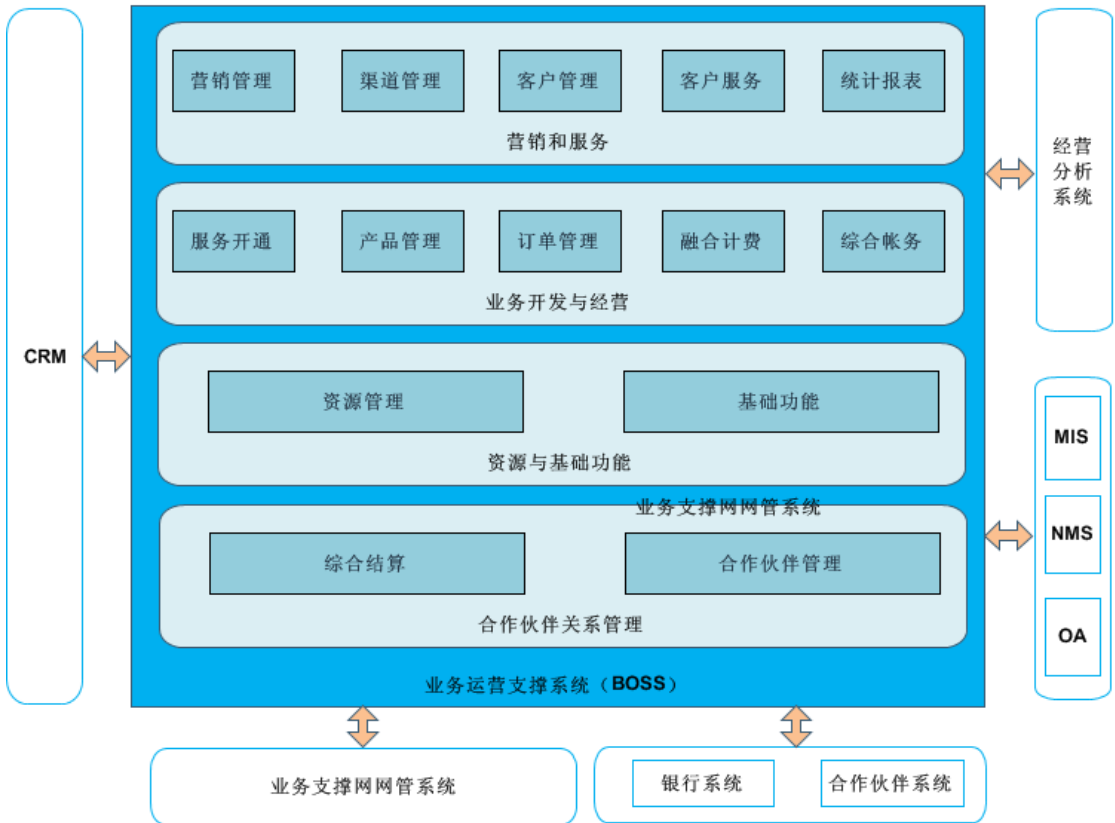


1.业务需求

1.1 业务目标

在电信领域中，业务运营支撑系统(BOSS)是电信企业业务处理**信息化管理**的基础平台，用于支持前台销售、**客户服务**, 内部支撑全部流程及分析管理的业务管理系统。



BOSS通常有以下几类主要业务及其功能：

(1) 面向多种业务的功能

多种业务有固定话音及数据业务、无线话音及数据业务、无线数据业务等。功能主要有工单调度、资源管理等融合的营业系统、多业务融合的计费系统与账务系统、统一的客户服务系统、统一的客户关系管理（CRM）系统、业务开通与保障、业务开发与**决策**、**服务水平协议（SLA）/服务质量保证（QoS）**管理以及应用集成等。

(2) 面向一般**消费者**及大众化IP业务的功能

主要包括：营业系统、账务系统、计费系统、**客户服务**、**客户分析**、业务开发与规划、业务激活、业务保障和应用集成等。

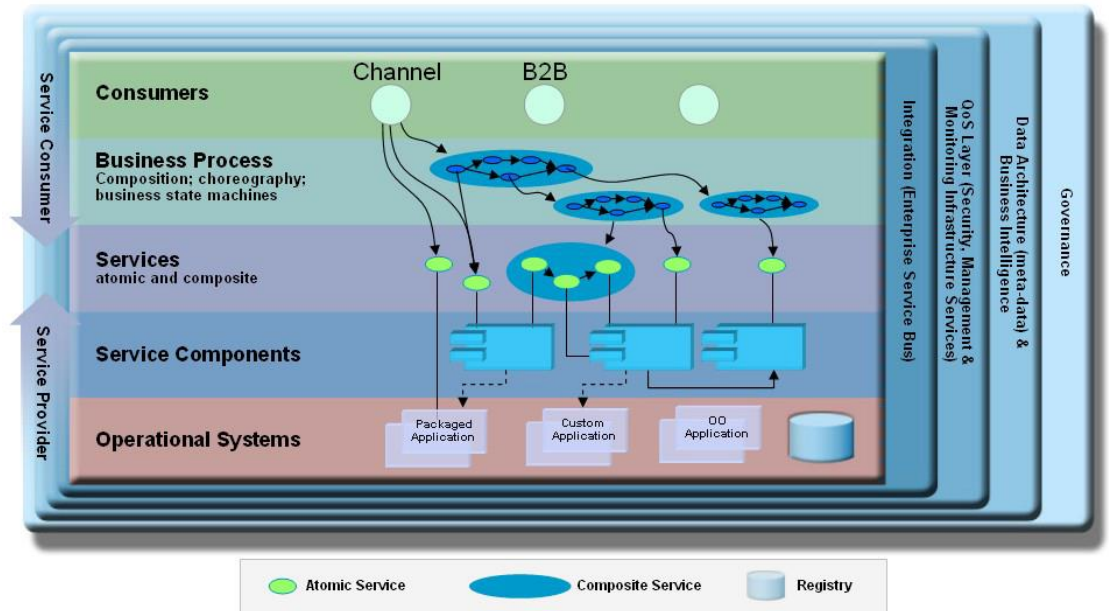
(3) 面向企业和个人用户的数据业务的功能

该功能主要是针对个人用户特别是VIP客户所需的个性化服务。其流程复杂，多样化。主要功能有：营业系统、工单调度、资源管理、计费系统、账务系统、客户服务系统、CRM系统、业务开通与保障、业务开发与**决策**、**SLA/QoS**管理以及应用集成等。

1.2 问题分析

对于传统的业务运营支撑系统而言，电信企业的各个系统往往以单一业务为出发点，内部缺乏有效的互连互通，形成“信息孤岛”。而BOSS系统的重要功能是实现统一管理，并为企业决策提供信息支持，如果集中程度不够，统一管理和决策支持能力都将受到影响，同时也会造成人力及资源的浪费。

以下将集中探讨订单可行性流程。在企业级用户向电信运营商提出业务开通要求时，电信运营商首先要检查该业务是否能够被运营商所满足，这里把业务的开通要求视作订单，判断订单是否能够被运营商所满足的流程称之为订单可行性流程。订单通常包括一个或多个产品和资源，所以订单最终是否能被运营商所满足取决于包含的产品和资源是否能够被提供。由于产品和资源的多样性和复杂性，验证一个订单是否可以被提供，需要和不同的系统通过不同的方式进行交互。如果没有明确的定义和建模的订单可行性流程，仅仅判断一个订单是否可以被满足，运营商就需要花费大量的时间，而且很难监控整个订单可行性流程的运行瓶颈以及了解订单为何不能得到满足。客户的满意度由于订单可行性流程的效率受到影响，同时企业也不能够及时的采取措施来应对现有的缺陷。

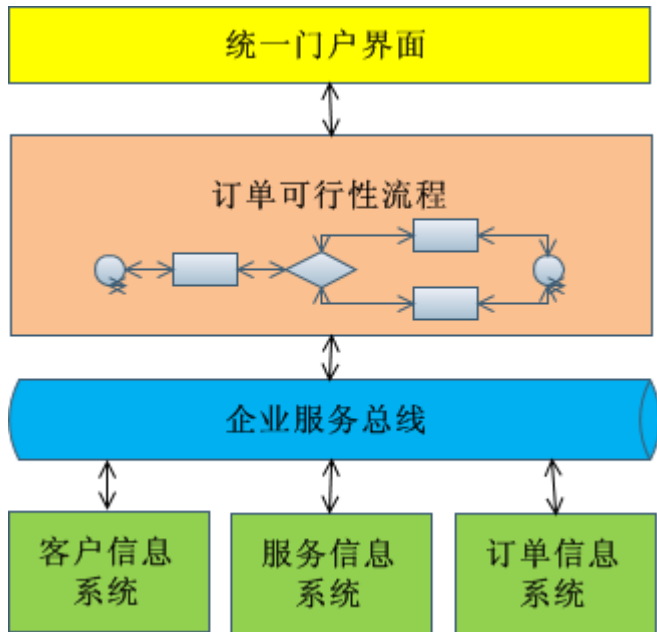


通过使用SOA/BPM的理念和方法来建模订单可行性流程，可以得到以下好处：

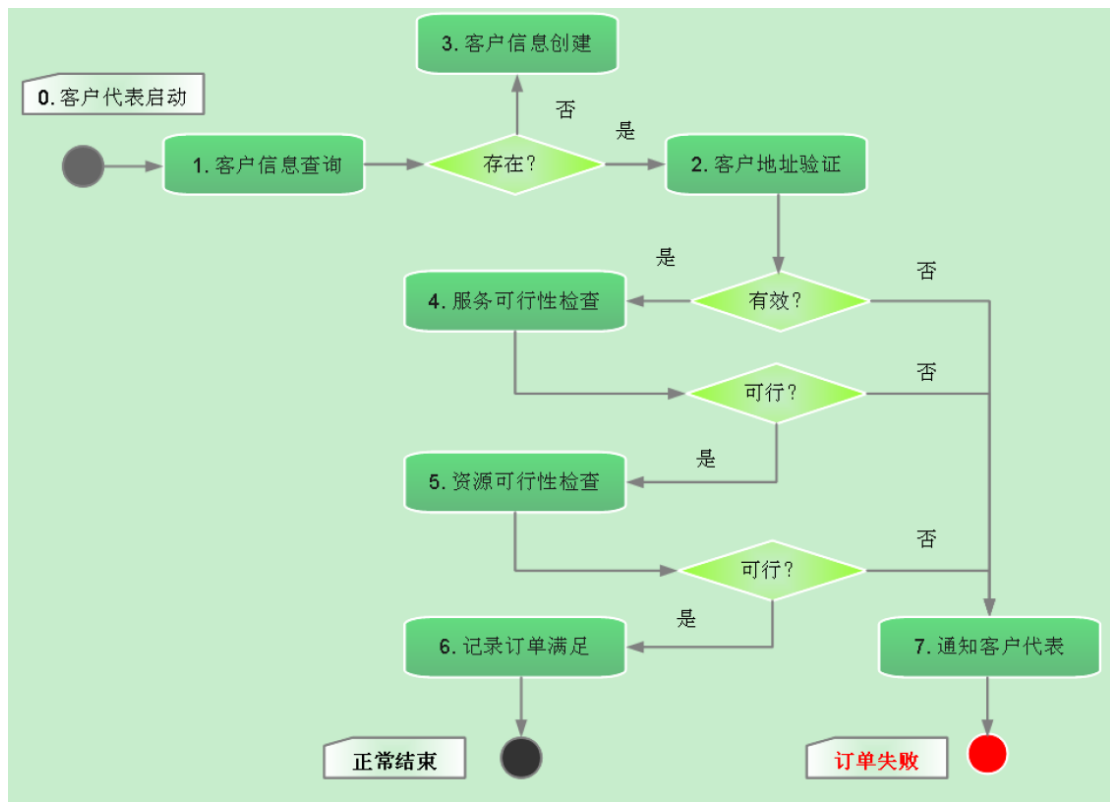
- (1) 在业务部门和IT部门之间维护统一的业务流程，因而业务人员和IT人员能够对流程有统一的认识和了解，业务人员的意图能够更好的映射到IT人员的实现。
- (2) 通过自动化订单可行性流程，消除和不同系统交互时不必要的人工干预过程，从而有效地降低流程的执行时间，最终提高客户的满意度和企业的收入。
- (3) 通过监控流程执行过程中的KPI (关键业务指标)，了解订单的趋势和满足程度，以及流程执行中的瓶颈所在；通过提供或者准备相应的资源和服务，来提高流程的整体执行效率和订单的满足比例。

1.3 目标业务流程

订单可行性系统架构如下图所示：



订单可行性流程逻辑如下图所示：



流程的总体逻辑简单描述：

0. 客户代表在收集完客户信息和客户需要提交的订单后，在系统中输入相关的信息并且启动订单可行性流程。
1. 流程调用客户信息模块验证客户是否存在。如果存在，进入步骤 2；如果不存在，进入步骤 3。
2. 流程调用地址验证模块验证地址是否有效。如果有效，进入步骤 4；如果无效，进入步骤 7。

- 3. 流程调用客户信息模块创建客户，然后返回步骤 2。
- 4. 流程调用服务信息模块查询服务是否可以被满足。如果可行，进入步骤 5；如果不可行，进入步骤 7。
- 5. 流程调用资源检查子流程验证资源是否可以被满足。如果可行，进入步骤 6；如果不可行，进入步骤 7。
- 6. 流程调用订单信息模块纪录订单信息，订单满足，正常结束。
- 7. 流程调用通知模块通知客户代表订单失败及失败原因。

1.4 业务规则

规则编号	规则名	备注
101	客户地址验证规则	如果客户地址无效，拒绝其订单
102	VoIP 服务可行性规则	客户地址所在区域是否具备宽带服务（客户需要预先安装宽带），上门安装服务和 ID 产生服务的条件，以上三种服务须同时具备 VoIP 服务才可行。
103	DSL 服务可行性规则	客户地址所在区域是否具备上门安装服务和 ID 产生服务的条件，以上两种服务须同时具备 DSL 服务才可行。
104	VoIP 资源可行性规则	客户需要具备 VOIP 网络电话机和路由器，以上两种资源须同时具备才能满足 VoIP 资源。
105	DSL 资源可行性规则	客户需要具备 Cable Modem，DSLAM，混合光纤/同轴电缆，以上三种资源须同时具备才能满足 DSL 资源。

2 系统需求

2.1 用户特征分析

表 1 地域说明

地点	描述	关键属性
X市1街	省公司所在地	M个相关部门, N名注册用户
X市2街X	市公司所在地	
Z市	Z地市公司所在地	

表 2 组织结构说明

名称	级别	描述
省公司市场部	一级	ABC 省公司市场部
地市公司市场部	二级	X 市分公司市场部

表 3 角色说明

角色	职责描述
客户代表	负责提交客户订单
资源管理员	负责资源调控

表 4 用户组说明

用户组名称	主要特征				
	访问方式	IT 使用经验	系统使用频率	工作时间段	系统访问要求
UG1: 客户代表	PC	中	中	9:00-18:00	中
	PC	中	中	9:00-18:00	中
	PC	低	低	9:00-18:00	中
UG2: 资源管理员	PC	高	高	9:00-18:00	高

2.2 系统边界

相关系统如下：

（1）客户信息系统

客户信息系统提供客户信息管理的功能，具备增、删、改、查功能。本系统将使用它来创建新客户，查询客户记录，以及核对信息的合法性。

（2）服务信息系统

服务信息系统提供电信运营商服务信息管理功能，具备增、删、改、查功能，涉及服务信息和资源信息的管理。本系统将使用它进行服务注册、服务查询和资源查询。

（3）订单信息系统

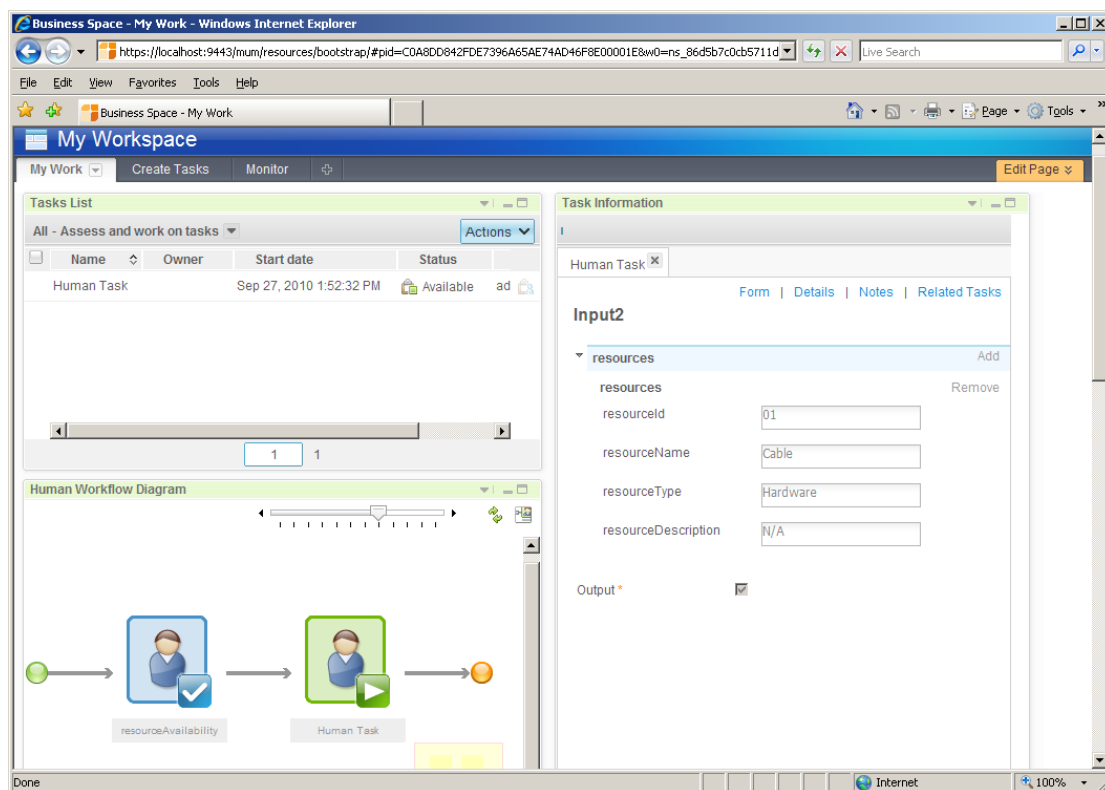
订单信息系统提供订单信息管理功能，具备增、删、改、查功能，本系统将使用它进行订单的管理。

2.3 系统用例

用例部分由学生在实验中完成。

2.4 用户界面需求

用户界面是基于 IBM Business Space 框架实现。界面风格如下：



2.5 非功能需求

(1) 性能（Performance）

订单管理平台工作时间段定义如下表所示：

表 5 订单管理平台工作时间段定义

波峰时间段	8:00 AM - 5:00 PM, Monday – Friday
波谷时间段	5:00 PM - 8:00 AM, Monday – Friday 0:00-24:00, Saturday and Sunday

表 6 订单管理平台所需支持的用户数

用户组名称	性能要求			
	注册用户数	在线用户数	功能域	并发用户数
UG1: 订单客户代表	1000	500	FA1	波峰: 100, 平均: 50
			FA2	波峰: 50, 平均: 20
			FA3	波峰: 50, 平均: 20

表 7 订单管理平台所需达到的平均响应时间

并发用户数	简单交易平均响应时间秒	中等交易平均响应时间秒	复杂交易平均响应时间秒
10	<2	<5	<8
20	<3	<6	<10
30	<4	<7	<12
50	<5	<8	<14

(2) 可用性（Availability）

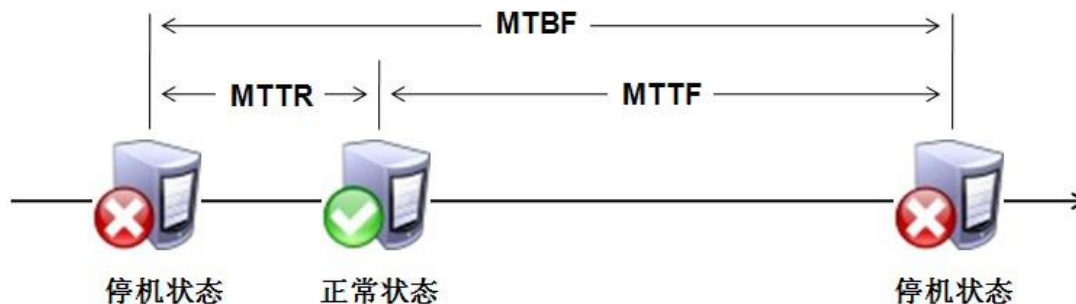
描述系统可用性的相关参数主要包括：

平均故障时间 MTTR（Mean Time To Restoration）

平均无故障时间 MTTF（Mean Time To Failure）

平均故障间隔时间 MTBF（Mean Time Between Failure）

各指标间的关系如下图所示：



通常，采用 MTTF/MTBF 来描述系统的可用性。例如，一个可用性为 99.9% 的系统一年内平均故障时间（MTTR）为 $365 \times 24 \times 0.1\% = 8.76$ 小时(525 分钟)，而要想让系统的平均故障时间在一年中只有 3 分钟的话，系统必须有 99.999% 的可用性。

（3）易扩容（Scalability）

采用负载均衡架构，系统负荷加大时，能够通过动态调整主机资源实现升级，确保所需的服务质量，而不需要更改整个系统的架构。

3.业务流程建模

3.1 WebSphere Business Modeler 实现要求

SOA 生命周期的第一阶段是业务人员开发业务模型。业务人员最了解业务逻辑，所以应该由业务人员定义业务模型。

注：这样做首先是更清晰的描述整个项目。业务模型中描述了企业组织架构，定义人员角色，描述业务流程，其次在业务部门和 IT 部门之间维护统一的业务流程，因而业务人员和 IT 人员能够对流程有统一的认识和了解，业务人员的意图能够更好的映射到 IT 人员的实现。

3.2 描述业务流程

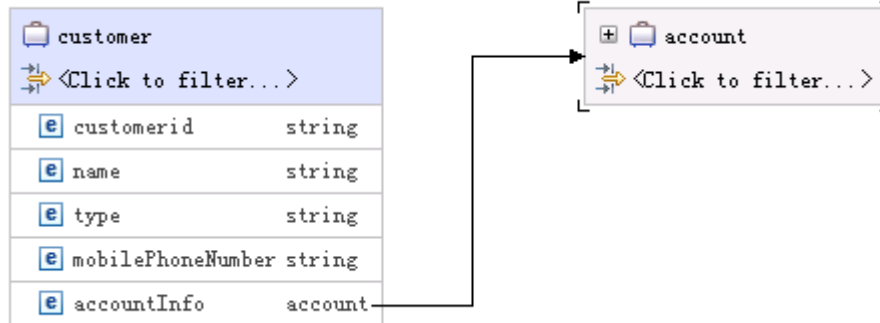
3.2.1 数据定义 (Business Item)

根据如下数据定义，使用 Modeler 建立各种数据类型。在 Modeler 中，用 Business Item 来进行数据定义。

1. account 类型

	account
	<Click to filter...>
	accountID string
	accountName string
	accountType string
	credit int

2. customer 类型



3. address 类型

	address
	<Click to filter...>
	city string
	street1 string
	street2 string
	postcode string

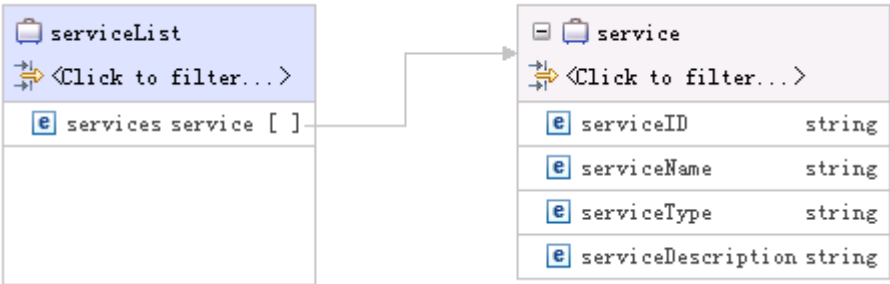
4. service 类型

	service
	<Click to filter...>
	serviceID string
	serviceName string
	serviceType string
	serviceDescription string

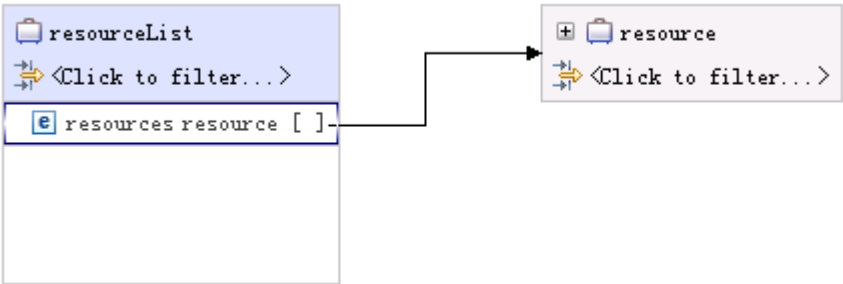
5. resource 类型

resource	
<Click to filter...>	
resourceId	string
resourceName	string
resourceType	string
resourceDescription	string

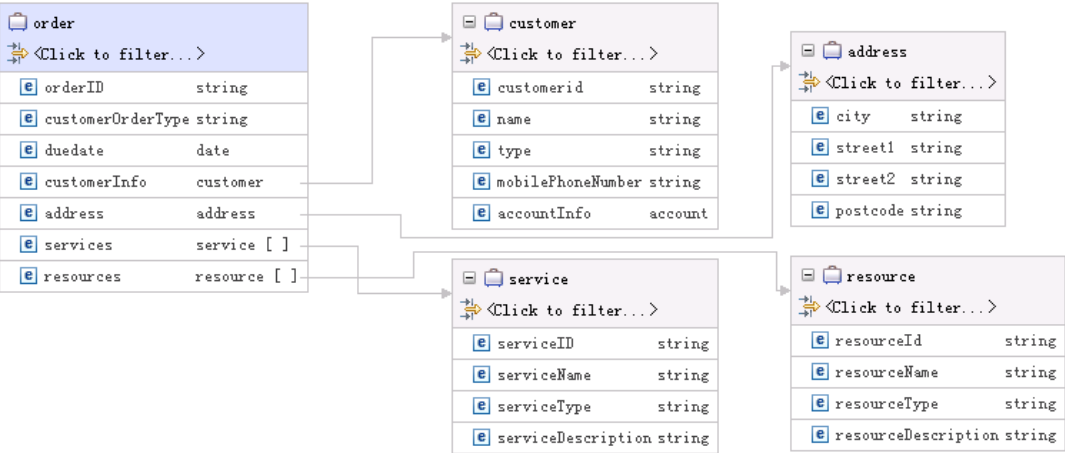
6. serviceList 类型



7. resourceList 类型



8. order 类型



3.2.2 服务定义(Business Service)

服务的接口和输入输出数据与如下节点定义相同，对外表现为 **Web Services**。先定义接口的 WSDL 文件（可以使用 WID 的图形化界面工具生成 WSDL 文件），再使用 Modeler 导入这些服务定义。在 Modeler 中，用 Business Service 来进行服务定义。

1. 流程启动节点

接口定义：



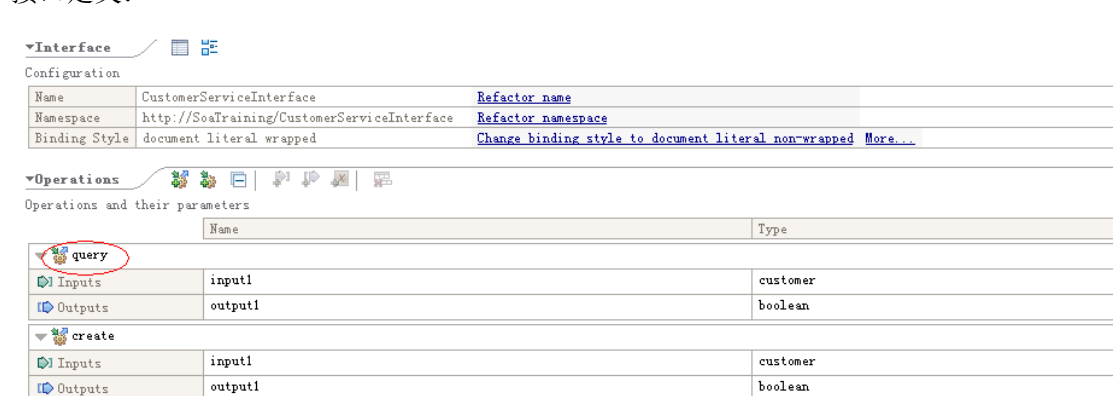
Configuration		
Name	OrderFeasibility	Refactor name
Namespace	http://SoaTraining/OrderFeasibility	Refactor namespace
Binding Style	document literal wrapped	Change binding style to document literal non-wrapped More...

Operations and their parameters		
	Name	Type
▼ start		
Inputs	input1	order

输入数据格式：order 类型

2. 客户信息查询节点

接口定义：



Configuration		
Name	CustomerServiceInterface	Refactor name
Namespace	http://SoaTraining/CustomerServiceInterface	Refactor namespace
Binding Style	document literal wrapped	Change binding style to document literal non-wrapped More...

Operations and their parameters		
	Name	Type
▼ query		
Inputs	input1	customer
Outputs	output1	boolean
▼ create		
Inputs	input1	customer
Outputs	output1	boolean

输入数据格式：customer 类

输出数据格式：boolean 类型, true 表示客户信息已存在，false 表示客户信息不存在

实现方式：通过调解流，调用客户信息系统的 Web Services

3. 客户地址验证节点

接口定义：



Configuration		
Name	addressServiceInterface	Refactor name
Namespace	http://SoaTraining/addressServiceInterface	Refactor namespace
Binding Style	document literal wrapped	Change binding style to document literal non-wrapped More...

Operations and their parameters		
	Name	Type
▼ verify		
Inputs	input1	address
Outputs	output1	boolean

输入数据格式：address 类型

输出数据格式：boolean 类型，表示地址是否有效

实现方式：通过调解流，调用客户信息系统的 Web Services

4. 客户信息创建节点

接口定义：

▼Interface 

Configuration

Name	CustomerServiceInterface	Refactor name
Namespace	http://SoaTraining/CustomerServiceInterface	Refactor namespace
Binding Style	document literal wrapped	Change binding style to document literal non-wrapped More...

▼Operations 

Operations and their parameters

	Name	Type
▼  query		
 Inputs	input1	customer
 Outputs	output1	boolean
 create		
 Inputs	input1	customer
 Outputs	output1	boolean

输入数据格式：customer 类型

输出数据格式：boolean 类型

实现方式：通过调解流，调用客户信息系统的 Web Services

5. 服务可行性检查节点

接口定义：

▼Interface

Configuration

Name	servicesAvailabilityInterface	Refactor name	
Namespace	http://SoaTraining/servicesAvailabilityInterface	Refactor namespace	
Binding Style	document literal wrapped	Change binding style to document literal non-wrapped	More...

▼Operations

Operations and their parameters

	Name	Type
▼ verifyAvailability		
Inputs	input1	serviceList
Outputs	output1	boolean

输入数据格式：serviceList 类型

输出数据格式：boolean 类型，表示服务列表是否可以被满足

实现方式：通过调解流，调用服务信息系统的 Web Services

6. 资源可行性检查节点

接口定义：

▼Interface

Configuration

Name	resourcesAvailabilityInterface	Refactor name	
Namespace	http://SoaTraining/resourcesAvailabilityInterface	Refactor namespace	
Binding Style	document literal wrapped	Change binding style to document literal non-wrapped	More...

▼Operations

Operations and their parameters

Name	Type
▼ verifyAvailability	
Inputs	input1 resourceList
Outputs	output1 boolean

输入数据格式：resourceList 类型

输出数据格式：boolean 类型，表示资源列表是否可以被满足。

实现方式：资源可行性检查包括多项资源的检查，较为复杂，需要人工干预。该节点的实现方式规定为调用包含人工任务的子流程。

7. 纪录订单满足节点

接口定义：

▼Interface

Configuration

Name	orderServiceInterface	Refactor name	
Namespace	http://SoaTraining/orderServiceInterface	Refactor namespace	
Binding Style	document literal wrapped	Change binding style to document literal non-wrapped	More...

▼Operations

Operations and their parameters

	Name	Type
▼record		
Inputs	input1	order
Outputs	output1	boolean

输入数据格式：order 类型

输出数据格式：boolean 类型

实现方式：通过调解流，调用订单信息系统的 Web Services。

8. 通知客户代表节点

接口定义

▼接口

配置

名称	a7notifyService	重构名称
名称空间	http://soaTraining/Processes/orderFeasibilityProcess/a7notifyService/a7notifyServiceInterface	重构名称空间
绑定样式	document literal wrapped	将绑定样式更改为 document literal non-wrapped 更多...

▼操作

操作及其参数

名称		类型
▼ a7notifyService		
输入	Input	order
输出	failReason	string

输入数据格式：order 类型

输出数据格式：string 类型

实现方式：由于通知客户代表根据实际情况和订单类型，可以是电话、短信或者系统通知，所以此处需要人工干预，该节点的实现方式规定为人工任务。

3.2.3 流程定义（Process）

使用 Modeler 描述业务流程。全局流程模型是由多种活动像搭积木一样拼接而成的。这些活动可以是外部业务服务（Business Service），可以是子流程（Sub Process），可以是人工任务（Human Task）。在这个流程中，通知客户代表节点用人工任务实现。资源可行性检查节点用子流程实现，要求子流程中嵌套一个人工任务。其他的所有节点都使用业务服务实现。

3.3 SOA 开发

业务人员在第一阶段定义的只是业务逻辑，缺乏相应的业务服务（Business Service），这些流程无法运行，所以 SOA 生命周期的第二阶段是需要开发人员开发相应的服务，修改流程，组装服务，并对所有的服务和实现进行相应的测试，保证系统最终能够上线运行。

3.3.1 用例的开发

运用面向对象的分析与设计的方法，对业务进行用例的开发。用例规约的格式如下：

用例名称	
用例编号	UC1201
简要描述	
主要参与者（角色）	
基本事件流	
可选事件流	
前置条件	
后续效果	
业务规则	
优先级	中
使用频度	次/月
备注	

3.3.2 Web Services 的开发

Web Services 开发人员根据服务定义，实现定义好的接口 wsdl 文件（课程教材提供），开发客户信息系统、服务信息系统和订单信息系统对外暴露的 Web Services。Web Services 开发人员还应该使用 WID 中的资源管理器对自己开发的 Web Services 进行单元测试，保证服务正确可用。各个信息系统如何保存信息，由团队自行决定。

3.3.3 调解流的开发

SOA 的核心思想是将各个系统的 Web Services 发布到企业总线上，然后各个系统的交互通过调解流进行路由和数据格式的转换。所以 ESB 开发人员需要开发调节流，实现调用各个 Web Services 的路由，并且为流程调用提供一个统一的 SCA 服务接口。

ESB 开发人员还应该使用 WID 中的测试组件功能对调解流进行单元测试，保证调解流能够正确运行。

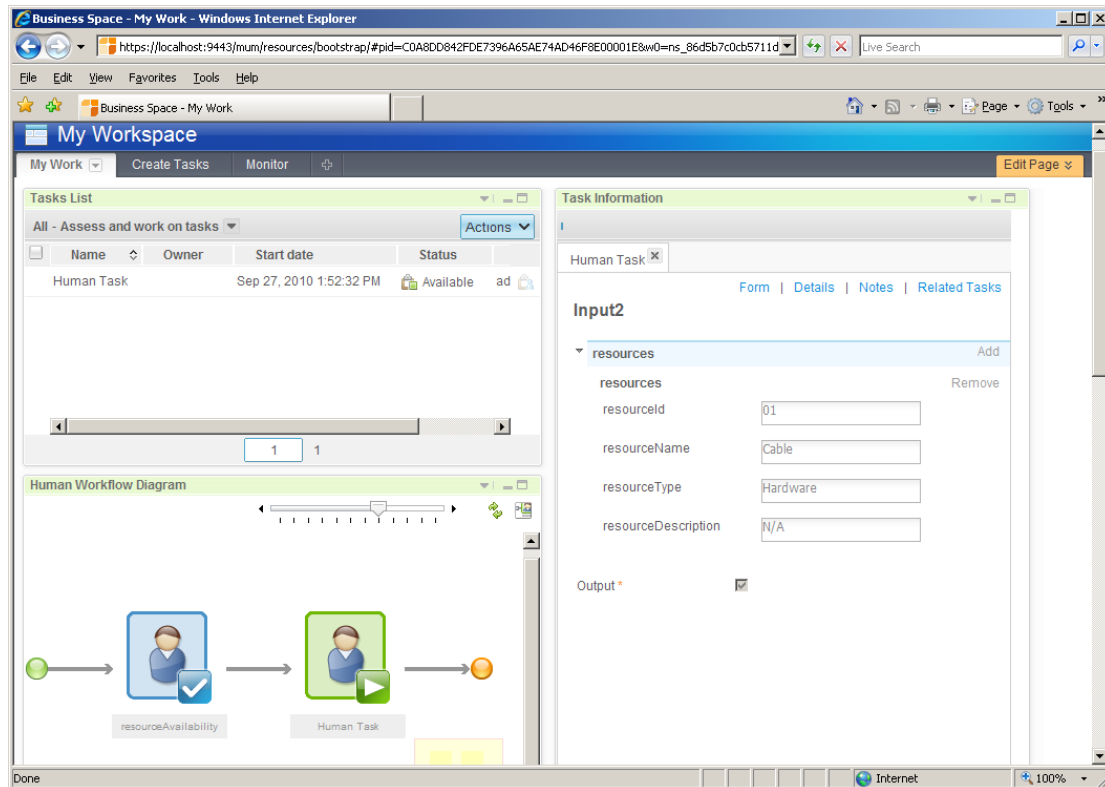
3.3.4 流程开发和服务组装

流程开发人员从业务人员手中获得业务流程定义，根据流程的接口和输入输出数据格式的描述，以及流程中每个调用节点的接口和输入输出数据格式的描述，对流程进行修改，包括修改映射、为人工任务开发能在业务空间上运行的 HTML-dojo 界面、集成组装总线上暴露的服务。

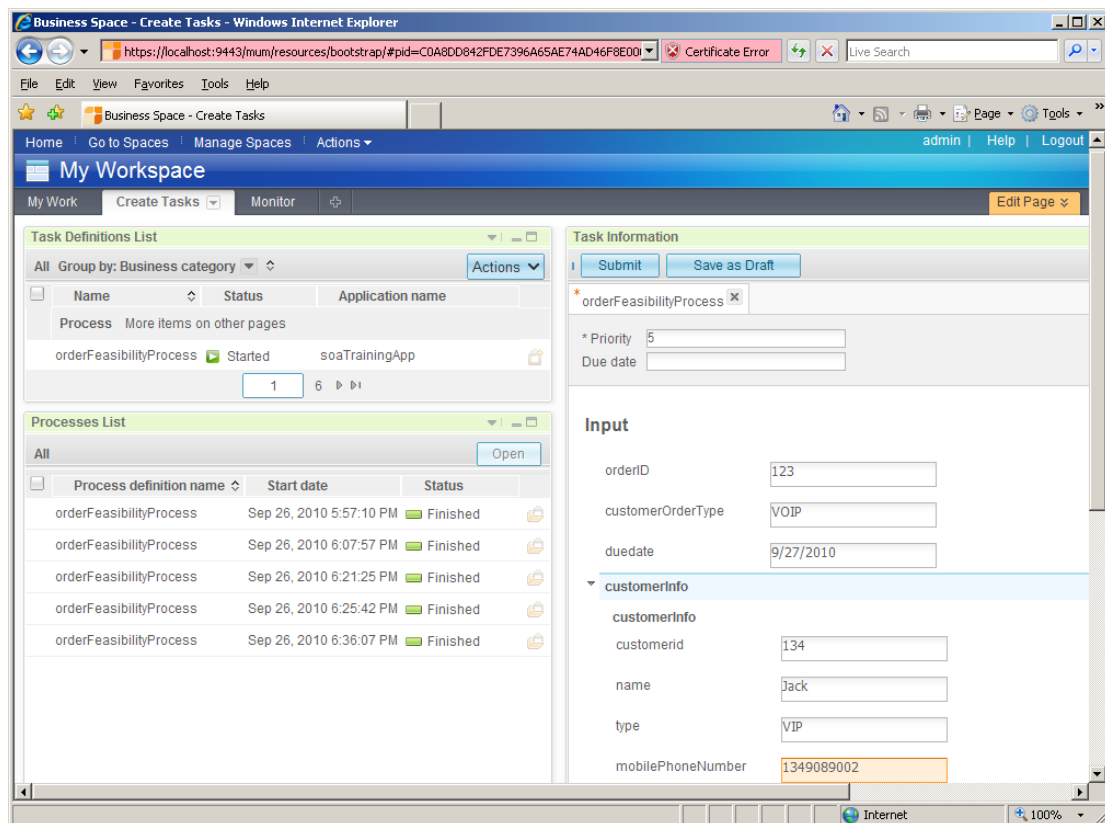
流程开发人员还应该使用 WID 中的测试组件功能或者 BPC explorer 对流程进行单元测试，保证流程能够正确运行。

3.3.5 界面开发

在 Business Space 上创建一个空白的空间，空间中包含两个页面：My Work 和 Create Tasks。My Work 页面用于处理待办任务，包含 Tasks List、Task Information 和 Human Workflow Diagram 三个 widget。Task List 用于展示我的待办任务列表；Task Information 用于在处理待办任务的时候展示表单；Human Workflow Diagram 用于展示当前流程的流程图。



Create Tasks 页面用于发起流程实例，包含 Task Definition List、Task Information 和 Processes List。Task Definition List 显示了流程模板，选中流程模板，可以创建流程实例；Task Information 用于展示发起流程实例的表单；Processes List 用于所有流程的状态和信息。



3.3.6 功能测试

测试人员需要将所有的服务、调解流和流程部署到 WPS 上，并且使用 Business Space，验证是否满足功能说明中描述的所有功能需求。

3.3.7 输出

工程文件：*.zip 文件，开发人员交互时使用的项目交换文件

工程文件：*.ear 文件，WPS 部署文件

门户界面设计文件 *.zip (Business Space)

3.4 SOA 监控

SOA 生命周期的最后一个阶段是 Monitor 监控器模型开发。Monitor 通过业务事件收集提供对业务流程的实时监控，并且图形化显示业务流程状态，还可以对业务历史数据进行多维数据查询分析。通过实时监控，业务流程相关人员可以及时发现并解决业务流程中出现的问题，通过多维数据分析，决策者可以基于分析结果做出更优化的选择。并且，可以为不同用户定制不同的使用界面和数据权限，这样每个用户都可以有定制的监控界面来显示所关注的的数据。

3.4.1 监控器模型开发

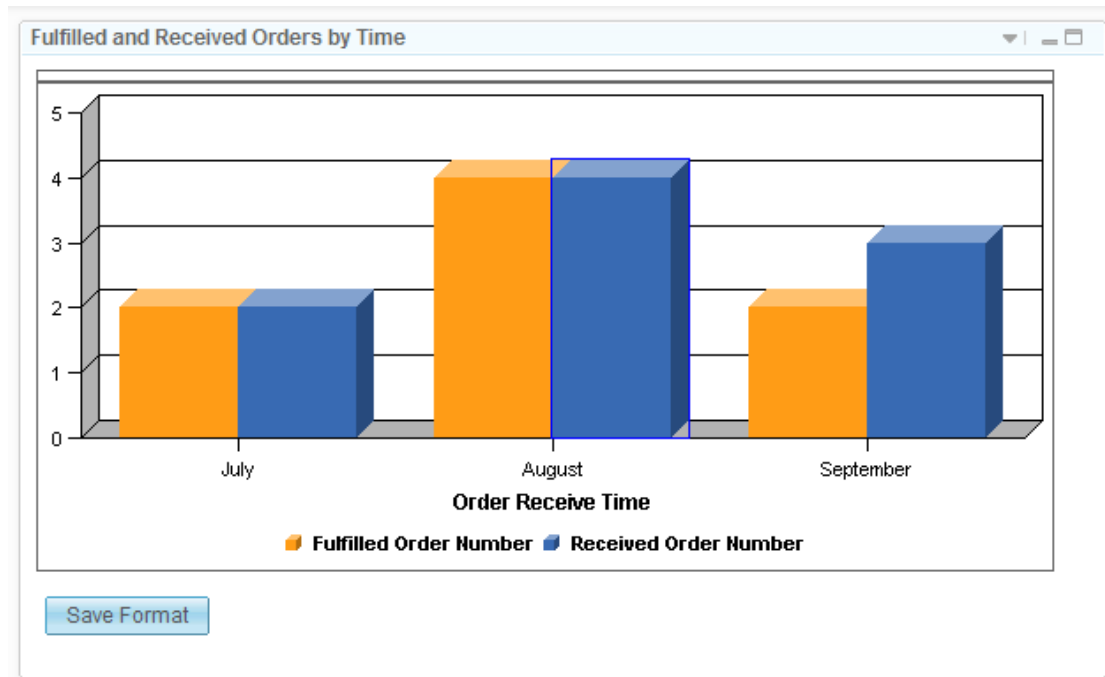
模型开发人员从流程开发人员手中获得业务流程定义，使用 WID 中的 Monitor Toolkit 产品生成监控器模型的框架，并继续开发监控器模型。

3.4.2 监控器模型需求

1. 创建一个基于 Telecom theme template 的 Business Space Dashboard.
2. 创建一个 Instance View 显示最近一月内收到的订单列表。
3. 创建一个 KPI View 显示平均流程执行时间，最近一个月收到的订单总数量和成功完成的订单数量，以及订单完成百分比。



4. 创建一个 Dimension View 基于订单时间维度显示度量值收到的订单数量和完成的订单数量。



3.4.3 功能测试

将监控器模型应用程序部署到 **Monitor** 上, 收集业务流程数据, 并且使用 **Business Space** 创建用户仪表盘, 验证是否满足监控器模型需求中描述的所有功能需求。

3.4.5 输出

监控器*.PI 文件, 开发人员交互时使用的项目交换文件

监控器*.ear 文件, **Monitor** 部署使用

监控器仪表盘(Dashboard)*.zip 文件, 开发人员创建的 **Business Space Dashboard**