

PCLAgenda: 一个基于过程的 CASE 环境*

武小鹏 黄涛 冯涛

(中国科学院软件研究所计算机科学开放研究实验室 北京 100080)

摘要 本文描述了 PCLAgenda(process controlling language agenda)的设计与实现. PCLAgenda 是 SCOP 组合框架软件工具系统中提供的以过程为核心的软件工程环境,其主要功能是定义软件开发过程,指导、控制和自动化软件开发工作,支持开发人员的合作和协调. PCLAgenda 的核心是基于 SCOP 软件开发方法学的、面向活动的软件过程模型.

关键词 软件过程,过程模型,活动,合作.

软件的开发是由具体的软件开发过程来完成的.从大量的实践中,人们认识到软件开发过程的质量是影响软件产品质量和软件生产效率的重要因素.软件过程成熟度(软件开发组织和个人对软件过程的认识与实践水平)已成为衡量一个国家软件产业水平的重要标志.^[1]

自从计算机软件产生,特别是 60 年代软件危机以来,软件开发过程一直是软件工程领域的重要研究方向之一.通过 40 年来的努力,从 1956 年 Benington 提出的九阶段 stage-wise 软件开发模型、1970 年 Royce 提出的瀑布型模型,到 1986 年 Boehm 提出的螺旋模型^[2]以及 80 年代中期以来兴起的面向对象的模型,人们已经提出了各种不同的软件开发方法模型.但是,这些软件开发方法对软件开发过程的描述是不严格和不详细的,其实施过程具有很大的随意性.开发人员需要按自己对开发过程的理解进行工作,而各类开发人员对开发过程常常有不同的理解,从而影响了相互间的交流与合作,妨碍了开发工作的顺利开展,项目管理人员很难保证实际开发过程与文档中描述的开发过程之间的一致.

为了解决上述问题,近年来,软件工程中一个新的研究领域——软件过程工程(Software Process Engineering)逐渐发展起来.软件过程工程试图通过某种形式化手段对软件开发过程加以系统严格的描述,为开发人员提供一个标准的无歧义的软件开发规范,使他们之间能更有效地进行交流.通过在计算机上执行所描述的软件过程,项目管理人员可以对实际的软件开发过程进行监控,从而保证实际的软件开发过程按规定的方式和步骤进行.这种对软件开发过程的形式化描述称为软件过程模型(Software Process Model).软件过程模型是实际软件开发过程中的开发活动、产品、资源、人员等各类信息的抽象.为基于软件过程模型

* 本文研究得到国家 863 高科技项目基金资助.作者武小鹏,1964 年生,博士生,主要研究领域为软件工程环境,过程模型,面向对象的软件方法学.黄涛,1965 年生,博士,副研究员,主要研究领域为软件工程,程序设计方法学,对象语义理论.冯涛,1974 年生,硕士生,主要研究领域为软件工程环境,面向对象技术.

本文通讯联系人:黄涛,北京 100080,中国科学院软件研究所计算机科学开放研究实验室

本文 1996-10-22 收到修改稿

的软件开发工作提供自动支持的软件工程环境就称为以过程为核心的软件工程环境——PSEE(process-centered software engineering environment).

本文所描述的 PCLAgenda(process controlling language agenda)是 SCOP 组合框架软件工具系统^[3]中提供的以过程为核心的软件工程环境,其核心是基于 SCOP 软件开发方法学的^[4]、用户可配置的软件过程模型. SCOP 是基于组合软件工程模型实现的框架软件工具系统,其基本含义是:

$$\text{Software Construction} = \text{Objects} + \text{Process Control}$$

即软件是对象组件的复合,而软件设计则是对象组件经进程控制的组合构造.

PCLAgenda 的主要功能是:

(1)软件过程定义:精确严格地定义软件开发过程.

(2)指导和控制:根据已定义的软件开发过程,提供用户工作的导航支持,对开发人员的工作进行指导和控制.

(3)软件过程自动化:在软件开发过程中,有很多工作是可以不需要人的参与的. PCLAgenda 可以根据软件开发过程的定义自动完成这些工作.

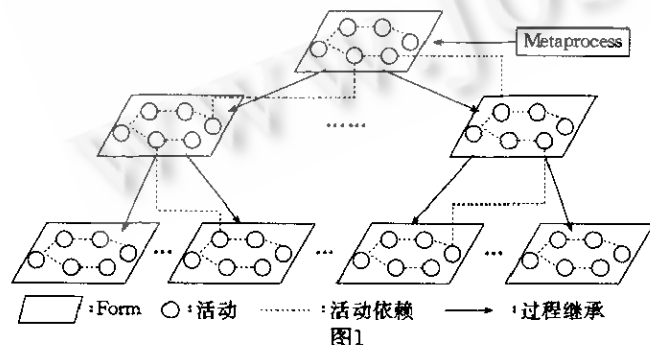
(4)合作和协调:软件开发是一项涉及很多开发人员的长时间的工作. PCLAgenda 自动将每一开发人员的工作进展情况通知给开发组的其他人员,从而协调各开发人员之间的合作.

本文首先介绍 PCLAgenda 的过程模型及系统总体结构,然后就 2 个有代表性的同类系统——MARVEL 和 MELMAC 与 PCLAgenda 进行简单的比较讨论,最后是进一步的工作.

1· PCLAgenda 过程模型

1.1 概述

PCLAgenda 的软件过程模型是基于 SCOP 软件开发方法学的、面向活动的过程模型. 其总体结构如图 1 所示. Form、活动和角色是 PCL 过程模型的 3 个基本构件.



1.1.1 Form

依照 SCOP 软件开发方法,软件项目的开发过程是其软件构件的开发与组装过程. 而软件构件的开发过程又是其子构件的开发与组装过程. 这样,如果我们把软件项目本身也看作构件,则整个软件项目的开发过程就是各构件的开发过程的集合.

在 PCLAgenda 中,过程模型是软件项目实际开发过程的抽象,Form 则是软件构件的实际开发过程的抽象. Form 是 PCLAgenda 过程(即 PCLAgenda 软件过程模型描述的软件开发过程)的子过程. PCLAgenda 过程就是一棵 Form 树.

1.1.2 活动(Activity)

PCLAgenda 过程模型由一组互相关联的活动组成. 活动是软件开发过程中具体开发活动的抽象. 每个活动有其具体目标, 完成一项特定的工作, 创造相应的产品. 这些活动由特定的角色(或计算机), 通过调用相应的工具或命令, 由用户交互或系统自动完成.

在 PCLAgenda 软件过程生命周期中活动有活动定义和活动实施 2 个阶段. 活动定义是指使用 PCLAgenda 软件过程建模语言定义活动的各个要素. 活动的实施则指通过调用活动中各种预定的工具和命令, 由系统自动或由用户交互完成活动预定任务的过程.

活动是相互关联, 部分有序的. 一个活动是否可以实施由其它相关活动的完成情况决定, 而活动的实施又会影响其它相关的活动. 活动的相关性不仅体现在同 Form 活动的互相关联, 更体现在不同 Form 的活动之间的互相关联上.

Form 和活动反映了 PCLAgenda 过程模型在不同方向上的分解.

1.1.3 角色(Role)

角色体现了软件开发过程中的一种责任和特权. 在 PCLAgenda 过程模型中角色是开发活动与开发人员联系的媒介. 它们之间的关系表示如图 2 所示.



图2

1.1.4 Form 的继承性及元过程(Meta-Process)

在 PCLAgenda 过程模型中, 子 Form 的过程定义继承其父 Form 的过程定义, 同时对继承的过程定义进行实例化, 即对继承的过程定义进行修改, 以便适应该 Form 的具体需求. 因此, Form 树同时也是过程继承树.

元过程是 Form 软件过程定义的模板. Form 树中顶层 Form 的过程定义继承元过程的过程定义. 因此元过程决定了整个 PCL 软件过程的基本结构.

1.2 过程定义

PCLAgenda 过程定义是指使用 PCLAgenda 的过程建模语言描述软件开发过程. 过程定义由活动定义和角色定义 2 部分组成. PCLAgenda 过程建模语言有外部表示和内部表示 2 种形式. 外部表示是一种表格驱动的语言, 用户通过填写这些表格来完成过程定义; 而内部表示(即 PCL 程序)是一种类 C 的过程控制语言. 用户通过外部表示定义的 PCLAgenda 过程存储在一个内部数据结构中, 过程实施时, 系统自动将其翻译成 PCL 程序, 并进行解释执行.

1.2.1 活动

活动的描述、前后条件、操作、输入和产出、实施类型、完成状态以及分配的角色是组成活动的 7 要素.

(1) 完成状态(Finishing State): 该活动完成预定任务的状态. PCLAgenda 定义了活动的 3 种完成状态: Incomplete: 活动预定的工作尚未完成; Successful: 活动预定的工作已成功完成; Unsuccessful: 活动预定的工作已完成, 但不成功.

(2) 实施类型(Enactment Type): 自动执行(Automatic)或交互执行(Interactive). 在 PCLAgenda 过程模型中, 每个活动有一个活动标志, 只有活动标志为真的活动可以实施. 活动标志为假的活动则处于等待状态. 活动标志由该活动的前条件决定. 自动活动在活动标志

为真时,系统自动完成该活动的操作.交互活动在其活动标志为真时,系统提示用户该活动已可以实施,由用户交互完成其具体操作.

(3)前条件(Precondition)和后条件(Postcondition)

活动的前条件和后条件反映了活动之间的关联性.前条件是活动是否可以实施的条件,只有前条件为真的活动可以实施.活动的前条件是由有关活动(包括该活动本身)的状态组成的一条逻辑表达式.后条件是该活动完成(包括成功完成和不成功完成)后,有关活动(包括该活动本身)应满足的完成状态及其应执行的有关操作.活动的后条件是一段 PCL 程序.

(4)描述(Description):指明该活动应完成的工作内容以及如何完成这些工作.

(5)操作(Operation):活动实施时应执行的操作及相应的工具或 UNIX 命令.例如,编码活动的操作可包括编辑、显示和打印等,其对应的 UNIX 命令是 vi, cat 和 lp.分析活动的操作应包括 OOA 分析,其对应的工具是 OOAnalysis.

(6)输入(Import)和产出(Product):活动的输入和产出分别指活动实施时的输入文件和输出文件.活动的输入必是相关活动的产出,而活动的产出又可作为另一活动的输入.

(7)角色(Role):授权实施本活动的角色.在 PCLAgenda 中角色是由人来担任的.因此,活动的角色实际上决定了由谁负责该活动的实施.自动活动是由系统自动实施的,因此不必为其指定角色.

1.2.2 角色

PCLAgenda 中的角色由系统缺省的定义和用户定义的角色组成.承担者和责任是角色的两大要素.承担者是担任该角色的人,责任是分配由该角色来实施的活动.系统缺省定义的角色是项目管理员(Project Owner)和过程管理员(Process Manager).用户定义的角色是由项目管理员根据项目的具体需要确定的,如分析员、程序员等.

项目管理员在整个工程项目中是唯一的,其承担者和责任是系统定义的.PCLAgenda 规定创建该工程的人自动担任项目管理员,其责任是过程定义和过程监控.

过程管理员的责任是系统定义的.PCLAgenda 规定过程管理员的责任是所属 Form 的过程实例化.过程管理员的承担者的确定方式如下:

- 项目管理员的承担者自动成为顶层 Form 过程管理员的承担者.
- 父 Form 过程管理员的承担者自动成为子 Form 过程管理员的承担者.
- 每个 Form 的过程实例化可修改该 Form 过程管理员的承担者.

用户定义的角色和责任和承担者是在所属 Form 的过程实例化时由过程管理员指定.

1.3 过程实施

PCLAgenda 的过程实施就是通过 PCL 语言解释器解释执行已定义的 PCLAgenda 过程,按预定的方法和顺序完成各项开发工作的过程.在 PCLAgenda 过程模型中,软件过程是一棵 Form 树,PCLAgenda 过程的实施也就是 Form 树动态递规地产生与发展的过程.

1.3.1 Form 树的产生

(1)顶层 Form 的产生:用户使用 PCLAgenda 开发软件项目时,应首先创建一个工程,并定义该工程的元过程,同时系统自动产生顶层 Form.顶层 Form 继承该工程的元过程定义.

(2)子 Form 的产生:如果一个 Form 所要开发的软件构件比较复杂,既不能直接编码,

也不能在构件库中找到,那么用户应在该 Form 的软件过程中定义一个 OOA 分析活动,以便调用 OOA 工具进行 OOA 分析.进而确定需开发的子构件.系统自动根据 OOA 分析的结果产生子 Form. OOA 分析文本^[5]中的每一个类对应一个子 Form.子 Form 继承父 Form 的过程定义.

(3)随着 Form 树中各 Form 的过程实施,子 Form 不断产生,直到 Form 树中所有的叶结点 Form 对应的软件构件均可以直接编码或可以在构件库中找到为止.由此可以看出,Form 树是随着软件开发过程的进展不断发展变化的.

(4)每个 Form 产生后,先进入过程实例化状态,由过程管理员配置满足该 Form 具体要求的软件过程.过程实施中,如果用户希望对过程进行某些修改,则可再进入过程实例化状态.由于对过程的修改可影响到该 Form 的所有子 Form,因此这时该 Form 的所有子 Form 也进入过程实例化状态.

1.3.2 过程实施

过程实施时,PCL 语言解释器解释执行已定义的 PCLAgenda 过程,根据活动的前条件确定活动是否可以实施.

对于自动活动,当其活动状态为真时,系统自动执行(调用)操作中定义的 UNIX 命令或工具,确定该活动的产出文件,并根据操作的结果设置活动的完成状态.

对于交互活动,如其活动状态为真并且实施该活动的角色由当前用户担任,则此活动出现在当前用户的工作活动列表中.工作活动列表中的活动即表示了用户在当前状态下应完成的工作.用户在活动列表中选择一项活动,则该活动应完成的工作描述、可调用的工具及输入文件自动出现在 PCLAgenda 用户界面上.用户可选择调用这些工具完成预定的任务,并设置输出文件.当用户希望退出该活动的实施时,系统打开对话框,由用户设定活动的完成状态.如果用户只是想暂时退出,活动的预定工作内容未完成,选 Incomplete,否则根据预定活动的完成情况选 Successful 或 Unsuccessful,这时系统自动发 Mail 通知有关的用户.

1.4 PCLAgenda 过程生命周期

PCLAgenda 过程的生命周期由过程定义、Form 产生、过程实例化和过程实施 4 个阶段组成(如图 3 所示).PCLAgenda 过程是一个循环和递归的过程.在过程实施阶段,可以对过程(包括元过程和实例化后的过程)进行动态修改,从而构成一个循环过程.另一方面,过程实施中可产生新的 Form,新的 Form 又要进行过程实例化,进而过程实施,从而构成一个递归过程.

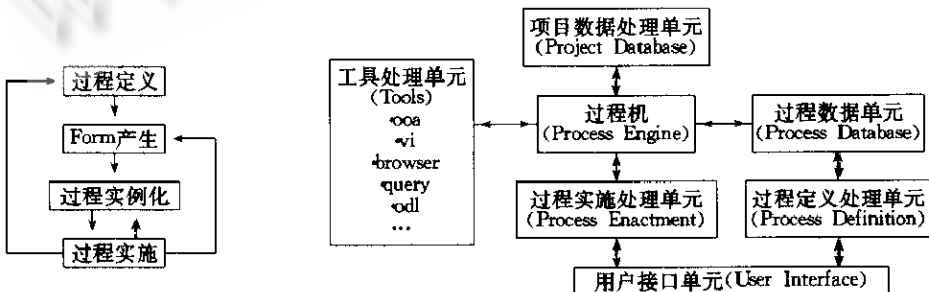


图3

图4

2 PCLAgenda 总体结构

PCLAgenda 的总体结构如图 4 所示.

(1) 用户接口单元(User Interface)是 PCLAgenda 的输入输出界面,是用户与 PCLAgenda 进行交互的媒介.

(2) 过程机(Process Engine)是 PCL 语言解释器,用于解释执行 PCLAgenda 过程模型内部表示(即 PCL 程序).

(3) 过程定义处理单元(Process Definition)的主要功能是将通过用户接口单元定义的 PCLAgenda 过程模型外部表示转换成内部表示,将其存储在过程数据处理单元中,并进行一致性和正确性检查.

(4) 过程实施处理单元(Process Enactment)用于 PCLAgenda 的过程实施.其主要功能是组织活动实施时的各种数据,对于自动活动完成该活动的具体操作,而对于交互活动则将活动有关的信息送往用户接口单元.

(5) 过程数据处理单元(Process Database)用于存储和维护 PCLAgenda 过程定义以及过程实施时产生的各种数据.

(6) 项目数据处理单元(Project Database)用于存储和维护所开发软件项目涉及的各种数据,包括各类文档、源代码等.以上 2 个数据处理单元均是基于 UNIX 文件系统实现的.

(7) 工具处理单元(Tools)用于调用各种 SCOP 工具和 UNIX 工具(命令).

3 同类工作比较

从发表各类文献^[6,7]中看出,基于过程的软件工程环境是一个非常活跃的研究领域,已有一些实验性的和个别商品化的系统问世.从所采用的过程模型来看,这些系统可大致分为:

- 基于规则(Rule-based)的系统,如 MARVEL,Merlin.
- 基于 Petri 网(Petri-net-based)的系统,如 MELMAC,SPADE.
- 基于过程编程(Process programming)的系统,如 ARCADIA.
- 基于多种形式化方法(Hybrid)的系统,如 EPOS.

下面就其中有代表性的 2 个系统与本文的工作进行简单的比较讨论.

MARVEL^[8]是 Columbia 大学开发的,以基于规则的过程模型为核心的软件工程环境. MARVEL 提供基于规则的过程模型语言 MSL. MSL 规则类似于 PCLAgenda 的活动,一条规则描述一项具体的软件开发任务. MSL 规则由名称(Name)、参数(Parameters)、动作(Action)、条件(Condition)和效果(Effects)组成,分别类似于 PCLAgenda 活动的名称、输入/产出、操作、前条件和后条件. MARVEL 通过称为 opportunistic processing 的机制,利用规则的向前链和向后链来驱动过程的实施. PCLAgenda 的过程实施类似 MARVEL 的向前链机制. 与 MARVEL 主要强调过程的自动化不同, PCLAgenda 区分交互和自动活动,对它们提供不同的实施方式,从而明确地体现了软件过程的自动化及指导和控制 2 个不同的侧面.

MELMAC^[9]是 Dortmund 大学开发的、以基于 Petri 网的过程模型为核心的软件环境。MELMAC 的过程模型是一种基于扩充的谓词/变迁网(Predicate/Transition)的高级 Petri 网系统——FUNSOFT 网。与 PCLAgenda 类似, MELMAC 过程模型提供软件过程的内外 2 种表示形式。外部表示主要由对象和活动、过程、反馈以及项目管理等视图(View)组成。对象和活动视图描述软件开发过程中产生的各类数据、文档及开发活动,过程视图描述开发活动的调度关系,反馈视图描述开发活动的修改及精化,项目管理视图描述软件项目的组织管理信息。外部表示的各视图被自动集成为一个统一的由 FUNSOFT 描述的内部表示。MELMAC 通过对内部表示的解释执行驱动过程的实施。与 PCLAgenda 类似, MELMAC 也对自动活动和交互活动提供不同的实施方式。此外, MELMAC 还提供了基于 FUNSOFT 网的过程分析手段,以确定所描述软件过程中可能存在的缺陷,从而保证其正确性和合理性。虽然 MELMAC 通过预先设置修改点的方式,支持增量方式(Incremental)的软件过程建模,但在支持过程的动态修改方面 PCLAgenda 更加灵活。

4 结束语

本文简要地描述了 PCLAgenda 的设计与实现。PCLAgenda 是以过程为核心的软件环境,其核心是基于 SCOP 软件开发方法学的、面向活动的过程模型。系统使用 C++ 语言和 Motif 图形界面开发工具,在 SUN 工作站上实现,源代码约 25 000 行。

目前的 PCLAgenda 还是一个实验性的系统。进一步的工作主要是:

- (1) 通过在各种不同类型和规模的软件开发项目中的实际使用,改进系统的实用性。
- (2) 改进软件过程模型,特别是增强数据模型描述能力。
- (3) 改善过程实施环境(Process Enactment Environment),提供过程的可视化支持。

参考文献

- 1 Humphery W S, Kitson D H, Gale J. A comparison of U. S. and Japanese software process maturity. In: Proc. 13th Conference on Software Engineering, 1991.
- 2 Boehm B W. A spiral model of software development and enhancement. IEEE Computer, 1988, (5): 61~72.
- 3 冯玉琳, 黄涛, 武小鹏. 面向对象的组合软件工程研究. 计算机学报, 1996, 19(3): 237~241.
- 4 黄涛, 武小鹏. SCOP 软件开发方法学. 中国科学院软件研究所计算机科学开放研究实验室技术报告, 1995.
- 5 倪彬. OOAnalysis 的设计与实现. 中国科学院软件研究所计算机科学开放研究实验室技术报告, 1995.
- 6 Curtis B, Kellner M, Over J. Process modeling. Communication of the ACM, 1992, 35(9): 75~90.
- 7 Fuggetta A, Ghezzi C. State of the art and open issues in process-centered software engineering environments. J. System Software, 1994, 26: 53~60.
- 8 Barghouti N S. Supporting cooperation in the MARVEL process-centered SDE. ACM SIGSOFT, 1992, 17(5): 21~31.
- 9 Deiters W, Gruhn V. The FUNSOFT net approach to software process management. International Journal of Software Engineering and Knowledge Engineering, 1994, 4(2): 229~256.

PCLAgenda: A PROCESS-CENTERED CASE ENVIRONMENT

Wu Xiaopeng Huang Tao Feng Tao

(Laboratory of Computer Science Institute of Software The Chinese Academy of Sciences Beijing 100080)

Abstract This paper outlines the design and implementation of the PCLAgenda. PCLAgenda is a process-centered software engineering environment provided by SCOP system. Its functionality includes defining the software development process, guiding, controlling and automating the steps of software development, and supporting the cooperative work among individuals and teams. The kernel of PCLAgenda is the activity-centered process model based on SCOP software development methodology.

Key words Software process, process model, activity, cooperative work.

中国自动化学会第 12 届青年学术年会(CYA'97)

征 文 通 知

为促进我国自动化领域及相关学科青年科技工作者的迅速成长,推动青年科技工作者的学术交流,拟定于 1997 年 8 月中旬在浙江杭州召开中国自动化学会第 12 届青年学术年会。本次年会由中国自动化学会主办,由浙江工业大学信息工程学院和浙江大学工业控制技术研究所联合承办,浙江省自动化学会协办。大会期间将邀请国内有关著名专家、学者作综述或专题报告;组织专题讨论;举办高新技术产品展示会。同时,由于明年是香港回归年,因此,年会还将邀请部分港、台青年学者和企业家参加会议。

现将有关事宜通知如下:

一、征文范围

- | | | |
|--------------------------|--------------------|------------------------|
| (1)线性与非线性系统控制 | (7)神经网络及其应用 | 仿真 |
| (2)自适应控制和预测控制 | (8)自动化仪表与过程控制 | (13)软件体系结构与开发方法 |
| (3) H_{∞} 控制和鲁棒控制 | (9)软件工程、并行处理 | (14)离散事件动态系统(FMS,CIMS) |
| (4)智能控制、模糊控制 | (10)人工智能与专家系统 | (15)系统工程理论、方法及其应用 |
| (5)系统辨识与建模 | (11)图象处理与模式识别 | (16)工业工程与生产管理 |
| (6)故障诊断与容错控制 | (12)信息管理、决策支持系统及系统 | (17)其它 |

二、征文要求

- (1)论文应内容充实具体,具有学术、推广价值,未在国内外公开发行的刊物或全国性学术会议上发表或宣读;
- (2)论文全文一般不超过 5000 字;
- (3)论文第一作者的年龄不超过 40 岁。

三、论文截稿时间:1997 年 3 月 1 日

1997 年 4 月 15 日前向作者发出论文录用通知

四、投稿地址

310014 杭州市 浙江工业大学信息工程学院 CYA'97 组委会
联系电话:(0571)8320365,(0571)8320200 联系人:吕丽民、梁丰

五、有关说明

- (1)投稿时请注明文章所属的研究方向(见上述征文范围)。
- (2)论文录用后收入会议论文集,会议论文集由出版社正式出版。部分优秀稿件将在全国核心期刊刊出。
- (3)请作者自留论文底稿,本会不退稿。