

一种可扩展的构件库数据模型^{*}

李 琰 邹艳珍 潘 颖 谢 冰 孙家骥

(北京大学信息科学技术学院软件研究所 北京 100871)

摘 要 在基于构件的软件开发(CBSD)中,构件库是描述、存储和检索软件构件的基础设施。随着软件复用技术的成熟,构件库中构件的种类日益增多,构件间关系日渐复杂。这需要构件库数据模型提供相应的支持。为了给构件库的实施者和管理者提供指导,有效地发挥构件库的作用,本文提出了一种可扩展的构件库数据模型 ECLDM。该模型以 UDDI 和 BIDM 为基础,对其结构进行了修改和扩充,并定义了扩展机制。利用本模型可以有效地描述和管理构件库中各种信息,并可以和 UDDI、BIDM 进行互操作。

关键词 构件库,可扩展性,构件分类,构件关系,互操作

An Extensible Component Library Data Model

LI Yan Zou Yan-Zhen PAN Ying XIE Bing SUN Jia-Su

(Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871)

Abstract In the Component-Based Software Development (CBSD), component library is an infrastructure which supports the description, storage and retrieval of software components. With the maturation of software reuse techniques, the categories of components stored in the library are growing rapidly, and the relations among components become more and more complex. Thus, the component library data model needs to support these changes. To guide developers and managers in the component library, we propose an extensible component library data model-ECLDM. Based on UDDI and BIDM, ECLDM modifies and extends their structure, and defines the policy for extension. The ECLDM can describe and manage different kinds of information in the component library effectively. Besides, it supports the interoperation with UDDI and BIDM.

Keywords Component library, Extension, Component classification, Component relation, Interoperation

1 引言

软件复用是一条提高软件开发效率和软件质量的切实有效的途径^[1]。在软件复用中,构件库提供了描述、存储和检索构件等功能,它在软件复用中起着关键的作用^[2-4]。

为了能够给构件库的实施者和管理者提供指导,有效地发挥构件库的作用,产业界和学术界提出了若干构件库模型。它们大致可以分为两类:一是面向构件库基本数据结构来进行描述的模型,如 W3C 提出的管理 Web Services 的 UDDI (Universal Discovery Description and Integration)^[5], ORES^[6], Aspect-based Component Repository^[7]等;二是为实现构件库之间互操作提出的模型,这方面主要的研究有 RIG (Reuse Library Interoperation Group) 提出并提交 IEEE 通过的用于构件库互操作的数据模型 BIDM (Basic Interoperability Data Model)^[8]及其扩展模型^[9,10],以及 RIG 推荐的 UDM (Uniform Data Model)^[11]等。

要减小复用成本,使复用者能够正确地理解构件,更好地复用软件构件,就需要对构件进行描述和分类^[12]。W. Frakes 的研究表明,构件库模型应该能够支持多种分类法^[13]。此外,构件之间的关系对于构件库而言也是很重要的资源^[14],构件库的模型也应该能够明确地表示构件之间的关系。

不仅如此,为了能够在各种不同的应用场景下更好地管理、利用构件库,构件库的实施者和管理人员往往需要根据情况对模型进行扩展。

现有的模型缺乏对分类、关系信息相应的描述机制,同时缺乏扩展机制的定义。例如:作为 Web Services 注册和发现机制的 UDDI,既不能很好地对其它类型的构件进行描述和管理,也没有表示构件关系的结构,而且其面向电子商务应用的性质使它很难满足作为构件库数据模型的需要。BIDM 仅包含了构件库之间为进行互操作所必需定义的最小信息集合。因此它的结构过于精简,对于构件分类、构件关系的描述缺乏必要的支持机制,难以充分地满足构件库管理的需要。另外,UDDI 和 BIDM 都没有定义如何对模型进行扩展。

随着软件复用技术的成熟,构件的种类日益增多,构件间关系日渐复杂,由于缺乏分类、关系描述信息而产生的问题更加凸现。缺乏明确定义的扩展机制,就无法对模型的扩展提供指导和约束。这样,用户进行的扩展容易带有随意性,而造成模型的混乱,进而给用户的理解和使用设置障碍。

本文提出一种可扩展的构件库数据模型 ECLDM (Extensible Component Library Data Model),ECLDM 在现有的模型的基础上增加了对构件分类等信息的描述,并且定义了如何对模型进行扩展的机制。ECLDM 既囊括了构件库中必

^{*} 国家高技术研究发展 863 计划(No. 2004AA112070, No. 2004AA115080)、自然科学基金项目(No. 60473059)。李 琰 博士生,主要研究领域为软件复用、软件工程;邹艳珍 博士生,主要研究领域为软件构件技术,软件工程;潘 颖 博士生,主要研究领域为软件复用、软件工程;谢 冰 副教授,主要研究领域为软件工程;孙家骥 教授、博士生导师,主要研究领域为软件逆向工程、软件再工程。

需的基本元素,又能够根据需要对模型进行规范的扩展,这为构件库的实和管理提供了明确的指导。基于 ECLDM 及其扩展机制实现的构件库管理系统不仅能够很好地支持构件的分类和检索,还能有效地描述构件间的关系,并且能够支持构件库间的互操作。

2 可扩展构件库数据模型(ECLDM)

2.1 模型概述

基于现有的各种构件库数据模型,我们提出了一种新的可扩展的构件库数据模型 ECLDM。模型由 19 类元素组成,分为 3 个部分,即基本描述部分、分类法描述部分和关系描述部分(如图 1)。

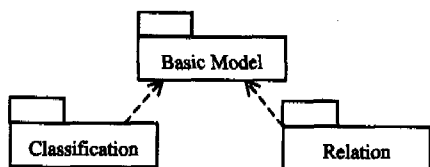


图 1 模型总体结构

2.1.1 基本描述部分

基本描述部分包含了模型中的最基本的构成元素。构件

库对象(Component Library Object,表示为 CLObjct)处于整个层次结构的最上层(如图 2),它表示的是构件库中管理的对象,因此是所有元素的父元素。CLObjct 包含了模型中所有元素所共有的属性:名称(Name)和用来唯一标识元素的 ID。

综合以往的各种模型,构件库中必需的对象包括:构件库(Library)、构件库中的用户(User)、构件库中的软件资产(Asset)及资产的构成成分(Ingredient)。另外,构件库中的各种元素往往满足某些约束或属于某种分类框架,我们可以利用这类信息更好地描述各种元素。为此,我们定义了规约(Specification),来表示约束和描述机制。而引用规约(Referenced Specifications)元素则包含了对 Specification 的各种实例化信息。

(1) Library 元素 记录构件库自身的信息,来帮助用户对构件库有概要性的了解。在 Library 中,包括了对外提供的访问构件库的地址(ElectronicAddress)。为了对构件的入库和管理进行指导,Library 元素通过一些文本类型的属性提供了介绍性的信息,包括关于构件的验证方法(CertificationMethods)、构件库的度量原则(LibraryMetrics)和库的分类机制(LibraryClassificationMechanism)。构件库的限制(LibraryRestriction)属性则介绍了使用构件库应该遵守的原则。

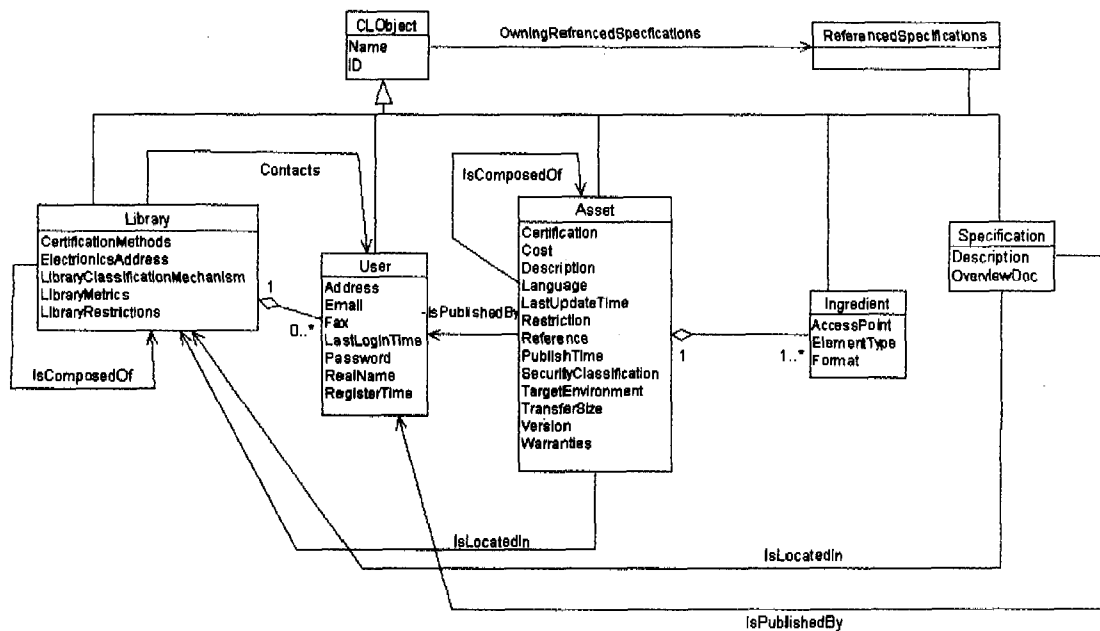


图 2 ECLDM 基本描述部分

构件库之间可以存在层次(IsComposedOf)关系,即一个构件库可能是另外一个构件库的子库。构件库同时也是资产和规范的存储场所,即 Asset 元素和 Specification 元素与 Library 元素存在 IsLocatedIn 关系。

(2) User 元素 构件库的用户使用 User 元素进行描述,它记录了用户的各种联系方式(如 Address、Fax 等)和登录相关的信息(LastLoginTime、Password 等)。

Library 元素和 User 元素之间的 Contacts 关系表示该用户是构件库的联系人。User 元素与 Asset 元素和 Specification 元素间的 IsPublishedBy 关系表明用户是构件库中软件资产和规约的发布者。

(3) Asset 元素 表示的是构件库中的构件和其他有用的各类软件资产。它包含了软件资产的版本相关的信息

(Version、LastUpdateTime 等)、实现工具及对外界依赖相关的信息(Language、TargetEnvironment 等)以及与使用相关的信息(Cost、TransferSize 等)。

一个 Asset 可能是其它 Asset 的子部分;在这种情况下,两个 Asset 元素构成了 IsComposedOf 关系。而一个基本的 Asset 则是由若干个成分(Ingredient)组成的。

(4) Ingredient 元素 描述了资产成分的类型(ElementType)和格式(Format),可以方便构件库的存储和用户的使用。另外,对于 EJB、WebService 等分布式构件,构件库中并不存放构件的实体,而是提供构件的访问点(AccessPoint)。

(5) Specification 元素 描述构件库中的规范和约束等信息,包含了描述信息(Description)以及相关远端描述和使用说明(OverviewDoc)。

(6) ReferencedSpecifications 元素 包含了对 Specification 的各种实例化信息的引用。它是 CLObject 的组成部分, 用来具体描述元素。

2.1.2 分类法描述部分

构件的有效分类对于构件的检索和获取起着非常重要的作用, 是软件复用成功的关键^[12]。W. Frakes 的研究表明, 构件库模型应该能够支持多种分类法^[13]。在对构件进行分类的研究中, 有 4 种常见的分类法:

- 关键词分类法: 每个构件用一组与之相关的关键词来编目。
- 属性-值分类法: 构件是根据一组固定的属性和对应的值来描述的。
- 枚举分类法: 通常将一个被关注的领域划分为不相交

的子领域, 依次构成层次结构。

• 剖面分类法^[15]: 由一组描述构件本质特征的剖面所组成, 每个剖面从不同的视角对构件库中的构件进行精确的分类。每个剖面具有一组术语(关键词), 术语之间有层次关系而形成结构化的术语空间。

以往的模型中, 对于构件的分类机制的支持机制或者不够直接明了, 如 UDDI 中并没有明确地用于描述分类法的结构; 或者缺乏支持, 如 BIDM 不能够为构件提供分类信息的描述。这都给构件的检索等操作带来了困难。同时, 如果模型中分类的方法不够全面, 那么难以满足不同场景的需要。因此, 我们在模型中对规范 (Specification) 进行扩展, 得到了分类法元素 (Classification), 提供对于分类机制的描述 (如图 3)。

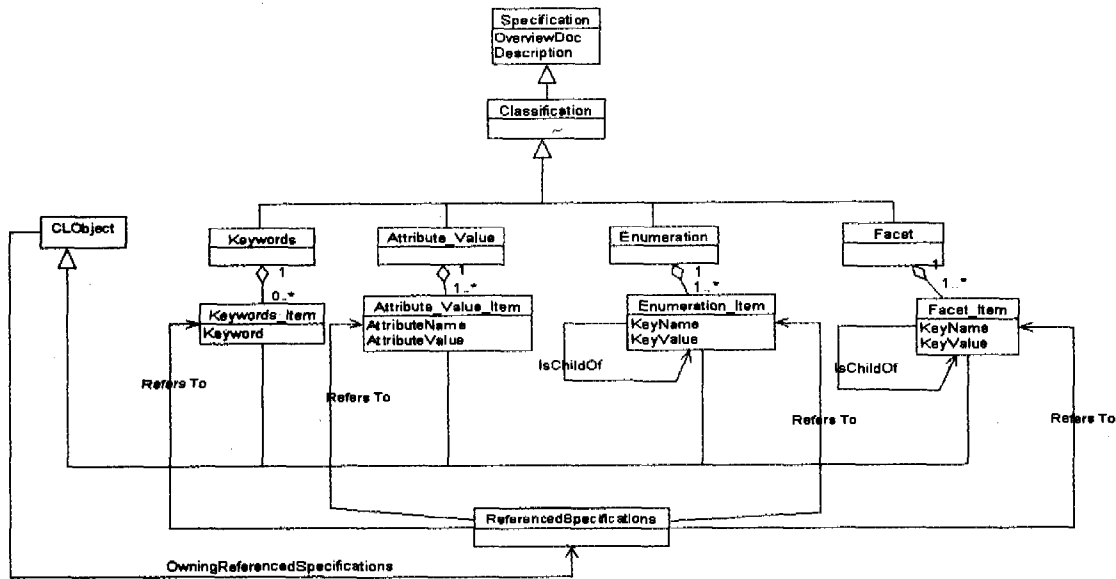


图 3 ECLDM 分类法描述部分

我们将分类法元素进行扩展, 将 4 种分类法同时引入。同时, 我们将 4 种分类法使用的术语/值或关键词也引入到模型中, 并和相应的分类法建立整体-部分关系。这些术语或关键词属于构件库需要管理的对象, 但是不能视为软件资产或资产的成分, 因此它们直接继承 CLObject。

(1) 关键词分类法。Keywords 元素包含若干个相应的分类术语项 (Keywords_Item)。Keywords_Item 元素中包含的唯一属性 (Keyword) 就是用来进行描述构件的关键词。关键词分类法可以支持受控词表和非受控词表^[13]两种形式。

(2) 属性/值分类法。Attribute_Value 元素对应的术语元素是 Attribute_Value_Item, 它的属性是属性名 (Attribute_Name) 和属性值 (Attribute_Value)。

(3) 枚举分类法。Enumeration 元素的术语元素是 Enumeration_Item。Enumeration_Item 之间可以依据描述的领域构成层次 (IsChildOf) 关系, 它包含的属性是键名 (KeyName) 和键值 (KeyValue)。

(4) 剖面分类法。Facet 元素由一系列 Facet_Item 术语元素构成。同样, Facet_Item 也可以依据术语空间中的关系构成层次 (IsChildOf) 关系, 它包含的属性是键名 (KeyName) 和键值 (KeyValue)。

用户在使用分类机制时, 可以发布某种类型分类法的实例, 也可以在已有的分类法实例下编辑术语或属性。在描述元素的时候, 可以将分类法中的术语或属性添加到元素的引

用规约 (ReferencedSpecifications) 中, 来表示元素属于某个类别。例如, 如果需要按照构件的使用环境进行分类, 可以发布一个 “Application Environment” 的剖面分类法, 并在该分类法发布操作系统环境 (OS)、数据库环境 (DataBase) 和网络环境 (Network) 3 个术语。还可以进一步在操作系统环境 (OS) 下发布 Unix、Dos、Windows 等子术语。在描述某个仅适用于 Windows 操作系统的构件时, 可以将 Windows 术语项加入到该构件的引用规约中。

2.1.3 关系描述部分

构件关系不仅可以用于有效地组织构件, 也是软件开发中可以使用有效信息, 更是体系结构中的基本元素。因此, 构件库系统中也应该管理与维护构件关系, 以降低软件开发的成本^[14]。

传统的构件库模型中往往缺乏对关系表示的支持, 或者是由构件库提供一些预定义的关系类型来对构件关系进行描述。用户仅能依赖预定义的关系类型来建立构件之间的关系, 这样很难为构件关系的描述提供灵活的支持。

我们将构件之间的一个具体的关系称为关系实例 (RInstance)。不同的关系实例可能具有相同的属性和表示。因此, 可以把它们抽象为同一类型的关系, 并用关系类型 (RType) 来进行描述。同样, 不同的关系类型可能隶属于相同的应用范畴, 我们利用关系空间 (RSpace) 来界定不同的应用范围。

在构件库中,用户可以针对不同的应用领域建立独立的关系空间,并在该关系空间下建立起一系列适用于该领域的关系类型。最后,引用这些关系类型来定义关系实例,来描述构件关系。这样,就可以清晰、灵活地描述构件间的关系(如图4)。

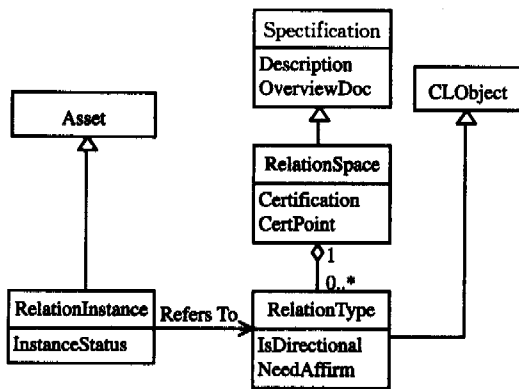


图4 ECLDM 关系部分

在RSpace元素中,约束验证信息(Certification)描述的是引用该关系空间的关系类型定义具体的关系实例时,是否需要进行约束验证。如果需要验证,还需要提供约束验证服务的访问地址(CertPoint),这可以为关系空间及其关系类型提供第三方的管理机制。RSpace元素下的RType元素是对一组具有相同属性的关系的抽象描述,它提供了是否为有向关系(IsDirectional)和引用该类型定义的关系实例是否需要被关连实体的确认(NeedAffirm)的信息。由于RType元素比较抽象,不具有可复用性,不能被视为软件资产或资产的成分,因此我们将CObject元素作为RType元素的父元素。这表示关系类型是构件库要管理的对象。关系实例(RInstance)是特定关系空间中特定关系类型在实体集合上的实现,即实体之间关系的实例化。关系实例连接着不同的构件,并且可以和它所连接的构件结合在一起,组成更大粒度的可复用构件。所以,关系实例也是一种可复用的软件资产,RInstance元素也应扩展资产(Asset)元素。

使用上述的关系表示模型不仅能够清楚描述构件的关系,还能够为广义的构件库中的实体之间的关系提供统一的描述。基于本模型来管理关系的构件库系统,可以支持用户自定义关系类型,并且支持关系的定义和关系实例的使用相分离的管理模式,使关系管理具有更大的灵活性,从而具有较高的可扩展性。

2.2 模型扩展机制

随着构件库技术的不断成熟与发展,用户对构件库的需要日渐复杂而且趋于多样化。因此,用户必然需要对基本的构件库模型进行扩展来满足各自的需求。传统的构件库模型并没有关于模型扩展机制的定义,如UDDI和BIDM并没有约束如何进行扩展。缺乏了必要的约束和指导,对构件库模型的扩展会出现随意、混乱的情况。这样,既给构件库用户的理解和使用造成了困难,也为将来构件库的互操作制造不便。

针对上述问题,在定义构件库模型的基础上,我们进一步定义了模型的扩展机制,对模型的扩展分为两类,即:对模型中已有元素的扩展和向模型中扩展新的元素。

(1)对已有元素进行扩展 用户在使用本模型中元素的时候,可能需要对该元素增加描述信息才能满足需要。这时,用户可以有两种选择来满足需要:①派生新的元素;②在已有

元素中添加新的属性。

派生新的元素,容易激增模型中元素的个数和元素层次,造成模型复杂、难以理解。而在已有元素中添加新的属性,会使得该元素过于庞大,同时难以区分元素中原有的属性和扩展的属性。

在本模型中,扩展的方法是由用户发布相应的属性/值分类法,然后将该分类法下的属性/值项加入到需要扩展元素的引用规约(ReferencedSpecifications)中来对已有元素进行扩展。这样既满足了描述元素的需要,又保持了模型的清晰性。

例如:如果想要将构件库中用户分级,即要在User元素中加入等级的信息,那么可以在属性值分类法下面发布一种关于用户等级(UserLevel)的分类法。该分类法包含一个术语项(UserLevel-Item),该术语项中属性名为等级(Level),属性值为具体的等级值(Value)。然后,将该术语项加入到User元素的引用规约中,便完成了对用户等级的描述。

(2)扩展新的元素 在使用构件库模型实现构件库管理系统时,用户往往需要添加新的元素。但是随意地增加新的元素会增加模型的复杂度,带来混乱的现象。

为了规范如何扩展新的元素,ECLDM规定必须通过继承已有的元素得到新的元素。泛化元素中的属性和泛化元素参与的关联都将继承到新扩展的元素中,而且只能进行单继承。这样,扩展新的元素不会破坏模型的基本架构。而如果模型中所有的元素都允许被继承,也会造成模型的混乱。因此,扩展新的元素只能在指定的元素上进行。如果某元素的实例具有多种不同形态,而且有必要区分不同形态的实例,那么该元素可以被继承。在ECLDM中,可以在其上进行扩展的元素包括:CObject、Asset、Ingredient、Specification、Classification和RSpace。除此之外,扩展还有具体的约束。

对Specification、Classification和RSpace的继承,将会得到新的形式约束条件和规范信息,我们把新扩展的元素表示为A。对于A及其子类可能会包含一些组成的元素,我们将这些元素表示为B。除此之外,还需要其它的元素(表示为C)来引用A元素或B元素,以实际描述构件库中的对象(如图5)。

约束1:如果C元素是ReferencedSpecifications,那么首先要区分B元素实例是否具有可复用性。如果具有可复用性,那么应将B元素视为一种软件资产,因此B应该继承Asset元素。如果B元素不具有可复用性,而且只能用来描述Asset元素,那么它成为Asset的一部分而继承Ingredient。其他情况下,B元素不具有可复用性,而且可以用来描述多种元素,那么它仅是构件库需要管理的对象,应该继承CObject,而且必须被ReferencedSpecifications元素所引用。

约束2:如果C元素不是ReferencedSpecifications,而是新扩展的元素,那么B元素是用来生成描述信息的模板或者标准。这样,B既不能被视为软件资产或资产成分,也不能视为是一种规约,因而只能继承CObject。对于C,它的约束要求和约束1中B元素情况一致。

约束3:对于继承Asset的元素,其组成成分只能是Ingredient。

约束4:对于继承Ingredient的元素,不可以再包含组成成分。

约束5:继承CObject的元素必须是构件库需要管理的对象。

例如:为了描述构件的质量,构件库的实施者和管理者在

Specification 元素下派生质量模型(QualityModel)元素(如图6),并在 QualityModel 元素中加入质量因素(QualityFactor)和质量级别(QualityLevel)等元素来作为评定质量的标准。这两个元素是用来生成质量描述信息的模板,因此继承 CLObject 元素。为了记录、评定 Asset 元素质量的结果,可以利用对 Ingredient 元素进行扩展得到质量信息(QualityInformation)元素来完成,这样就可以有效地描述构件的质量。

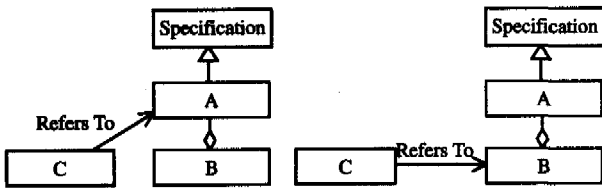


图5 扩展说明

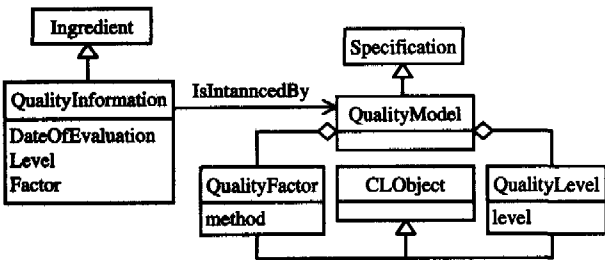


图6 质量扩展

通过定义模型的可扩展性,既满足了用户扩展构件库模型的需要,又保持了模型的基本架构。这样,不但控制了模型的复杂度,也为构件库的互操作打下了基础。

2.3 互操作性分析

伴随着软件复用技术的发展,Internet 上各种构件库也日益增多,例如:SourceForge (http://www.sourceforge.net), Component-Source (http://www.componentsource.com), Download.com(http://www.download.com)等。由于各个不同的构件库可以根据自己的用户群体的偏好来进行组织和管理,所以这些异构构件库的存在是合理的,也是必要的^[6]。如果构件库之间存在良好的互操作,那么构件库的用户可以通过单一的用户界面/接口访问多个构件库中的制品和服务。这样,就更能进一步推动软件复用的进行,从而达到降低软件开发的成本、提高软件产品质量的目的。而网络技术和软件技术的发展则使构件库之间进行互操作成为必然^[6]。

因此,构件库的模型必须能够支持互操作。我们将 ECLDM 与 IEEE 构件库互操作标准 BIDM 进行比较(如表1),从中我们可以发现 BIDM 中的各类元素在 ECLDM 中都有相应的元素。因此,ECLDM 显然可以和 BIDM 进行互操作。

表1 BIDM 与本模型的元素对应关系表

IEEE BIDM	ECLDM
RIGObject	CLObject
Asset	Asset
Element	Ingredient
Library	Library
Organization	User

除此之外,ECLDM 和 UDDI 模型中的核心数据结构也

存在很好的对应关系(如图7)。可以看出,两者之间大部分元素存在着直接的映射关系(为表示 UDDI 中的 publisher-Assertion,可以在 ECLDM 中发布一个针对 UDDI 的关系空间,然后将 UDDI 预置的关系类型发布在该关系空间下)。UDDI 中没有类似本模型中 Library 的元素,但是却可以通过 UDDI 中的 operator 属性来记录 Library 中的信息。由此可见,本模型和 UDDI 也是可以互操作的。

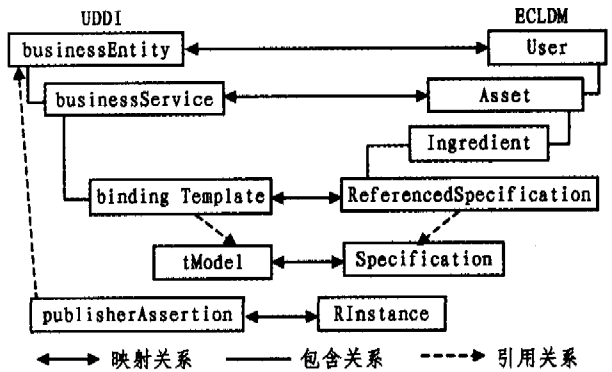


图7 UDDI 和本模型元素对应关系图

3 应用实例

下面我们以北京大学软件研究所开发的构件库管理系统 JBCLMS 为例,来说明如何使用本模型对构件库进行管理。

3.1 系统概述

北京大学在“基于 Internet,以构件库为核心的软件开发平台”项目中,以本模型为核心来对构件库进行建模,开发了公共构件库系统 JBCLMS。该系统使用 EJB 接口规范和 J2EE 技术来实现业务逻辑,对外提供了基于 Struts 技术的 Web 形式的用户访问界面和 Web Services 形式的 API 接口,实现了对软件构件进行描述、分类、存储和检索等功能。该系统已在北京、上海、沈阳、青岛等地的软件园正式发布和使用。

3.2 模型应用

构件库分类法子系统参照 ECLDM 中分类法描述部分来实现,它能够满足构件库中用户不同的分类需要,使得构件的管理和分类更加清晰。同时,丰富的构件分类手段也为构件检索提供了有力支持。当用户去了解构件进行复用时,构件所包含的分类信息也能够为他们提供帮助。

基于 ECLDM 中关系描述部分实现的关系管理子系统,将关系类型和关系实例的管理进行了分离,很好地实现了关系类型管理者和关系实例使用者的角色的区别。同时,通过关系进行构件查询,可以帮助用户在构件库中周游构件,从而浏览、比较、评价与某个构件相关联的构件。

在实现 JBCLMS 过程中,我们以本模型为核心,并利用模型的扩展机制完成了需要的扩展。

(1)对已有元素进行的扩展 为了使管理者更好地管理、调整构件库,使构件库的用户对构件有更加清晰的了解和认识,JBCLMS 在 Asset 元素的引用规约中添加了关于下载次数和运行记录等属性的引用。为了保证构件库的访问安全性,JBCLMS 还将用户(User)进行了分级,并且给出用户相应的权限。同时,JBCLMS 也在 Asset 元素中添加了访问权限的属性。两者结合,使系统具有灵活的权限分配和安全的访问控制方式。除此之外,JBCLMS 还对 Library,Ingredient 等元素进行了扩展。

(下转封四)

(上接第 286 页)

(2)向模型中扩展了新的元素 在现实世界中,存在着大量的标识事物的信息,如机构的银行账号、地理位置等等。这类信息具有很好的可解析性,即它们的含义是明确的,而且具有很好的可扩展性。借助于这类信息,可以更直观地描述和查询发布的元素。因此,JBCLMS 继承了规约(Specification)元素,引入了标识符(Identifier)元素。使用标识符可以更好地利用现实世界的信息来检索构件库中的元素。除此之外,为了更好地理解和评估构件,JBCLMS 还扩展了反馈(FeedBack)等新的元素。

ECLDM 中定义的扩展机制满足了 JBCLMS 构造过程中对模型的各种扩展需要,同时保证了模型的稳定性和易理解性。

在互操作方面,JBCLMS 实现了与 UDDI 兼容的 Web Service 访问接口,完成了与 UDDI 站点的互操作。同时,基于文[18]中的方法,我们以 ECLDM 为核心数据结构,实现了构件库互操作体系,从而为不同构件库的用户提供统一的查询接口,使他们更容易地获取分布在不同构件库中的构件。目前已经完成了系统开发工作。

总结 构件库是支持基于构件软件开发(CBSD)的重要基础设施。随着软件复用和构件技术的发展,构件的种类日益增多,构件关系也越来越复杂。构件库数据模型应该能够适应随着这些变化而产生的需求。本文提出了一种新的构件库数据模型 ECLDM,它既囊括了构件库中必需的基本元素,又规定了模型的扩展机制。该模型提供了明确的分类机制,并支持 4 种流行的构件分类方法,丰富了构件的分类和描述手段,为构件检索提供了支持。同时,模型包括了管理构件关系的部分,并采用关系类型定义和关系实例使用相分离的模式,为关系管理带来灵活性。该模型可以与 IEEE 的 BIDM 规范以及管理 Web Service 的 UDDI 规范进行互操作。北京大学基于本模型及其扩展机制研发了公共构件库管理系统 JBCLMS,从而进一步验证了模型的可扩展性和可用性。

参 考 文 献

- 1 杨美清,梅宏,李克勤. 软件复用和软件构件技术. 电子学报, 1999, 27(2):68~75
- 2 杨燕燕,梅宏,陈海文,等. 数据仓库技术和可复用构件系统. 计算机科学, 1999, 26(5):56~60
- 3 Guo J, Lu Qi. A survey of software reuse repositories. In: Proceedings of IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2000), Scotland, UK, 2000, 92~100
- 4 潘颖,赵俊峰,谢冰. 构件库技术的研究与发展. 计算机科学, 2003, 5:90~93, 156
- 5 Bellwood T, Clement L, Ehnebuske D, et al. UDDI version 3.0 published specification. [Technical report]. UDDI.org, 2002
- 6 Yen I-Ling, Khan L, Prabhakaran B, et al. An on-line repository for embedded software. In: Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence, USA, 2001. 314~321
- 7 Grundy J C. Component storage and retrieval using aspects. In: Proceeding of the 2000 Australian Computer Science Conference, Australia, 2000, 95~103
- 8 IEEE. IEEE standard for information technology - software reuse - data model for reusable library interoperability: Basic interoperability data model (BIDM). IEEE std 1420. 1, 1995
- 9 IEEE. Supplement to IEEE standard for information technology software reuse data model for reuse library interoperability: Asset certification framework. IEEE std 1420. 1a, 1996
- 10 IEEE. IEEE trial-use supplement to IEEE standard for information technology software reuse data model for reuse library interoperability: Intellectual property rights framework. IEEE std 1420. 1b, 1999
- 11 Reuse Library Interoperation Group, RIG Uniform data model for reuse libraries (UDM). [Technical Report]. RPS-002, 1994
- 12 Mili H, Mili F, Mili A. Reusing software: issues and research directions. IEEE Transaction on Software Engineering, 1995, 21(6):528~562
- 13 Frakes W, Pole T. An empirical study of representation methods for reusable software. IEEE Transaction on Software Engineering, 1994, 20:617~630
- 14 薛大宇,潘颖,谢冰,等. 一种支持构件关系管理的描述模型. 见:2004 全国软件和应用学术会议论文集. 北京:机械工业出版社, 2004. 50~55
- 15 Prieto-Diaz R. Implementing faceted classification. Communication of ACM, 1991, 34(5):88~97
- 16 Browne S, Moore J. Reuse library interoperability and the world wide Web. In: Proceedings of the 19th International Conference on Software Engineering, USA, 1997. 684~691
- 17 Pan Ying, Wang Lei, Zhang Lu, et al. Relevancy Based Semantic Interoperation of Reuse Repositories. In: Proceedings of the 12th ACM SIGSOFT Symposium on Foundations of Software Engineering (FSE), USA, 2004. 211~220

计算机科学

(1974 年 1 月创刊)

第 33 卷第 5 期 (月刊)

2006 年 5 月 25 日出版

国际标准连续出版物号 ISSN 1002-137X
国内统一连续出版物号 CN50-1075/TP

定价: 30.00 元 国外定价: 5 美元

邮发代号: 78-68

发行范围: 国内外公开

主管单位: 国家科学技术部

主办单位: 国家科技部西南信息中心

编辑出版: 《计算机科学》杂志社

重庆市渝中区胜利路 132 号 邮政编码: 400013

电话: (023) 63500828 E-mail: jsjcx@swic.ac.cn

网址: www.jsjcx.com

社 长: 牟炳林

总 编: 彭 丹

主 编: 朱宗元

主编助理: 徐书令

印刷者: 重庆科情印务有限公司

总发行处: 重庆市邮政局

订购处: 全国各地邮政局

国外总发行: 中国国际图书贸易总公司 (北京 399 信箱)

国外代号: 6210-MO