

第3章 需求与概念设计方法

3.1 概念设计内容及模式

3.1.1 概念设计内容

一个机电产品的设计过程一般从设计任务开始，经概念设计、具体结构设计，最后到详细设计。概念设计是系统设计中的一个关键步骤，在概念设计过程中一般需要通过分析设计任务、抽象概念，确定功能结构和工作原理，得到系统设计方案。概念设计过程是一个抽象的且极具创造性的过程，因此它在一个产品的设计过程中扮演着非常重要的角色。机械系统概念设计的基本流程如下。

1) 明确任务并建立需求

获取用户和设计人员的需求，并对这些需求进行分析和验证。从用户处获取的需求可能是不准确、模糊的，需要在此需求上进行抽象和规范，得到明确规范的且易于处理的需求，确定产品需要实现的总功能。

2) 功能结构分析

机电产品一般是一个复杂的系统，因此在机电产品设计过程中，从需求中提取出的总功能也会比较复杂，要想直接对该总功能进行求解将会非常困难。因此在总功能确定后，我们需要进行功能分解，将一个复杂的功能分解为多个子功能，一般采用功能结构来描述各个子功能之间的关系。对于某个子功能也可以进行多次循环分解，直到得到相对简单且易于实现的子功能为止。得到系统的子功能后再对这些子功能进行处理、求解，得到一个可以实现系统功能的原理方案。

3) 工作机理的选择

功能分解完成后需要确定通过何种工作机理来实现该子功能。工作机理一般是指通过某种工作原理来实现特定的工艺动作。一个系统的功能可以通过一些特定的工艺动作来实现。因此需要将各个子功能与某一个工艺动作对应起来，工艺动作又通过某种工作原理来实现，因此选择合适的工作机理就可以完成系统所需要的功能。

4) 机构设计

通过功能分解和工作机理的选择已经得到了实现系统需要的功能和工作机理。通常工作机理的实现需要一系列的执行机构来完成，因此需要通过工作机理来寻找合适的结构来实现这个系统的功能。

5) 方案评价与优选

在上述概念设计过程中，由于不同功能分解条件及工作机理的选择，将会产生许多能够实现系统功能的解决方案，因此需要采用合适的评价和决策方法来选出合理且最优的方案。概念设计的主要任务是通过功能得到结构以实现系统的设计。人们提出了功能-行为-结构（Function-Behavior-Structure, FBS）模型，FBS模型是一种可用于类比设计的探索模型，将设计知识表达为功能-行为-结构，然后采用功能、行为、结构交互映射的方法得到创新方案。该模型引入了行为描述，增加了功能与结构之间的联系，细化了映射过程，改善了概念设计过程。

3.1.2 概念设计模式

传统上，产品的需求与概念设计都是用文档描述的，没有适当的模型。系统建模语言（System Modeling Language, SysML）是一种支持复杂系统分析、规范、设计、验证和确认的通用图形化建模语言。SysML能够帮助实现系统的规范定义和架构设计，并定义组件的规范。这些组件可以使用其他领域的语言进行设计，系统工程实践正从基于文档的方法转变到基于模型的方法，即基于模型的系统

工程 (Model Based System Engineering, MBSE), SysML 建模语言是实现 MBSE 的有效工具。一个复杂系统可能包括硬件设备、软件、数据、人员、规程、设施,以及其他人造和自然系统元素,采用 SysML 的好处是其能增强规范和设计质量的一致性,增强规范和设计构件的重用,增强开发团队之间的交流,整体提高质量,能生成和控制一个连贯的系统模型,并使用这个模型来规范和设计系统。SysML 有助于 MBSE 方法论的应用,创造一个内聚的、一致的系统模型。

传统的生产质量控制是通过对生产的产品进行检查——用观察与测试的手段来实现的,这种措施通常也被归于检验质量的方法。质量功能展开 (Quality Function Deployment, QFD) 方法则帮助公司从检验产品外在质量转向检查产品设计的内在质量,因为设计质量是产品、工程质量的基石。QFD 早在产品或服务设计成为蓝图之前就已经引进了许多无形的要素,使质量融入生产和服务及其工程的设计之中。QFD 的基本原理就是用“质量屋”(Quality House)的形式,量化分析用户需求与工程措施间的关联度,经数据分析处理后,找出对满足用户需求贡献最大的工程措施,即关键措施,从而指导设计人员抓住主要矛盾,进行稳定性优化设计,开发出满足用户需求的产品。QFD 是产品或服务设计阶段中一种非常有效的方法,是一种旨在提高用户满意度的“用户驱动”式的质量管理方法。

这里将 SysML 与 QFD 两种方法结合起来,实现产品设计中的概念建模和质量控制,如图 3-1 所示。在产品 V 形设计模式基础上,用 SysML 描述产品的概念和系统结构,用 QFD 方法分析产品设计各阶段的质量影响因素,从而改变产品概念模型的文档描述方式,实现 MBSE 的产品开发。

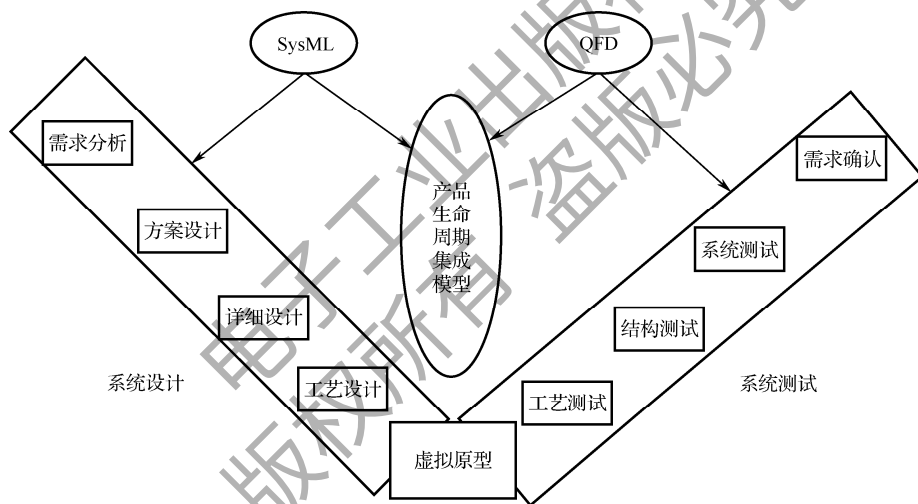


图 3-1 SysML 与 QFD 在产品生命周期中的作用

图 3-1 中, V 形设计模式包括系统设计和系统测试两大部分,其中,系统设计部分的主要任务是建立系统的模型,对需求分析、方案设计、详细设计等多个阶段进行规范描述和分析,这部分工作主要由 SysML 来描述;系统测试部分的主要任务是对设计的模型或结果进行测试,包括工艺测试、结构测试、系统测试、需求确认等,这部分工作主要由 QFD 来实现。需要指出的是,系统设计和系统测试不是简单的串行工作方式,而是交互、迭代的并行工作模式,也就是说 QFD 在产品开始的需求分析阶段就已经介入。

3.2 SysML 建模方法

3.2.1 SysML 基本组成

2003 年, UML 快速发展到了最后一个 1.X 版本 1.5。系统工程界看到 UML 在软件领域的成功,希望在系统工程领域也建立一个建模语言,于是国际系统工程协会 (INCOSE) 联合国际对象管理组织 (OMG) 共同制定了 UML for System Engineering RFP, 希望能将 UML 应用到系统工程领域。成熟

的 UML 作为基础, 加上联合工作组共同努力, 2006 年 6 月 SysML 标准被 OMG 采纳, 2007 年 9 月正式发布 SysML V1.0。

OMG 决定在对 UML2.0 的子集进行重用和扩展的基础上, 提出一种新的系统建模语言——SysML, 并将其作为系统工程的标准建模语言。和 UML 用来统一软件工程中使用的建模语言一样, SysML 的目的是统一系统工程中使用的建模语言。

SysML 为系统的结构模型、行为模型、需求模型和参数模型定义了语义。结构模型强调系统的层次及对象之间的相互连接关系, 包括类和装配。行为模型强调系统中对象的行为, 包括它们的活动、交互和状态历史。需求模型强调需求之间的追溯关系及设计对需求的满足关系。参数模型强调系统或部件的属性之间的约束关系。SysML 为模型表示法提供了完整的语义。

和 UML 一样, SysML 语言的结构也是基于四层元模型结构的: 元-元模型、元模型、模型和用户对象。元-元模型层具有最高抽象层次, 是定义元模型描述语言的模型, 为定义元模型的元素和各种机制提供最基本的概念和机制。元模型是元-元模型的实例, 是定义模型描述语言的模型。元模型提供了表达系统的各种包、模型元素的定义类型、标记值和约束等。模型是元模型的实例, 定义特定领域描述语言的模型。用户对象是模型的实例。任何复杂系统在用户看来都是相互通信的具体对象, 其目的都是实现复杂系统的功能和性能。

SysML 能够表示系统、组件和其他对象的结构组成、关联关系和分类; 表达基于流、基于信息和基于状态的物理行为和性能属性的约束; 描述行为、结构和约束之间的分配关系, 以及不同需求之间、不同设计元素和测试用例之间的关系。SysML 从结构、行为、需求和参数 4 个方面来构建系统模型。SysML 提供了 9 种图来支持用户进行系统建模, 如图 3-2 所示。

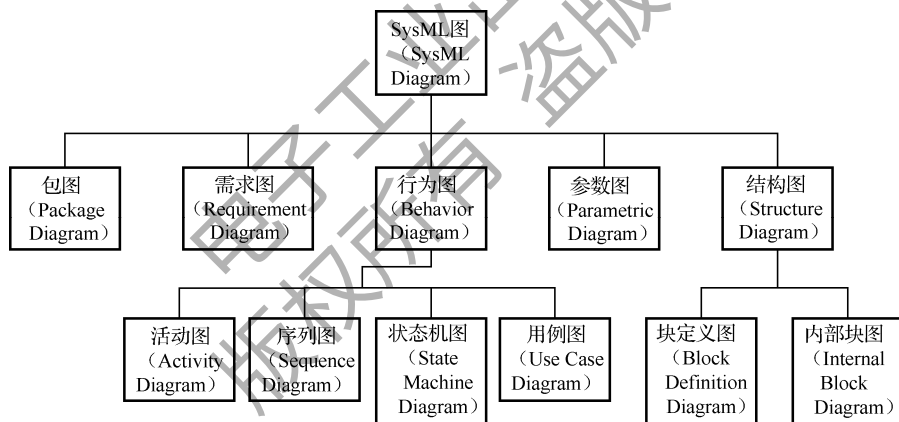


图 3-2 SysML 的 9 种图

1) 包图

包图是用来组织模型的图形, 它可以按照层次关系、图表类型和视点将模型进行分类。

2) 需求图

需求是指系统必须满足的能力或条件, 一个需求能够分解成多个子需求。需求图能够描述需求和需求之间及需求和其他建模元素之间的关系。需求的描述可以有图形、表格和树形结构等各种形式。

3) 活动图

活动图用于描述 workflow、业务流程, 或者是将执行流分解为一系列活动和子活动的算法。活动图可以是简单活动的序列, 或带有条件分支和并发的复杂系列的并行活动。泳道可以添加到活动图以显示负责执行每个活动的实体。活动图强调活动的输入、输出、顺序和条件。

4) 序列图

序列图用于描述对象间的信息交互序列。

5) 状态机图

状态机图通过状态及状态之间的转移对离散行为建模, 它把行为表示为对象的状态历史。在状态的转移、进入和退出过程中会调用活动, 并指定相关的事件和守卫条件。

6) 用例图

用例图描述外部参与者对系统的使用, 这是通过系统向参与者提供一系列服务来实现的。用例图包括用例、参与者及它们之间的通信, 参与者可能是用户、外部系统或其他环境实体, 它们与系统直接或间接交互。

7) 块定义图

块定义图显示系统和系统的基本结构元素(模块, Block), 以及它们之间的关系/依赖性。但是, 它一般用来描述复杂系统的层次结构, 而不显示模块内部的连接关系。

8) 内部块图

内部块图显示了块定义图所定义的系统结构的实现。它包含了一组套件的部件(模块的实例), 这些部件是由端口和接口彼此连接在一起的。

9) 参数图

参数图定义了一组系统属性及属性之间的参数关系。参数关系用来表示系统的结构模型中属性之间的依赖关系, 说明一个属性值的变化怎样影响其他的属性值, 参数关系是没有方向的, 可以是基本的数学操作符号, 也可以是和物理系统的性质有关的数学表达式(如 $F=ma$) 等。参数模型是分析模型, 把行为模型和结构模型与工程分析模型(如性能模型和可靠性模型)等结合在一起, 用来进行权衡分析, 评价各种备选解决方案。

3.2.2 SysML 建模过程

3.2.2.1 SysML 模型之间的逻辑关系

SysML 4 大模型(结构模型、行为模型、需求模型、参数模型)之间的逻辑关系如图 3-3 所示。系统模型包含系统的规格说明、设计、分析和验证信息, 包含需求、设计、测试用例、设计原理模型元素和它们的内部关系。正如功能-行为-结构的映射关系, 结构模型描述了产品的组成模块, 行为模型描述信息或动作运行过程, 需求模型描述产品的功能要求和技术指标, 参数模型为前三个模型提供约束或参数方程。

组成系统模型的模型元素被存储在一个模型库中, 并通过图形化标志绘制在图上。建模者使用建模工具生成、修改和删除个体模型元素和它们之间的联系。模型元素对应需求、设计、分析和验证信息, 即使它们在不同的图上描述, 也可追溯它们之间的相互联系。例如, 一个发动机在一辆汽车的系统模型中可以有許多联系, 它是汽车系统的一部分, 它连接到变速箱能满足一个动力需求, 它将油料转化为机械能, 有一个质量属性。

建模语言规定了约束关系可以存在的规则。例如, 模型不允许一个需求包含一个系统组件或一个活动, 附加的建模约束可以应用到方法中。所有的系统功能必须被分解, 并分配到系统的一个组件。建模工具用来在模型构建时增强约束的一致性和完整性。

基于模型的方法保证规格要求、设计、分析和验证过程的严谨性, 也明显增强了需求可追溯性的质量和时间, 建模质量超过了基于文档的方法。MBSE 解决基于文档的方法的许多限制, 它通过提供一个更严格的方法来捕捉和集成系统需求、设计、分析和验证信息, 并促进这些信息在整个系统生命周期的维护、评估和通信。MBSE 的一些潜在的优点如下。

1) 提高沟通效率

在整个开发团队和其他利益相关者之间, 共享对于系统的理解, 具有从多个角度集成系统视图的能力。

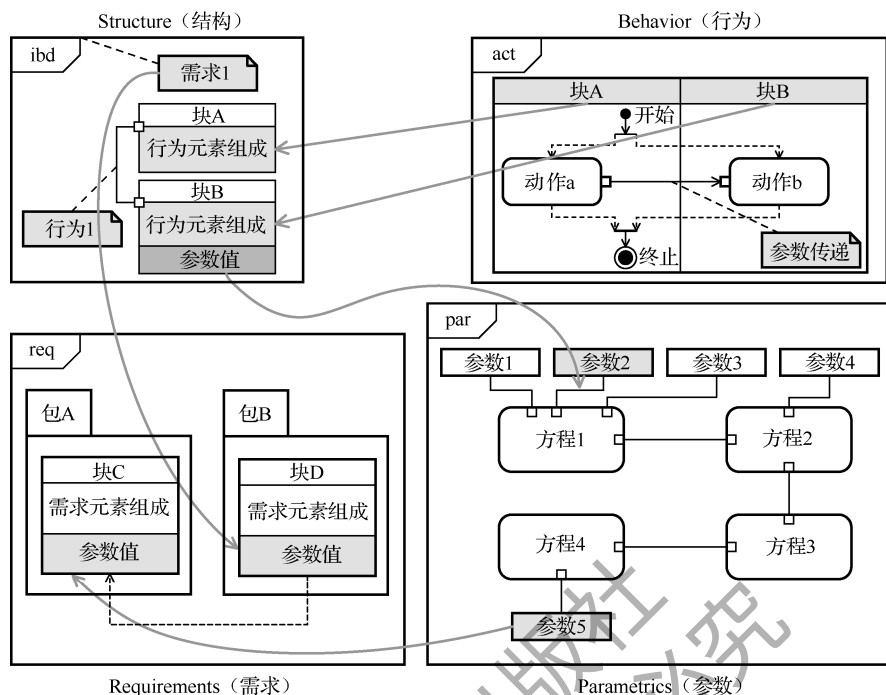


图 3-3 SysML 中系统模型的表示示例

2) 降低开发风险

随时进行需求验证和设计验证，针对系统开发有更准确的成本预估能力。

3) 提高质量

有更完整、无歧义、可验证的需求、设计、分析和描述，保证设计活动和测试活动之间更严格的可追溯性和设计完整性。

4) 增加可生产性

实现需求和设计变更的快速影响，重用已经存在的模型，支持设计演进，降低集成和测试过程的错误，减少产品迭代开发时间。

5) 自动文档生成

在整个生命周期中使用模型，可以自动生成各种设计文档，支持系统的诊断和维护。

6) 增强知识利用

对已经存在的设计和历史信息进行有效捕捉，保证信息、知识的有效存取和修改。当使用恰当的方法和工具时，在设计过程中，MBSE 可以提供附加的严格性规范。然而，这种严格不是没有代价的。明显地，转变到 MBSE 强调了在过程、方法、工具和培训上的前期投资。在这种转变过程中，MBSE 将结合基于文档的方法有效实施。例如，一个大型的、复杂的历史系统的更新严重依赖于历史文档，并且仅有系统的部分可以被建模。一个好的模型满足预期的目标，一个好的设计满足用户需求，并保证质量与设计原则的一致性。一个好的模型提供可视化方法，来辅助设计团队标识问题，并评估设计质量。

3.2.2.2 模型和 MBSE 方法定义

模型是物理世界中一个或多个概念的一种表示，它通常描述一个感兴趣的领域。一个模型是感兴趣领域的一个抽象，不能包含建模实体的所有细节。模型可以通过数学和逻辑表示，也可以用更具体的物理原型表示。更抽象的表示方法可以是一个图形化标志的组合，例如，节点和圆弧构成一个图像或一个几何体，由文本构成的一种编程语言也可以描述模型。

在 SysML 中，系统模型表示类似一个建筑蓝图，其明确地说明一个将被实施的系统是一个系统

的图形表示。SysML 模型表示行为、结构、属性、约束和系统的需求。SysML 有一个语义基础，能明确出现在系统模型中的模型元素类型和它们之间的联系。构成系统模型的模型元素被存储在一个模型库中，并用图形表示。如果一个执行环境支持仿真，那么一个 SysML 模型也可以被仿真。

方法是一组相关的活动、技术和约定，通常形成一组工具集，用来实现一个或多个过程。MBSE 是一种方法，其执行系统工程过程的所有或部分环节，并生成一个系统模型。

系统建模必须明确建模目的，明确定义建模工作的期望结果、利益相关者，以及结果如何被使用。根据建模目的，确定模型范围，模型的广度、深度和真实度。这个范围应该综合考虑可用的计划、预算、技巧和相关资源。理解目的和范围是建模工作的基础。

下面介绍的概念，可以用来评估模型的有效性和模型衍生的质量。质量属性可以用来建立模型偏好。建模工具可以用来检查模型质量，例如，模型是否被规范表达？模型的范围是否充分地满足它的目标要求？正如先前描述的，假设目标已经被明确定义，模型的广度、深度和真实度已经确定，模型范围将影响支持建模工作的资源需要。

模型的广度：模型的广度必须充分满足目标要求。明确需要对系统的哪些部分建模，并且考虑模型的可扩展性。尤其对比较大的系统，一般来说不需要对整个系统的所有部分建模。如果对一个已存在的系统添加新的功能，可以仅对新功能部分进行建模。例如，在一个汽车设计中，如果新需求为使其具有燃油经济性和解决加速问题，那么可以聚焦在传动系关联的元素上，而不聚焦在制动和转向子系统上。

模型的深度：模型的深度也必须充分满足目标要求。包括确定系统设计的层次、一个概念设计或初步设计的迭代过程，模型可以解决一个高层级的设计。例如，在汽车设计的例子中，初步设计迭代可以将发动机作为黑盒层次来建模，而不需要对发动机进行详细建模。

模型的真实度：模型的真实度也必须满足目标要求。通过确定层级的详细程度来实现。例如，一个低级真实度的行为模型可以是一个活动图中的动作顺序。如果行为模型被执行，那么需要附加的模型细节，这个附加的细节可以是系统响应的底层结构和逻辑。一个低级的真实度模型可以仅包含逻辑接口描述，相反，一个高级的真实度模型需要对信息结构和通信协议进行建模。同时，时间粒度也是系统动态性能需要考虑的维度。

模型的完整性：模型的广度、深度和真实度等特性必须匹配模型定义的范围。完整性准则可以关联模型的其他质量属性（例如，是否正确应用了命名约定）和设计完整性准则（例如，模型元素是否都能方便追溯等）。

模型格式良好：一个格式良好的模型符合建模语言规则，模型约束就能方便地附加上去。例如，在 SysML 中，原则上不允许一个需求只包含一个系统组件，尽管允许在组件和需求之间定义约束关系。建模工具应该增强约束的可实施性，通过建模语言规则保证模型格式良好。

模型的一致性：在 SysML 中，一些规则用来确保模型一致性。例如，兼容性规则可以支持类型检查，以确定接口是否兼容或单位制在不同的属性上是否一致。通过不同的方法可以附加其他约束。例如，一种方法可以施加一种约束，它的逻辑组件可以被分配到硬件、软件或操作过程。这些约束可以使用对象约束语言（Object Constraint Language, OCL）来表达。增强约束能维护整个模型的一致性，但它不阻止设计的不一致性。一个简单的例子是，两个建模者将相同的组件命名为不同的名称，通过一个模型检查器，可以将其区分。

模型的可理解性：有许多因素会影响基于模型的建模方法和建模样式，并影响到模型的可理解性。一个增强可理解性的关键因素是模型的抽象。例如，当描述一辆车的功能时，可以描述一个顶层功能，如“驾驶汽车”或提供一个更详细的功能描述，如“打开点火开关，使齿轮转动，踩下油门踏板”等。一个可理解的模型应该包含多层级的抽象，表示不同层级的细节，并相互关联。使用分解、详细说明、分配、视图和其他建模方法来表示不同层级的抽象。另外一个可理解性的影响因素是关联信息在图上有自己的表示。通常，使用工具可将信息在图上的不必要功能被隐藏，而仅显示信息关联到图的目的。

建模约定和标准的连贯性：建模约定和标准的连贯性是确保整个模型一致和连贯的关键。命名约

定对应每种类型的模型元素、图名称和绘图内容等。命名约定可以包含语言的文体方面,比如,何时使用大写和小写,何时在名称中使用空格。建模约定和标准也应该考虑工具施加的约束,比如,字母、数字和特殊字符的使用限制。建议为每种图的类型建立一个模板,这样可以确保模型协调一致。

模型是自描述的:如果引用是一致的,那么在整个模型中,符号和描述可以帮助提供增值信息。这些信息可以包含设计决策的原理,为弱问题或解析困难的问题提供模型元素的附加文本描述。这使模型能长期维护,保证与其他人员进行交流时更有效。系统模型可能需要集成电子、机械、软件、测试和工程分析等多种模型,这就需要特定的方法、工具和建模语言。例如,使用 SysML 描述系统模型的传递信息,使用 UML 定义特定的方法、工具和交换标准。

基于模型的测量:测量数据的收集、分析和报告可以作为一种管理技术,贯穿模型开发的整个过程。通过测量数据,可评估模型设计的质量和过程,评估技术、成本、计划状态和风险,并支持正在进行项目的计划和控制。基于模型的测量可以提供有用的数据,可以衍生一个表示在 SysML 中的系统模型。随着时间的推移,通过对数据大量的观察,可进行趋势评估和分布统计。

设计的质量:测量用来度量一个基于模型的系统设计的质量。这些测量已经应用在以文档为中心的设计评测中,例如,评估需求满足、需求验证和技术性能。SysML 模型可以包含明确的联系,用来测量扩展需求的满足程度,模型的细分粒度可通过模型元素满足的特定需求来评价。需求的可追溯性依赖于从任务层级需求向下到组件层级需求的关联。其他 SysML 关系可以用一种相似的方式来测量,判断哪些需求被实现、验证,这个数据可以直接从模型或间接从一个需求管理工具中捕捉。SysML 模型可以包含关键属性,在整个设计过程中都被监测,典型的属性可以包含性能属性,比如,恢复时间、物理属性(如质量)和其他属性(如可靠性和成本)。这些属性可以通过技术性能测量(Technical Performance Measurement, TPM)方法来监控。SysML 模型也可以包含属性之间的参数关系,并说明它们如何影响一个设计决策的结果。

设计划分:设计划分用来度量设计的层级聚集度和耦合度。根据接口的数量或根据不同模型之间的依赖关系来测量模型的耦合度。聚集度的测量非常困难,但可以通过一个组件的扩展性来评测,若一个组件的功能执行,不需要或较少需要外部数据,则组件的聚集度好。

设计的过程和开发:基于模型的测量方法提供了评估设计过程完整性的准则。质量属性的参考模型是否完整取决于建模工作的定义范围。评估设计完整性是必要的,但是不充分的,还需要评价需求的满足性,从而测量设计质量,当然,也可以用来评估设计的完整性。其他测量包括用例场景或逻辑组件及物理组件的测试。从系统工程的角度看,系统设计完整性的评测标准是扩展的组件是否被明确描述,这个标准可以用来评测组件接口、行为和属性规范的完整性。还需要评测扩展到的那个组件是否已经被验证和集成到系统中,并满足系统扩展的需求。在系统模型中,测试用例和验证状态作为评估的一个基础,需要同时被定义。

完成设计和开发的预估工作:利用系统工程成本模型(Systems Engineering Cost Model, COSYSMO)对系统工程活动进行成本预估。这个模型包含规模和生产性参数,其中,规模预测除了模型元素数量,还包括模型元素之间的联系,如模型的需求数量、验证需求的数量、实现的用户案例、模块的活动数目、分析的数目等。MBSE 规模参数被集成到成本模型中,此参数与它们关联的复杂性有关。例如,一个用例的复杂性不但取决于交互参与者的数目,还应考虑重用的和已存在模型的修改,以及新生成模型的数目。随着时间的推移,需要收集和确认规模参数和生产性参数,建立统计学上有意义的数据集,并评估与成本的关系,来支持准确的成本估算。早期使用的 MBSE 案例,可以明确分析建模工作的规模参数,并对模型数据进行局部预估,从而提高复杂系统的评估能力。

3.3 SysML 软件及其应用

3.3.1 SysML 软件

在人工智能时代,任何一种设计方法,当发展到一定阶段时,都会形成软件工具。SysML 建模方法也不例外。EA(Enterprise Architect)是澳大利亚 Sparx Systems 公司的产品,它覆盖了系统开发的整

个周期,除包括开发模型的功能之外,还包括事务进程分析、使用案例需求、动态模型、组件和布局、系统管理、非功能需求、用户界面设计、测试和维护等功能。EA 为用户提供一个高性能的、直观的工作界面,联合 UML 最新规范,为桌面电脑工作人员、开发和应用团队打造先进的软件建模方案。该产品可以配备整个工作团队,包括分析人员、测试人员、项目经理、品质控制和部署人员等。

EA 将 UML 和衍生建模语言(如 BPMN 和 SysML)的强大功能与高性能的、直观的工作界面相结合,为整个开发团队带来了集成的高级工具集,其主要功能如下。

1) 业务仿真

使用动态模型仿真将模型带入现实世界;验证行为模型的正确性,更好地理解业务系统如何工作;采用触发器(如按下按钮、扳动开关或接收信息)来控制仿真的执行;使用 JavaScript 编写的 Guards 和 Effects 来管理仿真流程;使用断点来分析决策和改善业务成果等。

2) 端到端跟踪能力

EA 具备从需求分析、设计模型到实施和部署的全程跟踪能力;利用 EA 的关系矩阵(Relationship Matrix)和层级视图(Hierarchy View)等功能,可以在整个产品生命周期内对系统进行有效性验证、确认和直接影响分析;结合内建的任务和资源分配,项目经理及质量保证团队将获得他们需要的正确信息来帮助项目成功进行。

3) 建模、管理和跟踪需求

EA 的内建需求管理功能可用来定义有组织的层次需求模型,跟踪从系统需求到模型元素的实施,搜索和汇报需求,对拟议的更改需求进行影响分析。

4) 复杂性管理

管理复杂性的 EA 内建工具,包括创建策略层面概念模型和业务层面概念模型的图、特定域的文件和可重复使用的模型模式;用于跟踪和集成更改的基准线和版本管理;基于角色的安全管理可使各级人员各司其职。同时,EA 利用“所见即所得”形式的模板编辑器,提供了强大的报表生成工具。

EA 系统建模软件的体系结构如图 3-4 所示,包括领域建模、扩展标准和扩展接口、基于模型的开发三大部分。MBSE 将系统领域建模与基于模型的开发、建模、仿真、测试、管理等集成起来。

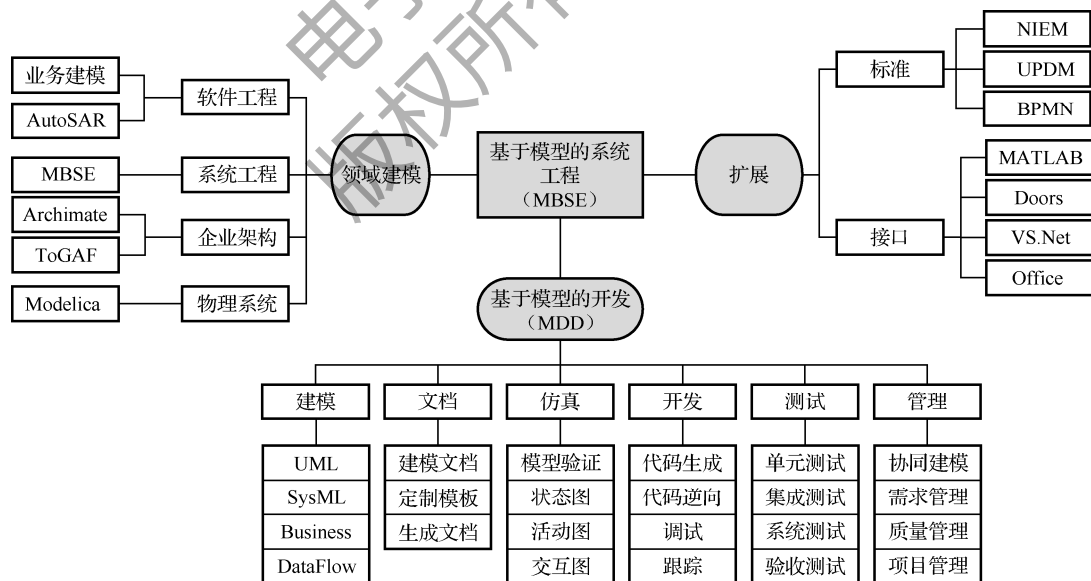


图 3-4 EA 系统建模软件的体系结构

EA 软件的主要功能如表 3-1 所示,包括基于模型的需求管理、需求条目管理、系统分析设计与建模、软件分析设计与建模、算法建模与仿真、系统运行框架建模、电子分析设计与建模、物理系统分析设计与建模、各种文档生成、软件开发、持续集成、工具总线等内容。