



扫码获取更多备考干货

论软件开发模型及应用

软件开发模型（Software Development Model）是指软件开发全部过程、活动和任务的结构框架。软件开发过程包括需求、设计、编码和测试等阶段，有时也包括维护阶段。软件开发模型能清晰、直观地表达软件开发全过程，明确规定了要完成的主要任务和活动，用来作为软件项目工作的基础。对于不同的软件项目，针对应用需求、项目复杂程度、规模等不同要求，可以采用不同的开发模型，并采用相应的人员组织策略、管理方法、工具和环境。

请围绕“软件开发模型及应用”论题，依次从以下三个方面进行论述。

1. 简要叙述你参与的软件开发项目以及你所承担的主要工作。
2. 列举出几种典型的软件开发模型，并概要论述每种软件开发模型的主要思想和技术特点。
3. 根据你所参与的项目中使用的软件开发模型，具体阐述使用方法和实施效果。

范例

摘要部分：

年月，我所在公司承担了**平台的开发工作，我有幸作为该项目的技术负责人参与整个开发过程，并负责了该项目的需求分析与系统设计的工作。该系统主要包括**等业务功能模块。本文以**平台为例，主要论述了统一过程开发模型在该项目中的具体应用。在初始阶段，我们建立了业务模型并且确立项目的边界；在细化阶段，主要对需求流程进行补充和完善；在构建阶段，主要是开发构件和应用程序功能，并将其集成为产品。整个项目历时 10 个月开发完成，到目前运行稳定。实践证明，这种开发模型有效的提高了开发效率，降低了开发成本和项目风险。

【注意：实际写作中相关项目情况应介绍清楚，摘要字数（包括标点符号）一般写到 300 到 320 字】

正文部分：

目前由于*****更迫切需要解决*****等一系列亟待解决的问题。【项目背景内容可分 2 段写，第 1 段简要说明下项目来龙去脉】

年月，我所在公司承担了**平台的开发工作，我有幸作为该项目的技术负责人，参与整个开

发过程，并负责了该项目的需求分析与系统设计的工作。该系统主要解决**，其中包括***业务功能模块组成，*****。因此，如何能够让项目开发顺利进行，选择一种合适的开发模型组织开发显得至关重要的。【第 2 段对系统整体情况进行细致介绍，项目背景第 1、2 段内容可以写到 400 到 450 字】

当前主流的开发模型主要有瀑布模型，演化模型，螺旋模型，统一过程（UP），敏捷开发模型等。其中，瀑布模型，是一种严格遵循软件生命周期各阶段的固定顺序，一个阶段完成再进入另一个阶段。适合于业务需求比较明确且很少变更的项目。演化模型，是从初始的模型中逐渐演化为最终软件产品，是一种“渐进式”原型法。适合于用户需求不明确的项目，且软件完善周期较长。螺旋模型，是一种结合了瀑布模型和演化模型的优点，最主要的特点在于加入了风险分析。适用于项目规模庞大，复杂且高风险的项目，由于流程复杂，增加了大量成本和时间消耗；敏捷方法，是一种以人为核心、迭代、循序渐进的开发方法。适用于小规模软件或者小团队开发；统一过程（UP），一个通用过程框架，适合于各种应用领域、组织类型、性能水平、项目规模的项目；采用了强大的 UML 建模语言，能够在团队中形成统一规范和模板，同时有很多成熟商业软件提供整个开发周期的相关支持，可以极大的降低开发和管理成本，提高开发效率。因此，我们选择采用统一过程（UP）开发模型。

基于统一过程（UP）的软件项目一般分为初始阶段，细化阶段，构建阶段和交付阶段四个阶段。本文主要着重从前 3 个阶段具体论述统一过程开发模型在该项目中的具体应用。

首先，初始阶段，我们主要为整个系统建立业务模型并确立项目的边界。由于养老机构很多业务流程基本相同，通过调研分析和整理，并利用 UML 工具 PowerDesigner 对系统业务模型进行梳理，识别出与系统交互外部实体，譬如，日常需要使用养老管理平台的老人，护工，主管和管理员，还有需要进行接口对接的医院设备和信息系统和各种医疗厂商的穿戴设备等，并利用 PowerDesinger 建立相关类结构图和数据库结构的概念模型。同时，还有这些实体如何与系统进行交互的各种流程，为了整理的更加准确和清晰，其间，我们与各个养老机构进行了反复沟通和确认，最终定义出尽量符合养老机构业务规范的通用流程，这样既满足了各种机构的业务要求，又能给成长中的养老机构提供更好的借鉴参考的空间，最后利用 PowerDesigner 的建立相关用例图和时序图，并整理相关的设计文档，以备后续系统设计使用。同时，我们对各个机构一些特殊需求进行了梳理评估，对其中暂时无法实现或者实现成本较高的，加入到风险列表，跟用户最后再进行协商和确认，有些进行调整变化，有些直接放到后续项目升级中加以实现。

其次，细化阶段，主要是基于初始阶段确定下来的需求流程进行补充和完善，同时，淘汰业务中风险最高的元素。识别出主要功能和一些重要的非功能需求，做出最佳的决策。我们基于初始阶段的成果，对实体进行进一步的梳理，譬如，剔除无用的实体类，抽象合并和留下真正需要的实体类，同时根据业务需要补充类之间的关系。我们对业务用例也进行进一步梳理，譬如，在养老管理平台中，我认为养老档案的管理是整个平台的主线，贯穿始终，不仅仅在机构版中重要，在未来扩展的社区版中也需要，所以项目在进行概念设计时我们对养老档案数据结构的进行扩展性设计，充分考虑到以后系统的扩展需要。根据系统业务功能需要，我们评估哪些构件需要自研，哪些是需要采用第三方的。譬如，一些业务组件，基于我们原有构件库进行升级和改造，可以直接使用，我们就采取自研，这样既可以提高开发效率，又节省项目成本。而譬如一些基于视频通信的功能，由于自研的成本和本身团队能力限制，所以采用第三方的构件来实现。由于养老管理平台是需要支持 IOS 和 Andriod 平台两种平台的手机和 PAD，所以，为了减少后期开发和维护的成本，对通信接口格式规范进行了统一的设计，同时，也在架构层面上降低项目的开发风

险。

最后，构建阶段，主要是开发所有剩余的构件和应用程序功能，把这些构件集成为产品。由于**平台分为机构版和社区版，同时包括 PC 端，移动手机端和 PAD 端程序设计。我们采用了基于组件的方式进行具体功能的构建，譬如，我们设计开发了通用登录模块，统一的权限管理模块，养老档案管理模块，工作查询模块，日志管理模块等。同时，我把团队又根据业务分为机构业务组和社区业务组，主要进行具体业务功能的开发；组件架构组，主要对通用构件进行封装和集成；移动业务组，主要进行移动端开发。由于采用了组件进行开发，既降低开发的风险和后续维护的成本，同时提高开发效率，项目推进过程中取得了不错的效果。由于涉及到全国几十家养老机构，同时，软件的产品质量是以一个企业生存的基础，我们在测试环境也进行充分的工作，譬如，在开发内部进行单元测试，系统模块进行功能测试；同时，由于涉及浏览器和移动端，在各种版本兼容性上也进行了充分的测试；最后，考虑到用户体验，还对系统进行充分的压力和性能测试。

整个项目历时 10 个月开发完成，最终于**年**月完成交付，到目前运行稳定。通过在生产环境一段时间的使用，用户普遍反馈良好。但同时，也存在一些不足，例如，在细化阶段没有充分考虑到各养老机构关于业务上区别，缺少了灵活的配置策略，在后期开发完成后无法统一调整，只能在临时为存在差异的机构进行代码层面的修改。以后在这种问题上准备结合原型法的一些思想进行调整，尽量减少类似问题的出现。

实践证明，这种开发模型的选择有效降低了开发成本和项目风险，提高了团队开发效率。**平台是一个通用的管理平台，接下来，作为项目的技术负责人，我会总结现阶段的经验教训，在后续系统升级完善中，不断思考和改进统一过程的方法在项目中的应用，充分发挥统一过程开发模型的更大作用，为公司创造效益的同时，也能够为客户开发出更稳定更高效的系统。



扫码获取更多备考干货

论 SOA 技术的应用

面向服务的体系结构（Service-Oriented Architecture, SOA）是一个组件模型，它将应用程序的不同功能单元（称为服务）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一种统一和通用的方式进行交互。

近年来这种技术开始广泛应用于各种开发领域。

请围绕“SOA 技术应用”论题，从以下三个方面进行讨论。

1. 简要叙述你参与分析和设计的信息系统（项目的背景、发起单位、目的、项目周期、交付的产品等），以及你在该项目中的工作。
2. 结合你的项目经历，说明为什么要使用 SOA，详细论述是如何利用 SOA 进行系统构建的。
3. 在对 SOA 的应用过程中，您遇到了哪些问题，是如何解决的，有何改进方法。

范例

摘要部分：

年**月，我公司承接了**项目。该项目是**，为解决*。我作为该项目系统架构师，主要负责系统整体架构设计工作。本文以**项目为例，主要论述了 SOA 技术在该项目的具体应用过程。在服务识别阶段，以客户需求为依据，识别各个系统应该提供的服务接口；在服务开发阶段，各个系统通过平台、构件的使用来实现 Web 服务；在服务组装阶段，以 MDS 为流程服务器，组装各个子系统，实现业务流程串接。通过以上技术的应用，使得项目整体集成度高且运行稳定可靠，最终于***年**月进入试生产阶段，获得客户一致好评。

【注意：实际写作中相关项目情况应介绍清楚，摘要字数（包括标点符号）一般写到 300 到 320 字】

正文部分：

目前由于*****更迫切需要解决*****等一系列亟待解决的问题。【项目背景内容可分 2 段写，第 1 段简要说明下项目来龙去脉】

年**月，我们公司承接了**项目。该项目于年**月，正式进入试生产阶段，历时 10 个多月。我作为该项目系统架构师，主要负责带领设计团队，对软件进行系统划分、服务集成以及架构设计工作。该项目是****，由此可见，这是一个大型多系统的线库一体项目，选择一种合适的系统架构是项目成败的关键。【第 2 段对系统整体情况进行细致介绍，项目背景第 1、2 段内容可以写到 400 到 450 字】

面向服务架构 SOA 是一种应用程序架构，在这种架构中，所有的功能都被拆分为独立的、粗粒度的服务，通过调用这些服务提供的接口，完成业务流程。SOA 的服务类型包括基础设施服务、开发服务、管理服务、业务创新与优化服务等十大类。SOA 的实现过程可以分为五个阶段进行，包括服务识别、服务开发、服务组装、服务测试和服务发布。在该检定线项目中，由于已经存在 IBS、WMS 这些现存的业务子系统，可以完成部分线体作业子流程，这为 SOA 奠定了可行性基础。但其中 IBS 和 MES 是由 C#开发，

WMS 是用 java 开发，不能直接使用 COM/DCOM 或 EJB 将其集成。且由于 MES 并不负责业务流程拆解和整合工作，导致三个子系统是三个信息孤岛，互相无法直接进行通信。

为了解决上述问题，我们最终决定采用 SOA 技术，选择了中立的 WebService 作为分布式应用集成框架，并选用 Json 作为数据交换格式。并开发了 MDS 作为流程服务中心，串接整个系统。以下就以该项目的服务识别、服务开发、服务组装三个步骤分别阐述系统的建设过程。

一、服务识别阶段，以客户需求为依据，识别 MDS、IBS、WMS、MES 应该提供的服务接口。

从客户的第一个需求开始，系统需要接收 MES 下发的生产工单，这个服务应该由 MDS 提供。之后调拨电池出库充电，应该是 WMS 的服务。充电之前充电设备需要读取电池的属性，这属于 WMS 应该提供的查询服务。但考虑到充电设备这种嵌入式设备不应该直接和 WMS 耦合，我们将其设计为 MDS 也提供 485 接口供充电设备调用，MDS 再透传给 WMS。之后的回库再出库等货物调拨接口都应该由 WMS 提供，由 MDS 调用。而产线相关的业务如检定、分拣等服务，则由 IBS 提供给 MDS 调用。另外，由于出入库或检定等工作都耗时很长，这些服务都不可能设计成业务完成时，才返回的同步服务接口。所以我将其设计成仅仅通知的异步接口，在业务完成时，再由 IBS、WMS 反向调用 MDS 提供的服务来通知 MDS。最后，客户的 MES 需要提供一个服务，以便 MDS 上报工单完成。

二、服务开发阶段，主要是改造 IBS、WMS、MES 的系统接口，并另行开发 MDS。

由于 IBS 以前使用的是微软的 WCF 进行通信，并无 WebService 模块，我们在 IBS 中加入了 COM+ 框架，使用其中的 WebService 构建来开发服务接口和调用其他的服务。并改造原有的 XML 格式的协议，使之成为该项目统一使用的 JSON 格式。对于使用 JAVA 开发的 WMS，我们则使用谷歌的 GSON 作为 JSON 的解析包，另外引入了 AXIS2 作为 SOAP 引擎，并结合了大量 Apache 的 WebService 相关构件，搭建起了可以工作的 WebService 收发平台。并在该平台的基础上，开发了服务识别阶段规划的接口。第三方的 MES 开发过程未知，我们只定义了使用 JSON 格式的数据交换格式。MDS 选择使用 Java 语言开发，继承了 WMS 的系统框架，在此基础上进行开发。除了开发了对外服务接口以外，还开发了串接整个业务流程的模型机制，该机制在组装阶段中进行详细描述。

三、服务组装阶段，以 MDS 为流程服务，组装其他子系统。

在我们的设计中，MDS 被设计成整个项目的业务集成中心，它通过依次调用 IBS 和 WMS 提供的服务，实现整个业务流程。MDS 首先将业务流程分解为：出库、充电、入库、静置、出库、检定、发货 7 个生产节点。其中出库节点调用 WMS 出库服务接口，实现电池出库到充电区或检定线。而入库节点则调用 WMS 的入库服务接口，实现已充电电池回库静置。检定节点用于调用 IBS 的服务，实现线体业务操作。静置节点不调用任何其他系统的服务，由 MDS 自己计时，当静置时间到达时，自动触发进入下一个节点业务。最后，发货节点在工单完成时，通过调用 MES 提供的“工单完成通知”服务，通知 MES。通过上述 MDS 架构设计以及服务组装，MDS 串接起来整个系统的业务流程，实现了业务流程集成。

通过以上技术的应用，项目于**年**月正式进入试生产阶段。由于项目集成度高，且运行稳定可靠，获得了客户的广泛好评。

但项目在实施过程中也遇到了一些问题。例如在集成多个系统时，由于过于信赖 WebService 的通信可靠度，没有考虑到通信丢包的场景。在做高负载压力测试时，出现了通信丢包，进一步导致业务流程中断的情况。为了解决该 BUG，且尽量避免业务层关注这些通信异常。在项目后期，我们又从公司的构件库

中选取我们公司设计的“握手补发机制”通信中间件，用以部署在 MDS-MES、MDS-IBS 以及 MDS-WMS 之间。该中间件的设计理念为：所有使用该中间件的命令都是异步命令。且每一条请求命令都有一条握手命令作为回复。如果请求的发送方 10 秒没有收到握手则自动补发请求。请求的接收方负责过滤重复补发的命令，避免二次处理业务。在部署了这些中间件之后，系统再也没有出现过因为丢包导致业务中断的情况。



扫码获取更多备考干货

论软件系统架构风格

系统架构风格（System Architecture Style）是描述某一特定应用领域中系统组织方式的惯用模式。架构风格定义了一个词汇表和一组约束，词汇表中包含一些构件和连接件类型，而这组约束指出系统是如何将这些构件和连接件组合起来的。软件系统架构风格反映了领域中众多软件系统所共有的结构和语义特性，并指导如何将各个模块和子系统有效地组织成一个完整的系统。软件系统架构风格的共有部分可以使得不同系统共享同一个实现代码，系统能够按照常用的、规范化的方式来组织，便于不同设计者很容易地理解系统架构。

请以“软件系统架构风格”论题，依次从以下三个方面进行论述：

1. 概要叙述你参与分析和开发的软件系统开发项目以及你所担任的主要工作。
2. 分析软件系统开发中常用的软件系统架构风格有哪些？详细阐述每种风格的具体含义。
3. 详细说明在你所参与的软件系统开发项目中，采用了哪种软件系统架构风格，具体实施效果如何。

范例

摘要部分：

本人于**年**月参与**项目，该系统为***，在该项目组中我担任系统架构师岗位，主要负责系统整体架构设计。本文以该车联网项目为例，主要讨论了软件架构风格在该项目中的具体应用。底层架构风格我们采用了虚拟机风格中的解释器，因该公交共有几十种不同的数据协议，使用解释器风格可以满足整车数据协议兼容性需求；中间层关于应用层的数据流转采用了独立构件风格中的隐式调用，以减低系统间耦合度、简化软件架构，提高可修改性方面的架构属性；应用系统层采用了 B/S 的架构风格，统一解决公交行业性难题“实施推广难、维护难”问题。最终项目成功上线，获得用户一致好评。

【注意：实际写作中相关项目情况应介绍清楚，摘要字数（包括标点符号）一般写到 300 到 320 字】

正文部分：

随着**，**集团自**年**月起****，规划**。

【项目背景内容可分 2 段写，第 1 段简要说明下项目来龙去脉】

年月我司**委托建设***项目。本项目****，我在项目中担任系统架构师职务，主要负责系统整体架构设计，该系统主要***，主要功能模块包括**。

【第 2 段对系统整体情况进行细致介绍，项目背景第 1、2 段内容可以写到 400 到 450 字】

在架构工作开始阶段，我们便意识到，架构风格是一组设计原则，是能够提供抽象框架模式，可以为我们的项目提供通用解决方案的，这种能够极大提高软件设计的重用的方法加快我们的建设进程，因此在我司总工程师的建议下，我们使用了虚拟机风格、独立构件风格以及 B/S 架构风格这三种较常用风格。虚拟机风格中的解释器架构风格能够提供灵活的解析引擎，这类风格非常适用于复杂流程的处理。独立构件

风格包括进程通讯风格与隐式调用风格，我们为了简化架构复杂度采用了隐式调用风格，通过消息订阅和发布控制系统间信息交互，不仅能减低系统耦合度，而且还提高架构的可修改性。B/S 架构风格是基于浏览器和服务器的软件架构，它主要使用 http 协议进行通信和交互，简化客户端的工作，最终减低了系统推广和维护的难度，以下正文将重点描述架构风格的实施过程和效果。

底层架构我们使用解释器风格来满足整车数据协议兼容性需求。解释器风格是虚拟机风格中的一种，具备良好的灵活性，在本项目中我们的架构设计需要兼容好 86 种不同 can 数据协议，一般来说这种软件编写难度非常高，代码维护难度压力也很大，因此这个解释器的设计任务便很明确了，软件设计需要高度抽象、协议的适配由配置文件来承担。具体的做法如下，我们对各个车厂的 can 数据结构进行了高度抽象，由于 can 数据由很多数据帧组成，每个数据帧容量固定并且标识和数据有明确规定，因此我们将 can 协议中的 ID 和数据进行关系建模，将整体协议标识作为一个根节点，以 canid 作为根节点下的叶子节点，使用 XML 的数据结构映射成了有整车协议链-数据帧-数据字节-数据位这 4 层的数据结构，核心的代码采用 jdom.jar 与 java 的反射机制动态生成 java 对象，搭建一套可以基于可变模板的解释器，协议模板的产生可以由公交公司提供的 excel 协议文档进行转换得到，解释器支持协议模板热部署，这种可以将透传二进制数据直接映射成 java 的可序列化对象，将数据协议的复杂度简化，后期数据协议更改不会对软件产生影响，仅仅更改协议模板文件即可，最终我们使用了 86 个协议描述文件便兼容了这些复杂的 can 数据协议，规避了 can 数据巨大差异带来的技术风险。

中间层我们使用独立构件风格中的隐式调用来简化构件间的交互复杂度，降低系统耦合度。主要的实现手段是，我们采用了一个开源的消息中间件作为连接构件，这个构件是 apache 基金会下的核心开源项目 activemq，它是一款消息服务器，其性能和稳定性久经考验。由上文提到的解释器解析出对象化数据经过 activemq 分发到各个订阅此消息的应用系统，这些应用系统包括运营指挥调度、自动化机护、新能源电池安全监控等，这种多 web 应用的情况非常适合采用消息发布与消息订阅的机制，能够有效解决耦合问题，我们在编码的过程中发现只要采用这种风格的 web 应用，整个迭代过程效率极高，错误率降低，而且我们使用的 spring 框架，消息队列的管理完全基于配置，清晰简单，维护性良好，例如整车安全主题、运营调度主题、机护维修主题等消息队列分类清晰，可以随时修改其结构也能够随时增其他主题的消息队列，不同的 web 系统监听的队列也可以随时变换组合，基于消息中间件的架构设计能够让系统的构件化思路得到良好实施，总体来说这种架构风格带来了非常清晰的数据流转架构，简化了编码难度，减低本项目的二次开发的难度。

应用系统层我们主要采用 B/S 的架构风格，主要用于解决公交推广难、维护难的问题。公交行业有一个明显的特点，公交子公司分布在全市各个地区，路途很远，且都是内网通讯，车联网络也是走的 APN 专网，一般是无法远程支持的，这给我们的系统推广以及后期维护带来了很大难题，我们可以想象如果使用 C/S 架构，更新客户端一旦遇到问题很可能需要全市各个站点跑一遍。这让我们在系统推广和维护方面面临较大压力。我们采用的 B/S 架构风格能够解决这个难题，并充分考量了现在相关技术的成熟度，例如现在的 html5 完全能够实现以前客户端的功能，项目中我们使用了大量的前端缓存技术与 websocket 技术，能够满足公交用户实时性交互等需求。这种风格中页面和逻辑处理存储在 web 服务器上，维护和软件升级只要更新服务器端即可，及时生效，用户体验较好，例如界面上需要优化，改一下 Javascript 脚本或者 CSS 文件就可以马上看到效果了。

项目于**年**月完成验收，这 1 年内共经历了 2 次大批量新购公交车辆接入，这几次接入过程平稳顺利，其中协议解释器软件性能没有出现有问题，消息中间件的性能经过多次调优吞吐量也接近了硬盘 IO 极限，满足当前的消息交互总量，另外由于我们的项目多次在紧急状态下能够快速适应 can 协议变动，得到过业主的邮件表扬。除了业主机房几次突发性的网络故障外，项目至今还未有重大的生产事故，项目组现在留 1 个开发人员和 1 个售后在维护，系统的维护量是可控的，系统运行也比较稳定。

不足之处有两个方面，第一在架构设计的过程中我们忽略 PC 配置，个别 PC 因为需要兼容老的应用软件不允许系统升级，这些电脑系统老旧，其浏览器不支持 html5，导致了系统推广障碍。第二在系统容灾方面还有待改善。针对第一种问题，我们通过技术研讨会说服业主新购 PC，采用两台机器同时使用方式解决。针对第二种问题我方采用了服务器冗余和心跳监测等策略，在一台服务暂停的情况下，另外一台服务接管，以增加可用性。



扫码获取更多备考干货

论高可靠性系统中软件容错技术的应用

容错技术是当前计算机领域研究的热点之一，是提高整个系统可靠性的有效途径，许多重要行业（如航空、航天、电力、银行等）对计算机系统提出了高可靠、高可用、高安全的要求，用于保障系统的连续工作，当硬件或软件发生故障后，计算机系统能快速完成故障的定位与处理，确保系统正常工作。

对于可靠性要求高的系统，在系统设计中应充分考虑系统的容错能力，通常，在硬件配置上，采用了冗余备份的方法，以便在资源上保证系统的可靠性。在软件设计上，主要考虑对错误（故障）的过滤、定位和处理，软件的容错算法是软件系统需要解决的关键技术，也是充分发挥硬件资源效率，提高系统可靠性的关键。

请围绕“高可靠性系统中软件容错技术的应用”论题，依次从以下三个方面进行论述。

1. 简述你参与设计和开发的、与容错相关的软件项目以及你所承担的主要工作。
2. 具体论述你在设计软件时，如何考虑容错问题，采用了哪几种容错技术和方法。
3. 分析你所采用的容错方法是否达到系统的可靠性和实时性要求。

范例

摘要部分：

年月，我所在**公司组织研发**平台，该平台主要****，我在该项目中担任系统架构师，主要负责系统整体的架构设计工作。本文以该平台为例，主要论述了软件容错技术和方法在该系统中的具体应用。通过采用以集群化的形式进行应用部署；通过主备形式的数据库部署进行软件容错；通过程序设计方面进行软件的容错与避错。事实证明，以上软件容错技术的应用对于系统的可用性、安全性和可扩展性方面起到了很好的效果，满足了该系统的性能需求，并且该平台从上线到目前一直稳定运行，得到了各业务线负责人和公司高层领导的认可和赞赏。

【注意：实际写作中相关项目情况应介绍清楚，摘要字数（包括标点符号）一般写到 300 到 320 字】

正文部分：

由于公司快速发展**，各业务线迫切需要****，用于***。【项目背景内容可分 2 段写，第 1 段简要说明下项目来龙去脉】

年月，我所在公司组织研发**平台，我作为该平台的架构设计师，主要负责该系统整体的架构设计工作。平台主要划分****等服务。其中****。对于**相关的服务必须要能满足 7*24 小时的稳定运行，对软件的可靠性要求非常高，这就要求在软件架构设计中必须考虑到软件容错技术的应用。【第 2 段对系统整体情况进行细致介绍，项目背景第 1、2 段内容可以写到 400 到 450 字】

提高软件系统可靠性技术主要分为容错技术和避错技术，容错技术的主要方式为冗余，冗余又分为结构冗余、时间冗余、信息冗余，冗余附加。结构冗余又分为静态冗余、动态冗余和混合冗余。软件容错技术主要有 N 版本程序设计，恢复块方法，和防卫式程序设计。结合互联网软件的性质，我主要采用了集群技术、数据库主从方式、和程序设计方面来进行软件的容错与避错处理。下面就从以上三方面详细讨论我所采用的容错技术和方法。

1、通过集群技术来容错

平台中的各服务如果在运行时部署在一台服务器上，那么当服务器发生故障，整个平台将不能再提供任何服务。所以一般非常小规模的应用才会采取这样的部署方式，像互联网应用这样的支付平台来说必须采用多机同时部署的方式，防止单台服务器宕机或者服务进程 Crash 导致整个平台不能提供服务的问题。通过多机同时部署，当一台服务出现问题时，可以很容易替换一台新服务器进行重新部署生效，通过服务客户端的软负载均衡功能，可动态剔除不可用服务机器，动态发现新加入集群的服务机器，使平台在出现故障时可平滑过渡，达到容错的目的。另外平台中各服务当首次获取到的不易变的静态数据会将其存入非本地缓存中，例如采用了 redis cluster 技术，可以很好的保证写入缓存中的数据获取的高可靠性，恰当使用缓存不但会提升平台性能，同时还可以起到容错的效果，例如当某个服务所依赖的后端存储发生了短时的故障或者网络抖动，在这个时候大量的并发请求发现存储获取失败直接从缓存中获取将数据返回给调用方，起到了很好的容错效果。

2、通过数据库主从部署方式来容错

对于该平台来说，所依赖的后端数据库存储的稳定性是非常非常重要的，所有的订单，交易、账户等等数据将直接存储到数据库中。如果数据库在运行过程中频繁宕机，那么带来的问题将是不能容忍的，因为会造成订单丢失，交易丢失，账户余额出错等等问题，所以在这里就要求数据库存储要具有非常高的可靠性，同时具有很强的容错性，在这里我主要采用了数据库的部署结构为主从式方式，要求部署在不同服务器上，在不出问题的情况下对于一些时效性要求不是很高的场合从库可以负责承担一部分读流量，当主库发生读写问题时，可快速由其他的从库升级为主库，继续服务，达到容错的效果。在这里我还采用了要求数据库宕机加报警的方式来防止宕机的主从数据库实例过多，导致在并发高的情况下没有可用的从库升级为主库提供服务，通过这样的方式也提高了整个平台的高可靠性。数据库的数据文件存储这里我也要求采用了 RAID 磁盘容错技术来防止单块磁盘损坏导致的数据文件丢失问题。

3、通过程序设计方面进行软件的容错与避错。

根据以往的架构经验，那么系统的不可靠大部分是由于程序内部的设计或者网络请求参数的配置或者连接池参数的配置不当所导致的。所以通过程序设计方面进行软件的容错是非常重要的。在程序设计方面的容错用的最普遍的就是防卫式程序设计，例如平台中的一键支付聚合服务，当在支付的过程中调用账户服务来进行账户金额扣减的时候，势必会调用账户服务传递请求对象来处理，如果说账户服务在被调用的这一刻网络抖动或者丢包的情况下，这个时候一键支付聚合服务必然会收到抛出的错误信息，如果没有通过恰当的容错处理，那么这次一键支付必然会给用户显示支付失败，不太友好，在这里我采用了 TRY CATCH 机制加 3 次重试的容错处理机制，就解决了该次支付因网络抖动导致的支付失败问题。平台采用的是微服务的架构风格，那么在服务之间的通讯过程中涉及到数据的传递，这里我采用了在数据传输协议的头部加 CRC 码来做到对错误数据处理的避错。

通过采用了以上容错技术的方法和措施后，平台最终于**年**月成功上线，各服务运行状态良好，通过日志分析来看，支付成功率很高，得到了公司高层领导和各业务线负责人的赞赏和认可。但在在我看来还有如下两点不足。1.各服务间调用事物一致性问题的容错处理。针对该问题，目前只能保证事物的最终一致性，因为根据 CAP 理论，要解决该问题确实存在一定的难度，后面我准备研究下 TCC 事物处理方式尤其适合支付平台场景，争取在不损失性能的前提下最大限度解决分布式事物的一致性问题。2.目前所采用

的最大努力推送型事物服务依赖 MQ 重复消息的问题。针对该问题我采用加了一张消息处理表的方式来解
决，当收到消息的时候，先查询该条消息是否已经处理，如果没有处理直接进行处理并将其进行记录，防
止重复处理导致支付数据出错。

软件容错技术对软件的稳定性起着至关重要的作用，尤其是针对互联网性质的软件并发高存在流量峰
值等问题，软件容错技术的应用的重要性就不言而喻了。经过这次我所采用的软件容错技术的方法和措施
的实施效果后，使我也看到了自己身上的不足之处，我会在今后的架构设计过程中，不断更新自己的知
识，不断完善自己的架构设计领域，设计出更好的软件架构，更好的支撑业务平台的运行，提高公司的竞
争力，为公司为社会尽一份绵薄之力。



扫码获取更多备考干货

论信息系统的安全性及保密性设计

在企业信息化推进的过程中，需要建设许多的信息系统，这些系统能够实现高效率、低成本的运行，为企业提升竞争力。但在设计和实现这些信息系统时，除了针对具体业务需求进行详细的分析，保证满足具体的业务需求之外，还要加强信息系统安全方面的考虑。因为如果一个系统的安全措施没有做好，那么系统功能越强大，系统出安全事故时的危害与损失也就越大。

请围绕“信息系统的安全性及保密性”论题，依次从以下三个方面进行论述：

- 1.概要叙述你参与分析设计的信息系统及你所担任的主要工作。
- 2.深入讨论作者参与建设的信息系统中，面临的安全及保密性问题，以及解决该问题采用的技术方案。
- 3.经过系统运行实践，客观的评价你的技术方案，并指出不足，以及解决方案。

范例

摘要部分：

年月，我所在公司承接了**平台，该系统***业务功能模块，我在项目中担任系统架构设计师，主要负责整个系统的架构设计。本文以该平台为例，主要论述了针对该系统的安全及保密性问题，我们所采用的技术手段及解决方案。在网络硬件层通过设置硬件防火墙，来保证内部网络安全性；在数据层通过设置数据加密及容灾备份机制，以此保证数据抗突发风险的安全；在应用层统一采用 RBAC 的授权机制，来保证整个系统认证环节的安全性。事实证明，以上技术方案的实现对整个系统的安全和保密性方面起到了很好的效果，最终项目顺利上线，运行稳定，受到广大用户的一致好评。

【注意：实际写作中相关项目情况应介绍清楚，摘要字数（包括标点符号）一般写到 300 到 320 字】

正文部分：

目前由于*****更迫切需要解决*****等一系列亟待解决的问题。【项目背景内容可分 2 段写，第 1 段简要说明下项目来龙去脉】

年月，我所在公司承担了**平台的研发，我在该项目中担任系统架构设计师，主要负责应用系统的软件架构设计。该系统****，其中包括***业务功能模块，****。【第 2 段对系统整体情况进行细致介绍，项目背景第 1、2 段内容可以写到 400 到 450 字】

经过前期对全国几十家养老机构和相关合作的医疗单位的调研分析，结合原有的经验，对整个系统业务进行详细规划和相关设备的初步选型，即我们内部所说的“两版三端”的方案，机构养老和社区养老两个子系统，同时能够在 PC 端，PAD 端，手机端三端进行呈现和交互。整个养老管理信息平台基于 B/A/S 模式的分层架构风格设计。同时，结合当前先进的基于租户理念搭建的一个养老云平台（SAAS）。当前信息安全问题严重，而养老云平台是一个养老大型综合管理平台，确保系统的安全性和保密性显得尤为重要，让使用我们平台的每一家机构、社区、老人及相关亲人都能够安心和放心。这样，只有最终得到了用户的信赖，系统的前景才会发展的更好。所以，我在系统的安全性和保密性方面做了充分的设计。

1、网络硬件层安全方案

首先，因为网络和硬件是整个系统运行的基础，也是很多外部攻击的主要途径，所以，在这块需要进行合理规划。我们对网络拓扑结构划分为外部网络，内部网络和 DMZ 三部分。在外部网络和内部网络之间设置了硬件防火墙，主要是防止外部的恶意攻击。在防火墙后，为了加强对病毒入侵的防范，又设置了硬件的防毒墙，实时保证最新的病毒特征库保证有效的病毒工具。为了防止外部客户通过 DNS 服务直接访问到我们服务器的目标地址，在内部网络中又增加了反向代理服务器，对外只暴露代理服务器的虚拟 IP，更好的保护了应用服务器被攻击的可能性。应用服务器和数据库服务器做了物理隔离，内部人员也无法通过 IP 直接访问，只能通过跳板机由服务器管理员进行操作，阻断了外界通过内部客户端代理的访问对服务器造成攻击。对于一些核心的 FTP 服务器，DNS 服务器，Web 服务器都规划到 DMZ 中。

2、数据层安全方案

其次，数据是整个系统平台的核心，其中涉及大量个人隐私数据和重要信息资产，安全不容忽视。主要从数据存储，数据访问和数据容灾几方面进行了规划。由于系统主要存放的养老档案数据，其中涉及很多个人隐私部分数据，为了防止泄露，我们在存储时进行了加密处理。虽然在处理中损失些性能，但是，提高了数据的安全性和保密性。我们采购商业数据库 Oracle 作为后台存储，由于其极佳的商业口碑，对数据存储的安全提供强大保证。为了防止数据信息的泄露，加强了访问权限的限制，在数据库的访问权限上，都根据不同角色进行了详细划分，包括对数据库、表、索、记录等的增、删、改、查进行了限制，严格保证非法用户对数据库恶意破坏。同时，在数据库本身，也制定了基于全量，增量，差量的按周备份计划，保证每时每刻的数据都可以及时恢复。在物理存储上也做了本地多机房的容灾备份，充分保证数据抗突发风险的安全。

3、应用层安全方案

最后，由于我们系统架构采用的是分层架构风格，同时，又是采用“三端”的形式，即手机端，PAD 端，还有 PC 端，所以，在用户认证，接口传输等方面做了充分设计。由于系统涉及到养老机构工作人员，老人及家属等，角色众多，统一采用 RBAC 的授权机制，既增强系统安全性满足业务需求，同时，减轻由于权限信息维护负担。在登录界面设计上，不仅仅需要提供用户名+口令，同时设置了校验码，而且同时还增加了动态口令，通过短信接收验证码，以此来防止暴露非法破解登录，保证整个系统认证环节的安全性。密码提交和存储过程中，一律通过 MD5 +salt 加密，即使密码泄露也不会被解密。接口设计是整个系统通信的安全保障，为了防止外部人员恶意调用和窃取信息，我们采用了令牌机制保障每次通讯的安全性，即在登录后，通过加密传输的用户信息和时间戳获取到 token 令牌，然后在后续每次通讯传输过程服务端对 token 进行校验，充分保障了接口调用的安全。

整个项目历时 10 个月开发完成，最终于**年**月完成交付，到目前运行稳定。通过用户一段时间的使用反馈，不论是 PC，还是 PAD 和手机端，在任何环境下使用系统，基本没有信息泄露的情况发生。但是，一些用户反馈，手机登录需要时间较长，查询和保存老人档案信息有时候需要等待，经过分析，登录时间较长，由于登录需要对客户端对用户和密码信息进行客户端加密处理，加密算法耗费了移动端 CPU 资源，同时在服务端也对数据进行加密处理返回客户端 token 也消耗了不少时间。老人档案也是由于敏感信息加密的问题，耗费了 CPU 的处理时间，我们后面通过选择效率更高的加密算法改善了这个问题。

实践证明，项目能够顺利上线，并运行稳定，使用良好，与系统在安全性和保密性方面设计密不可分

分。系统安全是一个永久的话题，我们对系统安全性的完善是一个持续的过程，我们接下来，还会继续不断完善系统安全方面的设计缺陷和不足，使整个平台更加安全好用。



扫码获取更多备考干货