

支持 MDD 的体系结构模型的形式化语义*

侯金奎, 马 军

(山东大学 计算机科学与技术学院, 山东 济南 250101)

摘 要: 为解决模型驱动开发中模型映射关系的定义和模型转换的正确性验证等方面存在的问题, 利用范畴理论形式化描述软件体系结构模型及其间的关系, 使之具有严格的语义。态射合成被用来追踪构件模型之间的关联和映射关系, 不同抽象层次的体系结构模型之间的一致性由函子来维持。范畴理论支持图形化建模, 可以使模型中的构件关系以及结构特征可视化, 有利于对模型转换的理解和追踪。应用研究表明, 该方法不仅可为验证模型之间映射规则的正确性提供依据, 还能为模型转换的具体实现提供理论指导。

关键词: 计算机软件; 模型驱动开发; 模型映射; 软件体系结构; 形式化语义

中图分类号: TP311 **文献标识码:** A **文章编号:** 0529-6579 (2008) 06-0109-06

模型转换是模型驱动开发 (Model-driven Development, MDD) 中的一项关键技术, 模型间的映射关系是模型转换的基础和依据^[1]。已提出的模型转换方法^[2]大都侧重于不同抽象层次模型之间转换的具体实现技术的研究, 而在映射关系的定义原则, 以及模型转换的正确性验证和分析等方面, 还缺乏成熟的理论基础和验证工具。Harel 等^[3]对模型描述语言的语法结构和语义表达作了详细的介绍, 但没有涉及模型转换过程中语义的迁移和重构。Bezivin 等^[4]从元模型语义的角度分别介绍了基于 UML 和 EDOC 的平台无关模型到 JAVA、Web service 和 JWS DP 三种目标平台模型的映射关系, 并简单讨论了模型描述之间的异构性对映射关系定义的影响, 但没有给出一般情况下的解决方案。Caplat 等^[5]以扩展形式化语言的方式对不同层次模型之间的转换分类作了说明, 该方法可以在一定程度上降低模型描述之间的异构性, 但模型之间的映射关系仍难以直接定义。

范畴理论^[6]为软件体系结构描述提供了恰当的数学抽象层次^[7], 其抽象框架能为复杂系统的构件配置提供正确的语义描述, 通过把应用合成算子后的运算结果描述为递归定义的系统构件, 从而用图表来表达配置。本文扩展了这种思想, 对范畴论框架作了改进, 使之同时为体系结构模型的配置以及模型间的映射关系提供精确的语义描述, 可用于 MDD 开发中指导不同抽象层次模型间映射规则的定义以及模型转换的实现与验证。

1 范畴理论

参考文献 [6], 本节给出了范畴理论的一些基本概念。

定义 1 范畴 (Category) 一个范畴 C 包含以下两个集合:

- (1) 一个称作是 C -objects 的对象集合;
- (2) 一个称作是 C -arrows 的态射 (箭头) 集合;

这两个集合必须符合以下特性:

(3) 每个态射 f 都有一个相关联的对象 A (称为 f 的定义域) 和一个对象 B (称为 f 的值域), 记作 $f: A \rightarrow B$;

(4) 对所有的态射 $f: A \rightarrow B$ 和 $g: B \rightarrow C$, 存在一个合成态射 $g \circ f: A \rightarrow C$, 并且满足结合律, 即对所有的 $h: C \rightarrow D$, 有 $h \circ (g \circ f) = (h \circ g) \circ f$;

(5) 对范畴中的每个对象 A , 存在单位态射 id_A , 满足条件 $\forall f: B \rightarrow A, id_A \circ f = f$ 并且 $\forall f: A \rightarrow B, f \circ id_A = f$ 。

定义 2 图表 (Diagram) 范畴 C 的一个图表 D 包含一个 C -objects 的集合 D_C 和一个 C -arrows 的集合 D_A , 并且对所有的箭头 $a \in D_A$, $\text{cod } a \in D_C$, $\text{dom } a \in D_C$, 这里 $\text{cod } a$ 表示 a 的值域, $\text{dom } a$ 表示 a 的定义域。

定义 3 余极限 (Colimit) 范畴 C 的图表 D 的一个余极限是一个 C -object C , 沿着从 D 到 C 的一个余锥 (Co-cone) $\{f_i: D_i \rightarrow C \mid D_i \in D\}$, 对所

* 收稿日期: 2008-09-10

基金项目: 山东省科技攻关资助项目 (2008GG10001026)

作者简介: 侯金奎 (1976 年生), 男, 博士研究生; 通讯联系人: 马军; E-mail: hjk@mail.sdu.edu.cn

有其它从 D 到顶点 C' 的余锥 $\{f'_i: D_i \rightarrow C' \mid D_i \in D\}$, 有唯一的 C -arrow $f: C \rightarrow C'$, 对 D 中所有的对象 D_i , 图 1 所示的图可交换, 即 $f \circ f_i = f'_i$ 。

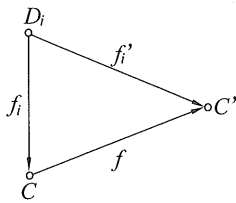


图 1 余极限

Fig. 1 Definition of a colimit

定义 4 函子 (Functor) 由范畴 C 到范畴 D 的一个函子 F 是一个映射, 它把每一个 C -object a , 都指派给一个 D -object $F(a)$, 把每一个 C -arrow $f: A \rightarrow B$, 都指派给一个 D -arrow $F(f): F(A) \rightarrow F(B)$, 并且保持了单位态射和结合律, 即对所有的 C -objects A , 都有 $F(id_A) = id_{F(A)}$; 并且若 C 中定义了 $g \circ f$, 则 $F(g \circ f) = F(g) \circ F(f)$ 。

2 体系结构模型的范畴论语义

2.1 构件规范

定义 5 构件标识 (Component signature) 一个构件标识是一个 3 元组 $\Sigma = (S = A \cup \Gamma, \Omega)$, 其中

- (1) S 是构件规范中所有标识符的集合;
- (2) A 是构件的所有属性的集合;
- (3) Γ 是由构件的接口构成的集合;
- (4) $\Omega: \Gamma \rightarrow 2^A$, 对每一个 $r \in \Gamma$, $\Omega(r)$ 表示通过接口 r 可以修改的属性构成的集合。

定义 6 构件标识态射 (Component signature morphism) 给定两个构件标识 $\Sigma_1 = (S_1 = A_1 \cup \Gamma_1, \Omega_1)$ 和 $\Sigma_2 = (S_2 = A_2 \cup \Gamma_2, \Omega_2)$, 从 Σ_1 到 Σ_2 的构件标识态射 $\delta: \Sigma_1 \rightarrow \Sigma_2$ 是一个标识符映射 $\delta_s: S_1 \rightarrow S_2$, 并且

- (1) $\delta_s(S_1) \subseteq S_2$;
- (2) 对每一个 $r \in \Gamma_1$, $\delta_s(\Omega_1(r)) = \Omega_2(\delta_s(r))$, 即构件标识态射保持某一接口所影响的属性集合不变。

命题 1 给定三个构件标识 $\Sigma_1 = (S_1 = A_1 \cup \Gamma_1, \Omega_1)$, $\Sigma_2 = (S_2 = A_2 \cup \Gamma_2, \Omega_2)$ 和 $\Sigma_3 = (S_3 = A_3 \cup \Gamma_3, \Omega_3)$, 如果存在态射 $\delta_1: \Sigma_1 \rightarrow \Sigma_2$ 和 $\delta_2: \Sigma_2 \rightarrow \Sigma_3$, 那么存在一个合成态射 $\delta_2 \circ \delta_1: \Sigma_1 \rightarrow \Sigma_3$ 。

由定义 6, 命题 1 显然成立, 它可用于跟踪不同构件标识态射之间的关系。

定义 7 构件规范 (Component specification)

一个构件规范是一个 2 元组 $CS = \langle \Sigma, \Phi \rangle$, 其中 $\Sigma = (S, \Omega)$ 是构件标识, Φ 是构件的功能目标和非功能目标的集合。规范 $CS = \langle \Sigma, \Phi \rangle$ 的一个 Σ -模型 $M[CS]$ 满足对所有的 $g \in \Phi$, 有 $M[CS] \models g$, $\models$ 表示逻辑蕴含。

定义 8 构件规范态射 (Component specification morphism) 给定两个构件规范 $CS_1 = (\Sigma_1, \Phi_1)$ 和 $CS_2 = (\Sigma_2, \Phi_2)$, 其中 $\Sigma_1 = (S_1, \Omega_1)$, $\Sigma_2 = (S_2, \Omega_2)$ 。由 CS_1 到 CS_2 的构件规范态射 $\sigma: CS_1 \rightarrow CS_2$ 是一个标识态射 $\delta: \Sigma_1 \rightarrow \Sigma_2$, 对所有的 $g \in \Phi_1$, $M[CS_1] \models g$ 当且仅当 $\sigma(g) \in \Phi_2$ 并且 $M[CS_2] \models \sigma(g)$ 。

构件规范态射描述了构件标识以及这些标识相关联的特性之间的关系, 同时它还是构件之间交互的机制。

2.2 体系结构模型

在面向构件的模型驱动开发中, 构件模型之间的关系有多种^[8]: 组合 (compose)、使用 (use)、扩展 (extend) 以及不同抽象层次构件模型之间的映射 (mapping) 关系等。Compose 关系表示一个构件是另一构件的一部分, 通过这种态射的递归应用, 再添加一些相应的约束, 复杂构件可以由简单构件通过层次组合配置来表示, 如图 2 中 $c_1, c_2, \dots$ 为组合态射, 复合构件 C 为构件 C_1, C_2, C_3 的余极限。Use 和 Extends 关系描述了两个构件规范或者两个构件实现之间的依赖, 即两个不同但又相关的构件之间的关系, 它表明一个构件中某些功能的实现以另一构件中的某些功能为基础, 如图 2 中的 $f_1, f_2, \dots$ 所示。Mapping 关系描述了抽象构件规范和具体构件规范之间的映射关系, 如图 4 中的虚线箭头所示。

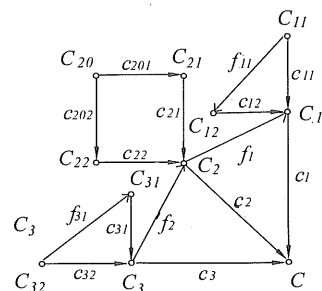


图 2 构件模型的层次组合

Fig. 2 Hierarchical composition of component models

本文借鉴陆汝钊院士提出的类型范畴^[9]来描述软件体系结构模型。类型范畴是一种可以为对象和态射定义类型的范畴, 且每一种类型都可以定义一系列的属性。

定义9 体系结构模型 (Architecture Model)

体系结构模型是一个五元组 $AM = \langle CO, CR, OT, RT, RuleS \rangle$, 其中: CO 是由作为对象的构件实例组成的集合, CR 是由作为态射的构件关系实例构成的集合, OT 是构件类型描述规范的集合, RT 是构件关系类型的集合, $RuleS$ 是构件间关系类型合成机制的集合, 除满足范畴定义的基本条件外, 还要符合以下特性:

(1) $CO = \{o_i \mid 1 \leq i \leq n, \text{sort}(o_i) \in OT\}$, sort 是类型函数;

(2) $CR = \{r_j \mid 0 \leq j \leq m \exists a, b \in CO, r_j = (a, b, t), t = \text{sort}(r_j) \in RT\}$;

(3) $RuleS: RT \times RT \rightarrow RT$, 对任意 $(t, s) \in \text{dom}(RuleS)$, 类型 $w = t \times s$ 称作是 t, s 的合成类型, 并且对给定的 $r_i, r_j \in CR, r_i = (a, b, t), r_j = (b, c, s)$, 存在 $r_k = (a, c, w) = r_j \circ r_i \in CR$.

一个体系结构模型就是由构件规范和构件规范态射及其实例构成的类型范畴。期望构件体现或者需要对构件进行证明的特性可以与规范相关联, 通过在规范框架中描述这些特性, 使用范畴理论来计算和管理这些特性 (如余极限规范的特性给出了特性的组合), 其优点是通过对态射和证明义务的管理, 以一种统一的方式来处理这些特性及其状态。

3 体系结构模型映射

本文中的“映射”特指由较高抽象层的构件规范到较低层规范 (也可以是实现) 的映射关系。

3.1 构件模型之间的基本映射关系

一般说来, 不同抽象层次构件模型之间的基本映射关系有3种: 直接映射、分解映射和组合映射。图3(a)表示直接映射, 这是最简单的一种映射关系, 也称作是“1-to-1”映射。它直接把构件规范 S 映射至另一构件规范 T , T 是 S 在较低抽象层次上直接对应的构件规范。分解映射又称为“1-to- m ”映射, 其目标 T 由若干子构件 $\{T_i \mid i \leq n\}$ 组成, 如图3(b)所示。这种情况下, 我们添加从 S 到每个 T_i 的态射 v_i , 并且 $c_i \circ v_i = m$, c_i 是从 T_i 到 T 的 compose 态射, m 是从 S 到 T 的 mapping 态射。态射集合 $\{S \rightarrow T_i \mid i \leq n\}$ 就表示源构件规范中功能目标和非功能目标的一种分解方式, 形成“责任-指派”关系。图3(c)表示组合映射, 也称作“ m -to-1”映射, 其源规范 S 是由若干子构件 $\{S_i \mid i \leq n\}$ 组成。与分解映射类似, 我们添加从每个 S_i 到 T 的态射 v_i , 并且 $m \circ c_i = v_i$, 其中 c_i 是从 S_i 到 S 的 compose 态射, m 是从 S 到 T 的 mapping 态射。态射集合 $\{S_i \rightarrow T \mid i \leq n\}$ 意味着在

每个子构件规范 S_i 中定义的公理都必须在目标规范 T 中得到满足。

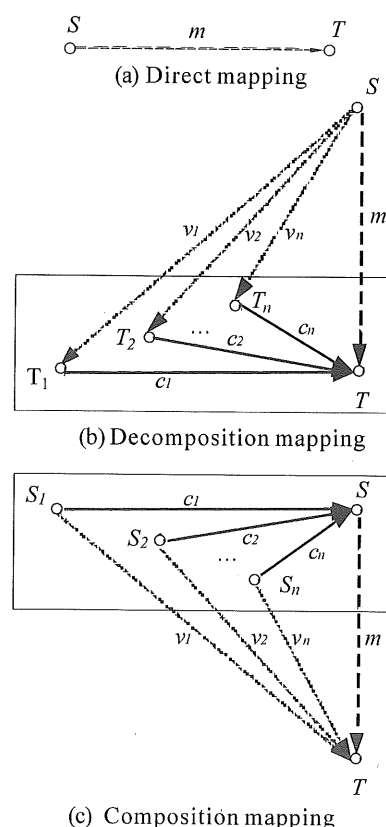


图3 构件模型之间基本的映射关系

Fig. 3 Basic mapping relations between component models

实际中, 如果两个构件规范之间的映射关系很复杂, 如“ m -to- n ”的映射形式, 则可通过添加中间规范的方式进行简化, 也就是说, 一个“ m -to- n ”映射关系可分解为一个“ m -to-1” (组合映射) 关系和一个“1-to- n ” (分解映射) 关系。

3.2 体系结构映射函子

定义10 体系结构映射函子 (Architecture mapping functor) 从体系结构模型 $AM_1 = \langle CO_1, CR_1, OT_1, RT_1, RuleS_1 \rangle$ 到 $AM_2 = \langle CO_2, CR_2, OT_2, RT_2, RuleS_2 \rangle$ 的结构映射函子 $F: AM_1 \rightarrow AM_2$ 是满足以下条件的映射:

- (1) 对所有的 $cs \in CO_1$, 都有 $F(cs) \in CO_2$;
- (2) 对所有的 $t, s \in RT_1$, 都有 $F(t), F(s) \in RT_2$, 并且 $F(t \times s) = F(t) \times F(s)$;
- (3) 对所有的 $cs, cs' \in CO_1, (cs, cs', t) \in CR_1$ 都意味着 $F(cs), F(cs'), F(t) \in CR_2$;
- (4) $F(f \circ g) = F(f) \circ F(g)$, 若 $g, f \in CR_1$ 且定义了 $f \circ g$ 。

体系结构映射函子本质上是模型之间 (或模

型与程序之间)的一个转换,它保持了源构件规范的实现方式,反映了设计决策。

4 实例研究

本节以一个构件化的邮件客户端为例验证本文所提理论和方法的可行性。

为了简化问题的描述,此处邮件客户端的主要功能有:收发邮件、撰写邮件、阅读邮件、邮件管理、地址簿管理、多账户管理。该邮件客户端支持文本格式和 html 格式邮件的接收、阅读和撰写。

基于构件的层次组装,图 4 左半部分用范畴图描述了其构件化的源模型结构,涉及到 7 个构件类型:①MainUI 负责邮件客户端与用户交互部分的界面布局 and 美工,用户通过它进行收信、发信、写信等操作;②EmailManagement 负责对存放在本地的所有电子邮件进行存储、读取和显示;③Editor 用于撰写文本格式或 html 格式的电子邮件;④Client 负责发送和接收邮件;⑤Protocol 负责邮件收发协议的设置;⑥AddressBook 负责地址簿管理;⑦Account 负责账户管理。

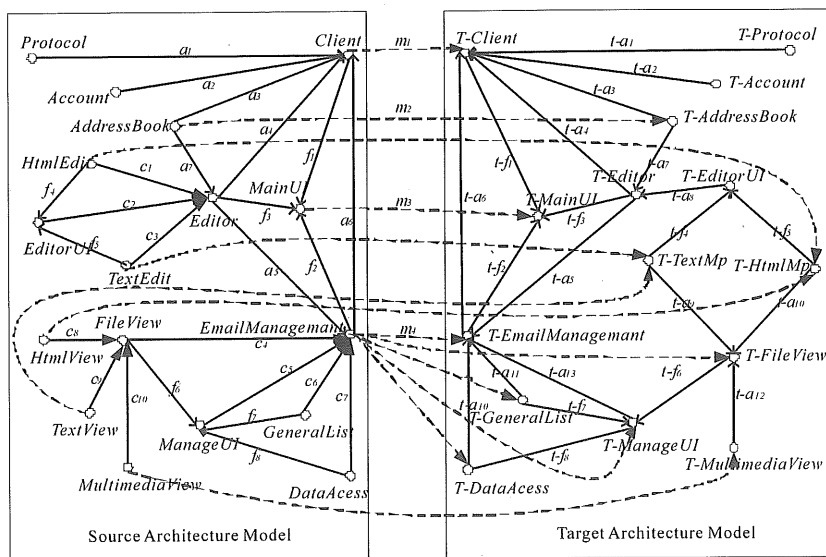


图 4 邮件客户端的源模型的目标模型

Fig. 4 The source model and the target model of the email client system

这里,EmailManagement 和 Editor 是两个复合构件。EmailManagement 内部使用一个通用的列表构件 GeneralList 来处理邮件列表,使用数据存取构件 DataAccess 来存取电子邮件,使用构件 FileView 来显示不同类型的电子邮件内容,ManageUI 是相应的界面展示构件。其中,构件 FileView 又是由 Html 处理构件 HtmlView、文本处理构件 TextView 和多媒体处理构件 MultimediaView 三个子构件复合而成的。在构件 Editor 内部,子构件 EditorUI 负责界面展示的处理,子构件 TextEdit 和 HtmlEdit 分别用于相应格式的电子邮件内容的编辑。

假定目标平台的实现不支持复合构件(如 C++ 不支持类的嵌套定义,EJB 规范仅允许一个 jar 包中有多个 EJB,但不支持复合 EJB),那么对于源模型中的大多数单一构件都能很容易与目标平

台的构件类型直接建立映射关系,但所有的复合构件都必须通过逐层分解建立到目标平台构件类型的分解映射。为了优化目标平台的体系结构,最后一般还需要对目标平台的构件类型进行整合,从而又会形成从源到目标的组合映射。如目标模型中将 html 文件的编辑和浏览功能合并在一个构件 T-HtmlMp 中,将文本文件的编辑和浏览合并在一个构件 T-TextMp 中。

图 4 右半部分是该系统在目标平台上对应的体系结构模型范畴图,映射关系如表 1 所示,包含了构件模型映射的三种基本形式。图 4 种的映射关系满足范畴图的可交换性,如 $t-f_1 \circ m_1 = m_3 \circ f_1$, $t-f_3 \circ t-a_7 \circ m_2 = m_3 \circ f_3 \circ a_7$ 等,表明该转换保持了构件间依赖关系的一致性。

表 1 构件模型间的映射关系

Tab. 1 The mapping relations between component models

Source model	Target model	Mapping forms
Client	T-Client	direct
MainUI	T-MainUI	direct
Protocol	T-Protocol	direct
Account	T-Account	direct
AddressBook	T-AddressBook	direct
Editor	{T-Editor, T-EditorUI, T-HtmlMp, T-TextMp}	decomposition
EmailManagement	{T-EmailManagement, T-DataAccess, T-GeneralList, T-ManageUI, T-FileView, T-HtmlMp, T-TextMp, T-MultimedisView}	decomposition
EditorUI	T-EditorUI	direct
HtmlEdit	T-HtmlMp	composition
TextEdit	T-TextMp	composition
DataAccess	T-DataAccess	direct
GeneralList	T-GeneralList	direct
ManageUI	T-ManageUI	direct
FileView	{T-FileView, T-HtmlMp, T-TextMp, T-MultimedisView}	decomposition
HtmlView	T-HtmlMp	composition
TextView	T-TextMp	composition
MultimedisView	T-MultimedisView	direct

5 总 结

本文以形式化、高抽象的范畴理论为基础, 为基于构件的体系结构模型及其间的映射关系提供了一种统一的语义描述框架。范畴理论支持图形化建模, 可以使模型中的构件关系以及结构特征可视化, 有利于对模型转换的理解和追踪。应用研究表明, 这不仅可为验证不同抽象层次模型之间映射规则的正确性提供依据, 还能为模型转换的具体实现提供理论指导, 从而为模型驱动的软件开发提供有力的支持。

今后的工作包括: ①进一步形式化定义构件规范和体系结构模型, 从而提高模型的语义表述能力; ②探讨模型映射所应保持的语义特性以提高转换的正确性; ③对一些比较复杂的一致性保持规约进行形式化描述, 并设计出可行的验证算法。

参考文献:

- [1] BRENT H, PERI T. Model-driven development: The good, the bad, and the ugly [J]. IBM Systems Journal, 2006, 45(3): 451 - 461.
- [2] KRZYSZTOF C, SIMON H. Feature-based survey of model transformation approaches [J]. IBM Systems Journal, 2006, 45(3): 621 - 644.
- [3] HAREL D, RUMPE B. Meaningful modeling: what's the semantics of 'semantics' [J]. IEEE Computer Magazine, 2004, 37(10): 64 - 72.
- [4] BEZIVIN J, HAMMOUDI S, LOPPE D, et al. Applying MDA approach for Web service platform [C]//Proc of Enterprise Distributed Object Computing Conference 2004, Washington: IEEE Computer Society, 2004: 58 - 70.
- [5] CAPLAT G, SOURROUILLE J L. Model mapping using formalism extensions [J]. IEEE Software, 2005, 22(2): 44 - 51.
- [6] BARR M, WELLS C. Category theory for computing science [M]. New Jersey: Prentice-Hall, 1990.
- [7] FIADREIRO J L, MAIBAUM T. A mathematical toolbox for the software architect [C]//Proc of 8th International Workshop on Software Specification and Design, Washington: IEEE Computer Society, 1995: 46 - 55.
- [8] GUO J. Using category theory to model software component dependencies [C]//Proc of the 9th Annual IEEE Int'l Conf. and Workshop on the Engineering of Computer-Based Systems, Washington: IEEE Computer Society, 2002: 185 - 192.
- [9] LU R Q. Towards a mathematical theory of knowledge [J]. Journal of Computer Science and Technology, 2005, (20)6: 751 - 757.

(下转第 119 页)

- hoc networks[C]//Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications 2003.
- [12] THEODORE S R. Wireless communications: principles and practice[M]. Prentice Hall PTR, 1996.
- [13] 王福豹, 史龙, 任丰原. 无线传感器网络中的自身定位系统和算法[J]. 软件学报, 2005, 16 (5): 857 - 868.

Environment Aware Node Localization Method for Wireless Sensor Networks

DENG Gang, CHEN Ying-wen, XU Ming, WANG Sen

(School of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: Node localization is one of the essential techniques for wireless sensor networks, especially for location - dependent applications. As is known, the RSSI range-based localization algorithms have a considerable error in distance measure. To address this issue, we propose the Environment Aware Node Localization Method (EAM). In EAM, the whole covering area of the sensor network is divided into several sub regions, and then the method can figure out the parameters of the propagation model in the corresponding sub region with the cooperation of each beacon node. Based on the propagation model in each sub region, EAM can improve the precision of the distance measurement. Simulation results show that the average distance measurement error can be reduced about 15 % to 30% by adopting EAM.

Key words: node localization; range-based; RSSI; environment aware

(上接第 113 页)

MDD-supported Formal Semantics Description of Architecture Models

HOU Jin-kui, MA Jun

(School of Computer Science and Technology, Shandong University, Jinan 250101, China)

Abstract: In order to resolve the problems in the definition of model mapping relations and in the validation of the correctness of model transformations in model-driven development, this paper use category theory to provide a precise semantics for architecture models and their relationships. Morphism composition is used to trace the interconnections and mapping relations between component models, while the consistency between architecture models at different abstract levels is maintained by functors. Category theory supports the diagrammatic representation of component models that visualizes the relationships between components and the structural features, which can be used to strengthen the understandability and traceability of model transformation. The application research shows that, the description framework does not only provide a measurement for validating the mapping rules, but also provide a theoretical guidance for model transformations.

Key words: computer software; model-driven development; model mapping; software architecture; formal semantics