一种面向服务的业务过程建模语言

邢少敏 周伯生 陈天影

(北京航空航天大学 计算机学院, 北京 100191)

摘 要: 通过对现有业务过程建模语言的分析指出了其对面向服务支持的不足,以可视化过程建模语言 VPML(Visual Process Modeling Language)为基础,提出了一种面向服务的可视化过程建模语言 VPML-S,并提出了该语言的概念设计框架,简述了其通用的建模元素;使用 UML类图设计了该语言的元模型,在此基础上建立了语言的形式化模型;分析和评估了 VPML-S中的服务元素所提供的将活动映射为 Web服务的面向服务功能,嵌入子过程元素所提供的多个业务过程之间的复用机制,管道元素所提供的多个业务过程的协作机制以及 VPML-S和其它语言所建立的业务过程模型的互操作支持机制;评述了 VPML-S的可视化图符替换、元模型扩展机制;最后通过一个网上购物流程实例说明 VPML-S为面向服务的环境下的业务过程复用、协作和互操作提供了有效的支撑.

关 键 词: 可视化过程建模语言; 过程工程; 形式化语言; 面向服务

中图分类号: TP 311. 5

文献标识码: A **文 章 编 号**: 1001-5965(2008) 11-1343-06

Service-oriented business process modeling language

Xing Shaomin Zhou Bosheng Chen Tianying

(School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract Through analysis of the existing business process modeling languages, their lack of service-oriented support was indicated. A new service-oriented visual business process modeling language VPML-S was proposed based on the existing visual process modeling language (VPML). The conceptual design framework for VPML-S was proposed and its general modeling elements were outlined. The meta-model for VPML-S was designed by using UML class diagram, and its formal model was established on this basis. The service-oriented function provided by service element, the reuse mechanism between business processes provided by nested sub process element, the collaboration mechanism between business processes provided by pipeline element and the interoperability support mechanism between VPML-S business processes and business processes established by other languages were analyzed and evaluated. Visual icons replacement mechanism and meta-model expansion mechanism provided by VPML-S were discussed. Finally, an online shopping flow example was presented to demonstrate the efficiency of the VPML-S for business process reuse, collaboration and interoperability in service-oriented environment.

Key words visual process modeling language; process engineering; formal languages; service-oriented

近年来,随着业务过程建模 BPM (Business Process Model)、面向服务的体系结构 SOA (Service-Oriented Architecture)等技术日益成熟,业务过程再一次成为人们关注的重点.现代 BPM的一个重要特点就是能够和 SOA技术相结合,将各种 Web服务编排成可执行的业务过程[1],执行逻辑可以用某种业务过程执行语言如 BPEL[2]来表达,并直接由业务过程执行引擎执行.但是 BPEL所表达的业务过程是面向软件设计人员的,业务人员难以理解,这就产生了对业务级别的可视化

收稿日期: 2007-12-05

基金项目: 北京市科技计划课题资助项目 (y0105001040521)

作者简介: 邢少敏 (1980-),男,陕西宝鸡人,博士生, xingshaomin@cse.buaa.edu.cn.

过程建模语言的需求,经过长期的分析和实践,本文认为现代 BPM 对可视化业务过程建模语言有如下要求: ①以业务人员能够理解的简明方式在业务级别来表达业务过程,不要涉及到技术细节问题; ②有严格的定义,包括形式化模型、无二义性的语法说明和清晰的执行语义,保证语言的正确性; ③有过程复用机制,保证业务过程模型的高度复用,避免重复的工作; ④能够提供组织内外多个业务过程的协作机制和支持使用不同底层技术架构的异构业务过程的互操作.

针对这些需求,在原有可视化过程建模语言 VPML[3]的基础上,扩展了面向服务的功能,形成了面向服务的过程建模语言 VPML-S

1 业务过程建模语言的相关研究

近年来,随着 Web服务技术、XML技术的发展,很多新的面向服务的业务过程建模语言相继提出. 比如: BPEL, XPDL[4], WS-CDL[5]和 OWL-S[6]. BPEL所表达的业务过程是面向软件设计人员的,业务人员难以理解. XPDL主要致力于分布式工作流,近年来的定位主要是作为一种通用业务过程存储格式. WS-CDL也是致力于 Web服务的编排,但是它侧重于系统设计和实现,而不是业务需求的分析和获取. OWL-S是一种基于语义的 Web服务过程建模语言,但是它主要面向研究人员,对于一般业务人员来讲,是不适用的.

以上这些语言都不能满足业务人员进行业务过程建模的需求,北航软件所开发的可视化过程建模语言 VPML是针对企事业过程建模的图形化语言. 它用可视化的过程图及其相应的正文规格说明来描述业务,其过程图用来描述过程的结构,正文规格说明用来描述该过程图中诸元素的属性. VPML过程图中的基本元素分成 5类: 活动、产品、资源、其它以及它们之间的关联连接. 用 VPML建造的过程模型可以模拟执行. 多年来的实践证明, VPML不仅能描述一个完备的过程模型,而且是一个功能丰富、直观易学、灵活适用的过程建模语言. 以 VPML为基础的企事业过程建模支持系统 EPMS[7]在国内外有长期广泛的应用, VPML语言的价值得到了广泛的认可. 但是随着近年来面向服务的体系结构的迅速发展, VPML逐渐暴露出很多问题: 比如使用 VPML建立的过程模型本身是一个封闭的过程,不支持开放环境下多个业务过程协作、互操作和复用; 而且用 VPML建立的业务过程始终无法执行; 再加上用户定制图符困难和没有提供元模型扩展机制,使得该语言越来越不能适应实际需求.

2 业务过程建模语言 VPML-S

2.1 VPML-S语言概述和框架

面向服务的过程建模语言 VPML-S继承了原有 VPML语言的过程建模基本思想,并根据现代 BPM 的特点和实际需要作了全新的设计,扩展了面向服务的特性,有效的支持了过程复用,协作和互操作,新增了灵活的元模型扩展机制和形式化模型,使得 VPML-S满足了当前业务过程建模的需求. VPML-S的语言的总体设计见图 1. 图 1中, VPML-S的语言总体框架由业务层、执行层和服务层组成. 其中执行层和服务层不属于 VPML-S语言的组成部分,执行层由 BPEL充当,服务层则直接利用网络上的开放 Web服务资源. 业务层才是 VPML-S语言的组成部分,位于业务层底层的是语言核心,由元模型 (规定 VPML-S的语法)、形式化模型 (提供严密的形式化理论模型,用于模型的形式化分析和验证)、模拟规则 (模型进行执行前的模拟验证)和执行语义 (执行规则). 位于业务层上层的是用户定义层,用户可以根据自己的需要,灵活的扩展 VPML-S,扩展包括图符自定义、扩展建模元素、扩展建模元素语义属性等方面. 使用 VPML-S建立的过程模型最终将映射为业界流行的 BPEL执行语言,并且可以放在任意兼容 BPEL的执行引擎中执行.

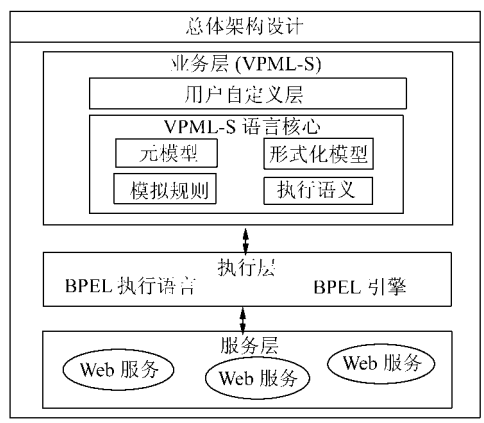


图 1 VPML-S的语言总体框架

本文主要介绍 VPML-S的主要建模元素,元模型和用户扩展机制和面向服务的特性,过程复用机制,协作机制和互操作支持,其它方面将在后续论文中继续讨论.

2.2 VPML-S通用建模元素

VPML-S过程图中的基本元素分成 5类,分别为活动、产品、资源、连接和事件. 活动表示过程中要执行的工作,是 VPML-S语言的核心. 其他

VPML-S元素主要用来描述活动的细节、活动的输入和输出产品以及各个活动之间的协调关系. 产品用于表示在过程活动中被使用、创建、修改和传送的数据. 资源用于模仿现实世界中的资源, 这些资源是执行活动所必需的, 用来描述人员和某些非人员资源 (如计算机、机器和程序). 连接分为 2类: ①流连接; ②逻辑连接符. 流连接不仅用在建立 VPML-S元素间的联系, 而且还用在活动间传送产品信息, 帮助协调活动的调度. 逻辑连接符使用逻辑运算符 (与、或)组合流连接, 用来表达过程中分支结构的路径选择. 事件在过程开始、结束或执行时产生, 反映了过程执行时的状态或发生的事情. 比如过程的开始和结束. 不同于原有 VPML的是, VPML-S通过引入服务资源来支持活动的 Web服务实现, 引入嵌入子过程, 解决了过程建模语言的过程复用问题, 引入管道解决了过程协作问题.

2 3　元 模 型

VPML-S的元模型如图 2所示, 它规定了 VPML-S各个元素之间的语法连接规则. 图 2给出了 VPML-S的简化元模型, 在这个元模型中, 为了清晰和突出重点省去了和本文主题无关的一些对象, 比如连接里面的数据流, 关联流, 还有逻辑连接符等等.

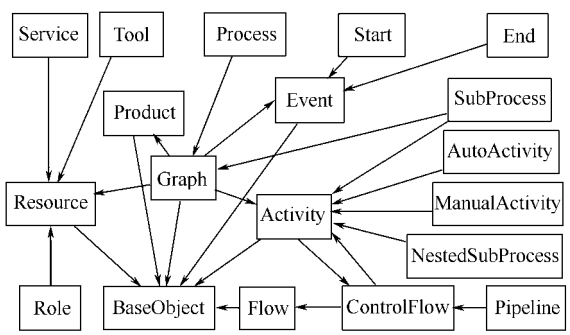


图 2　VPML-S元模型

过程开始和结束分别使用开始和结束事件, 活动之间可以直接用控制流连接表示活动执行的顺序关系. 资源中新增服务 (service)元素, 活动中新增嵌入子过程 (nested sub process)元素, 管道 (pipeline)是控制流的一种, 用于支持跨过程的协作. 另外图中没有画出的还有: 产品和活动之间的数据流连接, 表示活动的输入输出产品. 资源到活动用关联连接表示资源被活动引用. 在控制流上还可以使用逻辑连接符“与”和“或”表示过程执行路径的选择.

2 4　形式化定义

基于以上对 VPML-S的元模型的分析, 建立了相应的形式化模型.

定义 **1**　过程定义 process, 是实际业务过程的数学描述, 它可以形式化地表示为一个七元组 process= < Activities, Products, Resources, Connections, Events, Attr, partners>. 其中 Activities是活动的非空有限集; Products是活动执行所需要的产品和输出的产品的有限集; Resources是活动执行需要的资源的非空有限集. Connections是活动之间, 活动与产品, 活动与资源之间连接的有限集; Events是过程执行过程中产生的事件的有限集. Attr是五元组 < id, name, accessLevel, author, created>, 它是过程属性集合, 分别定义了标识 id, 名称, 外部访问权限, 创建者和创建时间. 外部访问权限, 有 public和 private两种权限, public表示过程可以被外部过程复用和与外部过程协作, private则不能, 只能供本过程内部使用. Partners表示业务过程伙伴的集合, 表示此过程可能复用和协作的外部业务过程的集合, 集合中每一个业务伙伴 Partner可以表示成一个四元组 < id, name, sharedSubProcess, sharedPublicActs>, 分别定义了业务伙伴的唯一 id、名字、可供本过程使用的公有子过程和公有活动的集合.

定义 **2**　活动集合 Activities = {ManualActivities} ∪ {AutomaticActivities} ∪ {Sub Processes} ∪ {NestedSubProcess}. 其中 Manual Activities表示人工活动, 是需要人工参与完成的活动. Automatic Activities表示自动活动, 是机器可执行的. 人工活动和自动活动表示业务过程中的一个不可再分的工作步骤. 它们均是六元组 a = < id, name, desc, type, resource, accessLevel>. 其中 id, name, type分别是 a的唯一标识, 名称和类型, 类型可以定义为人工活动或自动活动. resource是 a执行时需要的资源, accessLevel是 a的外部访问权限, 有 public和 private两种权限, public表示活动可以被外部过程复用和与外部过程协作, private则不能, 只能供本过程内部使用.

Sub Process表示子过程, 可以使用子过程表示整个过程中的一部分, 从而可以支持层次式的复杂过程. 子过程具有双重性质, 它既是一种过程, 又是一种活动. 子过程是一个六元组 Sub Process= < Activities, Products, Resources, Connections, Events, Attr>, 各个元素的含义均和过程 process相同, 只是没有业务伙伴属性.

Nested Sub Process表示过程引用的外部过程 (来自 process中定义的业务伙伴)的公开的活动或者子过程, 是一种过程复用机制, 嵌入子过程只

是对外部过程的共享活动和子过程的一种引用,引用的活动和子过程实际执行还是在外部过程中执行,只是执行结果会传给本过程. Nested SubProcess= < id, name, externalProcessId, externalActId, inProduct, outProduct>. externalProcessId表示嵌入子过程所表示的外部活动所在的过程的 id, externalActId表示外部活动的 id, inProduct表示本过程传递给外部活动的输入产品, outProduct表示外部活动传递给本过程的输出产品.

定义 **3** 产品 Product是一个五元组 < id, name, desc, type, behaviorAttrs>,其中 id, name, desc, type,分别表示产品的唯一 id,名称,描述和类型. behaviorAttrs表示根据产品类型不同,产品所具有的属性的集合 behaviorAttrs= { attr1, attr2 …, attrN},不同的产品 behaviorAttrs集合中属性的个数和含义都是不一样的,比如,如果产品是文件,就有大小属性,修改时间属性,如果产品是数据库,就有数据库类型属性等等.产品用于表示在过程活动中被使用、创建、修改和传送的元素.预定义的产品统称产品,如果需要创建其他类型的产品可以使用元模型扩展机制来添加产品类型,比如添加软件,文件,数据库等产品类型.

定义 **4** 资源 Resources= {Roles}∪{Tools}∪{ Services},表示活动执行需要的资源,分为人员角色、工具和服务,其中角色和工具是一个六元组 < id, name, desc, useNumber, sumCount, costPerUnit>, useNumber表示执行活动需要的该资源的数目, sumCount表示该资源的总数目, costPerUnit表示该资源的单位成本. Service表示执行非人工参与的活动所需要调用或引用的组织外部或内部的服务,这个服务一般是 Web服务.它是一个七元组 < id, name, desc, pt, location, operation, FP>,其中 pt是服务的提供者,是一个四元组 < name, address, email, telephone>,分别定义了服务提供者的名字,地址,邮箱和联系电话. location是服务的 URL地址, operation指明了要调用服务的操作, FP的是调用服务要传递的参数集合.

定义 **5** 管道 pipeline,是一个七元组 < id, fromProcessId, fromActId, toProcessId, toActId, synType, MsgQuene>,表示不同业务过程之间的控制关系与数据流动关系. fromProcessId和 fromActId表示管道从哪个过程的哪个活动开始, toProcessId, toActId表示从管道到哪个过程的哪个活动. synType表示管道的同步类型,有同步管道和异步管道两种. MsgQuene是管道中的消息队列,消息 Msg= < id, src, des, content>, src表示消息发送源端, des表示消息发送目的端, content表示消息传递的数据内容.

3 VPML-S关键特性

3.1 面向服务的特征

VPML-S专为面向服务设计的,具有明显的面向服务的特征.除了一些需要人工完成的活动外,自动活动,子过程都可以由一个组织内部或者外部提供的服务(即 Service)来完成. VPML-S过程已经不是象以前一样基本依靠组织内部来完成的封闭过程,而是一个高度开放,面向服务的业务过程,过程中的很多活动是依靠第 3方的服务来完成的. VPML-S以一种全新的面向服务的观点来看待业务过程,对于原有的业务过程建模观点有了根本性的改变. VPML-S把整个业务过程看作一个服务,或者服务过程,把子过程看作子服务,把非人工参与的自动活动都看作服务,业务过程的执行就是过程请求一系列公共服务最终完成该业务的一个过程,而该业务可能又作为服务提供给其他一些更大规模的业务.一般来说,一项业务的完成都要依靠多个企事业单位或者一个企事业的多个部门共同协作,很少有一个业务是由一家单独的企事业或单独的部门来独立完成的,也就是说,任何一个企事业或者企事业内部的部门都要依靠其它企事业或部门的服务才能正常工作;反过来,任何一个企事业或企事业内部的部门只有向其他企事业或者部门提供服务才有存在的价值.所以以服务的观点来看待业务过程,才符合现实世界的业务模型.

3.2 过程复用机制

在企业业务过程中,有很多活动是可以复用的,比如象文档审批过程、财务报账过程每一个企业都会有,而且大致都是类似的,甚至相同.每次建立业务过程时都要建立这些重复的活动和子过程,做很多无用的工作,而且不利于过程模型的维护.如果能够把这些普遍使用的业务过程建立为公共业务过程,对外提供业务过程服务,那么避免了每个企业都重复建立同样类似的业务过程,节约了企业信息化成本,提高了效率.利用 VPML-S提供的嵌入子过程可以将外部公共过程引入到企业内部过程,作为企业内部过程的一部分来执行,企业只需要提供输入,而得到输出结果就可以了.

3.3 过程协作和互操作

每一个业务模型都有业务伙伴,业务模型与业务伙伴之间有协作关系,共同协作完成一项业务.各个业务过程之间的协作模型可以分为链式

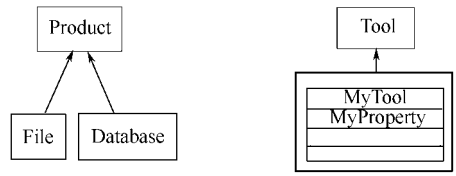
模型、嵌入子过程模型和并行同步模型. 链式模型表示业务过程之间的异步调用, 嵌入子过程表示业务过程之间的同步调用, 这两种模型可以使用本文的 Nested Sub Process分别加上同步和异步管道来实现. 而并行同步模型表示不同过程的多个活动要均到达某一个集合点, 比如某个时间点以后才可以继续向前执行, 先到的活动必须等待后到的活动, 并行同步模型可以直接使用同步管道来实现.

除了可以表示上述同步关系, 管道机制还可以表示共享资源同步关系和数据同步关系. 共享资源同步关系表示多个业务过程在共享资源时所产生的资源调度策略. 数据同步关系由于多个业务过程共同处理同一产品所产生的同步关系, 比如, 多部门会签等. 如果两个活动之间需要共享资源或者共享数据, 那么就把这两个活动用管道连接起来, 管道的箭头方向表示活动使用资源和数据的先后顺序关系. 业务过程的互操作指的是使用不同技术构建起来的业务过程在不同的执行引擎中执行的时候相互之间的调用和消息传递. 由于 VPML-S是面向服务的, 并且由它建立的业务过程在执行时是采用工业标准 BPEL语言, 所以在过程执行中, 可以通过 Web服务和 BPEL和大多数异构的业务过程互操作, 解决了异构业务过程的兼容性的问题.

3.4 用户扩展机制

VPML-S有两种用户扩展机制, 一种是图符替换机制, 另一种是元模型扩展机制, 即建模元素及其属性的扩展机制, 这样可以让用户自己来为自己的业务定制建模语言, 并且在将来业务发生变化时, 也可以自己扩展建模语言来满足业务建模的需要. VPML-S允许用户自己以多种图标格式设计自己的建模图符, 同时灵活的扩展建模元素及其属性, 最大限度的满足业务建模的需求.

VPML-S对于允许扩展的建模元素提供扩展点, 现有的扩展点是: 产品扩展点, 资源扩展点, 事件扩展点, 活动扩展点, 逻辑连接符扩展点和数据流扩展点. 下面以产品和工具为例说明, 见图 3.



a　通过继承 product元素　b　通过继承 Tool产生新的建模
增加文件和数据库两种产品　元素来扩展 Tool元素的属性

图 3　元模型扩展机制

4　应用实例

图 4是一个 C2C网上购物流程实例. 参与流程有买家流程, 卖家流程和快递流程, 它们互为业务伙伴. 为了突出重点, 图中只保留了活动和嵌入子过程, 管道, 控制流还有开始符和结束符, 其他相关元素均省去. 另外, 实例中的图符只是一种示例, VPML-S具有图符替换机制, 可以使用任意用户自定义的图符.

这个实例描述了 3个流程之间的协作交互和复用关系. 其中标号为①的活动所连接的资源都是 service, 即 C2C网站所提供的服务, 而标号为②的活动是人工活动, 需要连接角色资源, 标号为③的活动是嵌入子过程, 其所需要完成的活动由快递流程代理完成, 嵌入子过程只需要传递数据给代理流程即可. 快递流程由发货活动启动, 是链式协作模型, 由嵌入子过程 (发货)和异步管道 (发货请求)共同来实现. 带箭头的虚线表示管道, 三个流程之间通过管道来协作, 管道可以传递跨业务过程消息, 比如询问库存的消息等, 也可以传递跨业务过程的数据, 比如快递把货物传送给买家.

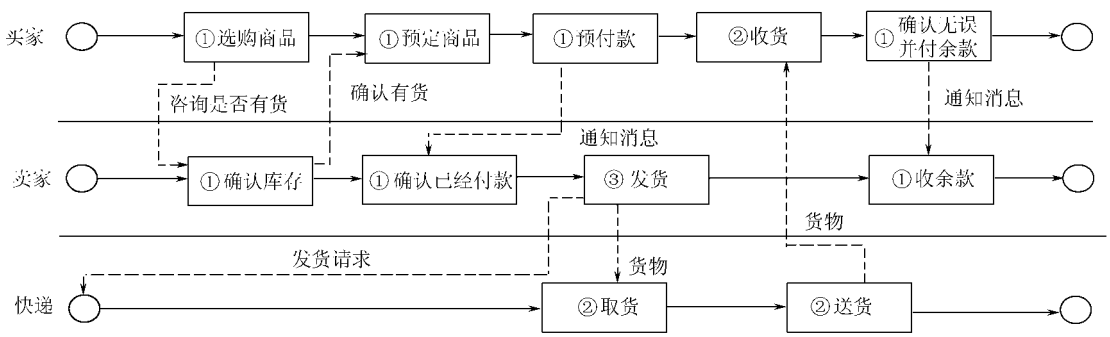


图 4　C2C网上购物流程实例

5 结束语

本文在原有 VPML语言的基础上,扩展出了一种面向服务的可视化过程建模语言 VPML-S,分析了它相对于其它过程建模语言的优势. VPML-S提供了对业务人员友好的可视化界面、灵活的用户扩展机制和严格的元模型定义,并且为面向服务的体系结构的业务过程建模提供了原生支持.有效解决了业务过程建模语言的过程复用,协作和互操作的问题.本文中 VPML-S的形式化定义只是初步的定义,下一步,将在 VPML-S的形式化 pi演算模型、模拟验证等方面做进一步的深入研究.

参考文献 (References)

[1] Ouyang C, Dumas M, HM A, et al. From BPMN process models to BPEL web services[C]// ICWS06. Chicago: IEEE, 2006: 285-292

[2] OASIS-wsbpel-v2.0-OS-2007, Web services for business process execution language[S]

[3] 周伯生,张社英.可视化过程建模语言 VPML[J].软件学报, 1997(增刊): 535-545
Zhou Bosheng, Zhang Sheying. Visual process modeling language VPML[J]. Journal of Software, 1997 (Supplement): 535-545(in Chinese)

[4] WFMC-TC-1025-2005, Process definition interface-XML process definition language[S]

[5] Kavantzas N, Burdett D, Ritzinger G, et al. Web services choreography description language version 1.0[R/OL], W3C, 2005. http://www.w3.org/TR/ws-cdl-10

[6] Martin D, Paolucci M, McIlraith S, et al. Bringing semantics to web services: the OWL-S approach[C]// SWSWPC 2004. California: Springer-Verlag, 2004: 26-42

[7] 张社英,刘又诚.过程模型建造环境 PMBE[J].软件学报, 1997(增刊): 563-564
Zhang Sheying, Liu Youcheng. Process model building environment PMBE[J]. Journal of Software, 1997(Supplement): 563-564(in Chinese)

(责任编辑:张 嵘)

(上接 1330页)

参考文献 (References)

[1] 常志明,周海芳,蒋艳凰,等.流域变换算法中过度分割的平滑解决方法[J].计算机工程与科学, 2005, 27(3): 29-31
Chang Zhiming, Zhou Haifang, Jiang Yanhuang, et al. A new method of solving over-segmentation in watershed algorithms[J]. Computer Engineering & Science, 2005, 27(3): 29-31(in Chinese)

[2] Wang Demin. A multiscale gradient algorithm for image segmentation using watersheds[J]. Pattern Recognition, 1997, 678(12): 2043-2052

[3] Hernandez S, Barner K. Joint region merging criteria for watershed-based image segmentation[C]// IEEE International Conference on Image Processing. Vancouver, Canada: IEEE, 2000, 2: 108-111

[4] 王小鹏,罗进文.基于形态学梯度重建的分水岭分割[J].光电子·激光, 2005, 16(1): 98-101
Wang Xiaopeng, Luo Jinwen. Watershed segmentation based on morphological gradient reconstruction[J]. Journal of Optoelectronics·Laser, 2005, 16(1): 98-101(in Chinese)

[5] Gauch JM. Image segmentation and analysis via multiscale gradient watershed hierarchies[J]. IEEE Transactions on Image Processing, 1999, 8(1): 69-79

[6] Park J, Keller JM. Snakes on the watershed[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(10): 1201-1205

[7] Haris K, Efstratiadis S N, Maglaveras N, et al. Hybrid image segmentation using watersheds and fast region merging[J]. IEEE Transactions on Image Processing, 1998, 7(12): 1684-1699

[8] 崔明,孙守迁,潘云鹤.基于改进快速分水岭变换的图像区域融合[J].计算机辅助设计与图形学学报, 2005, 17(3): 546-552
Cui Ming, Sun Shouqian, Pan Yunhe. An image region merging algorithm based on modified fast watershed transform[J]. Journal of Computer-aided Design & Computer Graphics, 2005, 17(3): 546-552(in Chinese)

[9] Wang Jia, Lu Hanqing, Eude G, et al. A fast region merging algorithm for watershed segmentation[C]//Proceedings of ICSP'04: the 7th International Conference on Signal Processing. Beijing: IEEE, 2004, 1: 781-784

[10] 王欣,王德隽.离散信号的滤波[M].北京:电子工业出版社, 2002: 161-165
Wang Xin, Wang Dejuan. The filtering of discrete signal[M]. Beijing: Publishing House of Electronics Industry, 2002: 161-165(in Chinese)

[11] Wu Xiaolin. Adaptive split-and-merge segmentation based on piecewise least-square approximation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1993, 15(8): 808-815

(责任编辑:赵海容)