

笔式界面软件快速开发工具

华 超^{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 笔式界面软件快速开发工具 Visual PBAP (pen-based application) Creator 可以根据用户的个性化需求快速设计和开发相应软件, 缩短开发周期。一方面, Visual PBAP Creator 通过场景和 UI 编辑方便开发人员和用户确定需求; 另一方面, 自动生成代码可以缩短开发周期。Visual PBAP Creator 以笔式操作平台为软件平台, 采用基于场景设计的开发方法, 设计结果形成 XML 格式的文档, 通过解析这些 XML 文档最终生成 C 代码。实践表明, Visual PBAP Creator 可以提高笔式界面软件的开发效率。

关键词 笔式用户界面 基于场景的设计 形式化文档 UI 编辑 代码生成

中图法分类号: TP391.72 **文献标志码:** A **文章编号:** 1006-8961(2010)07-1007-07

Rapid Development Tool for Pen-based Application

HUA Chao^{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 Visual PBAP (Pen-based Application) Creator is a rapid development tool, which can be used to design and develop Pen-based Application according to end user's specific requirement and can shorten the development cycle. On one hand, Visual PBAP Creator introduces scenario-based method and UI editor to facilitate the definition of requirement. On the other hand, it can automatically generate code to accelerate the development speed. Visual PBAP Creator is based on PBOP (Pen-based Operating Platform) and scenario-based method. The design results are saved as XML specification, and these XML files will be parsed to generator C code, which can be compiled and run on PBOP. The results of practices have shown that Visual PBAP Creator can improve the efficiency of development of Pen-based Application.

Keywords pen-based user interface, scenario-based design, formal documentation, UI editor, code generating

0 引 言

基于纸笔隐喻的笔式用户界面是普适计算的一个主要形态,其自然、高效的交互方式使用户在不增加认知和交互负担的同时,充分利用计算资源^[1]。

笔式交互固有的非精确性和强的信息表达能力,使得它有利于表达思想和进行自然的交流。最终用户开发方法(end user development)适合于笔式

用户界面系统。这一类系统的特点是:根据不同用户的业务逻辑开发符合用户要求的界面和交互。通常的开发过程为:1)挖掘用户需求;2)生成原型系统与用户进行交流,不断更新设计。由于用户的需求是个性化、不断变化的,所以原型系统的设计要与用户需求保持一致,开发周期要尽可能短。由此需要相应的笔式界面设计工具帮助开发人员进行快速开发。

本文提出并实现了一个笔式界面快速开发工具

基金项目:国家重点基础研究发展规划(973)项目(2009CB320804);国家自然科学基金项目(U0735004,60603073);国家高技术研究发展计划(863)项目(2007AA01Z158)

收稿日期:2010-03-18;**改回日期:**2010-05-18

第一作者简介:华 超(1984—),男。中国科学院软件研究所计算机应用技术专业硕士研究生。主要研究方向为人机交互与笔式界面软件。E-mail: huachao.iscas@gmail.com

Visual PBAP Creator, 该工具有如下特点:1) 通过基于场景的图形化方式设计最终用户的业务逻辑,方便与用户沟通;2) 根据最终用户的个性化需求定制用户界面与交互;3) 将高层的业务逻辑自动转换为形式化的描述文档;4) 根据形式化的描述文档生成代码,缩短开发周期。

1 相关工作

作为笔式用户界面研究的一个重要标志,1991年,施乐 Palo Alto 研究中心(Xerox PARC)研制了一个白板大小的、可以用专用笔进行直接交互的笔交互设备(LiveBoard),可以用于会议室和教室,并开发了基于该交互设备的软件,提出了许多现代笔式用户界面研究的基本思想和概念。

SATIN^[2]是 UC Berkeley 开发的一个基于 Java 的笔式用户界面工具箱。该工具箱提供了为笔式交互设计的 UI 组件,INK 笔迹的操作,用笔自由勾画的界面,设计者可以在设计中的任意时刻进行测试并作进一步的修改。基于 SATIN 工具箱已经开发出一些应用,如网站设计工具 DENIM^[3],它使网站设计者能够脱离技术细节进行高层的思考。然而 SATIN 对于用笔进行复杂交互的支持不够。

PIBG Toolkit^[4]也是一个用于开发笔式用户界面的工具箱。它基于 PIBG^[5]交互范式,由纸、框等交互组件支撑,定义了不同层次交互的结构。它可以解析原始消息到相应的笔式交互原语,并进行交互任务处理,同时有相应的手势集支持手势操作。对笔式用户界面的常用功能提供支持。在它基础上已经开发了基于数字笔的科学跳水训练管理系统、笔式数学公式计算器等多个笔式交互系统。本文的笔式界面范式 PGIS 是 PIBG 范式的扩展与提升。用 Sketch 取代了 Gesture 元素,更加突出笔的自由输入,并且在底层的 INK 引擎中加入了对 Sketch 的支持,比在应用逻辑层的 Gesture 速度更快。Gadget 也是新加入的一个元素,它是一组适合于笔式应用界面的 UI 组件,如调色板、饼形菜单。

SketchFlow^[6]是微软 2009 年发布的一款用 Sketch 方式进行原型系统设计的快速开发工具。主要有 3 个特点:1) 高效的原型建模,用户可以用草图绘制的方法构建起自己的应用程序框架和外观;2) 交互式的迭代设计,在原型建模的过程中用户可

以和 SketchFlow 进行直接的交互,方便用户对原型系统做出反馈;3) 自动将用户的建模过程保存为 Microsoft Word 的文档。

2 笔式界面软件快速开发工具

Visual PBAP Creator 由 3 个模块组成:场景设计模块、UI 编辑模块、C 代码生成模块。第 1 步,在需求分析阶段,开发人员将用户的需求映射到场景中,并且通过设计 UI 来向用户展示业务逻辑与应用程序的对应关系,并且定制用户个性化的需求,比如 UI 的布局与外观。第 2 步,Visual PBAP Creator 将图形化的场景树和 UI 保存为 XML 文档。第 3 步,Visual PBAP Creator 根据 XML 文档自动生成 C 代码。其总体架构如图 1 所示。

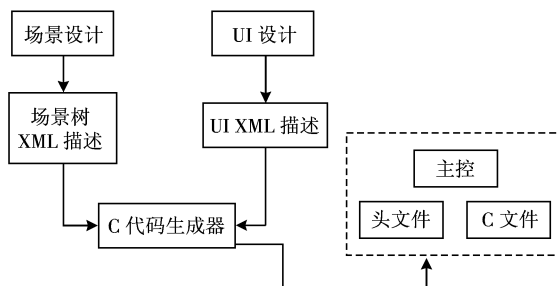


图 1 总体架构

Fig. 1 Architecture

2.1 场景设计模块

用户个性化的需求通过场景的方式来表达^[7]。每个场景包含一个状态图,其定义了场景的动态行为。每一个动态行为包含 3 个要素:1) 操作者;2) 被操作对象;3) 行为活动。笔式界面设计中操作对象就是用户,被操作对象就是每个场景的 UI,行为活动就是用户的交互方式。基于场景开发方法有以下优点:1) 促进设计人员对先前设计的思考;2) 场景善于适应变化;3) 对用户有多种表现形式;4) 与具体技术无关;5) 面向用户的具体工作行为^[8]。多个场景组成一个场景树,场景树其实就是 Storyboard。它处于用户需求和软件实现之间^[9],既可以描述用户的需求又可以作为一种整体设计。对于一个具体的需求,Visual PBAP Creator 生成场景任务列表和交互原语列表,这两个列表与用户的业务逻辑相关。开发人根据这两个列表设计与特定于用户需求的函数库。Visual PBAP Creator 自动地把图形化的场景树转换成 XML 格式

的形式化描述。场景树的图形表示如图 2，状态图的表示如图 3。

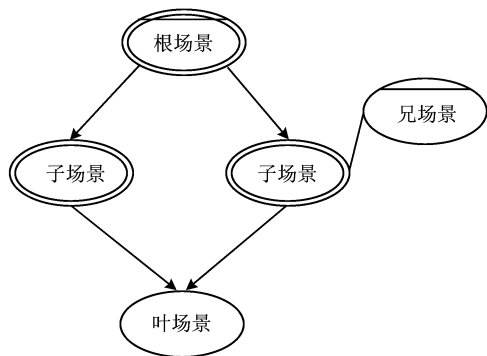


图 2 场景树
Fig. 2 Scenario tree

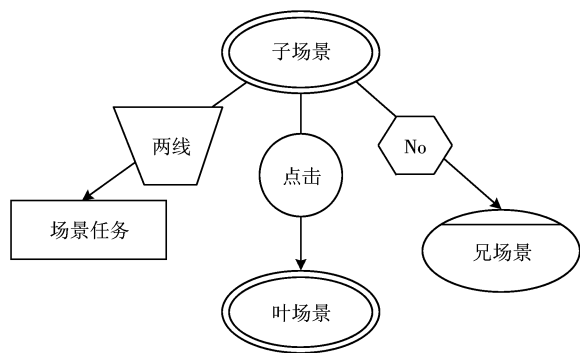


图 3 状态图
Fig. 3 State diagram

2.2 UI 编辑模块

UI 编辑器用来设计每个场景中包含的 UI 元素。Visual PBAP Creator 采用 PGIS 范式。PGIS 范式是新的笔式应用界面范式,由 4 个元素组成 Paper, Gadget, Icon, Sketch。Paper 用来呈现, Gadget 是一类适应于笔式应用界面的小工具,比如调色板,饼形菜单等。Icon 就是按钮,也是唯一保留的 WIMP 界面中的元素。Sketch 是 ink 笔迹和手势。每个场景 UI 由这 4 类元素组成,在设计场景时可以创建,修改这 4 个 UI 元素的属性来满足用户对界面外观的需求。在设计 UI 时还可以和场景任务关联,表示此 UI 执行一个场景任务。对于遵循 PGIS 范式的界面 UI 元素,UI 编辑器可以快速地构建和展示。在 WIMP 界面范式下,有许多类似的 UI 编辑器,如 Microsoft 的 Visual Studio。Visual PBAP Creator 会把这些图形界面转换成 XML 格式的 UI 形式化描述。

2.3 代码生成模块

代码生成器是一个 MDA Tool^[10],它将平台无关模型的业务逻辑转换为平台相关的代码。具体过程为:代码生成器根据业务逻辑的 XML 描述文件和 UI 的 XML 描述文件,对每个场景生成一个头文件和一个 C 文件,处理场景内部的操作,如场景任务函数和生成 PBOP 支持的 PGIS UI。代码生成器会自动地链接已有的库函数(PBOP 库,PGIS 库)和已有的特定于用户需求的库(如基于卡片操作的 CDM 库),如果已有的库中不存在满足用户需求的函数,代码生成器生成一个空的函数,待以后库中添加。对于任意一个场景,都有背景显示,设置交互,设置 UI,消息循环处理等基本流程,这样就形成了一个场景模板,代码生成器生成这个模板并填入特定场景的内容。场景树生成主控程序,主控管理各个场景之间的关系,在程序运行时,根据条件把控制权交给当前场景。所有生成的 C 代码可以在 PBOP(pen-based operating platform)平台上编译运行。

3 场景形式化描述与 UI 形式化描述

形式化的描述对于需求文档和代码生成都至关重要。一方面需求文档是系统设计和开发人员沟通时的重要依据;另一方面形式化的描述定义规范并且场景形式化描述与具体平台无关,UI 形式化描述与平台相关,这样从平台无关到平台相关的映射清晰,方便代码的生成^[11]。场景形式化描述与 UI 形式化描述在实现时均采用 XML 格式。

3.1 场景形式化描述

场景的形式化描述主要由 3 部分组成:1) 场景 ID;2) 场景步;3) 场景 UI。其中场景 ID 唯一标示场景。场景步由两部分组成:1) 交互原语,用来描述动作的交互类型,如点击,Stroke 等;2) 场景动作,用来描述动作的目的,如跳转到其他场景,执行一个场景任务。场景树包含多个场景,各个场景之间的逻辑跳转关系在场景动作中给出。场景有相应的 BNF 形式化描述。

- <场景树>::={<场景>}
- <场景>::=<场景名><UI>{<场景步>}
- <UI>::=<背景>{<交互对象>}{<应用对象>}
- <交互对象>::=<ID><类型><名称><位置><大小>
- <应用对象>::=<ID><类型><名称><位置><大小>
- <场景步>::=<交互原语><场景动作>

- 〈交互原语〉:: = 〈类型〉〈单字符名〉〈交互对象名〉〈说明〉
- 〈场景动作〉:: = 〈转子〉|〈转叶〉|〈场景任务〉
- 〈场景任务〉:: = 〈名称〉〈API〉〈说明〉

3.2 UI 形式化描述

UI 的形式化描述由两部分组成:1) 交互对象; 2) 应用对象。交互对象是 PGIS 中的 Icon, Gadget 和 Paper, 其中 Paper 就是背景。每个交互对象的信息都由两部分组成, 一是基本信息, 包括交互对象的 ID 号, 包围框, 名称等。另一个是辅助信息, 比如 Icon 可以有边框颜色, 背景色, 是否圆角, Icon 图片在操作系统上的存储路径等。又比如 Piemenu 的辅助信息包括有几个 Menu Item, 每个 Menu Item 的颜色是什么, Menu Item 的名字是什么等。应用对象只有一个抽象信息, 告诉 Visual PBAP Creator 它所在的位置和 ID。应用对象的例子是笔迹框, 媒体框。交互对象和应用对象在实现上并没有明显的区别, PGIS 中定义的 Gadget 其实就是一种应用对象, 如果某一类应用对象比较独立并且适合于笔式界面就可以添加到 Gadget 中。在 Visual PBAP Creator 中之所以这样做区分是为了将已经实现的 PGIS UI 与还在开发中的 UI 做出区别, 并且应用对象一般来讲比较独立于场景的业务逻辑。场景 UI 可以用 BNF 形式化描述如下。

- 〈UI〉:: = 〈背景〉{〈交互对象〉}{〈应用对象〉}
- 〈背景〉:: = 〈颜色〉|〈图片〉|〈GadgetId〉
- 〈交互对象〉:: = 〈ID〉〈类型〉〈名称〉〈大小〉〈位置〉
- 〈类型〉= 〈Icon〉|〈Slider〉|〈ITable〉|〈TTable〉|〈Piemenu〉|〈Palette〉
- 〈应用对象〉:: = 〈ID〉〈类型〉〈名称〉〈大小〉〈位置〉
- 〈Icon〉:: = 〈基本信息〉〈正常显示图片〉〈被按下后图片〉〈不响应图片〉〈Icon 状态〉〈矩形圆角率〉〈icon 边缘色〉〈icon 背景色〉〈text 信息〉
- 〈基本信息〉:: = 〈类型〉〈Id〉〈包围框〉〈是否响应〉〈Gadget Id 号〉
- 〈包围框〉:: = [left, top, width, height]
- 〈Icon 状态〉:: = normal|pressed|gray
- 〈图片〉:: = 图片在文件系统中的路径
- 〈text 信息〉:: = 字符串, 字号, 字体, 粗体|斜体, 颜色, 字符串偏移
- 〈slider〉:: = 〈基本信息〉〈Stable 信息〉〈Slider 信息〉
- 〈stable 信息〉:: = 〈图片路径〉〈Item 数目〉〈边缘颜色〉{〈item 颜色〉〈每个 item 所占空间比〉〈Text 信息〉}
- 〈Slider 信息〉:: = 〈对齐方式〉〈游标图片〉〈游标移动框〉

- 〈对齐方式〉:: = 0|1 (0 为纵向, 1 为横向)
- 〈游标移动框〉:: = 〈包围框〉
- 〈ITable〉:: = 〈基本信息〉〈Stable 信息〉〈IT 信息〉
- 〈IT 信息〉:: = 〈对齐方式〉
- 〈TTable〉:: = 〈基本信息〉〈Stable 信息〉〈TT 信息〉
- 〈TT 信息〉:: = 〈字符串宽〉
- 〈Piemenu〉:: = 〈基本信息〉〈Stable 信息〉〈PM 信息〉
- 〈PM 信息〉:: = 〈起始角度〉

3.3 实例

幼儿教学场景树的 XML 例子。共有 4 个场景, 其中根场景为 GRADE, 子场景为小班、中班、大班。根场景可以进入子场景, 从子场景返回根场景。在根场景中有一个场景任务为 QUIT, 在子场景中有场景任务: 播放媒体和绘画。幼儿教学场景树和状态图 XML 描述如下:

```
<scenelist count = "4">
  <scene _node id = "0" type = "ROOT" name = "GRADE">
    <boundingbox>507 84 253 169 </boundingbox>
    <ui>ui_GRADE.cfg</ui>
    <step>
      <primitive name = "I" type = "1">
        <boundingbox>411 294 64 86</boundingbox>
        </primitive>
        <action type = "0" scene_id = "1">
          <boundingbox>126 423 253 169 </boundingbox>
          </action>
        </step>
      <step>
        <primitive name = "E" type = "2">
          <boundingbox> 538 294 64 86</boundingbox>
          </primitive>
          <action type = "1" task = "INTOBIG">
            <buondingbox>126 423 253 169
            </boundingbox>
            </acction>
          </step>
        </scene_node>
      ...
    </scenelist>
  <tasklist count = "3">
    <tasknode>
      <name>QUIT</name>
      <description>QUIT API</description>
    </tasknode>
    <tasknode>
```

```
<name>PEN</name>
<description>PEN STROKE</description>
</tasknode>
<tasknode>
  <name>VEDIO</name>
  <description>PLAY VEDIO</description>
</tasknode>
</tasklist>

  幼儿教育 UI XML 例子。在根场景中有两个
  Icon,点击其中一个 Icon 可以进入大班场景。可以
  根据用户对 UI 的需求,替换 Icon 的外观。UI 和场
  景任务的挂接是通过任务列表选择完成的。幼儿教
  学 UI XML 描述如下。
<ui>
  <name>GRADE</name>
  <type>INTE_Bg</type>
  <base>INTE _ Bg, 41, [ 0, 0, 1024, 768 ], ,
0</base>
  <type>INTE_Icon</type>
  <base>INTE _ Icon, 1, [ 359, 166, 94, 65 ], ,
0</base>
  <icon_specific>
    <normal_path>filer://res/BigGrade. png
    </normal_path>
    <pressed_path>filer://res/testP. png
    </pressed_path>
    <gray_path>filer://res/testG. png</gray_path>
    <status>1</status>
    <roundedXY>0.00, 0.00</roundedXY>
    <edgecolor>0</edgecolor>
    <bghcolor>0</bghcolor>
    <textattri>( null), ( null), 0, &d, 0, [0, 0]
    </textattri>
  </icon_specific>
  <icon_api>
    <api_type>0</api_type>
    <api_name>INTOBIG</api_name>
  </icon_api>
  <type>INTE_Icon</type>
  <base>INTE _ Icon, 1, [ 366, 336, 94, 65 ], ,
0</base>
  <icon_specific>
    <normal_path>filer://res/SmallGrade. png
    </normal_path>
    <pressed_path>filer://res/testP. png
    </pressed_path>
    <gray_path>filer://res/testG. png</gray_path>
```

```
<status>1</status>
<roundedXY>0.00, 0.00</roundedXY>
<edgecolor>0</edgecolor>
<bghcolor>0</bghcolor>
<textattri>( null), ( null), 0, &d, 0, [0, 0]
</textattri>
</icon_specific>
<icon_api>
  <api_type>0</api_type>
  <api_name>INTOBIG</api_name>
</icon_api>
</ui>
```

4 代码生成流程

代码生成器分步解析场景描述 XML 文件和 UI 描述 XML 文件:1) 解析场景描述 XML 文件,根据 <step> 节点生成场景动作函数,链接已有函数库(如 CDM 库),如果函数库中没有,则生成空的函数。将这些场景动作函数填入场景模板中;2) 解析 UI XML 文件,生成 UI 创建函数、UI 消息响应函数。链接 PBOP 和 PGIS 库,将 UI 创建函数,消息响应函数具体内容填入场景模板;3) 解析场景树描述,生成主控函数。如下所示,场景模板中填入的内容以幼儿教育为例。

场景模板:

//ui 指针是特定于幼教每个场景 UI 的

```
Scenario_UIInit( * ui );
Scenario_UIShow( * ui );
Scenario_MsgHandle( msg ) {
  Switch( msg→owerId ) {
    //当前场景的某个 Icon
    case ICON_1:
    //与此 Icon 对应的场景任务或跳转
    INTOBIG();
    case GADGET_1:
    do sth.
    ...
  }
}
```

主控模板:

```
InitPGIS(); //初始化 PGIS 和 PBOP
InitCDM(); //幼教的 CDM 初始化,如果是其他应用,这边会
变化
for(;;) {
  msg = getMsg();
```

```
switch( scenario_id ) {  
    case 0:  
        //进入某场景,显示 UI,进入消息循环  
        scenario_0();  
    case 1:  
        scenario_1();  
}  
}
```

5 工具展示

5.1 场景设计器

以幼教场景设计为例。根节点为 GRADE,通过单击名为 BIG 的 Icon 可以进入大班场景。单击名为 QUIT 的 Icon 可以退出。其余的还未定义。图 4 为场景编辑器,图 5 为状态图编辑器。

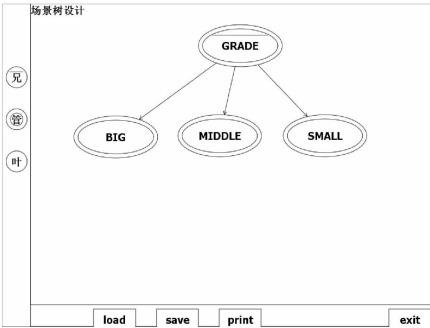


图 4 场景编辑
Fig. 4 Scenario edit

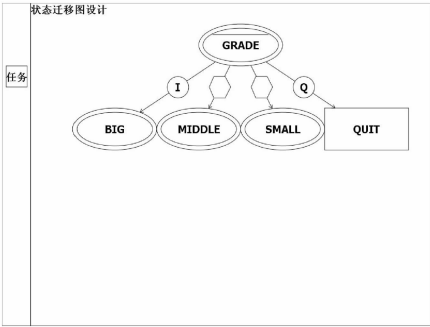


图 5 状态图编辑
Fig. 5 State diagram edit

5.2 UI 编辑器

在 GRADE 场景中设计了 4 个 Icon,可以通过修改属性来改变 Icon 的外观,拖动来改变 Icon 的布局。通过选择场景任务,将 UI 和场景动作挂接起来。UI 编辑如图 6,图 7。UI 挂载交互任务如图 8。

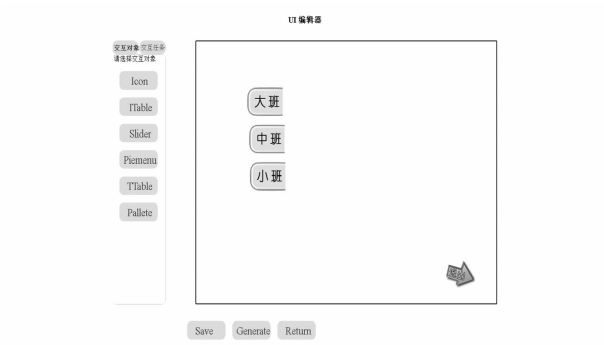


图 6 UI 编辑器
Fig. 6 UI editor

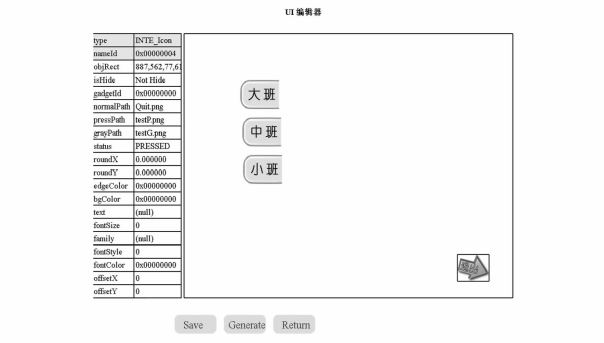


图 7 UI 属性修改
Fig. 7 Modify UI attribute

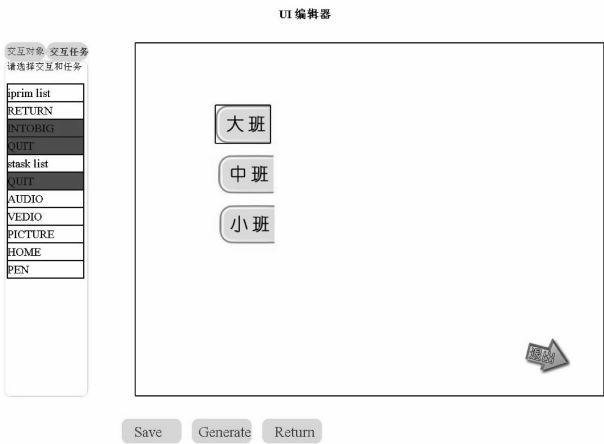


图 8 选择交互任务
Fig. 8 Select interactive task

6 结 论

阐述了建立 Visual PBAP Creator 软件设计工具的理论模型基础、软件架构以及各个模块的功能与设计流程。Visual PBAP Creator 已经应用于多个实际笔式界面软件的开发过程中,如表单系统、个人信息管理系统等。Visual PBAP Creator 能够以界面设

计人员为中心,快速确立用户需求,缩短开发周期,并能方便地完成笔式应用系统的需求、设计、实现的各阶段,为笔式应用系统的开发提供了很好的工具支持。下一步的工作是丰富 PGIS UI 的元素库,特别是 Gadget 的丰富为笔式界面软件提供更多的 UI。设计更多的库函数来支持最终用户的需求。

参考文献 (References)

[1] Tian Feng. Research on the Post-WIMP User 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.]

[2] Jason I H, Landay J A. 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, 2000: 63-72.

[3] Lin J, Newman M W, 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.

[4] Tian Feng, Qin Yanyan, Wang Xiaochun, et al. PIBG Toolkit: design and analysis of a pen-based user interface toolkit [J].

Chinese Journal of Computers, 2004, 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.]

[5] Tian Feng, Mou Shu, Dai Guozhong, et al. Research on the pen-based interaction paradigm of post-WIMP Environment [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.]

[6] Shormann C. SketchFlow Concepts: An Overview [EB/OL]. (2009) [2010]. <http://electricbeach.org/>.

[7] Liu Jian, Deng Changzhi, Dai Guozhong. Pen-based software design tool based on scenario-based method [J]. Computer Engineering and Design, 2009, 30 (8): 2011-2014. [刘建, 邓昌智, 戴国忠. 基于场景方法的笔式界面软件设计工具 [J]. 计算机工程与设计, 2009, 30 (8): 2011-2014.]

[8] Go K, Carroll J M, Imamiya A. Surveying scenario-based approaches in system design [J]. IPSJ SIG Notes, 2000, 12 (HI-87): 43-48.

[9] Wang Danli, Dai Guozhong, Wang Hongan, et al. Scenario-focused development method for a pen-based user interface: model and applications [J]. The Journal of Supercomputing (TJS), 2008, 46 (2): 139-149.

[10] Miller J, Mukerji J. MDA Guide version 1. 0. 1 [R]. Framingham, MA, USA: Object Management Group, 2003.