第 35 卷 第 19 期 计 算 机 工 程 2009 年 10 月
Vol.35 No.19 Computer Engineering October 2009
·安全技术· 文章编号：1000—3428(2009)19—0138—03 文献标识码：A 中图分类号：TP393

基于 MDA 和可执行形式化的可信软件设计

唐 艳[1]，杜玉越[1,2]，刘 伟[1]
(1. 山东科技大学信息科学与工程学院，青岛 266510；2. 中国科学院软件研究所计算机科学国家重点实验室，北京 100080)

摘 要：提出基于模型驱动架构的软件开发过程，利用可执行形式化规范，有效提高软件开发效率和可测试性，并通过基于可执行规范的运行时监控技术保证系统行为的可信性，降低由于软件测试阶段遗留的错误以及系统受到非法入侵所带来的风险。
关键词：可执行形式化；Petri 网；可信软件体系；模型驱动的体系结构

Design of Trusted Software Based on MDA and Executable Formalization

TANG Yan[1], DU Yu-yue[1,2], LIU Wei[1]
(1. College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510;
2. State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100080)

【Abstract】 This paper brings forward the software development process based on Model Driven Architecture(MDA) and improves efficiency & testability of software with executable formalization specification. By way of run-time monitoring technology, the credibility of system is guaranteed and the errors created by test stage and the risk of system unlawful invasion are reduced and eliminated.
【Key words】 executable formalization; Petri net; trusted software architecture; Model Driven Architecture(MDA)

1 概述

国际对象管理组织(Object Management Group, OMG)的最新战略是建立模型驱动体系结构(Model Driven Architecture, MDA)。MDA 支持软件设计和模型的可视化、存储和交换，它和模型驱动的开发(Model Driven Development, MDD)是当前软件研究和开发的新领域。基于 MDA 的软件开发方法是未来软件开发的必然趋势，将形式化方法、可执行模型技术引入 MDA 方法的模型构造活动中，是当前研究的一个热点。

统一建模语言(UML)是 MDA 的核心技术之一，平台无关模型(PIM)及特定平台模型(PSM)都是通过 UML 模型表示的。UML 形式化程度的不足导致 PIM 及 PSM 不能很好地仿真、验证系统的设计结果。文献[1]提出利用可执行模型技术解决建模的这些问题。

目前有 90%的软件开发项目采用了 UML 作为描述语言。在基于 UML 的开发过程中，主要用多种 UML 视图表达和分析系统，用对象约束语言 OCL 表示对象的功能和约束。但 OCL 的表达能力不够强，无法用来表达模型的动态语义，所以，许多研究机构进一步致力于 UML 建模的形式化支持[2]。此外，多数 UML 形式化方面的工作及支持工具，都是关注特定视图的模型(如类模型、状态图和顺序图)，以及将它们转换成现有的形式语言[3]。

因此，将基于 Petri 网、UML、时序逻辑及工作流分析技术的可执行形式化规范描述语言，作为 UML 描述方法的补充，用来实现基于 PIM 概念模型、PSM 概念模型、代码模型以及运行时对象模型等多个抽象级别的可执行形式化规范描述，在不同抽象级别通过关键业务功能规范的模拟执行，达到验证模型准确性、有效性，以及系统运行时行为可靠性、可信性的目的。

一个可信实体(包括组件、系统或过程)的行为在任意操作条件下是可预测的，并能很好地抵抗应用程序、病毒以及一定的物理干扰造成的破坏[4]。

2 基于 MDA 的可信软件开发

2.1 MDA 的开发过程

图 1 给出了基于 MDA 的软件开发过程。

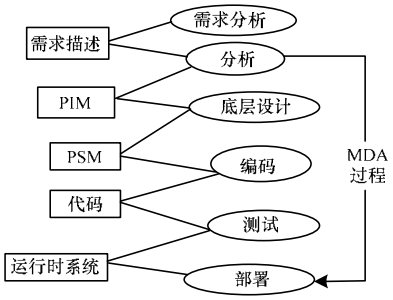

图 1 基于 MDA 的软件开发过程

由于目前 MDA 技术尚不成熟，可执行的 UML 模型的研究多侧重于从模型生成代码，往往对目标编程语言具有一定

基金项目：国家自然科学基金资助项目(60773034)；国家“973”计划基金资助项目(2004CB318001-03)；中国科学院计算机科学国家重点实验室开放课题基金资助项目(SYSKF0804)；山东省“泰山学者”建设工程专项基金资助项目
作者简介：唐 艳(1982－)，女，硕士研究生，主研方向：工作流，可信软件，模型驱动体系结构，Petri 网理论与应用；杜玉越，教授、博士、博士生导师；刘 伟，讲师
收稿日期：2009-06-04 **E-mail：**tangyan_0621@126.com

的依赖性。而软件的形式化验证需要的往往不是全方位的可执行，不需要模拟系统的全部行为，不需要为生成目标系统构造全部的行为规范细节，它侧重于针对特定需求约束进行有针对性的验证。所以，有必要针对形式化验证规范的需要，构造一种平台独立的简化可执行形式化规范描述语言。

2.2 可执行形式化规范

可执行形式化规范包括 Petri 网、规范描述语言及其解释推理引擎 3 个组成部分。

Petri 网具有直观的图形表现形式及丰富的模型验证和分析手段。

规范描述语言是一种依托领域需求定义的、有一定通用性的、可以解释执行的甚高级语言，具有基本控制结构及领域数据结构的描述能力，可以对需求分析过程中形成的业务功能、性能需求进行精确的描述。

利用解释推理引擎可以对 Petri 网及规范描述语言描述的业务规范模拟执行和分析推理。在系统开发阶段，其结果可以作为模型验证及系统验收的依据，在运行时期可以进行系统运行时的行为监管和监控。

3 基于 MDA 形式化规范的可信软件开发

3.1 基本思想

如图 2 所示，该可信软件系统是基于 PIM、基于 PSM 和基于运行时对象模型的可执行规范说明。三者之间形成一种精化关系，基于 UML 模型元素之间的精化关系形成一种跟踪关系，规范验证机制主要通过规范的模拟执行来验证各阶段模型与代码的完整性、一致性和有效性。运行时监控机制主要通过采集运行时系统的实时数据，利用可执行规范模拟运行，根据模拟运行情况推断系统行为是否符合预定规范的约束。如果发现异常，则给出相应检测监控报告，或进行适当干预。还利用模板、参数化等技术来提高领域规范的可复用性。

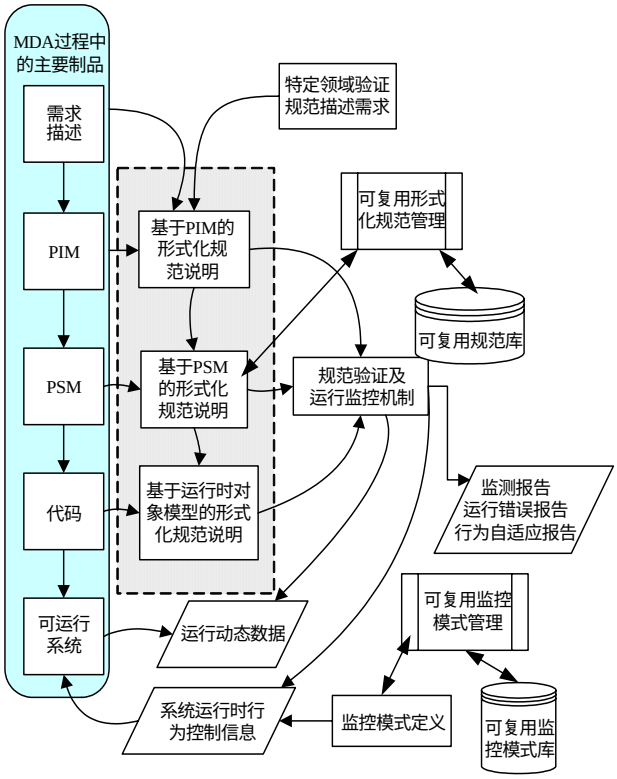

图 2 可执行形式化规范与 MDA 制品之间的关系

3.2 具体设计

将可以自动生成代码的形式化描述，统称为可执行形式化规范描述语言。

将形式化规范与 UML 自身的描述机制(如 OCL、状态图、动作语义等)相结合的方式描述系统模型。在 UML 模型基础上，对系统关键行为加以约束，并利用可执行形式化规范描述语言进行描述。由于 UML 本身缺乏严格的形式化基础，尤其是动态行为建模方面，这将导致 MDA 行为模型描述的不足和困难，因此该可信系统是基于可执行 Petri 网描述规范的 UML 模型检查工具，对 UML 模型的关键部分进行完整性、一致性、有效性检查。如图 3 所示，针对 MDA 中的 PIM, PSM 模块，增加了可执行形式化规范说明，对代码模块增加了运行时的形式化规范说明。其中，需求描述模块多数是用文本描述的，PIM 和 PSM 模块采用 UML 图形(如类图、对象图、顺序图等)描述，增强了 PIM/PSM 的描述精度和能力及系统的可信。

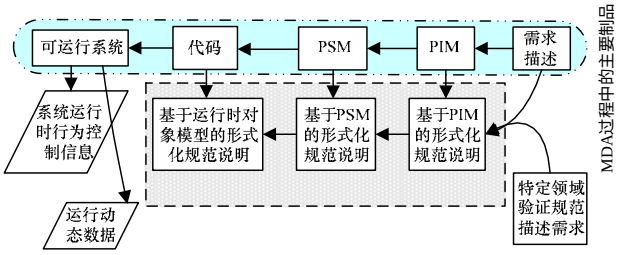

图 3 MDA 过程的形式化规范说明

如图 4 所示，在形式化规范说明的基础上添加了系统运行时控制信息模块、规范验证及运行监控机制等模块，实现行为控制和规范验证，从而实现系统运行时的可信。

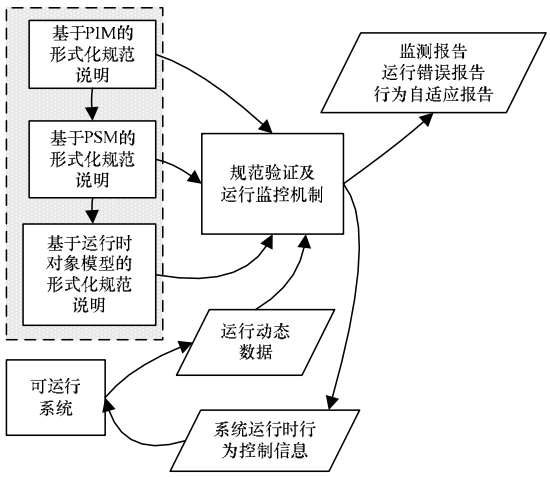

图 4 对可执行规范的验证及监控

对可执行规范的验证及监控包括如何收集形式化规范所需数据、如何利用采集的运行时数据进行可执行形式化规范的模拟执行、如何识别系统运行时行为的合法性和有效性，以及发现行为异常时监控系统应采取的处理策略和方法。可以依据跟踪关系矩阵及相应运行时数据接口来采集验证规范所需数据，通过可执行规范及系统约束的模拟运行情况，识别系统的当前运行状况及行为情况，并进一步判断是否与预期行为相符，从而识别出系统运行时行为的合法性和有效性。如果出现异常情况，可根据预定监测模式，利用实时数据启动基于可执行的形式化规范 PSM 模型验证及源代码验证，初

步判断可能的原因，例如设计逻辑错误、编码错误、怀疑受到攻击等。然后，根据预定监控处理模式采取相应的处理措施，例如记录日志及运行环境数据、给出相应警告信息、中止系统当前动作、终止当前任务，甚至终止整个系统的运行。

由于特定领域的规范描述是可以复用的，因此利用模板、参数化等技术来提高领域规范的可复用性，对可复用规范进行管理。如图 5 所示，对可执行规范描述应用领域规范复用，将一些成熟的软件复用技术及复用库管理技术应用于形式化规范库的复用过程中，可以有效提高定义形式化规范说明的效率，改善形式化规范说明的应用效果。利用监控模式、处理模式库对监控处理机制进行复用，系统控制监控处理模式对运行时和测试阶段进行管理和监控来实现系统的复用性和可信。

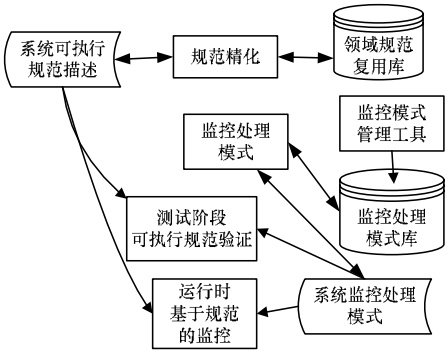

图 5　可信系统中的可复用

4　结束语

本文用可执行形式化规范与目前的 MDA 主流技术有效结合，基于可执行规范的软件验证及监控工具包，有效弥补目前主流基于 MDA 技术的软件开发工具的不足以及 UML 目前形式化程度不足带来的缺陷，有效提高软件运行时行为的可信性。通过模拟执行发现建模早期的错误，可以降低软件开发期间的成本。基于可执行形式化规范进行的测试，可以提高验收测试的效率，降低验收测试的成本，进而获得较大的经济效益。本文给出并论述了基于 MDA 和可执行形式化规范的可信软件体系设计方法的基本思想，更细致具体的实施过程将是笔者进一步的研究课题。

参考文献

[1] Raistrick C, Francis P, Wright J. Model Driven Architecture with Executable UML[M]. Cambridge, UK: Cambridge University Press, 2004.

[2] Dale P, Murray D J, Yu Chen. Object-oriented Design Patterns for Debugging Heterogeneous Languages and Virtual Machines[J]. 2005, 35(3): 255-279.

[3] Yeung W L. Checking Consistency Between UML Class and State Models Based on CSP and B[J]. Journal of Universal Computer Science, 2004, 10(11): 1540-1558.

[4] 周明天，谭　良. 可信计算及其进展[J]. 电子科技大学学报, 2006, 35(4): 686-696.

编辑　顾逸斐

（上接第 137 页）

件的新建、读、写、重命名、删除、防打印、防截屏、防拷贝等。

(3)客户端中断重连测试。系统工作过程中当客户端与服务器由于网络故障而中断，客户端应能够在网络恢复后自动连接服务器并恢复作用。

(4)网络通信访问控制及加解密测试。保证网络上的所有数据都以密文方式传送，且安装客户端软件的计算机终端只能够与部门服务器进行通信，客户端之间不能进行网络通信。

(5)权限实时回收测试。对于期限到期的权限，服务器应能够实时对权限进行回收，保证用户在期限范围外不能再对移动介质上的文件进行越权操作。

此外，测试还包括审计日志有效性测试、防外联测试、移动介质安全性测试等。经过 72 h 的不间断测试，系统各个功能组件工作正常，没有出现死机和蓝屏等现象。

4.2　服务器负载测试

在安全系统中当客户端的数量不断增加时，服务器的负载均衡问题可能会造成瓶颈，通过同时运行多个客户端对服务器负载进行测试。由于实验条件的限制，在部门内部部门服务器对 16 台客户端进行监控，在互联网上防外联服务器对 100 台客户端进行监控。测试结果表明，系统中的部门服务器和防外联服务器均可以正常工作。当客户端增多时，内存消耗趋于平缓，说明系统比较稳定。

5　结束语

系统从多级安全控制和内外兼防的角度出发，保证了存储在移动介质中的数据的安全。经过全面测试表明该系统可靠性强，可用于政府部门和中小型企业。PKI(Public Key Infrastructure)和 PMI(Privilege Management Infrastructure)的理论成果完全可以使用在系统的用户认证和权限管理体系中，PKI 和 PMI 如果能够得以应用到移动介质的安全管理中，则将会把整个安全范围推向互联网络，而不仅仅局限于内部局域网。

参考文献

[1] 洪　帆, 崔国华, 付小青. 信息安全概论[M]. 武汉: 华中科技大学出版社, 2005.

[2] Pieprzyk J, Hardjono T, Seberry J. Fundamentals of Computer Security[M]. 北京: 机械工业出版社, 2002.

[3] 尤晋元, 史美林. Windows 操作系统原理[M]. 北京: 机械工业出版社, 2001.

[4] 武安河. Windows2000/XP WDM 设备驱动程序开发[M]. 2 版. 北京: 电子工业出版社, 2005.

[5] 瞿　进, 李清宝, 白　燕, 等. 文件过滤驱动在网络安全终端中的应用[J]. 计算机应用, 2007, 27(3): 624-626.

编辑　任吉慧