# “嫦娥五号”探测器GNC应用软件高可信研制技术

李 轶，黎 藜，郭明姝，王同磊，张国峰，李晓锋
（北京控制工程研究所，北京 100094）

**摘 要：** 基于“嫦娥五号”（Chang’E-5，CE-5）任务高安全性、高可靠性、高复杂度、高自主性的功能以及高实时性、强时序性的需求，开展了导航、制导与控制（Guidance，Navigation and Control，GNC）分系统应用软件高可信研制保障技术研究。针对自然语言需求定义方式无法精确描述一些关键复杂时序的问题，在需求分析阶段建立了基于时序安全性属性描述的形式化建模语言模型验证技术，保证了系统时序的安全性；针对人工走查难以发现的代码深层次脆弱性缺陷，在设计编码阶段结合飞行任务剖面提取了程序切片，提高了源代码缺陷定位效率，保障了编码的规范性与软件构件的功能正确性；针对复杂软件的海量测试用例无法快速执行的问题，在确认测试阶段，研究了基于状态图和序列图的测试用例生成方法，搭建了一键测试的自动测试系统，实现了海量测试用例的快速自动执行，有效提升了测试效率与测试覆盖性。通过各阶段地面仿真实验和在轨飞行试验验证，表明所提出的高可信软件研制保障技术方法有效可行。

**关键词：** 嫦娥五号；GNC分系统；应用软件；高可信
**中图分类号：** TP311.1 **文献标识码：** A **文章编号：** 2096-9287(2021)03-0244-08
**DOI:**10.15982/j.issn.2096-9287.2021.20200065
**引用格式：** 李轶，黎藜，郭明姝，等. “嫦娥五号”探测器GNC应用软件高可信研制技术[J]. 深空探测学报（中英文），2021，8（3）：244-251.
**Reference format:** LI Y，LI L，GUO M S，et al. High confidence development technology of application software for GNC subsystem of Chang’E-5[J]. Journal of Deep Space Exploration，2021，8（3）：244-251.

## 引 言

制导、导航与控制（Guidance Navigation and Control，GNC）分系统是“嫦娥五号”（Chang’E-5，CE-5）探测器的重要组成部分，具有高实时性、高可靠性、高复杂性等特点。应用软件作为GNC分系统的灵魂，承担全过程的制导、导航以及姿态与轨道控制功能，属于安全攸关的软件。相比我国以往的月球探测任务，“嫦娥五号”任务更复杂、技术难度更高，软件产品的高可信性关乎任务成败。

深空探测任务的超远距离测控和超长通讯时延，以及空间环境的复杂性和不确定性都使得软件面临更加严峻的可信性问题[1]，软件一旦在轨出现异常，将会导致不可逆的后果，因此软件可信性关乎任务的成败。随着深空探测任务的自主智能化程度日益提高，应用软件的规模和复杂度也在不断提升[2-3]。在不依赖或者少依赖地面的情况下，准确地感知自身的状态和外部环境，通过自主任务规划的方式合理安排动作序列、分配星上资源，自主完成任务目标，已成为当前深空探测器软件的主要发展目标[4-5]；同时如何在有限的计算资源上最大限度地提高算法执行效率，降低时间空间复杂度，也是软件面临的主要约束[6]。近年来，依托我国的月球探测工程和火星探测工程，国内深空探测器GNC分系统应用软件研制能力取得了显著提升，软件规模和复杂度也在不断增加。以“嫦娥四号”（Chang’E-4，CE-4）巡视器控制分系统的应用软件为例，软件规模接近10万行，并具备自主任务规划和管理等多种高复杂度功能。

相比以往的任务，“嫦娥五号”软件研制面临更为复杂的挑战：①高安全性和高可靠性，近月制动、着陆下降、起飞上升、交会对接、再入返回等关键事件都属于不可逆任务，需软件全自主处理，一旦出现异常，将直接导致任务失败；②高复杂度和高自主性，探测器面临全新的液体晃动条件、环月轨道受晒条件，以及帆板挠性参数未知、测控盲区长、落点选取精度高等多种因素，软件设计需满足任务要求的最大包络范围并自主应对，复杂度高；③高实时性和强时序性，探测器由着陆器、上升器、返回器、轨道器四

收稿日期：2020-09-18 修回日期：2021-01-13
基金项目：国家自然科学基金资助项目（61802017）

器组成，各器之间存在形式多样且时序依赖的数据流交互，涉及多器联合的关键任务时序逻辑十分复杂。

面对上述复杂需求，现有的软件研制技术体系难以满足“嫦娥五号”任务的高可信保障要求。根据航天型号软件工程化标准[7]，软件开发环节中技术类的过程主要包括需求分析、设计编码与测试等3个阶段。3个阶段相互依赖相互影响，每个环节都对软件产品的可信性具有重要意义，必须在所有环节开展可行性保障措施，才能最终实现软件的高可信。以往的软件研制体系中，上述3个阶段的研制过程主要存在以下问题。

基于自然语言描述的需求分析和基于结构图与流程图的架构设计方法，无法精确量化地定义一些复杂时序逻辑，需求和描述二义性问题日益显著。关于需求的形式化建模方法，顾斌等[8]研究的模型仅关注了离散时间的动力系统；谭彦亮等[9]对中断管理的上下文保护进行了需求层和设计层的建模，周育逵等[10]对操作系统周期性任务队列管理进行了需求形式化建模，但两者都是针对典型的操作系统应用场景，无法精确描述基于模式转换的有限状态机自主迁移模型。针对上述问题，本文对上述方式中的建模语言进行了改进，加入了时序安全性属性，用于精确描述状态迁移前后的时序特性变化，可精确分析模式转换前后的时序安全性。

在设计编码阶段，基于人工走查和静态分析工具的源代码检查，无法从飞行任务剖面的角度定位一些深层次的脆弱性缺陷。李雷等[11-13]研究的基于聚类分析的软件多故障定位方法，对测试用例依赖性较强；王同磊等[14]研究的软件脆弱性自主定位方法，模型较为简单，没有结合实际的飞行任务过程建立数据流和控制流的约束刻画与依赖关系，导致定位状态空间膨胀，定位搜索过程时间开销大，无法直接应用于型号任务软件。本文研究了一种基于程序切片的代码缺陷定位理论和技术，并结合“嫦娥五号”任务飞行程序提炼了安全关键飞行任务剖面，在此基础上构建了基于典型飞行任务剖面的代码语句依赖关系图，有效提高了缺陷检测与定位效率。

面对任务需求导致的软件规模和复杂度的提升，常规的人工测试手段已无法满足“嫦娥五号”高自主性软件的海量测试用例快速自动测试需求。Yang等[15]介绍了国外通用的自动测试语言及应用情况，但对于测试系统的实现和测试脚本的生成没有展开描述。本文提出了一种基于状态图和序列图的测试用例生成方法，并建立了海量测试用例的一键快速自动测试系统。

本文提出的技术在“嫦娥五号”GNC分系统[16]应用软件研制过程中得到了应用验证，有力保证了软件产品的高可信性，实现了“多个首次”。

# 1　面向关键时序的形式化需求建模

## 1.1　基于建模语言的需求模型

GNC软件是一种时序性极强的软件系统，具有周期性、计算复杂、逻辑状态组合繁多、模式切换条件复杂等特点。因此对其开展的需求建模技术必须要具备时序性的特点[8]。

传统的基于自然语言描述的需求定义方式仅能描述顺序执行或并发执行的软件行为，机械地加入时序性逻辑会破坏其原有的可组合性。因此有必要引入时序性需求建模技术，确保软件需求得到清晰、无二义性的表述，并能在需求分析阶段进行分析和验证，以便尽早发现可能存在的问题，有效地提高软件质量。

顾斌等[8]描述的面向航天控制领域的需求描述语言SPARDL（SPAcecraft Requirement Description Language），适用于基于模式的、含有多种状态和有限周期的控制系统，与状态图类似，基于事件驱动状态变迁[17]。同时SPARDL在控制结构的描述上提出了自己的规范，对于一个控制系统模型，SPARDL用一系列模式来体现控制系统中的周期性行为特征，每个模式可以周期性地执行一系列的过程，系统可以周期性地运行在某个模式中，直到满足设定的某个迁移条件，系统可自主转入下一个模式。模式定义如表1所示。

**表 1　模式管理语法**

**Table 1　Syntax of mode management**

| 元素 | 条件 | 执行实体 |
|---|---|---|
| Mode | ::= | <ID，Entry，（Proc\|Mode$^+$），Trans> |
| Entry | ::= | stmt |
| Proc | ::= | stmt |
| Trans | ::= | <priority，condition，Action，target> |
| Action | ::= | stmt |

表1中，ID代表模式编号，Entry代表模式入口初始化，Proc表示模式处理过程，Trans表示模式迁移，模式迁移包括优先级、条件、迁移操作和目标模式。模式可以由若干子模式构成。上述语言可以精确描述周期性控制系统的实时状态转移过程，但无法描述模式转换前后周期时序特性发生变化的情况，尤其针对执行安全关键任务的特殊模式，如果没有完整的时序特性需求定义，则不能得到充分验证。本文针对上述情况，引入了模式时序特性Seq这一属性，用于定义变周期控制任务的模式周期特性，在每个模式的Entry控制流中，包含对Seq的初始化设置，在模式转换过程增

加如下验证规则RuleVery：

IF（Mode$^{+}$ - > Seq == definedSeq）

{ Trans = TRUE；}

ELSE

{ Trans = FALSE；}

根据规则，只要事先定义好各模式和子模式的时序特性，就可以对所有模式转换路径建立规则检查，确认模式转换的时序设置是否完整正确。

### 1.2 “嫦娥五号”轨道器与上升器的自主交会对接

图1给出了“嫦娥五号”任务中一个典型的模式转换工况——轨道器与上升器的自主交会对接。对于此类时序功能，原始输入形式通常以自然语言描述或是伪逻辑代码的方式为主，经过初步的需求分析提炼形成状态转换图。

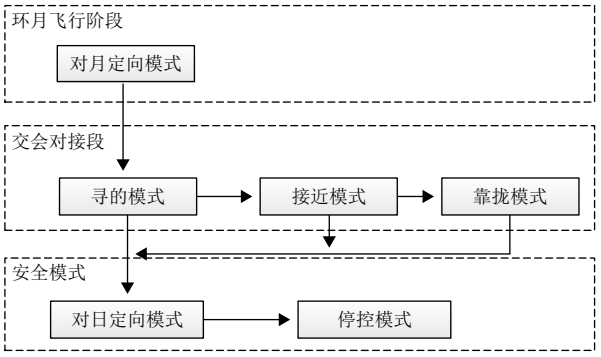


图 1 交会对接过程模式转换示意图

Fig. 1 Schematic diagram of mode transformation in rendezvous and docking process

1）环月飞行阶段，轨道器根据目标相对距离测量信息自主转入交会对接阶段，或根据地面遥控指令转入交会对接阶段；

2）交会对接阶段，轨道器根据目标相对距离测量信息，依次完成寻的模式、接近模式和靠拢模式的自主切换，也可根据地面指令转入相应的模式；

3）交会对接阶段，如果发生姿态异常等故障，则自主转入对日定向模式，对日定向模式期间如果发生推力器故障，则自主转入停控模式。

基于上述需求，对其中的寻的模式进行功能拆分与建模，采用与SPARDL配套的图形化工具生成的子模式转换状态图如图2所示。

根据图2的定义，寻的模式有5个子模式，其中寻的变轨子模式为快周期任务模式，其余4个子模式为慢周期任务模式；由Cond2条件转入的对日定向模式，也为慢周期任务模式。条件Cond2的隐含路径包含5条（即分别从5个子模式直接进入对日定向模式），从寻的变轨之外的4个子模式转入，则时序特性不会发生变化，从寻的变轨模式转入，则存在时序特性变化，如果在Entry元素中没有进行相关的时序特性设置，则会导致新的模式（对日定向模式）时序错误，引发安全性问题。按照本文提出的改进建模语言，对模式转换前后的时序特性属性进行规则检查，如果从寻的变轨子模式转出至对日定向模式遗漏了时序特性设置，则可以有效识别并报警。

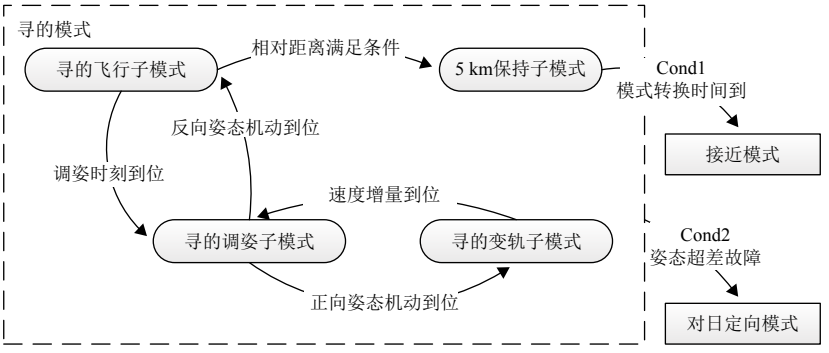


图 2 寻的模式状态转换图

Fig. 2 State flow in home mode

基于上述方法和相应的配套工具，在“嫦娥五号”GNC分系统应用软件研制中开展应用，在软件需求分析阶段共提炼功能需求点534项，对其中86项涉及时序特性转换的功能点应用上述方法开展形式化建模，后续阶段的测试和第三方评测数据表明，上述86项需求模型准确可信，没有因遗漏或错误导致的时序安全问题，有效避免了需求分析问题遗留到后续阶段。软件产品在轨表现进一步表明该方法有效提高了软件的可信性。

## 2 基于程序切片的代码缺陷定位

采用人工走查和静态分析工具扫描，可以保证源代码符合基本的编程规范，也能避免低层次错误，但无法定位一些深层次的脆弱性缺陷，这种缺陷一旦在

在轨运行中被激活利用，会导致不可预期的后果[14]。

本文设计了一种基于程序切片的源代码分析方法，并结合飞行任务关键事件构建了不同维度的典型飞行任务剖面，有效缩减了程序语句的数据依赖和控制依赖代码子集范围，提高了可疑代码探查效率，实现了源代码缺陷快速定位。

### 2.1　程序切片的构建

程序切片是程序语句集合的子集，由程序中的部分或者全部语句构成，包含的这些语句能够对某条语句$s$中特定变量$x$的值产生影响，其中将语句$s$称作兴趣点，变量$x$称为兴趣变量。给定源程序$p$、兴趣点$s$和兴趣变量$x$，其程序切片是源程序语句集合的一个子集，该语句子集中的元素：当程序运行到兴趣点$s$时，$p$中所有直接或间接对兴趣变量$x$有影响的语句[18]。

基于程序切片的故障定位技术主要的实现原理是根据程序执行情况选择和错误输出相关的所有语句或者通过在程序中进行插桩以构建程序依赖图来进行定位。

### 2.2　基于飞行任务剖面的缺陷定位

构造合理的数据依赖和控制依赖关系，是程序切片定位效率的关键因素。有效精准的程序语句的数据依赖和控制依赖，可以将整个源程序缩小到程序语句的子集中，缩小定位的排查范围，从而提高定位效率[13]。

本文结合飞行任务提取了各飞行阶段的飞行任务剖面，按照当前所处剖面的实际数据流、时序特性以及外部环境等特性构建程序数据依赖和控制依赖关系，形成当前剖面的子集，作为缺陷定位的程序切片。飞行任务剖面的提取过程如下。

1）飞行阶段识别

根据“嫦娥五号”任务特点，GNC系统的飞行阶段划分为入轨阶段、地月转移阶段、近月制动阶段、组合体分离阶段、环月飞行阶段、交会对接阶段、月地入射阶段、轨返分离阶段等若干阶段，不同阶段的控制策略、控制参数以及时序特性各不相同。

2）组合体构型识别

“嫦娥五号”探测器是四器组合体，随着任务过程的推进，组合体会发生若干次分离、对接，其构型存在“四器”“三器”“两器”“单器”等多种工况，每种构型的控制逻辑与控制参数都不相同。

3）时序特性识别

根据任务规划设计，不同任务过程GNC系统的时序特性有所区别，有短周期任务、长周期任务、随机异步事件任务等，不同时序特性下的数据依赖关系各有不同。

根据上述方案进行飞行任务剖面的提取，就可以建立精准刻画当前数据依赖和控制依赖的程序切片，进而完成源代码缺陷定位，具体实现步骤如下：①以源程序为基础并对源程序进行预处理，通过程序依赖性分析，得到程序的静态控制依赖图；②基于某特定的任务剖面，以某观测量的预期结果为输入，根据静态控制依赖图以及预处理后的程序在测试用例下的执行，计算得到程序切片；③由程序切片构造程序切片谱，根据选定的统计量及可疑度计算公式，计算程序切片中各语句的缺陷可疑度，从而定位代码缺陷。

以“嫦娥五号”GNC分系统应用软件源代码作为试验对象（ANSY C语言），通过词法分析和语法分析构建程序中的控制依赖关系（包括if语句块、switch语句块、for语句块、while语句块、do-while语句块）；对源程序进行插桩以获取程序运行时的动态信息，包括执行的语句行号及定义、引用的变量集。在专用平台上使用预先设计好的测试用例运行预处理之后的目标源代码，同时基于前向计算动态切片的算法，计算程序的动态切片，并记录可疑缺陷位置。

通过上述方法，有效识别出了走查时遗漏掉的源代码设计缺陷，现从中挑选2个典型案例：

**案例1**　特殊时序下的不可达分支

伪代码抽象如下：

```
IF（条件1 && 条件2）
{ 执行功能1 }
```

在条件1和条件2都满足的情况下执行功能1。但计算系统为多机架构，在特殊时序下，有可能出现分别在不同的周期实现条件1满足和条件2满足，这样就无法保证在同一个周期内条件1和条件2同时满足。该工况在走查和静态分析阶段没有识别到，在测试阶段也难以发现，采用本文的技术有效识别出了这一疑似缺陷。

**案例2**　代码升级不充分导致的不可达分支

伪代码抽象如下：

```
IF（工况1）                    语句块1
{ Flag = 1 }
IF（Flag == 1）                语句块2
{ 转入安全模式 }
```

设计本意：在工况1发生时，立Flag标志为1（语句块1），在执行到语句块2时，判断如果Flag标志立为1，则转入安全模式。

在某次版本升级时，任务总体提出了新的要求——在某个特定模式下，工况1发生时，Flag标志立为2（遥测判读需要）。跟进这一需求，只对语句块1进行了修改，但并未识别到这一更改对语句块2的影响

域，这一更动的后果会导致语句块2成为不可达分支。采用本文的技术有效识别出了这一缺陷。

为了进一步验证本方法的有效性，在“嫦娥五号”GNC分系统应用软件研制阶段，选取软件产品源代码作为实验对象，结合5种典型的飞行任务剖面，人为设置了类似于前文案例的5个缺陷。利用基于本方法的工具进行源代码扫描，结果表明5个缺陷全部被检测出，且定位消耗时间较参考文献中的方法有了显著改善，结果如表2所示。

**表 2 代码缺陷检测实验结果**

**Table 2 Result of source code fault localization**

| 任务剖面 | 预埋缺陷数 | 扫描结果 | 定位耗时降低/% |
|---|---|---|---|
| 对日巡航 | 1 | 正确 | 23 |
| 姿态机动 | 1 | 正确 | 31 |
| 轨道修正 | 1 | 正确 | 33 |
| 近月制动 | 1 | 正确 | 19 |
| 对月定向 | 1 | 正确 | 25 |

上述代码缺陷定位方法，已经在“天问一号”（Tianwen-1）火星探测器巡视器GNC软件和在研的“嫦娥六号”（Chang’E-6，CE-6）探测器GNC软件研制中得到了推广应用，有效地提高了软件源代码的产品质量。

## 3 海量测试用例的快速自动测试

伴随月球探测任务的复杂度的日益增大，GNC分系统应用软件的测试用例数量也呈现井喷。以“嫦娥五号”GNC分系统应用软件为例，全功能版本的测试用例数已接近1万个。

为了实现对海量测试用例的快速有效覆盖，本文设计了海量测试用例的快速自动测试系统。

### 3.1 全数字虚拟仿真验证测试环境

该测试平台利用软件仿真技术，模拟物理硬件目标系统所构成的软件仿真的虚拟目标系统，原本运行于真实目标系统上的嵌入式软件，可以不加修改直接在软平台上运行，并且其运行的动态特性与在真实目标机上一致。

在软平台中利用CPU模拟器、虚拟芯片、虚拟内存构成虚拟目标机。其中CPU模拟器对目标CPU的内核进行模拟，将目标机的指令转换为宿主机上的指令执行，从而实现在宿主机上执行目标机上的代码；虚拟芯片对IO芯片进行软件仿真，包括串口、并口、中断控制器、1553B总线控制器等；虚拟内存对内存芯片进行仿真，虚拟目标机与被控对象仿真以及其它子系统仿真一起构成虚拟目标系统。

全数字虚拟仿真验证测试环境配合其它软件工具，如调试器、人机界面、测试自动化工具等可以构成虚拟的开发、测试环境。由于整个环境全部由软件实现，克服了基于真实目标机的环境所固有的运行状态难以控制和监视、调试不便、故障注入困难等缺陷，因而能够为软件开发、测试提供更为理想的环境。

### 3.2 快速自动测试系统

在全数字虚拟仿真验证测试环境下，自主研发了星载软件全自动测试平台（Full-Automatic Spacecraft software Testing suite，FAST），达到一键测试的目的。

FAST由主应用程序（FAST）、数据处理平台和测试仿真环境3部分组成，其组成、功能和数据交互逻辑如图3所示。该平台提供脚本编辑界面、基于SVN的脚本管理、脚本批量执行、结果自动判读和每日自动分析测试数据（脚本数、通过数、未通过数、问题数、测试执行时间、测试覆盖星时、测试覆盖功能点数等）。

### 3.3 自动测试语言

软件自动化测试主要采用测试语言完成针对目标应用程序的测试。在自动测试技术发展的过程中出现了许多测试语言[15]。随着测试需求的不断增长，测试语言在实际应用中也存在一些不足之处：

1）测试成本较高，达不到节约资源的效果，如ATLAS语言的开发工具十分昂贵，测试人员学习ATLAS语言还需要额外的费用，且培训周期长；

2）测试语言中提供的程序控制语句不够完善，以ATLAS语言中跳转语句为例，跳转语句的实现是指向步骤数的，这样给修改程序尤其是添加代码带来很大不便[19]；

3）测试语言的灵活性与通用性之间存在矛盾，如GOAL语言在卫星测试领域中应用评价很好，但不具备经过简单修改应用到其它领域的能力。

为了解决以上问题，在分析GNC软件自动测试语言需求的基础上，设计了一种GNC自动测试语言，具有易于掌握、维护成本低、应用灵活的特点，可在不同领域的航天飞行器控制软件的自动测试中进行应用。

1）测试需求

（1）与自动测试系统的类型无关，具有较强的可移植性；

（2）对测试行为逻辑的定义与描述要精确、简介；

（3）可根据不同使用环境进行测试行为逻辑的扩充；

（4）要对测试过程有灵活的监控能力；

（5）测试语言要易维护；

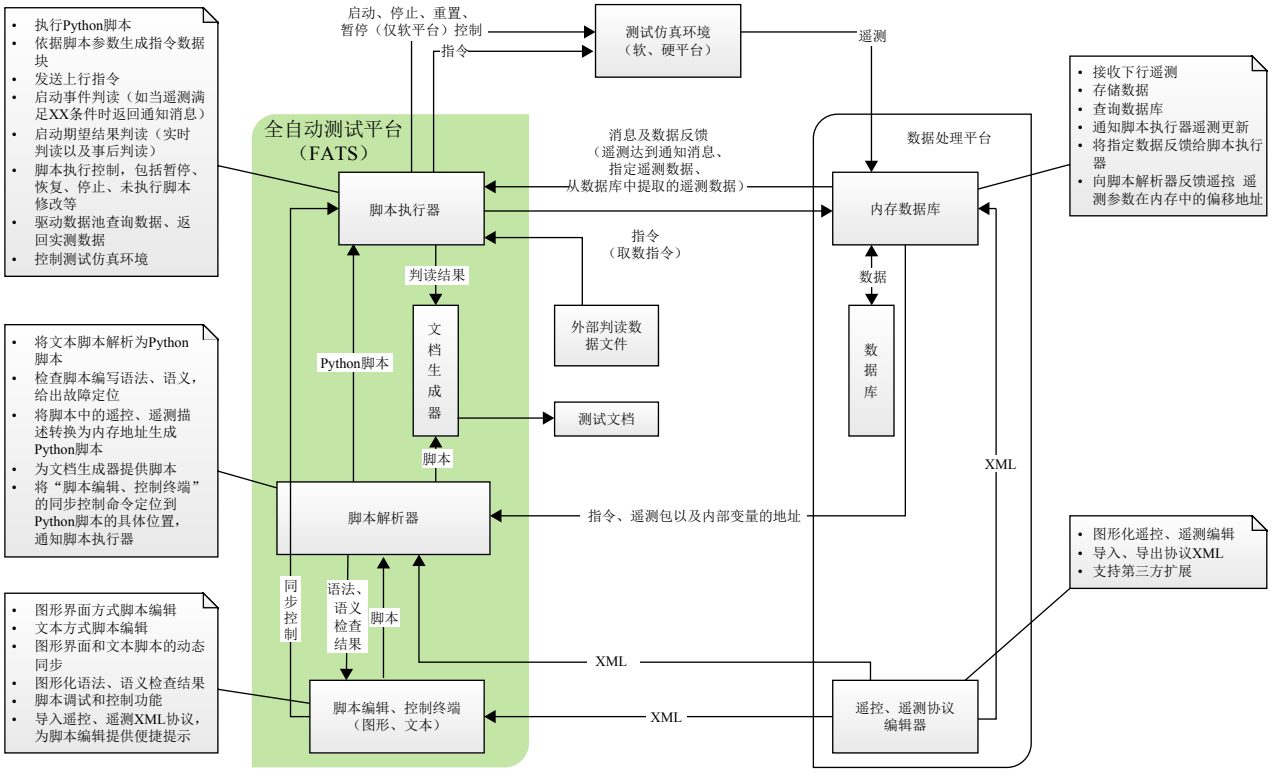


图 3　FATS系统示意图

Fig. 3　FAST system architecture

（6）测试语言的语法要直观、简单、易懂。

2）测试语言逻辑设计

为了满足以上需求，设计了一种控制软件自动测试语言，该测试语言在Python语言[20]的基础上，针对自动测试行为逻辑[21]构建了自动测试所需要的通用测试脚本模型库和专用测试脚本模型库。测试人员可通过调用相关操作逻辑的脚本模型，编写出测试用例脚本。在某个领域中，某些专用测试行为逻辑模型也可转化为通用测试行为逻辑模型，供测试人员在该领域的任何软件测试中应用。其中，通用测试行为逻辑为6大类，如图4所示。

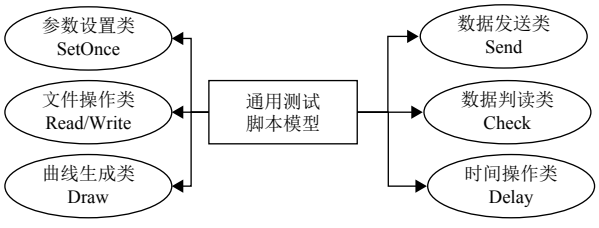


图 4　通用测试行为逻辑

Fig. 4　General test behavior logic

为了实现对不同功能的GNC软件进行自动测试，以及适应不同测试环境，需要对现有的通用测试行为逻辑模型进行扩充，形成专用测试脚本模型库。专用测试行为逻辑模型可以是对现有的行为逻辑的补充，封装为使用更为便利的模型，也可以是新定义的行为逻辑，实现新的测试行为。

针对GNC软件特点，大量参数是向量形式，逐个进行参数设置工作量巨大，因此在参数设置类行为中构建数组设置模型。在发送类行为中，指令的发送通常分为一次性发送和循环发送，Send通用模型解决一次性发送需求，对于GNC软件测试场景中经常遇到的循环发送需求，设计循环发送模型和停止循环发送模型。判读类行为是测试脚本模型中的重要模型，它实现了遥测信息的机器自动判读，替代了传统的人工判读，使整个测试过程有了灵活的监控能力。该模型可实现大量的数据判读，且能对更多的数据细节进行监控。现有的Check模型可实现对数据的一次判读，为了便于描述连续判读行为，构建循环判读模型、停止循环判读和停止所有循环判读模型。

本文选取“嫦娥五号”GNC分系统应用软件的历史用例作为实验对象。该软件配置项研制过程共计生成10个版本，测试用例共计9 846项，测试工作量600人天。本文对上述测试用例进行了筛选剔除，选取了2 000条典型测试用例，使用本系统开展自动测试实验，包含测试脚本编辑时间在内，共花费10个工作日，且测试结果准确可信，统计数据如表3所示。实验结果表明本方法有效解决了复杂任务软件的海量测试用例快速自动测试问题。

表 3 测试工作量统计

Table 3 Test workload statistics

| 软件版本 | 原测试用例数 | 原测试工作量/（人天） | 自动测试用例数 | 自动测试工作量/（人天） |
|---|---|---|---|---|
| 2.00 | 4 852 | 295 | 986 | 9 |
| 2.01 | 840 | 51 | 171 | 2 |
| 2.02 | 53 | 3 | 11 | 1 |
| 2.03 | 42 | 2 | 9 | 1 |
| 2.04 | 360 | 22 | 73 | 0.5 |
| 2.05 | 115 | 7 | 23 | 0.5 |
| 2.06 | 125 | 8 | 25 | 0.5 |
| 2.07 | 2 840 | 173 | 577 | 5 |
| 2.08 | 584 | 36 | 118 | 1 |
| 2.09 | 35 | 3 | 7 | 0.5 |
| 总计 | 9 846 | 240 | 2 000 | 20 |

## 4 结 论

本文面向“嫦娥五号”GNC分系统软件高安全性高可靠性的任务需求、高复杂度高自主性的功能需求以及高实时性强时序性的性能需求，开展了GNC分系统应用软件高可信研制保障技术研究。

本文的软件高可信构造与验证技术成功应用于“嫦娥五号”GNC分系统应用软件研制的全过程，任务在轨飞行23 d，软件零缺陷表现优异。结果表明，软件产品高可信研制技术有效可行，将在未来深空探测任务中推广应用。

## 参 考 文 献


[1] 沈国华，黄志球，谢冰，等. 软件可信评估研究综述：标准、模型与工具[J]. 软件学报，2016，27(4)：955-968.
SHEN G H，HUANG Z Q，XIE B，et al. Survey on software trustworthiness evaluation：standards，models and tools[J]. Journal of Software，2016，27(4)：955-968.

[2] 何熊文，郭坚，李玉庆，等. 深空探测器自主监控管理需求及其软件架构[J]. 控制理论与应用，2019，36(12)：2065-2073.
HE X W，GUO J，LI Y Q，et al. Autonomous health management requirements and software architecture for deep space probe[J]. Acta Automatica Sinica，2019，36(12)：2065-2073.

[3] 韩勇. 基于VxWorks的深空探测器姿轨控系统软件设计[D]. 哈尔滨：哈尔滨工业大学，2008.
HAN Y. The software of attitude and orbit control system ofdeep space probe based on VxWorks design[D]. Harbin：Harbin Institute of Technology，2008.

[4] 姜啸，徐瑞，朱圣英. 基于约束可满足的深空探测任务规划方法研究[J]. 深空探测学报(中英文)，2018，5(3)：262-268.
JIANG X，XU R，ZHU S Y. Research on task planning problems for deep space exploration based on constraint satisfaction[J]. Journal of Deep Space Exploration，2018，5(3)：262-268.

[5] 姜啸，徐瑞，陈俐均. 深空探测器动态约束规划中的外延约束过滤方法研究[J]. 深空探测学报(中英文)，2019，6(6)：586-594.
JIANG X，XU R，CHEN L J. Research on extensional constraint filtering method based on dynamic constraint sets[J]. Journal of Deep Space Exploration，2019，6(6)：586-594.

[6] 金颢，徐瑞，崔平远，等. 基于状态转移图的启发式深空探测器任务规划方法[J]. 深空探测学报(中英文)，2019，6(4)：364-368.
JIN H，XU R，CUI P Y，et al. Heuristic search based on state transition graphs for deep space task planning[J]. Journal of Deep Space Exploration，2019，6(4)：364-368.

[7] 中国人民解放军总装备部. GJB 5000A-2008，军用软件研制能力成熟度模型[S]. 北京：总装备部军标发行部，2008.

[8] 顾斌，董云卫，王政. 面向航天嵌入式软件的形式化建模方法[J]. 软件学报，2015，26(2)：321-331.
GU B，DONG Y W，WANG Z. Formal modeling approach for aerospace embedded software[J]. Journal of Software，2015，26(2)：321-331.

[9] 谭彦亮，杨桦，乔磊. 基于Event-B的SpaceOS2操作系统任务管理需求设计形式化建模与验证[J]. 空间控制技术与应用，2014，4(40)：57-62.
TAN Y L，YANG H，QIAO L. Formal modeling and verification method of task management requirement for SpaceOS2 based on Event-B[J]. Aerospace Control and Application，2014，4(40)：57-62.

[10] 周育逵，杨桦，乔磊. 基于Event-B的中断管理需求和设计形式化建模与验证方法[J]. 空间控制技术与应用，2017，3(43)：71-78.
ZHOU Y K，YANG H，QIAO L. Formal modeling and verification method of interrupt management requirement and design based on event-B[J]. Aerospace Control and Application，2017，3(43)：71-78.

[11] 李雷，陈朝晖，董晓刚，等. 基于聚类分析的软件多故障定位技术[J]. 空间控制技术与应用，2019，45(5)：55-62.
LI L，CHEN Z H，DONG X G，et al. Software multi-fault location technology base on cluster analysis[J]. Aerospace Control and Application，2019，45(5)：55-62.

[12] 李雷，陈朝晖，李轶，等. 软件故障定位技术研究综述[J]. 计算机测量与控制，2019，27(5)：1-8.
LI L，CHEN Z H，LI Y，et al. Overview of software fault localization technology[J]. Computer Measurement & Control，2019，27(5)：1-8.

[13] 李雷. 基于聚类分析的软件多故障定位技术研究[D]. 北京：中国空间技术研究院，2019.
LI L. Software multi-fault localization based on clustering analysis[D]. Beijing：China Academy of Space Technology，2019.

[14] 王同磊. 基于程序切片的软件脆弱性自动定位技术研究[D]. 北京：中国空间技术研究院，2017.
WANG T L. Software vulnerability localization based on program slicing[D]. Beijing：China Academy of Space Technology，2017.

[15] YANG Z，XIAO M Q，HU B，et al. Development of foreign automatic test language for aviation[J]. Computer Measurement & Control，2013，21(4)：833-842.

[16] 张玉花，梅海，赵晨，等. 嫦娥五号轨道器的创新与实践[J]. 上海航天(中英文)，2020，37(6)：1-10.
ZHANG Y H，MEI H，ZHAO C，et al. Innovation and practice of Chang'e-5 orbiter[J]. Aerospace Shanghai，2020，37(6)：1-10.

[17] 张丽芸，蒲戈光，王政，等. 一种面向控制软件需求分析的方法[J]. 计算机应用研究，2013，30(2)：465-468.
ZHANG L Y，PU G G，WANG Z，et al. Analysis method of control

system requirement[J]. Application Research of Computers，2013，30(2)：465-468.

[18] WEISER M. Program slicing[C]//Proceedings of ICSE'81：the 5th International Conference on Software Engineering. San Diego，CA,，USA：ICSE，1981.

[19] IEEE Standard Coordinating Committee. ATLAS 2000，ATLAS 2000 Introductory Guide Rev B[S]. USA：IEEE，1997.

[20] WU L J，JIAN Y，ZHANG K，et al. Technology about GUI test script based on python[J]. Computer Measurement & Control，2015，23(10)：3330-3337.

[21] 古天龙. 软件开发的形式化方法[M]. 北京：高等教育出版社，2005.

作者简介：

**李轶**(1984– )，男，博士，高级工程师，主要研究方向：航天嵌入式软件高可信研制技术。

通讯地址：北京市海淀区友谊路104号5142信箱150分箱

电话：(010)68111399-802

E-mail：116488832@qq.com

# High Confidence Development Technology of Application Software for GNC Subsystem of Chang'E-5

LI Yi，LI Li，GUO Mingshu，WANG Tonglei，ZHANG Guofeng，LI Xiaofeng

（Beijing Institute of Control Engineering，Beijing 100094，China）

**Abstract**：Facing to the high-safety and high-reliability mission requirement，the high-complexity and high-autonomy function requirement，and the high-real-time and strong-sequential performance requirement of mission Chang'E-5，this paper researches the high confidence develop technology of the application software for GNC Subsystem of Chang'E-5. During the requirement analysis，the sequential safety attribute is added to the formal modeling and verification language，which avoids the requirement duality. In phase of design and coding，program slicing is extracted based on the mission profile，which is used in the source code fault localization，comparing to the manual work，the normalization and correctness of source code is improved. The test case auto generate method base on state chart and sequence diagram is researched，and a Full-Automatic Spacecraft software Testing suite is established for the massive test case，the test coverage and efficiency is obviously improved.

**Keywords**：Chang'E-5；GNC subsystem；application software；high confidence

**Highlights**：

- Formal modeling face to the key temporal is proposed.
- Source code fault localization base on program slicing is implemented.
- Fast automatic testing for the massive test case is compared.

[责任编辑：宋宏，英文审校：刘勇]