[1. 概述 2](#_Toc385592377)

[1.1 目的 2](#_Toc385592378)

[1.2 总体部署 2](#_Toc385592379)

[1.3 静态视图 3](#_Toc385592380)

[2. 需求描述 4](#_Toc385592381)

[2.1 关键字库需求 4](#_Toc385592382)

[2.1.1 对接网管 4](#_Toc385592383)

[**2.1.1.1** **需求和基本框架** 4](#_Toc385592384)

[**2.1.1.2** **低级关键字库** 4](#_Toc385592385)

[**2.1.1.3** **OMM接口适配层** 5](#_Toc385592386)

[**2.1.1.4** **在测试用例中配置被测网元** 5](#_Toc385592387)

[**2.1.1.5** **与其他关键字库的配合** 5](#_Toc385592388)

[2.1.2 驱动测试工具执行业务 5](#_Toc385592389)

[**2.1.2.1** **测试工具的角色** 5](#_Toc385592390)

[**2.1.2.2** **测试工具接口适配** 6](#_Toc385592391)

[**2.1.2.3** **测试工具关键字库** 6](#_Toc385592392)

[**2.1.2.4** **多个测试工具间的配合** 6](#_Toc385592393)

[2.1.3 数据采集 6](#_Toc385592394)

[**2.1.3.1** **基本框架** 6](#_Toc385592395)

[**2.1.3.2** **数据采集关键字** 7](#_Toc385592396)

[**2.1.3.3** **接口适配** 7](#_Toc385592397)

[**2.1.3.4** **数据分析** 7](#_Toc385592398)

[2.1.4 基本辅助功能 7](#_Toc385592399)

[**2.1.4.1** **日志** 7](#_Toc385592400)

[**2.1.4.2** **文件操作** 7](#_Toc385592401)

[**2.1.4.3** **本地数据库** 7](#_Toc385592402)

[2.2 管理需求 8](#_Toc385592403)

[2.2.1 测试用例与关键字管理 8](#_Toc385592404)

[2.2.2 对接Onetest 8](#_Toc385592405)

[2.2.3 持续集成 8](#_Toc385592406)

[2.3 扩展功能 8](#_Toc385592407)

[2.3.1 智能化（暂不实现） 8](#_Toc385592408)

[**2.3.1.1** **测试数据自动配置** 8](#_Toc385592409)

[**2.3.1.2** **自动生成消息流程** 8](#_Toc385592410)

[2.3.2 分布式（暂不实现） 8](#_Toc385592411)

[**2.3.2.1** **测试工具共享** 8](#_Toc385592412)

[**2.3.2.2** **测试环境共享** 8](#_Toc385592413)

# 概述

## 目的

1. **简化测试设计，使得测试人员专注于自己最需要做的事情上。**通过使用Robot Framework中的高级关键字，可以简化测试用例的编写，设计测试用例时能够仅关注测试需要关注的内容；
2. **简化支撑库的开发，能够快速实现需要的功能。**Robot Framework有Python语言生态环境支持，各种通用功能基本都有成熟的模块可以使用，使用Robot Framework能够简化支撑测试的功能库的开发；另一方面，各种支撑库与测试框架只有一个统一的接口，不会与测试框架紧密耦合，便于支撑库的修改；
3. **支持复杂网络环境的部署，实现较复杂的组网测试。**Robot Framework支持远程调用，能够较为方便的在多台主机上灵活部署不同的组件；
4. **实现自动化测试执行与测试报告。**Robot Framework是自动化测试框架，因此其核心功能就是自动化测试执行控制，并能生成详尽的测试记录与报告；
5. **简化测试用例管理与开发。**Robot Framework的测试用例使用文本文件保存，因此可以通过各种版本管理工具对其进行管理，如SVN等，对测试用例的管理应该与代码的管理一致；
6. **与持续集成过程对接。**Robot Framework有官方的Jenkins插件，可以无缝嵌入持续集成的测试环节中。

## 总体部署



1. 框图为逻辑划分，测试PC、测试工具、OMM在物理上可以合并；
2. 测试PC中使用Robot Framework作为自动化测试框架；
3. 测试PC从测试脚本管理服务器中下载测试脚本和支撑脚本运行的库；
4. 测试工具提供各种工具控制功能，当工具与测试PC不在一台机器上时，通过在工具上驻留脚本包装远程调用接口，供测试PC远程使用；
5. 各种数据服务器中，通过在服务器（或代理服务器）中驻留脚本，包装各种数据查询检索功能，供测试PC远程调用；
6. 如果OMM客户端与测试PC物理合一，则可以通过Telnet直接操作网管命令，如果OMM客户端独立，则需要通过脚本包装数据配置与查询功能，供测试PC远程调用；
7. 测试脚本管理平台可以使用SVN，测试用例、测试关键字库和测试记录/报告独立保存；

## 静态视图



1. 测试用例的编写遵循Robot Framework的格式，使用文本格式编写，通过目录和文件管理；
2. 高级关键字随测试用例管理，不需要独立入库；
3. 自动化测试框架使用Robot Framework；
4. 从低级关键字到设备接口分为两层，一层为低级关键字，另一层为各种设备和环境的接口适配，以便屏蔽接口变化对关键字的影响；
5. 接口适配层可以和低级关键字库部署在同一台机器上，也可以和测试环境中的设备部署在同一台机器上；

# 需求描述

## 关键字库需求

### 对接网管

* + - 1. **需求和基本框架**

不同产品网管的连接方式、命令格式等，都有较大的差异，但是其背后的思路都是一致的：完成配置、读取信息。与不同网管对接的框架中，应该将各个组件间的耦合减小到最低，以便能够用统一的接口适应不同的网管，增强可扩展性。

为了减小耦合，网管的关键字库与网管服务器间，应该增加一层适配层，用来完成网管的连接方式和操作协议的适配。这一适配层屏蔽网管的传输协议特征，为关键字提供统一的操作接口，当网管传输发生改变，或增加新的网管类型时，关键字库可以不做变更。

关键字库中，提供网管命令的封装，将一个或若干个关联且有普遍适用性的命令封装成关键字，供测试用例使用。当网管命令发生改变时，关键字库中能够屏蔽这种改变，从而不会影响测试用例的可用性。



考虑到集中控制的需求，OMM接口关键字和适配层在物理上应该可以部署在网络中任意位置，包括测试PC和OMM Server中。低级关键字库需要同时提供本地调用和远程调用的功能。

* + - 1. **低级关键字库**

如前文所述，适配层和关键字库两层的分工，将测试用例与真实的网管命令做了屏蔽。适配层做连接管理等操作，关键字库对网管命令进行封装。测试用例中使用的，是关键字库中编写的封装后的操作。

当网管命令发生改变，则只需要修改关键字库中的命令封装，测试用例中调用关键字的方法不需要做任何修改。在网管命令发生改变时，可以使用工具进行扫描，通过工具辅助修改关键字库，提高修改关键字库变更的效率。

最小的低级关键字库中，应包含所有网管命令的封装。进一步的，低级关键字库还可以将一组相关联的命令组织在一起，封装成一个关键字，完成一项通用的操作。

关键字库中，还应该能够对网管的返回结果提供检查方法，并与对应的操作绑定。由于检查方法可能很多，因此在关键字库中仅提供基本的检查方法，用户可以在测试用例中扩展自己的检查方法以适应不同的需求。

关键字库还应该能够记录每个网管操作的逆操作，并提供单一接口供测试用例使用，一步完成所有网管操作的逆操作。

* + - 1. **OMM接口适配层**

适配层完成的功能包括网管的连接与释放、命令执行控制（单条、批量、循环、异步执行等）、命令执行结果判断等。

不管实际的网管类型是什么，接口适配层对关键字库提供统一的接口，实现网管连接、命令执行、结果检查功能，有适配层根据实际的网管类型，实现真实的传输层操作。

接口适配层提供的命令执行控制中，应能实现同步或异步调用。同步命令在适配层主线程中执行，异步命令则在子线程中执行。适配层还应该能够对命令执行是否成功做判断，成功的标志在关键字库中，针对不同的关键字分别定义。这里的成功判断仅作为传输层成功的判断，并不对命令的返回结果做语义解析。命令返回结果的语义解析有关键字库完成。

接口适配层还应该能够处理异常处理，包括网管中断、配置命令超时等。

* + - 1. **在测试用例中配置被测网元**

在测试用例中，使用关键字库中提供的各种操作对网管进行设置，并使用关键字库或自定义的检查对网管返回结果进行检查。

测试用例中使用的配置和查询操作，是由低级关键字提供的，这个接口相对稳定基本不会改变，网管命令的变化，由关键字库来屏蔽。网管类型和连接方式的变化，由适配层屏蔽。

恢复网管配置时，用户在测试用例中可以通过调用关键关键字库提供的接口，一次性恢复在本用例中所做的所有需要恢复的配置。

* + - 1. **与其他关键字库的配合**

在测试用例中，网管命令可能需要和其他操作组合来使用，如uMAC中有些动态管理命令会触发业务流程，这样的需求由测试框架来完成，不需要在关键字库中实现。

由于各个关键字库相互独立，因此各种操作的顺序完全由测试框架决定，只要相关的关键字库提供了异步操作的接口，就能够实现各个关键字库之间的并发，信息传递可以通过测试框架来转发，这样各个关键字库就不需要互相依赖了。

### 驱动测试工具执行业务

* + - 1. **测试工具的角色**

由于测试的调度、管理功能转移到了测试框架，因此测试工具只需要完成各类对接网元的模拟、模拟流程的编辑、消息编解码等工作即可。

工具中的模拟流程，将以库的形式独立保存，与测试用例解耦，在不同的测试用例中，可以同一个测试流程，并由测试用例为其设置不同的配置参数。测试工具与被测环境的对接参数，将是测试环境的一部分，需要同测试环境一起规划设计。

由于测试框架有远程调用的能力，测试工具可以集中管理，同一套模拟工具可以供多个测试PC共同使用。

为实现这些功能，测试工具需要提供操作接口，供关键字库及其适配层调用，接口的形式可以是API或其他网络传输接口。通过操作接口，能够在测试工具中加载指定的模拟流程、修改其参数配置，能够启动、停止测试流程，并能够获取执行完成后的详细信息，包括成功、失败等。

* + - 1. **测试工具接口适配**

测试工具目前主要是PSTT和Loadtest，以后可以将其他具有API接口的商用仪表也纳入自动化范围内。

上文已经描述了测试工具需要提供的接口需求，不同的工具基于自己的特征，提供的接口可能也有差别，因此，类似网管的框架，也需要引入工具适配层，对工具关键字库屏蔽工具的传输层差异。

适配层对关键字库需要提供工具连接、工具命令执行、数据采集等功能，针对不同的工具，适配层将这些操作转换为真实的传输层命令。

当工具的命令接口或命令格式发生改变，只需要修改适配层，不需要对关键字库进行修改，从而降低了关键字库与工具的耦合。

* + - 1. **测试工具关键字库**

测试工具的关键字库提供的操作工测试用例使用，其中屏蔽真实的操作命令以及工具连接、数据传输等细节。

测试工具的关键字库中，主要用于适配工具命令格式的改变。当命令发生改变时，仅需要修改关键字库中关于各个命令的实现部分，在测试用例中，这些改变是不可见的。

由于测试工具主要是自研的，因此工具的接口可以主动适配已有的测试用例，从另一个角度减少接口变化对测试用例的影响。对于第三方测试工具，一般都有API接口可供调用，可以将这些API封装成关键字，供测试用例使用。这些API相对稳定，其变化很小，即使有变化也能够通过关键字库屏蔽其影响。

在部署上，可以利用Robot Framework的远程调用能力，将工具部署在远程主机上，统一调度资源，从而简化测试环境组网。

* + - 1. **多个测试工具间的配合**

在测试过程中，不可避免的，会使用不同的对接网元，各个网元的业务流程需要配合起来，以满足不同的测试场景。在本需求的框架中，不同的网元可以使用不同的模拟工具来实现，通过RF的测试用例来控制不同的模拟工具间的消息序列，从而实现不同模拟工具间的配合。

### 数据采集

* + - 1. **基本框架**

数据采集的过程与网管、工具不同，这一过程将使用各种各样的接口，以便获取到需要的数据。可能使用的接口包括：数据库、FTP、Telnet、计费、抓包等等。不同的数据源，可能使用相同的接口，也可能使用不同的接口；另一方面，即使两个数据源使用相同的接口，如FTP，对数据的解析也是不一样的。因此数据采集模块需要区分接口适配与数据解析功能。在功能上，数据采集模块自上向下可以分为关键字库、数据分析、接口适配三部分。

* + - 1. **数据采集关键字**

数据采集关键字库提供的关键字，对测试用例屏蔽数据采集的各种细节。在测试用例中只需要描述需要采集的数据类型、位置、加工方法，就应该能够得到所需的数据。所有的处理细节，由关键字库对测试用例屏蔽。

另一方面，关键字库也应该相对稳定，因此关键字库也不需要直接操作各种数据源，而是借助接口适配与数据分析模块提供的操作来完成数据的获取。关键字库的实现中，主要是将各种基本操作组合在一起，完成某一特定的功能。

当需求发生改变时，关键字库对测试用例提供的接口保持不变，仅修改其内部实现。同样的，当数据采集与分析方法发生改变时，也不需要修改关键字库，仅修改底层对应的模块即可。

关键字库中，各种数据源需要按照功能组合在一起，不同的数据源，可能使用的协议一致，但是其功能不同，也应该将其划分为不同的分组。

* + - 1. **接口适配**

数据采集与分析过程，将面对各种各样的数据源，每种数据源都有各自的数据获取方法，因此需要对不同类型的数据源设计不同的操作。接口适配模块的目的就是对不同的数据源进行包装，方便处理。

在接口适配模块，主要获取数据的获取方法，将上层的调用转换为各种数据源真实的操作方法，并关注传输层是否成功。接口适配层不关注采集到的数据的语法与结构，而是将原始数据直接返回给上层，由上层调用数据分析提供的方法来实现数据的解析与加工。

接口适配层中，不针对数据源的功能进行分类，而是根据数据源的协议类型进行分类。如为所有的Telnet提供统一接口、为所有的FTP提供统一接口等。

* + - 1. **数据分析**

数据采集关键字驱动接口适配层获取到原始数据后，根据数据源的功能，可能还需要对数据进行分析，这一部分功能由数据分析模块来提供。数据分析模块中接收由接口适配层获取到的原始数据，经过各种处理，将加工过的有意义的数据返回给关键字使用。

数据分析模块中，需要根据不同数据源的功能来定制数据分析方法，与关键字库的组织一致，数据分析模块也按照数据源的功能来组织在一起。

### 基本辅助功能

* + - 1. **需求与基本框架**

辅助功能指的是在执行测试用例的主机本地上，执行的一些基本功能，如文件操作、本地数据库归档、数据分析计算、条件断言等，其他功能可以根据需求添加。

基本辅助功能的实现，可以根据功能的不同，分别确定各个功能的实现结构。各个功能间应该消除耦合。

* + - 1. **文件操作**

提供操作系统文件操作的关键字，支持目录、文件等各种操作。可以使用Robot Framework提供的OperatingSystem关键字库，也可以在其基础上进行扩展。

通过文件操作关键字，测试用例可以将数据、过程信息等写入文件中，以便进一步加工或者归档。

* + - 1. **本地数据库**

除文件外，辅助功能还应该提供将数据写入本地数据库的功能，通过使用数据库的机制，提供更有效的分析与检索能力。

在测试用例使用的关键字中，应该屏蔽表结构，只将用户需要关注的数据写入数据库，这样数据库结构的变更就不会影响测试用例，能够提高测试用例的复用性。

* + - 1. **数据分析计算**

从各种数据源以及测试用例输入得到的数据，可能需要经过各种加工计算，因此需要提供数据分析计算相关的关键字。这一组关键字中，应该包括统计计算、类型转换、排序、查找等各种通用的功能。这些关键字的接口应该尽量简洁，避免关键字的实现影响测试用例。

* + - 1. **条件断言**

判断各种条件是否符合预期，如果不符合预期则判定测试用例执行失败。可以使用Robot Framework内置的Assert族关键字，也可以对其进行扩展。

## 管理需求

### 测试用例与关键字管理

待定。（SVN？）

### 对接Onetest

待定。

### 持续集成

待定。

## 扩展功能

### 智能化（暂不实现）

* + - 1. **测试数据自动配置**
      2. **自动生成消息流程**

### 分布式（暂不实现）

* + - 1. **测试工具共享**
      2. **测试环境共享**