

# 服务模型分析设计

## Abstract

作为以服务为中心的架构设计，在整个 SOA 的实践过程中，如何合理、有效的获取服务并表达成为所开发系统成功实施 SOA 技术的关键之一。在本次比赛中我们依据开发灵活的以按需业务流程为中心的开发方法进行服务模型的分析与设计。详细描述了针对题目所给出的实例进行开发时的依据、思路、详细的设计流程和结果。

## Introduction

在本文中，我们会介绍一系列的基于流程建模的思想在我们这个项目中的实践。我们所关注的重点是如何从一段关于业务需求的描述中抽取我们所需的可灵活重组的服务和组件。这包括对业务流程的建模、服务建模和组件建模。在下面的讨论中，我们会从三个方面具体描述这个过程：如何发现并定义所需服务、服务的接口和服务实现的初步分析。

## Content

首先我们有必要再次回顾一下系统的实际需求背景。凤凰公司希望万合公司能够将该公司日益扩大的业务机会与销售订单相整合，通过先进的信息化手段提高工作效率并为企业后续发展做统筹。这是最为基本的需求。其次还有一些可视化信息聚合、网上智能信息系统等扩展需求。根据凤凰公司对上述业务流程的描述，我们使用 UML 用例图和活动图来描述这些基本的业务流程。为了发现实际商业活动中可能的发展方向，提高可重用服务划分的准确性，在对具有相似性质的企业实地调研后，我们修正并添加了一些实际商业活动中已存在的或者可能出现的流程，并使用修正后的业务流程活动图作为我们开发的输入进行服务模型的设计。

## 服务发现及其依据

### ➤ 场景和案例

在凤凰公司所描述的业务背景中，概括地说，我们需要关注的角色就是财务人员、销售

人员和公司高层顾问，所要关注的平台是他们各自应用的 ERP、CRM 及相应数据库。高层顾问这个角色是我们在调研过程中了解到订单的分级审批制度后预设的一个角色，它在目前的系统中可以与财务人员合并，但是在未来系统完善扩展至有市场方向、业绩考评等的辅助决策信息的提取后，该角色可分离成为服务的消费者之一，享受如前所述的服务。业务流程需求分析见后，在这里不再详述。基于这些需求和场景，我们列举出所有的案例，这里仅举例如下：

**P1:** 用户身份认证

**P2:** 通知用户身份认证成功/失败

**P3:** 浏览产品信息

**P4:** 产品信息检索

**P5:** 通知管理员系统异常

**P6:** 提交产品信息

**P7:** 通知用户产品信息更新成功/失败

**P8:** 订单审批

- ◆ 客户身份验证
- ◆ 创建新客户
- ◆ 修改订单数据
- ◆ 生成新订单
- ◆ 通知销售人员订单创建成功
- ◆ 写数据入 CRM 数据库

**P9:** 提交业务状态

**P10:** 等等

为了充分说明上述案例，以 1 为例具体说明如下：

**P1: 用户身份认证**

**Actor:** 操作人员

**前提:** 门户和业务集成服务器准备好并处于运行状态

**主要成功场景:** 操作人员登陆门户服务器→系统显示处理页面及提示信息→操作人员输入正确信息并确认→系统执行 P2→系统显示“认证成功”

**扩展:** 输入信息不正确，系统显示“认证失败”并返回处理页面

其它案例与此类似。

## ➤ 业务流程需求

通过讨论题目对业务流程的描述和实地调研,我们通过业务分析确定了如下一些核心活动:

**A1:** 服务消费者身份认证

**A2:** 产品信息浏览

**A3:** 产品信息更新

**A4:** 标志业务状态

**A5:** 业务提交

**A6:** 客户信息有效性验证

**A7:** 订单审批

**A8:** 确认信息通知

这些关键业务实施的目标是使得系统在运行之后能够自动快速地完成从发现业务机会到成功创建订单这个过程。同时,对于未来可能采用的可视化信息聚合方式、网上社区功能使用和高层管理参考数据业务等扩展实现快速灵活的开发。

## ➤ 业务流程建模

基于前一阶段成果,我们使用 Rational Rose 对业务进行了用例建模并绘制了活动图,获得了业务模型。它是实际的复杂业务的简化视图及业务如何运转的提取,给出了系统的内部组件的交互特性。同时为了更好地适应服务规范,我们在 Rose 图的基础上又创建了 BI Modeler 的组件交互图。业务流程活动图详见系统结构和组件部分文档中的描述,这里给出使用 WBI Modular 绘制的其中的一个非常简单的流程——客户信息校验来说明:

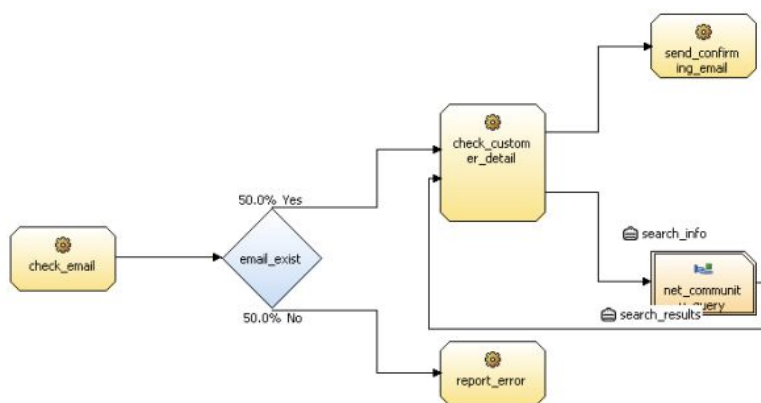


图1 客户信息校验流程

此流程属于订单审批下 ERP 子流程中。又如：

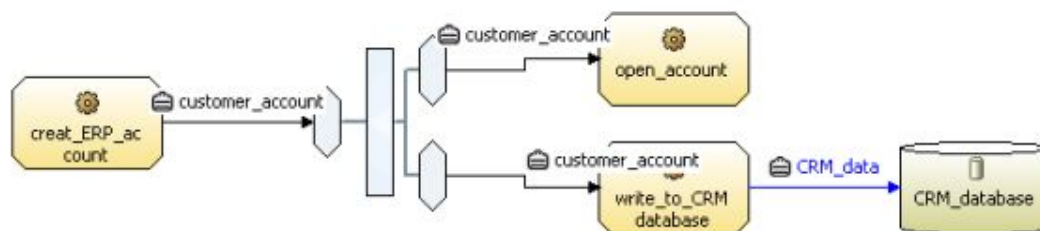


图2 创建新客户流程

这个创建客户流程属于上述客户信息校验流程的上一层面，它描述的是订单审批中的一步：预先创建一个账户，输入信息并校验，如果正确，确认创建账户，并继续订单审批的下一个子流程；否则错误，取消创建账户。可以看到，这类活动图中包含了流程的操作者、数据流向、关键决策等很多信息，能够帮助我们抽取一定的服务。

## ➤ 服务的确定

实际操作中，抽取服务是一项不简单的工作，原因在于没有固定的合适的标准。我们在操作时是粗细结合、与流程相一致、交互进行的。首先，对应于活动图的泳道视图，我们尝试找出每一个组织单元的出入流并且将它们定义成为候选的服务。在这种方式下，已经依照核心业务功能将任务分配给特定的组织单元，所以通过组织单元传递的信息都代表内部组件的交互，可以作为候选服务。同时，我们考虑了企业内部原有信息平台 ERP 和 CRM 系统中的数据服务和贯穿整个系统的安全及工具服务。如此我们得到服务清单如下：

1. 规则服务：系统所要用到的各种大大小小的规则都由这项服务来提供，比如：不同身份权限的定义，分级审批制度，更新ERP中哪些种类数据到CRM中等等。
2. 消息格式变换服务：是完成ERP和CRM系统交互的必需。这项服务保证了各个不同系统之间的接口一致。
3. 安全服务：使用数据加密等技术完成对流程安全性的保护以及其它。这是一项贯穿各个层面的服务。
4. 身份认证服务：对登陆使用者的身份进行确定，以便提供相适应的权限。
5. 日志服务：类似于备忘录的一个实时提醒组件，并且可以被用来传递简单的消息。
6. 短信服务：基于电信运营商平台，功能同上，主要用于各业务操作的确认。

7. 数据访问服务：在原有系统之外对数据库操作的集合。
8. 数据库状态监视服务：这项服务用于在ERP更新后，获得一个“已经更新”的信息，并且自动比对所有冻结订单。
9. CRM功能调用服务：所有CRM系统提供的服务功能均在其中。
10. ERP功能调用服务：所有ERP系统提供的服务功能均在其中。
11. 网上社区查询服务：把客户的信息作为查询关键词，在智能社区上搜索相关信息。
12. 事务性服务：常见的保护数据库的一些机制，比如在写数据库内容时禁止读等等。
13. 数据库同步服务：使用到规则服务和消息格式变换服务，调用此服务可以将ERP数据库中被规则服务指定的信息内容更新到CRM数据库中。
14. 员工状态服务：提供给ERP和CRM系统的查看企业内部人员工作状态的服务，如：繁忙、空闲、不在等，便于CRM业务请求的分配，使之可以被及时处理而不会因为员工的原因被耽误。
15. 数据库挖掘服务：提供内嵌的数据挖掘算法，使管理级的人员可以通过系统进行辅助决策，为将来更高级的决策辅助类服务提供基础。
16. 产品浏览服务：这一服务对销售人员、合作企业以及客户/潜在客户开放，对不同身份的角色提供不同的详细程度的产品信息以及不同的参考价位。  
**服务组合**：身份认证服务→规则服务→事务性服务→数据库访问服务
17. 产品信息提交服务：这一服务所实现的功能就是在更新ERP中产品信息的时候同步更新CRM中的数据，并对以冻结订单的状况作检查。  
**服务组合**：ERP 功能调用服务[提交产品变更请求]→ERP 功能调用服务[ERP 数据更新]→数据库状态监视服务→日历、短信服务→数据库同步服务
18. 业务成功提交服务：这一服务所实现的功能就是提交业务清单并获得成功确认。  
**服务组合**：CRM 功能调用服务[提交业务清单]→消息格式变换服务→ERP 功能调用服务[库存管理] →ERP 功能调用服务[提交订单]→员工状态服务→日历服务→短信服务→事务性服务→数据访问服务→消息格式转换服务→CRM 功能调用服务[业务提交成功]
19. 客户校验服务：认证客户的E-mail是否存在，发送确认函，并调用网上社区查询服务人工对客户的信息进行对比。
20. 订单审批服务：这一服务所实现的功能就是由财务人员对销售人员提交的订单进行审核。

**服务组合：**ERP 功能调用服务[客户存在否]→客户校验服务→ERP 功能调用服务[创建新客户]→ERP 功能调用服务[批准订单/拒绝批准]→日志, 短信服务→数据访问服务(前提: CRM 数据库相应项标记为“以批准”)

21. **业务状态服务：**这一服务所实现的功能就是销售人员获得业务机会后修改业务直到状态为赢, 提交并等待财务人员审核的流程。

**服务组合：**CRM 功能调用服务[客户存在否]→网上社区查询服务→CRM 功能调用服务[创建新客户]→CRM 功能调用服务[生成业务清单]→产品浏览服务→CRM 功能调用服务[修改业务清单]→CRM 功能调用服务[修改业务状态(为赢)]→CRM 功能调用服务[提交业务清单]

### ➤ 创建服务模型

创建服务模型时, 我们首先考虑将所有的候选服务, 即服务清单中所列服务进行映射和分类。它们的用法和调用关系如图所示:

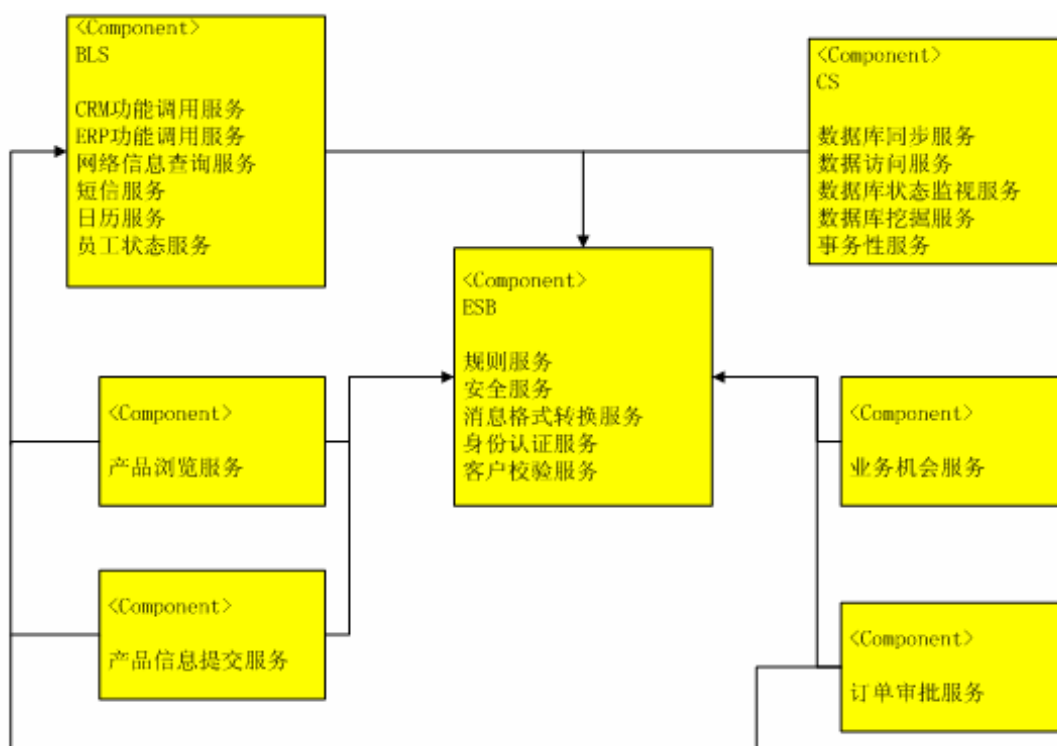


图 3 服务映射与分类

接下来我们确定哪些服务是可发布的。在发布服务时, 服务的粒度是最需要考虑的问题之一。划分服务粒度的时候, 我们参考了比较流行的粗粒度、细粒度和组合服务这样的划分形式。划分时考虑如下三点因素: 服务的功能、服务需要承受的数据流量和服务与外部的接

口属性。比如：服务消费者无法接触到的服务全部划归为细粒度服务，而组织单元的出入流则是粗粒度的最佳备选。具体说来，我们由细到粗编排出下列服务：

- 细粒度服务：S1～S15
- 粗粒度服务：S16～S19
- 组合粒度服务：S20 和 S21

以 S1 为例，由于规则服务与外部用户和系统是没有接口的，它负责为一些粗粒度的服务提供执行规则，从功能上看也比较单一，数据流量小，因此将它作为细粒度服务。在某些政策发生调整之后，比如税务审批制度的变化等等，需要变更业务流程时，只需要修改此规则服务即可，而其它服务不需要修改。这就是比较典型的服务组件的可替换性。

最终，对于题目要求的这一完整流程我们得到如下四个业务和相应的组件及服务，即服务模型。

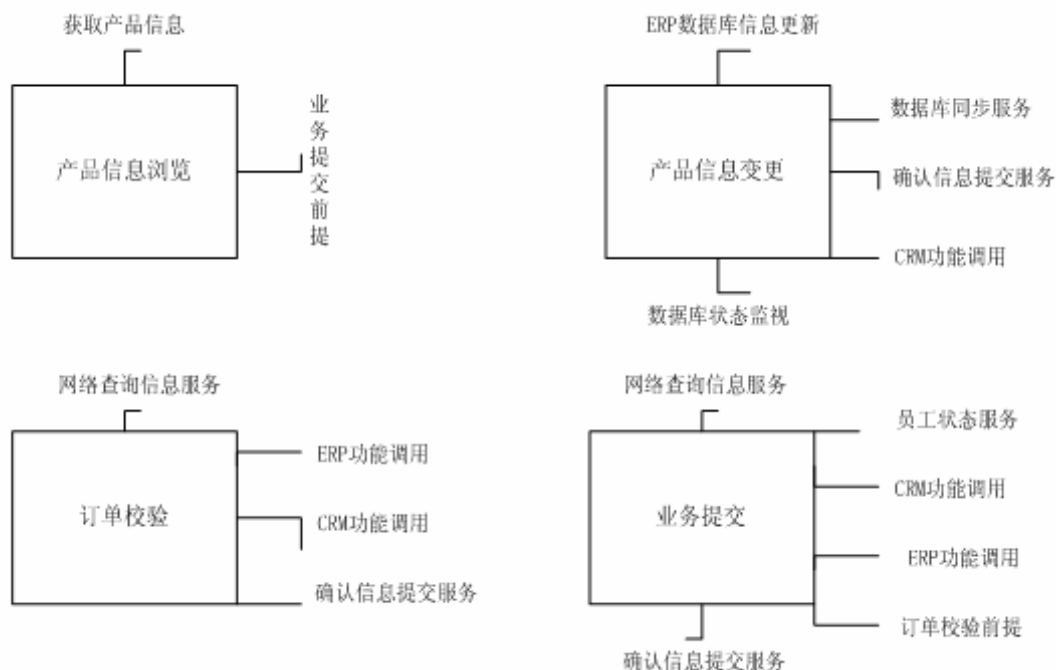


图 4 核心业务的服务模型

## 服务规约

所谓服务规约，就是对上述讨论的服务在操作上所作的约束和说明。在这个项目中我们主要关注服务的接口来解释服务规约。输入输出的数据类型可参考系统架构及组件设计中有关数据格式的描述。

表 1 服务接口的描述

服务名称	服务描述	输入	输出	函数列举
Calendar	<p>日历服务主要完成如下服务：</p> <p>1'把消息在规定的日期显示出来</p> <p>2'即时显示 instant message, 作为简易的即时通信工具</p> <p>3'为每个消息标注 state, 只有当 state= already_read &amp;&amp; date&lt;today's date 才可以删除消息, 否则 instant=true &amp;&amp; date=today's date</p> <p>4'收到消息后, 如果 ID 存在 &amp;&amp;date&gt;=today's date 则返回 success_info&amp;&amp;把消息存入数据库, 否则丢弃消息并返回 fail_info</p>	Calendar_message	Success_info Fail_info	<p>void trans_msg(calendar_message)</p> <p>void send_msg(int id,datetime data,string msg,Boolean instant)</p>
Data_accesses	<p>用通用的数据库操作语言对分布在网络中的各个数据库进行各种操作的服务, 用户只需要指出想访问的数据库名字以及所要进行的操作。</p> <p>特别注意点: 此服务内部需要调用 ruler 服务, 对所有操作的操作者进行资格省查, 对销售人员禁止其访问 ERP_database, 仅当其调用 bill_into_ERP 组件时允许组件对 ERP_database 的添加订单操作并强制订单状态为 freeze</p>	Data_request	CRM_data ERP_data	<p>void trans_meg(data_request)</p> <p>void data_access(string databaseID,string SQL)</p>
Data_mining (待拓展服务)	<p>此服务通过内建的关联规则和算法, 对销售信息和客户信息进行挖掘, 找出其中的关联项, 用以发现商机和辅助决策, 同时亦可以用来作为对销售人员的业绩、效率、商业</p>			



	机会把握的成功度和财务人员处理订单的时间等进行考核时的依据			
Data_monitor	此服务被 ERP_data 的改动触发, 自动核对 ERP_database 和所有状态为 freeze 的订单, 如果有库存满足订单的条件, 则解冻订单, 发送解除冻结状态订单的 ID	ERP_data	Bill_changed_info	void check_bill()
Database_synchronization	此服务由 ERP_data 的改动触发, 对改动的部分调用 ruler 服务判断是否需要写入 CRM_database, 如果需要, 则将改动部分更新到 CRM 数据库, 实现 ERP 和 CRM 数据库中相关内容的一致性	ERP_data	CRM_data	void data_access(string SQL)
Identification	通过对输入的用户名和密码的确认返回用户所属的组或角色信息, 建立 session, 实现单点登陆。	User_info	Role&group_info	string trans_id(string name, string password)
Message_router	此服务使同一局域网中跨平台的消息传递可以忽略细节特征, 只需要提供目标地址, 己方地址以及消息内容, 服务即自动把消息转换成目标系统的格式并发送出去	Message_info	Calendar_message SMS_message ERP_bill_info Data_request Success_info_for_CRM CRM_bill_info	void trans_msg(message_info e)  void trans_form(int send_address, string msg_type, string msg_content)  void send_msg(int send_address, string msg_content)
Net_community_query	使用户无需关心是在哪个网上社区查找或是登陆社区的用户密码之类具体信息, 直接进行查询工作	Search_info	Search_results	void send_msg(string search_info)
Ruler	对输入的用户信息和操作信息在规则列表中查询后返回用户信息、操作	Rule_info	Rule_results	void trans_msg(rule_info)

	信息和允许信息			<pre> void check_rule(string   jobtype,string   role_info) rule_results return_results(stri ng jobtype,string rule_info,Boolean permit_info) </pre>
Security	<p>对需要在 internet 上传输的敏感信息按安全协议进行加密处理，实现端到端的安全连接，一般使用现有的各种安全协议和算法。</p> <p>注：这一服务和系统架构的安全服务概念不同，系统架构中的安全服务主要是通过规则服务实现的</p>	ID&selection	ID&selection	
SMS	<p>此服务将消息发送到电信营运商的服务器上，若成功则返回 success_info，否则返回 fail_info</p>	SMS_messa ge	Success_info Fail_info	<pre> void send_msg(SMS_ message e) </pre>
State_moni tor	<p>此服务监视员工的在线状态，可分为空闲，忙，暂离，未登陆等，对于等待处理的帐单，通过接收处理帐单人员所需要的等级，自动把任务分配给对应等级中按 空闲&gt;忙、暂离&gt;未登陆 排列的人员。这一服务不但加强了效率，更重要的是使销售人员和财务人员无法一一对应，降低了两者串通后引发事故的风险</p>	Ask_for_idle ss	Worker_rem ain_idless	<pre> void check_idless()  int select_worker() </pre>
Products_o verview	<p>浏览产品信息，通过规则服务使得不同的角色（销售人员、客户、合作伙伴...）可以对同一商品获得不同得信息，如不同详细程度的介绍和不同的</p>	Viewer_info	Overview_re sults_info	

	价格			
Products_changed_information	财务人员通过这个流程改变库存信息，此流程会自动比较这些库存变动并解冻相应满足条件的订单，同时通知提交这些订单的销售人员	Viewer_info	Success_info Fail_info	
Bill_submitted	销售人员从发现业务机会到提交业务清单的全过程	Viewer_info	Success_info Fail_info	
Bill_checked	财务人员审核订单的全过程	Viewer_info	Success_info Fail_info	

## 服务实现分析

在现阶段，我们对服务实现分析的理解就是 IT 环境分析，即分析系统现有应用技术的特点，由此为服务实现和系统架构设计提供一些技术依据。

很容易发现，原有的体系是一个拥有大量异构系统的集合，涉及到 ERP、CRM、IISC 和网络短信平台这四个系统。它们在核心业务流程的实现过程中，有着如下的交互关系：

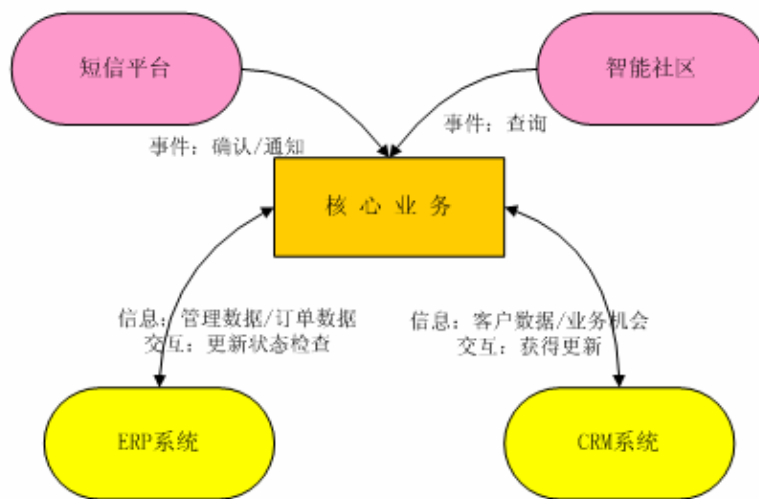


图5 IT系统与核心业务间的交互

此图可以看作是一个简化的 IT 环境视图，它描述了核心业务流程和已有四个系统间的交互状况。目前需要的交互包括这样几种：

- 企业内部工作人员从短信平台上获取确认/通知信息。
- 财务人员（或其它有权限人员）从智能社区中获取客户身份信息。
- 财务人员向 ERP 系统发送更新数据，收到确认信息；订单审核过程从 ERP 系统获

得各种数据/订单信息等。

- 销售人员使用 CRM 系统查询产品/客户信息、提交业务状态数据；CRM 系统获得数据更新，与 ERP 系统同步。

对于提到的四个系统而言，短信平台和智能社区都拥有相对开放的技术，它们可以较好地支持整合，不会使系统遇到过于困难的问题。而 ERP 和 CRM 运行在比较独立的主机上，如何在保证安全性的前提下使得这两个系统的信息按照一定的规则互相交互是我们需要解决的问题。具体的实现机制分析请见系统架构和组件设计第三部分。

## 结束语

在服务模型分析设计这部分内容中我们主要介绍了是如何通过对业务流程建模逐步获得服务模型的。在本次比赛中，由于我们希望能够更好地在企业整合中体现 SOA 的思想，真正做到以服务为中心，所以对服务获取和依据这部分下了很多的功夫，也考虑了不少的创新点，可参看项目综述部分的说明。但是相比较而言，在组件建模和设计方面，由于时间安排上的一些失误，没有前一部分做得详细，有些细节还需要讨论和细化，尤其是实际系统上的一些接口。这是需要再详加考虑的问题。

## Reference

- 以服务为中心的企业整合
- 以服务为中心的企业整合——案例分析
- 按需业务流程生命周期系列文章
- SOA 面向服务的业务转换在零售业中的最佳实践