

计算机辅助软件 engineering 研究

姜旭平 (清华大学经管学院信息系)

摘 要

计算机辅助软件工程(CASE)是目前计算机工业界和学术界研究的热点之一。本文在分析总结近年来CASE研究的基础上,以几个典型的CASE系统为背景,讨论了CASE的基本概念、结构框架、工具等,并研究了CASE从工具、到环境、进而成为一门系统开发方法学的演变过程,以及工具技术的发展对方法学研究的冲击和挑战。本文提出的观点是:CASE是一个广义的、发展中的概念;CASE是一个以工具和自动化开发环境为基础的方法学的概念;CASE工具是指一个不断丰富和完善着的软件工具的集合的概念;和CASE库是CASE的核心和围绕CASE库来展开CASE研究的概念,这些都是作者近年来研究总结的结果,可作为进一步开展CASE研究的参考,亦可作为实际开发适合我国国情的CASE系统时借鉴。

随着DEC、ORACLE、IBM等各大公司纷纷推出自己的计算机辅助软件工程(Computer Aided Software Environment, CASE)软件系统,九十年代初的软件市场异常活跃起来,各种CASE版本在各种不同类型的大、中、小、微型机上不断涌现,几乎主宰了软件市场。技术的发展同时也引起了学术界、理论界的高度重视,以Information and Software Technology 和 IEEE Computer 为代表的一些学术刊物也在大量征稿准备出CASE专辑。CASE究竟是什么?它有哪些特点?为什么会引起如此广泛的兴趣?下面结合笔者近年来的研究对上述问题给出一些粗浅的看法。

一、CASE的产生与发展

CASE是八十年代末期从计算机辅助编程工具、4GLs(第四代语言)以及绘图工具发展而来的大型综合计算机辅助软件开发环境。早先的CASE以工具和辅助开发环境的面貌而出现^[1,2,3],它以自动化的编程环境来取代原有的那些结构简单、功能较弱的开发工具(如PC-Base等等)。但随着技术的发展和人们认识的加深,CASE逐渐朝着从可进行各种需求分析,功能分析、生成各

种结构化图表(如数据流程图、结构图、实体关系图、层次化功能图、矩阵图)等演变成成为支持系统开发整个生存期的大型综合系统。CASE的概念也从工具发展成为一门方法学^[5],最为典型的例子是DEC公司的Digital Cohesion CASE^[6]和ORACLE公司的CASE-Method^[6]等等。目前CASE的发展已从支持结构化的约当方法^[14]、面向对象的杰克逊方法^[11]、原型方法到支持知识工程处理语言(如OPS5^[5]等)在内的大型综合软件开发环境。CASE目前的发展是一门工具和方法相结合的产物

二、CASE的结构

目前正作为开发环境工具发展的CASE是一个广义的概念,而不是特指某几个特定软件工具的组合,它没有具体的系统和结构。但就一个信息系统或软件工程的开发过程来看,它所需要的辅助或支持是有一定范围和共性的。为此要开发出一个支持这一开发过程的CASE工具,就必须有其最基本、最核心的功能结构。实际分析各家所推出的CASE软件后可以得出这样的结论:尽管它们形式多样,各有所长,但就其最基本的部分来说,却表现出一定的共性。

1. CASE的结构层次

CASE的结构是由应用软件开发过程必备的功能所决定的,所以支持从需求分析到系统分析、系统功能设计、数据库设计以至最后完成应用系统程序的自动转换(生成)就构成了如图1所示的CASE的基本结构。图中基础软件层是CASE的技术基础,包括关系数据库系统RDBS、4GLs、图形系统、以及从图形(主要是各种分析设计图形)到应用系统程序代码的自动转换。在这一层上,各家产品差别不大。建立在这个基础之上的是CASE内核层,由于这部分内容各家的命名和功能都不完全一致,为了叙述方便,我们暂称为CASE库。CASE库是整个CASE系统的核心,也是连接基础软件层和CASE工具层的纽带。在支持信息系统开发生存期中各种分析、设计、实现工作均由第三层即CASE工具层带到这一层次来完成,并对分析设计的结果建立数据库,产生相应的应用系统程序。所以这一部分是CASE最重要、最关键的部分。第三层是工具和应用系统层,对于这一层次,在各个公司推出的CASE中差别最

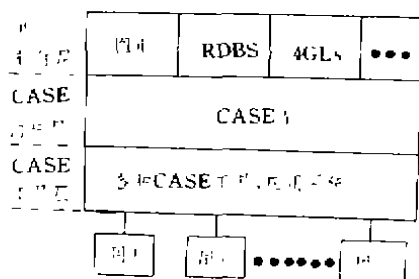


图1

大。它包括适用于系统开发不同阶段的各种窗口软件、分析设计工具、应用系统建立工具、数据库建立工具和一些专用的事务管理功能(Business Functions)软件包等等,它是CASE系统的对外接口。

2. CASE库

CASE库是CASE的核心部分,它在整个CASE开发策略中具有极其重要的作用。CASE库一般是传统字典系统的扩充和发

展,它不但保留了传统字典存储各种实体(主要指数据和数据定义,如字段、记录、关系、数据库、格式、程序和规则等功能,而且还增加了存储有关对象(Object)和方法(Method)的信息。这种对象是指任何与软件开发和信息管理有关的内容,如模型、编码、档案、图形、表格、文件等等;方法是指对一种或多种对象进行分析、操作、处理的形式,如创造、编辑、显示、综合、分析等等。

下面列举二个较为典型的CASE库来看一下其结构和主要功能。

(1) DEC公司的集成化CASE库

DEC公司推出的集成化Cohesion CASE-Engine中的CASE库(Digital Reposition, DR)是Cohesion CASE承上启下的核心部分,对上可支持围绕DEC数据库系统(包括它的分布式数据库系统)的定义、建立、应用程序原码的生成、应用系统的建立等等操作;对下可提供一个开放式接口和对所有工具都普遍适用的一系列服务,为用户创造了一个既方便灵活、又功能齐全的、能支持整个系统开发生存期的开发环境。

①DR的主要功能:

- 为系统开发者提供统一的结构框架。所有实体都在一个叫fashion的控制下统一管理,保证了定义的一致性。任何一个开发者都可自由地使用(如果没有权限、保密等要求的限制下)DR中的信息,从而提高了系统开发的质量和稳定性。

- 提供产生、转换、处理等支持系统开发的功能。它包括对数据和实体的检查、跟踪、设计能力、系统管理、效用分析以至最终形成应用系统的全部工作。

- 提供单个组织的数据库系统。通过各种外部的CASE工具,如决策支持工具、数据处理工具等,可帮助管理信息系统和决策支持系统开发者很方便地建立和使用一个复杂的实际应用系统。

②DR所支持的其它软件。DR对其下层

的CASE工具实现全开放式的支持。目前DEC的大部分软件工具都已与DR连接,而且还在继续开展这类工作,预计在不久的将来,DR将会支持所有的软件工具,为用户提供更加完备的软件开发环境。目前已连接的有: DEC design、VAX Rdb/VMS with SQL、VAX DBMS、VAX RALLY、RMS、VDA with DB2、VAX DATATRIEVE、VAX Link、DEC forms、VAX ACMS以及各种语言生成器等。

(2) ORACLE公司的CASE库

ORACLE公司推出的CASE*Method(C*D)中的CASE库是一个分布式多用户资料库。它可帮助系统开发者收集、管理、存贮系统开发过程中的信息,如数据定义、功能需求、设计分析、决策处理和实现细节。C*D在系统的内部自动产生统一的定义格式,并综合开发过程中的各种资料,对其结果进行分析、验证,确保系统开发工作的进行。C*D的结构和用户用C*D的关系如图2所示。

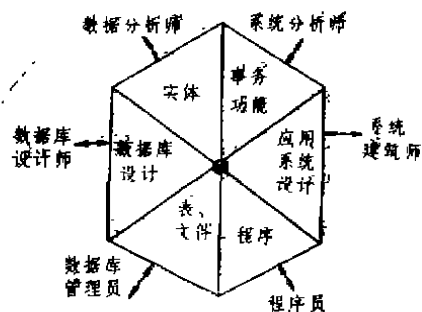


图 2

①C*D支持系统开发策略规划和需求分析的各个阶段,如各种需求分析工具,策略规划,功能分析,数据定义与数据流程分析等等。

②C*D支持以X Wondom标准建立的图形式多窗口的开发平台(platform)——CASE设计器(CASE*Designer),允许用户在这个平台上开发设计多种开发方法,如功能层次图、实体关系图、矩阵图等生成工具。

③C*D支持从各部分的分析设计到建立和维护应用系统的机器自动转换过程,也支持实际问题的最后求解,如图3所示。

3. CASE工具

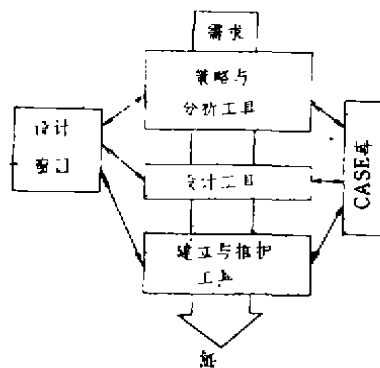


图 3

CASE工具是指CASE的最外层,即用户使用CASE去开发一个应用系统时所接触到的前述所有软件工具。这一层次的软件极为丰富且各家公司也不尽相同,各有所长。粗略地归纳起来大致有如下几类:

①图形工具。绘制结构图,生成系统专用图。

②屏幕显示和报告生成的各种专用系统。可支持生成一个简单的原型系统。

③专用检测工具。用以测试不一致的专用工具及其生成的信息。

④代码生成器。从原型系统的工具中自动产生可执行代码。

⑤文件生成器。产生结构化方法和其它方法所需要的用户系统文件。

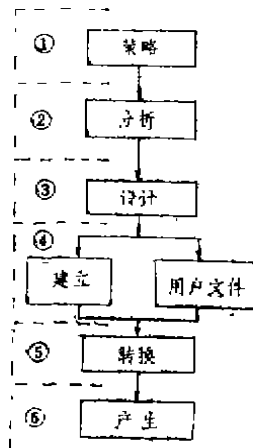
按目前技术的发展趋势,这一层次的大型化、综合化程度还在增加,有的甚至把以前研究的所有软件都与CASE库相联,归纳为CASE工具的范围,有关这一层次的内容在此不可能做全面的归纳。

三、CASE方法学

由于CASE工具对整个信息系统或软件开发过程的全局支持,引起了系统开发方法学领域从技术,方法到观念、认知体系的变化,方法学研究领域也受到了冲击和挑

战,从而使CASE演变成了一种独特的、以自动化支持环境为基础的系统开发方法学。CASE既支持自顶向下的结构化系统分析与设计的约当方法和SADT方法,也支持面向对象的杰克逊系统开发方法和原型方法。

CASE方法的系统开发过程和CASE工具对方法的支持如图4所示。传统的系统开发方法研究中的某些概念和界线在CASE方法中是模糊的、相容的。



①应用定义;功能层次;实体定义;实体图。②功能定义;数据详细定义;数据流程图。③缺省数据设计;数据库规模预测;程序/模块结构。④程序文档;数据库/文件定义。⑤更新现存数据库定义。⑥效果分析预测;重定义与重组文件。

图4

四、CASE的特点

与其它方法相比,CASE方法一般来说有如下几方面的特点:

- 强有力地支持软件/信息系统工程。
- 使结构化方法更加实用。
- 自动检测方法大大地提高了软件的质量。
- 使原型化方法付诸于实施。
- 简化了软件的管理和维护。
- 加速了系统的开发过程。
- 使开发者从繁杂的分析设计图表和程序编写工作中解放出来。
- 使软件的各部分能重用。

• 产生出统一的标准化系统文档。

• 加速了软件开发而且功能进一步完善。

五、关于CASE概念的探讨

本文结合笔者近年的研究谈了关于CASE一些粗浅的看法。笔者在工作中深深感到,就目前所定义的CASE来说是一个发展的、广义的概念,其中还存在着很多问题,需要作进一步的研究。下面提出几个问题希望与大家一起探讨。

1. CASE是一个发展中的概念。它从工具、环境发展到方法学,目前还在继续深入地发展。同时CASE也是一个广义的概念,是为了解决应用软件开发“瓶颈”问题,利用计算机系统自身的资源对整个软件开发过程进行支持,所以这就决定了CASE不会是指某一具体的系统。而且从发展的观点来看,今后也不会有一个确定的CASE模式。它的发展将会是随着人们对信息处理规律认识的不断深化和计算机技术的发展而不断得到丰富和完善。

2. CASE是一门全新的方法学。与自顶向下结构化方法和面向最终处理对象的原型方法相比,它更强调工具的作用、工具对方法的支持。CASE方法学的诞生,将会动摇传统方法学研究的某些基础,引起人们思想、观念上的某些变化。今后以CASE为基础的系统开发方法学的研究,将促使人们放弃从开发过程每一阶段实现的方便程度来研究方法的传统做法,朝着进一步研究信息系统自身的处理规律的纵深方向发展。

3. CASE工具是一个不断丰富和完善着的软件工具的集合。CASE的发展朝着全面的支持计算机在各个领域的应用发展。CASE工具已发展成为一个永无止境的,联系CASE库的软件工具的集合。根据CASE方法和系统开发策略的要求,CASE库正在联合更多的Compilers和软件开发工具。根据这一趋势的发展,今后的CASE工具将更加丰富多彩,各有千秋,而不会形成一个

基于类网络的OODB事务处理模型^{*})

李伟华 胡守仁 (国防科技大学计算机系)

摘

要

In the database field, transaction processing has long been recognized commonly as not only an advanced, but also a difficult topic. The transaction models for existing object-oriented databases(OODBs) are almost all extensions to, or borrowed from traditional transaction models. They captures only a few characteristics of OODBs. Our Contributions in this paper are three aspects. First, we take a new look at the relationship between database and its transaction processing. A database is viewed as an object's structure, and its transaction processing system as the method of this object. Second, we present a transaction model for an OODB—GOODbase, which is being developed by us on SUN workstation. The idea of this model can be applied generally to other OODBs. Third, we distinguish two kinds of concurrency in our model, namely, entra- and intra-object parallelisms. Some implementation issues are also considered here.

1. 引言

当前, OODB已成为数据库领域的热门话题, 并从实验上和理论上获得了高度重视^[1, 2, 4, 8, 10, 14, 16-20]。但是, 关于这种系统的定义还存在着不少争议, 有三点可以概括OODB初期研究阶段的状况: (1) 缺少形式化的定义, (2) 没有共同的数据模型, (3) 大量的实验性活动。这些问题同样存在于OODB事务处理的研究之中^[1, 4, 14]。

事务处理机制是数据库系统中广泛采用的控制程序并发执行的手段, 通常, 事务被看作数据库并发和恢复的基本单位。OODB比关系数据库(RDB)具有更丰富的语义和更强的数据模拟能力, 自然会包含许多可能固定的模式和确定的范围。

4. CASE库是CASE的核心。目前技术发展的趋势是所有的软件工具只要能与CASE库相连, 就成其为CASE工具的范畴。CASE

影响我们对事务认识的新特性。

1.1 面向对象数据模型概要

GOODbase^[10, 18, 19]使用的数据模型可以用若干概念来表征, 这些概念主要借自于Smalltalk-80^[5], ORION^[2, 4, 8]和[1]。

所有概念上的实体都模拟成对象, 每个对象都有一个唯一性标识符, 对象的状态存储在它的属性(或称实例变量)中, 属性的值本身也是一个对象, 对象的行为封装在它的方法中, 所以, 对象可以用来描述数据的结构和行为。由IS-PART-OF关系所联系的一组对象统称为一个组合对象。对象之间的通信是通过发送消息来实现的, 处理一个消息就是执行相应的方法。

SE库的发展不单纯是数据或实体集成的执行者, 而是整个开发环境中的驱动和控制力量^[6]。(参考文献共19篇略)

●863高技术及国家自然科学基金资助项目