## 论基于构件的软件开发在图书馆数据中台的实践

## 一、摘要

2019年11月，本人所在信息技术部承接了馆内数据中台的建设，该项目为期半年。通过数据中台，实现了我馆数据的分层与水平解耦，沉淀了公共基础数据。通过数据建模实现跨域数据整合和知识沉淀，通过数据服务实现对于数据的封装和开放，快速、灵活满足上层应用的要求，通过数据开发工具满足个性化数据和应用的需要，为全市文旅数据融合工作提供了有力的数据支撑。该项目时间紧任务重，涉及人员组织多， 涉及馆内 10 余个部门，外部配合协作 10 多个厂商团队。该项目于 2020 年 5 月正式上线， 2020 年 6 月通过最终验收。本人在该项目中担任系统架构师，主要完成技术方案评估与实现，项目立项论证等工作。本文结合笔者实际经验，以该项目为例，讨论微服务架构下，基于构件的软件开发方法的应用，包括基本构件开发的概念及特点以及在实现过程中遇到的问题及其解决方案。

## 二、正文

近年来，随着我馆服务质量的不断提升，大量软硬件系统在我馆得到部署。硬件设备包括自助借还机、交互机器人、电子书借阅机、人脸识别门禁、人流量计数器，软件系统包括图书管理系统、微信小程序、移动端APP。软硬件系统的增加带来了大量用户数据的流入，如读者借阅数据、读者信息、阅读偏好、读者常见问题、入馆人次、入馆时间等海量的数据，但是这些数据缺乏统一的接入和分析。因此，需要通过数据技术，对海量数据进行采集、计算、存储、加工，同时统一标准和口径，并在此基础之上，对数据进行归纳分析，对外提供统一的馆情数据服务。

笔者带领团队于2019年11月承接馆内数据中台的建设，并担任系统架构设计师。主要负责项目整体技术方案评估实现、立项论证以及项目管理工作。该项目的架构工作于同年12月完成，整个项目耗时6个月，于2020年5月上线测试。于2020年6月完成验收。

笔者经过一段时间的架构考察和评估，最终决定采用微服务架构建设新系统。采用微服务架构的目的是充分拆分庞大臃肿的系统，以促进软件的敏捷开发和部署。在此项目中，每个小组负责一个组件的完整生命周期，从开发到测试，再到运维以及后续迭代升级。最后，各个服务组件通过REST接口进行组装。在微服务的基础上，为了实现代码复用和快速开发的需求，通过基于构件的开发方法进行软件开发。基于构件的软件复用作为一种提高软件生产率和软件质量的有效途径,是近几年软件工程界研究的重点之一,被认为是继面向对象方法之后的一个新的技术热潮。一般来说,基于构件的复用包括 3 个相关的过程：构件的开发、构件的管理和基于构件组装的系统开发。在基于构件的软件开发中,系统开发的重点从程序设计变成构件组装。特别是近年来,在中间件技术的基础上,结合软件复用思想和面向对象方法,基于构件的软件开发技术受到了高度重视。

根据笔者的实践经验，基于微服务架构的馆情数据中台的构件可以分为中间件构件和业务构件。其中中间件构件包括微服务注册中心：注册并维护远程服务及服务提供者的地址，供服务消费者发现和调用，为保证可用性，通常基于分布式 KV存储器来实现，比如 zookeeper 等；服务框架：用于实现微服务的 RPC 框架，包含服务接口描述及实现方案、向注册中心发布服务等功能，常见的 RPC 框架包括 Spring Cloud、Dubbo、gRPC、 Thrift、go-micro 等；服务网关：介于客户端与微服务之间的网关层，可以理解为「门卫」的角色，以确保服务提供者对客户端的透明，这一层可以进行反向路由、安全认证、灰度发布、日志监控等前置动作；服务监控：对服务消费者与提供者之间的调用情况进行监控和数据展示；服务追踪：记录对每个请求的微服务调用完整链路，以便进行问题定位和故障分析；服务治理：服务治理就是通过一系列的手段来保证在各种意外情况下，服务调用仍然能够正常进行，这些手段包括熔断、隔离、限流、降级、负载均衡等。基础设施：分布式消息队列、日志存储、数据库、缓存、文件服务器、搜索集群等，用以提供服务底层的基础数据服务。

业务构件包括RBAC认证构件，OAUTH2.0 授权构件、数据接入构件、消息队列微服务构件、数据开发构件、数据模型构件、数据服务构件。这里重点从数据接入构件、数据开发构件、数据模型构件、数据服务构件三方面展开。

数据接入构件角度来说。数据接入是中台的基础业务，通过对接各个软硬件厂商，针对不同类型软硬件系统的特点，提出不同的数据需求,联合各个厂商一起讨论制定标准化的上报接口，以确保后续的同类型设备都能被快速接入。在开发阶段，依据接入对象的不同，将不同的接入任务分摊给不同的项目组进行开发。各个小组负载每个微服务从开发、部署到运维的整个生命周期，并且提供完善的构件开发文档，确保构件的复用。在各小组开发完毕之后，各个服务之间采用 Redis 作为微服务消息总线进行集成。

在数据模型构件方面，笔者这里概括为三层。分别是基础模型构件、融合模型构件和挖掘模型构件。模型构件可以为上层数据服务提供数据角度的构件基础，实现复杂数据业务的构件表达。其中基础模型一般是关系建模，主要实现数据的标准化，这也叫作“书同文、车同轨”。通过设计好的数据标准将接入的数据进行入库管理。其次是融合模型。融合模型一般是维度建模，主要实现跨域数据的整合，整合的形式可以是汇总、关联，也包括解析。这些融合模型表是通过各个业务部门梳理而得到的常用的宽表，比如读者终端使用行为表、读者阅读偏好表等，目前这类已经梳理出来的模型（宽表）已有20余个。最后是挖掘模型。常规理解，挖掘模型其实是偏应用的，但是在本馆的馆情分析中，例如读者常驻区域模型这类常用的挖掘模型，也把它规整到中台模型，以开放给其它人使用。

在数据服务构件方面，笔者将数据模型按照应用要求做服务封装，就构成了数据服务构件。数据服务构件调用之前，通过服务网关调用RBAC身份证认证构件实现鉴权。数据服务构件跟业务中台中的服务概念是完全相同的，只是数据封装比一般的功能封装要难一点，毕竟OLTP功能的变化有限，而数据分析受业务因素的影响很大，变化更快，导致服务封装的难度变大。但也不是毫无逻辑可循，通过分析整理常规的报表服务，笔者所在团队提取了大量的原子口径，这些原子口径可以方便的组装成各类次级服务。即使业务口径发生变化，也能第一时间通过基础服务组装的形式快速生成目标服务。

但有数据模型和数据服务还是远远不够的，因为再好的现成数据和服务也往往无法满足前端个性化的要求。这时候就得授人以鱼。数据中台的最后一层就是数据开发，其按照开发难度也分为三个层次，最简单的是提供标签库（DMP），用户可以基于标签的组装快速形成分析报表，一般面向普通馆员。前端馆员以拖拽的形式选择感兴趣的标签字段，比如性别、常驻区域、阅读偏好（分类号），通过添加逻辑运算符组合，并点击确定就可以快速生成目标读者群体。然后可以通过电话、短信、APP等推送方式对读者进行针对性的书籍、活动等推送服务。在标签库层次上，由于其包含前端交互层，因此笔者率领团队进行了大量适合数据中台的前端组件开发，并入数据中台UI组件库，确保软件复用。其次是提供数据开发平台，用户可以基于该平台访问到所有的数据并进行可视化开发，一般面向技术部SQL开发人员。最后就是提供应用环境和组件，这一块可以是基于数据服务构件也可以是基于数据接入构件，让技术人员可以自主打造个性化数据产品，以上层层递进，满足不同层次人员的要求。

通过基于构件的开发方法，该数据中台项目顺利上线并通过验收。通过数据中台的建立，提升了数据仓库的建设水平，为数据的开放和服务打下了坚实的基础。在项目的实施过程中，笔者体会最深的是，数据中台可以中构件的不仅包括传统的业务构件，还包含大量数据模型。但是笔者也发现，数据中台并不是一个稳定的业务，数据模型构件时刻会随着业务的演讲发生变化，因此构件的管理和维护是值得关注和研究的方向。除了传统的文档形式加以记录和管理之外，也应该利用数字化平台实现构件的集中式管理。