

## 第二部分 服务模型分析与设计

【内容提要】这部分内容作为面向服务建模与架构（SOMA）分析的开始，基于企业业务模型的分析与需求分析，对系统进行领域分解和目标服务模型分析，发现所需要的服务，并对其进行规约与实现分析。目的在于引入 SOA 思想，为架构人员提供支持。

### 1. 服务建模简述

下面简要介绍服务建模的概念，以及在服务建模中所遵循的 SOMA 方法。

#### （1）服务建模目的

SOA 中最重要的一个概念，就是服务。在 SOA 架构风格中，业务被组件化为“Human Task”，“Rule Group”，“State Machine”，“Process”，……这些服务都是粗粒度的，他们可以根据粒度划分为业务服务和业务流程。业务服务是独立的，自包含的，可重用的，业务流程是由业务组装而来的。给组件包装上接口，便可得到 SOA 中的服务。服务接口中默认会有一个输入输出，用户可以根据需求设定输入输出的个数以及输入输出的数据类型（SOA 中的数据类型通常为 Business Object）。一个服务除了定义与其业务功能相关的接口以外，还定义了约束该接口的契约，如服务质量要求，业务规则，安全性等等。

SOA 建模的目的就是将业务组件化，实现从业务领域跨越到 IT 领域，达到 IT 与业务对齐的目的。业务建模以后，得到一个由业务活动集组成的业务流程。这些业务活动的子集可以提取为服务，也就

是实现业务与 IT 对齐。所提取出来的服务通常是不能够满足特定的业务目标的，所以常常需要增加一些服务。服务建模以后，得到一个服务集，这些服务可以根据不同的业务目标来重新组装，也可以重用这些服务来实现多个较高级别的业务流程当业务目标改变以后，也可以随时添加所需要的服务或者去掉某个不需要的服务，从而实现业务的灵活性。

SOA 主要是为了支持业务的快速变化，所以做好 SOA 必须做变化分析。根据变化可以将业务领域中易变的部分和稳定的部分化分开，稳定部分提取为服务，不稳定部分提取为业务规则。通过将易变的业务逻辑及相关的业务规则剥离出来，保证未来的变化不会破坏现有设计，从而提升架构应对变化的能力。

## （2）SOMA 简述

众所周知，面向对象的应用构建在类和对象之上。

随后发展起来的建模技术将相关的对象按照业务功能进行分组，就形成了组件的概念；对于跨组件的功能调用，则采用接口的形式暴露出来。

进一步的将接口的定义与接口的具体实现进行解耦，就催生了 SOA。而作为业务和 IT 之间的契约的服务，是 SOA 最重要的概念。

因此面向对象、基于组件、面向服务是三个递进的抽象层次。

现在我们有 OOAD (Object Oriented Analysis Design) 和 CBD (Component Based Development) 来进行面向对象和基于组件的建模与开发。但是没有一个好的方法来进行 SOA 的分析、设计和开发。

SOMA (Service Oriented Modeling Architecture) 就是在这个背景下诞生的,其主要目的就是填补 OOAD 和 CBD 在建模领域留下的空白,为 SOA 实施提供一个方法学的指导。

需要特别指出的是, SOMA 的出现并不是要替代 OOAD 或者 CBD,正如 CBD 需要借助 OOAD 一样, SOMA 也要借助 OOAD 和 CBD 进行实现层面的建模。同时 SOMA 也是可供大家剪裁使用的,也不需要一定绑定于某个厂商的特定工具之上。

与 OOAD 和 CBD 相比较而言, SOMA 贯穿整个 IT 建设的生命周期,在项目规划、设计、实施、运行中都起到重要的作用。本文就不展开阐述了,相关信息可见参考资料。

SOMA 另外一个显著的特点就是将 IT 与业务对齐。在具体的实施过程中, SOMA 将业务特性,如:业务目标、关键业务指标等,延伸到 IT 的分析和架构决策过程,从而缩小业务与 IT 之间的差距。具体来看,业务组件模型(或者类似业务分析方法论的结果)、端到端的业务流程以及关键业务指标是 SOMA 的三项主要输入, SOA 的实现则是 SOA 的输出,从这也可以看出 SOMA 的定位是在业务和 IT 之间。

按照实施的阶段, SOMA 分为服务发现、服务规约以及服务实现三个阶段。

#### 1) 服务发现

通常采用自上而下、自下而上和中间对齐的方式,得到服务的候选者。

- 自上而下 (业务领域分解) 方式

从业务着手进行分析，将业务进行领域分解、流程分解，以及进行变化分析。

业务组件模型（CBD）是业务领域分解的输入。根据业务组件模型的详细描述，我们可以将业务领域按照业务职责细分为业务范围，并直接其映射到 IT 范畴的子系统，实现业务与 IT 的无缝连接。

顶级的业务流程是流程分解的输入。将业务流程分解成子流程或者业务活动，逐级进行，直到每个业务活动都是具备业务含义的最小单元。流程分解得到的业务活动树上的每一个节点，都是服务的候选者，构成了服务候选者组合。在大部分情况下，服务候选者组合都是一个很长的列表，加上自下而上和中间对齐方式还有可能发现新的服务，因此将服务候选者按照某种方式进行分类是一件非常必要的事情。业务领域分解的结果——业务范围是一个业务概念，同时可以无缝映射到 IT 范畴，因此它是一个好的分类原则。根据业务范围，服务候选者组合可以被划分服务候选者目录。

变化分析的目的是将业务领域中易变的部分和稳定的部分区分开来，通过将易变的业务逻辑及相关的业务规则剥离出来，保证未来的变化不会破坏现有设计，从而提升架构应对变化的能力。变化分析可能会从对未来需求的分析中发现一些新的服务候选者，这些服务候选者需要加入到服务候选者目录中。

- 自下而上（已有资产分析）

目的是利用已有资产来实现服务，已有资产包括：已有系统、套装或定制应用、行业规范或业务模型等。

通过对已有资产的业务功能、技术平台、架构以及实现方式的分析，除了能够验证服务候选者或者发现新的服务候选者，还能够通过分析已有系统、套装或定制应用的技术局限性尽早验证服务实现决策的可行性，为服务实现决策提供重要的依据。

- 中间对齐（业务目标建模）

目的是帮助发现与业务对齐的服务，并确保关键的服务在流程分解和已有资产分析的过程中没有被遗漏。

业务目标建模将业务目标分解成子目标，然后分析哪些服务是用来实现这些子目标的。在这个过程中，为了可以度量这些服务的执行情况并进而评估业务目标，我们会发现关键业务指标、度量值和相关的业务事件。

结合这三种方式的分析，我们发现服务候选者组合，并按照业务范围划分为服务目录。同时为服务规约做好其他准备，如：通过对已有资产分析进行的技术可行性评估、通过业务目标建模发现的业务事件等等。

## 2) 服务规约

定义实现服务的服务组件的细节，包括，数据、规则、服务、可配置概要、可能的变更，同时还会涉及到消息、事件的定义和管理。

经过服务发现的阶段，我们得到了候选服务目录，接下来就需要决定暴露哪些服务。理论上所有的服务候选者都可以暴露为服务，但是一旦暴露为服务，该服务候选者就必须满足附加的安全性、性能等方面的要求，企业还必须为服务的规划、设计、开发、维护、监管支

付额外的开支，因此我们会根据一定的规则来决定将哪些服务候选者暴露为服务。

这些规则包含以下几个方面：

- 业务对齐：该服务候选者可以支持相关的业务流程和业务目标；
- 可组装：该服务候选者满足技术中立、自包含以及无状态等特点，同时还满足复合应用的相关非功能性需求；
- 可重用：该服务候选者可以在不同的应用、流程中重用，从而减少重复的功能实现，降低开发和维护的成本。

基于企业应用开发的经验，我们还可以有其他一些方面的考虑。在决定暴露特定的服务候选者为服务以后，服务规约还需要定义服务的消息、非功能性需求以及服务之间的依赖关系、组合关系。

### 3) 服务实现

根据对业务领域的理解和现有 IT 系统的分析，将服务的实现分配到相应的服务组件，并决定服务的实现方式。具体的实现方式，可以由已有系统暴露相关功能为服务，或者重新开发相关功能提供服务，也可以由合作伙伴来提供服务。

## 2. 服务模型分析设计

### (1) 服务发现极其依据

服务发现是 SOMA 进行服务分析和设计的第一步，它的主要任务是确定在一定范围内可能成为服务的候选列表。

本项目采用的服务发现步骤如表 2-1 所示。

表 2-1 服务发现步骤

编号	步骤	详细描述
1	自顶向下分析（通过业务流程进行服务建模）	将业务流程描述清楚之后，我们需要分析已有的 Test 和 Process， 将可以提取为服务的发布为 Global Service。
2	自底向上分析（对遗留系统分析 ）	分析遗留系统，找出可以复用的模块或者组件。为服务提取做准备。
3	中间对齐（业务目标建模）分析	通过以上两种分析，所提取的服务还不足以满足特定的业务目标，我们需要生成新的服务。

1) 域分解（自顶向下）

从业务角度而言，域是功能块的集合，通过域分解来对功能块进行划分来发现服务的候选。下面是本系统的域分解图，参见图 2-1。

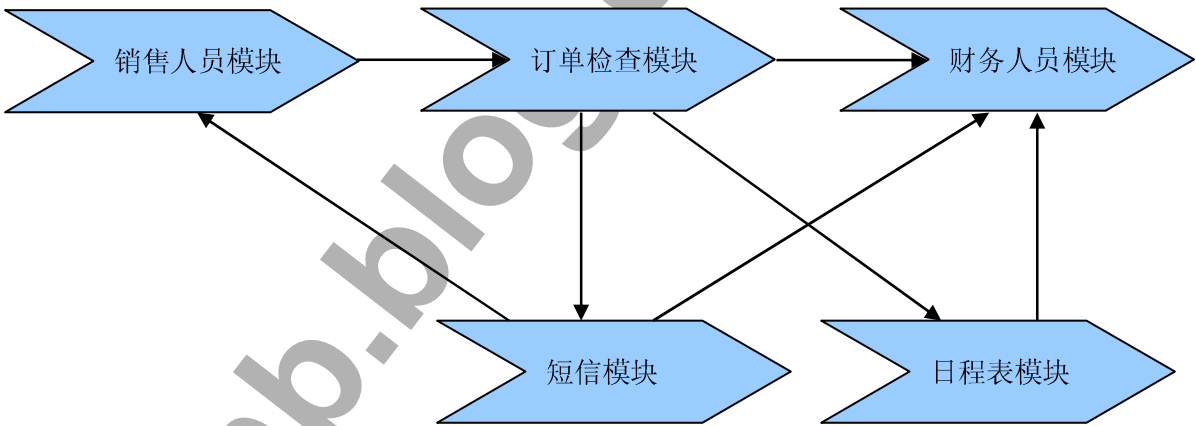


图 2-1 本系统的域分解图

我们采取自顶向下分解业务的方法来抽取服务。对于该业务流程，我们将企业相关业务进行业务活动的向下分解，直到每一个业务

活动都能够清楚进行描述为止。最后抽取出来的候选服务通过域分解，可以得到粗粒度的服务候选如下：

- U1：销售人员
  - U1.1：销售人员生成业务机会
  - U1.2：销售人员维护业务机会
  - U1.3：销售人员发送订单请求
  - U1.4：销售人员重新修订订单并且重新提交订单
  - U1.5：销售人员废除订单
- U2：订单检查
  - U2.1：判断业务机会状态
  - U2.2：订单请求检查
  - U2.3：订单冻结
  - U2.4：监视库存是否满足
- U3：财务人员
  - U3.1：财务人员审核订单
  - U3.2：财务人员生成销售订单
  - U3.3：财务人员输入修改意见并且打回订单
  - U3.4：财务人员修改订单并且打回订单
- U4：短信
  - U4.1：给财务人员订单处理通知
  - U4.2：给销售人员订单冻结通知
  - U4.3：给销售人员重新提交订单通知
  - U4.4：财务人员给销售人员的订单打回通知
  - U4.5：给销售人员发送订单请求通过通知
- U5：日程
  - U5.1：日程服务

通过以上域分解，获得了最粗粒度的服务候选，这些服务候选都包括在顶级流程中，如图 2-2 所示。



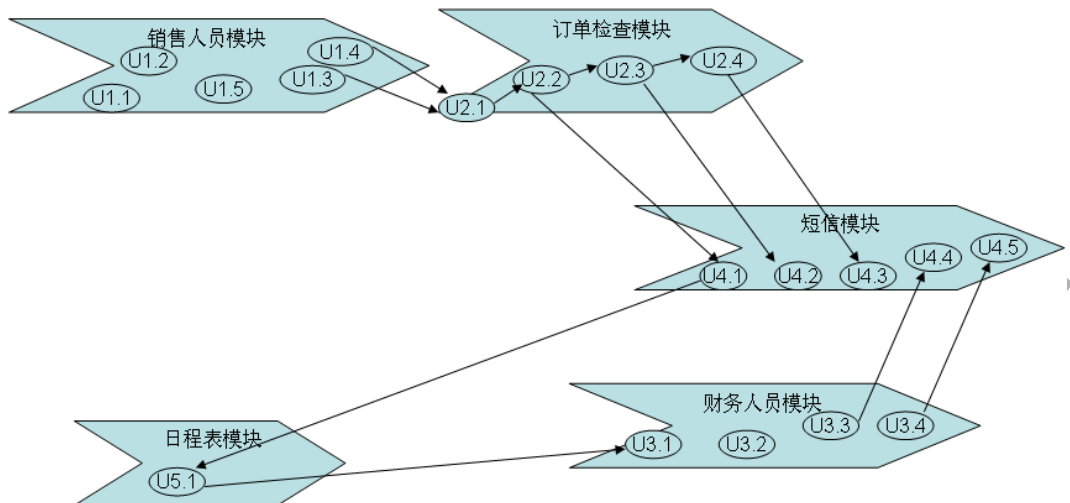


图 2-2 包括在顶级流程中的粗粒度的服务候选

## 2) 业务目标服务发现

在这个步骤中，我们通过业务需求和业务目标分析，业务目标建模来增加服务候选。我们的步骤是，首先提出主要目标，分解主要目标为子目标来实现这些主目标，通过增加服务候选来实现这些子目标进而来实现主目标。

在我们的案例中，企业的目标是增加企业收入，要达成这个目标，我们的首选方法是：创建更多的业务机会，除此之外，还需引入两个方法，一个是商机挖掘，提供了商机挖掘生成预业务机会供销售人员确认服务；另一个是服务社区，提供了信息订阅，信息定制，信息发布等功能，参见 2-2。

表 2-2 业务目标服务发现

业务目标、达成方法	服务抽取
创建更多的业务机会	<ul style="list-style-type: none"> <li>✧ 获取客户信息</li> <li>✧ 获取产品信息</li> </ul>

	◇ 自动创建业务机会 ◇ 增加订单类型
商机挖掘	◇ 商机挖掘生成预业务机会供销售人员确认
服务社区	◇ 信息订阅 ◇ 信息定制 ◇ 信息发布

通过业务目标的分析，所得到的服务候选（黑体的服务为新增的服务候选）如下：

- U1：销售人员
  - U1.1：销售人员生成业务机会
  - U1.2：销售人员进行商机确认生成业务机会**
  - U1.3：判断客户是否存在于 CRM 系统中**
  - U1.4：销售人员创建客户信息
  - U1.5：销售人员浏览产品信息**
  - U1.6：销售人员获取客户信息
    - U1.6.1：从 CRM 中获取客户信息**
    - U1.6.2：向服务社区模块订阅客户信息**
  - U1.7：销售人员维护业务机会
  - U1.8：销售人员发送订单请求
  - U1.9：销售人员重新修订订单并且重新提交订单
  - U1.10：销售人员废除订单
  - U1.11：判断业务机会状态
- U2：订单管理
  - U2.1：订单请求检查
    - U2.1.1：判断是否为订货型订单**
    - U2.1.2：判断库存情况**
  - U2.2：订单冻结
  - U2.3：监视库存是否满足

- U3: 财务人员
  - U3.1: 财务人员审核订单
  - U3.2: 财务人员生成销售订单
  - U3.3: 财务人员输入修改意见并且打回订单
  - U3.4: 订单不通过, 财务人员直接打回订单
- U4: 短信
  - U4.1: 给财务人员订单处理通知
  - U4.2: 给销售人员订单冻结通知
  - U4.3: 给销售人员重新提交订单通知
  - U4.4: 财务人员给销售人员的订单打回通知
  - U4.5: 给销售人员发送订单请求通过通知
- U5: 日程安排
  - U5.1: 日程表服务
- **U6: 商机挖掘**
  - U6.1: 商机挖掘生成预业务机会供销售人员确认**
- **U7: 服务社区**
  - U7.1: 信息订阅**
  - U7.2: 信息定制**
  - U7.3: 信息发布**

用这些服务候选分解模块如图 2-3 所示。

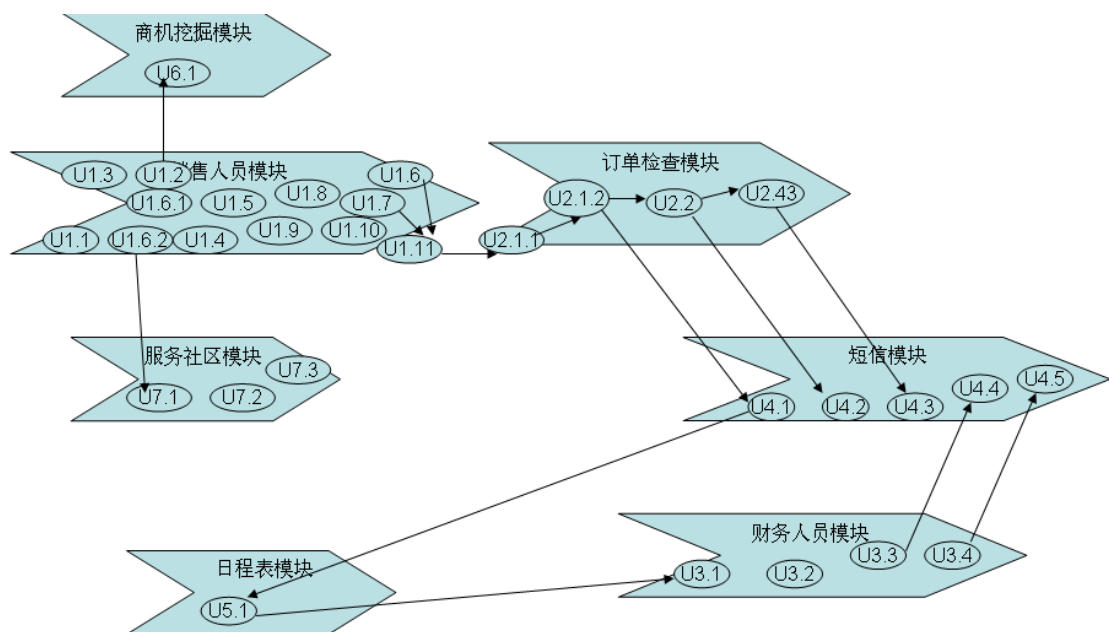


图 2-3 包括在顶级流程中的粗粒度的服务候选

### 3) 子系统分析

经过上面的分析，得到了各个功能模块之间进行业务交互的关系。下面将对业务用例进行进一步精化，成为描述业务过程的系统用例。同时对子系统进行分析，划分出其中的业务模块和技术模块。业务模块的发现，主要是通过对子系统内的业务流进行分析，而技术模块主要是通过对非功能需求的分析得到。

根据以上分析，得到了七个模块，每一个模块将和一个子系统相对应如下：

① 销售人员模块，销售人员进行相关业务机会操作并提交订单等，和遗留系统—CRM 系统相对应。

② 财务人员模块，财务人员将对销售人员的订单请求进行处理，和遗留系统—ERP 系统相对应。

以下模块没有遗留系统，都需要进行开发：

③ 订单检查模块，对销售人员的订单请求进行检查，需要新开发系统。

④ 短信模块，进行短信通知

⑤ 日程表模块，给财务人员提供日程表服务。

⑥ 商机挖掘模块，挖掘潜在的业务机会。

⑦ 服务社区模块，提供社区上的客户信息

每一个子系统都可以由两部分组件组成，分别由业务组件和技术组件，业务组件提供了业务功能服务，例如创建业务机会等等，参与到了业务流程中，而技术组件提供非业务功能服务，其帮助作用，列入提供客户信息目录等等。以下将对每个子系统进行分析，目的是找出遗漏的服务候选。

#### ● CRM 系统分析：

首先是业务组件，其过程图表示如图 2-4 所述：

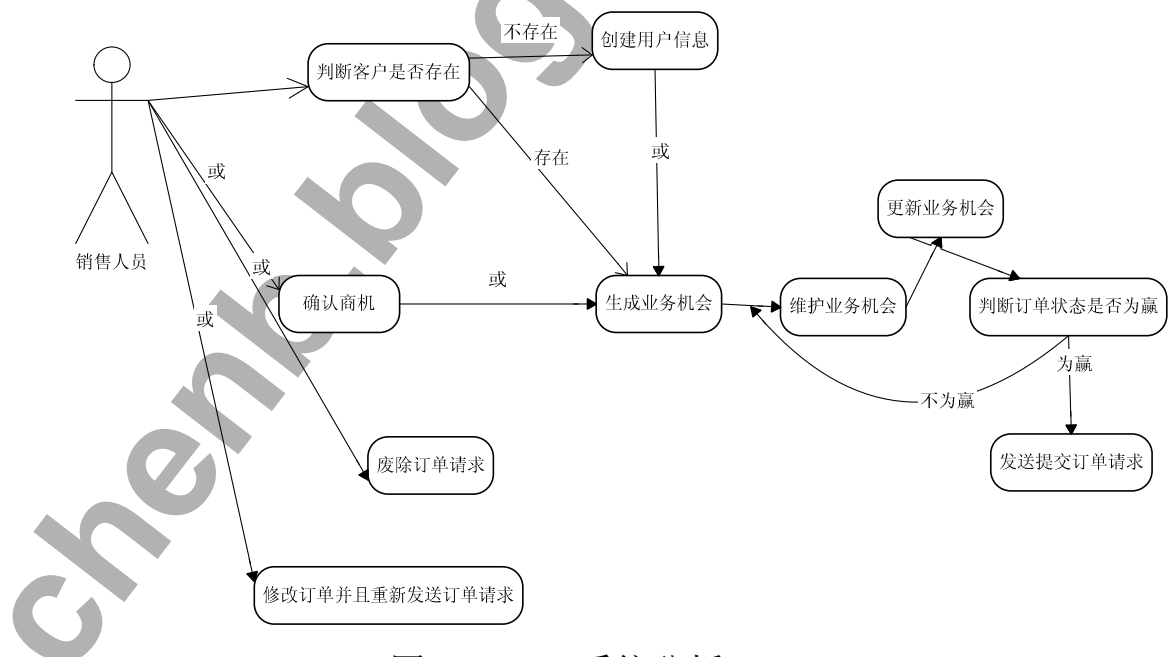


图 2-4 CRM 系统分析

在业务组件分析后加入了

① 更新业务机会

在技术组件中新加入的服务候选为：

① 展示产品列表

② 查看某个产品详细信息

③ 查看客户基本信息

④ 查看客户历史交易信息

⑤ 查看订阅的客户信息

● ERP 系统分析

业务组件分析，见图 2-5

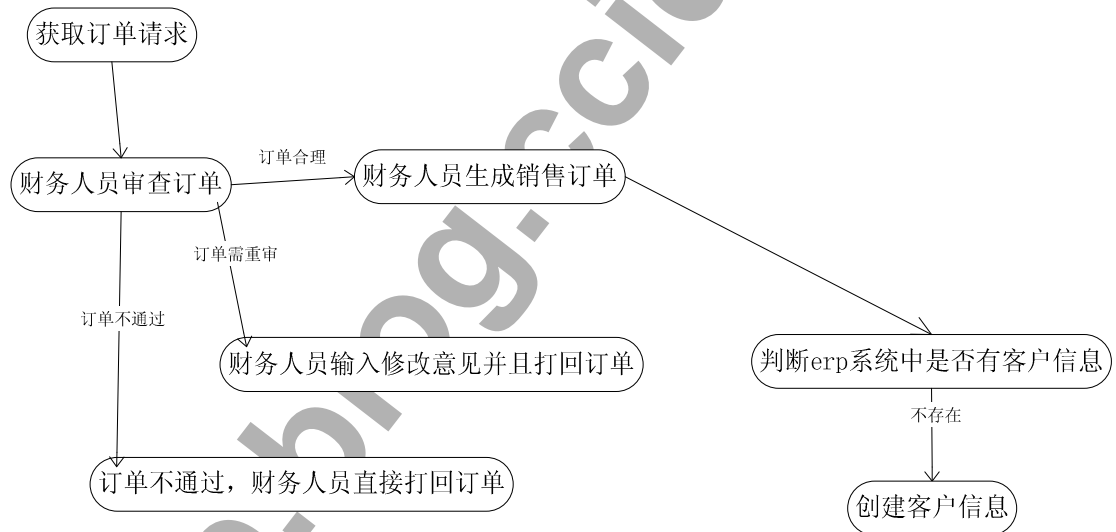


图 2-5 ERP 系统分析

由此分析，可增加服务候选为：

① 判断 ERP 系统中是否有客户信息

② 创建客户信息

③ 获取订单请求

● 订单检查模块分析

见图 2-6.

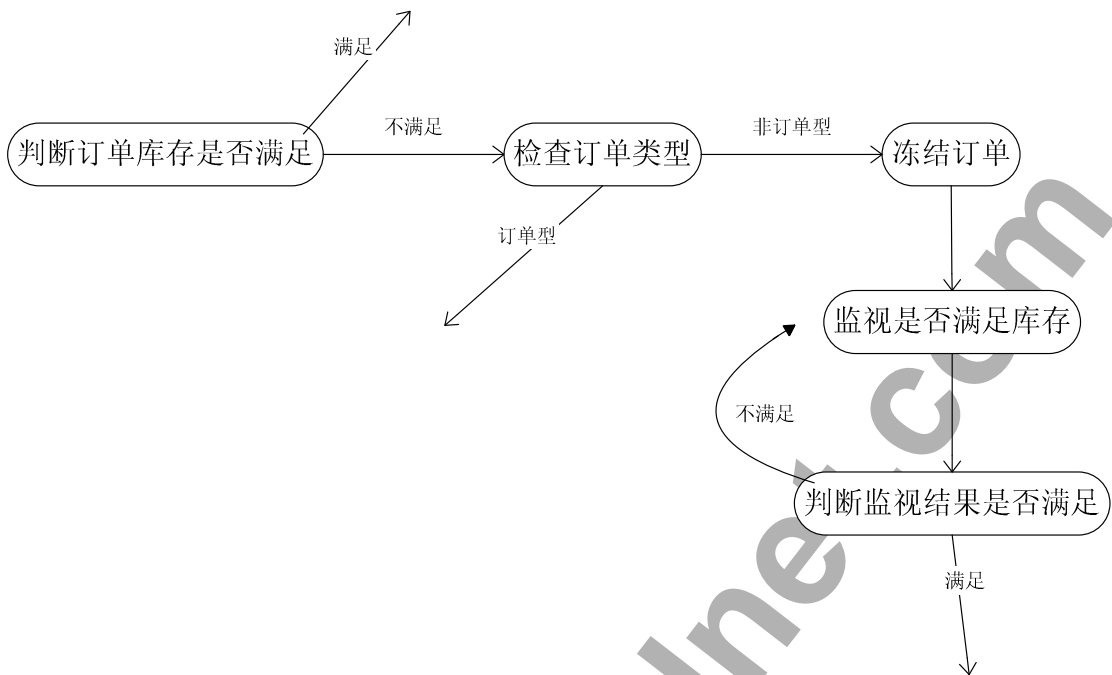
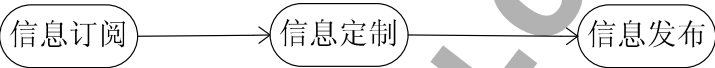
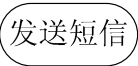


图 2-6 订单检查模块分析

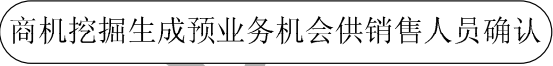
● 服务社区模块分析



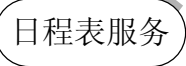
● 短信模块分析



● 短信模块分析



● 日程表模块分析



至此对七个子系统（包括两个遗留系统）分析完毕，增加了新的候选服务如下：

- ①. 展示产品列表
- ②. 查看某个产品详细信息

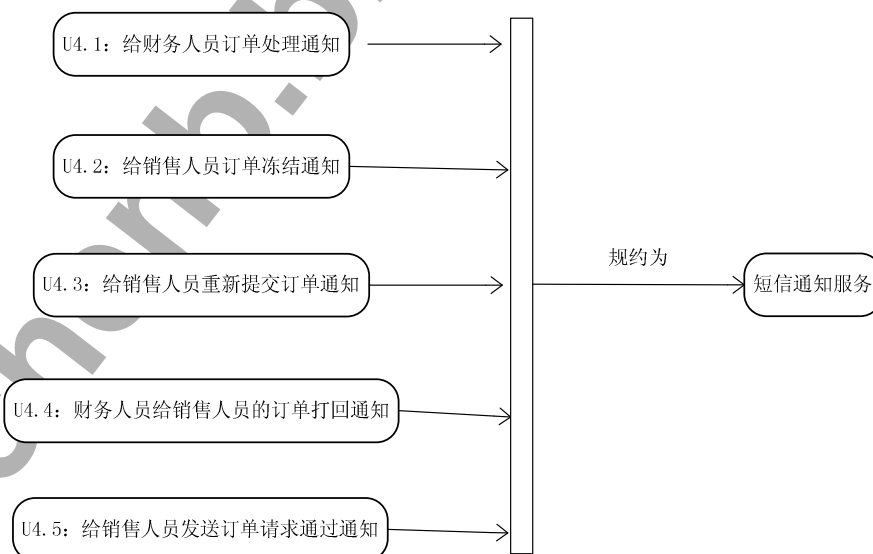
- ③. 查看客户基本信息
- ④. 查看客户历史交易信息
- ⑤. 查看订阅的客户信息
- ⑥. 判断 erp 系统中是否有客户信息
- ⑦. 创建客户信息
- ⑧. 判断监视结果是否满足

## (2) 服务规约

根据上面的分析结果，得到了候选服务目录。下面，对服务进行规约，也就是决定暴露哪些服务。理论上所有的服务候选者都可以暴露为服务，但是一旦暴露为服务，该服务候选者就必须满足附加的安全性、性能等方面的要求，企业还必须为服务的规划、设计、开发、维护、监管支付额外的开支，因此我们会根据一定的规则来决定将哪些服务候选者暴露为服务：

### 1) 对服务后选择进行归并

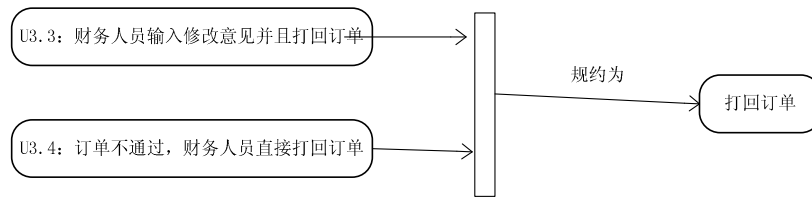
#### ① 短信服务



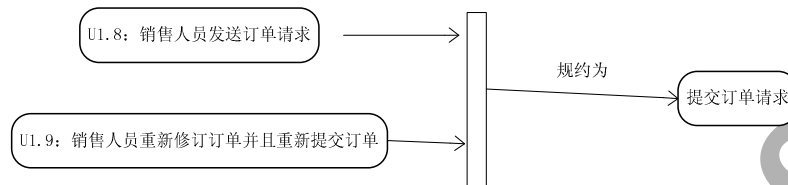


在短信系统中只提供一个服务—短线通知服务。

## ② 财务人员打回订单服务



## ③ 发送订单请求服务



通过的归并和以前的分析，最后定出了服务如下：

- U1: 销售人员
  - U1.1: 生成业务机会
  - U1.2: 销售人员进行商机确认
  - U1.3: 判断客户是否存在于 CRM 系统中
  - U1.4: 销售人员创建客户信息
  - U1.5: 销售人员浏览产品信息
    - U1.5.1 展示产品列表
    - U1.5.2 查看某个产品详细信息
  - U1.6: 销售人员获取客户信息
    - U1.6.1: 查看客户基本信息
    - U1.6.2: 查看客户历史交易信息
    - U1.6.3: 向服务社区模块订阅客户信息
  - U1.7: 销售人员维护业务机会
  - U1.8: 业务机会更新
  - U1.9: 销售人员发送订单请求
  - U1.10: 销售人员废除订单
  - U1.11: 判断业务机会状态

- U2: 订单管理
  - U2.1: 订单请求检查
    - U2.1.1: 判断是否为订货型订单
    - U2.1.2: 判断库存情况
  - U2.2: 订单冻结
  - U2.3: 监视库存是否满足
  - U2.4: 判断监视结果是否满足
- U3: 财务人员
  - U3.1: 财务人员审核订单
  - U3.2: 财务人员生成销售订单
  - U3.3: 财务人员打回订单
  - U3.4: 判断 ERP 系统中是否有客户信息
  - U3.5: 创建客户信息
  - U3.6: 获取订单请求
- U4: 短信
  - U4.1: 短信通知服务
- U5: 日程安排
  - U5.1: 日程表服务
- U6: 商机挖掘
  - U6.1: 商机挖掘生成预业务机会供销售人员确认
- U7: 服务社区
  - U7.1: 信息订阅
  - U7.2: 信息定制
  - U7.3: 信息发布

根据上面的分析结果，得到了需要实现的服务列表。下面，对服务逐个进行规约，为后续的构件规约和最后的实现提供支持。

## 2) 服务规约模版

- 标识：服务的名称或者代号

- 类别：对识别出来的服务进行归类。经过上面的分析，得到的服务是特定领域的业务服务，所以，在这里采用了 SOA 逻辑体系结构模型中的服务分类法[3]，作为候选的服务类别。

- Interaction Services: 指的是实现业务设计构件的界面逻辑的服务，一般用来支持在应用程序与终端用户之间的交互。这里的终端用户不只是限定于人，在某些情况下可以是机器人，传感器或者过程控制的设备等。其注重的是为用户其提供交互和展现信息的最佳方式。

- Process Services: 包括不同方式的组合逻辑，其中比较典型的有：业务流和业务状态机。当是不局限于这两种比较广泛使用的类型，还包括别的形式例如商业规则和决策树等组合服务的方法。

- Business Application Services: 是实现核心业务逻辑的服务。这种服务是业务模型中的核心模块，也是最基本的组成部分。它是不可再分解的，但是可以组合来形成更高层的服务。这些服务通常通过组合来完成业务流程，或者由 Interaction Services 来调用。

- Information Services 是包含业务设计数据逻辑的服务，一般是为业务应用提供服务。这里包括了两种层次的逻辑。第一种是数据层的服务，这种服务提供对业务中永久型数据，可以通过数据库的查询语句来实现，或者通过组合查询服务数据来得到。而另一种服务主要是

在多个数据库上提供联合的服务，给为其他服务提供一个获取数据的统一视图。

■ **Access Services:** 是把遗产应用或者功能整合到 SOA 中的。一般有两种方法，一种是通过简单的将现有功能包装成一个服务，而另一种是将现有多个功能组合成服务来满足不同的业务需求。这是根据应用环境的复杂程度而定。而后者通常包括了需要使用到多个遗产系统中提供的功能。

■ **Partner Services:** 能够实现商业伙伴之间的互操作的服务。这种服务通常需要包含商业合作之间的规则和约束，包括横行的行业信息和数据交换的标准，以及业务逻辑。这种类型的服务和 Interaction services 非常类似，只是把商业伙伴的服务作为是外部的实体。

- **功能:** 服务所提供的功能。自然语言描述。
- **输入:** 服务所需要的信息，自然语言描述。
- **输出:** 服务返回的信息，自然语言描述。
- **质量属性:** 描述对提供的服务在质量属性上的需求，包括安全性，健壮性以及可靠性等。
- **依赖:** 提供这个服务所需要的外部提供的服务。这里的外部服务主要指的是业务模块之外的。

### 3) 服务规约

标识: 订单请求检查
类别: Interaction Service

提供者: 订单管理
功能: 如果确定订单请求, 则将订单请求生成销售订单, 并将确认消息以短信方式通知相应的销售人员。之后, 订单信息更新到销售业务; 如果退回订单请求, 则将预订单删除, 并将退回理由以短信方式通知相应的销售人员
输入: 预订单标识, 确认标识
输出: 处理状态
质量属性:
依赖: 通知设施的发送短信服务, 通知设施的邮件地址检测服务, 数据更新的订单信息触发更新
异常: 过期未处理提醒

标识: 库存检查
类别: Access Service
提供者: 库存管理
功能: 对产品的库存需求列表, 检查是否目前的库存能否满足, 将相应的检测结果返回。如果库存满足, 则结束本次检查; 否则, 需将列表进行保存, 并将冻结信息以短信通知的方式发送给相应的销售人员。在库存信息更新的时候对冻结的列表进行再次检查, 如果库存能够满足, 则通知相应销售人员再次提交, 删除相应列表结束这次库存检查。
输入: 产品的库存需求列表
输出: 检测结果
质量属性:
依赖: 通知设施的发送短信服务
异常: 关键信息缺失, 过期自动删除

标识: 增加日程安排
类别: Access Service
提供者: 日程管理
功能: 在对应人员的日程安排上增加相应的通知
输入: 通知内容, 通知人员的标识
输出: 无
质量属性:
依赖:
异常: 关键信息缺失

标识: 短信通知
类别: Access Service
提供者: 短信
功能: 将短信内容发送到对应人员的手机上
输入: 短信内容, 目标手机的标识 (可以为手机号码或者人员标识)

输出： 短信发送状态
输出： 无
质量属性：
依赖：
异常： 关键信息缺失

项目中，服务规约会比较复杂，既包括具体的服务的操作、输入消息、输出消息，也包括相关联的业务目标、业务规则、业务事件，此外，非功能性需求等方面也是需要在服务实现以前定义。限于篇幅，上面仅仅列举几个方面做简单的示意。

除了对单个的服务本身进行规约，服务规约还包括服务之间关系的描述，例如服务之间的依赖关系和包含关系。

### （3）服务实现分析

无论怎样进行服务建模，服务最终都将由不同的服务组件来实现。因此服务实现是衔接服务建模和组件详细设计的关键步骤。服务实现首先将服务分配到相应的服务组件，然后逐个分析服务实现方式并进行技术可行性的验证。

在服务发现的过程中，我们根据业务领域的分析结果将服务按照业务范围进行分类。在服务实现的过程中，将业务范围直接映射到服务组件，从而实现业务与 IT 的一致性。

服务实现的方式如图 2-7 所示。“销售人员（销售业务）”业务组件将实现生成业务机会、销售人员进行商机确认、创建客户信息、... 等服务。“订单管理”业务组件将实现订单请求检查、订单冻结等服务。

在我们的项目中，对于服务实现方式的选择，可以分为以下几类：

- 映射已有功能服务：如生成业务机会、创建客户信息、产看客户基本信息、...。其好处非常明显，就是重用已有功能，保护企业

的投资；避免重复功能的存在，降低维护成本。但是在选择的过程中，需要考虑传输协议、消息格式的差异，是否可以通过引入中介来弥合服务调用者和实现者之间的差距。需要特别提出的是如果有的服务外部由合作伙伴提供，通过中介将外部的服务进行映射（还需要重点考虑安全性相关的问题），在业务流程中就可以无缝的使用了。

- 新建流程服务：如生成销售订单。这是一个短流程（Micro Flow）。而长流程（Long Running），由于有人工活动的参与，使得长流程的执行不能在可预期的短时间（如：几秒钟）内完成，需要相关人员在完成自己的任务以后，流程才能进入下一步，常常是几天甚至几个月才能完成整个流程。在传统的方案中，业务流程通常采用硬编码的方式将多个功能组装起来；与之相对，我们推荐采用工作流（如 BPEL）的方式将服务组装起来，从而达到灵活组装、灵活应对变化的目的。

- 新建人工服务：如销售人员进行商机确认、销售人员发送订单请求。人工服务是相对于自动化服务而言。自动化服务通常由 IT 系统来提供，不用人为的干预；人工服务则是由企业的员工、合作伙伴员工或者最终用户来执行，但是它同样具备完整的服务描述。采用统一的服务描述来定义人工服务，可以将人工服务与自动化服务统一对待，除了可以在多个应用之间重用人工服务以外，还可以在服务实现从人工活动迁移到 IT 系统的过程中保持系统的柔性。

- 新建业务规则服务：如判断业务机会状态。由于这部分功能不稳定，会随着竞争环境、社会环境和客户情况的变化而变化。将易于

变化的这部分逻辑从稳定的架构中剥离出来，可以增强 IT 应对业务变化的能力。采用业务规则来实现相应的服务，可以相对灵活的进行修改来适应业务的变化，业务规则引擎已经在大量的行业得到广泛的应用。

- 新建功能服务：如获取订单请求、短信通知服务等。针对以前没有的功能，或者以前采用人工方式完成的功能，现在可以引入自动化服务来提高业务流程的运行效率。在这里实现了新建功能服务以后，也能在其他的应用中逐步引入，从而达到在企业范围内重用的目的。



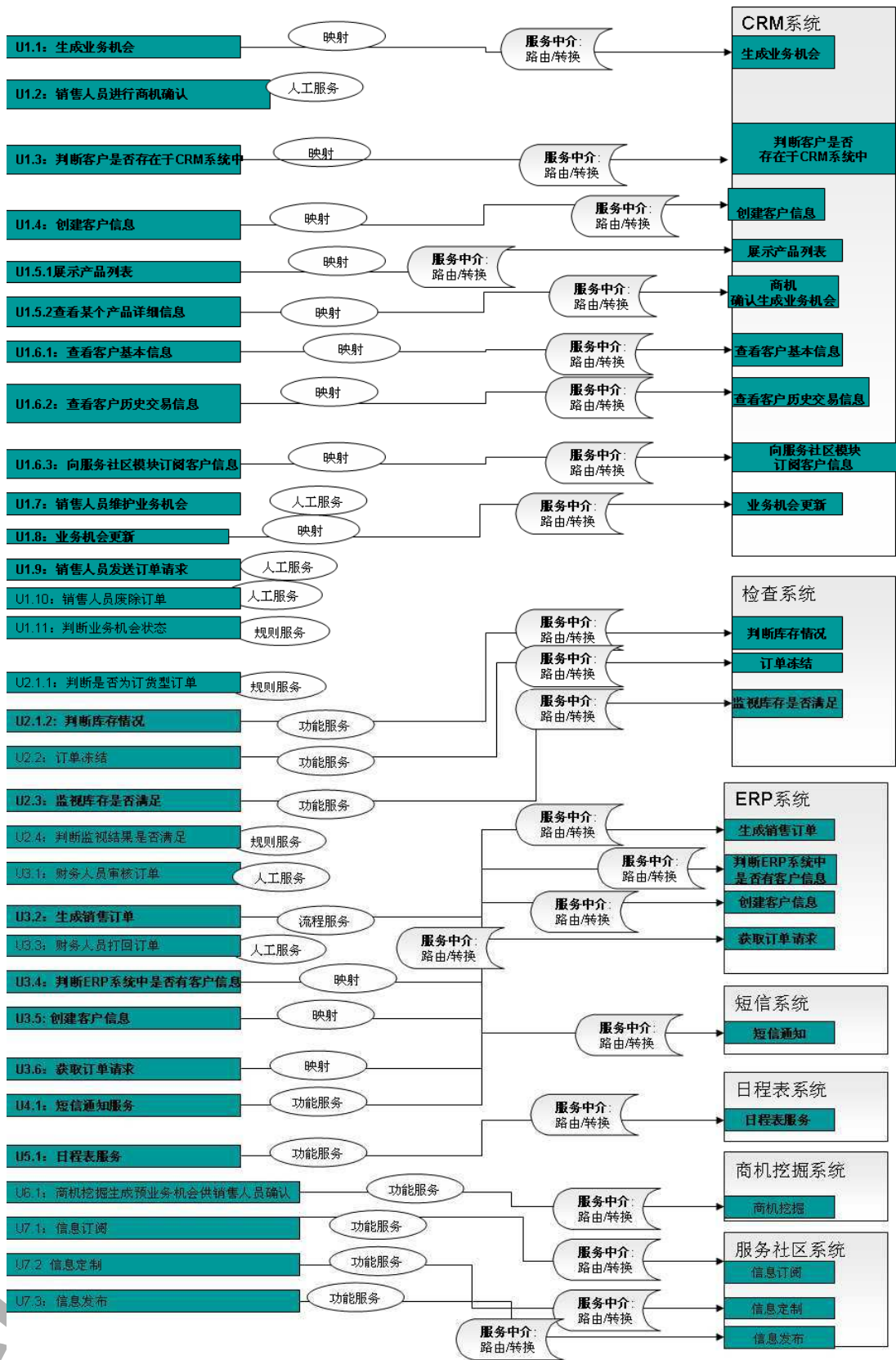


图 2-7 服务实现