

基于特定域建模方法的航空发动机 MBSE 实施方法

郝彬彬¹, 鲁金直^{2*}, 李军¹, 王国新³, 兰小平⁴, 陈建江⁵, 吴新¹

1. 中国航发沈阳发动机研究所, 沈阳 110015
2. 瑞典皇家理工学院工业技术管理学院, 斯德哥尔摩 10044
3. 北京理工大学机械与车辆学院, 北京 100081
4. 北方车辆研究所, 北京 100072
5. 北京机电工程研究所, 北京 100854

摘要 提出一种基于模型系统工程(MBSE)的方法,将基于系统工程航空发动机系统的研发与基于模型的研发有效结合起来:采用系统思维方法对航空发动机需求、架构及验证进行形式化表达,并采用特定域建模方法与代码生成方法实现发动机验证模型的自动生成及验证。该方法通过元元模型对航空发动机的需求、架构及验证进行元模型建模,将用于描述航空发动机研发流程的特定域模型分为3个视图,并根据不同视图实现航空发动机研发过程的信息描述。通过开发代码生成器实现航空发动机验证模型自动生成的过程,实现航空发动机验证模型的自动验证和确认。以某型航空发动机为例,验证了所提方法的有效性。

关键词 航空发动机研发;基于模型的系统工程;特定域建模;代码生成;自动验证

近年来航空发动机设计技术正在不断进步,性能指标也在不断提高。此外,航空发动机系统本身也由简单到复杂,从单纯的机械液压控制发展到全权限数字电子控制(FADEC)^[1]。其控制系统的组成主要包括燃油系统、电气系统、控制器、健康管理系统等。随着航空发动机系统复杂性的增加,其研发过程与研发组织的复杂性也随之增长,严重影响到航空发动机研发的效率及系统鲁棒性。

基于模型的系统工程(MBSE)是目前工业界比较关注的热点领域,其通过形式化的建模方法,可以有效

支持复杂系统全生命周期的需求、设计、分析、验证和确认活动,目前已经成为学术界与工业界关注的热点技术,其具有如下特点:1) 基于系统工程方法执行基于模型的设计和研发;2) 采用形式化的建模方法支持全生命周期中的系统需求、设计、分析、验证和确认;3) 通过相关技术实现需求追溯、系统架构驱动、行为及性能表达、仿真及测试等。鉴于MBSE工程是针对复杂系统的良好解决方案,因此其是一个面向航空发动机控制系统这一复杂系统的有效研究技术路径。

传统的MBSE方法提倡采用通用建模语言,如Sys-

收稿日期:2019-01-01;修回日期:2019-02-28

作者简介:郝彬彬,高级工程师,研究方向为航空发动机控制系统设计,电子信箱:cdhbb525@126.com;鲁金直(通信作者),博士研究生,研究方向为基于模型的系统工程,电子信箱:jinzhl@kth.se

引用格式:郝彬彬,鲁金直,李军,等. 基于特定域建模方法的航空发动机 MBSE 实施方法[J]. 科技导报, 2019, 37(7): 88-95; doi: 10.3981/j.issn.1000-7857.2019.07.013

ML、UML等支持全生命周期需求、设计、分析等研发活动。然后,这些通用性语言一般基于一整套方法论,采用固定表达方式,导致研发人员需要适应和学习新的研发模式,将会延长项目周期,不利于对项目的风险管控。同时,针对这些通用语言,由于其本身的形式化限制,导致与目前研发流程和设计方法脱节,使相关MBSE技术落地困难,无法真实地应用到工程设计当中。

因此,为了解决通用建模语言存在过于形式化、学习周期较长导致MBSE落地相对困难的这一问题,本研究引入特定域建模方法,该方法通过M0~M3元模型建模,实现用户自定义其所在领域的元模型,进而形成特定域建模语言,实现通过研发体系模型化支持全生命周期的设计活动。

1 研究背景

1.1 特定域建模与集成建模

目前,主流通用建模语言UML^[2]和SysML^[3]被广泛应用于产品系统、经济性及计划的基于模型的形式化。与通用建模语言相比,特定域建模语言能够实现不同领域及工业标准的细化,例如AADL(architecture analysis & design language)^[4]是一个将SAE标准用于嵌入式系统软硬件架构描述的特定域语言;EAST-ADL^[5]是基于AutoSAR^[6](开放汽车系统架构标准)的针对汽车机电系统开发的架构描述语言。在领域工程师采用建模语言形式化其对应领域知识时,通用建模语言通常需要二次开发和领域知识细化,如设计相应领域模型的Profile文件等。与特定域建模语言相比,领域工程师需要花费更多的时间做相关建模工作。而建立的针对不同领域的模型库通常可以被看做是一种新的特定域建模语言。因此,系统工程师可以使用UML或者SysML的语义和语法设计自己领域的特定域建模语言^[7]。

1.2 支持MBSE的自动验证

NASA的喷气推进实验室(JPL)在实际的工程中采用MBSE方法研制航天系统,主要的工作包括扩展系统建模语言(SysML)而形成其特定域建模语言,进行宇宙飞机的设计及标准化,提出了相应的设计流程以及接口设计方法^[8],并与佐治亚理工学院开展了Modelica建模工具MapleSim与SysML建模工具MagicDraw之间进行模型转换的应用研究^[9]。与此同时,佐治亚理工学院与福特公司采用基于SysML的系统架构用于对汽车的

开发过程进行形式化架构。该方法首先从在系统建模语言SysML定义汽车参考架构模型,然后将汽车SysML架构模型自动转换成Modelica和Simulink模板进行相应的分析。

2 研究方法

2.1 系统思维

系统思维是以系统概念对事物的分析和理解的认识论,即事物(系统)是由多个子系统组成,不同子系统之间存在相关联系^[10]。通过结合系统思维与航空发动机研发过程,基于特定域建模的航空发动机研发方法如图1所示。

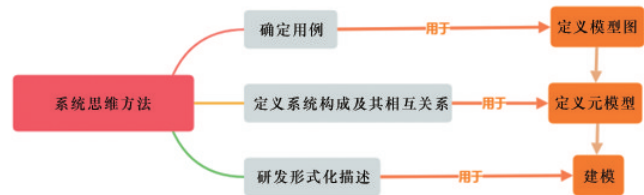


图1 系统思维方法支持特定域建模

Fig. 1 Systemic thinking supports DSM

1) 确认所需建模场景的用例。即模型所对应系统的边界。通过确认用例,决定所需建立特定域模型的图。

2) 定义图中的系统构成及相互关系。并应用其定义特定域模型的元模型。

3) 通过研发形式化的描述,采用特定域模型方法对其进行建模。

2.2 个案分析

以某型航空发动机为例,对特定域方法的可行性及有效性进行验证,验证指标见表1。

表1 验证指标

Table 1 Metrics and criteria

验证评估内容	验证指标	量度
特定域建模	架构描述	是否可以描述航发系统的架构
	验证描述	是否可以描述航发验证
	需求描述	是否可以描述航发需求
自动验证和确认	代码生成	是否可以代码生成
	自动验证	是否可以实现自动验证

3 特定域建模

特定域建模(domain-specific modeling DSM),是一种用于设计和开发系统(如软件系统,硬件系统)的建模方法^[11]。在特定域建模过程中,基于用户自定义的元模型库(metamodels)建立特定域模型(domain-specific models),进而表达系统的各方面特性。这些自定义的元模型库随着时间的积累也逐渐形成了特定域建模语言(domain-specific modeling language, DSL)。特定域建模语言倾向于支持比通用建模语言更具体化的表达,因此在对特定系统的建模过程中可以减少对模型进行二次开发和添加底层细节的表达。

3.1 特定域建模的语义和语法

根据语言学定义,设计语言(或建立模型)需要提前设计该语言的语义和语法^[12]。如图2所示,模型形式化(特定域建模语言)包括语义及语法;语法包括具体语法及抽象语法;语义包括语义映射及语义定义域。

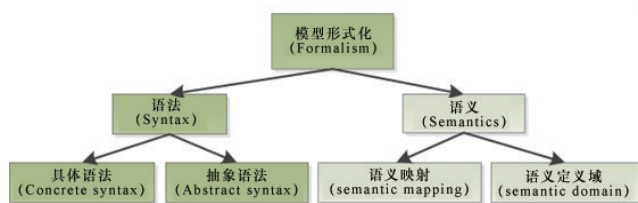


图2 领域建模语言术语
Fig. 2 Concepts in DSM models

具体语法用来显示特定域建模语言中的图形化表达或文本化表达,例如Modelica语言中的equation代表公式的含义(文本化表达),UML语言中class类的图标代表指定类(图形化表达)。

抽象语法定义了特定域建模语言的组成元素及它们之间所构成的关系。这些信息在模型的元模型(metamodel,模型的模型^[13])中表达(图3)。

语义定义模型在领域中的含义。其中包括语义定义域(什么含义)及语义映射(如何赋予模型含义),具体解释如下例。

例如,以图3中的1+2作为一个模型的例子。(1,+ ,2)可以被看成是文本式的具体语法。而1+2所代表的抽象语法是数字1与2之间可以添加“+”(模型的组成元素之间的描述规则)。这个例子中的语义定义域是自然数的运算,通过语义映射执行运算。因此,1+2的含义为数字的加法,即为3。

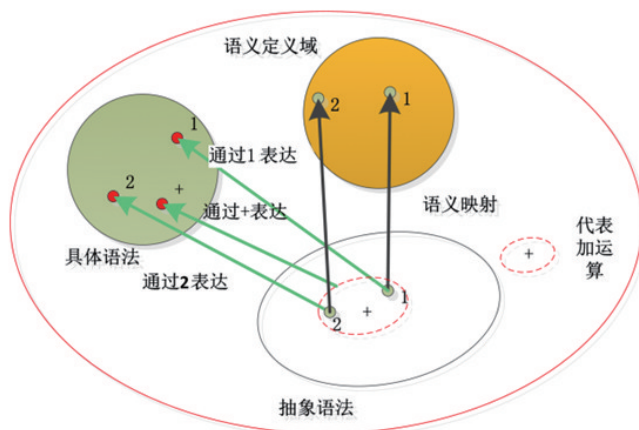


图3 术语含义
Fig. 3 Terminology

3.2 特定域建模的过程

在特定域建模过程中,一般遵循M₀~M₃的层次架构,如图4所示,其具体含义如下。

M₀,系统的视角^[14],即为一种特殊的协定模式用于建立、解释及确认视点表达某一个系统关切问题。视点是从某一个系统关切问题出发,采用一系列结构式及模块化的表达方法用于系统架构如何处理一个或者多个系统所关切问题。例如,某一个传感器组件的可靠性就是一个典型的视点。每个视角对应一个视点,是一系列形式化方法、模板及约定的集合用于描述该视点。每个视角包括:1)从项目或者组织中所认定或补充的信息;2)对相关视点的建立、解释和确认;3)从对应视点出发,以整个系统作为目标,用来确认该视点的所有关系问题的架构模型;4)包含可以用来确认对应视点的所有已知的条目。例如,采用Modelica进行仿真过程中,对应M₀的视角为对应模型需要验证和确认的系统问题。

M₁,所建立的特定域模型,用于描述其系统特性。

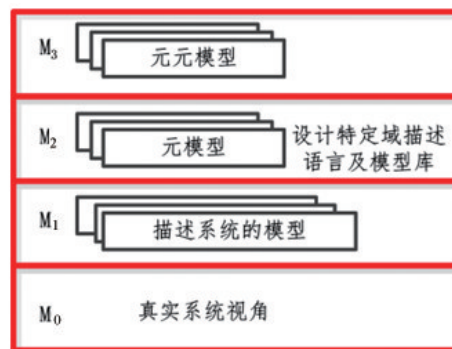


图4 M₀~M₃的特定域建模过程

Fig. 4 M₀~M₃ modeling framework

M_2 , 元模型, 即模型库。设计完成的模型库可积累成为一种特定域建模语言来描述对应领域问题。

M_3 , 元元模型, 用于设计元模型。一般为最基础的类, 将元元模型实例化后进而实现元模型的设计。

3.3 特定域建模实现自动验证

特定域模型在产品的设计过程中可以用于描述系统的不同视角, 例如研发体系、产品架构、仿真模型架构等。在产品研发过程中, 也可以与验证模型结合实现研发与验证的整合。以图5中的架构模型为例, 基于图、对象、点、关系、角色、属性(GOPPRR)6种元元模型, 设计领域的元模型库(如图5中的组件库)。通过其组件库, 对系统的架构模型进行建模(图5的架构模型), 进而表达对应系统架构。随后, 以研发流程为主线, 特定域模型与验证模型进行集成, 实现自动验证。

在研发流程的架构设计过程中, 采用特定域建模

对系统的物理架构进行建模: 首先, 将对应元元模型进行实例化, 建立元模型; 其次, 根据对应元模型建立物理架构模型用于表达其物理系统架构。

在研发流程的验证确认过程中, 采用不同的语言对该系统的行为或者组件进行面向对象的建模: 例如, 根据 Modelica 规范, 首先, 建立对应 Modelica 模型库; 然后, 用其模型库, 建立其对应的验证模型; 再次, 根据需求对该验证模型参数配置的描述, 对该验证模型进行配置; 最后, 通过该模型的仿真实现对该物理架构的验证和确认。

模型转化指将研发流程过程中的架构模型自动生成验证和确认模型的过程。以图5为例, 架构模型通过指定的代码生成器自动生成对应的验证模型(Modelica 模型)。通常, 在模型转化过程中, 一般采用的是元模型层次的组件映射。

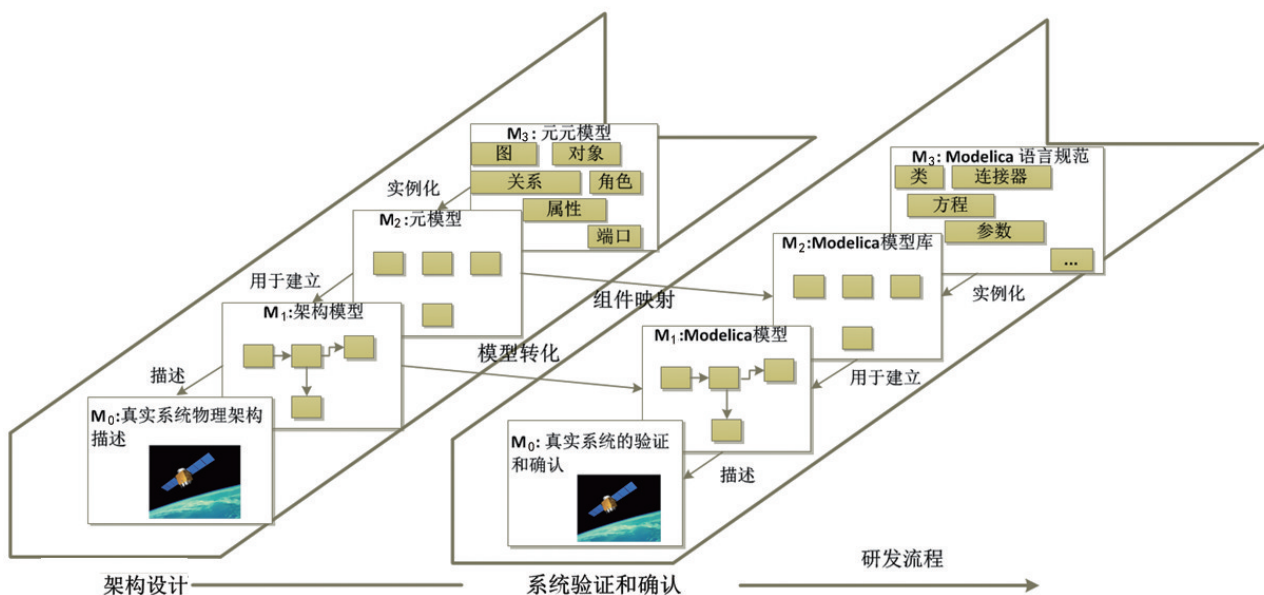


图5 特定域模型与验证和确认模型的集成

Fig. 5 Integration between DSM models and verification

4 应用案例

以某型航空发动机研发设计为例, 如图6和图7所示, 对其架构进行设计和开发, 图中所包括总图(需求表达图)、架构表达图及验证表达图, 各图中所设计的

元模型如表2所示。在总图中, 用于描述需求定义、架构定义及验证和确认之间的关系。验证图描述的是一个顺序的仿真过程, 每个仿真任务表示一个仿真过程, 对应的参数配置和求解配置与该仿真任务关联。

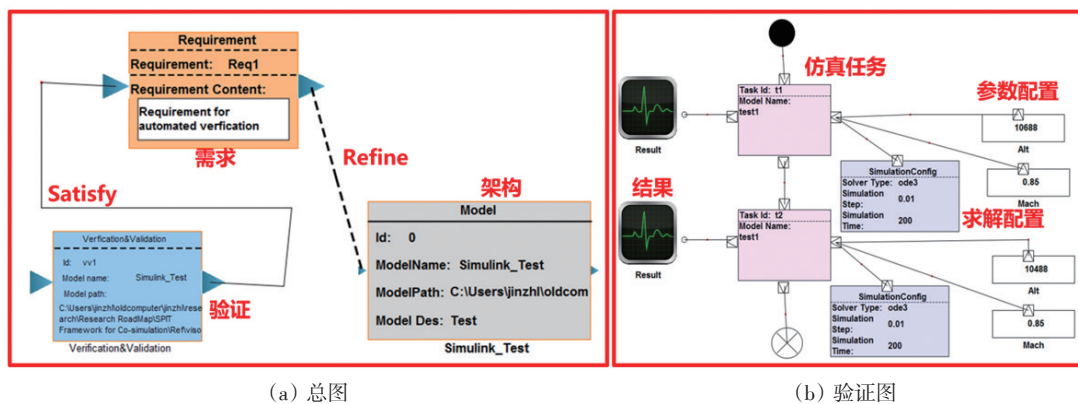


图6 航空发动机研发设计总图和构架

Fig. 6 General drawing and framework for aero-engine development and design

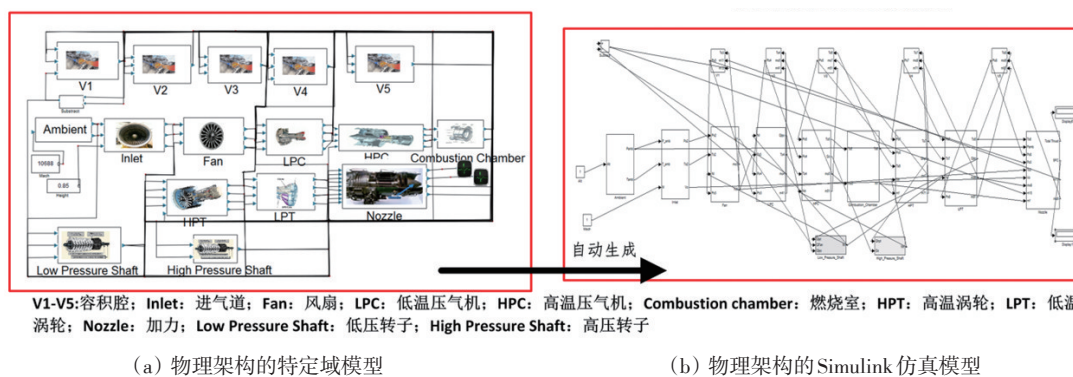


图7 航空发动机架构模型自动生成Simulink验证模型

图7 Mapping from architecture models to Simulink models

表2 元模型设计案例

Talbe 2 Meta-model design in case study

元元模型	元模型	描述
图	总体表达图(总图)	用于顶层描述,描述需求、架构与验证的关系
	架构表达图	用于描述架构
	验证表达图	用于描述验证
对象	举例,如Fan用于描述航发风扇组件	用于描述三个视图中的组件
关系	例如SysML中需求图的6种连接方式	用于描述对象之间的连接
角色	忽略	用于表达抽象语法,即不同对象与关系的连接方式
属性	忽略	图、对象及关系中所含的属性
端口	忽略	对象中与接口连接的端点

如图6所示,在特定域建模工具(MetaGraph, <http://www.zkhoneycomb.com/>)中,对航空发动机架构进行建模,该架构模型包括不同组件。如图7所示,(a)为基于特定建模的模型航空发动机物理架构模型,(b)为Simulink的航空发动机性能模型。在架构模型中,航空发动机控制系统的各级组件以元模型的形式建立。同时,组件之间的关系采用能量流的方式对不同组件进行连接。

本文研究了基于模型转化的自动生成和验证过

程,即代码生成技术,如图8所示。在特定域建模工具MetaGraph中,采用Karma语言实现对该代码生成器算法的表达。

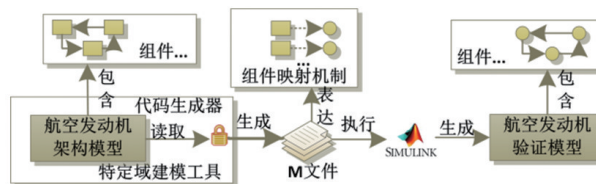


图8 模型自动生成过程

Fig. 8 Automated generation of Simulink models

通过基于 Karma 语言开发的代码生成器,读取该架构模型并生成对应 M 文件。该 M 文件通过生成 Simulink 模型所需要的所有操作执行代码,如 add_block、add_line 等表达对应的组件映射机制,即每个特定域模型的元模型到 Simulink 验证模型的组件映射规则。在特定域建模工具,通过 M 文件对 Simulink 进行操作来自动生成对应航空发动机的验证模型,其模型生成代码如算法 1 所示。

算法 1

1. Report .m file; %生成 M 文件
 2. Foreach architecture graph %遍历架构图
 3. If an object is found
 4. Write “add_block…” with object property;
 5. End %如果找到一个对象,根据对象属性,在 M 文件中写出 add_block 函数
 6. If an relationship is found
 7. Foreach ports with the relationship
 8. Obtain properties with such ports;
 9. Write “add_line…” with ports
 10. End
- %如果找到一个关系,遍历其输入输出端的属性,在 M 文件中写出 add_line 函数

同理,验证图生成对应 M 文件用于配置对应验证模型的参数和求解器,运行该文件,Simulink 自动仿真执行。

5 讨论

5.1 方法验证

通过对应用案例进行验证,验证指标及对应符合程度如表 3 所示,因此特定域建模方法可以满足目前基

表 3 验证指标

Table 3 Metrics and results

验证评估内容	验证指标	是否符合
特定域建模	架构描述	√
	验证描述	√
	需求描述	√
自动验证和确认	代码生成	√
	自动验证	√

于模型的航发的研制及形式化需求。

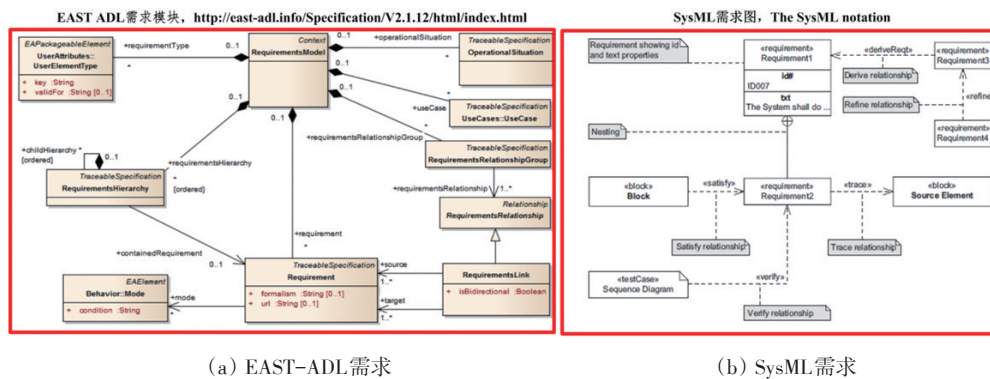
5.2 经验

通过特定域建模及验证模型自动生成的相关过程,优点如下。

- (1) 通过基于模型的表达方式增加航空发动机研发过程中形式化能力,进而可以实现整个研发体系的标准化。
- (2) 实现了研发流程中的研发信息可追溯及架构与验证和确认的单向可追溯功能。
- (3) 通过模型重用和自动生成验证模型提升了研发效率。

根据本团队在航空发动机控制系统 5 年的 MBSE 研究经验,对特定域建模方法和传统的基于 UML 或者 SysML 的通用建模语言进行比较,得到如下结果。

通用语言方法过于通用而特定域建模语言所表达的语义定义域更加具体。因此,通常,不同领域工程师可以根据 UML 与 SysML 的一些语义或语法设计他们所需的具有特殊功能的语言。如从 SysML 与 EAST-ADL 的需求图案例可以看出,工程师们根据 UML 与 SysML 语义语法和工业标准的相关要求设计自己的语义和语法,进而实现全生命周期或者部分研发流程的模型表达,语义比较如图 9 所示。



(a) EAST-ADL需求

(b) SysML需求

图9 EAST-ADL需求和SysML需求

Fig. 9 Meta models for requirement formalisms in EAST/ADL; B: Requirement diagram in SysML

目前部分支持通用建模语言的工具,过于形式化,不利于结合本单位或者本系统的研发、建模和开发。而特定域建模语言更加开放,在给定元元模型的基础上,可以根据工程师的需求进行开发或指定对应模型库。这样对推行体系与设计结合有很大好处。工程师不必拘泥于繁琐的形式化准则,从而更好地实现领域对接。

除了会使用对应工具设计元模型以外,设计人员还需要通过系统思维方法对系统工程核心进行理解,例如,如何将系统架构进行分解等实现系统中的各个实体相互之间关系的形式化描述及可视化表达。

5.3 展望

通过案例,定义的航空发动机的研发流程(需求定义、架构设计及系统验证)通过特定域建模技术实现形式化表达与自动验证。

根据案例中总结的经验,未来将针对航发控制系统的研发过程采用系统思维方法进行分析与定义,并逐级分析;其次,基于特定域建模技术对研发的不同视点进行元模型开发;再次,通过对应元模型建立模型实现对整个航发控制系统开发过程的描述。与此同时,开发对应代码生成器使架构模型与仿真验证模型实现单向打通。

6 结论

介绍了特定域建模的概念及其建模方法,研究了基于特定域建模的系统设计与验证确认相结合的方法,并应用于航空发动机案例。根据研究结果及团队使用MBSE在航空发动机控制系统实施的5年经验可以得到以下结论:特定域建模是一个比较新颖的MBSE方法,对于特殊领域或复杂系统的MBSE落地和实践具有良好的适用性。

参考文献(References)

- [1] Prencipe A. Technological competencies and product's evolutionary dynamics: A case study from the aero-engine industry [J]. *Research Policy*, 1997, 25: 1261-1276.
- [2] OMG. Software & systems process engineering meta-model specification[EB/OL]. (2008-04-01)[2018-05-10]. <http://depts.cnam.fr/Enseignement/CycleSpecialisation/MAI/UML1.4.pdf>.
- [3] Holt J, Perry S. SysML for systems engineering, 2nd Edition: A model-based approach[M]. London: The Institution of Engineering and Technology, 2008: 121-222.
- [4] Feiler P H, Gluch D P, Hudak J J. The architecture analysis and design language (AADL) [EB/OL]. (2006-02-01) [2018-05-10]. <http://www.aadl.info/aadl/currentsite/>.
- [5] EAST-ADL. EAST-ADL domain model specification[EB/OL]. (2013-11-28) [2018-05-10]. <http://www.east-adl.info/Specification.html>.
- [6] Cuenot P, Chen D, Gérard S, et al. Managing complexity of automotive electronics using the EAST-ADL[C]//12th IEEE International Conference on Engineering Complex Computer Systems (ICECCS 2007). Piscataway, NJ: IEEE, 2007: 353-358.
- [7] Selic B. A systematic approach to domain-specific language design using UML[C]//10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'07). Piscataway, NJ: IEEE, 2007: 2-9.
- [8] Cole B, Simmons J. An integrated systems modeling and analysis platform for flight project work[C]//AIAA SPACE 2016. California: AIAA, 2016: 1-11.
- [9] Schamai W, Fritzson P, Paredis C J J, et al. ModelicaML value bindings for automated model composition [C]//Proceedings of the 2012 Symposium on Theory of Modeling and Simulation-DEVS Integrative M&S Symposium. San Diego: Society for Computer Simulation International, 2012, 31: 1-8.
- [10] Lawson H W. A journey through a systems landscape[M]. London: College Publications, 2010: 1-20.
- [11] Kelly S, Tolvanen J P. Domain-specific modeling: Enabling full code generation[M]. Los Alamitos: Wiley-IEEE Computer Society Press, 2008: 1-30.
- [12] Mierlo S V, Tendeloo Y V, Meyers B, et al. The handbook of formal methods in human-computer interaction[C]//Cham: Springer, 2017: 435-463.
- [13] Lu J, Wang J, Chen D, et al. A service-oriented tool-chain for model-based systems engineering of aero-engines[J]. *IEEE Access*, 2018, 6(1): 50443-50458.
- [14] IEEE. International standard ISO/IEC/IEEE Systems and software engineering-Architecture description[EB/OL]. (2018-10-10) [2019-01-12]. <https://www.iso.org/standard/50508.html>.
- [15] Liu Z G, Yang M F, Chen Q, et al. An MBSE tool to support architecture design for spacecraft electrical power system[C]//INCOSE International Symposium 28. San Diego: INCOSE, 2018: 1-15.

Domain-specific modeling approach for model-based systems engineering in aero-engine development

HAO Binbin¹, LU Jinzhi^{2*}, LI Jun¹, WANG Guoxin³, LAN Xiaoping⁴, CHEN Jianjiang⁵, WU Xin¹

1. AECC Shenyang Engine Research Institute, Shenyang 110015, China
2. KTH Royal Institute of Technology, School of Industrial Engineering and Management, Stockholm 10044, Sweden
3. Beijing Institute of Technology, School of Mechanical Engineering, Beijing 100081, China
4. North Vehicle Research Institute, Beijing 100072, China
5. Beijing Research Institute of Mechanical & Electrical Technology, Beijing 100854, China

Abstract A model-based systems engineering (MBSE) approach is proposed to formalize aero-engine development with model-based design, which is developed based on a systemic thinking approach to support analysis of aero-engine systems, such as requirement, architecture, verification and validation. Moreover, a domain-specific modeling (DSM) approach is developed under an M_0 - M_3 modeling framework that includes meta-meta model, meta-model and model. The DSM approach is used to formalize views of products and implement automated verification and validation using code-generation. In this paper, a case study including four graphs is used to represent development information from different views based on GOPRRR meta-meta models. In addition, one code-generation algorithm is developed to support transformation from DSM models to Simulink models for verification. The result shows that the proposed aero-engine example indeed verifies the flexibility of the DSM approaches.

Keywords aero-engine development; MBSE; domain specific modeling; code generator; automated verification ●



(责任编辑 祝叶华)