

学校编码: 10384

分类号 _____ 密级 _____

学号: 200340014

UDC _____

厦 门 大 学

硕 士 学 位 论 文

模型驱动体系结构研究

Model Driven Architecture Research

田 英

指导教师姓名: 陈海山 副教授

专 业 名 称: 计算机应用技术

论文提交日期:

论文答辩时间:

学位授予日期:

答辩委员会主席: _____

评 阅 人: _____

2006 年 5 月

厦门大学学位论文原创性声明

兹提交的学位论文，是本人在导师指导下独立完成的研究成果。本人在论文写作中参考的其他个人或集体的研究成果，均在文中以明确方式标明。本人依法享有和承担由此论文产生的权利和责任。

声明人（签名）：

年 月 日

厦门大学博硕士学位论文摘要库

摘 要

模型驱动体系结构是对象管理组织针对软件产业所面临的压力提出的一种新的解决途径。MDA 的关键之处是，模型在软件开发中扮演了非常重要的角色。整个软件开发过程是由对软件系统的建模行为驱动的。MDA 的开发过程就是首先抽象出与实现技术无关、完整描述业务功能的核心模型 PIM，针对不同实现技术制定多个映射规则，然后通过这些映射规则及辅助工具将 PIM 转换成与具体实现技术相关的应用模型 PSM。最后，在一定程度上将 PSM 自动转换成代码。

本文以基于模型驱动的软件开发方法为主要研究内容，对 MDA 的基本原理、基本框架、软件开发生命周期、核心技术、模型变换的实现技术进行了深入的研究和探索。在对模型、平台无关模型、平台相关模型、变换、MDA 的软件开发周期等进行深入分析的基础上，深入研究了 MDA 的核心技术，包括 UML、MOF、XMI、CWM 等。深入研究了模型变换的各项变换特性，给出了 PIM 到 DBMS PSM、EJB PSM、Web PSM 的变换规则的形式化表示。同时，给出了从 DBMS PSM、EJB PSM、Web PSM 到代码模型的变换规则。最后，在一个财务管理系统项目中，应用了 MDA 的软件开发方法。实现了领域模型的建立、领域模型到应用模型的变换。最后，在一定程度上实现了应用模型到代码模型的变换。

关键词：模型驱动体系结构；平台无关；模型变换

厦门大学博硕士学位论文摘要库

Abstract

Model Driven Architecture is proposed by OMG in order to solve the problem which the software industry to face today. The key of MDA is that model is act as a very important role in software development period. The whole development process is driven by creating model for the system. The first period of MDA process is to abstract the core model (PIM) which is independent from the detail implement issue and can descript the business rules completely. Then define the different map rules for different implement platform. Then base on these map rules, transform the PIM into an application model (PSM) which depend on the implement platform. At last, transform the PSM into code in some extent.

This paper regards the MDA software development method as the main topic. Base on the deeply research on the basic principle, basic architecture, software development circle, model independent, and model transform implement of MDA, this paper also research deeply on the core techniques of MDA such as UML, MOF, XMI, CWM, etc. Base on these research, the formal descript of model transform from PIM to DBMS PSM, EJB PSM and Web PSM are given. This paper also gives the transform rules about how to transform these PSMs into a Code Model. At last, we implement MDA in a real project. This paper descripts the whole development process.

Keywords: Model Driven Architecture; Platform Independent; Model Transform

厦门大学博硕士学位论文摘要库

目 录

第 1 章 绪论	1
1.1 课题背景	1
1.1.1 软件产业的发展阶段	1
1.1.2 软件产业所面临的压力	3
1.1.3 模型驱动体系结构	3
1.2 本文的主要内容	4
1.2.1 本文的主要工作	4
1.2.2 本文的篇章结构	5
第 2 章 MDA 的基本原理	6
2.1 MDA 的基本概念	6
2.1.1 模型	6
2.1.2 平台无关模型和平台相关模型	7
2.1.3 模型变换	8
2.2 MDA 的基本框架	8
2.3 MDA 软件的开发生命周期	9
2.4 MDA 软件开发方法的优势	10
第 3 章 MDA 的核心技术	12
3.1 统一建模语言 (UML)	12
3.2 元对象设施 (MOF)	14
3.2.1 MOF 概述.....	14
3.2.2 MOF 元层次.....	15
3.2.3 MOF 自身模型.....	15
3.2.4 MOF 映射.....	16
3.2.5 模型驱动的元数据管理	16
3.3 XML 元数据交换 (XMI)	19
3.4 公共仓库元模型 (CWM)	20
第 4 章 模型变换实现	21
4.1 概述	21
4.2 变换特性	22

4.2.1 可调性	22
4.2.2 可追溯性	23
4.2.3 增量一致性	24
4.2.4 双向性	24
4.3 变换规则.....	25
4.3.1 变换规则需求	25
4.3.2 变换规则的形式化表示	25
4.3.3 变换定义	27
4.4 变换的实现.....	28
4.4.1 概述	28
4.4.2 PIM 到 PSM 变换的实现	29
4.4.3 PSM 到代码变换的实现	39
第 5 章 基于 MDA 的财务管理系统的实现.....	43
5.1 财务管理系统简介.....	43
5.2 总账模块的平台无关模型 (PIM)	44
5.3 PIM 到 PSM 的变换	45
5.3.1 PIM 到 DBMS PSM 的变换	45
5.3.2 PIM 到 EJB PSM 的变换	46
5.3.3 PIM 到 Web PSM 的变换	48
5.4 PSM 到代码模型的变换	49
5.4.1 DBMS PSM 到代码的变换	49
5.4.2 EJB PSM 到代码的变换.....	51
5.4.3 Web PSM 到代码的变换	51
第 6 章 总结与展望	53
参考文献.....	55

Contents

Chapter 1 Introduction.....	1
1.1 Background	1
1.1.1 Software Industry Development Periods.....	1
1.1.2 Software Industry Faces Stress.....	3
1.1.3 Model Driven Architecture.....	3
1.2 Main Topic	4
1.2.1 Main Research Work in this paper.....	4
1.2.2 Paper Structure.....	5
Chapter 2 MDA’s Basic Principle.....	6
2.1 MDA’s Basic Definition	6
2.1.1 Model.....	6
2.1.2 PIM and PSM	7
2.1.3 Model Transformation	8
2.2 MDA’s Basic Framework	8
2.3 MDA’s Software Development Lifecycle.....	9
2.4 Advantages in MDA	10
Chapter 3 MDA’s Core Technology	12
3.1 UML	12
3.2 MOF	14
3.2.1 MOF Introduction.....	14
3.2.2 MOF Meta-Layer.....	15
3.2.3 MOF Model for Itself	15
3.2.4 MOF Mapping	16
3.2.5 Model Driven Meta-data Management.....	16
3.3 XMI	19
3.4 CWM.....	20
Chapter 4 Model Transformation Implement.....	21
4.1 Introduction.....	21
4.2 Transformation Characteristic	22
4.2.1 Tunability.....	22
4.2.2 Traceability	23
4.2.3 Incremental Consistency.....	24
4.2.4 Bidirectionability	24
4.3 Transformation Rules	25
4.3.1 Requirement of Transformation Rules	25

4.3.2 Formal Expression of Transformation Rules.....	25
4.3.3 Transformation Definition	27
4.4 Implement Transformation.....	28
4.4.1 Introduction	28
4.4.2 Transform from PIM to PSM.....	29
4.4.3 Transform from PSM to Code	39
Chapter 5 MDA-based Accounting Management System Implement	43
.....	
5.1 Accounting Management System Introduction.....	43
5.2 PIM of GL Model.....	44
5.3 Transform from PIM to PSM	45
5.3.1 Transform from PIM to DBMS PSM	45
5.3.2 Transform from PIM to EJB PSM.....	46
5.3.3 Transform from PIM to Web PSM	48
5.4 Transform from PSM to Code	49
5.4.1 Transform from DBMS PSM to Code.....	49
5.4.2 Transform from EJB PSM to Code.....	51
5.4.3 Transform from Web PSM to Code.....	51
Chapter 6 Summary	53
Reference.....	55

第1章 绪论

1.1 课题背景

1.1.1 软件产业的发展阶段

1.1.1.1 以机器为中心的阶段

在软件产业的最早期，软件的开发是以机器为中心的。程序员用机器语言编写程序，这是一件十分繁琐的工作，编写程序花费的时间往往是实际运行时间的几十倍或几百倍。而且，编出的程序全是些 0 和 1 的指令代码，直观性差，还容易出错。软件的抽象层次很低。

汇编语言的出现是这个阶段的一个创举。汇编语言允许程序员使用简单的助记符来表示计算机所能理解的机器语言。程序员可以用“MOV AX, DX”来把数据从 D 寄存器移到 A 寄存器，而不必写出执行移数据指令的二进制代码。这样，程序员也变得更不容易犯错误。因此，程序的质量也提高了。汇编语言提高了软件的抽象层次，成功地指明了通向以应用为中心的软件开发阶段的道路。

1.1.1.2 以应用为中心的阶段

第三代编程语言（3GL）的出现使得生产力出现了飞跃。结构化的第三代编程语言（如 C 和 Pascal），使得程序员只需要用短短的几条指令，就可以完成一个计算逻辑。这极大地提高了程序员的生产率，从而降低了生产成本。同时，出错的可能性也降低了，提高了程序的质量。由于编译器可以根据不同的目标处理器来产生不同的机器代码，因此，第三代编程语言具有可移植性。但是，如果编译器不得不为磁盘读取和显示这样的例程产生详细的机器代码，编译器的压力就太大了。如果只是产生调用操作系统的磁盘和显示服务的代码则要轻松很多。因此，第三代编程语言进一步提升了软件的抽象层次，而操作系统则提高了计算平台的抽象层次，减轻了编译器的压力。

面向对象第三代编程语言的出现，使得在不同场合复用程序的一部分变得

更为简单。一些面向对象的语言（如 Smalltalk, Java, C#）引入了一个称为虚拟机的解释器。虚拟机执行语言编译器产生的中间代码。中间代码独立于处理器和操作系统，因此，在不同处理器和操作系统上实现虚拟机，使得以编译形式在不同计算环境间移植应用程序成为可能。

1.1.1.3 以企业为中心的阶段

基于组件的开发^[1]（Component-Based Development, CBD）吸取了工业生产过程的经验，致力于创建一个用可以互换的组件组装应用程序。组件化使得人们不必在应用程序中重复相同的解决方案，提高了软件的复用性。从而提高了生产率，降低了成本。由于组件化把功能分离开来，使得团队调试和升级某个功能时不必顾虑其余部分，因此有助于提高软件质量。

设计模式^[2]的概念也对提高软件开发效率和质量做出了巨大的贡献。程序员可以复用别人已经思考并证实过的通用设计模式。例如：J2EE（Java 2 Platform Enterprise Edition）。

中间件技术的出现，再一次提高了编程的抽象层次。中间件是位于操作系统和应用程序中间的一种基于分布式处理的软件。中间件屏蔽了底层操作系统的复杂性，使开发人员面对一个简单而统一的开发环境，减少程序设计的复杂性，将注意力集中在自己的业务上，不必再为程序在不同系统软件上的移植而重复工作，从而大大减少了技术上的负担。基于中间件技术的软件体系结构为软件的集成、互操作、可移植性提供了极大的便利。中间件提供了一个位于操作系统层之上的计算抽象层^[3]。

多层软件架构是分布式软件系统架构方案之一。这种架构提升了数据和逻辑的复用程度，从而提高了软件开发的效率和质量。同时，也更好的利用了中间件。将中间件应用于各个层次，从而最大程度上减少了重复编写这些层次用到的基本功能的需要。

以企业为中心的阶段还涌现出了很多其他的技术，如说明性规约（Declarative Specification）、契约式设计（Design by Contract）、企业应用集成（EAI）等等。这里不再一一赘述。

1.1.2 软件产业所面临的压力

用户对软件产业的需求不断增加。传统的软件开发方法，在软件的生产成本、质量和寿命等多个方面承受着巨大的压力。

在生产成本方面，针对更复杂的需求进行企业编程所需的密集劳动力给生产成本带来了压力。程序中很小的改动可能会影响到多处修改，这可能需要对整个系统的测试和验证。例如：添加了一个数据属性。可能会涉及到数据库中表格、Server 端、Client 端、以及用户界面的修改。如果这个系统还会被其他的系统调用的话，情况可能更糟。开发组织不得不竭尽全力来控制这一复杂性。对于程序员来说，手工管理并保持同步的确不是一件简单的事情。

在质量方面，增加复杂性本来就对质量造成了压力。而业界当前的趋势就是：代码就是一切，体系结构和设计并不是那么重要。文档只是在有时间的时候才去写。并且文档和具体的代码之间存在着不一致性。同时，测试也是在有时间的时候才去做。并且，很多情况下，测试工作也只是最终用户测试。在跟是否可以与其他的 product 实现互操作相比，产品中的 Bug 往往只是冰山的一角。因此，从某种程度上说，是程序员承担了对新的复杂性的压力。

程序设计界有些人努力试图找出管理压力的方法。极限编程（Extreme Programming, XP）的趋势就体现程序员想要控制这一情形的需求^[4]。而更为全面的方法学，比如统一过程（Unified Process）。但是，这些并没有从根本上解决软件复杂性对开发者带来的压力。

在寿命方面，平均每五年就会出现新的平台、新的技术。在五年前 J2EE、.NET、Web Services 都还是不存在的。在旧的平台开发的系统，是否可以在新的平台上运行，而且可以没有任何的损失呢？大多数情况下，唯一的解决方法就是进行代价高昂的手工编码升级。

1.1.3 模型驱动体系结构

模型驱动体系结构（Model Driven Architecture, MDA）是对象管理组织（OMG）针对软件产业所面临的压力提出来的一种新的解决途径。

MDA 的关键之处是，模型在软件开发中扮演了非常重要的角色，整个软件开发过程是由对软件系统的建模行为驱动的。MDA 的开发过程就是首先抽象出

与实现技术无关、完整描述业务功能的核心模型 PIM (Platform-Independent Model), 针对不同实现技术制定多个变换规则, 然后通过这些映射规则及辅助工具将 PIM 转换成与具体实现技术相关的应用模型 PSM (Platform-Specific Model)。最后, 在一定程度上将 PSM 自动转换成代码^[5]。

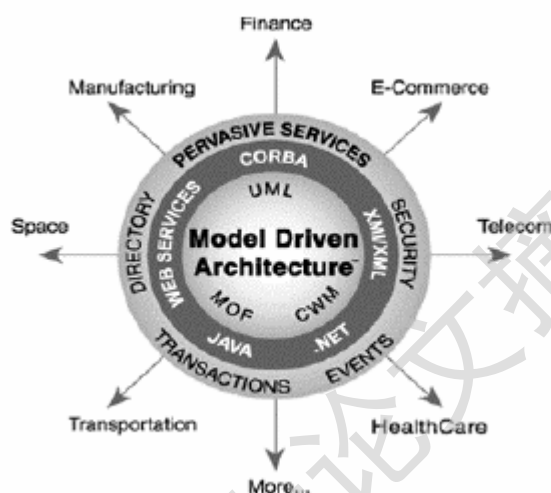


图 1.1 MDA 整体框架

图 1.1 给出了 MDA 的整体框架。MDA 不仅包含 OMG 已经建立的一系列重要的集成标准, 如 UML、CORBA 等, 而且还涵盖了 OMG 所定义的普遍服务, 包括目录服务、事件处理、持久性、事务和安全性。另外, MDA 还支持特定垂直行业, 如电信、运输行业的标准化领域模型的建立⁶。

MDA 以计算机可以理解的模型为中心, 把软件开发提升到更高一级的抽象层次——模型层次。

1.2 本文的主要内容

1.2.1 本文的主要工作

本文以基于模型驱动的软件开发方法为主要研究内容, 对 MDA 的基本原理、基本框架、软件开发生命周期、核心技术、模型变换的实现技术进行了深入的研究和探索。在对模型、平台无关模型、平台相关模型、变换、MDA 的软

Degree papers are in the "[Xiamen University Electronic Theses and Dissertations Database](#)". Full texts are available in the following ways:

1. If your library is a CALIS member libraries, please log on <http://etd.calis.edu.cn/> and submit requests online, or consult the interlibrary loan department in your library.
2. For users of non-CALIS member libraries, please mail to etd@xmu.edu.cn for delivery details.

厦门大学博硕士论文摘要库