

北京科技大学 硕士学位研究生 选题报告及文献总结



行为模型驱动的服务组合程序 测试用例生成技术研究

指导教师： 孙昌爱

单 位： 计算机与通信工程学院

学 号： S20150750

作 者： 贾婧婷

专业名称： 软件工程

入学时间： 2015 年 9 月 1 日

2016 年 9 月 2 日

目 录

1. 选题的背景和意义.....3

 1.1 选题的背景.....3

 1.2 选题的意义.....4

2.研究现状.....5

 2.1 国内外研究现状.....5

 2.2 本课题组研究现状.....6

3. 研究内容.....7

 3.1 基于服务行为的形式化描述模型的建立.....8

 3.2 基于行为描述模型的测试序列与测试数据自动生成.....9

 3.2.1 测试序列集的获取.....9

 3.2.2 测试数据集的生成.....10

 3.2.3 测试用例集的生成.....11

 3.3 基于行为描述模型的测试用例自动生成工具.....11

 3.4 实例验证.....12

 3.5 研究难点.....13

4. 研究工作进度安排.....14

参考文献.....14

1. 选题的背景和意义

1.1 选题的背景

随着软件系统的规模和功能不断增长,各种应用程序在不同域的相互作用也越来越依赖于彼此,它们必须能够相互调整,以适应迅速变化的市场条件。近几年以面向服务的体系架构 SOA (Service-Oriented Architecture) 为基础的 Web 服务逐渐受到重视,被认为是解决异构系统集成问题、快速实现企业流程的有效方案。

面向服务的架构是一种新兴的旨在促进松散耦合的应用的方法[1],它定义了一种松散的基于标准的面向服务的应用程序开发范式[2]。面向服务的架构以服务为基本构件,将软件资源与应用被封装成服务,提供者将开发的服务发布到注册中心,供使用者通过已发布的接口使用服务。SOA 架构可以支持快速的业务重整与优化、较好的解决分布、动态、异构环境下,数据、应用和系统集成等问题。目前,大量银行系统、各种行业供应链系统、航空服务系统、电子商务系统等已经接纳了面向服务的架构。

Web 服务是 SOA 概念的一种典型的实现方式,采用 XML 标准,使用简单对象访问协议(SOAP)进行通信、网络服务描述语言(WSDL)进行服务描述、UDDI 进行服务的注册和搜索[3]。许多软件资源与应用被封装成服务,服务以功能模块的方式对外发布,对外提供统一的调用接口,而屏蔽服务的实现细节[4]。这些技术及标准有效地屏蔽了运行环境的异构性,使得 Web 服务可以直接部署和运行在网络上。对于复杂多变的企业级业务需求而言,单一的 Web 服务往往无法满足实际需求,需要将多个服务协调组织起来以支持复杂应用,这些服务可以通过定义一个 workflow 将把多个 Web 服务组织起来,这个过程被称为 Web 服务组合(组装)[5]。

由于 Web 服务及其协同的动态性,开放多变的互联网运行环境,以及松耦合的 Web 服务开发模式所导致的开发和运行过程不确定,使得 Web 服务的正确性、可靠性、安全性、可用性、时效性等可信性质难以得到保证,因此,如何保证单个 Web 服务及多服务组合的可靠性是一个亟待解决的关键问题。

1.2 选题的意义

软件测试是一种广泛使用的提高软件可信性的方法。SOA 软件的新特点为软件测试提出了新的挑战[6-7]。

例如，对用户和系统集成商，服务只是提供了接口，实现部分是透明的，这阻碍了白盒测试技术的应用[6]；其次，服务的 WSDL 文档里信息有限，只包含参数数据类型的信息，依据这些信息，很难生成有效的测试用例[8]。同时，对 Web 服务组合程序来说，其正确性不仅仅取决于组合流程，也取决于各个组成它的服务的正确性[9]，软件组合的互操作性和集成性，网络状态等问题，使得即使单个服务已经进行了测试，也要在服务组合后进行进一步的测试，这给服务组合程序的测试引入了新的问题；最后，需求的快速变化导致参与组合的服务面临经常性的修改，服务的接口与实现之间可能潜藏不一致性。

分布式、缺乏控制与可监测性、与其他应用程序的动态整合、XML 标准的使用等是这种新类型软件的主要特点。如何保证 Web 服务及其组合的可靠性是一个亟待解决的关键问题。

在这种情况下，软件测试是保证服务质量和可靠性的必要手段。然而，由于 SOA 软件的上述特点，使得已有的基于程序的测试方法很难直接用来测试 Web 服务及其服务组合程序。

任何类型应用程序测试中的一个常见问题是自动获取有意义的测试用例。现有研究提出将模型驱动测试技术应用于 Web 服务。使用模型生成测试用例的策略称为基于模型的测试（MBT），模型驱动测试技术关注待开发系统的重要属性及约束，通过采用形式化或半形式化语言对复杂软件进行抽象与建模，依据建模进行测试的设计，具有高故障检测率，自动化程度高，能够更好的适应需求的演变等特点，能够较好的适应 Web 服务的测试需求，因此越来越多的研究人员从模型角度出发对 Web 服务及其组合程序进行测试[10]。

2.研究现状

2.1 国内外研究现状

目前，基于模型的测试技术逐渐得到关注，并出现了大量针对 Web 服务及其组合程序的测试模型建模方法。基于模型的软件测试思想来自于硬件测试，它被广泛的应用到电信交换系统的测试中，当被应用到软件测试时，其过程是首先构建待测试软件的模型及其派生模型（一般称作测试模型），然后从模型中生成需要的测试用例，在待测程序上执行该测试用例得到测试结果[11]。

国内外现有研究主要关注 Web 服务执行时的行为及其内部状态，首先根据 Web 服务或其组合程序的行为建立有限状态机[12-18]、事件序列图[3,9,19,20]等模型，或者结合 UML 建模技术[11,21-25]进行模型的建立，然后采取一定的覆盖准则，生成测试路径。

文献[14]提出了一种基于有限状态机 FSM[26]的 Web 服务建模测试方法及支持工具，该方法专注于服务测试场景中的状态及转移。有限状态机由有限个状态及状态之间的迁移组成，转换条件对应 Web 服务所执行的操作，即服务的请求及响应；状态对应待测程序目前所处状态。但该方法并未考虑各个状态之间的转移条件（输入、输出数据及程序内部变量值）。

Keum[15]和 Kalaji[16]等人提出了一种基于扩展的有限状态机（EFSM）进行 Web 服务的建模与测试方法，在传统有限状态机模型中添加状态转换的约束（操作输入输出参数要求，变迁的前置条件等）。通过扩展的 WSDL 文档（向其中添加服务行为约束）进行模型的建立，利用建立的模型生成测试用例。由于 EFSM 在 FSM 基础上添加了变量约束，因此它可以对待测服务的控制流和数据流进行建模，但该方法并未考虑服务组合程序的测试。

文献[17]针对如何实现服务组合程序的足够的测试覆盖率的问题提出了面向服务组合的分层有限状态机模型。

有限状态机可以比较精确的刻画软件系统的行为，因此被广泛应用于很多领域模型系统的建立，包括时序电路、程序验证、网络交互协议等。但是有限状态机和状态转换模型不能很好的体现 Web 服务的动态交互，描述具有典型并发特

征的工作流系统的能力较弱，也不利于系统的负面测试。

文献[19]提出了一种基于事件序列图的 Web 服务测试方法 (ESG4WS)，测试 Web 服务的功能行为，事件序列图模型是无状态的，他们不关注软件组件的内部状态，而是专注于事件[27]。事件序列图由节点及边组成，节点代表服务的请求或响应事件，边代表这些事件的序列，该方法能够有效地进行正面及负面测试。

文献[3]在其基础上添加了基于覆盖的测试方法，实现了控制流和数据流测试覆盖标准，但该研究仅针对单个 Web 服务且要求被测服务开放源码。

文献[9,20]在文献[19]的基础上，针对服务组合程序进行事件序列模型的建立及测试用例的自动生成，开发了相应的支持工具。

虽然上述模型描述了系统交互过程，但它忽略了测试中的系统的内部状态。

也有研究方向考虑结合统一建模语言进行模型的建立，统一建模语言 (unified modeling language, 简称 UML) 是一种通用的可视化面向对象建模语言，它提供了多种图元，可以从不同视角和层次描述复杂软件系统的静态结构和动态特性，其定义良好、易于表达，已广泛应用于各种领域[28]。

文献[21]提出了一种基于扩展有限状态机及 UML 顺序图相结合的服务组合程序测试用例生成框架 (EFSM-SeTM) 并定义了 5 种覆盖准则 (全路径覆盖、状态覆盖、转移覆盖、消息覆盖及谓词覆盖)，EFSM-SeTM 使用扩展的有限状态机对单个 Web 服务进行建模、UML 顺序图描述服务组合的信息传递顺序。但该工作对测试用例生成的具体实现没有做详细介绍。

文献[11, 22]提出了基于 UML2.0 活动图和通信图构建 Web 服务测试模型进行 Web 服务测试的方法，该方法扩展了 UML2.0 活动图，使其能够描述 BPEL 的语法元素和行为特性；同时，给出 UML2.0 活动图形式化定义及其测试覆盖准则，对测试用例生成的深度优先搜索算法加以约束，合理地减少了测试用例的数量，提高了测试的效率和精确性。但该方法基于 BPEL 的语法元素和行为特性，在应用到其他 Web 服务组合方式时有其局限性。

2.2 本课题组研究现状

课题组在本课题相关领域进行了大量的前期研究工作与技术积累，提出了面

向 Web 服务及其组合程序的测试方法[29-32]。

特别的。课题组已经尝试将模型驱动测试方法应用于业务流程执行语言 (BPEL) [33] 服务组合程序，提出了一种基于场景的 BPEL 测试用例自动生成技术及支持工具原型的开发，该方法将 BPEL 流程转换为 UNL 活动图模型，通过不同覆盖策略生成测试序列，利用约束求解工具求解出相对应的测试数据，产生面向场景的测试用例集。实验结果表明，模型驱动测试方法可以有效监测服务组装程序中潜藏的故障[34-35]。但该方法重点测试 BPEL 流程出现的常见错误，仅仅考虑针对控制流进行建模。

3. 研究内容

现有研究存在如下不足：

- 1) 主要针对单个服务层面，或是服务组合流程的测试；
- 2) 基于事件序列图的测试模型缺乏对调用服务内部状态的考虑；
- 3) 现有研究多集中在模型驱动测试用例生成框架的研究，缺乏支持所提框架的测试用例生成工具。

因此我们提出从服务组合视角对 Web 服务的行为进行测试，从服务组合上下文环境中服务行为(事件+状态)的角度出发，针对服务执行过程的内部状态，结合国内外出现的基于事件模型的测试用例生成方法，提出一种行为模型驱动的服务组合程序测试用例生成技术，并开发相应支持工具，辅助所提技术的测试用例自动化生成。

基于模型进行服务测试的一般过程如图 3-1 所示，首先构建被测试软件的模型，然后依据特定的遍历算法从模型中导出测试序列集，求解出对应的测试数据，在待测程序上执行该测试用例得到测试结果。

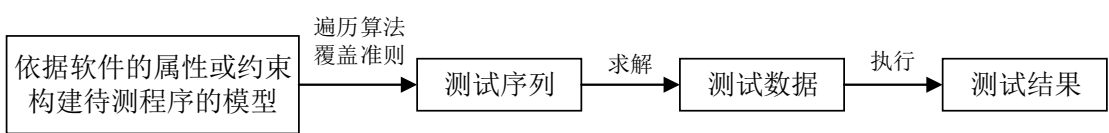


图 3-1. 模型驱动测试一般过程

因此本课题重点研究以下三个问题：

- 1) 如何从给定的 Web 服务组合程序中构建与存储 Web 服务行为的形式化

描述模型；

- 2) 如何从建立的模型中导出软件行为测试序列，如何求出某个特定测试序列对应的输入数据；
- 3) 设计与开发支持本课题所提出的“行为模型驱动的服务组合程序测试用例生成技术”的原型工具，提高该技术的可用性。

3.1 基于服务行为的形式化描述模型的建立

SOA 软件由一组分布在异构环境中的服务按照松散耦合的方式组合而成，服务由描述和实现两部分组成，服务描述通常借助于服务描述语言进行说明。由于参与服务组合的 Web 服务可能来自不同的管理域、为不同的组织机构所有，服务组合只能依据规格说明访问服务。然而服务规格说明仅仅包含了服务接口说明（调用的操作包含哪些数据以及数据格式要求），但是仅描述这些结构化的语法信息是不够的，参与组合的服务通过接口中的多个操作提供服务，这些操作潜藏的各种**数据与控制流**方面的约束往往是业务处理的关键。控制流约束表示操作之间的执行顺序约束，例如电子支付系统中“转账”操作依赖于“建立账户”这样的操作之后。数据流约束表示触发某个特定操作所需的特定数据状态，根据上下文数据决定是否执行相应的操作，例如，ATM 转账业务中，当转账的金额不大于卡内余额时可触发转账操作。同时对于同一操作内部来说，输入数据范围的不同可能导致该操作的内部执行逻辑的不同。

因此，我们将采用**模型**对这些**约束**进行描述。本课题将服务组合程序对 **Web 服务的每次调用（调用服务提供的操作）及接收操作返回的信号处理为一次事件**，事件的执行顺序描述了服务组合程序控制流约束；在事件与事件的转移中添加触发转移的**数据约束**，包括输入数据的**顺序、类型及范围**等的约束信息。拟采用**事件序列图或 UML 时序图**等**行为模型**对服务行为进行**行为描述建模**。事件序列图提供一种简洁、易于理解的形式化表达手段，由节点及边组成，节点代表触发的事件，边代表事件的转移，同时具有较强的理论技术，能够表达数据约束、控制依赖。UML 时序图能够直观的描述按照时间顺序的消息交换，可以方便描述不同服务之间的动态协作关系。

其基本流程如图 3-2 所示。

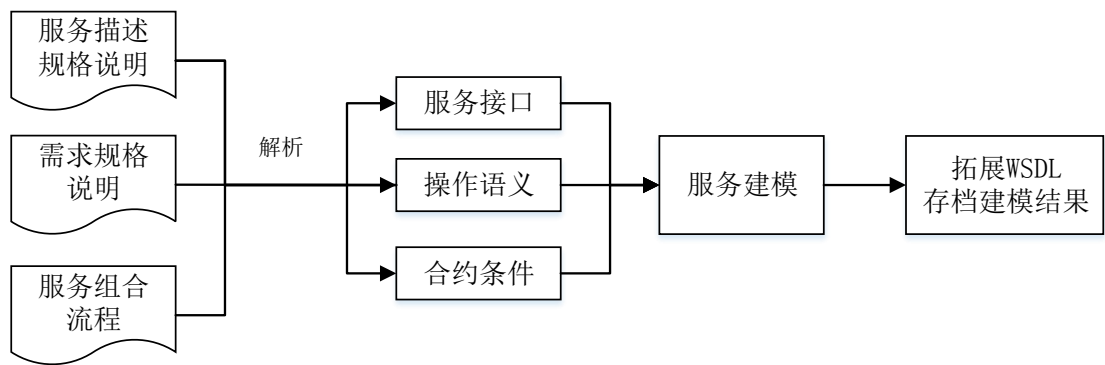


图 3-2 Web 服务组合程序行为模型的建立流程

首先，依据待测 Web 服务的**服务描述规格说明**、**需求规格说明**及**服务组合流程**，分析服务组合程序调用的**服务接口说明**、调用特定操作所需数据的要求，**数据区域划分**等。其次，根据解析的约束进行模型的建立，建立的模型重点关注服务进行了何种操作，操作的数据状态及服务之间的交互。最后，通过进一步对 WSDL 进行拓展，使得拓展后的 WSDL 存档服务行为建模的结果，以便后续测试序列的生成。

3.2 基于行为描述模型的测试序列与测试数据自动生成

理论上，服务应该是无状态的（即服务提供的操作可以任意顺序调用）。然而实践中，这些操作则存在一定的数据流、控制流约束关系（3.1 小节中提及）。为了验证服务行为的正确性和可靠性，必须充分的测试所有服务行为相关的数据流和控制流约束信息。每个从服务组合流程开始到结束的路径称为一条测试序列，遍历该模型即可获取所需测试序列以及程序执行该条序列的数据约束，可通过约束求解获取该测试序列对应的测试数据。

因此我们重点考虑如下三个问题：

- 1) 如何从给定模型中获取测试序列；
- 2) 如何将测试序列转换为相对应的测试数据；
- 3) 如何将测试数据转换为符合服务组合程序要求的测试用例。

3.2.1 测试序列集的获取

我们可以通过遍历给定的模型以获取测试序列集，但对于服务组合程序而言，可能的执行路径可能有很多，在有限的测试资源中要将所有的测试序列输出是不

可能的，因此可以定义一定的覆盖准则来避免路径爆炸的问题，同时避免测试路径集不完备的情况发生。

在给定的行为描述模型的基础上，依据服务行为描述模型的特征，定义各种**约束覆盖准则**，确定各类约束的选取策略以及优先级，例如**事件覆盖**（模型中的每个事件至少执行一次）、**迁移覆盖**（模型中的每个事件迁移至少执行一次）、**数据约束覆盖**（模型中的每个数据分区至少覆盖一次）等等，开发相应的测试序列生成算法，关键问题是面向各种覆盖准则的测试序列生成算法。

目前，已存在多种基于事件序列图模型生成测试序列的算法[9,20,21]。课题组前期工作中研究了如何基于 UML 活动图（扩展的状态图）模型自动生成测试场景对系统行为进行测试，我们也可以通过将行为描述模型转换为状态图，可以有效复用课题组前期工作中开发的基于 UML 活动图的测试场景自动生成技术与支持工具。

3.2.2 测试数据集的生成

在得到测试序列之后，应该生成测试数据使得测试序列可执行，一个测试序列是可行的，则至少存在一组测试数据满足这条路径中全部条件。通常情况下，测试数据的获取主要是由人来分析通过手动来完成，不仅繁琐而且耗时，并且只覆盖部分所需的数据。为了保证生成测试用例的完备与有效性并且提高测试质量，在这里将实现针对每条测试序列自动生成测试数据。

通常情况下，根据测试序列（路径）生成测试数据有以下几种方法：

（1）基于符号运算的静态方法

符号运算是指静态地分析给定路径的有关变量和谓词，对路径上赋值语句的局部变量及分支谓词的变量依次用输入的符号表达式进行替换。其对简单的线性路径很适用，但是即使是对线性问题也会遇到困难：**(1)**循环的运算；**(2)**过程调用；**(3)**数组和指针的处理；**(4)**不可达路径问题。

（2）基于遗传算法的程序直接执行

遗传算法作为一种基于自然选择原理和自然遗传机制的通用搜索算法，模拟自然界中的生命进化机制，在人工系统中实现特定目标的优化。遗传算法在解决大空间、多峰、非线性、全局优化等高复杂度问题时可以显示独特的优势和高效

性，但其面临一系列的实现问题，例如参数的编码方案、算子的选择与控制、适值的调整策略以及初始测试用例群大小与覆盖面的控制等。

（3）基于约束求解策略的方法

通过分析程序路径中所有的约束表达式，逆推得到测试用例的过程。约束求解模块需要处理好两个步骤，一是约束提取步骤，即从待测程序中分析出待求解的约束表达式组。二是约束求解步骤，即采用正确的算法与搜索策略，快速而有效的求出约束表达式组的若干个可行解。

在该研究方法中，采用第三种方法，在约束求解步骤中综合使用线性约束求解策略与迭代求解策略，用线性约束求解策略先求解出所有的解，然后用迭代求解的策略选出满足约束的解，从而得出满足特定测试路径的测试数据。

3.2.3 测试用例集的生成

通过前面的步骤，已经通过一系列覆盖准则遍历行为模型得到了测试序列的集合，又通过约束求解策略的方法得到了相对应的测试数据，因此我们可以根据服务组合程序调用接口的数据格式要求，将测试数据转换为符合输入格式的测试用例，就得到了测试用例集合。

3.3 基于行为描述模型的测试用例自动生成工具

为了提高行为模型驱动的服务组合程序测试用例生成技术的可用性，需开发出一个工具支持该技术，该工具的初步架构图如图 3-3 所示：

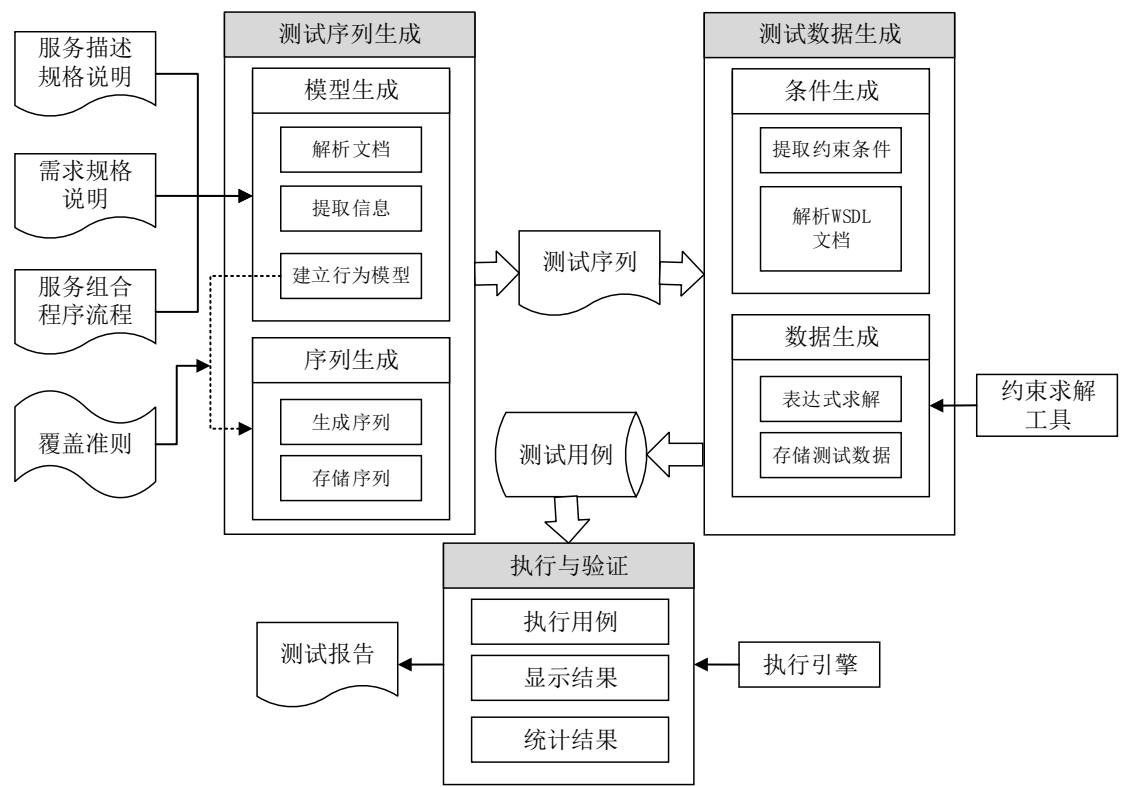


图 3-3 BM4WSC 工具架构

图 3-3 描述了支持工具的系统初步架构，灰色矩形框表示基本组件，对应测试序列生成器，测试数据生成器以及执行与验证。每个组件由多个模块构成。

测试序列生成器主要负责解析用户输入的服务描述规格说明、需求规格说明、服务组合程序流程，提取出约束信息进行行为建模；进一步依据选择的覆盖准则进行测试序列的生成。

测试数据生成器主要负责提取测试序列中的约束条件，依靠约束求解工具，求出相应的测试数据。

执行与验证，主要负责执行生成的测试用例，对比测试用例的实际执行结果与预期执行结果，生成测试报告。

3.4 实例验证

本课题拟采用实证研究与变异分析相结合的方式验证提出的模型测试技术与框架的可行性与有效性。将以课题组前期开发 Web 服务组合程序为研究对象。这些研究对象均基于业务流程执行语言实现服务组合。

采用变异分析[36]的方法进一步评估行为模型驱动的用例生成技术的有效性，在已有的 Web 服务实现中，采用变异分析植入故障，然后使用本文所

提及方案生成的测试用例执行变异前后的 Web 服务及其组合，分析与评估测试用例的故障检测能力。

采用变异得分作为实验的评价指标，我们将对比不同覆盖准则及随机测试生成的测试用例集的变异得分，对于规模相同的测试用例集来说，变异得分越高说明该测试用例集的故障检测能力越强。

3.5 研究难点

本课题提出一种行为模型驱动的 Web 服务组合程序测试用例生成技术，开发了相应的支持工具。在该技术的实现过程中可能会遇到以下几个问题：

- 1) **行为模型建立过程中约束信息的获取：**在进行 Web 服务行为模型建模过程中，需求规格说明文件并未有统一标准格式，因此自动化解析较为困难，且服务规格说明仅仅包含了服务接口说明，解析后获取的数据约束不够充分。因此可能需要服务开发商提供符合要求的需求规格说明文件，或人为添加额外的数据约束。
- 2) **如何存储包含约束的服务行为描述结果：**在进行服务行为描述结果存储时，由于组成模型的元素之间有多种顺序关系，因此如何较为合理的存储行为描述结果是该技术实现的一个关键问题。
- 3) **如何合理定义基于行为模型的覆盖规则以生成测试序列：**覆盖准则的不同导致生成测试序列所用时间及规模不同，各种约束的选取策略不同也导致最后生成测试用例的不同。如何选取合适的覆盖准则、设定不同准则间的优先级、或不同准则的组合方式是我们在生成测试序列中面临的问题。
- 4) **测试数据生成过程中存在的问题：**在针对每条测试序列生成测试数据时，需要分析约束条件，提取出约束表达式，有可能造成约束条件提取的不完全。且进行约束求解的过程中，由于变量类型复杂等问题，可能造成求解限制，导致无法求出所需解，这也是在具体实现中不可避免的研究问题。

4. 研究工作进度安排

本课题工作进度安排如表 4.1 所示。

表 4.1 工作进度安排

时间段	任务
2016.07—2016.08	查阅国内外文献，了解基于模型驱动的 Web 服务测试研究现状，初步拟定课题研究方案，完成开题报告
2016.09—2016.11	学习 Web 服务组合相关技术，了解 WSDL 文档及 BPEL 文件的结构，实现形式化描述模型的表达与存储
2016.12—2017.02	实现测试用例自动生成
2017.03—2017.06	初步开发支持工具
2017.07—2017.09	进一步完善、调试工具，进行实验验证
2017.10—2017.12	分析数据并得出结论，撰写论文，准备答辩

参考文献

- [1] M. P. Papazoglou, W. J. V. D. Heuvel. Service oriented architectures: approaches, technologies and research issues[J]. Vldb Journal, 2007, 16(3): 389-415.
- [2] M. P. Papazoglou, P. Traverso, S. Dustdar, F. Leymann. Service-Oriented Computing: a Research Roadmap[J]. International Journal of Cooperative Information Systems, 2008, 17(2): 223-255.
- [3] A. T. Endo, M. Linschulte, A. D. S. Simão, et al. Event-and Coverage-Based Testing of Web Services, in: Proceedings of the 4th International Conference on Secure Software Integration & Reliability Improvement Companion (SSIRI 2010), IEEE Computer Society, 2010, pp. 62-69.
- [4] 骆翔宇, 谭征, 苏开乐, 吴立军. 一种基于认知模型检测的 Web 服务组合验证方法[J]. 计算机学报, 2011, 34(6): 1041-1061.
- [5] N. Milanovic, M. Malek. Current solutions for web service composition[J]. IEEE Internet Computing, 2004, 8(6): 51-59.
- [6] G. Canfora, M. D. Penta. Testing services and service-centric systems: challenges and opportunities[J]. It Professional, 2006, 8(2): 10-17.

- [7] G. Canfora, M. D. Penta. Service-Oriented Architectures Testing: A Survey, Lecture Notes in Computer Science 5413, 2009, pp. 78-105.
- [8] 魏莹. 基于模型检测的 Web 服务测试方法研究[D]. 北方工业大学, 2012.
- [9] F. Belli, A. T. Endo, M. Linschulte, A. Simao. Model-based Testing of Web Service Compositions, in Proceedings of the 6th IEEE International Symposium on Service Oriented System Engineering (SOSE 2011), IEEE Computer Society, 2011, pp. 181-192.
- [10] W. Grieskamp, N. Kicillof, K. Stobie, V. Braberman. Model-based quality assurance of protocol documentation: tools and methodology[J]. Software Testing Verification & Reliability, 2011, 21(1): 55-71.
- [11] 张峻. 基于 UML2.0 动态视图的 Web 服务模型测试方法及其应用[D]. 苏州大学, 2007.
- [12] Y. Zheng, J. Zhou, P. Krause. An Automatic Test Case Generation Framework for Web Services[J]. Journal of Software, 2007, 2(3): 64-77.
- [13] A. Bertolino, G. D. Angelis, L. Frantzen, A. Polini. Model-Based Generation of Testbeds for Web Services, in: Proceedings of the 20th IFIP International Conference on Testing of Software and Communicating Systems (TESTCOM 2008), 2008, pp. 266-282.
- [14] A. T. Endo, A. Simao. Model-Based Testing of Service-Oriented Applications via State Models, in: Proceedings of the 8th IEEE International Conference on Services Computing (SCC 2011), 2011, pp. 432-439.
- [15] C. S. Keum, S. Kang, I. Y. Ko, J. Baik, Y. I. Choi. Generating Test Cases for Web Services Using Extended Finite State Machine, in: Proceedings of the 18th IFIP International Conference on Testing of Software and Communicating Systems (TESTCOM 2006), 2006, pp. 103-117.
- [16] A. S. Kalaji, R. M. Hierons, S. Swift. An integrated search-based approach for automatic testing from extended finite state machine (EFSM) models[J]. Information & Software Technology, 2011, 53(53): 1297-1318.
- [17] M. Kiran, A. J. H. Simons. Model-Based Testing for Composite Web Services in Cloud Brokerage Scenarios, in Proceedings of the 3rd European Conference on

- Service-Oriented and Cloud Computing (ESOCC 2014), 2014, pp. 190-205.
- [18]王任. 基于有限状态机的 Web 服务测试用例生成方法[D]. 江西财经大学, 2009.
- [19]F. Belli, M. Linschulte. Event-Driven Modeling and Testing of Web Services, in: Proceedings of the 32nd IEEE International Computer Software and Applications Conference (COMPSAC 2008), IEEE Computer Society, 2008, pp. 1168-1173.
- [20]F. Belli, A. T. Endo, M. Linschulte, A. Simao. A holistic approach to model-based testing of Web service compositions[J]. Software Practice & Experience, 2014, 44(2): 201–234.
- [21]C. S. Wu, C. H. Huang. The Web Services Composition Testing Based on Extended Finite State Machine and UML Model, in Proceedings of the 5th International Conference on Service Science and Innovation (ICSSI 2013), 2013, pp. 215-222.
- [22]G. Zhang, R. Mei, J. Zhang. A Business Process of Web Services Testing Method Based on UML2.0 Activity Diagram, in Proceedings of the Workshop on Intelligent Information Technology Application (IITA 2007), 2007, pp. 59-65.
- [23]V. Pretre, A. D. Kermadec, F. Bouquet, et al. Automated UML models merging for web services testing[J]. International Journal of Web & Grid Services, 2009, 5(2): 107-129.
- [24]S. Ali, L. C. Briand, J. U. Rehman, et al. A state-based approach to integration testing based on UML models[J]. Information & Software Technology, 2007, 49(11–12): 1087-1106.
- [25]M. E. Cambronero, G. Diaz, J. J. Pardo, V. Valero. Using UML Diagrams to Model Real-Time Web Services, in Proceedings of the 2nd International Conference on Internet and Web Applications and Services (ICIW 2007), 2007, pp. 24.
- [26]D. Lee, M. Yannakakis. Principles and methods of testing finite state machines-a survey[J]. Proceedings of the IEEE, 1996, 84(8): 1090-1123.
- [27]Model-Based Integration Testing with Communication Sequence Graphs. In Justyna Zander, Ina Schieferdecker, and Pieter J. Mosterman, editors: Model-Based Testing for Embedded Systems, CRC Press, 2011, pp. 223-243

- [28] 韩德帅, 杨启亮, 邢建春. 一种软件自适应 UML 建模及其形式化验证方法[J]. 软件学报, 2015, 26(4): 730-746.
- [29] C. Sun, G. Wang, B. Mu, H. Liu, Z. Wang, T. Y. Chen. Metamorphic Testing for Web Services: Framework and a Case Study, in Proceedings of the 9th IEEE International Conference on Web Services (ICWS 2011). IEEE Computer Society, 2011, pp. 283-290.
- [30] C. Sun, G. Wang, B. Mu, H. Liu, Z. Wang, T. Y. Chen. A metamorphic relation-based approach to testing web services without oracles[J]. International Journal on Web Service Research, 2012, 9(1): 51-73.
- [31] C. Sun, G. Wang, K. Y. Cai, T. Y. Chen. Towards Dynamic Random Testing for Web Services, in Proceedings of the 36th Annual IEEE International Computer Software and Applications Conference (COMPSAC 2012), IEEE Computer Society, 2012, pp. 164-169.
- [32] C. Sun, Y. Zhai, Y. Shang, Z. Zhang. BPELDebugger: An effective BPEL-specific fault localization framework[J]. Information and Software Technology, 2013, 55(12): 2140-2153.
- [33] C. Peltz. Web services orchestration and choreography[J]. IEEE Computer, 2003, 36(10): 46-52.
- [34] C. Sun, Y. Shang, Y. Zhao, T. Y. Chen. Scenario-Oriented Testing for Web Service Compositions Using BPEL, in: Proceedings of the 12th International Conference on Quality Software (QSIC 2012), IEEE Computer Society, 2012, pp. 171-174.
- [35] C. Sun, Y. Zhao, L. Pan, L. Hui, T. Y. Chen. Automated Testing of WS-BPEL Service Compositions: A Scenario-Oriented Approach[J]. IEEE Transactions on Services Computing, in press (accepted on 2 August 2015).
- [36] R. A. DeMillo, R. J. Lipton, F. G. Sayward. Hints on test data selection: Help for the practicing programmer [J]. IEEE Computer, 1978, 11(4): 34-41.