

《微机组装与维修》教学模型的构建及实践

尚 晋

(重庆社会大学 重庆 401147)

摘 要 针对《微机组装与维修》课程,本文提出了 FMI 教学模式,即“框架→模型→实例”模式,认为不同的施教对象有不同的框架内容 F 和教学模型 M, M 作用于 F 产生实例 I,某实例集合的执行过程就对应着某主题的 FMI 教学过程。文中给出了 FMI 的定义,列举了 FMI 的特性,并阐述了 FMI 的实例构建过程。

关键词 微机组装与维修,教学模型,框架,实例,FMI

1 前言

针对《微机组装与维修》这门课程,许多作者提出了良好的见解^[1~6]。文[1]提出了基于 VR (Virtual Reality)的课件教学设计,强调教学课件不是电子黑板,应该包括电子教案、试题库的生成、实验教学仿真和实习训练仿真。文[2]提出了任务驱动教学法,力求以任务驱动,以某个实例为先导,进而提出问题,引导学生思考,通过学和做,培养分析问题和解决问题的能力。文[3]强调教学时注重方法和手段,区分层次和角度。

笔者在多年的课程教学活动中,试图建立一套适宜的教学模式,以适应不断更新的教学内容并满足不同层次教学对象的需要,FMI(框架→模型→实例)就是在这种背景下提出的。FMI 综合考虑了施教对象、教学内容和教学方法三要素,具有层次分明的系统结构,易于转化为电脑化的教学管理系统,实现教学资源共享;同时其框架主题结构具有良好的分类性和数据扩充性,可随时刷新数据,保证教学内容与时俱进。

2 FMI 教学模型

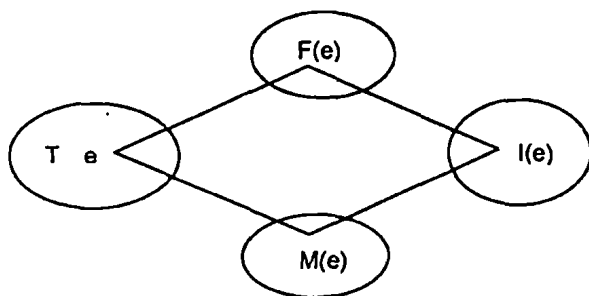


图1 MFI 教学模型图

2.1 FMI 的定义

FMI 教学模型如图1所示, T 是 FMI 教学方法

的总称, e 表示施教对象。因此 T(e) 就表示某个具体施教对象的 FMI。T(e) 用 F(e) 和 M(e) 进行描述, 用 I(e) 进行实现。F(e) 表示 e 下的课程内容框架, 用主题进行表达, 一个具有二级主题结构的框架定义如下:

$$F(e) = \sum_{i=1}^n \sum_{j=1}^m t_{ij} \quad (1)$$

其中 t_{ij} 表示第 i 个一级主题下的第 j 个二级主题, 可推广到多级主题结构。M(e) 表示 e 下的各种教学模型, 定义如下:

$$M(e) = \sum_{k=1}^q M_k(e) \quad (2)$$

主要的教学模型包括授课模型、实验模型和作业模型三种, 分别记为 M_T 、 M_L 和 M_E 。根据(1)和(2), 我们定义实例集合 I(e) 如下:

$$I(e) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^q M_k(t_{ij}) \quad (3)$$

它是 M(e) 作用于 F(e) 所产生的各种教学模型实例的集合, 其中 $M_k(t_{ij})$ 就是主题 t_{ij} 在教学模型 M_k 下的实例, 记为 $I_p(e)$, 关于该主题的所有实例集合就是关于该主题的 FMI 教学模型, 即:

$$FMI(t_{ij}) = \sum_{p=1}^q I_p(e) \quad (4)$$

2.2 FMI 的特性

(1) 参数 e 的变化使得 T(e) 适宜于多种施教对象的需要。例如, e 的取值范围可能是学生专业和学历层次的一个二元组, 即 $e = (\text{专业}, \text{学历})$, 专业包括计算机和非计算机, 学历包括本科、专科、成教等。e 的不同, 使得 T(e) 映射的 F(e) 和 M(e) 也不同, 最终导致 I(e) 集合的内容和大小不一样。就 I(e) 元素的个数而论, 本科总是多于专科, 专科总是多于成教。

(2) 参数 i 和 j 的自由取值保证了课程内容随时刷新。随着计算机的发展,《微机组装与维修》课程的内容经常需要增加和改变。数月推出一款

CPU、数周推出一款显示卡、数日出现一种工具软件、数小时发现一种病毒,要想把最新的内容及时补充到教学体系,其教学模型对应的信息系统就必须具备良好的可扩充性和友好的用户界面。

(3) 参数 k 的取值为多种模型教学创造了条件。一般情况是: $I_p(e) = M_T(e) + M_L(e) + M_E(e)$, 即主题的教学过程一般由授课、实验和作业三个环节完成。可能情况是: $I_p(e) = M_k(t_{ij}) = \phi$, 即某主题 t_{ij} 在某教学模型下为空, 表明并非所有主题的教学都要由全部教学模型来描述。

(4) $T(e)$ 是概念模型, 而 $F(e)$ 和 $M(e)$ 是逻辑模型, $I(e)$ 是物理模型。或者说, $T(e)$ 对应着客观世界的事物及其联系, 而 $F(e)$ 和 $M(e)$ 对应着信息世

界的实体及其属性, $I(e)$ 则对应着数据世界的记录及其数据项。通过 F (框架)、 M (模型) 到 I (实例) 的路径实现了 FMI 教学模型的物理化。

3 FMI 教学模型实例的构建

模型实例集合 $I(e)$ 的建立是 FMI 教学模式的核心, 一方面要保持模型中的实例处于“合用”状态, 即内容是最新的, 使用是方便的; 另一方面, 模型中的实例还应该具备: 表现形式要丰富多彩、内容难度要有层次。FMI 通过 $F(e)$ 的结构来保证数据刷新, 通过 $I(e)$ 的数据组织和管理来保证可操作性, 通过 $M(e)$ 的种类来展现教学风采。

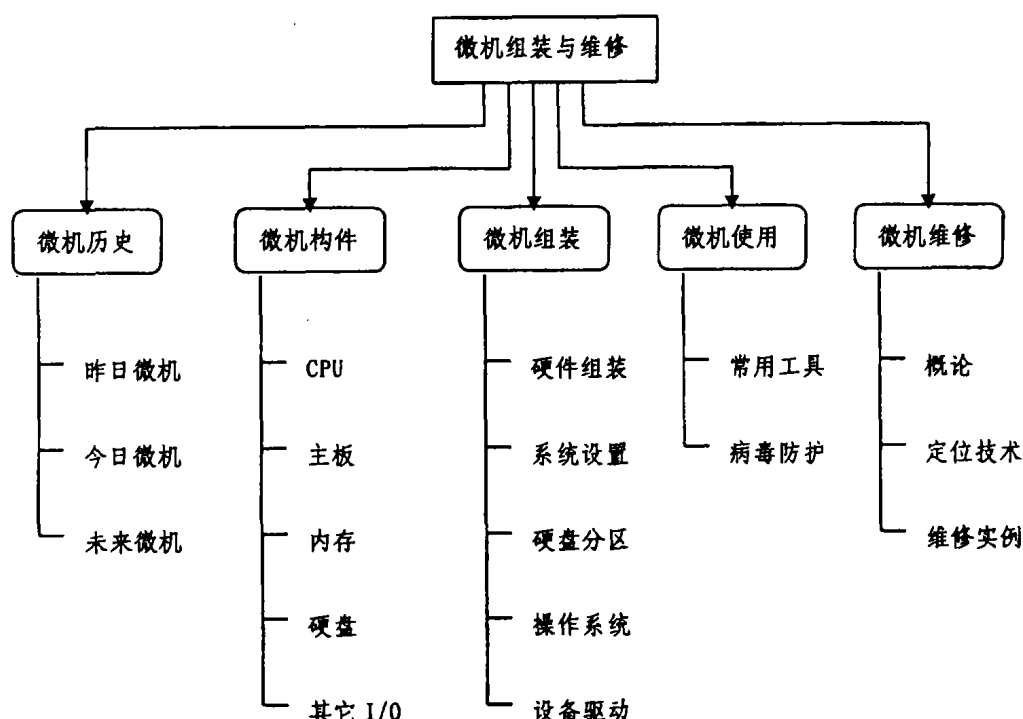


图2 课程框架图

3.1 $F(e)$ 的构建

根据对文[6~10]的研究和教学实践,图2描述了某施教对象 e 下的一个框架实例 $F(e)$ 。 $F(e)$ 包括微机历史、微机构件、微机组装、微机使用和微机维修五个一级主题,这五个一级主题的自然顺序符合人们认识客观世界的自然规律,也符合信息系统中面向对象的普遍法则:对象——属性——方法——事件。其中,“微机历史”涉及的内容主要是回顾微机的发展历史、了解微机的现状,以及展望微机的发展前景。“微机构件”就是将微机整体解剖成构件,再分析构件的各种属性:发展、原理、指标和安装,即了解构件的发展历史,掌握构件的工作原理,熟悉构件的性能指标,识别构件的接口和正确安装构件。“微机构件”是“微机组装”、“微机使用”、“微机维修”的基础,是该课程的核心内容。

3.2 $M_T(e)$ 的构建

授课模型 M_T 的构建以框架 $F(e)$ 中的一级主题为依据,举例说明如下。

对于“微机历史”这一部分,主要讲述微机的发展历程,作者认为应以讲故事的方式授课,也就是建立叙述型的故事模型,即“知识点→故事实例”模式。并且,考虑到学生的兴趣和特点,可以从今日的微机开始,追溯到昨日的微机,再展望未来的微机。可讲述的故事举例如下:伟大的 ENIAC、Bill Gates 的成长故事、世界上第一位程序员、Intel-x86 架构 CPU 光辉编年史、中国计算机产业发展大记事、IBM-未来的 PC 是这样的。要达到的教学目的是提高学生学习该课程的兴趣,并了解如下这些计算机之最:① ENIAC 是世界上第一台通用型计算机;② 计算机的鼻祖是巴贝奇分析机;③ 世界上第一台微机是

Altair8008;④世界上第一台通用型微机是 IBM PC;⑤中国第一台微机是长城 0520CH;⑥世界上第一台便携式电脑是 Toshiba T1100。

“微机组件”主要内容是全面了解微机的各种功能构件,其授课模型定义为: $M_T(\text{微机组件}) = \{\text{构件的发展背景,构件的工作原理,构件的性能指标,构件的选购,构件的安装连接}\}$ 。我们可以这样来理解:每种功能构件都有它自己的一段发展历史,并且是市场驱动的,是以时间为线索的。其次,每种功能构件都是一个子系统,可以用代数的形式来表达: $C = \langle X, R \rangle$,其中 C 表示功能构件的名称, X 表示该功能构件的组成部分,即组成元素, R 表示两重含义,一是 X 内各元素之间的关系,另一层含义是 X

与其它功能构件的关系,因此,一旦掌握了 $C = \langle X, R \rangle$ 这一表达式,也就理解了该功能构件的工作原理。第三,构件的性能指标,是功能构件外在表现的具体描述,主要从性能指标的名称、含义和特点、性能指标所使用的单位,以及一些示例共四个方面进行描述。表 1 列出了“硬盘”功能构件的性能指标授课模型。了解了构件的性能指标,构件选购就有了理论基础,这部分的授课模型定义为: $M_T(\text{构件选购}) = \{\text{要求或目标,性价比,假货鉴别}\}$ 。最后一部分是构件的安装连接,其授课模型定义为: $M_T(\text{构件安装连接}) = \{\text{构件的接口、信号连接、电源连接、与其它构件的关系}\}$ 。

表 1 M_T (硬盘性能指标)示例

框架名称:硬盘				
编号	性能指标	含义及特点	单位	示例或注意事项
25101	容量	存储数据的物理空间。	GB	容量 $X = C * H * S * \text{扇区内字节数 (B)}$ 工厂容量 $Y = X/10^9$ (GB) 电脑容量 $Z = X/23^0$ (GB)
25102	主轴转速	主轴电机的旋转速度。 决定了硬盘的档次。	RPM	5400RPM、7200RPM、15000RPM
25103	数据传输率	硬盘读写数据的速度。 外部传输率:从硬盘 Cache 读数据的速度; 内部传输率:将数据记录在盘片上的速度。	MB/s bps	UltraDMA/33:33.3MB/s UltraDMA/66:66.6MB/s Maxtor DiamondMax Plus60:456Mbps
25104	平均寻址时间	硬盘在盘片上找到数据访问地址所需要的时间。	ms	平均寻址时间 = 平均寻道时间 + 平均等待时间 IBM Deskstar 180GXP:8.5ms = 4.33ms + 4.17ms
25105	Cache	硬盘上的高速缓冲存储器。	MB	1MB、2MB
25106	接口	与其它功能构件连接的标准。		IDE、SCSI

“微机组装”包含了硬件组装、系统设置、硬盘分区、操作系统安装和设备驱动程序安装五大部分,五部分的总体授课模型可遵循“过程演示→要点分析→一般规律”来构建,即: $M_T(\text{微机组装}) = \{\text{过程演示,要点分析,一般规律}\}$,各部分的内容和形式也可有所不同。比如“过程演示”可以是一段详尽的录像,即视频(video)插件,“要点分析”包括正确的步骤、错误步骤、可能的后果和构件关联等方面的要素,“一般规律”是对某主题归纳总结的结果,用言简意赅的短语进行表达。

同样,我们可以构建 $F(e)$ 下其它一级主题的教学参考模型。 $M_T(\text{使用微机}) = \{\text{过程演示,要点分析,一般规律}\}$,它与 $M_T(\text{微机组装})$ 相同。 $M_T(\text{维修微机}) = \{\text{故障现象,故障分析,故障定位,故障排除}\}$,它是一个描述问题、分析问题、定位故障和解决问题的过程。

3.3 $M_L(e)$ 的构建

实验是本课程的重要环节,是训练动手能力,培

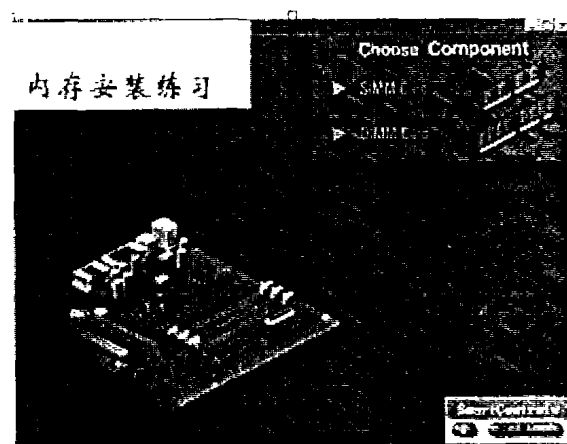


图 3 基于 VR 的 M_L (内存安装练习)

养创新能力的关键步骤。笔者认为,实验的基本目的是让学生“敢拆、敢装、敢修”,在树立学生信心的

基础上,启发学生的思维潜力,让学生的创造力和想象力得以发挥。该环节的教学步骤分为两步,一步是基于 VR 环境的练习,另一步是在实验室的实做。因此,实验模型 $M_L(e)$ 包含练习子模型 M_{L1} 和实做子模型 M_{L2} 两部分。

由于虚拟现实 VR 一般由计算机生成,包含文字、声音、图形、图像和三维动画等,给人身临其境的现实感觉,因此,它是练习子模型的良好工具,也是练习子模型构建的技术基础。基于 VR 的练习子模型定义如下: $M_{L1}(e) = \{\text{主题代码, 练习名称, VR 程序, 练习行为}\}$, 其中 VR 程序应包括 VR 界面、操作提示和练习评分等内容。图 3 是内存安装的练习模型实例,在这个 VR 界面中,首先要求选中 SIMM 或 DIMM 内存模块,然后在主板上选择相应的模块插槽,最后按正确的安装方法将内存模块插入相应的插槽。整个练习过程由实例程序进行操作提示,

并将练习成绩登录相应成绩库。

实做子模型 M_{L2} 的定义与练习子模型不同,它更注重实做准备和实做可行性,参照文[11],我们定义该模型为: $M_{L2}(e) = \{\text{实验目的, 实验内容, 实验要求}\}$ 。

实验模型应用时,一般先执行 M_{L1} ,再执行 M_{L2} 。例如,假设 Ghost 工具软件对应的两个模型实例为: $I_1 = M_{L1}(\text{Ghost})$, $I_2 = M_{L2}(\text{Ghost})$,则 $M_L(\text{Ghost}) = \{I_1, I_2\}$,即 Ghost 的实验教学需要运行 I_1 和 I_2 ,当然,如果 $I_1 = \phi$,则直接运行 I_2 。

3.4 $M_E(e)$ 的构建

构建作业模型 $M_E(e)$ 时应从两方面着手,一是 M_E 实例的内容,包括:题目编号、题目和题目答案共三项;另一是 M_E 实例的属性,包括题目类型、涉及的框架编号、难度系数等。表 2 列举了“硬盘构件”的一个作业模型实例。

表 2 M_E (硬盘构件)示例

框架名称:硬盘				
编号	题型	题 目	答 案	难度系数
25301	选择题	某 PC 中, IDE0 挂接 HDD1(主)和 CD-ROM(从), IDE1 挂接 HDD2(线选)。HDD1 分为 2 个区, HDD2 分为 3 个区,则这些 IDE 设备按 HDD1/HDD2/CD-ROM 排列的盘符顺序是: A、A:B:/C:/D:/E:/F: B、C:/D:/E:/F:/G:/H: C、C:/E:/D:/F:/G:/H: D、C:/F:/D:/G:/H:/E:	答案为 C。 在 PC 中,盘符顺序对应的设备顺序是: HDD/CDROM。而多硬盘的盘符顺序是 M1、N1、M2、M3、M4、N2、N3、N4。其中 M1、M2、M3、M4 是 HDD1 的分区, N1、N2、N3、N4 是 HDD2 的分区, M1 和 N1 是系统分区,其它是逻辑分区。	0.6
25302	选择题	某机器使用 40GB 硬盘,但在分区时大小不能突破 8.4GB,你认为,造成这种现象的主要原因是: A、操作系统版本太低 B、BIOS 版本太低 C、主板接口类型 D、硬盘制造技术	答案为 B。其原因是,在 BIOS 中,硬盘相关的中断服务程序 int13 的寻址能力有限制。BIOS 认为硬盘参数的返回值 CHS 最大可能为:16383、16、63,三者之积再乘以扇区内字节数就是 8.4GB。	0.4
25303	计算题	已知某硬盘驱动器柱面数 $C = 1024$,磁头数 $H = 16$,扇区数 $S = 63$,扇区内字节数 = 1024,请问该硬盘的工厂容量和电脑容量分别为多少 MB?	硬盘容量 $X = C * H * S * \text{扇区内字节数}$ $= 1024 * 16 * 63 * 1024 \quad (\text{B})$ 工厂容量 $Y = X/10^6 \approx 1057 \quad (\text{MB})$ 电脑容量 $Z = X/2^{20} = 1008 \quad (\text{MB})$	0.5
25304	简述题	对于采用 UltraDMA/66 标准的硬盘,其外部数据传输率理论上可以达到 66MB/s,你认为该硬盘的内部数据传输率至少应为多少 bps?	要想保证外部数据传输率达到 66MB/s,则内部数据传输率应该接近该值,将该值转换成 bps 即得到答案。计算过程如下: $66 * 2^{20} * 8 \text{bps} \approx 554 \text{Mbps}$ 。	0.4

4 FMI 的教学实践

本文提出的 FMI 教学模式正在实践和完善中。施教对象包括:重庆师范大学数学及计算机科学学院的 2000 级到 2002 级学生,重庆社会大学 1997 级到 1998 级学生,这些学生包括计算机专业和非计算

机专业,涉及本科、专科和成人教育三个层次。

FMI 教学模式是可电脑化的一个教学信息系统,可遵照信息系统开发的普遍规律来实现。在实施过程中,尤其要注意 FMI 中的代码设计,如框架的主题编号,模型的种类编号,以及实例的代码结

(下转第 48 页)

法,以提高 WCET 分析的可信度。

参考文献

- 1 Kligerman E, Stoyenko A. Real - Time Euclid: A language for reliable real - time systems. IEEE Transactions on Software Engineering, 1986, SE - 12(9):941 ~ 949
- 2 Mok A, Amerasinghe P, Chen M, Tantisirivat K. Evaluating tight execution time bounds of programs by annotations. In: proc. 6th IEEE workshop on real - time operating system and software, 1989.74 ~ 80
- 3 Puschner P, Koza C. Calculating the maximum execution time of real - time programs. Real - time systems, 1989,1(2):159 ~ 176
- 4 Shaw A. C Reasoning about time in higher - level language software. IEEE transactions on software engineering, 1989, SE - 15(7):875 ~ 889
- 5 Cousot P, Cousot R. Abstract Interpretation: A unified lattice model for static analysis of programs by construction or approximation of fix points. In: Proc. of the 4th ACM Symposium on Principles of Programming Languages, 1977.238 ~ 252
- 6 邹勇,李明树,王青.开放式实时系统的调度理论与方法分析.软件学报,2003,14(1):83 ~ 90
- 7 Park C Y. Predictable program execution time by analyzing static and dynamic program paths. Real - time systems, 1993,5(1):31 ~ 62
- 8 Alterbernd P. On the false path problem in hard real - time programs. In: ProcEuromicro Workshop on Real - Time Systems. L'Aquila, Italy, 1996
- 9 Pospischil G, Puschner P, Vrchoticky A. Developing real - time tasks with predictable timing. IEEE software, 1992,9(5):35 ~ 44
- 10 Puschner P. A Tool for high - level language analysis of worst - case execution times. In: proc. Euromicro workshop on real - time system, 1998.130 ~ 137
- 11 Vrchoticky A. Compilation support for fine - grained execution time analysis. In: proc. of the ACM SIGPLAN workshop on language, compilers, and tools for real - time system, 1994
- 12 Puschner P. Worst - case execution time analysis at low cost. Control Engineering Practice, 1998,6(1):129 ~ 135
- 13 Enghblom J, Ermedahl A, Altenbernd P. Facilitating worst - case execution times analysis for optimized code. In: ProcEuromicro Workshop on Real - Time Systems, 1998.146 ~ 153
- 14 Kim S, Min S L. Efficient Worst Case Timing Analysis of Data Caching. In: Proc. of the IEEE Real - Time Technology and Applications Symposium, 1996.230 ~ 240
- 15 Li Y S, Malik S, Wolfe A. Cache modeling for real - time software: beyond direct mapped instruction caches. In: proc17th IEEE real - time system symposium, 1996
- 16 Healy C A, Arnold R D, Mueller F. Bounding Pipeline and instruction cache performance. IEEE Transaction on Computers, 1999,48(1):53 ~ 70
- 17 Stappert F. Predictable pipelining and caching behaviour of hard real - time programs. In: procEuromicro workshop on real - time system, 1997.80 ~ 86
- 18 Stappert F, Altenbernd P. Complete worst - case execution time analysis of straight - line hard real - time programs. Journal of Systems Architecture, 2000,46(4):339 ~ 355
- 19 Theiling H, Ferdinand C, Wilhelm R. Fast and precise WCET prediction by separated cache and path analysis. Journal of Real time Systems, 2000,18:157 ~ 179
- 20 Colin A, Puaut I. Worst case execution time analysis for a processor with branch prediction. Journal of Real time Systems, 2000,18:249 ~ 274
- 21 Ottosson G, Sjödin M. Worst - case execution time analysis for modern hardware architectures. In: Proc. ACM SIGPLAN Workshop on Languages, Compilers and Tools for Real - Time Systems (LCT - RTS'97), June 1997
- 22 Enghblom J, Ermedahl A. Modeling Complex Flow for Worst - Case Execution - Time Analysis. In: Proc. 21th IEEE Real - Time Systems Symposium (RTSS'00), 2000
- 23 Puschner P, Burns A. Wiring temporal predictable code. In: Proc. 7th IEEE international workshop on object - oriented real - time dependable systems, 2002.85 ~ 91

(上接第 33 页)

构。举例来说,如果实例代码的结构类似于“ABCD”,其中 A 表示施教对象代码,B 表示框架主题代码,C 表示模型种类代码,D 表示实例序号,则关于某个主题 t_{ij} 的 FMI 教学活动 $FMI(t_{ij})$ 的检索就非常简单明了,只需要在 $I(e)$ 中提取代码为“AB * *”的实例子集即可。

结束语 学高为师,生正为范,在教学领域,由于学科不同,施教对象不同,因此可采用的教学模式也千差万别。让我们积极探索,潜心研究,以科学务实的教学活动适应新形势下的高等教育,培养出更多的新世纪创新人才。

参考文献

- 1 黄剑飞. VR 技术与教学课件[J]. 工程图学学报, 2001(增刊):285 ~ 287
- 2 张恩华. 任务驱动教学法初探[J]. 郑州铁路职业技术学院学报, 2003, 15(3):79 ~ 83
- 3 邵玉琳. 论计算机硬件维修教学[J]. 电子计算机与外部设备, 2000, 24(3):54
- 4 刘娜. 《计算机维修与维护》课的教学改革[J]. 机械职业教育, 2003(3):37 ~ 39
- 5 袁保社. “微机维修与维护”课教学内容探讨[J]. 高等理科教育, 2002(5):68 ~ 69
- 6 王金岗. 计算机维修实训教学的尝试[J]. 现代技能开发, 2002(7):91 ~ 92
- 7 张夷人. 微机组装与维护[M]. 北京:清华大学出版社, 2003
- 8 杨志翔. 微型计算机故障诊断与处理[M]. 重庆:重庆大学出版社, 2002
- 9 丁唐. 最新计算机组装与维修教程[M]. 北京:冶金工业出版社, 2000
- 10 封滢彦, 裴明德, 王世高. 最新电脑采购与组装完全手册[M]. 重庆:电脑报社, 2002
- 11 梁和. 微机组装与维修[M]. 北京:清华大学出版社, 2002
- 12 佟伟光. 微型机组装与维护实用教程[M]. 北京:高等教育出版社, 2003