

关键字驱动框架中关键字划分方法研究

钱月琴

(南京邮电大学 计算机学院, 江苏 南京 210003)

摘要:文中针对目前关键字驱动框架中的关键字划分方法进行研究,提出了在eTOM(增强的电信运营图,enhanced Telecom Operations Map)业务过程分解思想指导下的关键字划分方法,建立了关键字划分模型,并根据关键字划分模型设计了关键字驱动框架中脚本划分模型(高层脚本对应高层关键字、中层脚本对应中层关键字和低层脚本对应低层关键字)。结束了对关键字驱动框架中关键字划分无理论研究的历史,提高了关键字驱动脚本的执行效率。最后,简单介绍了生成高层脚本、中层脚本和启动脚本的方法。

关键词:关键字驱动;关键字;测试脚本;eTOM

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2010)09-0044-04

Research of Keywords Decomposing Method in Keyword-Driven Framework

QIAN Yue-qin

(School of Computer, Nanjing University of Posts & Telecommunications, Nanjing 210003, China)

Abstract: Aiming at research of decomposing keywords in current keyword-driven framework, put forward ideas of keywords decomposing method based on the guidance of eTOM business process decomposition model, and then build a keyword decomposition model. Based on keywords decomposition model designed a script decomposition model in keyword-driven framework. Put an end of script decompose model theory-free in keyword-driven script technology of history and increase the execution efficiency of keyword-driver script. Finally, introduce the methods of generating high-level, middle-level, low-level and launch scripts in brief.

Key words: keyword-driven; keyword; test script; eTOM

0 引言

关键字驱动的思想一方面继承了数据驱动的理念,将测试数据同测试脚本相互分离;另一方面为了适应业务逻辑的灵活变化,将测试步骤中的业务逻辑也同测试脚本相互分离;另外还为方便对测试脚本的理解,实现了被测对象界面元素名和脚本内容对象名的分离^[1,2]。

目前国内外对关键字的划分是仁者见仁,智者见智,没有统一的参照标准^[3]。很多关键字的划分是根据研究者的喜好来决定的,目前还没有人对关键字的划分做理论研究,而关键字划分在关键字驱动技术中起着一个非常重要的作用。针对这个问题,笔者对关键字划分问题做了深入的研究。笔者通过对NGOSS(软件自动化框架支持, Software Automation Frame-

work Support)体系架构中的eTOM^[4]业务过程分解模型进行研究后发现,eTOM业务过程分解模型可以指导关键字的划分。因此笔者提出了一种基于eTOM业务过程分解模型的关键字划分方法,将被测对象和测试工具有机地结合起来,这更有利于测试的进行。

根据关键字的划分模型可以得到脚本的划分模型,文中在详细讲述了关键字划分模型之后,对关键字划分模型指导下的脚本划分作了仔细的分析。下面首先简单介绍下eTOM的业务过程分解思想。

1 eTOM 业务过程分解模型简介

eTOM业务流程框架采用自上而下层次结构的方法对企业运营相关的所有过程单元和活动进行层次化的描述,目前eTOM版本提供了四级业务过程分解:零级、一级、二级和三级过程。eTOM业务过程框架包括了企业面向客户开展服务的方方面面的过程元素,以及这些过程之间的复杂关系,分层次对整体框架进行描述有助于清晰地定义eTOM的整体框架。所谓

收稿日期:2010-01-10;修回日期:2010-04-13

基金项目:国家科技支撑计划项目(2007BAH17B00)

作者简介:钱月琴(1984-),女,安徽池州人,硕士研究生,研究方向为软件测试、计算机通信网络。

的层次结构就是从不同的角度描述 eTOM 业务过程框架,不同的使用者可以根据需求关注相应的重点,参考 eTOM 业务过程框架的不同层次。eTOM 的零级过程从最高层面给出了 eTOM 业务过程框架的概念,从总体上设定了 eTOM 的研究范围和关注的领域。eTOM 一级过程显示了整个企业过程的处理细节,对于 CEO、CIO 和 CTO 来说,更关注这个层次上的结构。eTOM 二级和三级过程在上一级的基础上,对每个过程进行了更细节的分解。目前的 eTOM 版本提供了框架中所有过程的二级过程分解,但只对某些主要过程提供了三级过程分解,而相对次要的一些过程则没有提供三级过程。对于具体的设备提供商,软件开发人员和系统建设人员会更加关注二级和三级过程中定义的详细内容^[3]。

2 基于 eTOM 模型的关键字划分模型

在关键字驱动的架构中,关键字贯穿始终,怎样根据一个关键字导出下一个关键字是关键字划分着重要考虑的因素。分层次对关键字驱动的整体框架进行描述,有助于对关键字驱动思想的理解以及调用相应的脚本^[5,6]。总体规划是将关键字划分为三层:高层、中层和低层,三层关键字分别与三层脚本相对应。

从上文对 eTOM 业务过程分解思想的描述可以看到,其主要思想就是从不同人员所关心的主要内容不同从高往下划分。该思想可以指导关键字的划分,在划分关键字时也从参与测试的不同人员的角度来考虑所需要的关键字来划分,具体如下:

首先,设计关键字的目的是为了更好地执行测试任务,所以参照 eTOM 业务过程分解的思想,从测试执行人员的角度来划分高层关键字。在测试执行人员看来,脚本的具体内容对他们没有影响,他们只需关心脚本对应的测试用例是什么,最后脚本执行的结果是什么就可以了。所以将测试用例的名称作为高层关键字,从总体上规定了脚本的内容范围。图 1 给出了按照 eTOM 业务过程分解的框架图思想得出的关键字划分的一级框架结构。

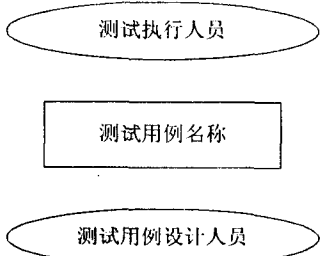


图 1 关键字第一级框架结构

根据 eTOM 业务过程分解的思想,高层关键字规

定了脚本的内容范围,再接下来中层关键字应该显示该内容范围的细节,对测试用例的设计人员来说,更关注的是这一级别的关键字信息。对 GUI 功能测试来说,测试用例中的主要内容就是测试步骤和测试数据,且用例中可能会包含一到多个步骤,所以将测试步骤作为中层关键字。这是从测试用例设计人员的角度考虑的,测试用例设计人员最关心的就是测试用例中的测试步骤以及测试步骤的执行顺序。依据这样的划分思想,可以看到,一个测试用例可能包含一个或多个测试步骤,所以一个高层关键字对应一个或多个中层关键字。图 2 给出了按照 eTOM 业务过程分解的思想得出的关键字划分的第二级框架结构。

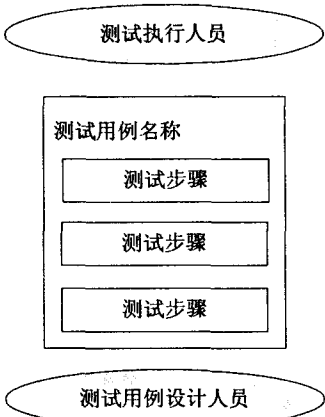


图 2 关键字第二级框架结构

在 eTOM 的业务过程分解中,对一些主要过程提供了三级过程分解,主要是方便对每个过程细节的了解。同样,在对关键字进行划分时,考虑到对测试步骤执行的难度以及脚本设计的方便,在中层关键字的基础上,对其进行划分,也即对测试步骤进行划分。在对 GUI 功能进行测试时,一个测试步骤会对应 GUI 界面上一个或多个操作,所以将“测试操作”定义为测试步骤对应的低层关键字。根据这样的划分,可以很容易地看到,一个中层关键字会对应一个或多个低层关键字。图 3 为测试步骤的第三级过程分解。每个测试步骤包含操作以及执行这些操作所需要的测试数据。

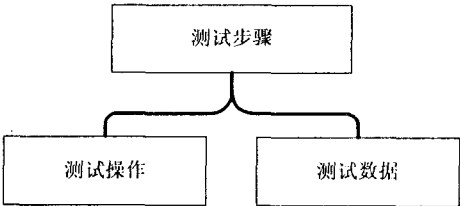


图 3 测试步骤的第三级过程分解

在本设计方案中,对 GUI 界面的操作是最低层的关键字,按照 eTOM 业务过程分解理念,再结合具体的测试过程,没有必要再往下细分关键字了,关键字划

分层次多了,可能起不到正面作用,反而还会给脚本的维护带来更大的开销。

对上面的关键字划分做一总结:高层关键字对应测试用例名称,中层关键字对应高层关键字中的测试步骤,低层关键字对应中层关键字中的测试操作。

3 基于关键字划分模型的脚本划分模型

在 SAFS 框架中,将测试脚本共分为三层,其实,这三层测试脚本对应着三层测试表,其内容也即原框架中三层测试表的内容^[7,8]。根据上文提出的关键字划分模型,可利用该模型指导脚本的划分:高层脚本对应高层关键字、中层脚本对应中层关键字和低层脚本对应低层关键字。下面将详细讲述三层脚本之间的关系及各个脚本所包含的主要内容,最后阐述了实现这三层脚本内容时所需要的命令。

高层脚本:是“测试用例级”的测试脚本,主要包含的内容是针对不同的测试目的所设计的测试用例组合。换句话说,一个高层脚本对应一个测试套件,一个测试套件包括多个测试用例,每个测试用例又对应一到多个测试步骤,而中层脚本所包含的关键字就是“测试步骤”,所以一个高层脚本会有一到多个中层脚本与其对应,也即一个高层脚本可以分解为一到多个中层脚本,是对多个中层脚本的组合。在 SAFS 框架中,提供了调用中层脚本的命令,只需按照该命令将中层脚本组合即可得到高层脚本。

中层脚本:是“测试步骤级”的测试脚本,主要包含的内容是对应高层脚本中每个测试用例所包含的具体的测试步骤的组合,为实现每个测试步骤,将要完成一到多个界面操作,例如输入文本框的值、点击按钮等操作,而低层脚本的关键字是“操作”,所以一个中层脚本会有一到多个低层脚本与其对应,也即一个中层脚本可以分解为一到多个低层脚本,是对多个低层脚本的组合。为实现中层脚本,只要在中层脚本中将多个低层脚本组合起来即可,换句话说,在中层脚本中提供调用低层脚本的命令以及相应的低层脚本的名称即可。在 SAFS 框架中提供了调用低层测试脚本的命令。

低层脚本:是“测试组件级”的测试脚本,主要包含的内容就是中层脚本中每个测试步骤的具体实现过程,按照关键字划分的思想,低层脚本将每个测试步骤的实现划分为实现该测试步骤所需要的操作过程和完成该操作所需要的参数,“操作”对应关键字划分中低层关键字。在具体实现过程中,低层脚本需要调用组件函数库中的关键字函数、应用映射表中的 GUI 界面对象,必要的时候会调用支持库中的内容来完成测试工作。

各个脚本所包含的内容以及脚本之间关系如图 4 所示。图 4 概括了脚本之间的关系并指出了各层脚本所包含的主要内容,在这需要说明的是:低层脚本和中层脚本都是可以复用的。在实现三层脚本的时候,按照该图的指示,从高层脚本开始逐步实现中层脚本和低层脚本。

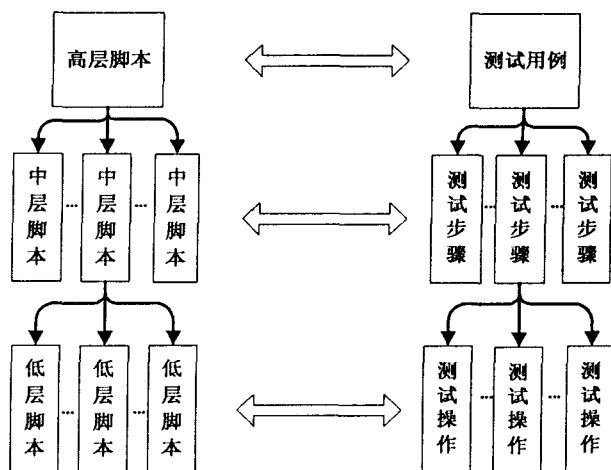


图 4 三层脚本所包含的主要内容及其对应关系

4 生成三层脚本的实现

根据上文介绍的脚本划分,已经清晰描述了三层脚本之间的关系,再利用脚本执行的思路来指导脚本的生成^[9~12]。在脚本执行时,首先在 Robot 中运行启动脚本,当启动脚本遇到调用高层脚本命令时,由高层驱动器调用高层脚本;在高层脚本的运行过程中,遇到调用中层脚本命令时,由中层驱动器调用中层脚本;在中层脚本的执行过程中,遇到调用低层脚本命令时,由低层驱动器调用低层脚本执行。所以在生成脚本时,首先生成高层脚本,其次生成中层脚本,然后手工编写低层脚本,最后生成启动脚本。

图 5 给出了生成三层脚本的流程图(阴影部分表示的脚本是自动生成的)。首先根据从测试套件中解析的高层脚本所需要的信息和高层脚本模板生成高层脚本;其次,根据从测试套件中解析出来的中层脚本所需内容和中层脚本模板生成中层脚本;最后,根据从测试套件中解析出的测试套件名称和启动脚本模板生成启动脚本。将得到的各个脚本保存到文件系统中。

高层脚本的主要任务就是将测试套件中各个测试用例组合起来。生成高层脚本的方案如下:首先根据 SAFS 框架对高层脚本的要求,定义一个高层脚本的模板文件,按行读取高层脚本模板内容,并将高层脚本模板文件中相应内容写入到缓存中,然后将从测试套件中解析出来的相应内容动态地根据高层脚本模板内容加载到缓存中,最后将缓存中内容全部写入到一个

文件中并将该文件命名为测试套件名称加上高层脚本扩展名.CDD,将该文件保存到文件系统中,该文件也即为生成的高层脚本。一个测试套件对应一个高层脚本,而一个测试套件通常包含多个测试用例,所以,高层脚本是由多个测试用例组合而成的,在高层脚本中只需要调用所需要的测试用例即可。

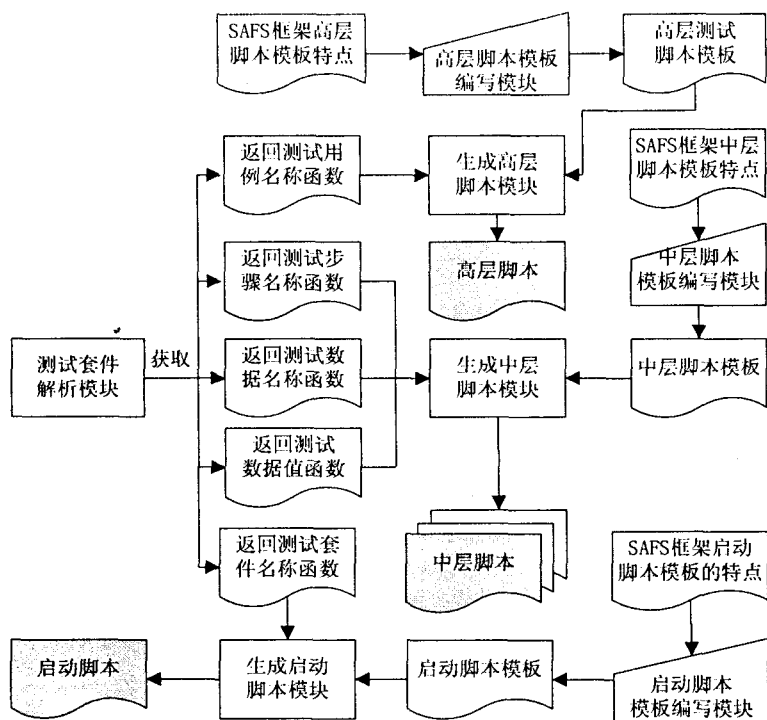


图5 生成三层脚本总流程图

生成中层脚本的方案是:首先根据SAFS框架下中层脚本的特点,定义一个中层脚本的模板,将中层脚本中一些不变的量写入到中层脚本模板中;然后读取中层脚本模板的内容并写入到缓存中,将从测试套件中解析出来的测试步骤、测试数据和预期结果验证动态地加载到缓存中相应的位置;最后将缓存中内容全部写入一个文件中,并将该文件命名为解析出来的测试用例名称加上SAFS框架下中层脚本的扩展名.STD,保存文件到文件系统中。在这里注意的是,一个测试用例对应一个中层脚本,所以一个测试套件可能会生成多个中层脚本。在SAFS框架下,中层脚本主要用来调用相应的低层脚本,每个中层关键字都对应一个低层脚本。

低层脚本对应低层关键字,也即其主要包含的内容就是测试操作,以及实现这些测试操作所需要的测试数据。在实现某个测试操作时,还需要考虑该操作是针对哪种类型的组件、该组件属于哪个窗口等信息。笔者还没有实现低层脚本的自动生成,低层脚本需要手动编写,主要原因是无法直接捕获到界面上的对象,纵然现在有工具可以捕获到界面上的对象,但捕获的

对象不能直接加载到脚本中,脚本中的对象是从应用映射表中获取的,所以将这两者结合起来有一定的难度。这也是笔者接下来将重点研究的内容,争取做到低层脚本也能自动生成。

在高层脚本和中层脚本都生成之后,接下来就是对这些脚本的执行了。启动脚本的主要任务就是使一

系列的高层、中层和低层脚本运行起来,并将运行结果保存到测试日志中。所以启动脚本中包含的内容还是挺多的,且启动脚本是在IBM Rational Robot中运行的,所以启动脚本所采用的语言是SQABasic。SAFS框架允许从高层脚本开始执行,也允许从中层脚本开始执行,具体的要看测试任务是怎样要求的。所以在笔者设计与实现中,首先判断测试套件中是否只有一个测试用例,如果只有一个测试用例,就没必要先执行高层脚本,然后由高层脚本调用中层脚本了,而是在启动脚本中直接调用中层脚本,这样节省了调用开销,提高了脚本执行的速度;若有多个测试用例,则从高层脚本开始运行。生成启动脚本的思想与生成高层和中层脚本的思想是一样的。先根据SAFS框架下启动脚本模板的特点及SQABasic语言的特点,制定一个启动脚本模板,将启动脚本中固定不变的量先写入模板中,然后读取启动脚本模板的内容到缓存中,并将从测试套件中解析出来的相应变化的量添加到缓存中相应地方,最后将缓存中的内容全部写入到以测试套件名称命名并加上扩展名.rec的文件中,保存该文件到文件系统即得到生成的启动脚本。

5 结束语

文中提出基于eTOM业务过程分解模型的关键字划分模型,该模型能有效地指导关键字驱动框架下测试脚本的划分,使人们能对关键字驱动框架有一个更清晰的认识,最为重要的是提高了脚本的复用性。最后,简单介绍了SAFS框架下三层脚本的实现方法。

参考文献:

- [1] Atif M, Adithya N, Xie Qing. Automating regression testing for evolving GUI software[J]. Journal of Software Maintenance and Evolution, 2005, 17(1): 27-64.
- [2] Peters D K, Parnas D L. Using Test Oracles Generated from Program Documentation[J]. IEEE Transaction on Software Engineering, 1998, 24(3): 161-173.

表 2 参数值和支持向量个数

模型	C	σ	ϵ	nSV
SVR	400	12	0.01	16(61.5%)
RS & SVR	40	11	0.01	18(69.2%)

表 3 电力产量预测值与实际值

年份	SVR	RS & SVR	实际值
2004	22230.19	21958.09	22033.09
2005	25577.35	25293.83	25002.60
2006	28815.96	28560.00	28657.26
2007	32684.36	32924.35	32815.53
2008	34448.61	35163.74	34668.82

单位:亿千瓦小时(100 million kWh)

表 4 模型预测精度比较

模型	RMSE	THEIL-IC	MAPE	MAX-APE
SVR	303.3361	0.0023	0.9564	2.2987
RS & SVR	267.0918	0.0021	0.7208	1.4276

其中:

(1) $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$

(2) $THEIL-IC =$

$\sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} / \left(\sqrt{\frac{1}{n} \sum_{i=1}^n \hat{y}_i^2} + \sqrt{\frac{1}{n} \sum_{i=1}^n y_i^2} \right)$, 希尔不等系数通常介于 0 ~ 1, 且值越小表明预测精度越高;

(3) $MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \times 100 \right|$, 一般认为 MAPE 的值小于 10, 则预测精度较高;

(4) $MAX_APE = \max \left(\left| \frac{\hat{y}_i - y_i}{y_i} \times 100 \right| \right)$

以上式中 \hat{y}_i, y_i 分别为预测值和实际值。

从支持向量个数来看,RS&SVR 模型与 SVR 模型大体相当,前者仅比后者多了两个支持向量;从预测效果来看,RS&SVR 模型具有更小的均方根误差和希尔不等系数,说明预测值更接近实际值,拟合效果更好。因此,RS&SVR 模型用于回归预测,方法是有效的。

4 结束语

支持向量机回归应用于电力供应量分析是一种较理想的模型,而粗糙集理论在处理不确定知识、消除冗余信息方面具有优势,将属性约简方法应用于数据的预处理,简化了样本的特征维。两者的结合比单一模型产生了更好的结果。相信粗糙集理论方法与支持向量机的结合在应用领域具有广阔的前景。

参考文献:

[1] Vapnik V N. 统计学习理论的本质[M]. 张学工,译. 北京:清华大学出版社,2000.

[2] 施燕杰. 基于支持向量机(SVM)的股市预测方法[J]. 统计与决策,2005(4):123-125.

[3] 滕卫平,俞善贤,胡波,等. SVM 回归法在汛期旱涝预测中的应用研究[J]. 浙江大学学报:理学版,2008,35(3):343-347.

[4] 王晓红,吴德会. 基于 WLS-SVM 回归模型的电力负荷预测[J]. 微计算机信息,2008,24(4):312-314.

[5] 程伟,张燕平,赵姝. 支持向量机在粮食产量预测中的应用[J]. 安徽农业科学,2009,37(8):3347-3348.

[6] 姜德民,王磊,徐义田,等. 基于粗糙集理论与支持向量回归的预测模型[J]. 统计与决策,2008(10):32-33.

[7] 王国胤. Rough 集理论与知识获取[M]. 西安:西安交通大学出版社,2001.

[8] 王定成,方廷健,高理富,等. 支持向量机回归在线建模及应用[J]. 控制与决策,2003,18(1):89-91.

[9] Cristianini N, Shawe-Taylor J. 支持向量机导论[M]. 李国正,王猛,曾华军,译. 北京:电子工业出版社,2004:98-104.

[10] 苗夺谦. Rough Set 理论中连续属性的离散化方法[J]. 自动化学报,2001,27(3):296-302.

[11] 袁卫,庞皓,曾五一,等. 统计学习题与案例[M]. 北京:高等教育出版社,2006:236-241.

[12] Gunn S R. Support vector machines for classification and regression[R]. Faculty of Engineering, Science and Mathematics, School of Electronics and Computer Science, University of Southampton, 1998.

(上接第 47 页)

[3] 侯勇. 关键字驱动的自动化测试系统的研究[D]. 西安:西安电子科技大学,2006.

[4] 徐志发. 从 eTOM/SID 看电信商业智能系统的发展思路[J]. 电信科学,2008(1):17-21.

[5] 徐振良,樊滨温,王志鹏. 关键字驱动技术在 SAFS 中的研究[J]. 软件时空,2006,22:270-272.

[6] 冯玉才,唐艳,周淳. 关键字驱动自动化测试的原理和实现[J]. 计算机应用,2004,24(8):140-142.

[7] 凌永发,张云生,郭秀萍. 软件测试自动化中的脚本技术[J]. 云南民族学院学报,2002(11):544-548.

[8] 芦彩林,丁刚毅. .Net 框架下测试脚本自动生成技术研究

[J]. 微计算机应用,2008,29(5):101-104.

[9] 赵斌飞,刘磊. 测试脚本自动生成器的设计与实现[J]. 计算机科学,2008(6):276-279.

[10] Fewster M, Graham D. 软件测试自动化技术与实例详解[M]. 舒智勇,包晓露,焦跃,等译. 北京:电子工业出版社,1999:450-560.

[11] Mosley D J, Posey B A. 软件测试自动化[M]. 邓波,黄丽娟,译. 北京:机械工业出版社,2003:440-442.

[12] Yuan Xun, Atif M. Generating event sequence-based test cases using GUI runtime state feedback[J]. IEEE Transactions on Software Engineering, 2010, 36(1):81-95.