

Web 服务组合系统研究综述

武云鹏 包卫东 张维明 黄金才

(国防科学技术大学 C4ISR 技术国防重点实验室 长沙 410073)

摘要 随着 Web 服务组合技术的发展,基于组件 Web 服务构建的复杂服务的相关系统也越来越多。随之而来的问题是如何选择适合 Web 服务组合系统构建需要的组合服务。近来,基于不同技术的服务组合工具系统越来越多。简要介绍了 Web 服务组合的相关技术,给出了通用 Web 服务组合系统的功能分析模型,并根据该模型对 12 个 Web 服务组合系统所具有的功能进行分析、对比,最后以表格形式给出结果。

关键词 组合服务,组件服务,服务组合系统,执行引擎

中图法分类号 TP319 **文献标识码** A

Web Service Composition Systems Survey

WU Yun-peng BAO Wei-dong ZHANG Wei-ming HUANG Jin-cai

(Information System and Management, National University of Defense Technology, Changsha 410073, China)

Abstract Web service composition has become an emerging and promising technology for designing and building complex applications out of single Web-based software components. Currently, there already are several such systems based on different technology. Question along with these systems is which system should be chosen as a composition tool. This paper gave a function model of general Web service composition system, based on which analyzed twelve systems which already exist. According to the functions systems have, it divided these systems into different categories and gave an outlook to essential future research work.

Keywords Synthesized service, Component service, Service composition system, Execute engine

1 前言

Web 服务由 URL 唯一标识,以基于 XML 的标准语言定义描述的网络架构。这种架构能够保证在不同平台之间的互操作性,其主要由 3 个部分组成^[3]。

1) 服务提供方:提供 Web 服务实体,定义服务的描述并将其发布到服务注册方。

2) 服务需求方:从 Web 服务注册方获取服务信息并调用服务实体。

3) 服务注册方:提供 Web 服务注册、查询机制。

服务需求方可以是终端用户,也可以是其它 Web 服务。若某 Web 服务的执行过程中调用了其它的 Web 服务,就称该服务为组合服务^[1],被调用的服务为组件服务。执行调用的过程为 Web 服务组合。

Web 服务组合依靠对现有 Web 服务的编排、调用,来满足用户的功能需求。依据绑定组件服务实体的机制,可以将 Web 服务组合过程分为静态与动态两种^[2]。静态服务组合是在执行组合过程之前即已经确定了绑定的组件服务实体,而动态服务组合是在执行组合过程当中确定组件服务实体。还可以根据组合过程基于的技术将服务组合分为基于工作流的 Web 服务组合与基于 AI-Planning 的 Web 服务组合^[3]。另外,文献^[4]根据服务组合执行机制将 Web 服务组合分为

集中式、分布式和混合式的服务组合。

获取用户需求是任何软件系统的首要任务,晦涩难懂的协议与建模语言很难让用户将需求表述清楚,采取较为人性化的人机界面会使需求获取过程更加有效。如何获取组件服务接口,是服务组合系统必须考虑的事情。通常系统会对组件服务接口有严格的要求,例如数据类型、接口逻辑格式等等,这些要求成为组件之间交互的主要障碍。采用合适的服务获取机制是构建 Web 组合服务的必要前提,目前较为常用的解决方法是基于本体的语义匹配查询^[5]。

组合过程中,需要建立两方面模型。一方面要对控制流建模,定义组件服务的调用顺序以及调用的前提条件,确保组合过程与功能需求的一致性。可以采用过程化的建模语言建模,例如 UML、Petri 网、状态图、规则集合、 π 演算等。另一方面要对数据流建模,定义数据在组件服务之间的传递方式,确保数据接口的正确性。

由于 Web 服务所处环境的高度不稳定性,组合过程中很有可能出现组合失败的情况,例如调用超时、组件服务不存在等等,因此有必要搭建监控环境,用以及时报告系统运行状态,同时建立容错机制,自动进行错误修复或系统回退^[6,7,20]。

2 Web 组合系统功能分析模型

本文提出一种用于分析 Web 服务组合系统的功能模型

(见图 1),模型根据组合过程分为 7 个子功能模块:人机界面、预处理模块、组合建模模块、执行模块、验证模块、控制模块、错误处理模块。

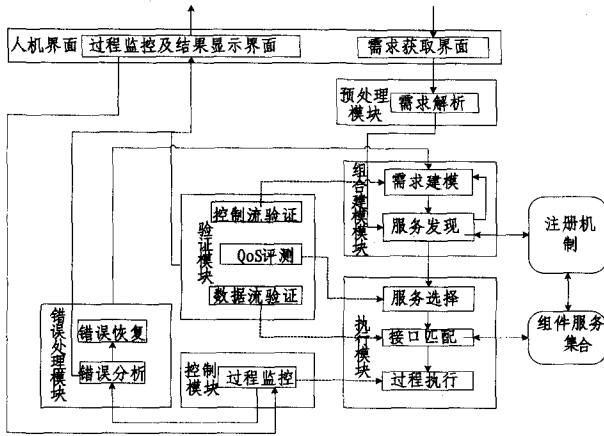


图 1 系统分析模型

人机界面:用于获取用户的需求并将组合过程的监控和结果信息反馈给用户。目前有部分系统采用图形化的界面获取用户需求,使用户不再需要了解各种协议标准,在很大程度上提高了系统的可用性。及时的信息反馈使用户可以根据自定义标准调整组合过程。组合的最终结果要尽量以直观的形式呈现给用户,形成良好的反馈。

预处理模块:不同系统获取需求的形式也各不相同。通常,以图形界面或自然语言获取的需求是不能直接被机器所识别的,需要将其转化为能够识别的标准格式,如 BPEL 等。预处理模块解析用户需求,将用户输入的需求信息转化为机器所能识别的标准格式。

组合建模模块:Web 服务组合通过编排组件服务的调用顺序及组建之间的接口关系得到满足用户需求的新功能。为了验证组合服务与需求功能的一致性,通常需要对组合过程构建抽象模型。基于工作流和基于 AI-Planning 是构建模型的两种主要方法。基于工作流的方法将服务组合过程中的组件调用看作功能操作,该方法需要建模人员了解组件服务与功能操作之间的对应关系。AI-Planning 是人工智能领域中的一个重要方向,根据组合与计划的过程相似性,基于 AI-Planning 的方法被服务组合系统广泛采用。服务发现是获取与需求功能相匹配的组件服务的重要手段,建立抽象模型之后,通过服务发现功能验证是否存在一组组件服务满足当前抽象模型的功能需求。不被满足时,需要重新构建抽象模型。通常这是一个迭代的过程,直至找到一组能够满足抽象模型全部功能需求的组件服务为止。

执行模块:服务选择根据用户需求标准选取较优的组件服务作为抽象模型绑定的服务实体。对于不同的选择标准及不同的选择算法,选取的组件服务也会不同。绑定服务实体后,通过接口匹配功能合理设置组件之间的数据流,使数据能够以统一的标准在组件服务之间传递。在完成接口匹配工作后,执行该服务组合过程,以满足用户的需求。执行组合过程需要相应执行引擎的支持。

验证模块:验证的最终目的是为了组合结果满足用户的需求。因此,需要验证逻辑模型的正确性和绑定的组件服务与功能需求的一致性以及数据接口的匹配性。验证方法有多种,例如基于逻辑的验证方法将需求及组件服务的功能、性质等用标准的逻辑符号表示,通过分析逻辑表达式的推理

进行验证。

控制模块:Web 服务组合所处环境的高度不稳定性决定了需要实时监控机制来保障执行过程的正确、有效。将服务组合的过程信息以及及时有效的方式反馈给用户,增加了系统的灵活性。用户通过控制机制根据反馈信息干预组合过程。

错误处理模块:虽然在组合过程当中有控制模块保证系统正确地运行,但由于环境的高度不稳定性,最终还是有可能发生系统组合失败^[6],例如结果与需求不匹配、调用组件服务超时等。一旦发生这种情况,需要系统自动对失败原因进行分析、匹配,找到相应的错误处理方法^[7],并采取正确的恢复措施。

3 Web 服务组合系统分析

以功能模型为系统分析基础,可以得到较直观的对比,使用户能够根据功能选择适合需求的工具系统,降低选择的盲目性。本文根据上述系统功能分析模型对 12 个 Web 服务组合系统进行了分析和对比(见表 1):

ASTRO^[8]是 ITC-IRST 与 Trento 大学合作开发的项目。它是以服务为导向的、基于本体描述的、分布式的 Web 服务组合系统。系统输入是以 BPEL 定义接口的组件服务集合和组合服务的功能需求描述,输出是以 BPEL 定义的可执行文件。ASTRO 分为若干子系统:WsRequirement 用以获取用户需求以及组件服务信息,以图形化符号为基础,提供类似于工作流图的描述机制获取用户需求,具有清晰直观的优点,但用户需首先了解符号意义。WsTranslator 将 WsRequirement 获取的需求及组件信息以形式化语言描述,便于验证。WSYNTH 子系统根据需求信息自动构建组合模型,得到由 BPEL 定义的可执行文件。给定监控属性,WNON 自动生成并执行监控代码,获取属性在运行过程中的相关信息。NuSMV 是一个形式化的验证系统,用于验证有限状态系统。WSConsole 是前端控制台,将监控信息显示给终端用户。WSAnimator 是一个 eclipse 平台插件,用于执行整个组合过程,提供用户调用该 Web 组合服务的接口。

Haley^[9]是 Georgia 大学 LSDIS 试验室开发的基于语义匹配的 Web 服务组合工具集合。系统输入为用户的需求描述及以 WSDL 描述接口、以 WSLA 描述非功能属性的组件集合,最终生成 BPEL 可执行文件。以半马尔可夫决策理论为基础优化服务组合过程。工具集合包含 7 个主要部分:SA-WSDL Parser 解析 WSDL 接口文件;WSLA Parser 用以解析非功能属性描述文件,获取组件服务的 QoS 参数作为绑定参考;Process Hierarchy Modeler 为用户需求获取界面,用户通过该子系统将需求以层级结构表示;Planning Domain Generator 根据用户需求及组件信息生成以一阶逻辑定义操作的描述文件;eDR-GOLOG Planner 根据上述操作描述文件自动生成组合计划;BPEL Generator 根据组合计划生成可执行的 BPEL 文件;KB based Process Monitor 负责监控组合执行过程。

SHOP2^[10]并不是专门用于 Web 服务组合的系统,它是由 Maryland 大学开发的一款基于 AI-planning 方法的任务分解系统。由于 Web 服务组合与任务计划过程的相似性,该系统在一定程度上可用于 Web 服务组合工作。它主要分为 4 个部分:人机界面,用以获取用户的需求;接口解析,SHOP2 要求组件服务接口以 OWL-S 标准描述;计划生成,负责生成

Web 服务组合计划;信息收集,负责收集绑定组件服务的相关信息,SHOP2 可以根据相应的信息确定任务计划的可行性。最后系统将计划以 DAML-S 描述。

SeGSeC^[11]具有图形化显示界面的 Web 服务组合系统。它与 CoSMoS(组件服务语义模型)及 CoRE(组件运行环境)共同完成 Web 组合任务。CoSMoS 用以描述组件服务的语义。CoRE 提供服务发现及调用功能。SeGSeC 允许用户以自然语言的形式输入需求。主要包含 4 个部分:RequestAnalyzer 将用户输入的需求转换为 CoSMoS 语义图;ServiceComposer 根据接收到的语义图生成可执行组件服务的调用顺序;Reasoner 负责检查需求和组件服务功能的匹配;最后 ServicePerformer 负责调用组件服务。

Argos^[12]由 SPPD 和 ISI 联合研制,利用 Web 服务技术重用数据资源及数据处理操作,提供灵活的数据查询和分析功能,以提高数据分析的效率。它是以本体描述和工作流分析为基础的自动 Web 服务组合系统。Argos 的服务组合过程分为 3 个步骤:领域资源和操作的本体描述,Argos 要求组件以 RDF/RDFS 标准描述;根据组件的描述信息及用户需求,自动生成工作流程;执行工作流,调用组件 Web 服务完成数据分析,并返回分析结果。新的工作流程可以作为独立的、以 WSDL 定义接口的 Web 服务独立存在。采用 IBM 的 BPEL 执行引擎,Apache Tomcat 和 Axis 框架部署组件 Web 服务。

SELF-SERV^[13]以状态转换图的形式描述组合流程。系统分为 3 个主要部分:服务发现引擎、服务编辑器、服务部署器。服务发现引擎通过 UDDI、WSDL、SOAP 等协议及相应机制发现与功能需求匹配的组件服务;用户可以通过服务编辑器定义新的或编辑已有的 Web 服务,并最终转换成 XML 文档,以便以后分析、调用;服务部署器根据组合服务的状态转换图和状态发生条件得到相应的组件 Web 服务调用顺序,并部署组件 Web 服务。系统在完成组合服务的定义及部署后能够将该组合服务发布到相应的注册机制,作为组件服务供其它 Web 服务调用。

Fusion^[14]根据用户需求生成组合计划,并能够自动对其验证和优化。系统分为 6 个子系统:需求获取子系统(USS)是一个图形化的人机交互系统,用于获取用户的需求并将其

转换成系统可识别的信息结构;动态组合计划生成子系统(WGS)根据用户需求生成正确并经过优化的组合计划;执行子系统(PES)根据执行计划调用相应的组件 Web 服务实体;验证子系统(VS)验证组合结果是否满足用户需求标准;一旦发生组合失败,恢复子系统(RS)根据 VS 的验证结果分析错误原因,将组合过程回退到错误发生处,重新启动组合过程。

MODiCo^[15]是一个原型系统。它具有自己的注册机制,用户首先需要将组件 Web 服务注册到该注册机制中,然后才能调用相应的组件服务。Web 服务发现子系统根据用户需求在上述注册机制中搜索能够单独满足需求的组件 Web 服务。如果不存在,则调用 Web 服务组合功能,对用户需求进行解析,使用基于 AI-Planning 的方法对能够满足子需求的组件 Web 服务进行组合。

Synthy^[16]是根据 AI-Planning 的计划分解和资源调配思想构建的 Web 服务组合原型系统。它主要由 3 部分组成:逻辑组合子系统(LC)以目标为导向,根据功能需求描述自动生成抽象的组合模型;实体组合模型(PC)根据非功能性需求描述优化选择满足抽象模型功能需求的组件 Web 服务实体,另外系统本身还具有基于本体描述的 Web 服务注册机制,用户需要将调用的组件 Web 服务注册到该注册机制中。

ServiceCom^[17]系统将组合过程分为描述、计划、构建、执行 4 个阶段。在描述阶段,系统提供组合编辑器,为用户提供可视化的服务组合编辑环境。计划阶段,系统提供一个组件服务库,用以完成组合服务的实例化过程。构建阶段,系统提供自动生成可执行代码文件功能,并最终执行组合代码、返回组合结果。

ZenFlow^[18]是基于 BPEL 语言的一款可视化程度较高的 Web 服务组合软件系统。它类似于标准的编译环境界面,分为编辑及结构显示部分、文件浏览部分、错误显示部分。用户可以在工作区的主要界面以工作流程图和树状图形式编辑数据流和控制流中的 BPEL 元素。文件浏览部分显示最近打开的 BPEL 项目。可视化组合生成器根据编辑区域信息自动生成 BPEL 文件,并负责检查 BPEL 文件的语法及语义错误,将其显示在底部的错误信息面板中,并最终 BPEL 引擎执行组合过程。

表 1 系统分析结果

系统	ASTRO	Haley	SHOP2	SeGSeC	Argos	SELF-SERV	Fusion	MODiCo	Synthy	ServiceCom	ZenFlow	eFlow
人机需求获取	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y
界面结果显示	Y	Y		Y	Y		Y				Y	Y
预处理需求解析	Y	Y		Y		Y	Y				Y	
组合需求建模	工作流	AI-Planning	AI-Planning	工作流	工作流	工作流	AI-Planning	AI-Planning	AI-Planning	工作流	工作流	工作流
建模服务发现				Y		Y		Y	Y	Y	Y	Y
执行服务选择	Y	Y		Y		Y		Y	Y			Y
过程执行	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
控制流验证	Y						Y					Y
验证数据流验证	Y				Y		Y	Y				Y
QoS 评测		Y	Y	Y			Y		Y	Y		Y
监控过程监测	Y	Y	Y								Y	Y
错误处理错误分析			Y	Y			Y					
错误恢复			Y				Y					

E-Flow^[19]是 HP 公司开发的一个比较成熟的 Web 服务组合系统,支持定义 Web 服务接口、定制 Web 服务、管理组

合 Web 服务、建立组合服务抽象模型、执行组合过程等功能。系统将一个组合服务抽象成工作流程图的形式,流程图中包

含两类节点:组件服务节点、决策节点。系统实现动态服务组合及修改功能,具有服务发现功能,能够在一定程度上适应动态的网络环境。可以将组合服务发布,以供其它实体调用。

ASTRO 系统以流程图方式获取用户需求,但用户需要了解系统自定义的符号涵义。系统不具备服务发现功能。在选择步骤不考虑 QoS 等因素对组合的影响。系统一旦组合失败,没有相应的错误分析及恢复机制。Haley 使用基于 semi-Markov 的选择算法,在确定调用实体时考虑了不确定性和 QoS 因素,但系统也不具备服务发现、控制流及数据流的验证以及错误分析和恢复等机制。SHOP2 没有对控制流及数据流的验证措施,但可以得到关于组件 Web 服务的 QoS 信息,并具有跟踪回退功能。该系统预先绑定组件实体,因此不能很好地控制复杂结构,鲁棒性较差。与 SHOP2 相同,SeGSeC 也预先绑定组建实体。它解析自然语言描述的需求,根据解析结果逐步实体化工作流程,一定程度地提高了系统的灵活性,但同时增加了组件服务搜索空间,使搜索匹配难度加大。系统没有监控措施,并且该系统假设所有的组建服务都有其自身的语义描述,这在当前的网络环境中是不能被满足的。Argos 是已经成功应用于商业领域的 Web 服务系统之一,但主要应用领域局限在数据处理方面,具有其自身的局限性。SELF-SERV 是较早出现的 Web 服务组合系统,它以状态转换图的形式描述组合流程,具备服务发现及服务选择功能,并可以将产生的组合服务注册到 UDDI 注册机制,以供调用。Fusion 具备了错误恢复机制及验证功能。系统需要根据自身本体集合定义组件 Web 服务的接口,并且不具备服务发现及选择功能,但能够根据 QoS 因素优化组合流程。MODiCo 和 Synthy 都是 Web 服务组合的原型系统,MODiCo 的主要特点是可以应用于多种本体描述的组件服务集合中。ServiceCom 系统支持 UDDI 服务发现功能,并可以将生成的组合 Web 服务注册到 UDDI。E-Flow 支持定义 Web 服务接口、定制 Web 服务、管理组合 Web 服务、建立组合服务抽象模型、执行组合过程等功能,以工作流程图的形式描述组合服务抽象模型,具有服务发现及注册功能,能够在一定程度上适应动态的网络环境。

结束语 任何软件系统本质上都是实现一定目标的功能集合,通常能够按照不同的子功能将系统划分为不同的功能模块。本文提出了一种基于功能模型的 Web 服务组合系统分析对比方法。根据 Web 服务组合的特点,将 Web 服务组合系统分为 7 个子模块:人机界面、预处理模块、组合建模模块、执行模块、验证模块、控制模块、错误处理模块,涵盖了 Web 服务组合系统应该具有的各项功能。需要注意的是,并不是 Web 服务组合系统都必须具备全部子模块的功能,而是需要根据用户和开发人员的需求进行选择。

根据给出的 Web 服务组合系统的功能模型,本文对 12 个 Web 服务组合工具系统进行了分析对比,并给出了对比结果。用户使用 Web 服务组合系统进行组合服务开发时,可以根据功能需求选取相应的系统,避免了因盲目选择系统而导致最终的组合服务不能满足用户需求。对 Web 服务组合系统开发人员来说,可以参考 Web 服务组合系统的功能模型,构建服务组合系统。

参 考 文 献

[1] Alonso G, Casati F, Kuno H, et al. Web Services-Concepts, Ar-

- chitectures and Applications [M]. Springer, November 2003
- [2] Bucchiarone A, Gnesi S. A Survey on Services Composition Languages and Models[C]// International Workshop on Web Services Modeling and Testing, 2006; 37- 49
- [3] Booth D, Haas H, McCabe F, et al. Web services architecture [S]. W3C Working Group Note, W3C, 2004, Vol. 11
- [4] Dustdar S, Schreiner W. A survey on Web services composition [J]. Web and Grid Services, 2005, 1(1)
- [5] Fujii K, Suda T. Dynamic service composition using semantic information[C]// Proceedings of the Second International Conference on Service Oriented Computing (ICSOC'04). New York, USA, 2004
- [6] Robinson W. Monitoring Web service requirements[C]// Proceedings of the International Conference on Requirements Engineering, 2003
- [7] Bhiri S, Gaaloul W, Godart C. Discovering and improving recovery mechanisms of composite Web services[C]// ICWS. Los Alamitos; IEEE Computer Society, 2006; 99-110
- [8] ASTRO Project. Supporting the Composition of Distributed Business Processes[EB/OL]. <http://astroproject.org>
- [9] Zhao Hai-bo, Doshi P. Haley: A Hierarchical Framework for Logical Composition of Web Services[C]// IEEE International Conference on Web Services(ICWS). 2007
- [10] Sirin E, Parsia B, Wu D, et al. HTN Planning for Web Service Composition Using SHOP2[J]. J. Web Sem, 2004, 1(4): 377-396
- [11] Fujii K, Suda T. Dynamic service composition using semantic information[C]// 2nd International Conference of Service Oriented Comput. 2004
- [12] Luis J, Giuliano G, Gordon P. ARGOS: Dynamic Composition of Web Services for Goods Movement Analysis and Planning[C]// Proceedings of the 8th Annual National Conference on Digital Government Research. Philadelphia, PA, 2007
- [13] Quan Z, Boualem S, Marlon B, et al. SELF-SERV: A Platform for Rapid Composition of Web Services in a Peer-to-Peer Environment[C]// Proceedings of the 28th VLDB Conference. Hong Kong, China, 2002
- [14] VanderMeer D, Datta A, Dutta K, et al. FUSION: A system allowing dynamic Web service composition and automatic execution[C]// IEEE International Conference on E-Commerce(CEC'03). California, USA, 2003; 399-404
- [15] Le D N, Tran B D, Tan P S, et al. MODiCo: A Multi-ontology Web Service Discovery and Composition System[C]// ICWE 2009, LNCS 5648. 2009; 531-534
- [16] Srivastava B. The Synthy Approach for End to End Web Service Composition; Planning with Decoupled Causal and Resource Reasoning[C]// Anerucab Assicuatuib fir Artificial Intelligence. 2006
- [17] Orriens B, Yang Jian, Papazoglou M P. ServiceCom; A Tool for Service Composition Reuse and Specialization[J]. Web Information Systems Engineering, 2003
- [18] Patino-Martinez M A, Jimenez-Peris M, Perez-Sorrosal R, et al. A Visual Web Service Composition Tool for BPEL4WS[C]// Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing. 2005
- [19] Casati F, Ilnicki S, Jin L, et al. Adaptive and dynamic service composition in eFlow[C]// Proceedings of the CAiSE Conference. Stockholm, 2000; 13-31
- [20] May Chan K S, Bishop J, Steyn J, et al. A fault taxonomy for Web service composition [C] // Proceedings of WESOA07. Springer LNCS. 2007