

目录

- 1. 背景介绍..... 1
 - 1.1 Web 服务 1
 - 1.2 Web 服务的测试 2
 - 1.3 动态随机测试技术 2
- 2. 面向 Web 服务的动态随机测试系统分析与设计 3
 - 2.1 需求分析 3
 - 2.2 系统设计 5
- 3. 系统实现与演示..... 6
 - 3.1 系统功能 6
 - 3.2 系统演示 8

1. 背景介绍

Web 服务基于 XML、SOAP、WSDL、和 UDDI 等技术，不仅提高了软件模块的复用性还能够降低系统集成的复杂性。Web 服务具有规范性、松散耦合、跨平台和高度可集成的特点，广泛用来构造分布式系统。软件测试是一种保障软件质量的重要手段，而 Web 服务的动态特征和分布式的特点给 Web 服务测试带来了挑战。服务提供者发布服务之前通常已经测试了 Web 服务，使得 web 服务的故障不容易被揭示。随机测试是广泛使用的软件测试技术，具有原理简单、易于实现自动化的特点。由于没有利用待测软件信息以及测试信息，随机测试技术的测试效率可能不高。动态随机测试是一种改进的随机测试技术，结合了软件测试控制论的思想将反馈机制引入到测试用例选取的过程中，旨在提高了随机测试的测试效率。很多研究工作表明动态随机测试技术的测试效率比随机测试技术的测试效率高。自动化测试可以节约测试资源，因此有必要研究面向 Web 服务的自动化动态随机测试技术。

1.1 Web 服务

Web 服务遵循面向服务的体系结构（SOA），具有基于组件和开放性的特点。SOA 包含三个实体和三个基本操作，具体来说：1）三个实体：服务提供者、服务请求者和服务代理（也称服务注册中心）；2）三个操作：发布服务、查询服务和绑定服务。服务提供者是服务的所有者并通过服务访问平台提供服务。服务请求者是需求特定功能的企业或组织，通过服务代理提供的目录搜索相关服务，得到如何调用该服务的信息，然后根据这些信息调用服务。服务代理是存储服务描述信息的信息库并且服务请求者在这里查找需求的服务。

Web 服务的核心技术是可扩展标记语言（eXtensible Markup Language, XML）。XML 是被设计用来传输及携带数据信息。它为 Web 服务提供了统一的数据格式。包括消息、服务描述以及工作流的描述等不同层次的协议，都采用 XML 作为定义语言。Web 服务的具体协议包括简单对象访问协议（Simple Object Access Protocol, SOAP），Web 服务描述语言（Web Services Description Language, WSDL）和通用描述、发现与集成规范（Universal Description, Discovery and Integration, UDDI）。

SOAP 是一个基于 XML 的可扩展消息信封格式，需同时绑定一个网络传输协议。它是用于交换 XML 编码信息的轻量级协议。

WSDL 遵循 XML 的格式来描述网络服务的端点，用来定义 Web 服务的功能、操作以及调用方式。

UDDI 提供了在 Web 上描述并发现服务的框架，是面向 Web 服务的信息注册中心的实现标准和规范。

1.2 Web 服务的测试

Web 服务是 SOA 软件的基本组成单元。服务描述与实现的一致性确保 SOA 软件质量的重要方面。与传统软件开发相比，SOA 不仅改变系统的构建和使用方式，也改变了测试方式。在 SOA 环境下，服务不是在用户的系统上物理集成而是通过调用的方式获取 SOA 的上述特点增加了测试的复杂性，具体来说：

- (1) 由于基于服务的系统分布不集中，用户需要在不同的配置下保证服务的质量。
- (2) 系统中服务独立变化的特点增加迭代测试的难度。
- (3) 由于服务的定义于实现分离，同一个服务描述在不同的时间可能使用不同的实现方式。系统的功能随需求不断变化，表现为添加或替换某些服务。上述情形增加了集成测试的难度。
- (4) 服务提供者提供服务描述，然而服务代理和用户难以判断服务描述的可信度、完整度，使得设计测试用例更加困难。

1.3 动态随机测试技术

随机测试和分区测试是两个主要的软件测试方法。随机测试在选取测试用例时，一般按照均等的概率或者根据功能剖面在输入域中随机选取测试用例。所谓功能剖面，是根据待测软件不同输入数据的使用频率人为设定概率：使用频率高的输入数据对应的测试用例则有较高的选取概率，使用频率低的输入数据赋予较低的选择概率。随机测试选取测试用例的方法较为简单，有助于估计待测软件的稳定性与可信性。

与随机测试不同，分区测试旨在同构地产生测试用例以提高测试效率。分区测试首先将输入域划分成若干个不相交的分区，然后从每一个分区中挑选测试用例并执行。在分区测试中，分区期望存在同构的性质，即相同分区下的测试用例应该使程序产生相似的行为。在实际中，这种同构的性质难以保证，分区测试的效率难以保障。

很多研究者发现引起故障的输入趋向于集簇在连续区域。基于上述观察，蔡教授提出了动态随机测试技术。动态随机测试策略的主要特点是在测试的过程中依据每一次测试用例的执行结果动态改变测试剖面，使得失效率大的分区被选择的概率高。动态随机测试策略依据测试剖面随机选择一个分区，然后在选中的分区中随机选择一个测试用例并执行。如果执行的测试用例揭示了软件中的故障，该策略增大执行的测试用例所在的分区被选择的概率。相反地，如果没有揭示软件中的故障，该策略减小执行的测试用例所在的分区被选择的概率。

课题组结合 Web 服务的特点与动态随机测试的基本原理，提出了面向 Web 服务的动态随机测试技术，并提出了测试框架，开发了面向 Web 服务的动态随机测试系统，相关成果以 Towards Dynamic Random Testing for Web Services 为题发表在 IEEE International Computer Software and Applications Conference。

2. 面向 Web 服务的动态随机测试系统分析与设计

2.1 需求分析

Web 服务是面向服务架构体系的基础组件。依据 Web 服务的 WSDL 文档，用户可以定位和请求 Web 服务提供的操作，也可以将已存在的 Web 服务集成到正在开发的系统中来提高开发效率。因此 Web 服务的质量受到人们广泛的关注。虽然 Web 服务提供者在发布 Web 服务之前会测试提供的服务，但是提供者很难保证测试的充分性。在集成已存在的 Web 服务之前，快速、高效地测试该服务是必不可少的步骤。广泛采用的随机测试技术没有利用待测软件的信息以及测试的历史信息，因此该技术的测试效率可能不高。动态随机测试技术改进了随机测试技术，依据测试历史信息选择下一个分区。很多研究工作表明：相较随机测试，动态随机测试技术具有更高的故障检测效率。因此动态随机测试技术适用于测试 web 服务。

面向 Web 服务的动态随机测试工具用例图如图 1 所示，各个用例的详细描述如下：

- (1) **解析待测目标：**测试者首先依据 web 服务的 WSDL 文档分析该服务提供的操作，然后针对具体的操作解析该操作的输入与输出，并将所得信息记录在文件中。
- (2) **获取测试用例：**有两种生成测试用例的方式：自动生成测试用例和上传包含测试用例的文件，具体来说：1) 自动生成测试用例：根据待测服务指定操作的参数信息，系统自动随机生成大量测试用例。2) 上传测试用

例文件：用户提供描述测试用例集的 XML 文件。该文件须符合指定的

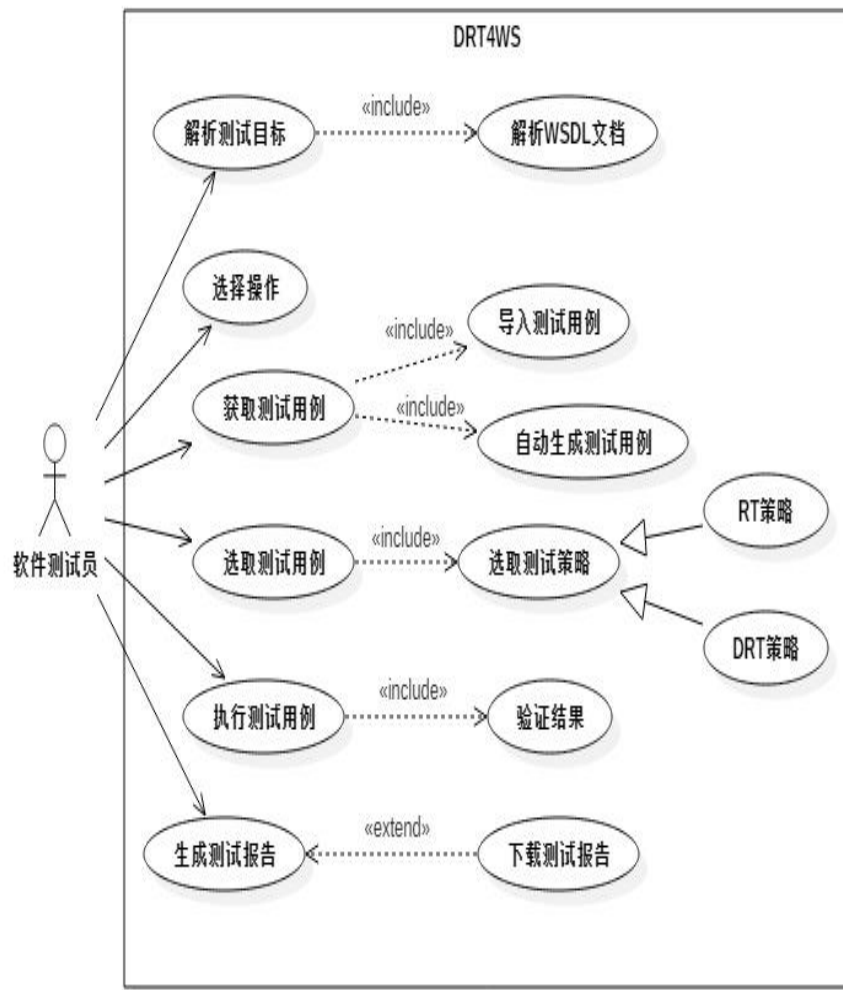


图 1 DRT4WS 用例图

输入格式。当文件内容与输入输出的接口不一致时，工具报错。

- (3) **选取测试用例：**工具依据用户指定的策略自动地选取测试用例并执行。不同的测试用例选取策略对应不同的测试技术，测试技术是影响测试效率的关键。目前工具提供随机测试策略和动态随机测试策略。
- (4) **执行测试：**工具根据测试用例中的数据发送特定的请求，然后解析响应的信息并得到返回结果，然后将响应的结果与预期结果比较。
- (5) **生成测试报告：**当测试过程完成后，工具在界面上显示测试报告。测试报告中包含执行的测试用例个数，被杀死的错误个数及杀死错误的测试用例的信息等。测试者可以下载完成的测试报告文档。

2.2 系统设计

根据上述需求分析，设计了 DRT4WS 的系统架构，如图 2 所示。该系统分为 6 个组件，每个组件的描述如下：

- (1) **预处理组件：**预处理组件负责为待测 Web 服务创建测试项目目录以及解析待测服务的 WSDL 文件，包括以下 2 个模块：1) **工作目录创建模块：**根据测试者提供的 URI 为待测 Web 服务创建测试目录。工具在测试过程中所产生的所有文档都存放在测试目录中；2) **Web 服务解析模块：**根据 URI 获取 Web 服务的 WSDL 文件，解析服务描述中定义的所有操作的信息。操作信息包括：操作名，操作输入接口所包含的参数个数、参数名称、各参数的数据类型，操作输出接口的数据类型。
- (2) **测试用例生成组件：**生成测试用例组件负责生成测试用例并将测试用例集以文件的形式保存下来，包括以下 2 个模块：1) **随机生成模块：**根据操作的信息，以随机的方式生成输入接口各个参数的值，并按照操作输入接口的格式封装成测试用例。所有测试用例组成测试用例集，以 XML 文件的形式保存在测试项目目录中；2) **人工导入模块：**接收 XML 格式的测试用例集文件，并根据待测 Web 服务的操作信息对文件内容进行验证。文件符合服务操作的接口格式时，人工导入模块将此文件复制到测试项目目录下；文件不符合接口格式时，将错误信息显示给测试人员。对于其他文件格式的测试用例集合的解析与验证，工具预留了扩展接口。分区规则，组件根据分区规则对测试用例集进行划分，并设定初始概率分布。
- (3) **分区设定组件：**分区设定组件负责对测试用例集进行分区并设置初始概率分布。该组件包括以下 2 个模块：1) **分区划分模块：**根据测试者提供的分区规则，将测试用例集合划分成若干分区。分区划分的结果是分区表，其中记录各分区编号以及分区内包含的测试用例的编号；2) **概率分布设定模块：**根据测试者的选择设定分区的初始概率分布。目前工具支持三种设定初始概率分布的方式：人工输入、平均分布和根据分区容量比例设定初始概率分布。
- (4) **测试用例选取组件：**用例选取组件负责根据测试者设定的测试策略，在测试过程中自动选取测试用例交给测试执行组件。用例选取组件是工具的关键组件。用例选取组件选取测试用例的过程分为两步，具体来说：首先，按照一定测试策略，从测试用例集中挑选符合策略标准的测试用

例构成测试用例集合的子集；然后，从测试用例集合的子集中随机挑选一个测试用例，子集中每个测试用例被选中的概率均等。分区选取模块是用例选取组件的核心模块，它负责从测试用例集合中挑选子集。它是具体测试策略在工具中的体现。目前工具提供两种分区选取模块——随机模块与动态随机模块。随机模块直接将整个测试用例集作为子集传递给用例选取模块；动态随机模块则根据概率分布选取一个分区 p_i ，再查询分区表，选出 p_i 中的测试用例形成子集传递给用例选取模块。动态随机测试技术在用例选取组件中对应两个模块：1) **分区选取模块**：负责根据分区概率分布和分区表挑选测试用例子集；2) **分区概率调整模块**则根据每个测试用例的执行结果来调整概率分布。分区选取模块也是工具扩展的关键。测试人员可以编写子集的分区选取模块，并在工具配置文件中“注册”此模块，就可以在工具中使用自己定义的测试策略。

- (5) **测试执行组件**：测试执行组件负责执行测试用例，截取输出并记录日志包括以下 2 个模块：1) **执行测试模块**：将测试用例中的数据按照待服务的接口信息封装成 SOAP 消息发送给待测服务，然后接收服务响应的 SOAP 消息，解析服务请求的结果并将结果传递给评估组件进行判断是否发现错误。DRT4WS 使用开源软件 soapUI 的 API 实现 SOAP 消息的封装、解封以及 Web 服务的调用。2) **日志记录模块**：负责在测试过程中采集测试信息并记录。从测试日志中提取并分析出测试报告。
- (6) **评估组件**：评估组件由两部分组成：结果评估模块和终止判定模块。1) **结果评估模块**：负责根据每个测试用例的执行结果与预期结果判断是否检测出待测服务的故障。2) **终止判定模块**：根据测试人员指定的测试终止条件来判断测试过程是否完成。目前工具提供两种测试终止条件：一种是执行若干条测试用例后停止测试，另一种检测出若干故障后停止测试。

3. 系统实现与演示

3.1 系统功能

DRT4WS 支持不同的测试策略测试 Web 服务，主要实现了一下功能：

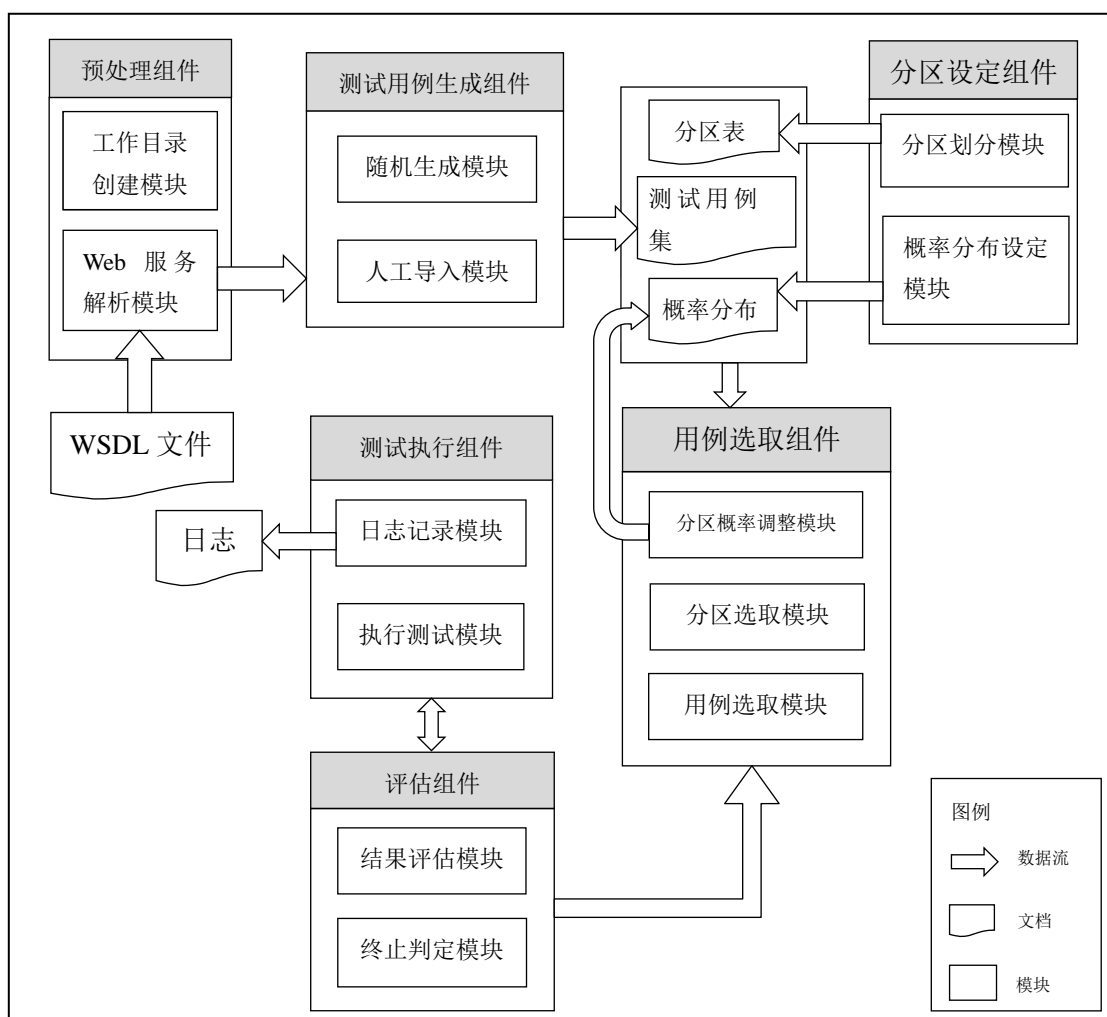


图 2 DRT4WS 系统架构图

- (1) 解析 WSDL 文档
 - 1) 获取待测服务对应的 WSDL 文档。
 - 2) 解析 WSDL 文档。
 - 3) 获取不同操作的参数信息。
- (2) 生成测试用例
 - 1) 根据解析的参数信息，系统随机地产生测试用例。
 - 2) 用户可以上传存贮在 XML 文件中的测试用例。
- (3) 执行测试用例
 - 1) 输入测试用例，自动形成请求消息，并发待测服务。
 - 2) 接受返回的消息并提取执行的结果。
- (4) 测试结果分析
 - 1) 将某个测试用例的执行结果与预期对比。

2) 统计揭示的故障数目。

3.2 系统演示

运用 AddService 服务来演示系统，展示系统测试 Web 服务的过程。该服务提供计算两数之和的功能，用户输入两个数字，然后返回它们的和。



图 3 DRT4WS 系统首页

如图 3 所示，系统主要由三个部分组成：待测服务地址栏、解析按钮和重置按钮。用户可以在地址栏中输入待测服务的地址。如果地址信息不正确，“重置”按钮可以快速清空地址栏。如果地址信息正确，用户可以点击“解析”按钮，然后系统进入操作选择界面，如图 4。在该界面中，用户可以在界面的最上端点击“下三角”按钮并选择要测试的操作。当用户选择要测试的操作之后，界面的中间位置将显示要测试操作的参数信息，主要包括：1) 参数的名称；2) 参数要求的数据类型。显示栏的右边是参数说明模块，用户可以补充更多参数的信息。在测试策略标签下，用户可以选择测试策略。系统提供两种测试策略：1) 随机测试策略；2) 动态随机测试策略。用户根据需求选择测试策略之后，点击“下一步”进入测试用例生成界面，如图 5 所示。在该界面支持两种方式产生测试用例，具体来说：

- (1) **随机生成测试用例：**如果选择这种方式产生测试用例，用户首先需要选中“自动添加”选项，然后在后面地文本框中填入生成测试用例地个数，最后点击“开始”按钮。根据上一个界面选中操作种参数的信息，系统可

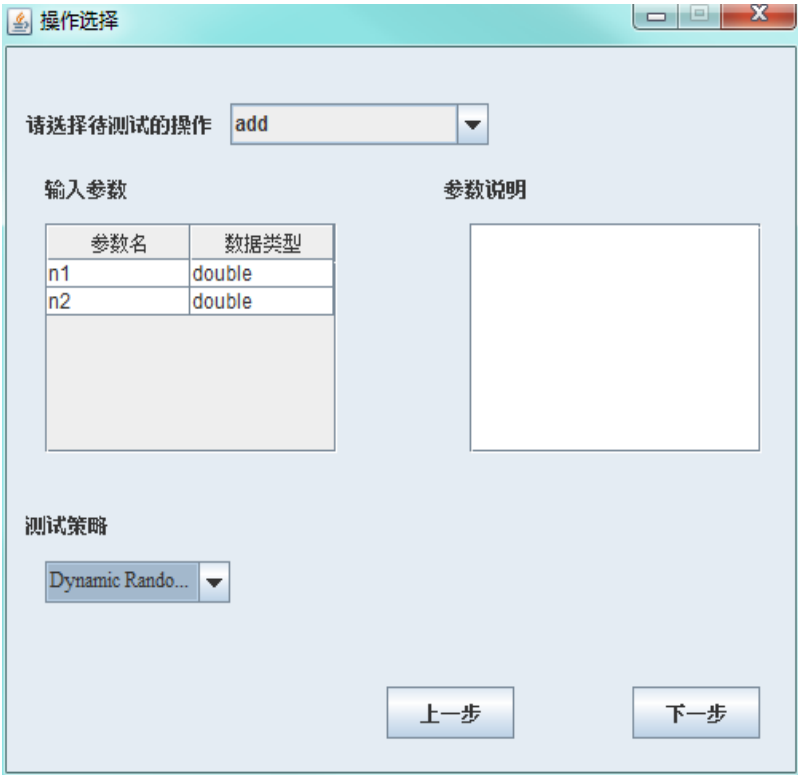


图 4 DRT4WS 系统准备界面

能需要一段时间随机生成要求数目的测试用例。

- (2) **人工添加测试用例：**如果选择这种方式产生测试用例，用户首先需要将待添加的测试用例按照一定的格式存贮在 XML 文件中，然后点击“导入文件”，接着再点击“浏览”按钮并选择 XML 文件，最后点击“导入”按钮。系统在后台解析 XML 文件提取测试用例。

这两种方式产生的测试用例最后都会显示在最上方的显示板中，显示的信息主要包括如下 3 个部分：1) 测试用例标号；2) 输入的数据；3) 测试预期。用户可以在最上面检查生成的测试用例。

依据上一个界面选择的测试策略不同，用户点击“下一页”后，测试界面分为两种，具体说来：

- (1) **动态随机测试策略：**动态随机测试策略是在分区基础上进行的，用户还需要在如图 6 的界面进一步配置分区信息。用户进入该界面之后点击“添加分区”按钮，然后用户在界面的文本区添加分区信息，如图 7 所示。该图中两参数满足 $0 \leq n1 < 10000$ ， $0 \leq n2 < 2000$ 属于第一个分区，该图中两参数满足 $0 \leq n1 < 10000$ ， $2000 \leq n2 < 10000$ 属于第二个分区。配置分区规则之后，用户通过点击“下一步”按钮进入测试界面。



图 5 DRT4WS 测试用例生成界面

(2) **随机测试策略：**生成测试用例之后，用户通过点击界面的“下一步”按钮进入测试界面。

配置测试信息之后，用户可以进入测试界面，如图 8 所示。该界面可以分为 3 个模块，具体来说：

- (1) **展示测试设置模块：**该模块展示的信息包括：1) 测试的操作名字。用户可以方便地观察目前测试的方法；2) Web 服务。该标签下方显示目前测试的 Web 服务的地址；3) 测试策略信息。在这里，用户可以了解目前的测试策略；4) 测试用例集。在这个标签的下面，系统显示测试用例的生成方式。如果使用随机的方法生成测试用例，则显示“随机产生”。如果是人工导入测试用例，则显示导入的测试用例集的路径以及相应的名字。
- (2) **测试控制模块：**在该模块中，用户可以选择测试的终止条件：1) 执行一定数目的测试用例之后程序停止。由于测试资源是有限的，长时间地测试是不可能的。用户可以设定执行的测试用例数目上限，当执行的测试用例数目达到用户设定的阈值时，系统自动停止测试并生成相应的测试报告。2) 揭示一定数目的故障。当系统揭示的故障数目与预期相同时，



图 6 分区配置生成界面



图 7 分区示例界面

系统自动停止测试并生成相应的测试报告。设置测试终止条件之后，如果选择的时随机测试策略，用户可以点击“start”按钮，开始测试。如果选择的是动态随机测试技术，用户需要进一步设置动态随机测试技术涉

及到的参数，然后点击“start”按钮并开始测试。

- (3) **测试过程模块：**该模块主要展示测试过程的信息。系统执行某个测试用例时，该模块显示如下内容：1) 该测试用例是否揭示了故障。如果揭示了故障则显示“发现错误”，否则显示“未发现错误”；2) 测试用例信息：执行的测试用例的编号、测试功能不同参数的实际值；3) 该测试用例预期的测试结果；4) 实际的测试结果。如果测试用例的执行结果与预期结果相同，该测试用例没有揭示软件中的故障；反之，该测试用例揭示了软件故障。

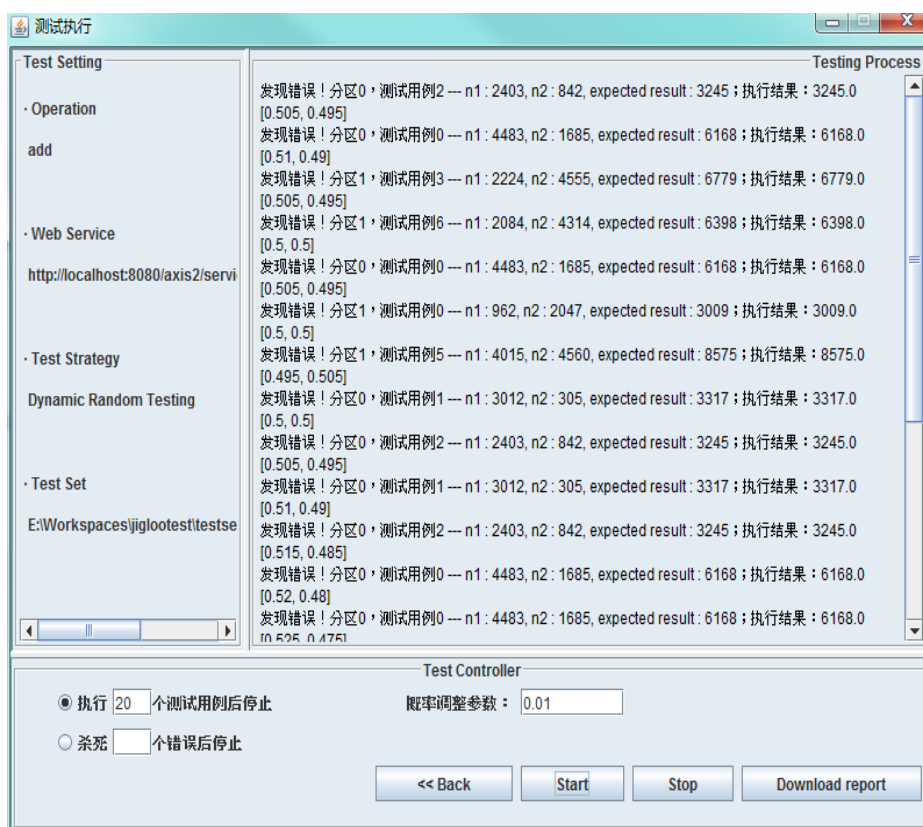


图 8 执行测试界面