

软件世界

SOFTWARE WORLD

(双月刊)

1984年9月创刊

社 长 周慕昌 副社长 杜娟
主 编 孙毓林 副主编 王锡生
主 办 电子工业部计算机与微电子发
展研究中心

编辑出版 《软件世界》编辑部
地 址 (100036)北京复兴路乙20号
电 话 821 2233-3431, 5052
印 刷 电子部科技情报所印刷厂
国内总发行 北京报刊发行局
订 购 全国各地邮局
邮发代号 82-469
刊 号 ISSN 1005-2348/CN11 3394
广告许可证号 京海工商广字 004 号
每期定价 1.50元 全年定价 9元
出版日期 1993年9月21日

MAIN CONTENTS

Object-Oriented Technology—A Key
Software Technology of 90s(3)
New Generation Spreadsheet—Combina-
tion of Integration Theory with
Electronic Windows(6)
Designing of SASS by Object-Oriented
Structure Method(9)
Universal FOXBASE Full-Screen Batch
Input Modification Programming(13)
A Trial of Embedded Operation among
Different Database systems—De-
sign and Implementation of DECnet-
PVDB System(16)
The Difference between Turbo C and C++
File Functions and Their Uses(20)
Designing of A Group of Novel Dynamic
Pictures(22)
The Main Features and New Functions of
ORACLE 7(26)
Several Features of DR-DOS 6.0(28)
Test and Evaluation of Non-Programmed
Database systems(29)
Microsoft Windows NT—A New Gen-
eration Operation System(36)
MS DOS 6.0 Perspectives(40)
Lectures on UNIX System V Programming
Chapter 7 Operations of Disk File(49)

目 录

1993年第5期(总第81期)

技术研讨

90年代软件核心技术 面向对象技术……于功第(3)
集成理论与电子窗口相结合的
新一代数表软件探索……杨传杰 黄效国(6)

开发与应用

用面向对象的结构化方法设计
—科技软件共享系统 SASS……李 军 张筠霞等(9)
FOXBASE 全屏幕通用批输入修改程序设计…冯 俊(13)
不同数据库系统间嵌入式操作的尝试
—DECnet-PVDB 系统设计与实现简述…吕 勇(16)
AutoCAD 在计算裂隙网络分数维方面应用…周启友(18)

实践与经验

Turbo C 和 C++ 两类文件函数的区别和使用…黄焕如(20)
—组新颖动态的画面设计……黄定杰(22)
WPS 加密文件的解密技巧……张 凝(23)
用 XENIX 系统修复 DOS 系统硬盘与文件…谢金钟(23)
—个 ZENIX 系统常见问题的处理办法……王 鑫(24)
利用 Debug 在文本中加入打印控制码……林小明(24)
dBASE III 程序逻辑框图的自动生成……郭乃林(24)

产品介绍

ORACLE 7 的主要特点与新功能……陈少昭(26)
DR-DOS 6.0 的几个特点……邱向群(28)

软件评测

非编程数据库系统的评测……肖文贵(29)

微软专栏

微软公司在中国……(33)
微软公司产品汉化策略……田本和(34)
Microsoft Windows NT——新一代操作系统……(36)
MS-DOS 6.0 透视……(40)
微软公司在中国的培训服务和图书出版……(42)

经营与管理

软件经营连载之五——软件利用的经营……陈幼松(43)

软件水平考试

程序设计试题解法探讨……朱慧真(45)

技术讲座

UNIX System V 程序设计讲座
第七讲 盘设备文件操作……周祖伦等(49)

知识园地

英汉对照软件专业时文选读
——解人之难的中件……刘 欣(54)

软件市场

出版软件简介(四)……(55)

信息之窗

国内要闻……(61)
国际信息……(62)
台港信息……(60)

计算机软件著作权登记公告

……(57)

九十年代的软件核心技术 ——面向对象技术

已发表 1993

于功弟

摘要 本文比较详细全面地介绍了面向对象软件技术的基本概念和基本方法。包括面向对象的程序设计语言、面向对象的程序设计方法、面向对象的操作系统、面向对象的数据库管理以及面向对象的系统分析方法等。

面向对象的软件技术于80年代开始兴起,现已成为软件技术研究中的重要领域。特别是近几年来,面向对象的研究遍及软件系统的各个领域,如面向对象的程序设计语言、面向对象的程序设计方法、面向对象的数据库管理系统、面向对象的操作系统以及面向对象的系统分析方法等。可以说,面向对象的概念的出现是程序设计方法学和软件工程方法学的一个新的里程碑。概括起来,面向对象的软件技术又主要分为:面向对象的程序设计语言、面向对象的数据库技术和面向对象的系统分析与设计方法。

目前,面向对象的程序设计(object-oriented programming),即 OOP 正在取代80年代的 structured(结构化)一词,而成为软件界当今最为流行的一种形式。人们希望这种新的形式能给软件工程带来新的生机与突破,使软件产业能像硬件产业那样设计与生产出各种型号的“软件集成块”,用于装配大型软件。专家们普遍认为,面向对象的软件技术是90年代软件行业的核心技术之一。

一、对象的基本概念

目前关于对象的概念有多种说法,但比较一致的说法如定义1所述。

定义1 对象是下列一系列现实世界事物的抽象。

- (1) 这些事物即实例具有相同的特征。
- (2) 这些事物都服从和遵守相同的规则。

在定义中,“具有相同的特征”是指:如果你制作一个二维表格代表一个对象,并将实例填入表中,那么,对象便是从这些实例中抽象出来的;同样,“都服从和遵守相同的规则”是指:你可以对对象数据定义

简单的操作,且容易理解,不易出错。

由上述可知,对象是现实世界中具有相同属性、服从相同规则的一系列事物的抽象,也就是将相似事物抽象化。其中的具体事物称为对象的实例。

例如:客车可视为一个对象。它具有属性,如制造厂家名称、类别、系列号、乘客容量等。

定义2 对象的属性是实体所具有的某个特性的抽象,而实例本身被抽象成对象。

属性是对对象某个特性的抽象。获得对象的一组属性的目的在于:

- (1) 完整性。属性反映了所定义对象的全部信息。
- (2) 完全分解性。每一个属性只反映对象这个抽象体的某个方面的信息。
- (3) 相互独立性。属性具有相互独立的取值。

属性的表示方法一般采用两种形式:表格表示形式和文本表示形式。

属性的种类有:描述性属性(对象的内在特性);命名性属性(任意的名字或标识符);参考性属性(将一个对象的实例与另一个对象的实例联结起来的事实)。

定义3 关系是指两个或多个对象之间具有的相互联系的属性。

关系是现实世界中不同种类的事物之间所具有的有规律性联系的抽象。关系是根据从现实世界中的实体抽象而来的,根据参与联系的对象来表述的。大多数的关系也可以用被动语态来描述。

信息模型是面向对象分析的基础。信息模型由问题领域中的对象所组成;根据对象、属性及对象之间的关系来规范化问题领域中的信息。信息模型的

基本思想是描述对象、对象的属性和对象之间的关系三个内容。

信息模型用两种基本的形式来描述,一种是文本说明形式,包括对系统中所有对象、属性、关系的描述说明;另一种是图形表示形式,它提供一种全局的观点来考虑系统中的相关性、完全性和一致性。

二、面向对象的程序设计语言

程序设计语言支持对象的概念是十分重要的。这种概念是让程序员自己去定义解空间对象,首先描述对象及其行为,然后用传统的方法实现。例如,数据堆栈可以作为一个对象,它必须有一个栈体存放数据,至于栈体中存放什么类型的数据,并不是堆栈行为的主要特征。面向对象提供了一种机制,使得私有数据有其私有操作,即描述这个对象(数据)的独特行为。这样就向着减少语义断层的方向迈进了一步,在某些问题上可以直接模拟问题空间的对象。因而,写出的程序易于理解和维护。✓

在结构化程序中是从小到大的构造系统,即是说对程序中的函数、过程逐一定义,然后得到总体结构程序,而在面向对象的程序设计中,则是针对某一对象首先把主程序运行起来,然后对各个过程、函数等进行定义,或者在某一环节预先不给予定义,而是根据情况等到程序执行到这一环节才给予定义。

历来传统的过程语言(从汇编语言、FORTRAN一直到Ada语言)都是以强调调作为主,而面向对象的程序语言则提供了以描述对象的数据结构为中心构成系统的机制。在面向对象语言中是没有“主程序”这一概念的,其程序的执行过程实质上是一个“见机行事”的过程。当需要对某种类型的数据进行处理,就对这一类的数据进行操作,但处理的结果有可能是另一种类型的数据。✓

面向对象的程序设计语言支持对象的概念可理解为:对象是一个封装数据和操作的实体。数据描述了对对象的状态,操作可操纵私有数据,改变对象的状态。每当其它对象向本对象发出消息而本对象响应时,其操作才得以实现。✓

自从80年代Smalltalk语言及其环境向计算机界推行以来,面向对象的程序设计技术引起了计算机界的极大关注,目前已取得丰硕成果。

Smalltalk是一种典型的面向对象的程序设计语言。此语言采用的面向对象的程序设计思想被认为是在数据抽象方面的一个变革性的进步。Smalltalk采用面向问题对象的程序设计范式来求解问题,它以对象和对象上的操作来描述软件系统,使其符合

人的一般思想规律和自然语言。

对象是Smalltalk语言的基本建筑块,相当于其它语言中的数据。对象是自描述数据结构,本身不能做任何事情,必须通过向对象发送消息,对对象进行处理。消息类似于其它语言中的函数调用。

在Smalltalk中,对象和消息构成了它的求解问题基本执行机制。对象原始数据,仅能通过消息起作用,储存在一个对象内部的数据也只能通过消息来存取和共享。

Smalltalk-80语言是80年代初研制出来的,尽管它还不完善,但它的语言思想形成了今天的面向对象系统的雏形。✓

目前,面向对象的语言种类繁多,发展迅速。在人工智能领域,由Flavors语言产生了普通的Lisp对象系统(Common Lisp Object System),简称CLOS。Pascal作为面向对象语言的基础,除了在Pascal基础上实现面向对象的概念——类之外,还产生了Clascal。最近又形成Object-Pascal。而Stepstone公司开发的Objective-C语言是一种预处理器,它可以标准C代码与Smalltalk混合起来,由AT&T贝尔实验室开发的C++语言是另一种产生C代码的预处理器。最近从C派生出来的面向对象的语言是NextStep。

Eiffel是一种相对新的面向对象的语言,它可以更有效地支持软件工程。

可以说,除了Eiffel语言以外,目前大多数面向对象的语言都起源于某种非面向对象的语言。由于面向对象的程序设计语言仍处于发展之中,因此,现有的多种面向对象的语言,没有一种是标准的。这方面的研究仍在继续。

二、面向对象的数据库技术

近年来,面向对象的数据库技术的研究也在不断深入,并取得了很大进展。在国外,已有商品化的面向对象的数据库管理系统投入使用。如美国TRW公司开发的PMDB(Project Master Data Base)。

目前提出的面向对象的数据模型是一种语义关联模型。其基本组成单元是数据对象,对象是现实问题中的一个实体,类似于关系模型中的实体。面向对象的数据模型用属性描述对象的特征,并用对象这个抽象的概念来表示任何存储和接受信息的单元,并指定某个属性(或多个属性组合)作为对象的标识符。对象反映用户的观点,对于不同的用户,对象有其不同的抽象级别。这样,面向对象的数据库中的数据就是一系列对象、对象的描述信息(即对象的属

性)以及对象之间的联系。对象是面向对象数据库的组成部分,它代表了与一个项目有关的各种数据类型,如产品、资源、人等。对象可以实例化,即每个实例表示某个对象的一个不同的事件。如实体类:人,作为一个对象,其属性有姓名、年龄、性别、……。属性描述对象的性质和提供关于对象的附加信息。每个对象有一组属性与它相联,一个对象的每个实例对那些属性具有不同的值。属性包括历史属性、种类属性和存取权属性。历史属性记录了对对象的产生方式及所有与该对象的产生有关的信息,这个属性为配置控制提供了基础。种类属性记录了对象所存放信息的种类。这主要是用来防止外层工具对该对象进行格式或内容不匹配的存取。存取属性指定了对对象的存取权。

在面向对象数据库中,对象之间的关系可概括为归纳关系、组合关系、关联关系。归纳关系反映一个对象类与若干个互不相容的子类之间的分类关系。组合关系反映对象之间的构成关系,对象实体可利用组合关系加以描述和抽象。关联关系反映对象之间的相互依赖、相互作用的关系。关联关系与关系模型中的关系是同义的。在某种意义上讲,面向对象的语义关联模型可以看成是关系模型的发展。它是在关系模型、网状模型和层次模型的基础上发展起来的。

我们知道采用合适的数据模型来描述数据无疑会提高数据库的功能和效率。面向对象的数据管理系统可以将数据和应用程序结合起来,显示出它的特点,应用于复杂环境。在面向对象数据库中面向对象模型与语义数据模型都以对象为中心,并把面向对象的语义关联模型转化成实体的结构,为所研究的问题领域提供适当的抽象层次,减少了数据模型和数据库设计语言的差别以及数据模型和应用程序之间的语义差别,满足了目前新的应用环境的需要。如前面提到的面向对象的数据库 PMDB,该数据库采用 ER 模型,具有 31 个对象,220 种属性和 170 种联系,可应用于项目数据库(Project Database)的开发。专家们认为面向对象的数据库的提出及应用,将会为数据库技术的发展提供美好的前景。

三、面向对象的系统分析方法✓

面向对象的分析方法是在 80 年代初提出的。虽然对象和操作并不是程序设计的新概念,但是面向对象的系统分析与设计方法却是一种新的方法。面向对象的设计方法是近年来发展起来的,它强调了系统设计之前的系统分析,强调以系统中的数据或

信息为主线,全面、系统、详细地描述系统的信息,建立系统的信息模型,指导系统的设计。从某种程度上讲,它是一种数据驱动的系统分析和设计方法。结构化分析方法是目前系统分析的主要方法。如面向外理的功能分解法、面向控制的事件响应法等。从本质上讲,面向对象的分析方法也属于结构化分析方法。它是一种面向数据的分析方法。

近年来,随着计算机技术的普及,诸如数据库系统、专家系统、实时过程控制系统、系统软件、自动代码生成、人工智能、计算机集成制造系统等变得愈来愈大,愈来愈复杂,从而导致了开发上的困难性、复杂性和不确定性。面对这些新的复杂的问题,传统的系统分析与设计方法,由于它们不能很好地正确反映和组织系统中的所有信息,从而不能满足分析和设计的要求。自从面向对象和数据驱动的思想提出以来,人们对面向对象的系统分析与设计方法产生了极大的兴趣,从而展开了对面向对象的系统分析与设计方法的研究,并把它应用于各个领域。

我们知道在软件开发过程中,系统分析主要完成三个目标:

- (1) 确定各种处理、操作和算法。
- (2) 确定施加于系统中的信息或数据上的操作。
- (3) 确定各种处理、操作的执行顺序,即确定系统的总体控制。

面向对象的分析方法首先定义对象及其属性,也就是从问题领域引申出数据。用状态模型描述对象的生命周期。建立状态模型的依据是问题域中的操作规则和内在性质,生命周期反映了系统的控制问题,并根据对象及其生命周期定义处理过程。面向对象的分析方法主要考虑:数据(信息模型)、控制(生命周期)和对象处理。采用面向对象的分析,其目的在于有效地描述与刻画问题领域的信息和行为。它实现这样一种描述,必须以一种全局的观点来考虑系统中的各种联系,包括系统的完整性和一致性。同时,这种描述能够说明系统中各种操作的细节。为了达到面向对象分析的目的,面向对象的分析提供了二种形式的模型:信息模型、一系列状态模型和一系列处理模型。这些模型也曾在软件工程中的其它分析方法中使用过,但是,在面向对象的分析中,这二种模型被有机地结合起来,它们相互影响,相互制约。面向对象的系统分析提供了一种独特的方法,它反映了软件问题的本质所在。软件问题的本质在于数据处理。面向对象的分析是一种以数据或信息为主线,数据和处理相结合的方法,这种方法主要是以建立系统的信息模型为基础的。

集成理论与电子窗口 相结合的新一代数表软件探索

安徽省经济信息中心 杨传杰 黄效国

摘要 自从计算机问世以来,软件开发技术经历了机器代码、汇编语言、高级语言、软件工程等几个阶段。但由于新的应用领域不断涌现和计算机应用的深入,对现实世界模型的描述变得越来越困难。再加上,软件工程技术本身的复杂性以及这些技术在现实应用过程中面临的实际困难,从而导致了“新的软件危机”,促使人们寻求新的途径,本文将简要阐述集成化理论与电子窗口相结合所形成的新一代数表软件。

一、概 述

由于社会的发展和进步,计算机已渗透到国民经济的各行各业。因此,对计算机软件的需求量成百倍地增加,原有的软件开发理论和技术已无法适应日益增长的社会需求,所以,软件开发迫切需要新的理论和新的技术。近年来,随着人们对软件内外都在

进行“集成软件”的尝试,力求软件开发有新的突破。目前对集成软件已有多种理解和解释,一般可把集成软件方法归纳为三个方面:

(1)不同工具的集成软件,该方法是把各种各样的软件工具所开发、操作、测试的软件产品信息存储在一起,并连接起来。

(2)不同技术的集成软件,把已发展成熟的软件

目前,面向对象的分析方法仍处于研究和发展阶段,而且逐渐被应用于由软件、硬件构成的各种应用领域。这种方法之所以具有强大的生命力,主要是因为它从以下几方面致力于改善软件产品的质量,缩短软件研制的周期。

(1)可修改性。面向对象的软件系统很容易增删软件对象而不影响系统的原来结构,使用户像把新的硬件插入计算机一样来扩展系统的功能。

(2)可重用性。创建新的对象时,应能继承现有对象的某种功能,而不用浪费额外的时间和存储空间去重复以前的工作。

(3)可移植性。对象可以从一个系统移到另一系统,而且在大范围内的软件产品中可重用。

(4)可靠性。由于对象是构成系统的基本软件单元,每个对象作为一个独立的单元被测试,而且易于保证其可靠性,从而可以保证整个系统的可靠性。

(5)兼容性。面向对象的系统与传统的软件是兼容的。

从目前应用上看,虽然面向对象的分析方法为实际问题建立了一个可用软件实现的模型,利用对象提供了一种描述信息的机制,是一种独特的方法,

但其抽象层次较高、思维方法与传统方法有所不同,其概念不易于掌握,特别是熟悉了过程型程序设计的软件人员,对新的又非一致的分析方法不习惯,因为它不如传统的方法那样易于为人们掌握,所以不能很快地普及。尽管如此,面向对象的分析方法以其独特新颖的设计思想、抽象机制和优点,必将成为计算机技术不可分割的一部分。采用这种面向对象技术,在软件开发、人工智能研究及其它领域中必将显示出巨大的优越性。

进入90年代以来,面向对象的技术研究进入了一个新的阶段,面向对象的技术已成为广大计算机工作者感兴趣的课题,面向对象的程序设计语言如 Smalltalk、C++、Eiffel 和 NextStep 等已逐步推广应用;面向对象的数据库系统已产品化(如 PMDB 等)并逐步投入使用;面向对象的系统分析方法,已从研究阶段转向应用,随着研究和应用的不断深入,面向对象的分析方法必定会逐渐成熟,并广泛应用。可以预见,面向对象的技术很可能成为90年代中后期的最重要的软件技术,它必将为广大软件人员所接受,有着非常广阔的应用前景,成为面向二十一世纪的重要技术之一。

技术集成在一起使用。

(3)理论和技术的集成软件:该方法是把计算机理论研究、软件开发的形式化方法与各种先进的技术集成在一起。

二、新一代数表软件的主要思想和研究内容

新一代数表软件所采用的“集成软件”思想,就是指相关的软件技术、相关的软件工具、相关的软件方法的高度集合,从而能获得下述三大优点:

(1)直接面向应用中的问题,不需开发者了解过多的软件技术问题。

(2)软件人员的劳动强度、劳力密集度得以降低,开发率得以提高。

(3)在软件开发和应用过程中,可大量减少各环节的错误,提高软件的可靠性。把“集成软件”思想用于计算机语言方面,现正处于刚刚起步阶段。与此同时国内外的“电子窗口”理论和技术也逐渐成熟起来,国外非常流行,它有下列三大优点:

(1)用户界面友好,人机交互形式直观。

(2)操作简单,易学。

(3)智能化程度高,有记忆存储功能。

无论是“集成软件”还是“电子窗口”,它们都有各自的优点,同时也有各自的缺点。“集成软件”具有灵活、多变的特点,解决问题面宽,但是仍需编写程序。“电子窗口”使用简单,操作方便,但是解决问题面窄,灵活性差。把“集成软件”与“电子窗口”有效地结合成一体,便产生了新的软件理论和新的软件技术。它的主要思想模式是:

“集成软件”+“电子窗口”=集成化智能软件。这种思想使得两种不同类型的软件理论,在结构上、框架上、界面上和机制上紧密地统一起来,最大限度地人的智慧与计算机的功能结合起来,可形成各种符合当前社会各行业迫切需求的、通用的智能软件。

新一代数表软件的主要研究内容分三个方面:

(1)集成语言

程序设计语言就是一台“抽象”的机器,目前的语言通常是经过词法分析或语法分析进行工作,也就是说语言的词法分析或语法分析的技术已成熟。如何使研制的语言具有集成度高、灵活通用、使用简便、功能丰富、稳定可靠以及符合国际标准等项特点,其关键是在语言的语句灵活性和集成度上,也就是在语句的参数选择上。主要研究内容是:

①确定语句规则,按规则把一条语句分为标识

符、命令和参数三个部分。

②利用灵活多变的参数,解决已往集成软件功能不全、应用面窄的问题。

③利用参数的调整改变其语言的功能。

④确定语言的运行方式是:词法分析、语法分析、翻译语句和运行。

(2)电子窗口

该方面的最大特点是用户不必编制程序就能得到预期的结果,所以电子窗口是计算机应用软件技术近几年来发展较快的一个领域。短短的数年时间从面向用户的应用程序生成程序(代表性软件:电子表格软件 VISICALC,文字处理软件 WORDSTAR,数据库管理软件 dBASE,还有制图和排版软件等),发展到一组式计算机交互图形(代表性软件:嵌入式 Lotus 1-2-3,操作环境式 Microsoft Windows);窗口技术的实现为用户提供一种方便、易懂、易学、直观、高效、可靠的、符合国际标准的软件工具。

①向用户显示“菜单”和各种提示信息,以指导用户下步操作。

②具有记忆存储功能。

③具有联动式或联想式操作,使得操作智能化。

(3)智能软件

研究内容主要是:

①研究出一种新型的集成语言加电子窗口技术和理论。

②设计出一种用户界面极好又非常直观的智能化界面。

③使智能软件的接口技术符合国际标准。

④解决智能软件指针、参数、状态、地址等关键技术参数的传递。

⑤最大限度地采用“集成语言”和“电子窗口”两者的优点,使之有效地结合在一起,而无任何障碍。

三、新一代数表软件的预期目标与研究方法

预期达到的目标是:

(1)含有高集成度的语句,也就是说,一条语句或几条语句就能完成对一件事务的处理;解决那些需要重复操作或多变的事情;解决突发事件,即对事务进行实时处理;设有活口,允许其它语言和软件嵌套使用。

(2)窗口对各集成模块进行控制、协调;向用户提供联动或自动操作和智能化手段。因此,窗口具有学习掌握快、处理事务快、运行时间快,灵活通用性强,功能丰富齐全,使用简单和稳定可靠。

(3)智能化后的软件,既保留集成软件全部的功能,又体现出窗口软件使用简单的特点。

(4)能直接读写文本文件和 dBASE 数据库文件。对硬件无特殊要求,便于推广使用。

(5)采用超大规模集成芯片技术固化智能软件,提高商品化程度。

拟采取的研究方法以及技术路线是:

(1)利用软件工程环境的理论与方法,分阶段细化和实施,给出软件研制的统一的方法、统一的标准、统一的结构、统一的接口,使研究出的软件具有统一的规范。

(2)采用多种数学方法建立智能模型,该模型对集成语言与电子窗口软件研制十分重要:

①这能使软件更多地具有人类思维的优点。

②能方便对客观事物进行结构和行为的模拟。

③使问题的直接求解过程变得简单、直观。

(3)集成化理论与电子窗口技术相结合,是一对既相互对立、又相互统一的矛盾,将它们有机地结合,必须解决它们之间的接口问题,保证接口稳定可靠不会产生降低功能的现象,同时为用户提供一致的操作环境。

(4)优化电路板线路的结构,尽量减少集成电路芯片,使商品智能软件可靠、稳定,商品化程度高。

四、新一代数表软件的 预期功能和性能

1. 功能要求

(1)能方便地对文件进行管理,被管理的文件大小不受机器内存限制;并能直接读写 dBASE、Foxbase 数据库和文本文件。

(2)文件结构定义简单方便,可根据需要任意组合、搭配和修改,使文件的域变化具有动态。

(3)编辑处理,操作流畅,功能齐全;含有人工录入、机器自录、更新、计算、查询、替换、排序、移位、插入、删除等十多种实用功能。

(4)能生成各种各样报表,即:过录表、汇总表、分类表;表的宾栏可连续生成五层;并且,具有对生成后的表再加工处理或多次加工处理的能力。

(5)提供图形功能,可制作折线图、直方图、圆饼图。

(6)将集成语言与窗口技术溶合为一体,使得新一代数表软件具有灵活、多变、高效、准确的特点,操作简单方便,解决问题面宽,处理事务质量高、效率高。

(7)具有信息共享功能,可在网络上形成共享资源,适用于 OA 工程。

(8)具有数据保护功能,能有效地保护数据的安全和完整,在突发故障的情况下,能保证数据不受任何损失。

2. 性能要求

(1)易理解:对于计算机技术人员来说易理解数表卡各项技术要求,有利于进一步开发研究,对用户来讲,易于了解各种人机界面,操作简单、方便、直观。

(2)易运行:为用户提供各种屏口操作显示,帮助用户进行每个层次、每一步骤、每项功能的操作。在用户误操作时,系统拒绝执行,并有错误诊断能力,绝不会因操作失误而死机。

(3)可靠性与稳定性:系统能准确按用户的要求进行各项操作,不会因系统本身而产生误差;在运行中停电,系统自身的保护功能使数据不会丢失;当发生故障等意外时,系统自保护能按预期要求作出适当处理。

(4)效率:系统对外部环境要求低,运行速度快,适应机型宽,能有效地使用计算机资源。

(5)精度性:系统中计算数据处理、表处理和输入输出均能提供用户所需精度。

(6)灵活性:

①系统适应强,能满足各种类型的数表处理;

②系统应用范围广,能满足企、事业单位中处理各种数表;

③系统功能强,能满足动态的数表处理;

④系统面向用户,用户可不受限制进行“盲操作”。

(7)智能化:以接近人类通常思维方式建立系统,使系统更有类人思维和记忆的优点。

五、新一代数表软件研究的意义

(1)首次把语言和电子窗口的理论和软件结合为一体,提出新的理论和新的技术。

(2)让智能技术走向应用,形成通用型的数据处理商品软件。

(3)为广大计算机非专业用户和计算机专业人员提供方便、易学、高效的智能工具,彻底解决社会对软件需求量大,而软件研制开发周期长的尖锐矛盾。

(4)填补国内外无理想的通用性集成化数表软件的空白。

用面向对象的结构化方法设计 科技软件共享系统 SASS

copied 9/6/93

中科院计算中心

李 军 张绮霞

倪咏东 董学军

一、关于 SASS 的介绍

科技软件共享系统 SASS (Scientific Application Software Sharing System) 是纳入《中关村教育和科技示范网络(NCFEC)》计划中的一个应用项目,是为了更好地让科学计算工作者、科技应用软件开发者共享中关村网络上的科技软件而设计的。

SASS 是个网络应用系统,资源分布在网络中各个网点上,用户也分布在各个网点的不同机器上使用,且资源规模十分庞大(软件产品数十个,程序构件数以万计,源语句百万量级)。故 SASS 的开发设计存在以下关键之处:

- (1)资源的组织与存放
- (2)系统运行效率和响应时间
- (3)资源的分布式管理和系统的分布式使用
- (4)资源的安全保密性和系统的一致性
- (5)系统的可维护性和可扩充性

鉴于以上特点,考虑到系统本身的开发效率和可维护性,我们在面向对象方法 OOM 基础上,提出了一种综合了 OOM 与 SM(结构化设计方法)的面向对象的结构化设计方法 OOSD,并用 OOSD 进行了 SASS 的分析、设计与开发。

设计中,我们把 SASS 分为下列三部分:

(1)SB:Software Base——软件资源库,存放共享的软件资源。软件库中的资源分布存放在若干网点的宿主机上。SB 中以软件为基本独立单位,一般类型有:

程序库(library):功能较完全的、通用的子程序集合。如 STYR,INSL 等。

程序包(package):功能专门、适用特定领域的程序集合。如 Linpack。

批式应用系统:如有限元分析系统 ADINA。

交互式应用系统:用交互方式求解问题。如代数

公式处理系统 REDUCE。

解题环境:集成化的多任务系统。如数值软件开发环境 NSDE。

(2)SBMS:Software Base Management System——软件库管理系统,它对软件库 SB 实施有效的管理,提供对资源的查询、存放、维护、运行、安全、记帐等功能。

(3)SASS SERVER:SASS 服务器;用户通过服务器使用 SASS。有三种 SASS 服务器:Host server, Subsb-server, User-server,分别面向系统管理级、资源级和应用级。

二、OOSD 的基本原理介绍

60年代末随着“软件危机”的出现,计算机科学家们提出了程序的结构化设计思想(structured design),由此又推动了软件工程(software engineer)的发展与研究。结构化程序设计与软件工程结合起来,就是程序的结构化开发方法 SM(structured method)。SM 在开发各类软件中发挥了巨大作用,独领风骚近20年,影响了好几代程序开发人员。

OOM(object oriented method)则是在 OOPL 的基础上发展起来的。Simula 1 首先提出了“类”与“对象”的概念雏形;随后,SmallTalk 的推出,使“对象”概念走向成熟与普及。而今 OO 引起了人们越来越强烈的兴趣,以至于将以 OO 概念为核心的面向对象设计方法 OOM 看成是继 SM 之后的又一大里程碑,并认为它是90年代软件的核心技术之一。OOM 与 SM 相比,有许多相同之处,但他们之间也存在一些本质区别:

(1)从整体来看:SM 是自顶向下的。系统结构的“顶”描述了对各函数功能的控制,相应的“底”描述了对数据的定义和处理功能;而 OOM 从整体上说是自底向上的开发方法。

(2)从开发过程来看:SM 认为函数和过程是主

要的,数据是次要的。函数和与其相关的数据要么被认为是相互独立的,要么被认为是数据附着在相应函数之上;而 OOM 则认为,数据是主要的,函数和过程是次要的,函数附着在相应数据上,它将对现实世界问题的描述和解答看成是一组相应类对象之间的通讯,并通过对象的公共属性,遵循一定的规则及其上的操作来刻画对象。

(3)在 OOM 中各阶段的过渡是自然平滑的,其界限是模糊的;而在 SM 中这种过渡则意味着分析和设计基本表示方法之间的实质性转换/改变(如从数据流图到结构设计图的转换),使得转换难度随着系统的复杂度而急剧变化。

(4)从软件生命周期来看,SM 是传统瀑布模型,维护费用巨大;而 OOM 更像是喷泉模型,前期工作量较大。

由上看来,OOM 与 SM 相比有很多优越之处,但是否能完全取代 SM?答案似乎是否定的,国际上关于 OOM 与 SM 有许多争论,不过有一点不少人都赞同,那就是:把 OOM 与 SM 综合起来应用。这是因为:

(1)OOM 与 SM 都是人们考虑问题的一种固有思维方式,只不过 SM 更注重功能,而 OOM 更注重结构(形式)。

(2)OOM 注重对象概念,是面向问题空间的;SM 注重功能刻画,是面向程序解空间的。这实际上反映了人们认识问题的两个层次,相应地可将 OOM 与 SM 在更广泛的含义上进行综合:全局性采用 OOM,而在某些局部则采用 SM。

(3)OOM 与 OOP 本身并不十分成熟与普及,并缺乏大型工程实践的经验,而 SM 发展这么多年,已经有了一套形式化的工具,程序开发人员也较为熟悉,这从必要性上也决定了 OOM 必须与 SM 进行综合。

那么 OOM 与 SM 如何综合?答案有许多种,目前这方面也已有了一些初步的工作:如 P. Ward 将 SM 扩展以支持 OOM;P. Jalote 将 OOM 扩展以支持 SM。常见的综合一般有以下三种:

(1)S-O-O;SA+OOD+OOP

从 DFD 与 ER 图中抽取对象,例如把数据存储和数据传接器标识为对象,把处理过程变为对象的行为操作,这种强行匹配是可行的,但增加了复杂性。

(2)O-O-S;OOA+OOD+SP

用过程语言实现 OO 的设计。

(3)OO+SD:

综合了面向对象思想和结构设计思想;全局采用 OOM,某些局部采用 SM。OOSD 就是一种 OO+SD,称之为面向对象的结构化设计方法。

三、用 OOSD 来设计实现 SASS

1. 原因

主要是从软件的可重用性、可维护性和可扩充性来考虑。运用 OOSD 方法,大大提高了软件的可重用性。而 SASS 本身就是一个软件可重用性的课题,这是系统采用 OOSD 的主要原因。

2. SASS 系统的总体框架设计

整个 SASS 系统由用户接口 UI (User Interface)、软件资源库管理系统 SBMS (Software Base Management System)、软件资源库 SB (Software Base)三大部分组成。

软件资源库由树型结构的文件系统构成,入库的软件构件(包括源程序、文件等成分),以软件产品(相当于子程序库)为单位,存放在统一的逻辑结构中,有统一的命名约定,但在物理上是可以分布的,可分散在不同网点的宿主机上。

软件库管理系统 SBMS 基于从 SB 中加工出来的属性信息和 CAMS 及 Keyword 分类模式,响应用户接口的请求,对 SB 中的资源进行统一的管理,并提供资源的查询、存取、运行、维护和安全等功能。

用户接口 UI 提供系统与各种用户之间的界面,接受用户请求,输出相应结果。

3. SASS 中类的划分

3.1 软件资源库 SB

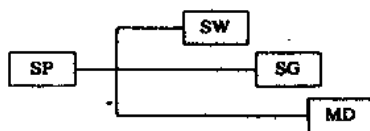
SB 存放共享的软件资源,下面主要说明组成及组织方式:

3.1.1 SB 的组成

软件资源库中,通常以软件产品为基本组成单位,我们称之为“软件”(Software),简称 SW,抽象出来为软件类:

(1)按产品结构分:SW 类下包括若干个“软件片”(Segment),简称 SG,每个 SG 下是一组“方法”构件(method)简称为 MD。

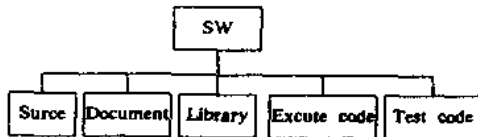
三个类关系如图:



三个类加以抽象,可统称为软件构件(Software Parts),简称 SP。

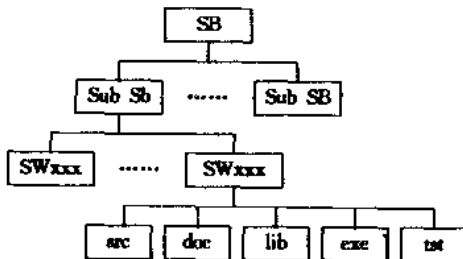
SP 用统一的标识符 ID 号来标识, ID 为 10 位整数, 前三位为 SW 编号, 4、5 位为 SG 编号, 6—10 位为 MD 编号。

(2) 按组成内容分: SW 通常由以下项或其中若干项构成:



3.1.2 SB 的组织

我们用一树状的文件系统来实现 SB 的存放。这里给出 SB 的逻辑组织结构:



SB 分布在网络上的若干宿主主机上(称为资源机), 各资源机上的资源构成资源子库 SubSB, SubSB 具有和 SB 相同的结构。

3.2 软件资源管理系统 SBMS

3.2.1 软件资源信息库 SIB(Software Information Base)

SIB 由两部分构成:

• SBMI (software base management information);

软件资源库管理信息。主要包括各种软件资源的分类模型以及从源程序和文档中加工整理出来的构件属性, 这是供管理 SB 用的信息。

• SYMI (system management information);

SASS 系统管理信息, 主要是供 SASS 系统管理用的信息, 如用户表、记帐文件等。这里我们主要讨论的是 SBMI。

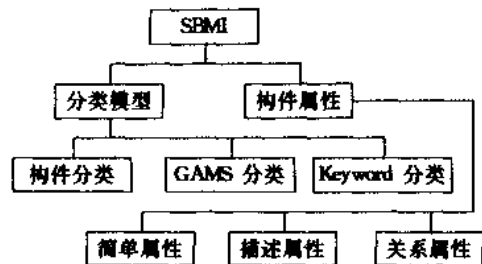
3.2.1.1 SBMI 的组成

3.2.1.2 分类模型

分类是将相类似的事物组织在一起的方法。

SASS 中采用三种分类模型, 一种是构件分类, 即按构件的原有结构: 软件→软件片→方法来分类, 第二种是 GAMS 分类, 第三种是按 keyword 分类。

GAMS 分类模式是科学计算领域的一个分类标准, 是一个面向问题的层次分类模型, 遵循科学计算



领域的惯例。第一层分为 19 个问题领域, 基本上覆盖了科学计算领域的各个部分, 对于每一个问题领域再细分, 最长达七层, 每次细分都是对问题领域的缩小, 最后得到一棵分类树, 树上每个结点都赋予一个标识符, 称为 GAMS 分类码 (gamscode), 分类码是交替出现的字母和数字, 指示从根结点到该结点的路径。

keyword 分类是 SASS 采用的另一种分类模式, 系统定义一套标准的按字典序排列的 keyword 分类表, keyword 也可层层细分, 最后得到一棵 keyword 树, keyword 树上每个结点也都赋予一个标识符, 称为 keycode, keycode 反映的也是 keyword 树上从根结点到该结点的路径。

SASS 中的分类模型是可扩充的, 可以根据要求再增添新的分类模式。

3.2.1.3 构件属性

构件属性刻画构件的可重用特性, 是重要的管理信息, 构件属性的选取是可重用软件构件库设计中极为重要的一环。基于本系统的目标和要求, 考虑到科学计算软件的特点, 我们反复推敲, 确定将构件属性概括为四个主要类:

简单属性: 又称为一般属性, 包括: 标识号, 标识名, 构件类型、数型、语言, 可移植性, 可复用性, 构件规模大小等属性, 对每一类构件: SW、SC、MD, 它们的属性略有差异, 如 SW 的属性中还包括了 SW 的组成、开放级别、管理水准等。

描述属性: 用以描述构件的简要信息, 包括构件名、版本号、构件功能概述、适用机型开发单位等信息, 构件的详细描述通过构件文档得到。

关系属性: 主要描述构件之间的相互关系以及构件与分类码 (Gamscode keycode) 的对照关系, 构件的调用与被调用关系 (usewhat, useme) 等。

分布属性: 又称为网点属性, 描述构件的分布信息, 主要是构件号 (ID) 与网点号 (netnode) 的对照表。

构件属性也是可以扩充的, 可以扩充新的属性类, 也可以在某属性类中增添新的属性。

3.2.2 SBMS 中的事件类 (SBMS 的功能)

SBMS 中对 SB 的管理以及对 SASS 系统本身的管理是通过对事件管理来实现的。事件类刻画了 SBMS 的功能。

SASS 中的事件分为几大类。为避免细节问题,这里主要介绍软件库操作与软件库管理所涉及到的几个主要事件:

查询(query)——查询资源的内容、功能、用法、属性关系和文档等。系统提供以下几种查询方式:

(1)按软件构成层次查询;

(2)按 GAMS 分类查询;

(3)按 Keyword 分类查询;

(4)按目标带约束查询,约束条件取自一个或几个属性项的条件规定。

提取(retrieval)——提取软件、软件片或方法,包括它的全部或部分组成,也包括所引用到的软件、软件片、方法。

装配(assemble)——提供连接装配用户应用程序的功能,产生源代码或可执行程序。

维护(maintain)——提供对软件库 SB、管理信息 SIB 以及系统的日常维护。包括构件的增、删、改以及对 SIB 的协调修改,负责对 SB 和 SIB 的完整性和一致性检查。

安全(secure)——负责对用户权限的监管和对软件不同开放级别的安全保护以及维护系统的安全。

记帐(accounting)——负责系统运行的日常管理和应用监督,作日记与帐本记录,输出统计表等。

事件类中操作也许十分复杂,这就要用到功能分解的思想,把复杂的操作分解成简单的操作。

(上接第32页)

Express 并称其为最新式的基于 Windows 的非编程数据库产品。虽然 Approach 不能再夸称最易使用的非编程数据库,但在某些特性上仍是独一无二。

Approach 的最佳特性之一是能以多种格式来直接、快速地存取数据。最初它只以本征方式支持 dBASE 和 Paradox 文件格式;2.0 版时又为更新本征索引文件(但不能建立新的本征索引文件)增加相应设施和对 FOXPro 的支持。此外,它还增加了全邮件合并功能及从 SQL 数据库查询输出结果这一非常见的功能。尽管 2.0 版具有以上优点,但易用性却不如以前。不过,主要的工作如果只是与单表的数据库打交道,Approach 胜任这类工作则是绰绰有余的。

下面简介其主要特点:

·容易的接口设计

SASS 中的事件类与 SASS 中别的类相比有很大差别,事件类中数据属性成分相当少,甚至没有,主要是以操作为主。

4. SASS 的界面设计——UI 的功能与实现

4.1 SASS 的用户分类

SASS 用户初步分为五级,由 userid 与 password 来识别:

(1)SASS 系统员——SASS 的特权用户,拥有系统的管理和维护的最高权限

(2)SB 管理员——是 SB 的特权用户,拥有维护 SB 的权限

(3)SubSB 管理员——在 SB 管理员授权下,有维护本 subSB 的权限

(4)注册用户——在 SASS 中注册的用户,可以查询、提取、装配 SB 中开放的资源

(5)临时用户——可以查询 SASS 中的资源

4.2 UI 分类

UI 分为二类三个接口:

(1)命令语言接口

①SSQL 语言及 SSQL 语言解释器:SSQL 是一种类似 SQL 语言的软件库查询语言,由解释器解释执行,提供给系统管理员及熟练用户使用。

(2)交互式接口

②图形终端用户接口:建在 X-Window 之上,以友好的多窗口和菜单驱动方式,提供给图形终端用户使用。

③字符终端用户接口:用 UNIX 中的 curse 库来实现,提供给字符终端的用户使用。

在 Approach 中定义新格式非常容易。利用跨越荧屏上部的工具条(toolbar)能快速地存取接口对象,以便进行表格或报告的布局编排。利用其它工具条图标(Icons)能使在设计方式和运行方式之间进行快速切换,以便能马上看到工作的结果。

当把一个字段放置的表格上时,Field Style 会话框会在屏幕上出现。该会话框提供一步式字段类型设定和字段显示,还可以为每个表格建立和保存约定设置。

·简单易用

Approach 的文档编排得很好,但由于它的设计直观,一般情况下,在线求助就可解决所有的问题。

一上所述,如果用户应用需求诸中,尤其是需要阅读和操纵众多的数据格式时,Approach 能以灵活而友好的用户工具而为用户所青睐。

原
书
缺
页

原
书
缺
页

原
书
缺
页

原
书
缺
页

•TEXT 和 VARYING STRING 型字段的第一个字符非空。

•SEGMENTED STRING 不转换。

•转换精度以 dBASE III 为准。

3. 总体设计

整个系统由三部分组成:

1) 命令识别与处理(含与 dBASE III 系统的命令接口)。

2) 通讯模块(由 PC 机模块和 VAX 机模块组成)。

3) 结果及错误处理(由 PC 机模块和 VAX 机模块组成)。

系统总体框图如图1。

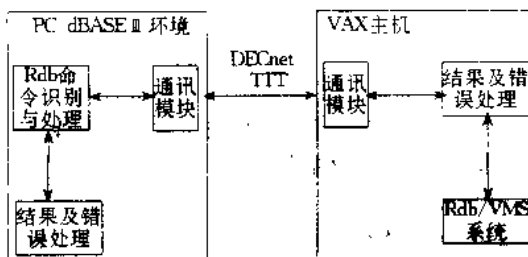


图1 系统总体框图

三、功能与实现

1. dBASE III 系统 Rdb 命令接口的实现

在 dBASE III 命令接收状态下,增加识别处理模块,对接收命令进行预处理,以区分 dBASE III 命令和 Rdb 命令并完成两者之间的切换。由于没有 dBASE III 源程序,故利用 debug 工具软件找到命令接收部分,再接合中断功能嵌入语言分析与处理模块。该模块由 C 语言编程,而接口通过汇编实现,利用驻留技术将两者同时驻留于内存。

语言分析与处理模块的主要功能见图2。

2. 通讯模块

这部分分别在 PC 机和 VAX 机上完成通讯初始化、建立通讯逻辑链路、命令与数据包的发送接收以及终止逻辑链路等功能。它采用网络技术,运用 DECnet 透明任务到任务通讯(TTT)功能,实现信息交换。通过通讯管理机制,视用户操纵 Rdb 的具体功能,系统地予以监视,以此提高运行效率,减少网上通信的系统开销。

3. 结果及错误处理模块

PC 机方面,完成结果及错误的显示与处理,以及两数据库之间的实时数据转换。

VAX 机方面,形成 RDO 提交批命令,将命令包交由 Rdb/VMS 执行,组织结果及错误信息传送包。

该部分由 C 语言编程,以模块方式调用执行。

PVIDB 系统除以上功能外,还提供独立的直接运行于 PC 机 DOS 系统和 VAX 机 VMS 操作系统之下的 dBASE III 与 Rdb 数据库文件静态转换程序。它由两个命令文件分别在 PC 机和 VAX 机上实现,操作简单方便。

dBASE III 命令接收状态

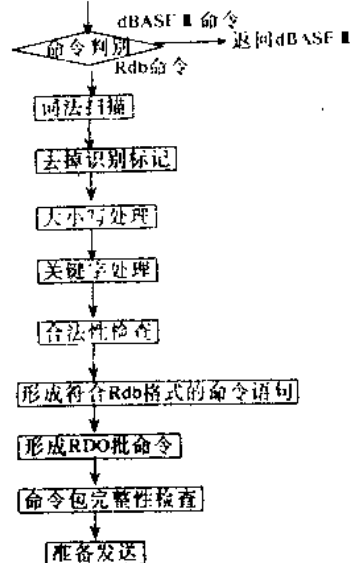


图2 语言分析与处理模块主要功能流程

CAD/CAM 基础与应用

大型讲座在京举行

计算机技术在 CAD/CAM 方面的应用在我国还处于基础状态,因此如何使企业能了解自身的技术状态,与国际国内相比,还有哪些差距,哪些优势,如何使 CAD/CAM 技术在我国有更广阔的市场,为解决广大企业和用户的渴求,机电部北京自动化所、北京宇思机电公司、中国兵器部工业计算机应用研究所、中国机械工程学会机械工业自动化分会、电子信息应用联合总公司等五单位联合举办了计算机、软件应用技术及开发技术、CAD/CAM 基础与应用大型讲座。

讲座内容十分丰富,有 CAD/CAM 技术的国内外最新动态;国内外流行的 CAD/CAM 支撑软件介绍和比较;有编程技术等。讲座期间邀请了国内著名的专家学者就 CAD/CAM 技术的最新发展、CIMS 技术做专题报告。在此期间教师与学员们共同探讨我国 CAD/CAM 技术发展中的实际问题。这种讲座形式深受广大企业界欢迎。因此以后还要不定期地举行。(淑)