

软件过程环境的设计与实现^{*}

郭江 黄涛 廖越虹

(中国科学院软件研究所 北京 100080)

摘要 本文主要讨论了软件过程环境 ISPE(integrated software process environment)的设计与实现. 首先, 讨论了 ISPE 的体系结构, 介绍了可视化的过程建模语言 VPML(visual process modeling language). 然后, 在这些内容的基础上, 详细讨论了 ISPE 的几个重要的组成部分: 过程编辑环境、过程分析和模拟环境以及过程运作环境. 最后, 本文给出了一个小结.

关键词 软件过程, 软件过程环境, 过程建模, 过程分析模拟, 过程运作.

中图法分类号 TP311

软件技术同计算机硬件一样发展了几十年, 但软件生产的水平还远远不能满足社会的需求. 从 80 年代开始, 人们逐渐认识到瀑布模型的不足. 为了开发出更可靠、执行效率更高、功能更完善的软件, 就需要软件工程师研究开发出新的软件开发模型. 此后提出了其它开发模型, 如快速抛弃原型法、增量式开发法、自动软件综合法、空间螺旋式开发模型以及面向对象的开发技术. 它们都从不同的侧重点出发, 试图避免瀑布模型的缺陷, 描述适合软件开发规律的模式.

但是困扰人们的“软件危机”并没有得到很好解决, 大规模软件生产仍然很困难. 于是, 从 80 年代中期起, 学术界和工业界开始注重对软件过程的研究.

尽管软件过程只是改进软件质量和组织性能的几个可控因素之一, 但是 M. Dowson 指出: “软件产品的质量在很大程度上依赖于软件过程, 尤其是大规模的软件开发更是如此”.^[1] 鉴于软件过程的重要性, 软件工程师对此进行了大量的研究工作, 不仅提交了很多报告, 还开发了许多支持软件过程的大型项目, 主要有 AP5, APPL/A, MSL/Marvel, STATE-MATE, EPOS/OOER, HFSP, MERLIN, OPIUM, Adele, BOOTSTRAP^[2] 等等.

除了上面介绍的大型研究项目以外, 随着学术界和工业界对软件开发过程越来越重视, 不断有新的过程管理环境出现. 据 David Sharon 和 Rodney Bell 在 1995 的统计^[3], 近一、二年推出的过程管理方面的环境主要包括以下几方面的功能: ①建模, 用于项目的建模并将之映射到由工具支持的任务上; ②工具协调, 用于将工具的使用协调起来以便支持过程中的各

* 本文研究得到国家自然科学基金、国家 863 高科技项目基金和国家“九五”攻关项目基金资助. 作者郭江, 1966 年生, 博士, 主要研究领域为软件工程, 数据库. 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. 廖越虹, 女, 1972 年生, 硕士, 主要研究领域为数据库.

本文通讯联系人: 郭江, 北京 100080, 中国科学院软件研究所

本文 1997-01-07 收到修改稿

个任务;③运作,用于指导和支持项目的有关人员遵循开发过程进行各项活动;④管理,用于帮助项目管理人员确定开发活动对完成过程的影响,根据实际情况做出相应的决策.这些工具的主要代表有:(1)InConcert;(2)Process Weaver;(3)SynerVision;(4)Process Engineer;(5)SE Companion;(6)Synergy 等等.在这些软件过程集成环境的基础上,我们开发了 ISPE (integrated software process environment)软件过程集成环境.

1 软件过程集成环境 ISPE 的目标

集成的过程工程环境是一个用于提高软件开发机构能力的集成环境,根据总的质量管理和连续不断的过程改进原则,软件开发机构必须首先定义用于进行关键过程的步骤,然后对这些步骤进行分析和控制,以便找出适当的方法来将工作做得更好、更快、更经济.

过程工程主要在 3 个阶段——过程理解和定义、过程分析和模拟、过程交付和实现——需要技术方面的支持,在每个阶段均有不同的要求.此外,还需要给软件过程工程师提供一个可视化的图示方法,以便帮助用户用一种明确和无二义的方式来表示过程模型,这样就可以描述和交流过程的有关信息.集成的过程工程环境及其工具不仅要满足所有这些需求,而且还要支持基本方法学和前端到后端的一致性.

在这里,过程建模的基础是可视化过程建模语言 VPML (visual process modeling language),其中关键的过程元素都是用图形来表示的.这就提供了一个集成的过程工程环境及其工具的构造技术基础.建立在 VPML 上的各种工具构成了过程工程环境的工具家族,其中包括过程编辑环境、过程分析模拟环境和过程运作环境,如图 1 所示.其它支持工具包括活动与工具的集成封装器等,活动与工具的集成封装器用于将活动和工具集成在一起,以便构成一个整体,这样在用户完成活动时就可随时调用相应的工具.

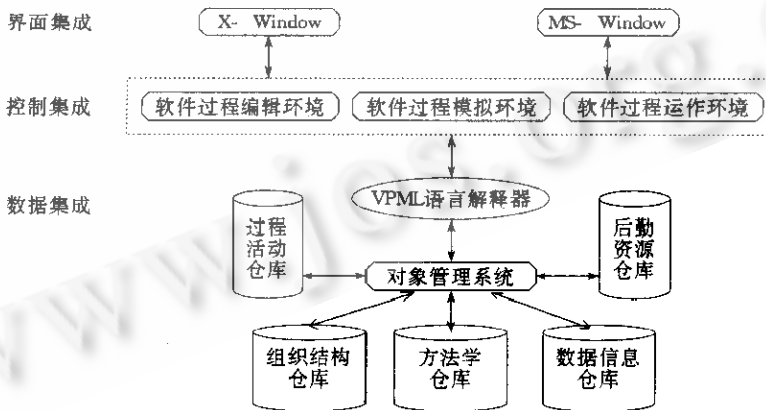


图1 ISPE的集成体系结构

2 软件过程环境 ISPE 的总体结构

过程工程环境是一个集成的过程开发与运作环境,集成不是一个工具的特性,它描述了工具与环境其它元素的关系.这些元素包括工具、平台和过程,而工具之间的关系以及这些关系的特性则是集成的关键.工具的集成是工具之间应有的一致性范围,包括数据格式、

用户界面、公共函数的使用以及工具结构等. 过程工程环境的系统总体结构主要体现了界面集成、数据集成和控制集成等特征, 如图 1 所示.

(1) 界面集成

界面集成的目标是使环境尽量给用户一个相同或相近的交互方式, 以便增加所有工具的共同特征. 这样就能给用户一个友好和熟悉的界面, 是提高生产率的一种手段. 在过程环境中, 我们采用了两种标准界面, 在 PC 上使用 MS-Windows, 在 SUN 工作站上使用 X-Window (Xview 和 Motif).

(2) 数据集成

数据集成是为工具所使用的信息提供共享, 包含有持久性和临时性信息. 目的是不管工具是使用还是交换部分信息都能保持信息的一致性和完整性. 因而要实现下面几个目标:

①交互运行能力: 即工具之间相互使用对方数据的能力. 在 ISPE 系统中我们采用了 VPML 对象作为统一的解释器, 所有工具均由 VPML 解释器来识别对象, 这样由一个工具生成的数据能为另一个工具所使用.

②数据一致性: 即工具对数据所进行的操作要使得环境中其它部分作相应处理, 从而维护整个环境的统一. 在 ISPE 系统中我们采用了对象管理系统 (OMS) 作为我们的数据管理系统, 所有数据的存取均通过 OMS 管理的动态库来完成, 这样就能较容易地保持数据的一致性了.

③数据的同步性: 即一个工具对公共数据的修改应通知其它与之合作的工具. 这与一致性类似, 但要用到非永久性数据上. 在 ISPE 运作环境中, 由于采用了客户/服务器体系结构和动态数据库的概念, 因而数据更改是由服务器来通知各个工具的.

(3) 控制集成

控制集成的目的是进行工具共享, 即某个工具应能使用其它工具所提供的功能. 在 ISPE 系统中, 我们在软件过程运作环境中就调用了软件过程编辑环境, 其中由于工具间传输的数据量较大, 因而使用了动态数据库, 工具间的控制采用了远程进程执行这种方式.

(4) 相关的工具集

过程工程环境中的工具集包含了从过程模型的定义、过程的静态分析和动态模拟到过程的执行和监控的主要支持工具. 主要有: 过程编辑环境、过程分析模拟环境以及过程运作环境.

3 软件过程管理模型 M-POID

一个软件开发机构的模型描述了如下 5 个方面: 方法学、过程活动、组织结构、后勤资源和数据信息. 这样就可以给管理人员、软件开发人员等提供必要的信息以便进行有效的管理. 因此, 软件开发机构模型可用下面的五元组来表示:

$$EM = (MM_{a=1,2,\dots,i}, PM_{b=1,2,\dots,j}, OM_{c=1,2,\dots,k}, IM_{d=1,2,\dots,l}, DM_{e=1,2,\dots,m})$$

其中 EM 表示软件开发机构模型, MM 表示方法学模型, PM 表示过程活动模型, OM 表示组织结构模型, IM 表示后勤资源模型, DM 表示数据信息模型. a, b, c, d, e 和 i, j, k, l, m 是有限整数, 用来表示一个软件开发机构中可以有多个模型的个数. 这就是软件开发机构的

M-POID 软件过程管理模型.

方法学模型是为了建立对软件开发机构和过程进行规划、进行决策的出发点和指导思想. 为了达到软件开发机构和项目的长期目标和短期目标, 就需要灵活的战略和战术方法学来指导规划和决策, 因此方法学模型是软件开发机构和过程的一个重要的组成部分.

过程活动模型用于描述软件的开发过程, 其基本部件是活动、产品(用作活动的输入/输出)、角色(负责完成活动的人员类型)、它们之间的关系(包括活动间的时序关系). 此外, 还可以给活动分配其它资源类型, 包括工具类型、机器类型和地点类型等. 过程活动模型用活动、从事活动的人员所扮演的角色、由活动所产生的输入输出产品、工具的类型、机器以及使用的地点来定义软件的开发过程.

组织结构模型用于描述软件开发过程中各种人员之间的相互关系. 因此它首先要确定软件开发中所需要的角色. 组织结构模型提供了一种角色视图. 一个项目由角色、问题(域分析)及其之间的沟通渠道、责任和依赖性组成. 注意某些角色对开发过程是通用的, 而且可以根据项目的具体情况分为更细的角色, 不仅如此, 某些人员依赖于项目的具体情况可担任多个角色.

后勤资源模型是用于描述构成软件开发机构的资源类型和资源实例的, 它是为软件开发机构或项目定义的, 这里所指的软件开发机构既可以是一个公司, 也可以是一个公司的部门, 甚至还可以是一个已拥有分配了资源的开发小组, 在一个软件开发机构中, 所有过程均要求有软件开发机构后勤(包括各种资源和资源类型)的支持.

数据信息模型主要用于定义在软件开发过程中所设计的文档、术语、规范以及标准等等. 数据信息是软件开发机构和过程的重要资源. 为了达到项目的目标, 就需要收集、保存、管理和利用这些信息, 它们是软件开发机构和过程的一个重要的组成部分.

4 可视化的过程建模语言 VPML 及其编辑环境

过程编辑环境是面向结构化的图形编辑工具, 它使用可视化的建模语言 VPML 来刻画和描述过程管理的 M-POID 模型(它包含 5 个相互关联的子模型: 方法学模型、过程活动模型、组织结构模型、后勤资源模型和数据信息模型). 它能帮助用户迅速而正确地构造软件开发机构的 VPML 过程模型(即过程图), 并使用 VPML 规则来连接模型中的各个部件. 过程编辑环境使用基于图形的接口来编辑 M-POID 中的各个子模型(过程图), 并使用基于表格的接口来编辑 VPML 的结构属性, 下面简要介绍 VPML 的主要结构^[4], 如表 1 所示.

过程编辑环境是使用可视化建模语言 VPML 来描述 M-POID 软件过程管理模型^[2]的一个结构化的图形编辑环境. 主要用于描述 M-POID 的 5 个子模型. 过程编辑环境帮助用户迅速产生和编辑 VPML 过程图, 使用 VPML 过程规则来连接各个组成部分. 它用一个基于图形的接口来编辑过程图, 并用一个基于表格的接口来编辑过程图中对象的属性. 这样过程编辑环境的用户就可以定义软件开发机构的 M-POID 软件过程管理模型. 形成一个过程模型, 即过程图.

过程活动模型用于描述软件的开发过程, 其基本部件是活动、产品、角色、它们之间的关系. 在过程活动模型中除了定义软件的开发过程, 还要定义活动的属性, 主要有: ①名称; ②描述; ③优先级; ④计划完成活动所需的时间; ⑤计划完成活动所需的工作量; ⑥计划完成活

动所需的费用;⑦计划执行活动的开始时间;⑧计划执行活动的完成时间等.过程活动模型中各个基本元素间的关系如图2所示.

表1 VPML的主要结构

活动	Activity 人为活动	Composite Activity 复合活动	Auto Activity 自动活动		
产品	工作产品	消息	文档	产品集	复合产品
资源	角色	机器	工具	地点	
里程碑	Milestone	计时器			
各种关系	用各种连线(虚线、点线、实线等)表示时序关系、成员关系、实例关系、关联关系、数据流等.				

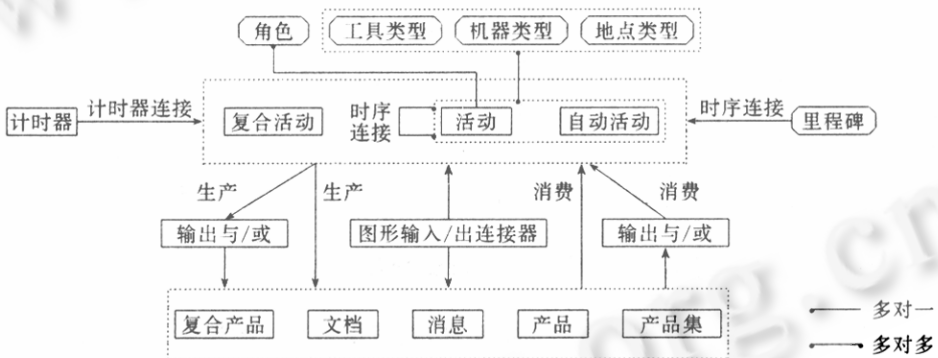


图2 过程模型中各基本元素间的关系

软件开发机构的后勤资源模型是由人员、他们所扮演的角色、机器类型(如工作站)、机器(如 Sun Sparcstation10)、工具类型(如桌面排版系统)、工具(如 FrameMaker)、地点类型(如会议室)、地点(如三楼会议室)等组成.在后勤资源模型中,还要定义机器、地点、工具和人等各自的属性.这些属性描述了资源的特征和性质,是联系过程建模和过程执行的纽带,即在后勤资源模型中定义的这些属性要在过程的执行中表现出来.在后勤资源模型中定义的资源属性主要有:①名称;②描述;③每周可以使用的时间;④每小时的成本;⑤工作效率等.

数据信息模型主要用以描述如下几方面的内容:①软件开发机构的战略和战术信息;②后勤资源信息,包括人力资源、机器设备、地点资源及工具资源的信息;③各种工作产品信息,包括输入/输出产品的信息;④组织结构模型产品信息;⑤过程活动模型产品信息;⑥应用领域的知识和有关术语;⑦特定版本的项目产品和项目需求的说明;⑧应用领域的模块和分解信息;⑨测试用例和现场数据库(从原型使用和实际使用中收集来的数据).

5 软件过程的分析模拟环境

过程分析的方法是使用静态分析和动态模拟来进行过程的费用、进度和资源风险的量化权衡分析,并提供过程的可视化分析报告,这种可视化和交互调试方法提供了对过程关键部分的理解能力。

其中静态分析工具产生成本和资源使用报告,这些均以过程定义中提供的时间期限和费用数据为基础。过程动态模拟允许用户使用静态分析的数据来激活(运行)过程,用户可以一次模拟一个任务,并增加一些随机因素。这样用户就可得到动态分析结果,如过程调度和成本的有关数据。静态分析提供了基于计划数据的进度和费用信息的图形显示,静态分析假设每个活动仅执行一次,而且不考虑产生调度冲突的可能性,这在动态模拟中考虑。

在动态模拟运行之前,过程必须通过过程编辑环境中的完整性检查,这意味着所有资源类型均分配给了资源,而且过程图中也没有语法错误。为了进行动态分析,就要模拟,因此用户不仅要给过程模型分配资源,而且还要给单个活动分配预计的费用和完成时间以及数据流分支的可能性。此外,用户还可以定义活动和里程碑计划的开始和完成时间。

模拟分析提供了有关的过程成本、完成和松弛时间(即人员等待工作的时间)等信息,如果输入了计划的开始和完成时间,过程分析模拟环境就能提供有关这些计划进度是否合理的信息。这些模拟描述了怎样使用循环来运行多次迭代过程。过程模拟生成了有关的性能数据,并给用户提供了项目在模拟过程中进行操作的可视化显示。

5.1 静态分析

静态分析提供了基于计划数据的进度和开销信息的图形表示,静态分析假定每个活动只执行一次,进度数据以图形方式来显示,而资源数据则以饼图和表的形式显示。

(1) 进度分析报告

① 软件开发机构:这里包括软件开发机构内各项目的时间表以及项目中所有活动的的时间表,该时间表反映了项目的计划开始时间和完成时间。

② 项目:这里要进行的是进度一致性检查,它是自动完成的。如果一个人为活动的最早开始日期与过程说明表格中定义的项目开始日期不同,就要给出一个警告信息。而如果一个人为活动的最迟结束日期与过程说明表格中定义的不同,也要给出警告信息。

③ 人员:显示每个被分配了活动的开发人员的时间表。

(2) 资源分析报告

① 软件开发机构:显示有关软件开发机构资源信息,包括可以满足每个角色的人员数目(含有那些可充当多个角色的人员)。

② 项目:显示与项目有关的角色信息,包括一个项目中需要的多个角色。

(3) 费用分析报告

① 软件开发机构:显示软件开发机构中每年各角色的费用信息。

② 角色:显示项目中各角色的费用信息。

③ 人员:显示项目中各人员的费用信息。

④ 阶段:显示人为活动、复合活动及自动活动在各个阶段的费用信息。

⑤ 阶段工作量:指人为活动、复合活动在各个阶段所需的工作量。

5.2 动态模拟

模拟在某些部分上是基于概率的,在有数据流分支存在时,概率就进入了模拟之中.在缺省情况下,给每个数据流分支赋予一定的概率,即如果数据流连接有两个分支,每个分支的概率就是 50%,这些概率用于决定在模拟时要采取的分支.随机数是由系统生成的,这些随机数决定了特定模拟的分支情况.用户可以通过修改项目数据(如项目的估计时间)在同一个随机数集上反复模拟运行来观察这些修改的影响,模拟也可以在同一个项目数据上的不同随机数情况下反复运行来观察项目随机变量的影响.

(1) 过程模拟变量

过程模拟的某些模拟变量直接对模拟的结果有影响,这些变量有:

①随机种子:项目模拟器有一个随机部分,用于负责包含产品输出连接的过程图的模拟.随机种子是在项目模拟中的随机数生成时使用的,用于决定生成时产生的随机数序列.随机数和分给产品输出或分支的概率相结合就决定了在通过产品输出连接时要执行哪个分支.通过使用相同的随机种子,变化项目变量,用户可以查看修改项目变量的控制方式的影响.通过改变随机种子而保持项目变量不变,用户就可以查看多重模拟运行的影响.

②时间单位:用于选择模拟所使用的时间单位,这样模拟的结果就可以以特定的单位来表示.这些时间单位可以是年、月、日、小时、分钟等.

③模拟速度:该度量单位用于决定模拟运行速度有多快,它影响着模拟期间用户在屏幕上所看到的速度.

(2) 过程模拟报告

在模拟期间主要产生两个报告:①进度报告,主要包括:等待人力资源的时间、完成活动的的时间、等待时序连接条件满足的时间;②费用报告,主要包括:一个人花费在一个活动上的时间、人员费用、两个人交互的次数、人员的空闲时间和每个活动执行的次数.

6 软件过程的运作环境

过程运作的目标可以在 3 个层次上考虑:软件开发机构、开发小组及具体开发人员.从软件开发机构的角度看,过程运作主要是在数据安全方面支持约束管理,并在必要时施加某种限制;从开发小组的角度看,过程运作主要是支持消息广播和变化控制,即以一种有序的方式管理共享信息的更新;从具体人员的角度看,过程运作主要是给管理人员和开发人员提供信息的可视性,并且只提供与当前工作相关的信息.这 3 个方面的一个共同点是支持软件开发机构信息的一致性,即在正确的时间给正确的人提供正确的信息.因此运作环境可以看作是一个工具,帮助生产和消耗消息,减少具体开发人员的“信息负担”.为此,支持过程运作的的环境包括:①用在软件开发机构过程中的访问控制工具;②用在开发小组过程中的更新控制、版本管理和消息广播工具;③用在开发人员过程中的可视控制和集成工具.

在过程定义完成以后,过程运作环境就可以运作过程模型了,过程运作环境对管理人员和开发人员均提供了较好的支持.运作环境的图形用户接口(GUI)有助于各种人员访问系统管理的数据,这不仅要支持访问他所需要的信息,而且还要确保去掉与他不相关的信息.^[5]由于与具体人员相关的信息是不断变化的,因此过程运作环境一方面要访问过程的描

述(即过程模型),另一方面还要自动监控过程的实际执行,并在此基础上预测用户的需求,随之动态改变给有关人员提供的信息.为了支持信息的相关性,运作环境需要给用户所提供管理的对象集视图,该视图是对象可视性和访问权限的限制方式.

在过程运作时,在过程模型中分配了任务的开发人员都负责完成相应的活动,这是由系统基于过程图来控制的.过程运作环境可以显示分配给开发人员的活动,并由他们一个接一个地完成,这样就将开发人员从下面的问题中解脱出来了:①哪些活动要执行;②哪些活动不要执行;③活动的输入和结果;④可以使用什么工具.在获得所有这些信息以后,开发人员就可以集中精力于完成特定的任务,而不必考虑下一步应该做什么.

因此,过程动作环境主要完成下面几方面的内容^[2]:

(1)活动的调度和管理

过程运作环境首先分析每个活动的依赖关系,这种分析揭示了活动间的潜在并发能力,即可用于确定可并发执行的活动和必须串行的活动.基于以上分析和过程模型中的资源信息,就可以调度活动的执行,这里可以使用数学分析来寻找优化和合理的调度.如果活动的输入/输出间不存在依赖性,而且资源也足够用来并发执行,那么就可以调度活动并发执行.如果在过程动作过程中发现结果不能令人满意,就需要确定产生这种情况的活动,以便能再次执行这些活动时取得正确的结果.通常这些活动只能在过程运作时发现,因而不能在过程中预先描述,这就要求管理人员介入.因此,执行活动的管理人员必须能随时处理这样的外部活动控制,也要负责例外情况的处理.运作环境所记录的活动执行情况不仅是管理和控制过程的基本信息,也是分析和改进过程的有用数据.

(2)工具调用

如果开始的活动是一个基本活动,可以调用工具来完成,那么完成该活动的开发人员就负责调用相应的工具来完成活动.工具调用时,可以将工具只看作一个函数,而不考虑其内部结构,这些均与工具调用无关.在工具开始执行以后,就不由开发人员控制了,而是由给定的支持环境包括人机交互来完成.在工具调用完成之后,就产生了相应的结果.

(3)人机交互

在过程运作中,人机交互随时发生,即不断要求开发人员的介入.运作环境动态地支持开发人员的需求,给他们提供活动的有关信息和相应的工具.在任务完成以后,开发人员在过程动作环境中可以发送消息通知项目组的其他有关人员,并产生相应的产品.运作环境还可提供完整的过程查询服务,开发人员不仅可以获得有关活动在整个过程中所处位置的信息,而且可以浏览活动就绪情况、截止日期、是否在关键路径上等.活动执行的管理人员要负责人机交互,以便使过程的运作具有更大的灵活性、更大的动态性.人机交互的另一种形式是在活动执行期间,工具从开发人员得到输入并将结果反馈回来.^[6]

7 小 结

本文主要介绍了软件过程集成环境 ISPE 及其主要组成部分,即可视化的过程建模语言、过程编辑环境、过程分析模拟环境以及过程运作环境.软件过程集成环境 ISPE 主要用于支持过程的 3 个阶段——建模/修订、模拟/分析、运作/监控.在第 1 个阶段,由过程编辑环境帮助用户使用可视化建模语言来描述过程模型,并给过程模型分配人力、物力以及时间

等资源. 在第 2 个阶段, 由过程分析模拟环境给用户有关过程模型的静态分析数据和动态模拟信息, 主要包括进度报告、费用报告、资源使用情况等等. 在第 3 个阶段, 由过程运作环境提供项目过程的真实运作环境, 给开发人员和管理人员提供相应的信息, 包括过程的完成情况、活动的状态、工具使用数据等等. 过程工程集成环境的基础是对象管理系统, 它负责存储和管理环境及具体项目过程有关数据. VPML 基于 MS Windows 和 X Window 的图形语言, 为用户提供了过程建模的直观表示手段.

参考文献

- 1 Dowson M. Software process themes and issues. Proc. IEEE 2nd Int. Conf., Feb. 1993.
- 2 郭江. 软件过程运作机理及其动态优化技术的研究[博士论文]. 北京航空航天大学, 1996.
- 3 David Sharon, Rodney Bell. Tools that bind; creating integrated environments. IEEE Software, March 1995.
- 4 Zhou Bosheng. VPML reference version 2.0. SEI/BUAA Tech. Report, Oct. 1994.
- 5 Scheer A. ARIS toolset; a software product is born. Information Systems, 1994.19(8).
- 6 Ambriola V *et al.* Software process enactment in oikos. In: Proceedings of the 4th ACM SIGSOFT/SIGPLAN Symposium on Practical Software Development Environments, 1990.

THE DESIGN AND IMPLEMENTATION OF INTEGRATED SOFTWARE PROCESS ENVIRONMENT

GUO Jiang HUANG Tao LIAO Yuehong

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

Abstract The paper discusses the design and implementation of ISPE (integrated software process environment). First, the paper describes the architecture of ISPE. Second, the paper introduces the VPML (visual process modeling language). Third, the paper discusses three important components of ISPE: process editing environment, process analyzing and simulating environment, and process enacting environment in detail. And finally, the paper gives out a conclusion.

Key words Software process, software process environment, process modeling, process analyzing and simulating, process enacting.

Class number TP311