**摘要:** 【问题】 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。 【摘要】 2018年4月，本人所在公司承担了一项公务用车系统平台的开发项目，该项目主要是响应国家机关事务管理局印发的《党政机关公务用车管理办法》，将公务用车的审批流程电子化，公务用车的使用也会更加透明。我在该项目中担任系统架构师一职，负责系统架构设计的工作。本文以公务用车系统平台的开发项目为例，主要论述了软件系统架构风格在该系统开发中的应用。在公车系统平台中，针对不同的角色的不同操作权限和任务逻辑我们采用了面向对象风格；在告警系统中我们使用了事件驱动风格，通过隐式调用执行相应的告警操作；在系统中整体使用了层次结构风格，将整个系统分成了数据层、DAO层、服务层和表示层。通过使用这些架构风格，规范了系统开发和组织方式，也便于后期维护，最终项目顺利上线，并获得用户的一致好评。

2018年4月，本人所在的公司承担了某省的公务用车系统平台的开发项目，该项目主要是响应国家机关事务管理局印发的《党政机关公务用车管理办法》，规范化公务用车流程，避免公车私用，并让结算有据可循。用户只需要定制业务服务即可，避免了直接采购设备带来的后期维护、设备维修、产品升级等问题。该系统分为网页管理平台和移动端操作App。在系统中有位置服务模块、事件提醒模块、车辆管理模块、统计分析模块、系统设置模块。位置服务模块包含车辆定位、电子栅栏和轨迹查询功能。通过在车上安装OBD终端设备实现位置上报。事件提醒模块主要负责异常用车、故障维修、保险年检、违规用车的告警，通知相关的负责人。车辆管理模块包含组织机构、派单管理、车辆管理和车辆公告四个子模块。而统计分析模块有里程油耗、成本、异常用车、派单、违规用车和司机考勤等项目的统计分析。系统设置模块主要包含账号权限管理、规则设置等功能。在该项目中，本人担任系统架构师，负责项目的架构设计。  
常见的软件系统架构风格有5类，分别是数据流风格、调用/返回风格、独立构件风格、虚拟机风格和仓库风格。其中数据流风格包含了批处理风格和管理过滤器风格，批处理按顺序处理数据，管理过滤器可以对多个输入分别同时处理。调用/返回风格包含了主程序子程序风格、数据抽象和面向对象风格、层次结构风格。主程序子程序风格的子程序的处理结果正确性取决于它调用的子程序的正确性。而面向对象风格侧重于将数据抽象成对象。层次结构风格将复杂的任务分成了多层。独立构件风格包含进程通信风格和事件驱动风格。事件驱动风格的构件通常触发或广播一个或多个事件。虚拟机风格包含解释器风格和基于规则的系统。具有解释器风格的软件含有一个虚拟机，可以仿真硬件的执行过程和一些关键应用。仓库风格包括数据库架构风格和黑板架构风格。数据库架构风格的构件有两大类，一类是中央共享数据源，保存系统的数据状态，一类是处理元素，处理元素对数据元素进行处理。黑板系统风格通常是用在解决问题没有特定性算法的系统中。  
在公务用车系统平台的开发过程中，我们综合使用了多种软件系统架构风格，本文着重对面向对象风格、事件驱动风格和层次结构风格三个风格在该项目中的具体应用进行介绍。  
一、面向对象风格  
在系统中有上级调度、调度、申请人、司机、用车人等角色。每个角色的使用权限不同，每个角色能够处理的任务也不同。派单管理中的车辆派单任务也有申请用车、驳回、司机接单、司机拒绝、审批通过、出发、到达、收车、调账等状态，不同状态的派单任务有不同的操作对象和操作逻辑。类似的还有告警信息、车辆、账单等数据。在进行系统设计时，我们采用了面向对象风格，将这些数据抽象成对象，将角色权限、派单任务状态、告警信息类型、车辆状态、账单状态等封装在对象中，这些权限和状态由对象自己管理，这样可以保证每个对象的独立性。这同时也有助于统计分析模块提取分析各个对象的信息，比如司机的考勤信息，只需要单独汇总司机角色的信息即可，里程燃油等信息也包含在了车辆对象中。不同对象在通过外部函数和过程方法调用和交互时使用也更加便捷。例如派单任务管理模块中进行派单操作涉及不同的角色与不同的订单状态进行交互，通过对象的封装这里就可以使用责任链模式将不同任务与角色关联起来，达到更好的协同作用。  
二、事件驱动风格  
在车辆管理模块中会有频繁的状态更新，同时它也需要与多个其他模块进行交互。比如异常用车、故障维修、违规用车等的告警需要通知相应角色，同时要触发告警管理模块、账单管理模块、派单管理模块相应状态的更新。因显示调用模块间的耦合性太高，所以我们使用了事件驱动风格，通过隐式调用触发事件。在程序设计过程中，设计者需要定义被观察的主题对象车辆类，从车辆对象创建时开始对该类进行监听，观察者分别为派单管理模块的派单管理类、告警模块的告警类和账单模块的调账类。当该订单出现异常例如故障维修的情况，告警类可以同步对该情况进行处理。调账类也会同步相应的调账申请的操作。同时派单管理类在自动派单时就会屏蔽该车辆及其对应的司机，避免派单异常的情况。使用事件驱动风格很好的实现了各个模块的解耦，模块职责也更清晰。  
三、层次结构风格  
在架构设计时，我们整体采用了层次结构风格。整个系统被分为数据层、数据库操作层、服务层、表示层。首先底层数据层使用了MySQL数据库，里面保存了各个模块的数据，比如车辆管理模块的车辆状态、里程油耗、派单状态等信息，位置服务模块保存了车辆定位、栅栏、历史轨迹等信息。而操作数据库使用的是DAO层，DAO层接受服务层传来的指令对数据库进行增删改查的操作。服务层主要是处理各种业务逻辑，然后将处理结果返回给表示层展示出来。比如统计分析模块里在表示层会有车辆里程油耗图表的展示，它的数据源就是服务层，服务层收到来自表示层的数据请求，就会调用DAO层对外提供的方法获取数据，DAO层通过查找方法从数据库的车辆表中获取到里程油耗数据返回给服务层，服务层对里程油耗数据进行处理，然后将处理后的数据返回给表示层，表示层刷新并展示数据，形成一个从顶层到底层又从底层返回到顶层的完整路径。  
整个项目在2019年2月份完成交付，获得了客户的一致好评。整个系统的业务逻辑比较复杂，同时也涉及了多种使用角色，每种角色也对应不同的业务线，这些业务线也有重合部分，使架构设计难度大大提高。在这里面架构设计风格帮了很大的忙，比如层次结构风格简化了整个系统的逻辑，事件驱动风格很好地实现了模块的协同和解耦。  
我们在使用这些架构风格进行设计的时候，也存在一些不足。车辆历史轨迹功能模块在后续使用时偶尔会出现轨迹交叉甚至是穿墙的问题。最初生成轨迹时，使用的是汽车上安装的OBD上传的经纬度加上百度地图SDK的鹰眼轨迹服务直接生成轨迹图，从后面用户反馈来说，部分轨迹生成效果不太理想。后面针对生成有问题使用了黑板架构风格，对大量的轨迹数据进行分析，在鹰眼轨迹算法基础上优化了相关算法，经过多轮迭代和验证后终于解决了这个问题。后面开发类似需要大量数据验证的模块时，尽量能够提供模拟数据多角度验证保证系统运行结果正确，避免在生产环境出现问题。