HZ BOOKS
华章教育

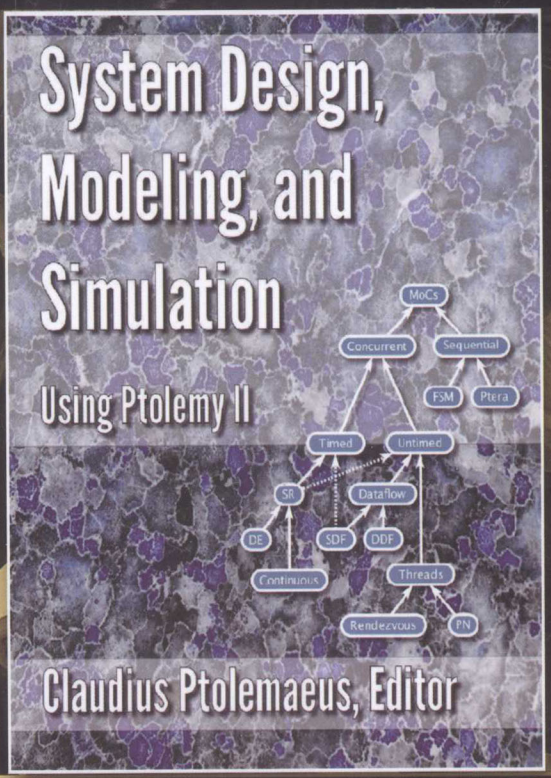
计 算 机 科 学 丛 书

信息物理融合系统（CPS）设计、建模与仿真

基于Ptolemy II平台

[美] 爱德华·阿什福德·李（Edward Ashford Lee） 等编著
吴迪 李仁发 译

System Design, Modeling, and Simulation
Using Ptolemy II



机械工业出版社
China Machine Press

计 学 丛 书

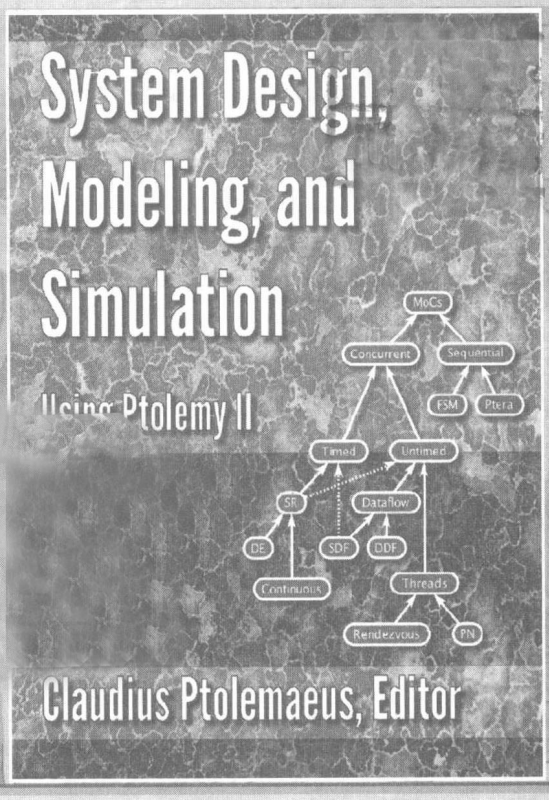
信息物理融合系统（CPS）设计、建模与仿真

基于Ptolemy II平台

[美] 爱德华·阿什福德·李（Edward Ashford Lee） 等编著
吴迪 李仁发 译

System Design, Modeling, and Simulation

Using Ptolemy II



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

信息物理融合系统（CPS）设计、建模与仿真——基于 Ptolemy II 平台 /（美）爱德华·阿什福德·李（Edward Ashford Lee）等编著；吴迪，李仁发译．—北京：机械工业出版社，2017.1
（计算机科学丛书）
书名原文：System Design, Modeling, and Simulation: Using Ptolemy II

ISBN 978-7-111-55843-9

I. 信… II. ① 爱… ② 吴… ③ 李… III. 异构网络－研究 IV. TP393.02

中国版本图书馆 CIP 数据核字（2017）第 012647 号

本书版权登记号：图字：01-2015-7582

本书是一本系统论述 CPS（集成了计算、网络和物理过程的信息物理融合系统）建模问题的专著，以 Ptolemy II 平台为基础，广泛讨论了分层、异构系统的设计、建模和仿真技术。本书共分为三部分：第一部分包括第 1 ～ 2 章，主要介绍系统的设计、建模与仿真的基础概念；第二部分包括第 3 ～ 11 章，涵盖系统设计、建模和仿真中常用的计算模型，其中每章都包括一个或一小类相关的计算模型，并解释它们如何工作、怎样使用它们建立模型以及哪些种类的模型与计算模型可以更好地匹配；第三部分包括第 12 ～ 17 章，重点介绍由 Ptolemy II 提供的模型系统的内部组件，讨论 Ptolemy II 跨模型计算的能力，对于那些想要扩展 Ptolemy II 或者想用 Java 写自己的角色的读者来说，可以从中得到具体指导。本书最后列出了大量的参考书目。

本书适合作为高等院校相关专业“嵌入式系统”课程的教材或教学参考书，也可作为专业技术人员在 CPS 系统建模过程中的参考书。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码 100037）
责任编辑：盛思源　　责任校对：董纪丽
印　　刷：北京诚信伟业印刷有限公司　　版　　次：2017 年 2 月第 1 版第 1 次印刷
开　　本：185mm×260mm　1/16　　印　　张：24.25
书　　号：ISBN 978-7-111-55843-9　　定　　价：79.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自 1998 年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum，Bjarne Stroustrup，Brian W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com
电子邮件：hzjsj@hzbook.com
联系电话：（010）88379604
联系地址：北京市西城区百万庄南街 1 号
邮政编码：100037

华章科技图书出版中心

译者序

System Design, Modeling, and Simulation using Ptolemy II

本书是Ptolemy II项目近20年成果经验的总结，也是项目组成员共同智慧的结晶。Ptolemy II项目是由美国加州大学伯克利分校教授、嵌入式系统领域的著名学者、CPS（Cyber-Physical System）研究的倡导者和引领者Edward Ashford Lee负责的。他长期专注于并发、实时嵌入式系统领域的研究，开发了集系统设计、建模及仿真于一体的Ptolemy II系统平台。

为了避免讨论过于抽象，本书以Ptolemy II为基础，广泛讨论了分层、异构系统的设计、建模和仿真技术。本书共分为三部分。第一部分包括第1～2章，主要介绍系统的设计、建模与仿真的基础概念。第1章首先概述了异构系统规范化建模的指导性原则，从较高的角度对第二部分详细描述的计算模型（Model of Computation，MoC）进行概述。第2章提供了一个通过图形用户界面Vergil使用Ptolemy II的操作指南，使读者可以在系统设计过程中熟练掌握开源的Ptolemy II来进行实验。第二部分包括第3～11章，涵盖了系统设计、建模和仿真中常用的计算模型。每章都包括一个或一小类相关的计算模型，并解释了它们如何工作、怎样使用它们建立模型以及哪些种类的模型与计算模型可以更好地匹配。第三部分包括第12～17章，重点介绍由Ptolemy II提供的模型系统的内部组件，讨论了Ptolemy II跨模型计算的能力。对于那些想要扩展Ptolemy II或者想用Java写自己的角色的读者来说，可以从第三部分得到具体指导。本书最后列出了大量的参考文献。另外，书中提到所有的方法、实例都可以在项目网站（http://ptolemy.org/systems）上下载开源代码。

本书的最初译稿来源于李仁发教授组织的两届“高性能嵌入式计算”研讨班所用教材之一。李仁发教授组织了本书的翻译工作。参加研讨班的全体博士生、硕士生为本书的共同译者，他们是：吴武飞、杜家宜、白洋、谌雨晴、黄雪伦、王娜、胡游、周兰花、黄一智、李坤明、黄晶、屠晓涵、马萌、彭理、何少芳、宋金林、袁娜、邓湘军、周佳、李万里和杨竞。吴迪、吴武飞负责全书的校对。吴迪负责全书的最后校审工作。

不同语言之间的转换是一件困难的事情。看似很直白的一个词，虽然理解其词义，但要换一种语言表达时往往煞费苦心，有的名词还很难给出确切的中文译名。译者力求忠实地表达书中所介绍的技术，保持原作者的行文风格。在本书翻译过程中，发现原书存在少量描述性错误，通过邮件与原作者确认核实后，在译文中直接进行了修订。限于时间以及译者水平和经验的不足，译文中难免存在不当之处，恳请读者批评指正。

本书的翻译工作得到了原书作者Edward Ashford Lee教授本人及Ptolemy项目组其他成员的鼎力支持，同时还得到湖南大学嵌入式与网络计算湖南省重点实验室同仁及机械工业出版社许多人士的帮助。对此，译者深表感谢。

错误反馈

如果你在阅读过程中发现本书中的错误或印刷错误，或者有任何改进的建议，请发送电子邮件到译者邮箱：enchnu@126.com。或者发送到原书作者邮箱：authors@leeseshia.org。

在邮件中，请注明该书的版本号和相关页面，非常感谢！

译者

“我”上次发表著作是在一千九百年前[⊖]。“我”很高兴从退休中复出，对以本人名字命名的工程（Ptolemy 工程）发表自己的看法。与“我”以往在天文和地理方面的工作相似，该项目也是对复杂系统进行处理。值得一提的是，类似“我”之前的许多著作，本书同样凝结了许多人共同的智慧和努力。

“我”以前在《The Almagest》（天文大全）中研究行星、太阳、地球和月亮的运动规律，这些运动都是**并发交互过程**（concurrent interacting process）。并且这些运动都是*确定性的*（deterministic），并不以神的意志为转移。这些模型的关注点不仅仅是对所观察到的行为进行精确匹配，更重要的是对行为的预测。类似地，Ptolemy 项目研究并发交互过程，并重点关注确定性模型。

理想情况下，求知欲推动着人类从迷信和盲目的信仰发展到逻辑和计量。现在所谓的“科学”深深根植于科学方法（scientific method），特别是在自然系统的研究中。利用科学方法，从设想开始，设计实验，并基于实验来对之前的设想下定论。当然，为了能够进行计量，待测量的工件或过程必须以某种形式存在。在“我”早期的研究中，不存在该问题，因为太阳、地球、月亮和行星是已经存在的事物。然而工程学科所关注的是人为的工件和过程，研究的是自然界中*本不存在的系统*。即便如此，科学方法也可用于并已经应用于工程设计中。工程师构建仿真和原型系统，将设想公式化，然后通过实验来进行设想的测试。

因为针对的是本不存在的工件和过程，所以工程设计不能单单基于科学方法。实验的目的是提高对所设计的工件、过程的认知。但是在进行实验前，必须将这些工件或者过程创造出来。在认识某些事物之前，不得不先把它们创造出来，这点注定了我们的设计会根植于“迷信”和盲目的信仰。

模型构造是与科学方法互补的重要科学部分。模型是物理现实的一种抽象，并且模型提供内视和行为预测的能力可以形成设想的核心思想，该思想核心等待被实验证实或证伪。建模本身更应归于工程学科，而非自然科学。从根本上讲，它并不是对于自然界已存在系统的研究。相反，它是人类主导的、对于自然界本身不存在事物的建造过程。一个模型本身就是一项工程。

好的模型甚至可以减少对计量的需求，因此可以减少对科学方法的依赖。比如，我们一旦有一个行星运动模型，我们就可以精确预测它们的位置，这样就减少了对其位置测量的需要。计量的角色从确定行星位置转变为改善它们的运动模型以及检测模型对运动的影响（工程上称为“**故障检测**”（fault detection））。

无论在自然科学还是在工程中，模型都可以通过迭代方法来进行优化。“我”提出的以地球为中心的宇宙模型需要很多次迭代来修正，以逼近实验观测到的行星运动情况。最终模型的预测能力让“我”引以为豪。并且，基于这些模型的预测方法可以通过星盘机械化，这点同样让“我”感到自豪。即便这样，不得不承认，令人尊敬的同行哥白尼（Nicolaus

⊖ 此处作者把自己比喻为古希腊天文学家、地心说体系创立者 Claudius Ptolemy。——译者注

Copernicus）为行星运动提出了一个更好的模型（日心学说）。他的模型从概念上讲是更简单的。这是一种概念上的飞跃：我们可观测到的宇宙的中心，即我们所在的大地，并非一定是宇宙模型的中心。更进一步说，相对于物理世界，对于模型我们有更大的自由度，因为模型不需要被自然界所限制。即便如此，“我”所建立的模型在将近 1400 年的时间里也是一流的。

Ptolemy 项目确实是一项关注系统模型的研究。但是，该系统与“我”之前关注的系统有很大的不同。之前的那些系统都是自然界提供的，但是本书中的系统都是人造的。在本书中，建模的目的是优化系统，我们不可能对自然界给予的行星系统做任何的优化。

简而言之，在与科学相反的工程中，模型要在被建模系统的设计阶段发挥作用。与科学一样，工程中的模型是可以被优化的，但是与科学不同的是，工程中的系统还可以被模型化。

更有趣的是，与科学不同的是，在工程中模型的选择对被建模的系统是有影响的。给予相同的目标，两位工程师可能会得出截然不同的系统设计和实现方案，这仅仅是因为他们在开始阶段使用了截然不同的系统模型。进一步说，若两位工程师提出了不同的模型，其原因可能仅仅是他们在开始阶段使用了不同的工具来构建模型。一位用纸和笔建模的工程师与一位用软件工具建模的工程师得出的模型可能很不一样。结果就是，他们很可能得出迥异的系统设计。

针对复杂系统，本书收集了非常丰富的建模工具和技术。它们中的一些毫无疑问在以后会被优化，正如“我”自己提出的本轮（epicycle）模型，其建模的复杂性被哥白尼学派证明为不必要的。即使如此，本书的目的是向工程师提供目前可用的最好的建模技术。可以确信的是，我们将做得更好。

如何使用本书

本书是为需要对各种系统建模的工程师和科学家，以及想了解如何为复杂、异构系统建模的人而编写的。这些系统包括机械系统、电气系统、控制系统、生物系统等，更有趣的是，还包括结合了这些领域或者其他领域元素的异构系统。本书假设读者熟悉仿真和建模工具及其技术，但不要求对这些内容有深厚的背景知识。

本书重点强调 Ptolemy II 中已实现的建模技术。Ptolemy II 是一个开源的仿真和建模工具，用于对系统设计技术进行实验，尤其是那些涉及各种不同模型组合的系统。它是由 UC Berkeley 的研究人员开发的，并且由于过去 20 年里世界各地研究者的努力，它逐渐演变成一个复杂而精巧的工具。本书基于 Ptolemy II，对分层、异构系统的系统设计、建模和仿真技术进行了广泛的讨论。同时本书使用 Ptolemy II 来避免这些讨论过于抽象化和理论化。所有这些技术都由精心设计且测试效果良好的软件实现来支持。关于 Ptolemy II 更详细的底层软件架构以及更为细节的操作和基础理论，可以在知识点、参考文献和网络链接中找到。

本书共分 3 个部分。第一部分是“入门”。第 1 章概述了本书所涵盖的建模方式所蕴含的准则，并简要概述了多种**计算模型**（Model of Computation，MoC）。第 2 章介绍了怎样通过图形编辑器 Vergil 使用 Ptolemy II。对于那些想直接开始建模的读者，该章是个很好的起点。

第二部分包括第 3 ～ 11 章，涵盖了几乎所有的计算模型。每一章都包括一个或者一小

类相关的计算模型，并解释了它们怎样工作、怎样使用它们建立模型以及哪些种类的模型与计算模型可以比较好地匹配。

第三部分讨论了 Ptolemy II 计算模型的可扩展性。对于那些想要扩展 Ptolemy II 或者想用 Java 写自己的角色（actor）的读者来说，第 12 章或许是最重要的一章，它描述了 Ptolemy II 软件架构。Ptolemy 是开源软件，并有完善的代码文档可供阅读。对于想要阅读代码并在此基础上做些工作的读者来说，该章可以提供很好的指引。第 13 章描述了用于规格化模型参数值和向角色（actor）中添加自定义函数的表达式语言。第 17 章描述了 Ptolemy II 标准库中包含的**信号绘图仪**（signal plotter）的功能。第 14 章讲解了 Ptolemy II 中的**类型系统**（type system）。Ptolemy II 是一个复杂的类型系统，当提供一个强调类型系统来使安全最大化时，其设计旨在把建模工具的负担最小化（通过强调类型推断而不是类型声明）。第 15 章讲述**本体**（ontology）。本体能将单元部件、尺寸和概念与模型中的数值相关联，它增强了类型系统。同样，重点在于推断和安全。最后，第 16 章描述了 Ptolemy II 中的 Web 界面。具体地说，它解释了从模型中导出页面以及在模型中建立 Web 服务和服务器的功能。

致谢

本书在 Ptolemy 项目中描述的建模技术，经过了 UC Berkeley 项目成员的多年开发。根源可追溯到 20 世纪 80 年代。雏形来自 Messerschmitt（1984）创建的名为 Blosim（Block Simulator）的软件框架，用于仿真信号处理系统。Messerchmitt 的博士生 Edward A. Lee，受 Blosim 的鼓舞，开发了同步数据流（Synchronous DataFlow，SDF）计算模型（Lee，1986；Lee and Messerschmitt，1987b）和针对该模型的调度方法（Lee and Messerschmitt，1987a）。Lee 和他的学生接着完善了一个基于 Lisp 的软件工具 Gabriel（Lee et al.，1989），通过它开发和完善了 SDF 计算模型。在 20 世纪 90 年代初期，Lee 和 Messerschmitt 开发的面向对象的方框图框架开始称为 Ptolemy（Buck et al.，1994）(现在称为 Ptolemy Classic)。在 20 世纪 90 年代后期，Lee 和他的小组基于最新的编程语言 Java（Eker et al.，2003），开始了一个称为 Ptolemy II 的全新设计。Ptolemy II 发展了面向角色设计（Lee et al.，2003）和分层异构的主要思想。

软件的发展趋势在许多方面都反映了当前计算的演变。Blosim 用 C 语言编写。Gabriel 用 Lisp 语言编写。第一代 Ptolemy 系统，现在称为 Ptolemy Classic，用 C++ 编写。下一个版本 Ptolemy II，用 Java 编写。每一次变化都反映了人们利用最有效的技术解决实际设计问题的努力。Lisp 超过了 C 语言是因为其鲁棒性以及对复杂逻辑设计的适应性。C++ 超过了 Lisp 是因为其能更好地发展（当时）面向对象设计的概念（特别是继承和多态）。Java 超过了 C++ 是因为其很好地支持了多线程和用户接口的可移植性。也许最重要的是，选择定期更换语言是为了强制这个小组重新进行设计，并扩充知识学习。

本书很大程度建立在基于 Java 的 Ptolemy II 上。即便如此，大部分的荣誉都应归功于 Ptolemy Classic（Buck et al.，1994）的设计者，尤其是 Joseph Buck、Soonhoi Ha、Edward A. Lee 和 David Messerschmitt 等。

参与人员

许多人都对本书和 Ptolemy II 软件做出了很大的贡献。除了每章的作者外，还有很多人对本书贡献很大，他们包括 Shuvra S. Bhattacharyya、David Broman、Adam Cataldo、

Chihhong Patrick Cheng、Daniel Lazaro Cuadrado、Johan Eker、Brian L. Evans、Teale Fristoe、Chamberlain Fong、Shanna-Shaye Forbes、Edwin E. Goei、Jeff Jensen、Bart Kienhuis、Rowland R. Johnson、Soonhoi Ha、Asawaree Kalavade、Phil Lapsley、BilungLee、Seungjun Lee、Isaac Liu、Eleftherios D. Matsikoudis、Praveen K. Murthy、Hiren Patel、José Luis Pino、Jan Reineke、Sonia Sachs、Farhana Sheikh、Sun-Inn Shih、Gilbert C. Sih、Sean Simmons、S. Sriram、Richard S. Stevens、Juergen Teich、Neil E. Turner、Jeffrey C. Tsay、Brian K. Vogel、Kennard D. White、Martin Wiggers、Michael C. Williamson、Michael Wirthlin、Zoltan Kemenczy、Ye（Rachel）Zhou 和 Jia Zou 等。Christopher Brooks 是该软件的总负责人，因此软件拥有如此高的质量，他功不可没。其他参与者详见 http://ptolemy.org/people。

特别感谢 Jennifer White 提供全面的编辑帮助，他帮助提高了全书的编写质量。为本书的编辑提供帮助的人还有 Yishai Feldman、William Lucas、Aviral Shrivastava、Ben Zhang 和 Michael Zimmer 等。

第三部分　建模的基础结构

第一部分

System Design, Modeling, and Simulation using Ptolemy II

入　门

本书第一部分主要介绍系统的设计、建模与仿真。第 1 章首先概述了异构系统规范化建模的指导性原则，从较高的角度对将在第二部分中详细描述的计算模型（Model of Computation，MoC）进行概述。另外，第 1 章还提供了一个高度简化的研究案例（一个发电机组），该案例阐明了多种不同计算模型在复杂系统设计中所起的作用。

第 2 章提供了一个利用图形用户界面 Vergil 使用 Ptolemy II 的操作指南。本书目标之一就是使得读者能够在系统设计过程中利用开源的 Ptolemy II 进行实验。这章的目的在于提供足够的信息，以使读者成为 Ptolemy II 的合格使用者。关于如何对 Ptolemy II 进行扩展，读者可参考本书第三部分。

第 1 章

System Design, Modeling, and Simulation using Ptolemy II

异构建模

Edward A. Lee

当前的许多工程系统通常都结合了异构且复杂的子系统。例如一辆汽车，就可能结合了一个复杂的发动机、很多的**电子控制单元**（Electronic Control Unit，ECU）、引擎控制系统、车身电子控制系统（用于控制车窗和门锁）、娱乐系统、空调控制和通风系统，以及各种安全子系统（如安全气囊）。每个子系统可能又由软件、电子及机械部分联合组成。实现如此复杂的系统的确是一项挑战，尤其在于：即使最小的子系统也跨越了多个工程学科领域。

这些复杂系统同样也对设计工具带来了挑战。工程师通过使用设计工具对系统进行规格化、设计、仿真及分析。如今，仅仅是画出机械结构的草图，然后列出一些等式描述机械部分之间的交互是不够的。而且，无论是完全依靠机械部分的 3D 建模软件工具，还是依靠用于软件系统的基于模型的设计工具，都是不够的。各个领域（机械、软件、电子、通信网络、化学、流体动力学以及人为因素）之间相互联系的复杂性削弱了只适用于单个领域工具的有效性。

本书重点针对**信息物理融合系统**（Cyber-Physical System，CPS）（Lee，2008a，2010a；Lee and Seshia，2011），这种系统将物理动力学与计算和网络结合。CPS 需要通过模型组合的方式将物理过程的连续动态（通常用微分方程描述）与软件模型集成在一起。有些应用需要结合组件间的计时交互和传统算法计算[⊖]来实现，这种情况下混合模型是最有效的。在那些算法组件间有并发[⊖]交互（concurrent interaction）的传统软件系统中，也可以使用混合模型。

补充阅读：关于术语“CPS”

术语“CPS”（Cyber-Physical System）出现于 2006 年前后，是由美国国家科学基金会的 Helen Gill 所提出。对于 William Gibson 的“**赛博空间**”（cyberspace）一词，我们都很熟悉。Gibson 在小说《Neuromancer》（神经漫游者）中用这个词来表示用于人类之间相互通信的计算机网络媒介。我们试图将术语赛博空间与 CPS 联系起来，但是术语 CPS 的根源更深远。或许，把 cyberspace 和 CPS 视为由同一词语“**控制论**”（cybernetics）演

⊖ **算法**指的是对解决某个问题所需的有限步骤序列描述。物理过程很少呈现如步骤序列那样的结构；相反，它们的结构组织类似于并发组件（concurrent component）间的连续交互。

⊖ **并发**，由拉丁文中“concurrere”一词而来，意味着同时发生；它在计算机科学中是指两个或多个步骤序列的任意交错。但是，这只是对基本概念的专业解释。本书认为“并发”这个概念是指同步操作，而并不表示交错或者一个步骤序列。特别地，两个连续过程可以并发操作，而并不要求它们被直接表示成一系列步骤。比如，放入一桶水中的一个电阻加热元器件。增大通过这个加热元器件的电流，会使水温上升。电流与水温都是连续过程，并且这两个过程相互影响。但是，这两种过程中的任一种都无法合理地表示为一系列步骤，并且这个总体过程也不是这些系列步骤的交错。

变出的同源词会更准确一些。

术语“控制论”由美国数学家 Norbert Wiener（Wiener，1948）提出，他对控制系统理论的发展有巨大影响。在第二次世界大战期间，Wiener 发明了高射炮的自动瞄准和射击技术。虽然他使用的机制并不涉及数字计算机，但涉及的原理与现今很多基于计算机的反馈控制系统相似。Wiener 发明的这个词起源于希腊语 κυβερνητης（kybernetes），意思是舵手、长官、领航员或船舵。该比喻对控制系统来说是很恰当的。

Wiener 将他对“控制论”的理解描述为控制和通信的结合。他对控制的概念根植于闭环反馈（closed-loop feedback），这里的控制逻辑由对物理过程的计量结果驱动，然后控制逻辑反过来也驱动物理过程。即使 Wiener 没有使用数字计算机，但控制逻辑从效果上来讲就是一种计算，因此“控制论”就是物理过程、计算和通信三者的结合。

Wiener 当时没有料到数字计算和网络的巨大影响力。“Cyber-Physical System”可能被模糊地解释为“赛博空间”和物理过程的结合，这一点削弱了 CPS 可能具有的巨大影响力。CPS 对信息技术带来的显著影响，甚至超越了 Wiener 时代最疯狂的想象。

1.1 语法、语义、语用

在撰写本书的过程中，工程工具和技术正处在巨大变化时期。这种变化推动着我们的工作，使我们有能力适应系统日益增加的复杂性和异构性。过去，整个行业都围绕着为单一的工程领域提供设计工具，比如数字电路、软件、3D 机械设计、采暖及通风领域。如今，我们看到越来越多设计工具的整合和组合；独立工具常常扩展到工具套件，并提供传统领域以外的功能。这种变革也带来了一些问题，即不良好的集成会导致一些不可预期的行为。工具集成常常导致怪异集成（frankenware）[⊖]，也就是说，由几乎不兼容的工具组成的不稳定组合很难得到有效维护和使用。

另外，在一个相对狭窄的领域中一贯良好的工具，在更广泛的领域中却并非那么有效。今天复杂的设计工具涉及**语法**（syntax）（如何表示一个设计）、**语义**（semantics）（一个设计表示什么，以及它是如何工作的）、**语用**（pragmatics）（Fuhrmann and von Hanxleden，2008）（工程师应该如何使设计可视化，并对其进行编辑和分析）的复杂组合，如图 1-1 所示。当一个设计工具被用于它的原始使用领域之外时，或者用于和其他工具的组合之中时，不兼容的语法、未充分理解的语义以及不一致的人机界面都有可能使得其不能有效使用。

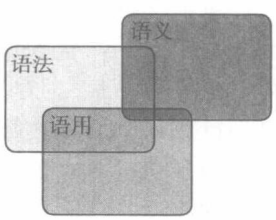


图 1-1 如今的设计工具涉及语法、语义和语用的复杂组合

会出现语法上的不兼容，是因为不同设计结构的本质是不同的（比如，软件的语法和 3D 设计几乎没有共同之处）。但是出现语法不兼容的一个更普遍的原因是，各种工具是在不同的工程领域以不同的技术开发的。以工具的语用为例，对于如何管理设计文件，如何追踪改动也会因不同发生时刻而不同。语义的差异也有一定偶然性，不同的理解会加大这种差异。语用在不同领域中可能具有不同意思。比如说，在控制工程师和软件工程师眼中，同一个框图可能代表着完全不同的意思。

⊖ 术语“frankenware”源于 Christopher Brooks。

本书使用 Ptolemy II 对异构建模的几个关键概念进行了检验。Ptolemy II[⊖]是一个开源的建模和仿真工具。与大多数其他设计工具不同，Ptolemy II 从开发之始就专注于异构系统。Ptolemy 项目（UC Berkeley 正在进行的一项研究）的一个关键目标就是将不同领域之间语法、语义和语用之间的差异最小化，并将不同领域设计之间的互操作性最大化。因此，Ptolemy II 为 CPS 系统的设计提供了一个有数的实验环境。

Ptolemy II 集成 4 种不同类型的语法：框图、弧线图（bubble-and-arc）图、命令式程序和算术表达式。这些语法是互补的，这使得 Ptolemy II 能够处理各种设计领域的问题。框图用来表示相互通信的组件之间的并发关系；图用来表示状态或模式的顺序；命令式程序用来表示算法；算术表达式用来表示函数的数值计算。

Ptolemy II 也集成了一些语义域。尤其是对于框图来说，它的语义有多种可能，彼此都有明显的不同。框图之间的连接表示设计中组件之间的交互（interaction）。但是什么类型的交互呢？是异步通信（如寄信）？是会话形式的通信（如打电话）？还是数据的定时更新（如在同步数字电路中那样）？在交互中，时间是否起到了作用？交互是离散的还是连续的？为了支持异构建模，Ptolemy II 支持以上提到的所有需求，并且它还可以被扩展以支持更多的需求。

1.2　域和计算模型

Ptolemy II 中的**语义域**（semantic domain），通常称为**域**（domain），它定义了设计中两个组件交互的“物理定律”。它为组件之间的并发执行以及两个组件之间的通信（如前文所述）提供了管理规则。这种规则的集合称为**计算模型**（Model of Computation，MoC）。在本书中，从技术上看尽管域是计算模型的实现，但术语“计算模型”和“域”是可替换的。计算模型是一个抽象模型，然而域是模型在软件上的具体实现。

模型规则分为三类。第一类规则指定了组件的构成要素，在本书中，一个组件一般是一个**角色**（actor），在下文中将给予更精确地定义。第二类规则指定执行和并发机制：角色调用是按序的？同时的？还是非确定性的？第三类规则指定通信机制：角色之间怎样交换数据？

本书中讨论的每一个计算模型都有很多可能的变体，这些变体中很多已经在其他的建模工具中实现了。本书把重点放在 Ptolemy II 中实现的计算模型，以及那些具有易读且书写良好的语义模型上[⊜]。为了进一步阐述，我们也提供了其他一些有用的、还未在 Ptolemy II 中实现但已在其他工具中实现的计算模型的简要说明和索引。

为了支持异构系统的设计，Ptolemy II 域之间可以交互操作。这要求语义域之间有一定程度的协议。但是，当不同的工具被分别独立设计再组合到一起时，这种协议几乎是不存在的。Ptolemy II 中域之间交互的法则在多篇论文中有所描述（Eker et al.，2003；Lee et al.，2003；Goderis et al.，2009；Lee，2010b；Tripakis et al.，2013）。本书重点在于域的互操作性的实践环节，而不是理论。

使用统一的、一致的软件系统使我们可以专注于域的交互操作，而不必过多担心不同工具集成过程带来的不兼容性问题。比如说，Ptolemy II 的类型系统（type system）（它

⊖ Ptolemy II 可从 http://ptolemy.org 下载。

⊜ 在本书的电子版中，大多数模型插图的图注都提供了超链接，你可以在线浏览这些模型。若你的机器支持 Java，你还可以编辑这些模型并执行它们。