

自适应笔式用户界面开发的概念框架*

徐礼爽⁺, 周明骏, 邓昌智, 田 丰, 刘媛媛, 戴国忠

(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)

A Conceptual Framework for Developing Adaptive Pen-Based User Interface

XU Li-Shuang⁺, ZHOU Ming-Jun, DENG Chang-Zhi, TIAN Feng, LIU Yuan-Yuan, DAI Guo-Zhong

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

+ Corresponding author: E-mail: xulishuang@mails.gucas.ac.cn, lishuang03@ios.cn

Xu LS, Zhou MJ, Deng CZ, Tian F, Liu YY, Dai GZ. A conceptual framework for developing adaptive pen-based user interface. *Journal of Software*, 2008,19(10):2681-2693. <http://www.jos.org.cn/1000-9825/19/2681.htm>

Abstract: This paper presents CFAPUI (a conceptual framework for developing adaptive pen-based user interface), a conceptual framework for developing adaptive pen-based user interface. CFAPUI proposes an architecture for adaptive pen-based applications, a set of guidelines to assist the design process of adaptive pen-based applications. To demonstrate CFAPUI's validity, the development process of an adaptive pen-based form application is illustrated.

Key words: adaptive user interface; pen-based user interface

摘 要: 提出了一个指导自适应笔式用户界面开发的概念框架 CFAPUI(a conceptual framework for developing adaptive pen-based user interface).该框架详细描述了自适应笔式用户界面的软件体系结构,阐述了自适应笔式用户界面的软件体系结构中的各组成模块及其相互间的关系.同时,在体系结构的基础上提出自适应笔式用户界面开发时的具体过程步骤和方法,并通过一个笔式表单应用来说明该开发框架在指导自适应笔式用户界面开发中的具体应用.

关键词: 自适应用户界面;笔式用户界面

中图法分类号: TP391 **文献标识码:** A

笔式用户界面基于传统的纸笔隐喻,因其自由勾画的输入方式,在思维捕捉和记录、观点研讨和交流、概念设计、文件审阅等众多方面有着其他界面无可比拟的优势,使得生活的诸多领域因数字化而更加便捷.目前已有一些笔式用户界面系统应用在教学、体育、办公、军事等众多领域^[1-4].由于其自然的交互风格,笔式用户界面已成为 Post-WIMP 用户界面的一种重要形态,从 20 世纪 90 年代起,以笔式用户界面等为代表的 Post-WIMP 界面迅速成为了国内外研究的热点^[5].

笔式用户界面摆脱了传统的鼠标、键盘输入方式,手写勾画的自然交互风格使得计算面向更广的人群.但同时,这种自然交互风格所固有的模糊和非精确性^[4],也加大了笔式用户界面设计和开发的难度.Fuller 指出,笔

* Supported by the National Natural Science Foundation of China under Grant Nos.60503054, U0735004 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z158 (国家高技术研究发展计划(863))

Received 2007-08-07; Accepted 2008-01-10

的手写自由输入特性很自然地让用户去期盼系统能够理解其交互意图,适应用户^[6]. 普适计算作为 21 世纪的计算模式^[7], 计算环境将遍及生活的各个角落, 而不再仅仅局限于传统的桌面, 因此方便、快捷的笔交互将成为普适计算环境下的重要交互方式. 在新的计算模式下, 笔交互设备和使用环境日益多样化, 不同笔设备的处理能力、屏幕大小、网络能力各异, 同时, 对于数字奥运、移动办公等重要领域而言, 由于用户对象、应用环境等方面的特点, 要求系统能够根据用户、环境等实时状态, 自适应地改变界面的显示方式和交互行为. 因此, 相对于传统精确交互的图形用户界面而言, 用户的个性化、设备的多样性以及多环境下的应用场景所带来的交互复杂性和多样化问题使得模糊和非精确特性在笔式用户界面中显得尤为突出. 而自适应是解决交互复杂性和多样化问题的很好的方法^[8].

在 ACM Intelligent User Interface 2006 年会上, Duarte 等人^[9]提出了用以指导开发自适应多通道用户界面的概念框架, 并用行为矩阵来表示多通道用户界面的自适应规则. 该概念框架描述了多通道用户界面的软件体系结构, 同时, 在此体系结构上提出了自适应多通道用户界面的开发流程和方法, 在该概念框架的指导下开发者可以方便、快捷地建立自适应多通道用户界面. 对于笔式用户界面而言, 虽然目前已有不少相关的自适应性研究方面的课题^[6, 10-12], 但是这些研究主要面向的是特定应用场景下笔式用户界面的复杂性和多样化的问题, 而并没有提出针对自适应笔式用户界面设计和开发的通用软件体系结构及软件开发方法.

本文针对笔式用户界面复杂性和多样化的问题, 提出了一个指导自适应笔式用户界面开发的概念框架 CFAPUI (a conceptual framework for developing adaptive pen-based user interface), 该框架主要由自适应笔式用户界面的软件体系结构、软件开发方法两大部分组成. 软件体系结构描述了自适应笔式用户界面的模块组成及各模块之间的关系; 软件开发方法在软件体系结构和开发实践的基础上总结出设计和开发自适应笔式用户界面的具体步骤及方法. 同时, 本文面向数字奥运领域, 利用 CFAPUI 设计和开发出面向国家运动队的笔式训练系统, 可以方便、灵活地应用在国家队的日常训练工作中. 目前, 该系统已经成功地应用于国家跳水队、国家羽毛球队和国家水球队.

本文第 1 节阐述相关工作. 第 2 节详细介绍框架的软件体系结构. 第 3 节介绍框架的软件开发方法. 第 4 节以具体的应用开发实例来介绍该概念框架的应用. 最后总结全文.

1 相关研究

笔式用户界面已经成为人机交互领域研究的热点, ACM CHI, UIST, IUI 等相关国际会议都将笔式用户界面作为重要研究内容, AAAI 也在 2004 年秋季专门召开了题为“Making Pen-Based Interaction Intelligent and Natural”的研讨会. 许多相关的研究集中在特定领域的笔迹结构理解和内容识别上, 如面向笔迹流程图理解^[13], 笔迹的隐式结构题解^[14], 数学^[15]、化学^[16]、音乐^[17]领域的理解和识别等, 针对手势的识别^[18]和手势的易用性^[19, 20]等方面. 从以上的这些研究可以看出, 笔式用户界面因其模糊性、非确定性的特点, 使其相对于传统的图形用户界面而言, 在智能理解和处理方面需要开展许多的研究工作. 另一方面, 笔交互是普适计算环境下的重要交互方式, 电子白板^[14]、手持设备^[21]中都使用到了笔交互设备, 笔交互设备及其应用场景的多样化给笔式用户界面的研究带来新的课题. 因此, 用统一而通用的框架去解决笔式用户界面的上述问题成为笔式用户界面研究的关键之一.

Dieterich 等人^[6]提出自适应用户界面是解决目前人机交互日趋复杂和多样性的很好的解决方法, 它的目的是根据用户的个性化需求和应用的环境及条件来裁剪定制系统的显示方式和交互行为, 从而提高用户界面的可用性. Fuller^[6]提出, 笔的自由输入特性自然地让用户去期盼系统的适应能力, 并且用机器学习的算法, 通过在线的自适应途径研究和实现了不同用户差异下的手写字符识别问题. Iwayama 等人^[10]引入自适应的上下文处理机制来提高在线手写字符的识别率. 孙正兴等人^[12]提出用户适应性是在线手绘草图识别的核心和关键课题, 并提出采用支撑向量机主动式增量学习和动态用户建模技术进行笔划和复杂图形的识别. 以上相关研究采用了自适应的方法解决了笔式用户界面中的一些特定应用场景下的模糊和非精确的交互和信息处理问题, 但并没有针对笔式用户界面提出通用的自适应的解决方法. Cao 等人^[11]提出了一个在线自适应的手势界面框架, 用

交互历史和识别器相结合的方式来提高手势的识别率,通过用户纠错机制在线、动态地更新手势训练集的样本来自适应不同用户的手势输入习惯,并用系统全面的实验评估方法验证了该框架下的手势界面相对于未使用在线自适应的手势界面的诸多优势.但是,该框架针对的是手势的在线自适应,不能通用到解决笔式用户界面的其他模糊和非精确问题.Duarte 等人^[9]提出了用以指导开发多通道自适应用户界面的概念框架,并用行为矩阵来表示多通道用户界面的自适应规则.然而他主要面向的是多个通道输入输出的多通道用户界面的通用概念框架,没有针对笔界面的自适应问题进行专门的描述,行为矩阵的自适应规则和自适应多通道用户界面的开发方法也不能方便地直接应用在自适应笔式用户界面中.

从上述相关研究中可以看出,虽然目前已有不少关于笔式用户界面的自适应性研究方面的课题,但都是用来解决具体的应用点上的自适应问题,并没有一个针对自适应笔式用户界面的通用框架和方法来指导自适应笔式用户界面的设计和开发.在此背景下,我们提出了面向自适应笔式用户界面设计和实现的开发框架,该框架描述了笔式自适应用户界面的软件体系结构,并在此体系结构的基础上给出了自适应笔式用户界面的软件开发方法.然后在数字奥运领域,以面向国家运动队的笔式训练系统示例了该框架在自适应笔式用户界面设计和开发中的应用.

2 自适应笔式用户界面软件体系结构

根据相关研究总结和开发实践,本文提出自适应笔式用户界面的层次式软件体系结构,如图 1 所示,该体系结构分为 3 层:底层为模型层①,中间层为自适应引擎层②,上层为平台层③.模型层由用户模型、环境模型、设备模型和任务模型组成,模型层是描述自适应所需的信息集合;自适应引擎层由自适应核心模块、交互上下文组成,引擎层是实现笔式用户界面自适应功能的算法实现和调度部分.平台层由 Ink 算法库、笔手势库和笔组件库组成,平台层是用来快速建立笔式用户界面的开发工具箱.层与层之间以通信相互关联:平台层的消息通过事件引擎形成交互历史,事件引擎和交互历史为引擎层提供了交互上下文,自适应核心模块通过交互上下文和模型层提供的信息,采取一定的自适应策略,为平台层提供自适应的功能.总体上看,引擎层通过与上层的平台层和底层的模型层以动态在线迭代的方式通信,从而实现了笔式用户界面的自适应功能.

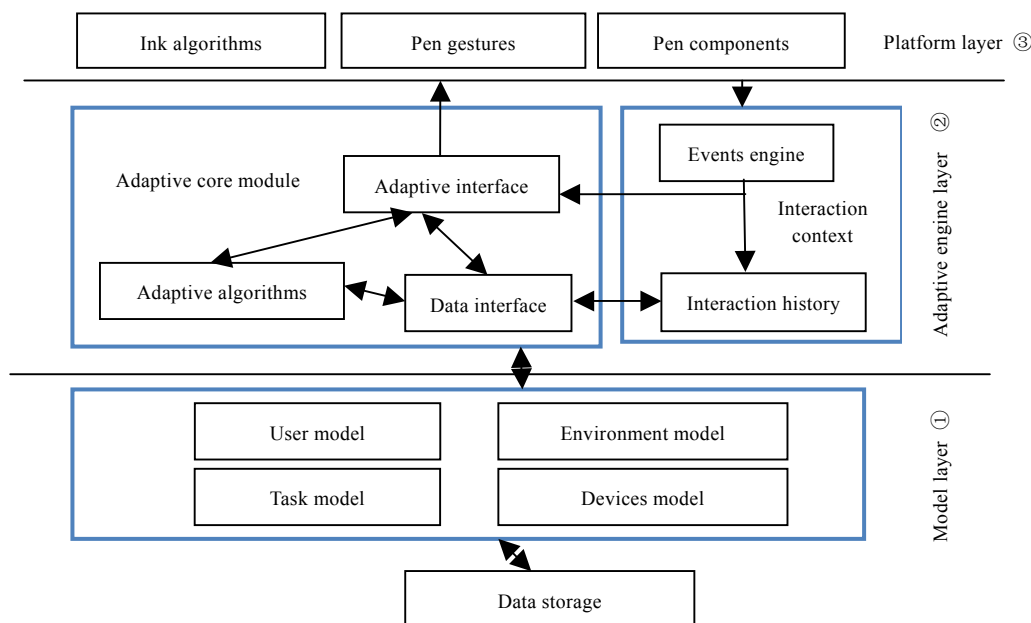


Fig.1 Software architecture of adaptive pen-based user interface

图 1 自适应笔式用户界面软件体系结构

可以看出,在线运行时,上层的模块需要下层的支持,同时该体系结构的各层之间相对独立,耦合度小,很好地体现了重用机制.下面对各层进行具体介绍.

2.1 模型层

模型是实现自适应所需信息在计算机的内部表示,模型层包括设备模型、环境模型、用户模型和任务模型.这些模型是对笔式用户界面运行时不同方面信息的建模.

设备模型是对设备特征的建模^[22],如设备的处理能力、笔设备所能提供的笔交互信息(如位置、倾角、压力);我们用元组 $d=\langle W,E,C_d\rangle$ 来描述特定设备的信息,其中 W 表示在特定设备下的笔式用户界面的界面组件集合, E 为特定设备下的各种类别识别引擎的集合, C_d 表示为与设备相关的约束集合. $D=\{d_1,d_2,\dots,d_n\}$ 描述了笔式界面中各种笔设备的信息,构成了完整的设备模型描述.

环境模型是将传感器感受到的环境的原始信息转换为方便系统应用的信息表示^[23],环境模型从传感器原始数据中提取出简单的语义信息,屏蔽系统不关心的数据,消除数据冗余^[24].

用户模型是计算机系统对当前的使用用户的知识表示^[24].在用户模型中,我们可以将用户的特征分为与特定应用独立和相关的两大类,与应用独立的特征有用户的偏好、心理、情感、运动技能、能力、认知和学习能力等,与应用相关的特征有用户具备的计算机技能,用户对现有系统和应用的知识掌握程度、目标、意图、期望等.有多种表示方式和技术来表示用户模型,如早期人们用人工智能中的知识表示技术,后来面向对象的新技术也被用户建模所采纳,为了能够方便地表达和处理用户的不确定信息,神经网络和贝叶斯的表示技术也逐渐得到应用.

任务模型是与应用领域相关的领域知识,为应用提供任务上下文信息.它描述了用户在使用过程中需完成的目标及完成目标所处的环境.我们可以根据任务树的结构将任务递归分解为子任务树来对任务进行建模.

以上模型的初始化和在线更新的方式也不尽相同,系统一般通过设备和环境传感器来提取和动态更新设备和环境模型.系统可以通过在线的方式动态地对当前的使用用户进行建模,如通过交互上下文,我们就可以提取出用户的使用模式(usage pattern),用户模型也可以通过传感器来获取,比如通过摄像头捕捉用户的表情来获取用户当前的情感因素.任务模型与具体的任务相关,应根据领域任务的特征来选择适合特定任务的表示方式.

以上这些模型是当前应用为了实现自适应所需要的信息表示,除了任务模型具有相对具体的应用固定外,其他模型的信息都是在线实时更新的,有些模型需要对相关信息作累积数据收集,这些信息在系统退出时需要保存到外部存储设备,在下次系统启动时再通过一定的策略载入到对应的模型中去.

2.2 自适应引擎层

自适应引擎层由自适应核心模块和交互上下文组成.自适应引擎层是界面实现自适应功能的关键和核心部分.它根据交互上下文和模型数据为界面提供自适应功能的算法实现和调度.

2.2.1 自适应核心模块

自适应核心模块由 3 个子模块组成(如图 1 所示):自适应调度接口、算法集和数据接口.自适应调度接口负责接收事件引擎中的事件,同时和数据接口进行通信来获取当前的状态,根据接收的事件和当前的状态进行自适应算法的调度,自适应接口还可以将必要的信息传给数据接口;算法集是一系列的自适应功能的实现函数,对应于笔式用户界面,自适应的算法集是组件库、Ink 算法库和手势库的自适应算法的总体集合,算法集中的算法同数据接口通信来确定算法执行时的状态参数;数据接口在与自适应核心模块内部的自适应调度接口和算法集进行通信的同时,还需要与交互历史和模型层中的模型信息进行通信,是提供模型信息、界面状态以及交互历史的通用接口.关于自适应的算法集的阐述,本文在第 2.2.3 节将专门描述一个通用的笔手势的自适应策略框架,在具体设计和开发过程中可以实现和扩展该框架,同时,Ink 算法库中的自适应算法也可以借鉴该框架的思想,由于具体组件的自适应策略因应用而有所不同,在此不作通用介绍,在第 4 节的实例中将有体现.

2.2.2 交互上下文

交互上下文由事件引擎和交互历史组成,事件引擎捕捉用户与界面交互而产生的事件,同时将捕捉到的消

息转发给自适应调度接口和交互历史.交互历史动态增量地对事件进行累积记录.交互历史本质上是对用户交互习惯的建模,可以通过数据接口和用户模型进行通信.在笔式用户界面中,我们可以将事件分为物理事件和逻辑事件.物理事件是笔交互设备所触发的消息事件,这些事件提供笔交互的实时信息(如位置、压力、倾角等).逻辑事件是用户连续输入后所产生的手势命令事件.这些事件提供连续交互后形成的交互逻辑信息.物理事件和逻辑事件将事件分为两个层次:词法层次和语法层次.为了更加方便地表达和记录,以及选择需要的事件层次进行建模,交互历史对事件进行记录时通常需要进行一些预处理.

交互式系统是一个运行的状态机^[23].我们用元组 $Op(t)=(Sr,E,Cmd,Sd)$ 来表示在某一时刻 t 系统对应着某一个状态 Sr ,并且在该状态下用户进行操作触发事件 E ,系统对事件进行响应执行相应的命令 Cmd 并切换到另一种状态 Sd .用户与系统交互的过程用元组以时序的方式记录保存下来,形成操作的偏序集合 $\langle S, < \rangle = \{Op(t) | t=t_1, t_2, \dots, t_n\}$.我们可以根据事件发生的时间密度来将交互历史聚类,从而产生偏序集 S 上的一个划分 $S = \{S_1, S_2, \dots, S_m\}$,每一个分块 $S_i (1 \leq i \leq m)$ 代表在其相应的时间内完成的系列操作,这些操作序列中蕴含着用户与系统的交互模式^[25],我们在子分块的基础上挖掘出交互历史库中的一些规律,从而发现用户与系统交互时的一些个性化的操作习惯.

2.2.3 笔手势自适应策略

笔手势是笔式用户界面的主要交互方式,也是笔式用户界面中通用的功能部分,在交互上下文的基础上,本文提出笔手势的自适应策略框架,用以支持用户个性化的手势输入.该框架由两部分组成:自适应手势识别框架和纠错机制.自适应手势识别框架如图 2 所示,我们通过交互历史对当前用户输入手势进行命令的预测,将预测和手势识别器识别结果相结合产生识别的命令,同时由于有些手势命令的参数与具体的手势位置相关,如选择命令,我们根据任务上下文和手势位置参数相结合计算出符合当前任务和用户意图的操作对象,从而在手势命令精确度提高的同时用领域知识提高了命令参数的精确度,形成符合用户意图的完整命令.图 3 说明了用纠错机制来支持用户手势的个性化输入,当用户输入的手势识别器未能识别,而又用户的意图输入时,用户可以通过纠错的方式让输入的手势与特定的手势命令加以关联,作为特定手势命令训练集的正样本;另一方面,当用户输入的手势被识别成非用户意图的手势命令时,用户可以通过同样的纠错机制,将其与正确的手势命令相关联,系统将其作为训练集的正样本,同时作为识别出来的错误命令手势训练集的负样本,通过上述纠错过程,系统能够快速学习到用户的个性化手势输入.

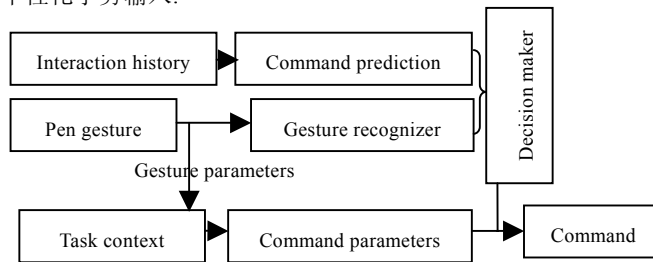


Fig.2 Framework of context based adaptive gesture recognition

图 2 基于上下文的自适应手势识别框架

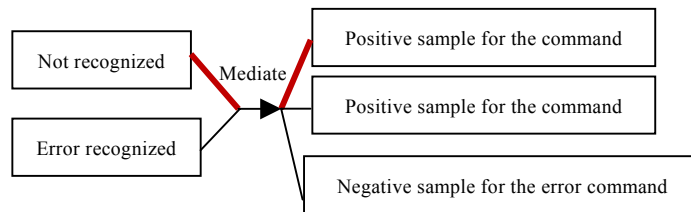


Fig.3 Mechanism of mediating to support the personified pen gestures input

图 3 纠错机制支持用户个性化的手势输入

2.3 平台层

平台层由 Ink 算法库、笔手势库和笔组件库组成.与非自适应笔式用户界面相比,自适应笔式用户界面的交互对话模型和界面显示不是固定不变的,为了实现笔式用户界面的自适应性,平台层应具有可适应功能.我们用一个描述元组 $Cd=(T,Af,Cc)$ 来描述每个构件的可适应功能.其中, T 为构件的类型(如能识别汉字字符型的手写输入组件,能识别数字型的手写输入组件,支持自由勾画的笔迹构件等), Af 为构件的自适应函数,对应于具体的自适应算法. Cc 为构件使用时的约束函数(如与设备相关的约束,自适应函数触发的条件等).自适应引擎通过数据接口确定当前的界面状态,从而在满足约束函数的条件下触发相应构件的自适应函数来实现自适应策略,实现自适应算法的调度.

笔手势库是当前界面的手势识别器所能识别的手势类型集合,在非自适应的笔式用户界面中,界面手势集是在设计和实现时静态固定的命令手势集合.但不同用户的手势输入方式各有不同,为了满足不同用户的手势输入习惯,用户界面应该能够学习用户的个性化手势输入习惯,我们可以开放手势识别器的相应接口,通过自适应引擎在线更新用户输入的手势训练样本,使得手势输入适应当前用户.笔手势库的自适应策略在第 2.2.3 节中有详细的说明.

Ink 算法库是一系列的 Ink 算法的集合,每一个 Ink 算法针对特定的笔迹计算的问题,比如笔迹流程图识别、数学表达式识别等,由于同一应用领域中不同用户会有不同的笔迹输入习惯,如数学表达式 a 不等于 b ,即可以写成 $a \neq b$,也可以写成 $a! = b$,算法识别库也可以通过动态的纠错机制或其他自适应方法来进行更新,自适应策略和途径与以上手势的自适应策略有相似之处.

3 软件开发方法

在基于上述自适应的软件体系结构的基础上,本节给出设计和开发自适应笔式用户界面的软件开发方法.

3.1 开发总体流程

自适应笔式用户界面软件开发的总体流程如图 4 所示,我们将开发过程分为需求分析(①)、设计和实现(②)及测试和评估(③)3 个阶段.

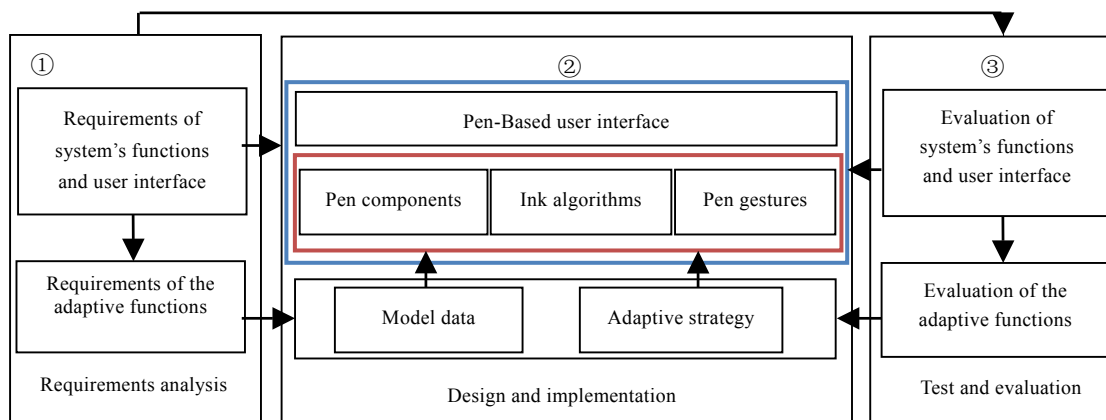


Fig.4 Software development method of adaptive pen-based user interface

图 4 自适应笔式用户界面软件开发方法

在需求分析阶段,需求分析人员在完成功能及界面需求分析的基础上进一步确定系统的自适应需求,规定在特定的界面状态下界面显示和交互行为的自适应规则和方式;在设计和实现阶段,设计和开发人员根据需求分析阶段的功能及界面的需求进行笔组件库、Ink 算法库和笔手势库的设计和实现,在设计和实现的同时,设计和开发人员应根据自适应的需求确定所需的模型数据和自适应策略,从而对笔组件库、Ink 算法库和笔手势库

的功能进行自适应的功能扩展;在测试和评估阶段,测试和评估人员根据需求分析得出相应的功能和界面评估以及自适应性评估的评估准则,根据它对设计和实现的结果进行评估,并将评估的结果返回到设计和实现过程中,从而实现迭代式的更新。

3.2 自适应设计和开发的具体过程指导

总体流程中的第 2 部分是软件的设计和实现部分,根据需求分析得出的结果,我们采用上述提出的层次式的自适应软件体系结构,由软件的功能及界面需求结果来确定平台层的笔交互组件和笔手势库,由自适应的需求来确定自适应引擎层和模型层中各个子模块的对应部分(如图 5 所示)。同时,自适应的需求确定上层平台层的组件库和手势库的可适应的接口。

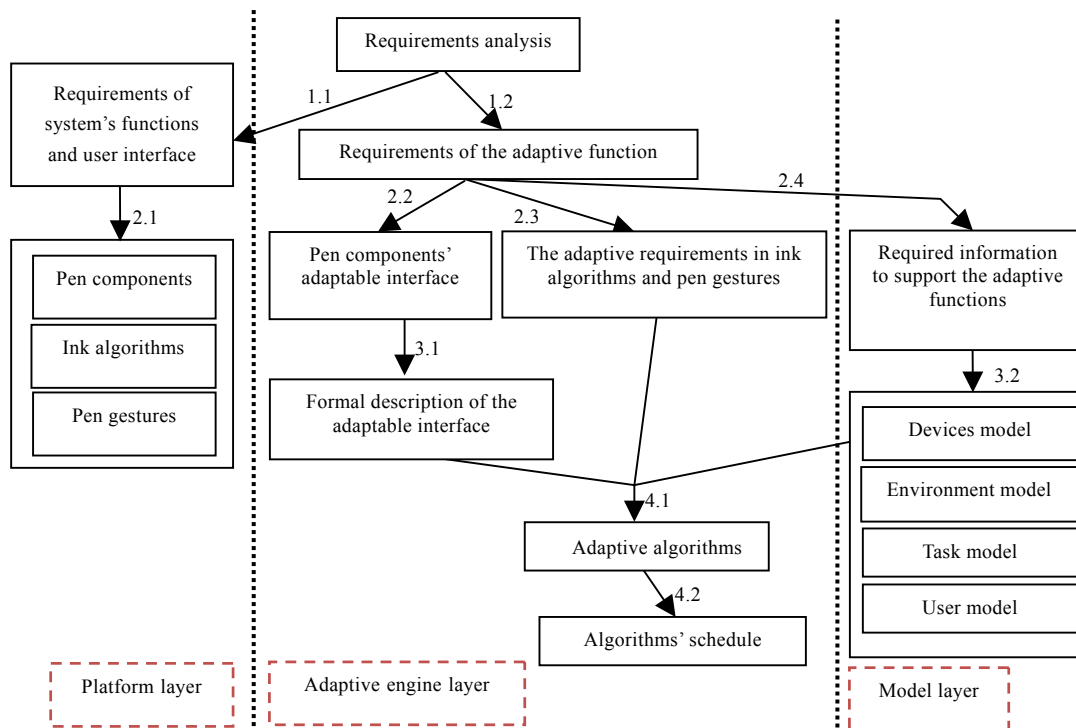


Fig.5 Design and development process of adaptive pen-based user interface

图 5 自适应笔式用户界面设计和开发详细流程

针对自适应笔式用户界面的设计和开发阶段,本节给出自适应部分的需求分析及设计和开发的详细流程,用以指导自适应笔式用户界面的设计和开发。

如图 5 所示,我们将设计和开发过程分为 4 个步骤:

第 1 步. 笔式用户界面需求分析.该步骤是从实际的应用需求出发确定系统的功能,我们将该步骤分为两个子步骤:

- 1.1. 确定应用功能及界面需求;
- 1.2. 确定自适应功能.

第 2 步. 笔式用户界面的自适应性分析.在确定了应用所需的的功能的基础上,该步骤从软件设计和实现的角度,对第 1 步的需求分析进行细化,我们将其又分成 4 个子步骤:

- 2.1. 根据第 1.1 步的应用功能及界面需求确定笔交互组件库、笔迹算法库和手势库;
- 2.2. 根据第 1.2 步的自适应功能确定笔交互组件库的可适应接口;
- 2.3. 根据第 1.2 步的自适应功能确定手势库及 Ink 算法库的适应性需求;

2.4. 确定为了实现第 2.2 步和第 2.3 步的自适应性所需要的信息支持.

第 3 步. 笔式用户界面的自适应性规范化描述. 该步骤对步骤 2 中确定的内容进行规范化描述, 形成能够方便计算机处理的数据结构, 我们将该步骤又分为两个子步骤:

3.1. 用体系结构中平台层部分描述的元组对第 2.2 步中确定的可适应笔交互组件进行规范化描述和表示;

3.2. 对第 2.4 步中确定的自适应所需信息进行规范化的表达, 建立对应的模型.

第 4 步. 笔式用户界面的自适应策略. 为了完整地实现自适应的策略, 第 4 步又分为两个子步骤来完成:

4.1. 设计满足上述第 3.1 步中规范化描述的可适应组件集的可适应算法集合, 为算法集中的每种算法确定好输入和输出, 在该过程中, 如果发现规范化描述在第 3 步没有完善, 可以返回第 3 步, 进一步完善对可适应组件和其他支持自适应的信息的规范化描述.

4.2. 在设计完成自适应算法后, 我们设计自适应引擎的接口, 以确定在特定的状态调用算法集中对应的算法, 实现算法的调度, 在该过程中, 如果发现第 4.1 步中过程的算法集的设计和实现有疏漏, 可以返回到第 4.1 步过程中进行完善.

以上设计和开发的过程和步骤是一个迭代渐进的过程, 通过在当前步骤的设计和开发中可以发现上一个步骤存在的问题和疏漏, 从而返回进行迭代设计, 最后形成一个比较完备的设计和开发过程. 同时可以看出, 设计和开发过程中具体的步骤中操作的对象和提出的层次式的软件体系结构存在着对应. 我们按照设计和开发的基本过程本质上是将应用分解为体系结构中的各个部分, 软件体系结构确保将各个子模块自然、有机地组合为一个相互关联的、统一的整体, 从而能够规范化地并且方便、快捷地建立自适应笔式用户界面.

可以看到, 与一般的笔式用户界面的开发相比, 自适应的笔式用户界面的开发相对复杂, 且实现上需要更多的智能技术(如机器学习)的支持, 因此本节描述了自适应笔式用户界面软件开发的总体流程以及软件开发中设计和实现阶段的详细流程, 用以指导开发者在自适应笔式用户界面的设计和开发中遵循特定的开发流程, 降低开发自适应笔式用户界面的成本和风险.

4 开发实例

本部分给出一个笔式表单应用设计和开发的过程实例, 该设计开发过程基于本文提出的概念框架, 采用了层次式的自适应软件体系结构, 并在开发过程中遵循自适应的软件开发方法, 下面详细介绍概念框架在设计开发该笔式表单过程中的应用.

4.1 第1步: 笔式表单的需求分析

按照上一节描述开发流程, 本步骤分为两个子步骤.

4.1.1 确认应用功能及界面需求

笔式表单应用区别于传统的 GUI 表单应用, 笔式表单应用的主要特征表现在用户能够方便地对表单进行自由手写输入, 并对输入的内容进行识别, 识别的内容格式可以有数字、英文字母、中文等, 甚至有面向行业的特定规范格式. 我们对笔式表单的应用功能及界面需求分析的结果见表 1 中“基本功能”一列.

4.1.2 确认自适应的功能

自适应功能是软件在满足应用功能及界面需求后, 应用能够随用户使用经历和环境变化动态地改变交互行为和界面显示, 从而更好地支持特定的用户和环境, 提高软件产品的可用性. 在笔式表单应用中, 我们从输入方式、布局外观、执行命令 3 个方面来确定自适应的功能, 得出自适应的功能需求, 见表 1 中“自适应”一列.

4.2 第2步: 笔式表单的自适应分析

该阶段过程按 4 个子步骤依次顺序完成.

4.2.1 根据应用功能及界面需求确定笔交互组件库和手势库

按照第 4.1.1 节确认的应用功能及界面需求, 从设计和实现角度进行分析, 提出了单元格组件和表格组件,

其中,表格组件采用单元格组件的组合模式来实现,是单元格组件的父组件.单元格组件的主要功能是支持自由勾画的手写输入,同时能够对手写输入的内容进行识别.

4.2.2 根据自适应功能确定笔交互组件库的可适应接口

按照第 4.1.2 节确定的自适应功能,单元格组件按应用的需求应有多种类型.如支持数字识别的单元格组件以及支持汉字或英文字母识别的单元格组件.对识别的结果进行约束有利于提高系统的运行时性能和识别结果.在输出上,单元格组件和表格组件要满足能够支持多种布局方式,同时需要确定改变布局的触发条件.因此确定将单元格的类型、单元格和表格的自动布局算法、触发单元格类型设定和布局改变的条件作为笔式表单的可适应接口.

4.2.3 根据自适应功能确定手势库及 Ink 算法库的适应性需求

在表 1 中的“自适应的功能”一列,对手势的自适应功能作了专门的描述,由于笔手势对于笔式用户界面的重要性和笔手势的自适应策略在本文的前面部分有比较详细的描述,这里不再作专门阐述.

Table 1 Analysis of adaptive functions and it's dependent information in pen-based form user interface

表 1 笔式表单系统的自适应性及其依赖信息分析

	Basic functions	Adaptive functions	Required information
Input	Handwritten Chinese recognition Handwritten digit recognition Sketch	When some implicit rules exist among rows or columns in the form, we can limit the recognition's type to improve the efficiency of the input.	Domain knowledge
Layout (View)	Standard form layout	When the cell is full, we can enlarge the width or the height even can reduce the size of the font according to the requirements of the use or application.	Users' habit Devices' constrains
Command	Can execute the commands (Select,Delete, etc) by the input of single stroke pen gestures	Improve the gestures recognition engine by updating samples which are from the feedback of the mediate interface. Improving the recognition by integrating the context. We can also get input parameters by using the context of domain knowledge.	Interaction context Mediate mechanism Domain knowledge Users' characteristic

4.2.4 确定为了实现第 4.2.3 节和第 4.2.4 节的自适应性所需要的信息支持

自适应的用户界面运行时需要一些相关信息的支持,因此自适应分析的最后一个过程的主要工作是在分析笔交互组件的可适应接口以及笔手势库和 Ink 算法库的适应性需求的基础上,确定支持这些自适应的功能所需要的设备、环境、任务以及用户信息.我们对笔式表单应用的自适应性所需要的信息支持分析的结果见表 1 的“实现自适应所需信息”一列.

4.3 笔式表单的自适应规范化描述

在完成对笔式表单的自适应性分析以后,需要将分析的结果进行规范化的描述,方便计算机进行处理.与分析过程的子步骤相对应,我们将该部分分成两个子步骤.

4.3.1 用体系结构中平台层部分描述的元组对第 1.2 节中确定的可适应笔交互组件进行规范化描述和表示

在表 1 分析的基础上,第 3 步主要完成信息的规范化表示,我们举例说明对构件的描述:

组件 1=(单元格(子组件),通用识别器 && 单元格填满时触发父组件自适应函数,NULL);

组件 2=(单元格(子组件),数字识别器 && 单元格填满时触发父组件自适应函数,NULL);

组件 3=(单元格(子组件),汉字字符识别器 && 单元格填满时触发父组件自适应函数,NULL);

组件 4=(表格(父组件),组合(组件 1,2,3),自适应函数).

其中,组件 1~组件 3 为单元格组件,它们在处理手写输入时调用不同的识别引擎,并且当单元格的内容被填满以后调用父组件的自适应函数,单元格子组件本身的自适应函数为空.组件 4 为表格组件,是单元格组件所组成的组合组件,它的自适应函数功能为:(1) 根据任务上下文为某些单元格确定类型;(2) 当某些组件的约束函数条件满足时会调用指定的自适应函数,如某单元格被填满时调用自适应的布局功能.这里要说明的是,单元格的内容并不一定是单一的文本或数字这种简单规律的内容,如表格的某些行或列的内容满足一定的文法句子,

我们也可以利用领域模型对这种类型来建模,从而提高处理效率和识别率.

4.3.2 对第 4.2.4 节中确定的自适应所需信息进行规范化的表达,建立对应的模型

该步骤要求对实现自适应所需的信息进行规范化的描述和表示,以形成自适应算法容易处理的数据结构,生成对应的模型.这里,我们示例描述笔式表单系统领域知识和对交互历史进行建模.

我们分别用两个特定类型的矩阵来描述表单的知识:假设界面中是一个具有 s 行和 n 列的表格,约定两个 $s \times n$ 矩阵分别为 A_{sn} 和 B_{sn} .其中, $a_{i,j}$ 和 $b_{i,j}$ ($0 \leq i < s, 0 \leq j < n$) 分别代表 A_{sn} 和 B_{sn} 中第 i 行和第 j 列的值.

$a_{i,j} = CT(i, j), 0 \leq i < s, 0 \leq j < n$ (第 i 行第 j 列单元格组件类型).

$b_{i,j} = LS(i, j), 0 \leq i < s, 0 \leq j < n$ (第 i 行第 j 列单元格排版策略).

$LS(i, j) \in \{None, Font, Width, Height, WH, All\}$ (6 个属性值代表不同的排版方案,分别为不采取任何策略、单单元格字体缩放、宽度缩放、高度缩放、宽度和高度同时缩放以及所有缩放方式的结合).在系统初始化时,我们将矩阵 A_{sn} 中每个元素的值初始化为通用识别器类型, B_{sn} 中每个元素的值初始化为 *None*.

4.4 笔式表单的自适应策略

第 4 步为设计和实现自适应的算法集和算法调度.在步骤 3 的基础上,我们能够比较方便地建立笔式表单系统的自适应策略,如我们可以通过统计和分析某一行或列中已填单元格的类型来估计未填单元格的类型.同时通过统计和分析某行或列中单元格已经排版布局的方案来估计当该行或列再次达到重新排版条件时的排版方案等.以上具体的策略本文不作说明.手势和 Ink 算法库的自适应策略在第 2 节有详细的说明.在具体应用中可以根据其技术方案作一定的扩展和改进.

我们从对构件的描述中可以看到,当构建处于某个特定的状态时,会触发相应的自适应函数,在这种情况下,系统能够自动地实现自适应算法的调度,但在某些状态下,需要用户显示调用自适应的策略,比如用户在输入的手势没有正确识别的情况下,可以放弃这次输入,或者可以通过在线纠错机制来让识别器动态更新输入样本来学习从而适应用户的手势输入习惯.

4.5 实例及其应用

目前,基于以上的开发流程指导,我们成功地开发了 3 个类似的笔式表单应用系统,分别应用在国家队的跳水、水球和羽毛球队中,用于教练的计划制定等诸多应用场景,并已在教练员的日常训练指导工作中得到了应用.

4.6 实例应用评估

根据自适应用户界面的评估标准^[26],为了检验本应用实例的可用性,我们观察记录了 5 个用户(教练员)接触系统一个月内的使用过程.我们从系统的效率、自适应行为相对于用户意图的准确度两个方面进行了初步的实验定量分析,通过用户的主观评分对系统进行了综合评估.我们从用户完成表单内容填写的时间上来验证系统的效率.为了评估自适应行为相对于用户意图的满意度,在用户使用的过程中,系统记录每次系统产生的自适应行为以及用户对该行为的反应,如果用户撤销自动执行的自适应行为,系统将记录此次的系统自适应行为与用户意图不符.经过一个月的使用,用户完成了 20 个表单内容的填写,用户完成每个表单内容的时间如图 7(a)所示,从图中可以看出,随着用户使用经验的积累,用户完成表单内容的填写任务将会逐渐减小,最后趋向于一个比较固定的时间区域.用户完成每个表单内容填写过程中产生预测的次数和预测不符合用户意图的次数如图 7(b)所示,从图中可以看出,随着用户使用经验的积累,自适应行为相对于用户意图的准确度也随之提高,而且比较快速地达到较高的准确度.

一个月后,我们对用户作了一次问卷调查,让用户对使用系统的速度,自适应行为的准确度、舒适度、愉悦度、满意度进行主观评分,问卷设计时我们将每一项从很差到很好分成 7 个等级,分数由低到高,对收回的 5 份调查问卷结果的分析如图 8 所示.

通过上述实验评估,我们初步得到,本文中的笔式自适应表单应用实例具有较高的可用性,从而进一步验证了文中所提出的概念框架对设计和开发过程的指导效果.

跳水队训练计划表					羽毛球队日训练计划		
日期: 2003-06-04 时段: 下午					运动类型: 羽毛球 日期: 2007-10-08		
弹网	陆台	陆板	一米板	三米板	时间	训练内容	完成指标
302A×3 3325D×5 304B×5 104B×5 106B×5	302C×5B×5 302C×5B×5 402C×5	陆板冰棍A×10B ×10 陆板棍A×10 102B×3 104B×5 302B×5 304C×5 302A×3 3325D×3 202C×3B×5 402C×3B×5	401C×5 单弹101C×3 105C×3 301A×3 306C×3 陆板102B×3 105C×5B×5 308B×3 302A×2 3325D×3 立龙201A×3 205B×3 401C×3 405B×3	单弹101B×3 走板100B×5 205B×5 双人401B×4 405B×4	8:15-9:00	预备运动 / / / / /	
动作总数:23	动作总数:25	动作总数:70	动作总数:50	动作总数:21			
三米台 例下×5	五米台 例下×10	七米台	十米台	身体素质			

Fig.6 Pen-Based form system for a sports teams' training management

图6 运动队笔式表单训练管理系统

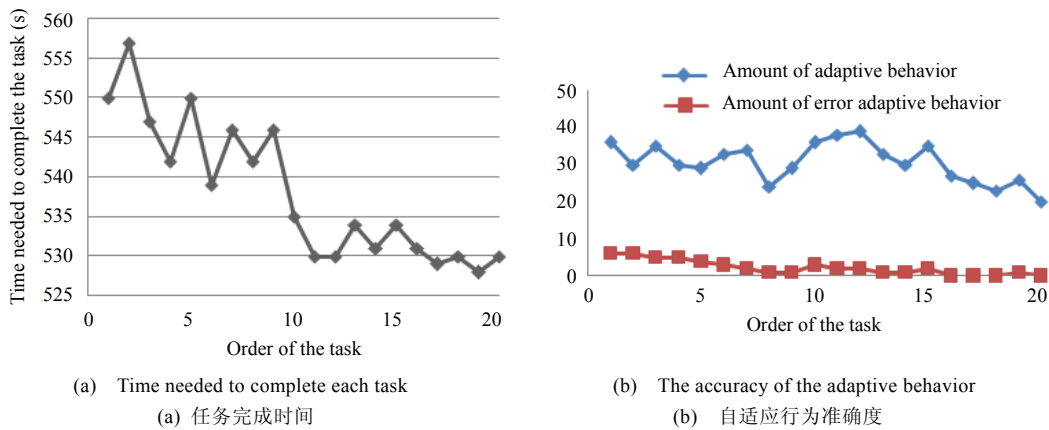


Fig.7 Experimental results

图7 实验评估结果

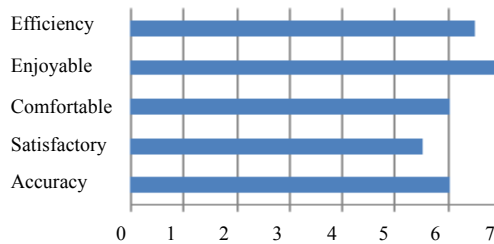


Fig.8 Subjective ratings

图8 用户主观评分

5 结论和进一步工作

本文给出了指导自适应笔式用户界面开发的概念框架,描述了自适应笔式用户界面的软件体系结构,并且在该体系结构的基础上给出了自适应笔式用户界面的软件开发方法,最后以实例来详细说明过程中的主要步骤流程的实现方法途径.说明了该框架在自适应笔式用户界面的设计和开发中的指导作用.

未来的工作将体现在进一步用该概念框架指导设计和开发其他自适应的笔式用户界面的原型和应用系统,且将进行更加严格而深入的比较实验,以及在这些实践中进一步检验和完善该框架.

References:

- [1] Landay JA, Myers BA. Interactive sketching for the early stages of user interface design. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press/Addison-Wesley Publishing, 1995. 43–50. <http://portal.acm.org/citation.cfm?id=223910>
- [2] Hong JI, Landay JA. SATIN: A toolkit for informal ink-based applications. In: Proc. of the 13th Annual ACM Symp. on User Interface Software and Technology. New York: ACM Press, 2000. 63–72. <http://portal.acm.org/citation.cfm?id=1185657.1185768&coll=GUIDE&dl=GUIDE&type=series&idx=SERIES382&part=series&WantType=Proceedings&title=SIGGRAPH&CFID=683760&CFTOKEN=10242240>
- [3] Lin J. DENIM: Finding a tighter fit between tools and practice for Web site. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 2000. 510–517. <http://portal.acm.org/citation.cfm?id=894366&jmp=cit&coll=GUIDE&dl=GUIDE>
- [4] Tian F, Qin YY, Wang XC, Ao X, Wang HA, Dai GZ. Analysis and design on PIBG toolkit: A pen-based user interface toolkit. Chinese Journal of Computers, 2005,28(6):1036–1042 (in Chinese with English abstract).
- [5] Dong SH, Wang J, Dai GZ. Human-Computer Interaction and MultiModel User Interface. Beijing: Science Press, 1999 (in Chinese).
- [6] Jr Fuller TH. Supervised competitive learning: A technology for pen-based adaptation in real time [Ph.D. Thesis]. Washington: Department of Computer Science, Washington University, 1994.
- [7] <http://nano.xerox.com/hypertext/weiser/SciAmDraft3.html>. 2008.
- [8] Dieterich H, Malinowski U, Kuhme T, Schneider-Hufschmidt M. State of the art in adaptive user interfaces. In: Schneider-Hufschmidt M, Kuhme T, Malinowski U, eds. Adaptive User Interfaces: Principles and Practice. Adaptive User Interfaces. Amsterdam: Elsevier Science Publishers B.V., 1993. 13–48.
- [9] Duarte C, Carriço L. A conceptual framework for developing adaptive multimodal applications. In: Proc. of the 11th Int'l Conf. on Intelligent User Interfaces. New York: ACM Press, 2006. 132–139. <http://portal.acm.org/citation.cfm?id=1111481>
- [10] Iwayama N, Ishigaki K. Adaptive context Proc. in on-line handwritten character recognition. In: Schomaker LRB, Vuurpijl LG, eds. Proc. of the 7th Int'l Workshop on Frontiers in Handwriting Recognition. Amsterdam: Int'l Unipen Foundation, 2000. 469–474.
- [11] Xiang C, Balakrishnan R. Evaluation of an on-line adaptive gesture interface with command prediction. In: Proc. of the 2005 Conf. on Graphics Interface. Victoria: Canadian Human-Computer Communications Society, 2005. 187–194. <http://portal.acm.org/citation.cfm?id=1089508.1089540>
- [12] Sun ZX, Peng BB, Cong LL, Sun JY, Zhang B. Study on user adaptation for on-line sketch recognition. Journal of Computer-Aided Design & Computer Graphics, 2004,16(9):1207–1215 (in Chinese with English abstract).
- [13] Szummer M, Yuan Q. Contextual recognition of hand-drawn diagrams with conditional random fields. In: Proc. of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR 2004)—Volume 00. Washington: IEEE Computer Society, 2004. 32–37. <http://portal.acm.org/citation.cfm?id=1033909&dl=GUIDE&coll=GUIDE&CFID=15151515&CFTOKEN=6184618>
- [14] Ronby E, McCall PK, Moran TP, Halasz FG. Tivoli: An electronic whiteboard for informal workgroup meetings. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 1993. 391–398. <http://portal.acm.org/citation.cfm?id=169059.169309&coll=portal&dl=ACM&CFID=15151515&CFTOKEN=6184618>
- [15] LaViola J. Mathematical sketching: A new approach to creating and exploring dynamic illustrations [Ph.D. Thesis]. Department of Computer Science, Brown University, 2005.
- [16] Tenneson D. ChemPad: A pedagogical tool for exploring handwritten organic molecules. Technical Report, No. RI 02912, Brown University, 2005.
- [17] Forsberg A, Dieterich M, Zeleznik R. The music notepad. In: Proc. of the 11th Annual ACM Symp. on User Interface Software and Technology. New York: ACM Press, 1998. 203–210. <http://graphics.cs.brown.edu/research/music/tpc.html>
- [18] Rubine D. Specifying gestures by example. In: Proc. of the 18th Annual Conf. on Computer Graphics and Interactive Techniques. New York: ACM Press, 1991. 329–337. <http://portal.acm.org/citation.cfm?id=122753>

- [19] Jr Long AC, Landay JA, Rowe LA, Michiels J. Visual similarity of pen gestures. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems. New York: ACM Press, 2000. 360–367. <http://portal.acm.org/citation.cfm?id=894169&coll=GUIDE&dl=GUIDE&CFID=15151515&CFTOKEN=6184618>
- [20] Jr Long AC. Quill: A gesture Design Tool for Pen-Based User Interfaces. Berkeley: University of California, 2001.
- [21] Pastel R, Skalsky N. Object-Action association: A HCI design model. In: Proc. of the 9th Int'l Conf. on Intelligent User Interfaces. New York: ACM Press, 2004. 295–297. <http://portal.acm.org/citation.cfm?id=964442.964510&coll=portal&dl=ACM&type=series&idx=SERIES823&part=series&WantType=Proceedings&title=IUI&CFID=15151515&CFTOKEN=6184618>
- [22] Yue WN, Wang Y, Wang GP, Wang H, Dong SH. Architecture of intelligent interaction system based on context awareness. Journal of Computer-Aided Design & Computer Graphics, 2005,17(1):74–79 (in Chinese with English abstract).
- [23] Samek M. Practical statecharts in C/C++: Quantum programming for embedded systems with CDROM (Paperback). In: CMP Books. Pap/Cdr edition (July 2002). http://www.librosaulamagna.com/libro/PRACTICAL_STATECHARTS_IN_C-C++:_QUANTUM_PROGRAMMING_FOR_EMBEDDED_SYSTEM_WITH_CDROM/7423/1800
- [24] Qin YY, Tian F, Wang XC, Dai GZ. An interaction-centered hierarchical post-WIMP user interface model. Journal of Software, 2006,17(4):691–702 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/17/691.htm>
- [25] Liu JM, Wong CK, Hui KK. An adaptive user interface based on personalized learning. Intelligent Systems, IEEE Intelligent Systems and Their Applications, 2003,18(2):52–57.
- [26] <http://www.isle.org/~langley/papers/adapt.exper.ps.gz>. 2008.

附中文参考文献:

- [4] 田丰,秦严严,王晓春,敖翔,王宏安,戴国忠.PIBG Toolkit:一个笔式界面工具箱的分析与设计.计算机学报,2005,28(6):1036–1042.
- [5] 董士海,王坚,戴国忠.人机交互和多通道用户界面.北京:科学出版社,1999.
- [12] 孙正兴,彭彬彬,丛兰兰,孙建勇,张斌.在线草图识别中的用户适应性研究.计算机辅助设计与图形学学报,2004,16(9):1207–1215.
- [22] 岳玮宁,王悦,王国平,王衡,董士海.基于上下文感知的智能交互系统模型.计算机辅助设计与图形学学报,2005,17(1):74–79.
- [24] 秦严严,田丰,王晓春,戴国忠.以交互为中心的 Post-WIMP 界面模型.软件学报,2006,17(4):691–702. <http://www.jos.org.cn/1000-9825/17/691.htm>



徐礼爽(1982—),男,安徽潜山人,博士生,主要研究领域为人机交互,笔式计算,自适应用户界面.



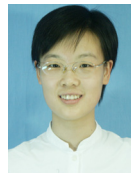
周明骏(1982—),男,博士生,CCF 学生会会员,主要研究领域为人机交互,协同计算.



邓昌智(1978—),男,博士生,主要研究领域为人机交互,个人信息管理,信息可视化.



田丰(1976—),男,博士,副研究员,主要研究领域为人机交互技术,界面工具.



刘媛媛(1982—),女,硕士生,主要研究领域为人机交互技术.



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,计算机图形学.