

基于领域特征空间的构件语义表示方法*

贾育, 顾毓清

(中国科学院 软件研究所, 北京 100080)

E-mail: jia_yu@263.net; guyq@sinosoftgroup.com

http://www.iscas.ac.cn

摘要: 在基于构件的软件开发范型中,构件模型是构件分类、检索和适应等关键任务的理论基础,然而构件模型中的语义表示方法一直是一个难点.提出一种以语义为核心的构件模型,采用基于领域分析的特征空间语义定义方法,从领域空间、定义空间和语境空间三个方面刻画语义结构,用描述逻辑表达语义内容及实现自动推理,尤其是文中的构件语义体概念和特征空间形式化方法,为构件复用自动化和工程化提供了一种可能的解决途径.

关键词: 软件构件;构件语义;特征空间;软件复用;描述逻辑;Z语言

中图法分类号: TP311 文献标识码: A

软件复用是一种重复使用软件资产或过程的软件开发方法^[1],目的是要提高软件质量和开发效率,目前,软件构件复用成为软件资产复用的主流技术.在基于构件的软件开发(component-based development,简称 CBD)^[2-4]过程中,人们发现,软件复用面临的主要问题不是如何从构件库中获得可用构件,而是怎样判断构件的有用性^[4].概括地,构件的有用性体现在语义和语法两个方面.当前,研究集中在语法有用性上,例如,CORBA^[5]注重对象封装规范和通信协议,但缺乏表达对象“做什么”的能力,使得语义只能以文档等媒介作为载体,或潜移默化地反映在 CBD 的活动之中,由此产生的非形式化和歧义性,无益于软件复用的自动化和工程化.因此,需要一种构件模型,把形式语义描述作为构件明确的组成部分,使构件的语义自描述成为可能,从而解决软件复用中的语义匹配和构件理解问题.

软件复用本质上是对知识的复用,构件功能的高度独立性和完整性,使其能够对应于不同层次和粒度的客观实体,如果能够获取客观实体的本质属性,那么相应的构件便有了直接的语义解释.因此,我们采用领域中用户和开发者都可理解的术语和概念“特征(feature)”^[6,7]来描述构件语义,建立一种层次化的特征空间(feature space)体系结构,使用描述逻辑(description logics,简称 DLs)^[8,9]表示特征、特征运算和推理规则,并从领域(domain)空间、定义(definition)空间和语境(context)空间 3 个侧面刻画语义内涵.本文中,说明性规格说明语言 Z^[10]被用作描述构件模型的工具.

1 构件语义模型

可复用构件模型是 CBD 的理论基础,在当今众多的模型中,3C 模型^[11]受到普遍认同,在此基础上,我们定义了一个具体的构件模型,充分强调了语义在构件中的重要地位.

定义 1.1 (构件 Component). 构件是指语义完整、语法正确和有可复用价值的单位软件,是软件复用过程中可以明确辨识的系统构成;结构上,它是语义描述、通信接口和实现代码的复合体.

[SEMANTICIZER, INTERFACE, IMPLEMENTATION]

$Component = SEMANTICIZER \times INTERFACE \times IMPLEMENTATION$

这里,SEMANTICIZER:语义体集合,语义体是对构件功能和属性的抽象和描述;INTERFACE:接口体集合,接口体

* 收稿日期: 2000-10-09; 修改日期: 2001-07-30

作者简介: 贾育(1966 -),男,安徽合肥人,博士生,助理研究员,主要研究领域为软件工程、人工智能;顾毓清(1940 -),男,江苏苏州人,研究员,博士生导师,主要研究领域为软件工程.

是描述构件的通信接口规范和语法约束;IMPLEMENTATION:实现体集合,实现体是满足接口规范和语义描述的实例。

以上三者之间,以语义体为主导,以接口体为条件,以实现体为目的,形成既逻辑统一,又物理独立的关系.接口体沿用 CORBA 等工业构件接口技术,语义体详细说明如下。

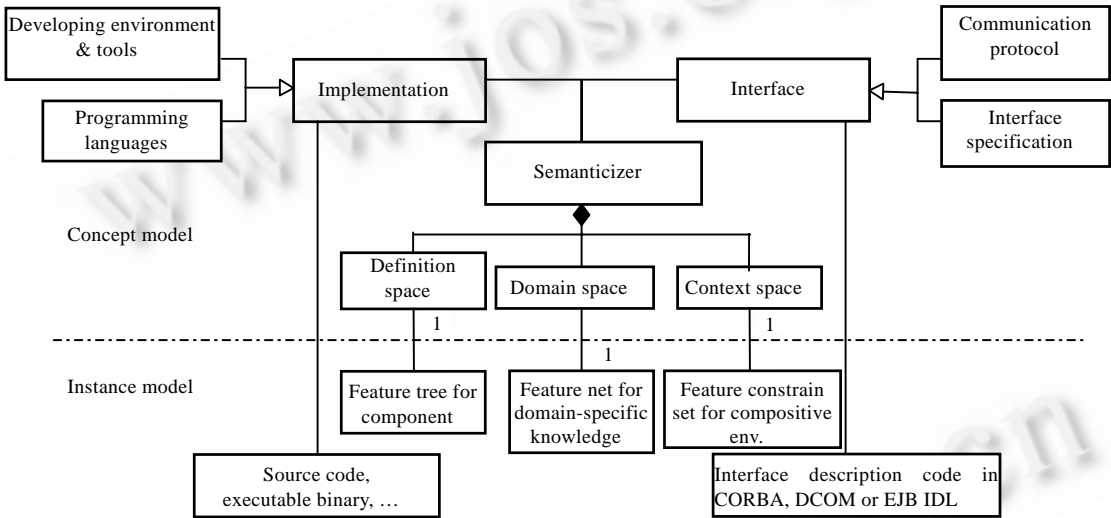
定义 1.2 (构件语义体, Semanticizer). 构件的语义指构件的含义和使用方法,是构件可复用价值的决定因素;构件语义体包括定义空间、领域空间和语境空间 3 个组成部分。

$$[\Omega_{def}, \Omega_{dom}, \Omega_{con}]$$

$$Semanticizer = \Omega_{def} \times \Omega_{dom} \times \Omega_{con}$$

这里,Ω_{def}:定义空间,表示构件语义的特征及其关系的集合;Ω_{dom}:领域空间,表示领域知识的特征及其关系的集合;Ω_{con}:语境空间,表示语义约束的特征条件及其关系的集合。

构件语义体又称 d²c 语义模型,它是用领域空间描述构件的领域知识语义网络,以定义空间表示构件的语义内涵,将构件的创建、检索、分类等活动限制在语境空间表示的语义环境中.3 个空间合理划分和相互约束,提高了构件语义表达的精确性、完整性和灵活性.图 1 是用 UML 绘制的构件模型示意图。



实现体, 接口体, 语义体, 定义空间, 领域空间, 语境空间, 运行环境, 程序设计语言, 通信协议, 接口规格说明, 构件特征语义树, 特定领域知识语义网络, 组装环境特征约束集, 源代码、可执行代码等, 用 CORBA、COM 和 EJB IDL 编写的接口规格说, 概念模型, 实例模型。

Fig 1. The semantics-oriented component model
图 1 面向语义的构件模型

2 特征的表达方法

2.1 特征

基于特征的方法^[7]广泛应用于许多领域,人工智能知识表示中的“框架-槽”、模式识别中的“特征”、多媒体数据库中的“特征向量”,所表达的概念都类似下面定义的“特征”。

定义 2.1(特征 Feature). 特征是描述构件语义的单位成分,是对现实世界知识的本质抽象,由领域中的术语或概念构成,其形式是属性-值的对偶。

[IDENTIFIER, DESCRIPTION]
Feature
id : IDENTIFIER
def : DEFINITION
interpretation : IDENTIFIER ↔ DESCRIPTION
∀ id ₁ , id ₂ : IDENTIFIER • interpretation (id ₁) = interpretation (id ₂) ⇒ id ₁ = id ₂

这里,IDENTIFIER 是特征标识符集合,DESCRIPTION 是特征的自然语言描述的集合,函数 interpretation 是用客观世界知识对特征的自然语言解释.约束条件说明,允许同一特征存在多个形式不同、含义一致的自然语言释义;DEFINITION 是定义特征的 DLs 断言的集合,被递归地定义为如下自由类型:

$$[META, CONSTRUCT]$$

$$DEFINITION ::= META \mid dl \ll \langle \langle META \times \langle \langle DEFINITION \times \langle \langle CONSTRUCT \rangle \rangle \rangle \rangle \rangle \gg$$

这里, META 是元特征的集合,元特征的类型是实型、字符型和布尔型等基本数据类型, CONSTRUCT 是 DLs 构造符的集合; $META \times \langle \langle DEFINITION \times \langle \langle CONSTRUCT \rangle \rangle \rangle \gg$ 是 DLs 合法字符组合的集合,构造函数 dl 按 DLs 语法从中构造出可满足的 DLs 断言表达式[8],规定了复合特征的合成方法.也即,对于被定义特征,可以通过不断分解和替换表达式中的复合特征,分层地将其展开成为仅含元特征的表达式.

特征值是特征的实例,定义如下:

$$FeatureValue \triangleq [id : IDENTIFIER; val : META \mid \exists x : Feature \bullet id = x.id \wedge val \in x.def]$$

2.2 特征空间

定义 2.2 (特征空间 Ω). 特征空间描述了语义的体系结构,它是特征和特征关系的集合.

$$\Omega$$

$fea : Feature$ $rel : Feature \times Feature$

按照特征之间的“一般-特殊”关系,可以把 Ω 看做一个树结构,树的每个结点是一个 Feature 结构的特征项,叶子结点代表元特征,非叶子结点代表复合特征.特征空间 Ω 通常只有一个根特征,由第 2.3 节的内容可知,特征关系隐含在特征定义 Feature.def 之中,因此我们简化 Ω 为仅包含 Feature 结构的集合.后面将要介绍,构成构件语义的 3 个空间在结构上都等同于 Ω .

2.3 特征描述逻辑

描述逻辑 DLs^[8]用论域中对象所属的类(称为“概念”)、类之间的关系(称为“角色”)和类的实例(称为“个体”)表示特定领域的知识库.DLs 的核心是概念描述语言,它通过一定构造符连接的表达式,描述概念之间的逻辑运算和相互关系,例如, \sqcap 表示特征的“与”运算, \sqsupseteq 表示“包含”断言,表达式“University \sqsupseteq INCLUDES.Departments”表示一所大学包含了若干系,有关构造符的深入介绍见文献^[8,9].

DLs 擅长业务领域建模,非常适合描述应用服务和特定领域服务,因此,在构件语义描述中,把它作为特征逻辑表达工具,概念成为特征的同义词,用含有特征和角色的断言表达式描述特征空间.此外,DLs 支持可确定的推理算法,为特征空间的合理性和一致性验证,以及 CBD 操作的自动化提供了保障,文献[9]给出了一个特征描述语言 FDL.

3 面向特征的构件语义表示

3.1 领域空间

领域空间是在分析领域知识的基础上,抽取客观实体的典型特征,按特征之间的联系建立的语义网络.领域分析强调以领域专家为主体,以建立领域体系结构标准为目的.

定义 3.1 (领域空间 Ω_{dom}). 领域空间是表示领域知识的特征以及特征之间的关系所组成的集合,表示领域空间的特征空间必须是合理的和完备的.

$$[DOMAIN]$$

$$\Omega_{dom} = \{x : \Omega \mid x.interpretation \in DOMAIN\}$$

这里, DOMAIN 是领域知识的集合. Ω_{dom} 用一个以 Feature 结构数据为元素构成的偏序集,表示一棵以领域特征为结点的语义树,其中,有且仅有一个根结点,元特征只能是叶子结点,称为原子域,非叶子结点称为复合域,复合域中父子结点之间呈现的 part-of 关系是对领域知识的划分,分区的并集要求覆盖父特征包含的领域知识.特定

领域软件体系结构(domain-specific software architecture,简称 DSSA)^[6,7]是 Ω_{dom} 中的一个分区,从现实世界中可以抽象出一个,也仅有一个完整的 Ω_{dom} ,它是我们定义构件语义的标准.

3.2 定义空间

描述构件语义的要素是特征值,它是对领域空间实例化的结果,由用户需求所决定.

定义 3.2 (定义空间 Ω_{def}). 定义空间 Ω_{def} 是一个特征实例 *FeatureValue* 的集合,每个集合元素或是多个特征值的逻辑组合,或是多个 Ω_{def} 的逻辑组合;构件静态语义是领域空间 Ω_{dom} 子集的实例的集合,构件动态语义是构件语义的变化规则.

$$\Omega_{def} = \{x : FeatureValue \mid \exists y : \Omega_{dom} \bullet x.id = y.id \wedge x.val \in y.def\}$$

Ω_{def} 是一棵描述构件语义结构的有穷特征树,每个树结点代表一个独立构件的语义集合,叶子结点表示该构件不再由其他构件组成,非叶子结点表示组合构件,其父子关系表示构件的功能包含关系.事实上, Ω_{dom} 与 Ω_{def} 的关系相当于 DLs 中 Tbox 与 Abox 的关系^[8].

3.3 语境空间

在 CBD 过程中,构件的创建环境和使用环境可能有所不同,语境是构件活动中指导构件语义变化的参照系,构件语义中要含有适应语境变化的参数.

定义 3.3 (语境空间 Ω_{con}). 语境是构件环境约束的总称,它决定构件语义内容的取舍;语境空间 Ω_{con} 是一个条件表达式的集合,条件表达式中的每个数据项或是一个特征值,或是仍然是一个语境空间 Ω_{con} .

$$\begin{aligned} & [CONFIG] \\ & condition : CONFIG \times \Omega_{dom} \rightarrow FeatureValue \\ & \Omega_{con} = \{(x, y) \mid (x, y) \in dom(condition)\} \end{aligned}$$

这里,CONFIG 是配置参数的集合;函数 *condition* 按照配置参数所决定的条件,将特征项实例化成特征值,该值表示了构件的语义.语境空间实质上表示了构件语义的变化部分.

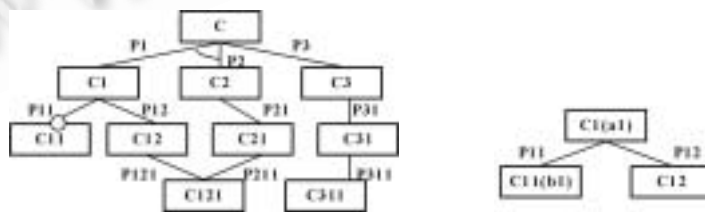
按照 Ω_{dom} 中特征之间的关系,可以把 Ω_{con} 形象地表示成一棵有限特征条件树,每个树结点代表一个条件表达式,它是对 Ω_{def} 中一个特征的约束,其中叶子结点表示的条件可以直接计算出来,称为简单条件,中间结点表示的条件建立在一组子条件的逻辑组合的基础上,称为复杂条件,计算复杂条件先要算出子条件的值.语境通常分为两类,定义语境是创建构件时的语义约束,操作语境是在基于构件的活动中对构件语义的限制.

3.4 实例

我们用与或图可视化地表示特征空间.图 2 中,形如 C1,P1 的数据项分别表示特征项和特征关系,形如 C1(a1)的数据项表示特征 C1 的实例 a1.

例子可以用特征描述逻辑 FDL 形式化表示如下,注意(b)中的 C12 需要在环境中实例化.

- (a) $\Sigma = \{C \sqsupseteq P1.C1 \sqcap P3.C3 \sqcup P2.C2 \sqcap P3.C3, C1 \sqsupseteq P11.C11 \sqcap P12.C12, C2 \sqsupseteq P21.C21, C3 \sqsupseteq P31.C31, C12 \sqsupseteq P121.C121, C21 \sqsupseteq P211.C211, C31 \sqsupseteq P311.C311\}$
- (b) $\Sigma = \{C1(a1), C11(b1), C12, P11(a1, b1), P12(a1, C12)\}$



(a) Domain space (a) 领域空间 (b) Definition space and context space (b) 定义空间和语境空间

Fig 2. Example of feature-oriented semantic representation for component 图 2 面向特征构件语义表示的例子

References:

- [1] Mili, H., Mili, F., Mili, A. Reusing software: issues and research directions. *IEEE Transactions on Software Engineering*, 1995,21(6):528~562.
- [2] Brown, A., Wallnau, K. The current state of CBSE. *IEEE Software*, 1998,15(5):37~46.
- [3] Brown, A. *Large-Scale Component-Based Development*. New Jersey: Prentice Hall, Inc., 2000.
- [4] Yang, Fu-qing, Mei, Hong, Li, Ke-qin. Software reuse and software component technology. *Acta Electronica Sinica*, 1999,27(2): 69~75 (in Chinese).
- [5] Object Management Group (OMG). *The Common Object Request Broker: Architecture and Specification V2.3.1999*. <http://www.omg.org>
- [6] Penix, J., Baraona, P., Alexander, P. Classification and retrieval of reusable components using semantic features. In: *Proceedings of the 10th Knowledge-Based Software Engineering Conference*. Boston, MA: IEEE Computer Society Press, 1995. 131~138.
- [7] Kang, K., Kim, S., Lee, J. *et al.* FORM: a feature-oriented reuse method with domain-specific reference architectures. *Annals of Software Engineering*, 1998,5:143~168.
- [8] Calvanese, D., Giacomo, G.D., Lenzerini, M., *et al.* Reasoning in expressive description logics. In: Robinson, A., Voronkov, A., eds. *Handbook of Automated Reasoning*. North-Holland: Elsevier Science Publishers, 2000. 316~327.
- [9] Jia, Yu, Gu, Yu-qing. Representing and reasoning on feature architecture: a description logic approach. *Workshop on Feature Interaction in Composed Systems, ECOOP 2001*. Budapest, 2001.
- [10] Spivey, M. *The Z Notation: a Reference Manual*. 2nd ed, Prentice Hall, 1992.
- [11] Tracz, W. Implementation working group summary. In: Baldo, J., ed. *Reuse in Practice Workshop Summary*. Alexandria, 1990. 10~19.
- [12] Edwards, S.H. Toward a model of reusable software subsystems. In: Philbrick, S., Stevens, M., eds. *Proceedings of the 15th Annual Workshop on Software Reuse*. Larry Latour, 1992.
- [13] Blom, M., Nordby, E.J. *Semantic integrity in component based development*. Project Report, Sweden: Mälardalen University, 2000.

附中文参考文献:

- [4] 杨芙清,梅宏,李克勤.软件复用与软件构件技术.电子学报,1999,27(2):69~75.

Domain Feature Space Based Semantic Representation of Component*

JIA Yu, GU Yu-qing

(Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

E-mail: jia_yu@263.net; guyq@sinosoftgroup.com

<http://www.iscas.ac.cn>

Abstract: In the paradigm of component-based software development, the component model is regarded as the foundation of almost all tasks such as classification, retrieval and adaptation. However, to date the semantic representation of component is a critical problem in the model. In this paper, a semantics intensive component model is proposed. Through a semantics defining method in principle of domain analysis on the feature space, the three views of component, that is domain space, define space and context space, are suggested to describe the semantic structure, and the feature description logics is used to express the semantic content. Especially, both the concept of semantizer and formal method of the feature space are one possible solution to component reuse automation and engineering.

Key words: software component; component semantics; feature space; software reuse; description logics; language Z

* Received October 9, 2000; accepted July 30, 2001