

终端自动测试引擎 (Terminal Auto Runner)

产品技术白皮书

上海泽众软件科技有限公司

2008 年 6 月

目录

1. 总述	3
2. 系统概述	6
2.1 系统定位	6
2.2 自动功能测试工具的概念	6
2.3 产品设计目标	7
3 系统体系结构特性要求	9
4. 系统基本功能	11
4.1 产品原理	11
4.2 产品功能说明	12
4.3 外部功能扩展	错误！未定义书签。5
5. 厂商支持能力	145

1. 总述

一、背景

随着软件规模的发展和对软件系统的依赖，人们发现：软件的质量对应用系统的影响日益增加，质量存在问题的软件会导致帐务出错，客户信息丢失，用户的服务出错。

因此，提高软件的质量成为一个重要的问题。而测试正是提高软件质量的有效手段。数据显示，在一个软件开发过程中，测试占到整个工作的 40%—60%。所以，如何能够在比较低成本的情况下大幅度提高测试的质量，对软件的最终质量起到非常重要的作用。

另一方面，当应用软件投入使用之后，随着应用的不断发展和变化，将会提出大量的新增需求。新功能对用户非常重要，能够给用户不断发展的业务提供更强大的支撑。

当开发人员修改软件的功能、增加软件功能，新增功能部分导致原有系统运行不够稳定的几率必然增加，可靠性降低：由于修改一个小错误造成大量业务无法正常运行的情况。这就需要做大量的回归测试来保证系统的可靠性，通过回归测试验证以往的功能是正确的、可靠的。

传统的回归测试是由人工来实现的，需要大量的人员来完成固定的输入和输出检查。

人工测试带来了一系列问题：

1、测试质量难以保证。临时参加测试的人员很多都是没有受到足够培训的人员，对应用软件本身的理解不够充分，对测试技术也不够了解，往往导致测试案例设计不够严密，测试的质量难以得到保证。

2、测试的成本很高。组织一次大规模的回归测试会导致大量成本发生：包括人工成本和管理成本等等。首先是测试人员自身的成本，然后是由于测试进度比较缓慢造成的开发人员延误造成的成本，此外，管理测试人员，协调测试和开发人员之间的关系也需要很多的工作和成本。

3、测试进度难以控制。由于人员、系统稳定性等众多方面的原因，导致测试的进度比较难以度量和控制，造成项目风险防范困难。

4、测试案例无法保存和管理。作为一个项目，测试案例是一个重要的财富：同软件代码一样，是具有版本和价值的。传统的做法是使用文档来保存测试案例，无法提供一种标准格式来保存测试案例。这样就会导致测试案例的存放非常困难，使用非常困难，造成了资源的浪费：但需要再做一次回归测试的时候，往往原来的案例都被丢弃了，还需要重新设计和完成测试案例。

二、解决办法

企业可以建立一整套软件自动测试体系，包括：需求管理、测试分析、测试管理、缺陷跟踪，并且把这个过程纳入整个软件项目开发和软件产品开发过程。

实际上，在 CMM 的规范中，测试本身就是 SQA 的一部分。

自动测试的基础就在于测试工具，只有采用了优秀的自动测试软件，才能够解决自动测试的问题。自动测试工具能够在两个阶段给软件开发企业带来价值：第一，对于软件开发人员来说，高效率的自动测试工具能够提供给程序员自己完成开发过程中的冒烟测试，便于在频繁修改的软件过程中迅速完成测试，保证编码的稳定性；第二，对于项目和产品的测试阶段来说，能够提供稳定的回归测试，保证产品的可靠性。

众所周知，在测试阶段发现问题的投入，相对与在软件投产之后出现错误再去解决问题要小的多。

目前的测试工具，基本上都是国外的产品，难以满足中、小型的软件公司的需要。正是在这种情况下，上海泽众软件科技有限公司开发出了国内第一个拥有自主知识产权的自动测试软件——自动测试引擎(Terminal Auto Runner)。

三、概述

1、本技术白皮书适用于上海泽众软件科技有限公司自动测试管理工具（TAR）。

2、本技术白皮书是上海泽众软件科技有限公司自动测试工具（TAR）的技术说明，也是技术谈判的主要内容，是采购方询价、系统选型以及系统测试和验收的主要技术依据。

3、本技术白皮书是根据信息产业部颁布的有关技术体制和技术政策并结合上海泽众软件科技有限公司的实际情况制定的。本技术白皮书没有提出而信息产业部的技术体制以及技术标准已有具体规定的内容，应按信息产业部的技术体制以及技术标准执行，如果存在不一致应以信息产业部颁布的最新技术体制及技术标准内容为

准。

4、本技术白皮书在内容或技术指标上如果存在错误（包括印刷错误），经双方确认后可对该错误内容或技术指标进行修正。

5、自动测试工具（TAR）版本升级之后，上海泽众软件科技有限公司有权对本技术白皮书进行修改，并不需要主动通知用户。

6、本技术白皮书以下内容为用户重点考察内容：

软件的功能、性能、技术指标和环境要求；

设备容量计算和配置方法；

所提供的数据库的功能和性能指标；

软件安装要求；

提供软件的接口、协议等工程技术要求；

乙方供货范围、交货能力和时间、运输、安装、调测验收和培训等项内容的日程安排；

其他有关技术资料。

7、本软件对涉及专利、知识产权等法律条款承担有限责任。

8、本技术白皮书提供了对上海泽众软件科技有限公司的自动测试工具（TAR）的相关技术描述，由于用户使用造成损失，上海泽众软件科技有限公司不承担责任。

9、本技术白皮书以中文编写，未经上海泽众软件科技有限公司同意或授权的其它语言或形式的技术白皮书无效。

10、本技术规范书的解释权归上海泽众软件科技有限公司。

2. 系统概述

2.1 系统定位

TAR (Terminal AutoRunner) 是上海泽众软件科技有限公司开发的, 具有自主知识产权的、面向终端系统的回归测试工具。TAR 支持字符终端 VT100 的功能测试, 能够支持命令行模式和窗口模式。

2.2 自动功能测试工具的概念

- 测试脚本

自动测试, 就是使用一个程序来测试另一个程序 (被测试的应用系统) 功能的正确性。如果用来测试的程序本身非常复杂, 也需要被测试, 或者编写困难, 那么自动测试就失去了意义。

因此, 用来测试另外一个程序的程序往往是非常简单的, 我们把这个程序称为“测试脚本”。

测试脚本通常在测试工具的 IDE 里执行, 并且获得 IDE 的支持。

- 自动记录

当我们编写测试脚本的时候, 往往发现编写脚本本身是很困难的: 了解脚本的语法、了解测试过程、把测试过程转换称为测试脚本语句。

自动记录, 就是通过记录一个操作过程来获得测试脚本的功能。通过自动记录, 我们就能够得到一个操作的基本的脚本, 通过修改这个脚本, 我们得到更通用的测试脚本。

- 同步点

在执行测试脚本的时候, 测试脚本语句的操作对象是 GUI 的组件。测试脚本通过这个组件的属性 (如: 名称、位置、winclass、disable 等) 来确定那个组件是我们需要操作的组件。

这个查找组件的过程如果失败, 意味着: 第一, 应用系统的响应比较慢, 需要等待一段时间再进行一次定位; 或者第二, 该组件不存在。

这个查找、定位组件的过程, 我们称为同步点。

AutoRunner 的同步点都是隐含方式的：在操作对象的时候进行自动同步，自动设置同步点。

● 检查点

测试的目的是检查数据是否正确。

在测试的过程中，我们需要检查某个组件的某些属性满足某个条件。这个检查的位置和条件，我们称为检查点。

在 TAR 中，使用 `checkpoint`，`checkText` 来作为检查点的脚本语句，它检查对象 `objectname` 的属性 `property` 是否和期望值一致。

在使用使用中，可以使用检查点来检验系统的各个方面，如数据库、GUI 属性等。

● 参数化与数据驱动

测试脚本是针对一个测试过程的。一个测试过程往往需要众多的数据来测试。通过自动录制得到的脚本，所有的输入数据都是常数，是固定的。

如果需要使用一个测试脚本测试多组数据，就需要对脚本进行参数化，把固定的常数修改为来自数据源变量。

这个过程我们称为参数化。

采用了参数化的脚本，我们称为数据驱动的模式。

2.3 产品设计目标

完成应用系统的功能测试和压力测试。

1、提高测试质量。通过设计严密的测试案例，可以达到比人工测试严密许多的测试案例，完成对系统的严密测试，大幅度提高测试的质量。

2、测试案例库支持。通过一次测试，得到一个测试案例库，并且把此测试案例库作为基线。在维护的过程中，只需要维护这个案例库就可以容易的实现快速测试、子系统回归测试、整个系统回归测试。

3、缩短测试时间，降低人员成本的测试的时间成本。自动测试要求在很短的时间内完成测试，并且不需要大量的测试人员因此降低了成本，并且由于测试的时间

短，也降低了整个开发过程中的机会成本。

TAR 产品能够自动完成基于终端系统应用的自动测试软件，特点如下：

测试案例和测试数据存放。TAR 采用.csv 文件和 xml 格式来存放测试数据和资源文件，也能够把测试结果以报表志的方式存放。存放的格式能够通过测试工具直接被执行，已经不再是一个文档，而是把文档和可执行的内容合并在一起。

案例自动执行。被存放的案例和测试数据可以被多次自动执行，不需要人员干预。测试人员只要预先定义好测试流程，就可以自动执行测试案例，完成自动回归测试过程。

测试案例可以被多次重复使用。当测试案例和测试数据建立之后，可以多次重复这个数据和案例。产品会保证每次都按照预先定义的顺序完成测试，从而保证了系统能够测试多轮。

测试的质量得到很好的保证。由于测试是由计算机自动完成的，输入数据的准确性能够得到非常高的保证，同时输出数据的校验也是通过计算机自动实现的，因此测试的质量远远高于人工测试。

测试的进度和时间得到保证。人工测试的时间是比较长的，只一点毋须讳言。自动测试工具由软件模拟人工不间断的输入，速度比人工快几十倍，能够在一夜之间完成所有的回归测试工作，第二天就可以提交错误报告，帮助开发人员定位错误。从而大大提高开发效率。

测试案例累加。测试案例本身也是有版本的，一个测试案例和一个应用系统的版本能够保持一致。当应用系统改进的时候，测试人员也可以同时改进测试案例，对测试案例进行累加，保护测试案例资源。同源代码一样测试案例也是一种资源，具有很高的价值，需要被保护。TAR 提供了一种方式能够很好的保护这种资源不被浪费。

3 系统体系结构特性要求



产品的体系结构分成两个层次：仿真系统层、运行和 IDE 层。下面，按照五个模块的功能来说明各个功能模块的具体功能，以及和其他模块的接口：

- **仿真终端系统**

TAR 软件的测试对象就是终端上面的应用。在终端系统上编写测试案例是比较困难的和难以使用的。

TAR 产品运行在 Windows 系统上，提供了对于终端的界面——仿真终端。在这个系统上运行终端上面的应用，感觉和在终端系统相同。

TAR 的仿真终端和一般的仿真终端软件不同，它是仿真了真实的硬件，从串行口到其他的硬件命令。

同时，仿真终端支持标准输入、中文输入和显示、虚拟设备接入等许多功能。

- **虚拟设备子系统**

运行在终端系统上的软件往往需要支持对设备的操作，从磁卡读写器到密码键盘再到票据打印机等。

虚拟设备子系统就实现了以上的几种设备——虚拟打印机、虚拟密码键盘、虚拟磁卡读写器。

虚拟打印机把实际设备打印在纸张上面的内容输出到文件中；

虚拟磁卡读写器把磁卡的内容读入、写出到虚拟的磁卡中（磁卡由文件来虚拟）；密码小键盘虚拟了实际的键盘，把输入专接到实际的键盘上面来。

虚拟设备的实现，使得终端上面的测试不再需要硬件环境，可以减少用户在测试环境方面的投资，另外也使得高速的测试能够持续进行，不被频繁的设备操作打

断。

● 资源交互定义子系统

测试过程是由测试脚本来控制的。测试脚本实际上是一个特殊的程序，自动测试过程就是使用自动测试程序来测试应用程序。

TAR 的资源文件支持“所见即所得”的模式，用户可以使用 TAR 来浏览用户测试案例所可见的各个窗口界面，不需要使用复杂的定栏位位置等方法。

测试脚本是通过访问资源文件来控制具体的屏幕组件，如同步、输入等工作。

让测试人员来编写严谨的测试脚本是非常困难的，它需要一定的编程基础、一定的培训才能够达到，这会增加用户在测试方面的开销。

TAR 可以通过录制的方法来得到一个完整的资源文件。这个资源文件可以被用户通过 TAR 的“案例编辑器”来查看，并且对屏幕上的一些同步内容进行交互式修改。

修改完毕的资源文件可以使用 TAR 的一个功能，把资源文件转换为测试脚本，从而提高了编写脚本的效率，降低了出错概率。

● 自动录制子系统

终端上的应用系统使用的是窗口的方式，并且支持如栏位、选择列表、表格、对话框等组件。

自动录制的功能能够使用户自己执行一次交易之后，就能够得到一个测试脚本。自动录制的脚本能够识别出所有的组件，并且能够马上被执行，提高了用户编写测试案例的效率。

TAR 目前提供了两种录制测试案例的方式：第一，动态录制。根据用户的操作来录制用户所有用到的屏幕，并且识别出屏幕上的组件；第二，静态录制。用户可以点击“录制”按钮，录制当前的屏幕，把当前的屏幕识别处理作为资源文件。

● 脚本自动执行

测试脚本被创建之后，就需要被执行。脚本执行子系统能够自动完成脚本的执行。

TAR 采用 java 作为标准的测试脚本语言，提高了测试系统本身的扩展性。

地址：上海市宁夏路 201 号绿地科创大厦 24 层 DE 座

Web: www.spasvo.com Tel: 021-31357887

4. 系统基本功能

产品的功能比较复杂，首先从产品的原理讲起。

4.1 产品原理

回归测试工具是黑盒测试工具，它不同于白盒测试工具，它不关心应用系统内部如何连接，如何工作，它只关心：有这样一个输入，应该有另外一个预期输出。如果预期输出不同，它就认为测试失败了。

从测试理论的交互来看，TAR 产品实现了针对终端测试软件的几个功能：录制、扩展、回放。

首先，TAR 完成录制的功能。测试人员把每一个功能都切分出来，成为一个个测试案例。测试案例对用户来说，就是一系列屏幕交互（和设备交互）。通过 TAR 的 IDE 环境，能够把一个测试案例过程录制下来。当回放这个案例的时候，TAR 系统模拟所有的键盘输入和设备输入，接受所有的输出，同时根据验证点来同步系统。

TAR 支持自动录制。具体的功能见上文的描述。另外，对特殊对话框的识别，TAR 支持使用配置文件的方式来确定哪些是需要识别的对话框，能够提高识别率，降低手工修改的部分。

参数化。当录制完成一个测试案例之后，测试人员得到的是一个测试过程的模版。如果所有的过程都需要录制，那对于用户来说，功能就过于复杂了，需要录制的案例也太多，以至于无法使用了——建立参数化的测试案例脚本过程比测试一次的工作量要大很多。TAR 提供了案例扩展功能，使得用户只需要对一个测试案例之需要录制一个过程就可以完成，而其他相同的只需要修改 csv 文件里的数据就可以实现。这样的结果是，只创建一个案例过程，其他的内容都是通过数据来实现的。从测试方法论的角度上考虑，达到了脚本与数据分离。

回放。执行脚本的过程，实际上就是回放过程。

4.2 产品功能说明

所有的这些功能，都是建立在：录制、扩展、回放基础上的。TAR 提供了方便的 IDE 集成开发执行环境，便于用户方面的创建测试脚本和测试数据。

TAR (Terminal Auto Runner) 终端自动测试引擎，提供了强大的功能，能够满足用户非常具体的要求：

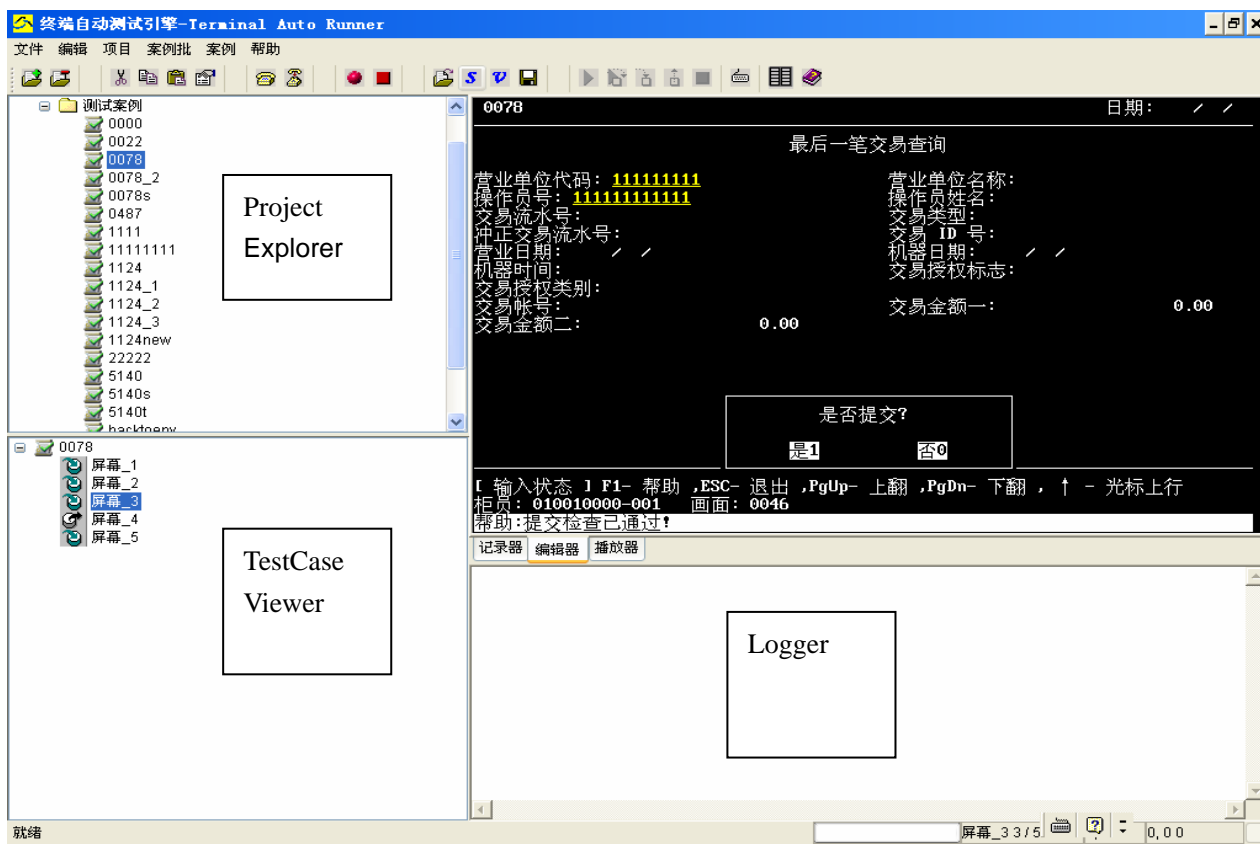
- 所见即所得的交互定义界面。

使用方便是产品的最大特色。

产品的使用对象是测试人员，不需要复杂的编程知识就可以自己完成案例录制等工作。方便的工具也使用户维护测试案例变得非常容易。

方便的使用最终使用户的二次开发成本降低为零——不需要进行产品的二次开发。

界面的方便性大大降低了用户的使用成本。



- 提供了虚拟设备，提供对设备的测试，提高了自动测试和回归测试的可测试范

地址：上海市宁夏路 201 号绿地科创大厦 24 层 DE 座

Web: www.spasvo.com Tel: 021-31357887

围

虚拟设备的提供，从两个方面提高了测试的效率和产品的可用性：

1、被测试应用。被测试系统往往支持这些设备的操作，如果测试工具不支持虚拟设备，就需要修改应用来满足测试工具的要求，当测试完成后又需要修改测试对象，来适应真实的情况。TAR 的虚拟设备不需要修改被测试系统，降低了修改系统的风险。

2、高效率。设备的操作需要人工干预，同时操作的速度很慢。是用实际的设备，会把测试过程变得很长，也很复杂。虚拟设备让测试过程非常的连贯，高效。

- **高可靠性自动录制，极大地提高了用户编写测试案例的效率。**

TAR 提供了高可靠性的自动录制算法，能够很大程度上保证用户通过录制就可以建立稳定可靠的脚本和资源文件。

为了提高识别效率，TAR 提供了默认对话框的预定义配置文件，通过预先定义对话框来提高识别效率。

通过自动录制的可靠性提高，可以得到高质量的测试脚本。

- **开放性。支持动态 java 类调用，灵活的扩展功能**

产品具有良好的开放性。

TAR 支持标准的 java 语法，支持对外部包的调用，任何需要扩展的功能都可以在 TAR 语法之下来实现，从而提高了系统的开放性。

对于一般的测试脚本，TAR 可以通过定义资源文件再生成的方法非常容易的得以实现。这种方法的优势在于测试脚本的可靠性高，开发效率高。

通过对自动生成脚本之上的修改和扩展，使得 TAR 的测试脚本能够支持对第三方包的调用、对其他测试案例的调用（比如标准模块）。

通过这种方式，TAR 很好的把开发的简便性和开放性性结合在一起。

通过这样的方式，能够满足各个方面的要求：绝大多数情况通过配置来实现，对测试人员要求很低；对极其特殊的情况，通过外部脚本来实现，实现了灵活扩展产品的功能。

- **自动识别。能够提供页面级别的自动识别功能。**

- **测试案例 checkpoint 的自动设置**

测试案例编写的很多问题设置验证点，或者叫做同步点，也就是你应该把一

个输入输入到应用应该输入的地方，而不是输入到另外一个地方或者不需要的地方。

无论哪两种结果，都是导致输入输出错位，系统混乱。

为了防止这种情况出现，都需要在测试案例中增加许多“同步点”或者称为“验证点”。

TAR 产品在录制的过程中，已经自动增加了屏幕验证点、栏位输入验证点等，这些都是自动实现的，用户甚至不需要了解。自动同步点的方式，极大地增强了用户的可用性，提高编写测试案例的效率。

- **自动截取屏幕。**

能够在完成测试的同时把实际的屏幕截取下来，供业务人员核对。

截取屏幕提供了对测试过程的重现：实际的输入是什么、实际的输出是什么，设备输出的是什么。

这些信息作为测试记录存放，可以查找错误和定位错误，也可以保证测试工具本身的可靠性。

5. 厂商支持能力

提供完备的用户手册，管理员使用手册，系统技术手册并再系统升级后及时修改更新服务。

厂商能够根据在实际应用中的问题，迅速给予解答（2 小时内），并给出解决方案（48 小时内）。