

系统建模语言概览

SysML 是一种应用广泛，表意丰富的图形建模语言，可以使系统设计的重要方面——结构、行为、需求和参数（数学模型）——可视化，便于就设计内容进行沟通。如第 1 章所讨论的，SysML 可以作为 MBSE 的第一支柱使用。

本章提供了对 SysML 的高层次概览：SysML 的总体目的、9 种 SysML 图各自的目的，以及对它们全部适用的普遍概念。这些讨论会为后续章节深入探讨每种图提供重要的背景信息。

2.1 SysML 是什么，不是什么

SysML 是多种图形建模语言中的一种。这里的关键词是语言。SysML 是一种语言——人与人沟通的媒介。它和我们所说的自然语言（例如：汉语、日语、英语）一样，有语法和词汇。SysML 是 MBSE 实践者在创建系统模型时“说”的语言，可以把他们的系统设计观点可视化，并与利益相关者沟通。

之所以给“说”字加引号，是因为 SysML 是一种图形语言。它的词汇包含能够表示特殊意义的图形标识。例如，带有箭头的虚线和带有箭头的实线（后续章节会详细介绍）拥有不同的意义。关键在于 SysML 的目的：在利益相关者之间可视化并沟通系统设计。

SysML 的语法和标识法都定义在标准说明书里面，它由对象管理组织（Object Management Group, Inc. OMG）发布，并归其所有。OMG 是由上百家计算机行业公司、政府机构和学术单位组成的联合体，它们合作以创建一系列企业集成标准，并推动业务技术。你可以在它的网站 www.omg.org 上找到更多关于 OMG 的信息。你在那里还会找到 SysML 的说明文档——关于 SysML 规则信息的重要资源。

SysML 说明书试图以精确、清晰的方式来定义这种建模语言的语法，很大程度上它已经成功达到了这个目的。但说明书中的文字有时可能难以解析和执行。这份 SysML 信息主要来源的目标读者是建模工具厂商以及建模语言的设计者（以及撰写这个主题相关书籍的建模极客们）。简言之，SysML 说明书并不是给初学者使用的。

如果你觉得那并不会难倒你，那么更可怕的还在后头。如下一节所述，SysML 并不是一种独立的语言。而是统一建模语言（Unified Modeling Language, UML）子集的一种形式（扩展）。因此，如果你想知道 SysML 语法和词汇表的完整定义，那么还需要参考 UML 说明文档的一部分内容（在 OMG 的网站上也可以找到）。

如果你第一次接触 SysML，我建议你跳到附录 A，从而对其各种各样的图形标识法有个初步认识。那些标识法将在你创建的 SysML 图中出现。这些标识法也组成了系统建模语言的词汇表。

那就是 SysML：一种建模语言。了解 SysML 不是什么也同样重要：它不是一种建模方法。说得更明白一点，正式的 SysML 说明文档只定义了语言本身（语法和词汇），而没有规定任何特定的建模方法。

注意 请参考第 1 章对建模语言和建模方法之间区别的讨论。

例如，SysML 规格说明书并没有告诉你应该在生命周期的哪个时间点创建用例图。也不会指定你必须使用活动图来详细说明用例。它不会要求你创建一系列内部模块图（Internal Block Diagram, IBD），每个模块专注于说明系统架构的特定方面。所有这些都是方法学上的决定，都不在 SysML 说明书的讨论范围之内。采纳建模方法并对其进行剪裁完全是你和团队的工作，你们要自己确定哪种方法能达到项目的特定目标。

2.2 SysML 确实基于 UML——但你可以从 SysML 开始

如前所述，SysML 并不是一种独立的语言。它是 UML 的一种形式或者扩展，而 UML 是特别为系统工程领域所创建的。UML 被设计为针对软件工程领域的标准化建模语言。系统工程师发现，使用标准的建模语言来构建系统模型很有价值，但他们并不觉得 UML 足以记录系统工程中所有有意义的概念。

例如，UML 模型可以包含 `DataType` 元素。软件工程师可以使用在 UML 模型中使用一种数据类型（例如，整型）指定类中一种属性的类型、能够在活动之间流动的对象类型，以及一项操作中参数的类型。然而，系统工程师会关注非数据的可流动的其他类型的事物，像事件和能量。这样，`DataType` 的概念就不够充分了。因此，SysML 引入了一种新的模型元素叫做 `ValueType`，它扩展了 `DataType` 的概念，从而为系统工程领域的广泛类型提供了更贴切的术语。

因为 SysML 是 UML 的一种扩展，所以 SysML 的某些规则实际上是在 UML 规格文档中定义的。这意味着 SysML 规格文档本身并没有对语言进行充分的定义。例如，如果你想要知道在系统模型中使用值类型的所有规则，那么仅仅阅读 SysML 说明中的 `ValueType` 入口是不够的；你还需要阅读 UML 说明中的 `DataType` 入口内容。那是语言扩展机制本身所决定的。

那么，你是否需要去买一本 UML 的书籍，来学习如何创建系统模型呢？

其实并不需要。这本书作为学习 SysML 的入门书籍已经足够，可以让你尽快开始创建模型。当在 3.9 节中讨论值类型的时候，就会告诉你所有需要知道的内容，让你可以在系统模型中正确、有效地使用 SysML。你根本不需要知道一个特定的细节是来自于 UML 关于基本元素 `DataType` 的定义，还是来自于 SysML 的扩展 `ValueType` 的定义。

2.3 SysML 图概览

一共有 9 种 SysML 图：

- 模块定义图 (Block Definition Diagram, BDD)
- 内部模块图 (Internal Block Diagram, IBD)
- 用例图
- 活动图
- 序列图
- 状态机图
- 参数图
- 包图
- 需求图

图 2.1 来自 SysML 规格 v1.2，很好地概括了 SysML 种类以及各种图之间的关系。但你要知道带空三角形箭头的线的意义，才能够理解这幅图的意思。那些线叫做泛化。你可以按照箭头的方向把它读作“……是……的一种类型”。（3.6 节详细讨论泛化）

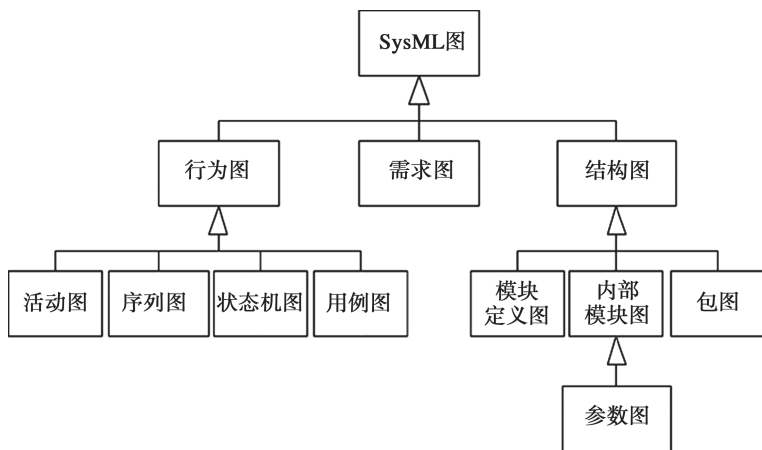


图 2.1 SysML 图分类

知道以上内容之后，我们就能够从图 2.1 中得到更多信息。活动图、序列图、状态机图和用例图都属于行为图。模块定义图、内部模块图和包图都属于结构图。参数图是内部模块图；根据传递性，它也是结构图。需求图自成一类——但它还是 SysML 图这个家族的重要组成部分。

以下是每种类型图的目的的简要总结。

- ❑ **模块定义图**用于表示模块和值类型之类的元素（定义能够在可操作的系统中存在的事物类型）以及那些元素之间的关系。BDD 的通常用法包括显示系统层级关系树以及分类树。
- ❑ **内部模块图**用于指定单个模块的内部结构。更精确的说法是，IBD 会显示模块内部组成部分之间的关系，以及它们之间的接口。
- ❑ **用例图**用于表达系统执行的用例，以及引起用例的行为者和其中的参与者。用例图是系统在行为者的协作下所执行服务的黑盒视图。
- ❑ **活动图**用于指定一种行为，主要关注控制流程，以及输入通过一系列动作转换为输出的过程。活动图一般用作一种分析工具，以理解和表达系统所需要的行为。

- **序列图**用于指定一种行为，主要关注模块的组成部分如何通过操作调用和异步信号交互。序列图通常用作详细设计工具，以精确地把一种行为指定为生命周期开发阶段的输入项。序列图也是指定测试案例的一种优秀机制。
- **状态机图**用于指定一种行为，主要关注模块的一系列状态，以及响应事件时，状态之间的可能转换。状态机图和序列图一样，都可以精确说明一个模块的行为，可以作为生命周期开发阶段的输入项。
- **参数图**用于表示一种或多种约束——特别是等式和不等式——如何与系统的属性绑定。参数图支持工程分析，包括性能、可靠性、可用性、电力、人力和成本。参数图还可以用于支持候选物理架构的优劣势研究。
- **包图**用于显示模型以包相互包含的层级关系形式组织的方式。包图可能还会显示包所包含的模型元素，以及包之间的依赖关系和它们包含的模型元素。
- **需求图**用于表示基于文字的需求、需求之间的关系（包含关系、继承关系以及复制关系），以及满足、验证和改善它们的其他模型元素。

2.4 通用的图概念

在你钻研特定类型 SysML 图之前，应该知道关于它们的一些总体概念。一幅示例 SysML 图如图 2.2 所示。

每幅图都会有外框、内容区域（也叫做画布）和头部。图的外框是外部的矩形。内容区域是外框内部的区域，其中可以显示模型元素和关系。头部位于图的左上角，如图 2.2 所示，其右下角被截掉一部分。

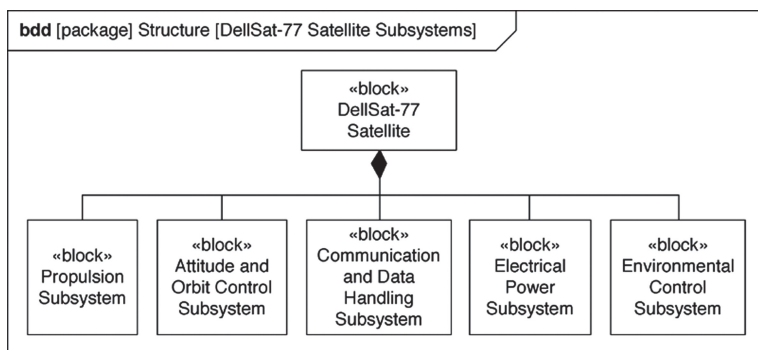


图 2.2 示例 SysML 图

在 SysML（和 UML 不同）中，外框必须显示。也就是说，本书中的一些图可能会在不闭合的外框中显示模型元素和关系。我那样做是为了隐藏后续的信息，并把你的注意力集中在特定的标识法上。但是，正式的情况下一定要使用外框。

头部信息的格式是一种最重要的图概念。头部一般包含四段信息：

- 图的类型
- 模型元素类型
- 模型元素名称
- 图的名称

那些信息的格式如图 2.3 的头部所示。

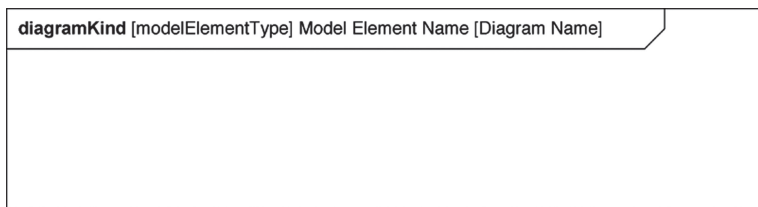


图 2.3 图的头部格式

我首先介绍两项直观的信息：图的类型以及图的名称。图的类型以 SysML 定义的缩写显示：

- bdd = 模块定义图
- ibd = 内部模块图
- uc = 用例图
- act = 活动图
- sd = 序列图
- stm = 状态机图
- par = 参数图
- req = 需求图
- pkg = 包图

基于上述内容，你可以判断出图 2.2 中的图是一幅模块定义图（第 3 章会详细讨论可以在 BDD 中显示的元素类型）。

图的名称可以任意设置。我建议你选择能够传达图所关注的、关于模型内容的名称。例如，图 2.2 中图的名称是“DellSat-77 卫星子系统”。这个图的名称表示图的

关注点在于一系列组成卫星系统的子系统。模型肯定会包含关于卫星系统的其他信息，但那些信息并不是这幅图的关注点所在。

头中接下来的两段信息是模型元素的类型和模型元素的名称。为了理解它们指的是什么，你首先需要了解关于 SysML 的另一个核心内容：你所创建的每个图都代表你已经在系统模型中某处定义的元素。更精确的说法是，图的外框代表模型中的一个元素。而那个模型元素的类型和名称正如图的头部所写。

在图 2.2 中，模型元素的类型是“包”，而模型元素的名称是“Structure”。这告诉我们，这个 BDD 的外框代表存在于系统模型层级关系中某处的 Structure 包。

要求每个图都代表一种模型元素看起来像是一种严格且没有必要的约束，但据 Frederick Brooks 先生在《设计原本》(The Design of Design)[⊖]中所说，“约束是朋友”。图是模型元素的交付物，也是 SysML 作者关于其组成部分做出的决定。原因在于 SysML 图概念的另一个关键方面：图所代表的模型元素会为图中所显示的其他元素定义命名空间，也就是模型层级关系中的容器元素。只要简单放置，图的头部显示的模型元素类型和模型元素名称就会指出图中的元素在模型中的位置。

图 2.2 中，图的头部告诉我们，内容区域中显示的 6 个模块包含（内嵌）在模型层级关系的 Structure 包中。这让我们了解到元素在模型中是如何分布的，并有助于查找。

模型元素可能是结构化元素（例如，一个包或者模块），或者可能是行为元素（例如，一种活动、交互或者状态机）。图能够代表的模型元素的类型取决于你所创建的图种类。它们之间的匹配关系如表 2.1 所示。

表2.1 每种图可表示的模型元素类型

图 类 型	可表示的模型元素类型
模块定义图	包、模型、模型库、视图、模块、约束模块
内部模块图	模块
用例图	包、模型、模型库、视图
活动图	活动
序列图	交互
状态机图	状态机
参数图	模块、约束模块

⊖ 本书由机械工业出版社引进并出版。

(续)

图 类 型	可表示的模型元素类型
需求图	包、模型、模型库、视图、需求
包图	包、模型、模型库、视图、概况

让我们回忆一下第 1 章的内容，想想系统模型（作为一种工程产出物）以及你创建的一系列图（它们都是底层模型的视图）之间的区别。这非常重要，所以我在这里更加正式地重新强调一下，并把它叫做：基于模型工程的基本规则。你的任务是聚精会神，心中默念：

模型的图永远不是模型本身；它只是模型的视图。

这种观点来自于很久以前的工程师，他们只使用纸、白板或者制图工具绘制设计草图。使用比喻能够很好地说明这个观点：模型是一座山，而图是山的照片。不管是否有人照照片，山都会存在。如果一个人从北面照，那么就创建了一幅视图，可以显示出一些特性，但不会显示其他特性。如果另一个人从西面拍摄照片，那么就为山峰创建了另一种视图，显示了一系列不同的特性（两种视图可能会有重叠）。

这两幅视图会关注整体的不同方面。但照片不管什么时候都只是山峰的视图，而不是山峰本身。山峰和它的特性会继续存在，即便摄影师从最终的照片上划掉了特定的细节，甚至销毁了那些照片。

这种比喻不适用于一种情况：你可以在任何时候向模型添加新特性并修改，或者从中删除现存的特性。当你修改现存的特性时，修改会立刻反映到显示特性的图上。当你从模型中删除特性的时候，它马上就会从显示那种特性的所有图上消失。当然，你相簿中的照片是不具备这种能力的。

在这一节前面的内容中，我提到在特定的图中删除不重要的信息。每种图都不能试图说明所有细节；那样的图是不可读的。你应该决定特定的图应该专注于哪些方面，并忽略不在关注之内的所有模型信息。这个观点会得出基于模型工程基本规则的推论：

你不能因为某个特性在一幅图中没有出现就断定它不存在；它可能会在模型的另一幅图中出现，也可能根本就不出现。

小结

SysML 是一种能够表达丰富内容的图形建模语言，你可以使用它把系统的结构、行为、需求和参数可视化，然后与其他人沟通那些信息。SysML 定义了 9 种图，可以使用它们说明所有系统设计信息；每种图都针对特定的目的，并说明系统一个方面的特定信息。

接下来的章节会详细说明这些图。你会学到在各种图中出现的不同类型的 SysML 模型元素以及它们之间的关系。这里包含了一些对 SysML 规则的讨论，想要正确构建你的系统模型，并确保与利益相关者的有效沟通，你需要知道那些规则。

