

基于 SysML 的健康监测系统软件设计

周潇雅, 李鹏程, 肖进, 顾黎, 赵博

(北京宇航系统工程研究所, 北京 100076)

摘要: 航天复杂任务场景导致航天产品规模与复杂度显著增长, 航天产品“智能化”“集成化”的发展趋势也使得软件在航天产品中的地位与功能要求不断攀升, 传统的基于文档的软件功能设计往往在软件需求文档尚未完全厘清下开展, 而软件测试则在软件设计完毕及编码实现后才开展, 需求文档的模糊性导致可能出现设计、测试反复等问题, 进而影响软件产品研制交付效率和质量; 为了避免基于文档的软件设计带来的问题, 采用 MBSE 方法, 针对基于 SysML 的健康监测系统软件设计进行了研究, 采用 SysML 软件模型在软件代码实现之前开展软件功能逻辑验证和软件功能测试; 仿真及测试结果表明在软件论证阶段使用该方法, 对软件功能设计方案正确性进行验证是可行的和有效的。

关键词: SysML; MBSE; 软件设计; 软件测试; 基于模型

Research on Health Monitoring Software Design Based on SysML

ZHOU Xiaoya, LI Pengcheng, XIAO Jin, GU Li, ZHAO Bo

(Beijing Aerospace System Engineering Institute, Beijing 100076, China)

Abstract: Complex aerospace missions bring a significant increase in the scale and complexity of aerospace products. And the development trend of intelligence and integration has led to a continuous increase in the status and functional requirements of software in aerospace products. Traditional software designs based on documents are often carried out before the requirement document of software is fully clarified, while software test is carried out after software design and coding are implemented. The ambiguity of requirement documents probably leads to design and test iterations, thus affecting the development and delivery efficiency and quality of software products. In order to avoid those problems, and based on system modeling language (SysML), model-based system engineering (MBSE) is used to study health monitoring software. The SysML is used to achieve the logic verification and test of software functions before software coding is implemented. Simulation and testing results show that it is feasible and available for this method to verify the correctness of software functional design in the demonstration stage.

Keywords: SysML; MBSE; software design; software test; model-based

0 引言

航天产品软件作为航天器电气系统的核心组成部分, 具有以下几个特点: 1) 航天产品软件往往需要实现多专业、多领域的功能, 导致航天产品软件功能复杂、规模较大; 2) 大多为嵌入式软件, 与运行环境密切相关, 软件的设计与测试依托于不同硬件平台, 增加了软件测试成本; 3) 软件决定了航天产品的成败, 应具有高安全性、高可靠性, 需要对航天产品软件开展全面有效的测试工作。这都给航天产品软件研制带来了挑战。

此外, 随着我国航天装备、运载火箭的任务场景愈加复杂, 航天产品的规模与复杂度显著增长。一方面, 航天产品系统功能、逻辑、信息流、软件协议、接口越来越复杂, 用户对岗位的划分越来越细, 以往依靠手工

完成或者由单一或零散的几个软件独立完成一个系统的设计方法越来越不能满足复杂系统集成设计、系统开发实现、系统集成测试的需求^[1-3]; 另一方面, 航天产品“智能化”“集成化”发展趋势使得软件在航天产品中的地位与功能要求不断攀升, 传统的基于文档的工程化方法已经显现出不同程度的不适应, 时序、状态、接口、协议、数据等设计信息的传递效率低且准确性差, 成为航天产品软件研制效率提升的主要制约瓶颈, 与现代主流的喷泉式、敏捷式软件产品集成开发存在明显差距, 无法适应未来多变环境的发展需要, 难以高质量生成装备能力, 未来或将严重滞后于发展新需求、新趋势。在这种严峻形势下如何对航天产品软件研制的进度、质量与成本进行有效控制, 成为亟待解决的问题。因此, 航

收稿日期: 2024-09-20; 修回日期: 2024-10-29。

作者简介: 周潇雅(1991-), 女, 博士, 高级工程师。

引用格式: 周潇雅, 李鹏程, 肖进, 等. 基于 SysML 的健康监测系统软件设计[J]. 计算机测量与控制, 2025, 33(10): 30-36.

天产品需求分析、软件设计、软件测试, 已经不能依赖传统方式进行验证, 亟需在软件功能设计与配置项测试方面开展革新。

近年来, 在航空航天领域国内外科研机构已广泛应用系统建模语言 (SysML, system modeling language) 开展系统设计, 有效验证了系统设计的功能逻辑、信息流、接口正确性等^[4-12]。目前, 将 SysML 模型深入应用到软件层面, 成为基于模型的系统工程 (MBSE, model-based systems engineering) 技术应用的新热点。件林博等人^[13]提出了一种基于 SysML 语言的嵌入式软件系统结构与行为需求建模方法, 通过 SysML 语言对嵌入式软件系统的结构和逻辑行为开展了层次化建模, 并将 SysML 模型与 modelica 仿真模型结合, 在系统需求论证阶段, 利用 SysML 语言对系统关键功能指标进行仿真, 结果表明该方法对嵌入式软件系统需求论证具有可行性。肖思慧等人^[14]提出了一种基于 SysML 状态机图子集的机载软件分层精化建模与验证方法。该方法首先使用 SysML 状态机图对机载软件的动态行为进行建模, 并根据提出的精化规则对初始模型进行手动逐层精化, 得到精化设计模型; 然后, 针对软件模型动态变化的特性, 将 SysML 状态机模型自动转换为时间自动机网络, 并从软件需求中手动提取形式化 TCTL 性质进行模型检验。其次, 将 SysML 模型自动转换至 Simulink, 利用 Simulink Coder 生成源代码, 实现编码自动化。最后, 以自动飞行控制软件为例进行了开发和验证, 实验结果表明了该方法的有效性。尹伟等人^[15]基于 Safety_SysML 状态机的语法, 设计了一种 Safety_SysML 一致性验证器, 包括静态数据检测和动态数据检测; 通过设计测试用例对核心算法及系统进行单元测试与集成测试, 结合错误推断及边界, 设计并执行功能测试, 用于发现验证器存在的缺陷。结果表明: Safety_SysML 一致性验证器能够有效识别航电系统软件中存在的二义性问题, 对于提高航电软件的可靠性具有重要意义。但是, 上述这些研究主要聚焦于软件的逻辑验证和单元测试, 尚未将软件通讯协议融入模型, 未考虑基于 SysML 模型开展软件配置项测试与分析, 进一步提升软件设计的可靠性和安全性。

本文面向航天产品, 提出了一种基于 SysML 的软件功能设计与配置项测试方法, 采用该方法能在软件设计早期及详细编码实现前, 基于 SysML 模型开展逻辑验证和软件配置项用例测试验证软件设计方案, 判断软件功能设计的合理性和正确性, 从而确保软件方案闭合, 支撑航天产品软件产品成功研制。

1 SysML 语言

统一建模语言 (UML, unified modeling language)

是软件设计人员开展软件设计的常用建模语言^[16-17]。系统建模语言 SysML 是一种标准的通用的图形化系统建模语言, 是在 UML 的基础上扩展而来。SysML 定义了块定义图、内部块图、包图、参数图、需求图、用例图、活动图、顺序图及状态机图共 9 种基本的图形, 从不同视角描述系统, 能对系统工程的各种问题建模, 适用系统工程的不同阶段, 特别是系统工程的详细分析阶段和设计阶段, 使用 SysML 能有效说明需求、系统结构、功能行为和分配^[18-20]。

2 基于 SysML 的健康监测系统软件设计方法

传统的软件设计是基于文档的、串行的研制模式, 如图 1 所示。软件设计主要流程为: 首先, 航天产品系统工程师将航天产品分系统或设备之间的指令信息数据交互要求、分系统或设备内嵌软件功能要求进行总结提炼, 形成软件设计要求; 接着, 软件开发工程师根据软件设计要求, 形成软件设计方案; 最后, 软件开发工程师基于方案进行软件编码实现, 开展软件配置项测试。采用传统的流程, 软件功能遗漏、软硬件接口不匹配等软件设计方案错误往往在软件实现后才被发现, 在系统集成或测试阶段出现设计反复问题, 严重影响了航天产品软件研制效率。

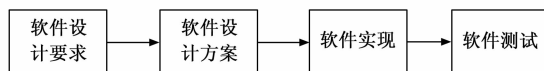


图 1 传统的软件设计流程

UDP (User Datagram Protocol) 协议是一种面向数据包的简单传输通讯协议, 具有启动传输时延短、开销小、传输效率高的特点, 被广泛运用于航天产品嵌入式软件中。航天产品健康监测系统软件嵌入健康监测系统内, 用于实时监测 3 个航天产品系统软件 (定义为系统 1、系统 2、系统 3) 运行状态, 软件之间通过 UDP 通讯协议进行数据传输。

航天产品健康监测系统软件的总体结构如图 2 所示, 其主要功能有 5 个: 1) 数据采集: 健康监测系统软件能够通过数据采集模块接收其他 3 个产品通过 UDP 协议上报的状态数据包, 并将 UDP 协议中各系统状态参数值按照协议格式进行读取、推送给数据存储模块和超差判读模块; 2) 超差判读: 超差判读模块接收数据采集模块推送的采集原码, 按照预设或动态设定的判读规则, 对采集原码中的监测参数进行自动判读, 每条判读规则中存储了各状态参数值判定正常时对应的有效范围, 包括上限和下限, 当超差判读模块采集的状态参数值超出有效范围时, 判断为超差, 反之判断为不超差, 形成各监测参数的判读结果, 推送给数据存储模块和数据图形化模块、超差显示模块; 3) 超差显示: 超

差显示模块将超差判读模块推送的判读结果以标签一值的形式通过人机交互接口推至界面，以数值显示的颜色标识数据是否超差，红色代表超差，黑色代表正常。通过不同的超差判读规则对测试过程中出现的异常故障进行初步判定，从而实时监控产品的运行情况和当前状态；4）数据存储：将数据采集模块、超差判读模块推送的采集原码、判读结果写入数据库固化，形成历史数据数据库，用于后续趋势比对和回溯查询；5）数据图形化：数据图形化模块将超差判读模块推送的判读结果通过人机交互接口以图表的形式推至界面，针对各产品上报状态参数值最近一段时间的实时数据自动绘制曲线图，从而辅助航天产品运维师进行监控与决策。

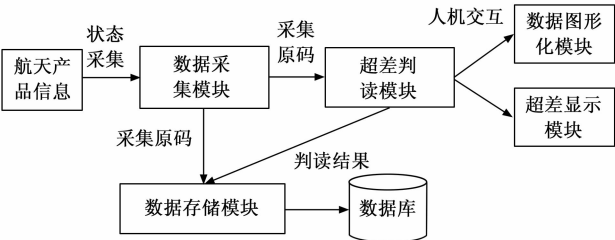


图 2 健康监测系统软件总体结构框图

基于模型的健康监测系统软件设计方法，流程如图 3 所示，包括基于模型的功能验证和基于模型的测试验证。该方法流程主要为：首先，基于健康监测系统设计软件要求，形成 SysML 软件模型（设计验证）描述软件功能，通过仿真开展一轮软件功能逻辑验证；接着，在 SysML 软件模型（设计验证）基础上，结合参数表、通讯协议要求进一步形成 SysML 软件模型（测试验证）；最后，基于 SysML 软件模型（测试验证）开展软件测试，健康监测系统软件模型与 3 个航天产品系统的软件等效器通过基于 UDP 通讯协议格式的数据包传输数据、指令，逐项验证健康监测系统软件功能是否满足设计要求，进一步验证软件方案闭合，两轮验证通过后实现软件，支撑软件产品设计一次成功。

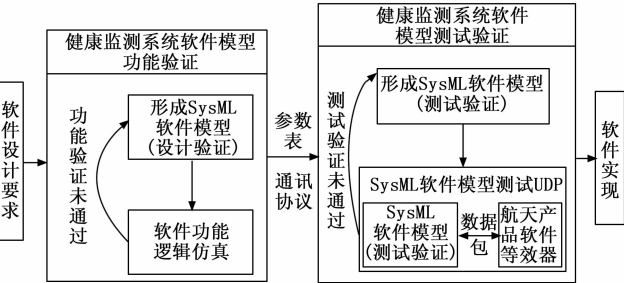


图 3 基于 SysML 的健康监测系统软件设计方法

3 基于 SysML 的健康监测系统软件设计实验分析

3.1 实验环境及参数配置

基于 SysML 的健康监测系统软件设计实验环境为

MagicDraw 功能建模环境和 Java 编译环境；实验参数配置包括通讯协议和参数表。

航天产品健康监测系统软件与 3 个航天产品系统间传输的 UDP 通讯协议主要包括两部分：帧头和数据区。其中，帧头即表号（2 字节）；数据区由参数名（8 字节）、参数值（4 字节）、参数上限（4 字节）和参数下限（4 字节）组成，如表 1 所示。

表 1 通讯协议格式

帧头	数据区			
表号 (2 字节)	参数名 (8 字节)	参数值 (4 字节)	上限 (4 字节)	下限 (4 字节)

航天产品健康监测过程参数表如表 2 所示。航天产品系统中系统 3 共有 5 个需采集监测的状态参数，表号为 100，参数 kz1 为 uchar 类型，其正常范围为 100~200，参数 kz2 为 ushort 类型，其正常范围为 1 000~2 000，参数 kz3 为 uint 类型，其正常范围为 10 000~20 000，参数 kz4 为 float 类型，其正常范围为 14.5~15.5，kz5 为 float 类型，其正常范围为 4.5~5.5；航天产品系统中系统 2 共有 5 个需采集监测的状态参数，表号为 200，参数 cl1 为 uchar 类型，其正常范围为 100~200，参数 cl2 为 ushort 类型，其正常范围为 1 000~2 000，参数 cl3 为 uint 类型，其正常范围为 10 000~20 000，参数 cl4 为 float 类型，其正常范围为 14.5~15.5，cl5 为 float 类型，其正常范围为 4.5~5.5；航天产品系统中系统 1 共有 5 个需采集监测的状态参数，表号为 300，参数 ck1 为 uchar 类型，其正常范围为 100~200，参数 ck2 为 ushort 类型，其正常范围为 1 000~2 000，参数 ck3 为 uint 类型，其正常范围为 10 000~20 000，参数 ck4 为 float 类型，其正常范围为 14.5~15.5，ck5 为 float 类型，其正常范围为 4.5~5.5。

表 2 参数表

所属系统	表号	参数名	参数类型	上限	下限
系统 3	100	kz1	uchar	200	100
系统 3	100	kz2	ushort	2 000	1000
系统 3	100	kz3	uint	20 000	10 000
系统 3	100	kz4	float	15.5	14.5
系统 3	100	kz5	float	5.5	4.5
系统 1	200	cl1	uchar	200	100
系统 1	200	cl2	ushort	2 000	1 000
系统 1	200	cl3	uint	20 000	10 000
系统 1	200	cl4	float	15.5	14.5
系统 1	200	cl5	float	5.5	4.5
系统 2	300	ck1	uchar	200	100
系统 2	300	ck2	ushort	2 000	1 000
系统 2	300	ck3	uint	20 000	10 000
系统 2	300	ck4	float	15.5	14.5
系统 2	300	ck5	float	5.5	4.5

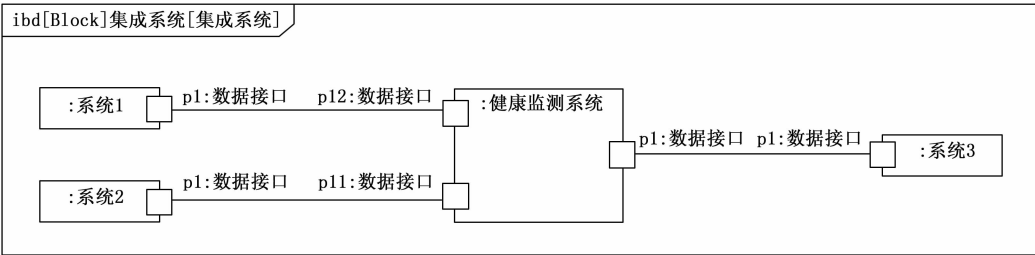


图 4 各系统间状态参数传递接口

3.2 基于 SysML 的健康监测系统软件设计实验过程

3.2.1 软件功能验证实验过程

首先, 通过内部模块定义图, 描述各系统间需进行监测的状态参数传递的接口。如图 4 所示, 健康监测系统软件需要对系统 3、系统 2 和系统 1 状态参数进行监测, 数据包分别通过数据接口 p1、p11 和 p12 进行传输。

接着, 在 3 个航天产品系统软件建立 SysML 软件模型 (设计验证) 状态机图, 描述控航天产品系统由“初始化”“发送数据”两个状态组成, 每间隔 1 s, 通过信号的形式发送需健康监测的状态数据。如图 5 所示, 描述了系统 1 每间隔 1 s 向健康监测系统软件发送参数 cl3。

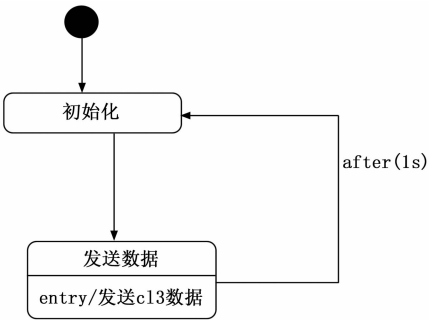


图 5 系统 1 发送状态参数

同时, 建立健康监测系统软件 SysML 软件模型 (设计验证) 状态机图, 描述健康监测系统软件不断地接收 3 个航天产品系统软件发送的状态参数, 如图 6 所示。

最后, 基于健康监测系统软件及 3 个航天产品系统软件状态机图, 形成最终的 SysML 软件模型 (设计验证), 开展功能逻辑仿真。

3.2.2 软件测试验证实验过程

在 SysML 软件模型 (设计验证) 的基础上, 根据健康监测系统软件功能要求, 形成 SysML 软件模型 (测试验证)。

首先, 针对健康监测系统软件数据采集、超差判读功能要求, 在 SysML 软件模型 (设计验证) 的基础上

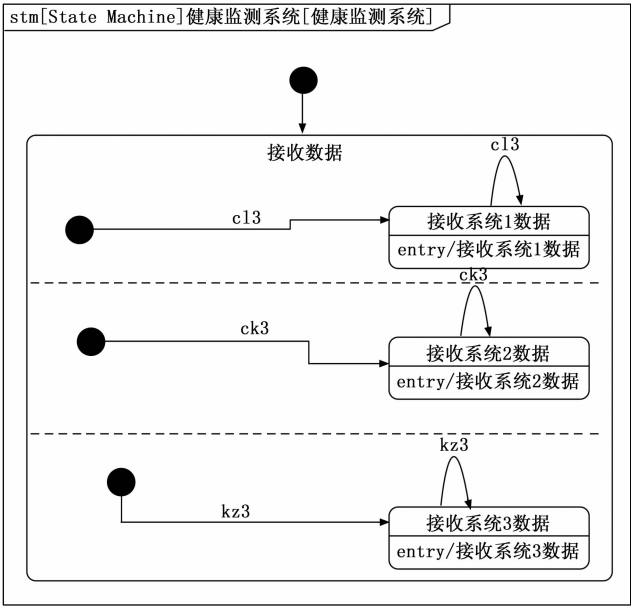


图 6 接收各系统状态参数

构建变量、内嵌代码, 包括以下步骤: 第一步, 按照参数表在 SysML 模型中定义需采集的状态参数变量, 即 kz1~kz5、cl1~cl5、ck1~ck5 共 15 个参数名及其对应参数类型; 第二步, 对应需采集状态参数, 在 SysML 模型中定义 kz1 _ boolean~kz5 _ boolean、cl1 _ boolean~cl5 _ boolean、ck1 _ boolean~ck5 _ boolean, 共 15 个布尔类型参数变量, 后续用于判断状态参数是否超差; 第三步, 构建健康监测系统软件 SysML 软件模型 (设计验证) 状态机图, 在状态中内嵌不透明表达式, 基于 Java 代码, 实现从网络端口采集符合 UDP 通讯协议格式的数据包, 识别出数据包中表号、参数名, 将其对应的状态参数变量数据赋值到 SysML 模型对应的状态参数变量中, 并在 SysML 模型中进一步判断状态参数值是否在正常范围内, 若在正常范围内, 将状态参数对应的布尔类型参数赋值为“true”, 若不在正常范围则判断该参数超差, 将状态参数对应的布尔类型参数赋值为“false”。

其次, 针对健康监测系统软件超差显示功能, 形成 UI 界面, 动态显示接收到的各状态参数数据以及该参数是否异常超差。UI 界面分为 3 个展示区域, 自上而下分

别为系统 3、系统 2、系统 1，各系统展示区域显示需监测的状态参数名及其具体的参数数据，若其对应的布尔类型参数值为“true”，则显示“? XX 参数正常”，若其对应的布尔类型参数值为“false”，则标红显示。

接着，针对健康监测系统软件数据存储功能，在 SysML 软件模型状态中内嵌不透明表达式。基于 Java 代码，实现打开 mysql 数据库，该数据库包括 7 列，即采集时间、采集系统、表号、参数名、参数类型、参数上限、参数下限和参数数据，当网络端口接收到基于 UDP 通讯协议格式的数据包时，将采集时间以及解析基于 UDP 通讯协议格式的数据包得到的信息对应存放在 mysql 数据库各列中，网络端口每采集到一次数据包，则存放一次数据至 mysql 数据库，最终形成 3 个航天产品系统 15 个状态参数随时间变化的历史数据。

并且，针对健康监测系统软件数据图形化功能，通过仿真配置进行实现。在 SysML 软件模型构建模拟配置图（Simulation Configuration Diagram），实现在 SysML 模型仿真时可选择 15 个状态参数之一，在仿真过程中同步生成该状态参数的曲线图，横轴为采集时间，纵轴为状态参数数值。

最后，运行 SysML 软件模型（测试验证）和软件等效器，开展基于 SysML 的健康监测系统软件测试，逐项测试健康监测系统软件各项主要功能，如图 7 所示。包括以下 3 个步骤：第一步，运行等效器，通过代码实现模拟 3 个航天产品系统软件，以 1 s 为时间间隔，给网络端口实时发送满足 UDP 通讯协议格式要求的数据包；第二步，运行 SysML 软件模型（测试验证），通过模型仿真开展数据采集、超差判读、超差显示、数据存储和数据图形化展示；第三步，根据 UI 界面、数据库及曲线图等仿真结果，测试健康监测系统软件各项主要功能是否正确。

3.3 基于 SysML 的健康监测系统软件设计实验结果及分析

3.3.1 软件功能验证实验结果及分析

基于 SysML 的软件功能验证实验结果如图 8 所示，从逻辑上验证了健康监测软件设计方案中软件间接口、主要参数传递、功能的正确性。

3.3.2 基于 SysML 的软件测试实验结果及分析

健康监测系统软件 SysML 软件模型（测试验证）数据采集、超差判读和超差显示功能测试结果如图 9 所示。由图可知，由模拟 3 个航天产品系统的等效器实时发

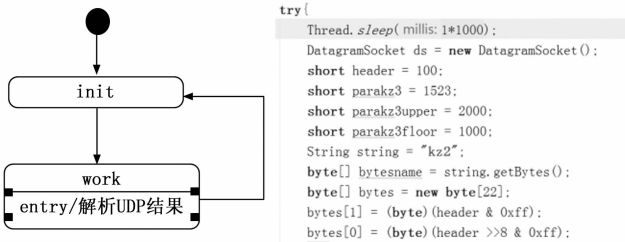


图 7 基于 SysML 的健康监测系统软件测试

送至网络端口的基于 UDP 通讯协议格式的数据包，能被 SysML 软件模型正确地识别、采集和显示，并进一步开展超差判读，超差判读结果符合 UDP 通讯协议格式判读规则，超差数据能正确地标红显示。因此，按照当前的健康监测系统软件设计方案能实现数据采集、超差判读和超差显示功能。

健康监测系统软件 SysML 软件模型（测试验证）数据存储功能测试结果如表 3 所示。通过 SysML 软件模型（测试验证）运行仿真，网络端口采集到的基于 UDP 通讯协议格式的数据包能正确地解析并将相应的采集时间、状态参数数据信息依次存储至 mysql 数据库，因此，按照当前的健康监测系统软件设计方案能实现数据存储功能。

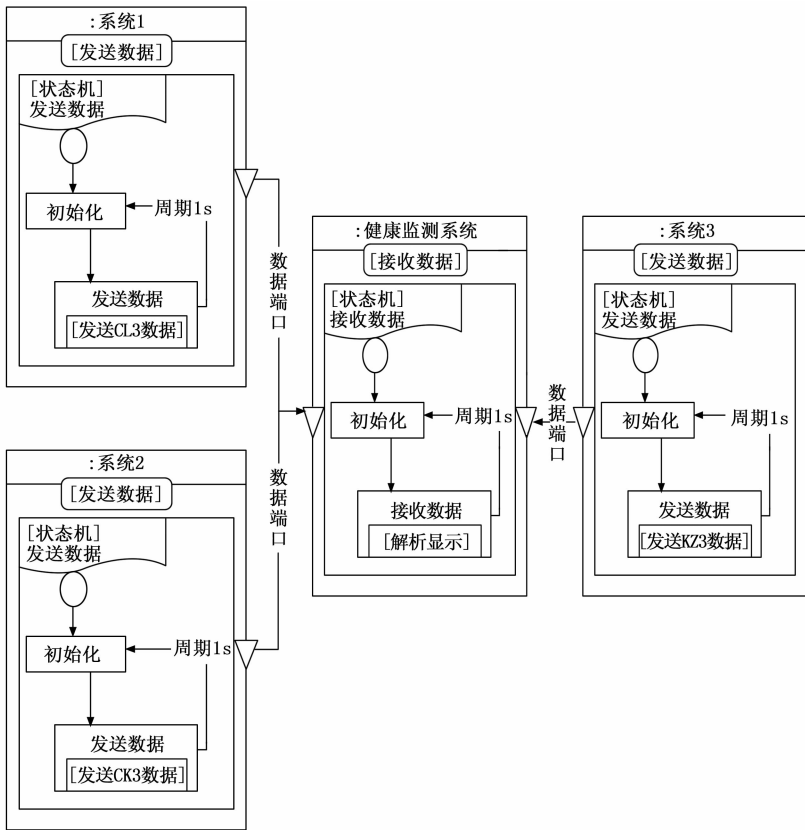


图 8 基于 SysML 的软件功能逻辑验证实验结果



图 9 数据采集及超差判读功能测试结果

表 3 存储数据功能测试结果

ts	system	tablename	paraname	type	upper	floor	para
aa	kzxt	100	kz3	int	20 000	10 000	16 870
aa	kzxt	100	kz3	int	20 000	10 000	17 370
aa	kzxt	100	kz3	int	20 000	10 000	17 870
aa	kzxt	100	kz3	int	20 000	10 000	18 370
aa	kzxt	100	kz3	int	20 000	10 000	18 870
aa	kzxt	100	kz3	int	20 000	10 000	19 370
aa	kzxt	100	kz3	int	20 000	10 000	19 870
aa	kzxt	100	kz3	int	20 000	10 000	20 370
aa	kzxt	100	kz3	int	20 000	10 000	20 870
aa	kzxt	100	kz3	int	20 000	10 000	21 370

健康监测系统软件 SysML 软件模型（测试验证）数据图形化展示功能测试结果如图 10 所示。由图可知，通过 SysML 软件模型（测试验证）运行仿真，能筛选出需监测的状态参数，并正确地自动绘制出该参数随时间变化的曲线。因此，按照当前的健康监测系统软件设计方案能实现数据采集、超差判读和超差显示功能。

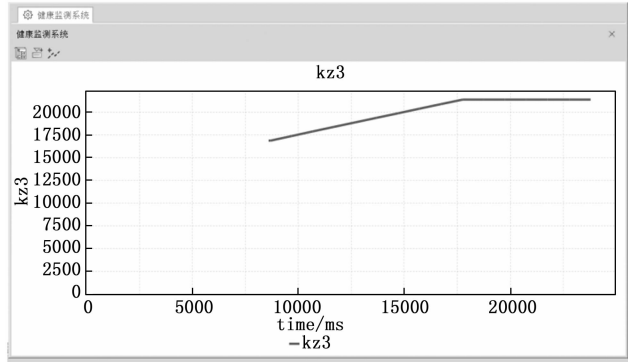


图 10 历史数据查询功能测试结果

针对健康监测系统 SysML 软件模型开展配置项测试，具体测试说明见表 4，测试结果如表 5 所示。

表 4 SysML 软件模型测试说明

序号	测试需求	测试用例	测试结果
1	数据采集	给定不符合 UDP 协议格式的产品信息	通过，丢弃不处理
2		连续上报合法的产品信息	通过，收到后及时上报
3	超差判读	给出超差的判据，确认超差判断	通过，判读结果为超差
4		给出超差判据边界值	通过，判读结果为不超差
5		连续上报合法的产品信息，能否正常处理	通过，判读结果及时上报
6	数据存储	给出采集原码，确认存储情况	通过，正常写入库
7		给出判读结果，确认存储情况	通过，正常写入库
8	超差显示	判读结果为超差的是否显示符合预期	通过，显示为红色
9		判读结果合法的是否显示符合预期	通过，显示为黑色
10	数据图形化	参数可否灵活绘制曲线	通过，显示为绘制对应参数的曲线

表 5 SysML 软件模型测试用例结果

测试项名称	用例总数	通过数	未通过数	未执行数
数据采集	11	9	2	2
超差判读	37	37	0	0
数据存储	12	12	0	0
超差显示	46	46	0	0
数据图形化	20	20	0	0

由表 4、表 5 可知，针对健康监测系统 SysML 软件模型开展配置项测试，验证了健康监测系统软件设计方案大部分满足主要软件功能要求，能实现基于 UDP 通讯协议的数据包接收、监测和分析，满足辅助航天产品设计师进行监控与决策的业务目标。在数据采集功能测试中，出现了未执行用例情况，经排查为当前健康监测系统软件方案中未考虑同一时刻收到 2~3 个航天产品系统参数，软件设计方案需要进一步优化，证明了采用基于 SysML 的健康监测系统软件设计方法验证软件方案是可行的、有效的；同时，基于模型从软件需求描述到软件模型建立，再由已验证的模型辅助软件实现的全过程，平均耗时为 9~10 天，相较于传统型号软件的串行的需求理解、设计、实现、调试的过程，一般为几周甚至按月计算，明显提升了设计效率。

4 基于 SysML 的软件设计优势

通过仿真与测试结果，可以得出与传统的软件设计

方法相比, 基于 SysML 的软件设计方法具有显著优势。首先, 采用 SysML 语言能以可视化、图形化的模型描述软件功能, 实现软件方案从系统设计专业到软件设计专业一致性传递, 避免软件需求与实现在航天产品软件开发工程师和系统工程师之间产生“二义性”, 减少由于协调沟通不彻底引发的软件设计问题; 其次, 通过 SysML 模型能对软件方案进行清晰明确的结构化描述, 避免基于文档带来的航天产品软件方案模糊性问题; 最后, 基于 SysML 的软件设计方法能将后端测试验证工作提前, 在航天产品软件设计初期阶段, 通过模型仿真验证软件功能逻辑, 并且通过开展基于模型的软件测试逐项验证软件方案, 能尽早地发现软件功能设计是否出现矛盾、遗漏、冗余等问题, 降低在软件产品实现后出现设计反复导致航天产品软件研制周期增加的问题, 从而减少软件修正成本。

5 结束语

本文提出了一种基于 SysML 的健康监测系统软件设计方法, 表明了该方法能在软件代码详细实现前通过仿真实验验证前移, 基于 SysML 软件模型提前高效完成一轮软件功能逻辑验证和一轮软件模型测试再实现软件, 极大地避免了软件功能遗漏、软硬件接口不匹配等错误造成软件设计反复等问题, 从而优化了航天产品软件研发流程, 缩短软件研发交付周期, 确保软件功能设计的合理性和正确性。

后续还需要进一步研究: 1) 将简化的通讯协议换成航天产品真实多样的复杂通讯协议进行建模, 并考虑多线程等复杂情况, 开展软件功能验证与软件测试; 2) 将多个航天产品软件的 SysML 软件模型进行集成并开展系统联试仿真, 分析多个软件之间逻辑和功能是否存在冲突和缺陷并改进; 3) 研究基于 SysML 软件模型自动生成代码框架或代码技术, 进一步提升系统软件设计可靠性和安全性, 支撑系统软件成功研制, 实现软件系统高效、高质量研发。

参考文献:

- [1] 侯 雄, 李东方. MBSE 技术及其在航天控制中的应用 [J]. 航天控制, 2023, 41 (5): 3-13.
- [2] 何 巍, 胡久辉, 赵 婷, 等. 基于模型的运载火箭总体设计方法初探 [J]. 导弹与航天运载技术, 2021 (1): 12-17.
- [3] 陈建伟, 肖 进, 周潇雅, 等. 基于模型的系统工程在航天产品研制的研究与实践 [J]. 航天标准化, 2020: 193-196.
- [4] 李 强, 胡元威, 董余红, 等. 基于模块的运载火箭电气系统匹配验证仿真 [J]. 计算机测量与控制, 2019, 27 (2): 256-259.
- [5] 周潇雅, 杨 亮, 张 茜, 等. 基于 ARCADIA MBSE 在运载火箭能源子系统的应用 [J]. 计算机测量与控制, 2020, 28 (11): 211-215.
- [6] 冯忠伟, 王 征, 丁 琳, 等. 基于 MBSE 技术的航天器电气系统设计技术 [J]. 计算机测量与控制, 2023, 31 (11): 255-259.
- [7] 关 锋, 葛 平, 邵艳利, 等. 基于 MBSE 的月球科研站任务分析 [J]. 航空工程进展, 2023, 14 (3): 84-99.
- [8] 浦 乐, 王西超, 杨 艺. 基于 MBSE 与 SysML 的空空导弹系统架构建模研究 [J]. 航空科学技术, 2018, 39 (2): 94-102.
- [9] 杨 锋, 王 南, 卢官凯, 等. MBSE 在活动发射平台支承销设计中的应用实践 [J]. 宇航总体技术, 2023, 7 (1): 37-45.
- [10] 张 兵, 陈建伟, 肖 进, 等. 基于模型的系统工程在航天产品研发中的研究与实践 [J]. 宇航总体技术, 2021, 5 (1): 1-7.
- [11] 李 娇, 隆金波, 彭文胜, 等. MBSE 模式下可靠性安全性测试性一体化建模与评估技术方法 [J]. 计算机测量与控制, 2021, 29 (7): 247-253.
- [12] 郭 坚, 张红军, 李 林, 等. 一种航天器软件系统设计方法 [J]. 质量与可靠性, 2021 (5): 56-60.
- [13] 仵林博, 陈小红, 彭艳红, 等. 基于 SysML 的嵌入式软件系统建模与验证方法研究 [J]. 计算机工程, 2019, 45 (1): 1-8.
- [14] 肖思慧, 刘 琦, 黄滢鸿, 等. 基于 SysML 的机载软件分层精化建模与验证方法 [J]. 软件学报, 2022, 33 (8): 2851-2874.
- [15] 尹 伟, 王 辉, 孙海英, 等. 面向航电系统软件的安全性 SysML 一致性验证器设计与实现 [J]. 航空工程进展, 2023, 14 (4): 177-188.
- [16] OZKAYA M, ERATA F. A survey on the practical use of UML for different software architecture viewpoints [J]. Information and Software Technology, 2020, 121 (4): 106275.
- [17] ORDINEZ L, EGGLEY G, MICHELETTO M, et al. Using UML for learning how to design and model cyber-physical systems [J]. IEEE Revista Iberomericana de Tecnologias del Aprendizaje, 2020, 15 (1): 50-60.
- [18] 王文浩, 毕文豪, 张 安, 等. 基于 MBSE 的民机系统功能建模方法 [J]. 系统工程与电子技术, 2021, 43 (10): 2884-2892.
- [19] 鲍 阳, 杨志斌, 杨永强, 等. 基于限定中文自然语言需求的 SysML 模型自动生成方法 [J]. 计算机研究与发展, 2021, 58 (4): 706-730.
- [20] 任浩亮, 张建超, 程会川. 基于 SysML 的武器装备体系能力需求建模分析方法 [J]. 系统工程与电子技术, 2023, 45 (9): 2843-2851.