

摘 要

自国家教育信息化逐步深入以来，资源建设成为信息技术教育应用关注的一个焦点。无论是为创设数字环境提供资源服务，还是进行自主探究协作学习，无一不以学习资源为中心展开。而学科工具，作为教育资源的重要组成部分，其设计和开发是当今教育资源建设的一个重点。学科工具具有生动活泼、直观形象、交互性强、可自主探索等特点，能够解决常规教学手段无法处理的问题。目前，国内外已经研究开发了各种学科工具，各自的着重点也不同，但总的看来，大多数学科工具的设计和使用上还存在着诸多问题，比如功能上比较独立、分散，不能很好地集成在一起使用，构件不易重用，构件之间的交互性差。

为了解决学科工具软件开发和应用中所存在的一些问题，使学科工具的在分析现存学科工具软件中存在的问题和国内外学科工具软件研究状开发变得简单、规范，提高构件的重用性，本文设计和开发了一个学科工具构建平台。首先，况的基础上，分析了学科工具构建平台的重要性；然后，基于软件重用技术，提出了基于领域知识的学科工具构件模型；接着，对平台进行了设计和实现；最后，以物理电学学科工具为例，对平台的可用性及其性能进行了验证。

关键词：学科工具；虚拟实验；领域构件；教育软件；计算机辅助教学

Abstract

With the fast development and broad applications of information technology, the education resources construction have become a focus of the information-based education. Subject Tool, as the important part of educational resources, play an important role. Subject tool have vivid, keep view characteristics and interactive mechanism which make it usefull in teaching when we are in difficulty with normal teaching method. Currently, there are many subject tools at home and abroad, and the emphasis are various. But, the problems in the existing tools are obvious. Owing to the lack of interactive mechanism and relation between different functions, subject tools can't work together to make an integrity application.

In order to settle above problems, make the design and development of subject tool easy, and make subject tools reusable. This article designed and developed a Platform for the constructing of subject tools. Firstly, analysed the existent problem in the existing subject tool software with the foundation of domestic and international subject tool software research. Secondly, put forward a subject tool model according to realm knowledge. Thirdly, carried on the design and implementation of the platform. At last, take the physical electricity subject tool as an example, carried on the function verification of the platform.

Key words: Subject Tool; Virtual Experiment; the Domain Component; Educational Software; CAI

独 创 性 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东北师范大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：马相春 日期：2007.6.1

学位论文版权使用授权书

本学位论文作者完全了解东北师范大学有关保留、使用学位论文的规定，即：东北师范大学有权保留并向国家有关部门或机构送交学位论文的复印件和磁盘，允许论文被查阅和借阅。本人授权东北师范大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存、汇编学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名：马相春 指导教师签名：钟绍春
日 期：2007.6.1 日 期：2007.6.1

学位论文作者毕业后去向：

工作单位：东北师范大学 电话：85683368

通讯地址：东北师大国家烟草学院9楼 邮编：130024

第一章 引言

1.1 学科工具软件的发展历程

20 世纪 50 年代以后，计算机辅助教学从传统的教学方式中脱颖而出，是教学史上一次伟大的创新。在这场变革中，两方面的因素发挥了重要的作用：一是现代心理学关于人类学习的研究的新理论和新方法的出现，二是以计算机技术为主的现代电子技术科学的兴起与发展。

20 世纪 90 年代以后，国外计算机辅助教学的研究和应用有了更加突出的进展，无论在人们对计算机辅助教学的认识、观念方面，还是计算机辅助教学理论研究和手段上都出现了较多的新动向和新发展，其中主要的可以概括为计算机辅助教学基础理论的发展和计算机硬件和软件技术的进步。早期的计算机辅助教学主要以行为主义学习理论为理论基础，但随着心理学尤其是认知心理学研究的发展，人们逐渐认识到了行为主义理论的局限性和片面性，并开始将目光投向强调个体的内部心理过程的认知理论。认知理论的引入为计算机辅助教学注入了新的活力，使得新一代计算机辅助教学方式更具灵活性和适应性，为学习者提供了更加生动、自由的学习空间，激发了学习者的积极性和主动性。认知理论从某种程度上弥补了行为主义理论的不足，拓宽了计算机辅助教学原有的个别化以及交互性的特色，将计算机辅助教学引入了一个新的高度。计算机硬件是计算机辅助教学得以产生、发展的关键物质基础，它的发展和进步也直接影响着计算机辅助教学的发展进程。

在理论、计算机硬件和软件已有了长足进步的前提下，计算机辅助教学发展到了智能计算机辅助教学阶段，这是以认知科学和人工智能为基础的一个新的阶段。它可以更加逼真地模拟教师的教学和指导行为，直接面向个别化的学生，成为一个综合应用教育心理学和教学法知识的全新的计算机辅助教学系统。它允许学生与计算机进行较为广泛的交互活动，而且可以根据学习者当时的状态与教学策略的要求进行的灵活的实时交互活动。ICAI 通常具有模糊化诊断功能(称为学生诊断模块)，不仅能简单反馈学生解题过程中的实际情况，还可以根据学习者的特点，设置自主学习，及时有效地针对学生的学习状态，来引导学生完成整个学习过程。这就使计算机辅助教学摆脱了以往的生硬而僵化的固定步骤，变得更具灵活性，更容易为学生所接受。

目前国外的学科工具种类繁多，根据这些系统在功能上的差异大致可分成以下六种类型：问题解答型、模拟型、人工智能型、计算机会议型、计算机网络型、简易型。计算机辅助教学软件，根据不同的用途可以细分为下面几类：教学测量软件、辅助自学的软件、辅助课堂教学的软件、辅助实验的软件、智力性游戏软件。

我国自 90 年代初期起，开始探讨学科教学工具软件的研究和设计。1998 年，中科院院士基于自动推理的教育软件《几何专家》、《立体几何》和《解析几何》推出，这些软件使老师们耳目一新，引起师生广泛兴趣。后来随着研究工作的深化，以及和第一

线教师的充分交流，张景中感到非常有必要在原有软件基础上做进一步优化，并根据我国数学课程改革的实际需要，吸取美国优秀教学软件《几何画板》的长处，给原有软件增加了大量新功能。2002年，代表最新成果的《Z+Z 智能教育平台-超级画板》诞生了。

《超级画板》易学易用，功能强大。用它能快速、精确地画动态几何图形和各种与参数动态关联的曲线，并具有跟踪、轨迹、动画、变换、测量、推理、公式编辑、图表绘制、符号演算、课件制作演示以及编程环境等多种用处。该软件一经推出，好评如潮，数学教育领域人士都对其表现出浓厚的兴趣。为了使这项成果能为尽可能多的师生所应用，

《超级画板》推出的免费版本也能满足中小学数学教学的需求，功能不亚于国外同类软件中的商业版本。

1.2 信息技术与课程整合对学科工具软件的需求

自国家教育信息化逐步深入以来，资源建设成为信息技术教育应用关注的一个焦点。无论是为创设数字环境提供资源服务，还是进行自主探究协作学习，无一不以学习资源为中心展开。学科工具，作为教育资源的重要组成部分，其设计和开发是当今教育资源建设的一个重点^{[1][2]}。

教育信息化过程中，现代信息技术被大量应用于教学过程中，但如何将这些先进的信息技术融合进教学中，实现真正的信息化教学，是当今课堂教学改革的关键所在，而大量信息化学科工具在课堂教学中的应用，是信息技术与课程有效整合的具体体现，为教学信息化提供了有力的支持，从而有力地推动了深化教育改革的进程。

学科教师充分发挥各学科工具软件在教育教学中的积极作用，有利于充分挖掘计算机和网络教育资源的利用率和潜力，有利于把学生从网上聊天、网上游戏、网上视听娱乐导向到学科专业学习以及以计算机、网络和外语为核心的信息技术的学习与掌握这样正确的方向上来；有利于学校和教师实施创造教育，把培养学生的创新意识，创新思维和创新实践能力落实到课堂教学和实践教学环节中去。

支持信息化教学的学科工具与普通应用软件类似，不同的是它的设计是针对教学中某一学科内容和特点，具有很强的学科实用性。具体来说，体现在以下几个方面。

1、方便快捷，具有很强的针对性。

很多学科工具软件是针对某一学科某一方面知识甚至是某一知识点而设计的，具有体积小、针对性强的特点，使用起来方便、快捷。

2、生动活泼、直观形象，具有很强的趣味性。

这个特点是所有多媒体软件的特色，许多学科工具为了吸引学生也都重点利用了这个优势，尤其是小学学科工具，画面设计美观，声图并茂，为学生的学习创造了良好的情境，使学生很容易进入学习状态。

3、交互性强，对个别化学习具有很好的支持性。

学科工具具有很强的交互性，能够让学生充分地参与对整个学习过程的自我控制和安排。如众所周知的几何画板，就是一款交互性极强的几何学科工具，它可以让学生自己设定不同的数学参数，然后显示出不同参数下几何图形的变化，学生所看到的不再是

书本上那种呆板枯燥的几何图形，而是变化多端的动态的几何动画，这些动画向他们展示了几何世界的无穷乐趣。

4、能够解决常规教学手段无法处理的问题。

一些常规教学中无法处理的问题，可以用工具软件来完成。例如，在函数教学中，需要观察函数的各个参数对函数图像的影响，常规教学中只能通过给出几个不同参数下的函数图像来大致观看一下，而在工具软件的帮助下，则可以通过参数的连续改变来观察函数图像的变化过程，从而深刻体验函数参数对函数图像的影响。

5、为学生提供一个自主探索的平台

在传统的实验过程中，考虑到实验的危险性和设备的安全性，学生只能按照规定的正确的操作方法进行实验，不允许学生违规操作，而在虚拟实验室里，学生可以尝试各种操作，进行自主探索。例如，学生可以反接电流表，可以将电流表的3个接线柱全部接上导线，进而根据实验结果去深入思考、探索。

6、跟踪实验过程，进行智能诊断

虚拟仿真学科工具可以用来进行实验，实验是在软件系统的控制下完成的，系统能够跟踪和记录学生的每一个操作步骤。如果在软件系统中加入诊断功能，就能够对整个实验过程进行分析、判断，找出操作中存在的问题，这不仅可以用来进行实验课的在线指导，也可以用来进行实验课的考试。

1.3 学科工具软件现状分析

随着信息技术教育发展至今，各种学科工具大量涌现，种类繁多，在很多大的软件网站上，如华军软件园和天下软件，一般把教育教学软件分成文科工具、理科工具、外语工具、天文地理、电脑学习、教育管理、考试系统、电子教室和学前教育这几大类，我们这里讨论的主要是课堂教学和学习软件，也就是文科工具、理科工具、外语工具、天文地理、电脑学习这几类。文科工具主要是帮助学生进行汉语语言学习和文学能力培养，如小学生写作技巧三十六计、成语接龙、汉语宝典等等；理科工具一般是数学、物理和化学这三科，针对这三科不同特点，分别有很多相应的学科工具。数学工具如几何画板、数学解析几何图像绘制大师等等；物理如金华科仿真物理实验室、金华科仿真电路实验室等；化学工具如ChemOffice、元素世界、分子量计算器、金华科仿真化学实验等等。

现有的学科工具涉及的学科很多，各自的着重点也不同，但总的看来，仍然存在以下一些问题。

1、应试教育色彩较浓。

信息技术的应用不会自然而然地创造教育奇迹，它可能促进教育革新，也可能强化传统教育。仔细看一下现有学科工具的功能，会发现有许多有着浓重的应试教育色彩，事实上它们还是在为应试教育服务^{[3][4]}。

2、没有充分体现计算机软件的优势。

有许多学科工具功能比较简单，基本上就是印刷图书的翻版。当然这也没什么不可，

但问题是当前我们最需要的是那些具有开创意义，能够革新教学和学习的学科工具，从而真正实现信息化教学的本质目标。在这一点上，计算机有着巨大的潜能，我们希望以后能够设计出越来越多的体现现代教育思想，充分发挥计算机软件优势的优秀学科工具^[5]。

3、设计思想没有充分体现信息化教学的本质特点。

4、学科工具缺乏相关的标准。操作方式不统一，不利于教师掌握；各种学科工具不能关联使用，不能集成。

对大量学科工具进行仔细审查后发现，像几何画板这么优秀的工具很少。很多都是功能极其简单，无法为教学和学习创设情境，只能简单作为学习辅助工具。

学科工具大大促进了教学信息化，也大大方便了学科教学，所以在教学中大量应用学科工具是教育信息化的必然趋势。但正像何克抗教授所说，当前信息技术与课程整合还没有实现质的突破。所以，学科工具的设计和开发如何能够充分体现信息技术与课程整合的本质特点，是以后设计和开发人员面临的突出问题。因此，工具软件的设计设计和开发应该：

1、以现代化教育思想和教学理论为指导，将技术的主要精力放在设计和开发高质量的学科工具上，为课堂教学信息化改革提供有力的支持。笔者从网上大概了解了许多学科工具，感觉很多在功能上并不比传统工具优越多少，甚至在重复传统工具的功能，这无疑是一种技术的一种浪费。

2、充分发挥计算机技术优势，如多媒体的交互性，而不仅仅是传统教材的电子化。利用计算机技术，我们可以综合多种媒体来表达教学内容。还有当今正处于研究热点的人工智能技术和虚拟技术，更为开发具有高智能、强交互性的教学软件提供了条件，能够为学生创设良好的学习情境。

3、要全面支持信息化教学，学科工具无论在内容上还是在表现形式上，都还有很多工作要做。在内容上，应该尽量全面，以有效支持学科教学和学习。在表现形式上，应做到尽量满足不同年龄层次、不同学习动机的学习者的需要。所以，要真正设计一个好的学科工具，需要有关教育专家和计算机技术专家的通力合作。

4、统一学科工具软件的操作方式，提高构件的可复用性，便于进行集成。

1.4 存在的问题及解决思路

通过对现有学科工具软件的分析，发现下面一些问题和现象。

1、现有学科工具很多，但是功能上比较独立、分散，不能很好地集成在一起使用。例如：函数图像绘制工具可以用来绘制各种函数图像，对于函数图像的讲解很方便，但是不能录入公式、图片和丰富文本，这样一来，教师只能在其他工具（例如 PowerPoint）中编辑文字、公式、图片，然后通过超链接来单独打开函数图形绘制工具，这带来的不便就是，教师需要在不同软件之间频繁进行切换，以便让学生深刻理解函数的各个参数对其图像的影响。

2、教师讲课时，主要用两类课件：Flash 集成课件和 PowerPoint 课件。Flash 集成课件将一堂课上要讲授的内容全部集成在一起，按照授课顺序进行讲解。由于 Flash 课件

制作的专业性，教师不能独立完成，需要 Flash 专业人员帮助制作，Flash 课件主要用于公开课、竞赛课。教师在平常讲课时，最常用的工具是 PowerPoint，因为教师可以自己制作，但这样又回到了上面谈到的问题，PowerPoint 没有提供学科方面的专用工具（如函数图像绘制工具），只能链接（不是集成）其他学科工具，然后用多个工具的切换来完成授课过程。实际上，PowerPoint 和学科工具是完全独立的，没有交互性的。

- 3、虽然 PowerPoint 功能强大，但是教师的 PowerPoint 课件中用到的功能不到 5%。
- 4、由于缺少一个整合的平台，学科工具软件没有形成规范，风格各异，操作方法不统一，教师需要学习不同学科工具的操作方法。即使是非常类似的功能，比如添加一个化学仪器，一些工具通过鼠标单击仪器图标的方式，另一些通过鼠标双击的方式，还有一些需要通过鼠标拖拽的方式，这给教师学习和使用工具软件带来了不便。
- 5、构件不能重用，每个新的学科工具构件的开发均需要从头开始。
- 6、各类构件之间不能互相调用和传递信息。
- 7、缺乏基于领域知识的学科工具构件模型。

针对上面这些问题，本文拟设计和开发一个学科工具构建平台，并提出一个基于领域知识的学科工具构件模型，使在此平台下基于学科工具构件模型进行学科工具的二次开发变得简单、规范。同时，提供一个构件编辑、连接、集成环境，辅以丰富的教学辅助功能，不仅满足了教师对学科工具构件的需要，也为教师提供了一个很好的备课和讲授平台，为教学过程提供帮助。

1.5 论文的研究工作

本文通过对学科工具软件现状的研究，分析了学科工具软件在开发和应用上存在的问题。针对这些问题，首先，探讨了学科工具构建平台的意义，提出了基于领域知识的学科工具构件模型。然后，给出了学科工具构建平台的总体方案，建立了基于领域知识的学科工具构件模型，并且详细分析、设计和实现了平台和构件模型。在此基础上，通过物理电学构件的设计和实现，对平台和构件模型进行了验证。

1.6 论文的研究意义

本文的研究意义主要体现在以下两个方面：

- 1、基于领域知识的可重用学科工具构件模型的建立，便于学科工具构件的重用。同时，学科工具构建平台为学科工具构件模型的使用提供了集成开发环境，使在此平台下基于学科工具构件模型进行学科工具的开发变得简单、规范，能够大大提高学科工具构件开发的效率。
- 2、平台不仅考虑了构件开发的方便性，也充分考虑了学科工具构件最终用户的需求。通过对各种文字、图片、表格、Flash、3D 动画等的支持，辅以丰富的教学辅助功能，不仅满足了教师对学科工具构件的需要，也为教师提供了一个很好的备课和讲授平台，满足教学过程的需要，具有很大的使用价值和现实意义。

1.7 本文结构

本文对学科工具构建平台的设计和实现以及基于领域知识的学科工具构件模型的建立进行了研究，以平台搭建、构件模型设计和具体学科构件的实现为主线展开论述，论文具体安排如下：

第一章 绪论：阐述了学科工具软件的发展历程，介绍了信息技术与课程整合对学科工具软件的需求，分析了学科工具软件现状，提出了学科工具软件存在的问题及解决思路，确定了本文的研究工作。

第二章 学科工具构件模型研究：介绍了软件重用技术相关知识，分析了组件与框架的关系，确定了本系统中将要研究的组件和框架，从模型的构成、属性分析和通信机制等方面阐述了学科工具构件的建立方法。

第三章 系统总体方案：首先，简要概述了本系统的定位；然后，进行了系统功能设计，包括：功能结构图、页面、公用构件、学科构件和生成独立的可执行文件几个部分；接着是数据设计，包括：数据的逻辑结构、数据的物理结构、版本控制；最后是界面方案和开发工具的选用。

第四章 原型系统实现：介绍了GDI+基本原理和GDI+绘图基本方法，包括：绘制线条、填充、坐标变换、映射模式、图形图像优化处理；进行构件数据设计和功能设计，包括添加、删除、恢复、选择、复制、剪切、粘贴、移动、缩放、旋转、关键点控制、属性编辑等功能的实现方法。

第五章 基于本平台的学科构件实例：设计和实现了物理电学构件，本章是对前面设计的平台框架和构件模型的验证。在本章中，分析了常规教学过程中的困难，设计了构件的数据、功能，讨论了构件在教学中的应用。

第六章 总结与展望：对全文的工作进行了概括总结，分析了不足之处，并对今后有待提高的工作进行了展望。

第二章 可重用学科工具构件模型研究

2.1 软件重用技术

2.1.1 软件重用技术介绍

软件重用 (Software Reuse, 又称软件复用) 是指利用事先建立好的软件部件创建新软件系统的过程。这个定义蕴含着软件重用所必须包含的两个方面: 其一系统地开发可重用的软件部件。这些软件部件可以是代码, 但不应该仅仅局限在代码。我们必须从更广泛和更高层次来理解, 这样才会带来更大的重用收益。比如软件部件还可以是: 分析, 设计, 测试数据, 原型, 计划, 文档, 模板, 框架等等。其二系统地使用这些软件部件作为构筑模块, 来建立新的系统。

2.1.2 软件重用的意义

软件重用有效地解决了软件开发效率、可靠性低下和维护开销巨大的矛盾, 有效地缓解了困扰软件工业多年的软件危机。采用软件重用技术具有以下几点好处:

1、提高软件生产率, 提高软件质量, 降低开发风险, 缩短开发周期。

通过使用已有的软件成分可明显提高开发速度, 而且已有软件成分是经过完善的测试, 并通过实践考验, 具有很好的可靠性和稳定性。软件产品重用能够节约成本和时间, 因为不必从头开始建造被重用构件。通过软件过程重用, 软件质量可得到提高^[1]。

2、开发的软件系统容易维护。

使用经过检验的软件成分除了可以减少可能的错误, 还可以减少软件中需要维护的部分。例如, 要对多个具有公共功能的系统进行维护时, 对公共部分的修改只需要一次即可, 而不是在每个系统中分别进行修改。

3、提高互操作性。

软件重用一个更为专业化的好处在于提高了系统间的互操作性。通过使用接口的同一个实现, 系统将更为有效地实现与其它系统之间的互操作。

4、支持快速原型。

重用的另一个好处在于对快速原型的支持, 即可以快速构造出系统可操作的模型, 以获得用户对系统功能的反馈。利用可重用构件库可以快速有效地构造出应用程序的原型。

5、降低开发和维护的费用。

2.1.3 软件重用的形式

目前, 软件重用主要在知识重用、方法和标准重用、软件成分重用等几个层次上进行。

1、知识重用。

知识重用主要是指软件工程知识的重用, 是软件重用的最高形式。虽然它不像代码级重用那样直观, 但对软件重用却是最具革命性的, 也是目前最为活跃的研究与实践范

畴，它涉及到知识工程和人工智能。

2、方法和标准重用。

方法和标准重用是软件工程方法或软件开发规范、标准、法律、法规等的重用。软件工程方法是长期以来软件开发成功的经验与失败的教训的总结，它表现为软件工程方法学和软件工程标准与规范。这一级别的软件重用往往是基于软件体系结构的，它对软件开发的成功率是至关重要的。

3、软件成分重用。

软件成分的重用是直接面向软件开发过程和软件开发人员的。在基于重用技术的软件开发中，提高可重用度是进一步提高软件生产率的关键，而可重用度在一定程度上依赖于抽象，由于在软件成分的重用中，可重用软件成分具有较高的抽象度，所以在这一级别上的重用是提高软件生产率和软件质量最有效的方法之一。一般地，按抽象程度的高低，软件成分的重用可分为以代码重用、设计结果重用、分析结果重用和测试信息重用几种形式。

2.1.4 软件重用的技术途径

自从软件重用思想产生以来，计算机科学家和软件工程师就致力于软件重用的技术的研究和实践。在 30 多年的时间内，出现多种软件重用技术如下：

1、库函数

库函数是很早的软件重用技术。很多编程语言为了增强自身的功能，都提供了大量的库函数。对于库函数的使用者，他只要知道函数的名称、返回值的类型、函数参数和函数功能就可以对其进行调用。缺点是重用受限于语言，不能做到与开发平台无关。它最根本的缺点在于无法与数据结合在一起，从而使程序员无法大规模使用。

2、面向对象

面向对象技术是近三十年来学术界和工业界研究和应用的一个热点。面向对象技术通过方法、消息、类、继承、封装和实例等机制构造软件系统，并为软件重用提供强有力的支持。面向对象方法已成为当今最有效、最先进的软件开发方法^[11]。与函数库对应，很多面向对象语言为应用程序开发者提供了易于使用的类库，如 VC++中的 MFC。

3、模板

模板相当于工业生产中所用的“模具”。有各种各样的模板（如文档模板，网页模板等），利用这些模板可以比较快速地建立对应的软件产品。模板把不变的部分封装在内部，对可能变化的部分提供了通用接口，由使用者来对这些接口进行设定或实现。

4、设计模式

设计模式是对被用来在特定场景下解决一般设计问题的类和相互通信的对象的描述。一个设计模式命名、抽象和确定了一个通用设计结构的主要方面，这些设计结构能被用来构造可重用的面向对象设计。设计模式使人们可以更加简单方便地重用成功的设计和体系结构。设计模式帮助你做出有利于系统重用的选择，避免设计损害了系统重用性。简而言之，设计模式可以帮助设计者更快更好地完成系统设计。

5、构件技术

构件又称软构件或组合软件(组件),它是以嵌入后马上可以使用的即插即用型部件的概念为核心,通过软部件的组合来建立应用技术体系。简而言之,软构件是二进制形式的可重用的代码和数据段。它们必须遵循一个外部的二进制标准,但是它们的内部实现却是没有限制的,可以用支持指针操作的任何语言来实现。从广义上说,软件构件是一种相对独立、可重用的二进制代码,包括功能模块、被封装的对象类、软件构架和软件系统模型等。构件技术是迄今为止最优秀的软件重用手段,并成为现在最受重视,最流行的技术。

2.2 组件与框架

2.2.1 组件的定义

可重用软件组件(RSC)是可以被重用的软件实体,它可以是设计、代码或软件开发过程中的其它产品,RSC有时也称为“软件资产”^[22]。当前,基于组件的软件开发(CBSD)模式已为广大研究人员所认同,人们在不同的背景下提出了各种组件定义,这些定义包括:

- 1、组件是一个具有规范接口和确定的上下文依赖的组装单元,软件组件能够被独立部署和被第三方组装。
- 2、组件是可单独生产、获取、部署的二进制单元,它们之间可以互相作用构成一个功能系统。
- 3、组件是一个不透明的功能实现,能够被第三方组装,符合一个组件模型。
- 4、组件是经过封装的、有定义完备接口的可发布的软件包。
- 5、组件是具有接口义务的合成单元,并且明确规定了所有背景的依赖关系;由第三方进行合成的主体,可以独立地部署。
- 6、组件是软件实现的一种内聚包:能够独立的开发、分发;所提供的服务被明确、完备地定义;能够与其他组件合成,可定制部分属性而不修改组件本身。
- 7、组件是一个通过接口向外界提供服务的软件包。

综合这些组件的定义,我们还可以发现组件的两个本质属性:重用和易用。重用是组件的目的,即它是用来重复使用的。易用是组件实现其目的的方式,即组件要能很方便地使用。从传统产业的成功经验来看,要达到易用,标准化是必不可少的。只有大家都遵循同一标准,才能方便使用组件。以上这两个本质属性就够成了组件的一个比较完整的定义,这个定义只是组件的概念定义,该定义具有抽象性,不涉及具体的实现技术,无法在实际应用中直接使用。为了在实际应用中使用的组件,必须对组件进行明确的、具体的描述,即建立组件模型。组件模型一般是指组件的实现模型和组件的分析设计模型。本文中所述的组件模型主要是指组件的分析设计模型。

2.2.2 框架的定义

在设计模式中, Gamma 等为框架给出了一个定义:“框架就是一组协同工作的类,它们为特定类型的软件构建了一个可重用的设计”。把这个定义做一些扩展:框架是一组相互协作的构件(component)的集合,能够处理一个或多个问题域(domain)中

的一系列问题。框架包含了以下关键特征：

- 1、框架包括一系列的抽象体（abstraction）——类或者构件。
- 2、这些抽象体相互协作，以完成任务。框架为一组相互独立的抽象体定义它们的协作方式，这是可重用解决方案的核心。
- 3、抽象体以及它们的行为方式都是可重用的。框架并不等同于类库（classlibrary），类库的目的是提供一组到处都通用的类和函数。而框架提供的重用级别要高于类库，它更关注通用的过程和功能。框架着眼于以通用的方式满足某类的需求，以一个可扩展（或可定制）的核心体系结构为基础，以适应不同用户的需求。支持扩展的地方被称为扩展点（extension point）或焦点（hotspot）。
- 4、框架解决的是某个特定范围即领域的问题。每个框架都有自己的目标领域，它并不能满足所有人的所有需求。

2.2.3 框架是软件产品的一种形态

不同的软件形态决定了软件开发目的的不同，以及软件开发过程的不同。项目、产品、框架都是软件的不同形态，虽然它们针对的目标领域可能是相同的，但是由于形态的差异，导致了它们之间极大的不同^[32]。项目针对的是特定的用户群，例如某个企业。它不需要很大的灵活性，因为需求是相对固定的，但有很强的特殊性，必须能够满足用户的一些极端需求。产品针对的是某个领域的用户，例如某种类型的企业。它需要满足这一类用户群中的通用需要，并可以根据不同用户的需要做一定程度的定制。框架则是针对那些开发应用（项目或产品）的人而设计的，它的目标是使那些开发应用的人能够快速制造出高质量的软件。这一目标决定了框架开发具有高度的灵活性，并允许将部分的实现留由应用开发人员完成。

框架是一门艺术。他在提供可重用的内容（如预定义的业务对象和业务流程和灵活性（允许对内容进行定制，以准确地满足框架用户需要）之间保持一种平衡。如果过度提供灵活性，那么框架就会分裂成微小的业务对象和低层次的业务流程——更像是一个类库。另一方面，如果过分关注内容，就会面临使定制复杂到难以使用的风险——就像使用一个现存的系统来构建另一个系统一样。

2.2.4 本系统中的组件与框架

本系统作为一个框架软件，是专门为学科工具软件开发服务的，该框架支持的组件是下面将要介绍的学科工具构件模型中定义的构件。框架由一系列学科工具构件构成，框架为学科工具构件之间的通信、交流、协作等提供支持。

2.3 学科工具构件模型的建立

2.3.1 建立学科工具构件模型的必要性及其意义

学科工具大大促进了教学信息化，也方便了学科教学，教学中大量应用学科工具是教育信息化的必然趋势。目前，学科工具软件种类繁多、风格各异、操作方法不统一，这给教师学习和使用工具软件带来了不便。同时，由于没有基于学科工具领域的参考模型，学科工具构件缺乏统一的接口，构件之间不能互相调用和传递信息，这不仅影响了

设计和开发效率，也给构件的复用和升级、维护带来了困难。

建立一个学科工具构件模型，不仅可以使学科工具构件的设计和开发变得简单、规范，提高学科工具的设计和开发效率；还可以通过规范操作和控制方式，为教师和学生的学习和使用带来便利。

2.3.2 学科工具构件模型的构成

通过对现有大量学科工具软件的研究和对用户不断变更的需求的分析，结合自己开发学科工具软件的经验，提出下面的学科工具构件复用模型。针对一个给定的用户需求，基于此模型，可以完成具体学科工具构件的设计。

学科工具构件 ::= {{几何属性集 GE}, {物理属性集 PH}, {行为属性集 AC}, {应用属性集 AP}, {反应机制集 RA}, {关系集 RE}, {操作方式集 CT}}。

几何属性(GE)主要影响仿真对象(如虚拟试验仪器、设备等)的展现形式,对于仿真对象的展示,可以是图片、矢量图形、3D模型等形式。物理属性(PH)是仿真对象的固有属性,如物理学科中的电阻的阻值、化学学科中的液体的沸点等。行为属性(AC)是仿真对象可能表现出来的各种行为,如灯泡烧坏变黑、2种组合的药品发生化学反应等。应用属性(AP)是从学科工具构件的使用角度来描述的,从大的方面可以划分为:教学展示用构件、考试测评用构件、自主探索用构件,一个构件的应用可能是多方面的,也可能是单方面的。反应机制(RA)是指构件对于其他构件传递过来的各种信息的加工、处理、传递方式,学科工具往往是多个构件组合起来使用,这时,只有构件之间具有有效的反应机制,才能充分发挥学科工具构件的作用。关系集(RE)中的元素是构件的其他任意两个集合中元素的关系,例如:在电流表构件中,经过电流表的电流是电流表的物理属性,指针位置是几何属性,指针位置是由电流决定的。操作方式(CT)是指人机交互方式,即用户对构件的控制方式,比如鼠标可以拖拽滑动变阻器的滑块。

有了学科工具构件,就可以应用这些构件组成一个完整的应用。

应用 ::= {{学科工具构件集}, {算法集}}

学科工具构件集由上述工具构件组成,算法集中的元素用来处理构件之间的关联关系。例如:在仿真电学实验室中,电池组、安培表、导线、灯泡、开关等几个构件就可以组成构件集,算法集中可以只有一个元素——节点法,节点法能够计算电路中的电压和电流。

2.3.3 构件模型的工作原理

对于模型各个部分之间的关系,可以从学科工具构件的工作过程来分析。对于一个具体的应用,不管是教学展示用,还是考试测评用或自主探索用,构件(组)的工作过程见图1。

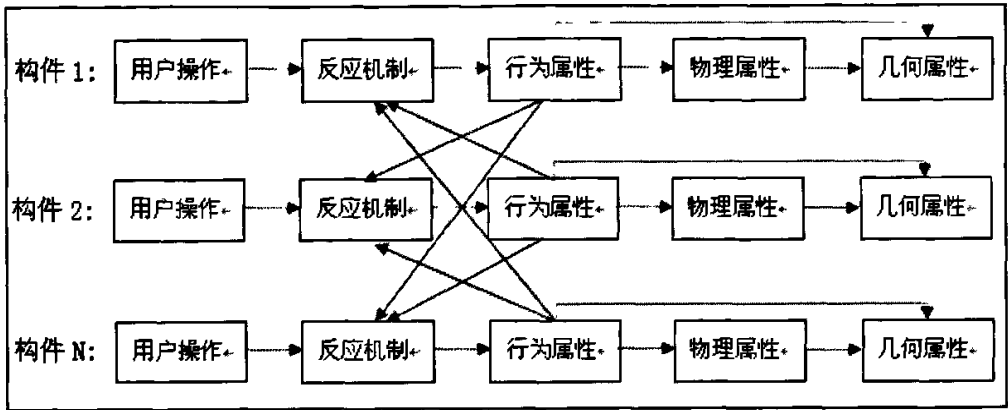


图 1 构件工作过程

从图中可以看到，在一个具体的应用中，一个构件的启动方式有两种，一是由用户的操作来启动，二是由其他构件来启动。第一种方式比较直接，比如用户按下开关的按钮，则启动了开关构件；第二种方式则需要构件能够对应用中的其他构件的行为做出反应，比如开关按下后，开关两端的导线会接收这一行为并交由反应机制进行处理，从而促使导线做出自己的行为。

对于一个独立的构件，模型的各部分之间的关系比较明了，即由反映机制集接收用户操作，做出自己的行为，而该行为会影响其物理属性和几何属性。

对于一组构件，情况就不同了。这时，在大多数情况下，构件是由第二种方式进行启动的，也就是说，构件是由其他构件的行为来启动的。那么，一个构件的行为是以怎样的方式传递给其他构件呢？在这里应该采用主动、直接传递的方式。在实现上，可以为构件的每一个行为都建立一个链表，记录对该行为感兴趣的构件的指针，对该行为感兴趣的构件需要主动将自己加入到链表中，从而才有被通知的可能性。构件启动一个行为后，为将该事件传递出去，需要遍历该行为对应的构件链表，并根据该行为自己特有的算法确定链表中应该接收这一事件的构件（可能是多个构件），从而将行为事件传递出去。下面以物理光学构件中光线与凸透镜、平面镜的关系为例来说明这一过程：首先，凸透镜、平面镜构件应该将自己加入到发射光线这一个行为对应的链表中，发射光线这一行为发生后，光线构件需要遍历这一行为对应的链表，找到在光线方向上距离发光点最近的链表中的构件（这里假设为凸透镜），然后将这一事件传递给该凸透镜构件，凸透镜构件接收到这一事件后，做出自己的行为（对光线进行折射），然后将这一行为继续传递下去（可能传递给平面镜）。

虽然构件之间可以自发地传递各自的行为事件，但这不足以满足一个完整应用的需要。比如计算闭合电路中电流这一个应用需求，用户闭合开关构件后，这一行为事件会依次传递到电路中的各个构件，但任何一个构件都不会触发计算电路中电流这一行为，因为这一行为是具体应用的行为，而不是某个构件的行为。对这个问题的解决办法是：任何构件的任何行为事件，均需要通知应用，应用每接收到一个行为事件后，都会调用应用算法集中的算法对全部构件进行检查、分析、计算，从而建立起构件之间的关联关系。

第三章 系统总体方案

3.1 系统定位

本系统将要达到两个目的：

1、设计和开发一个学科工具构建平台，使基于此平台的学科工具的开发变得简单、规范，便于在此基础上进行学科工具的开发。

2、基于此平台和学科工具构件模型设计和开发物理电学学科工具构件，以达到对平台的学科工具构件开发能力的验证。

本系统的优点主要体现在以下几个方面：

(1) 开放性：对整个系统进行模块化设计，各个模块的功能独立、接口明确，具有良好的扩展性。

(2) 智能性：能够根据学科构件间的关联关系进行自动计算、求解。

(3) 集成性：各个学科的各种构件集成在一起，便于交叉学科构件的使用。

(4) 界面友好：友好的个性化交互界面。把文字、声音、图片、视频及互动式构件结合起来，抓住了学习者的注意力和思维，充分调动了学习者学习的积极性和主动性。

(5) 简单易学：系统功能明确、清晰，软件操作简单。

3.2 功能设计

3.2.1 功能结构图

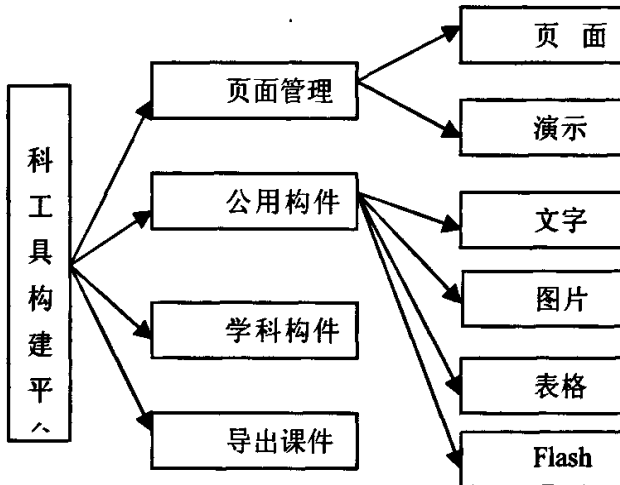


图2 功能结构图

3.2.2 页面管理

在本系统中，以页面的方式来管理各种构件，若干个页面构成一个最终的应用。各个构件的相关操作也是在页面上完成的，页面的功能包括两大方面：编辑状态时的操作

和播放状态时的操作（辅助讲授）。

1、编辑状态时的操作

- 添加页面：增加一个页面到应用中，可以通过菜单和键盘两种方式来添加页面，添加的页面默认为空页面，用户可以在页面上添加各种学科工具构件。
- 删除页面：可以通过菜单和键盘两种方式来删除当前选中的页面，删除页面后，选定一个其他的页面作为当前页面。
- 选择与切换：选择页面是为了对页面及其上面的构件进行各种控制和操作。可以通过 Shift、Ctrl 键进行页面的多选操作，选择一个页面后，对构件的各种操作也就切换到被选择的页面上了。
- 恢复：既可以恢复先前对页面的操作，也可以恢复对页面上构件的操作，恢复的次数可以由用户设置，便于优化系统的性能。
- 复制、剪切、粘贴：利用 Windows 剪贴板完成页面的复制、剪切、粘贴操作，在完成上述操作时，页面上的构件也随页面完成复制、剪切、粘贴操作。
- 隐藏：可以对页面进行隐藏和显示，被隐藏的页面在编辑状态可见（页面大纲上需要显示隐藏标记），在播放状态不可见。
- 移动：一个应用由多个页面构成，页面之间存在着顺序关系，可以通过鼠标拖拽页面大纲的方式来调整页面的顺序关系。
- 缩放：可以对页面的显示区域进行缩放，缩放后，页面上的构件按比例进行缩放。
- 属性设置：设置页面的背景、读写属性、操作方式等。

2、播放状态时的操作（辅助讲授）

- 黑板、白板功能：黑板、白板是一种特殊的页面，页面上没有任何构件，但用户可以用鼠标在其上面进行画线、绘图等，以满足课堂讲授的需要。
- 教鞭功能：教鞭功能是为了方便教师强调讲授内容而设置的，设置教鞭后，在页面上移动鼠标，教鞭会随着鼠标的轨迹而移动，教鞭的形状、长短、颜色、粗细等可以由用户进行设置。
- 粉笔功能：粉笔能够完成用鼠标在页面上进行书写的功能，粉笔的颜色、粗细等可以由用户进行设置。
- 页面切换和跳转：在播放状态时，既可以在各个页面之间进行顺序切换，又可以通过链接来跳转页面，以满足教师授课的需要。
- 开始、结束：控制播放的开始和结束，也就是在播放和编辑状态之间进行切换。

3.2.3 公用构件

为了方便用户更好地使用学科工具构件，满足其讲授知识、探索试验等方面的需要，本系统提供一些常用的简单的公用构件，这些构件独立于具体学科，主要起到一种辅助作用，主要包括：文字、图片、表格、Flash。对于这些构件的各种操作，很大一部分与学科构件的操作相同，在这里仅就一些特殊的功能进行描述，其他功能参考学科构件。

3.2.3.1 文字

文字作为最基本的公用构件，在一个应用中必不可少。在本系统中，文字是以丰富

文本格式(RTF)的方式存在的。文字构件能够完成如下几方面的功能。

- **基本编辑功能：**设置字体和字号、选择、删除、恢复、设置对齐方式、设置段落格式等。
- **状态切换：**文字构件有两个状态——编辑状态和显示状态，在页面中，用鼠标点击文字构件，可以激活文本编辑功能，进入编辑状态，此时可以进行各种文本编辑操作，完成编辑后，点击页面上的其它区域，又回到显示状态。
- **旋转：**可以对文字构件进行旋转操作，然后通过坐标变换，在旋转后的坐标系中输出文字块。对于旋转后的文字构件，如果进行编辑，必须暂时切换到旋转前的状态（便于编辑），待编辑后再恢复旋转状态。
- **自动缩放：**改变文字构件的边框尺寸时，无论文字是否处于编辑状态，文字大小能够随边框的改变而改变。

3.2.3.2 图片

图片也是常用的公用构件，本系统支持 BMP、JPG(JPEG)、GIF、ICO、PNG 等几种常用的图片格式。对于图片，除了基本显示功能外，还可以完成下面一些控制和编辑操作。

- **GIF 动画的控制：**可以对 GIF 动画进行播放、暂停、停止等控制，可以分离出 GIF 动画中的图片。
- **旋转：**可以对图片进行任意角度的旋转。
- **锁定纵横比例：**在对图片进行缩放处理时，既可以随意进行缩放，又可以在当前的纵横比例关系下进行缩放。
- **简单编辑功能：**可以对图片进行反色、黑白处理、对比度处理、局部拷贝等操作。

3.2.3.3 表格

表格用来限定其他构件的位置，使构件排列规则、有序，表格中的单元格可以容纳所有类型的构件，可以对表格的行、列、单元格进行下面的操作。

- **合并单元格：**选中若干连续的单元格后，可以将其合并成一个单元格，并将原来单元格中的构件全部放入合并后的单元格中。
- **拆分单元格：**选中一个单元格，将其拆分为两个单元格，并合理分配原单元格中的构件，设原单元格中的构件数为 M，则在拆分后的单元格中，一个单元格中的构件数为 $N=M\%2$ ，另一个单元格中的构件数为 $M-N$ 。
- **删除行：**删除选中的行中的所有单元格，同时删除单元格中的构件。
- **删除列：**删除选中的列中的所有单元格，同时删除单元格中的构件。
- **在下方插入行：**在选定的行的下方插入一个新行，同时选中新插入的行。
- **在上方插入行：**在选定的行的上方插入一个新行，同时选中新插入的行。
- **在左侧插入列：**在选定的行的左侧插入一个新列，同时选中新插入的列。
- **在右侧插入列：**在选定的行的右侧插入一个新列，同时选中新插入的列。
- **背景填充：**用设定的颜色填充所有单元格的背景。
- **选择行：**选择一行中的所有单元格及单元格中的构件。
- **选择列：**选择一列中的所有单元格及单元格中的构件。

- 平均分配行：按照表格的总高度平均分配各行的高度。
- 平均分配列：按照表格的总宽度平均分配各列的宽度。

3.2.3.4 Flash

Flash 动画在学科教学中使用比较广泛，为了方便地使用 Flash 动画，避免在学科工具构件环境和 Flash 动画窗口间频繁进行切换，本系统内嵌 Flash 播放插件，能够播放 Flash 动画，具体功能如下：

- 装入 SWF 文件：本系统支持的 Flash 动画文件是指 SWF 文件。
- 播放控制：启动、暂停、停止、快进、快退、跳转等控制方式。
- 状态切换：由于 Flash 窗口动画和其他学科构件工作于同一页面上，为了解决 Flash 窗口与其他构件的重叠问题，使 Flash 可以在窗口状态与图片状态之间进行切换。处于窗口状态时，可以控制 Flash 动画，进行播放、暂停、停止等操作；处于图片状态时，仅仅显示 Flash 当前帧的位图，这时不能进行播放等操作，只能够进行位置调整等操作。

3.2.4 学科构件

学科构件是指根据前面介绍的学科构件模型而建立的构件，对于根据构件模型建立的学科构件，提供下面这些通用的控制和操作。

- 添加构件：本系统提供两种构件添加方式，直接插入方式和鼠标拖拽方式。直接插入方式是指按照默认的规格将构件直接插入到页面的中间，鼠标拖拽方式是指将构件插入到鼠标拖拽出的区域内，在拖拽方式下，如果拖拽出的区域是一个点，则将构件的中心对准该点，按照默认的宽度和高度来自动计算构件所在的区域。
- 删除构件：选中一个或多个构件，可以将其删除。删除构件前，能够通知被删除构件进行预处理操作，如断开或删除与其他构件的连接等。
- 复制、剪切、粘贴：利用 Windows 剪贴板和序列化操作完成构件的复制、剪切、粘贴操作。
- 恢复：构件状态的每一次改变均被记录下来，在后面的操作过程中，如果需要，可以逐步向前进行恢复，恢复的次数可以由用户设定。
- 移动：在页面上选中一个构件后，可以通过鼠标拖拽的方式移动构件到一个新的位置或另一个页面上，移动前，能够通知被移动构件进行预处理操作，如断开或删除与其他构件的连接等。
- 缩放：选中构件后，在包围构件的矩形框上出现八个控制点，用鼠标拖拽控制点可以在上、下、左、右、左上、左下、右上和右下八个方向对构件进行缩放。
- 旋转：选中构件后，出现旋转控制点，用鼠标拖拽旋转控制点，可以使构件沿构件中心进行旋转，如果拖拽时按住 Shift 键，则可以沿构件底部中心进行旋转，按住 Ctrl 键则克隆一个构件。
- 关键点控制：关键点用来控制构件的各个组成部分的位置，例如：滑动变阻器的滑块的位置可以用一个关键点来控制。每个构件具有一个关键点列表，列表中的所有关键点均可以通过鼠标拖拽的方式进行位置调整，在调整关键点位置时，构件可以根据自己的需要对关键点的位置设置一些约束条件。

● 属性编辑：每类学科构件都有自己的相关属性信息，选中一个构件后，首先进行构件类型判断，然后调用该类型学科构件的属性表，在属性表中，可以编辑构件的相关属性。

3.2.5 生成独立的可执行文件

在本平台下编辑、连接的各种构件，只有在装有本套系统的电脑上才能进行工作。而教师在使用时，往往是在一台电脑上进行备课（如办公室或家里），而在其他的电脑上讲授（如多媒体教室中的电脑），这就需要在所有使用该课件的电脑上安装本系统，这会带来一些不便。为了解决这个问题，提供了生成独立的可执行文件的功能，教师在本平台下编辑好一堂课的内容后，生成的文件可以在没有安装本平台的机器上运行。解决办法是：制作一个精简的播放程序，可以播放（但不能编辑）本平台生成的文件，然后将教师编辑完成的文件以序列化的方式写入播放程序的尾部，在播放时，播放程序从自己的尾部读取数据，然后分析数据、运行各种构件，完成相关操作。

3.3 数据设计

本系统主要涉及到三类数据：与应用有关的数据、与页面有关的数据和与构件有关的数据。与应用有关的数据是指一些全局属性，如页面间的切换方式，播放窗口的控制等；与页面有关的数据包括页面的背景、顺序、模板类型、页面上的构件数量等；与构件有关的数据专指具体构件的相关属性，如构件的类型、位置、物理属性、几何属性、行为属性等。

3.3.1 数据的逻辑结构

在本系统中，以页面的方式来管理各种构件，若干个页面构成一个最终的应用。即一个应用（如教师制作的一个电子教案）由若干页面(CPage)组成，一个页面由若干个公用构件和学科工具构件组成。在组织方式上，每个应用对应一个页面对象数组，数组中的每个元素是一个页面；每个页面对应一个构件对象数组，数组中的每个元素是一个公用构件或学科工具构件，应用、页面、构件的关系见图 3。

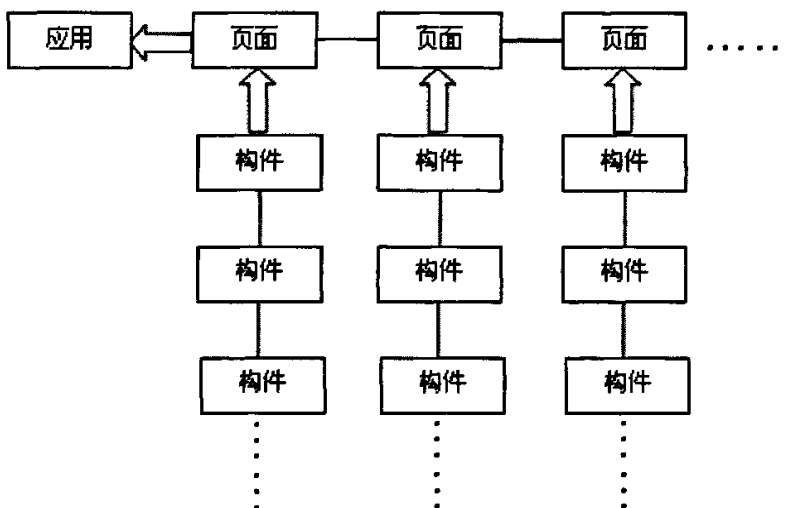


图3 应用、页面、构件的关系

3.3.2 数据的物理结构

一个应用中的数据全部保存在一个文件中，以序列化的方式进行顺序存储，具体存储方式见图 4。

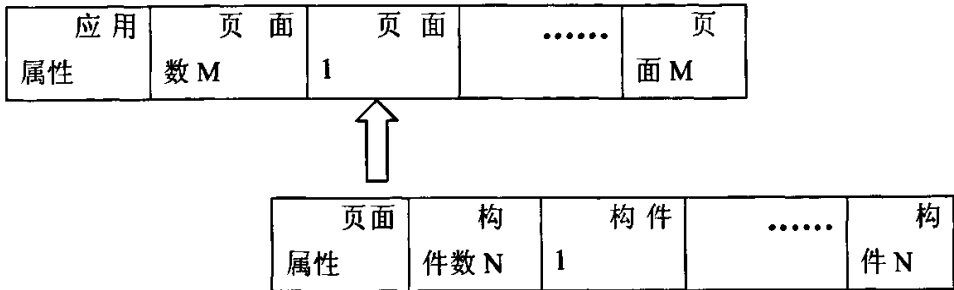


图 4 数据的物理结构

3.3.3 版本控制方案

本系统作为一个二次开发平台，随着学科工具构件数量的不断增加和修改，版本控制问题显得尤为重要，为了使新的版本能够兼容老版本的数据，采用了分层版本控制方案。

第一层为应用层的版本号，第二层为页面层的版本号，第三层为构件层的版本号，这三层版本号需要写入数据文件中。读取文件时，首先读取各个层次对象的版本号，然后根据其版本号进行不同的序列化操作，对于低版本的对象，需要将其转化成最新版本格式。

3.4 界面方案

为了让教师能够迅速熟悉系统的使用和操作方式，采用类似 PowerPoint 的界面方案，因为教师以往主要是使用 PowerPoint 作为备课和授课工具，比较熟悉其界面和操作方式。图 5 是本系统的界面截图，界面主要由四部分构成。左边是页面大纲，负责页面的组织和管理；右边是构件的工作区，是系统中最重要区域，构件的所有操作均在这里完成；下面是状态条，用来显示一些操作的辅助信息；上面是菜单栏和工具栏，工具栏有两类：公用构件工具栏和学科工具构件栏，公用构件工具栏主要由文字、图片、表格、Flash 等公用构件组成、学科工具构件栏是为各个学科构件服务的，学科工具构件栏可以在各个学科之间进行切换，这样既满足了不同学科教师的需要，又简化了界面布局。

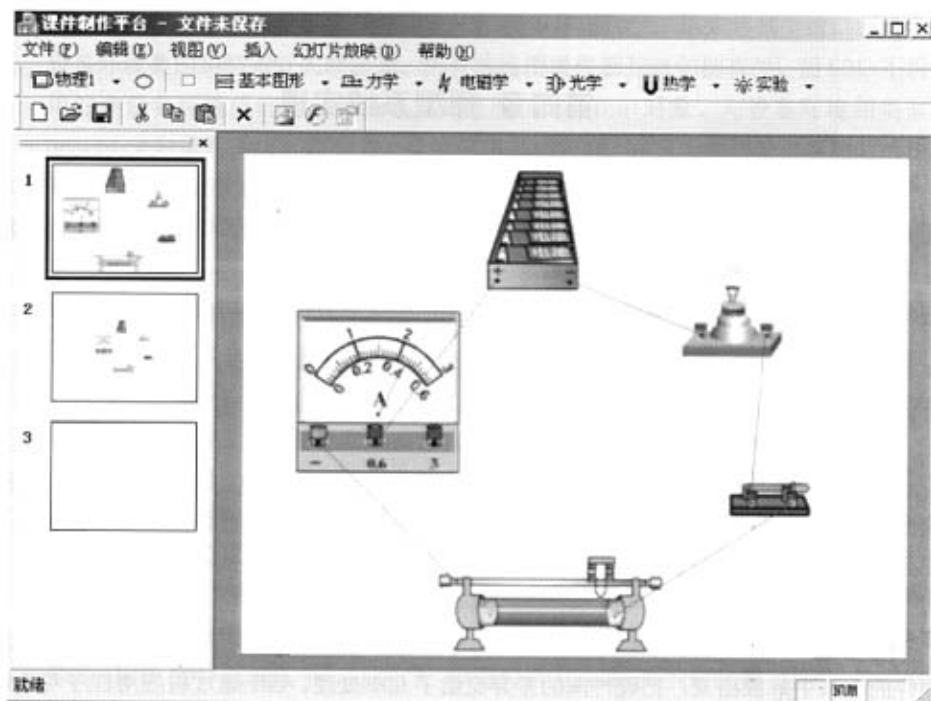


图 5 平台界面

3.5 开发环境和开发工具的选用

本系统实现选用的开发工具为 Microsoft 公司的 Visual C++.Net 2003。C++是最流行的面向对象语言之一，支持对象、类、方法、消息等概念，同时也支持面向对象方法的封装机制和继承、多态性机制。Visual C++.Net 2003 是 Microsoft 公司推出的基于 Windows 的可视化集成开发环境 Visual Studio.Net 2003 的一部分，将编辑、编译、链接和执行集成为一体，功能完善、使用方便。更重要的是，Visual C++.Net 2003 对于 GDI+和 Direct3D、OpenGL 的完好的支持，使二维学科工具构件的绘制和和 3 学科工具构件的建模变得相对简单。

第四章 原型系统实现

本系统的开发环境为 Visual C++.Net 2003，主要实现技术是 GDI+，本章首先就本系统中应用到的 GDI+ 技术进行介绍，包括绘制图形图像、坐标变换、映射模式、提高绘图速度四个方面，然后从数据存储和功能处理两个方面对系统进行了实现。

4.1 GDI+ 介绍

1、设备环境

设备环境是 Windows 用来给应用程序提供设备独立性的工具，它是 windows 系统为了处理输出设备而使用的一种内部数据结构，设备环境是 windows 程序，驱动程序，和输出设备(如打印机，绘图仪)之间的纽带，GDI 是一组 C++ 类，它在驱动程序的协助下把数据描绘在硬件上，它位于应用程序与硬件之间，把数据从一方传到另一方。

2、GDI

GDI 是位于应用程序与不同硬件之间的中间层，这种结构让程序员从直接处理不同硬件的工作中解放出来，把硬件间的差异交给了 GDI 处理。GDI 通过将应用程序与不同输出设备特性相隔离，使 Windows 应用程序能够毫无障碍地在 Windows 支持的任何图形输出设备上运行。例如，我们可以在不改变程序的前提下，让能在 Epson 点式打印机上工作的程序也能在激光打印机上工作。它把 windows 系统中的图形输出转换成硬件命令然后发送给硬件设备。GDI 命令是以文件的形式存储在系统中，系统需要输出图形时把它载入内存，如果转换成硬件命令时遇到非 GDI 命令，系统还可能载入硬件驱动程序，驱动程序辅助 GDI 把图形命令转换成硬件命令（如图 6 所示）。

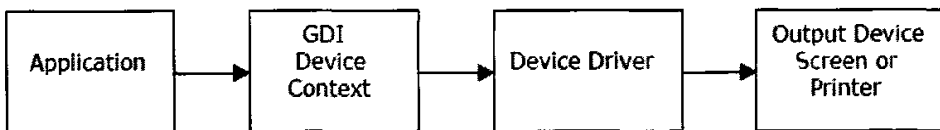


图 6 GDI 工作模式

3、GDI+

GDI+ 是 GDI 的下一个版本，它进行了很好的改进，并且易用性更好。GDI 的一个好处就是你不必知道任何关于数据怎样在设备上渲染的细节，GDI+ 更好的实现了这个优点，也就是说，GDI 是一个中低层 API，你还可能要知道设备，而 GDI+ 是一个高层的 API，你不必知道设备。例如你如果要设置某个控件的前景和背景色，只需设置 BackColor 和 ForeColor 属性。

4、GDI 到 GDI+ 编程模式的变化

GDI 是有状态的，GDI+ 是无状态的。

- 不再使用设备环境或句柄

在使用 GDI 绘图时，必须要指定一个设备环境(DC)，用来将某个窗口或设备与设备环境类的句柄指针关联起来，所有的绘图操作都与该句柄有关。而 GDI+不再使用这个设备环境或句柄，取而代之是使用 Graphics 对象。与设备环境相类似，Graphics 对象也是将屏幕的某一个窗口与之相关联，并包含绘图操作所需要的相关属性。但是，只有这个 Graphics 对象与设备环境句柄还存在着联系，其余的如 Pen、Brush、Image 和 Font 等对象均不再使用设备环境。

- Pen、Brush, Font, Image 等对象是图形对象独立的

画笔对象能与用于提供绘制方法的图形对象分开创建和维护，Graphics 绘图方法直接将 Pen 对象作为自己的参数，从而避免了在 GDI 中使用 SelectObject 进行繁琐的切换，类似的还有 Brush、Path、Image 和 Font 等。

- “当前位置”

GDI 绘图操作(如画线)中总存在一个被称为“当前位置”的特殊位置。每次画线都是以此当前位置为起始点，画线操作结束之后，直线的结束点位置又成为了当前位置。设置当前位置的理由是为了提高画线操作的效率，因为在一些场合下，总是一条直线连着另一条直线，首尾相接。有了当前位置的自动更新，就可避免每次画线时都要给出两点的坐标。尽管有其必要性，但是单独绘制一条直线的场合总是比较多的，因此 GDI+取消这个“当前位置”以避免当无法确定“当前位置”时所造成的绘图的差错，取而代之的是直接在 DrawLine 中指定直线起止点的坐标。

- 绘制和填充

GDI 总是让形状轮廓绘制和填充使用同一个绘图函数，例如 Rectangle。轮廓绘制需要一个画笔，而填充一个区域需要一个画刷。也就是说，不管我们是否需要填充所绘制的形状，我们都需要指定一个画刷，否则 GDI 采用默认的画刷进行填充。这种方式确实给我们带来了许多不便，现在 GDI+将形状轮廓绘制和填充操作分开而采用不同的方法，例如 DrawRectangle 和 FillRectangle 分别用来绘制和填充一个矩形。

- 区域的操作

GDI 提供了许多区域创建函数，如：CreateRectRgn、CreateEllipticRgn、CreateRoundRectRgn、CreatePolygonRgn 和 CreatePolyPolygonRgn 等。诚然，这些函数给我们带来了许多方便。但在 GDI+中，由于为了便于将区域引入矩阵变换操作，GDI+简化一般区域创建的方法，而将更复杂的区域创建交由 Path 接管。由于 Path 对象是与设备环境分离开来的，因而可以直接在 Region 构造函数中加以指定。

5、GDI+新特色

GDI+与 GDI 相比，增加了下列新的特性：

- 渐变画刷

以往 GDI 实现颜色渐变区域的方法是通过使用不同颜色的线条来填充一个裁剪区域而达到的。现在 GDI+拓展了 GDI 功能，提供线型渐变和路径渐变画刷来填充一个图形、路径和区域，甚至也可用来绘制直线、曲线等。这里的路径可以视为由各

种绘图函数产生的轨迹。

● 样条曲线

对于曲线而言，最具实际意义的莫过于样条曲线。样条曲线是在生产实践的基础上产生和发展起来的。模线间的设计人员在绘制模线时，先按给定的数据将型值点准确地“点”到图板上。然后，采用一种称为“样条”的工具(一根富有弹性的有机玻璃条或木条)，用压铁强迫它通过这些型值点，再适当调整这些压铁，让样条的形态发生变化，直至取得合适的形状，才沿着样条画出所需的曲线。如果我们把样条看成弹性细梁，那么压铁就可看成作用在这梁上的某些点上的集中力。GDI+的 Graphics:: DrawCurve 函数中就有一个这样的参数用来调整集中力的大小。除了样条曲线外，GDI+还支持原来 GDI 中的 Bezier 曲线。

● 独立的路径对象

在 GDI 中，路径是隶属于一个设备环境(上下文)，也就是说一旦设备环境指针超过它的有效期，路径也会被删除。而 GDI+是使用 Graphics 对象来进行绘图操作，并将路径操作从 Graphics 对象分离出来，提供一个 GraphicsPath 类供用户使用。这就是说，我们不必担心路径对象会受到 Graphics 对象操作的影响，从而可以使用同一个路径对象进行多次的路径绘制操作

● 矩阵和矩阵变换

在图形处理过程中常需要对其几何信息进行变换以便产生复杂的新图形，矩阵是这种图形几何变换最常用的方法。为了满足人们对图形变换的需求，GDI+提供了功能强大的 Matrix 类来实现矩阵的旋转、错切、平移、比例等变换操作，并且 GDI+还支持 Graphics 图形和区域(Region)的矩阵变换。

● Alpha 通道合成运算

在图像处理中，Alpha 用来衡量一个像素或图像的透明度。在非压缩的 32 位 RGB 图像中，每个像素是由四个部分组成：一个 Alpha 通道和三个颜色分量(R、G 和 B)。当 Alpha 值为 0 时，该像素是完全透明的，而当 Alpha 值为 255 时，则该像素是完全不透明。Alpha 混色是将源像素和背景像素的颜色进行混合，最终显示的颜色取决于其 RGB 颜色分量和 Alpha 值。它们之间的关系可用下列公式来表示。

显示颜色=源像素颜色 X alpha / 255 + 背景颜色 X (255 - alpha) / 255 GDI+的 Color 类定义了 ARGB 颜色数据类型，从而可以通过调整 Alpha 值来改变线条、图像等与背景色混合后的实际效果。

● 多图片格式的支持

GDI+提供了对各种图片的打开，存储功能。通过 GDI+，我们能够直接将一幅 BMP 文件存储成 JPG 或其它格式的图片文件。

除了上述新特性外，GDI+还将支持重新着色、色彩修正、消除走样、元数据以及 Graphics 容器等特性。

4.2 GDI+在本系统中的应用

4.2.1 绘制图形、图像

1、画笔的用法

画笔常用于绘制图形的轮廓。GDI+的画笔除了具有常见的色彩和宽度属性外，还具有对齐方式，线帽，变换方式等属性。GDI+中通过 Pen 类来定义画笔对象。

● 构造与使用画笔

```
Pen(color, width); //用颜色与线宽构造一个画笔
```

```
Pen(brush, width); //用画刷与宽度构造一个画笔
```

● 设置宽度与对齐方式

创建画笔时，可以把宽度当作参数传给构造函数，我们也可以使用 SetWidth() 方法来改变画笔的线宽，例如下面的两种方法。

```
Pen blackPen(Color(255, 0, 0, 0), 1);
```

```
blackPen.SetWidth(2);
```

对于画笔，可以通过 SetAlignment 函数设置对其方式，例如

```
blackPen.SetAlignment(PenAlignmentCenter); //把画笔设为中对齐
```

● 设置笔帽

可以用多种方式来绘制线条的头部与尾部形状，GDI+支持圆形，方形，菱形，与箭头等样式的笔帽。设置笔帽的函数为 SetStartCap() 和 SetEndCap()。

● 设置两条直线的连接形

GDI+提供了当两条直线连接时连接处形状的设置，有四种方式：斜接(miter)、斜切(bevel)，圆形(round)，剪裁斜接(miter clipped)。连接形的设置通过 SetLineJoin() 函数来完成。

● 自定义线型

GDI+为我们提供了好多线型，如果满足不了我们的需求，我们可以用成员函数 SetDashPattern 利用一个预定义的数组来描述画笔的虚实。

2、画刷的用法

基于线条的、封闭的图形需要画笔来表现，封闭的形状的典型的特点在于它可以用图形、颜色或模式来填充。画刷是具有颜色，图形或模式的对象，它用来填充图形内部，这种定义也意味着根据不同的目的有不同的画刷。为了满足这一目标，.NET 框架在不同的名字空间中使用不同的类来支持画刷。所有画刷的基类是 Brush 类。GDI+一共提供了四种画刷

● 实心画刷

最简单的画刷就是实心的，这种画刷只提供了一种颜色，习惯于用它来填充一个区域。为了获得实心画刷，需要使用 SolidBrush 类。它只有一个构造函数，语法如下：SolidBrush(Color color); color 参数必须是有效定义的颜色值。

● 网格画刷

网格画刷根据已画或已设计的模式来设置它的填充类型。为了支持网格画

刷，.NET 框架提供了画刷可以使用的各种模式，这些预先设计的模式对应于网格画刷的类型。这意味着在使用网格画刷时，必须在可用的各种类型中选择想要使用的模式，为了使填充区域更生动，还需要说明所使用的颜色。为了获取一个网格画刷，可以使用 `HatchBrush` 类，它的其中一个构造函数使用以下语法：`HatchBrush(HatchStyle style, Color foreColor)`；参数 `foreColor` 是用来绘制模式的颜色，`style` 参数是想使用的网格类型。

● 纹理画刷

网格画刷依赖于已经预先设计的模式来填充图形，在某些情况下，需要设计自己的图案来填充一个图形。要这么做，必须执行两个步骤，首先设计一个图形并存储为一个文件，然后创建一个纹理画刷，并将图案传递给它。

纹理画刷拥有图案，并且通常使用它来填充封闭的图形。为了对它初始化，可以使用一个已经存在的别人设计好了的图案，或使用常用的设计程序设计的自己的图案，同时应该使图案存储为常用图形文件格式，如 BMP 格式。有了图案，这时候就可以使用 `TextureBrush` 类，它有多种构造函数，最简单的构造函数只有一个 `Image` 对象做为参数，这个构造函数的语法是：`TextureBrush(Image *bitmap)`；这个构造函数使用位图作为参数。初始化画刷后，你可以使用它来填充封闭的图形。

● 渐变画刷

渐变画刷类似于实心画刷，因为它也是基于颜色的，与实心画刷不同的是，渐变画刷使用两种颜色，它的主要特点是，在使用过程中，一种颜色在一端，而另外一种颜色在另一端。在中间，两种颜色融合产生过渡或衰减的效果。渐变画刷有两种：线性画刷和路径画刷。定义函数分别为 `LinearGradientBrush()` 和 `LinearGradientBrush()`。

4.2.2 坐标变换

1、世界坐标系、设备坐标系和页面坐标系

GDI+为我们提供了三种坐标空间：世界坐标系、页面坐标系和设备坐标系。

世界坐标系是应用程序用来进行图形输入输出所使用的一种与设备无关的笛卡尔坐标系。通常，我们可以根据自己的需要和方便定义一个自己的世界坐标系，这个坐标系称为用户坐标系。例如，前面 `DrawLine(&newPen, 20, 10, 200, 100)`；中的坐标都是以这个用户坐标系为基准的，默认时使用像素为单位。

设备坐标系是指显示设备或打印设备坐标系下的坐标，它的特点是以设备上的象素点为单位。对于窗口中的视图而言，设备坐标的原点在客户区的左上角，x 坐标从左向右递增，y 坐标自上而下递增。由于设备的分辨率不同，相同坐标值的物理位置可能不同。如对于边长为 100 的正方形，当显示器为 640 x 480 和 800 x 600 时的大小是不一样的。

页面坐标系是指某种映射模式下的一种坐标系。所谓映射是指将世界坐标系通过某种方式进行的变换。默认时，设备坐标和页面坐标是一致的。

2、坐标映射和坐标原点的设置

为了保证打印或显示的结果不受设备的影响，GDI+定义了一些映射方法和属性来决定设备坐标和页面坐标之间的关系。这些映射方法和属性有：

- SetPageUnit 和 GetPageUnit

这两个属性函数是用来设置和获取每个单位所对应的实际度量单位。它通常可以有以下的值：

UnitDisplay — 每个页面单位为 1/75 英寸；

UnitPixel — 每个页面单位为 1 个像素，此时页面坐标与设备坐标相同；

UnitPoint — 每个页面单位为 1/72 英寸；

UnitInch — 每个页面单位为 1 英寸；

UnitDocument — 每个页面单位为 1/300 英寸；

UnitMillimeter — 每个页面单位为 1 毫米。

- SetPageScale 和 GetPageScale

GDI+的这两个属性函数分别用来设置和获取页面的缩放比例。

- TranslateTransform

GDI+ 的 TranslateTransform 方法用来改变坐标的原点位置，例如 TranslateTransform(100, 50) 是将坐标原点移到点(100, 50)。

4.2.3 映射模式

当 Windows 应用程序在其客户区绘制图形时，必须给出在客户区的位置，其位置用 x 和 y 两个坐标表示，x 表示横坐标，y 表示纵坐标。在所有的 GDI+绘制函数中，这些坐标使用的是一种“逻辑单位”。当 GDI+函数将输出送到某个物理设备上时，Windows 将逻辑坐标转换成设备坐标（如屏幕或打印机的像素点）。逻辑坐标和设备坐标的转换是由映射模式决定的。映射模式被储存在设备环境中。GetMapMode 函数用于从设备环境得到当前的映射模式，SetMapMode 函数用于设置设备环境的映射模式。

1、逻辑坐标

逻辑坐标是独立于设备的，它与设备点的大小无关。使用逻辑单位，是实现“所见即所得”的基础。当程序员在调用一个画线的 GDI+函数 DrawLine，画出 25.4mm(1 英寸) 长的线时，他并不需要考虑输出的是何种设备。若设备是 VGA 显示器，Windows 自动将其转化为 96 个像素点；若设备是一个 300dpi 的激光打印机，Windows 自动将其转化为 300 个像素点。

2、设备坐标

Windows 将 GDI+函数中指定的逻辑坐标映射为设备坐标，在所有的设备坐标系统中，单位以像素点为准，水平值从左到右增大，垂直值从上到下增大。Windows 中包括以下 3 种设备坐标，以满足各种不同需要：

(1) 客户区域坐标，包括应用程序的客户区域，客户区域的左上角为 (0, 0)。

(2) 屏幕坐标，包括整个屏幕，屏幕的左上角为 (0, 0)。屏幕坐标用在 WM_MOVE 消息中（对于非子窗口）以及下面的 Windows 函数中：CreateWindow 和 MoveWindow（都对于非子窗口）、GetMessage、GetCursorPos、GetWindowRect、WindowFromPoint 和 SetBrushOrg

中。用函数 ClientToScreen 和 ScreenToClient 可以将客户区域坐标转换成屏幕区域坐标，或反之。

(3)全窗口坐标，包括一个程序的整个窗口，包括标题条、菜单、滚动条和窗口框，窗口的左上角为 (0,0)。使用 GetWindowDC 得到的窗口设备环境，可以将逻辑单位转换成窗口坐标。

3、逻辑坐标与设备坐标的转换方式

映射方式定义了 Windows 如何将 GDI+函数中指定的逻辑坐标映射为设备坐标。我们将逻辑坐标所在的坐标系称为“窗口”，将设备坐标所在的坐标系称为“视口”。“窗口”依赖于逻辑坐标，可以是像素点、毫米或程序员想要的其他尺度。“视口”依赖于设备坐标（像素点）。通常，视口和客户区域等同。但是，如果程序员用 GetWindowDC 或 CreateDC 获取了一个设备环境，则视口也可以指全窗口坐标或屏幕坐标。点 (0,0) 是客户区域的左上角。x 的值向右增加，y 的值向上增加。对于所有映射模式，Windows 都用下面两个公式将窗口坐标转换成视口坐标：

$$xViewport=(xWindow-xWinOrg)*(xViewExt/xWinExt)+xViewOrg$$

$$yViewport=(yWindow-yWinOrg)*(yViewExt/yWinExt)+yViewOrg$$

其中，(xWindow,yWindows) 是待转换的逻辑点，(xViewport,yViewport) 是转换后的设备点。如果设备坐标是客户区域坐标或全窗口坐标，则 Windows 在画一个对象前，还必须将这些坐标转换成屏幕坐标。

Windows 还能将视口（设备）坐标转换为窗口（逻辑）坐标：

$$xWindow=(xViewport-xViewOrg)*(xWinExt/xViewExt)+xWinOrg$$

$$yWindow=(yViewport-yViewOrg)*(yWinExt/yViewExt)+yWinOrg$$

可以使用 Windows 提供的两个函数 DPtoLP 和 LPtoDP 在设备坐标及逻辑坐标之间互相转换。

4.2.4 提高绘图速度

1、双缓存技术

在图形图像处理过程中，当显示绘制的图像时，有时会出现闪烁的情况，解决的办法是使用“双缓存”技术。它的基本原理就是：先在内存中开辟一块虚拟画布，然后将所有需要画的图形先画在这块“虚拟画布”上，最后一次性将整块画布画到真正的窗体上。因为所有的单个图形的绘制都不是真正的调用显示系统来“画”，所以不会占用显示系统的开销，极大的提高绘图效率。实现双缓冲的具体步骤

- 1) 在内存中建立一块“虚拟画布”：`Bitmap bmp = new Bitmap(600, 600);`
- 2) 获取这块内存画布的 Graphics 引用：`Graphics g = Graphics.FromImage(bmp);`
- 3) 在这块内存画布上绘图：`g.FillEllipse(brush, i * 10, j * 10, 10, 10);`
- 4) 将内存画布画到窗口中：`this.CreateGraphics().DrawImage(bmp, 0, 0).`

2、局部刷新技术

GDI+为了提高重绘的效率设置了裁剪区。裁剪区的作用就是：只有在这个区内的绘图过程才会真正有效，在区外的是无效的，即使在区外执行了绘图函数也是不会显示的。

因为多数情况下窗口重绘的产生是因为窗口部分被遮挡或者窗口有滚动发生，改变的区域并不是整个图形而只是一小部分，这一部分需要改变的就是 pDC 中的裁剪区了。因为显示比绘图过程的计算要费时得多，有了裁剪区后显示的就只是应该显示的部分，大大提高了显示效率。比如，通过鼠标拖的方式改变正在绘制的图形，可能改变的仅仅是图形的一部分，而不是整个图形，这时，可以使用 SetClipBox() 函数来设置裁剪区，新设置的裁剪区仅仅是图形中需要重新绘制的区域。在绘制图形时，虽然调用绘图函数绘制了全部图形，但只有裁剪区内部的图形才会被绘制，从而大大提高了绘图速度。

4.3 数据及其存储

4.3.1 应用类

1、数据定义

在应用类中维护一个页面对象数组，数组中的每个元素是一个页面。

```
class CTeachDoc.: public CDocument
{
    CObArray m_objPages; //记录所有页面信息,每个元素为一个 CPage 对象
    int m_nCaretIndex; //记录闪烁光标在第几个幻灯片后面(0 表示最上端)
    CPage *m_pCurrentPage;
}
```

2、序列化操作

在应用的序列化操作中，依次调用对象数组中的各个页面对象的序列化操作。

```
void CTeachDoc::Serialize(CArchive& ar)
{
    CPage *pPage;
    int count;
    if (ar.IsStoring())
    {
        //写文件
        count = m_objPages.GetCount();
        ar<<count; //记录页面数量
        //依次写各个页面
        for (int i=0; i<count; i++)
        {
            pPage = (CPage*)(m_objPages.GetAt(i));
            ar<<pPage;
        }
        ar<<CPage::s_nIDForNewItem;
    }
    else
    {
        //读文件，建立各个构件对象
        ar>>count; //读取页面数量
        //依次读各个页面
```

```

        for (int i=0; i<count; i++)
        {
            ar>>pPage;
            m_objPages.Add(pPage);

        }
        ar>>CPage::s_nIDForNewItem;
        //设置当前页面
        SetSelectedPage((CPage*)(m_objPages.GetAt(0)));
        m_nCaretIndex = -1;
        //读完文件后进行构件绘制
        UpdateAllViews();
    }
}

```

4.3.2 页面类

1、数据定义

页面类中除维护一个构件对象数组外，还记录与该页面相关的一些临时属性，这些属性与页面的绘制相关或与页面上的构件的操作相关。

```

class CPage : public CObject
{
    DECLARE_SERIAL(CPage) //为了支持序列化操作
    static int s_nIDForNewItem;
    //记录文档对象指针
    static CDocument *s_pDoc;
    static CView *s_pLeftView;
    static CView *s_pRightView;
    //记录页面的缩放比例
    static float s_fScale_x;
    static float s_fScale_y;
    //是否正在进行某种操作
    static bool s_bAdding;           //添加
    static bool s_bMoving;          //移动
    static bool s_bResizing;        //缩放
    static bool s_bKeyPointMoving; //移动关键点
    static bool s_bRotating;        //旋转
    static bool s_bOutlining;       //框选
    static bool s_bActioning1;
    static bool s_bActioning2;
    //记录鼠标点击和移动的点
    static CPoint s_ptBegin; //记录点击点
    static CPoint s_ptEnd;   //记录移动点
    //将要添加的构件的类型
    static int s_nNewItemType;
    //自定义剪贴板数据格式标示

```

```

static UINT s_uClipboardPage; //页面
static UINT s_uClipboardItem; //构件
//当前页面的鼠标光标形状
static HCURSOR s_hCursor;
//当前是否有键盘按下
static bool s_bKeyDown;
//是否在播放幻灯片
static bool s_bPlaying;
static int s_nLeftMargin;
static int s_nTopMargin;
static int s_nClientWidth;
static int s_nClientHeight;
static bool s_bRefreshOutline;
}

```

2、序列化操作

```

void CPage::Serialize(CArchive& ar)
{
    CItem *pItem;
    int count;//记录页面上构件的数量
    if (ar.IsStoring())
    { //保存文件
        ar<<m_colorBackGround;
        count = m_objItem.GetCount();
        ar<<count;
        for (int i=0; i<count; i++)
        {
            pItem = (CItem*)(m_objItem.GetAt(i));
            ar<<pItem;
        }
    }
    else
    { //读取文件
        ar>>m_colorBackGround;
        ar>>count;
        for (int i=0; i<count; i++)
        {
            ar>>pItem;
            m_objItem.Add(pItem);
        }
    }
}

```

4.3.3 构件类

1、数据定义

构件基类中只记录所有构件的共有属性，各类构件自己特有的一些属性在各个子类中分别定义。


```

class CItem : public CObject
{
    DECLARE_SERIAL(CItem) //为了支持序列化操作
    //旋转构件时记录旋转前的信息
    static float s_fRotateAngle_Temp; //角度
    static Point s_ptRotateCenter_Temp; //中心点
    static Point s_ptRotateEnd_Temp; //另一端点
    static int s_nOldHeight; //旋转前高度
    //基本属性
    int m_nID;
    int m_nType; //构件类型
    bool m_bSelected; //是否选中
    float m_fRotateAngle; //旋转角度
    bool m_bChange; //是否改动(便于处理撤销操作)
    //与旋转有关的3个点
    Point m_ptRotateCenter; //旋转中心
    Point m_ptRotateEnd; //旋转的另一端点
    Point m_ptRotateEdge; //旋转射线与构件的交点
    //控制点,用来确定构件所在的矩形
    Point m_ptP1; //控制点1
    Point m_ptP2; //控制点2
    //关键点,用来控制构件的几何规格
    CArray<MyPoint,MyPoint> m_ptKeyPoint; //关键点集
    int m_nCurKeyPoint; //当前正在控制的关键点
    //移动或缩放构件前,记录m_ptP1和m_ptP2相对于鼠标落下时所在点的距离
    int m_nTempX1;
    int m_nTempY1;
    int m_nTempX2;
    int m_nTempY2;
    //其他
    Point m_ptP1_Old;
    Point m_ptP2_Old;
    float m_fScaleX;
    float m_fScaleY;
    //移动(或缩放)构件前,用来记录构件的宽度和高度的比例关系
    float m_fScaleXY;
    //记录缩放时应该改变的坐标
    int m_nResize_x; //0:不缩放x坐标,1:缩放m_ptP1.x,2:缩放m_ptP2.x
    int m_nResize_y; //0:不缩放y坐标,1:缩放m_ptP1.y,2:缩放m_ptP2.y
    //旋转时记录旋转前的角度
    float m_fRotateAngle_Old;
    //记录m_ptP2旋转前的坐标(按住Ctrl键进行旋转时需要)
    Point m_ptRotateCenter_Old;
}

```

2、序列化操作

```
void CItem::Serialize(CArchive &ar)
{
    MyPoint mpt;
    int nKeyPtCount;
    if (ar.IsStoring())
    { //保存文件
        //基本属性
        ar<<m_nType<<m_fRotateAngle<<m_ptP1.X<<m_ptP1.Y<<m_ptP2.X<<m_ptP2.Y;
        ar<<m_nID;
        //关键点
        nKeyPtCount = m_ptKeyPoint.GetCount();
        ar<<nKeyPtCount;
        for (int i=0; i<nKeyPtCount; i++)
        {
            mpt = m_ptKeyPoint.GetAt(i);
            ar<<mpt.kx<<mpt.ky;
        }
    }
    else
    { //读取文件
        //基本属性
        ar>>m_nType>>m_fRotateAngle>>m_ptP1.X>>m_ptP1.Y>>m_ptP2.X>>m_ptP2.Y;
        ar>>m_nID;
        //关键点
        ar>>nKeyPtCount;
        for (int i=0; i<nKeyPtCount; i++)
        {
            ar>>mpt.kx>>mpt.ky;
            m_ptKeyPoint.Add(mpt);
        }
        //设置旋转中心
        InitialRotatePoint();
        //初始化刷新区域
        AddRgnOfSelect();
        AddRgnOfRect(m_ptP1.X, m_ptP1.Y, m_ptP2.X, m_ptP2.Y, true);
    }
}
```

4.4 功能实现

由于页面类的大多数操作与构件类的操作实现方法类似，这里仅就构件基类的功能实现方法予以介绍。下面是与构件基类的各种操作有关的方法，在接下来的小节中将对各个操作的实现方法分别进行介绍。

```
class CItem : public CObject
{ //方法
```

```

//序列化操作
virtual void Serialize(CArchive &ar);
//绘制
virtual void Draw(Graphics &g){}; //在给定的 Graphics 上绘制该构件
virtual void DrawSelect(Graphics &g); //构件选中时,绘制控制点和旋转射线
virtual void DrawFrame(Graphics &g); //绘制构件的框架
//在点(x,y)处绘制小圆圈(主要是选中构件时绘制八个控制点用)
void DrawSelectedPoint(Graphics &g, int x, int y);
void DrawRotateLine(Graphics &g); //绘制旋转射线和点
void DrawKeyPoint(Graphics &g); //绘制关键点
void DrawArrow(Graphics &g, Pen &pen, int x1,int y1,int x2,int y2, int len, int width, bool
close=false); //绘制箭头
void FillArrow(Graphics &g, Brush &brush, int x1,int y1,int x2,int y2, int len, int width); //箭头
//鼠标光标在构件上的相对位置
//根据鼠标光标点 point,判断是否落在构件关键点或构件上,并返回相应的形状
virtual HCURSOR GetCursor(UINT nFlags, CPoint point);
//判断 point 点是否在构件上,如果在则返回准备移动的光标形状
virtual HCURSOR GetCursorForMove(UINT nFlags, CPoint point);
virtual bool PointInItem(CPoint point); //判断 point 是否落在构件上
//判断构件是否在 point1 和 point2 所界定的区域内
virtual bool ItemInRect(CPoint point1, CPoint point2);
virtual int AtControlPoint(CPoint point); //判断 point 是否落在关键点上
//鼠标和键盘事件
virtual bool OnLButtonDown(UINT nFlags, CPoint point){return false;};
virtual bool OnMouseMove(UINT nFlags, CPoint point){return false;};
virtual bool OnLButtonUp(UINT nFlags, CPoint point){return false;};
virtual bool OnLButtonDbkClk(UINT nFlags, CPoint point){return false;};
virtual bool OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags){return false;};
virtual bool OnKeyUp(UINT nChar, UINT nRepCnt, UINT nFlags){return false;};
virtual bool OnChar(UINT nChar, UINT nRepCnt, UINT nFlags){return false;};
virtual bool OnRButtonDown(UINT nFlags, CPoint point){return false;};
virtual bool OnRButtonUp(UINT nFlags, CPoint point){return false;}; //virtual
bool OnRDbkClk(UINT nFlags, CPoint point){return false;};
//添加构件
virtual void SetBeginPoint(CPoint point); //添加构件时,设置起始点
virtual void SetEndPoint(CPoint point); //添加构件时,设置结束点
virtual void EndAdd();
virtual int GetDefaultWidth(){return 100;}; //获得默认宽度
virtual int GetDefaultHeight(){return 100;}; //获得默认高度
//移动构件
//设置控制点相对于 point 的位置,以便在移动鼠标时保持控制点与光标的位置关系
virtual void SetRelativeLength(CPoint point);
virtual void Move(UINT nFlags, CPoint point); //移动构件
virtual void PrepareMove();

```

```

virtual void EndMove();
//缩放构件
//设置构件的当前缩放点, 将要缩放构件时记录缩放点, 以便在移动鼠标时让缩放点随鼠标光标的移动而移动
virtual void SetCurResizePoint(int arrowDir);
virtual void Resize(UINT nFlags, CPoint point); //缩放构件
virtual void PrepareResize();
virtual void EndResize();
//旋转构件
//旋转构件(绕旋转中心由 m_ptRotateEnd 旋转到 point)
virtual void Rotate(UINT nFlags, CPoint point);
//计算点 point 绕 center 旋转 angle 角度后的坐标点
CPoint RotatePoint(CPoint center, CPoint point, float angle, bool clockwise=false); //初始化旋转中心(序列化时不保存旋转中心,在打开文件时调用此函数设置旋转中心)
void InitialRotatePoint();
void ReSetRotatePoint();
virtual void PrepareRotate();
virtual void EndRotate(); //移动和缩放时,重新设置旋转中心
//关键点
virtual void InitialKeyPoint({}); //设置关键点
virtual void MoveKeyPoint(CPoint point); //移动关键点
//测试关键点的新位置是否满足构件自身的要求
virtual void TestKeyPoint(int index, float &kx, float &ky);
void AddKeyPoint(float kx, float ky); //添加关键点
virtual void PrepareMoveKeyPoint();
virtual void EndMoveKeyPoint();
//其他
virtual bool OverLayed(CItem *pItem); //判断 pItem 是否与自己重合
//判断控制点是否满足构件的几何约束,如果不满足,可以返回 false,取消或者进行强制设置
virtual bool IsValidate(){return true;};
bool IsAdjacence(CPoint point1, Point point2); //判断 2 个点是否接近
float GetP(float x1,float y1, float x2,float y2, float r, int type, bool getx);
float GetAngleOfTwoVector(CPoint p1, CPoint p2, CPoint ptCenter); //计算向量夹角
CRect GetRect(); //返回构件绘制区域所界定的矩形
Point LpToDp(Point point); //将逻辑坐标转成设备坐标 (控制 Flash 窗口时会用到)
//区域
void AddRgnOfPoint(int x, int y);
void AddRgnOfLine(int x1, int y1, int x2, int y2);
void AddRgnOfRect(int x1, int y1, int x2, int y2, bool fill=false);
virtual void AddRgnOfSelect();
void SetSelected(bool select);
void RemoveOverlapRgn(Region *rgn);
//窗体
virtual bool IsWindow(){return false;};

```

```
virtual bool IsPlaying(){return false;};  
virtual void InActiveWindow(){};  
};
```

4.4.1 添加

构件的添加可以采用两种方式：鼠标拖拽方式和菜单方式。

1、鼠标拖拽方式

当用户点击工具栏上的构件按钮时，将类型记录在 `CPage::s_nNewItemType` 中，在随后的 `LButtonDown` 事件中建立该构件并设置该点为构件的起始点，在 `LButtonDown` 事件中记录构件的结束点，起始点和结束点所界定的矩形区域便是构件的边界范围。

2、菜单方式

点击某类构件对应的菜单后，直接建立一个该类型的构件并加入页面的构件对象数组中，新建构件的默认宽度和高度由各构件类来确定。

4.4.2 删除

可以通过菜单和键盘两种方式删除当前选中的构件，删除构件前，需要通知被删除的构件进行一些预处理操作。如果该构件允许被删除，首先用一个临时指针记录该构件，然后释放掉该构件，最后将其从页面的构件对象数组中删除；如果构件不允许被删除，则给出不允许被删除的提示。

4.4.3 选择

有两种选择页面上的构件的方式：点击方式和框选方式

1、点击方式

用鼠标点击页面上的某个点时，依次遍历页面上的所有构件，调用构件各自的判断函数 `PointInItem()`，这个函数可以判断一个点是否落在构件上。遍历时需要按照构件加入页面的先后顺序进行（这也是页面上构件的绘制顺序），以便让上层的构件首先被选中。根据函数 `PointInItem()` 的返回值进行选中处理时，应该考虑 `Shift` 键是否同时按下。如果 `Shift` 没有按下，那么当 `PointInItem()` 第一次返回真值时，将当前构件选中并取消其他所有构件的选中状态，结束操作；如果 `Shift` 处于按下状态，对于所有构件均进行下面的处理：如果 `PointInItem()` 返回真值，将该构件在选中和非选中状态键进行切换，也就是说，如果该构件已经被选中，取消选择状态，否则，选中该构件。

2、框选方式

框选方式是指用鼠标在页面上拖拽出一个方框，然后判断构件是否在方框内，框选方式与点击方式的區別就是对函数 `PointInItem()` 的调用改为 `ItemInRect()`，以判断一个构件是否在矩形区域内。框选方式的其他处理步骤与点击方式相同。

4.4.4 恢复

在每一个应用中维护一个页面历史状态对象数组，记录页面对象的变动情况。同时，在每一个页面中也维持一个构件历史状态对象数组，记录页面上构件的状态变化情况。当构件状态发生变动时，首先将标记 `m_bChanged` 设置为 `true`，然后由页面类负责将构件原来的状态保存到页面的构件历史状态对象数组中；当页面对象发生变动时，则有由

应用类负责将页面原来的状态保存到应用的页面历史状态对象数组中。在恢复时，首先找到最近放入历史状态对象数组中的对象，然后分析其历史操作类型，根据不同的历史操作类型进行相应的恢复处理。例如：如果原来的操作是删除操作，则直接将被删除对象加入应用或页面的对象数组即可；如果原来的操作是修改了构件的某一个属性，直接恢复该属性即可。进行一次恢复后，需要将恢复的对象从历史状态对象数组中删除。

4.4.5 复制、剪切、粘贴

通过剪贴板来完成页面和构件的复制、剪切和粘贴操作。当对页面进行上述操作时，需要同时对页面上的构件进行相同的操作。复制时，首先将被复制的对象进行序列化，然后将序列化后的文件送入 Windows 剪贴板；剪切与复制操作基本相同，不过序列化后需要将剪切的对象删除。下面以复制页面及其构件为例来介绍 Windows 剪贴板的使用。

```
void CPage::CopyItems()
{
    //建立 Window 共享文件
    CSharedFile shareFile(GMEM_MOVEABLE|GMEM_DDESHARE|GMEM_ZEROINIT);
    //将共享文件与 ar 关联，以便进行序列化操作
    CArchive ar(&shareFile, CArchive::store);
    //序列化操作
    int count = GetSelCount(); //获取构件数量
    ar<<count;
    CItem *pItem;
    //依次序列化页面上的所有构件
    for (int i=0; i<m_objItem.GetCount(); i++)
    {
        pItem = (CItem*)(m_objItem.GetAt(i));
        if (pItem->m_bSelected)
        {
            ar<<pItem;
        }
    }
    ar.Close(); //关闭序列化
    HGLOBAL hMem = shareFile.Detach(); //获得共享文件的句柄
    if (!hMem)
    {
        return;
    }
    //将共享文件句柄送入剪贴板
    COleDataSource *pSource = new COleDataSource();
    pSource->CacheGlobalData(CPage::s_uClipboardItem, hMem);
    pSource->SetClipboard();
}
```

4.4.6 移动

构件的移动有两种方法：键盘控制和鼠标拖拽。

1、键盘控制

通过键盘上的方向键对选中的构件进行移动，这种移动方式是对构件位置的一种细微调整。

2、鼠标拖拽

点击选中的构件，按住鼠标左键可以移动构件。点击构件时，记录鼠标落下的点在构件上的相对位置，当鼠标移动到一个新的位置点后，根据上面记录的相对位置来确定构件的新的位置。这里需要注意的是，多个被选中的构件可以同时移动，所有选中的构件均需要记录鼠标落下的点在构件自身上的相对位置。由于用鼠标移动构件是一种连续的操作，为了提高速度，在拖拽过程中，默认情况下只绘制构件的轮廓，结束拖拽后再绘制完整的构件。如果移动时按住 shift 键，只能在水平或垂直方向进行移动，如果按住 Ctrl 键，则克隆一个构件。

4.4.7 缩放

点击选中的构件，用鼠标拖拽构件的八个方向控制点（图 7 中的八个小圆圈），可以对构件进行缩放。

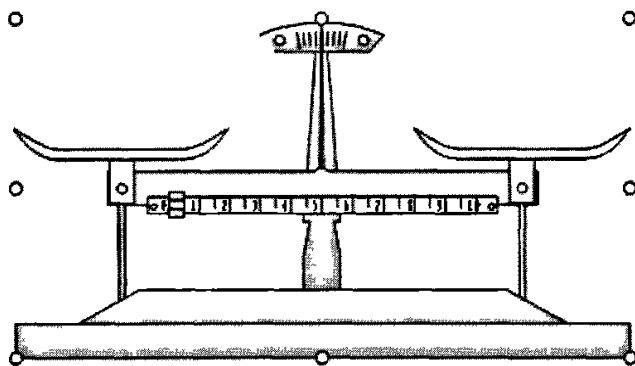


图 7 构件控制点

缩放前，根据鼠标所点击的控制点，记录下缩放的方向，在缩放过程中，根据鼠标当前位置和缩放方向，调整构件的位置。需要指出的是，当多个构件被同时缩放时，还需要在各个构件中记录鼠标落下的点在构件自身上的相对位置。再有，如果缩放时按住 shift 键，无论缩放哪个控制点，构件的纵横比例关系不能改变；如果按住 Ctrl 键，则以构件中心为对称点，进行对称缩放。

4.4.8 旋转

构件处于选中状态时，会显示一个旋转控制点（如图 8 所示），用鼠标拖拽该旋转点可以旋转构件。旋转时，根据鼠标旋转前后的位置计算出旋转的角度，然后用原来的倾斜角度加上本次旋转的角度，作为构件的新的倾斜角度。

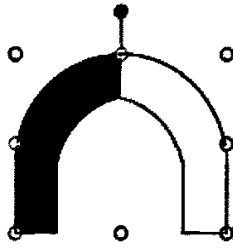


图8 旋转控制点

4.4.9 关键点控制

关键点用来控制构件的各个组成部分的位置，例如：滑动变阻器的滑块的位置可以用一个关键点来控制。一个关键点由2个参数 kx 和 ky 构成。关键点中的两个参数 kx, ky 分别表示关键点的 x, y 坐标与 m_ptP1 和 m_ptP2 的相对位置（如图9所示）。

$$\text{即 } kx = (x - m_ptP1.X) / (m_ptP2.X - m_ptP1.X)$$

$$ky = (y - m_ptP1.Y) / (m_ptP2.Y - m_ptP1.Y)$$

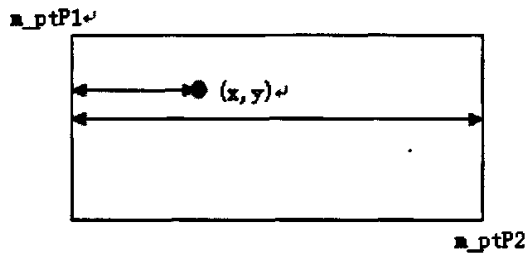


图9 关键点

每个构件具有一个关键点列表，列表中的所有关键点均可以通过鼠标拖拽的方式进行位置调整，在调整关键点位置时，构件可以根据自己的需要对关键点的位置设置一些约束条件。

4.4.10 属性编辑

在编辑状态下双击一个构件，可以打开该类构件的属性表，不同构件类具有不同的属性表，如下面图10是数学统计图学科构件的属性表。在打开属性表前，首先需要判断构件的类型，然后打开对应的属性表。



图10 统计图属性表

4.4.11 图形绘制

图形的绘制以页面为单位，绘制时，首先绘制页面背景，然后由页面类依次调用各个构件的绘制函数进行绘制，所有构件的绘制函数都接受一个 GDI+ 设备对象 Graphics，然后将图形、图像绘制到 Graphics 上。为了提高绘制速度，采用了前面介绍的双缓存技术和局部刷新技术。

第五章 基于本平台的学科构件实例

在本章中，利用前面提到的学科工具构件模型，开发了一套学科工具构件实例：物理电学实验构件。利用提供的这些电子元器件，能够模拟电学实验。操作者可以应用这些电子元器件，快速设计和搭建自己需要的电路、设置元器件的参数、探索物理电学现象和规律，系统还能够根据用户操作和电学原理自动生成可独立运行的教学动画软件。这不仅可供学生自主探究物理现象中蕴含的电学规律；又能满足教师制作个性化课件的需求，从而在信息化教学环境中丰富教学方法和手段，极大地减轻教师的负担、提高教学效率和质量。

5.1 建立电学构件的意义

实验是物理教学活动中非常重要的一个环节，对于培养学生的实际操作能力和解决问题能力至关重要，在物理电学知识部分，实验尤其重要。相对于传统实验方法，虚拟实验在如下几个方面具有其独特的优势。

1、借助于动画进行动态讲解，生动形象

在进行实验前，教师需要在课堂上（不是实验室里）讲解器件的操作方法和实验步骤。在传统的教学过程中，教师只能对实验过程进行讲解，而不能掩饰过程；借助电学构件，可以展现实验过程，使讲解变得生动、形象。

2、为学生提供一个自主探索的平台

在传统的实验过程中，考虑到实验的危险性和设备的安全性，学生只能按照规定的正确的操作方法进行实验，不允许学生违规操作，而在虚拟实验室里，学生可以尝试各种操作，进行自主探索。例如，学生可以反接电流表，可以将电流表的3个接线柱全部接上导线，进而根据实验结果去深入思考、探索。

3、弥补教学条件的不足

在一些学校中，由于缺少实验设备、场地，而使实验无法进行。利用虚拟实验室，可以弥补这些方面的不足，在虚拟的环境中便可以完成各种各样的实验，从而丰富感性认识，加深对教学内容的理解。

电学实验平台是面向中学生和物理教师的一个开放性的、提供多种电子元器件的模拟电学实验室。在平台下，操作者可以应用这些电子元器件，快速设计和搭建自己需要的电路、设置元器件的参数、探索物理电学现象和规律，平台还能够根据用户操作和电学原理自动生成可独立运行的教学动画软件。

5.2 构件的设计与实现

1、构件的种类

本套电学实验构件可以按照实际物理元器件的用途分为三类，电源和电表、开关、电器

和电阻。电源和电表类包括电池、电池组、蓄电池、电压表和电流表；开关类包括单刀单掷开关和单刀双掷开关；电器和电阻类包括灯泡、电铃、电阻、变阻箱和滑动变阻器。

2、构件的几何属性

这套物理电学构件均以图片的形式给出，在图片上设置关键点代表接线柱，导线只能连接在接线柱上，具体图片见表 1。









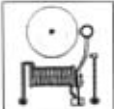

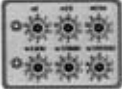
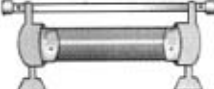
电源和电表类	 电池	 电池组	 蓄电池	 电压表	 电流表
开关类	 单刀单掷开关		 单刀双掷开关		
电器和电阻类	 灯泡	 电铃	 电阻	 变阻箱	 滑动变阻器

表 1 构件的几何属性

在上述器件中，电压表和电流表的指针根据电路中的电压或电流进行绘制，单刀单掷开关具有打开和闭合 2 个状态，单刀双掷开关有打开、左闭合和右闭合 3 个状态，灯泡根据实际功率具有不同的明亮程度，变阻箱的各个档位的指针根据具体数值进行绘制，滑动变阻器的滑块根据用户的调整情况在相应位置进行显示。

3、构件的物理属性

在本平台下，主要考虑的构件的物理属性见表 2。

类别	具体器件	物理属性
电源和电表类	电池	内阻
	电池组	单节内阻、节数
	蓄电池	单节内阻、节数、单节电压
	电压表	3V 和 5V 两个量程对应的内阻
	电流表	0.6A 和 3A 两个量程对应的内阻
开关类	单刀单掷开关	
	单刀双掷开关	
电器和	灯泡	最大电流、起亮电流、额定电压、额定功率

电阻类	电铃	起振电流、内阻
	电阻	内阻
	变阻箱	总内阻
	滑动变阻器	最大阻值、当前阻值

表 2 电学构件物理属性表

4、构件的行为属性及驱动事件

具体器件	行为	驱动事件
电压表	改变指针位置	量程或经过电压表的电流发生变化
电流表	改变指针位置	量程或经过电流表的电流发生变化
单刀单掷开关	打开与闭合	用户用鼠标双击开关的侧刀手柄
单刀双掷开关	打开、左闭合、右闭合	用户用鼠标双击开关的侧刀手柄
灯泡	明暗变化	经过的电流在起亮电流和最大电流之间发生变化
	变黑（烧坏）	经过的电流超出最大电流
电铃	振动（发声）	经过的电流达起振电流
变阻箱	改变各档位的指针位置	用户用鼠标双击各个档位按钮
滑动变阻器	改变滑块的位置	用户用鼠标拖拽滑块

表 3 电学构件行为属性及驱动事件表

5、算法分析

本部分构件涉及的核心算法是用基尔霍夫定律求解电压和电流(节点法)，基本步骤是：

- (1) 根据电路图，找到节点（数目为 N）
- (2) 设变量 $I_1, I_2 \dots I_L$ (共有 L 条之路)， $U_2, U_3 \dots U_N$ (节点电压，指定 $U_1=0$)，一共有 $L+N-1$ 个变量
- (3) 列出电流方程 (N-1 个)，只涉及变量 $I_1, I_2 \dots I_L$ ；
- (4) 列出 $U=RI$ 方程 (L 个)，注意：对于电源，用 $U=U_i-U_j$ 作为方程
- (5) 至此，共有 $L+N-1$ 个方程， $L+N-1$ 个变量，方程数目=变量数目，用高斯列主元小去法进行求解。

在利用节点法进行电压和电力求解中，寻找节点采用的方法是：将所有器件的接线柱作为临时节点，如果节点与三条支路相连，将其作为固定节点，然后遍历所有非固定节点，找出非固定节点所在的支路上的所有器件和支路所连接的固定节点，记录该支路两节点和之路的总电阻。

由于在一个具体的应用中，可能包含多个独立闭合电路，这就需要找出所有独立的闭合电路，采用的方法是：从头到尾遍历链表，如果一个器件能够从正极联出，经过一定的路径从负极联入，则表明该器件在闭合电路中，此时如果已经存在一些独立的闭合电路，判断该器件是否在这些闭合电路中（如果该器件 C 能够从正极联出，经过一定的路径（不走回头）从负极联入，并且经过某个闭合电路中 L 的一个器件，则表明器件 C

在电路 L 中)。

用基尔霍夫定律求解电压和电流(节点法)的实现代码如下:

```
void ItemHead::Calculate(int group)
{
    CString str;
    int N; 节点数
    double V[100]; 电源的电压
    double U[100]; 节点电压
    int L[100][4]; 支路,[[0]和[[1]分别记录支路的两个节点,取由编号大的节点流向编号小的
    节点电流方向为正方向
    double R[100]; 电阻
    int LCount; 支路总数
    double a[100][100],f[100],x[100];
    for (int i=0;i<100;i++)
    {
        f[i]=0;
        x[i]=0;
        V[i]=0;
        U[i]=0;
        for (int j=0;j<100;j++)
            a[i][j]=0;
        for (int j=0;j<4;j++)
            L[i][j]=0;
    }
    //[[0]表示器件ID,[[1]表示器件Port,[[2]表示连接的分支数(>=3即为固定节点),[[3]正式节
    点编号, [[4]-[[4*m]该节点各个支路的另一个端点的MidNode序号
    double MidNode[200][100];
    for (int i=0; i<200; i++)
    {
        for (int j=0; j<100; j++)
            MidNode[i][j] = 0;
    }
    int MidCount=0;
    Eltem *miditem=Item;
    int LC=0;
    / 首先进行电流初始化
    miditem=Item;
    while (miditem!=NULL)
    {
        if (miditem->Group==group)
            miditem->CurCurrent=0;
        miditem=miditem->Next;
    }
    miditem=Item;
    while (miditem!=NULL)
    {
```

```

    MidNode[MidCount][0]=miditem->ID;
    MidNode[MidCount][1]=1;
    MidNode[MidCount][2]=1;
    MidNode[MidCount][4]=MidCount+1;
    MidCount++;
    miditem=miditem->Next;
}
bool addVNode=false;//为电源增加虚拟节点，以便于简单串联电路的计算。
miditem=Item;
int One=0,Another=0;
miditem=Item;
int midID1,midID2,midPort1,midPort2;//记录导线的两个端点的ID和Port
//下面的循环在各个导线所连接的器件之间增加一条支路
miditem=Item;
while (miditem!=NULL)
{
    if (miditem->State==0 || miditem->State==2)//仅处理可见的器件
    {
        if (miditem->Type==1 && miditem->Group==group)//节点之间通过导线相连
        {
            if (!(miditem->NodeID1==miditem->NodeID2 ))
            {
                for (int i=0;i<miditem->Dots.GetCount();i++)
                {
                    MidCount++;
                    int x=miditem->Dots[i].x;
                    int y=miditem->Dots[i].y;
                    MidNode[MidCount][0]=miditem->ID;
                    MidNode[MidCount][1]=i;
                    MidNode[MidCount][2]=0;
                }
            }
        }
        miditem=miditem->Next;
    }
    miditem=Item;
    //取得各个正式节点间的电阻
    GetTrueLandResist(MidNode,MidCount);
    //求得节点数,并将节点编号
    N=0;
    int temp1,temp2;
    for (int i=1;i<=MidCount;i++)
    {
        if (MidNode[i][2]>=3)
        {

```

```

        temp1=MidNode[i][0];
        temp2=MidNode[i][1];
        MidNode[i][3]=N;
        N++;
    }
    else
        MidNode[i][3]=-1;
}
    取得各个支路
    LCount=0;
    for (int i=1;i<=MidCount;i++)
    {
        if ((int)MidNode[i][3]>=0)
        {
            for (int j=1;j<=(int)MidNode[i][2];j++)
            {
                if ((int)MidNode[i][4*j+1]>=0 &&
                    MidNode[(int)(MidNode[i][4*j+1])][3]>MidNode[i][3])
                {
                    LCount++;
                    L[LCount][0]=(int)MidNode[i][3]; //支路的起点的正式节点编号
                    L[LCount][1]=(int)(MidNode[(int)(MidNode[i][4*j+1])][3]);
                    L[LCount][2]=i; //支路的起点的序号
                    L[LCount][3]=(int)(MidNode[i][4*j+1]); //支路的终点的序号
                    R[LCount]=MidNode[i][4*j+2]; //支路LCount(节点i和节点j)之间的电阻
                    V[LCount]=MidNode[i][4*j+3]; //支路LCount(节点i和节点j)之间的电压
                }
            }
        }
    }
}
    if (N+LCount-1<1)
        return;
    //列N-1个电流方程
    for(int i=1;i<=N-1;i++)
    {
        for (int j=1;j<=LCount;j++)
        {
            if (L[j][0]==i)
                a[i][j]=-1;
            else if (L[j][1]==i)
                a[i][j]=1;
            else
                a[i][j]=0;
        }
    }
}
    //列L个U=IR方程

```

```

for (int i=1;i<=LCount;i++)
{
    if (fabs(V[i])>0.000001)
        f[i+N-1]=V[i];
    if ((int)L[i][0]!=0)
        a[i+N-1][LCount+L[i][0]]=1;
    a[i+N-1][LCount+L[i][1]]=-1;
    a[i+N-1][i]=R[i];
}
//显示结果
double result=gauss(a,f,x,N+LCount-1);
for (int i=1;i<=MidCount;i++)
{
    CString str1,str2,str3,str4,str5,str6,str7;
    str1.Format("%d",GetItemFromID((int)MidNode[i][0]->Type));
    str2.Format("%d",(int)MidNode[i][1]);
    str3.Format("%d",i);
    str=str3+":"+str1+"."+str2;
    if (MidNode[i][2]>=3)
    {
        str1.Format("%d",GetItemFromID((int)MidNode[i][0]->Type));
        str2.Format("%d",(int)MidNode[i][1]);
        str3.Format("%d",(int)MidNode[i][3]);
        str=str1+":"+str2+"."+str3;
    }
}
for (int i=1;i<=LCount;i++)
{
    CString str1,str2,str3,str4,str5,str6,str7;
    str1.Format("%d",i);
    str2.Format("%d",GetItemFromID((int)MidNode[L[i][2]][0]->Type));
    str3.Format("%d",GetItemFromID((int)MidNode[L[i][3]][0]->Type));
    str4.Format("%.12f",R[i]);
    str5.Format("%.12f",x[i]);
    str6.Format("%.12f",V[i]);
    str=str1+":"+str2+"."+str3+"."+str4+"."+str5+"."+str6;
}
//计算通过器件的电流
int CurL;
double d_cur;
for (int i=1;i<MidCount;i++)
{
    if (MidNode[i][0]==MidNode[i+1][0] && MidNode[i][1]==1)
    {
        CurL=(int)MidNode[i][98];
        d_cur=x[CurL]*MidNode[i][96];
    }
}

```



```

    GetItemFromID((int)MidNode[i][0])->CurCurrent=d_cur;
    if (GetItemFromID((int)MidNode[i][0])->Type==6 ||
        GetItemFromID((int)MidNode[i][0])->Type==550)
    {
        if (fabs(d_cur)>fabs(GetItemFromID((int)MidNode[i][0])->MaxCurrent))
            GetItemFromID((int)MidNode[i][0])->Valid=false;
    }
}
}
SetLineDir_current(x,group);
miditem=Item;
}

```

5.3 构件的具体应用

1、对教学过程的支持

本构件可以应用在如下几个教与学的过程中。

- (1) 教师课前备课；
- (2) 教师课堂上讲授；
- (3) 教师留实验作业；
- (4) 学生做作业；
- (5) 学生进行探索性试验。

2、构件的使用方式

进行电学实验的一般步骤如下（图 11）。

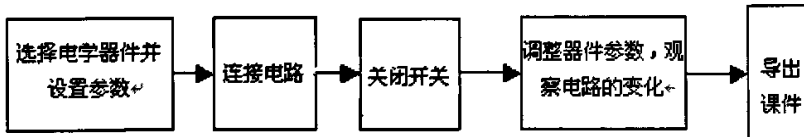


图 11 电学实验步骤

步骤 1：选择电学器件并设置参数。

选取实验将要用到的电学器件，如电池、电阻、开关、电表等，然后设置相应的参数（器件有默认的参数）。

步骤 2：连接电路。

将在上一步中选取的电学器件用导线连接起来，形成完整的电路。

步骤 3：关闭开关。

电路已连接并且器件参数设置完后，将电路中相应的开关闭合，开始研究电流的变化。

步骤 4：调整器件参数，观察电路的变化。

当电路闭合后，可以调整器件参数（如拖动滑动变阻器）观察电路中电流的变化情况。

步骤 5：导出课件。

如果想将已经完成的电路做成独立的课件，以便课堂上讲解，可以导出一个可独立执行的文件，脱离该平台使用。

图 12 是按照上述步骤制作的讲解滑动变阻器作用的一个电路图，在该电路中，可以用鼠标拖动滑动变阻器的滑块，观察电流表指针的变化，了解滑动变阻器的作用。

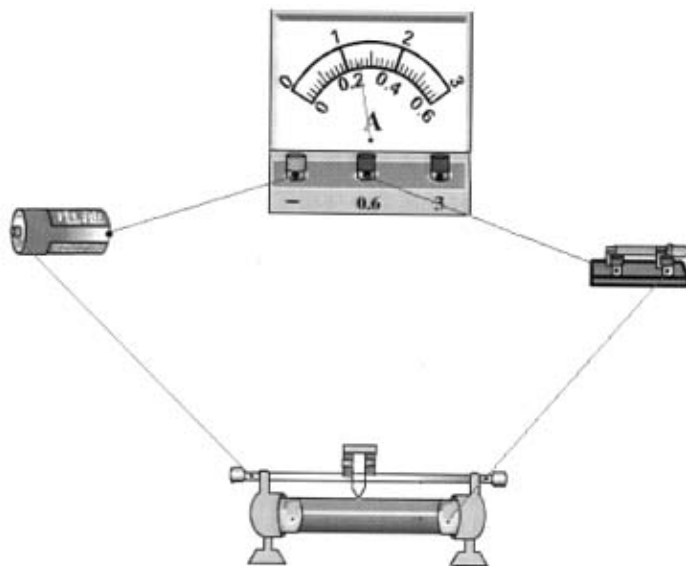


图 12 滑动变阻器的作用

第六章 总结与展望

6.1 全文总结

信息技术与学科课程教学整合已经走进课堂教学，并探索出一些新的教学模式，但制约信息技术与学科课程教学整合的关键因素是缺少能够灵活体现教师思想的资源和软件，工具软件在提高课堂教学的质量和效率方面具有重要作用。

本文通过对学科工具软件现状的研究，分析了学科工具构建平台的重要性，建立了学科工具构建平台的系统模型，并进行了数据和功能设计。同时，提出了基于领域知识的学科工具构件模型，并且详细分析、设计和实现了平台和构件模型。最后，通过物理电学构件的设计和实现，对平台和构件模型进行了验证。在本系统实现后，理想信息技术研究院虚拟仿真实验室利用该系统开发了物理光学、力学和数学统计图构件，进一步证实了平台和构件模型的实用性。对于整个系统的设计而言，本论文的研究成果在于：

1、基于领域知识的可重用学科工具构件模型的建立，便于学科工具构件的重用。同时，学科工具构建平台为学科工具构件模型的使用提供了集成开发环境，使在此平台下基于学科工具构件模型进行学科工具的二次开发变得简单、规范，能够大大提高学科工具构件的开发效率。

2、平台不仅考虑了构件开发的方便性，也充分考虑了学科工具构件最终使用者的需求。通过对各种文字、图片、表格、Flash、3D 动画等的支持，辅以丰富的教学辅助功能，不仅满足了教师对学科工具构件的需要，也为教师提供了一个很好的备课和讲授平台，满足教学过程的需要，具有很大的使用价值和现实意义。

6.2 工作展望

本文对学科工具构件的设计和开发进行了研究，提出了学科工具构建平台的思想，并给与了实现，取得了一些研究成果，但对于整个系统来说，还有待于进一步改进。

平台目前对学科工具构件模型的支持仅限于二维图形构件，没能支持三维学科构件；构建模型关注的仅仅是构件的行为、交互方式和工作流程，没能充分关注构件的几何模型。对三维模型的支持将是今后研究的重要方向。

理论上构建的模型必须到实践中进行检验，必须使之尽量满足用户的需求。同时，用户的需求也不是一成不变的，随着时间的推移和周围环境的变化，用户的需求也会发生相应的改变。教师不相信世界上有最好的教学工具软件，但他们相信世界上有最适合的教学工具软件，希望本系统最终能够成为信息技术与课程整合的有利助手。

参考文献

- [1] Yongjun Jing, Shaochun Zhong & Xiaochun Cheng. Teaching Models and Supporting Platforms in Modern Distance Education, CACS2003, September, 2003, Luton.
- [2] Xiaochun Cheng Teaching Model using Computer Network, ACE UK Journal, Vol. 11, 2004, ISSN 1476-5837, pp42-45.
- [3] Y. Li and X. Cheng Formal Analysis for Dynamic Planning of Intelligent Agents September 7-8, 2004, Londonderry, U.K. Proceedings of the IEEE SMC UK-RI Chapter Conference 2004 on Intelligent Cybernetic Systems.
- [4] Shaochun Zhong, Yongjun Jing & Yongjiang Zhong, The Research on Resources and Software on Integration of Information Technology and Courses, 2005 International Conference on Active Media Technology, pp. 169-172, May, 2005, Japan.
- [5] 李英军, 马晓星, 蔡敏, 刘建中等译. 设计模式: 可复用面向对象软件的基础. 北京: 机械工业出版社, 2000.
- [6] Peng Sun, Xiaochun Cheng & Shaochun Zhong, Knowledge Management for E-learning System, Proceedings of the 10th Annual Conference of Chinese Automation and Computing Society in UK, pp. 193-198, Liverpool, UK, September 2004.
- [7] Waller John C, Foster Natalie. Training via the web: a virtual instrument. Computers and Education, 2000, 35(2). pp161-167.
- [8] David Harris, An Exponentiation Unit for an OpenGL Lighting Engine Institute of Electrical and Electronics Engineers, 2004.
- [9] Plante. Thomas G., Aldridge. Arianne, Bogden. Ryan, Hanelin. Cara. Might virtual reality promote the mood benefits of exercise? Computers in Human Behavior. 2003, 4(19).
- [10] Yongjiang Zhong, Ju Liu, Shaochun Zhong, Yamei Zhang, and Xiaochun Cheng. Programming of Information Instructional Design Platform for Physics. Edutainment 2006: International Conference on E-learning and Games, April 16-19, 2006, Zhejiang University, Hangzhou, China.
- [11] 冯博琴, 冯岚, 薛涛, 崔舒宁译. 面向对象分析与设计. 北京: 机械工业出版社, 2003.
- [12] Shaochun Zhong, Yongjun Jing, Yongjiang Zhong, and Jinan Li. Research on the Method of the Integration of Information Technology and Courses. Edutainment 2006: International Conference on E-learning and Games, April 16-19, 2006, Zhejiang University, Hangzhou, China.
- [13] Burdea G, Philippe Coiffet. Virtual Reality Technology. New York: A Wiley-interscience Publication, 1994.
- [14] Li Shengqiang, Yang Jiajun, Zeng Kai. An interactive simulation system of mechanism based on VRML. Journal of HuaZhong University of Science and Technology, 2002, 30(9): 73~75.
- [15] Osman Balci. Guidelines for successful simulation studies. Proceeding of the 1990 Winter Simulation Conference, ed. Osman Balci, Randall P. Sadowski, Richard E. Nance,

- [16] 申蔚, 夏立文. 虚拟现实技术 北京: 北京希望电子出版社, 2002.
- [17] 侯捷. 深入浅出 MFC. 武汉: 华中科技大学出版社, 2001.
- [18] 余胜泉, 何克杭. 网络教学平台的体系结构与功能[J]. 中国电化教育, 2001(8):60~64.
- [19] 宋浩, 于靖熙. 关于网络教学平台智能化开发的技术可行性初析[J]. 现代教育技术, 2002(3):71~75.
- [20] Koop T, Broll W. VRML: Today and Tomorrow Computers and Graphics. An International Journal. 1996, 3(20). D. Kromker(ed.), J. Encarnacao(ed. In chief). Pergamon Press, Oxford, New York, Tokyo, Seoul(1996), 427~434.
- [21] Jeffrey Richter 著, 王书洪 刘光明译, Windows 高级编程指南(第三版), 清华大学出版社, 1999.
- [22] 吕凌燕, 赵小敏, 王昊翔, 等. 基于网络的教学支撑平台的设计[J]. 计算机工程与应用, 2003(12):174~177.
- [23] Jeffrey Richter 著, 王书洪 刘光明译, Windows 高级编程指南(第三版), 清华大学出版社, 2003.
- [24] 彭群生, 鲍虎军, 金小刚, 计算机真实感图形的算法基础, 科学出版社, 1999年.
- [25] 王琦电脑动画工作室. 3DS MAX5 白金手册. 北京: 北京科海电子出版社. 2003.
- [26] 王汝传, 姚旭敏, 王海艳, 刘丽. 基于 Java 和 VRML 虚拟场景通信方式的研究.
- [27] 王力, 冯武, 龚杰民. VRML 在仿真可视化系统中的应用. 计算机仿真, 2001.
- [28] 陈礼民, 秦爱红. 三维图形简化新算法. 中国图像图形学报, 1997, 2(23): 157-160.
- [29] 陶志良, 潘志庚, 石教英. 复杂场景中动态简化层次的构造. 中国图像图形学报, 1998, 3(12): 1032-1036.
- [30] 陈守满, 姜建国, 等. 虚拟现实技术在教学中的应用[J]1 计算机应用, 2002, (4):111-112.
- [31] 巩微. 关于用虚拟现实技术进行远程教学的研究[J]1 辽宁大学学报, 2002, (1):36-41.
- [32] Araki Y, Imamura H, Obo T, Kobori Y. Computer assisted instruction learning system using CAI [A]. Frontiers in Education Conference, 1993, Twenty-Third Annual Conference, Proceedings of Engineering Education: Renewing America's Technology [C], 1993, 804.
- [33] Poole N R. The application of simulators in teaching digital electronics [J]. Engineering Science and Education Journal, August 1994, 3 4: 177-18.
- [34] Taylor K D, Honchell J W, DeWitt W E. Distance learning in courses with a laboratory [A]. Frontiers in Education Conference, 1996, FIE '96, 26th Annual Conference, Proceedings of Volume: 1 [C], 1996. 44-46.
- [35] David Jones. Computing by Distance Education: problems and solutions [A]. roceedings of the conference on Integrating Technology into Computer Science Education [C]. epartment of Mathematics & Computing Central, Queensland University.
- [36] 史忠值. 高级人工智能. 科学出版社. 1998

致 谢

本文的研究工作是在我的导师钟绍春教授的精心指导和悉心关怀下完成的，我的论文研究工作及学业过程倾注了导师的辛勤汗水和心血。他扎实的专业知识、求实的工作作风和严谨的治学态度都令我深表敬佩。攻读硕士期间，我不仅学到了专业领域的知识和技能，在工作态度、方法上也深受启迪，导师对我的教导和影响必将使我终身受益。在此，我向导师钟绍春教授致以最衷心的感谢和最诚挚的敬意！

同时，在系统实现过程中，实验室的冯国松、王发斌、王伟、王波、康健、戴勇、潘秀龙等同学提出了许多宝贵建议，并参与了平台的测试工作，在此，一并表示感谢。

读研期间公开发表的论文

文章名称	发表刊物（出版社）	刊发时间	刊物级别	第几作者
The Deign and Realization of the Electrics Platform for Middle-school Physics	Journal of Computational Information Systems	2007. 2	EI	3