

## M-Design: 多域复杂机电产品系统建模平台\*

刘玉生<sup>+</sup>, 樊红日, 蒋浴芹, 曹悦

(CAD & CG 国家重点实验室(浙江大学), 浙江 杭州 310027)

### M-Design: System Modeling Platform for Multi-Domain Complex Mechatronics

LIU Yu-Sheng<sup>+</sup>, FAN Hong-Ri, JIANG Yu-Qin, CAO Yue

(State Key Laboratory of CAD & CG (Zhejiang University), Hangzhou 310027, China)

+ Corresponding author: E-mail: ysliu@cad.zju.edu.cn

Liu YS, Fan HR, Jiang YQ, Cao Y. M-Design: System modeling platform for multi-domain complex mechatronics. *Journal of Software*, 2012, 23(Suppl. (2)): 8-20 (in Chinese). <http://www.jos.org.cn/1000-9825/12020.htm>

**Abstract:** The system design has become an imperative part of the mechatronic system design with the increasing complexity. However, the current design platforms are all extended from the software modeling tools, which are essentially software-oriented and lack the support for the characteristics of complex mechatronics. Based on the analysis of key problems for the design of multi-domain complex mechatronics, a design-oriented modeling platform, M-Design, is developed for its design. First, the domain-independent fundamental semantics of SysML is represented in M-Design. Next, the domain-dependent design semantics is also modeled for various related domains of complex mechatronics. Finally, the transformation is designed to be conducted automatically from system design model to system simulation model to support the early simulation and find the irrational design schemes as early as possible. A case study is given to illustrate the proposed method.

**Key words:** SysML; multi-domain; complex mechatronics; system design; integration of design and simulation

**摘要:** 随着机电产品的复杂性增加, 针对其涉及的多领域的系统设计成为必不可少的一部分。然而, 现有的多域复杂机电产品系统设计平台均是在软件系统建模平台的基础上扩展而来, 其本质仍然是面向软件工程的, 且缺乏对多域复杂机电产品特有性能刻画的工具。在对多域复杂机电产品系统设计的关键问题进行分析的基础上, 以通用复杂系统建模语言 SysML 为基础, 研发了多域复杂机电产品系统设计的建模平台——M-Design。不仅对领域无关的 SysML 基础语义进行表示与建模, 同时更能对领域相关的多域复杂产品系统级设计语义进行表征, 且支持从多域统一的系统设计到多域统一的系统仿真自动转换, 从而及早进行仿真, 以发现不良设计方案。最后进行了实例验证。

**关键词:** SysML; 多域; 复杂机电产品; 系统设计; 设计与仿真集成

\* 基金项目: 国家自然科学基金(61173126, 61070064); 国家高技术研究发展计划(863)(2011AA100804); 浙江省自然科学基金(R1110377)

收稿时间: 2012-05-30; 定稿时间: 2012-09-29

## 1 介绍

复杂机电产品正呈现出功能高度复杂集成、领域间耦合关联、可重构等诸多特点.与一般产品相比,复杂产品所带来的挑战是:不同领域子系统间将产生不可预测的功能耦合、交叠甚至冲突,原本功能良好的子系统将可能产生不可预测的行为<sup>[1]</sup>.对复杂产品进行系统设计已成为不可缺少的重要部分.近年来,基于模型的系统工程(model-based systems engineering,简称 MBSE)及其标准建模语言 SysML 已引起产品设计领域的广泛关注,MBSE 正成为复杂机电产品系统设计与建模的重要方法<sup>[2,3]</sup>.它从需求阶段开始即通过模型(而非文档)的不断演化、迭代递增而实现产品的系统设计<sup>[4,5]</sup>,具有显著的优势,如:通过模型的形式化定义可以清晰地刻画产品设计初期在结构、功能与行为等各个方面的需求;基于模型可以及早模拟分析发现大量不合理的设计方案;同时模型还为各方提供了一个无二义性的设计信息交流工具.因此,近年来,MBSE 已成为复杂产品系统设计建模的研究重点.

目前,支持 SysML 设计建模的平台均是由已有 UML 建模平台扩展而来,其本质仍然是面向软件工程,而复杂机电产品具有许多与软件产品本质的区别,如涉及机械、电子、控制等多个领域;其行为复杂,往往是离散/连续的混合,普通的过程式描述不再能满足其建模的需求等.基于上述分析,我们在深入分析多域复杂机电产品系统设计需求的基础上,研发了一个支持 SysML 建模的多域机电产品系统设计建模平台,以满足日益复杂的多域机电产品系统设计建模的需要.

## 2 多域复杂机电产品系统设计的关键问题分析

为使研发的平台具有很好的通用性和很强的鲁棒性,需要针对系统研发中的一系列关键问题进行分析,然后才能对不同关键问题加以解决,本平台 M-Design 重点突破了以下 3 个问题:

领域无关的 SysML 基础语义的表示与建模.标准系统建模语言 SysML 的语法复杂,语义丰富,能对复杂系统通用的需求、结构、行为与约束进行无二义性的表征与建模,且它们间的语义关联与约束也很多.对 SysML 的基础语义进行严格、清晰的建模是建立鲁棒、可扩展、模块化、可重构的多域复杂产品系统级设计建模平台的基础.

领域相关的多域复杂产品系统级设计语义的建模表征.多域复杂产品本质上是由物理部分和控制部分组成.前者是基础与核心,可进一步分为机械、电子、液压等领域;后者是“大脑”部分,是灵魂与中枢.可正确地对各领域系统级设计语义知识如需求、结构、行为等进行统一建模表征.

支持多域产品设计与仿真的集成.复杂产品的系统级设计涉及机械、电子、液压、控制、软件等多个领域,其正确性难以保证.且由于多域产品行为的复杂性和不可预测性,设计者根本无法基于静态的设计模型验证其动态行为,必须加以仿真才能进行验证.

## 3 领域无关的 SysML 基础语义的表示与建模

SysML 在 UML<sup>[6]</sup>的基础上面向系统工程发展而来,它自然支持 OMG 提出的元模型机制 MOF,即四层元建模架构:实例层、模型层、元模型层和元元模型层.为此,要研发支持基于 SysML 的多域复杂产品建模平台,必须首先要有支持基于 SysML 的建模元模型,然后才能在此基础上建立基于 SysML 的模型及其实例.而 SysML 的语义、语法均十分复杂、丰富,必须对其进行分析与总结,并结合面向设计的多域复杂产品系统级建模的需要,给出 SysML 的元元模型语义.基于此,本文给出 SysML 通用建模的元模型及其关系的刻画方法,并基于 SysML 的语法语义约束与规则,给出基于元模型表示的 SysML 模型正确性评价算法.其总体过程如图 1 所示.

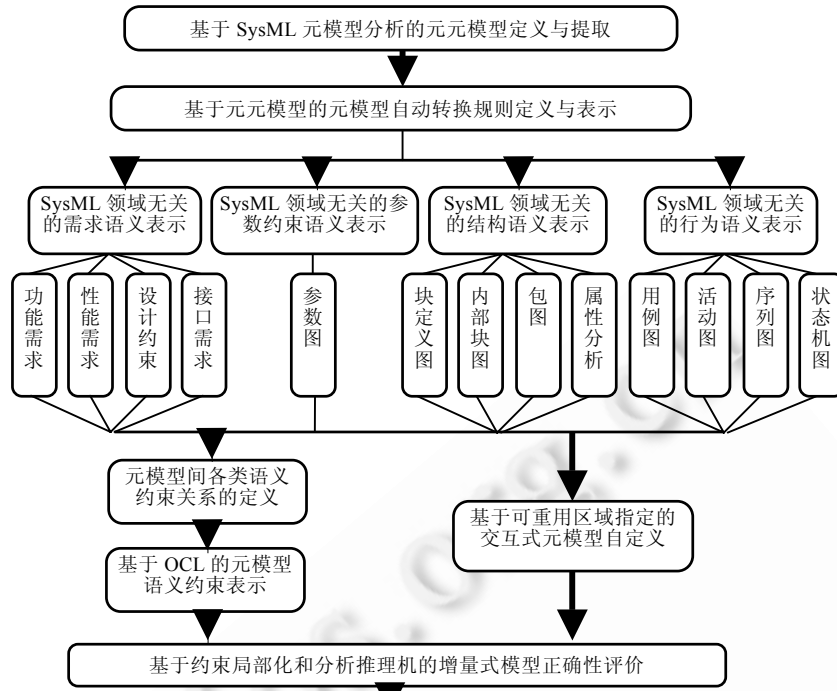


图1 领域无关的 SysML 语义建模

### 3.1 SysML元元模型语义定义与提取

建模的元模型层规定了模型的种类以及不同模型的属性,但这对于建模还不够.建模要创建用元模型体系表示的模型,并对元模型体系设定的模型属性进行修改、添加与删除,且在修改、添加与删除时,元模型还有各种不同的规则与语义约束.由于 SysML 元模型体系结构庞大、复杂,属性与属性之间具备各种关联,对一个属性进行修改,整个元模型体系内部可能会触发多个属性的修改,甚至触发雪崩般的修改.为此,要支持后续鲁棒、可重用的多域复杂产品建模,必须先对 SysML 的语法与语义进行分析,设计、定义一套满足支持 SysML 鲁棒、可重用、可扩展建模的 SysML 元元模型体系,其部分设计定义如图 2 所示.

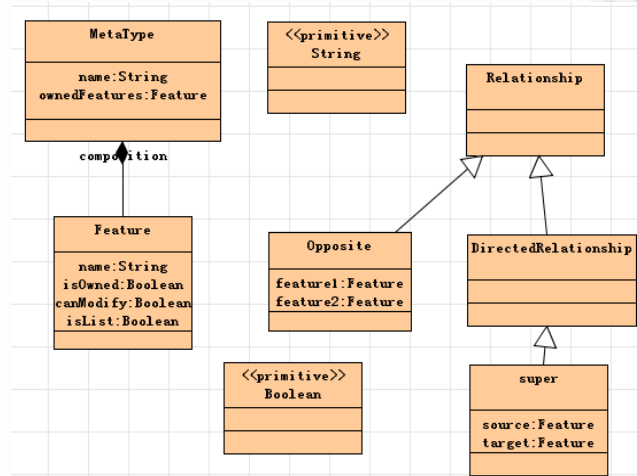


图2 部分设计定义的 SysML 元元模型

这里,元元模型 MetaType 的实例是具体的元模型,而元元模型 Feature 的实例则是某个具体的元模型的某个属性;元元模型 Opposite 关系用来连接两个 Feature 元元模型,以帮助定义这两个 Feature 元元模型之间的交叉引用关系;元元模型 super 关系也用来连接两个 Feature 元元模型,以帮助定义这两个 Feature 元元模型的自动包含关系.在定义好 SysML 的元元模型后,接着要探索给出基于元元模型的元模型的具体转换规则.基于上述元元模型体系的优势在于:① 统一对元模型进行表示,实现元模型属性与行为的可配置,从而便于生成文件,进行修改,同时实现文件与可执行元模型体系的转换,增加可扩展性,模型的结构清晰、明确,减少出错概率,增加稳定性;② 实现了很好的扩展性,从而便于扩展实现约束检测、便于回滚、便于添加新的约束规则等.

### 3.2 基于SysML语义与语法约束的模型正确性验证

若建立的模型不进行约束检查,不仅易于出错,而且人工查错也极易疏漏且劳动强度高,因而需要实现约束的正确性检查.为此,在 M-Design 研发时,在深入分析四层元模型体系结构的基础上,提出了两种约束机制:① 加载在元模型层,验证模型的语义符合性;② 加载在模型上,验证实例化后模型的正确性.在具体进行模型正确性验证时,重点解决以下 3 个问题:

(1) 触发何种正确性的约束检测:由于 SysML 是自描述语言,其元元模型与元模型均为 SysML 层,模型层为 Classifier 层,实例层为实例说明层,而实例说明层到 Classifier 层很容易,SysML 已经提供了这一实现的方法,因此这里我们重点介绍从 Classifier 到 SysML 映射的实现.M-Design 中采用的方法是将 SysML 层用 Classifier 层表示,即创建表征 SysML 体系结构的模型,并将该模型与 SysML 表示关联起来.此外,为了加速这种映射,使用枚举变量来作为映射的中间桥梁.

(2) 何时触发正确性的约束检测:为实现正确性检查的有效触发,同时不影响系统的运行速度,在 M-Design 中提出了两种触发处理方案.一是进一步扩充数据结构的底层信息,增加若干监听信息,以监视所有对属性的修改,当有属性被修改时,触发其对应的约束检测.这只适用于约束为查询语言,亦即约束不会对数据结构造成改变的情况.其优点是能够实现真正的动态、即查即用、即时检测的功能.但其缺点是,当进行非原子操作时,即需要分步操作,才能达到预期的正确结果,但在分布操作的过程中也会检测到约束并警报,从而造成 CPU 的浪费以及错误的警报;二是在系统中额外增加一个相关线程,当系统空闲时自动逐个扫描约束,并加以检查,其优点是会造成无响应.

(3) 如何实时验证设计模型的正确性:在 M-Design 中借用了现有的 OCL 约束规范来描述 SysML 的约束,对于 OCL 不能表示的部分,拟使用额外的非通用性质的函数来实现检查.这里需要解决的重要问题是全局约束的局部化问题.全局约束是否满足的验证工作十分费时、费力,为此,我们提出了基于弱链接处解耦的原理将全局约束问题进行局部化处理.在此基础上,拟探索约束的增量式验证方法.

## 4 领域相关的多域复杂产品系统级设计语义的建模表征

SysML 本身并不直接进行特定领域,如机械、电子、液压、控制等领域的系统级设计知识建模,但它具有极强的扩展能力,能对建模语言词汇、语义约束类型及属性值这 3 个方面进行扩展,使其具有支持复杂产品多域系统级设计知识的统一建模能力,以使设计人员在多域产品系统级设计时可方便地表达其设计意愿.考虑到多域复杂产品本质上是由物理部分和控制部分两部分组成,且它们有一些需要扩展的共同之处,因此,首先对其共性问题展开研究.总体方案如图 3 所示.

### 4.1 基于SysML的连续/离散混合行为的统一表征

多域复杂产品与其他复杂系统相比的一个最大特点就是其行为不仅包含一般系统具有的离散行为,还有许多的连续行为,且绝大部分是离散/连续的混合行为.且这种混合行为不仅在多域复杂产品的物理子系统中存在,在其控制子系统中也存在.现有的 SysML 不能表达这种复杂的行为,为此,必须对 SysML 加以扩展,建立基于 SysML 的混合行为建模方法,不仅能方便地支持设计人员在需求功能驱动的正向设计时方便地表达自己的设计意愿,也利于给出其面向仿真分析的统一行为模型表示,从而有利于与后续具体的仿真平台集成.

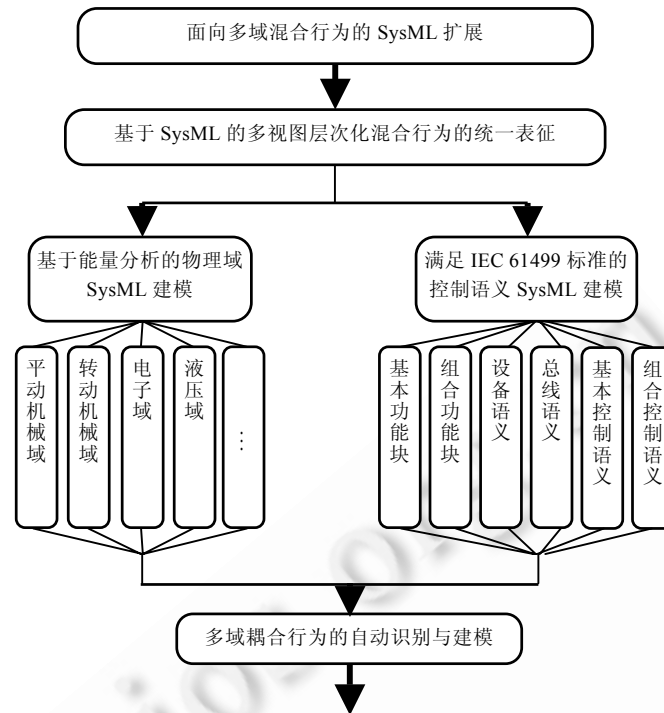


图3 领域相关的 SysML 语义建模

为此,通过对 SysML 进行轻量化扩展,即通过 SysML 的版型(stereotype)扩展机制,提出“条件约束块”概念,以表示离散/连续混合行为与多域耦合行为中对象的不同状态.其核心语义是其状态不同于约束块的“始终活动”,而是可以有条件地在“活动”与“不活动”间转化;随后,拟进一步在上述简单、有序参数图的基础上对其进行层次化、多视图表示能力的扩展,以支持包含多域行为耦合的多细节层次的离散/连续混合行为建模,形成基于有序参数图的层次式多视图系统层统一行为模型的表征.其中,视图将包含框架视图和行为视图.框架视图中表达的混合行为通过由一个顶层的条件约束块及其子条件约束块表示.子条件约束块还可进一步逐层地包含属于其自己的子条件约束块.这些条件约束块共同构成了混合行为及多域耦合行为的框架视图.而对于不包含子条件约束块的条件约束块,拟包含基于微分方程、差分方程或其他表示的一般约束块,以表示该状态下满足广义基尔霍夫定律的具体连续行为或具体的离散状态,构成模型具体的连续行为视图或离散行为视图,同时还将包含其活动转化条件等重要信息.

#### 4.2 基于 SysML 的多物理域建模构件统一表达

由于 SysML 已能表示需求、结构、行为等系统层设计的主要部分,但还不能表示复杂产品各个领域的知识,为此,需要在对物理域的本构语义加以深入的基础上定义基本的领域构件.此外,考虑到系统设计模型的正确性还需进一步通过仿真来进行验证,而多域统一的系统仿真平台与仿真建模语言已有多.为此,在 M-Design 平台研发时,先对典型的多域统一系统仿真语言 Modelica 和 Simscape 的共性进行分析,其共性见表 1.在此基础上,从能量流的角度出发,抽取复杂机电产品可能涉及物理域,如机械、电子、液压等的特定本构语义并以相应的领域参数进行表示;接着对具体的基础构件,拟先结合其领域参数及其具体属性,进行物理本构语义的定义与描述,然后考虑不同分析精度要求的需要,探索不同细节层次的本构语义划分与表示方法;基于此,运用 SysML 扩展技术,确定基础建模构件的层次式、模块化定义与表示方法;最后针对上述不同的物理域,建立与系统层设计构件相对应的、树状结构的基础分析构件组织方式,以便于其查找与重用.

表 1 Simscape 和 Modelica 元模型的对应关系

概念	Simscape 的元模型	Modelica 的元模型
物理系统(physical system)	Simscape diagram	Model (with hierarchical structure)
物理构件(physical component)	Component	Model (no hierarchical structure)
变量(variable,包括势变量 across 和流变量 through)	Variable (Across, Through)	Variable (Across, Through)
参数(parameter)	Parameter	Parameter
能量端口(energy port)	Node	Member class typed by connector
输入输出(input/output)	Input/Output	Input/Output
能量类型(energy type)	Domain	Connector
初始条件(initial condition)	Setup	Modification
本构方程(intrinsic equation)	Equation	Equation

考虑到需求功能对多域复杂产品系统级设计的极端重要性,接着重点关注基于 SysML 的功能建模方法,不仅方便于设计者在系统级设计时给定功能要求,同时更为后续的功能自动分解与推理等提供支持.这里,提出功能效应的概念,表示功能对其作用上的流所产生的效果,包括流的类型变化和流的属性变化.这样,功能总体上被表示为输入流和输出流关系,功能的作用是对输入流施加一些影响和效果,然后产生想要的输出流,在这个过程中,流的类型或者属性发生改变.通过这种表示,功能效应即包含了计算机能够理解的语义信息,从而支持自动的推理和分解.同时,考虑到工作原理对多域复杂产品需求功能实现的重要辅助作用,这里也对其进行显式建模,具体地对物理效应和几何材料特性进行建模.物理效应是自然界中客观存在的物理现象和规律,它描述了物体的属性在物理规律下是如何改变的.可以将功能效应和物理效应联系起来,根据不同的物理效应,功能可以按照不同的方式进行分解.

### 4.3 基于 SysML 的控制域设计统一建模表征

考虑到控制部分在现代复杂产品系统设计中的作用日益重要,国际电气工程学会从功能设计的层次提出了 IEC 61499 Functional Block 新标准<sup>[7]</sup>,突破了以往 IEC 61131 可编程算法级的局限,对多域复杂产品的系统级设计具有重要意义.为此,在功能驱动的复杂产品正向设计的系统级设计时,要以 IEC 61499 标准为基础,一方面符合设计人员的设计习惯,另一方面也能很好地与后续的详细控制设计相关联.为此,在 M-Design 平台研发时,我们进行了面向系统级控制设计的扩展.

首先基于 SysML 扩展,以对 IEC 61499 中定义的基本控制功能模块进行了表示.本质上,FB 首先可以分为外部接口和内部处理两部分.外部接口部分进一步分为事件接口部分和数据接口部分,且在事件与数据之间存在高度耦合.这样分开的好处在于内部处理时方便.FB 的内部处理部分也进一步对应地分为执行控制部分(execution control chart,简称 ECC)和具体控制算法部分.前者的本质是一个状态机,根据不同的输入事件,将调用不同的具体控制算法以实现不同的功能.而后者即是执行不同控制功能的具体算法.基于上述分析,我们以 SysML 中的事件作为元类扩展生成 FB 的事件输入与事件输出;以 SysML 中的变量类作为基本 FB 的输入数据和输出数据;以 SysML 中的依赖关系作为基本 FB 的数据与事件间关联关系的元类.而在内部处理部分,采用 SysML 中的状态机及相关概念,如状态、变换、约束、操作分别表示 ECC、执行状态、执行变换、执行条件及执行算法等.对于基于基本 FB 的 FB 网络,其核心是如何基于 SysML 的扩展对组合 FB 中的事件与数据流进行建模.我们提出了基于 SysML 中的块图的接口(interface)作为元类来扩展表示基本 FB 的输入与输出事件、以特征(feature)为元类来扩展表示基本 FB 的输入、输出及内部数据的表达方法,其优势在于能够尽量多地应用于其中的类(class)及其组成部分,从而尽可能地继承应用相关的功能.

限于篇幅,本文对多域耦合行为的识别与建模不作介绍.

## 5 基于模型变换的多域系统设计与系统仿真集成

如前所述,基于 SysML 建立的复杂机电产品系统设计模型非常复杂,很难基于静态的模型验证其正确性,而仿真是自动验证模型正确性的一种常用手段,为此,已有研究探索了将 SysML 与仿真语言相集成的方法,以支持 SysML 模型的自动仿真.在本平台 M-Design 的研发过程中,采用了 OMG 组织提出的模型驱动架构的模型转

换思想,将 SysML 表示的平台无关的系统设计模型自动转换为仿真平台相关模型,其核心思想是基于元模型的映射,将符合元模型 A 的模型 Ma 转换为符合元模型 B 的模型 Mb,如图4 所示.要实现模型转换,核心内容是定义源元模型和目标元模型以及它们之间的转换关系.之后,源模型便可以通过转换引擎转换到目标模型.

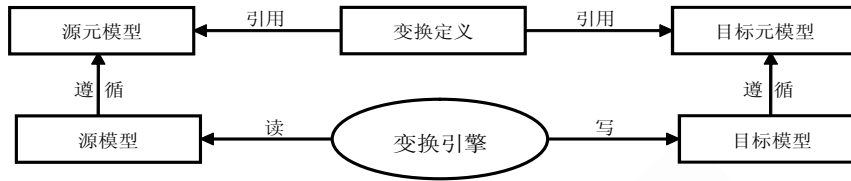


图4 模型转换基本思想

通过对常用的几种模型转换方法进行比较可知<sup>[8]</sup>,基于 Triple Graph Grammar(TGG)<sup>[9]</sup>的模型转换方法具备了如下的所有属性:(1) 支持陈述式或操作式转换规则的描述;(2) 支持模型一致性检查;(3) 支持单项或双向模型转换;(4) 文字或图形描述;(5) 基于元模型描述;(6) 与 MOF 2.0 标准相兼容;(7) 支持代码生成;(8) 可追溯性;(9) 变动传递.因而它是一种优秀的模型转换方法.其核心思想是,将需转换的两种元模型均视为图,元模型的元类为图的节点,元类之间的关系为图的边.在两图之间建立第 3 幅图,称为对应图(correspondence graph),该图的节点为对应关系,连接需要进行转换的元类.该方法的主要内容如图 5 所示.

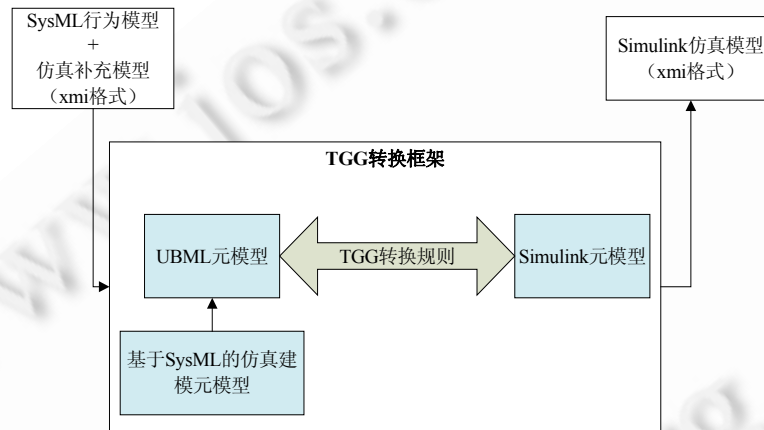


图5 基于 TGG 的模型转换方法

该方法的核心内容是构建图中的 TGG 转换框架.基于该框架,可以实现 SysML 模型与 Simulink 模型之间的双向转换.这两种模型均是以 XMI(XML-based metadata interchange)格式输入和输出.它使用可扩展标记语言(XML)进行描述,为使用不同语言和开发工具的工程人员之间提供了元数据信息交换的标准方法.构建 TGG 转换框架的过程完全在元模型层面上进行.在已知源模型和目标模型的建模语言元模型的基础上,建立二者元模型间的对应关系,并描述具体的转换规则是方法的关键所在.

### 5.1 对应图的表示

综上所述,基于 TGG 的模型转换由存在于两个元模型之间的对应图表示,其定义包括两部分内容:

(1) TGG 模式:由一系列集成链类型组成,这些链类型直接联系了需要进行转换的两个元类,显示了元类之间的对应关系,如图 6 所示.该链类型表示 SysML 中的 Class 元类与 Simulink 中的 Chart 元类相对应.链类型本身可以具有属性,如图 6 中的 n:String 属性,该属性可以在 TGG 转换规则中使用;也可以具有优先级,因为两个相对应的元类之间可能存在多条链类型,当这些转换规则同时得到满足时,采用优先级高的链进行转换.

(2) TGG 规则:仅有元类之间的对应关系还不足以描述转换的操作和实现过程,需要由 TGG 规则来描述具体转换的实现.每一条集成链类型均对应唯一一条 TGG 规则.它描述了源元素与目标元素是以何种方式同时出

现在 TGG 的 3 幅图中,如图 7 所示,即为上述 Class-Chart 链类型对应的 TGG 规则.它表示,当 Class 被版型 FunctionBlock 修饰,并且被 Package 包含时,才转换成 Chart,并且将该转换生成的 Chart 包含在 Lib 当中;反之亦然.值得注意的是,TGG 规则本身是不带有方向性的,当执行具体的向前或向后转换时,它可以生成有向的可操作规则,用于实现具体的转换任务.此外,对于无法转换的情形,则通过版形扩展定义相关对象后再定义 TGG 规则.

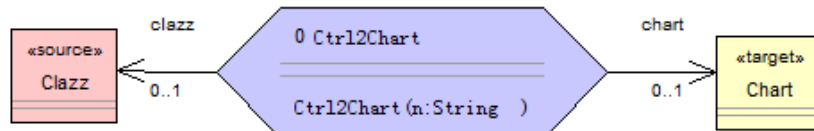


图 6 TGG 集成链类型示例

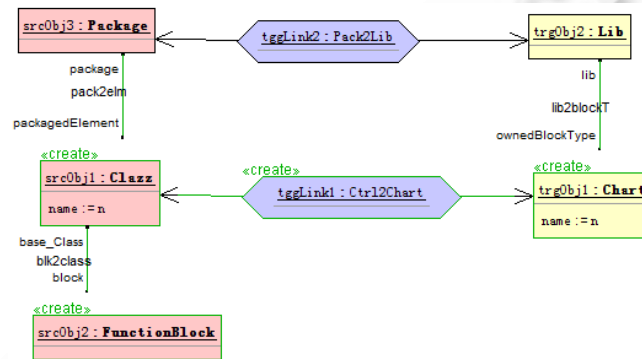


图 7 TGG 规则示例

## 5.2 TGG 模式

为了定义 TGG 模式,其核心内容是识别系统设计建模语言 SysML 元类与系统仿真建模语言,如 Simscape, Modelica 元类之间的对应关系.而两个元类之间存在对应关系的条件是,它们具有相同的功能和语义.因此,为了定义 TGG 模式,必须分析每个元类的功能和语义.在 M-Design 建模平台中,建模方法分为两个阶段:(1) 元件建模阶段.该阶段分别对物理元件和控制元件进行建模,每个元件包括结构建模和行为建模两部分.该阶段产生一系列元件模型并组成元件库.(2) 系统建模阶段.该阶段使用元件库中的元件并将其实例化为系统的一个组成部分,然后将这些元件连接成网络,形成系统模型.该阶段的产物为系统模型,它由元件实例和连接器组成.

这一建模方法对于系统设计仿真建模语言是一致的,因此,两者的元模型中的每个元类均对应到产物模型的相关概念上.例如,控制元件是元件建模阶段的产物,它可看作一个底层概念(本体);控制元件由 Control profile 中的 Function Block 表示;在 Simulink 中,控制元件由 Stateflow 中的 Chart 表示.因此,控制元件这一本体将系统设计中的元类 Function Block 与 Simulink 中的元类 Chart 联系起来,从而识别出二者之间的对应关系.

综上所述,TGG 模式的定义可归纳为以下步骤:(1) 分析复杂机电系统建模过程,分析每个步骤的相关产物,识别本体概念;(2) 寻找设计行为模型中表示该概念的模型元素;(3) 寻找仿真建模语言中表示该概念的元类;(4) 在表示同一概念的两个元类之间建立对应关系,即集成链类型.

### TGG 规则

TGG 模式中的每条链类型均对应唯一一条 TGG 规则,用于说明模型转换的具体实现过程.基于 SysML 与 Simscape/Simulink/StateFlow 元模型的特点,我们为 M-Design 抽象出了 TGG 规则定义的模板:它是一个四元组 (OT,PT,BC,CC).OT 和 PT 表示了本条规则所执行的任务(即生成哪些模型元素),BC 和 CC 表示本条规则适用的条件(即当哪些模型元素或关系已经存在时,本条规则才能发挥作用).

OT(objective transformation)是目标转换.它包含 3 个对象:本条规则所要转换的源元类实例 Os、目元类实例 Ot 以及链类型实例 L.一旦待转换的两个对象之一被创建,则另外一个须同时出现.



PT(property transformation)是属性转换.每个元类包括两类属性:简单属性是指其类型为基本类型(如String,Real 等)的属性;引用属性是指其类型为其他元类的属性.简单属性的转换可以直接通过参数共享来实现,即链类型的属性可以作为 TGG 规则的参数被 Os 和 Ot 共享,从而实现属性值的传递.而引用属性的转换需要用到作为属性类型的两个元类的实例(用 Ps 和 Pt 表示)及其之间的 TGG 链类型实例 Lp:创建 Ps 与 Os 以及 Pt 和 Ot 之间的关联关系(正是该关联关系定义了 Os 和 Ot 的两个引用属性),引用在 Ps 和 Pt 之间的链类型实例 Lp.

BC(belonging condition)是所属条件.待转换元类可能作为其他元类的属性,因此,在转换发生之前,其所属的元类(其实例用 Bs 和 Bt)之间必须已经存在转换关系.所属条件引用了 Bs 和 Bt 之间的转换关系,当该转换已经存在时,本转换才有可能发生.

CC(constraining condition)是约束条件.约束条件描述了待转换对象需满足的条件,只有这些条件已存在,本转换才能发挥作用.约束条件包括两类:属性约束是指待转换对象的简单属性值必须满足的条件;引用约束是指待转换对象必须被某个对象所引用,成为其引用属性,本规则才能适用.

图 8 显示了一个 TGG 规则的例子.

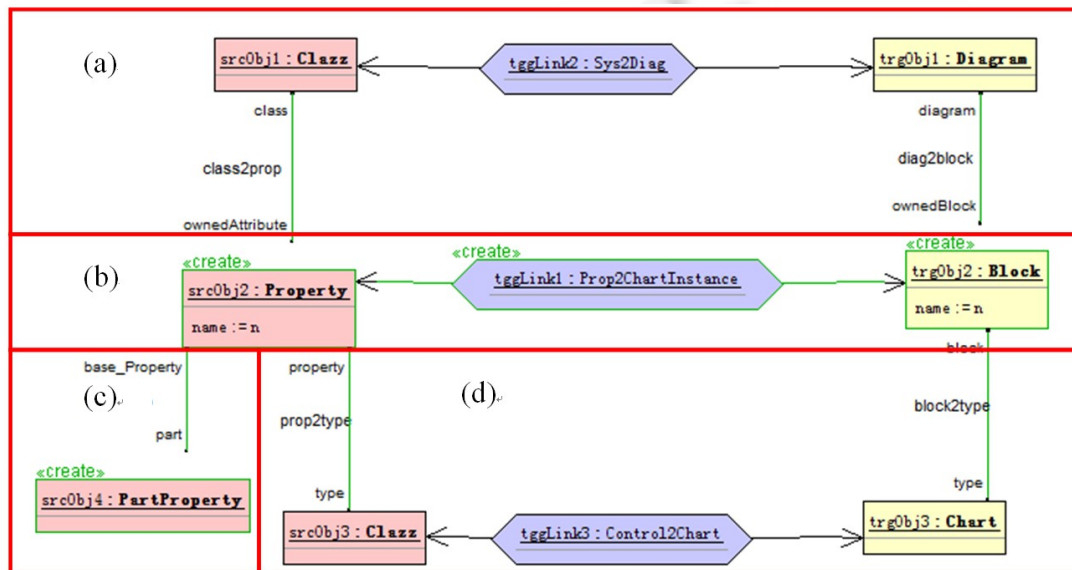


图 8 TGG 规则示例

该规则表示了系统设计的元件部件到 Simulink 中 Block 之间的转换.由于系统部件在系统设计行为模型中表示为系统的部件属性,因此,该转换规则的目标转换(如图 8(b)所示)发生在 Property 的对象与 Block 的对象之间.两个待转换对象均包含两个属性:部件名称 name 和部件类型.因此,属性转换中,简单属性 name 的转换直接通过链参数 n 传递;引用属性的转换(如图 8(d)所示)通过 Class 与 Chart 之间的转换加以描述.系统部件需包含在系统中,其所属的系统模型之间的转换必须已经存在.所属条件(如图 8(a)所示)描述了系统设计行为中表示系统的 Class 到 Simulink 中表示系统模型的 Diagram 之间的转换.虽然 Class 可以有多种属性,但只有部件属性才适用于本条转换规则.因此,约束条件(如图 8(c)所示)描述了这条约束,即只有当 Property 是 Part Property 版型时,才将该 Property 转换为 Simulink 的 Block.在具体应用时,转换引擎会采用模式识别的方法,判断所属条件和约束条件是否已经具备,如果具备,则运用本条规则,执行相应的目标转换和属性转换.

基于 SysML 与 Simspcae/Simulink/StateFlow 元模型以及二者之间的模型转换关系,可以生成 TGG 模型转换框架<sup>[10]</sup>.该框架能够自动实现 SysML 设计模型与 Simscape/Simulink/StateFlow 仿真模型之间的双向转换<sup>[11]</sup>.

## 6 建模实例

为验证本文研发的多域复杂产品系统设计建模平台,本文以图 9 所示的倒摆为例来进行建模.倒摆系统是一个复杂且不稳定的非线性系统,在控制理论中也是一个典型的范例.它是由机械子系统和控制子系统所构成,两个系统互相反馈,周而复始,简洁地表达了一个控制系统的流程:系统 A 提供输出给系统 B,系统 B 计算出反馈,提供给系统 A,系统 A 重新计算输出,反馈给 B.系统的需求及其外界交互分别表示如下:

主需求为:小车需要通过水平移动来控制,从而使得倒摆始终保持向上.

副需求为:(1) 即使杆被扰动了也要保持向上.(2) 用户可以修改反馈参数以达到需要的结果(更短的时间到达平衡).

其与外界的交互分为两种:(1) 外界对系统的扰动;(2) 用户对系统反馈参数的调节.

因此,在用例建模部分有两个用例,同时,这两个用例分别对应了两个需求.在图 10 中,我们对需求与用例进行了建模.

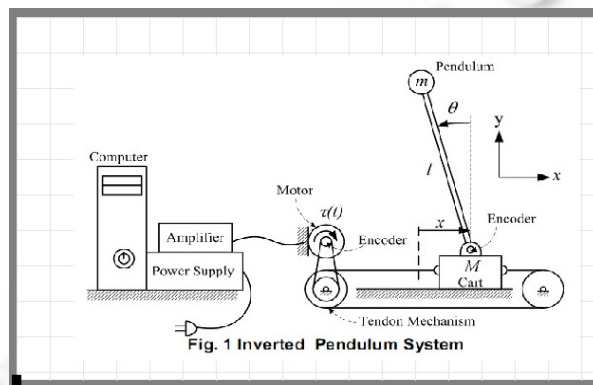


图 9 倒摆系统图

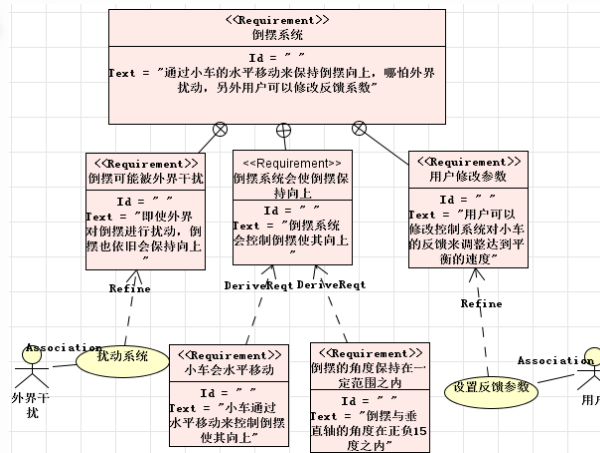


图 10 需求建模与用例建模

在结构建模时,将系统分为两块,一块是机械部分,另一块是控制部分,其次是接口设计.其内部结构建模表示如图 11 所示.

上述结构建模还很初步,机械部分与控制部分均可以进一步细化,如图 12 所示.机械部分具有小车和倒摆,小车和倒摆通过轴承连接在一起,小车有发动机,可以控制小车的加速度.小车和倒摆上都有感应器,用以为控制部分提供输入.其内部结构如图 12 所示.

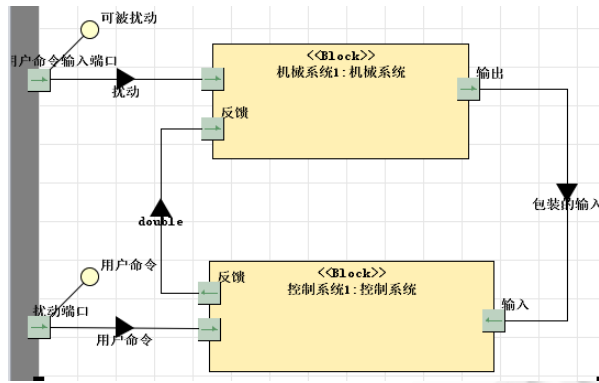


图 11 内部建模图

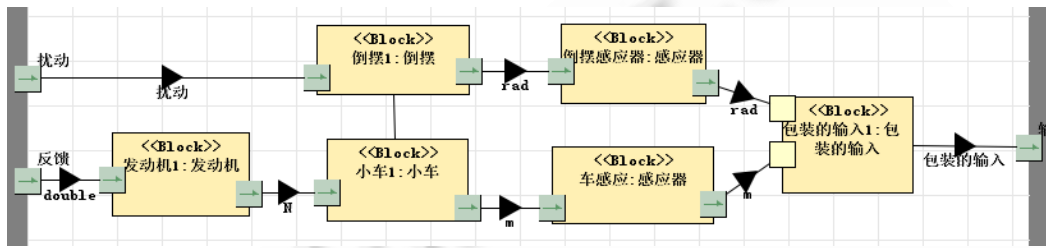


图 12 机械部分的内部结构图

接下来是行为建模,其行为部分的建模也极为容易,如前所述,可以通过结构内部建模图发展而来,如图 13 所示.

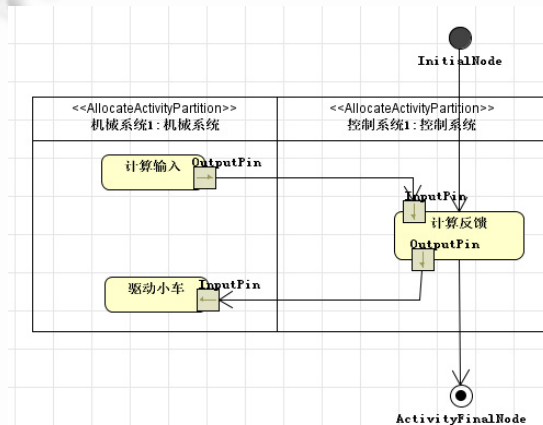


图 13 行为建模及其与结构的对应

在参数建模阶段,首先要引入领域知识与约束,如图 14 所示.

图 14 给出了各种公式,包括各种速度公式、矩阵化公式、矩阵计算公式等,涉及到了牛顿运动定律等物理知识,体现了不同物理构件的本构语义.

图 15 给出反馈的计算过程,首先是通过上一次输入保存的小车位置与倒摆位置来计算小车速度与倒摆速度,接下来是将小车位置、倒摆位置、小车速度、倒摆速度矩阵化,然后进行矩阵运算,最终得到反馈值.

在 M-Design 中设计建模结束之后,基于上述的模型转换技术可以生成 Simscape/Simulink/StateFlow 可接受的仿真模型,如图 16 所示,其仿真结果如图 17 所示,这里,初始偏角设为 0.002 弧度.

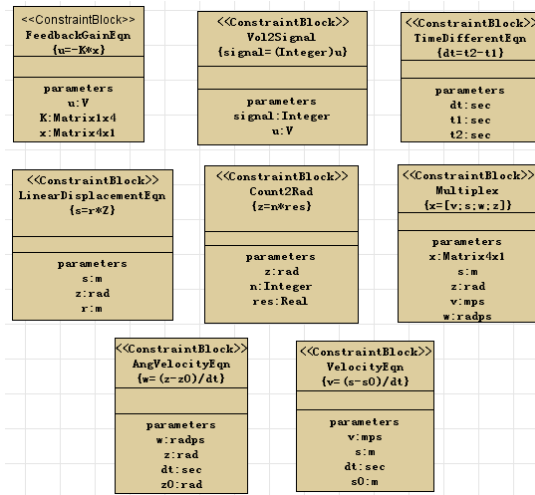


图 14 领域知识建模

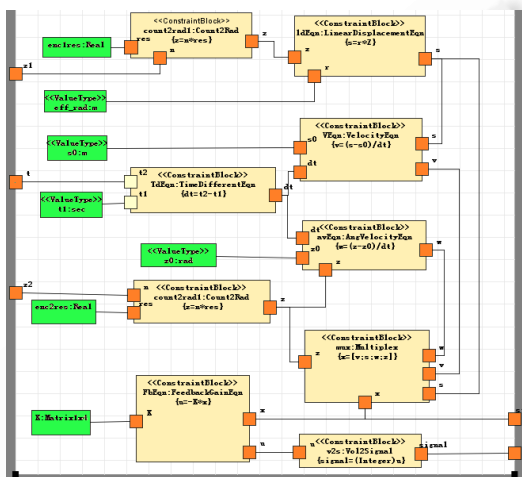


图 15 参数建模图

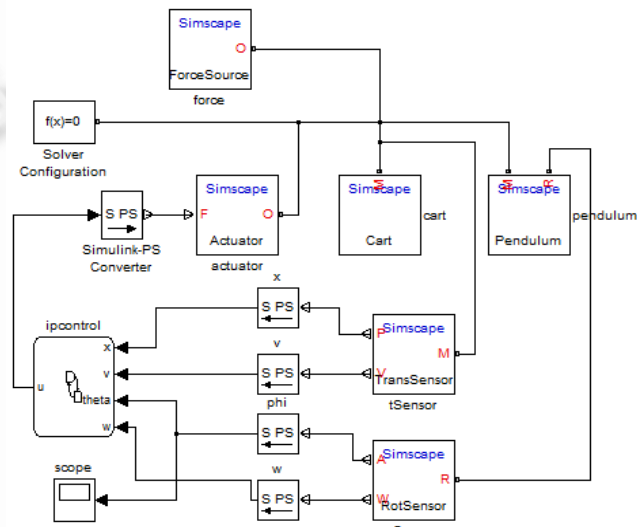
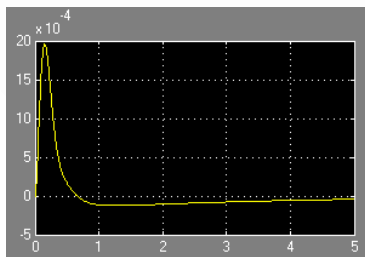
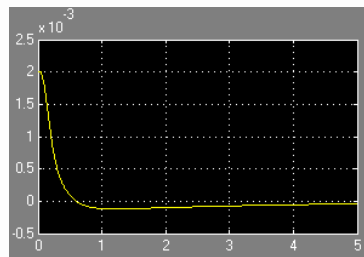


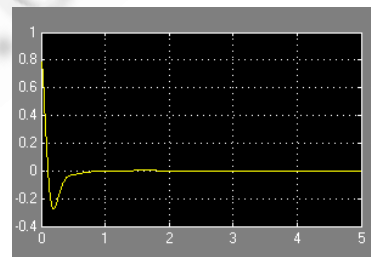
图 16 倒摆系统 Simulink 模型



(a) 位移仿真曲线



(b) 速度仿真曲线



(c) 加速度仿真曲线

图 17 倒摆的 Simulink 仿真结果图

## 7 结论

为支持多域复杂机电产品的系统创新设计,必须首先要有多域复杂机电产品的系统建模平台,以支持复杂

机电产品系统设计过程中所涉及的不同领域,如机械、电子、控制等设计知识的统一表征.本文以复杂系统设计建模语言 SysML 为基础,通过对其进行扩展,研发了能够支持多域统一建模的系统设计建模平台——M-Design,本文贡献如下:

(1) 探索了领域无关的 SysML 语义的表示方法;

(2) 针对复杂机电产品的特点,对其物理部分进行了统一处理,基于能量分析建立了物理域构件的 SysML 统一表达方式;

(3) 基于最新国际标准 IEC 61499,提出了基于 SysML 的 IEC 61499 表示方式,建立支持控制系统面向功能的自顶向下设计的表征方法;

(4) 提出了基于 TGG 的 SysML 系统设计模型与 Simulink/Simscape/StateFlow 系统仿真模型双向映射机制.

系统研发是一个长期的、艰巨的过程,本文工作还只是一个开始,还有许多工作需要加以完善,如支持多域系统协同设计等.

#### References:

- [1] 钟掘,等.复杂机电系统耦合设计理念与方法.北京:机械工业出版社,2007.
- [2] Michelle B, David H. System design: New product development for mechatronics. Benchmark Report, Aberdeen Group, 2008,1.
- [3] Object Management Group (OMG). Systems Modeling Language specification. 2010-06-01. <http://www.omg.org/spec/SysML/1.2/PDF>
- [4] Thramboulidis K. Model integrated mechatronics—Towards a new paradigm in the development of manufacturing systems. IEEE Trans. on Industrial Informatics, 2005,1(1):1-14.
- [5] Turki S, Soriano T. A SysML extension for bond graphs support. In: Proc. of the 5th Int'l Conf. on Technology and Automation. Thessaloniki, 2005.
- [6] Object Management Group (OMG). Official UML Specification. 2011. <http://www.omg.org/spec/UML/2.4.1/>
- [7] International Electrotechnical Commission. Technical Report, 61499-1: Function Blocks - Part 1: Architecture, 2005.
- [8] Schürr A. Specification of graph translators with triple graph grammars. In: Proc. of the WG'94 Workshop on Graph-Theoretic Concepts in Computer Science. 1994. 151-163.
- [9] Königs A. Model transformation with triple graph grammars. In: Proc. of the Model Transformations in Practice Workshop at MODELS Conf. Montego Bay, 2005.
- [10] Amelnxun C, Klar F, Königs A, Rötschke T, Schürr A. Metamodel-Based tool integration with MOFLON. In: Proc. of the 30th Int'l Conf. on Software Engineering. New York: ACM Press, 2008. 807-810.
- [11] Klar F, Rose S, Schürr A. TiE—A tool integration environment. In: Proc. of the 5th ECMDA Traceability Workshop, Vol.WP09-09. 2009. 39-48.



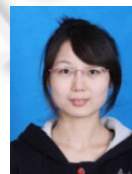
刘玉生(1970—),男,湖南醴陵人,博士,教授,主要研究领域为模型驱动系统设计, CAD/CAE/CAM 集成.



樊红日(1985—),男,博士生,主要研究领域为模型驱动工程.



蒋浴芹(1987—),男,硕士,主要研究领域为模型驱动工程.



曹悦(1986—),女,硕士,主要研究领域为系统设计,系统仿真集成.