

统一接入问题分析报告

王海阔

2015-01-27

Contents

1	概述	3
2	问题及解决思路	3
2.1	缺少业务分析	3
2.2	服务设计不严谨	4
2.3	没有数据通信协议规范	4
2.4	配置管理不到位	4
2.5	测试不充分	5
2.6	软件过程不严格	5
2.7	服务管理需要加强	5
2.8	架构和技术改进	6
3	小结	6

1 概述

负责统一接入的开发管理工作已经快一年了，对统一接入的架构和技术体系有了一定的了解；对统一接入的开发、测试、发布等活动也比较熟悉。经过一段时间的实践，尽管曾经提出并实施过一些方法和工具，但是从下一代交易结算系统整体的角度来看，统一接入还存在着一下问题和不足。由于时间和个人能力的关系，这里列出问题，只针对其中一部分问题提出的解决建议。

从下一代交易结算系统的角度来说，统一接入应该扮演着 **ESB** 的角色，是 **SOA** 架构的核心组件。统一接入已经实现了 **ESB** 最核心的功能，即支持分布式异构应用系统的集成。具体功能包括：

- 实现了服务的路由和寻址，使服务位置透明化
- 可以对服务的命名和寻址进行管理
- 实现了请求/响应（包括同步和异步）的消息机制，同时支持发布/订阅机制
- 统一了通信协议，对外使用 **T2** 协议，内部使用 **F2** 协议
- 能够连接 **KCXP**、**TBUS**、**WebService**、**FIX** 等多种协议
- 能够通过配置的方式进行数据交互协议的转换

下面围绕统一接入的上述核心功能，从服务定义、服务管理、协议规范、技术、软件过程、运维、高级 **ESB** 功能等角度，。。。。

2 问题及解决思路

2.1 缺少业务分析

在 **IBM SOMA**¹中，一个业务服务的确定，需要经过业务流程分析、功能区域分析、业务角色关联、业务对象梳理等一系列环节。在统一接入的服务分析和设计过程中，没有从业务层面进行分析，直接从技术层面开始着手设计，导致服务的目标不明确，也很难在应用层面进行前瞻性的规划。

建议使用 **CBM**²方法论指导整个业务分析的过程，首先在业务层面建立服务模型，并依次作为应用层面开展设计的前提和基础。在确定了一个业务服务后，要针对该服务设计业务用例，同时明确该服务必须要满足的指标 (**KPI**)，作为后续服务验收的标准，以及应用级监控的关键指标。

¹**SOMA**(Service Oriented Modeling and Architecture)，[面向服务的建模与体系结构](#)，是一种 **SOA** 解决方案实现的端到端方法。基于目标业务流程分析进行 **SOA** 的设计和构建。

²**CBM**(Component Business Modeling)，[组件业务建模](#)，按照功能区域和业务层级两个维度确定可管理、模块化、可重用的业务组件的一种 **SOA** 建模方法论，可以用于业务建模、分析、识别，以及评估现有的业务能力。

2.2 服务设计不严谨

在确定了业务服务及其 KPI 之后，应该以此为需求进行应用服务的设计，并且在设计中应该遵循一定的原则。我们的应用服务设计是由应用系统的开发人员在各自应用系统的角度进行设计，没有从整体 SOA 架构出发，也无法保证能够遵守整个下一代的架构原则。

建议在服务设计时遵守以下过程和原则：

1. 首先分析现有的应用系统的已有接口能否满足需要。如果能满足，复用现有的接口。
2. 如果现有接口不能直接满足需要，判断能否通过统一接入进行转换。如果可以，优先选用这种方式。
3. 现有接口无法满足，判断在已有应用系统上增加接口能否满足需要。
4. 以上各个层面的组件都服务复用的情况下，才考虑增加新的应用系统/架构组件。如果是这种情况，必需先进行技术验证工作。
5. 一定要先根据数据通信协议的相关规范进行接口设计，通过评审后再开始考虑如何实现。
6. 对应业务组件/应用组件/技术组件的设计，需要有配套的规范进行约束。
7. 如果增加了新的架构组件，需要进行设计评审。

2.3 没有数据通信协议规范

在分布式架构中，每个应用系统对外发布接口的《数据通信协议》非常重要，是各个节点之间相互连接和通信的基础。整个下一代体系中对于数据通信协议没有统一的要求，协议文档的必备的要素、文档格式、字段类型、数据字典等，都由个系统的各厂商自行决定。在系统间整合时，极易发生接口和文档不匹配的问题，增加了沟通和开发的成本。

建议参照《证券期货业数据通信协议应用指南》³，建立整个下一代的数据通信协议管理规范。

2.4 配置管理不到位

尽管已经有了初步的《配置管理规范》，但是从整个下一代的管理来看，最基本的配置管理仍然没有实现。有些系统甚至没有将所有的代码和配置入库，更遑论为各个版本创建分支和标签。

配置管理的不规范也是很多问题的来源，建议在整个下一代的研发体系中进一步加强配置管理的要求。

³《证券期货业数据通信协议应用指南》，是证监会发布的关于数据通信协议渐进式收敛演进的目标、路线、建议、方法与流程的行业标准，提供了设计通信协议的方法论。

2.5 测试不充分

我们的每个版本的发布都经过了业务测试，保证了基本的软件质量，但是还不够充分。主要体现在：

1. 没有严格遵守开发-发布-部署-测试的过程。在有些紧急的情况下，会采用开发-测试-发布-部署的过程，没有针对发布的升级包进行测试，往往这样发布出来的版本会发生问题。
2. 在接口层面，测试案例通常只能覆盖每个版本新增加或修改的接口，没有进行全面测试。目前正在自动化测试工具的基础上积累测试案例，当案例积累到一定程度后，需要将全面接口测试作为版本发布前的一个必备项。
3. 没有区分不同情况。上一节提到的各种不同的服务实现方式，其测试内容应该有所区别。比如，如果一个接口增加了新的 T2 插件，应该针对该插件进行更细致的边界条件测试；如果在整个分布式架构中增加了节点，针对该节点要进行全面的测试（包括性能测试）。

2.6 软件过程不严格

我们已经建立了问题管理和版本发布的机制，并区分了开发、联测、UAT、生产等环境。但是这些机制并没有被严格执行，经常有“特殊情况”发生。比较典型的如下：

- 没有在问题管理中登记需求和 bug，导致变更无据可查
- 对各套环境“混搭”使用，比如开发人员使用 UAT 环境进行测试
- 正常情况部署的版本应该是开发环境 > 联测环境 > UAT 环境 > 生产环境，但有时会发生 UAT 甚至生产环境的版本高于联测环境的情况，这反过来又会影响配置管理
- 每个版本的发布应该是经过测试组验证通过，并由我们的项目负责人发布给运行人员，实际情况是有时开发商会给运行人员直接提供升级包

建议再次明确软件过程的各种机制，严格执行。

2.7 服务管理需要加强

- 应该构建服务仓库，并建立服务的注册、准入、发布、查询等机制
- 针对业务服务，可以根据其 KPI 构建 SLA（Service Level Agreement，服务水平协议），再以 SLA 为依据设计应用级监控的指标（包括状态、可用性、容量等），将这些指标纳入 ITIL 的 SLM(Service Level Management，服务水平管理）体系中，作为服务评价的标准。

2.8 架构和技术改进

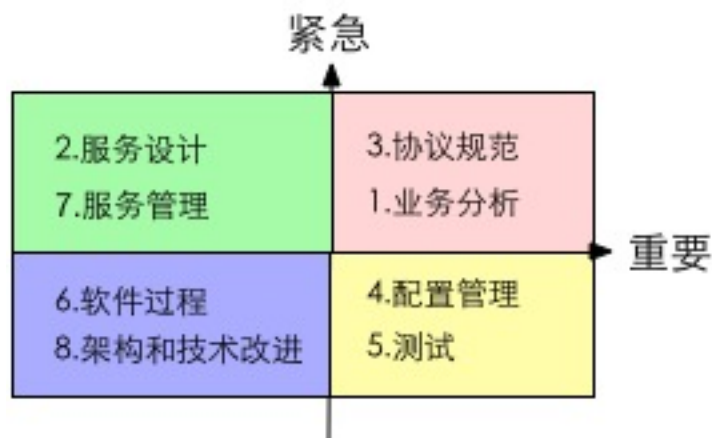
从统一接入的角度来看，在架构和技术上还可以改进，从而提供架构的灵活性，降低开发的成本，现罗列如下：

- SOA 架构上，没有与管理类系统的 Mule ESB 进行整体规划。应该形成整体、跨网络、跨协议，具有多个自治域的整体 SOA 架构
- 现有的统一接入，没有考虑服务编排的功能
- 基于内容的路由方面，可扩展性和灵活性不足，应该将“复杂的、动态的路由规则”作为一种服务组件而不是技术组件进行规划
- 目前外部访问统一接入都使用 T2 协议，过于单一。可以增加多种通信协议的支持，如 WebService, RESTful, Thrift 等，能够为其应用系统尤其是互联网系统方位统一接入提供更多的便利
- 目前统一接入的交互方式大量使用了请求/响应模式（支持同步或异步调用），极个别的接口使用了发布/订阅模式。为了在使用上更加灵活，还应该支持消息队列、推、拉等多种交换方式
- 从统一接入的配置来看，厂商标准化的配置和我们自定义的配置没有有效分离，在版本升级时带来很多配置合并的工作量。目前正在梳理，后续会进行改造。建议将这个问题列入统一接入的配置管理规范进行约束。
- 部署繁琐。尽管在联调环境试点了自动化部署工具，但是该模式并不适应运行部门现在的管理机制，还应该结合运行部门已经掌握的工具，进行自动化的工作。

这些架构和技术上的改进，可能更多的要有我们的开发人员推动，厂商只是配合和提供技术支持。

3 小结

上述的这些问题，有技术方面的，更多的是研发过程和管理方面的。有的更紧急，有的更重要。我个人认为其关系如下：



根据问题的紧急和重要程度不同，建议解决的优先级排序如下：

2.3 协议规范 -> 2.1 业务分析 -> 2.2 服务设计 -> 2.7 服务管理 -> 2.4 配置管理 -> 2.5 测试 -> 2.6 软件过程
-> 2.8 架构和技术改进