

# 一种笔式用户界面软件领域模型及其应用\*

秦严严<sup>+</sup>, 田 丰, 王晓春, 戴国忠

(中国科学院 软件研究所 人机交互技术与智能信息处理实验室, 北京 100080)

## A Domain Model of Pen-Based User Interface Software and Its Usage

QIN Yan-Yan<sup>+</sup>, TIAN Feng, WANG Xiao-Chun, DAI Guo-Zhong

(Laboratory of Human-Computer Interaction and Intelligent Information Processing, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-10-62540451 ext 8031, Fax: +86-10-62540451, E-mail: yanyan02@iscas.cn, <http://iel.iscas.ac.cn>

**Qin YY, Tian F, Wang XC, Dai GZ. A domain model of pen-based user interface software and its usage. *Journal of Software*, 2006,17(1):69-78. <http://www.jos.org.cn/1000-9825/17/69.htm>**

**Abstract:** Pen-Based user interface software is widely applicable due to its natural and efficient interaction style, whose important characteristics are interaction-centered and individual by user-required. In order to map the intention of users into the designs exactly and efficiently, PUIDM(Pen based User Interfaces software Domain Model) is presented, which is described and analyzed from the aspects of context, component, UI features and architecture, and implemented based on XML. Moreover, a user-centered design process based on the PUIDM is introduced. Experiments show that PUIDM can guide the process of design exactly and efficiently and make the designed software achieve the demand of usability.

**Key words:** pen-based UI software; user-centered design; end user; domain model; XML

**摘 要:** 笔式用户界面软件以其自然、高效的交互方式,在很多领域中有着广泛的应用.笔式用户界面软件具有以交互为中心、用户个性化需求高的特点,由此也决定了用户在软件设计中的主导地位.以用户为中心的设计的关键问题在于,如何使用户的思想如实地反映到设计中.通过建立笔式用户界面软件特征模型 PUIDM(the domain model for pen-based user interface software),构造了一个连接用户与软件设计的平台.从上下文、软件实体、界面特征、体系结构等角度,对该模型的建立进行分析,并给出了相应的 XML 描述.在此基础上,描述了用户使用该模型进行软件设计的过程.实例表明,PUIDM 能够正确引导用户进行设计,将用户意图充分地引入到软件设计和最终实现中,使软件满足可用性要求.

**关键词:** 笔式用户界面软件;以用户为中心的设计;最终用户;领域模型;XML

中图法分类号: TP391 文献标识码: A

\* Supported by the National Grand Fundamental Research 973 Program of China under Grant No.2002CB312103 (国家重点基础研究发展规划(973)); the National Natural Science Foundation of China under Grant No.60503054 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2003AA411330 (国家高技术研究发展计划(863)); the Key Innovation Project from Institute of Software "Research on Pen-based Interactive System and Its Applications in Personal Office Environment"; the Chinese Academy of Sciences (中国科学院软件研究所创新基金重大项目)

Received 2005-04-11; Accepted 2005-06-02

笔式用户界面由于其基于纸笔隐喻,接近人们日常生活的自然、高效的交互方式,以及在捕捉及表达用户模糊意图上的优势成为新一代用户界面的主要研究方向之一<sup>[1]</sup>。同时,研究表明,笔式用户界面软件是一个用户需求集中于笔交互任务的特定应用软件领域。笔式用户界面软件开发的主要任务是提供用户利用笔与界面进行交互的服务,因此,如何将用户意图正确地引导为系统设计就成为该领域软件设计中最为重要的问题之一。

Preece<sup>[2]</sup>提出交互式系统设计的 3 个关键特征是以用户为中心、确定的可用性标准和迭代式设计方法。目前的研究主要集中在两个问题上。首先是如何将用户纳入到软件的设计过程中。在传统的系统设计中,领域专家和 HCI 设计人员是设计过程的主体,而最终用户无法在设计开始时就处于决策地位,直接导致了用户与设计人员的交流误差和设计过程迭代次数的增加。第二个问题是如何进行快速的迭代并产生出用户易于理解的设计原型。缺乏对领域内可复用信息的抽象描述使得原型建立只能在构件级或代码级等较低层次进行,无法保证用户概念模型、设计模型与系统实现的统一。

对于上述问题,从笔式交互任务出发,通过领域建模来抽象笔式用户界面软件框架,利用领域模型对用户需求任务及其组织方式,领域内可复用信息以及软件体系结构进行抽象描述。笔式用户界面软件领域模型 PUIDM(the domain model for pen-based user interface software)为笔式用户界面软件的开发提供了一个基于文档的交流平台和软件描述。该模型支持以用户为中心、以设计人员为指导的快速迭代设计方法,能够良好地建立与用户概念模型的映射。下面将分别讨论此特征模型的结构、建立、基于 XML 的表示方法以及具体设计实例分析。

## 1 相关工作

Li<sup>[3]</sup>提出笔式用户界面所具有的非精确、连续和隐式交互特征以及笔交互设备的便携性和易用性,使得笔式用户界面在创造性工作、信息交流和捕捉以及思想捕捉等领域有着广泛的应用。在对笔式用户界面的研究过程中建立了大量的应用系统。由 CMU 机械工程系实现的 SketchIT<sup>[4]</sup>是一个支持机械概念设计的草图工具。之后开发的 SILK<sup>[5]</sup>则是通过勾画进行图形用户界面设计的系统。加州大学 Berkeley 分校设计并实现的 DENIM<sup>[6]</sup>是一个支持勾画的网站设计系统,之后又在此基础上开发了面向多设备的版本 Damask<sup>[7]</sup>。Davis<sup>[8]</sup>等人开发了基于自由笔迹的动画制作工具 Flatland<sup>[9]</sup>是由 Georgia 理工、东京大学和 Xerox PARC 联合研制的电子白板系统,以及 MIT 与 Microsoft 联合开发的利用勾画进行物理仿真的 ASSIST<sup>[10]</sup>等。然而,使用输入设备的差异、笔交互信息定义的互不兼容以及用户使用习惯的个性化,都使得这些应用并没有一个对笔式用户界面进行良好描述的框架。不同类型的笔输入设备由于输入区域和环境显示效果的不同,造成界面布局与界面风格上的差异;而同类型的笔输入设备之间缺乏对笔交互信息的描述规范,使得软件应用往往只能对特定的硬件设备进行设计实现。同时,不同用户在使用笔进行工作、办公或处理个人事务时都存在明显的个性化风格,缺乏对笔式用户界面的高层次抽象和可靠的软件复用技术支持,导致软件系统很难满足所有用户的使用需求和习惯。

以用户为中心设计的特点是让用户参与设计,一般存在两种形式:一种是让用户参与评估研究,另一种方式是让用户参与设计本身<sup>[2]</sup>。Keil 和 Carmel<sup>[11]</sup>的研究表明,用户是否直接参与直接影响着项目的成功与否。Kujala 和 Mantyla<sup>[12]</sup>通过对产品开发初期的用户研究成本收益的研究,认为用户研究的效益高于成本。随着用户在需求多变的应用开发中所处地位的不断提,“参与式设计(participatory design)”<sup>[13]</sup>逐渐成为以用户为中心开发的研究热点。PICTIVE<sup>[14]</sup>是一个基于纸张的原型制作技术,通过利用一些设计用品使用户能够参与设计过程并改进设计过程的知识获取方法。在另一个原型制作技术 CARD<sup>[15]</sup>中,用户使用卡片来发掘 workflow,确定一系列屏幕图像或任务决策点,以进行情节串连。用户参与开发的目的是将用户概念模型、领域任务模型良好地映射为系统实现。然而,在目前的用户参与设计方法中还缺乏将用户设计意图和系统实现技术进行统一抽象的交流平台,使得用户与其他设计人员或系统实现人员难以进行有效的沟通。

Arango<sup>[16]</sup>提出,领域的内聚性和稳定性是获取和表示领域内可复用信息的前提。Kyo<sup>[17]</sup>认为,对于相关软件系统的共性进行系统的分析和利用是成功的软件复用的基础。目前,针对不同的目标、产品和过程出现了多种领域工程方法。CMU/SEI 提出的 FODA<sup>[18]</sup>方法引入特征模型进行需求的抽象描述。其他如 FOOM<sup>[19]</sup>,PLA<sup>[20]</sup>等

方法也都利用多种模型进行领域内应用的共性与特性描述.其目的都是提供一个统一的模型来描述同一类应用系统以及潜在理论,并为新的软件应用提供一套快速实现机制.然而,现有的领域开发方法的主体仍然是领域专家、领域设计人员等,在面向采用可预测交互方式图形用户界面的特定软件领域时开发效率较高.但是,在面向用户个性化要求高、并采用在交互过程中产生大量模糊信息的新交互范型的软件领域时,由于缺乏最终用户对软件开发过程的积极参与,因而无法保证该类软件的可用性.

## 2 领域模型

软件开发的过程实质上就是开发人员将用户在问题域上的需求映射到方法域的求解过程.在这个过程中,软件开发人员必须面对两个关键问题:一个是如何将问题域中的问题进行准确而合理的表示,以便后期开发;另一个问题则是如何将该问题的描述快速映射到方法域中,得到求解系统.传统的软件开发过程往往将这两个问题分离为软件需求分析、软件设计、开发、评估等多个利用大量文档进行沟通的单独过程,并认为这样可以满足用户的要求并保证系统的可靠性.然而,大量附加产品的产生带来的不仅仅是开发效率的降低以及大量管理问题,同时也意味着对于用户不断改变的需求,软件开发无法给予快速的反馈,从而降低了系统的可用性.通过特征描述、领域术语词典、可复用构件等技术,将问题域和方法域的描述有机地结合在一起,使用户与系统的设计开发人员可以进行良好的沟通,允许用户进行自主设计与开发,使系统能够快速而准确地反映用户变化的需求,保证实现系统的可用性.

领域模型的建立分 3 个阶段进行:上下文分析、领域建模和体系结构建模.上下文分析确定领域边界;领域模型用于领域内问题的描述;而体系结构模型则提供了方法域的上层结构描述.

### 2.1 上下文分析

为了了解各类型笔式用户界面应用之间的差异与共性,我们针对不同类型的应用输入笔迹内容,面向应用人群、使用设备进行了分析.见表 1.

**Table 1** Application analysis of the pen-based user interface software

表 1 笔式用户界面软件应用分析

Application	Feature	Ink content	User	Device
Creative work		Sketch + Graphics	Domain expert, Software designer	Digital screen, Whiteboard, Tablet PC
Information exchange and sharing		Sketch + Text	Leader, Teacher	Whiteboard, Digital screen
Idea capturing		Sketch + Text	General people	Mobile phone, PDA, Tablet PC
Enhancing legacy software		Text + Graphics + Table	Domain expert	Digital screen, Tablet PC

应用领域任务和用户习惯的差异造成了不同的软件使用环境,在笔式用户界面软件领域中,突出体现在使用设备的多样性上.目前比较常见的笔式输入设备有手写板、手写屏、移动笔输入设备、电子白板、Tablet PC 以及 Anoto 笔.各种笔输入设备在界面设计上都有着特定的设计原则.

从书写内容和笔迹计算技术上来看,以用户输入笔迹为基础的草图、正文、图形和表格是笔式用户界面软件处理内容的主要部分.其中,在笔迹绘制、结构分析、笔迹识别以及笔迹搜索方面包含了多种笔迹计算技术,每项计算技术又同时包含很多相关的技术支持.对于各种技术的选择与使用,需要根据笔迹内容计算效率以及用户习惯等因素来决定,见表 2.

从笔式用户界面软件所涉及的用户群来看,用户人群范围较广.但从每一种应用类型所对应的用户群来看,其用户都是该应用领域的专家.即用户可以在完成任务的过程中根据自己的意图和习惯来决定需要进行的操作以及操作的方式.

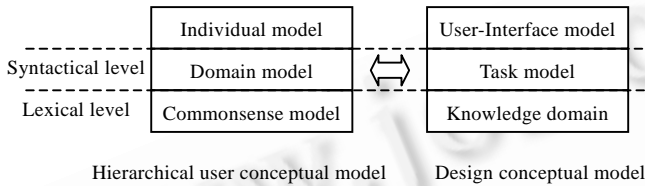
上面我们对笔式用户界面软件领域内的用户信息、使用设备、书写内容以及采用的基本技术进行了分析.领域内特定应用将由于面向的用户群体、输入的内容信息风格、采用的设备、技术等不同而产生各自的特性.

Table 2 Application analysis of the pen-based computing technology

表 2 笔式计算技术应用分析

Ink content Pen-Based computing tech.	Sketch	Text	Graphics	Table
Ink render	Fitting rendering	Rendering	Rendering	Inputting tip
Structure analysis	Ink typesetting User idea capturing	Text line analysis text char partition	Graphics primitive analysis text-graphics classification	Table primitive analysis text-table classification
Ink recognition	Gesture recognition	Text recognition gesture typesetting	Graphics recognition gesture typesetting	Table recognition gesture typesetting

2.2 领域建模



Hierarchical user conceptual model Design conceptual model

Fig.1 The mapping of the conceptual model

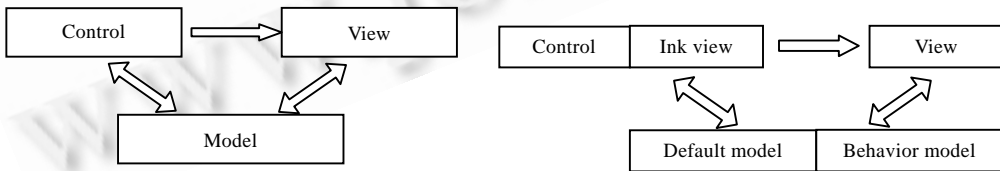
图 1 概念模型映射

在软件开发的同时,我们需要确定用户能否理解关于系统外观及行为的问题.Norman<sup>[21]</sup>提出了一个用于说明“设计概念模型”与“用户理解模型”之间关系的框架(如图 1 所示).理想状态下,这两个模型与系统之间能够完全相互映射,用户通过与系统映像相交

互,能够按照设计人员的意图(体现在系统映像中)去执行任务.但若系统映像不能明确地向用户展示设计模型,那么,用户很可能无法正确理解系统,造成用户在使用系统时效率较低,而且易出错.

用户个体之间的差异使系统在建立界面时需要针对每个用户进行适应和调整.由上下文分析可知,用户在日常生活中使用笔完成任务时,用户个体往往本身就是该任务领域的专家,对所使用的工具有着强烈的自我意识,因此在面对新工具时同样有着强烈的设计和改进欲望.用户与领域专家身份的一致性,使得在笔式用户界面软件领域中,用户的作用被更加突出于特定应用的软件设计中.用户参与设计使设计模型与用户模型得到了自然的映射.

MVC 模型<sup>[22]</sup>是较早提出的一种面向对象的交互式系统模型,该模型将系统抽象为 3 种相对独立的对象:模型(model)、控制(controller)和视图(view),如图 2(a)所示.模型试图将系统的交互输入和显示输出部分分离出来,然而在系统运行时,必须通过消息传递等机制来控制输入事件与数据显示输出事件,这样,视图和控制的改变就不可避免地关系到内部控制模型的结构,也就无法保证对象的独立性.之后的 PAC 模型<sup>[23]</sup>根据系统的功能划分了多个智能代理,其中每个代理都可以由 MVC 模型来描述.



(a) MVC model  
(a) MVC 构件模型

(b) Pen-Based UI component model  
(b) 笔式用户界面构件模型

Fig.2

图 2

在笔式用户界面中,当用户完成不同的交互任务时,系统在处理交互信息的上下文逻辑以及采用的具体实现技术上都存在着较大的差异.这种差异导致了多代理结构的形成.系统在完成不同的交互功能时,对接收交互信息时所产生的上下文处理在逻辑和技术实现上被独立成为不同的功能代理构件.然而在构件的对象描述上,

笔交互的特性使得 MVC 的对象结构不再适用于笔式用户界面领域。

在笔交互用户界面中,用户输入笔迹的同时需要进行笔迹显示输出事件.另一方面,由于笔式用户界面仍属于一种 Post-WIMP 界面,这就意味着,在 WIMP 构件以及构件对笔迹处理的控制部分和系统显示部分仍会保持一定的独立性.提出了笔式用户界面构件模型 MCRiv(model-control-representation(ink and view)),如图 2(b)所示.

考虑笔迹在系统中的作用,从该模型我们可以看出笔式用户界面构件的几个特点:首先,笔交互信息输入与笔迹显示部分直接连接,笔交互信息在输入后首先被转化为笔迹信息,进行笔迹显示,从而自然地保持了输入和显示的一致性;通过模型的设置,系统可以对笔迹的显示效果进行配置并允许视图对笔迹信息显示进行后处理;用户输入的笔迹直接映射至模型中的数据结构,实现了输入笔迹和数据模型的隐式连接;用户或设计者自行定义构件行为模型可以替换或共同作用于预定义的交互信息处理模型.

根据上述模型分析,我们将笔式用户界面上的实体分为 4 类.

界面构件实体.界面构件实体是笔式用户界面的主要实体,其主要任务是接收来自系统经笔事件生成器转化或由其他实体传送的笔式交互信息.这些笔式交互信息通过笔式交互信息处理模块,经上下文分析模块确定当前实体状态后,形成笔式交互原语或向其他实体发送交互信息,并通过显示模块完成词法反馈.当笔式交互信息转化为交互原语后,同样在上下文分析和外部手势识别模块分析后进行对实体内容的管理(包括对实体自身属性及其子实体的管理)或是向其他实体发送交互原语,并完成显示模块的词法反馈.在内容管理模块中,不仅要接收词法和语法层次的输入并进行语义反馈,同时还要接收用户对内容的直接决策(比如,在文字识别中,用户通过候选列表进行识别内容的修改操作).笔式用户界面构件实体结构如图 3 所示.

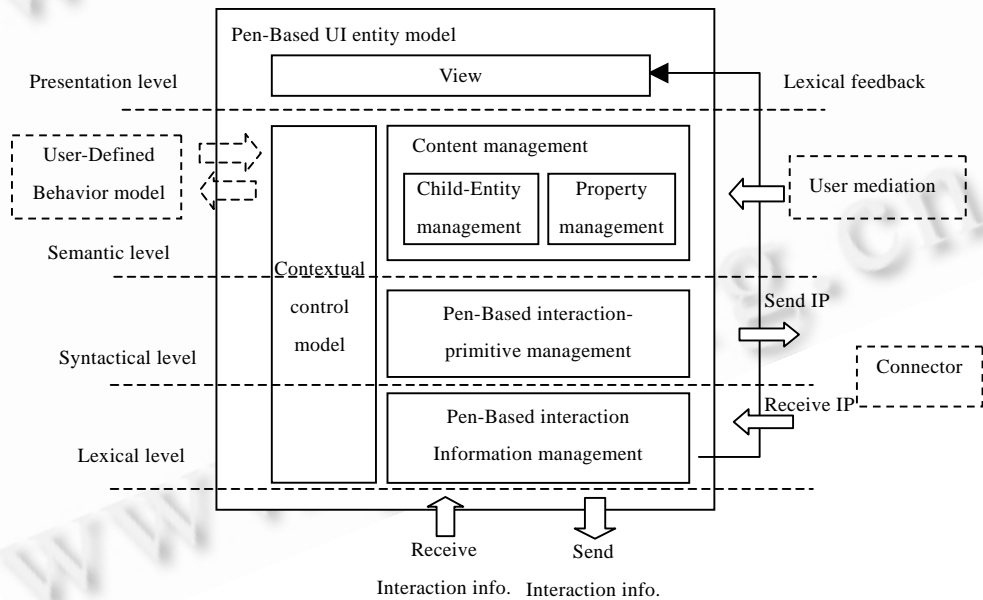


Fig.3 Pen-Based UI entity model  
图3 笔式用户界面构件实体模型

由于界面构件实体在实现中将以组件的形式出现,其内部处理模块,尤其是上下文分析模块和交互信息与交互原语的处理机制对于外部都完全透明.

行为实体.行为实体是用于表述构件实体在接收到笔式交互信息或笔式交互原语后所进行的处理行为的描述性实体.构件的行为(behavior)是一系列事件处理规则(rule)的集合.规则描述分为两部分:输入事件(event in)和事件输出(event out).输入事件是指可能引起行为的操作,考虑到分布式系统的事件并发性与事件发生条

件的多样性,我们引入事件的与或算子,将事件发生的必要条件同样作为一种事件来对待.对系统并发的事件以及事件发生的条件以事件与或树的方式进行描述.输入事件描述包括事件类型(event type)和事件输入源引用(source ref).输出事件是指在确定行为发生后的系统反馈事件.事件发生后通过构件的连接实体进行事件路由,得到系统反馈.

连接实体.连接实体就是上文所述的连接器,具体分为接收器(receiver)和发送器(sender)两部分.其中接收器主要功能有两个:接收(receive)数据或行为消息以及传递消息回应(reply).发送器的主要功能是进行信息发送(send)和发送数据请求(require).

实体关系.不同实体之间在逻辑和处理交互信息的层次上存在着不同的相互关系,关系模型用于描述实体之间的组织方式.关系类型主要包括:包含关系(consists-of association)和继承关系(inherited association).由于实体在接收处理笔式交互信息层次上具有不同,使得实体之间存在嵌套.各层次之间的实体由包含关系联系在一起;从同一层次的实体类型之间的关系来看,同层次实体对笔式交互信息的处理流程和内容管理功能上大体一致,因此利用面向对象中继承的概念可以很大程度地减轻实体的对应组件的开发和管理工作,同时也可以在于实体描述中自动继承父类实体的描述和生成方法,以防止描述文档无限制的增长.我们用继承关系来描述父子实体的关系.

特征模型是用于向用户描述系统应用的任务特性和交互特性的工具.该模型既要体现用户本体个性,同时也要表现出系统任务中的领域共性.对特征模型的剪裁和选取是联结问题域和方法域的主要手段,特征模型需要同时反映用户的需求以及笔式用户界面软件的实现接口.

在笔式用户界面的领域建模过程中,以纸笔交互为中心的领域特点使得我们可以将笔式交互的研究成果引入到建模过程中.确定领域的共性和变化性特征同样是笔式用户界面领域建模的主要任务.交互特征、实体特征和任务特征决定了笔式用户界面的共性特征;而笔式用户界面软件的可变特征则主要体现在软件应用的环境设备、特定应用对系统的任务需求以及特定应用的领域特征上.由于笔式用户界面的共性特征是以纸笔交互为中心,这与软件应用的主要需求任务相一致,因此,笔式用户界面软件的可变特征主要是对共有特征的原则剪裁.由此,特征建模的主要任务就是得到笔式用户界面软件的共有特征.

由前面的上下文分析可知,为了达到用户可用性标准,笔式用户界面软件应用所具备的任务能力并不仅仅是能够进行笔交互信息处理并提供反馈信息.在软件应用中,必须要考虑到用户个人的使用习惯、用户处理时的环境因素等.特征模型的建立必须要考虑到用户概念模型、设计模型与实际系统实现的统一.因此,我们给出特征模型的建立与用户概念模型可以相互参考的层次划分.笔式用户界面软件特征模型分为用户特征、交互风格特征、系统服务特征、表现特征、实现技术特征与质量特征几个层次,其中质量特征影响着其他各层次特征的选取和剪裁.

功能模型是特定应用在特征模型以外,或是对特征模型原有约束进行改变的用户需求描述.当面对特殊的用户需求时,需要进行新的实体组件或功能在代码级别上的开发.在功能模型中,对需要开发的实体组件或功能模块进行描述,并给出这些新开发部分与已有部分的关系和接口.功能模型包含对特定功能的操作描述以及与已有部分的关系接口描述.

领域术语词典是联系最终用户、领域专家和软件的设计开发人员的桥梁.通过领域术语词典,用户可以了解在领域内某个术语的具体含义,而开发人员也可以清楚用户在使用某个特定词语进行描述时的真正意图.术语词典的建立与各方面参与人员都有一定关系.其描述是术语单词与其说明文字一一对应的组合.

### 2.3 体系结构建模

软件体系结构模型是系统实现的重要设计部分.它的目的是为领域内定义的问题提供一个在实现层次的解决方案(solution).在新的构件模型的基础上,笔式用户界面模型根据其交互任务的不同采用多代理的结构,在构件之间使用连接器进行控制信息和数据信息传递以及对数据的显示约束.与 PAC 模型不同,这里,我们着重强调了连接器的作用.连接器是一个消息管道,连接着不同层次的构件,如图 4 所示.

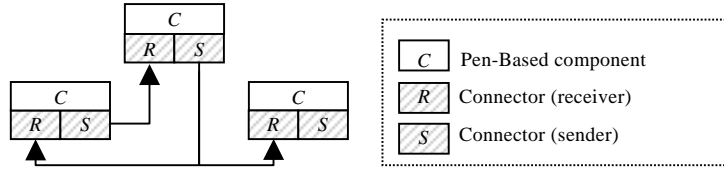


Fig.4 Pen-Based UI component-connector model

图4 笔式用户界面构件-连接模型

连接器的主要作用是通过接收器(receiver)和发送器(sender)进行构件之间的数据和控制信息传递,在连接器与构件之间采用统一的数据及控制接口,从而提高了构件以及连接器的可复用性.同时,连接器的存在使得构件可以分布在不同的物理空间,以支持分布式的界面交互.

笔式用户界面软件的主要任务是纸笔交互,这一点决定了其体系结构模型必然会反映笔式用户界面范式与笔式交互信息处理流程的特点.从笔式用户界面范式与笔式交互信息处理模型<sup>[24]</sup>来看,笔式用户界面的笔式交互信息处理实体可以分为两级:纸(paper)和内容区(region).从系统的角度来看,纸是内容区的容器.其主要功能是接收用户或其他实体发送的交互信息,对内容区进行管理,通过当前各内容区的状态将笔式交互信息发送到适当的信息接收实体,根据自身属性进行内容显示;内容区是不同格式的输入信息的容器.其主要功能是接收其他实体发送的交互信息,对输入信息进行处理(识别),形成笔式用户界面中的标准信息(笔迹、正文文字、图形、表格、公式等),并对其进行管理,进行内容显示.从用户角度来看,纸是交互的界面载体,数字世界中的纸是物理世界中真实“纸”的电子模板.一旦用户在设计过程中对某一种数字纸进行了个性化配置,便可以在系统运行时得到一本拥有无限张具有同样信息特征页面的“书”.因此,在实际的系统设计中,纸更多体现的是一种个性化模板的概念.在系统运行时,用户并不直接在模板上输入信息,而是通过由模板自动生成的经个性化定制的空白页面来进行内容输入.在运行时,笔式用户界面软件体系结构中将增加页面表示层进行实际笔式交互信息的处理和数据记录.整个系统的场景连接呈现为多个模板的相互切换.为了管理和控制模板切换及之间的数据传递,我们增加了一个更高层次的实体进行模板管理.在各个层次之间由连接器进行连接,负责信息和消息传递.因此,我们提出笔式用户界面软件的体系结构风格表示,如图5所示.

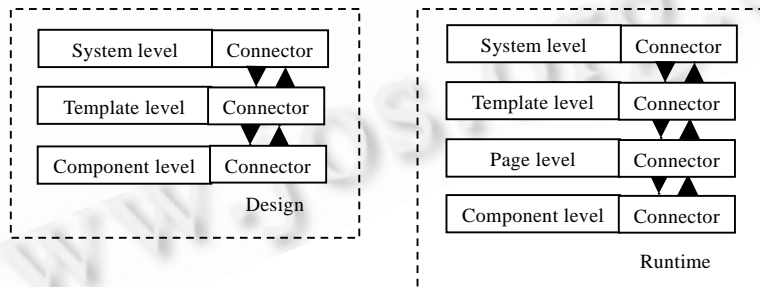


Fig.5 Architecture style of the pen-based UI software

图5 笔式用户界面软件的体系结构风格

### 3 应用实例的设计和评估

#### 3.1 应用实例设计

在目前的电子教学应用中,软件界面风格基本采用传统的图形用户界面风格,文字等信息输入主要依靠键盘和鼠标的输入设备组合.这样的信息输入方式不符合教师日常备课授课的习惯,使得这类软件无法得到教师们的认同,无法刺激教师使用软件.我们通过观察北京中小学16节语文、数学、英语和生物课,并结合教学实况

录像,对任课教师使用各种教学媒体(包括黑板、电脑投影机、实物投影仪)进行备课授课的意图、感受和态度进行访谈,考察教师对笔式教学媒体的使用状况、教学需求以及态度.大多数教师对于纸笔交互界面持较好的看法,认为手写笔会更方便、快捷,具有更多的互动性与人性化.

用户分析.该软件主要面向中小学教师.用户大量使用笔式交互完成日常工作,其工作根据环境的不同分为两个阶段:备课和授课.备课工作的核心为制作教案和课件,方法是使用笔进行教案信息输入或课件幻灯制作,形成教案文档.书写内容为文字、图形、公式、表格等多种笔迹信息;在授课过程中,教师与学生进行面对面的思想交流活动.教师通过黑板或幻灯进行笔迹书写开展教学活动.

用户使用环境分析.用户备课在桌面上完成,文档管理方式由个人习惯决定,笔迹直接书写在纸上.在讲课中教师使用粉笔在黑板上书写,引起学生注意,从而进行教学.根据用户对笔的使用环境的分析,以及用户分析结果,确定用户在备课时所使用的笔式输入设备为手写屏(板)类输入设备,而在授课阶段,用户则对使用电子白板类输入设备进行工作的认同程度较高.

应用场景分析.用户工作流程自然地划分为 3 个应用场景:文档管理、备课和授课.

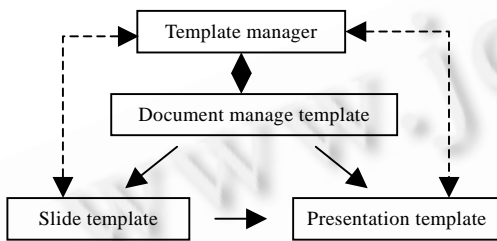


Fig.6 The scenario design of the PenOffice

图 6 PenOffice 的场景设计

软件体系结构设计(如图 6 所示).根据应用场景分析 PenOffice 具有 3 个场景模板:文档管理模板、备课模板和授课模板.根据用户日常工作的上下文,文档模板设计为:环境-室-工作台-文件夹逐层深入,以向用户呈现出一种符合日常工作的文档使用方式.确定文档后,该软件分为两个功能部分,即备课和授课两个功能应用.这两个应用之间具有一定的数据共享.在系统实现中,会存在顶级的模板管理层,用于负责 3 个模板之间的调用与数据传递.

特征分析过程是用户在设计人员指导下对已有的共有领域特征进行剪裁和选择的过程.设计人员的指导主要体现在帮助用户理解特征描述含义,对用户提出的设计方案进行实现技术可行性分析等方面.特征分析按照特征模型的层次逐层进行,并形成文档.根据特征描述文档,软件开发人员将其中领域公有特征与可复用信息库中的可复用信息(代码、组件、界面设计等)一一对应实现.对于特定应用的特定需求,在软件框架下,进行代码级的二次开发,完成原型的建立.

在设计过程中,始终由用户进行决策,在设计人员的指导下,用户对自己的设计有着清晰的了解,保证了用户概念模型与设计模型的映射一致.同时,从特征模型描述中可以看出,用户的大部分需求在笔式用户界面软件特征模型中得到了体现,能够进行软件复用.这就为原型的快速开发提供了条件,使用户可以更快地了解设计的实际运行效果,进行更为准确的评估.设计结果从实现功能的角度出发,用户需要系统提供的服务,除 Paper Manipulation 以外均在笔式用户界面软件模型中得以体现;从可用性角度来看,用户在设计的过程中根据个人的使用环境及书写习惯对软件的交互风格和表现细节进行决策,从而保证了用户概念模型和设计模型的一致.北京师范大学心理学系对该软件的可用性评估结果表明<sup>[25]</sup>,用户对 PenOffice 的设计结果比较满意.

### 4 总结

笔式用户界面软件开发的目的是满足用户可用性要求.通过对笔式用户界面软件特征模型的分析,对领域内应用进行了高层次的抽象描述.利用该模型将以用户为中心的思想引入到笔式用户界面软件的开发过程中,在用户与设计人员之间建立起良好的交流平台,使用户最终成为软件设计的主导,实现用户概念模型与设计模型及系统实现的良好映射.同时,利用特征模型可以建立起设计与可复用信息的对应,从而加快软件开发的迭代速度,提高软件的开发效率.最后,通过 PenOffice 的具体实例进行了上述说明.从该实例可以看出,通过 PUIDM 描述和设计人员的引导,用户意图能够正确地表现到软件设计中,设计实例的软件可用性评估结果良好.

领域模型的建立是一个不断补充、不断求精的过程.在未来的工作中,我们将逐步完善模型,并对模型中的各部分给出更为详尽的定义或描述.另外,我们也将试图向用户提供一个界面设计工具,使用户在设计过程中,



能够快速地了解设计的效果,明确设计思路,并自动生成特征描述文档,协助系统原型建立。

致谢 作者所在的 Pen UI Group 众多研究人员,包括王常青、李俊峰等人对本文的验证做了大量工作,在此表示感谢。

## References:

- [1] Dong SH, Wang J, Dai G, *et al.* Human-Computer Interaction and Multi Model User Interface. Beijing: Science Press, 1999 (in Chinese).
- [2] Preece J, Rogers Y, Sharp H. Interaction Design—Beyond Human-Computer Interaction. John Wiley & Sons, 2002.
- [3] Li Y. Research on pen\_based user interfaces, theory, technique and implementation [Ph.D. Thesis]. Beijing: Institute of Software, the Chinese Academy of Sciences, 2002. 49–56 (in Chinese with English abstract).
- [4] Stahovich T. SketchIT: A sketch interpretation tool for conceptual mechanical design. Technical Report, TR 1573, MIT AI Laboratory, 1996.
- [5] Landay JA. SILK: Sketching interfaces like crazy. In: Tauber MJ, ed. Proc. of the Human Factors in Computing Systems (Conf. Companion), ACM CHI 1996. New York: ACM Press, 1996. 398–399.
- [6] Lin J, Newman MW, Hong JI, Landay JA. DENIM: Finding a tighter fit between tools and practice for web site design. In: Turner T, ed. Proc. of the SIGCHI Conf. on Human Factors in Computing Systems, CHI 2000. New York: ACM Press, 2000. 510–517.
- [7] Lin J. Damask: A tool for early-stage design and prototyping of multi-device user interfaces. In: Shepherd D, ed. Proc. of the 8th Int'l Conf. on Distributed Multimedia Systems (2002 Int'l Workshop on Visual Computing). San Francisco, 2002. 573–580.
- [8] Davis RC, Landay JA. Informal animation sketching: Requirements and design. In: Proc. of the AAAI 2004 Fall Symp. on Making Pen-Based Interaction Intelligent and Natural. Washington, 2004. <http://guir.berkeley.edu/pubs/>
- [9] Mynatt ED, Igarashi T, Edwards WK, LaMarca A. Flatland: New dimensions in office whiteboards. In: Williams MG, ed. Proc. of the CHI'99 Human Factors in Computing Systems. New York: ACM Press, 1999. 346–353.
- [10] Alvarado CJ, Davis R. Resolving ambiguities to create a natural sketch based interface. In: Nebel B, ed. Proc. of the IJCAI-2001. Washington: Morgan Kaufmann Publishers, 2001. 1365–1371.
- [11] Keil M, Carmel E. Customer-Developer links in software development. Communications of the ACM, 1995,38(5):33–44.
- [12] Kujala S, Mäntylä M. Is user involvement harmful or useful in the early stages of product development? In: Tremaine M, ed. CHI 2000 Extended Abstracts. New York: ACM Press, 2000. 285–286.
- [13] Carroll JM. Human-Computer interaction: Psychology as a science of design. Int. J. Human-Computer Studies, 1997,46(4): 501–522.
- [14] Muller MJ. PICTIVE—An exploration in participatory design. In: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems: Reaching Through Technology. New York: ACM Press, 1991. 225–231.
- [15] Tudor LG, Muller MJ, Dayton T. A C.A.R.D. game for participatory task analysis and redesign: Macroscopic complement to PICTIVE. In: Ashlund S, ed. Proc. of the INTERCHI'93. Amsterdam: ACM Press, 1993. 51–52.
- [16] Arango G. A brief introduction to domain analysis. In: Berghel H, ed. Proc. of the 1994 ACM Symp. on Applied Computing. New York: ACM Press, 1994. 42–46.
- [17] Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS. Feature-Oriented domain analysis (FODA) feasibility study. Technical Report, CMU/SEI-90-TR-21 ESD-90-TR-222, 1990.
- [18] Tierney PJ, Ajila SA. The FOOM method—Modeling software product line in industrial settings. In: Proc. of the 2002 Int'l Conf. on Software Engineering Research and Practice (SERP 2002). 2002. <http://www.sce.carleton.ca/faculty/ajila/Publications-old.htm>
- [19] Chastek G, Donohoe P, Kang KC, Thiel S. Product line analysis: a practical introduction. Technical Report, CMU/SEI-2001-TR-001, Pittsburgh: Carnegie Mellon University, Software Engineering Institute, 2001. 1–42.
- [20] Norman DA. The Design of Everyday Things. New York: Basic Books, 1988.
- [21] Goodell H. The costs and benefits of end-user programming. 1997. <http://www.cs.uml.edu/~hgoodell/EndUser/cost-ben.htm>
- [22] Krasner GE, Pope ST. A cookbook for using the model-view-controller user interface paradigm in Smalltalk-80. Journal of Object Oriented Programming, 1988,1(3):26–48.
- [23] Balbo S, Coutaz J, Salber D. Towards automatic evaluation of multimodal user interfaces. Knowledge Based System, 1993,6(4): 201–208.
- [24] Tian F. Research on Post-WIMP software interface [Ph.D. Thesis]. Beijing: Institute of Software, the Chinese Academy of Sciences, 2003. 35–45 (in Chinese with English abstract).
- [25] Wu G, Tian F, Li J, Mu S, Dai GZ. Usability evaluation of a new pen-based user interface paradigm computer engineering and design. Computer Engineering and Design, 2005,26(3):570–572, 610 (in Chinese with English abstract).

## 附中文参考文献:

- [1] 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.  
 [3] 栗阳.笔式用户界面研究——理论、方法和实现[博士学位论文].北京:中国科学院软件研究所,2002.  
 [24] 田丰.Post-WIMP 软件界面研究[博士学位论文].北京:中国科学院软件研究所,2003.  
 [24] 吴刚,田丰,李杰,牟书,戴国忠.一种新的笔式交互范式的可用性评估.计算机工程与设计,2005,26(3):570-572,610.



秦严严(1977 - ),男,北京人,博士生,主要研究领域为人机交互技术,笔式计算.



王晓春(1976 - ),男,博士生,主要研究领域为人机交互技术,笔式计算.



田丰(1976 - ),男,博士,主要研究领域为人机交互技术,虚拟现实.



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.

\*\*\*\*\*

## 《软件学报》网格计算专刊 征文通知

本专刊瞄准与网格技术及其应用相关的基础性、前瞻性、战略性的重大理论和关键技术问题,以及网格技术与其他学科领域的交叉性,同时为了加强网格界和各应用领域专家之间的合作与交流,共同促进技术进步,推动网格技术在国民经济重大领域的应用普及,提高该领域信息化程度。本专刊面向全国征集论文,欢迎广大学者、专家、工程技术人员积极投稿,现将专刊论文征集的有关事项通知如下:

专刊题目:网格计算

特约编辑:金海(华中科技大学)、郑纬民(清华大学)

### 一、征文范围

#### 1. 网格基础理论研究

主要包括网格核心协议和核心服务的研究,如服务管理、网格执行管理、元数据管理、网络安全、数据管理等。

#### 2. 网格应用理论研究

针对面向科学活动和信息服务的典型应用领域,用于研究网格对这些领域提供关键技术的支持及改造,以及问题解决环境和使用门户。

#### 3. 网格技术与其他前沿技术的交叉研究

主要研究网格技术与前沿信息技术,如:语义网、对等计算、普适计算、自主计算、服务计算等的交叉结合。

论文内容并不仅限于此,所有与网格计算相关的内容均可投稿。

### 二、投稿要求

1. 投稿方式:采用“软件学报在线投稿系统”(http://www.jos.org.cn)投稿。请在投稿时,在备注栏中注明“网格计算专刊投稿”字样。

2. 稿件格式:参照《软件学报》论文格式(学报网站上提供了论文模版,可下载)。

3. 投稿文章未在正式出版物上发表过,也不在其他刊物或会议的审稿过程中,且不存在一稿多投现象;保证投稿文章的合法性(无抄袭、剽窃、侵权等不良行为)。

4. 其他投稿须知请参阅《软件学报》投稿指南 http://www.jos.org.cn/directory.htm

5. 投稿作者需提交投稿声明;专刊投稿文章不收审理费。录用刊发文章将收取软件学报标准版面费。发表之后,将按软件学报标准支付稿酬,并赠送样刊及单行本。

### 三、重要时间

截稿日期:2006年6月10日

录用通知发出时间:2006年8月10日

录用修改稿提交日期:2006年9月10日

出版日期:2006年第11期