Q/NSSC

中国科学院国家空间科学中心标准

Q/NSSC 013—2025

空间科学任务论证 系统级仿真模型描述规范

Space science mission study—Specification for system-level simulation model description

2025-05-15 发布　　2025-05-15 实施

中国科学院国家空间科学中心 批准

目 次

前 言

本文件按照《空间中心标准管理办法》和GB/T1.1-2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定给出的规则起草。

本文件由中国科学院国家空间科学中心复杂航天系统电子信息技术重点实验室提出。

本文件由中国科学院国家空间科学中心质量管理处归口。

本文件起草部门：中国科学院国家空间科学中心复杂航天系统电子信息技术重点实验室

本文件主要起草人：张玉珠、彭晓东、谢文明。

引 言

众多的空间科学任务使得我们对该类任务的论证、设计、研制、测试与运行周期有了更多、更新和更高的要求。为了提高空间科学卫星在工程设计、验证和运行等各个阶段完成的质量和效率，世界各航天机构都采用建模和仿真技术作为空间科学任务的辅助手段。空间科学探测任务的仿真属于大型复杂系统仿真，涉及复杂的学科领域和专业子系统、众多的协作单位和复杂的集成关系。因此，编制本规范，为能够快速将多种仿真模型组织起来，构建仿真系统，开展任务和系统级仿真，提供依据。

空间科学任务论证 系统级仿真模型描述规范

1 范围

本文件规定了构建空间科学任务系统级仿真系统所需的仿真模型的信息，包括仿真模型信息中所需元素、仿真模型配置文件的格式与使用的语言。

本文件适用于空间科学任务系统级仿真系统配置文件与仿真模型接口文件的编写，其他类似系统可参照使用。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 30114.1-2013 空间科学及其应用术语 第 1 部分：基础通用

GJB 6935-2009 军用仿真术语

GJB 7099.2-2010 作战模拟模型开发通用要求 第 2 部分：数学逻辑模型

GJB 7099.3-2012 作战模拟模型开发通用要求 第 3 部分：仿真程序模型

GJB 7860-2012 仿真管理模型通用要求；

3 术语和定义

下列术语和定义适用于本文件。

3.1

空间科学 Space Science

以航天、航空飞行器以及地面工作平台，研究发生在地球、日地空间、太阳系乃至整个宇宙的物理、化学及生命等自然现象及其规律的科学。

[来源：GB/T 30114.1-2013]。

3.2

数学逻辑模型

用数学表达方法、逻辑表达方法和数据来描述研究对象的本质属性及其运动特征的模型。通常由数学解析式、逻辑表达式和逻辑图等组成。

[来源：GJB 7099.2-2010]。

3.3

仿真模型 Simulation Model

将数学逻辑模型转换为计算机程序语言描述的软件模型。

[来源：GJB 7099.2-2010，有修改]。

3.4

仿真步长 Simulation Step

在离散系统中，仿真系统推演时所采用的时间间隔。

3.5

基模型 Base Model

作为一个整体，能够独立完成某种仿真的模型。

3.6

复合模型 Composite Model

由多个基模型组成的模型。

3.7

仿真要素类型 Simulation Element Type

仿真模型在真实世界中所对应的实物、现象或者方法的类别。

3.8

仿真驱动方式 Simulation Time Management

以时间、事件、数据或者前三种的任意混合的形式推进仿真系统的运行。

3.9

科学有效载荷 Science Payload

装载于空间技术平台上，用于执行特定科学实验、科学探测与应用研究任务的仪器及设备系统。

注：简称有效载荷或载荷。

[来源：GB/T 30114.1-2013，有修改]。

4 总则

空间科学任务论证是从技术上、工程上和经济角度评估和确定空间任务的可行性。空间科学任务论证系统级仿真系统需要对整个任务的仿真模型、时间、接口、数据交互等方面进行管理。因此，该类系统中的仿真模型描述需考虑仿真要素类型、仿真方法、建模工具以及仿真驱动方式这四项内容。

4.1 仿真方法

系统级仿真可采用的方法如下（但不限于如下方法）：

a） 蒙特卡罗法

b） 磁流体力学仿真法

c） 有限元法

仿真方法的选择应遵循以下原则：

a） 根据空间科学任务中涉及的载荷（或者单机）的学科特性，从系统层面考虑在仿真模型接入时其系统资源分配的潜在需求；

b） 根据任务在轨运行环境及其对探测器、探测流程、探测数据等方面的影响开展分析（通常包

括空间环境、热、力、磁等多物理场方面的仿真分析）。

4.2 仿真要素类型

空间科学任务仿真系统中包含的仿真要素，包括探测目标、载荷（或者单机）、探测数据、空间环境及其影响（如在轨运行环境、多物理场影响分析）、天体及其表面环境、数值计算等。

4.3 仿真驱动方式

仿真系统的驱动方式可以分为数据驱动、事件驱动、时间驱动和混合驱动方式。当仿真模型明确其驱动方式后，可以支持仿真系统时间管理算法的选择提供依据。

4.4 建模工具

当仿真模型是基于专业建模工具构建的，仿真系统可以基于该信息对此类模型匹配相应的接口，完成接口管理，便于数据交互。

5 模型描述

5.1 空间科学任务仿真系统的构成

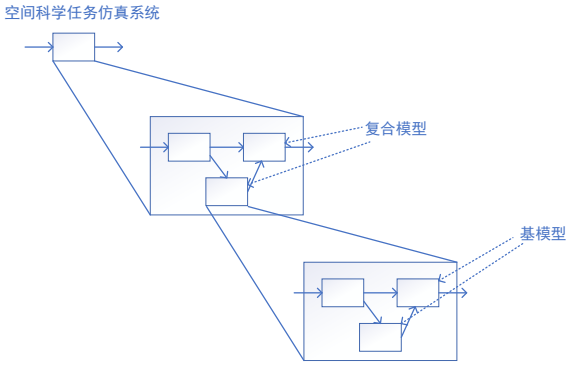

图 1 空间科学任务仿真系统示意图

为了满足空间科学任务的仿真，定义了整个任务仿真系统（如图 1 所示）的描述：

$$VS = \{T_s, T_e, C, Rel\} \quad (1)$$

式中：

T_s——仿真起始时间，$T_s > 0$，$T_s \in \boldsymbol{Z}^+$；

T_e——仿真结束时间，$T_e > T_s > 0$，$T_e \in \boldsymbol{Z}^+$；

C——复合模型集合；

Rel——与复合模型中的Rel_i定义相同，表示复合模型之间的交互关系。

5.2 复合模型的构成

复合模型（Composite Model ）定义如下，i即为复合模型的 ID，$i \in \boldsymbol{Z}^+$：

$$C_i = \{M, A_i, Rel_i, R_i\} \quad (2)$$

式中：

M——为模型集合，其元素为$(ID, InitFile)$，分别表示模型的 ID 和该模型实例的初始化文件，此集合中既可以包含基模型，也可以是复合模型；

A_i——A_i为模型i的属性集合（无模型i的仿真时间步长集合T_i）；

Rel_i——为模型集合M中模型之间的交互关系集合，定义如下：

$$Rel_i = \{Rel_{i1}, Rel_{i2}, \ldots\ldots, Rel_{ij} | j \in Z^+\} \quad \text{（3）}$$

式中:

$Rel_{ij} = \{SrcName, SrcID, DesName, DesID\}$，$SrcName$、$SrcID$、$DesName$和$DesID$分别为交互关系中输出方的方法名称、输出方的模型实例 ID、输入方的模型方法名称以及输入方的模型实例 ID。

R_i的定义见基模型的定义（见基模型中的定义）。

复合模型示意图如图所示。

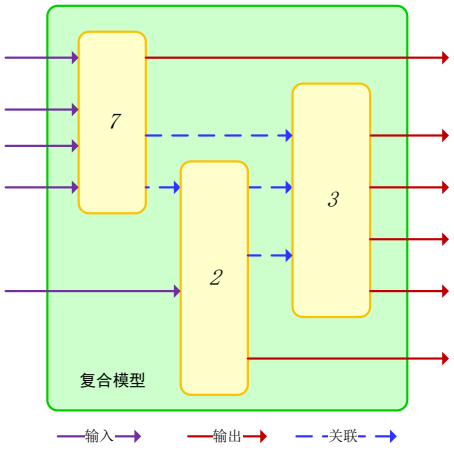

图 2 复合模型示意图

由于仿真模型中所采用的描述方法对外是透明的，因此不需要在模型文件中描述。

5.3 基模型的构成

M为模型集合，$m_i \in M, i \in \boldsymbol{Z}^+$（$Z^+$表示正整数），$i$即为模型的标识（Identifier，ID），模型描述具体如下：

$$m_i = \{A_i, F_i, E_i, R_i\} \quad \text{（4）}$$

a）A_i为模型i的属性集合，定义如下：

$$A_i = \{\mathrm{Sys}_i, \mathrm{N}_i, \mathrm{Met}_i, \mathrm{Mt}_i, \mathrm{Sim}_i, \mathrm{Obj}_i, \mathrm{T}_i, \mathrm{Pri}_i\} \quad \text{（5）}$$

式中：

Sys_i——模型i为空间科学探测任务中仿真要素；

N_i——模型i名称；

Met_i——模型i所采用的或者所需仿真方法，用字符串表示，此项可以为∅，表示无特殊仿真方法；

Mt_i——模型i所采用的建模工具，可使用其建模工具通用名称表示，此项也可以为∅，表示此模型构建时未采用专业建模工具，为自主研发的仿真模型；

Usagei——模型的用途；

Sim_i——模型i所需的仿真驱动方式，用字符串表示，此项也可以为$\emptyset$，表示默认为时间驱动方式；

Obj_i——模型i的实例 ID，其标识可自行定义；

T_i——模型i的仿真时间步长集合，定义为：

$$T_i = \{(step, unit)_{minStep}, (step, unit)_{maxStep}, (step, unit)_{optStep}\} \quad (6)$$

式中：

step——仿真时间步长大小

unit——*步长单位*

minStep——*最小步长*

maxStep——*最大步长*

optStep——*最佳仿真步长（根据仿真模型的计算精度要求确定）*

注：式（6）中都用二元组$(step, unit)$表示。此集合可以为∅，表示此模型没有仿真时间步长限制，如离散事件仿真系统模型。

Pri_i——模型i的优先级（$Pri \in Z+$用于指定模型运行的优先顺序）

b）F_i——为模型的方法集合，即模型的行为，定义如下：

$$F_i = \{I_i, O_i\} \quad (7)$$

式中：

I_i——*模型*i*的输入变量列表（定义为*$I_i = \{I_{i1}, I_{i2}, \ldots, I_{ij} | j \in \mathbf{Z}^+\}$*）；*

注：其中$I_{ij} = (name, num, frame, type, unit)$，$name$，$num$，$frame$，$type$和$unit$分别为输入变量的名称、变量包含数据个数、变量所属参考系（时间或者坐标系）、类型和单位；类型有普通和指令两种。

O_i——*模型*i*的输出变量列表（定义为*$O_i = \{O_{i1}, O_{i2}, \ldots, O_{ij} | j \in \boldsymbol{Z}^+\}$*）*

注：其中$O_{ij} = (name, num, frame, type, unit, u, gui, vis)$，$name$，$num$，$frame$，$type$，$unit$，$u$，$gui$和$vis$分别为输出变量的名称、变量包含数据个数、变量所属参考系（时间或者坐标系）、类型、单位、是否具有不确定性、界面显示名词和可视化类型；可视化类型分为二维、三维和二者皆有三种；

c）E_i为模型i的事件列表，其中按仿真时间排序，可以为∅，定义如下：

$$E_i = \{e_{i1}, e_{i2}, \ldots\ldots, e_{ij} | j \in \boldsymbol{Z}^+\} \quad (8)$$

式中：

$e_{ij} = \{N_{f_{ij}}, t_{f_{ij}}, T_{f_{ij}}\}$

$N_{f_{ij}}$——*事件*j*的标识，用于模型内部事件的解析；*

$t_{f_{ij}}$——*事件*j*发生的时间，随机、按照脚本等；*

$T_{f_{ij}}$——*事件*j*的类型，可以是指令、故障或者其他自定义事件；*

d）R_i为第i个模型的管理相关信息，定义如下：

$$R_i = \{G_i, IR_i, OR_i, B_i, D_i\} \quad (9)$$

式中：

G_i——*模型*i*的粒度（*$G_i \in \boldsymbol{Z}^+$*，为0时表示该模型为基模型；其他正整数表示该模型为复合模型，值越大表示其递归的层次越多）；*

IR_i——*模型*i*的输入分辨率；*

OR_i——*模型*i*的输出分辨率；*

B_i——*模型*i*属于哪个复合模型的模型ID集合；*

D_i——*模型*i*的存储位置，如模型存储的相对路径或者绝对路径。*

附录 A

（资料性附录）

仿真模型描述示例

A.1 基模型模型描述示例

为便于计算机读写，模型的描述采用类 XML 方式编写，如图 A.1 所示。

```
<BaseModel>
    <Attributes ID = "7" Sys = "Payload" ObjID = "7-1" Name = "Darkmatter" Priority = "1">
        <Step>
            <Min step = "1" unit = "ms"/>
            <Max step = "1000" unit = "ms"/>
            <Opt step = "50" unit = "ms"/>
        </Step>
    </Attributes>
    <Functions>
        <Input name = "power" num = "1" frame = "none" type = "string" unit = "V" source = ""/>
        <Input name = "angle" num = "3" frame = "PAYLOAD" type = "string" unit = "rad" source = ""/>
        <Input name= "satelliteAttitude" num = "3" frame = "SATELLITE" type = "string" unit = "rad" source = ""/>
        <Input name= "payload" num = "1" frame = "none" type = "command" unit = "none" source = ""/>
        <Input name= "payloadMal" num = "1" frame = "none" type = "failure" unit = "none" source = ""/>
        <Output name = "State" num = "1" frame = "none" type = "vis" unit = "none" u = "false" gui = "载荷状态" vis = "3D"/>
        <Output name = "DataQuantity" num = "1" frame = "none" type = "vis" unit = "MB" u = "false" gui = "数据量" vis = "2D"/>
        <Output name = "payloadAttitude" num = "3" frame = "PAYLOAD" type = "vis" unit = "degree" u = "true" gui = "载荷姿态" vis = "both"/>
    </Functions>
    <Events>
        <Event>
            <Name>PAYLOAD1</Name>
            <TimeofOccurance>Script</TimeofOccurance>
            <Type>Command</Type>
        </Event>
        <Event>
            <Name>PAYLOAD1</Name>
            <TimeofOccurance>Random</TimeofOccurance>
            <Type>Failure</Type>
        </Event>
    </Events>
    <ManagementInfo G = "0" IR = "5" OR = "3">
        <S>null</S>
        <Dir>null</Dir>
    </ManagementInfo>
</BaseModel>
```

图 A.1 基模型模型描述

XML schema 如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- Root Element -->
  <xs:element name="Models">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="BaseModel" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!-- Base Model -->
  <xs:element name="BaseModel">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Attributes"/>
        <xs:element ref="Functions"/>
        <xs:element ref="Events"/>
        <xs:element ref="ManagementInfo"/>
```

```
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- Attributes -->
<xs:element name="Attributes">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Step" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="ID" type="xs:string" use="required"/>
    <xs:attribute name="Sys" type="xs:string" use="required"/>
    <xs:attribute name="Name" type="xs:string" use="required"/>
    <xs:attribute name="SimMethod" type="xs:string" use="optional"/>
    <xs:attribute name="ModelingTool" type="xs:string" use="optional"/>
    <xs:attribute name="Usage" type="xs:string" use="required"/>
    <xs:attribute name="TimeManagementType" type="xs:string" use="optional"/>
    <xs:attribute name="ObjID" type="xs:string" use="required"/>
    <xs:attribute name="Priority" type="xs:positiveInteger" use="required"/>
  </xs:complexType>
</xs:element>

<!-- Step -->
<xs:element name="Step">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Min"/>
      <xs:element ref="Max"/>
      <xs:element ref="Opt"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- Min/Max/Opt -->
<xs:element name="Min">
  <xs:complexType>
    <xs:attribute name="step" type="xs:decimal" use="required"/>
    <xs:attribute name="unit" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>

<xs:element name="Max">
  <xs:complexType>
    <xs:attribute name="step" type="xs:decimal" use="required"/>
    <xs:attribute name="unit" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>

<xs:element name="Opt">
  <xs:complexType>
```

```
    <xs:attribute name="step" type="xs:decimal" use="required"/>
    <xs:attribute name="unit" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>

<!-- Functions -->
<xs:element name="Functions">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Input" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="Output" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- Input -->
<xs:element name="Input">
  <xs:complexType>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="num" type="xs:positiveInteger" use="required"/>
    <xs:attribute name="frame" type="xs:string" use="required"/>
    <xs:attribute name="type" type="xs:string" use="required"/>
    <xs:attribute name="unit" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>

<!-- Output -->
<xs:element name="Output">
  <xs:complexType>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="num" type="xs:positiveInteger" use="required"/>
    <xs:attribute name="frame" type="xs:string" use="required"/>
    <xs:attribute name="type" type="xs:string" use="required"/>
    <xs:attribute name="unit" type="xs:string" use="required"/>
    <xs:attribute name="u" type="xs:boolean" use="required"/>
    <xs:attribute name="gui" type="xs:string" use="required"/>
    <xs:attribute name="vis" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>

<!-- Events -->
<xs:element name="Events">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Event" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- Event -->
```

```
<xs:element name="Event">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="TimeofOccurance">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Script"/>
            <xs:enumeration value="Random"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Type">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Command"/>
            <xs:enumeration value="Failure"/>
            <xs:enumeration value="Custom"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- Management Info -->
<xs:element name="ManagementInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="S">
        <xs:simpleType>
          <xs:union memberTypes="xs:string">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="null"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:union>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Dir">
        <xs:simpleType>
          <xs:union memberTypes="xs:string">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="null"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:union>
        </xs:simpleType>
```

```
        </xs:element>
      </xs:sequence>
      <xs:attribute name="G" type="xs:integer" use="required"/>
      <xs:attribute name="IR" type="xs:positiveInteger" use="required"/>
      <xs:attribute name="OR" type="xs:positiveInteger" use="required"/>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

A.2 复合模型描述

以图 A.1 为例的复合模型描述如图 A.2 所示：

```
<CompositeModel>
<BaseModels>
    <Model ID = "7" File = "Init7.xml" />
    <Model ID = "2" File = "Init2.xml" />
    <Model ID = "3" File = "Init3.xml" />
</BaseModels>
<Attributes ID = "8" Name = "MyComposite" Priority = "1"/>
<Relationship>
    <Rel SourceName = "DataQuantity"  SourceID = "7" DestinationName = "dataquantity" DestinationID = "3"/>
    <Rel SourceName = "payloadAttitude"  SourceID = "7" DestinationName = "pAttitude" DestinationID = "2"/>
    <Rel SourceName = "ChannelState"  SourceID = "2" DestinationName = "cState" DestinationID = "3"/>
    <Rel SourceName = "velocity"  SourceID = "2" DestinationName = "Vel" DestinationID = "3"/>
</Relationship>
<ManagementInfo G = "0" IR = "5" OR = "6">
    <S>null</S>
    <Dir>null</Dir>
</ManagementInfo>
</CompositeModel>
```

图 A.2 复合模型描述示例

XML schema 如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- Root Element -->
  <xs:element name="Models">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="CompositeModel" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!-- Composite Model -->
  <xs:element name="CompositeModel">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Models"/>
        <xs:element ref="Attributes"/>
        <xs:element ref="Relationship"/>
```

```
        <xs:element ref="ManagementInfo"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!-- Models (Component References) -->
  <xs:element name="Models">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ModelSequence" type="xs:string" minOccurs="1"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!-- Model Sequence -->
  <xs:element name="ModelSequence">
    <xs:complexType>
      <xs:attribute name="ID" type="xs:string" use="required"/>
      <xs:attribute name="InitFileName" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>

  <!-- Attributes -->
  <xs:element name="Attributes">
    <xs:complexType>
      <xs:attribute name="ID" type="xs:string" use="required"/>
      <xs:attribute name="Sys" type="xs:string" use="required"/>
      <xs:attribute name="Name" type="xs:string" use="required"/>
      <xs:attribute name="SimMethod" type="xs:string" use="optional"/>
      <xs:attribute name="ModelingTool" type="xs:string" use="optional"/>
      <xs:attribute name="Usage" type="xs:string" use="required"/>
      <xs:attribute name="TimeManagementType" type="xs:string" use="optional"/>
      <xs:attribute name="ObjID" type="xs:string" use="required"/>
      <xs:attribute name="Priority" type="xs:positiveInteger" use="required"/>
    </xs:complexType>
  </xs:element>

  <!-- Relationship Network -->
  <xs:element name="Relationship">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Rel" minOccurs="1" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!-- Relationship Link -->
  <xs:element name="Rel">
    <xs:complexType>
```

```
      <xs:attribute name="SourceName" type="xs:string" use="required"/>
      <xs:attribute name="SourceID" type="xs:string" use="required"/>
      <xs:attribute name="DestinationName" type="xs:string" use="required"/>
      <xs:attribute name="DestinationID" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>

  <!-- Management Info -->
  <xs:element name="ManagementInfo">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="S">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="null"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Dir">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="null"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="G" type="xs:integer" use="required"/>
      <xs:attribute name="IR" type="xs:positiveInteger" use="required"/>
      <xs:attribute name="OR" type="xs:positiveInteger" use="required"/>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

附录 B
（规范性附录）
仿真模型接口定义

B.1 概述

仿真模型的接口可分为三类——固有接口、自定义接口和可选接口。固有接口为每个模型必须提供的接口；自定义接口为各模型自定义的，但是其命名必须符合虚拟卫星仿真模型接口规范；可选接口模型可以提供，也可以不提供。

B.2 接口定义

固有接口的语法格式固定，包括方法名称、返回值类型、参数顺序和类型，如表 B.1 所示。

表 B.1 固有接口

用途	名称	参数	参数类型	返回值
初始化模型	Init	文件路径	字符串	布尔型
运行	Run	无	无	布尔型
设置仿真步长	SetStep	步长数值	整型（64 位）	布尔型
销毁	Destroy	无	无	布尔型

自定义接口是为模型的输入和输出设计的，具体定义如表 B.2 所示。

表 B.2 自定义接口

用途	名称	参数	参数类型	返回值
输入	SetXXX	输入的数值或者文件名称	字符串	布尔型
输出	GetXXX	无	无	字符串

例如：图 A.1 中定义了 Input 变量名称 power，则该模型的方法应写为（以 C 语言为例）Setpower（ char * power ）

可选接口

模型若存在事件，需自定义处理方法，且仅在模型内被调用，对外是透明的，如 B.3 所示。其中，“XXX”自定义事件类型。

表 B.3 可选接口

用途	名称	参数	参数类型	返回值
执行事件	SetCommandXXX	事件	字符串	布尔型

B.3 仿真模型的组装

为了能够提高模型的可重用性、支持多种模型的集成，支持仿真的快速构建并确保模型组合的可

行性，结合上述内容，本节给出仿真模型组装方法示例，它是一种多分辨率、多粒度的模型组装方法，如图 所示。该过程将符合上述接口规范的模型，通过模型选择、模型可组合判定和方案确定三大步骤组装起来。在模型选择过程中都存在一定的计算复杂性，为了加速模型组装过程，这里的模型为已存在的一组仿真模型。模型组装方案确定过程则对模型信息，按照一定的数据结构进行组织与管理，从而为仿真的运行做准备。

利用仿真模型配置文件进行仿真应用系统组装是一个通用且有效的方法。仿真模型配置文件相当于仿真应用系统的初始化配置文件，在 HLA 系统的设计中，每个仿真应用系统都有 FOM 和 SOM 文件，FOM 文件提供了联邦成员之间的公共的、标准化的格式进行数据交换的规范，它描述了在仿真运行过程中将参与仿真的联邦成员之间的信息交换的对象类、对象类属性、交互类、交互参数的特性。

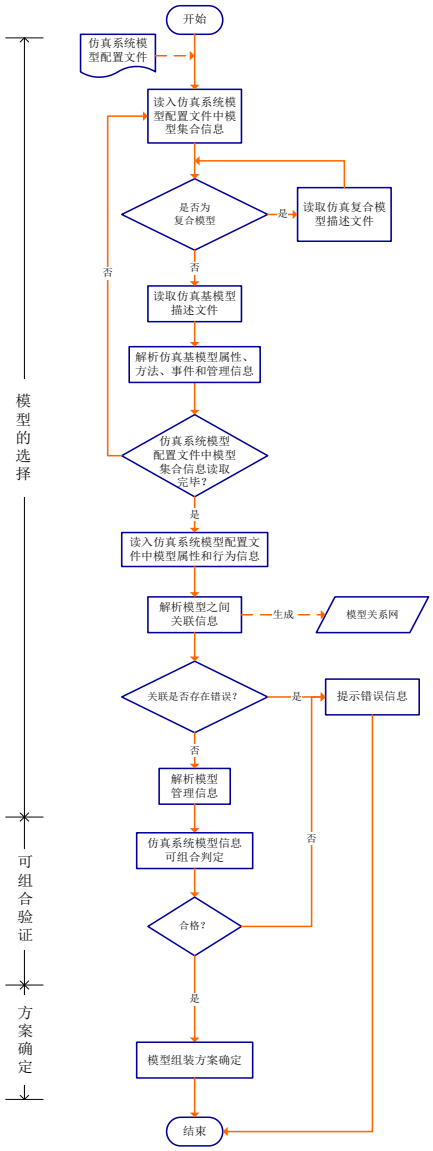

图 B.1 空间科学任务论证系统级仿真模型组装方法

这里采用了仿真模型配置文件的方法解决了仿真模型之间的交互问题，仿真模型配置文件成为仿真模型和仿真应用之间的桥梁和纽带。仿真系统读取配置文件，配置文件分为两个部分，一是仿真配

置部分；二是仿真脚本。仿真配置部分描述了仿真运行的基本信息，如起止时间、仿真步长、模型信息等。其中模型信息描述了参与仿真的模型标识，以及该模型的初始化信息，仿真系统在读取到模型标识之后，根据此标识到模型库去加载模型，利用初始化参数进行系统的初始化工作。

仿真模型配置文件内同样描述了仿真模型之间的信息交互关系，即需要公布那些数据，订购那些数据和相关模型的初始化参数。仿真模型在启动时读取该仿真模型的初始化配置文件。

参考文献

[1] 空间科学概论 科学出版社 2023.02

[2] Functional Mock-up Interface Specification，Version 3.0, 2022-05-1。