doi: 10.3969/j.issn.1006-1576.2011.11.024

# 基于 UML 的航天测试发射仿真系统建模

王华[1]，刘培杰[2]，肖艳彬[2]
（1. 装备指挥技术学院航天装备系，北京 101416；
2. 装备指挥技术学院研究生院，北京 101416）

**摘要**：为了实现仿真系统的开放性和可复用性，建立一套直观、通用的测试发射仿真体系模型。在对航天测试发射系统的结构、各部分关系等进行全面梳理和分析的基础上，通过对现有航天发射组织指挥流程的梳理和对训练系统的需求分析，先后对系统总框架和其中的测试发射分系统进行建模分析，通过对火工品测试间爆炸事件进行静态建模和动态建模，得到了从模型到软件实现的一般方法。结果表明：该方法便于从全局上纵览整个软件结构，把握了整个软件的结构和各分系统结构间的关系。

**关键词**：统一建模语言；训练仿真系统；系统仿真；航天发射
**中图分类号**：TJ86　　**文献标志码**：A

# Modeling Spaceflight Test and Launch Simulation System Based on UML

Wang Hua[1], Liu Peijie[2], Xiao Yanbin[2]
(1. Dept. of Spaceflight Equipment, Institute of Command & Technology of Equipment, Beijing 101416, China;
2. College of Graduate, Institute of Command & Technology of Equipment, Beijing 101416, China)

Abstract: In order to achieve openness and reusability of simulation system, a set of intuitive and common simulation system model is established. Based on analyzing the structure of spaceflight test & launch system, as well as the relationship between its various parts, general system framework and the test & launch subsystems models are fulfilled by sorting out the existing organizing and commanding process and making requirements analysis. And on this basis, by modeling the simulation incident about a fire of explosive components testing house in the static way and dynamic, a general approach for implementation from the unified modeling language (UML) models to software application is raised. The results show that the approach allows an overall overview of the entire software structure, to grasp the entire structure and the relationship between the structures of the subsystems.

Keywords: UML; simulation system; system simulation; spaceflight launch

## 0　引言

随着我国航天事业的快速发展，对运载火箭测试发射人才培养提出了更高的要求。现有的基于实物的测试发射人员训练模式受多种不利因素的影响，为此，建设相应的航天发射训练仿真系统，就显得尤为紧迫。而随着信息化要求的不断深入和航天发射任务的不断发展，航天发射训练仿真系统必须要具有开放性和可复用性，才能适应不断变化的航天试验环境和航天发射任务。因而，在仿真软件的开发过程中，系统建模的作用尤为重要。作为一种可视化建模语言——统一建模语言（unified modeling language，UML）提供了丰富、严谨、扩充性强的表达方式[1]。因此，笔者将利用UML对航天发射训练系统进行详细分析。

## 1　统一建模语言（UML）[2-3]

统一建模语言 UML 是一种用于描述、可视化和构架软件系统以及业务建模的语言，涵盖了面向对象的分析、设计和实现。UML 通过提供不同形式的图形来表述从软件分析开始的软件开发全过程，一个图就是系统架构在某个侧面的表示，所有的图组成了系统的完整视图。UML 的重要内容由下列 5 类图（共 10 种图形）来定义：

1）第 1 类是用例图（use case diagram），从用户角度描述系统功能，并指出各功能的参与者。

2）第 2 类是静态图（static diagram），包括类图、对象图和包图。其中类图描述系统中类的静态结构。对象图是类图的实例。包图由包或类组成，表示包与包之间的关系。包图用于描述系统的分层结构。

3）第 3 类是行为图（behavior diagram），描述系统的动态模型和组成对象间的交互关系，包括状态图和活动图。其中状态图描述类的对象所有可能的状态以及事件发生时状态的转移条件。而活动图描述满足用例要求所要进行的活动以及活动间的约束关系，有利于识别并行活动。

4）第 4 类是交互图（interactive diagram），描述

收稿日期：2011-07-06；修回日期：2011-07-25
作者简介：王华（1971—），男，四川人，博士研究生，从事运载火箭测试发控技术研究。

对象间的交互关系，包括序列图和合作图。其中序列图显示对象之间的动态合作关系，它强调对象之间消息发送的顺序，同时显示对象之间的交互。合作图描述对象间的协作关系，与序列图相似，显示对象间的动态合作关系。除显示信息交换外，合作图还显示对象以及它们之间的关系。

5) 第 5 类是实现图(implementation diagram)，包括构件图和配置图。其中构件图描述代码部件的物理结构及各部件之间的依赖关系，包含逻辑类或实现类的有关信息。配置图定义系统中软硬件的物理体系结构，可以显示实际的计算机和设备(用节点表示)以及它们之间的连接关系，也可显示连接的类型及部件之间的依赖性。

采用 UML 对软件系统进行建模，就是用上述 5 类图将软件开发过程中的各个步骤可视化地表现出来。其中，UML 提供用例图、类图(包括包图)、对象图、构件图和配置图等 6 种图来描述系统的结构，这些图构成了 UML 的静态建模机制。而状态图、活动图、序列图和合作图等 4 种图则用于描述可以执行的或者表示执行时的时序状态或交互关系，构成 UML 的动态建模机制。

## 2　测试发射分系统建模

按照岗位类别，航天发射训练仿真系统可划分为导调人员组、试验指挥分系统、测试发射分系统、试验通信分系统以及测量控制分系统。而测试发射分系统是整个训练仿真系统的核心。限于篇幅，笔者以测试发射分系统为例展开分析建模过程。

应用 UML 开发应用系统基本框架来对仿真系统进行建模，首先进行需求分析，理解仿真系统所需要完成的功能及其软件要达到的设计目标；其次进行静态建模，用 UML 的静态图描述软件的功能、类及其相互间的联系；最后进行动态建模，用 UML 的动态图描述软件中对象之间的交互关系。

### 2.1　系统需求分析

任务前期，由导调人员通过想定系统确定任务背景、发射场选择、火箭型号选择、演练日程安排等想定内容，作为航天发射训练仿真系统的系统输入。在任务实施过程中，参训者在导调人员的调理下，按照现有的发射场组织指挥流程模拟执行航天发射任务，期间导调人员可以按照想定预设触发特定的突发事件，考察参训者对其处理能力。导调人员可根据其表现对参训人员进行打分，同时系统也会按照预设逻辑对其操作进行自动评分。在任务结束后，评分系统依据评分规则表对各评分记录进行计算，得出该参训者的最终评定值，包括对其各项能力给出对应的评定值。

根据软件需求和实际功能，软件模块可分为导调管理模块、想定编辑模块、任务实施模块、评分模块和突发事件处理模块 5 部分。仿真系统逻辑结构如图 1。

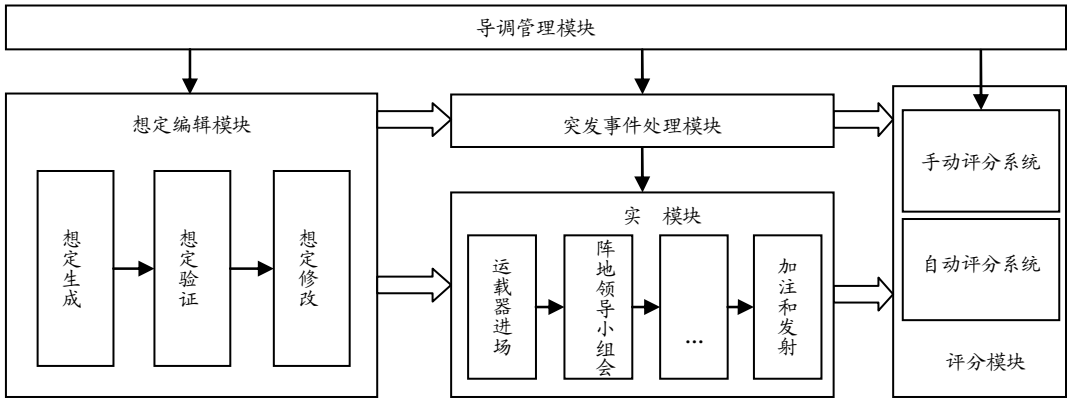


**图 1　仿真系统逻辑结构**

### 2.2　静态建模

根据上述需求分析，可以首先进行用例建模。用例建模是使用业务实践、发起业务事件的人以及系统如何响应这些事件来对系统功能进行建模的过程，是系统设计和开发的基础[4]。通过对航天测试发射仿真系统的功能需求分析，再结合参训岗位所属分系统的划分，可以构建出仿真实体用例模型。

图 2 所示为测试发射分系统用例图。测试发射分系统共有 14 个岗位，其中 01 指挥员、后勤岗位、装备岗位、工程师岗位、发射站参谋岗位、其它岗位 1、其它岗位 2 以及其它岗位 3 这 8 个岗位可以

抽象为发射站指挥所子系统，因为它们具有共同的属性和方法，在图中用泛化标记表明了其关系，即用一个三角箭头从子角色指向了父角色。类似地，11系统指挥、12系统指挥、13系统指挥、14系统指挥、15系统指挥、16系统指挥这6个角色抽象为了各子系统指挥角色。

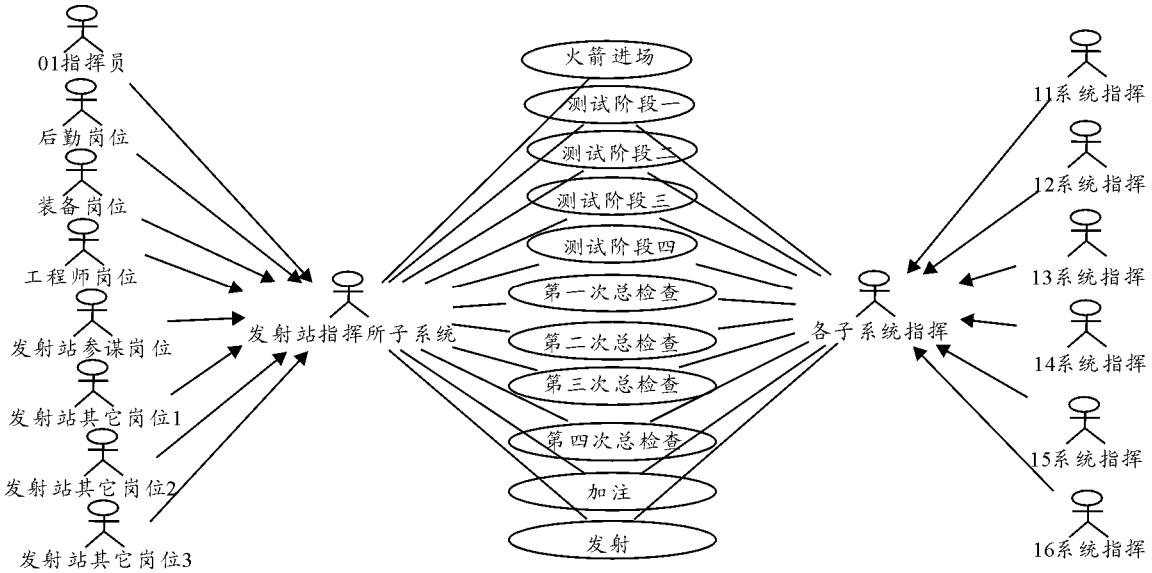


图2 测试发射分系统用例图

在UML中，泛化关系的含义是把某些参与者的共同行为抽取出来表示成通用行为，且把它们描述成为抽象参与者[5]。泛化是所有面向对象设计方法中都提供的一个概念，它和普通面向对象程序设计语言中的继承概念有着密切的关系。如将发射站指挥所的8个成员各自看作一个类，再加上它们的父类，就可以构建如图3所示的测发分系统部分类图。这样，图2中的角色泛化关系转变为了图3中类之间的继承关系。为便于分析阐述，类名改为了中文字符，实际工程中应该避免。

由于发射站指挥所子系统是事实上并不存在的一个角色，所以在图中用斜体字书写类名，表明它是一个抽象类。它拥有的一个属性ID和2个方法：SendOrder(发送指令)、SendFile(发送文件)，是8个类共同拥有的公共特性，可直接被继承，避免了重复定义。但由于01指挥员需要下发某些关键指令，父类中的SendOrder方法对其不再适用，故在该类中对SendOrder方法进行了覆盖，这和面向对象程序设计中的重写概念是一致的。

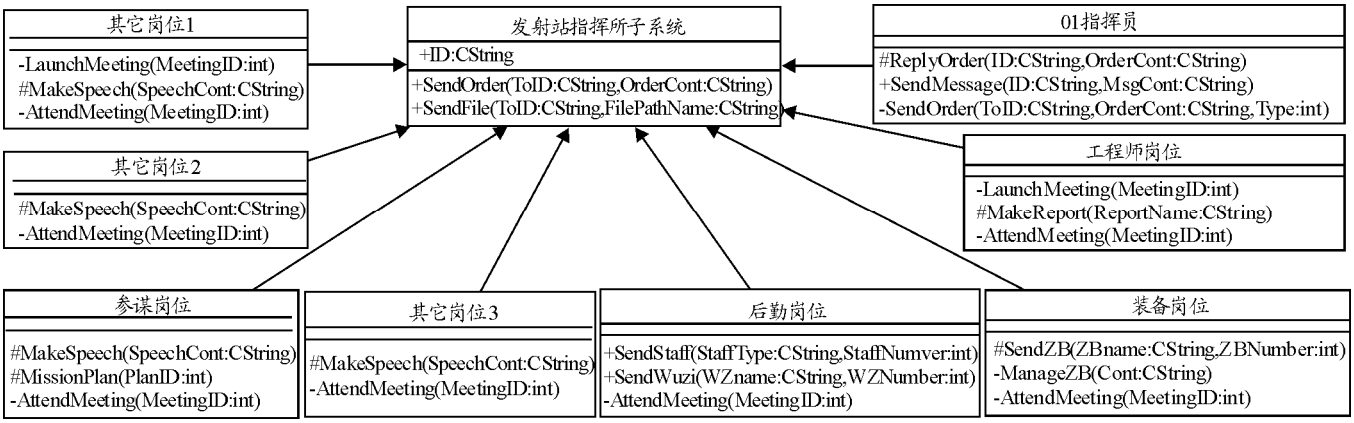


图3 测试发射分系统部分类图

## 3 突发事件处理流程建模

在航天发射训练仿真系统中，突发事件处理模块是不可或缺的一部分。通过在仿真系统中设置突发事件处理模块，不但可以考察参训者的综合能力，还能验证突发事件处置方式的有效性和合理性，探索应对突发事件的新举措。以技术区测试时火工品测试间爆炸这一突发事件为例，进行分析建模。

根据想定情况，预设的突发事件完整处理流程

如下：

1) 导调人员触发该突发事件；

2) 14 系统指挥按照预案实施对应紧急措施，如抢救伤员、紧急断电、灭火等；

3) 14 系统指挥将火情上报给 01 指挥员；

4) 01 指挥员收到火情汇报后，向装备岗位下达派发物资指令，向后勤岗位下达派车指令，同时赶往事发地点；

5) 发射站装备岗位收到指令后派发保障物资，实施物资运输；

6) 发射站后勤岗位收到指令后派车，实施人员运输；

7) 发射站工程师岗位、发射站参谋岗位、12 系统指挥、13 系统指挥乘车赴事发地点；

8) 当所有人到场后，01 指挥员执行灭火，并将火灭信号发给导调人员；若某些岗位操作失误或者不及时，将导致火势扩大，事件处置以失败告终，上报导调人员；

9) 若事件处置成功，火已熄灭，发射站工程师岗位撰写事故原因分析报告，上传给导调人员。至此该事件流程结束。

按照上述流程，可构建出涉及该事件的基本实体有：导调人员、14 系统指挥、01 指挥员、发射站后勤岗位、发射站装备岗位、发射站工程师岗位和其他相关岗位。它们构成了该突发事件流程的类图，如图 4 所示。其中“其他相关岗位”指的是发射站参谋岗位、12 系统指挥、13 系统指挥，因为在此事件中它们具有相同的属性和方法，将它们封装为了同一个类。

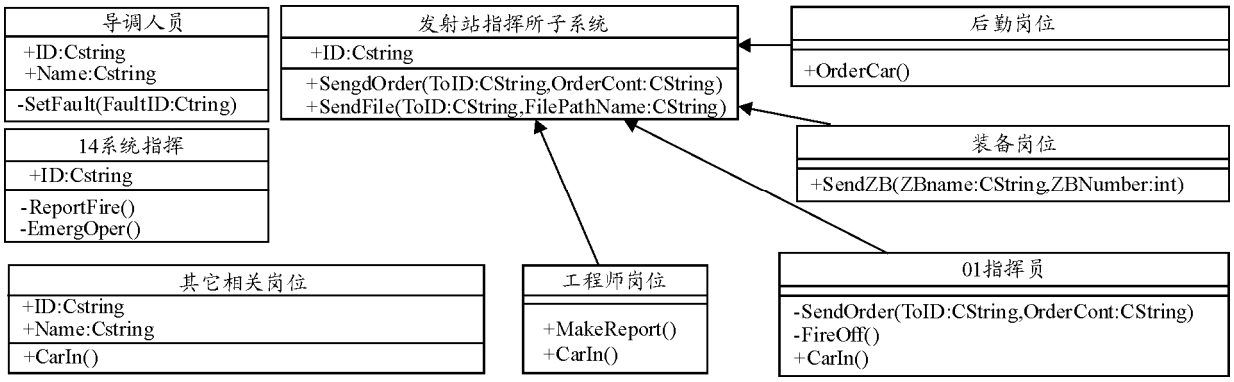


**图 4　火工品测试间爆炸突发事件类图**

导调人员是此事件的触发者，即该小系统的输入。导调人员类的属性有 ID 号和导调人员名称。ID 号用来对仿真系统中所有出现的岗位进行唯一标识；人员名称即岗位名称，是参训者扮演的角色名称。导调人员类具有 SetFault (FaultID: Cstring) 方法，该方法用来触发事件，其中的参数 FaultID 即突发事件的 ID 号。

14 系统指挥是突发事件的发现者和第一操作者。该类具有 ReportFire()(上报火情)和 EmergOper()(紧急措施)2 个方法。

01 指挥员类作为该小型指挥系统的核心，具有下达关键性指令的方法。通过设置 SendOrder (ToID: CString, OrderCont: CString)方法的 2 个参数，可以给装备岗位和后勤岗位下达不同的指令信息，组织应急分队按照预案顺利实施事件处理。

发射站装备岗位在该事件中具有派送装备的方法，在类中表现为 SendZB()函数。发射站后勤岗位在该事件中具有派发车辆的方法，在类中表现为 OrderCar()函数。发射站工程师岗位在该事件中具有撰写事故分析总结报告的方法，在类中表现为 MakeReport()方法。

以上这些类由于方法各异，各实例化为一个对象，而“其他相关岗位”则实例化为 3 个对象：参谋岗位、12 系统指挥、和 13 系统指挥，它们具有共同的一个方法 CarIn()，即登车。对象图未单独列出，因为序列图的最上面一行就是各个对象，如图 5 所示。

序列图用来描述对象之间消息发送的先后次序，阐明对象之间的交互过程以及在系统执行过程中的某一具体时刻将会发生什么事件。如图 5 消息沿纵轴按时间顺序排列。图中消息的序号表明了其时间顺序。

（下转第 96 页）