

SRA软件开发环境与工具的研究

郝克刚 乔广俊

(西北大学计算机科学系)

SRA (Software Research Associates, Inc. Company) 是日本一家有影响的软件公司, 建立较早, 研究范围广泛。目前开发的领域也多种多样。本文只介绍该公司在软件开发环境方面的研究工作。

为了提高软件生产效率和质量, SRA采用了集成化软件开发环境的概念, 其特点如下:

应用广泛性——应是内部结构灵活的复杂的环境;

用户友好性——整个系统应使用户好学易用;

工具重用性——采用工具箱办法, 包括一些模块化元素, 以便需要时重用;

功能整体性——用合理方便的机制把单一功能的工具组成复杂结构处理各用户的特殊要求;

集中数据库——信息由公共数据库管理。

完全具有上述特点的理想环境尚不存在, SRA采用UNIX作为环境原型的基础, 在VAX/UNIX上建立了适应于各类应用的软件工程环境。其中开发的一些公用软件工具, 代表了在UNIX上多年积累的经验, 可

以提高软件生产率和可靠性。这些工具包括:

·**交互式软件开发/维护工具** 从1968年开始, SRA集中了从事各种应用软件开发和熟悉UNIX系统本身的专家, 建立了一个内容丰富的程序设计环境, 其中包括不被UNIX本身支持的软件需求定义、程序设计、测试和维护等软件生存周期的各个方面。

·**用户界面软件** 初级用户界面软件可以使没有多少经验的人迅速从UNIX提供的程序设计辅助设施中获益; 普通用户界面软件, 能使用户方便地利用那些被看作UNIX专业人员独有的特点。

·**项目资料支持工具** 文件管理工具用一个数据库对整个项目所产生的资料进行控制; 更改控制工具用于管理源程序和测试数据的不同版本, 也支持软件配置的控制。

·**项目计划管理工具** 如PEGASUS可以用图示手段描述项目计划和实际开发过程。

下面介绍SRA研究的环境中的主要工具。

reasonable approach to system development" ACM SIGSOFT SEN Vol 7 No.5 1982

[4] John. L. Conneil "The Impact of Implementing a Rapid Prototyping on

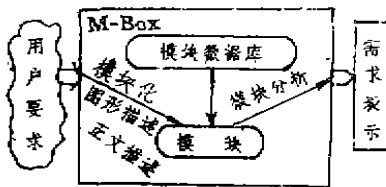
System Maintenance" AFIPS 1985

[6] Bernard H. Boar, Application Prototyping, John Wiley & Sons, Inc. 1984

一、 M-Box

模块设计工具箱M-Box是用于软件需求分析和定义的支持工具。它的目标是使不正规的用户需求表达变为清晰、精确的需求规格说明。

Box的概念图如下:



M-Box的概念图

M-Box根据用户的要求,可得到相应的较严格的需求表示。

M-Box的特征和功能如下:

(1) 通过各种模块化技术的使用,可以完成实际用户在需求分析和定义阶段的精确定义。

(2) 采用了支持面向视觉的需求的图形表达。

(3) 在多个图形终端上支持交互和多用户模式。

(4) 通过使用彩色图形终端和彩色画笔等能够表示丰富的图形模式。

(5) 提供了适应各种模式的语言,并且正文和图形模式是互相兼容的。模式语言还可根据用户的需求扩展。

(6) 模式需求表示(正文或图形表示)存储在模块数据库中,供分析时方便地访问。

(7) 提供了静态分析功能,以便检查模块需求表示的完整性、相容性和冗余性。

(8) 具有产生下列报告文档的功能。

- 模块分析报告
- 参考报告
 - 整体参考报告
 - 模式正文报告
- 模块管理报告

这样,既支持了产生分析和定义的结果,也支持了工作进展过程。

(9) 提供了菜单式、交互式用户界面。

从功能上考虑,M-Box包括下列七个子系统。

- 用户界面

- 交互式模块图形编辑器
- 模块化语言翻译器
- 模块分析工具
- 文档支持工具
- M-Box图形系统
- 模块数据库管理

二、 D-Box

D-Box是对集成化程序设计过程提供支持的工具箱。目前,虽然已经有很多程序设计方法,但是除了程序员手中积累的经验外,很少有软件工具能支持程序员的工作,这不仅容易产生错误,而且对设计文档的产生和管理需要花费很大的劳动量。

程序设计的共同问题是:

- (1) 缺少设计描述的一致性;
- (2) 设计情况的描述不精确;
- (3) 设计文档的堆积变得很难保存和管理;
- (4) 设计错误很难诊断,以后改正代价很大;
- (5) 不能轻易地转换到代码编制阶段;
- (6) 到正规表示的变化过程经常不能记录下来。

D-Box就是为了解决这些问题,用以提高设计阶段的效率和软件质量而开发的工具。它具有如下的功能和特征:

(1) 用户界面采用菜单的方式指导工作流程的进行,提高了工作效率。

(2) 用设计语言输入,系统提供了丰富的编辑功能。

(3) 支持图形的HIPO、PAD、HCP和NS-Chart方法的表示,这些都由D-Box图形功能交互式自动生成。

(4) 可以检查设计工作的当前进展状态。

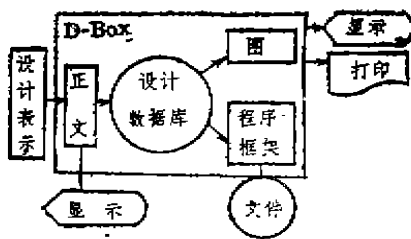
(5) 可以给出模块的树形结构。

(6) 用设计语言表示的规范描述可以选一种程序设计语言来输出框架程序。

三、 SPEX

软件工程的第一个任务就是从所开发的系统应用概念中创建较明确的规格说明,并且通过较严格的语言来表达。

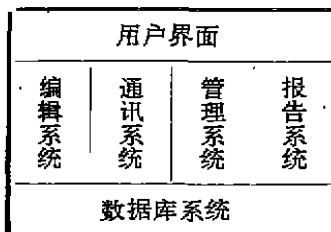
规格说明的分析和定义不仅占有软件研制任务的一大部分,而且对软件的质量和质



D-Box的概念图

量起着重要作用。然而，规格说明分析和定义阶段导出的信息经常是模糊的、不精确的，很难作为正式的资料被接收。

SPEX是SRA开发的规格表示和分析支持工具。它在规格说明分析和定义任务中可以允许有方便的、非形式的符号。SPEX的配置如下：



SPEX提供了用相对非形式的（自然的）语言表示的概念工具，并且对软件开发中每一阶段产生的与规格说明的分析和定义有关的信息进行对照分析和维护。这些信息包括：

- 思想、概念、想法；
- 决定性的方法和技术考虑；
- 系统的概念性进程；
- 中间结果；
- 可选择的最最终可能放弃的方案。

下面简要介绍一下配置成分。

(1) **用户界面** 管理所有SPEX的功能，并且对用户提供更方便的使用手段。

(2) **编辑系统** 支持所用符号的形式和用户环境的定义和修改，并且输入和修改符号的内容。

(3) **通讯系统** 支持所设计的系统中各个组成部分间的通讯和信息交换。

(4) **管理系统** 完成两个功能：管理概

念的更改和数据库存贮，并且管理版本。

(5) **报告系统** 报告每个开发成员的工作记录（进展状态和完成文档）及其它“历史”信息。

四、Testbed

Testbed是程序模块独立测试的交互式工具。过去十年中，在软件开发的测试和验证领域投入了大量的时间和人力。经过研究和试验，软件测试技术已从初始的直观、粗略的方法发展成具有一套合理原则的方法。在总结和比较现行方法的基础上，SRA提供了Testbed。这一工具不仅可以增加程序开发的效率，而且可以在与开发系统的其它部分绝对独立的环境下，进行严格的模块诊断和测试。交互式模块测试工具Testbed可以使程序员从复杂的应用系统中抽取单独的代码模块独立地进行测试。Testbed具有下列功能：

- (1) 交互式模块测试与诊断；
- (2) 自动记录测试历史；
- (3) 基于测试历史的自动重测。
- (4) 通过编辑以前的测试历史记录，可以容易地创建新的测试用例；
- (5) 动态确定进程；
- (6) 测试覆盖的方法。

Testbed可以看作作为一种命令翻译器，它从终端上接受命令控制测试执行的进程。

Testbed适用于C、COBOL、FORTRAN和PL/M等语言。

另外，Testbed具有存根自动产生的功能，所有测试结果都放在测试文件中，具有容易进行重测试等特征。

五、CCS

CCS是支持版本控制和配置管理的工具。目前软件系统工程有这样一种趋势，即开发项目的大小和系统中的模块个数都在增加。任何大型软件项目开发成功与否在很大程度上取决于对大量模块的管理。然而，模块管理不只是在开发过程中简单地保存最新版本，而且能允许采用次最新版本，或者比

较早的版本来进行测试。

为了进一步提高软件产量和质量,有必要在支持环境中建立软件管理功能。为了完成这一任务,需要管理:

- (1) 模块源代码(也可能是目标代码),
- (2) 测试数据(也可能是测试结果),
- (3) 测试计划。

从广义上讲,系统本身就是一个集成模块。可能因为如下原因,系统不必包括每个模块的最新版本。

- (1) 某些模块是从前面系统得到的。
- (2) 可能接受的是包括不同版本的库。
- (3) 可能决定返回到前面某些模块版本的暂时使用。

在系统级别上最重要的是管理配置信息。配置信息是决定用模块的哪一版本来实际产生系统的版本。CCS是SRA开发的、在上述领域支持程序员的工具。CCS具有下列两种功能:

- (1) 对构造系统的每个模块的版本管理。
- (2) 对系统本身的配置管理。

六、Midas

由于半导体技术的迅猛发展,微型计算机的应用也迅速和广泛地推广开来。然而,与飞速发展的微机技术不相适应的是,相应于硬件环境的软件的研制并不乐观,很难适应这种形势。

传统的微型机软件开发方法是直接在目标机器上完成的,这虽然有许多优点,但仍然存在着一一些问题。例如,依赖于特定目标机器的软件很难适应其它的目标机器,很难成为多用户的系统。

Midas为了解决这些问题,引入了“源/目标”的概念,使开发机器与目标机器独立开来。并且提供了UNIX的集成开发环境。这样就有效地保证了微机软件的产量和质量。用户可用UNIX开发环境和Midas提供的功能在UNIX上而不是在目标机器上来为目标微型机开发软件

Midas是一个在UNIX支持下的微机软件开发的集成环境。为了有效地开发高质量的

软件,Midas可以和UNIX一起工作,并且具有UNIX没有提供的很多特点。程序员可通过方便的用户界面,利用Midas提供的功能为目标系统开发软件。

Midas包括以下三个组成部分,它们都与目标机器有关。

(1) 交叉编译/连接器

①从C或PL/M源程序为目标机器产生目标代码。

②在C或PL/M源程序与编译成的目标代码机器语言间建立相应的联系。

(2) 模拟/信号诊断

①可以使目标机器的目标代码在源机器上执行。

②可以模拟由Midas的事件定义语言表示的严格的目标操作系统的运行。

③支持中断服务、实际访问、信号诊断等。

(3) 项目数据库

①对整个资料的统一管理。

②支持实现版本和配置管理。

七、Pygmalion

在过去几年里,已有越来越多的高质量工作站投入市场。其中包括二维图形显示器和鼠标定位设备等。许多这样的工作站也具有UNIX的特性和具有基于局域网络的分布式环境的特性。这些特性在研制软件的过程中有很高的价值和实用性。Pygmalion是一个支持C语言程序开发的工具,它在基于工作站窗口环境上充分利用目前存在的软件资源。Pygmalion提出了为创建、分析和诊断C程序的面向多窗口的用户界面。

Pygmalion包括管理模块和支持每个程序开发阶段的子程序。管理模块使用UNIX进程间通讯的能力来调用和控制各种子程序。每个子程序产生自己的窗口,利用UNIX虚拟终端控制make、vi和dbx这样的UNIX程序支持工具。

参考文献

1. Software Research Associates, Inc. Company Overview, Software Research Associates.

提高软件生产率(续)

Barry W. Boehm

四、提高

按照软件生产率机会树的组织,下面对提高软件生产率的几个主要可选方面进行讨论:1)挑选精干人员,2)提高阶段效率,3)消除人工阶段,4)减少重复劳动,5)建造简单产品,6)重用软部件。

挑选精干人员 象机会树所表示的那样,挑选精干人员有三个主要可选项:

职员 表1中生产率表明人员或队伍的能力差异对生产率有一个因子为4.18的影响,而他们对应用领域、计算机系统或虚拟机以及程序设计语言的相关经验还有一个2.52的综合因子。IBM的生产率分析也取得了类似的结果。

因此,如果想提高某个项目或组织的软件生产率,最行之有效的行动是搜罗最优秀的人才为该项目或组织工作,而让那些二流人员去别处工作。但通常经理们对这些关键人员的选定表示冷漠而且常常表示反对,经

常这样说:

“我们雇不起工资这么高的人。”

“我不能在这么‘昂贵’的人身上冒险。”

“我不能雇用你们的超级明星,除非你们的项目得到财政保障,差一个月也不行。”

“乔先生让这些都靠边站了,我得帮帮他们。”

“我不能等了。我希望有人能使这个项目下周马上有所进展。”

有时候两种情况需要你回答,但你一般也只能做一种暂时的而不是长久的安排。

另一个同等重要的问题是雇员的工资方面,它要求你去调整不合适的分配。无论你怎么精心挑选你的软件队伍,你总会发现一些人不愿意去做,为了全队的目标应做的一份工作。甚至在你企图为他们寻找或准备好了一个队中的恰当位置他们也还是这样。遇到这种情况,你可能想拖拖再说,或者假装不知道,或者说说了事,再不就请其它队员做些额外的工作。这样做在短期内可能是个办法,但长期下去,无疑会造成不良后果。

2. SRA Software Development Environment Tools on UNIX M-Box, Software Research Associates, Inc.
3. SRA Software Development Environment Tools on UNIX D-Box, Software Research Associates, Inc.
4. SRA Software Development Environment Tools on UNIX SPEX, Software Research Associates, Inc.
5. SRA Software Development Environment Tools on UNIX Testbed, Software Research Associates, Inc.
6. SRA Software Development Environme-

nt Tools on UNIX CCS, Software Research Associates, Inc.

7. SRA Software Development Environment Tools on UNIX Midas, Software Research Associates, Inc.
8. SRA Software Development Environment Tools on UNIX Pygmalion, Software Research Associates, Inc.
9. 郝克刚, 软件分析与设计的支持环境,《计算机研究与发展》, 1986.7.
10. 郝克刚, 设计结构编辑系统(DSE)功能说明书, 西北大学计算机科学系软件工程研究室, 1985.11.