事先定义的接口响应服务请求，服务本身并不依赖其他函数和过程的状态，而用什么技术来实现服务，也不在其定义中加以限制。SOA本质上是将网络、传输协议和安全细节留给特定的实现来处理。这些服务是可互相操作的、独立的、模块化的、位置明确的、松耦合的，以及可发现的。在SOA架构中有3种角色(参与者)，分别是：服务提供者(Service Provider)、服务请求者(Service Consumer)和服务代理者(Service Broker)。服务提供者提供符合契约的服务，并将他们发布到服务代理；服务代理者作为存储库、目录库或票据交换所，产生由服务提供者发布的软件接口；服务请求者(服务使用者，或者终端用户应用程序)发现并调用其他的软件服务来完成业务任务。

·SOA架构的特点：①服务的封装；②服务的重用；③服务的互操作；④服务是自治的功能实体；⑤服务之间的松耦合；⑥服务位置透明性。

·采用的工具或技术：①Web Service；②J2EE；③WebSphere、WebLogic；④.NET；⑤CORBA；⑥DCOM；⑦其他中间件技术。

·以J2EE为例子，阐述它对SOA的支持：J2EE平台通过新的JAX-RPC 1.1 API提供了完整的Web服务支持，这种APl支持基于Servlet和企业Bean的服务端点。JAX-RPC 1.1基于WSDL和SOAP协议提供了与Web服务的互操作性。J2EE 1.4平台也支持Web Services for J2EE规范(JSR 921)，它定义了Web服务的部署需求并利用了JAX-RPC编程模型。

·在应用中重点解决的问题。①服务粒度的控制：SOA系统中服务粒度的控制是一项重要的设计任务。对于暴露在整个系统外部的服务可使用粗粒度的接口，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。虽然细粒度的接口能够为服务请求者提供更加细化和更多的灵活性，但会使交互模式较难控制，服务的一致性较差。②无状态服务的设计：SOA的服务应该是独立的、自包含的，在实现这些服务的时候不需要前一个请求的状态，也就是说不应该依赖于其他服务的上下文和状态。当某一个服务需要依赖时，可以将其定义为具体的业务流程。

注意：论及一项即可，解决的问题是管理或工程实施方面的亦可。

12、结合项目实践经验，阐述SOA在系统集成中发挥的作用和优势所在。

SOA将应用程序的不同功能单元，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。这使得构建在各种这样的系统中的服务可以以一种统一和通用的方式进行交互。

这种具有中立的接口定义(没有强制绑定到特定的实现上)的特征称为服务之间的松耦合。松耦合系统的好处有两点：一是它的灵活性；二是当组成整个应用程序的每个服务的内部结构和实现逐渐地发生改变时，它能够继续存在。

SOA系统原型的一个典型例子是通用对象请求代理体系结构(CORBA)，它已经出现很长时间了，其定义的概念与SOA相似。

目前，SOA已经有所不同了，因为它依赖于一些更新的进展。这些进展是以可扩展标记语言(XML)为基础的。通过使用基于XML的语言(也称为Web服务描述语言)来描述接口，服务已经转到更动态且更灵活的接口系统中，比早期CORBA中的接口描述语言(Interfhce Definition Language，IDL)功能强大得多。