

关于 PCTE 的浅析

王铁男 冯克清 孟庆昌 (北京信息工程学院)

王成耀 (北京科技大学)

摘 要

可移植通用工具环境(The Portable Common Tools Environment, PCTE)是西欧八家公司联合开发的一个大型软件工程环境项目,也是欧洲Esprit信息战略计划的一个重要组成部分,在国际上具有重要影响.PCTE是为了获得软件产品设计的一致性而建立起来的一种标准软件系统,是构造现代软件工程环境的基础.PCTE的出现是软件工程环境走向集成化、标准化的重要一步,对我国软件工程环境的建造具有一定的借鉴和参考价值.本文首次较详细地向国内软件界介绍了PCTE的内容,并从其基本机制、用户接口、分布式结构三个方面加以分析和概述,以求抛砖引玉,供国内同行进一步研究和讨论.

一、引 言

六十年代末,软件危机导致了软件工程的诞生,随之出现了许多软件工具,但这些工具往往只支持软件生存期的某些特定阶段,而且基于不同的方法学,各工具间联系很小,甚至完全独立,不能完全地支持软件开发和维护的全过程,这就很难给软件开发以较高的工程支持.尽管软件工程思想对改进软件生产过程起到了一定的推动作用,但由于软件工程技术本身的复杂性及实现这些技术所面临的实际困难,因而并没有从根本上解决这一问题,这就导致了所谓的“新软件危机”。

为了提高软件质量及开发效率,需要建立良好的软件开发环境,因而出现了IPSE(Integrated Project Support Environment)的概念,为软件界提出了下列研究课题:

- 环境的集成化技术。即如何以一种统一的结构,将各种工具、方法等集成于一个整体,形成一个开放式的环境,并为新工具的加入提供方便。

- 环境信息库技术。将软件开发过程中的各种信息存放在一个信息库中,用统一的方式进行管理,提供有效的访问途径。

- 各种工具接口的标准化,以便不同厂家或用户开发的工具能不加修改或稍加修改后可以集成到统一的环境中去。

- 人们普遍认为,研制集成化的软件工程环境是提高软件生产率的有效途径。为此,产生了PCTE(Portable Common Tools Environment)的概念。

二、PCTE概述

PCTE,即可移植的通用工具环境,是为了获得软件产品设计的一致性而建立的一

天依然存在的两个基本障碍:标准化和性能问题,并列举了一些迄今尚未探讨的若干研究领域。

参考文献(略)

[范明译自Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, March 1990, pp1—15 仲源声钟校]

套标准，是构造现代软件工程环境的基础。

PCTE 标准的定义是在1983年底由欧共体六家公司——**Bull, GEC, ICL, NIXDO-RF, OLIVETTI**与**SIEMENS**共同发起的，旨在产生公共工具接口定义。1986年9月出版了关于**PCTE**接口功能说明的1.4版，并基本达到了预定目标。

PCTE 是为现代软件工程环境的建造而设计的一个宿主结构，基于**SEE (Software Engineering Environment)**设计的**PCTE**可看成是工具与服务程序的集成体，用于一个特定项目的生存期模型或应用领域。**PCTE**是基于**Stoneman**结构建立的，**Stoneman**结构用于**ADA**程序设计支撑环境。**PCTE**结构对应于**IPSE**的核心，为**IPSE**的研制提供了必要的可移植框架，其定义的公共通用工具接口 (**PCTI**) 可作为一种工具集成的标准。此外，**PCTE** 结构提供了一个开放式环境，适合新工具的开发，允许新工具集成到**IPSE**中。

PCTE 定义的公共通用工具接口提供了一套与机器无关的综合设施，主要涉及以下几个方面：

- 基本机制：用来操纵软件开发过程中的各种实体，例如可执行的程序（如工具，测试中的程序）以及由程序操纵的各种对象（如数据、I/O设备、文档）。

- 用户接口：在基本机制之上，**PCTE** 提供了许多功能，以帮助用户与工具之间进行交互。

- 分布式：根据**PCTE** 标准，环境要在工作站的网络上实现，这就要求定义各种机制和协议以支持分布式的透明性。

三、基本机制

PCTE 的基本机制可分为五个方面：执行、通信、进程通信、对象管理系统、并发活动。

1. 执行机制

PCTE 利用执行原语定义一个运行程序（进程）与其执行环境之间的关系，如定义

一个程序的执行如何开始或终止，如何控制，如何传递参数到程序中等。

2. 通信机制

通信原语涉及程序对文件型非结构化数据（对象的内容）的存取方式，这些数据存放在对象管理系统的数据库中。通信原语对应于传统的输入输出功能，与**UNIX**类似。

3. 进程通信

PCTE 提供了进程间的通信机制，使得不同进程之间可进行信息交换。尽管进程间通信可看作是通信或执行机制的一部分，但由于它们在系统其它方面的实现上扮演了重要的角色，因而将其单独对待。

除了传统的**UNIX**管道和信号之外，**PCTE**还提供了一个扩充的消息传送设施，这些机制与**UNIX V**保持兼容，且功能更强。

4. 对象管理系统

在软件工程环境中，各种用户和程序要处理大量的实体，如传统意义上的文件、外设、管道或一个程序静态环境 (**Context**) 的描述，项目管理及进度记录等信息。在**PCTE**中，现实世界的实体统称为对象 (**object**)。

在一个中规模项目中，就会涉及到大量具有复杂关系的对象，常见的有：

- 文档和程序源代码（可能由多个模块组成）。

- 一个特定对象特定版本的历史及演化信息。

- 实施在一个特定版本模块上的测试数据集及测试结果。

对各类对象采取统一的处理方式，提供有效的存取和访问机制，是对软件工程环境的重要要求。一个自然的解决办法是建立一个对象管理系统 (**Object Management System, OMS**)，对各种对象实施有效管理，同时用户能为对象建立若干属性及相互关系。

基本的**OMS**模型是从**E-R**数据模型演化

来的，它以对象和关系作为环境信息库的基本内容。从E-R观点看，对象是可被标识的实体，由下列几方面加以描述：

- 特定的内容
- 相关属性集
- 相关关系集

关系表示对象及结构化信息间的逻辑联系或依赖关系。在某些情况下，需要引入由多个对象组成的复合对象，或建立一个对象到另一个对象的参照 (reference)。关系也具有属性，以描述其特性。

关系是存取对象的基础。在大多数OMS操作中，欲存取一个对象，需在字符串型路径名所指定的关系序列引导下，在OMS的对象空间中航行。OMS所使用的路径名，在语法上与UNIX兼容。

此外，无论是对象、关系还是属性，都有一个定义其基本性质的类型，这是“面向对象”通常所具有的。对象、关系及属性的类型定义都包含在模式定义集 (Schema Definition Set, SDS) 中。SDS是一个特殊的对象，可专用于单个用户或工具，亦可为所有工具和用户通用。

在任何时刻，一个进程有其自己的工作模式 (Working Schema)，它是由一组SDS所组成的，用来描述该进程所操作的对象和关系的特定约束。为了改变一个进程的工作环境，可动态地重建其工作模式。因此，不同的用户和工具可在不同的工作模式下工作，即使是存取同一个对象，其方式也可能不同。

需要指出，OMS的许多功能与DBMS类似，或者说，它是一个特殊的DBMS，但具有下列区别：

- OMS的目标不是用对象值进行复杂的计算，而主要是对对象内容进行操作，其细节由工具管理。

- 用户有可能根据自己的需要使用系统，通过定义新的对象和关系类型来更新工作环境，无需更高一级的授权。

5. 活动

UNIX环境的缺陷之一是缺乏数据存取同步与恢复机制，PCTE对此作了改进，它将事务概念引入软件工程环境中。通常，事务可以看作是单个操作发生的框架。事务具有下列特性：

- 原子性。一个事务要么执行完毕，要么不执行，不能部分执行而被中断。

- 串行性。对于同一数据域，不能有多个事务同时执行。

在PCTE中，事务的概念推广为活动 (Activity)。活动以系统化和标准化的方法用来支持工具的建立，它是一组相关操作发生的框架，与一组资源相联系。与PCTE活动有关的一个重要概念是工具的粒度 (granularity)，就是说，将每个工具 (程序、脚本或程序片断) 看成是具有良好定义功能的模块化成份，若干简单工具可装配成功能更强的复杂工具，新的工具又可作为其它工具的构成成份。

每个工具可看成是一个原语，从而可由工具集建立功能层次。PCTE允许活动嵌套，基本的PCTE模型具有完整性、一致性及隐式或显式的恢复与同步机制，完全满足原子性与串行性的要求。此外，PCTE还支持活动类型与对象级操作，工具编写者可根据需要采用相应的数据一致性和并发控制级别：从UNIX式的“无保护”级到完全保护的“事务级”。

四、用户接口

随着计算机使用的普及，用户接口显得愈来愈重要。特别是它的标准化问题受到了高度重视。在许多应用系统和软件工程环境的研制过程中，用户接口的设计和开发工作占据了很大的比重，不少软件产品已将用户接口的优劣作为衡量其质量的一个重要指标。因而，用户接口技术在近年来取得了较大进展，尤其是随着一些相关技术的进展，用户接口技术更是大显神通，现在计算机系统可以向用户提供丰富多彩的人机界面。

PCTE 亦不例外，它在用户接口方面作了全面细致的考虑。

PCTE的用户接口是一个多窗口系统，支持具有一个或多个彩色或黑白显示器以及诸如鼠标的指示设备等先进的单用户工作站，工具编写者可通过联系一个或多个窗口去开发多进程环境，终端用户可通过窗口间的移动去开发多工具环境，实现多种工具的交互。

1. 交互方法

用户交互的一个主要问题是不同环境下各种命令含义不一致，编辑时尤其如此。为了提供一个较一致的交互方法，用户接口对象应尽可能以同一方式处理，如设置一个标准的操作集，这种技术就是熟知的“面向对象”方法。无论是窗口、字符、字符串，还是图形元素，均作为“对象”处理，这种方法由两步组成：首先选择一个对象，然后对对象实施相应的操作或命令。

2. 与显示相关的对象

PCTE 的用户接口为工具编写者提供了建立和操纵用户接口对象的手段，它提供了四种基本显示对象类型：屏幕 (Screen)、窗口 (Window)、视口 (Viewport) 与框架 (Frame)，实现了数据操作细节与显示操作细节的分离。

(1) 屏幕 用户接口支持具有一个或多个屏幕及一组输入设备 (键盘与鼠标) 的工作站。在任一时刻，一个工具可同时关联多个屏幕，但每个对象只能局部于一个屏幕。

(2) 窗口 窗口可具有不同大小，窗口之间可重叠甚至完全覆盖，这就使得运行工具可以对屏幕空间作出最合适的选择。用户通过同系统交互，可调整、改变或移动屏幕上的窗口，工具只能在其窗口限定的空间内同用户交互。在用户接口中，通过窗口可看到所需的输出信息。具体的输出信息存贮在“框架”中，工具使用“框架”输出文本和图形信息。在窗口与框架之间设置了一个中间结构“视口”。

(3) 视口 视口是窗口中的一块矩形

区，也是框架的一个输出端口。当输出时，将框架映射到视口的矩形区，以便在屏幕上输出相应信息。工具可以构造视口的层次，子视口可与其父视口重叠，以分享父级空间。窗口中视口的大小、位置由具体的工具来决定。

(4) 框架 框架是为工具进行信息输出提供的。框架是一个数据结构，用来存贮和操纵工具所产生的数据。通过相应的视口可在屏幕上看到框架的内容。框架具有三种基本类型：位象、图形与文本。位象框架用来存贮位象，图形框架用来存贮图形元素 (直线、弧等)，文本框架有二种子类：单字形与多字形。由于工具的输出不是直接写入屏幕，而是先放在框架中，因而当窗口大小或覆盖情况发生变化以及有新的信息写入框架时，可以很容易地更新相应窗口的信息，这样，工具就无需关心端点用户对窗口的操作问题了。

除了基本的框架—视口—窗口输出机制之外，工具编写者还可利用下列对象来改进 PCTE 的用户接口：

(1) 图符 (Icon) 一个图符是一个图形，它可显示在窗口中，以表示相关工具正在进行，此时在屏幕上看不到工具的输出。图符可以从屏幕上移走，以便将相关窗口恢复到先前状态。

(2) 光标 (Cursor) 光标是一个符号，用于跟踪指示位置。光标的位置由指示设备的动作加以改变，在不同的视口范围内，可使用不同的光标图形。

(3) 字形 (Font) 字形是值 (如 ASCII 码) 的图形表示，每个值对应于唯一的图形符号。PCTE 为用户接口定义了一个标准字形集，工具编写者可以选择字形的不同显示方式，如带下划线、倒置等。

3. 与输入相关的对象

框架—视口—窗口机制实现了输出信息的动态存储，利用视口可确定来自键盘或鼠标的输入事件应送往哪个工具。用户在使用

键盘或鼠标时,可确定与哪个工具以及相应工具的哪个视口进行通信。一旦工具接收到输入,必须决定其处理方式,例如,一个文本工具可能将来自键盘的输入写到一个文本框架。一旦将信息写入框架,需借助相关的视口、窗口更新屏幕内容。此外,系统提供了处理输入的缺省原语,这些原语有效地为PCTE工具定义了标准的交互方式。

像通常的软件系统一样,PCTE提供了三种与输入相关的用户接口对象:菜单(menu)、选择(Selection)与滚动条(Scrollbar)。

(1) 菜单 PCTE具有定义和处理菜单的功能,一旦用户选择了某个菜单项,一个菜单事件便送给了相应的工具。

(2) 选择 选择类似于菜单,若用户在视口选择了一个元素,一个选择事件被送到相应的工具。

(3) 滚动条 滚动条是与视口有关的图形对象,具有水平和垂直两个方向。用户通过激活滚动条,可看到信息的不同部分。若滚动条被激活,滚动事件便产生了。

五、分布式

PCTE基于局部网络结构,这种结构由功能很强的单用户工作站及相关资源组成。在这个环境下,项目开发者和用户可共享软件、数据以及外设,用户看到的总体结构就好像是一台计算机,每台机器是一个相对自主的单元,环境中的所有工作站具有同等权力。

分布式机制具有以下功能:

- 分布式的OMS功能、进程执行与通讯,且具有透明性。

- 网络管理,即通信层的控制管理,称为OSI传输。

- 分布式管理,即分布式本身的控制与配置。

1. 进程的分布式

在PCTE中,进程由可执行代码来描述,每个进程对应于唯一的系统范围内的标识

符,相关数据与信息构成了它的“执行环境”。

进程不能从建立它的工作站中移出,但访问远程资源的能力并未受到限制。分布式机制允许一个进程在其它工作站上建立子进程,这可通过显式引用站标识符或以隐含方式按照子进程本身的特征(如对硬件的特殊要求)来实现。同样,当访问OMS的对象时,进程无需了解其物理位置,分布式机制使进程独立于具体的拓扑结构。

2. 对象的分布式

OMS数据库被划分为卷(Volume),每卷由一个专门类型的OMS对象表示,对应于逻辑磁盘(相当于UNIX文件系统的概念)。在理论上(只要硬件允许),每卷可被动态安装到一个特定的工作站。分布式数据库的管理由对卷的静态与动态管理实现。在OMS中,卷可被静态建立或删除,也可动态安装或拆卸。

每个工作站将其它工作站看作是潜在的服务器,关系可存在于包含在不同卷的对象之间。从进程的一个参照结点开始,根据连接(link)序列航行对象库,可存取出现于已安装卷中的对象。用户对特定对象的存取,是由与OMS相关的分布式软件实现的,包括对象的定位及远程访问方法的提供。对用户来说,访问远程对象与访问局部对象没有任何区别。

3. 活动的分布式

PCTE活动具有分布式语境下的完整性与并发控制功能。系统范围内的活动语境标识以及分布式的活动均由分布式机制支持。

4. 进程通信

PCTE基本机制支持的进程通信形式是分布式的,包括命名或未命名的管道及消息队列。命名的管道与消息队列是特殊的OMS对象,由OMS加以管理。由于分布式进程结构的建立,未命名的管道也就变成分布式的了。进程通信的分布式,对用户来说,无需了解。

5. 网络管理

每个工作站由数据库中的一个对象表示, 具有全局唯一的标识符, 该标识符是在工作站最初建立时指定的, 可为所有 PCTE 原语加以显式使用。分布式机制除了管理工作站的联接与拆卸外, 还必须管理其静态拓扑结构。分布式机制本身要求配置和管理, 它隐藏特定安装的物理拓扑结构, 这个信息必须以复制的形式保存于所有工作站, 使得每个工作站自主地发挥作用。

6. 用户在分布式的PCTE中工作

用户由名字、口令、标识号、组号、参照对象及初始工作模式来标识, 口令文件中包含了每个注册用户的细节。在 PCTE 中, 每个用户在口令文件中有一个单独的入口, 且通用于所有工作站。这种方法使得工作站可共享资源, 用户可在任何已联接的工作站工作。

一旦一个工作站被启动和联接到 PCTE 的网中, 该站的用户就可以用完全统一的方式, 访问整个环境的所有资源 (软件、硬件及已安装卷中的对象), 仿佛所有资源都属于自己的工作站一样, 这充分体现了 PCTE 的思想。PCTE 环境中的用户不必了解他们所操纵进程及对象的物理位置。

PCTE 结构没有蕴含工作站的层次, 每个工作站只需访问充足的系统管理信息即可独立工作。

综上所述, PCTE 具有很强的分布式功能, 其主要特点可概括如下:

- PCTE 是基于局部网的。
- PCTE 好像在一台单机上工作 (对用户来说)。

- 分布式的 OMS 数据库、活动与进程通讯, 且具有透明性。

- 特定的拓扑结构由系统管理, 用户无需关心。

六、结束语

PCTE 产生的意义在于它提供了可以遵循且行之有效的一整套现代软件环境的标准, 对于构造集成化的软件环境具有极其重要的指导意义, 对八·五期间我国软件环境的开发也有很好的借鉴和参考价值。可以预见, 有了 PCTE 这样的通用结构, 不仅可减少软件工具的开发费用, 提高软件生产率, 而且将便于软件市场上的软件交换, 加速技术上的传播, 促进该领域思想的交流。

值得一提的是, 由法国三家公司—Bull, Eurosoft 与 Syseca 联合开发的具有实用价值的工业产品 Emerald 是基于 PCTE 标准实现的典型代表, 目前已投入使用。围绕 PCTE, 相关项目的开发仍在继续。

参考文献

- [1] PCTE Functional Specifications, Bull, GEC, ICL, Olivetti, Nixdorf, Siemens, June, 1988.
- [2] PCTE 1.5/C, Bull, GEC, ICL, Olivetti, Nixdorf, Siemens, 1988.
- [3] Industrial quality implementation of the PCTE interface standard, BullSA, Eurosoft, Syseca 1989, Printed by Cornwall Litho, Redruth, UK.
- [4] The Emerald Environment, BullSA, Eurosoft, Syseca 1989, Printed by Cornwall Litho, Redruth, UK.

下期主要内容预告

思维模拟——人工智能的根本途径; Uniface; 高级第四代应用开发和运行系统; OODB 的存储管理子系统的设计; WHYMX 存储系统的设计方法; 工程数据库中版本管理技术的研究; 基于 PCTE 标准的现代软件环境 EMERAUDE; 软件工程学科建设的若干问题