基于构件库/工作流的可视化软件开发

张 成[1,2],吴信才[1,2],罗 津[1],胡茂胜[1]
ZHANG Cheng[1,2],WU Xin-cai[1,2],LUO Jin[1],HU Mao-sheng[1]

1.中国地质大学 信息工程学院,武汉 430074
2.教育部地理信息系统软件及应用工程中心,武汉 430074
1.Faculty of Information Engineering,China University of Geosciences,Wuhan 430074,China
2.GIS Software Research and Application Engineering Center of the Ministry of Education,Wuhan 430074,China
E-mail:zhchwyl@163.com

ZHANG Cheng,WU Xin-cai,LUO Jin,et al.Visual software development based on software structural component library/workflow.Computer Engineering and Applications,2008,44(10):82-87.

Abstract: A new approach to make visual software development based on the combination of workflow technique and software component library technique is presented.The author brings up the idea of visual software development method which is based on software component library,which is to programming visually using the workflow visual edit tools to fabricate new software with the software components in some software component library.And also brings up the idea of separating the process control from software component,which is to using the workflow engine to drive software components to work together.Software component library is emphasized to be the foundation,it contains the components to build up complicated software,and the components can be any scale and level;it's an open system,supports assembling software with components and all development methods of B/S,C/S.It can be used to construct software project from bottom to top or from top to bottom.Finally the advantages and disadvantages are discussed,too.

Key words: software component;software component library;visual programming;workflow

摘 要:在研究构件技术的基础上,结合工作流技术提出了一种新的软件开发模式,即通过将可视化的构件库与可视化的工作流编辑系统全面融合实现零编程的软件搭建平台。构件库包含了多层次和多粒度的可视化软件构件集合。工作流以业务流程为核心来组装各种构件以实现可视化的软件开发。论文提出了软件构件运行与逻辑控制分离的思想,通过构件库不同层次构件为用户提供良好的软件扩展性和移植性,通过工作流引擎的流程控制取代程序流程的过程控制实现面向业务的快速软件搭建,并支持任意流程的实时测试。这种软件开发模式拓展了传统软件工程的过程开发方式,能较好兼容自上而下和自下而上的软件工程方法,并能适用于 BS/CS 开发模式。该模型已在新一代 GIS 平台——MapGIS 搭建平台中得到成功应用。最后分析了该软件开发方法的优缺点。

关键词:软件构件;构件库;可视化编程;工作流
文章编号:1002-8331(2008)10-0082-06 **文献标识码**:A **中图分类号**:TP311

1 引言

1.1 构件与构件库研究现状

自 1968 年 Mcilroy 在 NATO 软件工程会议中首次提出"软件复用、软件构件、构件工厂"等[1]概念以来(在商业软件和商务活动中,Component 通常也被称为组件),虽然软件构件的定义不断变迁[1-3],但基于构件的软件开发(CBSD)一直被视为解决软件危机,实现软件工业化生产切实可行的重要途径[4]。

近 40 年来,国内外同行从不同的角度对软件构件技术进行了许多有价值的研究[5]。这些研究涉及到了软件构件的整个生命周期过程,从软件构件的定义[1-3,6]到构件的属性[7];从构件的描述[7]和分类,构件的标准化[4,8]到构件之间的关系[7];从构件的制作与生产,构件的查询与获取及构件的表示和检索[6],构件的模型与裁剪,构件的组装与反馈[15]到最终的构件复用。在国内软件构件技术当前已经发展成为软件复用的一个非常重要的学科分支[8]。

随着软件构件技术的深入人心,各种异构的软件构件的数量愈来愈庞大,构件开发者关注的重点已经从最初强调构件的可复用性,转变到支撑软件构件整个生命周期管理的构件库的

基金项目:国家高技术研究发展计划(863)(the National High-Tech Research and Development Plan of China under Grant No.2006AA12Z218);国家科技公关计划项目(the Key Technologies R&D Program of China under Grant No.2002BA107B01)。

作者简介:张成(1975-),男,博士生,主要研究方向为地理信息系统、构件式 GIS 技术;吴信才(1953-),男,教授,博士生导师,主要研究方向为基础地理信息系统研究与应用;罗津(1979-),男,博士后,主要研究方向为地理信息系统应用软件开发;胡茂胜(1981-),男,博士生,主要研究方向为地理信息系统与空间数据库。

收稿日期:2007-09-19 **修回日期**:2007-12-17

建设。大部分构件库系统对构件的管理和维护原理类似,差异主要表现在各个构件库所采用的实现技术、系统的侧重目标,以及处理的构件类型和形态方面。

构件实现技术已经比较成熟。业界通用的构件模型主要有OMG(Object Management Group 对象管理集团)的 CORBA 技术;微软的 COM/DCOM(Component Object Model/Distributed Component Object Model);SUN 的 JavaBeans/EJB(Enterprise Java Beans 企业 JavaBeans)。它们都满足 Tracz 提出的 3C 模型。即构件三个部分的描述:概念(concept)描述构件的功能;内容(content)描述构件怎样完成概念所描述的功能;语境(context)描述构件与其他构件的关系。

按构件组织形式不同,构件库可以分为集中式构件库和基于网络的分布式构件库。前者由于构件功能相对集中,多用于专业领域的传统软件开发。后者所管理的构件集在物理上呈分布结构,在逻辑上是一个整体,为不同领域构件提供一个开发的注册、管理、检索、交易的场所,是目前构件库的主流趋势。

国内外比较有代表性的构件库有欧盟信息技术计划 ESPRIT 中开发的 REBOOT(Reuse Based on Object Oriented Techniques)系统,北京大学青鸟构件库管理系统(JBCLMS)(Jade Bird Component Library System),CMU SEI 开发的 Agora 构件搜索引擎,和美国 Colorado 大学开发的 CodeBroker 构件库[9]。这些模型均是学术界提出的指导性模型,抽象层次比较高,用户可以根据不同的问题域对其进行扩展[9,10]。其中 REBOOT 构件库系统由一个存储构件的复用库以及一组支持构件生产、考查、分类、选择、评估和适配的复用工具组成。JBCLMS 是国家九五重点科技攻关项目,它是一个基于 Internet 的软件资产库管理系统,包括构件库、构架库以及相应的库管理工具。JBCLMS 是国内比较成熟的构件库管理系统,诸多文献均有关于 JBCLMS 的应用实例描述。根据青鸟构件库模型,广义的构件包含分析件、设计件、测试件、代码件等多种构件。由于分析件、设计件和测试件难以形成一个可以让机器理解的形式化描述,因此这类构件并没有为软件生产效率的提高带来多大的价值,只有代码件即狭义的构件随着高级语言的不断发展而得到广泛应用。如无特殊说明,下文中描述的构件专指狭义构件。

Agora 构件搜索引擎[9]提供一种类似于 UDDI 的机制,允许构件开发者通过 Web 方式在线搜索或注册构件。CodeBroker 是美国 Colorado 大学开发的一个面向 Java 程序开发的构件库原型,其最大特色是构件库与源程序编辑工具实现无缝集成,为用户提供主动查询服务[9,11]。

除上述典型构件库外,商业构件库还有[9]Reuse Repository、SALMS 软件资产库管理系统、ASRR 自动软件复用库、RLT(复用库工具集)和 HSTX 复用库,政府级构件库如 DSRS 美国国防部软件库系统、LID 构件库互操作示范工程、I-CASE(计算机辅助软件工程集成环境)、MORE(面向多媒体的构件库)、SAIC/ASSET 面向软件工程的软件资产复用技术、PAL 公共 Ada 库、CAPS 软件可复用构件库和 DISA(Ada 库暨美国国防部信息系统代理复用库[12]等。

上述构件系统研究关注的重点集中于构件的注册、设计、查询、分类等构件的管理功能,其构件的产生仍然离不开手工编写代码,缺乏可视化编码的工具。即便构件的组装能够在相匹配的构架中实现热插拔,但却往往与构架紧密结合缺少柔性,很难适应其它异构环境,影响构件在不同的构架下的通用性,往往更多地强调构件的复用性可用性等方面[13,14],对构件库的研究局限于为了软件开发者查询,理解和选取构件的阶段,缺乏对构件库的分层分类描述和重构的设计。在一定的构架下进行组装业务时,当现有的构件不能够满足全部要求,需要进行重组时往往需要"大动手术"。对程序员而言,软件开发始终不能摆脱编写程序代码的窘境。

1.2　工作流研究现状

工作流的思想起源于 20 世纪 70 年代中期办公自动化(OA)和工业控制(CSCW)领域的研究工作[15]。在不同发展阶段和技术角度,研究者及工作流产品供应商对工作流和工作流管理系统有不同的定义[16,17]。1993 年工作流管理联盟(Workflow Management Coalition,WFMC)的成立标志着工作流技术开始进入相对成熟的阶段[18-20]。工作流管理联盟对工作流和工作流管理的标准定义是[17]:工作流是一类能够完全或者部分自动执行的经营过程,它根据一系列过程规则、文档、信息或任务能够在不同的执行者之间进行传递与执行。工作流管理系统是一个完全定义、管理和执行工作流的系统,它通过在计算机中预先定义好的工作流逻辑来驱动工作流事例的执行[21,22]。

90 年代后期,随着网络的普及和各种分布式技术的成熟,工作流的协作优势开始显现,更多、更新的技术被集成进来,文件管理系统、数据库、电子邮件、移动式计算、Internet 服务等都已被容纳到工作流管理系统之中[16]。根据所采用的任务项传递机制的不同,工作流系统分为 4 类[27]:(1)基于文件的工作流系统:以共享文件的方式来完成任务。代表产品有 FileNet 的 VisualWorkFlow,IBM 的 FlowMark。(2)基于消息的工作流系统:通过用户的电子邮件系统来传递文档信息。代表产品有 Novell 与 FileNet 合作开发的 Ensemble,JetForm 的 InTempo,Keyfile 的 Keyflow。(3)群件与套件系统:因为这类产品都需要依赖自己系统的应用基础结构,包括消息传递、目录服务、安全管理、数据库与文档管理服务等,它们本身就构成了一个完整的应用开发环境。代表产品有 IBM 的 Lotus Notes,Microsoft Office 的 Exchange,Novell 的 GroupWise。(4)基于 Web 的工作流系统:通过 WWW 来实现任务的协作,已成为一种最新的市场流行趋势[16]。如 Microsoft 大力推行的 Biztalk 和免费的 WWF 都比较适用于 Web 工作流开发[22]。国内的科研和学术机构对于工作流的研究起步较晚,近年来取得了一定的成果,但尚未形成产业化。

在工作流系统设计中,工作流建模最为关键。WFMC1994 年提出了工作流的参考模型及相关标准,接口组成与详细定义说明[18,22]。常见的工作流建模方法有流程图、状态图、活动网络图、IDEF 系列、ECAA(事件一条件一动作规则)、事件驱动的过程链模型、Petri 网等建模方法。其中,基于活动网络图和 Petri 网理论的建模方法应用最广,典型工作流产品都是基于这几类模型实现的。文献[18]对各种工作流模型优缺点进行了详细地比较,在此不再赘述。

2　基于构件库/工作流可视化软件开发方案

构件技术作为面向对象的软件开发过程中实现业务功能的技术应用已经比较成熟,而工作流作为解决业务流程的利器一直在局限于各行业的 OA 办公,企业信息化及工业控制领域[21,22]。两者都只部分解决了面向对象软件开发过程中的功能设计和流程设计的问题。程序指令流也是一种工作流,程序中各种操作都可以封装为构件库中的构件,包括各种基本的操作,程序流的各种操作都可以映射为工作流中的活动。能否将

两者优势有机整合到程序设计过程进一步提高软件开发的效率呢？基于这种思路，做了一定的探索性研究。

文献[15]提出了一种面向行业的工作流系统构件化开发方法。但这种方法仅适用于领域构件，通用性和移植性不强，且未能实现构件库对构件的可视化管理，可扩展性有限。本文将从可视化软件开发角度，结合软件构件及构件库设计和工作流的可视化流程建模实践提出一整套基于构件库/工作流的可视化软件开发解决方案。

一个现实的软件系统一般地由数据、功能、界面、过程几个层面的协同合作来完成。为了使构件库能更好面向实际应用，需要建立相应的界面控制和过程控制的子系统以协助软件开发，即工作流管理系统。以工作流驱动构件运行，不同于构件在特定构架下的运行模式。构件库作为通用功能和业务功能构件仓库，业务流程和权限的管理在工作流管理系统通过可视化界面预置定义。一个流程包含一连串按业务逻辑连接的节点，每个节点上绑定一个功能构件。系统调用起始节点即启动整个流程，根据一定条件依次执行被调用流程节点上绑定的功能构件以实现流程的功能目标。工作流负责管理业务控制流程和数据流。构件仅仅负责功能的执行，从而达到业务逻辑和功能的分离。

2.1 开放灵活的软件构件库设计

基于构件的软件开发必须建立相应的构件库以满足各种不同层次的软件开发需要。构件库支持纵向分层，横向分类管理，便于用户根据自身行业特点进行自定义细分子类和扩展。通过对构件库的分层管理，屏蔽构件的实现细节，保证构件属性的独立。多个低层细粒度的简单构件聚合可以组成复杂的粗粒度高层构件，这种结构不同于以往构件库的描述，极大地提高了构件库本身的完备性和扩展性，灵活性。用户可以根据需要用底层构件快速开发原型系统，并在适当的时候通过高层构件直接替换复杂的聚合构件来提高系统的运行效率。

构件库的分层图如图 1 所示。

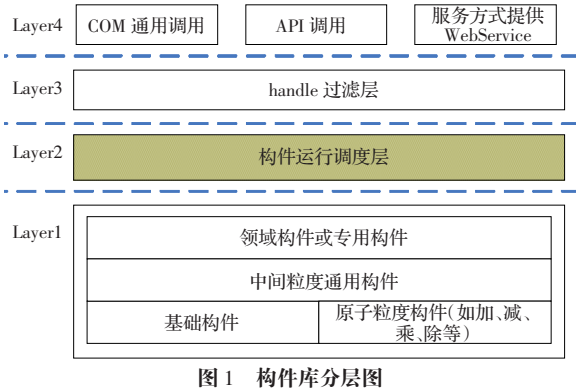

图 1 构件库分层图

Layer1：构件驻留层，其中按照构件粒度及应用性又可以划分为 3 个子层：

(1)基础构件，这一层提供原子粒度构件，可以由构件库系统自带，也可以由用户添加。例如：加，减，乘，除，乘方等基本构件，由于这些原子粒度构件的引入，增强了构件库的完备性，才使得基于工作流的可视化编程具有强大的功能以至于取代手工编写代码。这一层是基础性的构件，适应各种应用。本部分可以参考程序语言提供的基本操作，建立原子构件层，保证构件库的完备性。

(2)中间粒度的通用构件，可以通过基础性的构件组装而成，也可通过编写代码而得。这一层属于通用构件，与具体业务无关。通过建立标准构件层，保证易组装性和适应性。

(3)领域构件，由中间粒度构件与基础构件重组而成，往往面向某一个专业领域提供服务。该层提供具有高可扩展性。

Layer2：构件调度层，负责构件运行环境的准备及与调用端建立 Session，维护 Session 中各种参数的生命周期。

Layer3：Handle 过滤层，其作用主要是提供了扩展构件库系统的能力，用户可以写专用的插件来截获上层的每次调用及其参数，这个功能可以用来调试构件，观察构件的行为，也可以搜集调用数据做一些构件的调用统计，例如：在一个时间段上哪些构件被调用及其被调用的次数等等。

Layer4：用户调用接口层，对用户提供多样性的调用手段。这层是用户最关心的与用户关系最大的，这一层对外要提供丰富的调用接口，例如：API 调用、COM 调用，以服务的方式提供调用。注册到构件库的构件可能是某个 COM 组件，但在该层中可以自动生成其它类型的调用接口。

构件库的主要功能是提供对构件的管理，包括构件的注册、注销、删除、更新等。用户可以根据需要自行开发符合通用规范的各类构件，也可以直接导入已存在的外部构件。这些外部构件可以来自网络上远程分布式部署的子构件库、Internet 上的网络构件库，以及 WebService 构件。外部构件的存在丰富了构件库的功能，满足各种应用需求，增强了构件库的扩展性和完备性。

值得说明的是，由于开发平台的局限，所描述的构件通常指基于微软的 COM 技术(C++的 COM 组件，或者 C# 的程序集)实现的，因此构件库具有 COM 技术的部分特征，如跨平台的异构特征，它不依赖于加载构件的操作系统和运行环境，使得构件库具有良好的平台兼容性和适应性。同理而言，基于 CORBA 或 JAVA BEANS 技术同样可以建立与之适应的构件库。由此可见基于上述思路设计的构件库具有扩展性、移植性、兼容性、适应性。其软件开发流程如图 2 所示。

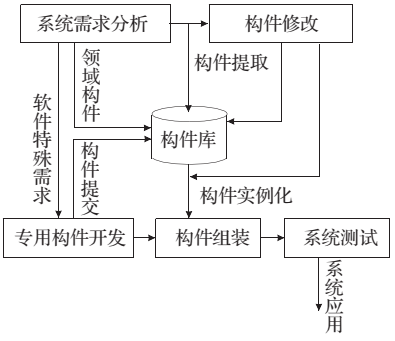

图 2 基于构件的软件开发流程

2.2 工作流系统设计

WFMC1994 年发布的工作流参考模型约定了工作流管理系统需要包含 6 个基本模块[15]：(1)工作流执行服务：激活并解释过程定义，完成工作流过程实例的创建、执行与管理，为工作流程的运行提供一个运行时环境。(2)过程定义工具：提供对实际业务过程进行分析、建模的手段，生成业务过程描述(过程定义)。(3)其它工作流执行服务：与其它异质的工作流执行服务来辅助完成复杂系统。(4)客户应用程序：提供人工干预手段以辅助过程实例运行。(5)被调应用程序：工作流执行服务在过程实例的运行过程中调用的、用以对应用数据进行处理的程序。(6)管理及监控工具：对 WFMS 中过程实例的状态进行监控与

管理。工作流参考模型如图 3 所示,下面结合该模型详细论述设计思路。

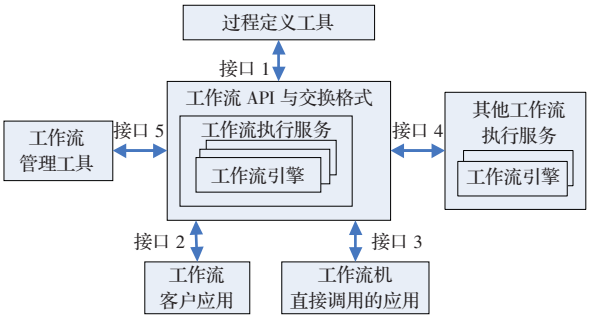

图 3　工作流参考模型

2.2.1　工作流执行服务

工作流执行服务即工作流引擎。它负责解释流程定义(被称为复合构件)及复合构件的实例化并向构件库发送构件执行命令,但在基于构件库的面向编程的工作流引擎中,主要依照定义的构件运行逻辑激活相应的节点,激活节点的过程就是将即将执行的任务发送给构件库,节点具体任务的执行是由构件库来执行。由此实现了构件与业务逻辑的分离。构件库则负责维护 Session 的状态信息负责维护对象的创建及对象的生命周期,功能的具体运行仍然在构件库的环境下运行。

2.2.2　过程定义工具

工作流的过程定义部分包括对流程实体、控制类型、用户权限等管理。其中用户权限管理在管理工具部分介绍。

(1)流程实体

对于工作流的过程定义的实体部分,采用活动网络图模型来实现工作流的过程建模,用 XPDL 来记录过程描述。为更好地结合软件构件建模,对活动网络图进行了适当的改进。

①将文献[23]所定义的 15 种节点抽象为 4 种节点:

起始节点:一个工作流的开始;

终止节点:一个工作流的结束;

普通节点:工作流执行过程中的活动节点,执行完当前功能后,程序逻辑过渡到下一节点。

子流程节点:用于嵌套子流程的活动节点,主要负责主流程与子流程的控制流的切换。

每一个独立的流程中只能有一个开始节点,但可以有多个终止节点。因为流程可能会在不同状态下结束。

子流程节点允许多层嵌套子流程,子流程节点不作为所负载子流程的起始节点和终止节点。它仅负责主流程与子流程的控制流切换。所嵌套子流程自身有起始节点和终止节点。子流程不允许自身嵌套。

②将文献[23]所定义的三种连接弧综合为一种有向连接弧。

实际工作流由一系列节点和有向连接弧组成。基于构件库的工作流可以理解为一个有向图构成。有向图中的节点元素表示可执行的任务,它被映射到构件库中的一个构件功能,节点间的连接弧代表过程中的控制流。以连接弧体现过程逻辑,数据的传递主要体现在构件参数之间的传递,再加上连接弧上的条件,就实现了用工作流的逻辑控制能力驱动构件的运行。

为了适应工作流的移植性和扩展性,系统为流程设计提供了一定数量的模板,用户在可视化的客户端程序通过简单的拖拽实现业务和逻辑控制的工作流建模,并允许设计模板的数据库记录与描述工作流设计的 XPDL 文档相互导入导出功能,极大地方便了资源的共享。

(2)控制类型

对于工作流的过程定义的控制类型部分。不同的文献有不同的分类[27]。简单起见,流程控制提供最简单的“顺序,选择,循环”控制以匹配可视化的程序设计逻辑控制过程。其他文献[23,24,27,28] 所描述的复杂控制流程均可由这三者复合而成。图 4–图 7 为三种简单逻辑控制示意图。

构件运行控制的工作流模型:

(1)顺序(如图4)

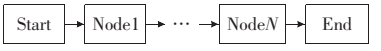

图 4　用工作流控制的构件顺序执行过程

顺序控制,是最简单的控制,只要连接弧上的条件设为 true,工作流引擎就会按照顺序向构件库发送调用请求信息。

(2)选择(如图 5)

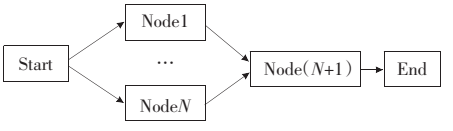

图 5　用工作流控制的构件选择执行过程

选择控制,同样依赖连接弧上的条件是 true 还是 false,哪条连接弧上的条件为 true,就按哪条路径执行。但这里与编程中的逻辑控制可以有不同,两条或分支出来的多条连接弧上的条件可以同时为真,这种控制能力实际上已经丰富了编程中逻辑控制能力,具体如何调度参见 2.4 节。

(3)循环(如图 6)

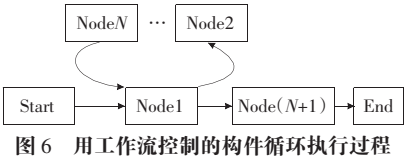

图 6　用工作流控制的构件循环执行过程

循环控制,同样依赖连接弧上的条件是 true 还是 false,放在封闭的回路上的构件都将被循环执行,在闭合回路中只要有一条连接弧上的条件为 false,那么整个循环宣告结束。在循环体及循环体向下连接部分,基于工作流引擎调度算法中要优先调度循环体,而仅当循环体执行完,才能继续向下执行。在循环控制中,同样要注意循环条件的构造,避免条件永远为真而造成死循环。

(4)带子流程的复合控制(如图 7)

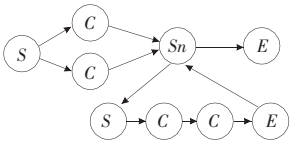

图 7　复合流程示意图

S 表示开始(Start)节点,*E* 表示终止(End)节点,*C* 表示普通(Common)节点,*Sn* 表示子流程(Subnode)节点。

2.2.3　其他工作流执行服务

系统数据库清晰地记录了工作流建模中设计活动网络图及其相互关系,只要将数据库记录按照一定的协议导出成 XPDL 文档,再转发给异构的工作流系统。对方按约定协议解

析 XML 文档即可转换为相应的工作流模型，从而实现异构工作流系统的交互式的操作。

2.2.4 客户端程序

工作流管理系统提供客户端可视化界面。目前工作流建模部分需要在服务器端进行配置，而权限管理部分可以通过 B/S 模式的浏览器直接登录系统进行配置管理。

2.2.5 被调用应用程序

被调用的应用程序在本文前面重点描述的功能构件。工作流建模中每一个节点都可以绑定一个构件。当流程驱动到当前节点时，系统将建模过程中节点预置的参数传给绑定的构件，执行构件代码实现其相应功能。节点间的流程跳转通过连接节点的有向连接线上的条件来实现。一个流程就是一连串按照条件执行预置构件的过程，以此来实现程序的逻辑控制从而实现实际的业务流程。

2.2.6 管理与监控工具

为适应各个业务环境下不同用户角色与权限设置，在设计工作流管理系统时，单独建立了一个开放式的机构管理模块。用来配置系统用户的机构，职务与特殊功能集。工作流节点的操作权限可以赋给机构（即机构内所有用户），单个用户，或者某个职级的部分用户。不同级别之间的操作权限允许向下传递。对于用户机构、职务交错的特别操作允许以功能集的方式给特定用户赋权。

在工作流监控方面，流程设计过程中允许任意节点的热拔插，提供所见即所得的实时调试功能，并支持在节点和连接线上设置断点跟踪，而无需更改设置和频繁编译，为设计人员节省了大量的调试时间。

该工作流设计模型提供了完整的工作流元素概念定义和工具，并支持界面化图例表示，具有完整的工作流分层与嵌套表达能力，且能随业务流程的变化而实时优化，总的说来，一定程度具备文献[26]提出的理想工作流模型的 6 点要求，即（1）形式化要求。（2）图形化特征。（3）较完整的表达能力。（4）层次性。（5）便于性能分析与优化。（6）柔性。

2.3 构件与工作流的相互依赖

基于工作流/构件可视化开发包括业务流程的建立和应用功能的实现两个部分，分别由工作流和构件库中的构件来实现。两者相对独立，而又紧密合作。

独立是指构件和工作流的设计开发均可以脱离于对方环境独立进行，彻底从软件开发角度将功能和控制逻辑分离，从而提高软件开发的效率。构件库负责管理构件运行的环境及调用构件运行，返回工作流引擎关心的结果，维护调用状态（如：变量的生命周期，参数传递等），而工作流引擎完全负责解释构件的执行逻辑，并将任务交给构件库去执行。

构件运行与流程控制分离还有如下优势：

（1）部署灵活：构件库的部署可以与工作流的部署分开，它们即可以运行在本地，也可以运行在远程；运行的 PC 环境可以相同，也可以不同；可以进程内，也可以在不同的进程中，这种特性使开发出来的应用系统可以实现分布式；

（2）开发快捷：构件库可以作为独立产品并行开发；流程与功能的分离简化了过程定义的复杂度，可先对已知的流程进行定义，把未知部分流程留待构件完备后进行设计，也缩短了开发周期；

（3）扩展方便：构件库和工作流允许用户随意扩展和改装。例如：可以增加构件库的负载均衡和系统过滤功能等；业务流程变化时只需要局部重建工作流模板，配置相应处理构件和权限即可，甚至不需要去编译，调试就能再次启动应用系统，保持了工作流系统的灵活性和扩展性；

（4）封装性高：通过构件库可以对外统一暴露调用接口，构件库内部的变化不影响上层调用。

合作是指构件与工作流统一于业务应用流程，流程节点将各种层次和粒度的构件与工作流节点的直接绑定来实现业务应用目标。对于程序员来说，工作流与构件库的结合就是将一个代码段（动态库，类，程序集）映射到一个流程，代码中的各个函数映射为构件库中相应的构件，构件需要工作流引擎的调度运行才能完成相应的功能。

2.4 编程的可视化

从汇编语言开始，人们就已经习惯用代码方式进行软件设计，随着计算机技术的发展，编程语言及手段都在不断提高，可视化编程的出现正在逐步取代传统的手工编码方式。现在的可视化编程还主要集中在界面的可视化，编码的可视化即搭建式开发领域较少有人涉足。在基于构件库/工作流的可视化软件开发中，假设基础构件足够完备时，任何程序功能的实现都可以归结为对现有构件的组装，编码的可视化即可转换为构件的可视化组装，极大地降低了软件开发的门槛，更多的程序员将从手工编写代码的工作中解脱出来。

将构件映射为流程节点上的活动，由连接弧上的条件控制来判定构件之间的运行次序。构件运行过程中参数的传递以及参数的生命周期由构件库进行维护[1]。在工作流可视化编辑环境中必须嵌入构件库的可视化表达。构件库可以采用树形结构按照功能进行分类，用户在进行编程时只需要将构件库中的构件向流程面板中拖放，鼠标操作连接弧上的各个节点，通过界面配置连接弧上的条件和节点上的传递参数，就可完成一个复合构件的可视化组合。在构件库完备的情况下，基于构件/工作流可视化软件开发可以实现无编码的快捷编程。

由于这种可视化开发方法对等于手工编写代码，因此适合自下而上的软件开发，也适合 C/S，B/S 的模式的开发。随着构件库对外服务能力的增强，构件库可以部署在 intranet/internet 上对外提供功能服务。使用这种方法成功地开发了 MapGIS 可视化搭建平台，为用户提供了全新的可视化开发手段，证实了该方法的可行性。

当然，不可否认，基于构件库/工作流的软件可视化开发方法目前还存在一些有待改进之处：构件运行与流程控制逻辑分离带来了一些性能损失，不适合做性能要求非常高的功能。原子粒度密集的复合构件也会影响系统的性能，建议首先采用编码方式实现中粒度的领域通用构件，然后再用领域通用构件来构建系统，有助于提高系统的性能。可视化工作流设计部分有待进一步丰富操作元素和界面，以增强系统的表现能力。

3 结论与展望

在基于构件的软件开发方法下，程序开发模式也相应地发生了根本变化，软件开发不再是算法+数据结构，而是构件开发+基于体系结构的构件组装[25]。在基于构件库/工作流的可视化软件开发模式中，软件开发将更为简捷：软件开发=构件组装+流程拼接，称之为搭建式开发。目前，软件开发者已经从注重构件重用性开始转向构件库的体系结构建设，尚缺乏通用成熟的构件库管理系统。本文认为构件库系统应该象数据库系

统一样去发展，通过不断抽象而最终建立相对成熟的符合一定通用规范的标准构件库。就像数据库系统通过实现数据与程序的分离而得到广泛应用一样,构件库/工作流的可视化开发模式实现了流程与功能的分离以及无编码软件开发,将来也一定会受到人们的青睐,成为构建各种应用系统必不可少的基础设施。

参考文献:

[1] Mcllroy M D.Mass-produced software components,software engineering concepts and techniques[C]//1968 NATO Conference on Software Engineering,Van Nostrand Reinhoid,1976:88-98.

[2] 马亮,孙艳春.软件构件概念的变迁[J].计算机科学,2002,29(4):28-30.

[3] 艾萍.构件柔性组装描述的形式化方法研究及其在水利领域的应用[D].南京:河海大学,2005.

[4] 赵俊峰.软件构件标准概述[J].信息技术与标准化,2006(6):10-13.

[5] 杨芙清.构件技术引领软件开发新潮流[J].中国计算机用户,2005(6):42-43.

[6] 李朝辉.基于构件复用技术的组态模型及平台研究[D].大连:大连理工大学,2005.

[7] 张建奋.基于构件的 GIS 软件开发研究[D].杭州:浙江大学,2002.

[8] 杨芙清.软件复用与软件构件技术[J].电子学报,1999,27(2):68-75.

[9] 潘颖,赵俊峰,谢冰.构件库技术研究与发展[J].计算机科学,2003,30(5):90-93.

[10] 费玉奎.构件技术发展综述[J].河海大学学报:自然科学版,2004,32(6):696-698.

[11] Ye Yun-wen.An active and adaptive reuse repository system[C]//Proc of 34th Hawaii Intl Conf on System Sciences(HIC-SS-34),Software Technology Track.Maui,HI:IEEE Press,2001-10.

[12] Guo J,Luqi.A survey of software reuse repositories[C]//Proc of the IEEE Intl Conf and Workshop on the Engineering of Computer Based Systems(IEEE ECBS'2000),Edinburgh,Scotland,UK,2000.

[13] Caballero R,Demurjian S.Towards the formalization of a reusability framework for refactoring[C]//Gacek Ced.LNCS 2319:Proc of the 7th Intil Conf on Software Reuse.Berlin:Springer-Verlag,2002:293-308.

[14] Alda S,Cremers A B.Towards composition management for component-based peer-to-peer architectures[J].Electronic Notes in Theoretical Computer Science,2005(114):47-64.

[15] 金正晔.工作流系统的构件化开发[J].计算机工程与设计,2006,27(23):4592-4595.

[16] 罗海滨.工作流技术综述[J].软件学报,2000,11(7):899-907.

[17] 付伟.工作流技术综述[J].河北北方学院学报:自然科学版,2007,23(1):68-70.

[18] 蒋国银.工作流过程建模理论综述[J].计算机系统应用,2006(3):90-93.

[19] 陈丽萍.基于构件的工作流系统[J].科技信息,2007(17):214-216.

[20] 赵瑞东.工作流与工作流管理技术综述[J].科技信息,2007(8):105-107.

[21] 朱金华.OA 系统中工作流引擎的设计[J].微计算机信息,2007,24(5):216-217.

[22] 谢熹.基于工作流的继电保护定值管理系统[J].电网技术,2006,30(16):64-68.

[23] 范玉顺.一种提高系统柔性的工作流建模方法研究[J].软件学报,2002,13(4):833-838.

[24] 范玉顺.工作流管理技术基础[M].北京:清华大学出版社,2001:63-66.

[25] 杨芙清.软件复用及相关技术[J].计算机科学,1999,26(5):1-4.

[26] 曾炜.工作流模型研究综述[J].计算机应用研究,2005(5):11-13.

[27] 蒋建民.基于组件和工作流技术的系统模型[J].计算机工程与应用,2002,38(15):63-64.

[28] 孙瑞志,史美林.支持工作流动态变化的过程元模型[J].软件学报,2003,14(1):62-66.

(上接 63 页)

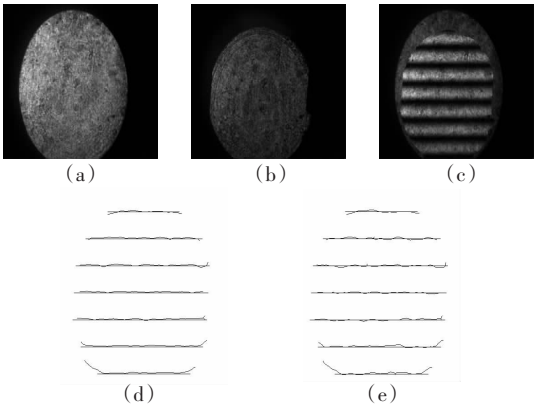

图 3　模拟图像条纹中心定位

移,而极值方法则在周边随机分布,对于后续的拟合计算来说,细化方法导致等相位面整体偏移更多。定位误差比较大的部分为条纹两端，当条纹被截断方向与条纹方向不垂直的时候,条纹延伸方向产生了改变,导致结果偏差特别大,需要进一步进行研究,可行的方法是摒弃两端的数据。

4　结束语

本文采用先去噪后分割条纹,再细化条纹的方法,最后用多项式拟合条纹垂直方向的灰度数据,用极值法计算条纹中心的精确位置,既避免了细化条纹方法的方法误差,又提高了条纹中心的定位精度。该方法适合于噪声比较严重的干涉条纹处理分析系统。

参考文献:

[1] Malgorzata K,Wolfgang O.Fringe pattern analysis methods:up-to-date review[C]//Rastogi P K,Gyimesi F.International Conference on Applied Optical Metrology,SPIE,1998,3407:56-66.

[2] Ge Zong-tao,Tkeda Mitsuo.A high precision 2-D angle measurement interferometer[C]//Wolfgang O,Interferometry XI:Applications,SPIE,2002,4778:277-287.

[3] 戴福隆,王朝阳.条纹图像的数字化自动分析处理技术之一:条纹中心法[J].光子学报,1999,28(8):700-706.

[4] 张伟,刘剑峰,龙夫年,等,基于 Zernike 多项式进行波面拟合研究[J].光学技术,2005,31(5):675-678.

[5] 章毓晋.图像处理和分析[M].北京:清华大学出版社,1999:91.

[6] 李自勤,王骐,李琦,等.激光成像系统图像散斑抑制算法比较[J].红外与激光工程,2003,32(2):130-133.

[7] 杨利红,施浣芳,陈智利,等.基于 CCD 采集的 Mach-Zehnder 干涉条纹图的处理算法[J].应用光学,2005,26(2):40-42.

[8] 陶卫,张敏,浦昭邦,等.基于多项式拟合的条纹图象处理方法[J].光电工程,2000,27(6):34-36.