

一种 Post-WIMP 界面:PGIS 的实现

石磊^{1),2)} 邓昌智¹⁾ 戴国忠¹⁾

¹⁾(中国科学院软件研究所,北京 100190) ²⁾(中国科学院研究生院,北京 100049)

摘要 笔式用户界面作为 Post-WIMP 界面的一种重要形式,成为当前研究的热点之一。基于笔交互的应用程序通常有如下需求:自然统一的风格、个性化界面定制的支持,以及各种终端设备上的移植。为了支持笔交互应用程序快速稳定的开发,提出了一种新的笔交互范式 PGIS,以及基于 PGIS 的应用程序开发平台。PGIS 交互范式包含了纸、小工具等交互组件,定义了各种组件的显示及交互形式。PGIS 平台采用模型驱动构架的设计方法,主要包括 3 个部分:界面的设计工具,PGIS 交互引擎,PGIS 底层核心库。目前,已经在 PGIS 平台上开发了几款优秀的笔式交互软件,完全满足上述笔式交互应用程序的需求。PGIS 的可行性也因此得到了很好的验证。

关键词 Post-WIMP PGIS 笔式用户界面

中图法分类号: TP301 文献标志码: A 文章编号: 1006-8961(2010)07-985-08

Implementation of PGIS: A Type of Post-WIMP User Interface

SHI Lei^{1),2)}, DENG Changzhi¹⁾, DAI Guozhong¹⁾

¹⁾(Institute of Software, Chinese Academy of Science, Beijing 100190)

²⁾(Graduate University of Chinese Academy of Science, Beijing 100049)

Abstract The pen-based user interface now plays an important role as a main type of Post-WIMP. It renders efficient and natural interaction between human and computer, and it is well suited to people's daily working habits. Applications incorporating a pen-based user interface should meet the following requirements: a uniform and natural display style; a customizable user interface; and high portability. To meet these requirements, this article describes a new paradigm: PGIS and a software-development platform to implement the PGIS. The PGIS paradigm, which describes some interactive objects, such as Paper, Gadget and so, is based on the pen-paper metaphor which is analogous to the real working environment. Adopting the model driven architecture design method, the PGIS software platform has a portable core, uniform management and description of the interactive objects, and a customizable user-interface design tool. Currently, under the PGIS software platform, several applications using pen-based interaction have been developed quickly and successfully. The feasibility of PGIS and the software platform therefore is well verified.

Keywords Post-WIMP, PGIS, pen-based user interface

0 引言

20 世纪 80 年代以前,人机交互的方式是多人共用一台大型计算机。八十年代后期,计算模式从

大型机向 PC 机转移,进入了桌面计算时代。进入 21 世纪,随着数字终端设备,如手机、PDA 等在人们生活中的大量普及,普适计算的时代到来,而 WIMP 界面的缺点也逐渐地体现出来。从 20 世纪 90 年代初开始,国外学者先后提出了 Non-WIMP/Post-

基金项目:国家重点基础研究发展计划(973)项目(2009CB320804);国家自然科学基金项目(U0735004);国家高技术研究发展计划(863)项目(2007AA01Z158)

收稿日期:2010-03-17; **改回日期:**2010-05-18

第一作者简介:石磊(1984—),女,中国科学院软件研究所计算机应用技术专业硕士生。主要研究方向为人机交互。E-mail: shilei0352@gmail.com

WIMP 界面。笔交互作为 Post-WIMP 交互的一种重要形式,模拟了人们日常的纸笔工作环境,具有便携性、使用的手眼协调性(视觉与手动的整合)和直觉性(草图的使用、思维连续性)^[1-2]。随着笔式界面软件的广泛应用,人们对软件进行快速开发,并能在多种平台上移植,同时支持个性化的定制。在这种情况下,提出了更为自然高效的 PGIS (paper, gadget, icon/button, sketch) 笔交互范式,并研究和设计了采用模型驱动构架 (MDA) 来支持应用程序需求的 PGIS 软件平台。MDA 是对象管理组织针对软件产业所面临的开发周期长,难以维护等压力提出出来的一种新的解决途径。MDA 的关键之处是,模型在软件开发中扮演了非常重要的角色,整个软件开发过程是由对软件系统的建模行为驱动的。采用 MDA 开发的软件开发周期短、可移植、并且适应于需求的频繁改变。

1 相关工作

SATIN^[3-5] 是一款由伯克利大学开发的基于 JAVA 的笔式软件开发平台,基于 SATIN 已经开发出一些不错的笔交互系统,如 DENIM, SketchySPICE 等。SATIN 是从软件模块重用的角度出发,将笔式交互中的常用功能封装起来,如字符识别、笔迹理解等,一定程度上简化了应用程序的开发量。但它没有提供一个指导笔式软件开发的构架,也缺少特定交互范式的理论支持。

PIBG ToolKit^[5-6] 也是一个笔式软件平台,在其基础上开发出笔式电子白板系统,电子教学系统等笔式软件。从交互角度讲,它基于 PIBG (paper, icon/button, gesture) 交互范式,缺少了小工具的概念。从系统开发的角度看,它在硬件和应用程序之间缺少一个抽象层,在不同平台上不易移植。另外,当需求改变的时候,开发人员不得不花费大量的时间做相应地改变。

目前市面上存在着不少软件支持 MDA 的软件。IBM Rational Software Architect 是一套设计与开发工具,它构建在开放的、可扩展的 Eclipse 3.0 平台之上。借助 UML 2.0 技术,实现了 MDA 的软件开发模式,可以帮助开发团队快速创建更加强壮的软件架构。Compuware OptimalJ 也是 MDA 标准的引领者,致力于快速开发高质量、可维护的 JAVA 程序。另外,AndroMDA 是一款开源的符合 MDA 标准

的软件,借助于 CASE-Tool 从模型自动生成代码。OpenMDX 也是非常领先的开源 MDA 平台,它可支持 J2SE, J2EE, .NET。这些平台基本上都针对业务逻辑的功能模块,缺少对于一些业务逻辑少且主要工作集中在界面交互的软件的支持。

提出的笔式软件平台基于新的笔交互范式 PGIS,主要用来支持业务逻辑少且大部分工作集中在界面交互的笔式软件,即界面软件。整个平台采用 MDA 方法,实现了文档到代码的一体化,并有以下特点:开发周期短、易于应对需求改变、可在多种平台下移植。

2 PGIS 软件平台的研究

2.1 PGIS 交互范式介绍

PGIS 交互范式从信息呈现和交互方式两个主要方面做了根本性的改变。从信息呈现的角度来看,PGIS 范式利用纸和框的信息呈现方式模拟人们在日常纸笔环境下的信息呈现方式,利用用户原有的(自然的)知识,以此来提高用户在交互过程中的数据质量,从而减轻用户的认知负担。

Paper 模拟现实世界中纸的概念,在 PGIS 界面范式中的作用相当于 WIMP 界面中的窗口。它主要作为界面的显示装置。图 1 为 Paper 的形象化表现形式。

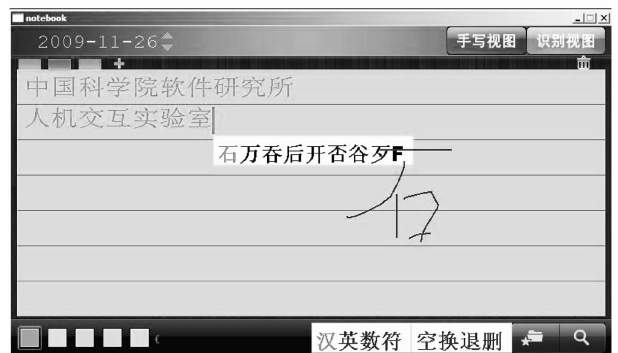


图 1 Paper 的视觉化表现示例

Fig. 1 The presentation of Paper

Gadget 包含功能多样的小工具,涉及文本列表、选择工具和面板等诸多功能。Gadget 的引入扩充了笔式用户界面的组件类型,使得笔式用户界面功能更加丰富。PGIS 界面中常用的 Gadget 工具有 Select 表:Text 列表,I 表,P-menu,Slider,Palette 以及树形控件等。

Icon 图标是具有指代意义的具有标识性质的图形,它不仅是一种图形,更是一种标识;具有高度浓缩并快捷传达信息、便于记忆的特性。通过图标看到的不仅仅是图标本身,而是它所代表的内在含义。Button 按钮是带有文字名称的圆角(或无圆角)矩形控件。短击按钮将触发与按钮名称相对应的动作。按钮的触发是即时的,如“确定”或“取消”。按钮的宽度与文字对应。

Sketch 是指常用的手势,如退格、回车、删除等。WIMP 范式执行对内容的操作时,用户所关注的焦点将会由内容转向交互组件,当操作完成后,焦点再转回内容。关注焦点的来回切换会给用户的认知加工过程带来难度,从而影响交互的效率^[1,7-8]。基于 Gesture 的交互方式通过笔在特定的信息上进行 Sketch 直接操纵,解决了这一问题。

2.2 PGIS 软件平台的分析研究

目前,移动设备的用户群越来越庞大。而这些移动设备通常运行着不同版本的操作系统。界面软件是移动设备上主要的软件形式,这就要求界面软件有良好的移植性。另外,类似于 WIMP 式的对于所有用户的一模一样的单调界面忽略了人们个性化的要求。不同环境下不同的用户往往在界面上有不同的需求。一款好的界面软件必须支持个性化的定制。而基于 PGIS 交互范式的界面软件又要求有统一自然的显示风格。综上所述,对好的界面软件有如下需求:可移植性、支持个性化的定制、自然统一的显示风格。开发的 PGIS 软件平台采用 MDA 很好地解决了前两点的需求。MDA 标准定义了平台无关模型(PIM)和平台相关模型(PSM)这两个术语。PIM 是具有高抽象层次、独立于任何平台及任何技术的模型。PSM 是为某种特定平台量身定做的;PSM 只对了解相应平台的开发者才有意义。MDA 的开发过程就是首先抽象出与实现技术无关,完整描述业务功能的核心模型 PIM,针对不同的实现技术制定变换规则,然后通过映射规则及辅助工具将 PIM 转换成与具体实现平台及实现技术相关的应用模型 PSM。最后,在一定程度上将 PSM 自动转换成相关平台及相关技术下的代码。MDA 流程如图 2 所示^[9-11]。

采用 MDA 开发的 PGIS 软件平台具有可移植性。PSM 作为底层平台库,对于常见的平台以动态链接库的形式各维护一个版本。PSM 底层平台库向上提供统一的与平台无关的接口,使得应用程序逻辑即 PIM 是独立于任何平台可移植的,并同 PSM

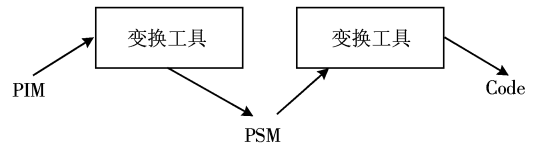


图 2 MDA 开发方法流程

Fig. 2 The flow of MDA development method

提供的与平台相关的动态链接库一并发布。

另外,采用 MDA 开发的 PGIS 软件也满足个性化定制的需求。通过遵循 MDA 方法,在 PGIS 软件平台下开发的应用程序是由用户需求模型来驱动的。应用程序的运行过程正如一场话剧演出,话剧中有若干个场景,每个场景有其特有的道具(交互对象),场景之间会发生切换,演员(用户)和道具之间需要大量交互动作。因此,在需求分析阶段,必须知道应用程序的话剧中有几个场景,每个场景中的交互对象是什么,用户与交互对象是怎样交互的,场景之间是如何切换的^[12-13]。PGIS 软件平台提供一个形象的设计工具去完成这样的需求分析,并将分析转换成相应平台下的代码。生成的代码调用底层的核心库而成为一个完整的应用程序。显而易见,应用程序的生成是需求分析结果驱动的,个性化的需求会自动生成个性化的应用程序。

另外,在 PGIS 软件平台开发的软件具有统一自然的显示风格。PGIS 交互范式借用纸笔隐喻,本身就是一种自然的交互方式。而 PGIS 软件平台提供 PGIS 交互引擎对应用程序中的交互对象进行统一的管理,包括其显示及其对消息的相应。这又形成了统一的显示风格。

3 PGIS 软件平台的设计

3.1 PGIS 软件平台的设计

界面软件要求响应时间短、多媒体播放流畅,这就决定了应用程序必须有较高的运行速度。同时,考虑到多个平台下的开发,选择标准 C 作为编程语言。但并没有摒弃面向对象的设计方法,而是用 C 来实现面向对象的思想。

PGIS 软件平台包含两个 PSM 模块(与平台相关的 PGIS 底层核心库、与实现技术相关的 PGIS 交互引擎)以及一个与平台无关的 PIM 模块(自动生成代码的界面设计工具)。PGIS 底层核心库相当于平台与应用程序之间的抽象层,对应用程序提供平

台无关的应用程序编程接口 (API)。对于基于 PGIS 的应用程序来说,首先调用界面设计工具与用户交流,分析需求;在设计工具中设计运行场景;场景之间的跳转关系;场景内的交互对象;以及在交互对象上发生的交互行为。设计结束后,设计工具调用 PGIS 交互引擎,将初期的设计转换成代码。图 3 为在 PGIS 软件平台下开发应用程序的时序图。

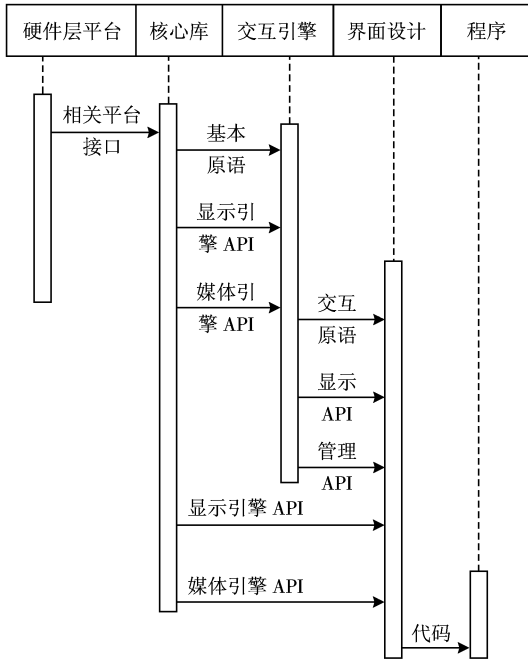


图 3 PGIS 平台下开发时序图

Fig. 3 The sequence diagram of the development

3.1.1 PGIS 底层核心模块的设计方案

PGIS 底层核心模块相当于应用程序和平台之间的抽象层,分别对相关平台的消息响应机制,显示层 API,媒体层 API 做了封装,以动态链接库的形式向上提供与平台无关的 API,其中包含 Ink 引擎、显示引擎、多媒体引擎。Ink 引擎结合笔式交互的特点,将底层的消息根据笔的停留时间和操作方式抽象成短击、长击、拉框、移动框及改变框的大小,以及划线 5 个基本原语。显示引擎类似于 Windows 下的图形设备接口 (GDI),包括基本的 2D 绘制,文字、图片的渲染,以及事件的捕捉等。媒体引擎主要提供音频、视频、Flash 的播放及控制函数。

这三个引擎对相关平台封装不同版本的动态链接库,随应用程序发布。当应用程序移植到另一个不同的平台,只需要更换 PSM 库,而不需要改变程序逻辑和代码。目前,PGIS 平台支持 Windows 98/2000/XP/Vista, WinCE, Redhat, 红旗, Ubuntu,

MiniLinux, Mac 以及各种手机版操作系统。

3.1.2 PGIS 软件平台交互引擎的设计

首先,对应用程序中的对象作了较为合理的分类:交互对象和应用对象。交互对象即为在应用程序运行前的设计阶段就确定的,数据不需要保存的对象,如按钮、图标、滑条、选择表等。而应用对象是程序运行过程中用户通过一定的交互行为动态产生的,在运行前无法确定,而且数据需要保存的对象,例如一个讲课系统中用户通过点击“新建”按钮新建的文本框、笔迹框等。PGIS 平台所提供的 PGIS 交互引擎将交互对象的描述、显示以及管理从应用程序中分离出来,不仅保证了统一的显示风格,而且简化了应用程序的设计,同时便于代码的自动生成。PGIS 交互引擎主要有 4 个模块:

1) 交互对象的描述

交互引擎抽取 PGIS 交互范式下常用的交互对象的特征,形成一棵描述的继承树(图 4),第 1 层为基对象,第 2 层为简单交互对象,第 3 层为组合交互对象。PGIS 下交互对象的继承关系如图 4 所示。

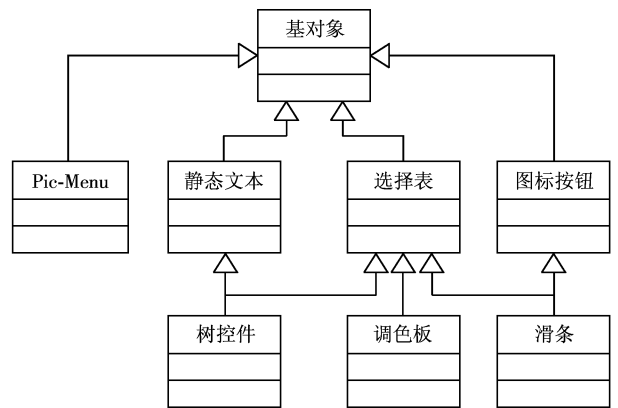


图 4 PGIS 交互引擎下的交互对象继承树

Fig. 4 The inherited tree of the interactive objects

以 Icon 图标为例,交互引擎对其属性作了详细描述,包括其大小、形状、位置、背景、状态、文字、交互状态、显示状态等。

另外,交互引擎对常见的手势进行了管理和描述,其手势库模块可自由添加新的手势和删除已有的手势。目前,交互引擎包含 20 多种手势,涵盖了笔交互应用程序中常见的操作,如删除、回车、退格、上下翻页等。图 5 为手势管理库的界面。

2) 交互对象的管理

从整体来看,交互对象由一个统一的抽象工厂来管理,抽象工厂对外提供统一的接口,包括交互对













识别	模板取样	保存模板	删除模板
向左箭头 	向上箭头 	向右箭头 	向下箭头 
V 字 	倒 V 字 	水平来回折线 	竖直来回折线 
竖直叉 	水平叉 	顺时针圆 	逆时针圆 

图 5 手势管理子程序的界面

Fig. 5 The user interface of the gesture management

象的创建、释放、显示等函数。用户不需要关心是对哪种特定交互对象的操作,而是调用统一的 API。而在引擎内部,每一类交互对象作为一个具现的工厂,对统一的抽象接口有自己的实现。各类工厂之间是松耦合的,新建或删除一个工厂不会对其他模块产生任何影响。具体结构如图 6 所示。

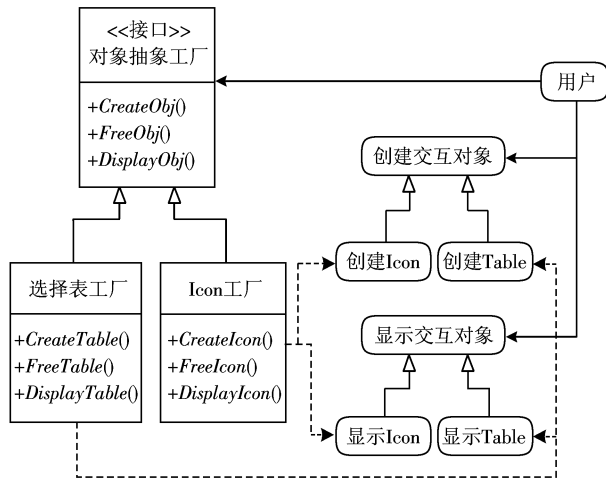


图 6 PGIS 交互对象工厂管理

Fig. 6 The management of the interactive object factory

从个体来看,交互引擎对交互对象提供链式的基本操作,如添加、插入、删除、设置、获取等。

3) 交互对象的显示

交互对象的显示模块是作为一个可插拔的组件来实现的。也就是说,当用户对显示有特殊的要求,可以隐藏现有的显示组件,向引擎中插入用户自己定制的显示组件。默认组件提供绘制和贴图两种显

示模式,根据用户不同的输入参数,提供多态的显示效果。以 Icon 图标为例,其贴图和绘制的显示效果如图 7 所示。



图 7 PGIS 交互对象显示示例

Fig. 7 The display of the interactive objects

4) 交互对象的消息响应模块,也是 PGIS 交互引擎的核心模块

首先,用一个有限状态机维护轻量级的内核,内核从 PGIS 底层核心库得到基本原语消息:即短击、长击、拉框、移动框及改变框的大小,以及划线 5 个基本原语,再将基本原语分发到各类交互对象工厂维护的消息响应机制中,对象工厂将基本原语转化成与交互对象相关的交互原语,返回给应用程序。消息转换过程如图 8 所示。

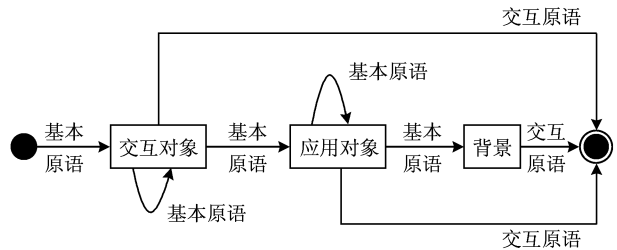


图 8 消息响应内核的状态机图示

Fig. 8 Generated responses of the finite-state machine

不同种类的交互对象有不同的消息响应机制。例如,图标 Icon 响应短击、长击;而滑条响应短击、划线;树控件响应短击、长击、划线。

3.1.3 PGIS 界面设计工具的实现

界面设计工具是 PGIS 软件平台下开发应用程序的入口点。而设计工具是在 PGIS 交互引擎的基础上开发的,对交互引擎中抽象的描述图形化,使设计过程更加自然直观。同时,该设计工具定义了代码映射的法则,将图形化的表示映射到交互引擎的各类 API。当设计结束时会自动生成可运行的代码,程序运行过程中调用交互引擎的 API,而交互引擎又会调用 PSM 底层封装的 Ink 引擎、显示引擎、多媒体引擎。当然生成的代码只是涉及到交互对象部分,对于应用对象部分在代码中以哑库(即空调用)的形式显示。而界面软件的特点在于多数工作集中在界面交

互上,所以需要补充的代码量仅占一小部分。

设计过程主要确定有哪些场景;场景间如何跳转;每个场景的界面(背景以及交互对象);每个交互对象上响应的交互行为,如长击、短击等。当设计过程结束,用户得到自动生成的可运行的代码。因为程序框架和交互相关的代码都已完成,应用程序 70%的工作已经完成。因此,PGIS 软件平台大幅度地缩短了开发周期。

3.2 PGIS 可行性验证

PGIS 可行性从两方面加以验证:一是开发已有软件查看其成本是否降低,性能是否提高。二是其开发流程是否简单易学,即有较低的开发门槛。本节从这两个方面进行了较详尽的阐述。

3.2.1 PGIS 可行性分析

在 PGIS 交互范式的指导下,在 PGIS 软件平台上重新开发了实验室在 PIBG 交互范式及平台^[5]下已开发的幼儿教学系统。下文从可运行平台、是否支持用户定制、稳定性、及可维护性上对在 PGIS 下开发的新幼教系统与在 PIBG 下开发的老幼教系统进行了对比。

从可运行的平台来看,新幼教系统可在以下平台上运行:Windows 98/2000/XP/Vista, WinCE, Redhat, 红旗, Ubuntu, MiniLinux, Mac, 以及各种手机版操作系统。而老幼教系统仅在 Windows XP 上运行。从是否支持用户定制来看,新幼教系统充分支持用户定制,并可用界面设计工具与用户进一步地交流。而老幼教系统很难支持用户定制。从稳定性来看,新幼教系统稳定性较高,在各种严格的测试过程中表现良好。而老幼教系统稳定性较低,在操作过程中偶尔出现系统崩溃的情况。从可维护性来讲,新幼教系统中文档即程序,易于维护。而老幼教系统依赖于程序员手工维护,成本较高。

由对比结果可以看出,在 PGIS 平台下开发的应用程序在软件质量上有较大地提高。另外,在较短时间内还成功开发了几款优秀的基于笔交互的软件,如便签本系统,透明白板系统等。

3.2.2 应用实例

现在以幼儿教学系统中备课系统为实例简单介绍在 PGIS 软件平台上的应用程序开发流程。

第 1 步 分析用户需求,确定场景:页场景、缩略图场景、框场景、内容场景。利用 PGIS 软件平台提供的界面工具绘制场景图,确定场景跳转关系。场景图如图 9 所示。

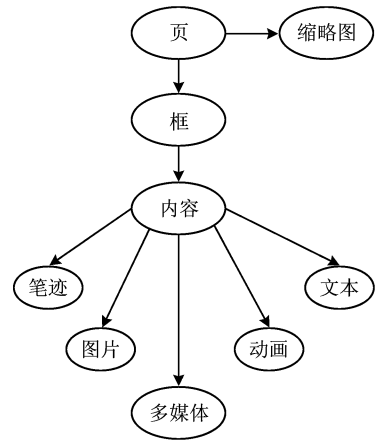


图 9 备课系统的场景图

Fig. 9 The scene diagram of the teaching system

第 2 步 状态图设计。即对于某一个场景的详细设计,涉及该场景中的交互对象和状态任务。以缩略图场景为例,其交互对象含一组操作图标:新建、复制、粘贴、撤销、删除;以及一组手势:上下翻页、换页。其状态任务及交互对象和手势产生的交互行为。

第 3 步 用户界面的设计。PGIS 设计工具有可选的各类交互对象。用户可以通过拖拽的方式将需要的交互对象放入界面,并对其属性做个性化的设置。对每一个场景重复第 2 步和第 3 步,当设计完毕,点击 PGIS 设计工具中的生成按钮,就得到了可运行的代码。对于无法生成的程序的应用对象部分,生成代码以哑库的形式显示。

第 4 步 对生成的代码的哑库做补充,得到完善的应用程序。

在 PGIS 软件平台生成的应用程序截图如图 10—图 13 所示。

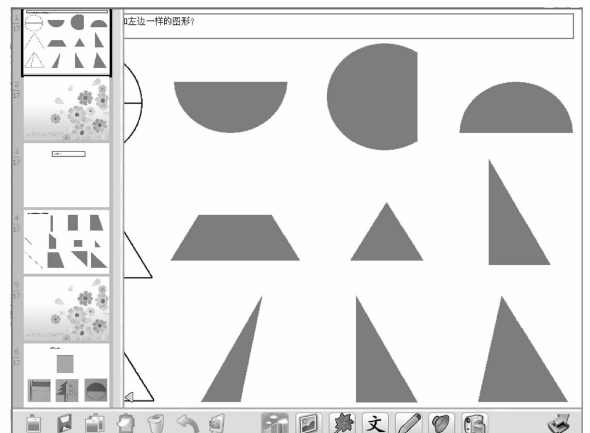


图 10 备课系统部分截图

Fig. 10 The screen capture of the teaching system



图 11 讲课系统部分截图

Fig. 11 The screen capture of the teaching system

4 结 论

在对 PGIS 交互范式研究和分析的基础上,提出了基于 MDA 的 PGIS 笔式软件平台。该平台包含 3 个部分:界面设计工具、交互引擎,以及底层核心库。这 3 个部分贯穿应用程序开发的整个流程,实现了设计文档和代码实现一体化,促使开发人员的重点放在了设计上而不是代码的实现上。简言之,PGIS 平台不仅大幅度缩短了基于笔交互软件的开发周期,满足可移植性,而且支持界面的个性化定制。

下一步的工作重点是改进 PGIS 界面设计工作,生成更为高效的代码。另外,在 PGIS 交互引擎的基础上维护一个基于笔的较全面的交互对象库,使笔交互的应用程序界面有更为丰富的选择。

参考文献 (References)

- [1] Dai Guozhong, Tian Feng. Pen-based User Interface [M]. Beijing: University of Science and Technology of China Press, 2009: 20-43. [戴国忠, 田丰. 笔式用户界面[M]. 北京: 中国科学技术大学出版社, 2009: 20-43.]
- [2] Tian Feng. Research on Post-WIMP Software Interface [D]. Beijing: Institute of Software Chinese Academy of Sciences, 2003. [田丰. Post-WIMP 软件界面研究[D]. 北京: 中国科学院软件研究所, 2003.]
- [3] Jason I H, James A L. SATIN: a toolkit for informal ink-based applications [C]//Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology, New York, NY, USA: ACM Press, 2000: 63-72.
- [4] James L, Mark W N, Jason I H, et al. DENIM: finding a tighter fit between tools and practice for Web site design [C]// Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York NY, USA: ACM Press, 2000: 510-517.
- [5] Tian Feng, Qin Yanyan, Wang Xiaochun, et al. Analysis and design on PIBG toolkit: A pen-based user interface toolkit [J]. Chinese Journal of Computers, 2005, 28 (6): 1036-1042. [田丰, 秦严严, 王晓春, 等. PIBG Toolkit: 一个笔式界面工具箱的分析与设计 [J]. 计算机学报, 2005, 28 (6): 1036-1042.]
- [6] Tian Feng, Mou Shu, Dai Guozhong, et al. Analysis and design on PIBG toolkit: A pen-based user interface toolkit [J]. Chinese Journal of Computers, 2004, 27 (7): 977-984. [田丰, 牟书, 戴国忠, 等. Post-WIMP 环境下笔式交互范式的研究 [J]. 计算机学报, 2004, 27 (7): 977-984.]

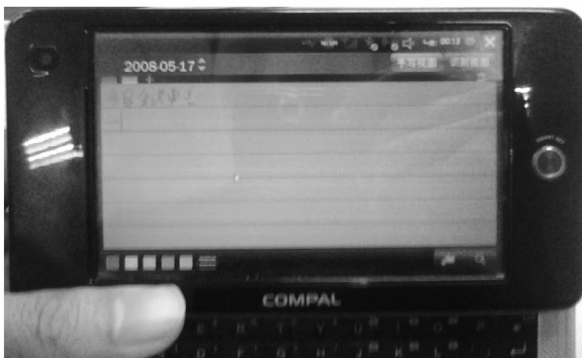


图 12 便签本系统部分截图

Fig. 12 The screen capture of the scratch pad system

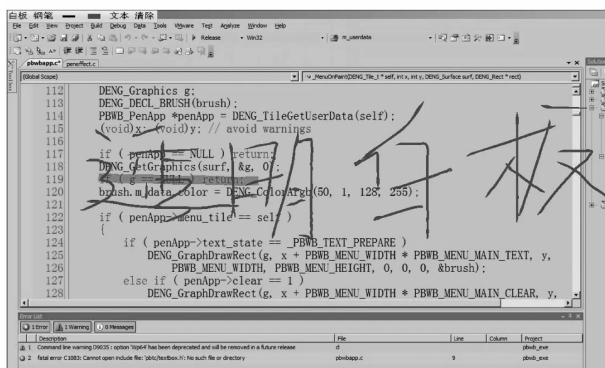


图 13 透明白板系统部分截图

Fig. 13 The screen capture of the transparent whiteboard system

- [7] James A L, Brad A M. Interactive sketching for the early stages of user interface design [C]//Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York NY, USA: ACM Press, 1995: 43-50.
- [8] Beryl P, Mark D A. Interacting with sketched interface designs: an evaluation study [C]//Proceedings of CHI ' 04 Extended Abstracts on Human Factors in Computing Systems. New York NY, USA: ACM Press, 2004: 1337-1340.
- [9] Pane J F, Ratanamahataa C A, Myers B A, et al. Studying the language and structure in non-programmers' solutions to programming problems [J]. International Journal of Human-Computer Studies, 2001, 54(2): 237-264.
- [10] Luckham D C, Kenney J J, Augustin L M, et al. Specification and analysis of system architecture using rapide [J]. IEEE Transactions on Software Engineering, 1995, 21(4): 336-355.
- [11] Luckham D, Vera J. An event-based architecture definition language [J]. IEEE Transactions on Software Engineering, 1995, 21(9): 717-734.
- [12] Carroll J M. Five reasons for scenario-based design [J]. Interacting with Computers(IWC), 2000, 13(1): 43-60.
- [13] Kentaro G, John M C, Imamiya A. Surveying scenario-based approaches in system design [J]. IPSJ SIG Notes, 2000, 12(HI-87): 43-48.