

doi: 10.3969/j.issn.1006-1576.2012.08.026

## 基于 WF-net 的卫星在轨测试过程建模

吴云<sup>1</sup>, 史建伟<sup>2</sup>, 王益忠<sup>1</sup>

(1. 中国卫星海上测控部, 江苏 江阴 214431; 2. 装备学院研究生院, 北京 101416)

**摘要:** 为解决卫星在轨测试过程建模问题, 以导航卫星在轨测试为例, 应用工作流网(workflow-net, WF-net)建立卫星在轨测试过程模型, 描述卫星在轨测试活动过程及其逻辑关系, 并应用遍历树算法对模型进行分析验证。分析结果表明: 该模型不仅可形式化的描述在轨测试过程中涉及到的活动及其逻辑关系, 还可动态分析在轨测试过程的逻辑性。应用工作流网对卫星在轨测试过程进行建模是可行的。

**关键词:** 工作流网(WF-net); 卫星; 在轨测试; 过程建模

**中图分类号:** TJ861 **文献标志码:** A

## Process Modeling of Satellite On-Orbit Test Based on WF-Net

Wu Yun<sup>1</sup>, Shi Jianwei<sup>2</sup>, Wang Yizhong<sup>1</sup>(1. China Satellite Maritime Tracking & Control Department, Jiangyin 214431, China;  
2. College of Graduate, Institute of Command & Technology of Equipment, Beijing 101416, China)

**Abstract:** In order to solve the model problem of satellite on-orbit test process, take navigation satellite on-orbit test as example, use workflow-net to establish satellite on-orbit test process model. Describe satellite on-orbit test process and logic relation, and adopt coverability tree algorithm to analyze and validate model. The analysis result shows that the model not only describes the action and logic relation of in on-orbit test formally, but also dynamically analyzes the logistic of on-orbit test process. The method of workflow-net is suitable to the process model of on-orbit test.

**Key words:** workflow-net (WF-net); satellite; on-orbit testing; process modeling

### 0 前言

在轨服务是为了延长在轨系统使用寿命, 扩展和提高系统性能, 而对系统进行在轨检查、校正、维护和升级更新等服务任务的空间活动<sup>[1]</sup>。在轨测试是在轨服务的一项重要内容, 而卫星是目前在轨服务的主要对象。出厂后, 卫星需要进行严格的测试, 以查找卫星存在的设计缺陷、质量问题和潜在隐患。在轨运行中, 卫星同样需要测试, 卫星的在轨测试有利于延长卫星寿命, 节约成本。

在卫星的地面型号测试中, 有许多测试模式是常用的、固化的, 如整星加电序列、整星断电序列以及载荷测试的许多模式。利用图形化的编辑方法和手段, 借助细则编辑器编制测试序列和判读准则, 将编制好的测试序列装载到控制台软件中, 测试序列就可以在整星测试中按照顺序和时间运行, 并能够进行指令与参数关联的自动判读<sup>[2]</sup>。卫星在轨测试, 将更多地依靠自动测试技术, 分析卫星在轨测试的过程, 并进行建模研究, 是实现卫星在轨测试的一个基础。

工作流网(workflow-net, WF-net)是一种可用图形表示的组合模型, 具有直观、易懂和易用的优

点, 其对过程的描述和分析有独到的优越之处。同时, 利用工作流网分析方法和技术既可用于模型静态的结构分析, 又可用于模型动态的行为分析, 具有灵活的建模和强大的分析能力。笔者利用 WF-net, 分层构建了在轨服务体系下的卫星在轨测试过程模型, 并应用遍历树算法对卫星在轨测试过程模型进行了分析验证。

### 1 WF-net

#### 1.1 WF-net 的形式化定义

WF-net 是从 Petri 网发展而来的, 其定义<sup>[3]</sup>如下: 一个 Petri 网  $PN=(P,T,F)$  被称为 WF-net, 仅当它满足以下条件:

- 1) PN 有 2 个特殊的库所:  $i$  和  $o$ 。库所  $i$  是一个起始库所, 即  $*i=\Phi$ ;
- 2) 库所  $o$  是一个终止库所, 即  $o^*=\Phi$ ;
- 3) 每一个节点  $x \in P \cup T$  都位于从  $i$  到  $o$  的一条路径上。

#### 1.2 在轨测试活动间的逻辑关系

在轨测试活动之间的逻辑关系是指在轨测试活动之间的相互约束, 主要是由执行的因果关系等形

收稿日期: 2012-03-16; 修回日期: 2012-04-16

作者简介: 吴云(1985—), 男, 江苏人, 助理工程师, 从事航天器海上测控技术研究。

成的在轨测试活动之间的约束关系。笔者将在轨测试活动之间的逻辑关系定义为: 顺序、并发、条件、或、与、循环 6 种, 将复杂的测试活动分成若干子活动。下面借助 WF-net 的图形化表示方式, 对这 6 种基本的逻辑关系和子活动的表示进行分析研究。

1) 顺序关系。

顺序关系描述的是在轨测试活动一个接一个的顺序执行情况。此关系的模型描述比较简单, 如图 1 所示, 其中,  $T_1$  和  $T_2$  表达了 2 个具有前后顺序关系的在轨测试活动。

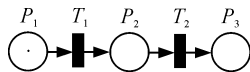


图 1 顺序关系

2) 并发关系。

并发关系描述的是前一在轨测试活动执行结束后, 其后多个在轨测试活动能同时执行的情况, 如图 2 所示, 当在轨测试活动  $T_1$  执行结束后, 其后的 2 个输出库所  $P_2$  和  $P_3$  同时各移入一个令牌, 2 个并发的在轨测试活动  $T_2$  和  $T_3$  能够被同时执行, 且互不影响。

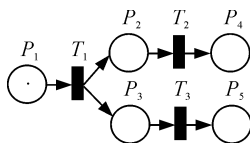


图 2 并发关系

3) 条件关系。

条件关系描述的是前一在轨测试活动执行结束后, 基于一定的判断条件, 在其后的 2 个或多个在轨测试活动中选择其中之一进行执行的情况, 如图 3 所示。

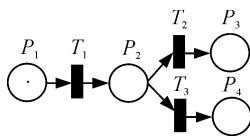


图 3 条件关系

4) 或关系。

或关系描述的是前面多个在轨测试活动中的任意一个在轨测试活动执行结束后, 后面的在轨测试活动就可以被执行的情况, 如图 4 所示。

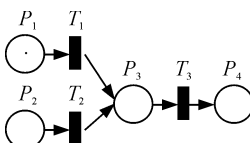


图 4 或关系

5) 与关系。

与关系描述的是前面多个在轨测试活动中的所有在轨测试活动执行结束后, 后面的在轨测试活动才能被执行的情况, 如图 5 所示。

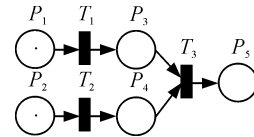


图 5 与关系

6) 循环关系。

循环关系描述的是基于一定的判定条件, 反复多次执行一个在轨测试活动的情况, 如图 6 所示, 在轨测试活动  $T_1$  每执行一次, 其输出库所  $P_2$  中移入一个令牌, 则  $T_2$  可以被激发, 激发的同时从  $P_2$  和  $P_3$  中移出一个令牌, 当在轨测试活动  $T_1$  执行  $m$  次后才能够结束。

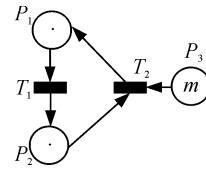


图 6 循环关系

## 2 卫星测试示例及测试过程建模

### 2.1 卫星测试过程示例

以在轨服务体系下对某颗导航卫星在轨测试过程为例, 设想在轨测试的过程如下:

- 1) 在轨服务系统收到测试任务请求;
- 2) 在轨服务系统根据请求判断任务等级;
- 3) 在轨服务系统选择执行该任务的航天器;
- 4) 根据航天器的当前位置确定其机动路径;
- 5) 执行任务的航天器进行轨道机动;
- 6) 机动到适当位置后, 执行航天器与目标航天器进行交会对接;
- 7) 由执行航天器发送测试命令, 测试活动开始;
- 8) 目标航天器上的敏感器进行数据采集;
- 9) 展开接收与发射性能测试和卫星信息处理性能测试;
- 10) 进行数据分析;
- 11) 进行结果分析;
- 12) 测试完毕, 航天器返回。

对于 workflow 中的复杂活动, 可以应用分层的思想, 将复杂活动看作一个整体活动, 在下一层中, 再将该活动分解。层次化的方法, 将 workflow 模型分解, 有助于降低模型的复杂度, 将复杂问题简单化。在上一层模型中, 对于含有子活动的活动, 用图 7

所示的图标进行表示，以区别于其它活动；在下一层模型中，属于上一层模型的活动用图 8 所示的图标进行表示，以区别于子活动。



图 7 含有子活动的活动



图 8 上一层活动

整个测试过程如图 9 所示。将“导航卫星信息处理性能测试”这一活动展开，如图 10 所示，其中，对于上一层活动，采用了如图 8 所示的图标。

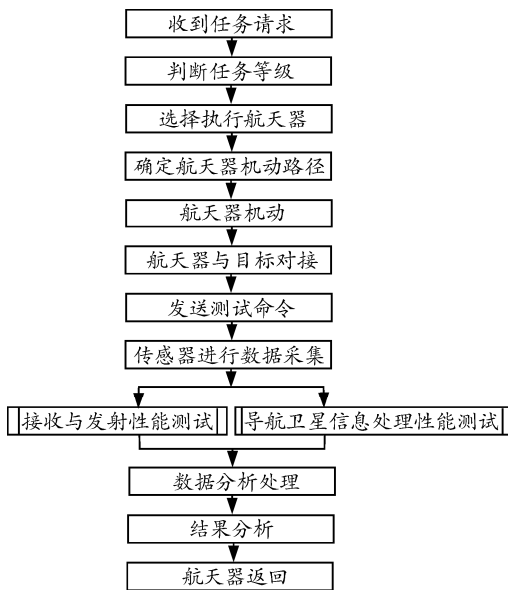


图 9 卫星在轨测试过程图

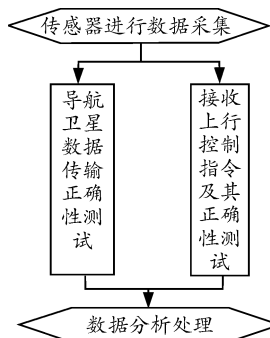


图 10 导航卫星信息处理性能测试活动分解图

### 2.2 卫星测试过程的 WF-net 模型

卫星测试过程用 WF-net 模型进行表示，如图 11 所示，“导航卫星信息处理性能测试”子活动用 WF-net 模型进行表示，如图 12 所示。

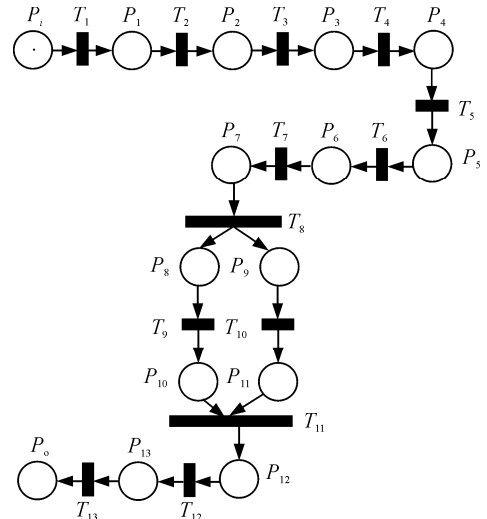


图 11 在轨测试过程 WF-net 模型

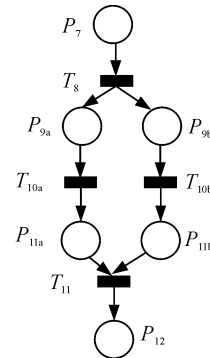


图 12 “导航卫星信息处理性能测试”活动 WF-net 模型

限于篇幅，在此仅给出变迁的含义，如表 1、2。

表 1 在轨测试过程 WF-net 模型中变迁的含义

变迁	含义	变迁	含义
$T_1$	收到任务请求	$T_8$	传感器进行数据采集
$T_2$	判断任务等级	$T_9$	接收与发射性能测试
$T_3$	选择航天器	$T_{10}$	信息处理性能测试
$T_4$	确定航天器机动路径	$T_{11}$	数据分析与处理
$T_5$	航天器机动	$T_{12}$	结果分析
$T_6$	航天器与目标对接	$T_{13}$	航天器返回
$T_7$	发送测试命令		

表 2 “导航卫星信息处理性能测试”活动 WF-net 模型中变迁的含义

变迁	含义
$T_8$	传感器进行数据采集
$T_{10a}$	导航卫星数据传输正确性测试
$T_{10b}$	接收上行控制指令及其正确性测试
$T_{10c}$	自主完好性功能测试
$T_{11}$	数据分析与处理

### 3 正确性验证和性能分析

卫星在轨测试过程模型建立以后，需要对模型结构正确性进行分析，即分析卫星在轨测试过程在逻辑上的正确性，从而保证模型中不会出现死锁、

运行结果不正确等现象。因为一个错误或者不完整的模型会产生无法预料的结果, 如死锁, 使得卫星在轨测试过程不能被正确地执行; 因此, 在卫星在轨测试过程模型被执行之前对其正确性进行分析和验证就显得很重要。

### 3.1 过程模型的正确性定义

**定义** 卫星在轨测试过程模型是正确的, 当且仅当<sup>[4]</sup>:

1) 对于任何一个由  $i$  可达的状态  $M$ , 都存在一个实施序列使  $M$  可以到达状态  $o$ ,  $i$  和  $o$  分别为卫星在轨测试过程模型的初始状态和结束状态, 即  $\forall M(I \xrightarrow{*} M) \Rightarrow (M \xrightarrow{*} o)$ ;

2) 对于任何一个由  $i$  可达的状态  $M$ , 如果此时在位置  $o$  中存在至少一个令牌, 该状态就是终止状态。终止时, 卫星在轨测试过程模型除了有一个令牌在  $o$  位置中外, 其它的位置都应该是空的状态, 即:  $\forall M(I \xrightarrow{*} M \wedge M \geq o) \Rightarrow (M = o)$ ;

3) 在卫星在轨测试过程模型中没有死的变迁, 即:  $\forall t \in T \exists M, M': I \xrightarrow{*} M \xrightarrow{*} M'$ 。

### 3.2 正确性验证

应急空间军事任务过程模型的正确性验证即为 WF-net 模型的正确性验证, 而对 WF-net 模型的正确性进行验证的方法主要包括: 遍历树 (coverability tree) 算法、关联矩阵、 $P/T$  变量、图形化简方法等。

由于遍历树 (coverability tree) 算法能够对 WF-net 模型中的所有状态进行遍历, 且在遍历过程中能够发现模型中的死锁等问题, 因而能有效地对 WF-net 模型进行验证。因此, 笔者选择遍历树算法对应急空间军事任务过程模型的正确性进行验证。为了遍历树算法有效, 先要声明下面这个引理。

**引理** 如果一个 workflow 网是无界的, 那么该 workflow 网不是正确的。

验证应急空间军事任务过程模型正确性的遍历树算法<sup>[5]</sup>如下:

1) 初始化, 可达状态集合  $M_R$  初始值为  $\{M_i\}$ , 可达变迁集合  $T_R$  为空, 把初始状态  $M_i$  为遍历树的根, 并且把  $M_i$  标识为 “new”;

2) 当  $M_R$  中有标识为 “new” 的状态存在时, 循环执行下列步骤:

① 选择一个标识为 “new” 的状态  $M$ , 并清除  $M$  的标识;

② 如果  $M$  与从遍历树的根到  $M$  的路径的一个

状态相同时, 表示已经存在循环, 则把标识置为 “repeat”, 返回步骤①;

③ 如果在状态  $M$  下, workflow 网中不存在可以被激发的变迁, 则表示  $M$  是一种死锁状态, 违背模型正确性定义的性质 3), 模型不是正确的, 算法退出;

④ 当在状态  $M$  下, workflow 网中存在可以被激发的变迁, 对其中每个可被激发的变迁  $t$ , 循环执行下列步骤:

A. 根据经典 Petri 网的点火规则, 在状态  $M$  下, 求得激发  $t$  后 workflow 网的状态  $M'$ , 向集合  $T_R$  中添加变迁  $t$ ;

B. 如果  $M' > M_0$ , 则表示违背模型正确性定义的性质 2), 模型不是正确的, 算法退出;

C. 如果从遍历树的根  $M$  到的路径上存在一个状态  $M''$ , 其中  $M' > M''$ , 则表示存在库所  $p$ , 使得  $M'(p) > M''(p)$ , 而对于其他库所  $p'$ , 都有  $M'(p') \geq M''(p')$ , 这意味着如果继续在状态  $M'$  下执行则会使得  $p$  拥有的库所数目无限递增, 则模型是无界的, 根据引理 1, 模型不是正确的, 算法退出;

D. 如果  $M' = M_0$ , 表示已经正常执行到终止状态, 把  $M$  的标识置为 “end”, 否则把  $M'$  的标识置为 “new”, 在遍历树上加入节点  $M'$ , 并把  $M$  与  $M'$  通过标识  $t$  的连接弧相连, 把  $M'$  加入集合  $M_R$ ;

3) 如果  $T_R$  中没有包含所有 workflow 网中的所有库所, 则表示违背模型正确性定义的性质 1), 模型不是正确的, 算法退出;

4) 对  $M_R$  其中每个标识为 “repeat” 的状态  $M$ , 循环执行下列步骤:

取得从遍历树的根到  $M$  的路径上与  $M$  相同的状态  $M'$ , 如果在以  $M'$  为根的子树中, 存在标识为 “end” 的叶子, 则表示循环是有可能退出的, 否则 workflow 网执行过程中会出现无限循环, 模型不是正确的, 算法退出;

5) 模型是正确的, 算法结束。

卫星在轨测试过程模型正确性定义的 3 条性质在图 13 和图 14 中得到确认。根据上面的遍历树验证算法, 可以得到卫星在轨测试过程模型的遍历树, 如图 5 所示。在这个遍历树中, 有一个标识为 “end” 的叶子:  $M_{14}$ , 同时该遍历树的连接弧变迁标识, 已经包含了卫星在轨测试过程模型中的所有变迁, 所以卫星在轨测试过程模型是正确的。