

基于 SysML 的系统设计-仿真模型可视化转换

李新光, 刘继红*

(北京航空航天大学机械工程及自动化学院 北京 100191)
(ryukeiko@buaa.edu.cn)

摘要: 针对基于模型的系统工程中形式化描述的系统设计模型仿真能力不足的问题, 结合元模型的转换思想, 提出一种基于 SysML 的可视化模型转换方法, 将系统设计模型转换为可执行的仿真模型. 根据模型转换特点, 抽象定义基于 SysML 语言及其扩展机制的转换规则和转换活动等 2 种转换元模型. 在构建设计-仿真元模型映射关系的基础上, 通过转换元模型的实例化对系统设计-仿真模型实现可视化转换, 以降低模型转换的复杂性. 最后以简单机械系统为例, 验证了该方法的可行性.

关键词: 基于模型的系统工程; 设计模型; 仿真模型; 可视化转换
中图法分类号: TP391

A Method of SysML-based Visual Transformation of System Design-Simulation Models

Li Xinguang and Liu Jihong*

(School of Mechanical Engineering & Automation, Beihang University, Beijing 100191)

Abstract: Aiming at the problem of weak simulation capability of the system design models described formally in model-based system engineering, by applying the meta-model transformation method, this paper presents a visual method of transforming system design models to executable simulation model based on SysML. According to the characteristics of model transformation, two transformation meta-models, transformation rules and transformation activities, are defined abstractly based on SysML and its extension mechanism. The mapping relationships between design meta-model and simulation meta-model are established firstly, and then the visual transformation of the system design-simulation models is achieved by instantiating the transformation meta-model to reduce the complexity of the model transformation. Finally, a case of simple mechanical system is used to verify the feasibility of the proposed method.

Key words: model-based system engineering; design model; simulation model; visual transformation

近年来, 复杂产品系统呈现出功能高度集成、各学科复杂耦合、设计信息庞杂等发展趋势, 导致其研发过程面临新的问题: 需求更新困难、系统层设计验证不足、理解不一致性等. 因此, 基于模型的系统工程(model-based systems engineering, MBSE)

作为问题的有效解决方案被提出^[1-2].

MBSE 以模型代替文档, 采用标准系统建模语言(system modeling language, SysML)构建系统设计模型并作为可追踪的信息集成框架, 从概念设计的需求阶段开始通过该模型的不断演化、迭代

收稿日期: 2016-05-10; 修回日期: 2016-08-11. 基金项目: 国家科技支撑计划(2015BAF17B01). 李新光(1990—), 男, 硕士研究生, 主要研究方向为基于模型的系统工程; 刘继红(1966—), 男, 博士, 教授, 博士生导师, CCF 会员, 论文通讯作者, 主要研究方向为复杂产品数字化设计与制造、现代设计理论与方法、基于模型的系统工程.

支持系统设计、分析、验证和确认等活动,并贯穿整个研发过程和系统的后续生命周期阶段^[3-4]。这种新的系统研发模式具有以下优势:集成表达产品研发早期需求、功能、结构和行为 4 方面的信息,为研发过程的需求管理和追踪提供便利;通过持续动态仿真环节验证发现概念设计方案存在的不合理之处并及时修正;系统设计模型具有唯一性,为研发的不同参与者提供了直观的、统一的、无二义性的设计信息交流平台^[5]。

由于 MBSE 中的系统设计模型无法直接执行仿真验证,因此需要将其与系统仿真模型(Modelica 或 Simulink 模型等)进行集成转换以弥补不足。Johnson 等^[6]基于 SysML 扩展机制建立与 Modelica 等价的模型元素,采用基于 TGG(triple graph grammars)的转换方法建立 2 种语言元模型的映射关系,实现自动转换。Schamai 等^[7]通过扩展 SysML 并定义新的方程图和仿真图以构建可执行的系统仿真模型。Cao 等^[8]以 Simulink/MapleSim 作为目标仿真平台,通过建立面向不同平台的仿真补充模型,将 SysML 系统混合行为模型自动转换到 Simulink/MapleSim 中进行系统行为仿真。

上述研究能够实现特定系统领域内设计模型向仿真模型的转换过程,通过仿真验证系统的性能需求,但存在平台依赖性强,转换过程复杂,适用性低等方面的不足。本文在现有研究基础上,借鉴系统设计-仿真模型转换的一般过程,抽象出其中的常用概念,提出一种基于 SysML 的可视化模型转换方法;结合元模型思想,抽象定义基于 SysML 语言及其扩展的转换规则和转换活动 2 种转换元模型;根据系统设计与仿真建模语言(SysML 与 Modelica)的元模型映射关系对照表,通过将转换元模型实例化实现系统设计-仿真模型的可视化转换。与现有的研究成果相比,本文方法采用与系统设计建模过程相一致的可视化图形建模语言 SysML 对转换规则与活动进行表达,达到建模与转换集成在统一环境中的目标。最后以简单机械系统为例验证了该方法的可行性。

1 系统设计-仿真模型转换

1.1 系统建模语言

SysML 作为一种图形化系统设计建模语言,以其具有明显优势的描述能力对系统需求、功能、结构和行为 4 方面信息进行表达^[9],为复杂系统设

计建模提供支持。目前, SysML 已被 NASA、欧洲航天局等机构广泛应用到实际项目研发中^[10-11]。

本文以系统仿真建模语言 Modelica 为例,说明方法应用过程。Modelica 采用面向对象思想和基于方程的表达形式^[12],为描述多领域的系统仿真模型提供了统一建模表达机制,通过扩展模型库已广泛应用在汽车、航空等多个领域系统的建模仿真中^[13-14]。

1.2 系统设计-仿真模型转换过程

系统设计-仿真模型的转换集成能够在系统设计过程中充分结合 SysML 与 Modelica2 种模型的优点,将仿真验证环节引入到系统设计的早期阶段^[15]。

尽管现有研究大多针对不同的仿真平台或语言提出各自的方法将 SysML 系统设计模型转换为各类仿真模型进行验证,但它们采用的转换过程具有一定的相似性,其中, Paredis 等^[16]对这类方法的一般思路进行了提炼:1) 分析特定仿真工具及相应仿真语言的特点,采用元模型表达;2) 对应仿真元模型中的每个模型元素,采用 SysML 扩展机制建立其等价模型;3) 在仿真元模型与 SysML 扩展模型之间建立转换关系,并采用特定的转换方法对相应转换规则进行表达;4) 通过仿真平台 API 实现源模型与目标模型的动态关联,使转换过程自动化。目前,系统设计-仿真的模型转换技术正在发展中,尚未形成统一的规范标准,仍存在着无法完全自动转换的挑战^[17]。

1.3 可视化模型转换

随着转换中涉及模型的结构复杂性不断提高,导致对转换的需求和目标增加。为了控制转换的复杂性同时提高复杂转换的效率和质量,将模型转换视为一种特殊的建模过程,因此利用可视化图形建模语言(如 SysML 等)实现对转换模型的构建,可以很好地支持模型转换过程的定义与执行,并有利于提高转换的效率和转换规则的重用性。同时,这种可视化转换过程具有直观性的特点,可以一定程度上降低转换的复杂性。

2 基于 SysML 的可视化模型转换方法

基于元模型映射的思想,将设计-仿真 2 种异构模型转换为同构模型进行集成。在上述通用转换思路的基础上,本文采用统一的元模型描述方式对模型转换进行抽象化表达,基于转换规则和转换活动 2 种元模型实现的具体转换过程可以将

SysML 模型转换为 XML 格式的 Modelica 模型。

本文方法的框架如图 1 所示, 重点在于定义元模型级别的转换规则与活动, 并遵循上述定义实现具体模型的转换过程。具体包括 5 个步骤:

Step1. 抽象 Modelica 元模型, 并采用元模型建模语言 MOF(Meta Object Facility)表达, 实现与 SysML 元模型的同构化, 为与 SysML 建模元素建立相应的映射关系提供基础。

Step2. 分析 SysML 系统设计模型与 Modelica 仿真模型, 建立两者具有相同语义功能的元模型元素之间的映射关系。

Step3. 根据定义的模型转换元模型, 对上述映射关系进行表达。

Step4. 利用 SysML 活动图对转换活动的执行过程及顺序进行定义。

Step5. 采用上述定义的转换规则和转换活动两类元模型并结合转换方法对具体模型的转换过程进行表达, 最后执行转换活动, 得到输出结果。

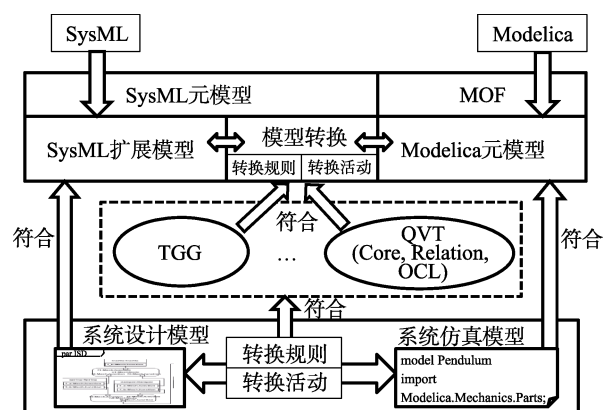


图 1 本文方法框架

本文方法涉及的 SysML 到 Modelica 模型转换实质上是一种重构转换, 是依据同一抽象层的 2 种模型之间的映射关系, 定义相应的映射转换规则, 通过规则的执行从而按照目标模型结构重新组织模型信息, 即实现所谓的“转换”。对于简单的映射关系可以直接定义转换规则进行表达, 当映射关系变得复杂化时, 需要满足一定的约束条件才能实现转换时, 可以通过一组有序的建模活动来进行控制转换的过程, 这也是所提转换活动模型的重要作用。

2.1 Modelica 元模型

Modelica 语言缺乏标准的抽象语法或元模型, 为了建立 SysML 与 Modelica 建模元素之间的映射关系, 需要对 Modelica 语言进行分析, 抽象出其元模型, 采用 MOF 表达, 基于 Modelica 语言的模

型可看作所建元模型的实例。

图 2 所示为 Modelica 元模型的简图, 元模型中分别用 ModelicaModel 和 ModelicaConnector 表示模型和连接器的概念, 最基本的模型包括 Behaviour, Component 和 Function。Behaviour 指模型中的 equation 部分; Component 指模型的组成属性, 包括子模型和连接器等; Function 指模型内部或由外部导入的算法代码。

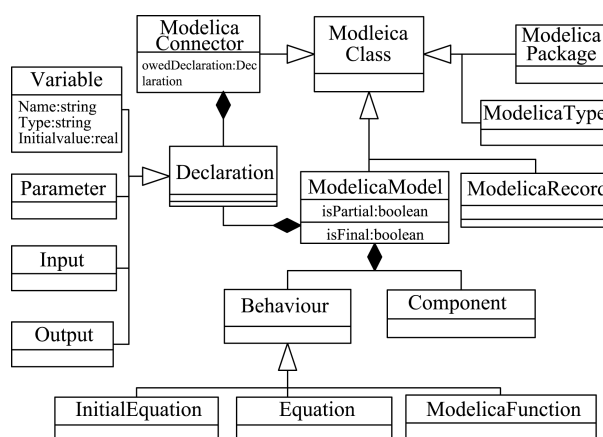


图 2 Modelica 元模型简图

2.2 SysML-Modelica 元模型映射关系

通过对 2 种建模语言的元模型进行分析, 建立 SysML-Modelica 中具有相同语义和功能的元模型间的映射关系, 为转换规则的定义提供基础。表 1 所示为部分映射关系, 还有部分无法实现一一映射的模型元素, 需利用 SysML 本身的扩展机制进行对应 Modelica 语义的表达, 如扩展定义 variable 以对应 Modelica 中的变量单位类型 unit。

表 1 SysML-Modelica 元模型映射关系(部分)

序号	SysML	Modelica
1	Package	Package
2	Block	Model
3	Constraints	Equation
4	Values	Variable
5	Flow Specification	connector
6	Connector	Connection

2.3 转换规则元模型

为了实现模型转换, 本文从实际模型转换活动中总结抽象得到一组建模元素, 通过定义转换规则元模型(如图 3 所示)支持开发人员对模型转换中模型间的映射关系进行刻画。

图 3 中 Transformation 本身是一个抽象转换类,

其子类是原子转换(AutomicTransformation), 是不可再分的转换表达元素, 多个原子转换组成一个转换任务; 每个原子转换包括一组转换规则(TransformationRule), 每条转换规则具有一个修饰符属性(Modifier), 用来标识规则的类型信息. 在不同的转换描述语言中规则具有不同的类型, 具体的规则类型则通过规则分类器的子类实现. 除了类型修饰符属性, 每条规则由若干区域(Domain)构成, 区域是一种特殊的模型类型, 代表着不同转换语言中对于规则定义的组成部分, 每个区域也有标识所属类型的修饰符.

2.4 转换活动元模型

模型转换过程可看作多个转换规则的顺序执行, 因此可以将这个过程通过一组有序的转换活动来实现. 本文采用 SysML 的活动图来完成模型转换过程的约束表达, 其元模型如图 4 所示.

图 4 中, Activity 表示模型转换的活动, 包括活动节点(ActivityNode)和边(Edge). 活动节点可以实例化为动作节点(ActionNode)、对象节点(ObjectNode)和控制节点(ControlNode). 动作节点子类 Transformation Action 表示所要执行的转换动作, 与包含相应转换规则的 Transformation 相关联;

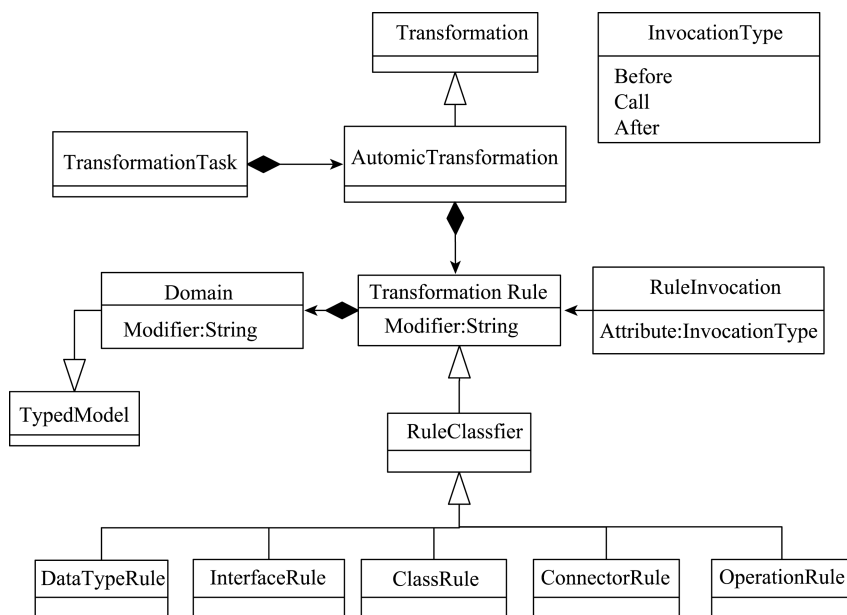


图 3 转换规则元模型

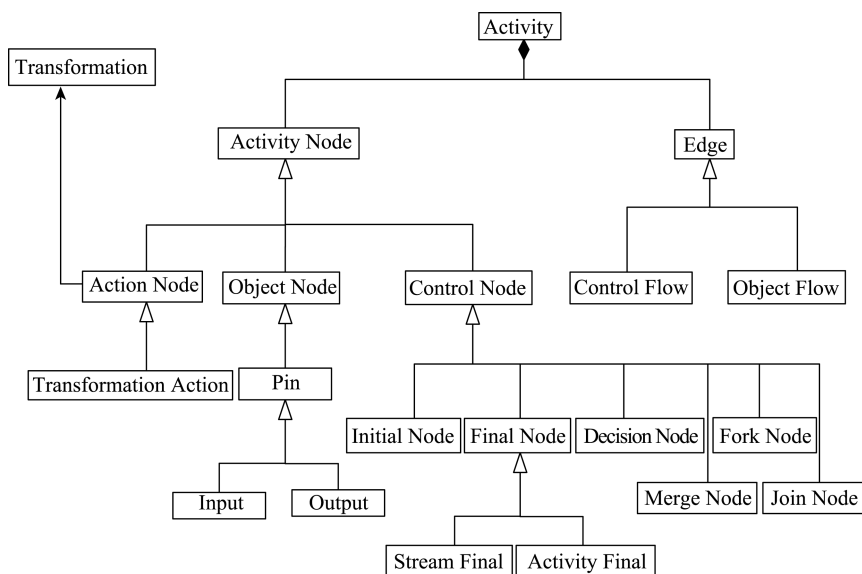


图 4 转换活动元模型

对象节点具体表现为 Pin, 即转换动作的输入、输出; 控制节点可以细化为 InitialNode, FinalNode (Stream-Final 和 ActivityFinal), DecisionNode, MergeNode, ForkNode, JoinNode 等 7 种类型。活动图中的边 (Edge)用来表示活动的执行序列, 分为 ObjectFlow 和 ControlFlow。

通过对上述转换活动元模型的实例化, 建立包含多个转换规则的不同转换活动之间的约束关系, 并最终实现所有转换活动的执行过程。以目前常用的模型转换技术 TGG 为例, 图 5 所示为符合

TGG 规范的方程转换规则, 采用本文方法通过转换活动图表达所包含转换规则的实现过程。对于继承关系的转换, 按照先父后子的顺序进行转换, 如图 6 所示。将 SysML 模型的约束属性转换为表示 Modelica 模型行为的方程, 总体需要 2 步: 1) 保证完成父类模块到父类组件的转换, 这是进行子类约束与子类方程之间转换的前提; 2) 子类转换活动又可进一步分解为更详细的转换活动, 此例中包括是否为初始方程的条件判断和方程参数转换 2 部分。依此类推, 任何复杂转换活动都可进行

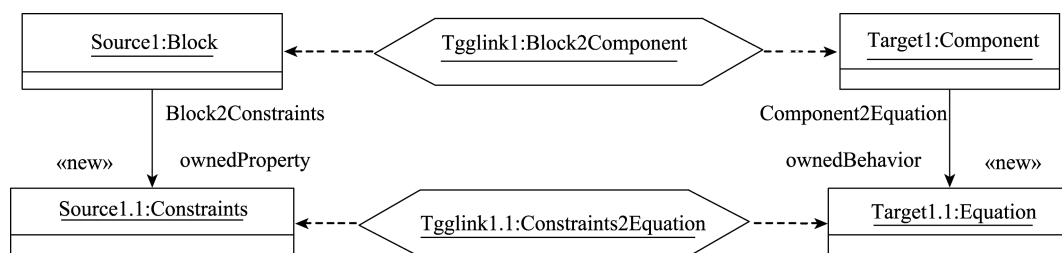


图 5 方程的 TGG 转换规则

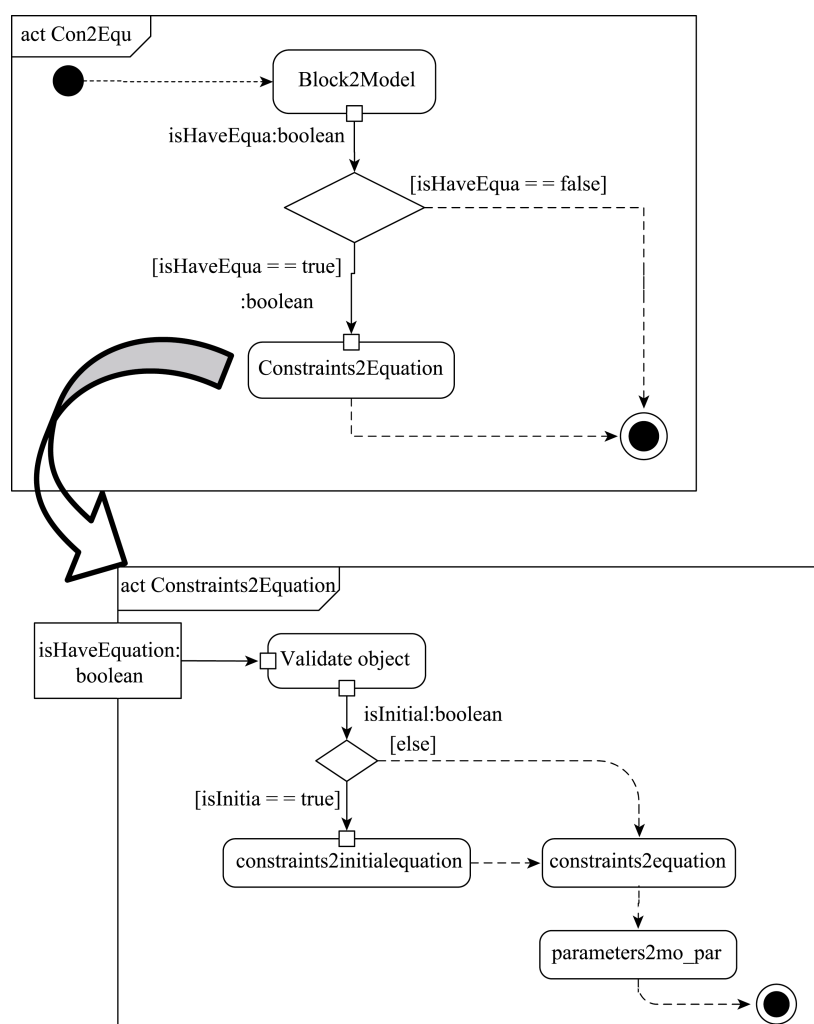


图 6 活动图定义的方程转换规则

分解,直到能够直接执行的转换动作为止.

2.5 转换过程

利用转换规则可以对转换内容实现静态描述,转换活动图可以定义转换的动态行为,以此来表达转换规则的执行顺序,即转换过程.转换动作用于执行与之相关联的转换规则,每个动作包含一个可执行引擎,判断是否满足执行条件并将执行后的数据流或控制流传递给下一个关联动作节点,保证转换过程执行的连续性,直到出现最终活动控制节点可结束整个转换活动.

3 实例验证

对带有双质量元素的惯性-弹簧-阻尼器(inertia-spring-damper, ISD)一维机械转动系统(如图 7 所示)

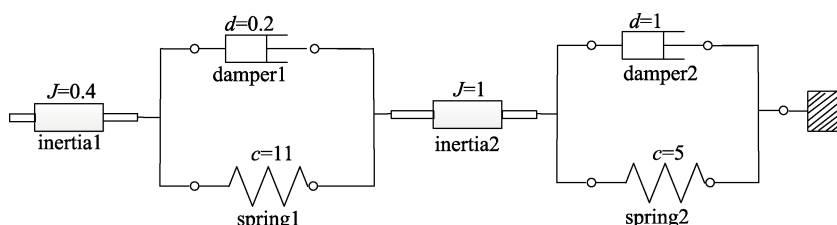


图 7 ISD 系统示意图

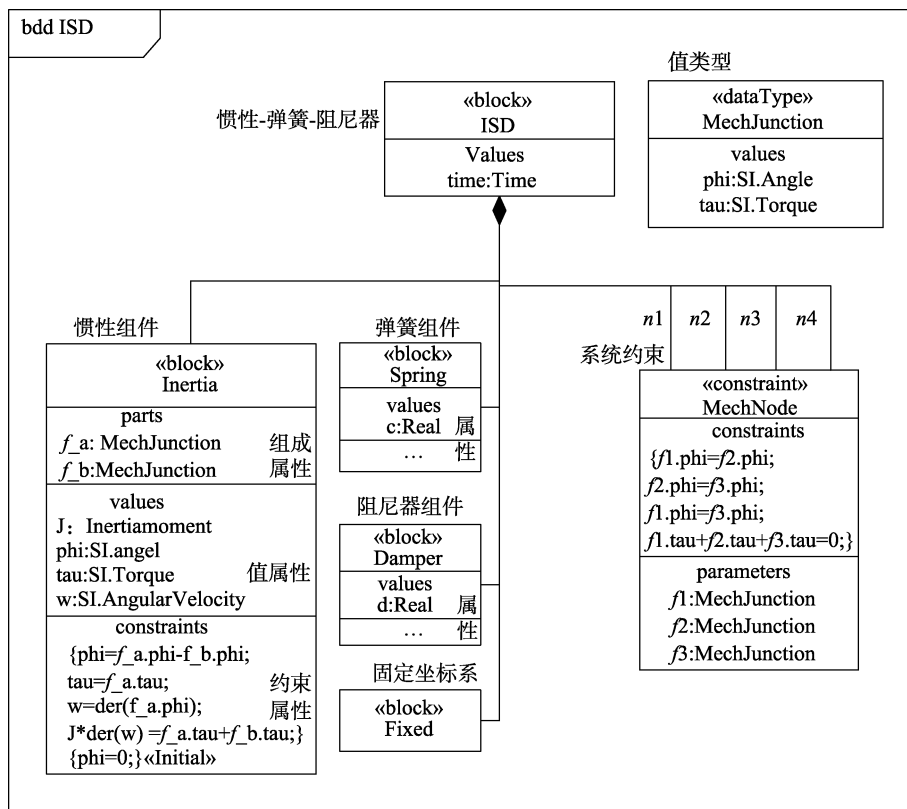


图 8 ISD 系统模块定义图

进行 SysML 建模,并通过所提可视化模型转换方法得到 XML 格式的 Modelica 模型,用于本文方法的可行性验证.本文选择 Eclipse Papyrus 作为 SysML 建模环境和 OpenModelica 作为系统仿真环境.

首先对该转动系统进行 SysML 建模,并能够完整地描述系统行为模型.图 8 所示为系统结构的模块定义图,该系统包括惯性组件、弹簧、阻尼器以及坐标系 4 种类型的组成部分和系统本身的值属性、约束属性.其中,惯性组件、弹簧和阻尼器两端各有一个法兰作为组件的连接,法兰中的变量满足不同组件的内部物理规律并作为组件的约束属性,每个类型的组件又具有属于自己的值属性,如惯性组件具有的转动惯量、转动角度、扭矩和角速度.根据图 8 所示系统组件间的连接关系以及系统遵循的物理规律构建表达系统行为的

参数图, 如图 9 所示。

利用前面定义的 SysML-Modelica 转换规则和转换活动元模型, 对 ISD 系统设计-仿真模型转换的规则和过程进行建模, 执行该过程输出 XML 格式的 Modelica 模型文件。

如图 10 所示, 采用转换规则模型建立与系统模型对应的静态转换模型, 并通过活动图对其进行详细说明, 从而构建不同层次的动态转换模型, 如图 11 所示。最后执行活动图中所定义的转换动作, 得到 XML 格式的系统 Modelica 模型。

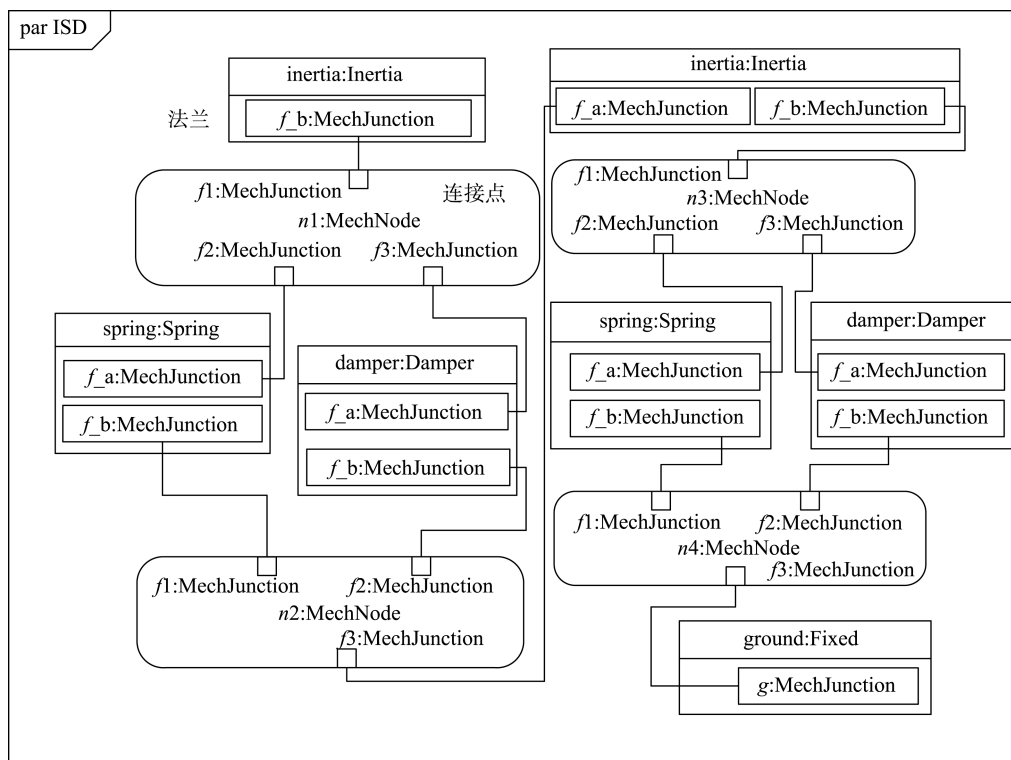


图 9 ISD 系统参数图

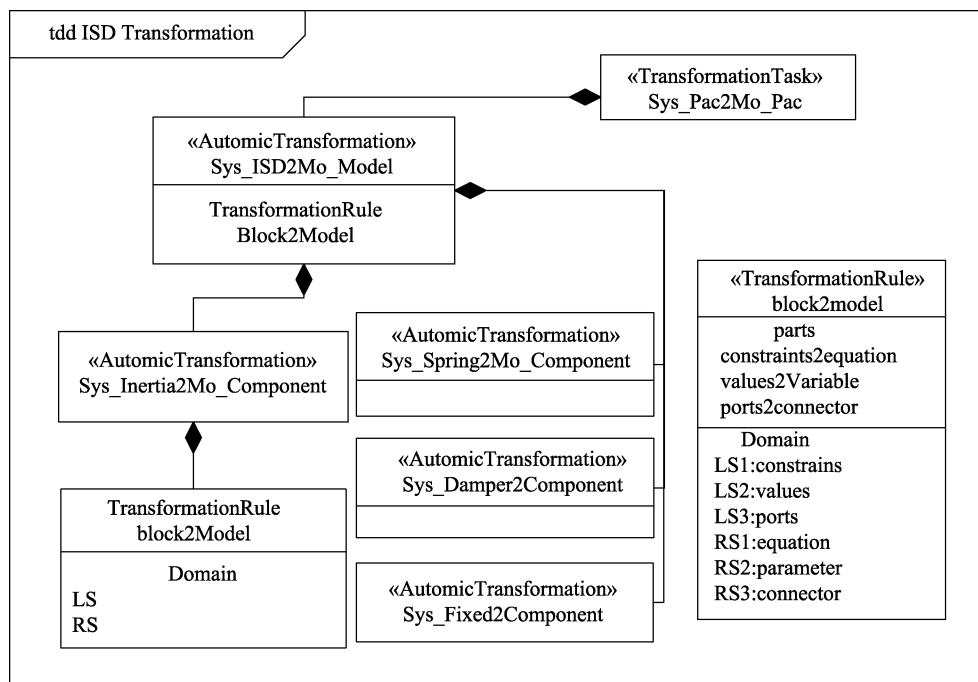


图 10 ISD 系统转换规则静态描述

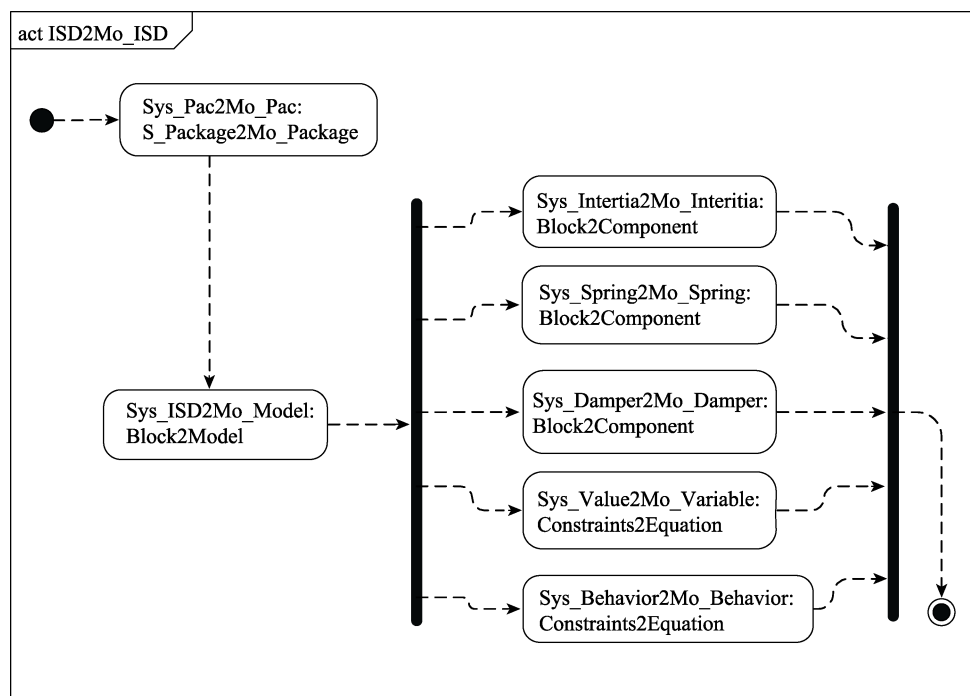


图 11 ISD 系统转换活动图

4 结 语

在复杂产品系统研发模式由基于文档向 MBSE 转换的发展背景下, 本文分析了当前 MBSE 缺乏支持仿真能力的不足. 为了实现系统设计模型向仿真模型的转换, 本文在基于元模型体系的模型转换思路基础上, 提出一种基于 SysML 的系统设计-仿真模型可视化转换方法, 重点包括基于元模型的转换规则定义和转换活动定义. 针对具体的 SysML 系统设计模型, 通过对转换规则与转换活动元模型的实例化层次建模确定 SysML 源模型与 Modelica 目标模型之间的转换关系, 执行动态转换活动实现转换. 最后, 以 ISD 系统作为实例进行该方法的可行性验证. 下一步的工作重点是对转换规则和活动的自动化建模进行研究.

参考文献(References):

- [1] Buede D M. The engineering design of systems: models and methods[M]. 2nd ed. New York: John Wiley & Sons, 2009
- [2] Estefan J A. Survey of model-based systems engineering (MBSE) methodologies[R]. Seattle: INCOSE MBSE Focus Group, 2007
- [3] Han Fengyu, Lin Yiming, Fan Haitao. Research and practice of model-based systems engineering in spacecraft development[J]. Spacecraft Engineering, 2014, 23(3): 119-125(in Chinese) (韩凤宇, 林益明, 范海涛. 基于模型的系统工程在航天器

研制中的研究与实践[J]. 航天器工程, 2014, 23(3): 119-125)

- [4] Friedenthal S, Griego R, Sampson M. INCOSE model based systems engineering (MBSE) initiative[OL]. [2016-05-10]. https://www.researchgate.net/publication/267687693_INCOSE_Model_Based_Systems_Engineering_MBSE_Initiative
- [5] Liu Yusheng, Yuan Wenqiang, Fan Hongri, et al. Research on information integration framework of SysML based model-driven design of complex products[J]. Chinese Journal of Mechanical Engineering, 2012, 23(12): 1438-1445(in Chinese) (刘玉生, 袁文强, 樊红日, 等. 基于 SysML 的模型驱动复杂产品设计的信息集成框架研究[J]. 中国机械工程, 2012, 23(12): 1438-1445)
- [6] Johnson T, Kerzhner A, Paredis C J J, et al. Integrating models and simulations of continuous dynamics into SysML[J]. Journal of Computing and Information Science in Engineering, 2012, 12(1): 011002
- [7] Schamai W, Fritzson P, Paredis C, et al. Towards unified system modeling and simulation with ModelicaML: modeling of executable behavior using graphical notations[C] //Proceedings of the 7th International Modelica Conference. Linköping: Linköping University Electronic Press, 2009: 612-621
- [8] Cao Y, Liu Y S, Fan H R, et al. SysML-based uniform behavior modeling and automated mapping of design and simulation model for complex mechatronics[J]. Computer-Aided Design, 2013, 45(3): 764-776
- [9] Jiang Caiyun, Wang Weiping, Li Qun. SysML: a new systems modeling language[J]. Journal of System Simulation, 2006, 18(6): 1483-1487(in Chinese) (蒋彩云, 王维平, 李 群. SysML: 一种新的系统建模语言[J]. 系统仿真学报, 2006, 18(6): 1483-1487)
- [10] McKelvin M, Jr., Jimenez A. Specification and design of electrical flight system architectures with SysML[OL]. [2016-05-10].

- <http://arc.aiaa.org/doi/pdf/10.2514/6.2012-2534>
- [11] de Lange D, Guo J, de Koning H P. Applicability of SysML to the early definition phase of space missions in a concurrent environment[M] //Complex Systems Design & Management. Heidelberg: Springer, 2012: 173-185
- [12] Wu Yizhong, Liu Min, Chen Liping. Development of hybrid modeling platform for multi-domain physical system[J]. Journal of Computer-Aided Design & Computer Graphics, 2006, 18(1): 120-124(in Chinese)
(吴义忠, 刘 敏, 陈立平. 多领域物理系统混合建模平台开发[J]. 计算机辅助设计与图形学学报, 2006, 18(1): 120-124)
- [13] Deuringa A, Gerl J, Wilhelm H. Multi-domain vehicle dynamics simulation in dymola[C] //Proceedings of the 8th International Modelica Conference. Linköping: Linköping University Electronic Press, 2011: 13-17
- [14] Dong Zheng, Zhang Hongchang, Ding Jianwan. Hardware-in-loop simulation of aircraft landing gear using Modelica language[J]. System Simulation Technology, 2015, 11(4): 277-282
(in Chinese)
(董 政, 张洪昌, 丁建完. 基于 Modelica 的起落架半物理仿真设计与实现[J]. 系统仿真技术, 2015, 11(4): 277-282)
- [15] Yan Xiqiang, Li Yan, Li Wenqiang, *et al.* A multidisciplinary information modeling method for complex products based on meta-model[J]. Journal of Computer-Aided Design & Computer Graphics, 2013, 25(10): 1540-1548(in Chinese)
(闫喜强, 李 彦, 李文强, 等. 元模型的复杂产品多学科信息建模方法[J]. 计算机辅助设计与图形学学报, 2013, 25(10): 1540-1548)
- [16] Paredis C J J, Johnson T. Using OMG's SysML to support simulation[C] //Proceedings of the 4th Conference on Winter Simulation. Los Alamitos: IEEE Computer Society Press, 2008: 2350-2352
- [17] Nikolaidou M, Kapos G D, Tsadimas A, *et al.* Simulating SysML models: overview and challenges[C] //Proceedings of the 10th System of Systems Engineering Conference (SoSE). Los Alamitos: IEEE Computer Society Press, 2015: 328-333