

# 一种基于知识复用的动态服务组合方法

刘颖<sup>1</sup> 胡海涛<sup>2</sup>

(清华大学中文系 北京 100084)<sup>1</sup> (华北电力大学计算机系 北京 102206)<sup>2</sup>

**摘要** 如何在业务层面即时动态组合服务,按需创建应用,是一个具有挑战性的课题。提出了一种适合业务用户使用的基于知识复用的服务组合方法,该方法充分利用领域专家的知识 and 经验,将其封装成组合模板透明地提供给用户,使用户可以在业务层面以一种大粒度的方式描述需求,建立业务应用,从而提高了业务用户主动参与面向服务的应用构造的能力和构造效率。

**关键词** 服务组合,组合模板,知识复用

## Approach to Dynamic Services Composition Based on Knowledge Reuse

LIU Ying<sup>1</sup> HU Hai-tao<sup>2</sup>

(Department of Chinese Language and Literature, Tsinghua University, Beijing 100084, China)<sup>1</sup>

(Department of Computer Science, North China Electric Power University, Beijing 102206, China)<sup>2</sup>

**Abstract** How to make full use of Web services and construct application by composing existing services on demand in the dynamic and open environment is a challenge issue. This paper presented an approach to larger-granularity service composition for business-users, which encapsulated domain knowledge into prefabricated and modifiable templates. By this approach, business-users can construct applications just like assembling hardware by composing reusable and larger-granularity modules, which facilitates automated service composition and enhances abstract level to some extent.

**Keywords** Services composition, Composition template, Knowledge reuse

## 1 引言

今天,Web服务作为炙手可热的技术,如何应用到企业的IT系统和商业流程之中,并给企业带来直接的经济效益,一直备受国内外企业界和学术界的高度关注。随着面向服务架构(SOA, Service-Oriented Architecture)的流行,人们开始着手构建面向服务的应用。面向服务的计算技术为解决广域网范围内应用集成带来了许多好处,可以使企业应用摆脱面向技术的解决方案的束缚。企业环境中传统的单个应用程序是无法包容业务用户的各种需求的,即使是一个大型的ERP解决方案,仍然不能弥补这个需求在不断膨胀、变化的缺口,因而无法对市场变化做出快速反应。业务用户只能通过不断开发新应用、扩展现有应用程序来艰难地支撑其变化的业务需求。通过将注意力放在服务上,应用程序能够集中起来提供更加丰富、目的性更强的商业流程。因而,基于SOA的企业应用系统通常会更加真实地反映出与业务模型的结合。

相比传统构件“购买而不创建(buy, don't build)”的开发哲学<sup>[1]</sup>,服务“使用而不拥有(use, don't own)”的特点更体现了软件所有权和使用权的分离,用户只需根据服务提供者注册的信息直接调用所需服务,这种“按需调用”的软件使用模式更多地体现了以用户为中心的设计理念。作为构造应用的基本模块,为提高重用性和应用构造的效率,各个服务并不是

孤立地使用,很多情况下用户需要将服务组合起来以满足复杂的需求。然而在动态、开放的计算环境下,如何让最终业务用户像组装硬件一样自行组装出面向服务的应用,从而更好地适应业务变化,满足即时的、个性化的需求等问题倍受关注。为此,我们提出了一种基于知识复用的动态服务组合方法——Keen(A Knowledge Reuse Based Approach for Dynamic Services Composition),方法的核心思想是将领域专家的知识 and 经验封装成组合模板,透明地提供给用户,使用户可以在业务层面以一种大粒度的方式描述需求,建立业务应用。

本文第2节介绍Keen方法的基本思路;第3节给出服务组合模板的定义;第4节介绍业务用户如何应用组合模板构造应用;第5节讨论国内外的相关研究;最后是结论和下一步的工作。

## 2 Keen方法的基本思路

Keen方法基于两个基本假设:(1)客观世界是有规律的,发生在某一领域的现象必定遵从一定的规律,特定领域具有内聚性和稳定性<sup>[2]</sup>。一个领域的规约和实现知识的内聚性,使得可以通过一组有限的、相对较少的可复用信息来解决大量问题。领域的稳定性使得获取的信息可以在较长的时间内多次复用。(2)在特定领域存在这样的一类人,他们对领域中的业务知识较为了解,同时又具备一定的计算机知识,能完整

到稿日期:2010-04-07 返修日期:2010-08-15 本文受清华大学亚洲研究基金(2005C-2),华北电力大学博士基金(200722018)资助。

刘颖(1969—),女,博士,副教授,主要研究方向为中文信息处理、知识管理、机器翻译,E-mail:yingliu@tsinghua.edu.cn;胡海涛(1973—),男,博士,讲师,主要研究方向为软件集成、面向服务计算、服务网络。

地描述业务过程涉及到的资源、信息等不同方面的视图,比一般的新手或普通用户更具有丰富的经验。如果能将他们的这些知识和经验很好地抽取出来加以利用,无疑会提高普通的业务用户进行应用构造的效率。

假设(1)说明了在特定领域的相似问题具有相似的求解,而且相似问题有可能重复发生,因此提供服务组合模板将这些解决相似问题的知识和经验表示出来,是必要的。假设(2)则说明由于存在这样的领域专家,因此提供可复用的知识是可能的。

基于以上的分析和假设,Keen方法遵循“构建与使用相剥离”的原则,其基本思路是:面向服务环境下业务用户通过组合服务主动参与应用构造还存在一定的困难;组合服务的过程涉及到分析、设计与优化业务过程,这是一项依赖于专业知识和经验的工作;通过将面向服务的应用构造从基础设施构造和应用构造两个层面按角色分工,领域专家可以通过服务组合模板抽取组合知识,提供一些业务过程的较好实现;最终业务用户根据自己的个性化需求,通过使用组合模板提高应用构造的效率与质量。具体如图1所示。

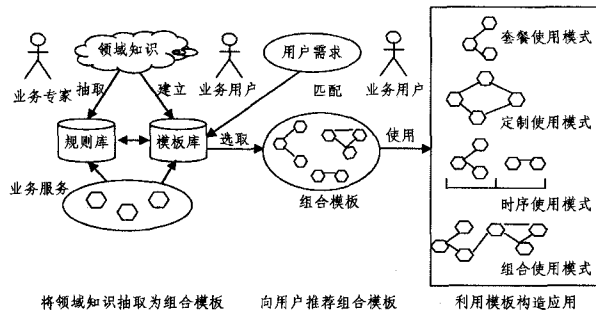


图1 Keen方法的思路图

正如文献[2]中的介绍,基于复用的思想是提高软件生产率的重要途径。而复用的核心不是软件,而是知识。Keen通过创建服务组合模板将领域专家的组合知识表示出来,通过利用组合模板多种使用模式来辅助用户主动参与应用构造。这样通过服务组合模板,业务用户就可以以一种大粒度、可重用的方式组合业务服务进而构造应用,从而提高面向服务应用构造的自动化程度和抽象层次。

### 3 服务组合模板

服务组合模板从软件复用的角度可以看作是一种组合构件,它是抽象的系统特征单元,具有封装和信息隐蔽的特性,其功能由它的接口定义。服务组合模板是可配置和共享的,这也是基于构件开发的基石。

服务组合模板封装了领域专家的知识和经验,通过对领域需求的分析,结合行业背景、业务规则,归纳总结出一些规则性的工作模式。利用业务服务和基本的组合逻辑,构造好一些可重用的业务流程框架。组合模板将业务流程中各个业务服务间的逻辑关系封装在模板内部,屏蔽掉了很复杂的构造细节,对用户来说类似于一个“黑盒式”的业务服务,具有输入、输出调用接口和非功能属性、业务服务、分支路由等配置接口。这样利用它们就可以省去许多专业的编程工作,用户可以通过直接套用、裁剪、扩充、参数配置、滚动组合等手段,类似于使用一个基本的业务服务一样来使用它们构建应用。下面给出它的形式化定义。

定义1 服务组合模板是一个六元组:  $SCTem = \{BasicInfo, BSSet, CallInterface, ConfigInterface, TemStructure, DataLink\}$ ,其中:

- *BasicInfo* 是记录组合模板的基本信息,通过它业务用户可以识别不同的组合模板;
- *BSSet* 是组合模板包含的业务服务的集合;
- *CallInterface* 是服务组合模板的调用接口,是运行时使用者和模板交互的描述;
- *ConfigInterface* 是组合模板的配置接口,包括模板的非功能属性、服务资源、分支路径和运行起止时间的配置信息;
- *TemStructure* 是组合模板的逻辑结构,是由业务服务、组合逻辑和括号构成的表达式;
- *DataLink* 是组合模板的内部数据关联。

图2为服务组合模板的结构图。

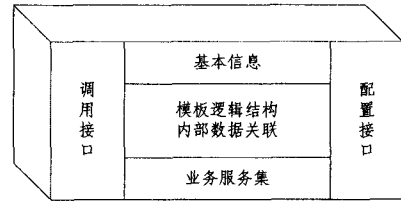


图2 组合模板的结构图

定义2—定义12进一步介绍了服务组合模板中各组成元素的基本内容。

定义2 组合模板的基本信息 *BasicInfo* 是业务用户用以识别不同组合模板的信息,它包括多项条目,可形式化表示为  $BasicInfo = \{FPDes, Name, Semantic, Provider, Version, Date\}$ ,其中:

- *FPDes* 是模板的功能描述,对应一个业务用户可以理解的功能描述字段,通过该功能描述业务用户能够获得组合模板的较为详细的功能信息,类型为 *String*;
- *Name* 是模板的名称,类型为 *String*;
- *Semantic* 是模板的语义标签,是模板按照业务功能分类对应的一个领域业务本体;
- *Provider* 是模板的提供者,类型为 *String*;
- *Version* 是模板的版本信息,类型为 *String*;
- *Date* 是模板颁布日期,类型为 *Date*。

这些信息由领域专家在创建模板时提供,用户可以根据这些信息进行基于关键字的模板检索,如根据模板名称、提供者、模板颁布日期等,检索满足条件的组合模板。

定义3 组合模板的业务服务集  $BSSet = \{(bs_1, optional), (bs_2, optional), \dots, (bs_n, optional)\}$ 。其中  $bs_i$  为业务服务,  $i=1, \dots, n$ 。  $optional \in \{true, false\}$ ; *optional* 表示该业务服务在组合模板中是否可选,是 *boolean* 型变量, *optional* 取 *true* 时表示该业务服务是可选的,取 *false* 时表示该业务服务是必须的,默认情况下为 *false*。

实际上,在组合模板的逻辑结构中已经包含了业务服务,这里之所以将业务服务集单独定义出来,是为了在用户选取组合模板时,直接通过业务服务集的包含关系向用户推荐组合模板。详细的推荐算法参见文献[3]。业务服务是服务组合模板中的一个核心构成要素,下面给出它的定义。

定义4 业务服务是一个三元组  $BS = \{BasicInfo, Fun-$

$Info, NFPInfo\}$ , 其中:

1)  $BasicInfo$  表示服务的基本信息, 与组合模板的基本信息类似。

2)  $FunInfo$  表示服务的功能属性,  $FunInfo$  是一个二元组:  $FunInfo = \{Input, Output\}$ , 其中:

•  $Input = \{Input_1, Input_2, \dots, Input_m\}$ , 表示服务的输入参数集合, 其中每一个输入参数  $Input_i$  是一个三元组  $Input_i = \{name, semantic, type\}$ , 其中  $name$  表示参数名称,  $semantic$  表示参数所对应的语义标签,  $type$  表示参数的数据类型。

•  $Output = \{Output_1, Output_2, \dots, Output_n\}$  表示服务的输出参数集合, 每一个输出参数与输入参数的定义类似。

3)  $NFPInfo$  表示服务的非功能属性, 由多个非功能属性条目组成。每个非功能属性条目是一个四元组:  $NFPInfo = \{Name, Semantic, Value, Unit\}$ 。其中  $Name$  表示非功能属性名称;  $semantic$  表示该名称所对应的语义标签;  $Value$  表示非功能属性的取值, 取值可以为离散型或连续型两种类型;  $Unit$  表示非功能属性取值的单位。

这里定义的业务服务是从功能上定义的, 本质上是一组功能相同或相近的具体的 Web 服务的一个聚集, 而不是一个可执行的 Web 服务。它本身也是一个模板, 这样定义的好处是用户可以根据需要在运行时根据非功能属性的要求进行动态绑定, 这样可以提高服务组合模板的灵活性。关于业务服务更详细的讨论和定义参考文献[4], 这里不再赘述。

**定义 5** 组合模板的调用接口  $CallInterface$  是运行时使用者和模板交互的描述, 包括  $k$  个有顺序的输入输出调用接口, 即  $CallInterface = \{Interface_1, Interface_2, \dots, Interface_n\}$ ,  $k=1, \dots, n, n$  为模板的交互次数。其中第  $i$  次交互的调用接口表示为  $Interface_i = \{Input_i, Output_i\}$ , 其中  $Input_i = \{Input_{i1}, Input_{i2}, \dots, Input_{ik}\}$ ,  $k$  为本次调用接口种的分支路径数,  $Input_{ij} = \{\langle input_{ij1}, optional \rangle, \langle input_{ij2}, optional \rangle, \dots, \langle input_{ijm}, optional \rangle\}$ ,  $Output_i = \{Output_{i1}, Output_{i2}, \dots, Output_{im}\}$ ,  $Output_{ij} = \{output_{ij1}, output_{ij2}, \dots, output_{ijn}\}$ ,  $input_{ijl} \in BSSet$ .  $Input, output_{ijl} \in BSSet$ .  $Output, j = 1, \dots, m, k = 1, \dots, n$ .  $BSSet$ .  $Input$  是模板中所有业务服务的全部输入集,  $BSSet$ .  $Output$  是模板中所有业务服务的全部输出集,  $optional \in \{true, false\}$ ,  $optional$  表示是否来自分支结构,  $optional$  取  $true$  时表示来自分支结构, 取  $false$  时表示不是来自分支结构。

说明: 通过对调用接口的分批次定义, 可以避免一次输入过多的信息, 也解决了由于组合运行不同的分支的不确定性带来的输入冗余问题, 具有一定的灵活性。

**定义 6** 组合模板的配置接口  $ConfigInterface = \{NFPConfig, BSConfig, RouteConfig\}$ , 其中:

•  $NFPConfig$  是模板的非功能配置接口, 我们用它描述模板的非功能属性信息, 包括执行时间、使用价格、可靠性、服务范围、调用的前提条件和运行的效果、访问的权限等。它是一个四元组, 具体定义与业务服务的  $NFPInfo$  一样。

•  $BSConfig$  是业务服务的配置接口, 表示在应用构造时, 业务用户可以对模板中的某些业务服务进行具体配置。  $BSConfig = \{BSSource, BSTarget\}$ , 其中  $BSSource$  为待绑定的业务服务,  $BSTarget$  为绑定的服务, 可以是业务服务, 也可以是具体的 Web 服务。

•  $RouteConfig$  是分支结构的路径配置信息。  $RouteConfig = \{Condition, BranchDes\}$ ,  $Condition$  是分支的选择条件,  $BranchDes$  是分支路径的描述信息。

说明: (1) 这里的非功能属性配置接口是对整个组合模板而言的。因为针对单个业务服务的非功能属性在 VINCA 流程已经存在, 而业务服务配置接口和分支路径配置接口是对组合模板中的业务服务而言的。

(2) 对于业务服务的配置接口  $BSConfig$ , 其配置目标可以是业务服务也可以是 Web 服务, 这是服务组合模板的一个特点。即模板中的业务服务可以是抽象的业务服务, 通过这个配置接口实例化为一个特定的业务服务。这体现了模板抽象共性、分化个性, 可灵活配置服务资源的特点。

**定义 7** 组合模板的逻辑结构  $TemStructure$  是由业务服务、基本组合逻辑和括号构成的表达式, 表示服务组合模板中包含的各业务服务间的逻辑关系。

从逻辑结构的完备性和便于业务用户使用的角度出发提出了 4 种基本组合逻辑, 作为组合模板中各业务服务的连接器。通过这些基本组合逻辑对业务服务进行组合, 可以构造出服务组合模板和业务应用。这 4 种基本组合逻辑是顺序、并行、选择和重复。顺序 (Sequence) 表示两个服务之间的关系, 前一个服务结束之后, 后一个服务才能开始。并行 (Parallel) 表示两个服务之间的关系, 两个服务可以并发地执行, 没有先后的制约关系。选择 (Choice) 表示两个服务之间的关系, 两个服务只能有一个可以执行, 一个服务执行后, 另一个服务将失去执行的机会。重复 (Repeat) 表示一个服务在满足一定条件的情况下, 可以多次反复地执行, 直到终止条件发生。

这里我们用  $BSSet$  表示业务服务的集合,  $bs_i \in BSSet$ ,  $bs_i$  为业务服务集中的一个业务服务。顺序、并行、选择和重复的形式化定义如下。

**定义 8**  $Sequence$  是  $BSSet$  上的二元关系, 简记为  $Seq$ 。

$Sequence = \{\langle bs_i, bs_j \rangle \mid bs_i, bs_j \in BSSet \text{ 当且仅当 } bs_i \text{ 结束后 } bs_j \text{ 才能开始}\}$ 。

**定义 9**  $Parallel$  是  $BSSet$  上的二元关系, 简记为  $Par$ 。

$Parallel = \{\langle bs_i, bs_j \rangle \mid bs_i, bs_j \in BSSet \text{ 当且仅当 } bs_i, bs_j \text{ 能够并行执行}\}$ 。

**定义 10**  $Choice$  是  $BSSet$  上的二元关系, 简记为  $Cho$ 。

$Choice = \{\langle bs_i, bs_j \rangle \mid bs_i, bs_j \in BSSet \text{ 当且仅当 } bs_i, bs_j \text{ 之间只有一个可以执行}\}$ 。

**定义 11**  $Repeat$  是  $BSSet$  上的  $n$  元关系, 简记为  $Rep$ 。

$Repeat^n (bs_i) = \{\langle Repeat^{n-1} (bs_i), bs_i \rangle \mid bs_i \in BSSet \text{ 当且仅当 } bs_i \text{ 满足一定条件后 } Repeat^{n-1} (bs_i), bs_i \text{ 顺序执行}\}$ 。

领域专家可以利用这些逻辑联接词构造服务组合模板。业务用户也可以利用它们对组合模板进行适当的修改, 以构造能满足其个性化需求的应用。

组合模板内部封装的逻辑结构就是一个由业务服务和这些基本组合逻辑以及括号构成符号表达式, 如表达式  $bs_1 Seq (bs_2 Par bs_3) Seq bs_4$  就可以表示业务服务  $bs_1$  执行后,  $bs_2$  和  $bs_3$  并行执行, 再执行  $bs_4$  这样的业务流程。

**定义 12** 组合模板的数据关联  $DataLink = \{DateSource, DataTarget\}$ 。其中  $DateSource$  表示数据联接中的数据来源, 其可能取值于某个业务服务的输出, 也可能来自手工

输入; *DataTarget* 表示数据联接中的数据目标, 是业务服务的输入。

图 3 是服务组合模板的元模型。

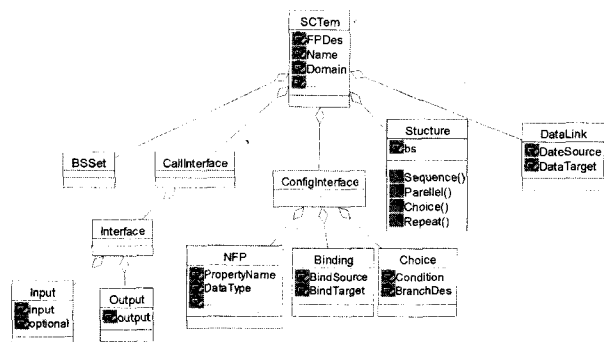


图 3 服务组合模板元模型

服务组合模板的一个主要作用就是隐藏关于业务流程的某些细节, 为用户提供业务流程在不同层次上的视图, 也就是业务流程的抽象, 即不同的使用者从不同的角度去观察业务流程中的逻辑所得到的结果。对最终用户(模板使用者)来说, 了解业务流程中用来表示业务流程的逻辑结构的细节没有太大的必要, 他们关心业务流程提供哪些功能以及实现这些功能在服务质量上的差别。对于领域专家(模板设计者)来说, 他们关心的是各个业务服务以怎样的逻辑结构连接在一起才能使整个业务流程效率更高、健壮性更强。

服务组合模板是和具体的应用领域密切相关的, 因此它的设计依赖于特定领域专家的知识、经验和相关的业务流程。如医疗专家诊断的服务组合模板是将专家的知识 and 经验用 4 种逻辑结构进行连接而成, 而保险理赔模板则是保险公司理赔业务流程的具体服务化实现。由于服务组合模板和领域密切相关, 因此现阶段我们只是针对一些特定领域制定了一些模板, 解决一些具体的问题, 而它们之间的完整性和正交性还缺乏相应的理论支持, 这将是今后工作的方向。

#### 4 基于组合模板构造应用

Keen 中服务组合模板的使用者是业务用户, 在定制上和使用上又与特定领域密切相关。因此在组合模板的使用方式上根据用户需求复杂度和用户“编程”能力并结合不同领域特色分为如下 4 类。

(1) 套餐使用模式: 又称为简单使用模式或直接使用模式。这种情形比较简单, 用户使用的只是一个服务组合模板, 并不涉及几个组合模板之间的交互或组合。之所以称为套餐使用模式, 是因为这种情况下用户不对组合模板进行任何改动, 要么不用该组合模板, 要么就全部拿来直接套用该组合模板。

(2) 定制使用模式: 这种情形相对也比较简单, 用户使用的只是一个服务组合模板, 也不涉及几个组合模板之间的交互或组合。但是与套餐使用模式不同, 这个组合模板又不能完全满足用户的实际需求, 需要在此基础上对组合模板进行适当的修改, 根据现有服务组合模板及用户需求的覆盖程度和冗余程度分为裁减使用模式和扩充使用模式两种。

(3) 时序使用模式: 这种模式适用于需求相对复杂的用户, 涉及到多个组合模板之间的交互。但是多个模板之间的交互又相对比较简单, 每个组合模板都与一个时间属性密切

相关, 可以按时序逻辑来安排这些模板, 而并不涉及其他如选择、循环等复杂逻辑。这种情况比较适合于个人行程安排类的业务应用。

(4) 组合使用模式: 又称为高级使用模式。这种方式对用户的要求最高(需要懂得一些计算机知识, 主要针对能力较强、需求又比较复杂的用户)。这样的组合模板相当于一个基本的编程单位, 即业务服务。用户可以通过前面介绍的基本组合逻辑将这些服务组合模板连接起来, 以实现其复杂的需求。

限于篇幅, 下面重点对套餐使用模式进行讨论。

对于组合模板, 我们按其复用的方式将其分为黑盒模板和白盒模板两种。对于组合模板定义中的 *BSSet*, 若 *BSSet* 中存在可选的业务服务  $bs_i$ , 则该组合模板为白盒模板, 否则模板为黑盒模板。用户可以对白盒模板的内部结构进行更改, 也可以将其作为一个类似于业务服务的组合单元通过组合逻辑关联起来, 形成新的模板或应用。对于黑盒模板不可以对其再进行修改或组合, 只能对其进行适当的个性化配置从而形成应用。

为了确保用户在使用服务模板解决实际业务需求时, 能够使用尽可能少的模板, 从而快速构造业务系统, 我们在文献 [3] 中定义了模板的覆盖度和冗余度的概念。而套餐使用模式主要适合用户的需求和组合模板的覆盖度为 1 而冗余度为 0 的情形。套餐使用模式是一种直接使用模式, 不对组合模板进行任何改动, 用户的需求完全包含在服务组合模板中, 因此这种使用模式也最简单, 适合没有任何“编程”能力的“傻瓜型”用户, 只需用户检索出相应的组合模板后直接对其非功能属性进行配置即可使用。具体使用步骤如下。

Step1 利用文献 [3] 中的模板推荐算法查找到满足需求的组合模板, 这里要求用户设置的参数  $user\_cov$  为 1,  $user\_red$  为 0;

Step2 若查找到的服务组合模板为多个, 可通过预览功能从中选择一个, 继续执行 4);

Step3 若查找不到满足要求的组合模板, 可以放宽冗余度(令  $user\_red > 0$ ) 或降低覆盖度(令  $0 < user\_cov < 1$ ), 选择有冗余的或部分满足用户需求的组合模板, 通过裁减或扩充模式构造应用;

Step4 对选择的模板进行个性化参数配置, 包括对模板的调用接口和配置接口的配置。分别说明如下:

- 对调用接口的配置

*CallInterface* 是组合模板的调用接口, 是用户和模板交互的描述, 包括  $k$  个有顺序的输入输出调用接口,  $CallInterface = \langle Interface_1, Interface_2, \dots, Interface_n \rangle, k = 1, \dots, n$ .  $n$  为模板的交互次数。因此有两种交互模式: 一种是简单交互模式, 即  $k=1$  时, 也就是用户和模板的交互次数为 1, 这时用户可以在模板使用之前全部设定输入和有关的数据关联; 另一种是分时交互模式, 即  $k > 1$  的情况, 这主要是由于分支结构的存在使得模板的输入具有不确定性。对于顺序执行的业务服务, 后一步的输入可能需要参考前一步的输出。由此在使用模板时需要多次交互、分批的输入。服务组合模板的一个特点——分时交互就是通过这个接口体现的。

- 对配置接口的配置

*ConfigInterface* 是组合模板的配置接口, 描述了模板定

义中的不确定部分,包括非功能属性配置信息 *NFPConfig*、业务服务配置信息 *BSCConfig* 和分支结构的路径配置信息 *RouteConfig*。

- 对 *NFPConfig* 是指允许用户设定整个模板或模板中的部分业务服务的非功能属性,也就是设定模板定义 *NFP* 中的 *Value* 值。根据不同的情况又可分为单值的设定和值区间的设定两种情况。

- 对 *BSCConfig* 的配置包括两个方面:一方面对于模板中可能提供的抽象业务服务由用户配置成特定的业务服务,其原理是根据业务服务“动词+名词”的定义将名词部分替换成具体事物名词,如将“旅游模板”中的“预订景点门票”这个抽象业务服务通过名词替换配置为“预订长城门票”;另一方面是指允许用户把部分具体服务预先绑定到组合模板的特定服务资源上,这是因为用户虽然使用的是整个服务组合模板,但并不是对模板中的业务服务的具体资源都要到运行时才由系统动态推荐,而是对其中的一些特定资源预先指定(这些资源可能是用户经常使用的、信誉好的或用户偏爱的等等),即分别设定 *BSCConfig* 中的 *BSSource* 和 *BSTarget*。

- 对 *RouteConfig* 的配置是指允许用户设定模板的分支结构中的判定条件。即重新设定模板中的 *RouteConfig* 的 *Condition*。比如“旅游模板”中对某个分支的设定可能是这样:若服务“查询天气”的结果为“晴”,选择下一步服务为“浏览故宫”。但是某个用户可能更喜欢雨天游故宫,所以他可将分支重新设定为若服务“查询天气”的结果为“阴”,选择下一步服务为“浏览故宫”。

## 5 国内外的相关工作

一般说来,从自动化程度方面对服务组合的方法可以分为手工(Manual)服务组合、半自动(Semi-automated)服务组合和自动(Automated)服务组合<sup>[5]</sup>。

手工服务组合要求用户通过图形化界面或文本编辑器手工编写一个流程脚本语言,然后提交给流程的执行引擎解释执行。Triana 提供了一套图形化的用户交互界面,允许用户使用关键字搜索的方法从 UDDI 注册中心选择合适的服务并通过拖拽的方式将服务组合起来,并最终通过 P2P 的方式运行组合的应用<sup>[6]</sup>。BPWS4J 通过内嵌的 Eclipse 插件帮助用户编辑基于 XML 的组合图,并将组合后的 WSDL 文件一并提交给解释执行引擎。这些工作都有着共同的问题:当服务数量比较大时,服务的发现和选取的扩展性差;其次这些系统对用户的要求较高,它们都要求用户具备很底层的专业知识,尤其是 IT 相关的知识。如在 BPWS4J 情形下,要求用户能在 XML 层次上建立 workflow,这对于不具备编程知识的业务用户来说有些不太现实。尽管 Triana 提供了图形化的拖拽界面,但是对于规模较大的组合流程来说还是不可行的。最近, Mandell 等人的工作扩展了 BPWS4J,允许在运行时选取和替换服务<sup>[7]</sup>。

半自动服务组合技术在人工服务组合的基础上前进了一步,通过在组合过程中向用户提供“语义建议”以支持服务的选取。用户仍需要选取特定的服务并按照一定的逻辑组合起来以构造应用。Sirin 提出了一个系统,可根据每个阶段的语义兼容性来选取服务,产生的 workflow 可以被执行<sup>[8]</sup>。Cardoso 和 Sheth 等提出了一个框架,通过为用户推荐服务来帮助用

户组合,实现的过程是将用户的特定服务模板(Service Template)与服务对象(Service Object)进行匹配来完成<sup>[9]</sup>。尽管这些系统克服了手工组合方式的一些缺点,但是它们仍然存在不足:系统的扩展性问题依然存在。服务的筛选过程可能会为用户推荐大量的服务,从而造成用户无法完成对合适服务的选取。另外,对于服务不可用后也没有对应的处理机制。如文献<sup>[9]</sup>的工作中,如果 Service Template 无法匹配到合适的 Service Object,那么组合将会失败。文献<sup>[8]</sup>中,组合的服务逻辑在执行过程中如果被使用的服务不可用,那么组合也会失败。

自动服务组合技术主要是通过人工智能规划或相似的技术来实现整个的组合过程的自动化。McIlraith 和 Son 使用一个基于 Agent 的 Web 服务组合框架<sup>[10]</sup>,通过使用通用的程序和语义标签服务来指导服务组合。该框架假定存在一个通用的程序,但实际上如果不存在对应的通用程序,那么组合将无法继续。这有点类似一个模板。SWORD 通过使用基于规则的服务描述来实现自动化服务组合<sup>[11]</sup>。服务被表示成逻辑规则,这些规则表示服务相关联的输入输出。用户指定组合的初始状态和最终状态,基于此,planner 尝试着将服务连接起来满足用户的需求。该系统要求用户能够明确地描述组合的两个状态,而且没有自动的服务发现机制。因此,如果不能发现需要的服务,组合可能会失败。此外,组合是基于特定的服务来实现的,当服务不再可用时共享会很困难。

通过对当前服务组合技术的简单讨论,可以看出手工的服务组合对用户的知识要求很高。而当前越来越多的参与面向服务应用开发的是直接面向业务应用的业务用户,因此手工的服务组合方法灵活性差,而且对于用户要求过高。而全自动的组合方式则过于理想化,当服务数量极为庞大时自动组合的效率是很大的瓶颈,同时对于个性化的应用开发来说目前很难顺利完成。而半自动的服务组合方式恰好介于二者之间,本文提出的 Keen 方法是一种基于自动的“组合套餐”和手工的“自助餐”相结合的服务组合方式。一方面通过借助专家的知识 and 经验提供大量可重用的组合构件,来降低对用户的要求,提高用户主动参与应用构造的可能性;另一方面通过用户的适当参与来表达他们的个性化需求。这样通过为用户推荐合适的服务组合模板和恰当的服务,进而为用户构造应用和选取服务提供辅助手段,降低了组合服务时对用户的要求。因此本文的 Keen 方法是一种半自动的服务组合方式。

**结束语** Web 服务的出现为软件 DIY 带来了曙光。然而服务组合是一个复杂的领域,没有现成的解决方案。本文针对业务人员给出了一种大粒度、可重用、便于操作的服务组合方法,并详细地介绍了该方法的核心元素——组合模板以及它的使用模式。目前我们只针对旅游领域提供了少量的模板,服务组合模板只有达到一定的数量,才能支持有效的复用。而大量的组合模板的获得需要有很高的投入和长期的积累,是一个循序渐进、逐步完善的过程。在模板达到一定数量后,对组合模板库的管理和维护,对组合模板的性能评价和使用模式,也都需要做深入的研究。

## 参考文献

[1] Clements P C. From Subroutines to Subsystems: Component-

based Software Development[J]. American Programmer, 1995, 8(11)

- [2] 杨美清,王千祥,梅宏,等.基于复用的软件生产技术[J].中国科学(E辑),2001,31(4)
- [3] 胡海涛,李刚,韩燕波.一种面向业务用户的大粒度服务组合法[J].计算机学报,2005,28(4):694-703
- [4] 房俊,虎嵩林,韩燕波,等.一种支持业务端编程的服务虚拟化机制 VINCA-VM[J].计算机学报,2005,28(4):549-557
- [5] Majithia S, David W, Gray W. A Framework for Automated Service Composition in Service-oriented Architectures[C]//1st European Semantic Web Symposium. Heraklion, Greece, 2004: 269-283
- [6] Taylor I, Shields M, Wang I, et al. Grid Enabling Applications Using Triana[C]// Workshop on Grid Applications and Programming Tools. Seattle, USA, 2003
- [7] Mandell D J, McIlraith S A. A Bottom-Up Approach to Automating

Web Service Discovery, Customization, and Semantic Translation[C]// Proc. of the 12th Int'l. WWW Conference Workshop on E-Services and the Semantic Web. Budapest, 2003: 89-96

- [8] Sirin E, Hendler J, Parsia B. Semi-automatic composition of Web services using semantic descriptions[C]// Web Services, Modeling, Architecture and Infrastructure Workshop in conjunction with ICEIS. 2003
- [9] Cardoso J, Sheth A. Semantic e-Workflow Composition [J]. Journal of Intelligent Information(JIIS), 2003, 12(3): 191-225
- [10] McIlraith S, Son T C. Adapting golog for composition of semantic web services[C]//Proc. of the 8th Int'l. Conf. on Knowledge Representation and Reasoning. Toulouse, France, 2002: 482-493
- [11] Shankar R P, Armando F. SWORD: A Developer Toolkit for Web Service Composition[C]//Proc. of the 11th International World Wide Web Conference. Honolulu, 2002: 786-810

(上接第 149 页)

管理员能够更直观地了解当前网络的流量信息。图 17 为局域网的数据流量视图,显示的是流入网段 125. 223. 118. 0/24 的数据流量大小。从图中可以看出,网络的数据流量大约介于 2MB/s 和 5MB/s 之间。图 18 显示 Netflow 传感器事件分析结果,主要是水平扫描和垂直扫描。此外,该类传感器还筛选并保存了安全事件所对应的原始 NetFlow 记录,便于进一步关联和取证安全事件。

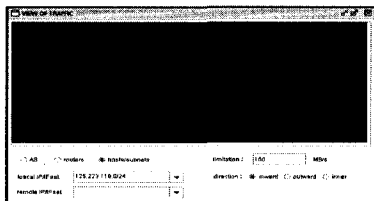


图 17 局域网的流量视图

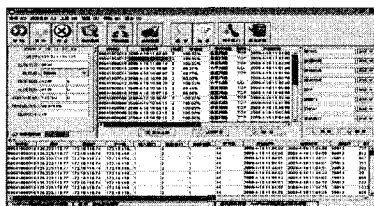


图 18 NetFlow 传感器事件分析结果

服务类传感器针对特定服务,监视和采集服务的安全状态数据,并对其进行分析。在服务发生严重偏离或失效时,以事件的形式报告给上层应用系统或网络管理员。图 19 显示了所监测服务的运行状态数据,主要涉及流量异常、性能异常、报文异常、进程状态异常、配置错误异常等,红色标记为紧急,绿色标记为正常。

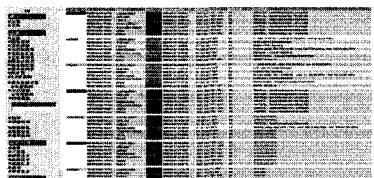


图 19 服务运行状态数据

**结束语** 本文在分析系统功能需求的基础上,提出了一种基于多源异构传感器的网络安全态势感知系统分层结构,并给出了相应的系统环形物理结构、系统层次概念模型以及各层次的功能实体详细设计。此结构采用“分布式获取,分域式处理”的思想,是一个开放、可扩展的环形结构,域内域间相互协作,能有效地降低系统实现复杂性,避免单点失效问题。此外,还从整体上明确了层次与层次、组件与组件关系,从而指导工程实践和关键技术的进一步开展。下一步在态势评估和预测具体实现中对整个系统架构进行必要的微调和完善。

### 参考文献

- [1] Bass T. Intrusion detection systems and multisensor data fusion: creating cyberspace situational awareness[J]. Communications of the ACM, 2000, 43(4): 99-105
- [2] Ganame A K, Bourgeois J, Bidou R, et al. Evaluation of the intrusion detection capabilities and performance of a security operation center[C]// Proceedings of the International Conference on Security and Cryptography. Setúbal, Portugal, 2006: 48-55
- [3] Zhang Z, Li J, et al. A hierarchical network intrusion detection system using statistical preprocessing and neural network classification[C]// Proceeding of the 2nd Annual IEEE Systems, Mans, Cybernetics Information Assurance Workshop. NY, 2001: 85-90
- [4] Ganame A K, Bourgeois J, Bidou R, et al. A global security architecture for intrusion detection on computer networks[J]. Computers & Security, 2008, 27: 30-47
- [5] Engelhardt D, Anderson M. A distributed multi-agent architecture for computer security situational awareness[C]// Proceedings of the 6th International Conference of Information Fusion. Cairns, Queensland, Australia, 2003
- [6] 崔玉华,李涛,周仲义.远程监控 Agent 的体系结构及其环境安全态势评估模型[J].四川大学学报:工程科学版,2007, 39(2): 127-132
- [7] 王慧强,赖积保,胡明明,等.网络安全态势感知关键实现技术研究[J].武汉大学学报:信息科学版,2008, 33(10): 995-998