

Web 服务组合测试综述*

丁志军^{1,2}, 周泽霞¹



¹(嵌入式系统与服务计算教育部重点实验室(同济大学),上海 201804)

²(上海市电子交易与信息服务协同创新中心(同济大学),上海 201804)

通讯作者: 周泽霞, E-mail: 1252385@tongji.edu.cn

摘要: 随着面向服务技术和云计算技术的不断成熟,尤其是面向服务体系结构 SOA 的不断完善以及推广,其主要内容 Web 服务已经被广泛应用.为了充分利用 Web 服务并解决单个 Web 服务功能有限的问题,业界将多个原子 Web 服务按照一定的规则和业务逻辑进行组合,以提供更多功能更强大的服务,实现了 Web 服务的增值和复用.为了保证 Web 服务组合的质量,需要对 Web 服务组合进行全面、充分的测试.然而,Web 服务组合的动态特性和分布式特点使得其测试技术和方法与传统的软件测试有很大的区别,存在很多挑战.针对 Web 服务组合测试,对近年来 Web 服务组合测试研究中的测试用例生成技术、回归测试技术、测试执行和度量方法进行了系统的总结和分析.此外,还对 Web 服务组合测试中有待研究的问题进行了分析和展望.

关键词: Web 服务组合;测试;测试用例生成;测试执行和度量;回归测试

中图法分类号: TP311

中文引用格式: 丁志军,周泽霞.Web 服务组合测试综述.软件学报,2018,29(2):299-319. <http://www.jos.org.cn/1000-9825/5383.htm>

英文引用格式: Ding ZJ, Zhou ZX. Survey on Web service composition testing. Ruan Jian Xue Bao/Journal of Software, 2018, 29(2):299-319 (in Chinese). <http://www.jos.org.cn/1000-9825/5383.htm>

Survey on Web Service Composition Testing

DING Zhi-Jun^{1,2}, ZHOU Ze-Xia¹

¹(Key Laboratory of Embedded System and Service Computing of Ministry of Education (Tongji University), Shanghai 201804, China)

²(Shanghai Electronic Transactions and Information Service Collaborative Innovation Center (Tongji University), Shanghai 201804, China)

Abstract: As service-oriented technology and cloud computing technology continue to mature, and especially the service-oriented architecture (SOA) continue to improve and proliferate, content Web services have become widely used. In order to take full advantage of general Web services and overcome the problem of limited functionality of individual Web service, the industry embraces approach of combining multiple atomic Web services in accordance with certain rules and business logic to provide more functionality and more powerful services, enabling the increment and reuse of Web services. To ensure the quality of the Web service composition, comprehensive and adequate testing of the Web service composition is required. However, due to the dynamic and distributed features of Web service composition, the testing techniques and methods are different from those of traditional software, thus bringing many challenges. This paper systematically summarizes and analyzes test case generation techniques, regression testing techniques, test execution and measurement methods in Web service composition testing research in recent years. The paper also provides an analysis and outlook on the issues that need to be studied in the Web service composition testing.

Key words: Web service compositions; testing; test case generation; test execution and measurement; regression testing

* 基金项目: 国家自然科学基金(61672381)

Foundation item: National Natural Science Foundation of China (61672381)

收稿时间: 2017-05-24; 修改时间: 2017-07-31; 采用时间: 2017-09-03; jos 在线出版时间: 2017-10-09

CNKI 网络优先出版: 2017-10-09 16:20:54, <http://kns.cnki.net/kcms/detail/11.2560.TP.20171009.1620.005.html>

Web 服务是一个平台独立的、低耦合的、自包含的、基于可编程的 Web 应用程序,可使用开放的 XML(标准通用标记语言下的一个子集)标准来描述、发布、发现、协调和配置这些应用程序,用于开发分布式的互操作的应用程序.Web 服务的兴起伴随着面向服务的软件体系结构(SOA)的提出,为 Internet 上的分布式计算提供了一种基于标准的、松散耦合的跨平台的新范型.单个 Web 服务提供特定的功能,为了满足用户的需求,越来越多的实际项目需要将多个 Web 服务集成、组合起来,以便提供综合、复杂的增值服务——组合 Web 服务.该服务的组成成员之间可以互相通信,它们按照一定的逻辑处理用户操作和请求.随着 Internet 应用的进一步发展,Web 服务组合必将得到广泛关注和应用.为了保证组合 Web 服务的质量,需要对 Web 服务组合进行全面、充分的测试.然而,由于 Web 服务组合本身动态特性和分布式特点,很多传统的软件测试技术对于 Web 服务组合已失去其原有的有效性.因此,我们需要研究针对 Web 服务组合的新的测试技术和方法,从而为组合服务的性能、功能和可靠性方面提供有力的支持.

目前,国内外已经对 Web 服务组合测试进行研究并取得了一定的研究成果,本文的目的是对现有的一些 Web 服务组合测试的方法和技术进行系统的总结和分析.虽然部分研究者已经就该问题进行了一定的剖析和探讨,但我们认为,该问题仍具有进一步深入调查的必要.文献[1-4]提供了 Web 服务组合测试的调查情况.Hong 等人^[1]、Bozkurt 等人^[2]和 Ebrahim 等人^[3]对 Web 服务测试进行分析和总结,但没有重点阐述 Web 服务组合测试的研究现状,而只将其作为其中的一章或部分内容讲解.Rusli 等人^[4]根据测试方法来分类,进而讨论 Web 服务组合测试.然而,Rusli 等人的研究是在 2011 年完成的,之后已经有很多其他优秀的文献发表.另外,该研究没有对组合服务的回归测试进行调查,因此,做一个全新且全面的调查综述是十分有必要的.

为了对 Web 服务组合测试研究问题进行全面系统的分析、比较和总结,我们首先在 IEEE,ACM 和 CNKI 等论文数据库中进行检索,检索时采用的主要英文关键词包括“Web service composition test”“BPEL test”“SOA test”等;然后,对检索出的论文进行人工审查,删除与研究问题无关的论文,并通过查阅论文的参考文献和作者发表的论文列表找出遗漏的论文;最终,我们选取了 70 余篇与该研究问题相关并且高质量的论文(论文发表日期截止到 2017 年 1 月),绝大部分论文发表在软件工程领域或者 Web 服务的权威会议或期刊上.

本文第 1 节对 Web 服务组合及其特性进行介绍.第 2 节对 Web 服务组合测试存在的挑战进行总结.第 3 节对传统软件测试和 Web 服务组合测试进行分析和比较.第 4 节对 Web 服务组合测试的第 1 阶段测试用例生成技术和方法进行总结.第 5 节对测试的维护阶段,即回归测试的相关技术和方法进行总结和分析.第 6 节对测试的执行和度量阶段的已有研究工作进行总结.最后总结全文,并对未来值得关注的研究方向进行初步探讨.

1 Web 服务组合

面向服务体系架构(SOA)的概念最初由 Gartner 公司于 1996 提出,由于当时的技术水平和市场环境尚不具备真正实施 SOA 的条件,因此,当时 SOA 并没有引起人们的广泛关注.伴随着互联网的浪潮,越来越多的企业将业务转移到互联网领域,但由于大部分企业的信息技术架构都不易扩展、不够灵活、开发周期长和利用率低等问题,使得公司要花费大量的资金来升级、维护原有的架构来适应业务的演化和技术的发展.面向服务体系架构(SOA)的应用,上述问题都能得到有效的解决,SOA 能够有效提高企业系统的复用性、灵活性和可扩展性.

Web 服务是 SOA 的主要实现技术,支持分布式软件的开发、集成和使用.W3C 将 Web 服务定义为:“Web 服务是由 URI 标识的软件应用,其接口和绑定可以用 XML 来定义和使用,并且可以被发现,基于 Internet 协议通过交换 XML 消息与其他软件交互.”Web 服务保证了不同平台、不同编程语言下的应用服务之间的互操作问题:一方面,Web 服务所有的规范都基于 XML,任何与平台、语言相关的数据都可以映射为 XML 文档;另一方面,由于 Web 服务对外只提供接口文档(WSDL),在服务接口不变的情况下,可对服务的实现方式进行更改而不影响对该服务的正常调用,从而解决了客户端与服务器端之间的紧耦合问题.

Web 服务组合是将多个原子 Web 服务按照一定的规则和业务逻辑进行组合得到的功能更强大的服务,该组合 Web 服务既可以作为最终的 Web 服务提供给用户,又可以作为新的 Web 服务发布到网络上.Web 服务组合是通过确定组件 Web 服务的执行顺序和各组件 Web 服务之间的交互来实现的,其中,WS-BPEL(Web 服务的商

业流程执行语言)是专为组合 Web 服务而制定的一项标准的流程语言.Web 服务组合具有层次性、动态性、松耦合和可扩展性等特性。

2 Web 服务组合测试存在的挑战

Web 服务组合测试是保证 Web 服务组合质量的有效手段,但测试是一项非常复杂和耗时费力的工作.在实际应用中,测试的资源往往很有限,同时又需要我们全面检测 Web 服务组合以满足更多的测试需求.一种理想的测试方法需要同时具有高检错能力、低成本消耗和较为广泛的适用性等特点.

与传统的组件相比,Web 服务组合有许多独特之处,如 Web 服务组合自身的动态特征和分布式特点等,这些特征给测试工作带来了许多挑战,主要集中在以下几个方面.

(1) Web 服务组合的组件服务黑盒化

SOA 架构中的相关人员包括服务提供者、服务开发者、服务集成者、用户以及 UDDI 注册中心等.服务组合过程由服务集成者完成,所以服务组合的测试主要由服务集成者完成.Web 服务组合测试包括组件服务测试和业务流程正确性测试.服务集成者拥有描述组合服务业务流程的 WS-BPEL 文档,该文档是代码性质的文件,因此,测试者可以对业务流程进行白盒测试以检测业务流程的正确性.然而,由于 SOA 架构的松耦合、跨平台和互操作性,Web 服务的程序代码只对服务开发者开放,服务集成者只能获取 Web 服务的接口描述文档(WSDL),无法获取具体的代码信息.因此,服务集成者只能对 Web 服务进行黑盒测试.如何根据 WSDL 文档产生有效测试的测试数据对组件服务进行充分的测试,是一个亟需解决的问题.

(2) Web 服务组合的动态演化

组合服务在其生命周期中要进行不断的演化以满足现实用户的需求.Web 服务组合的演化主要是由运行环境的改变、用户需求的变化和缺陷修复等原因引起的.为了满足快速变化的环境,服务集成者需要对组合服务进行适时演化,使其适应不断变化的外部环境;用户可能需要增加某些额外的业务功能,集成者需要对原有的组合服务进行扩充改进,从而完善组合服务的性能、增强服务的效率;在组合服务构建过程中,由于技术人员的经验不足、测试不充分,导致组合服务存在缺陷,因此需要对组合服务进行演化以修改这些错误.综上所述,Web 服务组合演化就是对单个的 Web 服务进行修改、升级或者利用新的服务进行替换,以及对组合服务的工作流进行修改.这些改变时刻在发生,为了确保正确性,需要反复进行回归测试.然而,由于传统软件中的演化都是静态的,所以传统的回归测试技术不再适用于解决 Web 服务组合的动态演化问题.因此,提出适用于解决 Web 服务组合动态演化问题的回归测试技术,是 Web 服务组合测试研究的一大挑战.

(3) Web 服务组合的动态绑定

Web 服务组合支持静态绑定和动态绑定.在大多数业务流程设计中,外部 Web 服务的绑定地址是固定的,即该流程调用指定的服务、该方法适用于高度定向的系统.而在大型系统中,业务流程更加复杂.它们与多个外部 Web 服务交互,且部分 Web 服务的地址是不固定的,即组合定义过程中不为活动指定固定的服务提供者,将具体的绑定延迟到组合服务执行时动态完成,根据业务流程运行时的输入动态调用 Web 服务,该过程为 Web 服务的动态绑定.动态绑定使开发人员可以通过配置或运行时的输入添加新服务,不需要在设计时预计和管理 Web 服务组合中的所有父子关系,是静态 Web 服务组合的扩展,得到学术界和工业界广泛的认可与应用.然而动态绑定的“运行时”特征,使得传统的静态测试方式(开发时测试)不再适用于 Web 服务组合测试环境中.如何针对“动态”的业务流程进行运行时测试,成为亟需解决的难题.

3 传统软件测试与 Web 服务组合测试

由第 2 节中介绍的 Web 服务组合测试存在的挑战可知:如何对 Web 服务进行足够的测试、如何解决 Web 服务组合动态演化问题的回归测试问题以及如何针对动态的业务流程进行运行时测试,是在对 Web 服务组合进行测试时必须考虑的问题,也是研究者们一直在关注的重点问题.和传统软件测试一样,Web 组合服务测试过程分为测试用例生成、测试执行、测试结果度量和回归测试这 4 个阶段.但 Web 服务组合测试与传统软件测

试存在很大不同,二者之间的区别见表 1.从表 1 可以看出,与传统软件测试具有专门的测试小组和软件开发测试人员相比,Web 服务组合的测试者是服务集成者,测试也由传统的、集中的多阶段测试变为分布的、远程的在线运行时测试.

Table 1 Traditional software testing and Web service composition testing

表 1 传统软件测试与 Web 服务组合测试

比较项	传统软件测试	Web 服务组合测试
测试者	专门的测试小组或软件开发人员	服务集成者
测试分布	集中的、多阶段测试	分布的、远程的、多阶段测试
测试覆盖	对软件可以进行白盒测试和黑盒测试	对组件服务进行黑盒测试,对描述业务流程的 BPEL 文档进行白盒测试
测试类型	单元测试、集成测试、系统测试、验收测试、回归测试、冒烟测试等	单元测试、集成测试、系统测试、回归测试
测试模型	拥有软件代码,可以根据软件的特征建立丰富的测试模型	由于测试者不拥有组件服务的源代码,所以测试者只能建立可控性和可观性低的测试模型
测试执行	离线测试	运行时测试
测试客户端	软件本身	需要构建客户端调用被测组合服务的组件服务,如代理
测试预言	软件的行为是可预知的,因此,测试预言的生成比较容易	由于组件服务的动态绑定,导致组合服务的状态和行为是很难预测的,测试预言的生成比较困难
软件演化	静态演化,程序内部结构或变量更改	动态演化,组件服务进行升级或替换、业务流程(BPEL 文档内部)进行更换、组件服务接口信息(WSDL 文档内部)进行更改
回归测试	离线的,软件为静态演化,可以充分了解软件的变更情况,及时进行回归测试	基于运行时收集数据的回归测试,是在线的;组合服务具有动态演化特征,很难掌握服务的演化情况,且在回归测试过程中同样会有演化发生,因此需要额外的信息支持回归测试进行

与传统软件测试相比,由于 Web 服务组合自身的特征,Web 服务组合测试者无法拥有全部的测试信息,因为组件服务是黑盒化的,测试者无法拥有组件服务的源代码,所以无法获取组件服务的所有特征建立丰富的测试模型,因此,在 Web 服务组合测试中,部分学者就如何扩展文档解析技术或者模型建立技术来获取充分的测试信息进行了深入研究.再者,Web 服务组合运行时的动态绑定,使得组合服务的运行环境和行为很难预测,测试预言的生成比较困难,而传统的软件测试技术针对的软件的行为是可预知的、静态的和非分布式的,因此无法适用于 Web 服务组合测试.最后,和传统软件一样,Web 服务组合同样存在软件演化过程,但该演化是动态的,不是仅限于程序内部结构或变量的更改,而是对组件服务进行升级或替换、对业务流程进行更换以及对组件服务的接口信息进行更改,且这些演化过程存在于系统的整个运行过程中,因此需要额外的信息支持回归测试进行.

从上文的比较可以看出,Web 服务组合测试与传统软件测试之间的差异存在于测试的整个过程中.本文根据测试过程对 Web 组合服务测试不同阶段的技术和方法进行总结与分析.接下来,第 4 节将对 Web 服务组合测试的第 1 阶段测试用例生成技术和方法进行总结.第 5 节对测试的维护阶段,即回归测试的相关技术和方法进行总结和分析.第 6 节对测试的执行和度量阶段的已有研究工作进行分析.

4 测试用例生成

测试用例生成的 Web 组合服务测试的第 1 阶段.测试用例是为某个特殊目标而编制的一组测试输入、执行条件以及预期结果,以便测试某个程序路径或核实是否满足某个特定需求.测试用例生成研究领域中,问题主要集中在运行时测试、测试用例精确度和测试用例冗余度 3 个方面.现有研究中,Web 服务组合测试用例生成阶段采用的方法和技术主要分为基于模型的测试技术、基于突变的测试技术、基于文档解析的测试方法和基于启发式的测试方法等.由于 Web 服务组合自身的多样性、动态性和并发性,测试用例产生存在很多挑战.基于模型的测试技术具有自动化产生测试用例的优点,可以有效解决该问题.根据基于模型的测试技术,测试用例生成方法进一步可以分为两种:形式化方法和非形式化方法.

4.1 基于模型的测试技术

4.1.1 形式化方法

基于形式化方法的测试用例生成可以分为基于模型检测器的测试用例生成和基于形式化分析技术的测试用例生成两种。基于模型检测器的测试用例生成方法是将 BPEL 等描述的服务组合转化为某种检测器的输入模型,并用形式化方法描述该组合服务应该满足的需求模型,将二者作为该模型检测器的输入,进而产生测试用例,大多数研究中用到的检测器包括 NuSMV, SPIN 和 BLAST 等^[5-8]。基于形式化分析技术的测试用例生成方法就是将 BPEL 等描述的服务组合用某种形式化方法描述,如 Petri 网、自动机或进程代数,然后采用该种形式化方法已有的分析技术实现测试用例的生成,如文献[9-11]。

Petri 网是分布式系统的一种建模和分析工具,是一个由库所、变迁、有向弧构成的有向图,它便于描述系统中进程和部件的顺序、并发、冲突以及同步等关系。和其他系统模型相比,对真并发的描述是 Petri 网的独特优势。基于 Petri 网的建模方法可以描述组合流程中的各种控制流,但不能直接体现组合服务的全局状态。

自动机是一种具有清晰语义的数学模型,适用于描述具有离散输入和输出的系统。该系统具有有限个状态,不同的状态代表不同的意义,按照实际的需要,系统可以在不同的状态下完成规定的任务,从而转移到另一个状态。基于自动机的建模方法可以直观地描述 Web 服务组合的内部状态,但无法描述两个服务之间的交互行为,对组合流程中并发活动的描述能力有限,且存在空间爆炸问题。

进程代数是基于代数方法的形式化建模语言,语法上用一组算子作为进程构件,算子的语义采用结构化操作语义方法定义,这样,进程就可看作是带标号的变迁系统(LTS)。进程代数的一个显著特征是把并发性归结为非确定性,即将并发执行的进程的行为看作是各单进程的行为的所有可能交错,适用于描述并发的交互系统。进程代数的建模方法有着很强的描述能力以及严谨的计算推理的能力,但其表达方式比较复杂,不够直观且不易理解。

以上 3 种形式化模型都能很好地描述 Web 服务组合的行为,且都具有相关的技术支撑和工具支撑,只是在计算复杂度上存在些差异。然而这些方法都需要人员具备相关的专业背景知识,且大都忽略了组合流程中数据流信息的建模。因此,本文主要集中讨论非形式化测试用例生成方法。

4.1.2 非形式化方法

非形式化方法^[12-34]是将 BPEL 等描述的组合服务中的控制流、数据流、消息流、行为等转换成图模型,然后采用搜索技术、约束解析技术实现测试用例的生成。进一步地,测试用例产生方法的步骤具体为:根据 Web 服务组合某一特征进行建模;基于模型遍历产生测试路径;基于路径中的约束条件生成测试用例。以下研究工作属于这一范畴。

部分研究者在研究 Web 服务组合测试用例生成问题时,首先关注的是 Web 服务组合的控制流特征。Yuan 等人^[12]提出了一种基于图搜索的 BPEL 测试用例生成方法,并用该方法来处理 BPEL 的并发语义。该方法通过定义控制流图——BPEL 流图(BFG)来描述 WS-BPEL 程序,BFG 包含 BPEL 程序的控制流和数据流。遍历 BFG 模型可以产生测试路径,测试路径中的约束条件作为约束解析器的输入,进而产生测试数据。由于产生的测试数据是抽象的,所以该方法进一步将测试数据转化为可执行的测试用例,该过程是自动化完成的。然而在该方法中,将 BPEL 文档转换为 BFG 以及测试路径的搜索过程却是手动完成的。

Yan 等人^[13]所提出的方法与 Yuan 等人^[12]的方法类似。他们将 WS-BPEL 程序转换为扩展的控制流图(XCFG),基于 XCFG 产生测试路径,然后利用约束解析器从测试路径中产生测试数据。然而与 Yuan 等人的方法不同的是,为了考虑组件服务的调用情况,Yan 等人使用符号执行的方法从测试路径获取一系列约束条件。然而,该方法产生的是抽象的测试用例,需要手动转换为可执行的测试用例。

Mei 等人^[14,15]解决了 BPEL 流程中 XPath 可能引起的集成问题,他们认为,WS-BPEL 应用程序是一种面向服务的应用程序,广泛使用 XPath 来整合松散耦合的工作流程步骤。但是,XPath 可能从收到的 XML 消息中提取错误的信息,从而导致集成过程中的错误。因此,Mei 等人提出一种基于 XPath 重写图(XRG)的测试方法。该方法结合 BPEL 流程的控制流图(CFG)和 XRG 来模型化 XPath 的重写方式:首先,利用 XPath 构造数学定义,将其作

为重写规则;XRG 不仅可以模拟 XPath 的重写方式,还可以逐步跟踪单个重写;应用重写规则中的数学变量作为程序变量,并用它们来分析信息如何在 XPath 中重写.Mei 等人也因此开发了构建 XRG 的算法以及一系列新型的数据流测试标准来测试 WS-BPEL 应用程序,实验结果表明,该测试方法十分具有前景.

随着研究的逐步深入,有学者认为:当下,基于模型的测试用例产生技术都是基于路径产生测试用例的,该方法通过产生消息参数来表示测试数据,生成的测试用例冗余度较高,而检错率比较低.因此,如何大幅度提高测试用例的有效性成为当务之急.

为了解决该难题,Ni 等人^[16,17]首次将面向对象的程序(OOP)中基于消息流的测试技术应用到 Web 服务组合测试中.他们认为:由于 WS-BPEL 程序定义一个业务流程,在 activity 中组合多个 Web 服务,对程序进行测试的测试用例通常是一序列的 SOAP 消息;再者,WS-BPEL 程序中的消息接收者能够创建状态实例.所以在测试时,如何使 WS-BPEL 程序实例在业务流程执行中进入所期望的状态,是十分有必要研究的.

Ni 等人认为,Web 服务组合交互中的消息路由机制中最重要的 3 个要素分别是:结构活动中消息接收者的顺序关系;消息之间的关联集 correlation sets 的值;消息接收者中的属性 createInstance,该属性决定是否产生新的实例.基于该路由机制,Ni 等人将 WS-BPEL 程序中的结构活动和消息关联集进行消息序列图(MSG)模型化.MSG 的构建过程分为 3 步:第一,模型化顺序关系,消息接收者的顺序关系反映在 WS-BPEL 程序中的 receive 结构活动,因此解析 WS-BPEL 文档,得到每一个包含消息接收者的结构活动,按照消息接收者顺序构建初始的 MSG;第二,解析 WSDL 文档,获得消息(message)的类型定义,记录在初始 MSG 节点中;第三,进一步解析 WS-BPEL 文档,获得 correlation sets 信息,将其填充到 MSG 节点中,得到最终的 MSG.通过设计消息序列产生技术(即深度优先遍历 MSG)得到消息序列集,为消息序列产生满足关联条件的参数,得到最终测试 WS-BPEL 的测试用例.在实验部分,Ni 等人使用模拟的组合服务进行测试,实验结果表明,该方法的错误暴露率明显比随机生成法(RAND)和基于图搜索法(GS)要高.

不同于以往的基于路径的测试方法都是基于消息参数生成测试用例,Ni 等人所提出的基于 MSG 的测试技术能够同时产生消息参数和消息序列,二者结合生成测试用例,测试用例的有效性必然会更高.但他们主要提取 WS-BPEL 中的信息,对 WSDL 文档只提取了消息类型信息,因此对原子服务的检错能力很薄弱.除此之外,只集中对 WS-BPEL 文档解析,必然导致 XML 文档解析不完整,提取到的约束条件不充分,进而导致生成的测试用例不精确,冗余度较高.他们也阐述了该方法在实验时会产生无效的测试用例,这也验证了测试用例的冗余度仍旧有待降低.

Web 服务组合内部的各个组件 Web 服务通常是抽象的,只有在运行时才能绑定到具体的 Web 服务.这种运行时绑定会延迟测试的执行过程,并且需要在组合服务运行时才能进行测试,因此传统的集成测试技术是不适用的.为了解决该问题,大量学者对 Web 服务组合的运行绑定特征进行深入研究,提出有效的运行时测试技术,进而提高 Web 服务组合测试的有效性.

Wu 等人^[18]认为,运行时绑定与单个服务的内部状态、服务之间的执行序列和服务的行为有密切关系,因此,针对 Web 服务组合提出了一个 EDSM 序列测试模型(EFSM-SeTM).该模型的构建过程如图 1 所示.

Web 服务分为有状态的服务和无状态的服务.无状态的服务即为可以被重复调用的服务.该服务无需维持上下文或状态,每当用户与服务交互一次,就会完成一个处理.有状态的服务需维持上下文或状态,即维持不同操作调用之间的状态.在 Web 服务组合中,服务之间的交互和服务的动态行为与 Web 服务的状态具有紧密联系.因此,Wu 等人认为:有必要在每个组件服务不同的状态下,对组件服务之间的交互和服务的动态行为进行测试.

文献[18]首先解析组合服务中的状态 Web 服务的 WSDL 文档,获取各个状态服务的所有操作信息、操作的前置条件和操作的后置条件,即获取 Web 服务的所有状态.然后利用扩展的有限状态机(EFSMs)刻画状态服务内部的所有状态和状态之间的转换;其次,利用 XSLT 和 XMI 两种技术从组合服务 WS-BPEL 描述文档中提取出 UML 时序图,UML 时序图是以时间为序的表示方法,主要用来描述某个原子服务为完成某个行为而与其他原子服务之间所传递的消息的时间顺序;最后,将 EFSMs 模型和 UML 时序图结合,得到最终的测试模型 EFSM-SeTM.该模型包含原子服务之间的执行序列、服务的行为和服务的内部状态.

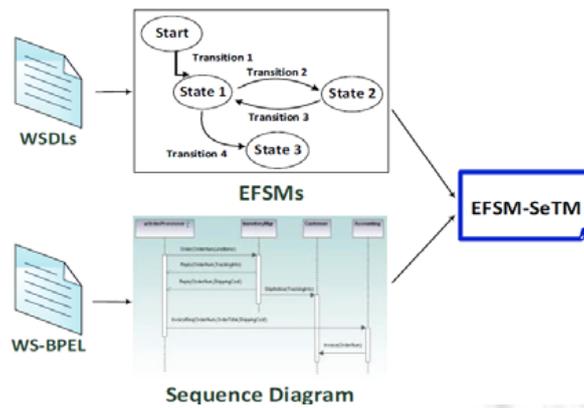


Fig.1 Production process of EFSM-SeTM [18]

图 1 EFSM-SeTM 产生过程[18]

基于 EFSM-SeTM 模型,选取能够覆盖所有场景的覆盖策略,基于该覆盖策略,遍历 EFSM-SeTM 模型,产生测试路径.文献[18]提出的 EFSM-SeTM 包含单个服务的内部状态、服务之间的执行序列和服务的行为,这些信息有利于实现运行时测试.但该模型也存在一些缺陷:首先,基于扩展的有限状态机或有限状态机来描述 Web 服务的内部状态,会出现状态空间爆炸问题;其次,UML 时序图反映服务行为和交互信息的传递过程,但是不能刻画组合服务内部的并发活动;再者,该文基于 EFSM-SeTM 模型,只得出测试路径,没有考虑路径中的约束条件,因此无法得到具有实际意义的测试用例;最后,该文未对所提出的模型进行实验评估,因此无法证明该方法的有效性.

Sun 等人[19,21]从 workflow 角度对运行时测试进行了深入的研究,并取得了一定的成果,但也存在一些问题需要进一步解决.他们从 workflow 角度分析,认为系统的功能可以描述成一系列的场景,一个场景即一种执行路径,因此可以对 WS-BPEL 程序进行场景测试,即寻找一切可行的场景,然后利用相应的测试数据执行场景,观察场景执行是否按预期进行.

所以,Sun 等人针对 Web 服务组合提出面向场景的测试框架如图 2 所示.首先,该框架对 BPEL 文档进行解析,得到结构化活动、基本活动和约束条件;其次,针对 WS-BPEL 中的结构化活动和基本活动制定映射规则,基于该映射规则,将 WS-BPEL 转化为 BPEL 图模型;然后,基于 BPEL 模型,结合覆盖策略进行深度优先遍历得到测试场景,将场景中的约束条件作为 Z3 约束解析器的输入获得测试用例;最终,结合 BPEL 引擎提供的两个组合服务 SupplyChain 和 SmartShelf 进行测试,对该方法进行有效性评估.

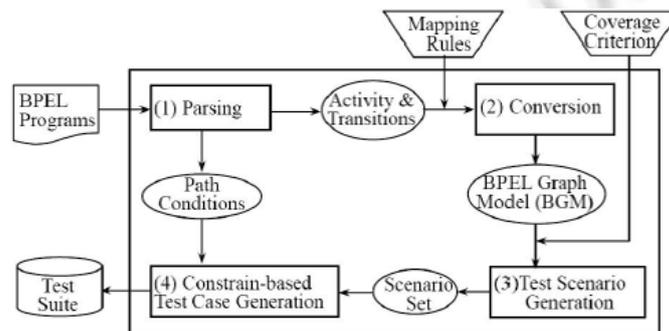


Fig.2 Scenario-Oriented testing framework [21]

图 2 面向场景的测试框架[21]

如果测试框架要适应 Web 服务组合的动态性,不仅要求测试用例生成过程完全自动化,而且整个测试过程

也需要完全自动化,且自动化程度越高,越能适用于 Web 服务组合环境测试^[21].因此,文献[21]进一步提出 Web 服务组合测试原型工具,如图 3 所示.

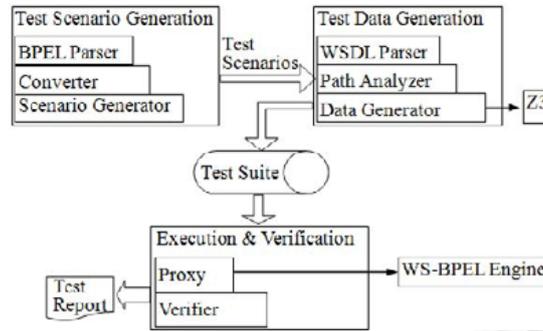


Fig.3 Automated test prototyping tool^[21]

图 3 自动化的测试原型工具^[21]

该原型工具是自动化的,支持运行时测试,解决了运行时绑定的问题.然而,基于 BPEL 模型的测试用例生成方法同样存在不足,主要体现在该方法只基于 WS-BPEL 文档信息进行测试用例的生成,对于原子服务内部存在的错误并没有检错能力;再者,本文提出的方法主要对控制流(路径)进行了分析,并没有充分解析 XML 文档获取约束条件,导致得到的约束条件并不充分,进而导致生成的测试用例不精确,冗余度高,检错能力较低.

综上所述,本节对基于模型的非形式化测试方法进行了总结和分析.具体情况见表 2.测试分为静态测试和运行时测试.在 Web 服务组合测试研究初期,研究者主要关注 Web 服务组合的静态测试,大部分都是基于控制流、数据流产生测试用例,能够产生有效的测试用例,但测试用例的质量有待提高.由于 Web 服务组合的运行时绑定特征,需要对其进行运行时测试,从表 2 中可见,针对该问题进行测试的文献比较少,但 Sun 等人^[19-21]、Ni 等人^[17]和 Wu 等人^[18]分别从场景、消息序列和服务内部状态模型来解决该问题,有一定的实际意义.再者,从表 2 可以看出,上述测试用例生成技术绝大多数都是半自动化的,即使是针对运行时测试提出的技术.因此,如何实现完全自动化,是可以继续进一步研究的问题.其次,没有任何一种测试用例生成技术涉及到 Web 服务,Web 服务的质量决定整个 Web 服务组合的正确性,因此,对 Web 服务测试是有必要的.

Table 2 Classification of informal methods

表 2 非形式化方法分类

类型	侧重点	模型	文献	测试用例质量	自动化测试	是否测试 Web 服务
静态测试	控制流, 数据流	控制流图扩展的控制流图, Xpath 重写图	Yan 等人 ^[13] , Mei 等人 ^[14,15]	抽象的测试用例	半自动化	否
	消息流	消息序列图	Hou 等人 ^[16]	测试用例的检错率高于随机生成法(RAND)和基于图搜索法(GS), 但冗余度较高	半自动化	否
运行时测试(运行时绑定问题)	服务交互与动态行为	状态转换图, 消息交互时序图	Wu 等人 ^[18]	没有考虑路径中的约束条件, 无法得到具有实际意义的测试用例	半自动化	否
	基于场景	BPEL 模型	Sun 等人 ^[19-21]	只基于控制流(路径)进行测试, 即没有充分解析 XML 文档, 测试用例不精确, 冗余度高, 检错能力较低	自动化	否
	消息流	消息序列图	Ni 等人 ^[17]	测试用例精确, 检错能力有待提高, 冗余度高	半自动化	否

4.2 其他方法

除了基于模型的测试用例产生方法外,有少部分研究提出了基于突变、基于文档解析或基于启发式的测试

技术来产生测试用例。

第 4.1 节中所阐述的基于模型的测试方法都存在一个共同缺陷——将组件 Web 服务看作黑盒,致力于对它们的接口进行交互测试,前提条件假设这些组件服务在组合之前已经测试正确,这样必然导致这些测试技术所生成的测试用例无法检测出原子 Web 服务内部存在的错误。在 SOA 架构中,这个假设不总是成立,因为 Web 服务组合是根据业务流程逻辑和 Web 服务的接口将各个 Web 服务联系在一起,Web 服务功能的正确性决定组合服务运行的正确性,所以在 SOA 背景下,对 Web 服务的测试同样重要。

然而,目前常见的 Web 服务测试方法主要为黑盒测试,包括利用 WSDL 中的接口信息、结合本体知识库或者基于用户输入等方式对组合 Web 服务进行测试,但这些方法生成的测试用例冗余较高,测试覆盖率不理想,并且 Web 服务领域内本体知识库获取比较困难。

针对这些问题,Zhou 等人^[35]提出了一种基于文档解析和约束求解的 Web 服务组合测试用例生成方法(GTCWS)。GTCWS 首先通过对 XSD,WSDL,WS-BPEL 这 3 类 XML 文档进行联合解析,获得能够丰富测试信息的各种约束条件,然后利用 Z3 约束求解器对各约束条件进行求解,并结合不同的覆盖率准则生成测试用例。

由于黑盒测试方法无法获取输入变量有效的取值范围,导致生成的测试用例集冗余度较高,因此,GTCWS 使用约束求解的方法来生成有效的测试用例。约束求解方法即将接口文档中提取的约束条件归类为变量类型约束、输入输出约束、条件约束及活动次序约束,并提出了相应的约束编码规则来指导与约束求解器相关的约束文件的生成,然后利用约束求解器并结合 SOAP 协议生成有效的测试用例。在对接口文件进行解析时,无法获得外部服务调用的结果,所以他们通过符号执行来表示其执行结果。

针对 Web 服务组合中测试覆盖率较低的问题,GTCWS 根据中间文件生成 Web 服务组合控制流图,并基于不同覆盖率标准对与约束求解器相关的约束文件进行修改,包括节点覆盖、路径覆盖和 MC/DC 覆盖,并对约束文档进行修改,生成满足覆盖率准则的测试用例集。

文献[35]在对 GTCWS 方法的有效性进行评估时,将生成的测试用例在一些随机选取的真实的 Web 服务组合上进行了测试,并将实验结果与现有的 Web 服务测试用例生成方法 TCG(基于控制流图的方法)进行了对比分析。实验结果表明:GTCWS 测试用例生成方法通过解析接口文档,能够获取更加详细的约束条件,从而降低测试用例的冗余度;同时,结合覆盖率标准的约束条件修改方法提高了测试用例覆盖率。

测试用例的质量是执行有效的 Web 服务组合测试的根本条件,如何生成高质量的测试用例,是软件测试中的重要问题。基于突变的测试用例生成技术是基于错误的测试技术,能够生成高质量的测试用例。基于突变的测试用例生成技术通过突变产生器对源程序进行轻微的修改(产生突变),测试套件测试修改后的程序,对突变进行识别,如果无法检测到一些突变,则对测试套件进行改进,使其能够检测到这些突变,即生成高质量的测试用例。在 Web 服务组合测试研究领域,部分学者将传统软件测试中基于突变的测试方法代入 Web 服务组合测试中,如文献[36-40]。

Estero-Botaro 等人^[36,37]提出了一个 WS-BPEL 服务组合的遗传测试用例生成框架。该框架如图 4 所示。

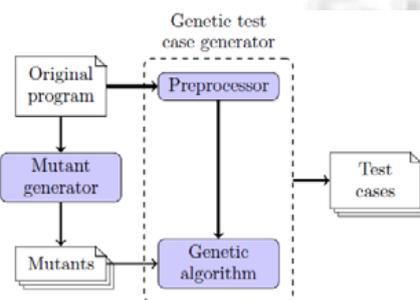


Fig.4 Architecture of the test-case generator^[36]

图 4 测试用例产生的框架^[36]

图 4 中的 preprocessor 可以生成遗传算法(GA)所需的信息:利用随机方法产生测试用例的初始种群,以便提供 GA 算法使用.在源程序中注入突变得到变异程序.该过程通过突变产生器完成.变异程序与测试用例的初始种群作为 GA 算法的输入.本文采用基于 bacteriological algorithms 特征设计的 GA 算法对测试用例进行择优选择,迭代遗传产生优化的测试用例.实验结果表明,基于该方法产生的测试用例的检错率和覆盖率得到很大的提高.

Blanco 等人^[41,42]提出了一种基于元启发式技术的分析搜索方法,用于 BPEL 业务流程的自动测试用例生成.测试用例生成器称为 TCSS-LS-for-BPEL.它将多样性属性与本地搜索相结合.多样性属性用于扩展测试用例的搜索,以实现业务流程的不同转换.当多样化属性有问题时,通过本地搜索来加强搜索的效果,可以找到转换的测试用例中尚未涵盖的部分.在实验部分,使用两个样本组合服务呈现测试用例生成器获得的结果,并与随机生成器进行比较.结果表明,TCSS-LS-for-BPEL 可用于生成 BPEL 业务流程的测试用例.

综上所述,除了基于模型的测试用例产生方法外,还有部分研究者提出基于突变、基于文档解析和基于启发式的测试技术来产生测试用例.本节对这些测试方法和技术进行了总结和分析,具体情况见表 3.

Table 3 Classification of other methods

表 3 其他方法分类

技术	文献	测试用例质量	自动化测试	是否测试 Web 服务
基于文档解析和约束求解	Zhou 等人 ^[35]	通过解析接口文档,能够获取更加详细的约束条件,从而降低测试用例的冗余度	半自动化	是
基于突变	Estero-Botaro 等人 ^[36,37] , Dominguez-Jimenez 等人 ^[38]	由于测试用例经过变异、进化和择优遗传,测试用例的质量比较高	半自动化	否
基于启发式	Blanco 等人 ^[41,42]	将多样性属性与本地搜索相结合,得到有效的测试用例,但其检错率有待提高	半自动化	否

基于文档解析和约束求解的测试用例生成技术涉及到 Web 服务的测试,通过解析接口文档,能够获取更加详细的约束条件,从而降低测试用例的冗余度;并结合控制流图对测试用例覆盖率进行改进,有效地提高了测试用例的质量.基于突变的测试用例生成技术主要目的是产生优化的测试用例,因此对测试用例进行了择优选择和进化,但是现有技术的测试用例初始种群都是通过随机技术产生的,这对最终测试用例质量是否具有影响还有待进一步研究;再者,该部分的测试用例生成技术都是半自动化的,因此,如何实现完全自动化,是可以继续进一步研究的问题.

5 回归测试

由于功能增加、组合部件重构、性能调优、错误修复等原因,Web 服务组合通常处于动态演化中.Li 等人^[43]将 Web 服务组合演化分为 3 类:业务改变、绑定改变和接口改变.为确保软件的正确性,需要对其进行回归测试.

统计数据表明:回归测试占了软件测试总预算的 80%,软件维护阶段总费用 50%以上^[44,45].尽管回归测试的代价如此之高,但它却是不可或缺的测试.为了大幅度减小回归测试开销,国内外研究者对自动化回归测试技术进行了深入研究,许多新的高效的测试技术不断涌现,取得了阶段性的成果.但由于 Web 服务组合特性与传统软件特性存在差异,导致一些回归测试技术不能直接应用到 Web 服务组合中,需要结合 Web 服务组合的特性进行调整和修改.根据调查和分析,可以认为,Web 服务组合回归测试研究还处于初级阶段.我们将传统软件回归测试与 Web 服务组合回归测试之间的区别与联系归纳为表 4.

典型的回归测试技术包括测试套件约简、测试用例选择、测试用例优先排序、测试用例修复与扩增.测试套件约简用于确定并消除过时或者冗余的测试用例,临时地产生一个测试套件子集.然而,删除测试用例有可能丢失发现错误的机会,因此,有的研究者提出测试套件约简仅选择 1 个测试用例子集而不删除测试用例.测试用例选择是从测试套件中选择一个测试用例子集.该子集能够用于测试修改的程序单元或者潜在地受修改影响的程序单元.测试用例优先排序通过优化测试用例的执行次序来提高回归测试效率.但在回归测试的实际应用中,仅采用测试套件约简、测试用例选择和测试用例优先排序技术并不能完全解决软件演化带来的挑战^[44].

- (1) 软件演化可能导致部分测试用例不可用,但直接丢弃这些测试用例将降低错误检测能力。
- (2) 软件演化引发模块的增加和修改,已有测试用例不能完全覆盖这些模块。

为了迎接上述挑战,部分研究者对回归测试技术进行了深入研究,提出了测试用例修复和测试用例扩增技术:测试用例修复是指对旧版本程序测试用例集中的不可用测试用例进行修复,使得修复后的测试用例能够在新版本程序上执行;测试用例集扩增技术是指根据新版本程序和已有测试用例来生成新的测试用例,促使新测试用例能够覆盖新版本程序的修改部分和新增部分^[44]。

Table 4 Traditional software regression testing and Web service composition regression test

表 4 传统软件回归测试与 Web 服务组合回归测试

区别/联系	比较项	传统软件回归测试	Web 服务组合回归测试
区别	软件修改识别方法	程序切片、数据流分析方法、Unix 工具、程序依赖图、UML 模型等	基于模型的方法,模型包括程序依赖图、控制流图、扩展的控制流图等
	剖面结构获取技术	动态插桩技术	运行时记录
	获取的剖面信息	函数覆盖信息、函数调用的时序信息、函数调用的关系信息以及函数调用的结构信息等	workflow 覆盖、数据流覆盖、XPath 覆盖、WSDL 覆盖
	剖面结构	函数覆盖向量、函数调用序列、函数执行序列与函数调用树	记录各个测试用例的覆盖量总和
	TCP 主要技术	贪婪法、机器学习法、专家融合知识法	贪婪法
联系	回归测试技术包括测试套件约简、测试用例选择、测试用例优先排序、测试用例修复与扩增		

测试用例选择、测试用例修复和扩增都需要进行修改识别,即,获取程序修改部分和被修改部分影响的内容,并且需要收集测试用例运行时的覆盖信息,针对修改部分的程序体,结合测试用例的覆盖信息对测试用例进行重新配置;而测试套件约简和测试用例优先排序只需获取测试用例运行时的覆盖量,根据各个测试用例的覆盖量对测试套件进行约简或优先排序。

传统的软件回归测试中,可以通过许多技术对当前版本 P 和修改版本 P' 的源代码进行分析,以此来确认 P' 的修改部分,如程序切片、数据流分析方法、Unix 工具、程序依赖图、UML 模型等。通过动态插桩技术可以收集运行测试用例时产生的剖面信息,比如函数覆盖信息、函数调用的时序信息、函数调用的关系信息以及函数调用的结构信息等。并构建 4 种不同类型的剖面结构,包括函数覆盖向量、函数调用序列、函数执行序列与函数调用树。这 4 种模型都可以反映测试用例对程序体的覆盖量,但是效果存在差异。

在 Web 服务组合回归测试技术的研究中,所有论文都集中探讨测试用例选择、测试套件约简和测试用例优先排序,未找到针对测试用例修复、测试用例扩增的研究文献。现有研究主要通过分别对当前版本 P 和修改版本 P' 的描述文档建立模型,比较两个模型的差异获取修改部分的内容,典型的模型包括程序依赖图、控制流图、扩展的控制流图。在组合服务测试运行时依据建立的模型,记录各个测试用例的工作流覆盖、数据流覆盖、XPath 覆盖或者 WSDL 覆盖。

在传统的软件测试中,为了实现测试用例优先排序(TCP),涉及的主要技术方法可以分为 3 类,分别为贪婪法、机器学习法和融合专家知识法^[45],而 Web 服务组合 TCP 的现有研究中,大部分都是基于贪婪法^[46-63]进行的,基于机器学习法和融合专家知识法的研究几乎没有。

下面将详细介绍 Web 服务组合回归测试研究现状。

5.1 测试套件约简

在 Web 服务组合回归测试研究领域,对测试套件约简(TSM)的研究还比较少。TSM 技术用于临时地产生一个测试套件子集或者是永久性地删除冗余的测试用例,并且这种选择与版本变化是无关的。由于 TSM 问题是 NP 完全问题,在传统的 TSM 技术研究中,部分研究者利用启发式的方法来解决;另一部分研究者则利用贪婪法来解决。之后,Yoo 等人^[64]针对 TSM 提出了 HSNGA-II 算法。该算法是启发式方法和贪婪法的结合,能够有效地解决 TSM 问题。

在 Web 服务组合的 TSM 研究领域,Bozkurt 等人^[65]也尝试利用 HSNGA-II 算法来实现 TSM。基于 HSNGA-II

算法,Bozkurt 等人提出了测试成本感知的 TSM 技术,从最小化测试成本的角度出发,删除冗余的测试用例,达到约简测试套件的目的.Bozkurt 等人将影响测试成本的因素可以归为 3 类,分别是:

- 在运行时测试中,频繁地进行服务调用;
- 由于广泛的测试会出现服务中断的现象;
- 在一些特殊的系统中进行测试,如证券交易所系统,每次使用该业务都需要进行业务交易,会产生额外的测试成本.

其中,由服务调用引起的运行时测试成本是 Web 服务组合测试的主要限制之一.然而,传统的 TSM 是针对线下测试设计的,无法解决实时测试代价的降低问题.因此,他们提出了全新的成本感知测试套件约简方法.该方法在不降低测试套件检错率的前提下,利用测试成本感知约简技术删除冗余的测试用例,缩减测试套件的规模,降低测试成本,提高测试效率.该技术由两个阶段组成.

(1) 测试套件衡量值计算

该方法利用随机测试技术产生测试用例.当测试套件产生后,该方法首先为测试套件中的每个测试用例计算 3 个衡量值:分支覆盖率、可靠性和执行代价(服务调用代价).测试用例的可靠性由其自身输入内容的可靠性决定;与可靠性不同,分支覆盖率和执行代价的信息不能从外部源获取,一般通过实时执行整个测试套件获取这些信息,但这往往会产生额外的代价,因此导致最终获取的执行代价是不准确的.因此,为了不产生额外的测试代价,Bozkurt 等人利用 stub/mock 服务模拟测试,进而获取测试用例的分支覆盖率和执行代价进行.

(2) 多目标最小化

TSM 是 NP 完全问题,因此可以利用多目标优化实现最终的测试套件约简.TSM 中的多目标优化表现为:测试套件的检错率不受约简影响、测试执行代价最小化以及可靠性最大化.为了实现多目标优化,文献[45]针对上述 3 个目标分别定义了 3 个目标函数,并结合 HSNGA-II 算法对冗余的测试用例进行删除,得到“最优”的测试子集.实验结果显示:该方法不仅能够有效地减小测试用例的规模,还能提高测试套件的执行效率和有效性,具有重要的实际意义.

5.2 测试用例选择

测试用例选择技术(TCS)也用于降低测试套件的规模,但 TCS 与程序版本变更相关.TCS 技术是为了选择足够且尽量少的测试用例来覆盖回归测试中被修改影响的部分.在测试用例选择的研究中,最主要的研究内容是如何进行修改识别,通过修改识别可获得当前 Web 服务组合被修改部分的内容.TCS 技术可以分为基于语义分析和基于语法分析两类.基于语法的 TCS 技术通过为新旧程序分别建立控制流图、扩展的控制流图或者数据流图,比较程序的数据流、消息序列或者工作流的差异获取修改的部分;基于语义的 TCS 技术则是通过为新旧程序分别建立程序依赖图来比较程序的数据依赖、控制依赖和异步调用依赖这些能够反映服务行为的信息,进而获取新旧程序的不同之处.

5.2.1 基于语法的 TCS

基于语法的 TCS 技术根据修改识别方法的不同可以分为基于控制流、基于数据流或者同时基于控制流和消息流这 3 类.基于控制流或消息流的 TCS 都是基于路径的方法,这也是目前最常用的选择方法.文献[66-74]都是基于路径的 TCS.

Tsai 等人^[66]提出了一种基于模型的自适应 TCS 方法.该方法通过构建覆盖关系模型对测试用例进行等级划分和选择.覆盖关系模型本质上移除了具有相同覆盖的测试用例.Ruth 等人^[67,68]将各原子服务的控制流图(CFG)组合为组合服务的全局 CFG.他们利用图遍历技术比较新旧程序的全局 CFG,识别出图中的危险边缘,从测试套件中选择能够测试危险边缘的测试用例.该方法选择的测试用例可以覆盖业务改变.Tarhini 等人^[69]提出了另一种基于模型的回归测试技术.该技术利用时间标记转换系统(TLTS)模型化 Web 服务组合,该模型不仅描述 BPEL 内部的活动,还包含原子服务的内部行为;然后进行修改识别,可以从 TLSL 中识别出组合服务中的 3 类修改.

- (1) 将新的服务增加到系统中;

- (2) 将现有服务的内部功能进行了修改;
- (3) 对描述文档进行了修改.

识别出修改部分后,选择可以覆盖该部分的测试用例.该方法选择的测试用例可以覆盖接口改变与绑定改变.Li 等人^[43]基于扩展的 BPEL 流图(XBFG)来进行测试用例选择.该模型不仅能够描述 BPEL 的流程,还能借助消息序列来表达 BPEL process 与合作服务之间的交互过程.基于该模型,Li 等人首先分别计算出新旧程序的控制流路径;然后利用符号定义的方法分别产生新旧版本程序的消息序列;最后比较新旧程序的控制流路径和消息序列,获得修改部分,根据该部分选择测试用例,以及确定是否需要派生新的测试用例.但文中没有叙述测试用例是如何产生的,并且该方法假定系统无论何时都需要进行全面的测试,这是耗时和多余的.

基于路径的 TCS 使用全路径(all-paths)的覆盖准则,然而该准则并不能暴露所有的错误.因为为了预防路径数量出现空间爆炸,在使用该准则时,需将循环结构非循环化,这就导致在测试中很难覆盖所有的路径.程序的本质是获得给定输入的期望输出,输入和输出之间的关系是通过一系列定义和变量的使用来实现的.从本质上看,控制流的设计是为实现正确的数据流.因此,进行数据流测试是非常重要的.所以, Ji 等人^[70]提出了基于数据流的测试用例选择方法.

数据流测试即通过变量的定义-使用关系得到一组测试策略用于衡量功能性测试的覆盖率.典型的数据流测试策略包括:

- (1) All-definition:每个变量的每一个定义值至少使用 1 次;
- (2) All-Uses:要求在测试中每个定义-使用(def-use)对都得到覆盖,其中一个 def-use 对 (x, n, n') 可定义为在语句 n 中定义的变量 x 在语句 n' 中使用.

策略 All-Uses 的覆盖标准比策略 All-definition 要强,能够更好地达到数据流测试的要求,因此, Ji 等人选择 All-Uses 策略作为数据流测试的覆盖准则.他们所提出的方法为新、旧两个版本源程序建立扩展的控制流图(XCFG),通过比较新、旧程序的定义-使用对和 XCFG 模型来获取受演化影响的 def-use 对;然后,根据当前程序的 XCFG 模型寻找能够覆盖受演化影响的 def-use 对的路径,从原始测试套件中选择包含这些路径的测试用例.该方法可以检测到 3 种演化:业务改变、接口改变和绑定改变.

5.2.2 基于语义的 TCS

BPEL 应用程序中,活动的行为是在执行过程中在该点出现的值的序列,程序的语义能够刻画活动的行为.因此, Wang 等人^[72]提出对于一些语法存在差异但语义相同的活动,就没有必要重新测试,这样,测试用例数量将显著减少.所以,他们基于语义分析来捕获新旧程序之间的行为差异部分,再基于差异的部分选择测试用例.他们的工作主要分为 4 个步骤.

- (1) 基于 3 个规则(死亡路径删除(DPE)的特性、交流控制、多赋值)将 BPEL 程序转换为控制流图,在转换过程中,将 BPEL 程序中的死亡路径、同步调用和单 assign 活动多赋值这些隐式结构转换为条件选择、异步调用和单 assign 活动单赋值的显式结构.
- (2) 根据控制流图进行数据依赖、控制依赖和异步调用依赖关系分析,基于得到的依赖关系构建程序依赖图.
- (3) 基于该程序依赖图,可以捕获活动之间的 3 种依赖信息(控制依赖、数据依赖、异步调用依赖).
- (4) 最后,通过捕获到的依赖信息,可以利用程序切片的方法识别出被影响的组件,选择对应的测试用例去重新执行.

实验结果表明,该方法能够有效减少回归测试用例的数量,具有很高的优化性.但也存在两个弊端:一是程序依赖图的分析 and 比较增加了工作负载,所以需要寻找更高效的算法来权衡效率和效益;二是方法中只覆盖了基本的结构和元素,(scope)结构没有完全考虑.

5.3 测试用例优先排序

由于回归测试中受测试资源的限制与制约,不是所有的测试用例都能被执行.在这种情况下,测试用例的执行顺序就显得特别重要,因此,衍生出测试用例的优先排序技术(TCP).TCP 技术采用不同的策略对测试用例的

执行顺序进行优化排序,以期较早地运行能够发现错误的测试用例,为错误修复与定位节省更多的资源.在本节开始部分,我们已经阐述在传统的软件回归测试中,可以采用贪婪法、机器学习法和融合专家知识法来实现 TCP,但在 Web 服务组合 TCP 的现有研究中,大部分都是基于贪婪法进行的^[46-63],基于机器学习法和融合专家知识法的研究几乎没有.所以可以认为,Web 服务组合 TCP 技术的研究还处于初级阶段.

Web 服务组合中的 TCP 大多数都是基于覆盖的,其中,结构覆盖被广泛地研究.该方法的主要目标是尽可能地覆盖所有的程序单元,以期更快地发现所有的错误.覆盖粒度主要包括 branch-total,branch-additional,statement-total,statement-additional,Fault Exposing Potential(FEP)-total 和 FEP-additional.实验结果表明:覆盖粒度对优先排序算法有重要的影响,基于粗粒度覆盖的优先排序算法的错误发现效率较低.根据是否引入反馈机制,可以将覆盖策略分为基于总(total)覆盖还是附加(additional)覆盖:Total 覆盖策略直接计算出各个测试用例的程序实体覆盖率,并从高到低进行排序;Additional 覆盖策略则引入反馈机制,从整体角度出发,当一部分程序实体被已执行测试用例覆盖后,对剩余测试用例进行排序时,则并不需要考虑对上述程序实体的覆盖.然而,无论是基于 total 覆盖还是 additional 覆盖,都是通过贪婪算法进行优先排序的,容易产生局部最优解.

针对 Web 服务组合的 TCP 研究最早始于 2009 年.Mei 等人^[47]发现,传统的测试用例优先排序技术并不适用于 Web 服务组合,就针对 SOA 架构的特征,提出了面向服务的业务应用程序的测试用例优先排序技术.他们提出了一个多层次的覆盖模型.该模型可以捕获测试用例的业务流程覆盖、XPath 覆盖和 WSDL 覆盖信息,并将这些覆盖信息分为 3 层:第 1 层为业务流程覆盖,第 2 层为业务流程和 XPath 覆盖,第 3 层为业务流程、XPath 和 WSDL 覆盖.基于该多层次的覆盖模型,进一步提出适用于 Web 服务组合的 Total 和 Addtl 优先排序技术.

基于多层次覆盖模型的优先排序技术填补了 Web 服务组合 TCP 研究中的一个空白,但是该技术仍然存在很多不足:该技术为一般的优先排序技术,即对所有的测试用例都进行了优先排序,可见,效率是比较低的.此外,该多层次覆盖模型所捕获的覆盖信息是否充足?针对这些问题,Mei 等人^[50]提出了一种包含层次的优先排序技术.他们认为:修改过的 XPath 表达式会提取错误的 XML 信息,需要将 Xpath 的寻找关系考虑进测试用例的排列中.他们在多层次覆盖模型的基础上提出了 ROLE 策略,该策略能够根据保留的包容关系探索不同层次的细节.实验结果表明,包含层次的优先排序技术能够有效弥补基于多层次覆盖模型的 TCP 的执行效率低、捕获的覆盖信息信息不充足的缺陷.

然而,上述研究都只针对回归测试运行之前的组合服务演化情况,而忽略了在回归测试运行时,组合服务同样会存在演化的问题.

因此,Mei 等人^[51,52]进而提出了抢占式的优先排序技术 PRT.该技术能够在当前回归测试运行时,检测到被测服务的演化.当演化发生,并且当前测试用例变为无效时,PRT 就在未执行的测试用例集中选择测试路径相同的测试用例代替当前测试用例,同时产生新的测试子进程抢占当前测试进程,利用替换的测试用例对改变的服务进行测试,该过程称为 fix.当子进程执行的测试任务结束时,原先的测试进程就在暂停点恢复执行,继续执行之后的测试任务.如果整个测试套件已经执行完毕,且两个测试用例之间没有子进程发生,则终止 PRT.

PRT 的抢占式有效地解决了测试过程中出现服务演化的问题,相对于其他 TCP 技术,检错率得到极大的提高.但该方法也存在缺陷,即只考虑了路径覆盖信息,这将对最终的检错率存在一定的影响.选择更加充分的覆盖信息来评估测试用例的优先级,或者根据修改的部分创建新的测试用例对演化的服务进行 fix 能否获得更高的检错率,是一个有待研究的问题.

Wang 等人^[53]认为:在 Web 服务组合 TCP 的研究中,不仅需要考虑覆盖信息,也需要考虑内部结构信息.Web 服务组合的内部结构可以看作是为实现预期目标的活动之间的交互,包括执行过程逻辑、交换消息、调用外部的 Web 服务.这些交互可以通过构建活动之间的依赖关系来分析.一项活动的修改或错误肯定会传播到其他直接或间接依赖它的活动中,因此,可以利用活动的修改影响度来衡量活动的测试必要性.他们通过构建 BPEL 活动依赖图来表示活动之间的控制依赖、数据依赖、异步调用依赖、相关依赖和同步调用依赖关系.基于该依赖图,计算每个活动的修改影响度,某一个活动的修改影响度越高,说明该活动的优先级越高,然后根据优先级对相应的测试用例进行排序.与基于覆盖的方法的比较实验显示,该方法能够实现更高的检错率.

6 测试执行与度量

本节首先总结测试执行阶段常用的评测程序,随后分析缺陷植入类型对研究结论最后的影响,最后总结常用的评测指标。

6.1 评测程序

本节将现有研究中采用的来自实验室或开源 Web 服务组合平台中的评测 Web 服务组合进行了搜集和汇总,结果见表 5。表 5 依次罗列了程序名称、程序来源、程序规模、程序特征描述和累计使用次数。在表 5 中,我们将评测程序按照累计使用次数从高到低进行排序。从表 5 可以看出,使用最多的评测程序是由 BPEL 引擎提供的 LoanApproval 和 ATM,程序规模在 100 行~200 行之间,分别与 3 个组件服务交互。此外,表 5 中总结了两个实际应用中的 Web 服务组合(real-world case):CityGuide 和 eBayfinder,程序规模都在 2 000 行以上,组件服务的数量都由用户需求决定;而其他 21 个评测服务都是来自不同的 BPEL 引擎,这些组合服务的程序规模和组件服务数量远远小于 real-world case,可以将该类评测服务称为合成服务(synthetic service)。然而现有研究中,只有几篇文章将 CityGuide 或 eBayfinder 作为评测程序,超过 70%以上的文章都使用 synthetic service 作为评测服务。这揭示了 Web 服务组合研究的重要问题之一:缺少 real-world case 来验证测试技术的有效性。

Table 5 Evaluation program used in existing studies

表 5 现有研究中使用的评测程序

Name	Source	Size	Description	Number
LoanApproval	ActiveBPEL	125	Interact with 3 Web services	16
ATM	ActiveBPEL	180	Interact with 3 Web services	12
MarketPlace	IBM BPWS4J	68	Interact with 2 Web services	11
GymLocker	IBM BPWS4J	52	Interact with 1 Web service	10
purchase	OracleBPEL	125	Interact with 2 Web services	7
triphandling	BPEL specification	170	Interact with 4 Web services	7
buybook	Oracle BPEL	532	Interact with 3 Web services	6
dslservice	ApacheODE	123	Interact with 3 Web services	6
Travel agent	Oracle BPEL	90	Interact with 2 Web services	3
Loan	Oracle BPEL	147	Interact with 4 Web services	3
Auction	Oracle BPEL	138	-	2
RiskAssessment	Oracle BPEL	118	-	2
CityGuide	Real-World case	3 289	Location-Based service composition, it interacts with Web services based on user's demands	2
Shipping service	IBM BPEL4WS	100~200	Description of the basic shipping service for processing orders	2
SupplyChain	Apache ODE	-	Interact with 3 Web services	1
SmartShelf	Apache ODE	-	Interact with 14 Web services	1
eBayfinder	Nguyen ^[60] 开发 real-world case	20 000	It allows users to search for items listed on eBay ⁷ , to filter search results	1
SquaresSum	Oracle BPEL	69	-	1
TacService	Oracle BPEL	57	-	1
CalculatorProj	Oracle BPEL	190	-	1
LargeProcesses	Oracle BPEL	129	-	1
TripHandling	Oracle BPEL	170	Interact with 4 Web services	1
BPEL1-5	Apache ODE	50	-	1

6.2 错误植入

选择评测程序之后,需要对评测程序进行错误植入,以便对测试用例的有效性进行评估。部分研究在评估 Web 服务组合测试技术有效性时,一般采用手工植入的缺陷进行评估,并假设这些缺陷与实际开发中产生的缺陷性质相似。但也有部分研究利用突变测试技术来产生变异缺陷,如此产生的缺陷与实际缺陷更为相似,如文献 [17,19-21]。

突变测试技术的基本原理是:对源程序中的某些语句进行一次性变换,即进行错误植入,形成 Web 服务组合的错误版本,然后输入测试序列进行测试,可以根据观察测试序列能否检测到程序中的变更来判断测试序列集

的充分性.程序的变异过程可以通过变异算子来实现,每个变异算子对应一种错误类型.越高质量的突变,越能暴露测试套件的缺陷.

Ni 等人^[17]在实验部分利用突变测试技术进行错误植入,他们在实验部分选择 6 种突变产生器:ERR,AIE,ACI,ASI,ASF 和 ER,用 A~F 表示.利用这些产生器为 WS-BPEL 源程序产生 34 种错误种子,将这些错误种子植入到源程序中,得到可以进行测试的变异程序.错误种子在 WS-BPEL,WSDL 和 XPath 文档中的分布见表 6.

Table 6 An instance from Ref.[17]

表 6 来自文献[17]中的一个实例

Ref	A	B	C	D	E	F	SUM
WS-BPEL	1	3	2	4	3	0	13
WSDL	0	1	1	1	6	2	11
XPath	3	0	1	2	1	3	10
Total	4	4	4	7	10	5	34

6.3 评测指标

Web 服务组合测试领域所用到的评测指标可以分为两类:一类是测试用例生成技术的评测指标,另一类回归测试的评测指标.经过调查总结,现有研究通常选择测试用例生成数量、约束条件数量、测试用例的覆盖率、测试用例生成时间以及测试用例的检错率 *FAIL* 这些条件中的 1 个或几个作为测试用例生成技术的评测指标.

错误检错率 *FAIL* 定义为能够暴露错误的测试套件占总测试条件数量的比例,计算公式为

$$FAIL = \frac{\text{Number of FAIL verdicts}}{\text{Number of total instances}} \times 100\% \quad (1)$$

Zhou 等人^[35]使用生成的测试用例数量、约束条件数量以及约束文档求解时间来对提出的测试用例生成方法进行评价,从而说明测试用例的充分性以及开发工具的实用性.他们选择 6 个程序进行评测,评测结果见表 7.在表 7 中,Solving time 表示测试用例生成时间,在文献[35]中即为约束文档求解时间;*TC Num* 表示覆盖率达到最大值时已生成测试用例的数量;*CNN*,*CEN* 分别表示覆盖的节点数量和覆盖的路径的数量;*MC\DC Num* 表示覆盖的独立条件\判断语句的数量.*TC Num*,*CNN* 和 *CEN* 的值越大,说明当前执行测试的测试用例程序体覆盖率越大,测试用例的充分性也越高;Solving time 越小,则说明开发工具的实用性越高.

Table 7 An instance from Ref.[35]

表 7 来自文献[35]中的一个实例

Composite Web service	BPEL LOC	Node coverage criterion			Edge coverage criterion			MC/DC coverage criterion		
		<i>TC Num</i>	<i>CNN</i>	Solving time	<i>TC Num</i>	<i>CNN</i>	Solving time	<i>TC Num</i>	<i>CNN</i>	Solving time
SquaresSum	69	1	4	131	1	7	13	2	2	262
SquaresSum (in 4 loops)	69	1	4	127	1	7	1 127	-	-	-
SquaresSum_1	117	1	7	123	1	11	123	2	2	245
TacService	57	1	4	107	1	5	107	2	2	207
Loan approval service	117	3	6	369	3	15	369	3	3	369
CaculatorProj	190	3	5	393	3	12	393	3	3	393
LargeProcess	129	1	10	168	1	15	168	4	4	636

不同的是,在回归测试研究领域,研究者选择错误检测平均累计比例 APFD、平均相对位置 RP 和故障检测的调和平均数 HMFD 这些指标中的 1 个或几个作为 TCP 的评测指标;一般从两个方面来评价测试套件约简技术(TSM)的效率——测试套件规模减小的百分比和错误发现减少的百分比,如文献[65];将最终选择出的测试子集的规模大小和错误暴露比例作为 TCS 的评测指标,如文献[69,70].

文献[50,51]同时将 APFD,RP 和 HMFD 作为 TCP 技术有效性的评测指标.评测指标 APFD,RP 和 HMFD 的取值均在 0~1 之间.APFD 取值越高,则缺陷检测速度越快;RP 和 HMFD 的取值越低,则意味着缺陷检测速度越快.假设有测试用例集 *T*,其中包含 *n* 个测试用例,*m* 个缺陷,*F* 为 *T* 检测到的缺陷集合,给定一个测试用例执行次序,其中,*TF_i* 表示首个可检测到第 *i* 个缺陷的测试用例在该执行次序中所处次序,则 APFD 的计算公式为

$$APFD = 1 - \frac{TF_1 + TF_2 + \dots + TF_m}{nm} + \frac{1}{2n} \quad (2)$$

HMFD 的计算公式为

$$HMFD = \frac{m}{\frac{1}{TF_1} + \frac{1}{TF_2} + \dots + \frac{1}{TF_m}} \quad (3)$$

$P(TF_i, i)$ 为缺陷 i 导致的第 1 个测试失败的测试用例所在位置为 TF_i 的可能性, 缺陷 i 的值可计算为 $RP(i)$:

$$RP(i) = \frac{\sum_{TF_i=1}^n TF_i \cdot P(TF_i, i)}{n} \quad (4)$$

7 总结与展望

根据上述分析可以看出, Web 服务组合测试问题已经得到学术界和工业界的广泛关注, 并获得了阶段性的成果. 本文从 Web 服务组合的特性出发, 提出 Web 服务组合测试存在的挑战和 Web 服务组合测试与传统测试的区别, 并对 Web 服务组合测试的已有研究成果、评测程序和评测指标进行了系统的总结和分析.

从上述内容的介绍中可以看出, 测试用例生成、测试结果评估和回归测试问题是 Web 服务组合测试的重要研究问题. 虽然已有研究取得了一定的成果, 但是在这些领域内还存在一些进一步研究的问题. 主要表现在以下几个方面.

(1) 高检错率、高覆盖量、低冗余度的测试用例

测试用例质量是执行有效的 Web 服务组合测试的根本条件, 测试用例的高质量表现在高检错率、低冗余和高覆盖. 已有研究工作为提高 Web 服务组合测试的高效性提出了许多测试用例生成技术和方法, 以获得高质量的测试用例. 从研究成果分析, 现有技术生成的测试用例质量不断得到提高, 但仍有进一步提高的空间. 如何获取更多能够丰富测试信息的约束条件来生成测试用例, 是当今学术界和工业界仍需进一步解决的问题.

(2) 自动化的测试工具

Sun 等人^[21]实现了能够自动化进行 Web 服务组合测试的工具. 该工具能够自动化产生测试用例、自动化执行测试, 是实现 Web 服务组合测试自动化的领先者. 但该工具没有实现测试流程的整体自动化且存在一些性能缺陷, 表现为: 测试结果验证与评估需人工完成; Sun 等人没有对该工具的执行时间进行评估, 因此无法了解该工具的性能和效率; 该工具自动化执行测试的评测程序为 BPEL 引擎提供的合成 Web 服务组合, 该服务规模较小. 因此, 该工具是否适用于复杂度更高的系统还有待进一步研究.

(3) 实际应用中的组合服务作为评测程序

目前, 已有的 Web 服务组合测试技术一般采用对照实验方式进行有效性评估, 从第 5.1 节可以得出结论: 已有测试技术采用的评测服务绝大部分为 BPEL 引擎提供的合成组合服务, 只有极少部分研究者选择实际应用中的 Web 服务组合作为评测程序, 合成组合服务的程序规模较小, 选择该组合服务作为评测程序所得的结论不一定适用于来自实际应用中的大型 Web 服务组合. 若要将学术界的 Web 服务组合测试技术应用到实际应用中, 则需要进一步提高测试工具的自动化程度, 并保证工具具有一定的鲁棒性. 若对上述问题提出有效的解决方案, 则将有效提高研究成果的应用价值, 并大幅度提高实际应用的 Web 服务组合测试效率.

(4) 运行时绑定

部分研究者已经关注 Web 服务组合的运行时绑定问题, 如 Sun 等人^[21]、Ni 等人^[17]、和 Wu 等人^[18]分别从场景、消息序列和服务内部状态模型化这 3 个方面实现测试用例生成技术. Sun 等人认为, 测试用例生成过程的自动化程度是决定运行时绑定问题解决与否的关键, 因此他们设计了测试自动化的原型工具, 而 Ni 等人和 Wu 等人没有阐述测试技术的自动化程度. 但遗憾的是, Sun 等人没有就自动化程度和运行时绑定问题之间的联系进行实验性的验证, 因此无法确定他们所提出的技术是否真正能够解决运行时绑定问题.

(5) 回归测试动态演化

解决动态演化问题,是 Web 服务组合回归测试技术的关键所在.已有研究不断对该问题进行深入研究,从开始着眼于解决回归测试之前的动态演化问题,到提出解决回归测试过程中的动态演化问题,已有研究已经取得一定的成果.但这仅限于 Web 服务组合的 TCP 技术研究领域,在测试用例选择、测试用例修复等其他回归测试技术研究领域处于空白期.

(6) 尝试新 TCP 测试方法

在第 6.3 节,我们对 Web 服务组合已有的 TCP 技术进行了系统的总结和分析,可以看到,现有的 TCP 技术都基于贪婪法进行,并取得了一定的研究成果.除了对基于贪婪法的 TCP 技术展开更为深入的研究之外,我们还可以积极寻找可以行之有效的 TCP 技术方法,如传统软件 TCP 研究中的机器学习法和专家融合知识法.

References:

- [1] Hong Z, Feng ZY. Collaborative testing of Web services. *IEEE Trans. on Service Computing*, 2012,5(1):116–130. [doi: 10.1109/TSC.2010.54]
- [2] Bozkurt M, Harman M, Hassoun Y. Testing and verification in service-oriented architecture: A survey. *Software Testing, Verification and Reliability*, 2013,23(4):261–313. [doi: 10.1002/stvr.1470]
- [3] Ebrahim SM. A survey of service-oriented architecture systems testing. *Int'l Journal of Software Engineering & Applications (IJSEA)*, 2012,3(6):19–27. [doi: 10.5121/ijsea.2012.3602]
- [4] Rusli HM, Puteg M, Ibrahim S, Tabatabaei S. A comparative evaluation of state-of-the-art Web service composition testing approaches. In: *Proc. of the 6th Int'l Workshop on Automation of Software Test (AST)*. 2011. 29–35. [doi: 10.1145/1982595.1982602]
- [5] Huang H, Tsai WT, Paul R. Automated model checking and testing for composite Web services. In: *Proc. of the IEEE Int'l Symp. on Object-Oriented Real-time Distributed Computing*. 2005. 300–307. [doi: 10.1109/ISORC.2005.16]
- [6] Garcia-Fanjul J, de la Riva C, Tuya J. Generating test cases specifications for BPEL compositions of Web services using SPIN. In: *Proc. of the Int'l Workshop on Web Services—Modeling and Testing*. 2006. 83–94.
- [7] Garcia-Fanjul J, de La Riva C, Tuya J. Generation of conformance test suites for compositions of Web services using model checking. In: *Proc. of the Testing: Academic and Industrial Conf.—Practice and Research Techniques*. 2006. 127–130. [doi: 10.1109/TAIC-PART.2006.11]
- [8] Zheng YY, Zhou J, Krause P. A model checking based test case generation framework for Web services. In: *Proc. of the Int'l Conf. on Information Technology*. 2007. 715–722. [doi: 10.1109/ITNG.2007.8]
- [9] Li B, Zhang WS, Zhang XG. Describing and verifying Web service using CCS. In: *Proc. of the Int'l Conf. on Parallel and Distributed Computing*. 2006. 1571–1576. [doi: 10.1109/PDCAT.2006.52]
- [10] Long HY, Ma D. Checking compatibility of BPEL4WS based on CCS. In: *Proc. of the Int'l Conf. on System Science, Engineering Design and Manufacturing Informatization*. 2011. 255–258. [doi: 10.1109/ICSSEM.2011.6081199]
- [11] Du YH, Tan W, Zhou MC. Timed compatibility analysis of Web service composition a modular approach based on Petri nets. *IEEE Trans. on Automation Science and Engineering*, 2014,11(2):594–606. [doi: 10.1109/TASE.2013.2253552]
- [12] Yuan Y, Li Z, Sun W. A graph-search based approach to BPEL4WS test generation. In: *Proc. of the Int'l Conf. on Software Engineering Advances (ICSEA)*. 2006. 1–14. [doi: 10.1109/ICSEA.2006.261270]
- [13] Yan J, Li Z, Yuan Y, Sun W, Zhang J. BPEL4WS unit testing: Test case generation using a concurrent path analysis approach. In: *Proc. of the 17th Int'l Symp. on Software Reliability Engineering (ISSRE)*. 2006. 75–84. [doi: 10.1109/ISSRE.2006.16]
- [14] Mei L, Chan WK, Tse TH. Data flow testing of service choreography. In: *Proc. of the 7th Joint Meeting of the European Software Engineering Conf. and the ACM SIGSOFT Int'l Symp. on Foundations of Software Engineering (ESEC)*. 2009. 151–160. [doi: 10.1145/1595696.1595720]
- [15] Mei L, Chan WK, Tse TH. Data flow testing of service oriented workflow applications. In: *Proc. of the 30th Int'l Conf. on Software Engineering (ICSE)*. 2008. 371–380. [doi: 10.1145/1368088.1368139]
- [16] Hou SS, Zhang L, Lan Q, Mei HJ, Sun S. Generating effective test sequences for BPEL testing. In: *Proc. of the 5th Int'l Conf. on Quality Software*. 2009. 331–340. [doi: 10.1109/QSIC.2009.50]
- [17] Ni Y, Hou SS, Zhang L, Zhu J, Li ZJ, Lan Q. Effective message-sequence generation or testing BPEL programs. *IEEE Trans. on Service Computing*, 2013,6(1):7–19. [doi: 10.1109/TSC.2011.22]

- [18] Wu CS, Huang CH. The Web services composition testing based on extended finite state machine and UML model. In: Proc. of the IEEE Int'l Conf. on Service Science and Innovation. 2013. 215–222. [doi: 10.1109/ICSSI.2013.46]
- [19] Sun CA, Shang Y, Zhao Y, Chen TY. Scenario-Oriented testing for Web service compositions using BPEL. In: Proc. of the 12th Int'l Conf. on Quality Software (QSIC). 2012. 171–174. [doi: 10.1109/QSIC.2012.47]
- [20] Zhang PC, Leung H, Li WR, Li XD. Web services property sequence chart monitor: A tool chain for monitoring BPEL-based Web service composition with scenario-based specifications. IET Software, 2013,7(4):222–248. [doi: 10.1049/iet-sen.2012.0047]
- [21] Sun C, Zhao Y, Pan L, Liu H, Chen TY. Automated testing of WS-BPEL service compositions: A scenario-oriented approach. IEEE Trans. on Services Computing, 2015. [doi: 10.1109/TSC.2015.2466572]
- [22] Zhang G, Rong M, Zhang J. A business process of Web services testing method based on UML 2.0 activity diagram. In: Proc. of the Intelligent Information Technology Application Workshop. 2007. 59–65. [doi: 10.1109/IITA.2007.83]
- [23] BellF, Endo AT, Linschulte M, Simao AS. Model-Based testing of Web service compositions. In: Proc. of the 6th IEEE Int'l Symp. on Service-Oriented System Engineering (SOSE). 2011. 181–192. [doi: 10.1109/SOSE.2011.6139107]
- [24] Endo AT. Model-Based testing of service oriented applications [Ph.D. Thesis]. Sao Paulo: Universidade de São Paulo, 2013.
- [25] Endo AT, Simao A. Model-Based testing of service-oriented applications via state models. In: Proc. of the IEEE Int'l Conf. on Services Computing (SCC). 2011. 432–439. [doi: 10.1109/SCC.2011.77]
- [26] Bernardino M, Rodrigues E, Zorzo A, Marchezan L. An industrial experience on using models to test Web service-oriented applications. In: Proc. of the IET Software. 2017.
- [27] Gao H, Li Y. Generating quantitative test cases for probabilistic timed Web service composition. In: Proc. of the APSCC. 2011. 275–283. [doi: 10.1109/APSCC.2011.13]
- [28] Cavalli AR, Cao TD, Mallouli W, Martins E, Sadovykh A, Salva S, Zaïdi F. WebMov: A dedicated framework for the modelling and testing of Web services composition. In: Proc. of the Int'l Conf. on Web Services Pages. 2010. 377–384. [doi: 10.1109/ICWS.2010.24]
- [29] Zhao H, Sun J, Liu X. A model checking based approach to automatic test suite generation for testing Web services and BPEL. In: Proc. of the IEEE Asia-Pacific Service Computing Conf. 2012. 61–69. [doi: 10.1109/APSCC.2012.29]
- [30] Petrova-Antonova D, Ilieva S, Manova D. Tassa: Testing framework for Web service orchestrations. In: Proc. of the 10th IEEE IACM Int'l Workshop on Automation of Software Test. 2015. 8–12. [doi: 10.1109/AST.2015.9]
- [31] Cao TD, Felix P, Castanet R. WSOTF: An automatic testing tool for Web services composition. In: Proc. of the 5th Int'l Conf. on Internet and Web Applications and Services (ICIW). 2010. 7–12. [doi: 10.1109/ICIW.2010.9]
- [32] Bartolini C, Bertolino A, Marchetti E, Parisis I. Data flow based validation of Web services compositions: Perspectives and examples. In: Proc. of the Architecting Dependable Systems V. LNCS 5135, 2008. 298–325. [doi: 10.1007/978-3-540-85571-2_13]
- [33] Zhang J, Yang R, Chen Z. Automated EFSM based test case generation with scatter search. In: Proc. of the 7th Int'l Workshop on AST. 2012. 76–82. [doi: 10.1109/IWAST.2012.6228994]
- [34] Gu Y, Ge Y. Search-Based performance testing of applications with composite services. In: Proc. of the 2009 Int'l Conf. on Web Information Systems and Mining (WISM). 2009. 320–324. [doi: 10.1109/WISM.2009.73]
- [35] Zhou LB, Xu L, Xu BW, Yang HJ. Generating test cases for composite Web services by parsing XML documents and solving constraints. In: Proc. of the 39th IEEE Annual Int'l Computers, Software & Applications Conf. 2015. 304–309. [doi: 10.1109/COMPSAC.2015.51]
- [36] Estero-Botaro A, Garcia-Dominguez A, Dominguez-Jimenez JJ, Palomo-Lozano F, Medina-Bulo I. A framework for genetic test-case generation for WS-BPEL compositions. In: Proc. of the 26th IFIP WG 6.1 Int'l Conf. on Testing Software and Systems (ICTSS). 2014. 1–16. [doi: 10.1007/978-3-662-44857-1_1]
- [37] Estero-Botaro A, Palomo-Lozano F, Medina-Bulo I. Quantitative evaluation of mutation operators for WS-BPEL compositions. In: Proc. of the 3rd Int'l Conf. on Software Testing, Verification, and Validation Workshops (ICST). 2010. 142–150. [doi: 10.1109/ICSTW.2010.36]
- [38] Dominguez-Jimenez J, Estero-Botaro A, Garcia-Dominguez A, Medina-Bulo I. GAmEra: An automatic mutant generation system for WS-BPEL compositions. In: Proc. of the IEEE Int'l Conf. on Web Services. 2009. 97–106. [doi: 10.1109/ECOWS.2009.18]
- [39] Casado R, Younas M, Tuya J. A generic framework for testing the Web services transactions. In: Proc. of the Advanced Web Service. 2014. 29–49. [doi: 10.1007/978-1-4614-7535-4_2]

- [40] Boonyakulsrirung P, Suwannasart T. A weak mutation testing framework for WS-BPEL. In: Proc. of the 8th Int'l Joint Conf. on Computer Science and Software Engineering (JCSSE). 2011. 313–318. [doi: 10.1109/JCSSE.2011.5930140]
- [41] Blanco R, Garcia-Fanjuland J, Tuya J. A first approach to test case generation for BPEL compositions of Web services using scatter search. In: Proc. of the Int'l Conf. on Software Testing, Verification, and Validation Workshops. 2009. 131–140. [doi: 10.1109/ICSTW.2009.24]
- [42] Blanco R, Fanjul JG, Tuya J. Test case generation for transition-pair coverage using scatter search. *Int'l Journal of Software Engineering and Its Applications*, 2010,4(4):37–55.
- [43] Li B, Qiu D, Leung H, Wang D. Automatic test case selection for regression testing of composite service based on extensible BPEL flow graph. *The Journal of Systems and Software*, 2012,85(6):1300–1324. [doi: 10.1016/j.jss.2012.01.036]
- [44] Zhang ZY, Chen ZY, Xu BW, Yang R. Research progress on test case evolution. *Ruan Jian Xue Bao/Journal of Software*, 2013,24(4):663–674 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4379.htm> [doi: 10.3724/SP.J.1001.2013.04379]
- [45] Chen X, Chen JH, Ju XL, Gu Q. Survey of test case prioritization techniques for regression testing. *Ruan Jian Xue Bao/Journal of Software*, 2013,24(8):1695–1712 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4420.htm> [doi: 10.3724/SP.J.1001.2013.04420]
- [46] Zhang Z, Chan WK, Tse TH. Test case prioritization for regression testing of service-oriented business applications. In: Proc. of the 18th Int'l Conf. on World Wide Web (WWW 2009). 2009. 901–910. [doi: 10.1145/1526709.1526830]
- [47] Mei L, Chan WK, Tse TH, Merkel RG. Tag-Based techniques for black-box test case prioritization for service testing. In: Proc. of the 9th Int'l Conf. on Quality Software (QSIC). 2009. 21–30. [doi: 10.1109/QSIC.2009.12]
- [48] Mei L, Chan WK, Tse TH, Merkel RG. Merkel. XML-manipulating test case prioritization for XML-manipulating services. *Journal of Systems and Software*, 2011,84(4):603–619. [doi: 10.1016/j.jss.2010.11.905]
- [49] Mei L, Cai Y, Jia C, Jiang B, Chan WK. Prioritizing structurally complex test pairs for validating WS-BPEL evolutions. In: Proc. of the Int'l Conf. Web Service (ICWS). 2013. 147–154. [doi: 10.1109/ICWS.2013.29]
- [50] Mei L, Cai Y, Jia C, Jiang B, Zhang Z, Tse YH. A subsumption hierarchy of test case prioritization for composite services. *IEEE Trans. on Computing*, 2015,8(5):658–673. [doi: 10.1109/TSC.2014.2331683]
- [51] Mei L, Zhai K, Jiang B, Chan WK, Tse TH. Preemptive regression test scheduling strategies: A new testing approach to thriving on the volatile service environments. In: Proc. of the IEEE Int'l Conf. Computer Software and Applications. 2012. 901–910. [doi: 10.1109/COMPSAC.2012.17]
- [52] Mei L, Zhang Z, Chan WK, Tse TH. Preemptive regression testing of workflow-based Web services. *IEEE Trans. on Services Computing*, 2015,8(5):740–754. [doi: 10.1109/TSC.2014.2322621]
- [53] Wang H, Xing J, Yang Q, Han D, Zhang X. Modification impact analysis based test case prioritization for regression testing of service-oriented workflow applications. In: Proc. of the 39th IEEE Annual Computer Software and Applications Conf. 2015. 288–297. [doi: 10.1109/COMPSAC.2015.11]
- [54] Maggie H, Katerina GP. Common trends in software fault and failure data. *IEEE Trans. on Software Engineering*, 2009,35(4):484–496. [doi: 10.1109/TSE.2009.3]
- [55] Zhou YM, Leung H. Empirical analysis of object-oriented design metrics for predicting high and low severity faults. *IEEE Trans. on Software Engineering*, 2006,32(10):771–789. [doi: 10.1109/TSE.2006.102]
- [56] Chen L, Wang Z, Xu L, Lu H, Xu B. Test case prioritization for Web service regression testing. In: Proc. of the 5th IEEE Int'l Symp. on Service Oriented System Engineering (SOSE). 2010. 173–178. [doi: 10.1109/SOSE.2010.27]
- [57] Askarunisa A, Abirami AM, Arockia K, Karthik B, Arunkumar R. Sequence-Based techniques for black-box test case prioritization for composite service testing. In: Proc. of the IEEE Int'l Conf. on Computational Intelligence and Computing Research (ICCIC). 2010. 1–4. [doi: 10.1109/ICCIC.2010.5705784]
- [58] Hou S, Zhang L, Xie T, Sun JS. Quota-Constrained test-case prioritization for regression testing of service-centric systems. In: Proc. of the IEEE Int'l Conf. on Software Maintenance (ICSM). 2008. 257–266. [doi: 10.1109/ICSM.2008.4658074]
- [59] Wang D, Li B, Cai J. Regression testing of composite service: An xfbg-based approach. In: Proc. of the IEEE Congress on Services. 2008. 112–119. [doi: 10.1109/SERVICES-2.2008.28]

- [60] Nguyen CD, Marchetto A, Tonella P. Change sensitivity based prioritization for audit testing of Web service compositions. In: Proc. of the 2011 4th IEEE Int'l Conf. on Software Testing, Verification and Validation Workshops. 2011. [doi: 10.1109/ICSTW.2011.50]
- [61] Raju S, Uma GV. Factors oriented test case prioritization technique in regression testing using genetic algorithm. European Journal of Scientific Research, 2012,74(3):389-402.
- [62] Zhai K, Jiang B, Chan WK. Prioritizing test cases for regression testing of location-based services: Metrics, techniques and case study. IEEE Trans. on Services Computing, 2014,7(1):54-67. [doi: 10.1109/TSC.2012.40]
- [63] Harrold MJ, Orso A. Retesting software during development and maintenance. In: Proc. of the Frontiers of Software Maintenance (FoSM). 2008. 99-108. [doi: 10.1109/FOSM.2008.4659253]
- [64] Yoo S, Harman M. Using hybrid algorithm for Pareto efficient multi-objective test suite minimisation. Journal of Systems Software, 2010,83(4):689-701. [doi: 10.1016/j.jss.2009.11.706]
- [65] Bozkurt M. Cost-Aware Pareto optimal test suite minimisation for service-centric systems. In: Proc. of the 15th Annual Conf. on Genetic and Evolutionary Computation. 2013. 1429-1436.
- [66] Tsai WT, Zhou X, Paul R, Chen Y, Bai X. A coverage relationship model for test case selection and ranking for multi-version software. In: Proc. of the 10th IEEE High Assurance Systems Engineering Symp. Dallas: IEEE Computer Society, 2007. 105-112. [doi: 10.1109/HASE.2007.64]
- [67] Ruth M, Tu S. Concurrency issues in automating RTS for Web services. In: Proc. of the IEEE Int'l Conf. on Web Services. Salt Lake City, 2007. 1142-1143. [doi: 10.1109/ICWS.2007.65]
- [68] Ruth M. Concurrency in a decentralized automatic regression test selection framework for Web services. In: Proc. of the 15th ACM Mardi Gras Conf. 2008. 7:1-7:8.
- [69] Tarhini A, Fouchal H, Mansor N. Regression testing Web services-based applications. In: Proc. of the IEEE Int'l Conf. on Computer Systems and Applications. 2006. 163-170. [doi: 10.1109/AICCSA.2006.205085]
- [70] Ji SH, Li BX, Zhang PC. Test case selection for data flow based regression testing of BPEL composite services. In: Proc. of the IEEE Int'l Conf. on Services Computing. 2016. 547-554. [doi: 10.1109/SCC.2016.77]
- [71] Li B, Qiu D, Ji S, Wang D. Automatic test case selection and generation for regression testing of composite service based on extensible BPEL flow graph. In: Proc. of the 26th IEEE Int'l Conf. on Software Maintenance (ICSM). 2010. 1-10. [doi: 10.1109/ICSM.2010.5609541]
- [72] Wang H, Xing J, Yang Q, Wang P, Zhang X. Optimal control based regression test selection for service-oriented workflow applications. Journal of Systems and Software, 2017,124(2017):274-288. [doi: 10.1016/j.jss.2016.06.065]
- [73] Cooray MB, Hamlyn-Harris JH, Merkel RG. Test reconfiguration for service oriented applications. In: Proc. of the IEEE Int'l Conf. on Utility Cloud Computing. 2011. 300-305. [doi: 10.1109/UCC.2011.48]
- [74] Cooray MB, Hamlyn-Harris JH, Merkel RG. Dynamic test reconfiguration for composite Web services. IEEE Trans. on Services Computing, 2015,8(4):576-585. [doi: 10.1109/TSC.2014.2312953]

附中文参考文献:

- [44] 张智轶,陈振宇,徐宝文,杨瑞.测试用例演化研究进展.软件学报,2013,24(4):663-674. <http://www.jos.org.cn/1000-9825/4379.htm> [doi: 10.3724/SP.J.1001.2013.04379]
- [45] 陈翔,陈继红,鞠小林,顾庆.回归测试中的测试用例优先排序技术述评.软件学报,2013,24(8):1695-1712. <http://www.jos.org.cn/1000-9825/4420.htm> [doi: 10.3724/SP.J.1001.2013.04420]



丁志军(1974-),男,江苏泰兴人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为服务计算,并发系统,Petri 网理论,移动互联网.



周泽霞(1994-),女,硕士生,主要研究领域为嵌入式系统与服务计算.