

E-UIDL 用户界面描述语言下的开发方法*

杜一^{1,2}, 田丰², 戴国忠²

¹(中国科学院 计算机网络信息中心 科学数据中心, 北京 100190)

²(人机交互北京市重点实验室(中国科学院 软件研究所), 北京 100190)

通讯作者: 杜一, E-mail: duy1@cnic.cn

摘要: 当前的用户界面开发过程需要产品经理、交互设计师、视觉设计师及用户界面开发工程师等多个角色的共同参与,各个角色的分工越来越细致,并且不同的角色在开发过程中使用的工具各不相同,这导致了界面开发过程中沟通成本的增加以及工作效率的降低.介绍了一种基于用户界面描述语言的开发方法,该方法能够降低用户界面开发过程中不同角色之间的沟通成本及转换成本,提高用户界面开发的效率.首先介绍 E-UIDL(extensible user interface description language)——一种新的用户界面描述语言,并在此基础上设计了基于 E-UIDL 的开发方法及辅助开发工具,最后,通过实例介绍了利用基于 E-UIDL 的开发工具进行界面开发的过程,证明了基于用户界面描述语言的开发方法的可行性及优势.

关键词: 界面描述语言;交互技术;用户界面开发方法;计算机辅助设计

中图法分类号: TP311

中文引用格式: 杜一,田丰,戴国忠.E-UIDL 用户界面描述语言下的开发方法.软件学报,2015,26(7):1772-1784. <http://www.jos.org.cn/1000-9825/4584.htm>

英文引用格式: Du Y, Tian F, Dai GZ. Development approach based on extensible user interface description language. Ruan Jian Xue Bao/Journal of Software, 2015, 26(7): 1772-1784 (in Chinese). <http://www.jos.org.cn/1000-9825/4584.htm>

Development Approach Based on Extensible User Interface Description Language

DU Yi^{1,2}, TIAN Feng², DAI Guo-Zhong²

¹(Scientific Data Center, Computer Network Information Center, The Chinese Academy of Sciences, Beijing 100190, China)

²(Beijing Key Laboratory of Human-Computer Interaction (Institute of Software, The Chinese Academy of Sciences), Beijing 100190, China)

Abstract: The division of roles becomes more and more complex in user interface development, and the tools used in the development process vary greatly. These problems result in increase of communication cost and decrease of efficiency. This article introduces a development method based on user interface description language. First, a new user interface description language E-UIDL is proposed. Then, a series of supporting tools based on E-UIDL are designed. Next, the development process of the user interface based on E-UIDL and related tools is presented. Finally, the feasibility of the proposed method is validated.

Key words: user interface description language; HCI; user interface development method; computer aided design

1 引言

1.1 问题提出

用户界面开发,在整个软件开发过程中占有很重要的地位.研究表明:在交互式应用程序中,有 48%的代码与用户界面的开发直接相关;在软件开发过程中,50%的时间消耗在了与界面代码有关的工作中^[1,2].随着交互设

* 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(61402435, 41371386)

收稿时间: 2013-04-22; 修改时间: 2013-09-09; 定稿时间: 2014-02-17

备及交互技术的不断发展,以及软件工程在软件产品开发中的应用不断深入,当前的用户界面开发产生了新的问题:

一方面,在软件产品的开发过程中,各个角色的分工越来越细,在整个过程中,需要产品经理、交互设计师、视觉设计师、界面开发工程师等不同角色的参与,各个不同的角色,掌握不同的专业技能,在各自的领域内支持整个产品的开发.但是,正是由于各个不同角色掌握着不同的专业技能,各自通过不同的工具来辅助自己的工作,使得各个角色之间的沟通存在一定的问题;并且各个角色将交付物交付给产品开发流程中的下一角色时,下一角色需要将其进行转换,以使得自己的角色能够理解和继续工作.沟通的不顺畅以及不同角色交付物之间的转换工作,降低了产品开发的效率.

另一方面,各种交互设备及交互技术层出不穷,即使仅为智能手机开发软件产品,就需要考虑移动操作系统、设备尺寸、设备支持的交互方式等多个要素,并根据不同的要素设计软件的不同用户界面,这更增加了用户界面设计及开发的工作量.

上述两方面综合作用,使得当前产品开发遇到了极大的挑战.而当前主要通过产品经理的管理与及时的沟通来降低该风险,很少有通过工具来解决这些问题的方法.

模型驱动的用户界面开发^[3,4]是用来辅助用户界面开发的一种有效方式,该方法使用模型来对用户界面进行描述,并且以模型为中心,通过设计和建模工具、实现和生成工具来完成整个用户界面的开发流程.用户界面描述语言^[5]在整个应用系统的开发中扮演了核心模型的角色,它通过一系列规范的语言对用户界面的各个组成部分进行描述.使用用户界面描述语言对界面进行描述有多种方法,其中,XML 等标记语言由于其较强的可扩展性及文档间的互操作性,成为用户界面描述语言的主要研究方法.在此基础上,有很多基于用户界面描述语言的工具产生,并用以辅助用户界面的开发.我们发现:一方面,由于界面描述语言本身的描述能力及扩展性等方面的不足,使其在整个用户界面开发过程中发挥的作用并不明显;另一方面,虽然各种描述语言大都有自己的工具来辅助界面的开发,但是都没有考虑整个界面开发过程中角色的划分问题,使得各种工具在当前细分工的环境下的应用存在一定的障碍.

1.2 问题分析

为了对当前产品中用户界面开发流程及其中存在的问题有更清晰的了解,我们展开了一项用户研究.这里指的用户,是用户界面开发流程中涉及到的主要人员.我们找到了产品经理、交互设计师、视觉设计师及界面开发工程师这 4 个角色的用户,其中每个角色的用户至少 1 人,在所有用户中,有 1 人同时从事交互设计师及视觉设计师的角色.我们主要采用用户访谈的方式,每名用户访谈时间约为 40 分钟,其中,访谈涉及的主要问题包括:用户所在公司的产品,特别是用户界面的开发流程;用户在产品开发中扮演的角色及主要职能;在产品开发过程中的接收物及提交物以及工作中所使用的工具;如何避免因为各个角色之间沟通而导致的用户界面的缺陷;存在的因为沟通问题导致的用户界面的缺陷及相关的处理办法.除此之外,我们还针对不同的角色提出了一些更加具体的问题.

虽然在约谈的用户中,所在的企业背景各不相同,但其产品开发的流程基本类似.首先,由产品经理或需求工程师确定产品的需求及主要功能点;然后,由交互设计师及视觉设计师进行交互设计及视觉设计;在交互设计师展开设计后,界面开发工程师开始进行产品的底层开发及界面开发;在此过程中,还需要由交互设计师、视觉设计师及界面开发工程师沟通确定所需资源的具体属性.整个过程是一个不断迭代的过程,主要通过一定的规则来防止过度的迭代对开发进度的影响.所有约谈的用户都认为:在整个开发过程中,有很多需要优化的工作,而且自己都碰到过因为沟通导致的问题.除此之外,在整个过程中还存在的问题主要有:

- 不同的角色要了解各个角色使用的工具,并能够理解不同工具的不同输出物,将这些输出物转换成自己工作需要的信息.对交互设计师而言,需要从需求或功能列表(多为 Excel 文档)中抽象出软件的架构信息,在此基础上进行交互设计;对视觉设计师而言,需要从交互设计文档(多为 PPT 或 Flash 原型)中提取具体应用的布局信息,然后使用设计工具作进一步的视觉设计;界面开发工程师需要从交互设计文档进行界面的开发及实现.

- 当前的开发流程,如果要为不同的设备设计不同的用户界面,如为非智能手机、智能手机或平板电脑设计并开发用户界面,需要完全开始一个新的开发流程,流程只能进行细粒度、低级别上的复用,重新开发的效率较低;并且由于人力资源有限,在多数情况下,只将产品或项目的关注点放在某一具体的目标设备或目标人群,放弃了其他类型的目标设备以及目标人群。

基于对以上问题进行分析和调研,本文采用基于 E-UIDL 的方式进行分析.E-UIDL 使用层次化、模块化的描述方法加以描述,既能够在不同的抽象层次上对用户界面相关的内容进行描述,也能够支持不同设备的用户界面的描述,模块化的描述方式使得开发过程中不同角色通过同一工具进行工作成为可能.在此基础上,我们进一步提出了基于 E-UIDL 的用户界面开发方法,该方法对传统的基于模型的用户界面的开发加以优化,能够处理开发过程中出现的 inconsistent 的情况.接下来,我们介绍在基于 E-UIDL 的用户界面开发方法中用到的设计工具及生成工具,这种工具允许产品经理、交互设计师使用同一工具进行工作,并且通过界面的自动生成技术直接生成界面的目标代码.在使得视觉设计师及界面开发人员更直观地看到最终产品的同时,降低了界面开发人员的工作量,使得他们能够从一个原型界面进一步地开发,实现最终的用户界面.最后,我们给出基于 E-UIDL 及辅助工具的应用实例及讨论.

2 相关工作

用户界面描述语言(user interface description language)是通过一系列规范的语言对用户界面的各个组成部分进行描述,在整个基于模型的用户界面的开发过程中扮演了核心模型的角色.使用用户界面描述语言对界面进行描述有多种方法,如使用图表^[6,7]、状态转换图^[8]、XML 等.其中,XML 等标记语言由于其较强的可扩展性及文档间的互操作性,成为用户界面描述语言的主要研究方法.

学术界与工业界对基于 XML 的用户界面的描述语言都做了大量工作.其中,PUC Specification Language^[9]是专门为描述各种电器控制设备界面而设计的描述语言,它能够以图形用户界面和语音界面两种方式描述多达 30 多款电器设备.该语言从功能和内容两个方面对基于电器设备的用户界面进行描述.除此之外,PUC 还针对电器设备界面的特性,提出了“智能模板(smart template)”来解决电器控制设备界面的特殊展示方式的问题,如播放器的通用界面展示.在工具支持方面,有两款基于 PUC 的辅助工具,分别是 UNIFORM^[10]与 HUDDLE^[11].UNIFORM 主要面临的是在普适计算环境下,各种家庭中的电器设备的整合经常出现各种电器界面的前后不一致或控制方式不一致等,如在 VCR 与闹钟上设置时间是完全不同的流程.该工具通过一种对 PUC Specification Language 进行相似性比较的方法来解决这一问题,并生成最终界面.Huddle 也是基于 PUC Specification Language 的一个自动生成工具,该工具主要面向的是类似家庭影院、电视会议系统之类的相互关联的电器(controlled appliance)的控制界面.这类系统的问题是:虽然它们的功能整合到了一起,但界面却没有,要想操作它们,必须在每种设备的界面上进行操作以达到目的.这两种工具都是基于该语言的实用工具,成功地将各种算法应用到界面自动生成的技术中,解决了电器设备界面上很多关键问题.

UsiXML^[12-15]是一种基于 XML 的用户界面描述语言,该语言能够对命令行用户界面、图形用户界面、听觉用户界面以及多通道用户界面等用户界面进行描述.它通过对语言进行细致的模块化设计,将界面描述分成领域模型、抽象用户界面模型、具体用户界面模型、任务模型、转换模型、上下文模型、资源模型等,来提高用户界面描述语言的可复用性及可扩展性.它允许在一个 UsiXML 描述中对多个用户界面进行描述,这样可以使得一次描述的用户界面适用于多个不同的物理设备,但是界面描述的标签的数目会相应地增加.UsiXML 有比较丰富的工具支持.

- 在设计端,有 GraphXML,VisiXML,SketchiXML,IdealXML,PlastiXML,ComposiXML 等设计工具,通过这些不同的设计工具,可以设计不同的用户界面,并且生成遵循 UsiXML 格式的描述;
- 在生成端,有 KnowiXML,IKnowYou,CodeGenerators,FlashiXML,QtKiXML,RSSRenderer 等多种生成工具,可以将 UsiXML 的描述生成不同的平台上的代码或者可执行程序;
- 除此之外,为了进一步支持界面的设计,还有很多知名设计工具的扩展,如 VisiXML 是基于 MS Visio 的

一种扩展,该工具能够支持在 Visio 上设计 UsiXML 的界面,并把界面生成为多种目标代码。

XAML(<http://msdn.microsoft.com/en-us/library/ms752059.aspx>)是微软提出的一种用户界面描述语言,该语言适用于 .Net 平台,通过设计基于 XAML 的用户界面描述,将用户界面与运行时逻辑分离,具有一定的创新性。该语言属于微软推出的一种商业用户界面描述,用来支持基于 .Net 平台的应用程序的开发。由于商业上的考量,这类商业的用户界面描述语言与科研领域中的用户界面描述语言有很大的不同。最主要的一点是,这类商业的界面描述语言只面向特定的开发平台或高级程序开发语言。一方面,面向特定的开发平台降低了界面描述语言本身以及支持其发挥作用的设计工具与生成工具的复杂度与开发成本,如果最终用户界面没有多平台或多设备的需求,则是一种很好的选择;另一方面,如果应用程序需要运行在多个平台上,或者运行在不同的设备上,那么对用户界面会有不同的需求。此时,如果使用商业的界面描述语言,则不仅需要为不同设备上的用户界面使用不同的描述语言进行描述,甚至对于同一个用户界面,也需要使用不同的用户界面描述语言进行描述,这在很大程度上增加了开发成本。而各种交互设备的出现,使得这样的需求越发紧迫,更加暴露出这种传统的商业用户界面描述语言的问题。

除此之外,我们还分析了 UIML, XIML, ChasmXML, XUL^[16-18]等多种用户界面描述语言及对应的工具。我们发现:大多数界面描述语言描述能力有限,只能对传统的图形用户界面进行描述,并且界面描述语言没有很好的扩展性。虽然很多界面描述语言有对应的辅助工具,但都没有考虑真实的软件产品的开发流程及开发过程中的角色分工,要求所有的工作都由一个角色完成,该角色需要掌握开发过程中相当多的相关知识。造成这种现状的原因:一方面是由于工具开发者没有很好地对开发过程进行分析;另一方面,也是由于界面描述语言本身的可扩展性及层次化的描述能力不够。

3 基于 E-UIDL 的用户界面开发

3.1 E-UIDL 概述

E-UIDL 是一种新的用户界面描述语言^[19-22],该语言能够支持对界面的层次化、模块化的描述,并具有无关性及良好的可扩展性、可复用性等特点。图 1 所示为 E-UIDL 的 XML Schema 主体结构图,在图中,E-UIDL 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。

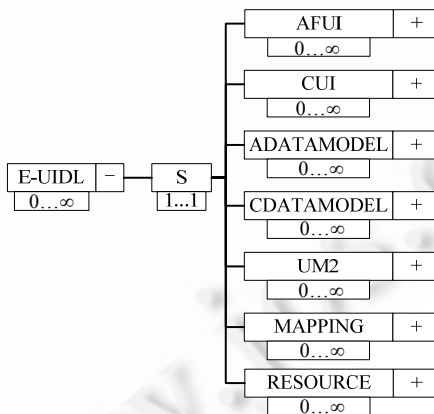


Fig.1 XML Schema of E-UIDL

图 1 E-UIDL 的 XML Schema

如图 1 所示,E-UIDL 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^[20]模块、映射模块(MAPPING)以及资源模块(RESOURCE)等。其中,抽象功能界面模块、具体用户界面模块主要对界面的

组成部分及组成元素进行不同抽象层次的描述;抽象数据描述模块及具体数据描述模块主要对界面中需要输入或输出的数据进行不同抽象层次的描述;UM2 模块主要实现对用户的建模;映射模块主要实现各个不同子模块之间的关联;资源模块主要描述模型描述过程中使用到的资源.E-UIDL 与每个子模块都是一对多的关系,即,E-UIDL 可以包括零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.在文献[19-22]中对 E-UIDL 有更加详细的阐述.通过 E-UIDL 各个模块之间的组合,可以实现对笔式用户界面设计与开发、多尺度界面生成以及自适应界面设计等多种功能.

3.2 基于 E-UIDL 的用户界面开发流程

图 2 给出基于 E-UIDL 的用户界面开发方法示例,其中,矩形框表示流程中涉及到的模块及子模块,实线箭头表示数据在整个流程的流转,虚线箭头表示数据在不同角色之间的流转.该流程以 E-UIDL 为核心,在传统的基于模型的用户界面开发方法的基础上,明确指出流程中各个环节的不同角色、工作方式及所使用的工具的类型.基于 E-UIDL 的开发方法由 5 部分组成,在图中,从左到右分别代表开发流程中的相关部分.这 5 部分分别是用户(user)、设计过程用到的工具(design tool)、用户界面模型(UI model)、生成过程用到的工具(generation tool)以及用户界面(user interface).

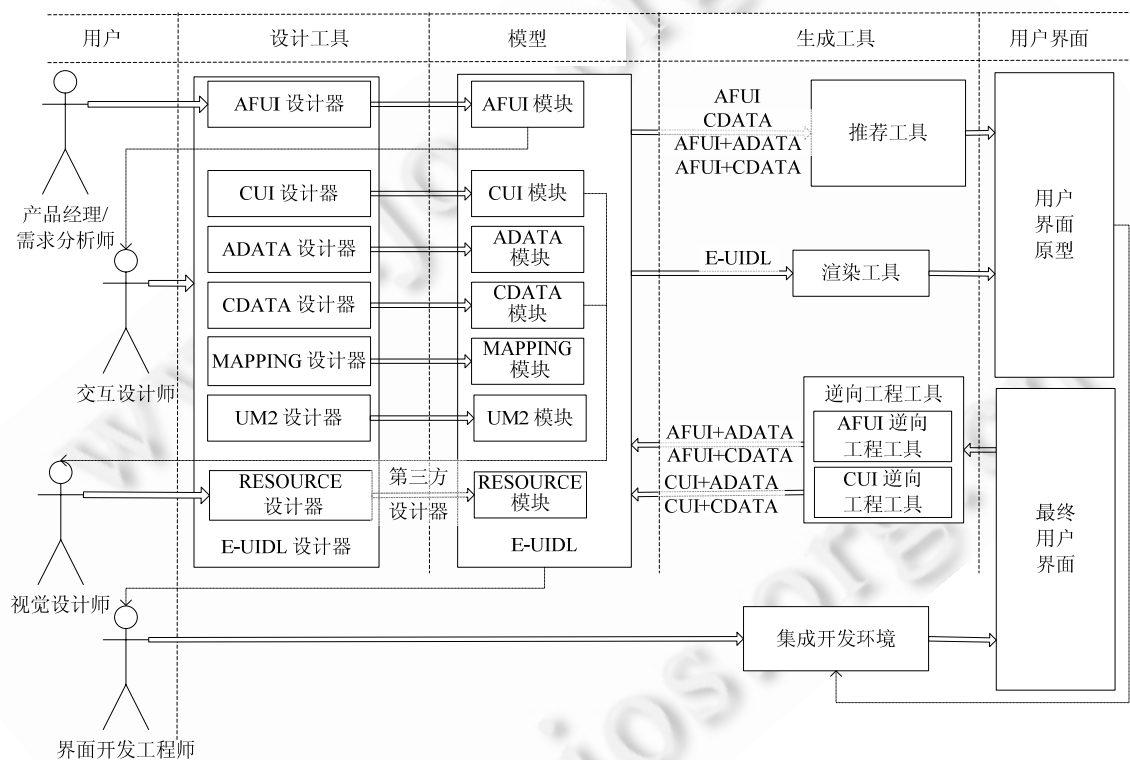


Fig.2 User interface development process based on E-UIDL

图 2 基于 E-UIDL 的用户界面开发方法

整个开发流程中,主要的相关角色包括 4 种,分别是产品经理/需求分析师(product manager/requirement analyst)、交互设计师(HCI designer)、视觉设计师(visual designer)与界面开发工程师(UI developer).用户界面模型部分使用 E-UIDL 进行描述,如上文所述,它由 7 个不同的模块组成.在模型的前端,是开发流程中设计过程要用到的工具,这些设计工具与界面模型相对应,即界面模型的每个组成模块都有对应的设计工具,这些设计工具在整个流程中由不同的角色使用.在模型的后端,是通过工具自动或半自动生成用户界面的过程,该过程也由一

些工具进行辅助的支持.在生成过程部分中,我们可以利用所生成模型的不同描述方式,根据映射描述的映射关系,通过界面推荐工具(recommendation tool)自动生成一些不同粒度的用户界面,推荐给不同的角色.在生成部分,还包括渲染工具(rendering tool),该工具利用模型的具体描述及数据和资源描述,根据映射描述的映射关系,实现界面的生成.通过渲染工具,可以直接生成用户界面的具体实现.除此之外,在这一部分还包括一些逆向工程工具(如 AFUI RE tool, CUI RE tool),这些工具从现有的用户界面中提取出界面模型的 E-UIDL 描述,从而实现一些系统移植等需求.在用户界面部分,有原型界面(user interface prototypes)及最终界面(final user interfaces)两种不同的界面,其中,原型界面是指通过界面推荐工具及渲染工具自动生成的用户界面,这些界面为不同的角色服务,用于不同角色了解产品的界面以及界面的二次开发.最终界面则是产品设计的阶段性的终点,该界面可以作为测试版本或最终的发布版本,由界面开发工程师在原型界面的基础上二次开发.在传统软件开发领域中,由于对用户界面的要求较低,因此从原型界面到最终界面之间,仅需要很少的二次开发工作.

基于 E-UIDL 的界面开发流程如下所示:

- (a) 流程开始时,产品经理或需求分析师利用 AFUI 设计器对产品的需求及功能点进行描述,生成遵循 E-UIDL 的 AFUI 模块的描述或 ADATA 模块的描述(可选).在此过程中,他可以利用推荐工具,浏览基于 AFUI 模块或 ADATA 模块生成的推荐界面,对整个界面及布局情况有总体的了解.其中,推荐界面是指推荐工具基于 E-UIDL 的不同模块生成的界面,该界面可以提供给界面相关的参与人员,以辅助相关人员进行决策及进一步的开发;
- (b) 生成 AFUI 模块的描述后,交互设计师则根据 AFUI 模块的描述及自己的经验进行竞品分析、交互设计等.在该过程中,交互设计师也可以利用推荐工具来生成原型界面辅助设计.利用 E-UIDL 设计器的各个部分,在原型界面的基础上,或者重新对界面的布局、形态及交互进行设计,生成 E-UIDL 的较为完整的描述,该描述可以作为提交物提交给视觉设计师及界面开发工程师;
- (c) 视觉设计师在拿到交互设计师设计的 E-UIDL 的描述后,参考渲染工具生成的原型界面及 E-UIDL 对视觉资源的具体描述进行视觉设计.在此设计过程中,主要利用一些第三方的视觉设计工具进行设计,然后通过 Resource 设计器进行资源的映射;
- (d) 界面开发工程师在获取交互设计师的 E-UIDL 的描述后,即进行用户界面的开发工作,他可以利用渲染工具生成原型界面;并在此基础上,利用第三方的集成开发环境,对界面进行实现.并且,在视觉设计师的视觉设计交付后,可以对各种资源加以替换,形成最终产品的用户界面.

整个的界面设计与开发过程是一个不断迭代的过程,在该过程中,需要各个角色之间进行沟通,我们给出在访谈中遇到的两个问题,并介绍如何通过我们提出的工具及流程来改善这两个问题.

第一,交互设计师与产品经理对交互设计进行沟通.

我们了解到,交互设计师会通过草图绘制出多种可选的线框图,并与产品经理等其他角色进行讨论,最终确定产品的界面及交互.在此过程中,交互设计师需要进行频繁的设计,并且最终只有一种设计被交付给流程的下一角色.而使用推荐工具,可以在 AFUI,ADATA 模块的基础上生成界面的可选结果,供交互设计师与各个角色进行交流,并选择一种或两种作进一步的设计.这样就减少了交互设计师的工作,而且具体实现的原型界面相比线框图更加直观,易于各个角色之间的沟通.

第二,界面开发工程师进行界面开发.

在被访谈的两位界面开发工程师中,一位提到了界面开发中一些重复的工作,需要将交互设计师的设计加以实现,而很多界面实现有许多重复的编码工作,仅需对界面的布局及控件的位置等进行修改.使用渲染工具,可以利用 EUID 的描述生成原型界面,并允许界面开发工程师在此基础上进行二次开发,减少了界面开发工程师的工作,使其能够将更多的精力投入到界面开发过程中更加细节的部分.

与传统用户界面开发或产品开发流程相比,基于 E-UIDL 的开发流程允许各种不同的角色在同一个平台上进行工作.辅助工具的输出物都为遵循 E-UIDL 的描述,因此各个角色之间可以更加流畅地进行沟通.通过工具的推荐,一方面能够使产品经理在产品被设计出之前即对产品的用户界面有一个总体的把握;另一方面,能够给

交互设计师推荐不同的界面实现,使设计师能够在一种较为合理的推荐界面的基础上作进一步的设计,减少了最初设计的工作量.对界面开发工程师而言,通过生成工具及渲染工具的辅助,可以允许它们在一个已经实现的原型界面的基础上进行开发,减少了初期界面开发的工作量,也降低了交互设计师与界面开发工程师之间的沟通及转换成本.各种基于 E-UIDL 的辅助工具在下一节中将作详细介绍.

4 E-UIDL 辅助开发工具

在上节中,我们给出了基于 E-UIDL 的开发流程.在整个开发流程中的各个部分,都需要相应的工具以辅助不同的角色完成当前的工作.因此,为辅助基于 E-UIDL 的用户界面的开发,给出了相应的基于 E-UIDL 的工具.辅助工具的系统架构如图 3 所示.在图中,下半部分是支持各种工具运行的平台及第三方库支持,上半部分是工具的主体.整个工具的主体部分由设计工具和生成工具两部分组成.

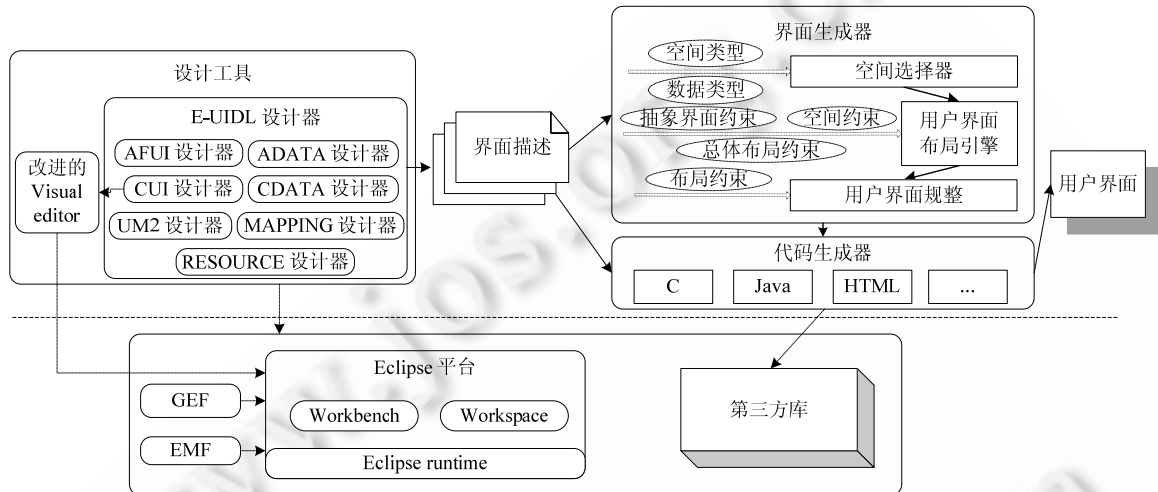


Fig.3 Architecture of tools

图 3 辅助工具体系结构

设计工具基于 Eclipse 平台,该平台是一款基于插件的开源开发平台.我们以插件的形式实现设计工具,可以充分利用 Eclipse 平台本身的多种扩展点.界面设计工具使用了 UIProject 的概念,在一个 UIProject 中,包括了 AFUI, CUI 等多个模块的设计.在架构上,每个模块的设计工具相对独立,针对其设计的模块类型进行特殊处理,并利用了第三方的插件.其中,各个设计工具均使用了视图(view)、属性(property)、编辑器(editor)等扩展点.CUI 由于需要实现界面的布局、控件的详细设计等功能,利用了基于 Visual Editor 的一个改进. Visual Editor 是 Eclipse 平台上一款所见即所得的界面设计的工具,该工具能够实现基于 Swing, AWT 以及 SWT 的界面,并能支持从界面到代码的转换.我们在其源码的基础上进行二次开发,取消了对 Java 项目的约束,并且实现了从界面到 E-UIDL 的 CUI 描述的主要实现,使得 CUI 可以设计 Visual Editor 支持的各种界面布局及控件.

在设计时,首先创建一个新的 UIProject 项目或指定一个已经存在的 UIProject 项目,项目的创建与管理通常由项目经理或需求分析人员使用.在 UIProject 的基础上,选择要设计的 E-UIDL 的各个模块.在创建某一特定的模块后,可以进入各个不同模块对应的设计界面,对 E-UIDL 的各个模块进行设计,各个不同模块的设计界面有相同的组成部分,它们主要由工程导航视图、主编辑区、组件面板、属性编辑视图组成.其中,工程导航视图用于在一个 UIProject 各个不同模块之间加以切换,帮助界面开发流程中的各个角色快速定位到感兴趣的模块;主编辑区为各个模块不同的编辑区域,实现所见即所得的设计;在主编辑区的设计过程中,可以通过将组件面板中定义各个组件拖入主编辑区来完成组件的创建;对于一些高级特性,如 AFUI 的 group 的权值及排序选项以及 CUI 中控件的颜色等选项,则可以通过属性编辑视图来完成.

界面生成工具^[23-25]可以通过不同模块之间的组合对界面进行不同抽象层次上的描述,既可以通过 AFUI+ADATA, AFUI+CDATA, AFUI, ADATA 或 CDATA 对界面进行相对抽象的描述,并且可以这种描述为基础,利用界面生成器来生成原型界面;也可以通过 CUI+CDATA 或 CUI+ADATA 对界面进行相对具体的描述,这种界面描述能够具体描述控件级别的展示,可以不经界面自动生成而直接生成依赖于目标语言的用户界面代码。

界面生成工具包括两个模块,分别是界面生成器及代码生成器。其中,

- 界面生成器的主要目的是利用对界面的抽象描述,生成统一的可以被代码化的具体描述。该模块是实现界面自动生成的核心部分,它包括控件选择器、布局引擎以及界面规整器 3 个部分。它们对界面描述的不同模块的内容加以利用,并且将前一部分的输出作为后一部分的输入。
 - 控件选择器利用界面描述中界面模块及数据模块的定义,并对照移动界面中控件的特性,确定某一具体的功能或功能组所使用的控件类型。例如,如果在界面描述使用了 AFUI 及 CDATA,那么经过控件选择器后,Label 会被作为候选,成为界面布局器的输入;如果界面仅使用了 CDATA 作为数据模型的描述,那么通过控件选择器后,Switch,CheckBox,RadioButton 将会被作为候选,成为界面布局器的输入;
 - 布局引擎是在确定了控件的候选之后对界面进行布局,该部分以控件选择器的控件选择结果作为输入,并且利用了以下 3 种规则作为约束:第一,界面描述语言的 AFUI(如果存在)描述的各功能组的排序规则以及权重规则,这些规则在 AFUI 的描述中以 group 或 atomgroup 的属性存在;第二,各个控件在不同移动操作系统上的尺寸、对齐等约束,这些约束主要从各个移动平台提供的界面指导准则中获得,如在 Android 平台上,按钮(button)的最小尺寸为 7mm,最优尺寸为 9mm,以 dp 为单位,为 48dp;第三,总体的布局约束,这主要是指应用程序的总体约束,如总的排版规则、目标用户类型等,这些信息除了从界面描述获得外,还从目标设备的软、硬件条件获取部分信息。将以上规则作为约束,并使用增加了约束的布局算法对界面进行布局,形成初步的界面形态。在下面的章节中,我们会介绍一种融合了自底向上与自顶向下方法的自动布局算法,该算法能够充分利用各种规则,实现对界面的布局;
 - 在形成了初步的界面形态后,我们利用一些规整化的规则对界面作进一步的优化,这些优化主要集中在对布局的微调,这些微调主要参考文献[26]对界面布局准则的描述,如物理特性、组合特性、联合分解特性、顺序特性及排版特性等。该部分作为界面生成部分的输出,为代码生成器提供输入;
- 代码生成器部分的主要功能是利用界面生成器生成的界面,或者利用 CUI+CDATA 或 CUI+ADATA 的描述,生成特定语言的界面。

5 实例应用

5.1 移动设备上的注册程序

我们使用基于 E-UIDL 的设计流程,并利用辅助工具,以一个移动设备上的注册程序为例,说明通过基于 E-UIDL 的辅助工具来支持用户界面开发的过程。

传统的开发流程需要产品经理对功能点进行定义,并通过 Word,PowerPoint 等工具对功能点及需求进行描述,交付给其他角色。交互设计师根据该文档,对界面布局及交互方式等进行详细定义。在此步骤中,不同的交互设计师使用不同的工具,并利用工具的输出内容与各个角色进行沟通。在交互设计定稿并交付到视觉设计师及界面开发工程师之后,视觉设计师需要从该文档中提取出需要进行视觉设计的详细内容,然后进行视觉设计;界面开发工程师需要建立项目的源代码工程,开始界面的开发。在此过程中,出现了本文提到的沟通上的问题及各个角色之间交付文档的转换问题,因而降低了开发的效率。

使用基于 E-UIDL 的用户界面开发方法,项目开始时,由产品经理或项目负责人创建一个 UIProject,并将该项目共享,以允许开发过程中的各个角色能够访问到该项目。产品经理或需求分析人员确定产品的主要功能点

及需求,该角色在 UIProject 下创建 AFUI 模块及 DATA 模块,其中,创建各个模块的导航界面如图 4 所示.从图中可以看到,当前的工具可以支持创建 E-UIDL 的各个模块.AFUI 的编辑界面如图 5 所示,使用前文的规则对 AFUI 模块进行所见即所得的编辑.由图 5 可知:根据 AFUI 描述规则,最大化地对功能组进行细分.在相关模块编辑完成后,可以通过界面自动生成工具生成推荐的界面,使得产品经理或需求分析人员能够在产品开发前对整个产品的界面有直观的认识.图 6 是在指定了 AFUI 及 CDATA 模块,Android 2.3 平台基础上,屏幕类型为 WVGA 的环境下生成的界面.该界面的生成利用了类似 SmartTemplate^[9]的技术,核心算法使用了一种传统的布局算法,迭代次数为 23 次.

在产品经理或需求分析人员定义了产品的主要功能点及需求后,交付给交互设计师,交互设计师使用工具进行界面及交互的设计,在 E-UIDL 环境下,交互设计师需要对 CUI,CDATA,MAPPING 等模块进行设计,其中,图 7 所示为交互设计师设计 CUI 模块的界面.在交互设计师完成相关设计工作后,也可以通过生成工具来浏览所生成的界面的效果.视觉设计师收到交互设计师工作后的 UIProject,进行视觉设计,由于大部分的视觉设计相关的属性如控件布局、控件尺寸等因素都在 E-UIDL 的描述中,因此,视觉设计师可以直接利用第三方的工具对界面需要的各种资源进行创作.界面开发工程师接收到交互设计师工作后的 UIProject,进行界面开发,其可以利用工具生成界面的基线版本,并在此版本上进行二次开发,以降低开发的工作量.

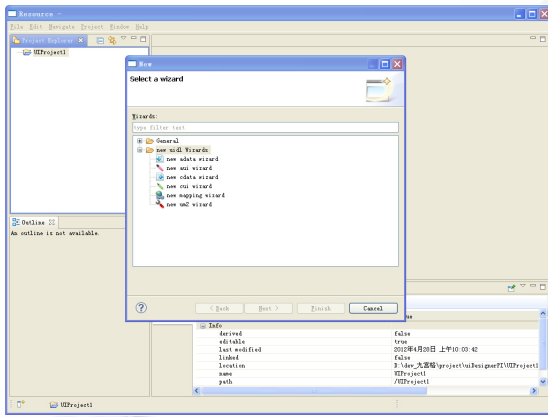


Fig.4 UIProject wizard of design tool
图 4 设计工具的 UIProject 界面

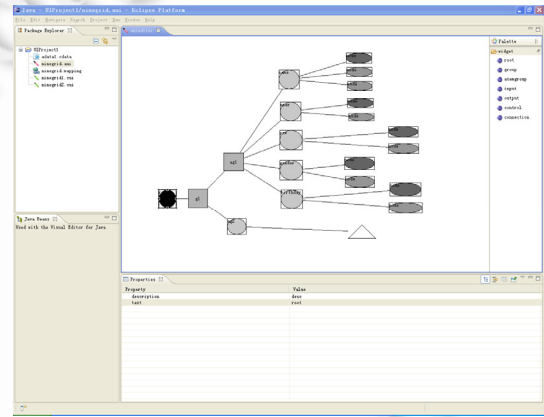


Fig.5 AUI design interface of design tool
图 5 设计工具的 AUI 设计界面

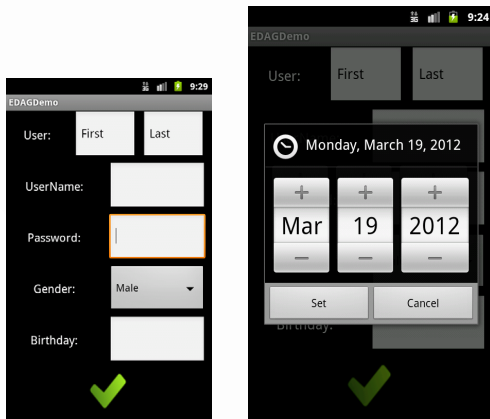


Fig.6 UI generated by tools
图 6 生成工具生成的推荐界面

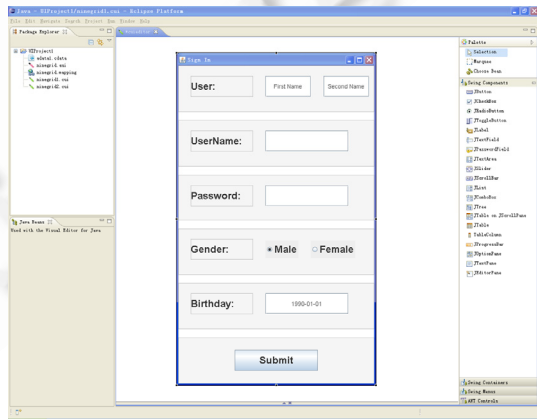


Fig.7 CUI design interface of design tool
图 7 设计工具的 CUI 设计界面

从该实例可以看出:使用基于 E-UIDL 的开发方法,可以在一定程度上降低各个角色之间因使用工具的不同及工具输出之间的差异导致的沟通及转换问题;并且通过自动界面的生成,不仅降低了界面开发工程师的工作量,还使得产品经理能够在产品开发初期即可对产品的界面有直观的认识.

5.2 基于E-UIDL开发方法的其他实例

为了进一步验证基于 E-UIDL 的开发方法的实用性,在 E-UIDL 及一系列的开发工具及开发方法的支持下,开发了 3 个应用系统,其中包括九宫格导航系统、手写邮件系统及日历记录系统.我们使用 E-UIDL 设计工具对系统进行不同层次的描述;然后,使用生成工具生成系统不同平台下的初始界面;最后,通过简单的二次开发以及后台逻辑功能模块的添加,完成整个系统的开发.

图 8 是基于 E-UIDL 的界面导航系统的界面.3 个界面使用了同一个 E-UIDL 的 AFUI 模块的描述,使用了不同的资源描述.其中,左上角的界面是生成的基于 PBOP(pen based operating platform)^[22]平台的代码运行后的界面,该界面在 Windows 及 Android 平台上均可运行,且显示效果相同;左下角及右侧的界面是生成的 Android 平台的代码运行后的界面,该界面能够实现自动的屏幕布局,且在重新布局后能够保持界面的合理性.在任意 Icon 上点击,都可以进入到相应的应用中去,其中,PBOP 代码导航应用使用了配置文件的方式来实现 Android 平台下运行时的导航,而 Android 代码使用了 Intent 的方式来实现运行时的导航.

图 9 是手写邮件系统的界面呈现,其中,

上半部分的两张截图是生成的 PBOP 平台的代码运行后的界面;

下半部分两张截图是生成的 Android 平台的代码运行后的界面,其中,Android 平台的代码运行后的界面在前文章节中已有所阐述.

在使用 E-UIDL 相关工具生成界面后,实现了手写邮件系统的内部逻辑,其中,较为核心的收发邮件的功能使用 JavaMail API 来实现.

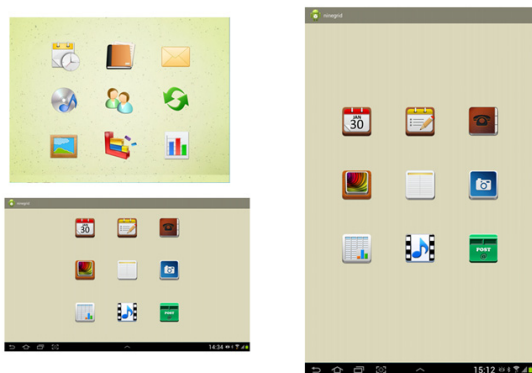


Fig.8 User interfaces of ninegrid navigation system
图 8 九宫格界面导航系统界面

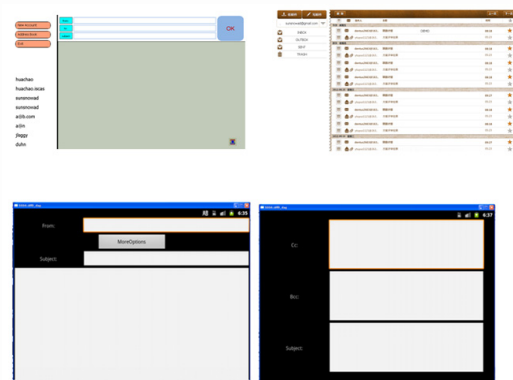


Fig.9 User interfaces of pen email system
图 9 手写邮件系统界面

图 10 是日历记录系统的界面截图.在人们办公的过程中,日历的作用越发重要,特别是在移动计算环境下,对日历及记录有了更多的要求.比如,如何协调待办事件、如何调整事件优先级等.在此背景下,我们基于 E-UIDL 及相关工具和方法,开发了日历记录系统.

在图 10 中:

左上角是生成的 PBOP 平台代码运行后的界面;

左下角及右侧的界面是生成的 Android 平台代码后运行显示的界面,在此界面中我们能够发现:生成工具可以根据生成的目标平台来选择合适的控件及布局方法,并给出相对较优的解.

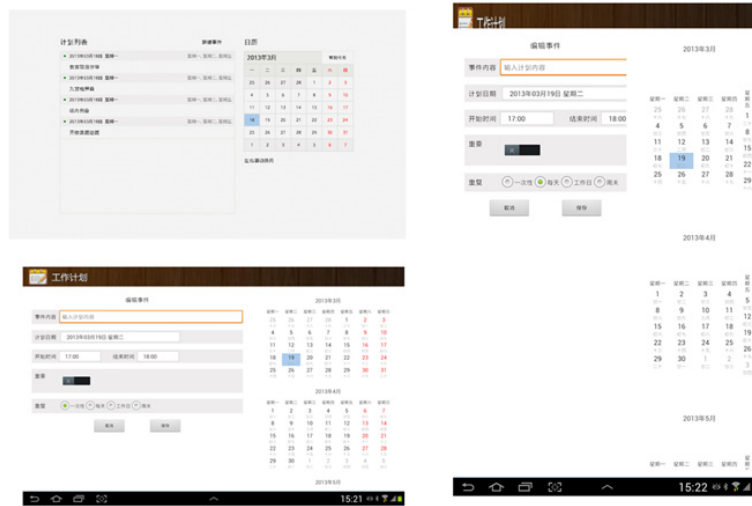


Fig.10 User interfaces of calendar record system

图 10 日历记录系统界面

6 总结及下一步工作

基于当前用户界面开发流程中由于角色的细分导致的各个角色之间沟通成本增加以及工作效率降低的问题,本文提出了一种基于 E-UIDL 用户界面描述语言的解决方案,介绍了一种基于该描述语言的开发方法.首先介绍了 E-UIDL 的组成部分及各部分之间的相互关系,详细介绍了基于 E-UIDL 的用户界面开发流程.在此基础上,我们介绍了基于 E-UIDL 的辅助工具,并通过一个注册程序的实例说明,如何使用基于 E-UIDL 的工具来支持整个界面开发的流程.该工作说明,通过基于用户界面描述语言的开发方法能够提高开发效率,但该项工作还存在一些不足:

第一,对 E-UIDL 及工具的可用性及可学习性的评估不足.

评估主要表现在两个方面:一方面是 E-UIDL 本身的可用性及可学习性的问题,另一方面是基于 E-UIDL 的工具的可用性及可学习性的问题.我们将参考文献[9]中对界面描述语言的评估方法,对 E-UIDL 及工具的可用性及可学习性进行评估.

第二,当前的设计工具存在一些不足.

CUI 设计工具基于改进的 Visual Editor 插件,可以解析 E-UIDL 的 CUI 模块,并生成对应的 CUI 描述,但是并不包含 Android 等移动设备的组件,这导致了 E-UIDL 对 Android 平台界面的支持不能反映在设计工具上,因此需要对 Visual Editor 的组件支持进行修改,使其能够支持更多的移动设备的组件.

致谢 在此,对所有参与审稿的老师及编辑致以衷心的感谢.

References:

[1] Myers BA. User-Interface tools: Introduction and survey. *Software*, 1989,6(1):15-23. [doi: 10.1109/52.16898]

[2] Myers BA, Rosson MB. Survey on user interface programming. In: *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. Monterey: ACM Press, 1992. 195-202. [doi: 10.1145/142750.142789]

[3] Szekely P, Luo P, Neches R. Beyond interface builders: Model-Based interface tools. In: *Proc. of the INTERACT'93 and CHI'93 Conf. on Human Factors in Computing Systems*. Amsterdam: ACM Press, 1993. 383-390. [doi: 10.1145/169059.169305]

[4] Schlungbaum E. Model-Based user interface software tools-current state of declarative models. Technical Report, GIT-GVU-96-30, GA: Georgia Institute of Technology, 1996.

- [5] Josefina GG. A theoretical survey of user interface description languages: Preliminary results. In: Juan MGC, Jean V, Jaime MA, eds. Proc. of the LA-WEB 2009 on Web Congress. 2009. 36–43. [doi: 10.1109/LA-WEB.2009.40]
- [6] 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]
- [7] Shaer O, Jacob RJK. A specification paradigm for the design and implementation of tangible user interfaces. ACM Trans. on Computer Human Interaction, 2009,16(4):1–39. [doi: 10.1145/1614390.1614395]
- [8] Navarre D, Palanque P, Ladry JF, Barboni E. ICOs: A model-based user interface description technique dedicated to interactive systems addressing usability, reliability and scalability. ACM Trans. on Computer Human Interaction, 2009,16(4):1–56. [doi: 10.1145/1614390.1614393]
- [9] 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]
- [10] Nichols J, Myers BA, Rothrock B. UNIFORM: Automatically generating consistent remote control user interfaces. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. Montreal: ACM Press, 2006. 611–620. [doi: 10.1145/1124772.1124865]
- [11] Nichols J, Rothrock B, Chau DH, Myers BA. Huddle: Automatically generating interfaces for systems of multiple connected appliances. In: Proc. of the 19th Annual ACM Symp. on User Interface Software and Technology. Montreux: ACM Press, 2006. 279–288. [doi: 10.1145/1166253.1166298]
- [12] Limbourg Q, Vanderdonck J, Michotte B, Bouillon L, López-Jaquero V. Usixml: A language supporting multi-path development of user interfaces. In: Proc. of the Engineering Human Computer Interaction and Interactive Systems. 2005. 200–220. [doi: 10.1007/11431879_12]
- [13] Limbourg Q, Vanderdonck J, Michotte B, Bouillon L, Florins M, Trevisan D. Usixml: A user interface description language for context-sensitive user interfaces. In: Costabile MF, ed. Proc. of the Citeseer. New York: ACM Press, 2004. 55–62.
- [14] Vanderdonck J, Limbourg Q, Michotte B, Bouillon L, Trevisan D, Florins M. UsiXML: A user interface description language for specifying multimodal user interfaces. In: Proc. of the W3C Workshop on Multimodal Interaction. 2004. 35–42. <http://www.w3.org/2004/02/mmi-workshop-cfp.html>
- [15] Limbourg Q, Vanderdonck J, Michotte B, Bouillon L, Florins M. Usixml: A user interface description language supporting multiple levels of independence. In: Matera M, Comai S, eds. Proc. of the Engineering Advanced Web Applications. Munich: Rinton Press, 2004. 325–338.
- [16] Dewan P. Increasing the automation of a toolkit without reducing its abstraction and user-interface flexibility. In: Proc. of the 2nd ACM SIGCHI Symp. on Engineering Interactive Computing Systems. Berlin: ACM Press, 2010. 47–56. [doi: 10.1145/1822018.1822028]
- [17] Raneburger D, Popp R, Kaindl H, Falb J, Ertl D. Automated generation of device-specific WIMP UIs: Weaving of structural and behavioral models. In: Proc. of the 3rd ACM SIGCHI Symp. on Engineering Interactive Computing Systems. Pisa: ACM Press, 2011. 41–46. [doi: 10.1145/1996461.1996492]
- [18] Paterno F, Santoro C, Spano LD. MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. ACM Trans. on Computer Human Interaction, 2009,16(4):1–30. [doi: 10.1145/1614390.1614394]
- [19] Du Y, Deng CZ, Tian F, Ren L, Dai GZ. User interface description language. Ruan Jian Xue Bao/Journal of Software, 2013,24(5): 1127–1142 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4321.htm> [doi: 10.3724/SP.J.1001.2013.04321]
- [20] 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>
- [21] Du Y, Tian F, Ma CX, Dai GZ, Wang HA. A user interface generation framework based on multi-scale description method. Chinese Journal of Computers, 2013,36(11):2179–2190 (in Chinese with English abstract).
- [22] Du Y, Ma CX, Teng DX, Dai GZ. CONCEPT-SKETCH: A tool for cooperative visual analytics. Int'l Journal of Advanced Intelligence (IJAI), 2011,3(1):95–113.
- [23] Tran V. UI generation from task, domain and user models: The DB-USE approach. In: Proc. of the 2010 ACM Sigchi Symp. on Engineering Interactive Computing Systems (EICS 2010). 2010. 353–356. [doi: 10.1145/1822018.1822079]

- [24] Wu H, Hua QY, Chang YS, Zhu HY, Yang JF. Lightweight multi-device user interface description language MDUIDL. Computer Engineering and Applications, 2011,47(35):14-21 (in Chinese with English abstract).
- [25] Vanderdonckt J, Gillo X. Visual techniques for traditional and multimedia layouts. In: Proc. of the Workshop on Advanced Visual Interfaces. Bari: ACM Press, 1994. 95-104. [doi: 10.1145/192309.192334]
- [26] Yap PS, Hosking J, Grundy J. Automatic diagram layout support for the Marama meta-toolset. In: Proc. of the 2011 IEEE Symp. on Visual Languages and Human-Centric Computing (VL/HCC). 2011. 61-64. [doi: 10.1109/VLHCC.2011.6070379]

附中文参考文献:

- [19] 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> [doi: 10.3724/SP.J.1001.2013.04321]
- [20] 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>
- [21] 杜一,田丰,马翠霞,戴国忠,王宏安.基于多尺度描述方法的移动用户界面生成框架.计算机学报,2013,36(11):2179-2190.
- [24] 吴昊,华庆一,常言说,朱海阳,杨建峰.一个轻量级多设备用户界面描述语言 MDUIDL.计算机工程与应用,2011,47(35):14-21.



杜一(1988-),男,山东聊城人,博士,助理研究员,CCF 会员,主要研究领域为人机交互技术,笔式用户界面.



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.



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.