

特征模型驱动的 Web Services 组装方案及其工具支持*

邢岩⁺, 谷放, 梅宏

(北京大学 信息科学技术学院 软件研究所, 北京 100871)

Feature Model Driven Web Services Composition Approach and Its Support Tool

XING Yan⁺, GU Fang, MEI Hong

(Institute of Software, School of Information Science and Technology, Peking University, Beijing 100871, China)

+ Corresponding author: Phn: +86-21-50819100 ext 806, E-mail: xingyan@pku.org.cn, <http://www.sei.pku.edu.cn/belljointlab/>

Xing Y, Gu F, Mei H. Feature model driven Web services composition approach and its support tool. *Journal of Software*, 2007,18(7):1582–1591. <http://www.jos.org.cn/1000-9825/18/1582.htm>

Abstract: Web Services are gradually becoming main platform-independent software components and will widely exist in the distributed environment, INTERNET. This kind of software developing method that the creation of new software systems will mainly depend on Web Services composition is attracting more and more researchers' attention. There is still not a systematic and mature approach for Web Services composition which can be used to instruct the whole composition process. This paper provides a "feature model driven Web Service composition" approach, which makes Feature Model as the modeling tool to cross through the whole Web Services composition process and can be used to drive and direct all the phases in this process, including requirement analysis, business flow modeling, services composition, business flow deployment and execution. Then, combining with a concrete example, the basic idea, rationale and key steps of FWSC (feature model driven Web Services composition) solution are introduced. The work presented in this paper enhances the speed of modeling, development and quality of Web Services composition, and increases the abilities which can make the system dynamically adjust and quickly evolve when requirement changes.

Key words: Web Services; Web Services composition; feature model; business flow engine

摘要: Web Services 正逐渐成为主流的平台独立的软件构件,并广泛地存在于 INTERNET 分布式环境中,通过组装 Web Services 生成系统的开发方法正在逐步引起人们的关注并受到重视,但目前还没有提出一套相对成熟的、系统化的组装方案用于指导 Web Services 组装系统的整个开发过程.将特征模型作为贯穿整个 Web Services 组装过程的模型工具,利用它驱动 Web Services 组装系统的需求分析、流程建模、组装、部署和执行,提出了“特征模型驱动的 Web Services 组装”解决方案,并结合一个具体的实例对 FWSC(feature model driven Web Services composition)方

* Supported by the Key Project of the National Natural Science Foundation of China under Grant No.60233010 (国家自然科学基金重点项目); the National Basic Research Program of China under Grant No.2002CB312003 (国家重点基础研究发展计划(973)); the National Science Foundation for Distinguished Young Scholars of China under Grant No.60125206 (国家杰出青年科学基金); the Major Project of Science and Technology Research of Ministry of Education of China under Grant No.MAJOR0214 (国家教育部重大科技项目)

Received 2004-06-28; Accepted 2006-08-14

案的基本思想、原理以及关键步骤进行了介绍.该工作可以有效地提高 Web Services 组装系统的建模、开发速度和质量,并增强 Web Services 组装系统在需求发生变化时的动态调整和演化能力.

关键词: Web Services; Web Services 组装;特征模型;流程引擎

中图法分类号: TP393 文献标识码: A

软件技术的发展趋势之一是构建出平台独立的软件构件^[1],现阶段,Web Services 正在不断实现着这一目标.不远的未来,它们将广泛地存在于 Internet 分布式环境中,那时,新应用系统的生成将不再主要依靠人工编写代码,而主要依靠选择一组恰当的 Web Services 通过组装的方式生成.这种基于 Web Services 组装的开发方法对于提高软件的开发效率以及可复用性,促进企业内部的信息集成(enterprise application integration)以及企业之间的商业合作(business to business)具有重要作用,因此,针对 Web Services 组装的研究已经引起很大关注.

目前,针对 Web Services 组装的研究主要有两大流派.其中,IBM,Microsoft 等产业界的研究部门主要侧重于“如何使 Web Services 能够顺利地互联、互通”.为此,他们制订了一组基于 XML、用于刻画 Web Services 的规约描述、交互协议以及服务查询机制的标准规范,分别是 Web Service 描述语言、简单对象访问协议、全局描述、发现和集成协议,并在此基础上提出了 Web Services 流程组装和执行的规约描述,包括面向 Web Service 的业务流程执行语言和 Web Service 编排组装接口.现阶段,大部分的 Web Services 组装方案都是基于已有的 Web Service 标准规范实现组装功能的,这些方案为 Web Services 以及 Web Services 组装的研究奠定了坚实的基础.

另一研究流派的典型代表^[2-7]主要是利用预定义的 ontology 术语,按照 RDF(resource description framework)格式对 Web Services 的功能和语义进行描述,并为 Web Services 的所有操作定义了前置条件和后置条件.在此基础上,利用人工智能领域较为成熟的规划方法,通过给出一个初始状态和一个明确的目标定义以及所有可能的状态转移,并在某种推演算法的指导下自动选择一组 Web Services 完成组装.这类组装方法的前提是假设每个业务流程都有一组明确的、无二义的业务目标定义^[3],并假设 Web Services 的每个操作都有精确的语义约束.

综上所述,在针对 Web Services 组装问题的研究领域中,大部分研究人员还集中于对 Web Services 的组装技术以及组装过程中存在的某些核心技术点进行探讨的阶段,还未提出一套相对成熟的、系统化的组装方案用于指导 Web Services 组装系统的需求分析、流程建模、组装、部署和执行.为此,本文在已有 Web Services 标准规范的基础上,以 Web Services 的实用组装为前提约束,利用特征模型驱动 Web Services 组装系统的开发过程,将自底向上的集成组装方法和自顶向下、逐步求精的特征建模方法结合起来,提出“特征模型驱动的 Web Services 组装(feature model driven Web Services composition,简称 FWSC)”方案.

本文第 1 节将已有组装方案和本文提出的 FWSC 方案进行比较.第 2 节介绍“特征模型驱动的 Web Services 组装”方案的基本思想和流程,并结合一个具体的实例对 FWSC 方案中的关键步骤予以介绍,最后针对本方案的不足以及未来工作方向进行阐述.第 3 节对 FWSC 方案的支撑平台——BiFrost 组装平台进行简单介绍.最后对全文进行总结.

1 相关工作比较

产业界研究部门所提出的组装方案确实一定程度上解决了 Web Services 组装的流程化、标准化以及规范化问题,但是不可避免地也存在着一些问题.如现阶段流程模型的建立主要依靠人工完成,而且一旦建立,则难以在运行时刻对已部署流程的执行逻辑以及流程中某些 Web Services 的调用进行修改和调整.

另外,这类组装方案还假设需要组装的 Web Services 都存在并已经或者很容易地发现,只需按照 BPEL4WS(business process execution language for Web Services)或者 WSCI(Web Service choreography interface)的语法格式定义出流程模型即可.这是一种自底向上的集成组装方法,并未对“如何指导 Web Services 系统的整个组装过程”以及“如何才能获到正确的、合适的 Web Services 完成组装”这些关键课题充分进行考虑.

同样,利用语义网辅助 Web Services 组装的研究也存在着一些不可回避的问题.它们的一个重要假设就是

每个业务流程都具有一组明确的、无二义的业务目标定义^[3],并假设 Web Services 的每个操作都具有精确的语义约束.但实际情况是:一组明确的、无二义的业务目标通常是不存在的,往往是利用非形式化语言在一个更高的抽象层次上予以刻画,且 Web Services 操作对整个业务流程所体现的功能也难以用精确的语义约束来表示.

与上述两大主流研究流派相比,FWSC 方案既充分吸收了已有 Web Services 组装方案经验的基础性技术,又针对他们未解决的问题创造性地提出将需求分析阶段获得的特征模型作为贯穿 Web Services 组装系统开发的模型工具,将自底向上的集成组装方法和自顶向下、逐步求精的特征建模方法结合在一起.与已有的各类组装方法相比,本方案首次针对 Web Services 组装系统的开发过程提出了一整套方法论、指导流程以及工具支持.

具体而言,本方案的特点主要体现在以下几个方面:

- (1) 利用特征模型作为系统组装过程的模型工具,可以在 Web Services 组装系统需求分析人员、流程建模人员、组装和部署人员以及服务调用人员之间建立一个统一的术语空间,为理解系统需求、设计决策、实现方案以及进一步的系统演化提供交流的必要基础;
- (2) 在特征模型中还体现了 Web Services 组装系统中最早的一组需求决策,对系统的整体特性、后续开发和组织进行了初步的约定;
- (3) 整个组装过程围绕特征模型展开,易于被开发人员理解和应用,也易于维护需求和运行实体之间的可追踪性,并能根据需求的变化迅速地在流程模型的设计、组装过程中予以体现,从而降低系统需求变化导致的演化成本;
- (4) 使用特征模型和支持工具维护了需求和实现之间的映射关系,不仅可以提高需求分析人员对系统的理解程度,也可以提高流程建模和组装人员对底层 Web Services 的认识程度,这从某种程度上也提高了底层 Web Services 以及高层规约模型(特征模型和流程模型)的复用价值.

2 特征模型驱动的 Web Services 组装

2.1 特征和特征模型

“特征”一词最早在电信领域中被大量使用,用于表示电信网络向电话客户或系统管理者提供附加服务的功能单元^[8].在电信领域,特征被广泛接受并作为术语描述电信系统的基本功能单元,而且建立了关于特征的一致术语空间,极大地方便了特征的使用和维护.例如,“呼叫转移(call forwarding)”和“呼叫等待(call waiting)”就是电信领域中比较经典的两个特征.电信系统对电话客户或系统管理者提供的服务都是由众多的特征通过相互之间的交互展现给用户的.

而在软件领域中,文献[9]把“特征作为系统需求规约的组织方式”,文中认为,特征是从用户角度对系统的感知,用特征对系统需求规约进行模块化组织是一种非常自然的手段.在 FODA(feature-oriented domain analysis)方法中,文献[10]也把特征定义为系统中用户可见的、显著的或具有特色的方面、品质和特点等.

显然,电信领域中对特征的定义与软件领域中对特征的定义是有区别的,电信领域中的特征强调作为电信系统运行时刻的有机构成单元,而在软件领域更强调将特征作为需求分析阶段对系统需求规约的描述和组织单元.本文对 Web Services 组装系统中特征的定义沿用软件领域中的特征定义,将特征作为 Web Services 组装系统需求分析阶段需求规约描述信息的基本组织单元,而将具体的 Web Services 作为系统运行过程中的功能实现体.

但是,Web Services 组装系统中的特征和组装系统涉及的 Web Services 在概念上具有一定的相似性,如两个概念都描述了 Web Services 组装系统中用户角度可见部分的行为特点,而且它们都代表一个相对独立的功能或非功能需求;甚至对于 Web Services 组装系统而言,功能性的特征和 Web Services 之间基本上可以实现一一映射,可以根据特征的描述直接查找相应的 Web Services 实现体.但是即便如此,两个概念还是有一些本质上的区别:特征强调从需求分析的角度对系统规约进行描述和组织,而 Web Services 更强调从系统实现的角度对系统予以刻画.

特征模型是从最终用户的角度对 Web Services 组装系统的行为能力和特点的描述,它使用特征作为描述

Web Services 组装系统问题域的一阶实体,由一组特征以及这组特征之间的关系组成.特征模型的元模型^[1]如图 1 所示:它将特征以及特征之间的关系作为主要描述对象,以不同抽象层次的特征之间的组成关系作为特征模型的基本组织方式,以部分特征相对整体特征的可选性以及变化性维度和值的机制表现系统需求的变化性,采用依赖、互斥、多选一、多选多等关系反映特征之间存在的约束关系.

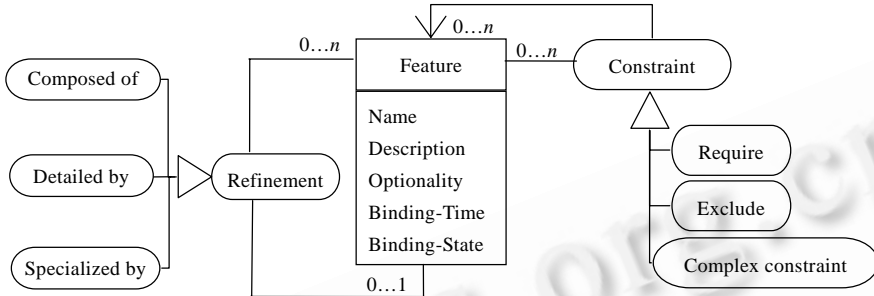


Fig.1 Meta model of feature model

图 1 特征模型元模型

图 2 描述的旅游预定服务的特征模型图是在图 1 所示的元模型基础上建立的,图 2 对特征之间的组成关系、精化关系、依赖/互斥关系以及子特征相对父特征的可选性等主要性质进行了描述.图 2 中所有的叶节点特征与其父特征之间的关系都是精化关系,每个叶节点特征都由一个具体的 Web Service 绑定并实现.

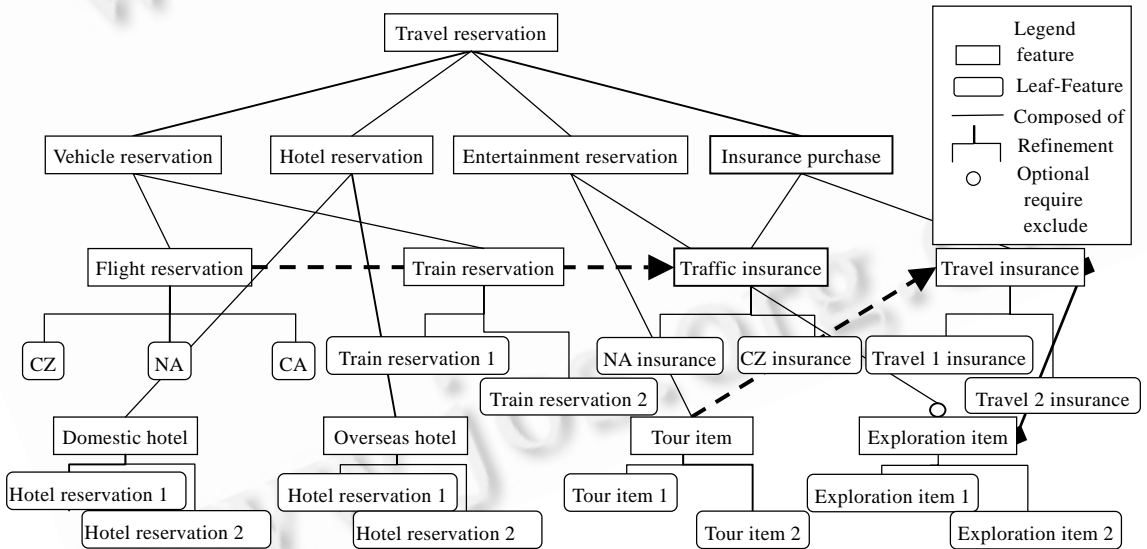


Fig.2 Feature model of travel reservation service

图 2 旅游预定服务的特征模型

2.2 基本思想

特征模型作为软件系统需求规约的组织模型已经得到广泛的认可,是需求分析阶段的重要产物.在开发 Web Services 组装系统的过程中,我们基于特征模型建立了基于特征的 Web Services 组装流程模型,并在此模型的基础上实施服务组装、部署和运行,将自底向上的集成组装方法和自顶向下、逐步求精的特征建模方法结合在一起.这就是“特征模型驱动的 Web Services 组装”(FWSC)方案的核心思想.

特征模型不仅可以应用于 Web Services 组装系统的需求分析和建模阶段,还可以应用到系统的流程组装、部署和执行阶段,从而更加充分地发挥特征模型在 Web Services 组装系统开发过程中的指导作用.例如:

- (1) 根据特征模型中记录的特征之间的拓扑结构以及依赖、互斥关系,完成对 Web Services 组装系统的流程建模;
- (2) 根据特征模型和支持工具中维护的特征与 Web Services 的映射关系,协助流程组装人员完成对流程模型的绑定工作;
- (3) 服务部署人员在运行时刻对特征模型予以维护,可以自动地将高层发生的需求变化以及环境的变化反映到运行时刻的特征模型中,并将对某些 Web Services 的选择、调用和演化功能交由 Web Services 调用平台完成,从而使之与流程模型的建模、组装、执行相分离。

图 3 展现了 FWSC 方案的全过程。

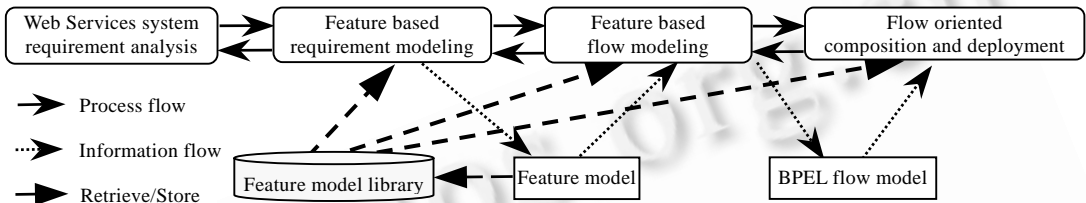


Fig.3 Feature model driven Web Services composition approach

图 3 特征模型驱动的 Web Services 组装方案

FWSC 方案的整个组装过程主要由 4 个阶段组成:

- (1) 针对 Web Services 组装系统的需求进行分析,并利用特征模型对需求分析的结果进行建模.这一阶段主要采用自顶向下、逐步求精的分解策略对系统需求进行划分.在此步骤中,由特征建模人员初步完成对特征模型的建模,在进一步的完善和精化过程中,由特征模型库^{*}协助完成整个建模工作;
- (2) 在特征模型库的协助下完成特征模型建模后,由流程建模人员根据特征模型中记录的特征之间的拓扑结构和依赖、互斥关系,并利用特征模型库中存储的特征与 Web Services 之间的映射关系,实现对 Web Services 组装系统的流程建模,将特征模型中的“服务代理特征”^{**}按照一定的逻辑结构组织在一起,以 BPEL4WS 语言的语法形式展现,此时的流程模型还只是一个抽象的、不可执行的静态流程模型;
- (3) 在流程模型建立后,由流程组装人员可以根据 Web Services 组装系统的具体需求以及用户的使用偏好将流程模型中的“服务代理特征”与某一个或者任意一个对应的子特征进行绑定,此时的流程建模过程就从一个抽象的、不可执行的流程模型转变为一个具体的、可执行的流程模型;
- (4) 最后,由流程部署人员将绑定完毕的流程模型部署到 Web Services 流程引擎上,并在 Web Services 调用平台上动态地维护 Web Services 组装系统的特征模型以及特征与 Web Services 的映射关系,藉此完成流程执行过程中对具体 Web Services 的选择、调用和演化。

上面介绍了 FWSC 方案的基本思想和流程,该方案利用特征模型驱动整个 Web Services 组装系统的开发,使之作为贯穿需求分析、流程建模、组装、部署和执行阶段的模型工具.FWSC 方案结合 Web Services 组装系统的具体特点,利用特征与 Web Services 之间的映射关系辅助完成 Web Services 组装系统的流程建模和组装,并在运行时刻将特征模型作为需求组织模型辅助实现对 Web Services 的选择、调用和演化。

2.3 FWSC方案的基本步骤

2.3.1 面向 Web Services 组装系统的特征建模

特征模型是一个需求分析模型,它采用特征对系统需求进行描述,通过特征模型库维护了叶节点特征和 Web Services 之间的映射关系.因此,特征模型在系统需求和底层实现之间建立起良好的映射关系,这不仅满足

* “特征模型库”将在第 3.3 节中予以详细介绍。

** 后面将对“服务代理特征”给出明确的定义。

了需求分析人员、流程建模人员以及组装/部署人员交流的需要,也提高了模型之间的可追踪性以及建模结果的可复用性。

需求建模人员在建模特征模型时,首先对 Web Services 组装系统的业务需求进行分析,将系统需求按照分治的策略划分成若干需求单元——特征,并综合考虑特征模型库中检索到的可以与系统需求相关或类似的特征模型或特征模型片断,从而得到满足 Web Services 组装系统业务需求的特征模型。

特征模型库是 FWSC 方案中的重要基础设施,它是在对已有或者目前可获得的 Web Services 集合进行分类整理的基础上建立的模型复用工具,可以基于目前业界应用广泛的 UDDI Web Services 库扩展而来。特征模型库将目前可获得的 Web Services 集合中具有相似或相同功能的 Web Services 子集采用特征模型中定义的“精化”关系进行组织,在为每个 Web Service 建立抽象特征描述的同时,将这组特征按照“精化”关系组织在一起,从而建立特征模型片断,并在此基础上将已有的针对其他 Web Services 组装系统建立的特征模型按照功能进行分类并予以存储,作为未来新的 Web Services 组装系统需求建模的基础数据。

当新的 Web Services 组装系统需要进行需求建模时,将分解得到的需求单元与特征模型库中存储的特征模型或特征模型片断进行匹配,将能够满足需求的特征模型或特征模型片断集成到新的特征模型中,这样就通过复用特征模型库中的特征模型或特征模型片断,最大限度地复用了以前的分析结果。

特征模型库的另一个重要作用就是和特征模型一起维护特征与 Web Services 的映射关系,特征模型的叶节点特征都可以在特征模型库中找到与之匹配的 Web Services,它们之间的关联关系在特征模型片断中予以维护。

在考虑特征模型库的检索结果时,如果在特征模型库中查询到某个特征模型或者特征模型片断已经满足当前某一部分的业务需求,则直接将其集成到新建的特征模型中。特征模型中的叶节点特征可以通过在特征模型库中查询满足其需要的特征模型片断,从而维护特征和 Web Services 之间的映射关系。图 2 中给出旅游预定服务的特征模型,下面将结合它对建模过程中如何建模特征间关系进行介绍。

在特征模型中,不同粒度或抽象层次的 Web Services 系统需求将按照特征模型元模型中定义的“组成关系”组织在一起。图 2 中的“旅游预定”特征是由相应的 4 个更小粒度、更具体的特征——“交通工具预定”、“酒店预定”、“娱乐项目预定”和“保险购买”特征组成的,它表明“旅游预定”特征的功能可以由它所包含的 4 个子特征按照一定的逻辑结构组织在一起来实现。

一个大粒度或者更高抽象层次的 Web Services 组装系统需求可以由一组“必选”的特征和一组“可选”的特征组成。必选的特征表示子特征相对于父特征是必须的,具体而言,一旦父特征被选定,子特征也必须被选择;可选的特征表示子特征相对于父特征是可选的,具体而言,即使父特征被选定,子特征也不一定被选择。图 2 中“探险项目预定”特征相对于“娱乐预定”特征而言就是可选的特征。

在特征模型中,我们利用精化关系对叶节点特征进行组织,将众多具有相似或者相同功能的叶节点特征划分成一些子集,从而利于对与叶节点特征对应的 Web Services 的查找、选择和调用。图 2 中的“南方航空预定”、“北方航空预定”和“国际航空预定”这 3 个叶节点特征既是对“航空预定”特征的精化,也作为底层 Web Services 在需求分析阶段的抽象描述而存在。这 3 个叶节点特征之间是“多选一”的关系,具有很强的可替代性,可以由流程组装人员在组装阶段选择其中的任意一个满足父特征的需求,父特征则作为这组具有可替代关系的子特征的共性抽象,我们将具有类似结构的特征定义为“服务代理特征”。

特征模型中采用元模型中定义的“依赖”、“互斥”等关系对特征之间的逻辑关系进行了限制和约束,从而便于流程建模和组装人员对“服务代理特征”及其子特征进行选择。图 2 定义的“航班预定”特征依赖于“交通险购买”特征,这意味着,通过旅游代理服务预定航班,必须同时选择相应的“航空保险”服务。反映到流程组装过程中,一旦在流程建模过程中选择了“航班预定”,就必须选择购买某个相应的“航空保险”。“探险项目”特征和“保险购买”特征之间是互斥的关系,意味着预定“探险项目”就不能预定“旅游保险”。

2.3.2 基于特征的 Web Services 流程建模

流程建模人员在定义流程模型之前,需要对 Web Services 组装系统的需求以及建模得到的特征模型进行深

入的理解和分析.在流程建模过程中,出于可扩展性以及 BPEL4WS 语言规范要求的考虑,流程建模人员建立的只是一个抽象的、不可执行的流程模型,他们将特征模型中定义的“服务代理特征”作为 Web Services 组装系统流程建模的主要对象,根据定义在特征模型中特征之间的拓扑结构以及依赖、互斥关系,建立一个能够满足 Web Services 组装系统需求的、符合 BPEL4WS^[12]语法规则的流程模型.

图 4 给出了一个利用图 2 特征模型中定义的“服务代理特征”建立的流程模型,实现了一个针对国内用户的、远程观光旅游预定服务.

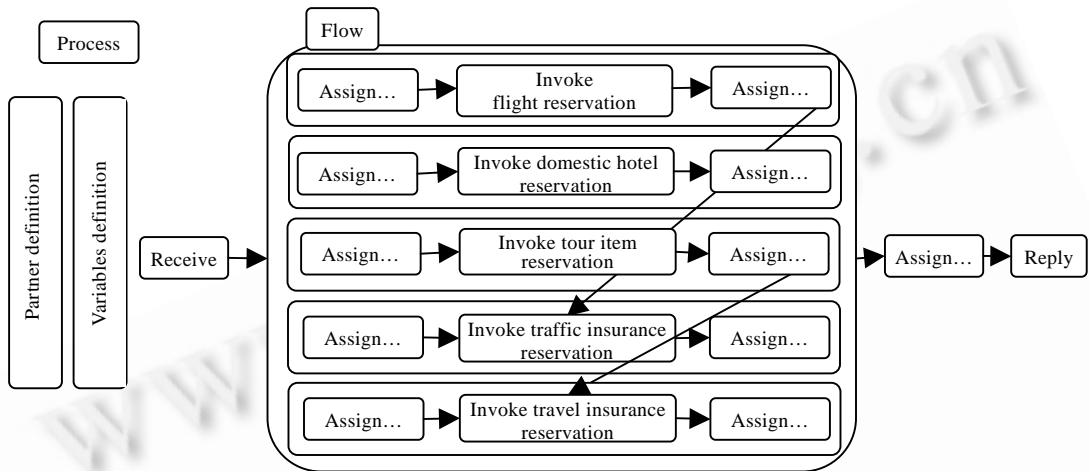


Fig.4 Flow model diagram

图 4 流程模型图

在流程建立之初,首先是确定流程中可能涉及的合作伙伴,也就是定义流程模型中的服务参与者.流程中合作伙伴的选择过程是对特征模型中“服务代理特征”的选择和匹配过程,根据特征模型中定义的子特征相对于父特征的可选性和必选性,并结合特征之间的依赖、互斥关系,对可能涉及到的特征模型中的“服务代理特征”进行选择 and 确定.表 1 定义了图 4 给出的流程模型中所涉及到的 6 个合作伙伴.

Table 1 partnerLinks definition in flow model

表 1 流程模型中的 partnerLinks 定义

```

<partnerLinks>
  <partnerLink name="caller" partnerLinkType="The service type which is defined in the WSDL file"/>
  <partnerLink name="airLineProvider" partnerLinkType="Flight Reservation"/>
  <partnerLink name="hotelProvider" partnerLinkType="Domestic Hotel Reservation"/>
  <partnerLink name="sightSeeProvider" partnerLinkType="Tour Reservation"/>
  <partnerLink name="aviationInsuranceProvider" partnerLinkType="Traffic Insurance Reservation"/>
  <partnerLink name="sightSeeInsuranceProvider" partnerLinkType="Travel Insurance Reservation"/>
</partnerLinks>

```

流程模型的合作伙伴确定后,还需要对整个流程中可能涉及的变量信息予以定义.流程中涉及的变量主要根据各合作伙伴在各自 WSDL(Web Service description language)中声明的数据结构进行定义的,变量定义的目的是使业务信息能够在调用方的请求和各个被调用方之间清晰、正确地传递,并最终将流程的执行结果返回给服务调用方.

流程模型中的变量定义完成后,就开始定义流程的业务逻辑.流程建模人员使用流程模型语言元素,将流程中各合作伙伴的具体操作通过预先声明的变量信息,按照一定的逻辑结构组织在一起,最终实现流程建模人员的设计目标.

图 4 给出了一个流程模型的例子,下面结合此例对流程模型建模进行大致介绍.

在流程的设计过程中,一般而言,首先是建立一个 receive 活动,由它负责监听服务调用者的调用请求.建立 receive 活动后,按图 4 所示建立一个由 5 个 Sequence 活动组成的 Flow 活动,Flow 活动中的每个 Sequence 活动将进一步调用相应“服务代理特征”的操作,分别是针对“航班预定”、“国内酒店预订”、“观光项目预定”、“旅游保险预定”和“交通保险预定”的调用.在 Sequence 活动调用“服务代理特征”的前后,需要根据服务调用者传递过来的请求信息以及 Web Services 的执行结果,针对具体操作的输入、输出参数进行设置,这是由一组 Assign 活动利用变量定义中定义的变量完成的.在图 2 定义的特征模型中,定义了“航班预定”和“观光项目预定”分别依赖于“交通保险预定”和“旅游保险预定”,所以在流程模型的设计上需要使用 Link 结构将这两对具有依赖关系的特征进行协调,保证流程在执行完“航班预定”和“观光项目预定”后能够触发各自相应的“保险购买”特征.完成上述的建模后,通过一组 Assign 活动对各“服务代理特征”的执行结果进行组织,通过 reply 活动将执行结果采用 SOAP 应答消息的方式返回给流程调用者.至此,流程模型的建模过程结束,特征模型在此过程中为流程建模提供了基本的建模元素,并利用自身蕴涵的特征之间的拓扑结构以及依赖、互斥约束对流程建模提供了基本的指导意见.

2.3.3 基于特征模型的 Web Services 组装和部署

流程模型定义完成后,流程组装人员首先根据具体的 Web Services 组装系统的业务需求以及用户的使用偏好,对建立的抽象、不可执行的流程模型中涉及到的“服务代理特征”进行绑定.流程组装人员根据特征模型中定义的“服务代理特征”与其子特征之间的精化关系,从一组具有相似或者相同功能的子特征中选择某一个作为其对应的“服务代理特征”的绑定实现.如果业务需求以及用户对某个“服务代理特征”的绑定要求没有任何偏好,流程组装人员可以通过在绑定过程中设置“任意”,从而将该“服务代理特征”的绑定工作推迟到运行时刻的 Web Services 调用阶段,由 Web Services 调用平台根据运行时刻的具体情况在特征模型中动态地选择某一个相应的叶节点特征对应的 Web Service 满足系统需求.

流程组装人员在完成对流程模型中“服务代理特征”的绑定后,将这个绑定后的流程模型部署以及前期建立的特征模型到组装平台上.组装平台会自动解析特征模型,并启动流程引擎解析绑定后的流程模型,与此同时,会对部署在平台上的各项信息进行一致性验证,包括特征模型和流程模型之间的映射关系、根据叶节点特征查找到的 Web Services 的 WSDL 描述和流程模型中具体活动的映射关系等.完成上述验证之后,根据流程模型的定义生成运行时刻的动态流程模型,并启动运行时刻流程引擎,创建相应的监听进程并等待来自外界用户的调用.

在整个部署、运行过程中,特征模型被作为 Web Services 组装系统的组织模型维护在 Web Services 组装平台上,由它维护系统需求并最终实现体之间的映射关系.根据流程模型的定义,某个具体的 Web Services 调用参数在传递的过程中会首先传递给组装平台维护的特征模型,由流程模型中决定的“服务代理特征”在其对应的子特征集合中进行选择和绑定,进而将调用参数传递给选定的子特征所对应的 Web Service 实例,从而实现了对 Web Services 的查找、选择和调用与流程模型的建模、组装、运行相分离.

此外,Web Services 调用平台也支持在运行时刻对已部署的特征模型进行修改和调整,当特征模型由于系统演化的需要发生变化时,比如某“服务代理特征”对应的子特征集合发生了变化,或者某个“服务代理特征”被其他新特征代替等,Web Services 调用平台中维护的特征模型可以动态地进行调整而不需要重新被部署,这就使得需求层或者运行环境发生的变化能够迅速地反映到系统的运行时刻模型中,从而增强了 Web Services 组装系统的动态演化和调整能力.

2.4 FWSC方案存在的不足

在本方案中,我们在 Web Services 组装的过程中引入了需求建模阶段的特征模型,通过建立基于特征的 Web Services 组装流程模型对服务进行组装和部署.整个组装过程围绕特征模型展开,易于被分析人员、开发人员以及流程建模人员所理解和应用,也易于维护需求和运行实体之间的可追踪性,并降低由需求变化所导致的演化成本.

但是,本方案也存在着一些不足,主要包括:

- (1) “特征模型”以及“特征模型库”的建立需要较丰富的领域知识以及行业知识,特征模型建立的质量将直接影响到后续的流程建模以及组装工作,未来将针对如何建立特征模型给出更详细的操作指南;
- (2) 目前的“特征模型”中并没有包括流程的概念,需要流程建模人员根据具体的业务需求将不同的“服务代理特征”进行流程重组.实际上,流程信息也是一种非常重要的可复用信息,如果将流程或者流程的模式信息在特征模型的基础上通过某种方式表示出来,将对提高流程建模的效率具有非常重大的指导作用.未来针对这一方面的研究工作将进一步展开.

3 工具支持

前文介绍了 FWSC 方案的基本思想和执行步骤,本节对该方案的支撑平台——BiFrost 组装平台进行简单介绍,该平台整体结构如图 5 所示.

BiFrost 组装平台是北京大学信息技术学院软件研究所针对 Web Services 组装问题研究的基础上,按照 FWSC 解决方案的要求设计并开发的 Web Services 组装平台.它主要由两个建模工具和两个支撑工具组成,分别是:特征建模工具、流程建模工具以及基于 BPEL4WS 规范的 BiFrost 流程引擎和基于特征模型的 Web Services 调用平台.BiFrost 组装平台基于 J2EE 框架,是一个能够部署在北京大学信息技术学院软件研究所自主研发的应用服务器 PKUAS^[13]上的应用系统.

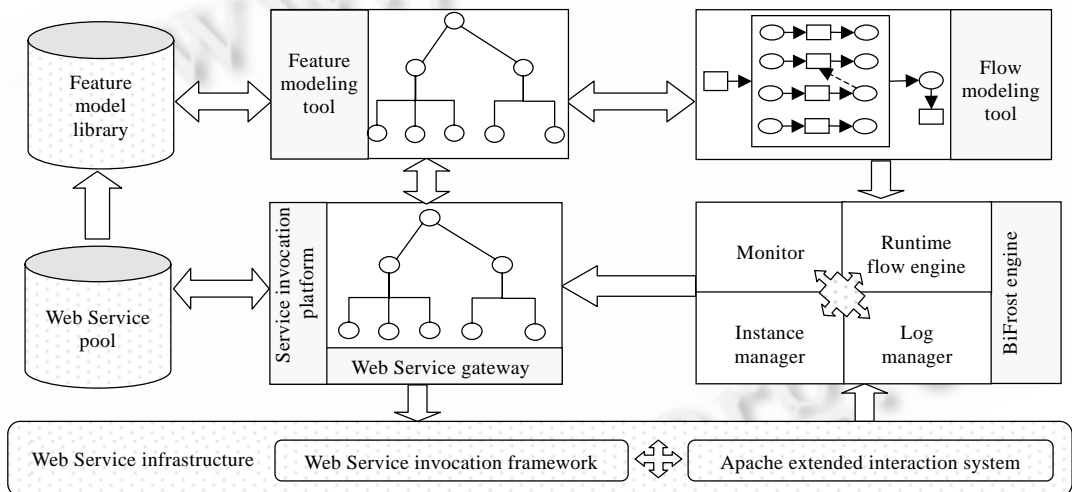


Fig.5 Support platform of FWSC approach

图 5 FWSC 方案的支撑平台

4 结束语

本文在已有研究成果的基础上,提出将特征模型作为贯穿整个 Web Services 组装过程的模型工具,利用它驱动 Web Services 组装系统的需求分析、流程建模、组装、部署和执行,提出“特征模型驱动的 Web Services 组装”解决方案.利用特征模型和特征模型库维护了需求模型和运行时刻模型之间的映射关系,其中特征作为 Web Services 组装系统需求在需求建模过程的基本构成单元,而 Web Services 作为特征在运行时刻的具体实现体.利用特征模型驱动整个 Web Services 组装系统的开发过程,这不仅满足了需求分析、流程建模、组装/部署人员以及最终调用者在系统开发和使用过程中交流的需要,也提高了系统内部的可追踪性以及建模结果的可复用性,并且大幅度提升了 Web Services 组装系统的建模、开发速度和质量,并增强了 Web Services 组装系统在需求发生变化时的动态调整和演化能力.本文结合一个具体的实例对 FWSC 方案的基本思想、原理以及关键步骤进行了介绍,并对该方案的支撑平台——BiFrost 组装平台进行了简要的介绍.

References:

- [1] Srivastava B, Koehler J. Web Service composition—Current solutions and open problems. In: Proc. of the 13th Int'l Conf. on Automated Planning & Scheduling (ICAPS 2003) Workshop on Planning for Web Services. 2003. 28–35. <http://icaps03.itc.it/>
- [2] Berners-Lee T, Hendler J, Lassila O. The semantic Web. Scientific American. 2001. http://www.ryerson.ca/~dgrimsha/courses/cps720_02/resources/Scientific%20American%20The%20Semantic%20Web.htm
- [3] McIlraith S, Son TC. Adapting golog for composition of semantic Web Services. In: Proc. of the 8th Int'l Conf. on Knowledge Representation and Reasoning (KR 2002). 2002. 482–493. <http://kr.org/KR2002/>
- [4] The OWL Services Coalition. OWL-S: Semantic markup for Web Services. 2003. <http://www.daml.org/services/owl-s/1.0/owl-s.html>
- [5] Limthanmaphon B, Zhang YC. Web Service composition with case-based reasoning. In: Schewe KD, Zhou X, eds. Database Technologies 2003 of the 14th Australian Database Conf. (ADC 2003) Conf. in Research and Practice in Information Technology, Vol. 17. Adelaide: Australian Computer Society, 2003. 201–208.
- [6] Rachid Hamadi, Boualem Benatallah. A Petri net-based model for Web Service composition. In: Schewe K-D, Zhou X, eds. Database Technologies 2003 of the 14th Australian Database Conference (ADC 2003) Conf. in Research and Practice in Information Technology. Vol. 17. Adelaide: Australian Computer Society, 2003. 191–200.
- [7] Carman M, Serafini L. Planning for Web Services the hard way. In: Proc. of the 3rd Int'l Symp. on Applications and the Internet (SAINT 2003) Workshop on Service Oriented Computing. Orlando: IEEE Computer Society Press, 2003. 27–31. http://www.saintconference.org/2003/saint_main.htm
- [8] Bowen TF, Dworack FS, Chow CH, Griffeth N, Herman GE, Lin YJ. The feature interaction problem in telecommunications systems. In: Proc. of the 7th Int'l Conf. Software Engineering for Telecommunications Switching Systems. London: Institution of Electrical Engineers, 1989. 59–62. <http://ieeexplore.ieee.org>
- [9] Davis AM. The design of a family of application-oriented requirements languages. IEEE Computer, 1982,15(5):21–28.
- [10] Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS. Feature-Oriented domain analysis (FODA) feasibility study. Technical Report, CMU/SEI-90-TR-21, Pittsburgh: Carnegie Mellon University, 1990. 1–52.
- [11] Zhang W, Mei H. A feature-oriented domain model and its modeling process. Journal of Software, 2003,14(8):1345–1356 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/14/1345.htm>
- [12] BPEL2003. 2003. <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
- [13] Huang G, Wang QX, Cao DG, Mei H. PKUAS: A domain-oriented component operating platform. Acta Electronica Sinica, 2002, 30(12A):39–43 (in Chinese with English abstract).

附中文参考文献:

- [11] 张伟,梅宏.一种面向特征的领域模型及其建模过程.软件学报,2003,14(8):1345–1356. <http://www.jos.org.cn/1000-9825/14/1345.htm>
- [13] 黄罡,王千祥,曹东刚,梅宏.PKUAS:一种面向领域的构件运行支撑平台.电子学报,2002,30(12A):39–43.



邢岩(1979 -),男,河北衡水人,硕士生,主要研究领域为软件工程,特征交互.



梅宏(1963 -),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件工
程与软件工程环境,软件复用与软件构件
技术,(分布)对象技术,软件工业化生产技术
及系统.



谷放(1973 -),男,博士生,主要研究领域为
软件工程,领域工程.