

## 论基于架构的软件设计方法（ABSD）及应用

基于架构的软件设计（Architecture-Based Software Design, ABSD）方法以构成软件架构的商业、质量和功能需求等要素来驱动整个软件开发过程。ABSD 是一个自顶向下，递归细化的软件开发方法，它以软件系统功能的分解为基础，通过选择架构风格实现质量和商业需求，并强调在架构设计过程中使用软件架构模板。采用 ABSD 方法，设计活动可以从项目总体功能框架明确后就开始，因此该方法特别适用于开发一些不能预先决定所有需求的软件系统，如软件产品线系统或长生命周期系统等，也可为需求不能在短时间内明确的软件项目提供指导。

请围绕“基于架构的软件开发方法及应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与开发的、采用 ABSD 方法的软件项目以及你在其中所承担的主要工作。
2. 结合项目实际，详细说明采用 ABSD 方法进行软件开发时，需要经历哪些开发阶段？每个阶段包括哪些主要活动？
3. 阐述你在软件开发的过程中都遇到了哪些实际问题及解决方法。

### 范例

摘要部分：

\*\*年\*\*月，某集团公司开始了\*\*系统的开发，该系统主要实现\*\*等功能。主要包括\*\*等主要功能模块。我在该系统中担任系统架构师，主要负责该系统的架构设计工作。本文以该系统为例，主要论述了基于架构的软件设计方法在该系统中的具体应用。在架构需求阶段，以构件图、包图、类图和对象图描述系统结构，为系统整体的上层结构进行建模；在架构复审阶段，采用架构权衡分析法来对架构设计方案进行复审，以应对评估质量属性的满足程度问题；在架构演化阶段，采用需求变动管理工具对收集的需求进行归类，以应对需求变动管理问题。经过多轮迭代演化，本系统最终顺利上线，并运行稳定，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：

某集团公司\*\*为了应对\*\*\*，因此\*\*研发\*\*系统。【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】

该项目在\*\*年\*\*月正式启动，旨在通过\*\*，优化\*\*。我在该项目中担任架构师，并负责整体系统的架构设计工作。本系统主要由\*\*等主要模块构成。此外，该系统能\*\*，通过如\*\*得到\*\*。\*\*从而最终实现加强集团在银行业的竞争力的目的。【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】

基于架构的软件设计方法，简称 ABSD 方法，主要包含了架构需求、架构设计、架构文档化、架构复审、架构实现和架构演化六个阶段。其中，架构需求主要包含了需求获取、标识构件、需求评审等活动。架构设计，主要包括提出架构模型、映射构件、分析构件相互作用、产生架构、设计评审等活动。架构文档化，用于把架构设计的成果进行分析与整理并进行文档化。架构复审，用于对架构设计进行复审并

标识其中潜在的风险、缺陷和错误。架构实现，包括了架构分析与设计、构件实现、构件组装、系统测试等活动。架构演化，主要活动包括需求变化归类、架构演化计划、构件变动、更新构件的相互作用、构件组装与测试、技术评审等活动。

考虑项目建立初期的需求不稳定，且后续的需要应对大量的新需要，我们决定使用 ABSD 方法来对系统进行构建与演进。下面将着重描述 ABSD 方法在本项目应用过程中所遇到的问题和采取的应对方案。

### 一、架构需求阶段，以构件图、包图、类图和对象图描述系统结构

为了应对架构需求阶段描述系统架构结构的问题，我们在标识构件活动中，使用构件图、类图等来对系统的整体结构进行建模，实现了通过标识构件活动描述架构结构的目的。首先，我们根据获取的需求以类图和包图的形式为系统的下层结构进行建模。若某个构件的业务构成较为复杂，则以对象图进行辅助说明。然后，我们对这些类图和包图进行分组，从而拟定构件的边界。接着，我们安排项目负责人基于上一步得出的类图和包图的分组，并以需求获取活动得到的会议记录和用例图作为辅助，使用构件图设计出需要的构件，再使用这些构件为系统整体的上层结构进行建模。最后，我们使用《需求构件关系表》标识并记录构件与需求之间的关系，用以指导后续的工作。通过这个方案，我们在标识构件活动中，使用构件图、包图、类图等结构更为清晰的视图为系统的整体结构建立了一个更为稳定的基线，从而达到了提升需求到系统设计的转化效率的效果。

### 二、架构复审阶段，使用 ATAM 对架构的质量属性进行复审

为了应对对架构设计方案的质量属性满足程度进行评估的问题，我们在架构复审阶段，使用架构权衡分析法（简称 ATAM）来对文档化后的设计方案进行复审，从而实现了确定非功能性需求在本系统的架构设计中是否得到满足的效果。首先，考虑到参与 ATAM 会议的与会人员来自于不同领域的部门，为了便于会议的进行，我们在 ATAM 的描述和介绍阶段中，为相关人员提供 ATAM 评估方法、系统业务动机和架构整体设计作为知识基础。然后，我们在 ATAM 的调查和分析阶段中，使用基于质量属性的评估方法（如建立质量效用树等），对评估系统架构满足非功能需求的情况进行评估。接着，我们在 ATAM 的测试阶段中，基于场景的验证方案来对上一步评估结果进行验证，从而通过质量场景和场景的优先级对架构方案进行调整。通过这个方案，我们在架构复审阶段中，使用 ATAM 来为架构设计方案的质量属性进行评估，从而确定非功能需求在架构设计中的满足程度并通过质量场景调整架构方案的目的。

### 三、架构演化阶段，对需求变化进行归类

为了应对架构演化阶段的需求变动管理问题，我们在需求变化归类活动中，使用数个工具对收集的需求变动进行整理与分类，实现了对需求变动与构件之间的关系进行管理的目的。首先，我们把新的需求变动登记到《需求管理列表》中，并通过该表获得新的需求变动与旧需求的对应关系。而当某个新的需求变动和旧需求建立起关系时，我们可以通过上次迭代生成的《需求构件关系表》和《开发工作登记表》，推导出新需求变动与系统已有构件及这些构件的开发人员的对应关系。然后，把新需求与系统中已有构件和候选处理人员的对应关系回填到《需求功能关系表》中。而当某个新需求的需求点和旧需求无法建立起关系时，我们直接把新需求写到《需求构件关系表》中并标记为新增。然后我们再通过召开相关干系人的会议，议定每个变动的实现优先级和跟进人员。通过这个方案，我们不但达到了对需求变动与构件之间的关系进行管理的目的，还能划定较适合处理这些需求变动的候选人名单以指导后续工作之用。

由于使用了以上 ABSD 方法来对项目进行开发与演进，该系统改进工作十分顺利，经过多轮迭代演

化，本系统最终于\*\*年\*\*月顺利上线，并运行稳定，用户普遍反馈功能可以满足工作需要且使用体验良好。实践证明 ABSD 方法适用于该项目的变动管理。

然而，我们在应用 ABSD 方法的过程中还存在着一些不足，如我们发现因人手更替，导致一些构件的跟进人员产生变动，这将会产生额外的学习成本。因此，我们决定在后续的演化中，每个构件都需要维护一份更详尽易懂的文档，并在技术评审阶段对本轮迭代中的这些文档进行审核来缓解这个问题。通过本次项目，我认识到紧贴业务需求、数据和开发小组的构成情况来选择具体的软件设计方法将会大大地影响一个项目的演进能力。而在选定软件设计方法后并不意味着结束，通过对方法的细节进行不断改进，将是项目成功的关键。



## 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）是每个企业都必须要面对的实际问题。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。

- 1、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。
- 2、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。
- 3、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。

### 范例

#### 摘要部分：

\*\*年\*\*月，我参与了\*\*系统的开发工作，该系统旨在\*\*等问题。我在该项目中担任技术负责人，主要负责该系统的技术方案制定及技术指导。本文以\*\*系统为例，主要论述了 SOA 在企业集成架构设计中的具体应用。服务提供者主要完成服务的设计、描述、定义和发布等相关工作；服务注册中心保证该系统各个模块、服务的相互独立性与松耦合；服务请求者通过 WebService 技术调用服务。实践证明，通过以上技术的应用有效实现了资源共享和系统间的互操作性，提高了系统的灵活性，最终系统顺利上线，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：

由于\*\*因而要求\*\*需要\*\*，因此，\*\*系统应运而生。【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，我参与了\*\*系统的开发工作，担任该项目的技术负责人，主要负责该系统的技术方案制定及技术指导。该系统的主要功能模块包括\*\*等。旨在帮助\*\*，解决\*\*等问题，通过\*\*，及时生成\*\*信息，提高\*\*等各项职能。因此，要满足该系统的需求，选择一种合适的架构技术至关重要。【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】

SOA 是一种应用程序架构，在这种架构中，所有功能都定义为独立的服务，服务之间通过交互和协调完成业务的整体逻辑。SOA 指定了一组实体，包括服务提供者、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。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。这些服务是自我包含的、无状态的实体，可以由多个组件组成。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事先定义的标准化接口，使得服务可以提供给在任何异构平台和任何用户接口使用。这种松散耦合和跨技术实现，使各服务在交互过程中无需考虑双方的内部实现细节、实现技术、以及部署在什么平台上，服务消费者只需要提出服务请求，就可以发现并调用其他

的软件服务得到答案。SOA 作为一种粗粒度、松耦合的架构，具有松散耦合、粗粒度服务、标准化的接口、位置和传输协议透明、服务的封装和重用、服务的互操作等几个特点。

该系统要求开发周期短，系统灵活性高等，结合 SOA 的特点，我们最终采用了面向服务的、基于 SOA 的企业应用集成。下面具体论述其应用过程。

## 1、服务提供者

服务提供者主要完成服务的设计、描述、定义和发布等相关工作。经过对水利行业施工工程及施工工艺的深入研究，通过查阅《胶凝砂砾石坝施工指南》等相关资料，根据企业应用集成的要求，对胶凝砂砾石坝施工质量监控系统的业务流程进行梳理；综合考虑服务粗粒度、松耦合、自包含和模块化等特点进行服务的设计。为了避免服务通信期间，信息量过大，服务之间交互过于频繁，尽量的减少了服务的数量。同时，为了保证服务自身功能的完整性，尽可能的减少服务与系统之间的通信，在胶凝砂砾石坝施工质量监控系统的分析与开发过程中，先行设计，提取出了两个必要，急需的服务便于日后集成使用，其中包括拌合监控中标准拌合比对比服务和碾压监控中的碾压轨迹生成服务。在标准拌合比对比服务中主要实现针对现有拌合配比与标准拌合比的对比，以判断现有拌合配比是否符合标准的工作。由于胶凝砂砾石坝就地、就近取材的特性，因此在不同的水利施工工地所使用的采料也不尽相同，标准拌合比对比服务预留了标准拌合比的输入接口，以适应不同的需求。在碾压轨迹生成服务中主要实现读取定位信息绘制碾压轨迹，以监控是否存在漏碾和欠碾的情况。由于受到胶凝砂砾石坝选址和机密程度的限制，定位信息可以选择 GPS 或者超宽带技术，但是两种定位的方式的数据格式并不相同，因此碾压轨迹生成服务的开发中预留了两种数据格式的接口来读取定位信息。待完成服务设计之后，服务提供者采用 WSDL 进行服务描述，然后再利用 UDDI 技术将这些服务信息发布至服务注册中心，公布查找和定位服务的方法。

## 2、服务注册中心和服务请求者

在胶凝砂砾石坝施工质量监控系统采用了服务注册中心。服务注册中心不是一个必选角色，但是为了保证该系统各个模块、服务的相互独立性与松耦合，在该系统中依然保留了服务注册中心。同时，服务注册中心的存在也使得服务请求者与服务提供者之间进一步解耦。在服务注册中心包含有已发布的标准拌合比对比服务与碾压监控中的碾压轨迹生成服务的描述信息，其描述信息主要包括服务功能描述、参数描述、接口定义、信息传递等相关信息。

服务请求者通过 WebService 技术调用服务。当服务完成发布，在服务请求者要使用已发布的服务。利用 Web Service 技术在拌合监控阶段，通过服务注册中心获取拌合监控中标准拌合比对比服务的相关功能，接口，参数及其返回值等相关服务信息；之后使用 Web Service 技术传递服务所需的标准拌合比等相关参数，进而调用该服务相关的运算、处理和分析。利用 Web Service 技术在碾压监控阶段，通过服务注册中心获取实时施工数据采集处理服务的服务定义和功能，接口，参数及其返回值等相关服务信息；之后根据施工工地的具体情况选择不同的定位方式，传递服务所需相关参数，最后实时施工数据采集处理服务运行结束返回的绘制碾压轨迹坐标点同样是利用 Web Service 技术传递至服务请求者。服务请求者接收到碾压轨迹的坐标点后最终完成碾压轨迹的绘制工作并在界面中将其呈现出来。在这期间服务请求者无需了解服务是如何对数据进行处理和分析的。

整个项目历时\*\*个月开发，于\*\*年\*\*月完成交付，到目前运行稳定。通过在水利施工工地等恶劣环境下的一段时间的使用，用户普遍反馈良好。总体来讲，选用 SOA 有如下优势：1、系统更易维护。当需求



发生变化时，不需要修改提供业务服务接口，只需要调整业务服务流程或者修改操作即可。2、更高的可用性。该特点是在于服务提供者和服务请求者的松散耦合关系上得以发挥与体现。这种没有绑定在特定实现上、具有中立的接口定义的特征称为服务之间的松耦合。松耦合有两个明显的优势，一是它的灵活性，其独立于实现服务的硬件平台、操作系统和编程语言；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。3、更好的伸缩性，依靠业务服务设计、开发和部署等所采用的架构模型实现伸缩性。使得服务提供者可以互相彼此独立地进行调整，以满足新的服务需求。

实践证明，SOA 技术的使用大大提高了系统开发效率，节省了开发和维护成本，使得系统具有更好的开放性、易扩展性和可维护性，从项目完工后的使用效果来看，达到了预期目的。

## 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过问题，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。



## 论数据访问层设计技术及其应用

在信息系统的开发与建设中，分层设计是一种常见的架构设计方法，区分层次的目的是为了实现“高内聚低耦合”的思想。分层设计能有效简化系统复杂性，使设计结构清晰，便于提高复用能力和产品维护能力。一种常见的层次划分模型是将信息系统分为表现层、业务逻辑层和数据访问层。信息系统一般以数据为中心，数据访问层的设计是系统设计中的重要内容。数据访问层需要针对需求，提供对数据源读写的访问接口；在保障性能的前提下，数据访问层应具有良好的封装性、可移植性，以及数据库无关性。

请围绕“论数据访问层设计技术及其应用”论题，依次从以下三个方面进行论述。

1. 概要叙述你参与管理和开发的与数据访问层设计有关的软件项目，以及你在其中所担任的主要工作。
2. 详细论述常见的数据访问层设计技术及其所包含的主要内容。
3. 结合你参与管理和开发的实际项目，具体说明采用了哪种数据访问层设计技术，并叙述具体实施过程以及应用效果。

### 范例

摘要部分：

\*\*年\*\*月，我司承接了\*\*系统的项目，该系统主要实现\*\*，主要包括\*\*等功能模块。我在该项目中担任技术负责人，主要负责该项目技术方案的制定及技术指导工作。本文以\*\*系统为例，主要论述了数据访问层技术在该项目的具体应用。首先在技术选型上，综合需求层面及技术特点的考虑，确定了 Mybatis 技术；接着通过导入资源包、配置核心文件及配置映射文件三大步骤来具体进行数据访问层的开发；最后针对业务执行层面、一二级缓存的配置及数据量大的查询等方面进行性能调优。通过以上技术应用，项目最终顺利上线，达成客户期望，获得用户一致认可。

【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\*\*，\*\*支持\*\*，\*\*主导研发\*\*系统。该系统将\*\*进行了严谨精准的管控。【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，我司承接了\*\*系统的建设。\*\*系统作为\*\*，包括\*\*等十几种任务管理模块。主要实现\*\*，对\*\*。\*\*系统拥有\*\*，且存储\*\*。因系统\*\*，因此如何使用数据访问层技术来满足本项目的功能及性能需求成了我们考虑的技术问题之一。【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】

数据访问层技术也称持久层技术，主要负责将系统数据通过调用数据库的 SQL 语句对数据进行读取和存储。java 语言的 JDK 中自带一种数据访问层技术——JDBC。市面上主流的数据库均实现了 JDBC 中所定义的对数据库访问的各类接口，因此 JDBC 能够较为高效的完成对各类数据库的数据存取操作。但是，单纯的使用 JDBC 进行开发，效率低下的同时会增加人力成本的支出。基于这种情况，Hibernate 和 Mybatis 技术应运而生。Hibernate 是一个对 JDBC 进行轻量级封装的全自动框架，能够实现数据库道 POJO 的完整映射，开发人员无需编写复杂的 SQL，仅需对 POJO 进行操作便可同步处理数据库中的数

据。而 Mybatis 相对 Hibernate 而言则是一个半自动的框架，可以支持自定义的 SQL 语句的操作，相对于复杂的数据查询操作支持性更佳。

针对 SPS 项目，如何应用数据访问层技术，是 SPS 项目组所关注的问题。总体来讲，我们将这个问题分成了三个阶段来处理，分别是技术选型、技术应用以及性能调优。

### 技术选型

从 SPS 的整体层次上来讲，为了保障 SPS 系统的处理拣料数据的效率，我们将九成以上的事务处理放到了数据库层面进行执行，也就是通过数据库层面的存储过程来完成事务的处理，因为数据库本身的事务处理效率比起 Spring 框架中的事务机制要更加高效，且彼此间的网络开销会更低。而作为上层的 JavaWeb 工程，主要是为用户提供对应的查询检索机制，主要包括两大块：其一，将每一种拣料任务的完成进度为用户提供查询的接口；其二，为用户提供有关拣料任务的各种类型的复杂报表，报表的关联性查询较多。Hibernate 本身的优势在于对数据的更新处理可通过对 POJO 的处理直接映射到对应的数据表/视图中，由于 SPS 系统的事务处理并不在 Java 层面，显然在这个系统中作用不大；另外 Hibernate 是一个全自动框架，对自定义的 SQL 查询支持性并不好；而 Mybatis 相对能够很好的支持自定义 SQL 的开发，对于 SPS 系统而言，不仅能够很好的支持各种简单对象的自定义查询，在复杂报表的开发支持上较 Hibernate 更加强大，能够显著的提升开发效率。因此，从 SPS 系统的需求层面及两种数据访问技术的特点上考虑，Mybatis 更适合 SPS 系统。

### 技术应用

SPS 系统的开发根据拣料指令业务类型，一共分成了十多个 JavaWeb 工程，每一个 Web 工程专门负责处理一种拣料指令。所以，在应用 Mybatis 进行数据访问层开发时，所有的工程都将引用 Mybatis 的资源 jar 包。在应用过程中，主要包括三大步骤。首先将 Mybatis 的 jar 包资源，导入到一个公共目录中，并将该路径通过 IDE 开发集成环境工具设置到公共资源库中为各个工程包所引用。其次，为所有的工程包配置 Mybatis 的一个名为 sqlMapConfig.xml 的核心配置文件，主要包括整个系统运行时的公共配置信息，例如数据库的基本配置信息（数据库连接路径、驱动、帐号及密码）。然后，针对每个模块的开发为其实体类单独配置 Java 对象；例如 SPS 系统用户要查询来自 APA 系统中的计划指令的信息，其实体类为 PlanOrderVO，对应的数据访问接口文件为 PlanOrderMapper.java，该接口文件内定义了一系列的查询计划指令的方法，则在开发过程中同样需要配置一个名为 PlanOrderMapper.xml 的映射文件；映射文件中定义的每一个查询语句皆与接口文件的方法保持一致，且通过<resultMap>标签来定义 Java 对象及其属性与数据库对象及其字段之间的映射关系，同时将自定义的 SQL 语句在该配置文件中完成。

### 性能调优

从整个系统的角度上讲，性能涉及到方方面面，但从数据访问层技术的应用上看，调优主要包括以下几个方面。首先是业务执行层面的确定。众所周知，事务级操作在数据库层面比在 Java 层面要快得多。因此，对于复杂的事务处理，乃至复杂的报表查询，都定义到了数据库层面（存储过程、视图），Java 层面只需要负责传参和接受结果即可。其次，对于 Mybatis 里面的相关设置的应用，也决定了系统处理数据的效率，最典型的，就是一二级缓存的配置。对 Mybatis 而言，如果配置了缓存，系统会优先在一二级缓存中寻找目标数据，当在一二级缓存中找不到数据时，才会到数据库中进行查询。但是缓存的配置因为缓存的有限容量也存在使用上的讲究。根据二八定律，我们将最常用的上月数据统计报表结果会放入缓存

中，避免系统在数据库层面对百万级的数据进行多次的复杂关联查询。最后，对于数据量太大的查询，SQL 语句的写法本身也是决定着查询效率的因素。例如，数据库执行 SQL 语句的过滤条件时，是从后往前执行，因此，如果某个条件能够最大限度的降低数据范围，那么这个条件在 SQL 语句中应该放在后面。

通过以上技术的应用，项目历经\*\*，\*\*系统最终于\*\*年\*\*月顺利上线，目前系统运行稳定，获得了用户一致认可。无论是从业务执行层面，还是在面临数据量大的查询，系统的稳定性都达到了预期。

但在该系统实施过程中，其实也发现了 Mybatis 技术相对于 Hibernate 技术的不足。如 Hibernate 拥有可通过配置方言属性（Dialect）屏蔽数据库类型的特点，让用户不用考虑不同数据库的操作方式，但 Mybatis 因为要支持灵活的 SQL 语句，导致无法支持方言。当某天系统数据库需要进行切换时，对于 Mybatis 而言是一个重大问题，届时需要将所有涉及到与数据库的对接写法全部维护，无疑增加了维护的工作量。要真正解决这个问题，恐怕需要从更高的层面，建立统一的数据库标准后才能实现。当然，这也是我们在下一代技术应用中需要去完善的地方。

## 论软件系统架构评估

对于软件系统，尤其是大规模的复杂软件系统来说，软件的系统架构对于确保最终系统的质量具有十分重要的意义，不恰当的系统架构将给项目开发带来高昂的代价和难以避免的灾难。对一个系统架构进行评估，是为了：分析现有架构存在的潜在风险，检验设计中提出的质量需求，在系统被构建之前分析现有系统架构对于系统质量的影响，提出系统架构的改进方案。架构评估是软件开发过程中的重要环节。

请围绕“论软件系统架构评估”论题，依次从以下三个方面进行论述。

- 1.概要叙述你所参与架构评估的软件系统，以及在评估过程中所担任的主要工作。
- 2.分析软件系统架构评估中所普遍关注的质量属性有哪些？详细阐述每种质量属性的具体含义。
- 3.详细说明你所参与的软件系统架构评估中，采用了哪种评估方法，具体实施过程和效果如何。

### 范例

#### 摘要部分：

\*\*年\*\*月，我参与了\*\*项目的开发，该项目主要功能是\*\*。我在该项目担任系统架构师，主要负责系统整体架构设计及评估等工作。本文以\*\*项目为例，主要论述了软件系统架构评估在本项目中的具体运用。在描述介绍阶段，组织项目团队共同学习 ATAM，促进甲乙双方达成架构评估的共识；在调查分析阶段，建立质量属性效用树，完成各场景的风险点、敏感点和权衡点分析；在测试评估阶段，进行场景的优先级排序，输出评估报告。通过以上技术的使用，使得系统开发的质量得到了保证，为后续项目的顺利实施提供了有力的支撑，最终项目于\*\*年\*\*月正式上线，获得甲方各级管理层的一致好评。

【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：

随着\*\*的速度发展，\*\*多次出现\*\*，造成\*\*。【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，\*\*企业开始了\*\*项目的开发，该系统主要整合了\*\*等子系统，以此解决\*\*等问题。从而支持\*\*满足\*\*\*多元化需求。该\*\*系统将\*\*整合为\*\*系统。比如\*\*门店的\*\*涉及\*\*等功能模块。项目总投入\*\*万，我在该项目中担任系统架构师，主要负责系统整体架构设计及评估等工作。团队成员\*\*余人，涉及\*\*等方面专家，历时\*\*个月，于\*\*年\*\*月正式上线。【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】

项目范围广、系统复杂、涉及各种异构系统整合，因此选择一种合适的架构评估方法显得非常重要。架构所关注的质量属性主要包括性能、可用性、可修改性、安全性等。性能是指系统的响应能力，比如在某段时间内系统所能处理的事件的个数。可用性是指系统能够正常运行的时间比例，经常用两次故障之间的时间长度来表示。安全性是指系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或者拒绝服务的能力。可修改性是指能够快速地对系统性能价格比进行变更的能力，通常以某些具体的变更为基准，通过考察这些变更的代价，来衡量可修改性。常用的架构评估方法有：基于问卷调查的评估方



式、基于场景的评估方式和基于度量的评估方式。

\*\*项目在实施企业应用集成时，为了解决 30 余个子系统的集成问题，提高架构的稳定性，主要采用了基于场景的架构权衡分析法 ATAM。下文具体从描述介绍阶段、调查分析阶段、测试评估阶段三个方面来进行论述。

### 1、描述介绍阶段。

通过组建项目评估小组，组织项目团队共同学习 ATAM，促进甲乙双方达成架构评估的共识。甲方已有 30 余个系统，涉及干系人较多，需要统一的架构评估方法达成共识、保证系统质量。首先根据项目需要成立了项目评估小组，主要成员包括评估小组负责人、项目决策者、架构设计师、用户、开发人员、测试人员、系统部署人员等项目干系人。然后组织召开项目列会，介绍 ATAM 方法，它是一种基于场景的软件架构评估方法，对系统的多个质量属性基于场景进行评估。通过该评估确认系统存在的风险，并检查各自的非功能性需求是否满足需求。该系统的目的和商业动机，主要是通过新零售 iStore 项目建设，完成业务中台、数据中台、AIOT 中台三大系统，支持堂吃、外卖、零售业务的快速增长。最后安排架构组长向项目组成员描述了系统将要采用的 SOA 架构，以及各业务线的领域功能，系统服务端采用 Linux 系统、Java 架构开发。通过以上措施，我们通过 ATAM 的研讨，完成了架构评估的前期准备工作，达成了架构评估的共识。

### 2、调查分析阶段。

通过收集各项目干系人的质量属性要求，建立质量属性效用树，完成各场景的风险点、敏感点和权衡点分析。智慧门店涉及大量的会员营销活动，需要权衡业务需求方、终端用户、开发人员等项目干系人的质量属性要求。首先组织项目例会，邀请项目组成员发表意见。比如甲方信息中心提出系统要保证其可靠性，系统发生故障必须在 5 分钟内恢复，此优先级最高。而开发人员则提出接口众多，无法预估 BUG 修复的时间。然后组织项目评审，讨论重要的质量属性目标，生成属性质量效应树，例如性能指标的并发数、QPS、平均响应时间。最后分析场景的风险点、敏感点、权衡点。风险点：当第三方聚合支付供应商无法提供服务时，会直接影响业务可用性。敏感点：在线支付信息的加密级别。权衡点：为提高可用性，需要增加集群线路，但集群线路增加，势必影响系统调试的工作量，影响可修改性。通过以上措施，我们建立了质量属性效用树，完成了风险点、敏感点、权衡点分析，提高了架构评估的有效性。

### 3、测试评估阶段。

通过组织项目例会，进行场景的优先级排序，评估最佳的应对方案，输出评估报告。数据中台、业务中台、AIOT 中台涉及的业务、数据需要全部在线化、迁入云端，如果不能支持高并发，则会导致出现大范围的服务无响应。首先组织评估小组集体讨论，确定了不同场景的优先级。系统的可用性最高，性能其次，可修改性及安全性优先级较低。主要是通过 Nginx 集群、Tomcat 负载均衡、Redis 集群、MySQL 集群，良好的支持了营销活动的抢红包、秒杀等场景。然后评估服务注册中心、分布式开关的限流、降级的有效性。采用令牌桶算法实现限流策略，控制访问速率以及应对突发性的流量。另外考虑网络的异常情况，增加监测机制，针对超时无响应的情况，采取回滚机制，提高可用性。最后制定了评估报告，结合系统的风险点、敏感点、权衡点和非风险点，递交了架构分析文档、架构场景及优先级、质量属性效应树、风险点决策等文档。通过以上措施，我们实施了 ATAM 架构评估，制定了架构评估报告，提升了架构评估的全面性。

通过使用软件系统架构评估，使得系统开发工作完成得非常顺利，系统开发的质量得到了保证，对后续项目的顺利实施提供了有力的支撑。项目于\*\*年\*\*月完成上线，通过中台系统的订单聚集分发，各项性能指标达到甲方要求，并经受住了 100 万用户参与的双 11 秒杀活动，系统宕机的问题得到解决，营销活动的千人千面也取得了很好的效果，项目获得了甲方管理层的表扬。

虽然项目整体上取得了成功，但具体实施过程中，其中对于 APP 架构评估的重视度不够，没有考虑微信小程序与微信公众号之间的衔接问题。通过堂吃、外卖、商城界面的导航栏统一，优化了小程序和 H5 的访问入口，提高了用户体验。另外由于需求的变化，频繁的升级，影响独立构件的稳定性，通过微服务的部署，提升了独立构件的灵活性。通过这次的评估工作，使我对软件架构的评估工作方方面面都有更深入的理解，规范的、正确的软件架构评估方法会给系统的开发阶段带来效率的大幅提高，基于经过评估的架构开发出来的系统质量有很好的保证，大幅减少的了需求实现的偏差。