

基于 UCM 的领域 SOA 资产库构建方法

陈廷伟, 杨艳辉, 关长城

(辽宁大学信息学院, 沈阳 110036)

摘要: 面向服务的体系架构(SOA)资产之间的复杂关系导致 SOA 资产管理困难。为解决上述问题, 提出一种基于 UCM 的领域 SOA 资产库构建方法。该方法通过 UCM 描述领域需求, 将领域需求转化为 UCM 模型中的标记及标记之间的关系, 将 UCM 模型中的标记及其之间的关系映射到 SOA 资产中, 建立领域 SOA 资产库, 从而实现 SOA 资产的有效管理, 并支持 SOA 资产复用。

关键词: SOA 资产复用; SOA 资产关系; SOA 资产库; UCM 语言; 映射规则

Domain SOA Asset Repository Construction Method Based on Use Case Maps

CHEN Ting-wei, YANG Yan-hui, GUAN Chang-cheng

(School of Information, Liaoning University, Shenyang 110036, China)

【Abstract】 Because of complex relationships among SOA assets, it is hard to manage SOA asset. This paper presents a construction method for domain SOA assets repository based on Use Case Maps(UCM). It uses UCM to describe the domain demands, and translates the domain demands into the marks and the relationships among marks in the UCM model. The marks and the relationships in the UCM model are mapped to SOA assets, so that domain SOA asset repository is obtained. The method realizes effective SOA asset management and supports SOA assets reuse.

【Key words】 SOA asset reuse; SOA asset relationship; SOA asset repository; Use Case Maps(UCM); mapping rule

DOI: 10.3969/j.issn.1000-3428.2011.02.017

1 概述

面向服务的体系架构(SOA)通过整合服务来构建软件解决方案, 目前已得到了广泛的应用^[1]。现有的 SOA 资产复用活动扩展到了设计模式、业务流程、合作模板、服务等较为宏观的程度^[2-5]。面向服务建模一个重要的环节是 SOA 资产复用, 方法诸如: (1)SBMC 方法^[2], 但 SOA 资产之间错综复杂的关系导致了该方法在进行资产复用时效率和成功率较低。(2)文献^[3]提出的一种全过程复用的服务建模方法, 其建立在本体系统之上, 而本体系统本身构建的复杂性限制了该方法的应用。

有效和成功复用 SOA 资产的关键是能否有效准确地梳理和组织 SOA 资产以及 SOA 资产之间的关系。对软件复用的研究和实践证明, 特定领域内的软件复用活动相对容易取得成功^[6]。

本文提出了基于 UCM(Use Case Maps)的领域 SOA 资产库构建方法: 采用 UCM 描述和分析应用领域需求, 并将 UCM 标记及其之间的关系映射到 SOA 资产上, 从而建立 SOA 资产库。另外, SOA 资产库中的 SOA 资产在应用领域内被广泛复用验证, 意味着它们的出错几率相对较小, 相应地会提高复用的成功率。

UCM 是一种表示场景的图示语言, 旨在方便人们在较高的抽象层次上理解系统行为的可视化表示方法^[7]。本文采用 UCM 描述领域需求是因为: (1)UCM 求精能力强, 场景可以被逐步地结构化和分层表示, 便于对应用领域子领域进行划分。(2)UCM 的存根、插件等标记可以很容易地将应用领域的可变性需求反映到模型中。(3)UCM 中的责任点、路径与 SOA 资产中的服务、业务流程的含义相通, 便于将领域需求反映到 SOA 资产中。但是利用 UCM 描述和分析领域需求也

有很多不足之处。本文需要明确每一个标记的具体功能和参数以及每一个标记之间的关系, 而传统 UCM 是非形式化的, 责任点、构件、存根等一般使用自然语言描述, 不能满足以上要求, 所以, 对传统的 UCM 进行了扩展。

2 领域 SOA 资产库构建思想

本文建立的领域 SOA 资产库由领域业务流程库(BP_B)、合作模板库(CT_B)、服务库(S_B) 3 个子库组成, 如图 1 所示。

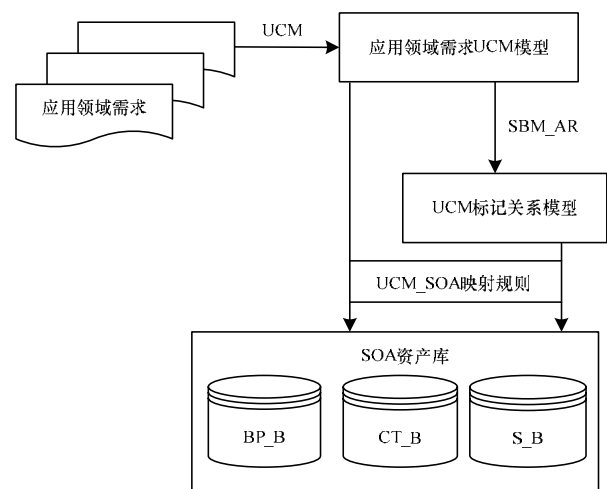


图 1 领域 SOA 资产库框架结构

该框架利用需求工程需求元模型方法进行领域需求分析,

基金项目: 辽宁省博士启动基金资助项目(20091031)

作者简介: 陈廷伟(1976 -), 男, 副教授、博士, 主研方向: 服务计算, 云计算; 杨艳辉、关长城, 硕士研究生

收稿日期: 2010-06-20 **E-mail:** yanhui988@163.com

并采用 UCM 描述应用领域需求，建立领域需求 UCM 模型。领域需求一般很复杂，本文通过存根-插件的方式将领域需求分为多个子领域需求。然后对子领域需求进行再次分解，直到分解为可由相应的服务完成的原子需求。每次分解得到的结果均采用 UCM 描述，即可得到需求 UCM 模型。

为了分析领域需求 UCM 模型标记之间的关系，提出了 SBM_AR(Scenario Based Model_Analysis of Relationships)方法，其主要思想是：根据得到的领域需求 UCM 模型，以路径为基准，分析责任点、存根、UCM 图之间的关系。本文分 3 种情况进行关系分析，分别是：存根-UCM 图的关系分析，同一路径上责任点(存根)的关系分析，不同路径上责任点(存根)的关系分析。另外，为了便于将 UCM 标记及标记之间的关系映射到合作模板上，采用三元组来表示标记之间的关系。领域需求 UCM 模型是从应用领域需求中得到的，所以，需求之间的依赖关系与 UCM 标记之间的关系是一致的。

以上工作的主要任务是分析和建立应用领域 SOA 资产库各种类型和各种层次的可复用资产。本文通过提出的 UCM_SOA 映射规则，将 UCM 标记及其之间的关系映射到应用领域 SOA 资产中，从而建立领域 SOA 资产库。其中将路径映射为业务流程，责任点映射为服务，关系映射为合作模板。所得到的领域 SOA 资产库包含领域业务流程库、合作模板库、服务库，可支持复用过程的不同阶段。

3 UCM 扩展和 SBM_AR 方法

3.1 UCM 扩展

针对采用 UCM 描述领域需求存在的不足之处，本文对 UCM 进行了如下扩展：

(1)为了便于分析标记之间的关系，为所有的 UCM 标记确定唯一的标识和名字。

(2)为了完整地表示业务流程、合作模板、服务等 SOA 资产概念，根据每种标记的功能做相应的扩展。具体扩展方法如下：

1)责任点采用 $RPE=<RPE_Id, RPE_Name, RPE_Function, PE_Id, UCM_Name>$ 表示，其中， RPE_Id 和 RPE_Name 分别表示责任点的唯一标识和名字； $RPE_Function$ 表示责任点的任务，由三元组 $<Action, In, Out>$ 组成， $Action$ 表示任务要执行的动作名， In 和 Out 分别表示此动作的输入、输出参数； PE_Id 表示责任点所在的路径； UCM_Name 表示责任点所在的 UCM 名字。

2)存根采用 $SE=<SE_Id, SE_Name, SE_Function, PE_Id, UCM_Id>$ 表示，其中， SE_Id 、 SE_Name 、 $SE_Function$ 与责任点的内容类似； UCM_Id 表示细化存根的子 UCM 图。

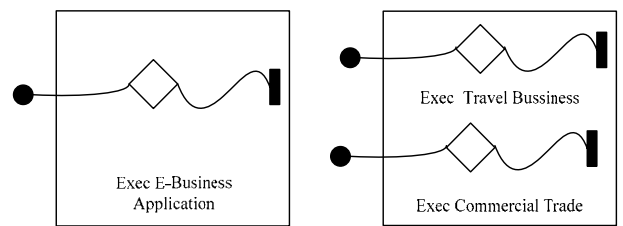
3)UCM 图采用 $UCM=<UCM_Id, UCM_Name, UCM_PS, SE_Id>$ 表示，其中， UCM_PS 表示在该 UCM 图上所有的路径集合； SE_Id 表示 UCM 图所细化的存根。

4)路径采用 $PE=<PE_Id, PE_Name, UE_S, UE_SE, UCM_Id>$ 表示，其中， UE_S 是该路径上所有责任点和存根组成的集合，其由 2 种类型的元素组成，分别是责任点(存根)和由责任点(存根)组成的集合 UE_SB ， UE_SB 指该路径分支上的责任点(存根)的集合，需要特别指出的是， UE_S 集合中的责任点(存根)按照该路径的方向先后有序，同一责任点后的 UE_SB 集合中元组属并列关系； UE_SE 表示该路径上互相排斥的责任点(存根)，由二元组 $<RPE(SE), RPE(SE)>$ 表示； UCM_Id 表示路径所在的 UCM 图标识。

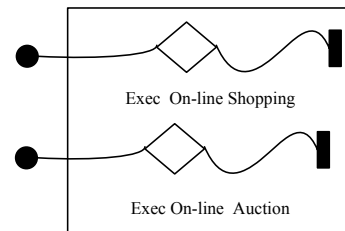
5)路径连接点表示不同路径的交汇点。路径连接点采用

元组 $PJP=<PJP_Id, PJP_Name, PJP_Pre, PJP_E, PJP_Post, UE_SE>$ 表示，其中， PJP_E 表示路径连接点责任点(存根)； PJP_Pre 和 PJP_Post 分别表示连接点前后的责任点(存根)，其内容与 UE_S 相同。

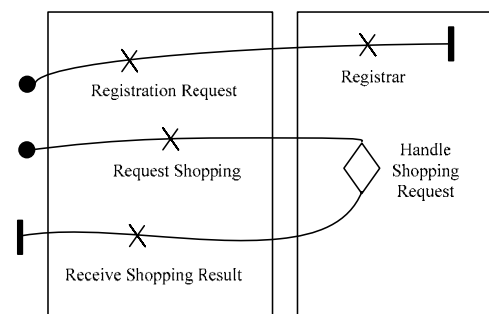
本文以一个简化电子商务应用领域展示 UCM 描述的领域需求模型，如图 2 所示。其中，存根表示具有较大功能但需要进一步细化的需求；路径表示系统活动的流程；责任点表示具有一定功能的原子需求(活动)。通过对 UCM 标记的扩展，UCM 图中的每一个责任点、存根、路径等都由一个唯一确定的元组表示。图 2(a)表示 E-Business Application 最外层需求；图 2(b)是对 E-Business Application 的细化，分解为 Travel Business 和 Commercial Trade 2 个子需求；图 2(c)是对 Commercial Trade 子领域的细化，分解为：On-line Shopping 和 On-line Auction 2 个子需求；图 2(d)是对 On-line Shopping 子需求的细化，包括 Registration 和 Shopping 2 条路径；图 2(e)是对 Handle Shopping 存根的细化。



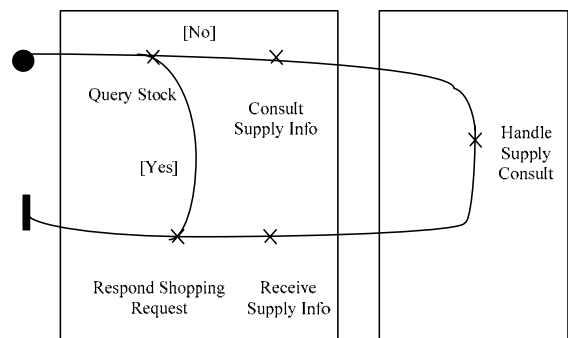
(a)E-Business Application 最外层需求 (b)E-Business Application 的细化



(c)Commercial Trade 子领域的细化



(d)On-line Shopping 子需求的细化



(e)Handle Shopping 存根的细化

图 2 简化的电子商务应用领域需求 UCM 模型

3.2 SBM_AR 方法

需求之间的依赖关系被分为静态依赖关系和动态依赖关系^[6]。为了便于将 UCM 图标记之间的关系映射到 SOA 资产上,本文采用如下的三元组来表示标记之间的关系: $UE_R = \langle UE_R_Id, UE_R_UE, UE_R_Type \rangle$ 。其中, UE_R_UE 表示关系所涉及的 UCM 标记,此项由元组 $\langle UE_Id, \dots, UE_Id \rangle$ 组成, UE_Id 表示 UCM 标记标识,需要特别指出的是,此元组的关系为前者 UE_Id 与后者 UE_Id 的关系; UE_R_Type 表示关系类型,包含文献[6]中提到的所有关系类型。

3.2.1 需求 UCM 模型中标记的形式化表示

为了方便关系的分析研究,本文将需求 UCM 模型所包含的 UCM 标记按不同的标记类型采用不同的集合表示,其中包括:

(1) $UCMS = \{UCM_i | 1 < i < n\}$ 。需求 UCM 模型中的 UCM 图表示为 $UCMS(UCM\ Set)$, 设该应用领域需求 UCM 模型的所有 UCM 图数量为 n 。

(2) $SES = \{SE_i | 1 < i < t\}$ 。每个 UCM 图中的存根表示为 $SES(SE\ Set)$, 设该 UCM 图上的所有存根数量为 t 。

(3) $PES = \{PE_i | 1 < i < q\}$ 。每个 UCM 图中的路径表示为 $PES(PE\ Set)$, 设该 UCM 图上的所有路径数量为 q 。

(4) $RPES = \{RPE_i(SE_i) | 1 < i < m\}$ 。该路径上的责任点(存根)表示为 $RPES(RPE\ Set)$, 设该路径上所有责任点(存根)的数量为 m 。

(5) $PJPS = \{PJP_i | 1 < i < p\}$ 。不同路径之间的所有连接点表示为 $PJPS(PJP\ Set)$, 设该领域需求 UCM 模型的路径连接点数量为 p 。

(6) $UE_RTS = \{Generalization, Decomposition, Required, Excluded, Serial, Collateral, Synthetic, Change\}$ 。需求之间的依赖关系表示为 $UE_RTS(UE_RT\ Set)$, 关系包含上文分析的所有关系类型。

(7) $UE_RS = \{UE_RS_i | 1 < i < r\}$ 。该领域所有的关系表示为 $UE_RS(UE_R\ Set)$, 该领域内根据下文规则获得的所有关系数量为 r 。

3.2.2 存根_UCM 图关系对应规则

存根_UCM 图关系对应规则包括:

规则 1 存根与所对应的具体化存根的 UCM 图为泛化关系。

$(\exists SE_i SES, UCM_j UCMS) (SE_i = UCM\langle SE_j \rangle) \rightarrow \langle UE_R_Id, \langle SE_i, UCM_j \rangle, Generalization \rangle$

规则 2 具体化存根的 UCM 图与所对应的存根为分解关系。

$(\exists UCM_i UCMS, SE_j SES) (UCM_i = SE\langle UCM_j \rangle) \rightarrow \langle UE_R_Id, \langle UCM_i, SE_j \rangle, Decomposition \rangle$

3.2.3 同一路径上责任点(存根)关系对应规则

设 $PE_i PES, UE_S \langle PE_i, UE_S \rangle, RPES = UE_S$, 则同一路径上责任点(存根)关系对应规则为:

规则 3 无分支路径上责任点(存根)之间为顺序关系。

$(\exists RPE_i RPES, RPE_j RPES) (\langle RPE_i, RPE_j \rangle UE_S) \rightarrow \langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Serial \rangle$

规则 4 路径不同分支上的责任点(存根)之间为静态约束关系。根据该路径的 UE_SE 项确定责任点(存根)之间是排斥关系还是需要关系。

排斥关系为:

$(\exists RPE_i RPES, RPE_j RPES) (\langle RPE_i, RPE_j \rangle UE_SE) \rightarrow \langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Excluded \rangle$

需要关系为:

$(\exists RPE_i RPES, RPE_j RPES) (\langle RPE_i, RPE_j \rangle UE_SB) (\langle RPE_i, RPE_j \rangle \notin UE_SE) \rightarrow \langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Required \rangle$

规则 4.1 若分支责任点(存根)之间为需要关系,则责任点(存根)之间为并行关系。

$\langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Required \rangle \rightarrow \langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Collateral \rangle$

规则 4.2 若分支责任点(存根)之间为需要关系,且该分支上的责任点需要在某一点会合,则它们之间为协同关系。

$(\exists RPE_i UE_SB, RPE_k RPES) (\exists RPE_j UE_SB, RPE_k RPES) (\langle RPE_i, RPE_k \rangle UE_S) (\langle RPE_j, RPE_k \rangle UE_S) \rightarrow (\langle RPE_i, RPE_j \rangle \notin UE_SE) \rightarrow \langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Synthetic \rangle$

3.2.4 不同路径上责任点(存根)之间关系对应规则

本文以路径的连接点为基准,分析不同路径上责任点(存根)之间的关系。

规则 5 不同路径上责任点(存根)之间为静态约束关系,根据路径连接点内 UE_SE 进一步确定责任点(存根)之间是排斥关系还是需要关系。

排斥关系为:

$(\exists PJP_i PJPS) (\exists \langle RPE_i, RPE_j \rangle (PJP_Pre\ PJP_Post)) (\exists \langle RPE_i, RPE_j \rangle UE_SE) \rightarrow \langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Excluded \rangle$

需要关系为:

$(\exists PJP_i PJPS) (\exists \langle RPE_i, RPE_j \rangle (PJP_Pre\ PJP_Post)) (\exists \langle RPE_i, RPE_j \rangle \notin UE_SE) \rightarrow \langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Required \rangle$

规则 5.1 若责任点(存根)之间是需要关系,则责任点(存根)之间为并行关系。

$\langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Required \rangle \rightarrow \langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Collateral \rangle$

规则 5.2 若责任点(存根)属于 PJP_Pre 元组,则责任点(存根)之间为协同关系。

$(\exists \langle RPE_i, RPE_j \rangle PJP_Pre) \langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Required \rangle \rightarrow \langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Synthetic \rangle$

规则 5.3 若责任点(存根)属于 PJP_Post 元组,则责任点(存根)之间为并行关系。

$(\exists \langle RPE_i, RPE_j \rangle PJP_Post) \langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Required \rangle \rightarrow \langle UE_R_Id, \langle RPE_i, RPE_j \rangle, Collateral \rangle$

规则 6 路径连接点处的责任点(存根)与其前后责任点(存根)之间的关系为顺序关系。

$(\exists \langle RPE_i, RPE_j \rangle PJP_Pre) (\exists RPE_k PJP_E) (\exists \langle RPE_m, RPE_n \rangle PJP_Post) \rightarrow \langle UE_R_Id, \langle \langle RPE_i, RPE_j \rangle, RPE_k, \langle RPE_m, RPE_n \rangle \rangle, Serial \rangle$

将以上映射规则应用到简化的电子商务应用领域需求 UCM 模型中,即可显式地得到 UCM 标记之间的关系。

4 UCM_SOA 映射规则

根据上文的分析可知,需求 UCM 模型中的责任点与 SOA 资产中服务的基本特征相同。存根可表示不同的子领域。路径是系统活动的流程,与业务流程的概念相同。责任点(存根)之间的关系表示为了完成某一活动不同责任点(存根)之间的协作关系,与合作的概念相同。为了将需求 UCM 模型中的标记和标记之间的关系映射到 SOA 资产中,本文提出了 UCM_SOA 映射规则。此规则分为如下 4 个部分:

(1) 存根-领域对应规则

将每一个存根的名字设为一个领域名,其中,应用领域

设为根领域。根据需求 UCM 模型,将 UCM 图中的存根设为该图的子领域,存根所对应的子 UCM 图中的存根为该领域的子领域,以此类推,直到搜索完所有 UCMS 集合中的 UCM 图为止。

(2)路径-业务流程对应规则

将路径的名字定义为业务流程的名字,将路径上 UE_S 元组中的责任点(存根)对应为业务流程中的活动,其活动的顺序与 UE_S 元组中责任点(存根)的顺序相同,从而完成一条业务流程的对应。按照以上方法,将 PES 集合中所有的路径均对应为相应的业务流程。由于领域的实现需要通过业务流程来体现,因此本文将领域和业务流程放到领域业务流程库中。根据路径所在的 UCM 图,将相应的业务流程对应到 UCM 图所细化的领域(存根)中,从而形成领域层次化、业务流程领域化的领域业务流程库。

(3)责任点-服务对应规则

根据责任点的 RPE_Function 元组,确定责任点要完成的任务和输入、输出参数。从现有的可复用服务库中搜索与所确定要完成的任务和输入、输出参数相匹配的可复用服务。

(4)关系-合作模板对应规则

一个合作模板关联一组为实现某一活动期望服务的合作,期望服务指活动期望由具有某种功能的服务相互协作实现活动所要求的功能。通过 SBM_AR 方法得到的关系较为零散。业务流程所要完成的活动关联一组可实现该活动的合作模板。为了方便关系的复用,本文以路径为基准对所得到的关系进行整合。由于 SBM_AR 方法的关系对应规则分为 3 种类型,因此分为 3 种整合方法:1)同一路径上责任点(存根)关系对应规则所得关系的整合方法。挑选出每条路径上 UE_S 中的所有责任点(存根),将该路径上责任点(存根)之间的所有关系整合为一个合作模板,合作模板的名字用相应的路径命名。2)不同路径上责任点(存根)关系对应规则所得关系的整合方法。由于不同路径之间是以责任点进行连接的,责任点分属不同的路径,因此以连接点为基础对所得的关系进行整合。每个连接点整合为一个合作模板,合作模板的名字用相应的连接点的名字命名。3)存根-UCM 图关系对应规则所得关系的整合方法。在需求 UCM 模型中存根主要用于划分子领域。把具体化该存根的 UCM 图中的合作模板作为该子领域(存根)的内嵌子合作模板。采用以上方法即可得到有组织、结构化、便于搜索复用的合作模板。

通过以上对应规则,可得简化的电子商务应用领域的 SOA 资产库。采用方框表示领域,框架矩形表示业务流程,领域业务流程库如图 3 所示。

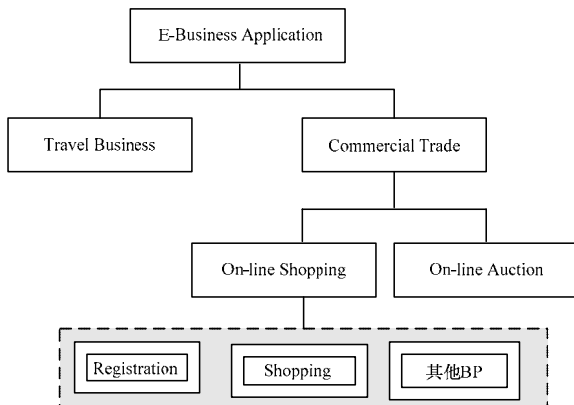


图 3 领域业务流程子库

在图 3 中,最上方为根领域(电子商务应用),然后将根领域分解为旅游业务和商业贸易 2 个子领域,最靠近灰色框的一层子领域为原子领域。灰色框内为该原子领域实现所需的业务流程。限于篇幅,选取部分图示罗列。

图 4 为合作模板库中在线购物子领域的合作模板,图中连线表示不同服务间的关系。

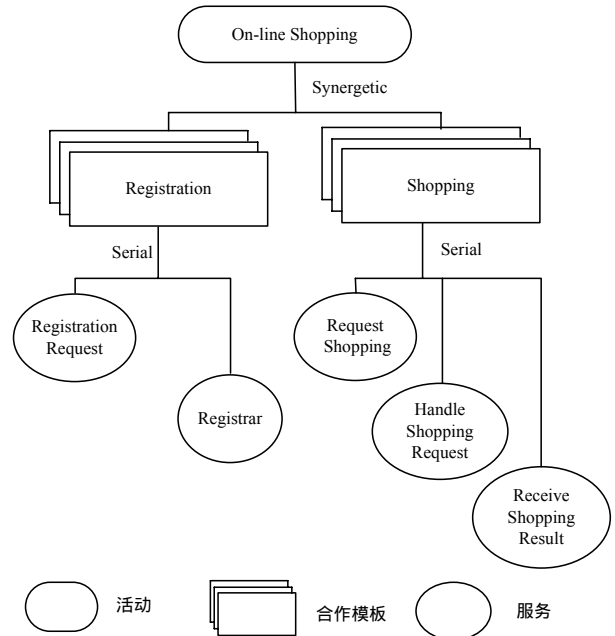


图 4 合作模板子库中在线购物合作模板

5 结束语

本文针对 SOA 资产复用的效率和成功率较低的问题,提出了组织管理 SOA 资产的方法,即基于 UCM、以领域为基础构建领域 SOA 资产库,在领域内进行 SOA 资产复用时,可在 SOA 资产库中方便、快捷、准确地找到所需的 SOA 资产,从而在很大程度上缩短 SOA 资产复用活动的时间。进一步的工作主要包括研究如何以领域 SOA 资产库为基础建立服务模型。

参考文献

- [1] 张良杰, 张嘉, 蔡弘. 服务计算[M]. 北京: 清华大学出版社, 2007.
- [2] 李长云, 阳爱民, 满君丰, 等. 一种面向按需集成服务的业务模型构造方法[J]. 计算机学报, 2006, 29(7): 1095-1104.
- [3] 吴步丹, 金芝, 赵彬. 面向服务的建模: 一种全过程复用的方法[J]. 计算机学报, 2008, 31(8): 1293-1308.
- [4] 唐文伟, 沈备军, 陈德来. 模型驱动的业务流程建模工具[J]. 计算机工程, 2009, 35(24): 262-264, 267.
- [5] Jamshidi P, Sharifi M, Mansour S. To Establish Enterprise Service Model from Enterprise Business Model[C]//Proc. of 2008 International Conference on Services Computing. [S. l.]: IEEE Press, 2008: 93-100.
- [6] 李玉琴. 需求工程与软件产品线若干关键技术研究[D]. 上海: 复旦大学, 2007.
- [7] Buhr R J. Use Case Maps as Architectural Entities for Complex Systems[J]. IEEE Transactions on Software Engineering, 1998, 24(12): 1131-1155.