

# 基于 MDA 的企业数字化管理平台

冯娟, 史浩山, 李实

(西北工业大学电子信息学院, 西安 710072)

**摘要:** 针对传统面向对象技术中系统自动化程度较低、难以适应需求和技术变化的问题, 提出基于模型驱动的企业数字化管理平台, 基于模型驱动(MDA)设计系统总体框架, 利用本体库和规则库的推理机制实现模型间的相互映射。该平台能实现企业对产品开发和管理的智能化和数字化, 保障系统与企业的同步发展。

**关键词:** 模型驱动架构; 本体库; 规则库; 模型映射

## Digital Management Platform in Enterprises Based on MDA

FENG Juan, SHI Hao-shan, LI Shi

(School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072)

**【Abstract】** In order to improve that automation of systems is low level in traditional object-oriented technology which is difficult to adapt to changes in the two aspects of demands and technology, this paper facilitates the development of products and the management of military enterprises, designs the digital management platform of military enterprises based on the model-driven. It uses the idea of model-driven design the overall framework of system and has the inference mechanism of ontology library and rules to implement the mutual transformation of models. The development and management of enterprise product are implemented, intelligently and digitally. The whole system and enterprise development's synchronization is ensured.

**【Key words】** Model Driven Architecture(MDA); ontology library; rule library; model mapping

传统的面向对象技术的缺点是未区分应用领域建模技术和软件技术, 使系统在构造过程中, 应用领域专家和软件专家不能严格区分。本文提出企业数字化管理平台的设计思路和实现方法, 用来开发适合该领域的建模方法和工具, 以支持对应用软件等系统的成功实施。在应用软件运行阶段, 当企业的组织、信息和流程发生变化时, 通过对企业模型的更改, 实现对应用软件系统的更改, 即可快速实现系统与企业发展的同步, 保障系统的成功应用。

### 1 MDA 技术

模型驱动架构(Model Driven Architecture, MDA)是对象管理组织(OMG)为大型系统应用架构的设计与开发提出的一种方法与思路, 其核心是建模和模型映射技术, 它将模型划分为: (1)计算无关模型(Computation Independent Model, CIM), 在系统需求分析阶段从纯业务角度描述系统应完成的工作。(2)平台无关模型(Platform Independent Model, PIM), 是对系统功能和结构的形式化描述, 是抽象的、与技术细节无关的。(3)平台特定模型(Platform Specific Model, PSM), 是在特定目标平台上的系统功能的描述。采用 MDA 方法可为不同人员提供不同的模型工具, 从而区分应用领域人员和计算机专家。向应用领域人员提供一种容易掌握的工具, 来建立与技术平台无关的系统模型 PIM, 它是最原始的模型, 此后, 可对该模型进行自动优化和变换, 直到产生面向平台的模型 PSM, 最后再由 PSM 自动产生可执行程序, 从而实现软件开发自动化。不同应用领域对建模工具的要求是不同的。在本系统中, 建模工具应能支持企业制造过程建模。

### 2 主要建模工具

目前针对企业数字化管理平台的建模工具主要有 2 类:

(1)以统一建模语言 UML 为基础的工具, UML 是一种通用的可视化建模语言, 用于对软件进行描述、可视化处理, 它包括概念的语义, 表示法和说明, 是为支持大部分现存的面向对象开发过程而设计的。

UML 建模的优点是 UML 语言比较普及, 且使用的约束较少, 灵活, 便于一般非计算机专业人员使用。但也有缺少严格的语义, 使用方法不规范, 人机难以理解等不足。

(2)企业过程建模系统(Enterprise Process Modeling System, EPMS)工具。主要针对企业业务流程的过程建模, 所谓企业业务流程就是企业在实施一个项目过程中, 各参与要素(部门、岗位、角色等)按一定的参与顺序对项目进行建设和作用的过程。它建模的主要对象就是描述各参与要素的作用和路线。

EPMS 建模的优点是语义严密, 容易支持对模型的优化, 但普及性不好, 应用领域工程师掌握较难。

目前 MDA 的发展方向是设计一种可执行的 UML 语言, 即 xUML, 但 xUML 还未能普及和实用。因此, 领域人员使用普通 UML 工具为企业信息系统和过程建模仍是目前最可行和实用的方法。但由于用 UML 所建的模型不能进行模拟执行和优化, 因此, 须将此模型映射为 EPMS 模型, 从而进行模拟与优化, 再将优化后的 EPMS 模型映射为 UML 模型, 用于支持对信息系统软件实体的开发或自动构建, 其转换思路如图 1 所示。本系统实现了 UML 和 EPMS 模型之间的自动映射, 为企业完成自动构建奠定基础。

**作者简介:** 冯娟(1983 -), 女, 硕士研究生, 主研方向: 计算机网络与通信; 史浩山, 教授、博士生导师; 李实, 硕士研究生

**收稿日期:** 2008-09-19 **E-mail:** fengjuankh@hotmail.com

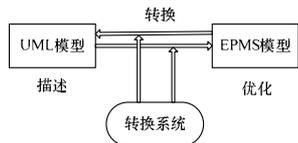


图1 模型转换思路

### 3 系统总体设计和实现

在系统中针对企业的 UML 和 EPMS 参考模型建立模型库，并开发本体规则库，建立 UML 与 EPMS 模型的映射关系。现行企业的 UML 模型作为源模型进入模型映射系统，系统根据本体规则库中所规定的对源模型中各个模型元素的变换规则，向目标模型的元素进行变换，输出 EPMS 模型。然后将 EPMS 模型与企业实施应用软件的需求模型相结合，并仿真、分析、优化和重组，形成符合实施应用软件的 EPMS 模型。再将优化后的 EPMS 模型经过本体规则库的映射，形成实施应用软件所需要的 UML 模型。系统中模型仓库存储和管理涉及到的各种模型的元模型。本系统遵循 OMG 的 CWM(Common Warehouse Meta)模型标准建立并管理模型库。在转换过程中，对输入的源模型，须结合模型仓库中该模型的元模型进行语法层面的分析，隔离出各个模型元素或子模型，再遵循本体规则库中的相应转换规则进行转换。因此，模型仓库用于系统在语法层面对输入模型的理解，而本体规则库则在语义层面上支持对源模型的转换。

在本系统中，模型映射是个核心问题，主要包括：UML 和 EPMS 模型的 CIM 层相互映射，UML 模型的 CIM 层到 PIM 层模型的映射，UML 模型的 PIM 层到 PSM 层模型的映射。

本系统实现了 UML 与 EPMS 模型之间的相互转换，并设计一个相对通用的模型映射工具以支持多种模型之间的映射。模型映射部分的结构如图 2 所示。

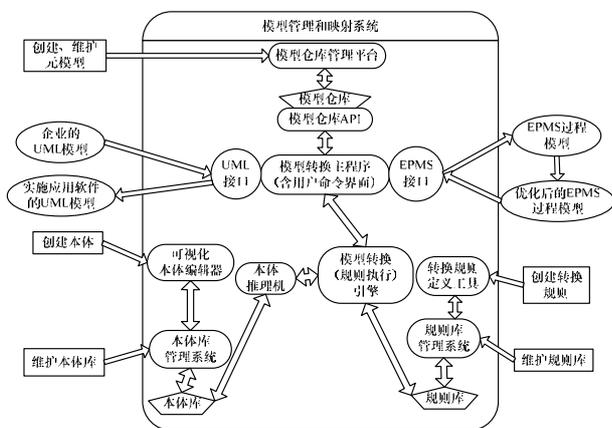


图2 模型映射的总体结构

#### 3.1 本体规则库

本体规则库由本体库和规则库组成，管理企业本体和模型转换规则。由于企业数字化涉及多个不同系统，每个系统的模型又分布在不同的 UML 视图中，而不同的视图可能会使用不同的模型定义，如组织机构视图、工具视图、过程视图等，可能会有多人、多单位参与系统建模。这些复杂的因素会导致模型中概念或术语的不一致，因此，采用本体<sup>[1]</sup>来解决这一问题，以增强系统对模型的理解能力。本体是对概念的明确描述和说明<sup>[2]</sup>，其目的是达成领域知识的共享及计算机的理解和可操作。本系统采用 protégé 来建立本体库<sup>[3]</sup>。本体的描述和推理基于描述逻辑(DL)，其表示和交换可借助

于 OWL(Ontology Web Language)。OWL 是一种本体描述语言，由类、数据类型、对象属性、数据类型属性、个体和数据值组成，它们之间有不同的语义关系，最常见的关系在 OWL 中的语义标签如表 1 所示。

表 1 3种语义关系在 OWL 中对应的语义标签

	父/子关系	等价关系
类	rdfs:subClassOf	EquivalentClass
对象属性	rdfs:subPropertyOf	EquivalentProperty
数据类型属性	rdfs:subPropertyOf	EquivalentProperty
个体	-	sameAs

在基于描述逻辑表示的本体中，概念是一个三元组  $C=(A, B, R)$ 。其中， $A$  是对象的集合， $B$  是属性的集合， $R$  是  $A$  和  $B$  之间的一个二元关系，即  $R \subseteq A \times B$ 。 $aRb$  表示  $a \in A$  与  $b \in B$  之间存在关系  $R$ ，读作对象  $a$  具有属性  $b$ 。这是一种清晰的概念层次结构，本质上体现了实体(对象、文件、记录)和属性(特征、词汇)之间的联系。

为支持多种不同模型间的映射，统一管理不同模型间的映射规则，在系统中建立了模型转换规则库，用于定义、管理模型转换规则，并通过统一的转换引擎实现模型的变换。转换规则和具体的模型元素映射一一对应，它是一种形式化的规则，具体的描述一个源模型元素是如何映射为目标模型元素的。模型转换引擎负责利用规则库对输入的源模型进行转换，产生目标模型。在此转换过程中会调用本体推理机，判断源模型中的某 2 个概念是否等价。规则库可以通过管理、维护人员利用可视化的模型转换规则定义工具、创建模型之间的转换规则，由规则管理模块加入到规则库中。规则库和本体库也可以由管理人员通过它的管理模块对其进行增加、删除、修改和查询。

#### 3.2 模型映射的层次和思路

模型映射问题涉及语法和语义 2 个方面<sup>[4]</sup>。在语法层面上，不同的模型均遵循统一的模型体系结构，模型体系共分 4 层：M3(最高层)是 MOF(Meta Object Facility)，它用来定义所有模型的元模型，又称为元元模型层。M2 层是元模型层，它定义了系统中所有模型的语法。M1 层是具体的模型，它符合 M2 层的语法规则。M0 层是各个模型的最终应用。模型转换规则在 M2 层定义，而实际转换发生在 M1 层。

首先在 M2 层定义系统所涉及的模型，并将这些模型纳入到模型仓库中进行统一的模型管理，此后再定义模型映射规则，存储在规则库中，用于支持实际的模型转换。本系统使用 MOF 原语定义 EPMS 的元模型，通过 UML Profile 扩展 UML，以定义企业的 CIM 模型。UML Profile 将对军工企业的组织机构、产品型号、制造工具和场所等模型进行定义，还要对活动图、泳道图、顺序图等进行规范，从而为企业过程的建模提供元模型。得到元模型后，采用基于规则和描述语言相结合的方法定义模型映射规则。规则中会涉及被变换的元素、变换的条件、变换后的结果、变换后应满足的条件、在变换过程中须采取的其他动作等。例如：若过程模型  $P$  中的活动用  $x$  表示， $x$  的属性用  $Attribute(x)$  表示，UML 模型中的功能节点用  $n$  表示，节点的属性用  $Attribute(n)$  表示。对于  $\forall x, x \in P$ ，至少存在一个  $n \in N$ ，使  $Attribute(x) = Attribute(n)$ ，则称活动  $x$  与功能节点之间存在映射关系，记作  $x \rightarrow n$ 。

模型元素的语义映射是模型映射的第 2 阶段，在此阶段通过使用基于 EPMS 和 UML 建模语言的形象化描述，给出模型元素和语义映射描述的例子：模型等价转换以

“Transformation”作为开始，其后是要进行等价转换的源模型和目标模型，具体的内容放在花括号中，源模型和目标模型用←和→连接，UML模型后的括号中的数字表示等价转换所对应的模型的个数，具体的元素映射部分以“Mapping”来开头，用→来连接2边的操作数，表示将左边的模型映射为右边的模型。

Transformation : {ClassDiagram(1+x), UseCaseDiagram(1+x), ActivityDiagram(1+x)} ↔ EPM

{ Mapping

ClassDiagram(1)Class.N → EO.N

ClassDiagram(1)Class.ATT → EProcess.G

Mapping

ActivityDiagram(1)ACT.N → EProcess.N

ActivityDiagram(1)ACT → EA

ActivityDiagram(1).trans(act1,act2) → EAA

(ea1,ea2)

ActivityDiagram(1)inout(infro,act2) → EOA

(eo1, ea1)

Mapping

UseCaseDiagram(1)N → PEA.N

UseCaseDiagram(1)Role.N → POSI.N

UseCaseDiagram(1)contion(usecase1, usecase2) → EIAA(ea1, ea2)

UseCaseDiagram(1)use(actor,usecase) → PEA(posi,ea) }

其中，EPM为EPMS企业过程模型；Mapping为映射域；1+x为表示须映射的模型数量；ATT为UML类图属性；N为名称；Contion表示包含关联描述；Use为用例关系描述；Trans为活动转换关系描述；Inout为信息对象输入输出关联描述。

某企业成件到货登记的EPMS过程模型如图3所示。通过模型转换系统转换后生成的UML模型如图4所示。

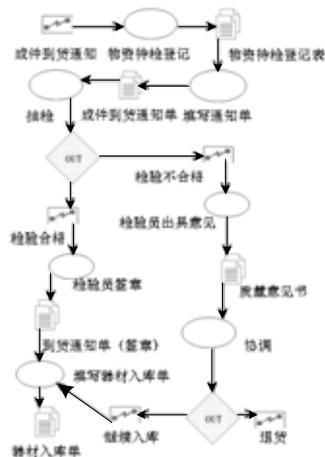


图3 EPMS过程模型

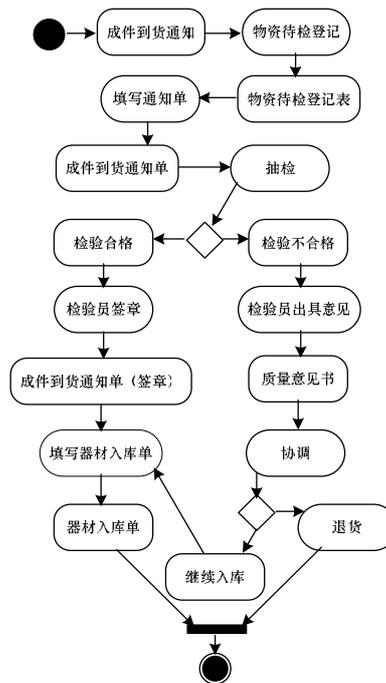


图4 EPMS转换后的UML模型

#### 4 结束语

本文提出基于模型驱动架构的企业数字化管理，实现了从CIM向PIM层企业模型转换的目标，提高了应用软件实施的成功率，利用模型的快速转换机制，在运行期可以及时对应用软件系统进行适应性升级，方便了企业的统一管理和产品的设计开发。采用基于本体规则库的统一模型映射技术，可避免整个项目在模型映射方面由不同的机构重复类似的研究工作，从而加快项目进度、节省工作量。

#### 参考文献

- [1] Uschold M, Gruninger M. Ontologies: Principles, Methods and Applications[D]. Edinburgh, UK: The University of Edinburgh, 1996.
- [2] Gruber T R. A Translation Approach to Portable Ontology Specifications[J]. Knowledge Acquisition, 1993, 5(2): 199-220.
- [3] 陈壮生, 瞿裕忠. 基于本体的信息处理系统的设计与实现[J]. 计算机工程, 2005, 31(11): 165-167.
- [4] Rodriguez M A, Egenhofer M J. Determining Semantic Similarity Among Entity Classes From Different Ontologies[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(2): 442-456.

编辑 金胡考

(上接第239页)

#### 参考文献

- [1] 王荣山. 大气数据计算机原理[M]. 西安: 空军工程学院出版社, 1995.
- [2] 黄胜, 张吉广, 张宗麟, 等. 一种惯性导航仿真系统的设计研究[J]. 系统工程与电子技术, 2004, 26(9): 1276-1278.

- [3] 武安河, 郜铭, 于洪涛. Windows 2000/XP WDM 设备驱动程序开发[M]. 北京: 电子工业出版社, 2003.
- [4] 邓建中, 葛仁杰, 程正兴. 计算方法[M]. 西安: 西安交通大学出版社, 1985.

编辑 张帆