

一种可扩展的用户界面描述语言^{*}

杜一^{1,2}, 邓昌智¹, 田丰¹, 任磊³, 戴国忠¹

¹(中国科学院 软件研究所 人机交互与智能信息处理实验室, 北京 100190)

²(中国科学院 研究生院 信息科学与工程学院, 北京 100190)

³(北京航空航天大学 自动化科学与电气工程学院, 北京 100191)

通讯作者: 杜一, E-mail: duyic.as@gmail.com

摘要: 用户界面描述语言是实现模型驱动的用户界面开发的重要方式. 当前的用户界面描述语言一方面在对不同物理特性的交互设备上的用户界面的描述能力不足; 另一方面, 缺乏可扩展性及界面描述的组成部分的可复用性. 针对上述问题, 设计出一种界面描述语言——E-UIDL(extensible user interface description language). 该语言遵循层次化、模块化的设计原则, 能够支持多设备、多通道的用户界面的描述, 并通过实例说明描述语言对笔式用户界面开发、多设备界面自动生成以及自适应用户界面开发的支持, 深入地阐述了 E-UIDL 的特性.

关键词: 用户界面描述语言; 笔式用户界面; XML; 模型驱动; 用户界面开发

中图法分类号: TP312 **文献标识码:** A

中文引用格式: 杜一, 邓昌智, 田丰, 任磊, 戴国忠. 一种可扩展的用户界面描述语言. 软件学报, 2013, 24(5): 1127–1142. <http://www.jos.org.cn/1000-9825/4321.htm>

英文引用格式: Du Y, Deng CZ, Tian F, Ren L, Dai GZ. Extensible user interface description language. Ruan Jian Xue Bao/ Journal of Software, 2013, 24(5): 1127–1142 (in Chinese). <http://www.jos.org.cn/1000-9825/4321.htm>

Extensible User Interface Description Language

DU Yi^{1,2}, DENG Chang-Zhi¹, TIAN Feng¹, REN Lei³, DAI Guo-Zhong¹

¹(Intelligence Engineering Laboratory, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

²(School of Information Science and Engineering, Graduate University, The Chinese Academy of Sciences, Beijing 100190, China)

³(School of Automation Science and Electrical Engineering, BeiHang University, Beijing 100191, China)

Corresponding author: DU Yi, E-mail: duyic.as@gmail.com

Abstract: In model-driven approaches, user interface description languages (UIDL) often existed as the model and played an important role in the user interface development process. With the increase of the scale of software and the appearance novel devices and interaction techniques, user interface have changed greatly. Former UIDLs did not perform well in the support of new interaction paradigm. Because of this, the study proposes an extensible UIDL: E-UIDL. This UIDL follows several roles such as hierarchy and modularity. It can support the description of pen based user interfaces and multi-device user interfaces. This paper describes the features of E-UIDL in depth.

Key words: user interface description language; pen-based user interface; XML; model driven; user interface development

用户界面是指人与计算设备进行信息交换的媒介. 它在整个交互式应用程序的开发过程中占有重要的地位^[1,2]. 对用户界面的开发而言, 一方面需要界面设计及开发人员掌握很多与界面本身无关的技术, 如 HTML, Flash, Java 等各种语言以及 Adobe Dreamwaver, Adobe Flash, Eclipse 等集成设计或开发环境, 这极大地增加了用户界面开发的难度; 另一方面, 新设备及新交互技术的出现, 使得人们对用户界面, 特别是软件界面有了更多的

* 基金项目: 国家重点基础研究发展计划(973)(2013CB329305); 国家自然科学基金(61173058, 61103096, 61202217, 61100151); 国家高技术研究发展计划(863)(2012AA02A608)

收稿时间: 2012-04-12; 定稿时间: 2012-09-06

需求,如为不同类型的设备设计用户界面、为基于笔交互等不同交互方式的设备设计用户界面等等,这增加了用户界面的开发成本.如何高效、快速地设计不同交互设备以及基于不同交互方式的用户界面原型,一直是一个很重要的问题.

用户界面描述语言(user interface description language)以模型驱动的开发方法^[1]为基础,允许界面开发人员 and 设计人员使用一种规范的方式对用户界面进行描述,并利用各种生成工具,通过自动或半自动的方式,生成用户界面或最终代码.使用用户界面描述语言及相应的设计和生成工具辅助用户界面开发,能够降低界面开发人员的开发难度;并且,用户界面描述语言还能很好地适应同一界面需求的变化.用户界面描述语言有多种表现形式,包括有限状态自动机、图表、XML(extensible markup language)等,由于 XML 的平台无关性及自描述性等特点,基于 XML 的用户界面描述语言成为用户界面描述语言的一种重要的形式.

现有的基于 XML 的用户界面描述语言存在一些问题:其一,这些界面描述语言大都基于传统的基于 WIMP 界面范式的图形用户界面,并没有很好地考虑对笔式用户界面等新兴的用户界面及对应的交互技术的支持;其二,新的交互设备的出现,对多设备用户界面提出了一定的要求,但当前的用户界面描述语言几乎没有对这种需求提供支持;其三,大多数界面描述语言在可扩展性及可复用性方面考虑不足,这增大了使用界面描述语言对界面进行描述的工作量,使得界面描述语言的使用受到了局限.

基于此,本文给出了一种可扩展的用户界面描述语言——E-UIDL(extensible user interface description language),该语言使用 XML 进行描述,借鉴了现有界面描述语言的优点,通过模块化描述的方式对界面描述语言进行定义,并对不同尺寸的交互设备的用户界面设计、数据建模、模型复用及扩展提供良好的支持.

本文第 1 节对用户界面描述语言的研究现状进行介绍.第 2 节详细介绍 E-UIDL 的设计原则及各个模块的组成部分,并对各个模块之间的关系及相互作用进行阐述.第 3 节通过实例来说明使用 E-UIDL 描述笔式用户界面、E-UIDL 对不同尺寸设备的用户界面设计的支持以及 E-UIDL 对自适应用户界面描述的支持.第 4 节给出总结和下一步的工作.

1 相关工作

用户界面的发展经历了命令行界面、WIMP(windows,icons,menus,pointer)界面及 Post-WIMP/Non-WIMP^[2] 界面 3 个阶段.在对用户界面的早期研究中,研究人员提出了用户界面管理系统(UIMS)的概念.该想法来源于数据库管理系统,初衷是形成一套像数据库管理系统那样的通用、统一的规范.这类用户界面管理系统有两种不同的设计思路:一种是通过各种设计工具来对界面进行设计,并生成高级语言;另一种是独立设计交互式语言并完成全部的解析运行工作.在这些用户界面管理系统中,Kasik 等人开发的 TIGER^[3] 系统由一个定义的语言与解释器构成,能够实现对菜单的描述.多伦多大学设计开发的 MENULAY^[4] 由一个设计和实现交互式程序的工具以及一系列的运行库构成.它可以通过对界面的描述生成高层的代码.这些用户界面管理系统为用户界面的设计提供了一种思路,但它们的描述能力有限,并且受限于当时用户界面发展的约束,使得界面描述语言只能对图形用户界面或者图形用户界面的一部分进行描述,并没有得到很好的应用.

随着 XML 技术的使用以及 MDA 思想的产生,使用用户界面描述语言来辅助模型驱动的用户界面的开发成为用户界面开发的研究热点.2009 年 11 月,人机交互领域的顶级期刊 ACM Transaction on Computer-Human Interaction 刊出了用户界面描述语言的专刊,专刊中进一步说明了用户界面描述语言对于辅助用户界面开发的作用,并对很多用户界面描述语言进行了总结与分析.UIML(user interface markup language)^[5,6] 是一种扩展性很强的界面描述语言.它使用树状结构对用户界面进行描述,通过在元语言之外的各种词汇表描述的界面元素、属性和事件来增强自身的描述能力,并且能够与很多通用的标准同时工作,如 HTML, VoiceXML 等,其特点是在语言中指定界面事件对应的逻辑.但一方面, UIML 的设计在一个较高的通用层次上,其对词汇表的定义非常复杂,并且词汇之间不能重用,目前仅有有限的几种词汇表;另一方面, UIML 设计之初便以 Java 作为默认的目标语言,因此其扩展性并不如描述的那么强;再一方面, UIML 不提供友好的界面设计工具,要完成界面的设计需要完全手写基于 UIML 的界面描述文档,其花费的时间比较长.XIML(extensible interface markup language)^[7,8]

定义了多种模型,如任务模型、对话模型、用户模型、展示模型等,并通过关系和属性使得各个独立的模型相互联系.除此之外,对交互数据的良好定义是其一大特点,但与 UIML 类似,其对模型的重用以及模型设计工具方面都没有很好的支持.UsiXML(user interface extensible markup language)^[9,10]与 XIML 类似,都是对语言进行细致的模块化设计.它将界面描述分成领域模型、抽象界面模型、具体界面模型、任务模型、转换模型、上下文模型、资源模型等.在该语言中,对具体界面模型中的界面组件的描述非常丰富,但对领域模型、上下文模型的定义相对简单,并不能满足快速发展的界面需求.MARIA 以 TERESA 语言为基础,可以灵活地实现界面语言到相关实现的转换,并且对 Web 服务有比较好的支持.其主要的不足在于没有在描述时对用户界面进行抽象,使得模型的重用非常困难.以上界面描述语言本质上都是以传统的图形用户界面为基础的,并没有很好地考虑对笔式用户界面、实物用户界面、3D 用户界面等 Post-WIMP/Non-WIMP 界面的支持.

在对各种新用户界面的描述中,PUC(personal universal controller)^[11]是一种针对电器设备控制界面描述的界面语言,利用这些语言,可以自动生成桌面、手持设备等多种设备上的图形用户界面或语音用户界面.它能够描述很多复杂的电器,并且可以根据设备的不同属性生成不同的控制界面.Shaer, Jacob 及 Wingrave^[12]等人分别对实物用户界面及 3D 用户界面的描述语言进行了详细的阐述.以上几种界面描述语言面向特定的新型用户界面,虽然其扩展能力不强,仅能对界面进行有限的描述,但为设计新的界面范式下的界面描述语言提供了良好的指导.

在工业界,也有很多界面描述语言,如微软的 XAML(extensible application markup language)、Mozilla 的 XUL(XML user interface language)以及 Adobe 的 MXML 等等.一方面,这类语言主要针对传统的图形用户界面进行描述.另一方面,其描述能力都受限于企业自身的开发语言及开发平台.另外,国内也有一些通过界面描述语言来辅助开发的研究,栗阳^[13]使用混合自动机来描述 Post-WIMP 界面,特别是对笔式用户界面的描述,并在此基础上开发出一款开发工具,用来指导笔式用户界面的开发.其优势是对基于笔式用户界面的底层支持非常丰富,但对基于模型的开发方面的研究不够深入.

文中提到的用户界面描述语言,一部分具有很好的可扩展性,但不支持对界面进行不同抽象层次的描述;另一部分支持对界面的抽象,但所支持的用户界面的范围受限.本文提出一种基于 XML 的用户界面描述语言.该语言借鉴了现有界面描述语言的优点,通过模块化描述的方式对界面描述语言进行定义,支持对界面进行不同抽象层次上的描述,并对不同尺寸的交互设备的用户界面设计、数据建模、模型复用及扩展有良好的支持.

2 E-UIDL

在本节中,我们介绍 E-UIDL.首先定义 E-UIDL 的设计目标,从模块化、层次性、无关性、可扩展性及可复用性等方面对 E-UIDL 的设计目标进行描述;然后介绍抽象用户界面模块设计的理论支持:功能组模型,该模型是实现层次化界面描述和无关性的重点;第 2.3 节将对 E-UIDL 的组成模块进行详细的描述,这些模块包括抽象用户界面模块、具体用户界面模块、抽象数据描述模块、具体数据描述模块、UM2 模块、映射模块及资源模块,并对模块间关系进行详述.

2.1 E-UIDL 的设计目标

E-UIDL 应该使用模块化的描述方式,能够支持对界面的多层次的描述,并具有一定的无关性及良好的可扩展性、可复用性等特点.

E-UIDL 应该使用模块化的描述方式对用户界面进行描述.用户界面由多个部分组成,除了组成界面本身的各种组件或元素以外,还包括界面相关的交互数据、事件处理组件与交互数据之间的映射关系等等.如此复杂的用户界面,必须使用模块化的方式进行描述,将用户界面从上述不同的角度进行描述.通过这种描述方式,一方面增加了描述语言本身的可读性,降低了直接对描述语言进行编辑开发的界面开发人员的认知困难,提高了开发效率;另一方面,模块化的方式可以支持对组成界面的各个模块进行复用,而对模块的复用能力取决于描述语言本身对模块化的划分粒度.

E-UIDL 应该支持多层次的界面描述.多层次的界面描述是指描述语言应该支持从不同的抽象层次上对用

户界面进行描述.当前,用户界面的开发越来越复杂,界面的设计和开发可能需要需求分析人员、界面设计师、程序员等不同角色的人参与,不同角色的人有不同的技术专长.一种好的界面描述语言要能够对界面进行不同级别的抽象.对所有参与界面开发的人员而言,可以允许不同的角色对界面进行特定粒度的描述,而不需要其他粒度描述的特定知识.对于界面设计与生成工具而言,一方面通过使用层次化的界面描述对不同角色对界面的不同粒度的描述,可以对界面进行验证,从而增加了界面设计的安全性;另一方面,通过对粗粒度界面描述的分析,可以实现用户界面的推荐,降低了开发的复杂度,提高了界面开发的效率.

E-UIDL 应该具有无关性的特点.这里提到的无关性包括设备无关性以及通道无关性.当前存在各种不同尺寸、不同类型的交互设备,设备无关性指界面描述语言能够忽略设备的差异,其描述能力不会因为设备的不同而有所不同.通道无关性则是指界面描述语言能够描述除了鼠标点击、键盘按键、屏幕显示等传统的输入输出方式之外的其他输入输出方式,比如语音的输出输出、笔的输入等.

E-UIDL 还应具有良好的可扩展性和可复用性.交互设备的发展日新月异,基于不同交互设备的用户界面的发展变化很快.因此,界面描述语言必须有很好的扩展性,对新的用户界面形式有很好的支持.这里的扩展性表现在两个层次上:首先,要能在元语言层次上对组成语言的模块进行添加和修改;其次,要能在组成语言的模块内部对模块的组成部分进行添加和修改.可复用性不仅包括组成用户界面的各个部分的重复使用,也包括对诸如用户及环境上下文、领域知识等非界面元素的复用.通过不同层次的复用,一方面可以随着不断设计基于描述语言的用户界面,提高开发新的用户界面的效率;另一方面可以降低新的用户界面对用户和环境的学习,加快界面自适应的过程.

除此之外,界面描述语言还应该有其相适应的界面开发工具及一系列的目标代码生成工具.对于界面开发工具,应尽量提高其可用性及易用性,并且降低使用者的学习周期.任何一款强大的开发工具,如果开发者认为它很难掌握,那么这注定是一款失败的产品^[14].

2.2 功能组模型

通过对界面进行抽象,可以辅助实现层次化用户界面描述等多个优点.如何对界面进行抽象以及进行何种层次的抽象,是其中的关键问题.通过分析多种任务模型我们发现,这些任务模型大都要求对任务的描述粒度达到非常细的程度(如鼠标按下、抬起)或者对任务的分类方法过于繁琐,使得实际应用建模时难度极大.在本文介绍的 E-UIDL 中,使用我们定义的以状态转换为基础的功能组模型.功能组模型与 E-UIDL 中抽象界面模块相对应,它包括组(group)、原子组(atomgroup)、输入(input)、输出(output)以及控制块(control)这 5 个主要组成部分,每个组成部分包括自己特定的属性.功能组模型的规则如下:

- 状态属性:一个应用程序由 1 个或多个界面构成,每个界面是一个状态.
- 功能属性:在某一个界面状态,能够完成一个或多个功能.
- 功能组属性:功能组分为普通功能组及原子功能组,其中,原子组是功能的最小集合.为保证界面的流畅性及完整性,同一原子组中的部分不能出现在不同的界面中.
- 细分组属性:对于每个功能组内部,除了细分的功能组以外,还包括输入、输出及控制这 3 种不同类型的功能,或者它们的组合.
- 粒度最细原则:功能组模型的设计过程,建议设计者尽可能地降低功能组划分的粒度,以支持后续的界面推荐及界面生成.

我们设计了一种简单的符号来表示描述功能组模型.在符号表示中,实心圆表示某一特定的界面状态,分别使用不同的形状代表 Group,AtomGroup,Input,Output 及 Control.图 1 是对一种简单的笔记软件的功能组模型的符号表示.该软件主要功能为用户登录以后文字的记录,其中包括新建笔记、保存笔记、退出以及文字编辑功能.在文字编辑区域,除了基本的文字输入、输出以外,还包括上下翻页的功能.在图 1 中,登陆界面主要由用户名(AG_user)及密码(AG_psw)输入两个原子组组成登陆组(G_login),在各原子组内部,分别由一些输入、输出及控制块组成.图 2 是主编辑界面的功能组模型,其主要包括控制(AG_control)及编辑(AG_edit)两个原子组.图 3 是一种九宫格样式的功能导航界面的功能组模型的符号表示,九宫格的说法来自书法临帖写仿的格界,与黄金分割

的 4 条格线很相似,当今很多交互设备,特别是移动设备的导航界面大多采用这种界面.在图 3 中,主要的导航分为办公(G_office)、娱乐(G_entertain)及教育(G_education)这 3 个原子组,各原子组内部的导航功能都属于同一类型.根据功能组模型的描述以及定义的 XML Schema,可以生成对应的抽象用户界面模块的描述.

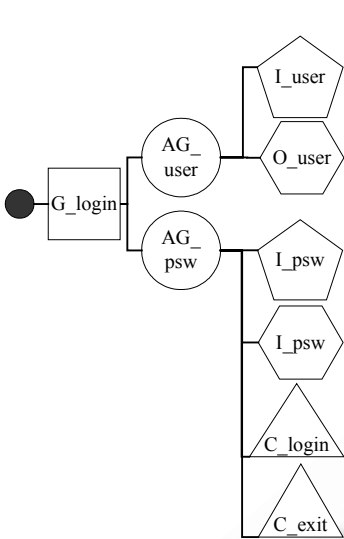


Fig.1 Symbolic notation 1 of note taking software's function group model

图 1 笔记软件功能组模型符号表示 1

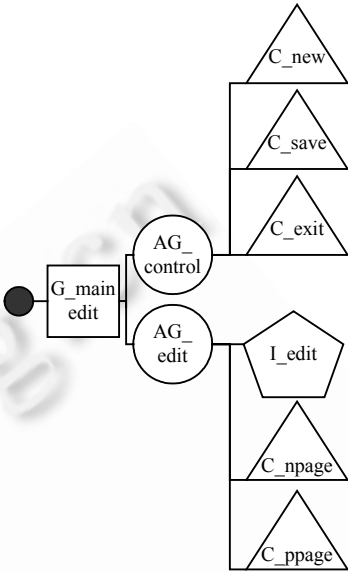


Fig.2 Symbolic notation 2 of note taking software's function group model

图 2 笔记软件功能组模型符号表示 2

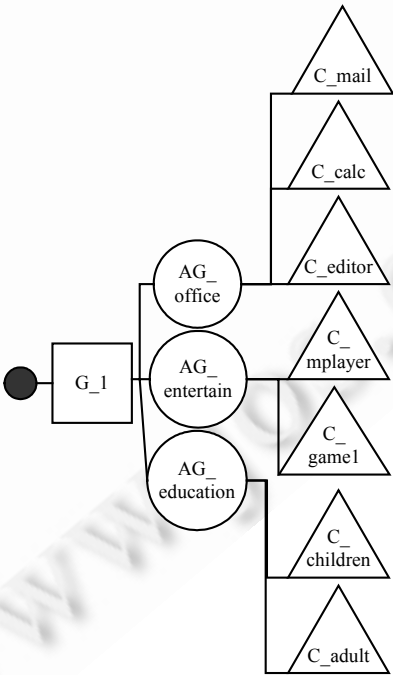


Fig.3 Symbolic notation of a navigation UI' function group model

图 3 九宫格导航界面功能组符号表示

2.3 E-UIDL组成模块

基于上述设计原则,我们给出了一种用户界面描述语言——E-UIDL.图 4~图 9 给出了 E-UIDL 主要部分的 XML Schema 结构图.图中每个矩形框表示一个节点元素,矩形框跟随的加号和减号分别表示该元素是否展开,矩形框下方的数字代表节点允许出现的次数.另外,我们分别使用内容为“S”,“C”,“A”的矩形框表示 XML Schema 描述时的“Sequence”,“Choice”及“All”等 3 种模型.并且,我们通过半形式化的方式对 E-UIDL 进行描述,该描述方法参考 EBNF 的规则,如使用大括号代表 0 或多次出现,中括号代表 0 或 1 次出现,但不对数值类型进行详细的定义,而在描述中进行说明.

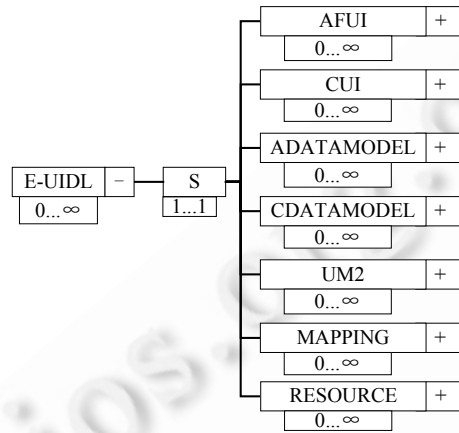


Fig.4 XML Schema of E-UIDL module

图 4 E-UIDL 模块的 XML Schema

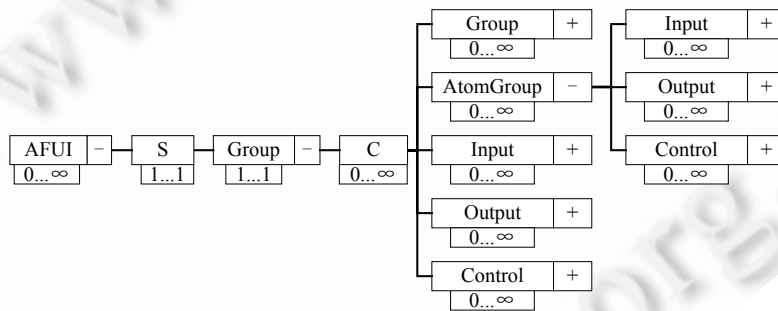


Fig.5 XML Schema of AFUI module

图 5 AFUI 模块的 XML Schema

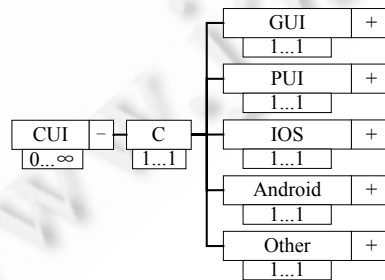


Fig.6 XML Schema of CUI module

图 6 CUI 模块的 XML Schema

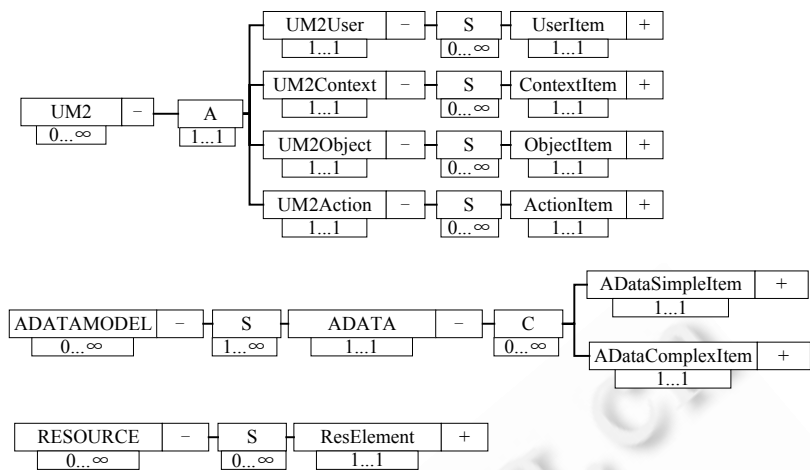


Fig.7 XML Schema of UM2 module, ADATAMODEL module and RESOURCE module
图 7 UM2,ADATAMODEL 和 RESOURCE 模块的 XML Schema

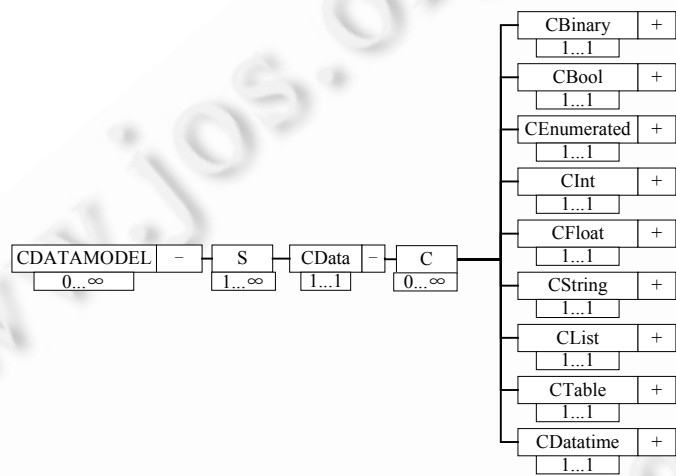
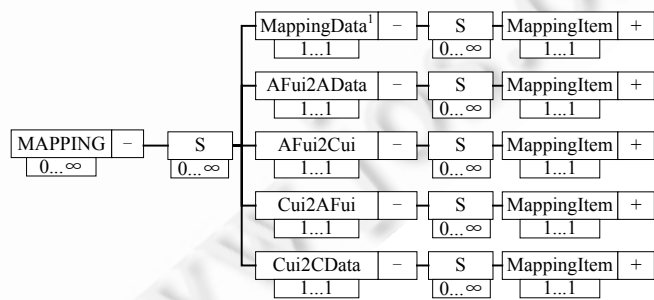


Fig.8 XML Schema of CDATAMODEL module
图 8 CDATAMODEL 模块的 XML Schema



MappingData¹:该节点是为与之前CUIDL版本兼容而保留的节点元素.

Fig.9 XML Schema of MAPPING module
图 9 MAPPING 模块的 XML Schema

在图 4 中,E-UIDL 结点是整个描述语言的根结构.它包括整个界面描述文件的基本参数,如该界面文件标示

符、名称、版本信息等等.如图 4 所示,它由多个子模块组成.这些子模块包括抽象功能界面描述模块(AFUI)、具体用户界面描述模块(CUI)、抽象数据描述模块(ADATAMODEL)、具体数据描述模块(CDATAMODEL)、UM2 模块、映射模块(MAPPING)及资源模块(RESOURCE)等.EUIDL 与每个子模块都是一对多的关系,即 EUIDL 可以包括 0 个或多个子模块的实体,如一个界面描述文件可能包括多个上下文模块、数据模块等等.另外,我们使用 UIDLHEADER 来表示所有子模块的公用的数据结构,在结构上,它是所有子模块的父类,包括子模块标示符、名称以及对子模块的描述等.其中,标示符是必须选项,通过使用标示符来保证子模块的唯一性,既可以在子模块重用通过标识符来实现外部链接,也降低了同一个描述文件中子模块之间的耦合程度,使得子模块之间相互独立.模块之间的关联方式通过独立的映射模块来实现.这部分使用半形式化方法来描述:

- $E\text{-UIDL} = \text{HeaderAttr}, \{AFUI, CUI, ADATAMODEL, CDATAMODEL, MAPPING, RESOURCE, UM2\}.$
- $\text{HeaderAttr} = Id, Name, [Desc], [Date], Version, [InheritFrom].$

AFUI 是抽象用户界面描述模块.它从一个抽象的层次对界面进行描述.该模块不与特定的界面范式有关.特定的界面范式包括对组成用户界面的各个部分的具体描述,如图形用户界面中的按钮的形状、位置或实物用户界面的实物类型等.并且,该模块除了一个默认的实现外,还允许对抽象层次自行定义.通过这种方式,一方面可与任务树等各种任务建模方法很好地结合,另一方面可以辅助实现设备无关及层次化的用户界面描述.如图 5 所示,在描述抽象用户界面时,我们使用了功能组模型的概念.它由 Group, AtomGroup, Input, Output 和 Control 这几部分组成.对于功能组模型,我们在第 2.2 节中进行了详细的介绍.这部分使用半形式化方法来描述:

- $AFUI = \text{UidlHeader}, [Group].$
- $\text{UidlHeader} = Id, Name, [Desc], [InheritFrom].$
- $Group = \text{AuiBaseType}, \{Group|AtomGroup|Input|Output|Control\}.$
- $\text{AuiBaseType} = Id, Name, [Desc], [DateType], [WeightType], [WeightValue], [HasOrder], [OrderIndex].$
- $\text{AtomGroup} = \text{AuiBaseType}, \{Input|Output|Control\}.$
- $\text{Input} = \text{AuiBaseType}, \text{AdataTypeLink}.$
- $\text{Output} = \text{AuiBaseType}, \text{AdataTypeLink}.$
- $\text{Control} = \text{AuiBaseType}, \text{DirectedUi}, \text{AdataTypeLink}.$
- $\text{WeightType} = \text{'SingleCenter'}|\text{'DoubleCenter'}|\text{'Fair'}|\text{'Customized'}|\text{'NotDefined'}.$
- $\text{HasOrder} = \text{'True'}|\text{'False'}.$

CUI 是具体用户界面描述模块.与抽象用户界面描述模块对应,具体用户界面描述模块是对用户界面的具体描述.该模块与特定的界面范式有关,描述了基于特定范式的用户界面的各个组成部分.它与抽象用户界面描述模块是一种映射关系.一种抽象用户界面描述模块可以有多种具体用户界面描述模块的具体实现.使用抽象用户界面描述模块与具体用户界面描述模块两种模块对界面进行描述,既可以保障用户界面在一定的抽象层次的可复用性,也可以使两种界面模块相互验证,保证界面的正确性.当前能够描述的 CUI 元素的集合在图 6 中呈现,各个 CUI 元素之间相对独立,便于添加新的 CUI 元素的支持.这部分主要内容使用半形式化方法描述:

- $CUI = \text{UidlHeader}, \{Gui, Pui, Ios, Android, Other\}.$
- $Gui = \{CLayout|CListView|CButton|CIcon|CScrollPane|CTabbedPane\}.$
- $Pui = \{CNinegrid|CPaper|CFrame|CIcon\}.$
- $Ios = \{IosLabel|IosTextfield|IosScopeBar|IosSearchBar|IosPicker|IosProgress|IosSlider|IosSwitch|IosIndicator|IosListView\}.$
- $Android = \{AStatic|ATextView|AEditText|APicker|ASpinner|AProgress|AButton|ACheckbox|ASwitch|AActivityBar|AActivityCircle|AList|AGridList\}.$
- $Other = \{CMetro|CPart|CCustomized\}.$

ADATAMODEL 是抽象数据描述模块.E-UIDL 包含了对数据的定义,这里所讲的数据既包括界面本身的输入输出数据,也包括界面描述语言在描述界面时用到的数据,其定义数据模块的方式也是以一种分层的方式实

现.抽象数据模块是在一个抽象的层次上对数据进行的定义,它不包括数据的具体内容.在该层模块中,可以对数据的类型及结构进行定义,这样可以允许在数据内容未知的情况下根据抽象用户界面描述模块或具体用户界面描述模块快速生成界面.从图 7 中我们可以看到,抽象数据描述主要分为简单数据和复杂数据,这与具体数据相对应.其中,简单数据用来描述包括布尔型数值、整数、浮点数、字符串等类型的数据结构,复杂数据则用来描述诸如列表、表格等由简单数据组成的数据结构.这部分使用半形式化方法来描述:

- $ADATAMODEL=UidlHeader,\{AData\}.$
- $AData=\{ADataSimpleItem|ADataComplexItem\}.$
- $ADataSimpleItem=Id,[Desc],[TypeValue],[Value].$
- $ADataComplexItem=Id,[Desc],[TypeValue],[Value].$

CDATAMODEL 是具体数据描述模块.它可以对数据进行详细而具体的描述,当前版本可以描述的数据类型包括单一的值(如二进制数值、布尔型数值、整数、浮点数)以及链表、表格等.在每种数据类型的定义中包括了数据本身的属性以及获取数据的方式.具体数据模块与抽象数据模块也可以实现相互验证,以保证界面所使用的数据的准确性.其组成参见图 8.这部分使用半形式化方法来描述:

- $CDATAMODEL=UidlHeader,\{CData\}.$
- $CData=Id,[Name],[Desc],\{CDataItem\}.$
- $CDataItem=Id,[Name],[Desc],CDataType,Source,IsFixed,Value.$
- $CDataType='CBinary'|'CBool'|'CEnum'|'CInt'|'CFloat'|'CString'|'CList'|'CTable'|'CDatetime'.$

UM2 模块是对用户行为进行描述的模块.它是我们提出的一种用户模型.该模型基于活动理论,通过分析用户日常使用交互设备的场景,将用户、上下文环境、对象及行为等统一纳入活动的研究框架,并建立了用户模型.在静态结构方面,UM2 是一个由用户(user)、上下文(context)、对象(object)、行为(action)构成的四元组.对于 UM2 的介绍不作为本文的重点,文献[15]对其进行了详细的阐述.通过在 E-UIDL 中对用户模型的支持,可以辅助自适应用户界面的描述以及生成.这部分使用半形式化方法来描述:

- $UM2=UidlHeader,\{Um2Element\}.$
- $Um2Element=Um2Context,Um2Action,Um2User,Um2Object.$

MAPPING 是映射模块.它关联了组成整个界面描述文件的各个模块,实现了这些模块之间的松耦合.通过映射模块,既可以实现抽象界面模块与具体界面模块的关联,也可以实现各种层次的界面模块与数据模块之间的关联,同时还可以实现某一界面模块与上下文模块之间的关联.通过单独的映射模块的方式,可以使组成界面模块的各个模块相互独立,提高了复用性.这部分使用半形式化方法来描述:

- $MAPPING=UidlHeader,\{MappingData|Afui2Adata|Cui2Cdata|Cui2Afui|Afui2Cui\}.$
- $MappingData=Id,[Name],[Desc],[MappingType],\{MappingItem\}.$
- $MappingItem=FromId,ToId.$

RESOURCE 是资源模块.该模块是对界面描述涉及的各种资源的描述,在对界面的描述中,可通过该模块,使用一种映射的方式将文本、图片、语音、视频等各种资源与资源模块外的其他模块关联.将资源模块进行单独管理,使得对界面的描述更加清晰,更易于对界面的管理及资源的替换.这部分使用半形式化方法来描述:

- $RESOURCE=UidlHeader,\{ResElement\}.$
- $ResElement=Id,Path,[Desc].$

利用 E-UIDL 的各个模块间的不同组合,可以使界面描述语言在不同的场景中发挥作用.如,通过具体界面模块中笔式界面相关的描述与具体数据模块的组合,可以实现对笔式用户界面的描述.这部分将在第 3.1 节中进行详细阐述.通过对抽象界面模块、具体界面模块、抽象数据模块及具体数据模块之间的组合,可以在不同的抽象层次上对界面的布局及界面的内容进行描述,通过不同的描述,实现不同设备上界面的自动生成.这部分将在第 3.2 节进行详细阐述.通过 UM2 模块及其他相关模块的组合,可以根据不同的用户模型提供不同的用户界面,实现界面的自适应.这部分将在第 3.3 节进行详述.

3 应用场景及实例

基于 E-UIDL,可以完成各种用户界面设计及开发的辅助工作,如,辅助 E-UIDL 的各种设计工具及生成工具、各种基于 E-UIDL 描述的界面自动生成以及辅助生成自适应用户界面等等.在本节中,我们通过介绍 E-UIDL 的几个应用场景来说明 E-UIDL 的各种特性.其中,使用 E-UIDL 来辅助笔式用户界面开发一节,主要阐述如何使用 E-UIDL 来辅助用户界面的开发,这包括如何使用 E-UIDL 描述用户界面特别是笔式用户界面、如何利用设计工具及生成工具来完成整个界面设计的流程等.E-UIDL 辅助多设备界面的自动生成一节,主要阐述如何利用 E-UIDL 的层次化及模块化的描述,通过各个不同抽象层次的描述模块的组合来实现不同设备的界面的自动生成.我们利用一种界面自动生成的框架,并通过一种界面自动布局算法,给出了几种 Android 平台下自动生成的用户界面.在 E-UIDL 辅助自适应用户界面生成一节,我们给出了利用 UM2 模块与其他模块的组合来实现在设计及运行时,界面的自适应特性的可能性.

3.1 E-UIDL辅助笔式用户界面的开发

界面描述语言的一个最普遍的应用在于辅助用户界面的开发.其辅助作用主要体现在:以界面描述语言作为界面模型,通过前期的设计工具设计用户界面,并生成对应的界面描述;以生成的界面描述为基础,通过各种生成工具,生成目标的用户界面.毋庸置疑,E-UIDL 也可以以该方式辅助用户界面的开发.除此之外,我们为 E-UIDL 提供了很好的笔式用户界面描述的支持,本节将以上一节提到的笔记软件的主编辑界面作为实例来验证 E-UIDL 对辅助用户界面的开发,特别是笔式用户界面的开发的支持.

笔式用户界面与传统的图形用户界面相比有很大的不同,人与计算机的交互方式从桌面环境变成了模拟的纸笔环境,界面隐喻发生了变化,使得 WIMP 交互范式已不适用于笔式用户界面^[16].戴国忠提出了一种笔式用户界面的交互范式 PIBG^[16],这是一种 Post-WIMP 交互范式.其中,P(physical objects)代表物理对象,它提供一个人机交互的基础,与窗口(window)的作用相当,但更接近于现实生活场景.I(icon)与 B(button)分别代表图标和按钮,与图形用户界面中的 IM(icon, menu)相比,IB 能够降低人机交互过程中用户注意力的转移以及认知负担.G(gesture)是指手势,使用 Gesture,使得用户不必花费精力从菜单中寻找命令.

3.1.1 工具支持

为了辅助该界面的设计,我们开发了一款设计工具.该工具基于 Eclipse 平台,可以实现对 AFUI,CUI,RESOURE 及 MAPPING 几个模块的描述.该工具使用了 UIProject 的概念.在一个 UIProject 中,包括 AFUI 设计、CUI 设计等多个模块的设计.图 10 中左图是该设计工具的 UIProject 界面.在设计时,首先创建一个 UIProject 项目,在项目内部,可以根据需求选择要进行设计的 E-UIDL 模块.图 10 中右图是工具的 AUI 的设计界面.它由一个主编辑区及各种辅助视图组成.在主编辑区中,设计人员可以对各个模块进行所见即所得的编辑,并且在主编辑区,根据不同的模块类型,提供不同的组件支持.设计工具提供的各种辅助视图可以简化模块的设计,其中,项目导航视图可以帮助设计人员方便地在各个不同的模块之间进行切换,属性视图可以对主编辑区中的某一特定内容的属性进行详细的编辑.

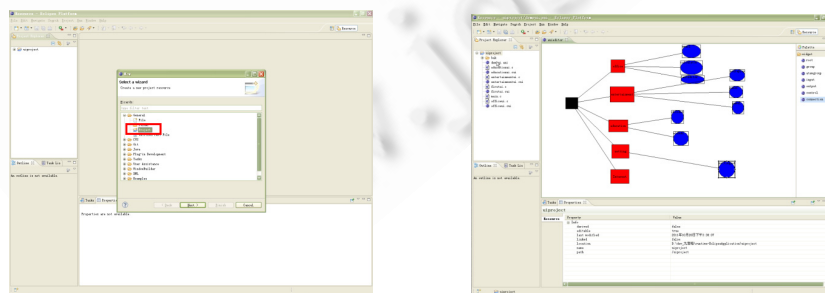


Fig.10 User interfaces of design tool

图 10 设计工具的界面

3.1.2 CUI 描述

我们使用 E-UIDL 来描述上一节提到的笔记软件的主编辑界面.该界面由两个主要部分组成,其示意图为图 11:第 1 部分为 Icon 区域,由 3 个 Icon 组成,分别为新建、保存和退出对应的图标;第 2 部分为 Note taking 区域,它是一个显示区域,允许使用笔进行输入操作,并且有两个 Icon 分别作为上翻页、下翻页对应的图标.使用 E-UIDL 描述的语言如图 12 所示.为了更清晰地把握 E-UIDL 描述界面的结构,我们简化对该软件界面的描述,在图 12 中,只使用 CUI 和 RESOURCE 两个模块进行描述,且在布局时使用绝对布局方式(layout="null").可以看出,图 12 描述了一个界面,该界面是一个“part”类型的节点.该节点包括界面的一些基本属性.同时,图 12 定义了该节点下的两种类型的子节点:“cpaper”和“cicon”.在标示符为“cuipaper01”的“cpaper”类型的节点中,定义了两个类型为“cicon”的子节点,分别对应界面中的上翻页及下翻页的图标.在“icon”类型的几个节点中,除了基本的属性的定义以外,还包括资源文件的链接.资源文件本身使用 RESOURCE 模块单独定义,通过这种方式使得资源文件与界面元素分离,可以使得资源文件的可替换性及重用性增加,方便后续界面设计的使用及界面的升级.

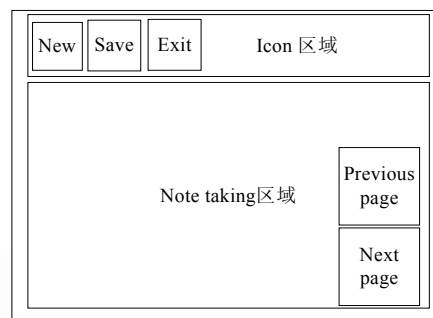


Fig.11 Sketch user interface of note taking software

图 11 笔记软件草图界面

```
<?xml version="1.0" encoding="UTF-8"?>
<E-UIDL name="puidemo" id="puidemo01" desc="This is a demo of pui uidl" date="2011-07-13"
version="0.0.11" xsi:noNamespaceSchemaLocation="E-UIDLv0.0.7.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <CUI name="puidemoCui" id="cui01" desc="cui used in puidemo">
    <cpart id="cpart01" isentrancecui="true" isfullscreen="true">
      <cuichild childid="cuipaper01"/>
      <cuichild childid="cuiicon02"/>
      <cuichild childid="cuiicon03"/>
      <cuichild childid="cuiicon04"/>
    </cpart>
    <cpaper id="cuipaper01" name="cuipaper" isentrancecui="true" isfullscreen="true" linable="true"
linecount="10" background="resitem01" layout="null" strokevisible="true">
      <cuichild childid="cuiicon05"/>
      <cuichild childid="cuiicon06"/>
    </cpaper>
    <cicon id="cuiicon02" res="resitem02" startx="5" starty="5" width="64" height="64"/>
    <cicon id="cuiicon03" res="resitem03" startx="74" starty="5" width="64" height="64"/>
    ...
  </CUI>
  <RESOURCE name="puidemoRes" id="res01" desc="resources used in puidemo">
    <reselement path="..." id="resitem01" desc="background of paper">Lgray</reselement>
    <reselement path=".../res/newnote.png" id="resitem02"></reselement>
    <reselement path=".../res/savenote.png" id="resitem03"></reselement>
    ...
  </RESOURCE>
</E-UIDL>
```

Fig.12 The first CUI description for note taking software

图 12 笔记软件的第 1 种 CUI 描述

3.1.3 界面生成

为辅助生成笔式应用程序,我们开发了一款笔式操作平台 PBOP(pen based operating system).PBOP 是一个基于笔式用户界面的开发平台,该平台对笔式用户界面有很好的支持.基于该平台的程序,可以无须修改,直接在 Windows, Linux, Android 等操作系统上运行. PBOP 平台的核心架构如图 13 所示.其中, PBAL(PBOP abstract layer)是操作系统/驱动程序与笔式应用程序之间的软件层.它隐藏了系统的细节,并且为多媒体提供了灵活的

支持.在其基础上,Ink Engine 可以根据应用程序的输入生成笔的原语.交互引擎(INTE)对各项功能进行封装,提供了一个类似 Android SDK 的开发库,PGIS 是在交互引擎基础上专门针对笔式应用程序进行的定制,基于笔的应用程序可以在其基础上进行开发.根据提供的 E-UIDL 的描述,使用生成工具生成基于 PBOP 平台的用户界面,界面如图 14 所示.

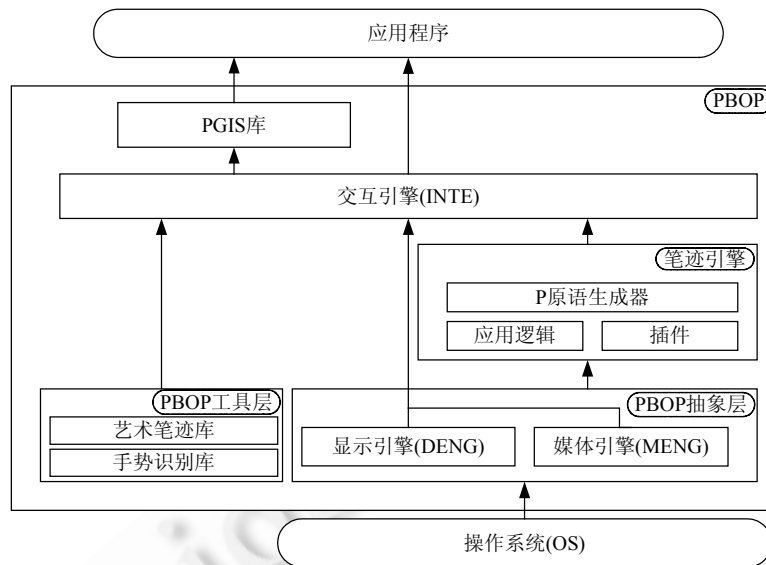


Fig.13 Architecture of PBOP

图 13 PBOP 平台架构

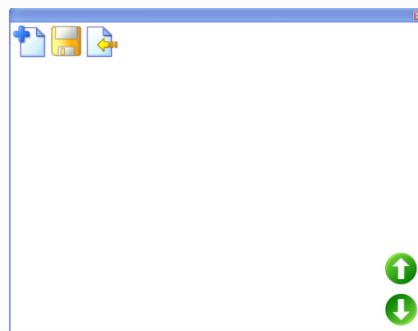


Fig.14 User interface generated by generation tool

图 14 使用生成工具生成的界面

3.2 E-UIDL辅助多设备界面的自动生成

通过对具体用户界面模块的定义,我们可以对某类特定的用户界面进行描述.但是,如果只存在具体用户界面的描述,对于各种不同的交互设备,依然需要界面开发人员通过设计具体用户界面模块的方式来实现所有的交互设备上的界面.在本节中,我们将通过实例来具体说明抽象用户界面模块及数据模块的不同组合在多种设备的用户界面的描述上发挥的作用.

在 E-UIDL 中,抽象用户界面和具体用户界面分别在不同的层次上对界面进行描述.对于具有相同功能的软件来说,其功能组模型是相同的,即其具有相同的抽象用户界面描述;但具体用户界面描述会根据目标设备的不同而各不相同.其中,抽象用户界面描述是实现多种设备的用户界面描述支持的核心.

3.2.1 多设备上的笔记软件界面描述

图 15 是对笔记软件的抽象用户界面描述的描述.图 12 是对笔记软件的一种 CUI 的描述.它使用绝对坐标进行描述,适用于普通 PC 上的界面展示.图 16 是两者之间的映射模块,表示两者之间的各个组成节点之间的映射关系.当设备变化时,屏幕尺寸、交互方式等也发生变化.在图 15 中,CUI 的描述就不适用于当前设备,需要根据 AFUI 的描述,设计或通过工具生成新设备上的用户界面.图 17 是当设备的屏幕尺寸变为当前尺寸的一半时的一种用户界面的选择.从图 17 中可以看出,由于尺寸缩小,当前屏幕已经不能在一个界面中容纳所有的功能.我们将 Icon 区域与 Note taking 区域两个原子组(atom group)使用两个单独的界面进行描述,并适当地缩小了各个控制部分(control)的尺寸,使得新界面的描述可以很好地适应新设备的尺寸.

从这个实例中我们可以看出,使用 E-UIDL 对界面进行描述以支持多种设备的用户界面的方式的一个优点是能够使用对界面的抽象描述将对不同设备的用户界面的描述关联起来.以此为基础,我们可以设计一系列的基于 E-UIDL 的工具.这些工具使用对界面的抽象描述,并通过定义相应的设备约束来自动或半自动地生成用户界面.在这种自动或半自动地生成用户界面的过程中,除了界面中各个控件的尺寸等一般属性以外,还包括界面的分屏、从图形用户界面移植到语音用户界面时的控件的重新定义等.通过这种方式,可以降低为不同目标设备生成用户界面的复杂度.

```

...
<group id="group01" desc="rootgroup">
  <group id="group011" desc="leaf1">
    <input id="input01" name="notetaking" desc="notetaking"/>
    <control id="control01" name="previous" desc="previous note"/>
    <control id="control02" name="next" desc="next note"/>
  </group>
  <group id="group012" desc="leaf2">
    <control id="control03" name="new" desc="new note"/>
    <control id="control04" name="save" desc="save note"/>
    <control id="control05" name="exit" desc="exit"/>
  </group>
</group>
...

```

Fig.15 AFUI description of note taking software

图 15 笔记软件的 AFUI 描述

```

...
<mappingdata mappingtype="aui2cui" id="mappingitem01">
  <mappingitem toid="cpart01" fromid="group01"/>
  <mappingitem toid="cuipaper01" fromid="group011"/>
  <mappingitem toid="cuiicon02" fromid="control03"/>
  <mappingitem toid="cuiicon03" fromid="control04"/>
  <mappingitem toid="cuiicon04" fromid="control05"/>
  <mappingitem toid="cuiicon05" fromid="control01"/>
  <mappingitem toid="cuiicon06" fromid="control02"/>
</mappingdata>
<mappingdata mappingtype="aui2aui" id="mappingitem02">
  <mappingitem toid="group01" fromid="group02"/>
  <mappingitem toid="group011" fromid="group021"/>
  <mappingitem toid="group012" fromid="group022"/>
</mappingdata>
...

```

Fig.16 MAPPING description between modules

图 16 各模块之间的 MAPPING 描述

```

...
<CUI name="puidemoCui" id="cui01" desc="cui used in puidemo">
  <cpart id="cpart01" isentrancecui="true" isfullscreen="true">
    <cuichild childid="cuipaper01"/>
    <cuichild childid="cuiicon02"/>
    <cuichild childid="cuiicon03"/>
    <cuichild childid="cuiicon04"/>
  </cpart>
  <cpaper id="cuipaper01" name="cuipaper" isentrancecui="true" isfullscreen="true" linable="true"
    linecount="10" background="resitem01" layout="null" strokevisible="true">
    <cuichild childid="cuiicon05"/>
    <cuichild childid="cuiicon06"/>
  </cpaper>
  <cicon id="cuiicon02" res="resitem02" startx="5" starty="5" width="64" height="64"/>
  <cicon id="cuiicon03" res="resitem03" startx="74" starty="5" width="64" height="64"/>
...
</CUI>
<RESOURCE name="puidemoRes" id="res01" desc="resources used in puidemo">
  <reselement path="..." id="resitem01" desc="background of paper">Lgray</reselement>
  <reselement path=".../res/newnote.png" id="resitem02"></reselement>
  <reselement path=".../res/savenote.png" id="resitem03"></reselement>
...
</RESOURCE>
...

```

Fig.17 The second CUI description for note taking software

图 17 笔记软件的第 2 种 CUI 描述

3.2.2 多设备上的界面自动生成

有了 E-UIDL 对多尺度界面描述的支持,我们就可以实现多尺度界面的自动生成.对于各种用户界面,使用 E-UIDL 进行不同尺度的描述.对于传统的界面开发过程来说,需要充分考虑到在需求分析过程中获取的需求内容的完备程度以及具体目标设备的类型,通过这些界面设计时依赖的前期约束,为每种不同类型的约束都生成目标界面.这样,界面设计师需要设计多次(需求完备程度种类 $\times$ 目标设备种类)界面;并且,界面使用设计工具进行描述,需要界面开发人员使用目标开发语言重新开发界面.而通过模块化、多尺度的界面描述,我们可以仅对界面进行一次模块化的描述,并指定其目标设备,通过符合我们提出的界面生成框架的步骤,自动生成符合要求的用户界面,将生成的界面提供给界面设计人员,界面设计人员在做进一步的修改后即生成目标界面.生成的界面使用目标开发语言进行描述,进一步降低了界面设计人员与界面开发人员之间沟通的成本,在一定程度上也减少了界面开发人员的工作量.我们给出一个表单应用实例,Form 是一个经典的表单界面,主要功能是填写并提交表单.该实例使用了 AFUI+CDATA 的界面描述,并且其目标设备是 WVGA 尺寸的屏幕.3 个实例的目标平台均为 Android,版本号为 2.3.3.图 18 是生成的部分界面的截图,其中,图 18(a)~图 18(d)是经典的表单输入界面.这里介绍了两种不同的界面,其差别主要在 CDATA 的定义上.在具体生成界面阶段,我们使用了类似 SmartTemplate^[17]的方式来解决目标语言在同一个控件接收不同数据时的不同呈现问题.

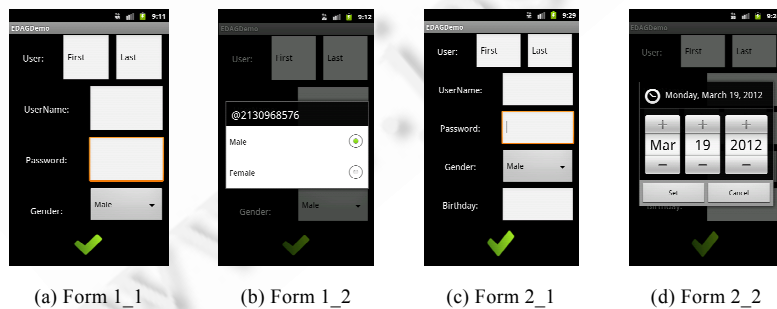


Fig.18 Generated user interfaces following by the framework

图 18 利用框架生成的用户界面

3.3 E-UIDL辅助自适应用户界面的设计

要实现自适应用户界面的设计与开发,需要有较好的模型以及建模的方法.这种模型能够对用户、上下文环境等进行建模,并实现自适应用户界面的输出.在文献[15]中,我们提出一种用户模型 UM2.该模型基于活动理论,能够将用户、上下文环境、对象及行为进行综合利用.我们将该模型引入 E-UIDL 当中,通过 E-UIDL 中 UM2 模块与 CUI 模块分别对模型及用户界面进行描述,利用 MAPPING 模块将界面与模型进行关联.在具体的代码中,我们使用文献[15]中提到的改进的向量空间模型来实现推荐的用户界面,其中生成的两个界面截图如图 19 所示.其中,两种界面均使用 E-UIDL 的 CUI 模块进行不同的描述,并且对应 1 种或多种 UM2 描述.在系统探查到当前的各种信息后,通过算法计算相似度,然后利用 MAPPING 模块中的映射关系找到对应的 CUI 描述,并给出推荐的用户界面.

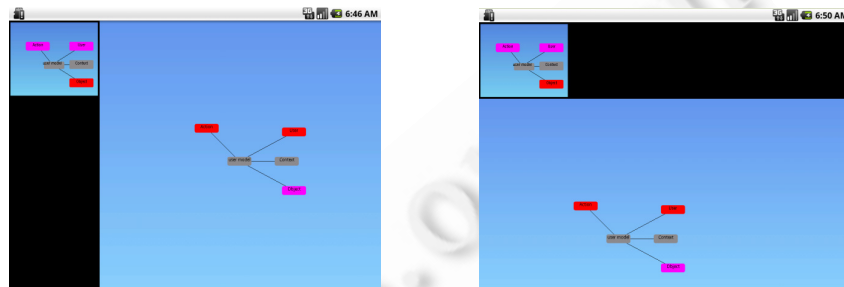


Fig.19 Screenshot of adaptive user interface

图 19 自适应用户界面截图

4 总 结

本文以模型驱动的开发方法为基础,定义了一种用户界面描述语言——E-UIDL. E-UIDL 使用模块化的描述方式,支持对界面的层次化描述,并且具有可扩展性及可复用性的特点.到目前为止,对 E-UIDL 的定义使用 XML Schema 来描述.详细定义了 AFUI, CUI, ADATAMODEL, CDATAMODEL, UM2, MAPPING, RESOURCE 这 7 个模块,能够很好地对笔式用户界面及基于笔式用户界面的多尺度界面进行描述,并通过 E-UIDL 在笔式用户界面的描述、E-UIDL 辅助多设备界面的自动生成、E-UIDL 对自适应界面的支持等方面,对 E-UIDL 的特性及使用方式进行了详细的介绍.

与较为成熟的界面描述语言相比, E-UIDL 在对上述几个方面的支持有明显的优势,并且能够通过层次化、模块化的描述,实现在不同抽象层次上的模块级别的复用.但是,一方面,本文定义的界面元素的数量还比较有限,在 E-UIDL 中,界面元素是指具体用户界面模块中的节点定义.当前的版本中仅定义了图形用户界面的基本组件以及构成笔式用户界面的大部分元素,虽然能够对一部分的图形用户界面及大部分的笔式用户界面进行描述,但充足的界面元素的定义能够进一步提升 E-UIDL 的描述能力,另一方面,随着界面的发展,对于界面的动态行为及界面间复杂逻辑的要求会进一步加强,因此对动态行为进行描述的要求会越来越高.由于 E-UIDL 本身的可扩展性,在下一步工作中,我们参考文献[13]中的方法,使用形式化的方法对界面的动态行为及更加复杂的界面逻辑进行描述,以增强 E-UIDL 的描述能力.

References:

- [1] Szekely P. Retrospective and challenges for model-based interface development. In: Proc. of the Design, Specification, and Verification of Interactive Systems (DSV-IS). Namur: Springer-Verlag, 1996. 1–27. [doi: 10.1007/978-3-7091-7491-3_1]
- [2] Green M, Jacob R. Software architectures and metaphors for non-wimp user interfaces. Computer Graphics, 1991,25(3):229–235. [doi: 10.1145/126640.126677]
- [3] Kasik DJ. A user interface management system. SIGGRAPH Computer Graphics, 1982,16(3):99–106. [doi: 10.1145/965145.801268]

- [4] Buxton W, Lamb MR, Sherman D, Smith KC. Towards a comprehensive user interface management system. In: Proc. of the 10th Annual Conf. on Computer Graphics and Interactive Techniques. Detroit: ACM Press, 1983. 35–42. [doi: 10.1145/800059.801130]
- [5] Phanouriou C. UIML: A device-independent user interface markup language [Ph.D. Thesis]. Blacksburg: Virginia Polytechnic Institute and State University, 2000.
- [6] Abrams M, Phanouriou C, Batongbacal AL, Williams SM, Shuster JE. UIML: An appliance-independent XML user interface language. Computer Networks, 1999,31(11-16):1695–1708. [doi: 10.1016/S1389-1286(99)00044-4]
- [7] Eisenstein J, Vanderdonckt J, Puerta A. Applying model-based techniques to the development of UIs for mobile computers. In: Proc. of the 6th Int'l Conf. on Intelligent user Interfaces. New York: ACM Press, 2001. 69–76. [doi: 10.1145/359784.360122]
- [8] Puerta A, Eisenstein J. XIML: A Universal Language for User Interfaces. White Paper, 2001.
- [9] Limbourg Q, Vanderdonckt J, Michotte B, Bouillon L, López-Jaquero V. Usixml: A language supporting multi-path development of user interfaces. Engineering Human Computer Interaction and Interactive Systems, 2005,3425:200–220. [doi: 10.1007/1143187_9_12]
- [10] Limbourg Q, Vanderdonckt J, Michotte B, Bouillon L, Florins M, Trevisan D. Usixml: A user interface description language for context-sensitive user interfaces. In: Proc. of the ACM AVI 2004 Workshop “Developing User Interfaces with XML: Advances on User Interface Description Languages”. Gallipoli: Citeseer, 2004. 55–62.
- [11] Nichols J, Myers BA. Creating a lightweight user interface description language: An overview and analysis of the personal universal controller project. ACM Trans. on Computer-Human Interaction, 2009,16(4):1–37. [doi: 10.1145/1614390.1614392]
- [12] Wingrave CA, Joseph J, Laviola J, Bowman DA. A natural, tiered and executable UIDL for 3D user interfaces based on concept-oriented design. ACM Trans. on Computer-Human Interaction, 2009,16(4):1–36. [doi: 10.1145/1614390.1614396]
- [13] Li Y, Guan ZW, Dai GZ. Modeling post-WIMP user interfaces based on hybrid automata. Ruan Jian Xue Bao/Journal of Software, 2001,12(5):633–644 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/12/633.htm>
- [14] Myers BA. User interface software technology. ACM Computing Surveys, 1996,28(1):189–191. [doi: 10.1145/234313.234394]
- [15] Du Y, Tian F, Dai GZ, Wang F, Wang HA. A user model based on mobile environment. Ruan Jian Xue Bao/Journal of Software, 2011,22:120–128 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/11032.htm>
- [16] Dai GZ, Tian F. Pen Based User Interface. Changsha: University of Science and Technology of China Press, 2009. 25–45 (in Chinese).
- [17] Nichols J, Myers BA, Litwack K. Improving automatic interface generation with smart templates. In: Proc. of the 9th Int'l Conf. on Intelligent User Interfaces Funchal. Madeira: ACM Press, 2004. 286–288. [doi: 10.1145/964442.964507]

附中文参考文献:

- [13] 栗阳,关志伟,戴国忠.基于混合自动机的 Post-WIMP 界面的建模.软件学报,2001,12(5):633–644. <http://www.jos.org.cn/1000-9825/12/633.htm>
- [15] 杜一,田丰,戴国忠,王锋,王宏安.一种移动环境下的用户模型.软件学报,2011,22:120–128. <http://www.jos.org.cn/1000-9825/11032.htm>
- [16] 戴国忠,田丰.笔式用户界面.长沙:中国科学技术大学出版社,2009.25–45.



杜一(1988—),男,山东聊城人,博士生,主要研究领域为人机交互技术,笔式用户界面.
E-mail: duyicase@gmail.com



任磊(1979—),男,博士,讲师,主要研究领域为信息可视化,云制造.
E-mail: leo.renlei@gmail.com



邓昌智(1978—),男,博士,助理研究员,主要研究领域为人机交互技术,个人信息管理,信息可视化.
E-mail: dcziel@gmail.com



戴国忠(1944—),男,研究员,博士生导师,CCF 高级会员,主要研究领域为人机交互技术,笔式用户界面.
E-mail: dgz@iel.iscas.ac.cn



田丰(1976—),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为人机交互技术,虚拟现实.
E-mail: tf@iel.iscas.ac.cn