

青鸟工程现状与发展

——兼论我国软件产业发展途径

杨芙清

(北京大学计算机科学技术系, 100871)

摘要 本文回顾了青鸟工程发展的历程, 全面总结了“八五”期间青鸟工程的成果, 提出了青鸟工程进一步发展的方向, 同时, 结合青鸟工程的实践, 阐述了作者对我国发展软件产业的若干看法及思路。

1 历史的回顾

青鸟, 又称“三足鸟”, 传说是我国远古神话中西王母的信使。我国“七五”攻关的成果——一个集成化的软件工程开发环境——被命名为青鸟 I 型系统, 其取义为: 软件产业是信息产业的核心, 软件正是专司信息传递(服务)的使者。从青鸟命名至今, 已历经一个五年计划, 青鸟已由一个具体系统演化为一项系统工程, 即青鸟工程。

我国的软件产业起步于 80 年代初, 从一开始, 人们就认识到软件产业的发展离不开软件工程的支持, 因此加强对软件工程学科的研究和探索软件产业发展策略和模式便成为学术界和产业界紧迫的任务。在“六五”及“七五”期间, 众多的研究者在国家支持下致力于软件工程基础研究和软件产业的基础设施建设, 出现了很多研究成果, 包括众多的软件工具、天马系统、XYZ 系统及青鸟 I 型(JB1)环境等。同时, 创建了一批软件企业。

然而, 应该看到, 软件开发的主要手段仍然是手工作坊式的劳动, 软件工程的思想远未普及, 软件开发中的低效率和低质量现象仍然存在, 这些都制约了软件产业的发展。为此, 在国家“八五”计划中, 列入了“软件工程开发环境(CASE)的标准化和实用化”课题的攻关。其目标是在上述成果基础上进一步发展, 为建立我国软件产业基础, 形成软件产业发展所必要的工业化生产手段, 为实现软件产业的规模经济提供人才储备、技术储备和产品储备。该课题以实用的软件工程技术为依托, 集成已有成果, 形成具有自主知识产权的、规范的、实用的软件工程开发环境; 为软件产业提供基础设施——软件工具、平台和环境, 建立工业化生产手段, 促进我国软件开发从手工作坊式转向用计算机辅助开发, 提高软件开发效率, 改善软件产品质量, 增强我国软件在市场上的综合竞争力。

经过五年的攻关, 青鸟工程规模日益增大, 成果斐然。

2 青鸟“八五”成果总结

“八五”期间，青鸟工程发展迅速，圆满实现了国家确定的攻关目标，取得了丰硕成果，受到国家有关部门及同行专家的好评，已引起国内外广泛关注。总结青鸟工程的成果，主要体现在如下几个方面：

2.1 青鸟 I 型系统的实用化和推广应用

根据我国软件市场的需求，在“七五”攻关成果的基础上，形成了青鸟系统的实用化版本系列，包括JB1工作站版、JB1微机Windows版、JB1微机Unix版等，这些系列产品已进入软件市场，取得了较好经济效益和社会效益。

配合JBCASE系列产品的发布，在一些大型软件项目中使用了青鸟系统，这些项目涉及系统软件、证券交易、大型MIS、商业自动化等领域。青鸟系统的使用为缩短开发周期，提高开发效率，改善系统质量起到了很好的作用。

为了普及软件工程思想和模式，我们先后在北京、上海两地，以集中、分散形式举办了多次软件工程及青鸟系统培训班，并通过各种方式和各种宣传媒介，传播青鸟系统，使得青鸟系统更广泛地被了解，软件工程思想被更广泛地理解和接受。

2.2 青鸟 II 型 (JB2) 系统的研制开发

JB2的研制是“八五”攻关课题中最重要，也是最大的一个专题，其目标是在跟踪、分析、消化、研究相关的国际标准的基础上，制订符合中国国情的CASE技术标准和开发规范，进而在标准和规范的框架内，研制开发实用的、集成化面向对象软件开发环境。

JB2的研制开发始终坚持既符合国情又与国际接轨的标准规范及与市场接轨的工程实践，技术上瞄准前沿，注重实用。JB2采用了与国际标准接轨的开放的体系结构和统一的工具结构模型，提供了良好的环境机制和支持结构化开发方法、面向对象开发方法及面向特定应用领域的工具集，以对象管理系统(OMS)为核心构成了开放的可剪裁组装的集成化软件开发环境，体现了紧密集成性和开放性的高度统一。通过对JB2的剪裁和组装，形成了两个支持不同软件开发方法的环境：结构化软件开发环境JB2/SM和面向对象软件开发环境JB2/OO以及三个面向特定应用领域的开发平台：管理信息系统开发平台JB2/MIS，地理信息系统开发平台JB2/GIS和实时监控系統开发平台JB2/SCADA。

JB2系统在技术上及系统规模上均达到90年代国际先进水平。JB2系统已在系统软件、MIS、GIS和实时监控等领域进行试用，效果良好，表现出很好的可用性和实用性。

2.3 青鸟标准规范的制订和质量保证体系的建立

青鸟工程是一项庞大的系统工程，涉及的单位及研究人员众多，要保证青鸟系统的研制具有统一的概念、统一的行为和统一的产品形态，相应的标准规范是必需的。在研究、分析国内外相关标准的基础上，我们制定了十四种标准规范，可分为JB2技术规范、面向

对象开发规范、文档规范及测试规范等四大类，这些标准规范在课题组内的贯彻执行，为课题的成功完成起到了很好的保障作用。

为了保证青鸟工程的质量，我们建立了青鸟质量保证体系，在组织机构上，采用了专家总体组—工作组(技术组、质量保证组、管理组)—专题组三级管理模式，在开发过程中全面贯彻质量意识、产品意识和攻关意识，由质量保证组负责质量监督工作，制定有关标准及规范，实施对有关专题或专题的技术评审、质量检查和功能测试，有效地保证了系统开发的质量。

2.4 新技术和新工具研究

以当今软件工程新技术为重点研究方向，研究面向应用系统的开发模型和方法，研制新型工具，为青鸟系统的进一步发展提供技术储备。成果包括实用智能化软件工具集、分布式系统辅助开发工具包、实用自动化工具集和一批应用领域系统模型等。同时，还在国内外学术刊物上发表了一批高水平的学术论文，对软件工程学科的发展起到了良好推动作用。

2.5 成果转换机制的建立

软件产业的发展需要一批技术领先、队伍整齐的软件企业。为了加快科研成果向生产力的转化，作为青鸟工程的重要组成部分，在国家计委、电子工业部和北京大学的支持下，我们成立了北京北大青鸟有限责任公司，一方面作为青鸟成果转化的良好机制，使青鸟产品尽快走向市场，并借助青鸟系统开发大型应用系统，探索应用软件系统开发方法和途径；另一方面积极推行软件工业化生产方式，传播软件工程思想、技术和方法，探索软件产业发展模式。青鸟公司成立仅一年多，在青鸟产品销售、大型应用系统开发等方面获得初步成功，资金总值达四千万，成为国内外一些著名公司竞相联合的对象。

同时，基于青鸟工程的软件工程国家工程研究中心的筹建，将成为青鸟工程科技成果工程化的机构和产品储备的重要保证。

2.6 人才培养和队伍建设

软件开发是智力密集型工作，软件产业的竞争归根结底是人才竞争。通过“七五”、“八五”十余年攻关，我们已经培养、建立并稳定了一批水平高、结构合理、可从事软件工程技术攻关和软件产业建设的骨干队伍。“八五”期间，全国20多所著名大学、科研院所和产业部门300多人参加了国家“八五”重点科技攻关工作，其中有中科院院士3人，80%以上为中青年科技人员，形成了以北京为中心向全国各地辐射的科技攻关网。

青鸟工程的集中和分散相结合的攻关方式，增加了各单位间的协作和交流，形成了有机结构，产生出科研工程的规模效应，使得有限的资金投入，得到最大化的产出。同时，锻炼出一批足以担当重任的年青软件人才。这一点对我国软件产业的发展具有重大意义。

3 青鸟工程的进一步发展

“八五”攻关成果为我国软件产业建设和发展建立了良好的基础。但是,面对当今信息技术迅猛发展的形势,面对软件市场激烈竞争的局面,面对国家国民经济信息化建设的需求,发展中的中国软件产业正面临着极好的机遇和严峻的挑战。为了满足国民经济信息化对软件产品的需求,满足我国软件产业发展的需求,青鸟工程进一步发展目标是:在国家支持和政策的指导下,迅速突破软件工业化生产的关键技术,形成具有中国特色的软件产业模式,推动我国软件生产方式的变革,使之从分散、小型、作坊式的生产过程,转变到工业化生产的轨道上来,使我国的软件产业能在先进装备和先进管理模式的支撑下迅速壮大起来,在自主的系统软件和支撑软件基础上,大力发展应用软件,支持大型信息工程建设,成为名符其实的信息产业中的核心产业,成为国民经济支柱产业之一。

“九五”期间,青鸟工程的任务是:进行“软件工程环境(青鸟CASE)工业化生产技术及系统(JB/SEIMS)的研究开发”,旨在借鉴国外“软件工厂”的概念及实践,将关键技术攻关,JB/SEIMS实验基地建设以及软件生产基金的设立融为一体,实现科技经济一体化,为软件企业提供装备(工具、平台、环境),形成软件产品的工业化生产方式,使软件产业走上规模经济。图1为青鸟工程在“九五”期间的工作框架设想。

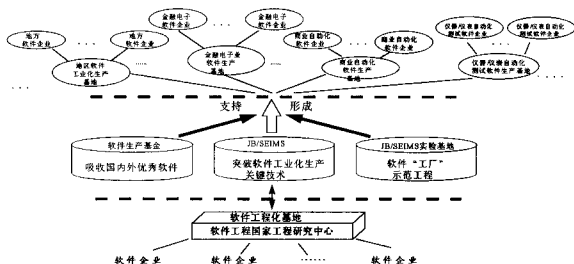


图1

其中作为国家支持的重点科技攻关项目JB/SEIMS系统的研究开发又可概括为如图2所示:

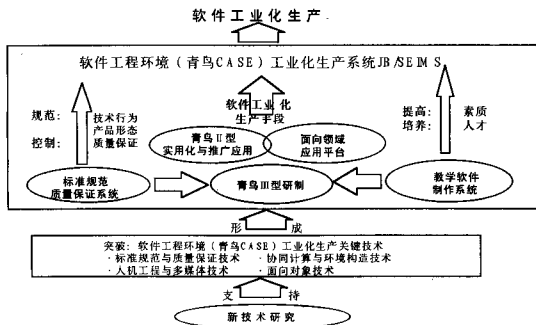


图2

青鸟系统的目标为：面向应用领域，在JB2的基础上，继续强化采用面向对象技术，建立基于重用，以构件库、构架库等多库机制为基础的构件—构架模式的应用软件系统集成环境。选择重点应用领域与领域专家相结合，通过领域分析和构件提取，充实构件库和构架库，研究面向领域的开发模型和方法，研制专用工具，组装专用（应用）平台，支持开发应用软件系统。图3表示了青鸟系统的发展思路。

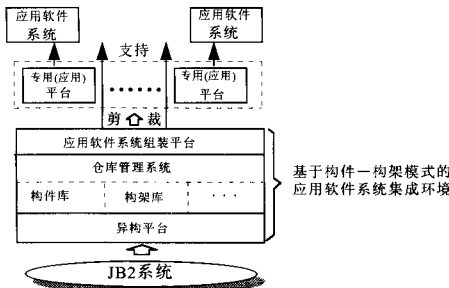


图3

从软件生产的角度来看，新一代的青鸟系统将为形成软件生产线(如图4)，完善软件的工业化生产模式，实现软件规模化生产提供有力的保障。

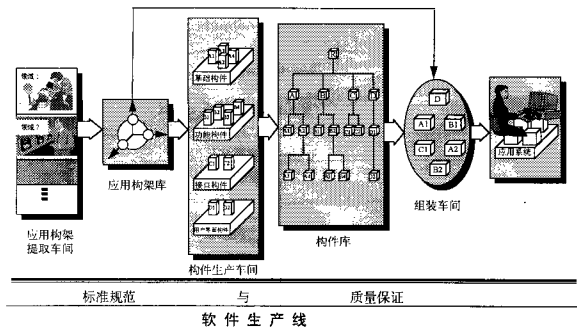


图4

可以预料，在国家支持下，经过众多科技人员的协同攻关，我们一定可以实现“九五”攻关目标，用更先进的技术来装备我国软件产业，为软件工业化生产方式的推行、软件产业的发展和国家信息化建设作出贡献。

4 软件产业发展思路与途径

我国软件产业在国家正确方针的指导和支持下，通过从“六五”、“七五”到“八五”十多年的关键技术攻关和成果推广应用，已建立起软件产业的初步基础，系统软件和支撑软件都有了自己的产品，在产品化、商品化及包装技术方面也正在适应市场而不断完善，在市场竞争中民族软件企业正在发展成长。当然，目前我们面临着严峻的竞争和挑战局面，只要政府采取扶持政策，必将会有力地促进中国软件产业的发展，充分发挥中国的软件优势，使软件产业在竞争中成长。

软件产业是信息产业的核心，中国的社会和经济信息化必须建立在我国自主软件产业的基础上。系统软件和支撑软件作为基础软件是软件产业的根本，发展应用软件必须建立在自主的基础软件上，否则将成为无源之水，无本之本。

我国在基础软件技术方面有优势，十多年的攻关已涌现大量成果，且有产品问世，因此，“九五”期间，依托和发展自己的基础软件，不断引进新技术，突破应用软件系统是可能的，也是合适的。

基于我国软件产业的现状，为了使其尽快成为具有一定技术实力、一定规模经济，一定应变能力，并能有力地参与国际竞争的产业，促使我国信息产业的发展，我们需做好如下几方面的工作：

4.1 重视软件科学和技术的研究

发展软件科学技术是建设软件产业的根本，软件科学技术的研究将为软件新产品研制、应用系统的开发，信息工程建设提供新思路、新技术、新方法。就我国软件产业的现状而言，当前应重视软件工业化生产技术，系统软件技术及中文处理技术的研究，为软件工业化生产提供先进手段，促进软件生产方式的变革，为开发应用软件系统提供良好的工具、平台和开发环境，保证信息的安全性、系统的自主性和文化性。

4.2 建立合理的软件产业结构框架

充分认识软件产业的特点和自身发展规律，建立合理的软件产业结构框架，以形成技术和经济的良性循环。一个合理的软件产业结构应由四个层次组成。第一个层次是基础研究和应用基础研究，重点研究有关软件的各种新理论、新技术和新方法，它是推动软件产业发展和促进软件产品革新的理论基础；第二个层次为模型研究和原型开发，针对国民经济与科学技术发展的需要，将第一层次中经评价确认有发展前途的研究成果，用于研究与开发软件新产品的模型和原型，解决关键技术问题；第三个层次是软件的产品化开发与生产，将科技成果转化为实用的软件产品，使其在国民经济的各个领域真正产生效益，重点解决软件产品化技术问题；第四个层次是市场服务，面向不同用户，做好技术培训、产品维护和信息反馈工作，重点解决软件商品化和软件产品维护技术问题。

4.3 重视软件产业基地建设

为建设软件产业，形成软件产业的规模经济，应统一规划、通力合作，重点建设几个软件产业基地，并在基地的基础上，发展各类软件企业以加速我国软件产业的发展。软件产业基地应包括研究开发、产品开发和质量评测等三个组成部分。研究开发实体主要进行第二层次的工作；产品开发实体主要进行上述第三层次和第四层次的工作；软件质量评测实体主要研究软件质量度量与软件产品评价技术，研制并推广软件质量保证工具与软件质量评测工具，完善软件产品开发与生产体系。软件产业基地的建设要考虑到分布的合理性，要有特色，要有利于优势力量的组合，形成综合效应。

4.4 重视人才培养、加强队伍建设

软件产业是智力密集型和技术密集型产业，人才是关键。从软件生产的角度看，软件生产的队伍应是具有高水平的技术队伍，其开发经验是开发高质量产品的重要因素之一；从软件开发方式来说，软件开发是一种群体合作的模式，软件产品是群体劳动的结晶，并

发人员间的默契配合(包括技术、作风等)、紧密合作是成功的关键,因而队伍建设是重要的环节。同时,特别要重视软件经营销售人才的培养和经销、管理队伍的建设。

4.5 开展国际合作、开拓软件市场

开展国际合作、参与国际竞争,是发展软件产业的重要措施,也是开拓软件市场的一个重要方面。目前国外软件大批进入中国市场,形势逼人。竞争推动技术进步,竞争促进产业发展。软件产业也必然会在市场机制中,在激烈竞争中形成、成长、发展。

5 结束语

青鸟工程已历时十余年,它应我国软件产业发展的需求而诞生,也将随着软件产业的发展而发展。任重而道远,我们只有不断努力,刻苦攻关,努力促进成果转化才能使青鸟满足我国软件产业不断发展的需要,并不断地为软件产业发展推波助澜。青鸟展翅高飞,是我们的殷切期望,也是历史赋予我们的使命。

Windows版青鸟软件工具系统的设计与实现

黄柏素 陈钟 杨芙清

(北京大学计算机科学技术系, 100871)

摘要 Windows版青鸟软件工具系统是国家“八五”科技攻关课题《软件工程开发环境(CASE)的标准化与实用化》的一个组成部分, 它是一个支持结构化软件开发方法的CASE工具系统。本文着重阐述了该系统的设计思想和关键技术途径, 介绍了系统的体系结构和主要功能, 最后, 分析了进一步要做的工作。

关键词 软件工程, 软件工具, 面向对象, 结构化方法

1 引言

软件的开发, 尤其是大型软件项目的开发是一个非常复杂的过程, 如何降低这种复杂性, 是目前软件界研究的重点。解决问题的方法之一是采用CASE工具和环境, Windows版青鸟软件工具系统(以下简称JBCASE for Windows)就是微机Windows平台上的一个CASE工具系统, 它支持结构化的软件开发方法, 主要由以下五个互相关联的工具组成: 结构化分析工具SAT、结构化设计工具SDT、文档追踪工具DAT/T、数据库设计工具DDT以及文档出版工具DPT。

JBCASE for Windows的支持覆盖了软件开发的主要阶段, 包括需求分析、一般设计、详细设计及数据库设计等, 使得软件开发从传统的手工作坊方式转变成使用计算机辅助设计的工作方式, 这种软件开发“好、快、省”的方式是广大软件开发人员所期盼的, 使用青鸟软件工具系统一方面加强软件开发和文档的规范化, 使软件开发工作上规模、上档次, 另一方面可以提高软件开发质量和效率, 降低开发和维护成本, 提高软件的可维护性和可重用性。本文着重介绍了该系统的主要功能和关键技术特点。

2 系统概述

2.1 系统目标

JBCASE for Windows的目标体现在以下几方面:

- (1) 为软件开发人员提供自动化的辅助工具，减轻开发人员工作量，提高软件开发的效率和质量。
- (2) 支持软件工程化方法。
- (3) 使结构化方法更加实用化和规范化。
- (4) 生成符合规范的标准化文档。
- (5) 集成性和开放性的统一。

2.2 系统体系结构

图1给出了JBCASE for Windows的体系结构，整个系统基于微机Windows平台，以文档出版工具为中心，通过OLE共享信息，实现工具间的互操作，同时提供了与工作站教育鸟环境的信息通讯功能，并支持外部工具的集成。

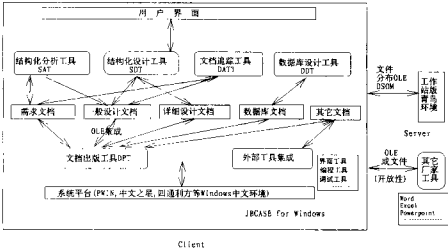


图1 JBCASE For Windows体系结构

2.3 功能简介

JBCASE for Windows中各个工具的主要功能可以通过图2来简要说明。

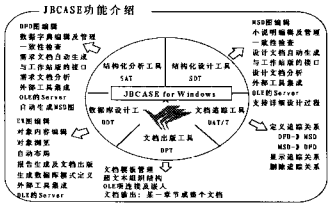


图2 JBCASE for windows功能简介

2.4 系统开发及运行环境

JBCASE for Windows采用面向对象程序设计语言Visual C++1.5语言编程,可运行于Windows的多种中文环境上,如PW1N、中文之星等,具有较好的可移植性。

3 系统主要特点

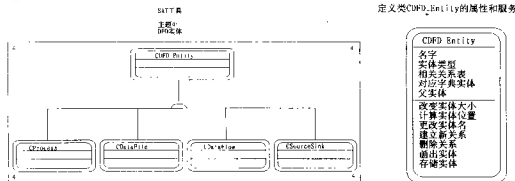
JBCASE for Windows的设计兼顾了两方面的因素,一是方法上技术上的先进性,一是产品化实用化的要求。这里,根据图1中给出的JBCASE for Windows的体系结构,来具体分析系统的主要特点。

3.1 系统开发方法

JBCASE for Windows工具系统支持结构化的软件开发方法,但该系统本身的开发却全面地采用了面向对象的方法和技术[3][4]。首先,采用了面向对象的方法进行系统分析,提取出系统中的类及对象,建立了问题域模型,接着,对模型不断扩充,并增加实现时所需要的细节,平滑过渡到设计阶段,建立OOD模型,最后,在OOD模型基础上采用Visual C++语言编程。Visual C++是一种面向对象程序设计语言,OOD模型中描述的类及对象、一般

特殊结构、整体一部分结构等在该语言中均有与之对应的实现机制,选择这种语言就是因为它能在实现阶段更好地体现面向对象的思想。

面向对象的软件开发方法在表达上具有分析、设计和实现等各个阶段平稳过渡的特点,而且支持多个层次上的软件复用。在系统建模层次上,它支持从系统分析、设计到实现之间的系统框架和模式的复用;在系统构造层次上,它支持对已有类库的复用;在程序设计层次上,面向对象程序设计语言具有封装、继承、多态等特性,有效地支持了复用。JBCASE For Windows的设计充分利用了这些特点,复用了Document/View框架和MFC类库,同时,抽象出各个工具共同的属性和操作,以动态连接库的形式构造了系统的可复用类库,使各个工具可以复用该类库。这样,不仅降低了系统的复杂性,而且降低了开发人员的工作量,缩短了开发周期。这里,给出一个SAT的开发片断,描述了数据流图的实体类,它是一个抽象类,数据流图的四中具体图元都设计为该类的子类,以实现代码级的复用。



相应Visual C++程序如下:

```
class CDFD_Entity: public CObject{
public:
    CString        m_entityName;
    DFD_Type       m_type;
    CObList        m_relaList;
    CDD_Entity*    m_pDDEntity;
    CDFD* m_pParentDFD;
protected:
    virtual void    ChangeEntitySize(float scaleX,float scaleY);
    virtual void    CalcEntityPos(CPoint point);
    virtual BOOL    Rename();
    CRelation*      newRelation();
    BOOL            RemoveRelation(CRelation* pRelation);
    virtual void    DrawEntity(CDC* pDC,int nErase);
    virtual void    Serialize(CArchive& ar);
};
```

3.2 JBCASE for windows的集成性

从图1中可以看到在JBCASE for Windows中包含三个方面的工具集成,它们分别对应不同的层次。

3.2.1 系统内部工具之间的集成

JBCASE for Windows内部工具之间的集成是通过OLE(对象连接与嵌入)来实现的。JBCASE for Windows的五个工具均设计为OLE构件对象,构件对象由两个部分组成,如图3所示。

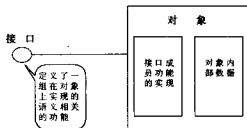


图3 JBCASE for Windows构件对象

其中,结构化分析工具SAT、结构化设计工具SDT及数据库设计工具DDT均设计成支持OLE的Server,而文档出版工具DPT则做成OLE的Client,这样,由SAT、SDT、DDT工具产生的文档,包括DFD图、MSD图、程序流程图、ER图及相应说明信息均可以连接或嵌入到DPT工具中,而且,这种关系一旦建立,那么,Server端对这些文档的任何修改均会自动反映

在Client端,同时,从Client端的DPT工具中也可以启动和访问server端的工具,支持工具间的互操作性。

不仅如此,因为OLE定义了一种标准接口,提供了可扩充的构件式的应用环境,使得不同应用程序可在同一个Client中运行,所以JBCASE for Windows不仅支持系统内部工具之间的互操作及信息共享,而且能够与其它厂家开发的软件互操作及共享信息,使得用户能够以更加自然、更加方便的方式进行系统分析及设计工作,可以选择他所熟悉的工具来编辑文档,并能够创建更加符合用户要求的复合文档,在该文档中可以编辑由不同应用程序生成的各种格式的数据,支持图文混排。

3.2.2 与工作站版青鸟环境之间的集成

目前,用户的软件开发环境是复杂的,用户具有的软件投资也是多方面的,尤其是大型软件项目,需要协同工作,共享信息。基于这些考虑,JBCASE for Windows提供了与工作站版青鸟环境的信息通讯功能,使得两者之间可以通过局域网或广域网互相交换信息,从而解决了多平台工具集成的问题。

这样,JBCASE for Windows即可以在单机环境下独立运行,也可以作为client端的工具集,同server端的工作站版“青鸟”环境进行通讯,构成一个client/server结构(如图4所示),支持跨平台进行软件开发工作,从而有效地支持大型软件工程项目中的协同工作方式。

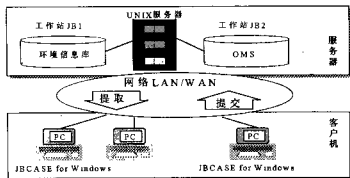


图4 JBCASE for Windows与工作站版青鸟环境的集成

3.2.3 与外部工具之间的集成

提供了开放式工具接口,使外部工具可以方便地以软插件的形式集成到JBCASE for Windows中,并产生语义相关的连接,使之成为工具中有机的一部分,同时,还提供了外部工具管理机制,使得用户可以自由增删工具,并将工具信息作为一种配置信息保存起来。

3.3 建立了SA和SD之间的双向联系

结构化方法不象OO方法，在OO方法中，从系统分析到设计是一个渐进的模型扩充过程，是平滑过渡的；而在结构化方法中，系统分析和设计是两个截然分离的过程，分析的结果不能直接过渡到设计阶段，而设计阶段对系统的改动也难以反映到分析中，造成分析和设计之间的一致性，使得系统难以维护。

JBCASE for Windows中，是通过两种方式来解决这个问题的，如图5所示，第一是利用文档追踪工具DAT/T，在需求分析和软件设计之间建立一个双向追踪关系，使得用户可以了解到分析中某些部分的定义或环境发生了变化，会影响到设计中哪些模块，或者对设计中某些部分的修改又会影响分析中哪些加工，并对这些部分作相应修正，从而保持了分析和设计之间的一致性。第二，JBCASE for Windows中提供了从DFD图自动生成MSD图的功能，这样，从分析阶段的结果可以自动得到部分设计的信息，同时，转换时所产生的对应关系也可以自动记录到DAT/T工具中。

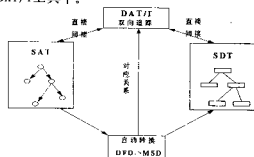


图5 SAT和SDT的双向联系

3.4 友好的用户界面、统一的操作方式

所有工具具有统一的界面风格，由于JBCASE for Windows既可作为单独的工具集独立应用，也可作为工作站版青鸟环境的Client端开发工具使用，因此，其界面设计兼顾了微机Windows的界面风格和工作站青鸟环境的界面规范。同时，在界面功能的布局中，突出了面向对象的特点，使界面的对象及其上的操作合理地组织在一起，并采用了先选对象，后选操作的方式，更符合人机工程学的规则。

JBCASE for Windows中全部操作采用鼠标"Drag and Play"方式，并提供了超文本形式的上下文相关帮助，方便用户使用。

4 结束语

目前，JBCASE for Windows已经得到一定程度的应用，但在方法和技术上仍需要进一步研究和探索，在功能上仍需要不断改进和完善，具体讲，可以从以下几方面做进一步的工作：

- (1) 多平台及环境机制

将JBCASE for Windows工具系统移植到Windows NT, Windows 95及OS/2平台上, 并增加配置管理、版本管理等。

(2) 工具的改进

大型软件项目的开发需要协同工作, 每个人的分析和设计结果应该能合并在一起, 并提供必要的一致性检查手段, 使得合并的结果是严格而没有二义性的, JBCASE for Windows工具下一步要实现这个功能, 支持小组工作方式。

参 考 文 献

- [1] David Canfield Smith & Joshua Susser. A Component Architecture for Personal Computer Software.OOPSLA'94
- [2] Kraig Brockchmidt. Inside OLE. Microsoft Press. 1994
- [3] 邵维忠等译. 面向对象的分析. 北京大学出版社
- [4] 邵维忠等译. 面向对象的设计. 北京大学出版社

一个实时软件原型建造环境的设计与实现

王振宇 汪 洋

(中船总第七〇九研究所, 武昌74223信箱 430074)

摘要 本文报告了在Sun工作站上基于Unix环境实现的一个实时软件的快速原型建造环境(PERTS)。结合实时软件的特点, 提出将实时需求分为周期需求、非周期频繁需求和非周期偶发需求三类的新思想。在事件—动作模型的基础上, 设计了一种实时软件原型描述语言RTSPDL, 突出了对实时需求的优先级及异常和违章机制的描述。该环境包括一组原型建造工具和原型的仿真环境, 支持软件重用的开发方法。

关键词 原型建造, 事件—动作模型, 实时系统, 实时需求

1 实时系统的描述模型

我们采用事件—动作模型(Event-Action Model) [2]。事件—动作模型是描述实时系统的常用模型, 该模型是基于以下几个概念:

- **事件** 事件是在某一特定时刻发生、具有瞬时性, 是系统行为的标志。有四种类型的事件:

- (1) 外部事件: 来自系统之外的动作产生的事件。
- (2) 开始事件和终止事件: 用来标识动作开始和结束的事件。
- (3) 转移事件: 用来标识系统操作条件改变的事件; 同时标识系统状态特征的改变。
- (4) 内部事件: 在实时系统内产生或激活的事件。

- **动作** 动作是系统对事件的反应。动作有简单动作和复杂动作:

- (1) 简单动作 也称原动作, 它是能消耗一定的系统资源的操作。定义三个简单动作:
 - a. 命名动作 由唯一的名字来标识;
 - b. 转移动作 标识系统内状态的改变, 由一些条件值来表示;
 - c. 激活动作 激活系统的内部事件。

- (2) 复杂动作 是由其他动作所构成的一个动作序列, 构成复杂动作的每个动作称为子动作; 每个子动作必须是先前定义过的。顺序动作, 选择动作和并行动作是系统中的三种复杂动作。

- 条件 条件是用来表示系统的状态集，动作的执行或外部事件的发生可引起条件的改变，而条件的改变又可引起事件的发生。

事件—动作模型就是由这三个基本概念组成，其特征是：当某一动作满足一定的条件时作用于某一事件，系统就由一种状态转换成另一种状态；继而引起系统行为的改变。

2 实时需求的分类及原型描述语言

2.1 实时需求的分类

实时需求描述了系统对单一事件的反应；是事件—动作模型基本概念的有机整体。它指明了各事件—动作之间的时序关系。

以往的研究者只将实时需求分为周期需求和偶发需求两种。我们认为这样分类并没有准确地反映出系统中需求的特征，特别是没有顾及到语言的实现效率。因为，任何语言都不能回避效率问题。若一个语言需要的编译程序过分复杂，或一个语言导致内存和运行时间的使用效率低，那么这个语言必定会把这些低效因素带给所有机器和所有程序[2]。所以，在设计一个语言时，一方面要找出应用领域需要的功能，另一方面要考虑到语言的最终执行效率。我们将实时需求分为三类：

(1) 周期需求：该类需求所描述的动作每隔一定时间就重复执行一次；其特征是要有一个动作执行周期以及动作执行所需的时限Deadline；周期需求一般实现实时系统中的监控功能，而且其执行不会对系统中的事件产生影响。

(2) 非周期频繁需求：该类需求描述的动作是由某一事件的发生而被执行，是系统中经常发生而且由一个开始事件来激活的需求，它描述的是在某个事件上发生什么动作，动作的执行受到哪些时间条件的制约；这些制约除了 Deadline外，还定义了两个时间约束条件Separation和Timeout；其中Separation用来限制同一事件连续发生所需的最小间隔时间；Timeout则是指两同一事件连续发生之间的最大间隔时间。实时系统中各操作的时序关系在非周期频繁需求中得到充分体现。

(3) 非周期偶发需求：该类需求的特征同非周期频繁需求一样，它同非周期频繁需求的区别在于这类需求在系统中出现的次数极少。

非周期需求是实时系统中的主要成分，它承担了系统的主要工作，因而它们的执行效率影响着整个系统的效率。有些非周期需求在系统中出现的次数极少，而有些却频繁出现；将它们分为非周期频繁需求和非周期偶发需求不只是在出现的频率上进行了区分，更重要的是它们在执行时将以不同的机制来完成。在系统最终实现时，将非周期频繁需求映射成Ada的任务，这些任务一旦被激活，就长驻内存，并始终处于活动状态；而将非周期偶发需求映射成Ada的过程，当需要时，才进行一次调用。这样可有效地提高整个原型系统的执行效率。

2.2 需求的优先级

在实时系统中，需求的优先级是一个十分重要的概念。由于各个需求的紧迫性是不一样的，有些需求一旦发生，就要求系统及时作出反应。为了区别这些需求的紧迫程度，必须为它们指明相应的优先级。需求的优先级由系统设计者根据实际应用的的需要，在对需求进行