

ICEMDA : 一种可互操作可配置可执行的模型驱动体系结构

战德臣,冯锦丹,聂兰顺,徐晓飞

(哈尔滨工业大学计算机科学与技术学院,黑龙江哈尔滨 150001)

摘要: 国际对象管理组织提出模型驱动软件体系结构(MDA),以模型建立与模型转换为中心的思想为软件开发带来了新变革,但 OMG 并未给出各层次具体模型与建模理论.为了对按需应变的企业管理软件开发提供一组理论与方法支持,本文提出一种可互操作、可配置与可执行的企业模型驱动体系结构 ICEMDA:面向过程的 ICE-CIM、面向对象的 ICE-PIM、面向模式的 ICE-PSM 和面向构件与配置的 ICE-CODE. 本文阐述了 ICEMDA 各层次模型的建模思想、核心概念及以核心概念为主的模型之间的转换. ICEMDA 可有效指导用户快速开发与实施复杂管理软件.

关键词: MDA; 企业模型; ESA; 可互操作性; 动态配置; 可执行性

中图分类号: TP3 **文献标识码:** A **文章编号:** 0372-2112 (2008) 12A-120-08

ICEMDA : An Interoperable Configurable Executable Model Driven Architecture

ZHAN De-chen, FENG Jin-dan, NIE Lan-shun, XU Xiao-fei

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China)

Abstract: MDA based on modeling and transformation brings novel thought to improve software development, but OMG doesn't give the concrete modeling methodology and reference model. This paper proposes an ICEMDA framework of software development for enterprise transforming on demand, including the process-oriented ICE-CIM, object-oriented ICE-PIM, pattern-oriented ICE-PSM and business component based ICE-CODE. The paper presents modeling thinking on every layer of ICEMDA, core concepts and model transformation based on them. The contribution of ICEMDA is helpful to develop and implement the large scale complicated software for enterprises.

Key words: model driven architecture (MDA); enterprise model; enterprise software and application (ESA); interoperable; configurable; executable

1 引言

面对全球范围内激烈的市场竞争,企业总是通过改进管理模式、业务重组、资源优化等手段,以快速响应不断变化的市场需求,提高其自身的竞争力.在企业业务持续完善的过程中,软件需求也总是不断变化的,因此,企业管理软件必须具备快速有效地适应不断变化的软件需求的能力,这就需要软件从体系结构、实现机制乃至开发方式都能支持和实现这种按需应变的能力^[1,2].

模型驱动的软件体系结构与软件开发(Model Driven Development, MDD)被认为是实现此类软件的一种有效途径.国际对象管理组织(Object Management Government, OMG)提出 MDA 的主要目标是通过关注点的结构性分离来实现复杂抽象系统不同层面的可实施性、互操作性和可重用性^[3],将大型软件系统的开发分解为不同层面、分而治之,降低了系统整体设计和开发的难度,有效

提高了复杂系统开发的成功性与可信性. MDA 的基本思想是将系统的需求、功能规约与其在特定技术平台上的实现规约相分离,分别建立描述业务需求的计算无关模型(Computation Independent Model, CIM),描述软件系统设计方案的平台无关模型(Platform Independent Model, PIM),描述具体实现机制的平台相关模型(Platform Specific Model, PSM),通过各层次模型要素的映射和转换实现逐层精化以及软件系统自动生成,其中建模与模型转换是 MDA 研究的两个核心问题^[4]. MDA 思想改变了传统的软件开发进程,将软件开发者从繁杂的编程中解放出来,提升为以模型为中心,进行模型构造、转换和精化^[5,6],并通过代码自动生成来开发系统,其相关技术正在成为国内外研究的热点.

OMG MDA 给出一系列模型驱动的相关概念及概念之间的关系和应用原则等,推荐软件开发者采用 UML、MOF 和 CWM 等技术实现建模,基于元模型、模

收稿日期:2008-08-25;修回日期:2008-12-20

基金项目:国家自然科学基金(No. 60773064);国家 863 高技术研究发展计划(No. 2006AA01Z167, No. 2006AA04Z165, No. 2006AA04Z150, No. 2007AA01Z128, No. 2008AA04Z101, No. 2008GGI000401028)

式、模型融合等技术实现模型转换^[7,8]。然而 OMG/MDA 只是从抽象层面对软件设计与研发过程予以指导,并未给出各层次模型的具体建模要素、建模方法及相关的模型转换方法等^[9]。软件开发者仍需依据其研究领域的特征和软件系统的需求等,选择或提炼具体的建模与模型转换方法,以指导其模型驱动的软件开发过程。

当前已出现许多企业建模方法和语言,如 CIMOSA 作为一种开放系统参考框架为制造企业的系统集成和企业建模提供了方法论^[10];GRAI/LAPs 提出的 GIM/ GRAI 方法将建模过程划分为“As Is”和“To Be”两个阶段,以确保面向需求的描述系统信息^[11];及面向结构分析的 IDEF,面向过程/控制的 ARIS 和面向对象的 UML 等方法^[12]。然而 CIMOSA, GRAI, IDEF 和 ARIS 是针对业务分析层次建模,并未提供如何转向 ESA 系统设计的方法。虽然 UML 建模语言可贯穿于软件开发的各个阶段,但其严格的规范更适合于 PIM 以下层次。UML 提供了不同的图来描述一个系统的不同方面,但仍缺少语义要素将各图围绕一个线索关联起来。上述建模方法都没能形成一套贯穿 MDA

的建模思路,即使综合应用各方法建立相应层次的模型,也很难实现相互集成与自顶向下的平滑转换。

因此本文在 MDA 思想指导下,借鉴与整合上述建模方法的一些思想,提出了一套可互操作,可配置,可执行的模型驱动体系结构(Interoperable Configurable Executable Model Driven Architecture, ICEMDA)。在 ICEMDA 各层次分别提出了一系列建模要素(即核心概念)及其关系,依据相应的建模规范使用这些要素进行建模,既能够准确的描述模型与系统,又可以实现 CIM PIM PSM CODE 之间的模型映射与转换,最终实现自动生成目标代码和系统。

本文首先简要介绍 ICEMDA 的整体框架;其次分别阐述各层次模型所要解决的核心问题、建模目标、核心内容及其关联关系,以及这些模型的建模过程等;最后给出模型转换的实现思想和支持工具等。本文试图为 ESA 软件研究人员提供一套基于模型驱动的指导思想和以核心概念为中心的建模与转换参考框架,限于篇幅各层模型的建模细节未在文中详细讨论。

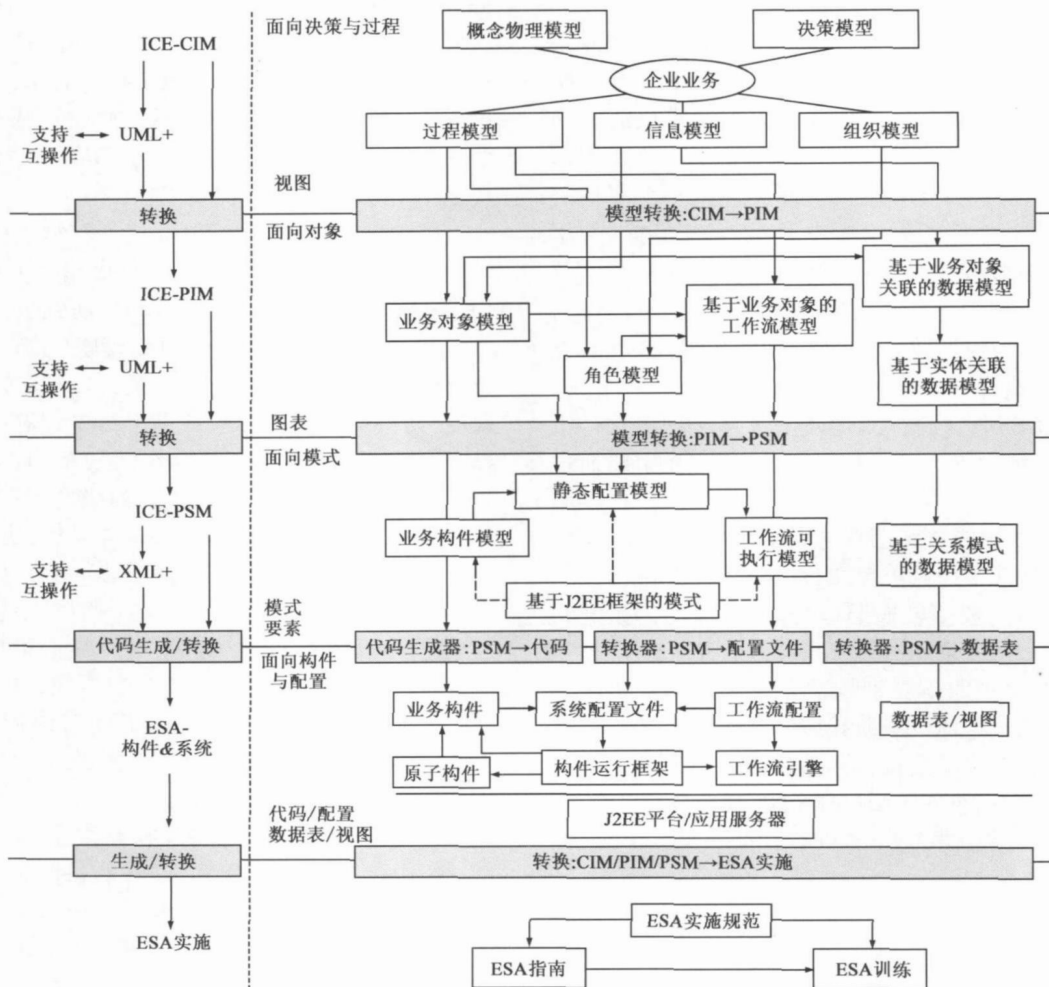


图1 ICEMDA框架

2 ICEMDA 概述

ICEMDA 遵循 MDA 的一般结构形式,包括三个模型层、一个系统层和各层次间的转换. ICEMDA 总体框架如图 1 所示.

ICE-CIM 是面向决策以业务过程优化为中心的集成模型体系,其中每个模型都描述了业务抽象过程中的阶段性或目标性成果.建模者通过建立概念物理模型分析实物流和发现过程优化的关键点,由业务过程模型的输入/出执行关键点控制,将企业组织优化模型和信息模型关联起来,从而为 ICE-PIM 建模提供业务需求.

ICE-PIM 是面向对象的软件设计描述模型.将 ICE-CIM 的信息实体映射为面向业务语义的业务对象,实现了从“面向过程的 CIM”到“面向对象的 PIM”的转换.以业务对象及其特性为基础进一步将过程模型转换为显式表达系统动态特性的 workflow 模型.基于面向对象的模型组装思想,workflow 模型将若干个描述不同视角的子模型关联成一个集成模型.ICE-CIM/PIM 都可以通过扩展 UML 与其他类型的企业模型进行互操作.

从软件实现角度描述系统的 ICE-PSM 具有面向软件模式的特征.ICE-PIM 业务对象模型被转换为基于模式的业务构件 PSM 模型,并建立基于控制模式的全局静态配置模型和基于调度模式的可执行 workflow 模型以及关系数据模型.ICE-PSM 通过对系统内的可变属性的配置控制可从模型层面响应变化的业务需求.采用 XML Schema 描述模式特征和软件结构,支持与其他类型构件实现模型的互操作.同时本层的模型也可作为代码生成工具与模型转换器的输入.因此 ICE-PSM 具有可互操作、可执行、可配置的显著特点.

ICE-CODE 层上的软件系统是由一系列业务构件组装而成,在系统配置文件的控制下,由 workflow 引擎通过解析 workflow 配置文件动态调度业务构件为用户提供服务.业务构件是 ICE-PSM 层业务构件模型的程序实现,系统配置文件和 workflow 配置文件分别由静态配置模型和可执行 workflow 模型转换形成.这种柔性的体系结构使软件在运行态配置相关模型来进行业务转型.

ICEMDA 自顶向下分别构造业务语义与可实现性都逐渐增强的构造体(模型,代码和系统),且能够明确界定上层映射源和下层目标构造体以及核心概念之间的映射关系.为从业务模型(ICE-CIM)到软件模型(ICE-PIM/ICE-PSM)再到系统(ICE-CODE)的转换提供了理论与技术支持.

3 ICE-CIM

ESA 主要用于辅助企业的经营、生产与管理,强调信息管理与服务过程,帮助企业解决如何提高效益与

效率的问题.效益优化集中体现为一系列过程的整体优化,既包括每个环节的优化又强调这些过程之间的协调和集成.ICE-CIM 需要帮助企业与 ESA 开发者解决的主要问题是:(1)如何提供面向系统效益与效率的分析手段,并为企业决策者与需求分析师提供交流的手段;(2)如何描述能被转换到 ICE-PIM 的系统需求.可见 ICE-CIM 应以企业效益优化为最终目标来描述对 ESA 的系统需求.借鉴业务过程重组的建模思想,以业务过程为中心,从实物流、信息流、组织结构和全局决策分别刻画软件的业务需求.

概念物理模型描述了生产过程中的实物转换和实物状态的变迁过程,也描述了业务处理流程和其中的资源使用情况.概念物理模型(Conceptual Physical Model, CPM)可被定义为 $CPM = (PP, ORG, R, ORDER, ACTION)$,其中 PP 是实物 p 的集合,可区分不同实物或同一实物的不同状态; ORG 是企业组织集合; R 是资源集; $ORDER$ 是指令集; $ACTION = (io, control, support, action)$ 是动作集,其中 $io = \{(p_i, p_o) | p_i, p_o \in PP\}$ 为动作 $action$ 的输入和输出实物集, $control = \{order | order \in ORDER\}$ 是控制 $action$ 的指令集, $support = \{s | s = r \wedge r \in R, org \in ORG\}$ 表示执行者 org 利用资源 r 执行 $action$.概念物理模型是从操作层面描述实物在指令支配下的流转过程,更关注于 $ORDER$ 的控制作用,且优化 $ORDER$ 有助于优化过程.

决策模型是一个由 GRAI 模型扩展而形成的二维模型,进一步描述了 $ORDER$ 的层次控制关系与纽带作用. $Decision Model = (H/P, FUNC, ORDER, DecisionCenter)$, H/P 代表行维度,表示时间跨度(如年、月)或决策层面(如战略层、战术层、操作层),体现了不同的决策粒度.而一个企业目标通常是由跨专业跨组织的具有衔接关系的若干项业务协作实现的.因此 $FUNC = \{func | func \text{ 是一个功能域}\}$ 代表列维度,是指为实现某些企业目标而衔接在一起的业务关系集.决策中心 $DecisionCenter = \{order | order = f(h/p, func), h/p \in H/P, func \in FUNC\}$,它是由行和列交叉所确定的单元格,核心项是 $ORDER$ 集. $ORDER$ 沿 H/P 纵向将形成贯穿企业决策、计划、执行等层次的控制主线,沿 $FUNC$ 横向将形成各功能域之间决策的集成与协同.决策者对 $ORDER$ 的深入分析将有助于挖掘 ESA 所能带来的效益.

过程模型位于概念物理模型之上以信息流为主线,描述如计划单、指令单、数据单和报表等信息实体的转换及状态变迁过程.既可描述由指令构成的 $ORDER$ 流,也可定义由非指令文档 $DOCUMENT$ 构成的 DOC 流.过程模型可以定义为 $Process Model = (IP, ORG, DOC, ORDER, ACTION)$,其中 $IP = \{ip | ip = order \text{ 或 } ip = document\}$

$= doc$ 且 $order \in ORDER, doc \in DOC$ 是信息实体集, 用于区分不同信息实体与其状态; 信息实体 ip 是指用户所采集、传递、处理和使用的单据、报表等, 在模型中体现为指令或文档; ORG 是企业组织集合; DOC 是文档集; $ORDER$ 是指令集; $ACTION = (io, control, support, action)$, $io = \{(ip_i, ip_o) \mid \text{当 } ip_i \in ORDER \text{ 则 } ip_o \in ORDER, \text{ 或者当 } ip_i \in DOC \text{ 则 } ip_o \in DOC\}$, 表示动作 $action$ 的输入和输出信息实体集, $control = \{order \mid order \in ORDER\}$ 表示控制 $action$ 的指令集, $support = \{s \mid s = doc \wedge s = org \wedge doc \in DOC, org \in ORG\}$ 表示执行者 org 利用文档资源 doc 执行 $action$ 。

通过分析企业信息流, 建模者可快速识别信息实体集和其间的关系集, 以形成信息模型。信息模型为概念物理模型的动作提供了控制条件, 为决策模型提供了决策对象。进一步分析过程模型可抽象出组织模型, 描述企业中各职能部门及其结构关系, 它以施动者或资源提供者的形式为其他模型提供支持, 反映了业务优化基础上组织优化的成果。

4 ICE-PIM

ICE-PIM 是连接业务模型和软件实现模型的桥梁。将过程模型中信息实体抽象和规范化为一个个业务对象, 采用面向对象技术对每个实体的数据结构、数据依赖与操作和其间的衔接关系进行设计, 为后续基于构件的软件开发奠定了基础。然而面向过程的 ICE-CIM 与面向对象的 ICE-PIM 具有不同的侧重点, 为了实现这两者逼近等价的模型转换, 可将过程模型向下映射为业务对象模型和工作流模型两个部分。

业务对象模型涵盖了业务对象 (Business Object, BO) 以及业务对象的集成模型 (BO-Relationship, BO-R)。从用户角度而言, 业务对象是指信息文档以及在其上的状态处理和操作处理的集成体。从系统角度而言, 业务对象是指具有独立标识的、具有一定生命周期的、由若干数据集和操作集构成的集成对象。依据信息实体的功能可将业务对象抽象为不同的类, 如订单类、报表类、计划类等。每类业务对象依据其具体值内容的不同被进一步分类为业务对象值类型, 如生产性物料采购计划、设备备品采购计划等。业务对象可表示为 $BO = (ID, DS, AT, OP, DPR, SS, ST)$, 其中 ID 是业务对象的唯一标识; $DS = \{ds \mid ds = (name, \{d\})\}$, ds 是依据数据项 d 之间的语义关系组成的数据集 $name$; AT 是 BO 的活动集, 每个活动 at 都是 BO 中若干原子操作 op 的封装结果; $OP = \{op \mid op \text{ 是业务对象的操作}\}$; $DPR = \{(op, ds)\}$, 表示操作对数据集的控制关系集; $SS = \{s \mid s \text{ 是 } BO \text{ 的一种状态且 } (\exists s) (s \in op_1 \dots op_n)\}$; $ST = \{st \mid s_o = st(s_i, at), at \in AT, s_i, s_o \in SS\}$ 表示 BO 的状态

变迁关系集, at 可触发 BO 的状态从 s_i 转换到 s_o 。一个业务对象内部各要素之间存在相互关联与影响的动态或静态关系。业务对象集成图 (BO Integrated Diagram, BOI) 可从四个视角分别建立子模型对其进行描述: (1) 业务对象数据图采用 IDEF1x 描述 BO 内部数据集之间的联系; (2) 业务对象类图体现 DPR , 采用 UML 类图描述 BO 中 OP 及活动 AT 之间的封装关系; (3) 业务对象状态图刻画 BO 的状态 SS 和变迁 ST 等与生命周期相关的特性; (4) 业务对象用例图刻画 BO 的角色相关特性, 如权限约束关系。

若干个业务对象之间的关系可由 $BO-R$ 模型从两个抽象层次予以描述。业务对象层次描述两两 BO 之间的数据依赖联系, 如关联关系、集成关系等; 实体联系层次则以 BO 内部的数据集为实体, 通过两个 BO 内部实体之间的联系实现上层抽象的宏观联系 $BO-R = \{f_R (BO_i, BO_j) \mid f_R = (r_1 \dots r_k), r_1, \dots, r_k \in BE-R\}$, $BE-R = \{r \mid r = f(ds_m, ds_n), ds_m \in BO_i.DS, ds_n \in BO_j.DS, 1 \leq m \leq |BO_i.DS|, 1 \leq n \leq |BO_j.DS|\}$, 其中 i, j 是自然数且 $i \neq j$ 。业务对象 BO_i 和 BO_j 之间的连接规则 f_R 是通过一组实体联系 $r_1 \dots r_k$ 来描述和具体实现的, r 表示若干 BO 之间的实体联系。

数据模型是全体业务对象的逻辑结构和特征的描述, 是基于业务对象内部实体及其之间联系的全局逻辑模式。 $Data Model = (E, ER)$, $E = \{e \mid e = ds_m, ds_m \in BO_i.DS, 1 \leq m \leq |BO_i.DS|, i = 1, \dots, l\}$ 表示该层所有数据集实体的集合; $ER = \{(e_i, e_j) \mid e_i, e_j \in E\}$ 表示所有实体之间的关系集, 其包含一个业务对象内的实体联系与两个业务对象之间的实体联系。 BO 内部的 DS 、 $BE-R$ 等是 $Data Model$ 中语义关联性较强的那部分数据的描述, 缺少任何一个数据都不能构成具有完整语义的 BO 。

不同企业和环境中的业务流程是不同和多变的, 这种差异可区分为业务对象和其间连接关系。前者由业务对象模型来描绘, 后者可由工作流模型来刻画。工作流模型描述了一个业务对象的多个活动和多个业务对象之间的活动及其运行关系, 按照用户需要的方式、顺序、衔接关系、控制关系所进行的动态组合。工作流模型 $WFM = (BO.DS, BO.AT, CON, ATRelation, RoleSet)$, 其中 $BO.DS$ 表示指定业务对象的数据集; $BO.AT$ 是指定业务对象的活动集, $BO.AT = (control IO, data IO, support)$, $control IO = \{(x_i, x_o) \mid \text{当 } x_i \in event \text{ 则 } x_o \in event \text{ 或当 } x_i \in s \text{ 则 } x_o \in s\}$, $data IO = \{(d_i, d_o) \mid d_i, d_o \in BO.DS\}$, $support = \{s \mid s = role, role \in RoleSet\}$; $CON = \{condition_x \rightarrow y \mid condition \text{ 是活动 } x \text{ 到 } y \text{ 之间的转换条件集}\}$; $ATRelation$ 是活动与活动之间的转换关系, 可表示

为 $ATRelation = \{ f | y = f(x) \text{ iff } condition_{x, y} = true, x \in BO_i, AT, y \in BO_j, AT, i, j = 1, \dots, n \}$; $RoleSet = \{ role | role \text{ is } BO. AT \text{ 中涉及到角色} \}$.

ICE-CIM 层组织模型描述了企业组织部门、岗位设置、人员和职能等信息,但这些都是面向企业人员且存在于现实世界中的概念.因此需要将上述概念转换为软件范畴内可操作可实现的术语.ICE-PIM 层采用“角色”概念映射“岗位”概念,通过角色模型来描述企业组织、岗位和人员及其之间的隶属关系等内容.角色模型 $RoleModel = (RS, RF, PS, PF, RPR)$,其中 $RS = \{ r | r \text{ 是一个角色} \}$, $RF = \{ f | f = f_r(r_i, r_j), f_r \text{ 是 } r_i \text{ 与 } r_j \text{ 的关系}, r_i, r_j \in RS, i, j = 1, \dots, n \}$, RF 是角色之间包含或分类关系的集合. $PS = \{ p | p \text{ 是一个岗位} \}$, PF 是岗位之间的隶属关系, $RPR = \{ (r, p) \}$ 是角色承担岗位的对偶集.

综上所述,业务对象模型是 ICE-PIM 的核心,通过 workflow 模型将 BO 连接起来以完成不同的业务功能, BO 的数据、操作和状态为 workflow 模型提供了输入、输出和活动触发条件等必要信息.数据模型是全体 BO 的数据模式,是所有角色的业务对象视图.角色和岗位关系明确定义了角色承担的权限,具体体现在其对业务对象活动的支撑和约束作用中.

5 ICE-PSM

ICE-PSM 向上需要承接软件设计内容并将其转换为特定运行平台上的系统实现模型,向下它是模型转换与自动生成代码的基础,其正确性与优化程度决定了软件的性能,它也是软件设计与开发人员沟通的媒介.软件系统的一个 PIM 模型映射到不同的实现平台将得到不同的 PSM 模型.即使在相同平台上实现的软件也体现出不同的软件模式,如风格、操作流程或实现机制等差异.在指定平台上找到一组能够实现需求软件所有特征的软件模式成为当务之急.本文基于 J2EE 平台明确了一组软件模式 $ESA-P = (BC-Ps, Control-P, Call-P)$, $BC-Ps = \{ BC-P | BC-P \text{ 业务构件的一种实现与运行模式} \}$, $Control-P$ 表示应用框架对全体构件的统一控制模式, $Schedule-P$ 表示 workflow 对全体构件的统一调度模式,其中每个模式又都是一系列小粒度模式的组合.

ICE-PSM 模型相对于 ICE-PIM 模型增加了基于 $ESA-P$ 的实现机制和系统结构等平台相关细节.业务构件模型(BC Model, BCM)是业务对象基于某种 $BC-P$ 的大粒度构件实现模型. $BCM = (BCID, BC-P, AM, CM, SI)$,其中 $BCID$ 表示业务构件的唯一标识; BC 总体上由基本构成部分模型 AM 、配置模型 CM 和服务接口声明模型 SI 构成. AM 包括表示层(Present Level, PL)、业务

逻辑层(Logic Level, LL)和数据层(DataBase Schema, DBS)三个层次以及其间的映射关系.表示层是与用户交互的接口 $AM = (PL, LL, DBS, Mapping(PL \rightarrow LL), Mapping(LL \rightarrow DBS))$, $PL = \{ UI | UI \text{ 是基于某种用户模式的用户界面} \}$,每个界面包含强语义关联性的数据与操作集.业务逻辑层是响应用户请求并处理的核心逻辑程序 $LL = (CP, AP, EP)$,核心控制器 CP 负责编排服务与调用方法,默认生成部分 AP 由代码生成工具自动获取,定制开发部分 EP 是根据用户需求由程序员扩展的部分.这三者相分离有助于提高模型的复用度. DBS 是业务构件的数据外模式. $Mapping(PL \rightarrow LL) = (PL. Data \rightarrow LL. variable, PL. Oper \rightarrow LL. method)$,表示 PL 的数据与 LL 层中变量的对应关系, PL 的操作被事件触发后通过方法选择符调用 LL 中相应的方法. $Mapping(LL \rightarrow DBS)$ 表示 LL 与 DBS 的成员变量与数据库字段之间的映射. BCM 具有许多随角色权限和用户关注点而不同的可变属性,通过控制这些属性能够进一步提高模型的可用度.从可变属性与其配置逻辑关系的角度 CM 描述了业务构件的功能、功能与角色/用户的映射关系. $CM = (CA, F, Role, User, Mapping(Role \rightarrow F), Mapping(User \rightarrow F), Mapping(F \rightarrow CA))$. F 为业务功能集, CA 是可变属性组, $Mapping(F \rightarrow CA)$ 定义业务功能由不同属性组构成. $Mapping(Role \rightarrow F)$ 和 $Mapping(User \rightarrow F)$ 分别表示角色和用户对功能的使用关系.

可执行 workflow 模型是基于先前抽取的统一调用模式在 workflow 模型基础上增加了更多的实现细节,依据选定的 workflow 引擎所支持的语言进行描述.全局静态配置模型(Static Configurable Model, SCM)由业务对象和角色模型中的控制相关内容转换得到. $SCM = (SCMID, Schedule-P, BC Config, Role-BC Config, Role Config, Position Role Config)$. $SCMID$ 表示 SCM 的唯一标识, SCM 以应用框架对 BC 的统一控制模式为基础,负责管理系统中业务构件($BC Config$)、业务构件与角色的映射关系($Role-BC Config$)、角色($Role Config$)、角色与岗位的映射($Position Role Config$)等,为系统组装和动态配置提供公共信息.全局数据模式是 ICE-PIM 数据模型在关系型数据模式上的映射结果,将数据实体都映射为数据库模式,将业务对象相关的数据模型映射到数据库的外模式.并建立数据模型与企业实际运行环境中数据结构的映射关系,从模型级支持数据视图的复用以降低适配工作量.

模式化的 ICE-PSM 严格规范了软件实现模型,简化了复杂系统的构造过程,可支持构件开发工业化.采用 XML+ 描述为软件的自动生成提供了可操作可执行的标准数据源.

6 ICE-CODE

ICEMDA 基于转换模式和局部转换再组装的思想来指导系统自动生成与构造的全过程。归纳业务构件模型中各组成部分的映射规则抽象出可信的转换模式。即将各部分映射为相应的代码片段,再将其组装成完整的业务构件。因此,业务构件是由若干小粒度构件(如文本框或下拉列表等基本控件)和其间的关联过程,按照用户的需求组装成一个具有业务语义完整性的构件。其内的可变属性的实例化过程可被分解到构件生成、部署与应用三个阶段。业务构件解析不同阶段的配置文件体现出多种形态以满足不同的业务需求。在构件生成阶段体现为尚未实例化的干业务构件(Stem Business Component, SBC),其权限相关或个性化相关的属性仍为参数形式;在部署阶段体现为面向特定需求的复用业务构件(Reused Business Component, RBC),是将 SBC 中可变属性转变为固化属性后的角色/用户相关构件。在应用过程中则是个性化构件,可依据使用者的个性化需求而不同。业务构件的形态划分与动态配置方法可有效降低具有按需应变能力的软件系统的开发难度。

基于业务构件的增量式软件开发首先是利用代码生成工具获取干业务构件;再借助工作流、关联机制和扩展机制,链接形成完整的业务流程。软件部署人员配置构件权限或个性化相关内容,更新系统部署模型确定实际环境中的管理软件组装。基于业务构件和工作流技术的 ESA 软件能够满足由业务需求变化引起的系统重构、流程改造等优化要求,这种软件具有高度的适应性和可生存性。ICE-CODE 利用一系列的自动化工具从模型角度描述所见即所得的软件系统,屏蔽了复杂的技术细节。另一方面,基于动态配置的系统重构技术,降低了业务过程重构成本,提高了对需求变化的响应速度。

7 转换建模与支持工具

ICEMDA 框架中的三个转换集中体现了 ESA 软件的可构造性:(1) ICE-CIM 到 ICE-PIM 的转换:将一个个具体的信息实体转换为抽象的业务对象,使其尽可能地接近直观的单据。将过程模型进一步抽象为基于业务对象的工作流模型,同时业务流程中涉及到的组织信息将被转换为面向软件的角色模型;(2) ICE-PIM 到 ICE-PSM 的转换:将 PIM 层的设计模型映射到确定平台上基于模式的系统实现模型,描述以

具有可变属性的业务构件模型为中心的全局动态和静态关系;(3) ICE-PSM 到 ICE-CODE 的转换:由模型自动生成构件的可执行程序,为组装具有按需应变能力的软件系统提供大粒度要素。ICEMDA 的模型转换思想是建立各层次上核心概念之间的映射与转换关系,如图 2 所示。

自顶向下的贯穿了若干条以概念为主体的转换关系链,如“信息实体 业务对象 业务构件模型 干业务构件 复用业务构件 个性化构件”,“信息流 工作流模型 可执行工作流模型 部署态的可执行工作流模型+工作流引擎”,“业务对象的数据集 业务对象的数据实体 面向业务对象的数据子模式 全局数据模式 基于关系模式的数据模型 SQLServer 数据表/视图”,“企业组织 角色/岗位 静态配置模型 静态配置文件”。这些转换关系链将企业中具体的概念转换为抽象的概念,进而转换为面向软件实现与运行的要素,为模型驱动的软件开发提供了一套顺畅的映射与转换思路。若辅以严格详细的映射规则和转换机制将有效支持各层次上模型要素的一致性和可追溯性。

基于 ICEMDA 的软件开发需借助一系列自动化工具予以支持才能真正实现模型构造到系统自动生成的通路。本文已基本实现了支持工具集原型 ICEMDA-Tools,如图 3 所示。ICEMDA-Tools 的主要功能包括:(1) 各层友好的图形化建模:用户使用标准的概念和图元借助建模工具即可描述各层模型;(2) 建模工具的可配置功能:将三层模型的要素相集成,用户通过设置建模工具的图元或环境形成不同的建模配置方案,以支持不同层次模型的建模;(3) 支持 ICEMDA 模型与 UML 模

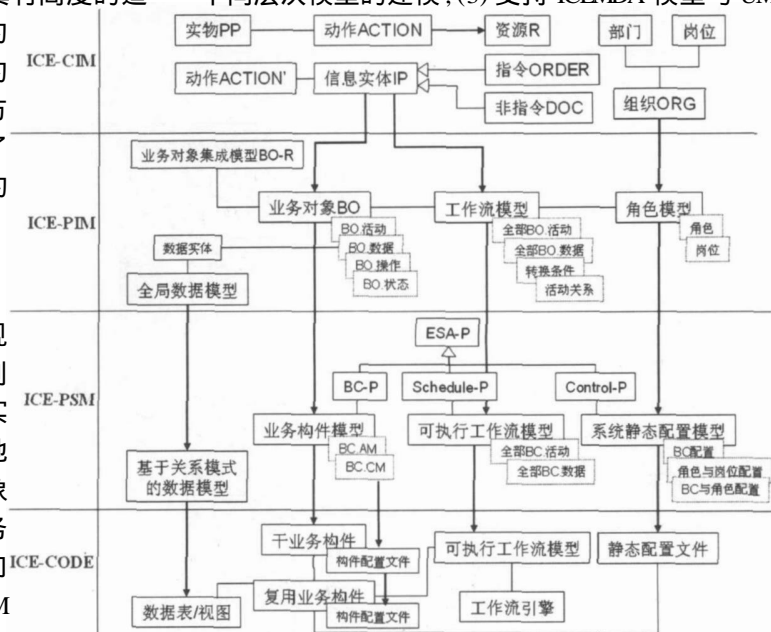


图2 以概念为主体的模型转换思想

- [11] D Chen, B Vallespir, G Doumeings. GRAI integrated methodology and its mapping onto generic enterprise reference architecture and methodology[J]. Computers in Industry, 1997, 33 (2 - 3) : 387 - 394.
- [12] 尚文利, 王成恩, 张士杰, 尹朝万. 基于 IDEF 与 UML 的系统建模方法[J]. 计算机集成制造系统, 2004, 10 (3) : 252 - 275.
- Shang Weng-li, Wang Cheng-en, Zhang Shi-jie, Yin Chaowan. IDEF & UML based system modeling method[J]. Computer Integrated Manufacturing Systems, 2004, 10 (3) : 252 - 275. (in Chinese)

作者简介:



战德臣 男, 1965 年生. 博士, 教授, 博士生导师, CCF 高级会员. 主要研究领域为软件复用与软件体系结构、模型驱动架构、现代企业管理、数据与知识工程.

E-mail: dechen@hit.edu.cn



冯锦丹 女, 1980 年生于辽宁锦州. 哈尔滨工业大学计算机科学与技术学院博士生, 研究方向为软件复用、软构件与代码生成.

E-mail: fjd1127@163.com

聂兰顺 男, 博士, 讲师, 主要研究领域为软件体系结构、软件复用、大型管理软件.

徐晓飞 男, 院长, 博士, 教授, 博士生导师, CCF 高级会员. 主要研究领域为管理与决策信息系统、数据库、企业资源计划与供应链管理技术, 服务科学.