

47 | 用机器设计测试用例：基于模型的测试

2018-10-15 茹炳晟

软件测试52讲

[进入课程 >](#)



讲述：茹炳晟

时长 14:14 大小 6.52M



你好，我是茹炳晟。今天我和你分享的主题是：“用机器设计测试用例之基于模型的测试”。

我在前面 4 篇文章中，和你分享的探索式测试、测试驱动开发 TDD、精准测试，以及渗透测试的内容，你是否已经掌握了呢？有没有尝试将这些比较新的理念用到你的工程项目中呢？如果你在应用的过程中，遇到了任何问题，也欢迎给我留言一起讨论。

那么，现在我们就正式开始测试新技术系列的最后一个话题：基于模型的测试。

可以说，软件测试是一款软件产品质量的最后一道防线，是产品上线前必不可少、最重要的一个环节。每一款高质量的软件产品背后，都蕴含了大量的测试工作。而且，这些测试工作很可能是整个软件开发过程中最昂贵、劳动最密集的工作。

虽说从最简单的功能性黑盒测试，到涉及定理证明的复杂测试，已经有很多种方法可以帮助我们提高测试的可靠性和有效性。但是，在设计测试用例的过程中，总还是存在着这样那样的问题，使得软件测试的结果没那么理想。

为此，我们新引入了基于模型的测试，即 Model-Based-Testing，简称 MBT。

MBT，是自动化测试的一个分支。它是将测试用例的设计依托于被测系统的模型，并基于该模型自动生成测试用例的技术。其中，这个被测系统的模型表示了被测系统行为的预期，也可以说是代表了对被测系统的预期。

从质量保证的角度来看，我们可以制定测试内容，但却无法保证测试会覆盖所有可能的组合。而 MBT 则允许软件开发人员和测试人员，只关注建立系统的正确性以及模型的规范性，再通过专门的 MBT 工具根据不同的测试用例设计策略从系统模型生成可靠的测试用例。

那么，MBT 的原理是什么，而什么样的应用又适合进行 MBT 呢？接下来，我就重点为你回答这两个问题。

MBT 的基本原理

MBT 的基本原理是通过建立被测系统的设计模型，然后结合不同的算法和策略来遍历该模型，以此生成测试用例的设计。

我用下面的一张图片，为你描述了 MBT 的过程：

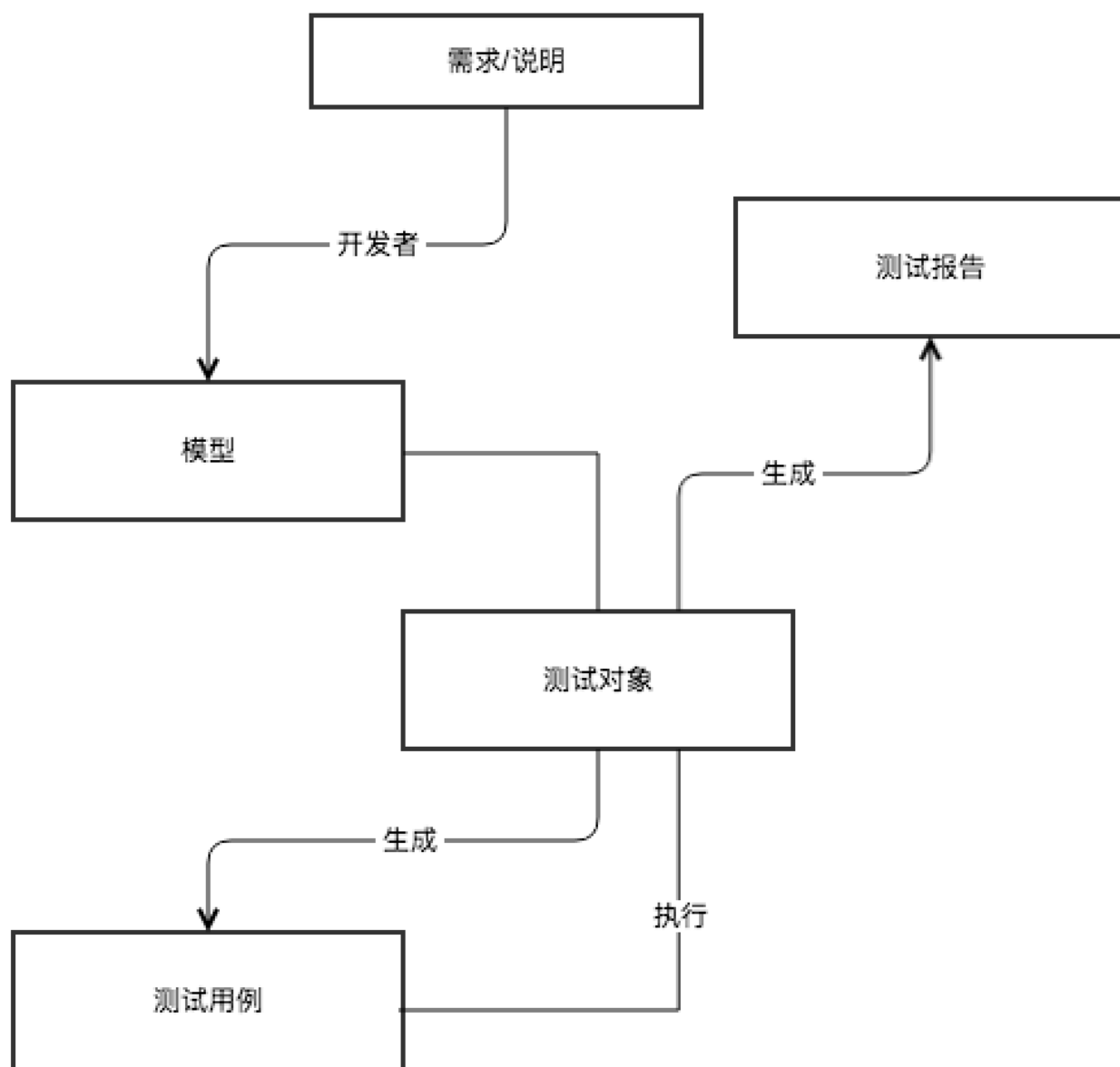


图 1 MBT 的一般过程

如图 1 所示，开发者首先根据产品需求或者说明来构建模型，然后结合测试对象生成测试用例，测试用例针对测试对象执行完之后，生成测试报告比对测试结果。

接下来，我以简单的登录系统为例，和你说明如何建模。

当用户访问网站时，网站需要识别用户是否已经登录：

如果已经是登录状态，则让用户进入，结束这一分支；

如果用户还没有登录，那么页面需要返回登录框给用户。用户在登录框输入用户名和密码后，由后台服务验证用户名和密码是否正确，如果通过验证，则用户登录成功，结束分

支；否则，返回错误信息，并再次返回登录框供用户登录。

根据这个逻辑，我们可以建模如下：

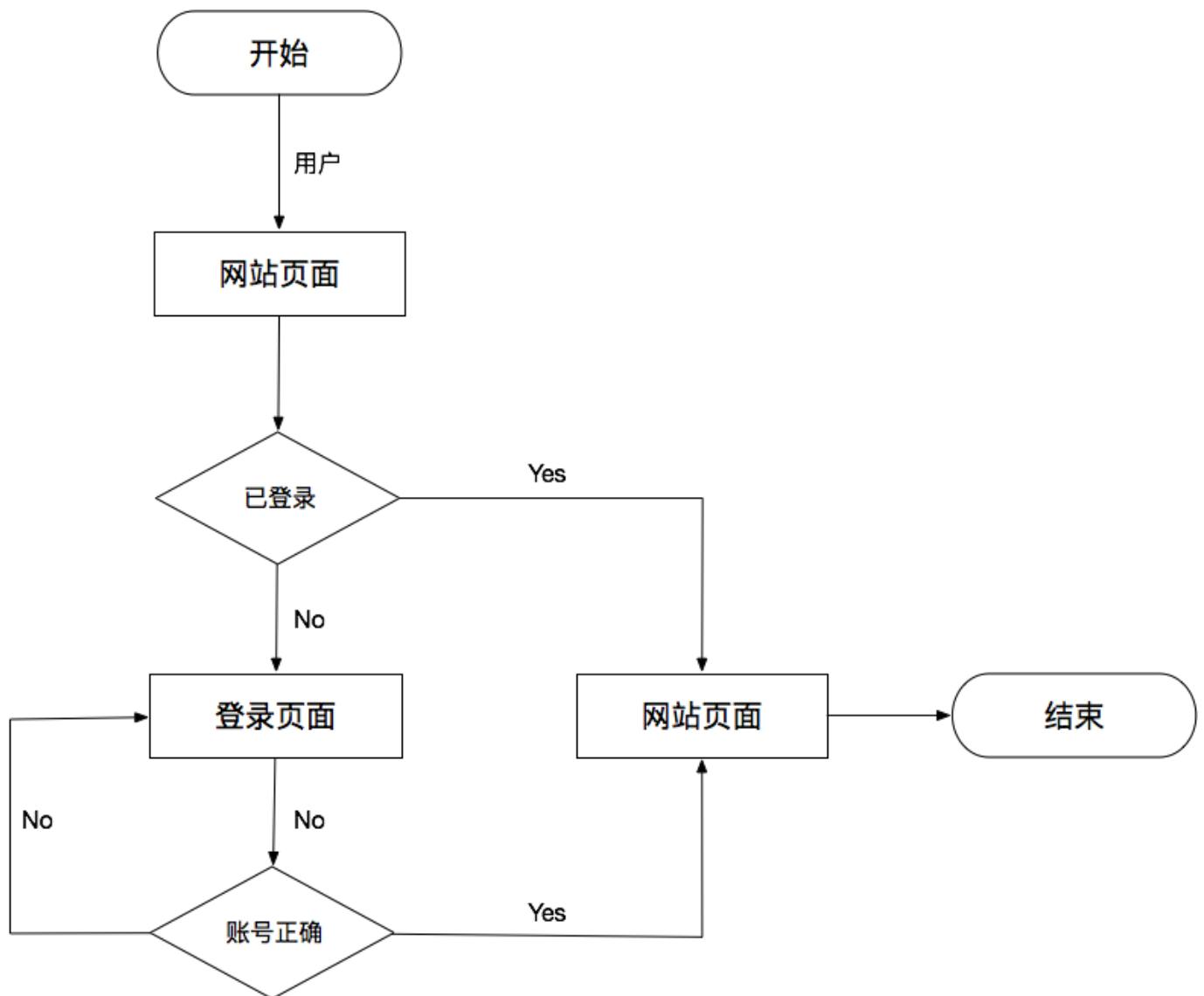


图 2 网站登录系统建模

至此，我们就完成对这个登录系统的建模工作了。然后，我们通过具象化被测产品的需求行为，再通过工具来遍历模型中的各个路径，就可以得到我们需要的测试用例了。

所以，执行 MBT 的过程就好比你把软件系统的设计画为了一张由节点和边构成的数据结构意义上的“图”，然后通过一定的算法（比如，深度遍历或者广度遍历）来尽可能完整覆盖图中全部的可能路径的过程。

而根据被测系统的特点，我们可以创建不同类型的模型完成 MBT。接下来，我们就一起看看有哪些常用的 MBT 模型吧。

常用模型简介

根据被测系统本身的特点，我们常用的模型主要有限状态机、状态图，以及 UML 三种。其中，有限状态机和状态图比较适合于用状态或者事件驱动的系统，而 UML 比较适合于靠业务流程驱动的系统。

第一，有限状态机。

有限状态机可以帮助测试人员根据选中的输入来评估输出，不同的输入组合对应着不同的系统状态。

在登录系统这个例子中，员工在未登录时的状态是“未登录”，一旦登录成功状态就会变为“已登录”。在已登录的状态下，员工可以访问各类资源，使用系统内的工具。

第二，状态图。

状态图是有限状态机的延伸，用于描述系统的各种行为，尤其适用于复杂且实时的系统。

状态图有一定数量的状态，系统的行为可以以事件的方式来驱动状态的变化。比如，缺陷管理工具中出现了缺陷，其初始状态为“new”；缺陷被开发人员修复后，就必须将其改为“Fixed”；但是，如果此时测试人员发现缺陷并未修复或者只是部分修复时，则需将状态更改为“Reopen”（重新打开）。

状态图的设计方式，要求为每个不同的状态创建一个事件。

第三，UML。

UML 即统一建模语言，是一种标准化的通用建模语言。UML 有自己定义的图形库，里面包含了丰富的图形用以描述系统、流程等。

UML 可以通过创建可视化模型，来描述非常复杂的系统行为。

当我们完成被测系统的建模工作后，接下来就要将模型转化为可执行的测试用例了。这个转换过程，需要借助工具来完成。

因为不同领域的产品风格迥异，其使用的自动化框架和编程语言也各不相同，所以我们需要花费一些精力去寻找与自己产品匹配的 MBT 工具。其实，在很多情况下，我们还需要根据产品特点，去自行开发和定制工具。

MBT 工具简介

这里，我为你罗列了一些常见的 MBT 工具，包括：BPM-X、fMBT、GraphWalker。在这里，我为你简单介绍这些工具的目的是，让你可以对 MBT 工具本身有个感性的认识，让你知道此类工具的应用场景和上下文。至于说如何来选择使用这些工具，这在很大程度上取决于被测系统的特点。

第一，BPM-X

BPM-X 根据不同的标准（比如，语句、分支、路径、条件）从业务流程模型创建测试用例。

它还可以从多个建模工具导入模型，并可以将测试用例导出到 Excel、HP Quality Center 等。这个工具，适用于业务流程比较清晰直观的场景。

第二，fMBT

fMBT 是一组免费的、用于全自动测试生成和执行的工具，也是一组支持高水平测试自动化的实用程序和库，主要被应用在 GUI 测试中。

fMBT 包括用于多平台 GUI 测试的 Python 库，用于编辑、调试、运行和记录 GUI 测试脚本的工具，以及用于编辑和可视化分析测试模型和生成的测试工具。

第三，GraphWalker

GraphWalker 以有向图的形式读取模型，主要支持 FSM、EFSM 模型。它读取这些模型，然后生成测试路径。GraphWalker 除了适用于 GUI 测试外，更适用于多状态以及基于事件驱动的状态转换的后台系统。

另外，GraphWalker 还支持从有限状态机中生成测试用例。

除此之外，市面上还有很多 MBT 测试工具，比如 GSL、JSXM、MaTeLo、MBT Suite 等。这里就不再一一介绍了，你可以自行百度了解它们的特点和适用场景，从而选取合适自己的工具。

MBT 的优势

其实，MBT 并不能算是一种新颖的测试技术，早在七八年前就已经被提了出来并且试图应用于软件产品的测试工作中。但是，MBT 在很长一段时间内，却一直停留在概念阶段，主要原因是一直没有普适的工具支持，所以很少有成功实施的实际案例。同时，业界一直以来都缺乏高效的测试用例设计生成策略，所以虽然大家都能看到 MBT 的优势，但能在实际项目中应用执行的却是寥寥无几。

与传统测试相比，MBT 的优点如下：

1. **测试用例的维护更轻松。** 由于测试用例是基于被测系统的模型生成的，因此我们只需维护好模型即可，而无需关注测试用例的细节。
2. **软件缺陷发现得更早。** 由于我们在构建被测系统模型的过程中，已经对被测系统有了比较全面的理解，加之要根据系统需求 / 说明完成建模过程，所以我们可以早期建模时发现被测系统可能存在的明显缺陷，而不用等到执行了大量的测试用例以后才发现这些缺陷。
3. **测试自动化的水平更高。** 由于 MBT 只需建好模型便可以自动生成测试用例，所以不再需要人工编写测试文档。而更高级的应用，甚至可以直接生成可以直接执行的自动化测试脚本。
4. **测试覆盖率变得更高，使得彻底的测试（即：穷尽测试）成为了可能。** 由于我们需要做的只是正确、详尽地用模型描述被测系统，而生成测试用例完全由 MBT 工具实现，所以这就避免了人工设计测试用例时的思维局限性，能够有效地提高测试覆盖率，让彻底的测试变为可能。
当然，是否有必要开展彻底的测试还是要由风险决定。这里的风险指的是，由于漏测导致产品问题对业务的影响程度。MBT 只是从技术上提供了可能性。
5. **基于模型间接维护测试用例的方式更高效。** 在传统测试中，如果被测系统的流程或者功能发生了变化，我们需要耗费大量的人力和时间成本，去重新设计与之匹配的测试用例。而在 MBT 中，我们只需要更新被测系统的模型即可，剩下的测试用例生成工作可以由 MBT 工具自动完成。

MBT 的劣势

虽然 MBT 相对于传统测试有很多优点，但它也不是完美的测试方案。在实际开展 MBT 时，我们往往需要应对很多挑战，并克服很多困难。所以，到现在为止，MBT 并没有被广泛使用于软件测试领域。

在这里，我总结了开展 MBT 的三大难点：

1. **学习成本较高。** MBT 要求开发人员和测试人员都精通建模，这就需要一定的培训成本，需要让开发人员去学习测试的技能，让测试人员去学习建模概念。这其中还牵涉到建模工具的选择，以及学习等成本。
2. **使用 MBT 的初期投资较大。** 在很多情况下，我们并不能找到适合自己产品的建模工具，而需要自行创建 MBT 工具。
在自行定制 MBT 工具时，我们要考虑到这个工具必须是可扩展的，并且能够处理复杂的测试逻辑，提供足够高的测试覆盖率，因此刚开始的工具建设就需要花费大量时间和精力。
更糟糕的情况是，当工具建好后，我们却发现它并不能满足所有的建模需求，因此还要在建模的同时对工具进行微调。而，这种微调工作的难度，也比较大。
3. **早期根据模型生成测试用例的技术并不是非常成熟。** 很多时候只是根据图论的算法来遍历模型，这就会导致生成的很多测试用例在业务上根本没有任何实际意义，也因此阻碍了 MBT 在实际项目中的落地。
不过好在近一两年来，基于人工智能（AI）生成测试用例的概念不断成熟，所以将基于 AI 的测试用例生成和 MBT 相结合，将会是接下来的一个发展方向。

总的来说，MBT 和传统测试各有优劣。所以，测试的方法多种多样，MBT 只是其中的一种。

如果一个应用的任何组件都可以通过模型来模拟、通过驱动程序来驱动，并可以通过测试结果来比较的话，那么这个应用就是 MBT 的最佳候选者。

如果我们的产品特征符合开展 MBT 的要求，并且团队各方面的条件都支持使用 MBT 时，我们便可以尝试用这种方法来改革我们的测试方式。尤其是将 MBT 结合基于 AI 的测试用例生成技术，将可以大大加速 MBT 产业应用的步伐。

但是，不管是否采用 MBT，开发人员或测试人员在接触到一款软件产品时，首先都会有一个心理建模的过程，自己先去理解并在脑中勾勒出系统的功能结构和流程。其实，这些内容很容易就可以转换成实际的模型，也就为 MBT 创造了条件。

总结

今天，我和你分享了 MBT 的基本概念和方法。

MBT 是一种基于被测系统的模型，由工具自动生成测试用例的软件测试技术。所以，这也就决定了 MBT 相对于传统测试技术可以在测试用例维护、软件缺陷发现时机、测试自动化水平，以及测试覆盖率等方面有其独到的优势。

但同时，这也使得 MBT 相对于传统软件测试，在初期阶段投入较大，学习应用的成本较高。也正因为如此，MBT 概念虽然已经提出了七八年的时间，但却并没有被广泛应用于软件测试领域。

所以，关于是否要在自己的项目中开展 MBT，我们需要综合考虑项目本身的特点和人员的技术水平，以此决定是否有必要开展 MBT。

思考题

假如你要在项目中开展 MBT，你会如何判断你的项目是否适合采用 MBT，以及你认为会遇到哪些问题可能会阻碍 MBT 的开展呢？

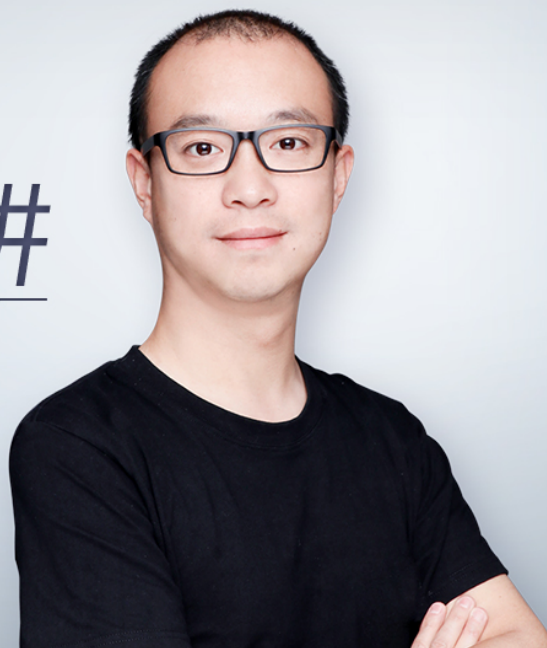
感谢你的收听，欢迎你给我留言一起讨论。




软件测试52讲

从小工到专家的实战心法

茹炳晟 eBay中国研发中心
测试基础架构技术主管



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

上一篇 46 | 安全第一：渗透测试

下一篇 测试专栏特别放送 | 答疑解惑第一期

精选留言 (11)

写留言



西兹兹

2018-12-11

1

类似于BPM

展开



口水窝

2019-05-23

1

首先，对于测试技术人员人员的挑战，就是对于建模并不熟悉，不知道如何建模。其次，对于工具的选择，不知道怎么才是适合自己产品的工具。然后，现在上线周期短，导致没有这么多时间去研究技术、工具。

展开



小老鼠

2018-12-01

1

对于unhappy的路径，MBT会涉及到吗？

展开



Alice

2018-10-30

1

MBT只是生成测试用例吗？

展开



无心

2018-10-23

1

以前有听过UML建模，但是没有执行过，可以很细致的画出模型

展开 ▾



红娟

2018-10-22



脑洞大开，第一次听说MBT概念

展开 ▾



Eve P...

2018-10-19



第一次接触MBT概念，以往项目有画过uml业务流，用于测试用例，但是从来没有用过具体的MBT工具。



小北

2018-10-16



学习到了新的思路，受教了。

展开 ▾



朝如青丝暮...

2018-10-16



这些软件使用起来非常的不便，感觉无从下手

展开 ▾



鱼贯而过

2018-10-15



我们研究生时期常做模型检测，将模型检测认为是不同于软件测试的一种方法。再扩大一些，模型检测属于形式化方法的一种，确实是门槛高，成本高，只有关系生命的大型软件，比如飞机上的关键软件，才需要形式化方法

展开 ▾



涅槃Ls

2018-10-15



打卡47

展开 ▾

