文章编号:1671-4598(2020)03-0072-07 DOI:10.16526/j.cnki.11—4762/tp.2020.03.016 中图分类号:TP311.5 文献标识码:A

基于任务的舰船装备软件测试技术研究

何 伟,沈晓美,刘泊江,韩新宇,唐龙利
(中国船舶工业集团公司 软件质量与可靠性测评中心,北京 100081)

摘要:针对基于任务的舰船装备软件进行测试时,缺乏任务需求的系统化分析与描述手段、难以从任务执行层面有效生成测试用例、不支持跨平台测试的自动化执行等问题,对舰船装备软件任务分析与建模、基于任务模型的舰船装备软件测试用例生成及舰船装备软件测试自动化执行等三方面关键技术进行了研究,研制了配套的软件测试工具,形成了一整套基于任务的舰船装备软件测试技术,以满足基于任务的舰船装备软件测试的需要;在此基础上,开展了基于任务的舰船装备软件测试技术实例应用,验证了该技术的工程适用性和配套工具的有效性,并形成了典型应用实例,为该技术的推广提供支撑。

关键词:扩展有限状态机;舰船装备软件;任务分析与建模;测试用例生成;测试自动化执行

Warship Software Testing Technique Based on Mission

He Wei, Shen Xiaomei, Liu Bojiang, Han Xinyu, Tang Longli
(China Shipbuilding Software Quality & Reliability Testing Center, Beijing 100081, China)

Abstract: Testing mission-based warship software usually involves many technical challenges, such as lack of systematical mission analysis and describing technique, hard to generate test cases with respect to mission requirements, unable to support automated cross-platform test execution, etc. To tackle these challenges, this paper presents an integrated approach to test mission-based warship software, including key steps on mission analysis and modeling, mission-model-based test case generation, and automated test execution. A tool for testing mission-based warship software is developed according to this approach. Besides, an example is provided to exhibit the feasibility of the approach and the effectiveness of the tool. In practice, this example can be looked up for reference so as to promote the usability of this approach.

Keywords: extended finite state machine; warship software; mission analysis and modeling; test case generation; automated test execution

0 引言

基于任务的舰船装备软件侧重于以军方用户的实际使用场景为出发点和落脚点。在设计上,此类软件主要采用开放式架构使组件松散耦合,以便对系统资源进行自动配置和动态重组,力求软件能用、好用,符合执行实际任务的要求。

对于此类基于任务的舰船装备软件,当前型号软件测试技术手段的适用性不强,主要体现在:①当前型号软件测试技术缺乏对任务需求的系统化分析,即使测试中考虑到隐含任务需求,有意设计了一些测试用例,也无法覆盖被测软件的全部可测任务需求,测试充分性难以保证[1];②当前型号软件测试技术主要关注于对软件各功能点实现的正确性进行验证,较少从任务完整执行过程的角度来设计测试用例,因而很难暴露出软件在任务执行层面的缺陷[2];③当前型号软件测试技术大多是在单机环境下运行被测软件实施测试,但基于任务的舰船装备软件常需要跨平台部署于采用不同处理器、操作系统的分布式网络基础设施和公共计算环境,现有测试工具难以针对此类软件进行有效的跨平台测试[3]。

针对上述问题,本文将深入探讨基于任务的舰船装备软件测试技术,包括:舰船装备软件任务分析与建模、基于任务模型的舰船装备软件测试用例生成和舰船装备软件测试自动化执行等三方面关键技术,给出相应的工具实现。在此基础上,开展基于任务的舰船装备软件测试技术实例应用,验证该技术的工程适用性和工具的有效性。总体研究思路如图1所示。

1 任务分析与建模技术

舰船作战任务具有分阶段、多层次、目标动态演化、协同关系复杂等特点,通常可视为由一系列子任务和元任务及它们之间的依赖关系(如顺序、并发、选择等)所组成的集合。其中,元任务是任务执行过程中系统对应关系相对固定、能够实现一定任务目标的最小任务单元,其特征为互不包含、不可再分。

对于基于任务的舰船装备软件而言,每项任务均可视为在达到规定目标过程中经历的一系列状态和事件,因而适于采用扩展有限状态机(Extended Finite State Machine, EFSM)来建立准确、有效的舰船装备软件任务模型,并以此为基础开展软件测试工作。

1.1 舰船装备软件任务特征分析

通过收集并分析现有舰船装备软件的用户任务需求,

收稿日期:2019-09-25; 修回日期:2019-11-07。

作者简介:何 伟(1984-),男,北京人,博士,高级工程师,主要从事程序分析与软件测试技术方向的研究。

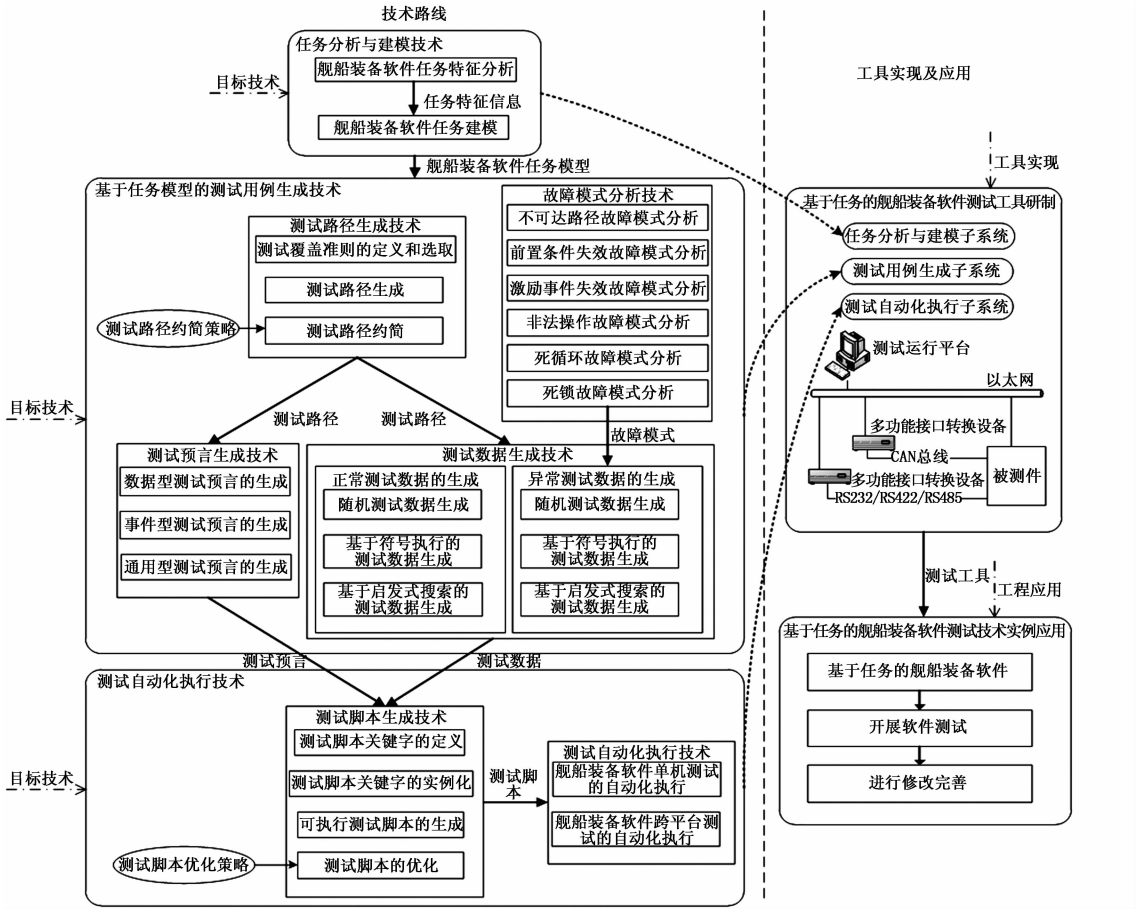

图 1　基于任务的舰船装备软件测试技术总体研究思路

开展任务特征分析，明确任务执行过程涉及的主要状态、关键变量以及各主要状态间的迁移关系，从而归纳出舰船装备软件的任务特征信息，包括：状态特征、数据特征、状态迁移特征。

在状态特征分析中，首先找出被测软件任务需求所包含的主要状态，然后明确初始状态和终止状态；在数据特征分析中，识别出任务执行过程涉及的关键变量，包括：输入变量、输出变量、可能影响状态变化的中间变量；在状态迁移特征分析中，理清各主要状态之间的迁移关系，进而分别确定每个状态迁移的源状态、目标状态、激励事件、前置条件、行动操作、后置条件。这些要素共同组成了舰船装备软件的任务特征信息。

1.2　舰船装备软件任务建模

根据任务特征信息，建立被测舰船装备软件任务 EFSM 模型。该模型由一个七元组 $M=<S,s_0,s_n,V,I,O,T>$ 构成[4]，其中 S 是一个非空的状态集，表示被测软件任务需求所包含的主要状态，s_0 是初始状态集，s_n 是终止状态集，V 是任务执行过程涉及的关键变量集，I 是输入变量集，O 是输出变量集，T 是非空的状态迁移集。T 中的每个元素 t 是一个六元组 $<s_{src},s_{tgt},event,pre-cond,action,post-cond>$。$s_{src}$ 为源状态，s_{tgt} 为目标状态，$event$ 是 t 的激励事件，$pre-cond$ 是 t 的前置条件，$action$ 为 t 引发的行动操作，$post-cond$ 是 t 的后置条件。

以某舰船装备软件设备初始化任务为例，建立 EFSM 模型如图 2 所示。在该模型中，状态集 $S=\{s_0,s_1,s_2,s_3,s_4,s_5\}$，初始状态为 s_0，终止状态为 s_5，迁移集 $T=\{t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8,t_9\}$。表 1 明确了每个迁移的源状态、目标状态、激励事件、前置条件、行动操作和后置条件。

在任务模型 M 中，一些复杂的行动操作可分解为若干个子任务，相应的模型由七元组 $M'=<S',s_0',s_n',V',I',O',T'>$ 构成。在此基础上，子任务可进一步分解为元任务，相应的模型由七元组 $M''=<S'',s_0'',s_n'',V'',I'',O'',T''>$ 构成。

针对子任务，采用上述方法构造状态迁移视图和状态迁移表，即可建立相应的子任务 EFSM 模型。类似地，针对元任务，也可建立相应的元任务 EFSM 模型。这样，就可以自顶向下建立完整的舰船装备软件任务 EFSM 模型。

2　基于任务模型的测试用例生成技术

基于任务模型的测试用例生成思路如下：首先，根据舰船装备软件任务 EFSM 模型，采用深度优先搜索策略，遍历舰船装备软件任务模型，生成满足指定测试覆盖准则的测试路径。然后，分析舰船装备软件的常见故障，识别出所蕴含的故障模式。根据每条测试路径涉及的前置条件

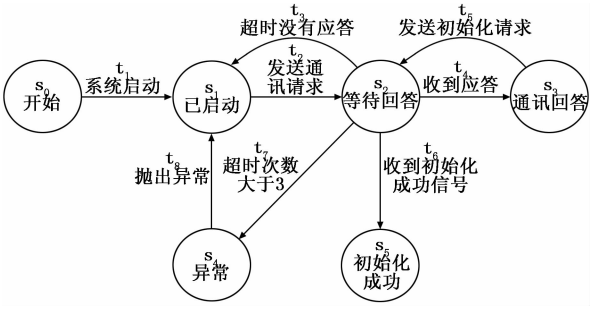

图 2 设备初始化任务 EFSM 模型的状态迁移视图

集合，生成正常测试数据；辅以舰船装备软件故障模式，生成异常测试数据。另一方面，利用每条测试路径涉及的后置条件或隐含的蜕变关系、容错机制、通用要求等信息，生成相应的测试预言。

表 1 设备初始化任务 EFSM 模型状态迁移表

迁移	源状态	目标状态	激励事件	前置条件	行动操作	后置条件
t_1	s_0	s_1	收到系统启动信号	无	无	无
t_2	s_1	s_2	无	无	发送通讯请求报文	通信请求报文发送成功
t_3	s_2	s_1	无	应答时间超过 1000ms	无	无
t_4	s_2	s_3	收到应答信号	无	无	无
t_5	s_3	s_2	无	无	发送初始化请求报文	初始化请求报文发送成功
t_6	s_2	s_5	收到初始化成功信号	无	设备初始化任务结束	无
t_7	s_2	s_4	无	超时次数超过 3 次	无	无
t_8	s_4	s_1	无	无	抛出异常	无

2.1 测试路径生成

对于基于任务的舰船装备软件而言，每条测试路径 p 可以用一个从初始状态 s_0 到终止状态 s_n 的状态－迁移序列表示。

根据被测舰船装备软件的规模、复杂度，可以定义和选取合适的测试覆盖准则，生成满足指定测试覆盖准则的测试路径，然后分别针对每条测试路径设计测试用例，实现对被测软件的测试，同时达到测试资源和测试质量之间的良好折衷。

本文采用深度优先搜索策略，遍历舰船装备软件任务 EFSM 模型，生成满足指定测试覆盖准则的测试路径。具体而言，测试路径生成分为 2 步实现：

1）深度优先搜索各元任务、子任务 EFSM 模型，生成原子测试路径；

2）深度优先搜索完整的任务 EFSM 模型，连接所经原子测试路径，构成复合测试路径，即目标测试路径。

随着软件复杂度增加，相应的 EFSM 模型中状态数和迁移数也会明显增加，导致测试路径生成的搜索空间急剧增大，测试路径生成效率严重下降。为此，本文根据软件外部组件调用和异常处理机制的特点，提出了两种搜索空间约简策略，通过约简软件任务 EFSM 模型中的外部组件状态节点和异常处理状态节点，在保证测试覆盖效果的同时，可减小测试路径搜索空间规模，提高测试路径生成效率。

2.1.1 针对外部组件状态节点的约简策略

软件测试主要关注被测软件本地组件内的状态迁移以及本地组件与外部组件间的临界状态迁移是否正确。因此，本文提出在构建舰船装备软件任务 EFSM 模型时，针对外部组件状态节点进行约简，以减小生成测试路径时的搜索空间，约简步骤如算法 1 所示。

该算法以舰船装备软件任务 EFSM 模型为输入，其核心思想是通过广度优先遍历 EFSM 模型，识别并删除外部组件状态节点以及相应的迁移边，从而得到约简后的 EFSM 模型。

算法 1：外部组件状态节点约简策略

```
输入：约简前的 EFSM 模型 mission_efsm
输出：约简后的 EFSM 模型 reduced_efsm
s0 = mission_efsm.getEntry();
foreach state∈mission_efsm.getNodes() - {s0} do
  visited[state] = FALSE;
Queue Q = Ø;
enqueue(Q, s0);
while Q≠Ø do
  si = dequeue(Q);
  foreach transition∈cg.edgesOutOf(si) do
  sj = transition.tgt();
  if visited[sj]=FALSE and
   sj.belongToExtComponent()=TRUE then
  mission_efsm.nodes=
       mission_efsm.nodes - {sj};
  mission_efsm.edges=
        mission_efsm.edges - {<si, sj>};
  enqueue(Q, sj);
  visited[si] = TRUE;
reduced_efsm = mission_efsm;
return reduced_efsm;
```

2.1.2 针对异常处理状态节点的约简策略

软件测试首要目标是检测软件核心功能是否正确。作为测试覆盖目标，测试路径不必包含异常处理状态信息。因此，本文提出在构建软件任务 EFSM 模型时，针对异常处理状态节点进行约简，以减小生成测试路径时的搜索空间。

与针对组件调用状态节点的搜索空间约简策略的实现方式相似，针对异常处理状态节点的搜索空间约简策略可通过广度优先遍历软件任务 EFSM 模型，识别并删除 EFSM 模型上的异常处理状态节点及其相应的边。

2.2　故障模式分析

开展舰船装备软件故障模式分析，针对舰船装备软件的常见故障，归纳故障特征（包括故障类型、原因、现象等），形成舰船装备软件故障模式库，目的在于根据故障模式生成可能触发软件故障的异常测试数据，并施加于被测软件，从而对其进行针对性的测试[6]。考虑到实际舰船装备软件中故障种类繁多、难以穷尽，因此本文主要考虑不可达路径、前置条件失效、激励事件失效、非法操作、死循环、死锁等 6 种故障模式。

1）不可达路径故障模式指软件设计或实现中分支判断间相互矛盾，造成相应的测试路径无法执行。例如，某条测试路径上的约束条件要求 var＜10 和 var＞100 同时为真，但实际上不存在这样的数据，导致该路径永远无法执行。

2）前置条件失效故障模式是指软件设计或实现中出现分支判断错误，导致软件前置条件失效，常见情况包括：在分支判断中对浮点型变量进行相等/不等比较、对无符号数与有符号数进行比较等。

3）激励事件失效故障模式是指软件中涉及并发任务时信号量处理不当，导致激励事件未发挥预期作用，软件状态出现错误。

4）非法操作故障模式是指软件执行任务（尤其是用户输入操作、接口输入操作）时，输入的数据由于访问控制不当等问题，威胁到软件的安全性和健壮性，导致软件出现异常。

5）死循环故障模式指的是软件由于循环或递归逻辑设计、实现不当，陷入重复执行某段程序的过程中，通常表现为软件失去控制从而无法响应外部输入事件或者程序调用栈溢出。

6）死锁故障模式是指软件中通过多线程访问共享资源，由于访问的顺序不当，造成线程间互相等待，软件无法继续执行。

2.3　测试数据生成

为舰船装备软件生成测试数据，首先要获取各测试路径上涉及的数据约束，其步骤如下：

1）针对被测软件的每条测试路径 p，收集路径上的全部状态迁移，构成状态迁移集 T_p；

2）提取状态迁移集 T_p 中每个状态迁移 t 涉及的数据约束，构成该路径的数据约束集 C_p。

根据测试路径及数据约束信息，可生成所需测试数据[7]。本节讨论的测试数据生成方法及其适用情形如表 2 所示。

表 2　测试数据生成方法一览表

适用情形			测试数据生成方法
黑盒测试	离散型变量		常量池法
	连续型变量	简单约束	随机生成法
		复杂约束	符号执行法
白盒测试			启发式生成法

2.3.1　常量池法

通过分析舰船装备软件的数据约束，可以发现某些通用字段的取值范围是一些离散点，甚至是固定取值。根据这些字段的取值范围/典型值，通过划分等价类取值、边界取值等方式建立常量池。在测试数据生成阶段，为相应字段高效生成所需测试数据。

例如，执行某舰船装备软件任务时需要输入 1 个航速字段，该字段为 int 类型，占用 4 个字节，正常取值范围为［－40，40］kn。为了检验该软件能否接收航速输入数据并正确响应，可以在常量池中为该字段添加｛－41，－40，－39，－1，0，1，39，40，41｝等备选值。这样在测试数据生成阶段，就能由常量池为该字段高效生成所需测试数据。

2.3.2　随机生成法

对于被测舰船装备软件中未明确取值范围的连续型字段，本项目根据相应字段的数据类型通过随机取值的方法快速生成测试数据。对于被测软件中有明确取值范围的连续型字段，可采用更为先进的自适应随机测试方法生成测试数据，该方法的核心思想是在各字段取值范围内尽可能生成均匀分布的测试数据，以提高发现潜在故障的概率[8]。

随机生成法便捷易行，而且适用于不同规模的被测软件，但是常常难以生成满足复杂约束的测试数据。因此，对于涉及复杂数据约束的连续型变量，建议采用符号执行法。

2.3.3　符号执行法

符号执行的核心思想是使用符号表示软件输入变量，而软件中的变量以及输出结果则由符号表达式（由代表输入变量的符号以及运算符构成）表示，通过模拟软件的执行过程，根据每条路径的执行条件，利用现有的约束求解器（如 SMT、Z3 等）对执行条件构成的方程组进行求解，即可生成测试数据。

舰船装备软件常采用收发报文的方式进行组件交互，以实现某些指挥、监控和通信任务，常用报文均涉及数组、结构体等复杂数据结构。然而，符号执行法通常难以生成此类涉及复杂数据结构的测试数据。因此，本文进一步探讨基于被测软件源代码的启发式生成法，作为常量池法、随机生成法和符号执行法的有益补充。

2.3.4　启发式生成法

针对含复杂数据结构输入变量的舰船装备软件测试数据生成需求，利用启发式测试数据生成方法意在为被测舰船装备软件搜索到一组覆盖指定路径的测试数据。具体来说，首先需要构建初始种群，其中每个个体表示一条随机生成的测试数据，每个个体的基因数量取决于测试路径上输入变量的个数，基因的排列顺序由测试路径上输入变量的出现顺序决定。每个基因由二元组 $< f,d >$ 构成，其中

f 是一个标志位，用于表示该基因对应的输入变量的数据类型，0 为基本类型数据，1 为复合类型数据；当基因对应于基本数据类型输入变量时，d 为该输入变量取值范围内的一个具体数据值，当基因对应于复合数据类型输入变量时，d 为一个索引，指向表示该复杂数据的所有基因。然后，将个体解码为可驱动测试数据，在被测软件执行测试数据并评估实际执行路径与当前目标路径的距离。如果其中某条测试数据能够覆盖当前目标路径，就将该测试数据添加到结果数据集合，并为下一条目标路径搜索相应的测试数据；否则，如果该种群中每个个体对应的测试数据都无法覆盖当前目标路径，就利用遗传算子再生成一个新的种群，并评估新种群中每个个体对应的实际执行路径与当前目标路径的距离。依此重复种群再生、个体执行和距离评估过程，直到满足指定的测试覆盖准则或达到其它终止条件。结果数据集合中即为所生成的具备良好测试覆盖能力的测试数据。

2.4 测试预言生成

测试预言用于判定被测软件在特定场景下是否正确运行。随着软件测试过程中程序分析、测试数据生成、测试执行等环节的自动化水平不断提升，测试预言已成为制约整体测试效率的主要瓶颈[9-10]。本文利用舰船装备软件测试路径上的后置条件和隐含的蜕变关系、容错机制、通用要求等信息生成测试预言。

2.4.1 测试预言的分类

根据测试预言所要检测的内容，可将测试预言分为 2 类：数据型测试预言、事件型测试预言。

数据型测试预言用于判定被测软件执行过程中指定变量的取值是否正确。例如，在某软件设计文档中描述了 GetRSSpeed 函数的返回值为串口通信波特率代号（0：9600bps，1：38400bps，2：115200bps），则其测试预言为函数返回值必须为 0，1 或 2，若实际返回值不是这三者之一，则认为函数实现有误。

事件型测试预言用于检查被测软件在执行过程中是否发生特定事件。例如，在调用 OpenRS 函数后，可以通过执行 WriteRS 函数向串口写入数据并在另一端读取数据来判断串口是否成功打开，也可通过执行 GetRSState 函数获取当前串口状态来判断串口是否成功打开。

此外，还可以根据被测软件的隐含要求，归纳出通用型测试预言。在很多情况下，这类通用型测试预言在软件相关的文档中没有明确进行定义，但是其作为对软件产品普遍适用的要求，也应当作为测试预言。

2.4.2 测试预言的生成

2.4.2.1 基于后置条件的测试预言生成方法

根据软件用户需求，采用任务分析与建模技术可快速构建软件任务 EFSM 模型。在被测舰船装备软件 EFSM 模型中，状态迁移的后置条件可直接转换为测试预言。例如：某后置条件要求输出字符串 string1≠NULL，则其预期输出就是一个非空字符串。

2.4.2.2 基于蜕变关系的测试预言生成方法

蜕变关系（metamorphic relations）是指被测软件在多次执行中必然成立的特定关系。根据被测舰船装备软件任务中隐含的蜕变关系，也可生成测试预言。例如：某项元任务用于计算 $f(x)=e^x$，则 $e^a\times e^{-a}=1$ 必然成立就是一项蜕变关系，若分别采用 0.3 和 −0.3 作为测试数据，执行实现该元任务的组件，则预期输出为两次执行的结果乘积为 1。

2.4.2.3 基于容错机制的测试预言生成方法

根据被测舰船装备软件任务中的容错机制生成测试预言是指：舰船装备软件中的某些关键组件会通过多个版本实现，且在并行执行后以大多数实现版本所支持的运算结果作为实际输出，从而实现容错。也就是说，给定某组件的 N 个实现 p_1，p_2，…，p_N 以及某门限值 $\sigma\in(1/2,1]$，若 p_1，p_2，…，p_N 中有 n 个实现版本的实际输出均为 x 且 $n/N\geqslant\sigma$ 时，则将 x 作为该组件的实际输出。在这种情况下，该组件的预期输出也认为是 x。一旦以某测试数据为输入，执行该组件，相应的实际输出结果不是 x，则认为该组件实现有误。

2.4.2.4 通用型测试预言生成方法

有些对软件产品普遍适用的要求，虽然未在软件文档中直接言明，但是作为隐含要求，被测软件也应当满足。例如：软件不允许崩溃、不允许发生死锁、不允许出现内存泄露、不允许缓冲区溢出等。这类必须遵守的隐含要求可作为通用型测试预言。

需要补充说明的是，考虑到开发文档对软件预期行为描述详细程度不一、软件任务流程的某些中间状态有时难以观测、软件实际运行场景存在不确定性等因素，理想的测试预言是几乎无法实现的。在具体实践中，需要考虑多方面因素，在测试预言的成本和效益之间做出权衡，利用有限的资源尽可能对软件中可能出错的环节进行充分测试。

3 测试自动化执行技术

生成满足测试覆盖准则的测试用例后，就需要将测试用例翻译为可执行且便于复用的测试脚本，然后按照测试环境要求在测试平台上执行测试脚本，通过比较实际输出与预期输出是否相符，判断各测试用例是否通过。实现软件测试的自动化执行能够有效提高测试效率，降低测试成本。

3.1 测试脚本生成技术

舰船装备软件测试脚本生成可通过 4 步实现：①定义测试脚本关键字；②测试脚本关键字实例化；③生成可执行测试脚本；④测试脚本优化。

首先，根据测试用例中描述的要素信息，定义测试脚本关键字集合。测试脚本关键字是根据测试用例描述的脚本要素信息。从舰船装备软件测试脚本的共性入手，可以归纳出两大类关键字：脚本执行关键字和脚本说明关键字。脚本执行关键字用于描述测试场景，涵盖测试脚本中的函数名称、参数名称、控制逻辑等，在测试脚本中实例化为

可执行语句；脚本说明关键字则用于描述测试脚本的标识信息、测试条件、预期输出以及补充信息，在测试脚本中实例化为断言语句和注释语句。

其次，生成测试脚本框架，并对测试脚本关键字进行实例化。测试脚本框架是指根据测试用例预先定义的具有一定流程控制功能的脚本单元，采用 XML 格式表示。测试用例执行行为与测试脚本的执行逻辑密切相关，因此可以根据测试用例建立测试脚本框架。测试脚本关键字实例化就是将测试脚本框架中的关键字替换为测试用例要素信息的过程。在测试用例信息完备和测试脚本关键字定义清晰的基础上，根据测试用例描述，在测试脚本框架的标识信息、测试前提、测试步骤、预期输出、测试备注这五个部分中查找关键字，将相应的测试用例要素信息替换到关键字位置，即可实现测试脚本关键字实例化，得到 XML 格式的测试脚本。

接下来，采用 DOM 或 SAX 对 XML 格式的测试脚本进行解析，根据测试用例要求及可执行测试脚本的语法要求，将 XML 格式的测试脚本中各元素逐项转化为可执行脚本语句，从而生成完整的可执行测试脚本。

3.2　测试自动化执行技术

现有研究和工程实践中，单机运行环境下的测试执行技术已相对较为成熟。然而，基于任务的舰船装备软件常需要能够跨平台运行于 i386、sparc32、x86 _ 64、sparc64、amd64、ppc64 等不同处理器以及 Windows、Linux、Solaris 等不同操作系统上的分布式网络基础设施和公共计算环境，传统的单机测试技术已不适用。如果通过手工操作来执行跨平台测试并逐一判断测试结果，不仅人力成本高昂，还可能难以满足某些测试用例中对特定行为的时序要求，而无法实现相应的复杂测试场景[11]。

实现舰船装备软件的跨平台测试，主要步骤如下：①脚本分发：将需要执行的一组测试脚本分发至指定测试机；②远程环境部署：远程配置测试脚本所需的运行环境；③远程测试执行：远程驱动测试脚本批量化执行；④远程结果收集：收集各测试脚本的执行结果并回传至本地计算机；⑤结果显示：利用本地计算机直观地显示各测试脚本的执行结果。

针对舰船装备软件的跨平台测试需求，本文提出利用 Software Testing Automation Framework (STAF)[12] 的可重用组件实现跨平台测试的自动化执行框架，如图 3 所示。

具体而言，跨平台测试的自动化执行框架负责接收并处理测试人员的测试执行指令，并利用 STAF 后台服务将测试脚本自动分发到指定运行平台，然后驱动测试脚本批量执行，全程监控测试执行过程、收集测试结果并提供可视化显示，从而实现测试脚本的跨平台分发、批量化执行、全过程监控以及测试结果的自动归集，以满足基于任务的舰船装备软件的跨平台测试要求。

4　基于任务的舰船装备软件测试工具研制

基于任务的舰船装备软件测试工具是上述任务分析与建模技术、测试用例生成技术和测试自动化执行技术的工程实现，其研制方案如图 4 所示、组成结构如图 5 所示。

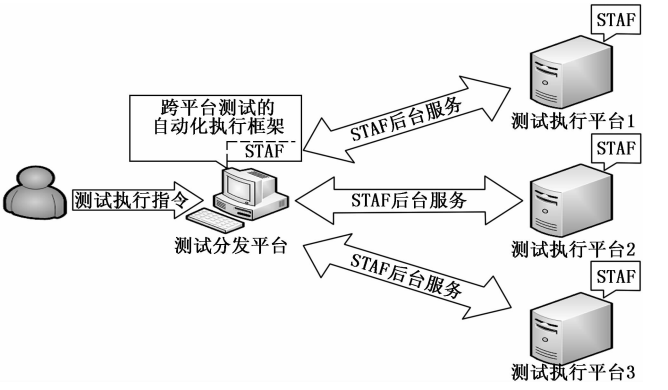

图 3　舰船装备软件跨平台测试的自动化执行技术方案

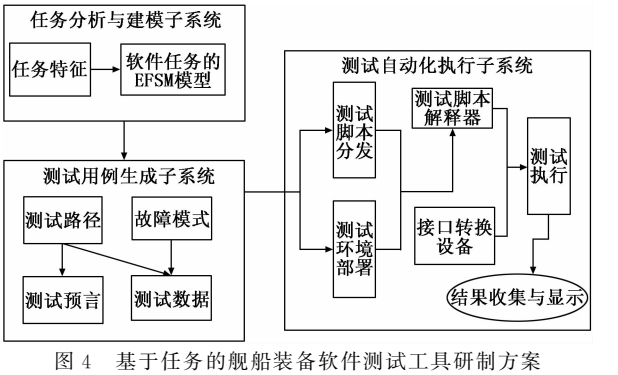

图 4　基于任务的舰船装备软件测试工具研制方案

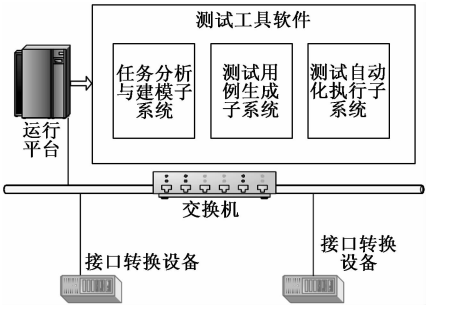

图 5　基于任务的舰船装备软件测试工具组成结构图

基于任务的舰船装备软件测试工具主要包括三个子系统：任务分析与建模子系统、测试用例生成子系统、测试自动化执行子系统。任务分析与建模子系统用于根据软件用户需求建立被测舰船装备软件任务 EFSM 模型；测试用例生成子系统包括测试路径生成、故障模式分析、测试数据生成和测试预言生成四大功能；测试自动化执行子系统通过人机交互将需要执行的测试脚本从本地计算机分发至指定测试机、配置测试脚本的运行环境、驱动测试脚本批量化执行、收集并显示测试执行结果。其中，测试执行所使用的接口转换设备为硬件系统，主要实现 CAN 总线接口、RS232/RS422/RS485 串口与以太网接口的多功能转换。

5　实例应用

为了验证上述技术方法的技术可行性和测试工具的工

程适用性，选用某装置监控软件、某概貌系统软件、某跟踪应用软件等 3 型舰船装备软件作为应用对象开展实例应用，应用方案如图 6 所示。

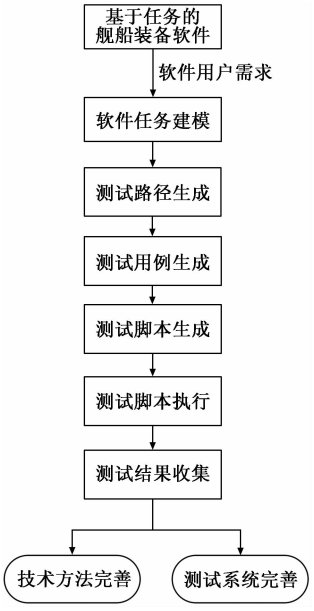

图 6 实例应用方案

应用过程中，采用本文提出的技术方法和测试工具可根据软件任务需求，成功建立软件任务 EFSM 模型。从测试用例编写和测试脚本生成的时间来看，此前人工对应用对象进行测试时，由于测试数据量较大，3 人用了 8 天时间完成相关测试用例编写和测试脚本编辑工作；本次实例应用过程中，采用基于任务的舰船装备软件测试工具，由 1 人在 1 天内完成了测试用例的编写和测试脚本的生成。从测试执行情况来看，此前对应用对象进行测试时，每个用例都需要人工逐步执行，3 个人用了约 2 天执行完相关测试用例；在本次实例应用过程中，采用基于任务的舰船装备软件测试工具，由 1 名测试人员执行测试、1 名测试人员操作被测软件和查看结果，全部测试用例在 2 小时内执行完毕，与未采用测试工具开展测试相比，测试设计和执行效率提高了 70%以上。

此外，基于任务的舰船装备软件测试工具可实现对多线程死锁故障测试，支持跨平台远程测试执行，有效解决现有舰船装备软件采用不同处理器、操作系统的分布式网络基础设施和公共计算环境复杂的问题。

实例应用结果表明：本文提出的基于任务的舰船装备软件测试技术是切实可行的，以此为基础研制的配套工具有助于提高舰船装备软件测试质量和效率，达到了预期效果。

6 结束语

本文以基于任务的舰船装备软件为对象，开展软件任务分析与建模、基于任务模型的测试用例生成、测试自动化执行等三方面技术方法研究。在软件任务分析与建模技术研究中，通过分析舰船装备软件任务特征，获取状态、数据和迁移信息，准确建立舰船装备软件任务 EFSM 模型。基于任务模型的测试用例生成技术研究中，根据舰船装备软件任务 EFSM 模型，高效生成测试数据和测试预言。在测试自动化执行技术研究中，将 XML 格式测试用例转化为可执行测试脚本，实现了跨平台运行环境下的测试自动化执行，可显著提升测试自动化水平。在此基础上，研制了基于任务的舰船装备软件测试工具，并以某装置监控软件、某概貌系统软件、某跟踪应用软件等 3 型舰船装备软件为对象开展实例应用，取得了良好的应用效果，能够为实施基于任务的舰船装备软件测试提供技术支撑。

本文研究成果的适用范围主要针对舰船装备软件的测试设计与执行阶段，后续可探讨测试需求分析、测试策划、测试总结等阶段的自动化测试技术，从而进一步提升舰船装备软件自动化测试水平。

参考文献：

[1] 黄贝贝，何 伟，韩新宇，唐龙利. 基于 EFSM 模型的舰船装备软件任务分析与建模方法 [A]. 第二十六届测试与故障诊断技术研讨会 [C]. 北京：中国计算机自动测量与控制技术协会，2017.

[2] 徐 佳，夏惠诚，吴媛媛，等. 面向作战任务的舰载指挥控制系统结构化匹配分析 [J]. 舰船科学技术，2010，32 (10)：34 - 37.

[3] 戴剑浩. 一种多平台自动化测试框架的设计与实现 [D]. 北京：北京大学，2013.

[4] 年晓玲. 基于扩展有限状态机软件测试用例自动生成的研究 [D]. 成都：西南交通大学，2005.

[5] 杨 瑞. 基于 EFSM 的测试用例自动化生成关键技术研究 [D]. 南京：南京大学，2015.

[6] Walton G H，Poore J H. Generating transition probabilities to support model－based software testing [J]. Software Practice and Experience，2000，30 (10)：1095 - 1106.

[7] 王雅文. 基于缺陷模式的软件测试技术研究 [D]. 北京：北京邮电大学，2009.

[8] 何 伟，韩新宇，唐龙利，等. 舰船装备软件测试数据自动生成方法及其应用研究 [J]. 计算机测量与控制，2015，23 (8)：2633 - 2636.

[9] Chen T Y，Leung H，Mak I K. Adaptive random testing [J]. Lecture Notes of Computer Science，2004，320 - 329.

[10] 王 馨，王 戟，齐治昌. 基于时序规范的测试预言自动生成技术评述 [J]. 计算机工程与科学，2006，28 (7)：127 - 133.

[11] Barr E，Harman M，Mcminn P，et al. The Oracle Problem in Software Testing：A Survey [J]. IEEE Transactions on Software Engineering，2015，41 (5)：507 - 525.

[12] 沈晓美，何 伟，韩新宇，等. 舰船装备软件跨平台测试执行技术研究 [A]. 第二十六届测试与故障诊断技术研讨会 [C]. 北京：中国计算机自动测量与控制技术协会，2017.

[13] 姜秀丽. 基于 STAF/STAX 的自动化测试平台的研究与实现 [D]. 大连：大连海事大学，2012.