

基于 PCTBTA 任务模型的用户界面开发方法*

李娟妮^{1,2}, 华庆一¹, 吴昊¹, 陈锐¹, 苏荟¹, 周筠³

¹(西北大学 信息科学与技术学院 计算机科学与技术系, 陕西 西安 710075)

²(西安石油大学 理学院, 陕西 西安 710065)

³(陕西师范大学 教育学院, 陕西 西安 710062)

通讯作者: 华庆一, E-mail: huaqy@nwu.edu.cn



摘要: 为了适应普适计算环境中用户、设备、使用环境和开发平台的多样性,基于模型的方法被应用于用户界面开发过程中,试图在抽象层次上描述界面,通过模型转换,使其适用于不同的平台.然而,由于目前基于模型的用户界面开发方法(model-based user interface development,简称MBUID)中所采用任务模型的限制性,致使生成的界面难以满足动态环境下用户的可用性需求.提出一种基于任务模型的用户界面开发框架,旨在建模和生成有效、高效、令用户满意的用户界面.在可用性方面,为了准确描述普适计算环境中用户任务,提出一种基于感知控制理论的任务分析方法(perceptual-control-theory-based task analysis,简称PCTBTA),将使用上下文信息引入到任务分析过程中,并且在较高的抽象层次上反映交互的内容,给可用性设计提供任务空间;在技术方面,为PCTBTA任务模型向界面模型的转换提供技术支持.最后,通过实例说明所提出方法的可行性,并通过与其他方法在可用性和性能方面的比较,表明该方法的有效性.

关键词: 任务模型;感知控制理论;可用性;基于模型的用户界面开发

中图法分类号: TP311

中文引用格式: 李娟妮,华庆一,吴昊,陈锐,苏荟,周筠.基于 PCTBTA 任务模型的用户界面开发方法.软件学报,2018,29(12): 3692-3715. <http://www.jos.org.cn/1000-9825/5432.htm>

英文引用格式: Li JN, Hua QY, Wu H, Chen R, Su H, Zhou Y. Method for user interface development with perceptual control theory based task analysis. Ruan Jian Xue Bao/Journal of Software, 2018,29(12):3692-3715 (in Chinese). <http://www.jos.org.cn/1000-9825/5432.htm>

Method for User Interface Development with Perceptual Control Theory Based Task Analysis

LI Juan-Ni^{1,2}, HUA Qing-Yi¹, WU Hao¹, CHEN Rui¹, SU Hui¹, ZHOU Yun³

¹(Department of Computer Science and Technology, School of Information Science and Technology, Northwest University, Xi'an 710075, China)

²(School of Science, Xi'an Shiyou University, Xi'an 710065, China)

³(School of Education, Shaanxi Normal University, Xi'an 710062, China)

Abstract: Nowadays, a number of methods on model-based user interface development (MBUID) have been applied to deal with the diversity of users, devices, environments, and development platforms in a pervasive computing environment. In general, those methods attempt to specify a user interface once on an abstract level, and to run anywhere by use of model transformation. Due to the limitation of task model used in the current MBUID methods, however, it is still an open question whether the generated user interfaces can meet

* 基金项目: 国家自然科学基金(61272286); 高等学校博士学科点专项科研基金(20126101110006); 陕西省教育厅科研计划(16JK1602)

Foundation item: National Natural Science Foundation of China (61272286); Specialized Research Fund for the Doctoral Program of Higher Education of China (20126101110006); Scientific Research Program Funded by Shaanxi Provincial Education Department (16JK1602)

收稿时间: 2017-07-31; 修改时间: 2017-08-13, 2017-09-14; 采用时间: 2017-10-16

usability requirements in a divergent context of applications. In this paper, a task model based user interface development framework is proposed for modeling and implementing effective, efficient and satisfactory user interfaces. In order to cope with the usability requirements, a novel perceptual control theory based task analysis (PCTBTA) method is presented to specify the user tasks in a divergent environment, in which the context information is introduced into the task analysis process, and the interaction content is reflected at a higher level of abstraction, providing the task space for usability design. For model transformation, a method is provided for converting PCTBTA task model into a variety of interface models. Finally, a case study is provided to illustrate the feasibility of the proposed method, and the effectiveness of the method is demonstrated by comparing it with other methods in terms of availability and performance.

Key words: task model; perceptual control theory; usability; model-based user interface development

随着各种类型的计算设备大量涌入消费者市场,信息技术已经渗入人们生活的物理空间,当前的计算环境已由个人计算演化为普适计算.用户从具有专业知识和技能的工作者扩展到不同年龄、不同背景的一般大众;他们的活动也由办公桌面延伸到社会的各个领域.在普适计算环境中,人们希望随时随地、无所不在地获取信息和计算服务,用户的非专业性、使用上下文的动态性、任务的不确定性、设备和开发平台的多样性给用户界面开发带来了很大的挑战.

- (1) 可用性方面,如何为不同设备开发同样方便易用,并具有一致用户体验的界面;如何为不同类型的用户在不同的使用环境中提供不同的界面;如何获取环境中的隐式数据,使桌面计算模式下的显式交互转换为普适计算模式下的隐式交互;
- (2) 在技术方面,开发者如何有效地为同一应用在不同技术环境(设备、交互通道、开发平台等)中分别设计实现不同的用户界面;如何有效管理多个版本界面开发的工程复杂性.

为了减少开发者的劳动负荷,基于模型的方法被引入用户界面的开发过程中.其主要思想是在较高的抽象层次上描述用户界面,然后根据不同的使用上下文(用户、平台、环境),将抽象描述转化为适用于特定设备的界面,以适应普适计算环境中技术和设备的异构性^[1].在可用性方面,为了让这种基于模型和模型转换方法生成的界面能够被用户所接受,MBUID 方法将用户任务模型作为起点,通过对用户任务进行分析、建模获取系统的功能性和可用性需求;在技术方面,对界面模型在不同的抽象层次上进行划分,如平台无关模型(platform independent model,简称 PIM)、平台相关模型(platform specific model,简称 PSM)等,借助模型之间的转换和映射方法,将 PIM 精化为特定平台相关模型 PSM.最后,通过特定的开发语言将 PSM 转换为具体平台的代码.

然而,当前对 MBUID 的研究中工作大多集中于技术方面,而忽略了可用性方面,这使得采用 MBUID 方法开发的用户界面在适应开放的社会环境需求方面存在一些不足.

- 首先,当前的 MBUID 研究中所采用的任务模型构建方法多在桌面系统盛行时代甚至字符界面时代产生,使用这些方法建模普适计算时代的用户任务已经难以胜任.随着信息技术与人们生活的日益结合,用户任务的范围从静态的办公业务拓展到复杂多变的娱乐和生活领域.设备的便携性使得软件的应用环境从封闭空间转移到开放空间,任务执行过程中使用上下文的动态变化给任务的执行带来了许多不确定性因素,用户任务呈现出多样性、容易受到外界环境影响的特点.在这种趋势下,需要任务模型能够描述外界环境的信息,并且根据不同的使用上下文执行不同的用户任务序列,最终映射成为不同的用户界面.然而,传统的任务模型反映封闭环境中任务世界的维度和体系构建规则,仅描述了用户与系统之间的交互,没有考虑外界环境的影响.
- 其次,由于 MBUID 方法中所采用任务模型的局限性,基于此类任务模型所生成的界面虽然在呈现方面(布局、风格等)和实现方面(不同的开发语言)具有一定的适应性^[2-5],但它们不能根据使用环境和用户需求的不同动态改变.目前,MBUID 中所采用的模型之间的转换和映射技术都是以当前任务模型为基础,根据相应的规则将任务模型中描述的对象映射为界面对象,通过任务之间的关系提取用户与系统之间的对话,并建立界面之间的导航关系.在界面开发过程中,对象、选择策略和行为过程是在设计阶段预先定义的,不能根据使用环境的不同而动态改变.

本文提出一种基于 PCTBTA 任务模型的用户界面开发框架,旨在探索生成高质量用户界面的建模方法和

技术.在可用性方面,为了更好地描述动态环境中,用户、系统和周围环境的交互过程,提出一种 PCTBTA 任务模型,它扩展传统任务模型的描述范围,将使用上下文信息引入到任务分析过程中,并且在较高的抽象层次上(接近目标层次)反映交互的内容,给可用性设计提供任务空间;在技术方面,为了使这种概念化方法更好地应用于用户界面开发过程中,提供了从 PCTBTA 模型向界面模型转换的技术支持.本研究对提高交互式系统的可用性、降低开发过程中的沟通成本、提高用户界面的开发效率具有重要的科学研究意义和推广应用前景.

本文第 1 节对基于 PCTBTA 任务模型的用户界面开发框架进行介绍,描述 PCTBTA 任务建模过程和任务模型向界面模型的转换过程.第 2 节针对普适计算环境中用户任务特征,提出一种 PCTBTA 任务建模方法及支持 PCTBTA 方法的任务描述符号.第 3 节具体阐述从 PCTBTA 任务模型到界面模型的转换过程.第 4 节以“校园导航系统”为例,说明利用 PCTBTA 方法分析和建模用户任务的过程,并且详细描述了从 PCTBTA 任务模型到界面模型的转换过程.第 5 节通过与其他方法对比评估,表明该方法的有效性.第 6 节介绍了与本文相关的工作.最后总结全文和未来工作.

1 基于 PCTBTA 任务模型的用户界面开发框架

基于 PCTBTA 任务模型的用户界面开发方法是在 CAMELEON 参考框架^[6]的基础上,为了满足普适计算环境中用户可用性需求的改变,对现有 MBUID 方法的改进.具体框架如图 1 所示,主要包括两部分内容:PCTBTA 任务模型的构建和 PCTBTA 任务模型至界面模型的转换.

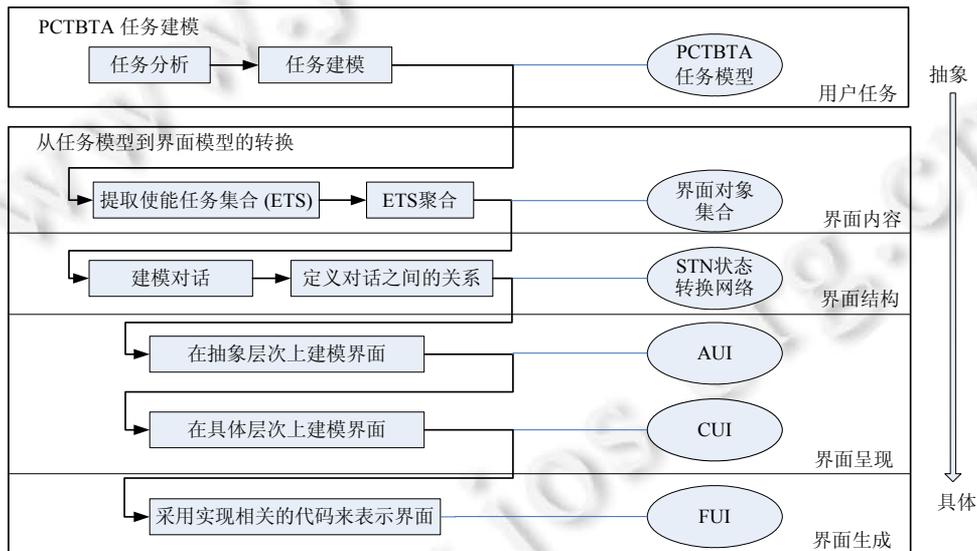


Fig.1 User interface development framework based on the PCTBTA task model

图 1 基于 PCTBTA 任务模型的用户界面开发框架

(1) PCTBTA 任务模型(见第 2 节).

在普适计算环境中,用户任务与环境信息的关联非常紧密,用户输入不再是任务执行的唯一驱动,实时感知的环境信息也将作为触发和改变任务序列的因素.在此应用背景下,为了让系统能更好地适应多变的用户需求,任务模型应该增添动态的属性.与其他任务模型相比较,PCTBTA 任务模型的特点主要表现在:

- 将感知控制理论(perceptual control theory,简称 PCT)^[7]作为任务分析和建模过程的心理学基础.

PCTBTA 方法认为:在动态环境中,用户是以一种感知控制的方式与外界环境进行交互活动,根据目标来感知环境信息,并将其与自己的目标作比较,如果不一致则采取相应的行为作用于环境变量,使其达到目标状态.因此,PCTBTA 方法分析用户任务时,假定所有的任务都包括控制,将用户执行任务看作用户通过控制系统界面

上的交互对象,使系统达到参考状态的过程.在任务分析时,始终以用户目标为导向,收集并组织任务相关的信息;通过分析外界环境信息,引导设计者在设计初始阶段考虑各种可能影响目标状态的因素,为开发高可用性的产品提供支持.

- 感知控制任务单元(perceptual control task unit,简称 PCTU).

在普适计算中,开放的应用环境使得用户的时间、兴趣和行为都呈现碎片化状态,用户不适合从事复杂任务,而更加希望利用碎片时间完成简单的任务.PCTBTA 方法允许设计者将任务分析过程中反复出现,或能完成某一具体上下文中用户目标的任务序列集合定义为一个任务单元(PCTU),每个 PCTU 都可以作为独立的描述原语与其他任务建立联系,完成任务目标.相对于传统的静态环境中大粒度的任务结构,这种小型的任务构件、灵活的组织关系更适应于描述动态环境中的任务.

- 任务组织方式.

PCTBTA 方法中,任务组织方式可以分为具体化和抽象化:具体化过程指通过高级层次上的抽象用户任务进行逐步求精,最终由若干个 PCTU 来实现;反之,抽象化过程指通过 PCTU 之间建立联系,能够实现更高层次的用户目标.

(2) PCTBTA 任务模型向界面模型的转换(见第 3 节).

对于一个有效的概念化方法,至少需要满足两方面的需求:既易于描述用户的抽象,所建立的抽象又易于映射为面向系统的模型.因此,任务模型除了能够描述用户与系统的交互过程,所描述的内容也应该能够映射为用户界面模型.任务模型向界面模型的转换过程主要包括以下 4 个方面:

- 界面内容:根据 PCTBTA 任务模型中任务之间的暂态关系,提取每个界面所包含的内容,将其定义为使能任务集合(enabled task set,简称 ETS)^[8].ETS 指同一时刻可以被激活的任务集合,即逻辑上可以呈现在同一个用户界面上的任务集合.ETS 所包含的对象采用抽象方式描述,不涉及具体的呈现方式;
- 界面结构:通过任务模型中对象之间的执行顺序,建立对话之间的导航关系以及致使对话转换的动态行为.采用状态转换网络(state transition network,简称 STN)表示对话状态之间的改变^[9];
- 界面呈现:根据每个对话中涉及界面元素的类型和属性以及对话之间的转换关系,实现从对话模型到抽象界面模型(abstract user interface,简称 AUI)的转换.在此基础上,通过添加不同软硬件平台属性,完成从 AUI 到具体用户界面(concrete user interface,简称 CUI)的转换;
- 界面生成:采用不同的开发语言,生成适应于特定平台的最终用户界面(final user interface,简称 FUI).

2 PCTBTA 任务模型

在静态的办公环境中,用户的行为空间是封闭的,相互干扰较少,有利于用户采用一种结构化的认知模型去独立的思考,进行复杂的工作,任务模型的范围是用户与系统之间的交互过程.而在多变的普适计算环境中,用户的行为空间是开放的,有很多不确定的因素会影响任务的执行;任务的描述范围也扩展到用户、系统和外界环境之间的交互过程.在执行任务的过程中,环境信息的改变将引起用户生理或心理效应,这种效应会以一种外在的行为方式在用户与系统交互的过程中表现出来.

PCTBTA 方法将 PCT 作为引导分析过程的理论基础,认为用户在使用软件系统完成目标时,与界面的交互过程也是一个感知控制过程.在使用系统时,用户将系统的当前状态与期望的目标状态相比较,并据此通过界面完成一系列交互行为.当外界环境的变化影响到任务目标完成时,针对不同的影响因素,用户需要执行不同的行为序列,最终实现任务目标.因此,在 PCTBTA 方法中并不存在一个达到用户目标的“正确”任务序列,而是考虑了各种影响因素的任务序列集合.

2.1 PCTBTA 任务建模

PCTBTA 任务建模方法的主要思路(如图 2 所示)是:通过分析用户的任务世界,获取用户、环境和任务描述的相关信息,为引导任务建模提供数据支持;采用抽象建模技术对分析过程所获取的数据进行建模,所生成的任务模型刻画了用户、系统和环境之间交互的任务空间.

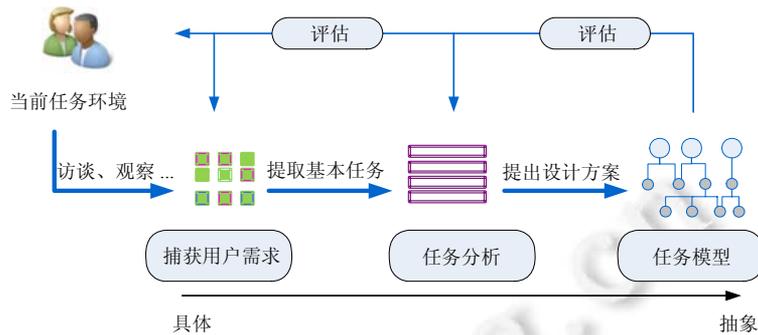


Fig.2 PCTBTA task modeling process

图2 PCTBTA 任务建模过程

该过程包含3个主要活动:获取用户需求、任务分析和任务建模.

- (1) 获取用户需求:采用观察、问卷、访谈等方法获取与用户需求相关的数据,如用户特征、用户意图和目标、当前用户执行任务的过程、执行任务的环境等;
- (2) 任务分析:对所收集的数据进行分类、整理,并以非形式化的方式描述用户在特定环境中实现目标可能需要执行的任务集合.PCTBTA 任务分析过程主要包括:
 - 第1步:初步分析.
 - 解释所收集的数据:识别用户(类型、特征等)、用户目标、任务执行的环境;
 - 抽象层面分析:通常情况下,完成特定目标用户需要执行的任务;
 - 具体层面分析:对任务进一步求精,分析执行任务用户需要控制的界面元素和完成任务时用户感知到的界面元素及其理想状态;
 - 第2步:延展分析.
 - 分析外部环境中可能对任务执行产生影响的因素.影响因素分为两类:一类是阻碍目标实现的因素(如原来可用的网络服务突然中断等);另一类是促进目标实现的因素(如通过智能设备传感器获取的环境信息,可以代替用户的输入等);
 - 分析不同影响因素下的任务序列:一方面,干扰因素存在时,采用什么样的操作能够解决异常状况;另一方面,有助于目标实现的因素可用时,如何引导用户更加便捷地实现目标;
 - 第3步:评估.通过直接与用户交流或根据所收集的需求信息对任务分析的结果进行评估,根据评估结果进行迭代设计,保证在逻辑上和技术上的正确性;
- (3) 任务建模:任务建模在任务分析之后,通过构建任务模型,精确地描述完成用户目标各种可能的任务、任务属性及任务之间的关系.PCTBTA 任务建模主要包括:
 - 第1步:建模任务的结构.
 - 系统目标层面:从整个软件系统设计或组织的层面考虑系统所包含的功能、子功能,此过程是从顶向下的建模过程,将实现高层目标的抽象任务逐步具体化;
 - 用户目标层面:在普适计算环境中,不同用户在不同的使用上下文中有不同的目标.对用户任务建模时,PCTBTA 方法允许设计者将与特定用户目标相关的任务序列封装在一个 PCTU 中,它可以作为独立的任务描述原语与其他任务结合完成高层抽象任务;
 - 第2步:进行任务分配.PCTBTA 方法中,设计者需要根据可用的技术辅助手段(如网络服务等),对任务进行重新分配和重组.一方面,试图将用户执行的任务由机器自动执行,从而简化任务操作,减轻用户的脑力负担;另一方面,要保证用户对系统的控制.依据执行代理角色的不同,可将任务分为用户任务、交互任务、感知任务和系统任务,具体定义见表1.

Table 1 Conceptual notation in the PCTBTA task modeling process

表 1 PCTBTA 任务建模过程中包含的概念符号

符号	名称	语义
	用户任务	由用户完成的任务,一般指用户的认知行为
	系统任务	由系统内部执行的任务,处理的数据来源于系统本身
	用户交互任务	属于交互任务类型,特指用户与系统交互的任务
	感知任务	指通过感知接口获取信息的任务,感知接口是系统与外部环境信息交流的通道
	抽象任务	是一种复合任务,由其他不同类型的任务组成
	外部影响	一切可以影响用户目标实现的因素,是对外部环境信息的描述
	任务单元	封装了特定环境中特定用户完成既定目标所执行的任务序列

- 第 3 步:建模任务(任务的描述维度).
 - 用户目标:完成任务后,用户希望系统所呈现的状态.目标可以通过执行一个或多个任务来实现;
 - 显式变量:在执行任务过程中,用户界面呈现的物理/信息变量,包括可被用户操作的可控制变量、可被用户感知的仅输出变量和多个变量的容器.可控制变量中包括选择、编辑、激活某个事件的控件和导航控件;仅输出变量是系统呈现给用户的信息,没有交互功能,如文本信息、警示信息、反馈信息、音频、视频等;
 - 外部影响:指所有对用户目标实现有影响的因素.影响因素与使用上下文相关,可以来源于环境、技术或者社会关系等.如果这个影响是促进目标快速实现的因素,称为兴奋效应的影响;如果是阻碍目标实现的,称为抑制效应的影响;
 - 操作:指为了使显示变量的状态达到用户理想的目标状态需要执行的一系列活动,以及当显示变量受到影响因素的干扰偏离目标状态时,能够减小或消除偏差所需要执行的活动;
- 第 4 步:建模任务之间的关系.在 PCTBTA 方法中,任务之间的关系可以分为 3 类.
 - 同一层次之间的任务,即兄弟任务(sibling task)之间的关系.PCTBTA 方法继承了 CTT 中定义的暂态关系^[10],它是建立在 LOTOS 基础上的一组描述事件驱动和状态转变的时序关系符号;
 - 相邻层次的任务,高层任务称为父任务(parent task),低层任务称为子任务(children task);
 - 作为外部影响因素存在的任务之间的关系(详细定义见表 2).

Table 2 Relationship notation between factors in the PCTBTA task modeling process

表 2 PCTBTA 任务建模过程中影响因素之间的关系符号

关系类型	符号	关系名称	语义
作为影响因素的任务之间的关系 (F_1, F_2 为外部影响因素)	$F_1()F_2$	无序选择	F_1, F_2 无先后顺序,被选中的比例相当
	$F_1(*)F_2$	有序选择	在做选择时考虑优先级,先选择 F_1 ,如信息没有成功获取,再选择 F_2
	$F_1 \Rightarrow F_2$	完全替代	在运行过程中,不执行 F_2 ,只执行 F_1
	$F_1 \nRightarrow F_2$	非完全替代	在运行过程中,不执行 F_2 ,只执行 F_1 ,但当 F_1 的信息获取不到时,再来执行 F_2

- 第 5 步:评估.通过用户参与评估或者对照数据收集、任务分析的结果评估任务模型,根据评估结果进行迭代设计,保证任务描述的完整性和一致性.

2.2 PCTBTA任务表示符号

为了准确记录 PCTBTA 方法的分析结果,定义了一套带有语义的图标式符号,试图更好地刻画动态环境中的用户任务.一方面,能够从用户的角度去描述用户完成目标的任务过程;另一方面,能够将这种描述映射为系统构建过程中的对象,从而保证软件开发过程中,用户、设计者、开发者之间的无障碍交流,提高了产品的开发效率.构建 PCTBTA 任务模型过程中相关概念的代表符号见表 1.

3 PCTBTA 任务模型向界面模型的转换

从 PCTBTA 任务模型到界面模型的转换过程如图 3 所示.

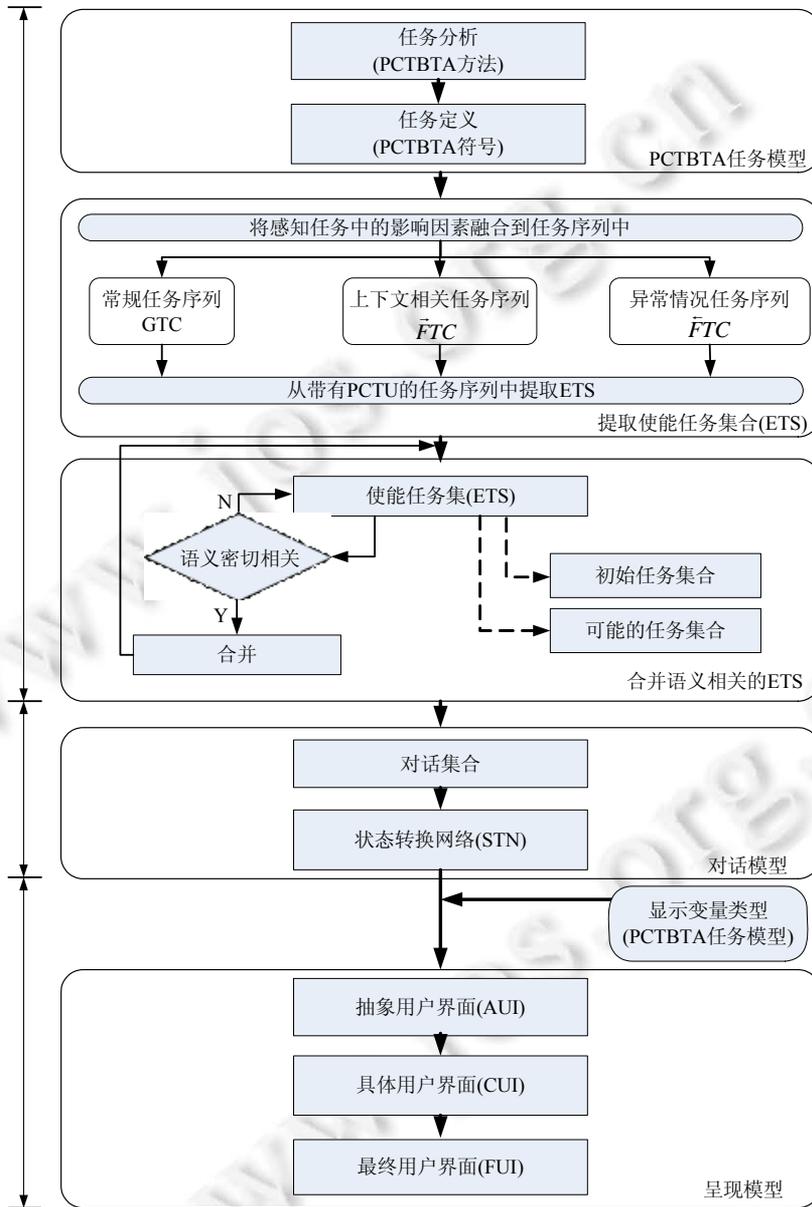


Fig.3 Transformation process from PCTBTA task model to persentation model

图 3 PCTBTA 任务模型到界面呈现模型的转换过程

从 PCTBTA 任务模型到界面模型的转换过程主要分为以下 3 个活动.

- (1) 从 PCTBTA 任务模型中提取 ETS.通过 PCTBTA 任务模型中定义的任务之间的暂态关系,识别哪些任务应该呈现在同一界面上;
- (2) 从 ETS 到对话模型的转换过程.包括 ETS 到对话的映射(在抽象层次上表明界面的内容)和对话之间

导航关系的建立;

- (3) 从对话模型到界面呈现模型的转换过程.根据每个对话中所包含的显示变量的类型以及对话之间的转换关系,实现从对话模型到抽象界面模型的转换.在此基础上,逐步增加设备、交互通道、开发平台等属性,生成不同技术环境中所运行的界面.

3.1 从PCTBTA任务模型中提取ETS

本节介绍了从 PCTBTA 任务模型中提取 ETS 的过程,为了更加准确地描述这个过程,给出以下定义.

定义 1. 任务集合 T 是一个任务的有限集合, $T = \{t_1, t_2, \dots, t_n\}$; 任务之间的关系可以表示为 $t_i \xrightarrow{R} t_j | r$, 其中, R 表示任务之间的关系集合, r 表示任务 t_i 与任务 t_j 之间的关系.

定义 2. 感知任务集合表示为 $\tilde{T}, \tilde{T} \subset T; t' \in \tilde{T} | t$ 表示任务 t 是感知任务 t' 的缺省任务, 即当没有感知到其他匹配信息时所执行的任务.

若用户在通过地图软件获取自己当前位置的任务中, $currentlocation \in \tilde{T} | t = user_input$ 表示任务 $currentlocation$ 属于感知任务, 当感知不到其他外界影响因素时, 它的缺省任务是 $user_input$, 即由用户自己输入当前位置信息.

定义 3. 任务的外界影响因素集合表示为 $F, F = \bar{F} \cup \bar{F}$. 其中: \bar{F} 为兴奋效应的影响因素集合, $\bar{F} = \{\bar{F}_1, \bar{F}_2, \dots, \bar{F}_n\}$, \bar{F}_i 表示兴奋因素; \bar{F} 为抑制效应的影响因素集合, $\bar{F} = \{\bar{F}_1, \bar{F}_2, \dots, \bar{F}_n\}$, \bar{F}_i 表示抑制因素.

定义 4. 将通常情况下完成用户目标的任务序列称为常规任务序列, 记作 GTC ; 包含兴奋效应影响因素的任务序列称为上下文相关任务序列, 记作 \bar{FTC} ; 应对抑制效应影响因素的任务序列称为异常情况的任务序列, 记作 \bar{FTC} .

定义 5. $T(id_PCTU)$ 表示某个具体的 PCTU 所包含的任务集合, 其中, id 为 PCTU 的标识.

下文针对从 PCTBTA 任务模型提取 ETS 中的每个步骤进行具体描述(如图 3 所示).

(1) 第 1 步: 将感知任务中的影响因素融合到任务序列中.

在文献[11]中, 我们给出了详细的算法, 主要的过程为: 通过分析 PCTBTA 模型中感知任务的兴奋效应影响因素及它们之间的关系, 得到上下文相关任务序列 \bar{FTC} ; 通过分析抑制效应因素, 将解决异常情况的任务序列放入异常任务序列集合 \bar{FTC} .

过程. 将感知任务包含内容合并到任务序列中

输入: 感知任务 \tilde{T} ;

输出: 外部影响因素作用下的任务序列集合.

步骤 1. 查找对感知任务产生兴奋效应的影响因素;

步骤 2. 根据这些产生兴奋效应的影响因素与缺省任务之间的关系以及关系优先级, 将兴奋因素转换为完成任务目标的一般任务, 其中, PCTU 单元作为一种特殊的抽象任务参与转换. 具体的方法如下.

- 兴奋因素与缺省任务之间为完全替代关系, 移除缺省任务;
- 兴奋因素与缺省任务之间为非完全替代关系, 则生成同层非完全替代关系的任务;
- 兴奋因素与缺省任务之间为无序选择关系, 则生成同一层选择关系的任务;
- 兴奋因素与缺省任务之间为有序选择关系, 相同优先级的任务生成同一层的选择关系; 不同优先级, 根据优先级的高低, 依次生成非完全替代关系的任务;

步骤 3. 重复步骤 1 和步骤 2, 直至所有的兴奋因素都转换为一般任务, 此时, 生成正常情况下, 考虑外部环境中不同兴奋效应影响后的任务序列 \bar{FTC} ;

步骤 4. 查找对感知任务产生抑制效应的影响因素;

步骤 5. 将找到的解决异常情况的任务序列归纳到 \bar{FTC} 集合中;

步骤 6. 重复步骤 4 和步骤 5, 直至考虑了所有的抑制因素.

通过以上步骤, 可以生成考虑外界影响因素所有可能的任务序列.

(2) 第 2 步:从带有 PCTU 的任务序列中提取 ETS.

如上文所述,ETS 指可以被同时激活的任务集合,逻辑上,同一个 ETS 中的任务可以显示在同一个用户界面上.因此,对 ETS 的提取是从任务模型到界面呈现模型转换过程中的关键环节.

针对采用 CTT 符号表示的任务模型到 ETS 的提取,在文献[8]中已经定义了相关规则.在 PCTBTA 任务模型中,除了包含 LOTOS 暂态关系外,还定义外部影响因素之间的关系.本文采用文献[12]中的方法对这些关系进行形式化的描述,为从任务模型中提取 ETS 提供理论基础(见表 3).

Table 3 Executive logic of influence factor relationship

表 3 影响因素关系的执行逻辑

非完全替代关系 $\left(t_1 \xrightarrow{R} t_2 \mid r = \text{replace_incomplete} \right)$	
	$\begin{aligned} &Invoked(\text{Root}) \rightarrow Invoked(t_1) \wedge \neg Invoked(t_2) \\ &Invoked(t_1) \rightarrow Performed(t_1) \\ &Perfored(t_1) \rightarrow Performed(\text{Root}) \\ &\neg Perfored(t_1) \rightarrow Invoked(t_2) \\ &Invoked(t_2) \rightarrow Performed(t_2) \\ &Performed(t_2) \rightarrow Performed(\text{Root}) \end{aligned}$
完全替代关系 $\left(t_1 \xrightarrow{R} t_2 \mid r = \text{replace_complete} \right)$	
	$\begin{aligned} &Invoked(\text{Root}) \rightarrow Invoked(t_1) \wedge \neg Invoked(t_2) \\ &Invoked(t_1) \rightarrow Performed(t_1) \\ &Perfored(t_1) \rightarrow Performed(\text{Root}) \end{aligned}$
无序选择关系 $\left(t_1 \xrightarrow{R} t_2 \mid r = \text{choice_unorder} \right)$	
	$\begin{aligned} &Invoked(\text{Root}) \rightarrow (Invoked(t_1) \wedge \neg Invoked(t_2)) \vee (Invoked(t_2) \wedge \neg Invoked(t_1)) \\ &Invoked(t_1) \rightarrow Performed(t_1) \\ &Invoked(t_2) \rightarrow Performed(t_2) \\ &Performed(t_1) \vee Performed(t_2) \rightarrow Performed(\text{Root}) \end{aligned}$
有序选择关系 $\left(t_1 \xrightarrow{R} t_2 \mid r = \text{choice_order}, \text{Priority}(t_1) > \text{Priority}(t_2) \right)$	
	$\begin{aligned} &Invoked(\text{Root}) \rightarrow Invoked(t_1) \wedge \neg Invoked(t_2) \\ &Invoked(t_1) \rightarrow Performed(t_1) \\ &Perfored(t_1) \rightarrow Invoked(t_2) \\ &Invoked(t_2) \rightarrow Performed(t_2) \\ &Performed(t_1) \wedge Performed(t_2) \rightarrow Performed(\text{Root}) \end{aligned}$

表 3 中,Invoked 表示触发状态,¬Invoked 表示未被触发状态,Interrupted 表示中断状态,¬Interrupted 表示未被中断状态,Performed 表示完成状态,¬Performed 表示未被完成状态.

根据以上关系之间的执行顺序的描述,下文给出了从带有 PCTU 的任务序列中提取 ETS 的方法.其中: first(PCTU)指 PCTU 的常规任务序列或上下文相关序列中第 1 个被执行的任务;Body(PCTU)指除了 first(PCTU)以外,任务单元中所包含的其他任务.

- 预处理:如果 PCTU 中有感知任务,根据上一步骤将外部影响因素转换为普通任务;
- $PCTU \xrightarrow{R} t \mid r = \text{choice_unorder}$, first(PCTU)与 t 属于同一个 ETS,Body(PCTU)与 t 属于不同的 ETS;
- $PCTU_1 \xrightarrow{R} PCTU_2 \mid r = \text{choice_unorder}$, first(PCTU₁)与 first(PCTU₂)属于同一个 ETS,Body(PCTU₁)与 Body(PCTU₂)属于不同的 ETS;
- $PCTU \xrightarrow{R} t \mid r = \text{choice_order}$, PCTU 与 t 属于不同的 ETS;
- $PCTU_1 \xrightarrow{R} PCTU_2 \mid r = \text{choice_order}$, PCTU₁ 与 PCTU₂ 属于不同的 ETS;

- $PCTU \xrightarrow{R} t | r = \text{replace_complete}$ 或 $PCTU \xrightarrow{R} t | r = \text{replace_incomplete}$, $PCTU$ 与 t 属于不同的 ETS;
- $PCTU_1 \xrightarrow{R} PCTU_2 | r = \text{replace_complete}$ 或者 $PCTU_1 \xrightarrow{R} PCTU_2 | r = \text{replace_incomplete}$, $PCTU_1$ 与 $PCTU_2$ 属于不同的 ETS;
- PCTU 中所涉及的异常序列都加入到 $\bar{F}TC$ 中.

(3) 第 3 步:ETS 的聚合

如果每个 EST 都对应一个用户界面,很可能出现过深的界面层级关系,这将分散用户的注意力,致使他们在使用软件的过程中迷失方向.因此在转换的过程中,对那些经常出现且有信息交换的任务,即使经过逻辑推理得出它们不属于同一个 ETS,但在语义关系上它们是紧密相连的,对于这些任务可以被融合成为一个新的 ETS,在同一个界面上呈现.

3.2 从 ETS 到对话模型的转换

对话模型用来刻画用户与界面之间的交互过程.在静态方面,它描述了用户与系统对话的内容,即对话过程中所涉及的用户界面元素;在动态方面,它描述了对话与对话之间的转换过程.对话模型是任务模型到呈现模型转换过程中一个必不可少的环节,为用户界面的抽象描述提供了技术支持.从 ETS 到对话模型的转换过程包括:

- 第 1 步:从 ETS 到对话的映射.

对话被认为是完成任务目标的步骤,每个对话包含的内容对应一个 ETS.从任务模型向对话模型转换的过程中,通过 ETS 将任务分析过程中的显示变量映射为用户与系统对话的内容,对话之间的转换反映了显示变量状态的改变.对话可以分为同一界面内部的对话和不同界面之间的对话.在基于 PCTBTA 方法的用户界面生成过程中,对话与任务始终保持一致性.对话模型是从任务模型中产生的,任务模型发生变化,对话模型也会随之改变.

- 第 2 步:建立对话之间的状态转换生成 STN.

为了更好地描述对话之间动态转换关系,PCTBTA 方法采用 STN 方式表示对话状态之间的转变.在 STN 中,每一个节点表示一个状态,每条边上可以标记引起状态发生改变的任务事件和任务事件执行的结果.同样一个任务事件,不同的执行结果可以使系统处于不同的状态.任务事件执行的结果可以分为 3 类:执行后成功、执行后失败和退出.其中,执行后失败又可以根据失败的原因分为行为执行后获取的数据不可用和行为执行后没有获取到数据.每个 ETS 对应 STN 中的一个节点,致使每个节点发生转变的行为在相应的边上标记.通过对任务状态的描述,可以得到抽象界面之间的逻辑关系.

3.3 从对话模型到界面模型的转换

基于任务模型的用户界面开发中,最后一个环节是在一个可用的输出设备中渲染对话模型,即,从对话模型到界面呈现模型的转换.呈现模型指界面给用户提供的在视觉上、触觉上和听觉上的表示,它包含了界面元素的各种属性,如呈现方式、交互方式和实现方式等等.从对话模型到呈现模型的转换过程包括:

- 第 1 步:从 STN 到 AUI 的映射过程.

主要包括两个方面:一是根据 PCTBTA 任务模型中已经定义的显示变量类型,抽象表示 ETS 中所包含的任务,从而确定了界面呈现的内容;二是通过 STN 中对话状态转换的描述,生成抽象交互对象(abstract interaction object,简称 AIO)之间的逻辑关系,以及界面之间的导航关系.

- 第 2 步:从 AUI 到 CUI 的转换.

CUI 是由具体交互对象(concrete interaction object,简称 CIO)构成的,CIO 是用户可以感知到的 UI 实体,如对话框、按钮、图标、下拉列表等.CIO 从 AIO 中转化而来,相同的 AIO 依据设备和交互通道的不同可以被映射为多个 CIO.在转换的过程中,主要考虑以下两个因素:呈现方式(大小、布局、形状、颜色等)和交互方式(图形、语言、触摸等).

- 第 3 步:从 CUI 到 FUI 的转换.

最终通过具体的界面设计语言(如 C,C++,Java,Html 等)实现 CUI,得到用户与之交互的界面为最终用户界面 FUI.

4 应用实例

针对 PCTBTA 方法的特点,本节以校园导航系统为例,描述基于 PCTBTA 任务模型的用户界面开发过程.

4.1 任务分析和建模过程

校园导航系统依据学校的地理信息给用户提供服务,不同的用户在不同使用环境中对系统的需求不同,如:呈现校园地图,为用户提供相关的信息查询服务、导航服务;为新生或者游览用户提供校园介绍服务;为来参加学术会议等其他活动的用户提供与近期活动相关的服务;具体分析的内容如图 4 所示.

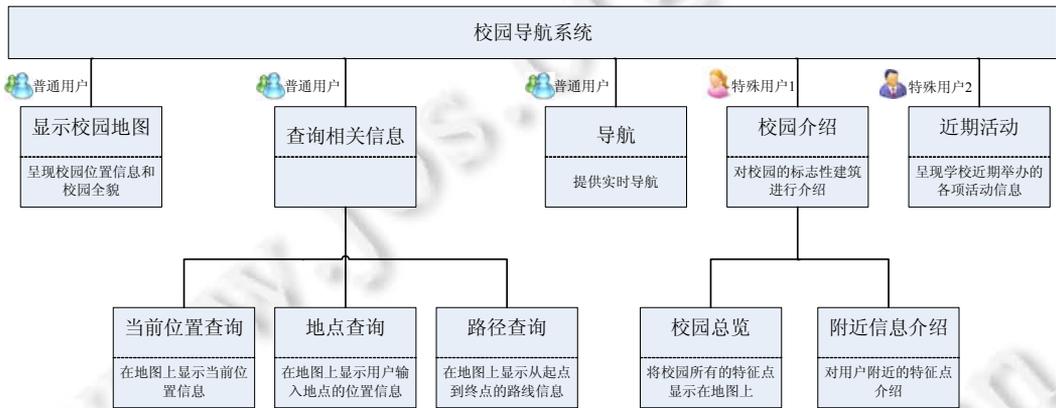


Fig.4 Task decomposition of campus navigation system on goal-oriented level

图 4 校园导航系统系统目标层面的任务分解

在 PCTBTA 方法中,允许设计者将特定场景中能够解决用户的具体问题、或者在设计过程中能被重复使用的任务单元定义为 PCTU,它可以作为一个独立的描述原语与其他任务建立联系,完成用户目标.根据对校园导航系统的分析,设计者定义的 PCTU 见表 4.

Table 4 Definition of PCTU in the campus navigation system

表 4 校园导航系统中 PCTU 的定义

名称	显示地图 (DisplayMap)	当前位置 (CurrentLoc)	地点查询 (QueryLoc)	路径查询 (QueryPath)	信息介绍 (Introducing)	近期活动 (RecentAct)	常用地址 (AddressBook)
目标	用户通过系统来查看地图信息	系统能够获取用户所在的位置信息,并显示在地图上	将用户希望查找的地理位置信息显示在地图上	将用户输入的起始点到终点的路径信息显示在地图上	对学校标志性的建筑进行介绍	呈现学校近期举办的各种活动相关信息	用户常用地址信息
用户类型	普通用户	普通用户	普通用户	普通用户	特殊用户	特殊用户	普通用户

由于篇幅所限,下文将选取校园导航系统中的路径查询这个具有代表性的实例来说明采用 PCTBTA 方法的任务建模及界面模型的生成过程.

在某个特定的情境下,一些对校园不熟悉的用户希望校园导航系统能够提供路径查询服务.采用 PCTBTA 方法分析路径查询任务的具体步骤为:

(1) 第 1 步:初步分析.

- 识别用户、用户目标和环境.
 - 用户:普通用户;

- 用户目标:在地图上显示起点位置到终点位置的路径信息;
- 环境:静态环境或者动态环境;
- 抽象层面的分析.完成“路径查询”需要执行的任务:获取起点位置→获取终点位置→用户确认→系统显示路径信息→用户退出;
- 具体层面的分析(见表 5);

Table 5 Type of interface element in path inquiry

表 5 路径查询中的界面元素类型

任务	元素名称	元素类型
获取起点位置	from	可控变量
获取终点位置	to	可控变量
用户确认	determine	可控变量
显示路径信息	display paths	仅输出变量
退出	exit	可控变量

(2) 第 2 步:延展分析.

- 分析可能外部影响因素.
 - 获取起点位置任务中兴奋效应的影响因素包括 PCTU(当前位置)、PCTU(近期活动)和 PCTU(常用地址),它们为用户提供多种获取起点位置的途径,以满足不同用户的需求,帮助用户快速实现任务目标;抑制效应的影响因素包括系统没有正确获取起始位置信息;
 - 获取终点位置任务中兴奋效应的影响因素包括 PCTU(近期活动)和 PCTU(常用地址),抑制效应的影响因素包括系统没有正确获取终点位置信息;
- 分析不同影响下的任务序列,包括通常情况下的操作、针对不同使用上下文的操作以及处理异常事件的操作.获取起点位置和获取终点位置任务,一般情况下由用户自己输入位置信息.当兴奋效应的影响因素可用时,系统给用户多种可能达到目标的途径由用户选择.当抑制效应的影响因素存在时,系统能够及时给用户反馈信息和应对策略.

(3) 第 3 步:通过与用户交流讨论,或根据收集的数据评估任务分析的结果.

路径查询属于用户目标层面的任务,通过对任务分析内容建模,得到的 PCTBTA 任务模型如图 5 所示.

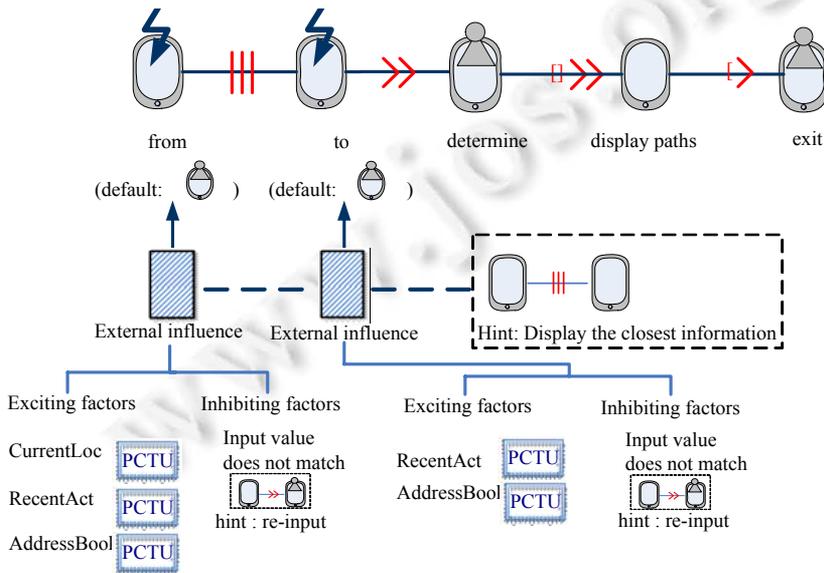


Fig.5 PCTBTA task model of path query

图 5 路径查询的 PCTBTA 任务模型

通常情况下,完成用户目标的任务集合 $GTC\{from,to,determine,displaypaths,exit\}$.其中,获取起点位置(from)和获取终点位置(to)任务的输入可以从外部获取,所以将其定义为感知任务类型,它们的缺省任务为用户输入(user_input),即:没有感知到其他影响因素时,需要用户自己输入起点或终点位置.每个子任务的具体定义见表 6.

Table 6 Task definition in path query

表 6 路径查询中任务的定义

Task name	from	to	determine	display paths	exit
Task type	perceptual task	perceptual task	interaction task	system task	interaction task
User goal	starting position	final position		get path information	exit
Displayed variables	control variables	control variables	control variables	only_output variables	control variables
External influence	Y	Y	N	N	N
Exciting factors	CurrentLoc_PCTU RecentAct_PCTU AddressBook_PCTU	RecentAct_PCTU AddressBook_PCTU			
Inhibiting factors	input_notmatch	input_notmatch			
Action	hint: re-input	hint: re-input			

4.2 界面模型的生成过程

4.2.1 从任务模型中提取 ETS

根据第 3.1 节所描述的内容,从 PCTBTA 任务模型到 ETS 的转换包括以下步骤.

(1) 第 1 步:将感知任务中的影响因素融合到任务序列中.

路径查询中包含两个感知任务:获取起点位置(from)和获取终点位置(to).其中,

- 起点位置的缺省任务为:用户输入起点位置信息 $from \in \tilde{T} | t = user_input()$;
- 产生兴奋效应的因素有: $\bar{F}_1 = CurrentLoc_PCTU$, $\bar{F}_2 = RecentAct_PCTU$ 和 $\bar{F}_3 = AddressBook_PCTU$;
- 兴奋因素当前位置与缺省任务之间的关系为不完全替代关系 $\bar{F}_1 \xrightarrow{R} \tilde{T} | t | r = replace - incomplete$, 即:首先执行 *CurrentLoc* 任务,如果失败,则执行 *user_input* 任务;
- 兴奋因素常用地址和近期活动与缺省任务之间是无序选择关系:

$$\bar{F}_2 \xrightarrow{R} \tilde{T} | t | r = choice - unordered, \bar{F}_3 \xrightarrow{R} \tilde{T} | t | r = choice - unordered;$$

- 产生抑制效应因素为:用户输入的数据与数据库中的信息不匹配 $\bar{F}_4 = input_notmatch$.

与起点位置分析过程相似,终点位置的分析结果如下.

- 缺省任务 $to \in \tilde{T} | t = user_input()$;
- 产生兴奋效应的因素: $\bar{F}_1 = RecentLoc_PCTU$ 和 $\bar{F}_2 = AddressBook_PCTU$, 它们与缺省任务之间的关系: $\bar{F}_1 \xrightarrow{R} \tilde{T} | t | r = choice - unordered, \bar{F}_2 \xrightarrow{R} \tilde{T} | t | r = choice - unordered$;
- 产生抑制效应的因素: $\bar{F}_1 = input_notmatch$.

根据以上分析,可以得到考虑外界影响因素后“路径查询”的任务序列集合,如图 6 所示.

(2) 第 2 步:从带有 PCTU 的任务序列中提取 ETS.

在路径查询的任务序列集合中包括 3 个 PCTU:当前位置、近期活动和常用地址,对这 3 个 PCTU 进行预处理,结果如图 7 所示.

依据第 3.1 节中描述的 ETS 提取和聚合的方法,可以得到:

- **ETS1** {from(GPS/wifi),from(input),from(RecentAct),form(AddressBook),to(input),to(RecentAct),to(AddressBook),determine,exit};
- **ETS2** {from(input),from(RecentAct),form(AddressBook),to(input),to(RecentAct),to(AddressBook),determine,exit};

- **ETS3** {from(GPS/wifi),from(input),form(AddressBook),to(input),to(AddressBook),determine,exit};
- **ETS4** {from(GPS/wifi),from(input),from(RecentAct),to(input),to(RecentAct),determine,exit};
- **ETS5** {from(input),from(RecentAct),to(input),to(RecentAct) ,determine,exit};
- **ETS6** {from(input),from(AddressBook),to(input),to(AddressBook),determine,exit};
- **ETS7** {from(input),to(input),determine,exit};
- **ETS8** {display,exit};
- **ETS9** {hint(GPS/wifi_notavailable),choice};
- **ETS10** {hint(from_notmatch),re-input};
- **ETS11** {hint(to_notmatch),re-input};
- **ETS12** {hint(task_fail),display the closest information}.

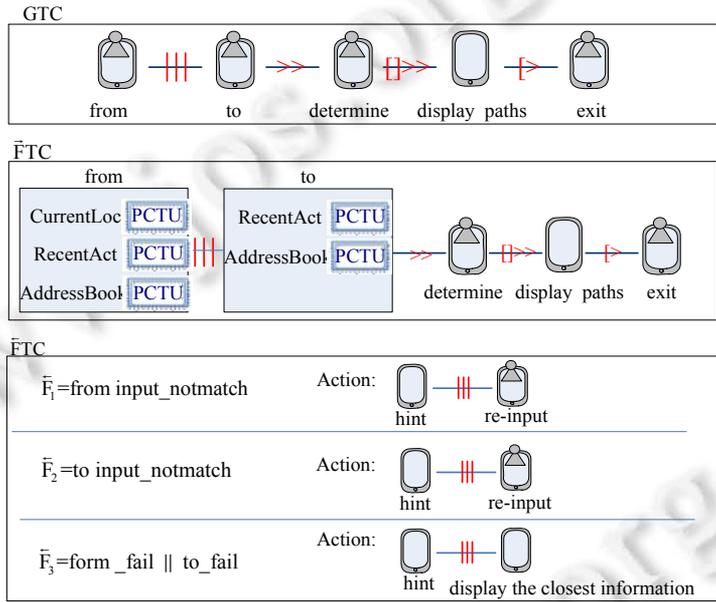


Fig.6 Task sequence set of path query

图 6 路径查询的任务序列集合

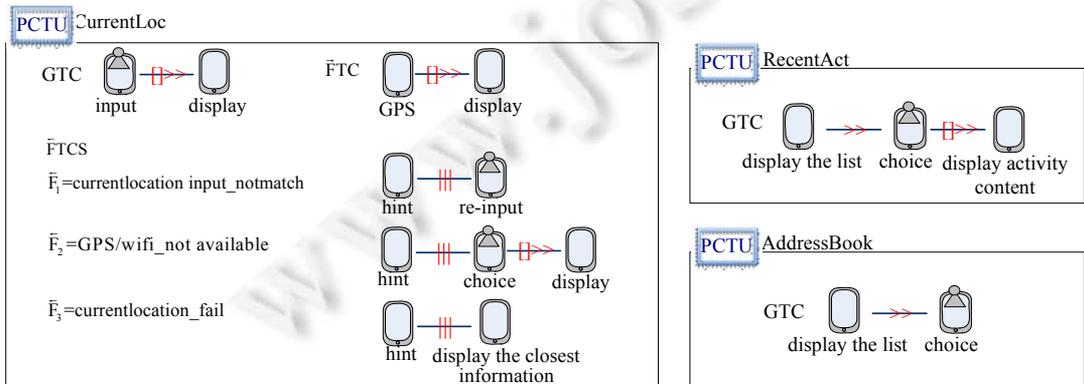


Fig.7 Pretreatment process of current location, recent activity and common address

图 7 当前位置、近期活动和常用地址预处理过程

4.2.2 从 ETS 到对话模型的转换

每个 ETS 表示逻辑上可以呈现在同一界面的内容,通过 ETS,将任务分析过程中的显示变量映射到用户与系统的对话过程中,对话之间的转换关系可以通过 STN 图来表示.路径查询的状态转换图如 8 所示,其中,ETS1 为初始状态,ETS8 和 ETS12 为任务执行过程中的可接受状态.

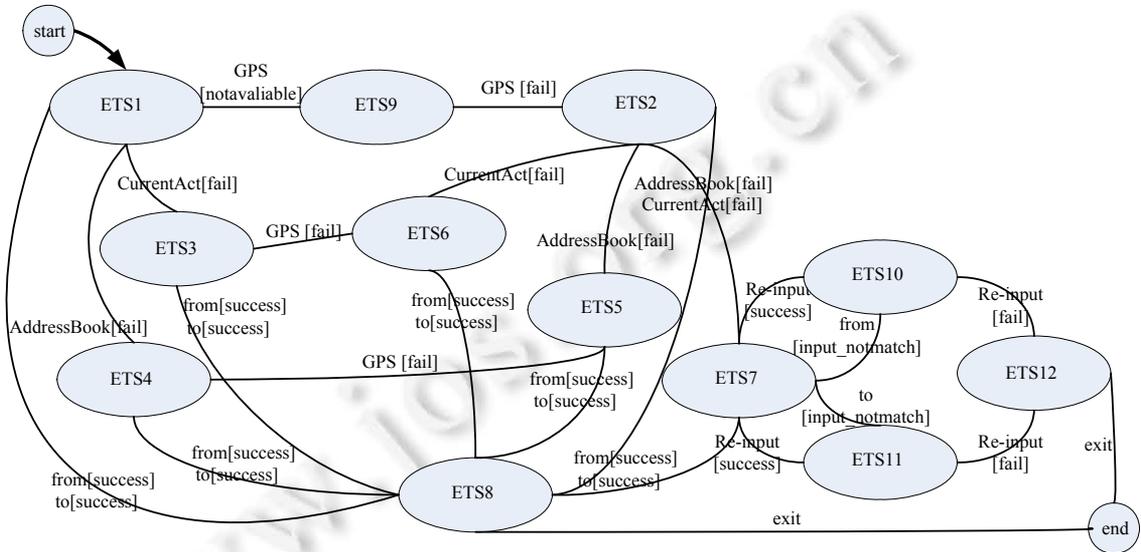


Fig.8 STN diagram of path query

图 8 路径查询的状态转换图

4.2.3 从对话模型到呈现模型的转换

ETS 中所包含的任务均为原子任务,采用 PCTBTA 方法定义时,每个原子任务被定义为一个显示变量,根据显示变量的类型,可以完成从对话模型到抽象用户界面的转换.ETS 中的任务与显示变量类型的对应关系见表 7,通过 STN 图中每个节点所包含的显示变量类型,可以生成路径查询的抽象界面和用户界面原型如图 9、图 10 所示.

Table 7 Relationship between the task and the display variable type in ETS of path query

表 7 路径查询 ETS 中的任务与显示变量类型的对应关系

ETS	ETS中包含的任务	显示变量类型	ETS	ETS中包含的任务	显示变量类型	ETS	ETS中包含的任务	显示变量类型
ETS1	from(GPS)		ETS3	from(AddressBook)	single_choice	ETS6	to(input)	text_edit
	from(input)	text_edit		to(input)	text_edit		to(AddressBook)	single_choice
	from(RencentAct)	single_choice		to(AddressBook)	single_choice		determine	activator
	from(AddressBook)	single_choice		determine	activator		exit	activator
	to(input)	text_edit	ETS4	from(GPS)		ETS7	from(input)	text_edit
	to(RencentAct)	single_choice		from(RencentAct)	single_choice		to(input)	text_edit
	to(AddressBook)	single_choice		to(input)	text_edit		determine	activator
	determine	activator		to(RencentAct)	single_choice		exit	activator
exit	activator	ETS5	from(input)	text_edit	ETS8	display	object_output	
from(input)	text_edit		determine	activator		exit	activator	
from(RencentAct)	single_choice		exit	activator		ETS9	hint(GPS_notavailable)	feedback_output
to(AddressBook)	single_choice		from(input)	text_edit		choice	single_choice	
determine	activator	ETS12	from(RencentAct)	single_choice	ETS10	hint(from_notmatch)	feedback_output	
exit	activator		to(input)	text_edit	from(re-input)	text_edit		
from(GPS)	text_edit		to(AddressBook)	single_choice	ETS11	hint(to_notmatch)	feedback_output	
from(input)	text_edit		determine	activator	to(re-input)	text_edit		
ETS2	from(GPS)	text_edit	ETS6	from(input)	text_edit	ETS12	hint(task_fail)	feedback_output
ETS3	from(input)	text_edit		from(AddressBook)	single_choice		display the closest information	object_output

5 对比分析与评估

为了说明基于 PCTBTA 任务模型用户界面开发方法有效性,本文从可用性和性能方面将几种典型的建模方法,如 HTA,Use Case,CTT,与 PCTBTA 方法进行了比较.在可用性方面,由于基于任务模型的用户界面开发方法并非完全自动化的过程,有很多存在于任务分析和建模过程的可用性缺陷可以由设计者经验及后期迭代过程中通过用户反馈进行修正.因此,本文在建模方法的比较过程中采用了此领域较为常用的可用性研究方法^[13-15],对不同方法,在任务建模及界面生成过程中所包含的可用性因素进行对比分析.在性能方面,则采用实验的方法,针对不同方法的支持工具在完成设计任务所消耗的时间和单位时间生成有效任务序列数目方面进行对比评估.

5.1 可用性方面的对比分析

可用性领域的杰出代表人物 Nielsen 指出:对于许多从事可用性研究的学者来说,最大的成就是发明一种分析方法,这种方法可以让设计师在测试之前预测用户界面的可用性^[16].界面的可用性是在设计开发过程中所确定的,每个阶段所考虑的可用性指标不同.如果在前期设计阶段缺失了对某些影响可用性因素的描述,在后期的开发过程中必须依靠开发者的经验来完善设计.如图 11 表示了不同方法建模路径查询任务及所生成的界面,其中采用 HTA 和 CTT 方法建模任务时,由于缺少对环境信息的描述,导致所生成的用户界面仅包含通常情况下的任务,而没有给用户提供更多快捷交互的可能,如利用智能设备感知到的地理位置信息替代用户的输入等.

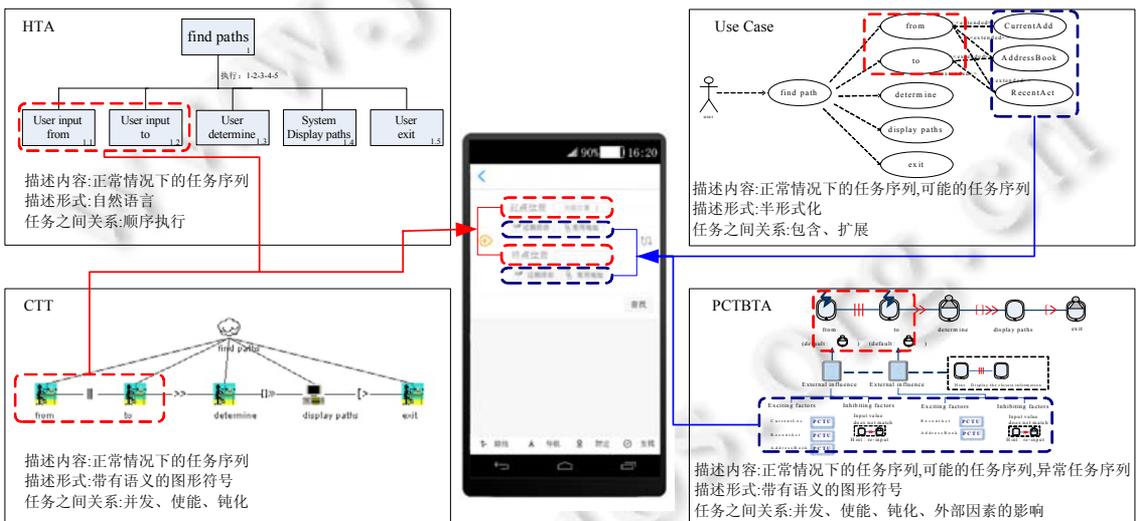


Fig.11 Path query task models building with different methods

图 11 不同方法建模路径查询任务

国际标准 ISO 9241^[17]中将可用性定义为一个产品可以被特定的用户在特定的使用上下文中,有效、高效并且满意地达到特定目标的程度.可用性概念本身是不可直接度量的,但与其相关的一些参数具有可度量的性质.根据 Welie 在文献[18]中对可用性层次的划分,定义了与有效性和高效性相关的可用性指标.它们大多都基于设计经验产生,用于指导设计,在设计后期(界面设计阶段)又被细化为相关的设计规范.对于一种设计开发方法而言,如果在设计阶段满足了这些指标和规范,就能够在一定程度上保证生成界面的可用性^[19].本文对 PCTBTA 方法在任务建模及界面生成过程中所包含的可用性因素进行了总结(见表 8),并与其他 3 种方法比较,用以说明本方法在生成高可用界面方面的作用.

Table 8 Usability factors considered at different stage of interface development

表 8 不同方法在界面开发各阶段包含的可用性因素

可用性	指标	设计规范细则	PCTBTA		HTA		Use Case		CTT		
			任务层面	界面层面	任务层面	界面层面	任务层面	界面层面	任务层面	界面层面	
有效性	错误/安全	提供系统状态反馈	◎	●	◎	●	◎	●	◎	●	
		用户操作的一致性	●	○	○	○	◎	○	◎	○	
		限制用户输入值种类	●	○	○	○	●	○	●	○	
		对可能引起严重后果的操作提供补救措施	●	○	○	○	●	○	◎	○	
		对错误操作支持可逆处理	●	○	○	○	●	○	◎	○	
	有清晰的错误提示	●	●	○	●	●	●	◎	●		
	易记性	让用户知道自己的当前位置和系统状态	◎	●	◎	●	◎	●	◎	●	
		迅速响应用户动作并显示动作结果	●	○	○	◎	◎	◎	●	○	
		采用用户的语言,而非技术语言	●	●	●	●	●	●	●	●	
		考虑人类处理信息的限度,控制任务复杂度	●	○	●	○	●	○	●	○	
		提供视觉线索	○	●	○	●	○	●	○	●	
	任务完成	界面信息呈现的一致性	○	●	○	●	○	●	○	●	
		给用户提供清晰的任务结构	●	○	●	○	●	○	●	○	
	高效性	易学习性	对话应该为用户提供成功完成任务的相关信息	●	○	◎	○	●	○	●	○
			对用户适当的帮助	●	○	●	○	●	○	●	○
使用户操作最小化			●	○	○	○	●	○	◎	○	
执行速度		在增加自动化的同时,确保用户对系统的控制	●	○	○	○	◎	○	◎	○	
		数据输入业务流程的一致性	●	○	◎	○	●	○	●	○	
		保持功能及内容描述的一致性	●	●	○	○	●	●	●	●	
		对话逻辑的一致性	●	○	○	○	◎	○	●	○	
		界面元素显示的一致性,如:字体、格式、颜色等	○	●	○	●	○	●	○	●	
		系统能够适应不同的使用环境	●	○	◎	○	●	○	◎	○	
		系统能够适应多个用户角色	●	○	◎	○	●	○	●	○	
		按功能将界面进行划分	○	●	○	●	○	●	○	●	
		完成相同或相近功能的界面元素应包含在相同的呈现单元中	○	●	○	◎	○	◎	○	●	
		用户控制数据显示的灵活性	○	●	○	●	○	●	○	●	
用户控制数据输入的灵活性		●	◎	◎	◎	●	●	◎	●		
为专家用户更快地完成任务, 给其提供易于操作的快捷方式		●	●	●	●	●	●	●	●		
令用户满意	满意度	○	○	○	○	○	○	○	○		

●包含 ◎部分包含 ○不包含

(1) 有效性方面

有效性指用户完成特定目标的准确性和完整性.在设计过程中,反映有效性的可用性指标有错误/安全、易记性和任务完成.

- 错误/安全包括两个方面:错误的预防和提供解决错误方法.在错误预防方面,PCTBTA 方法允许设计者将重复出现的任务序列定义为 PCTU,保证了每次 PCTU 出现时,其任务序列的一致性,从而降低了用户的出错概率;在错误处理方面,PCTBTA 方法显式表示了抑制效应的影响因素,引导设计者分析可能出现的异常情况以及应对异常情况的任务序列;
- 易记性可以通过在设计过程中给用户反馈、采用用户语言以及让用户的记忆负担最小化等方法实现.PCTBTA 方法在建模过程中,始终以用户目标为导向组织任务序列,根据使用上下文的变化给用户及时的反馈信息和可能简化任务操作的选择;

- 任务完成.为了让用户更好地完成任务,在设计时可以为提供导航和规范的表示.PCTBTA 方法中,系统层面的任务组织过程可以为用户提供清晰的结构表示,形式化的图形符号保证了任务描述的规范化.

(2) 高效性方面

高效性指用户实现特定目标的准确性和完整性所消耗的资源.在设计过程中,反映高效性的可用性指标有易学习性和执行速度.

- 易学习性可以通过提供帮助文档,或提高系统的自动化程度等方法实现.PCTBTA 方法中,结合普适计算环境中用户任务的特点,增加了感知任务类型,使得通过智能设备感知到的环境信息可以替代用户的输入,从而简化用户的任务,使用户目标完成有最少的输入动作;
- 执行速度可以通过保证一致性、提高适应性、有效的界面元素组织、灵活性、提供快捷方式等方法实现.PCTBTA 方法中,对话模型是从任务模型中产生,而呈现模型又是在对话模型的基础上生成的,因此该方法保证了整个设计和开发过程中的一致性.除此之外,PCTBTA 方法通过描述不同上下文的任务执行序列为开发具有适应性的界面提供了基础.

(3) 满意度方面

满意度指使用产品的舒适度和可接受程度.满意度是用户主观上对系统的总体评价.在设计和开发过程中,仅能够根据用户需求及用户反馈,通过迭代的方法试图开发出令用户满意的产品,而没有完全量化的标准.对满意度的评测大多采用问卷调查或访谈的方法.

通过表 8 可以看出,PCTBTA 方法在可用性指标分析方面优于存在的方法.HTA,Use Case 和 CTT 方法对于上述可用性指标的某些方面隐式地表现在建模过程中,这导致有经验的设计者可以根据个人经验在开发后期进行补充,而没有经验的设计者可能会遗漏相关内容.如在 HTA 和 CTT 方法中没有明确说明有关异常情况的描述,需要设计者根据个人设计经验添加到在界面开发过程中.Use Case 方法没有详细定义任务执行过程的时序关系,使得在开发过程中很难保证任务执行顺序的一致性.

5.2 性能评估

为了使 PCTBTA 方法得到更广泛地应用,依据该方法的特点,我们开发了设计工具原型(如图 12 所示),并将其与 HTA,Use Case,CTTE 工具在消耗时间和生成有效任务序列的数目方面进行了对比和评估.

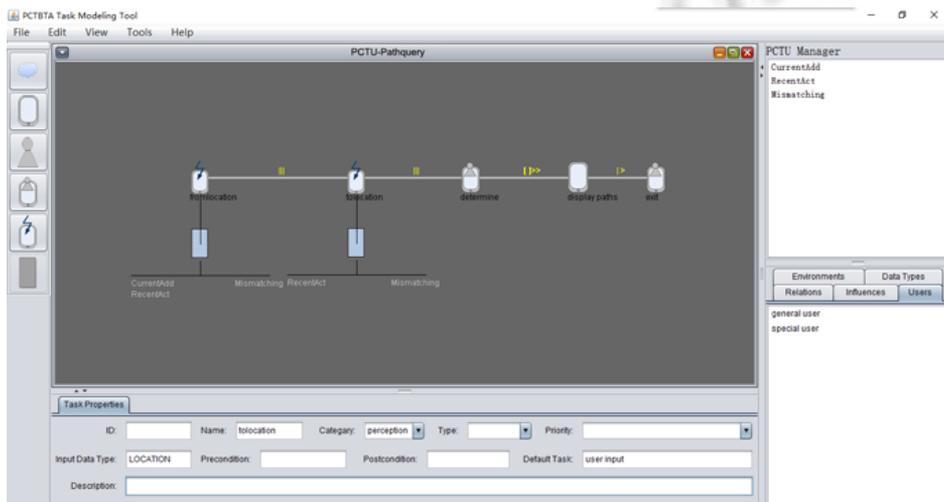


Fig.12 PCTBTA task modeling tool

图 12 PCTBTA 任务建模工具

评估过程招募了 16 位被试人员,他们都具有人机交互方面的学科背景,了解基于任务模型开发方法的理论及技术。其中,8 位被试者具有设计开发经验;另外 8 位被试者仅了解基本的设计原理和开发技术,实际开发经验较少。测试内容为建模校园导航系统中查询相关信息任务,包括当前位置查询、地点查询和路径查询。测试前要求被测者阅读各种任务模型相关的资料和分析实例,并且熟悉设计工具环境。测试维度包括设计过程所消耗时间和设计完成后经统计得到的单位时间内生成的有效任务序列数目。

比较结果表明(如图 13 所示):在消耗时间方面,无经验设计者采用 HTA 工具所消耗的时间较短,而有经验的设计者则采用 PCTBTA 工具消耗的时间较短。其原因在于:HTA 工具以它简单的表示、层次化的结构,非常容易让无经验设计者掌握;然而对于有经验的设计者而言,HTA 工具采用自然语言的描述,如果要增加任务属性等信息,需要采用其他方式,因而描述相同设计内容所耗费的时间较多。与之相比较,Use Case,CTT 和 PCTBTA 工具均采用带有语义信息的符号来构建任务模型,虽然学习复杂度高,但任务模型中包含了丰富的语义信息,并且易于向其他模型转换,有较强的扩展性。除此之外,PCTBTA 工具允许设计者定义并且重复使用 PCTU,增加了任务片段的可复用性,进一步提高了开发效率。在单位时间生成的有效任务序列数目方面,对于有经验的设计者,一旦设计思想形成,基本不受建模方法的影响,相对于其他工具,PCTBTA 工具能更有效地帮助他们构建模型;而对于没有经验的设计者,采用 PCTBTA 方法能够引导他们分析完成的不同途径和可能出现的异常情况及其解决方案,使其更全面地描述用户任务,提高设计开发效率。

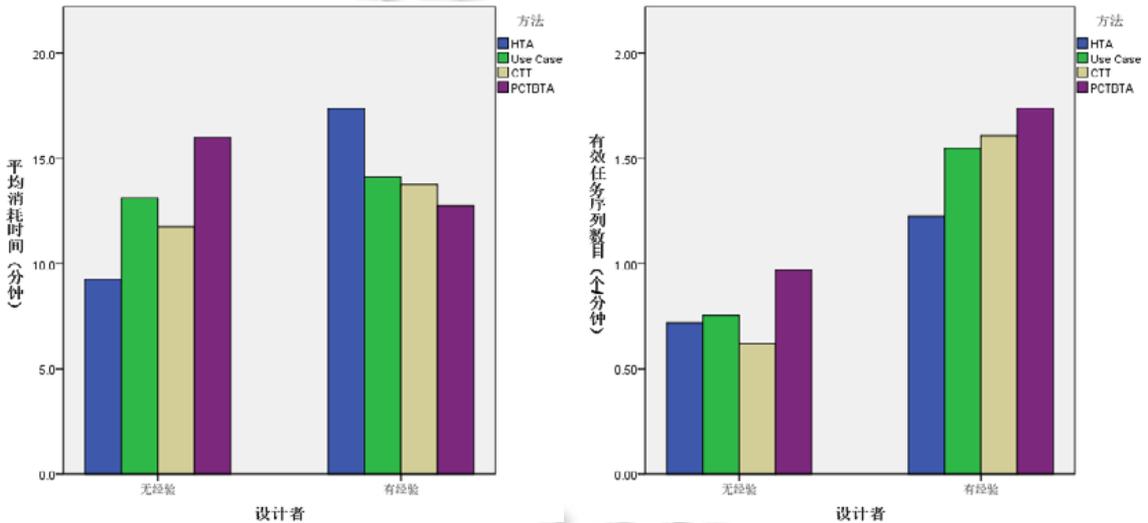


Fig. 13 Comparison of different methods in terms of time consuming and number of task sequences per unit time

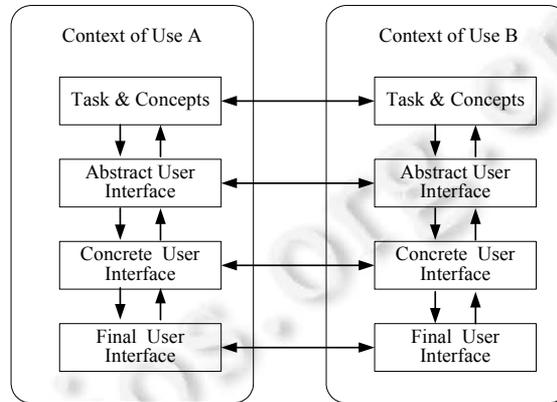
图 13 采用不同方法在消耗时间与单位时间生成的任务序列数目方面的比较

6 相关工作

在多设备、多通道、多平台的环境下,界面开发工作的复杂程度大大提高,开发人员需要为同一功能核心开发出适应不同软硬件平台的多个界面版本。为了简化开发过程,降低开发成本,目前人机交互领域的研究趋势是采用 MBUID 方法,使设计者能够从高级抽象层次描述用户界面,再通过模型转换将抽象模型逐步映射为最终代码,实现了界面原型的自动(半自动)生成。CAMELEON 参考框架^[6](如图 14 所示)表示了当前研究的特征。

- 任务和概念(task & concepts):指为了完成用户目标而执行的任务,以及在任务中所包含的领域对象;
- 抽象用户界面(AUI):指采用抽象交互对象 AIO 从高级抽象层次对用户界面的描述。AUI 是一种与平台无关的模型,它独立于具体的设备、交互通道、交互技术、开发平台;

- 具体用户界面(CUI):指采用具体平台的程序设计构件 CIO 对用户界面的描述.CUI 表示了与具体平台相关,但和实现语言无关的界面模型;它显式地表示了特定设备、特定交互通道下用户界面的内容与呈现风格(look and feel),但仅是界面的设计原型,而非最终开发的实现界面;
- 最终用户界面(FUI):指由编程语言或者标记语言构成,可以被解析或编译的最终呈现给用户的界面。

Fig. 14 CAMELEON reference framework^[6]图 14 CAMELEON 参考框架^[6]

CAMELEON 参考框架支持多目标、多使用上下文的用户界面开发,对开发上下文敏感用户界面提供了统一的解决方法.该框架将使用上下文定义为一个三元组 $\text{context}=(\text{user},\text{platform},\text{environment})$:

- **user** 指与系统交互的最终用户;
- **platform** 指用户在执行任务时使用的硬件和软件计算平台;
- **environment** 指用户使用系统的物理环境。

使用上下文被应用于不同层次的模型转换中,从顶向下的转换过程是对抽象描述逐步精化的过程,同样的抽象描述根据不同的使用上下文信息可以映射为不同的用户界面^[20]。

目前,基于 CAMELEON 参考框架用户界面生成的研究工作依据所解决问题的领域,主要分为两类:开发适应不同软硬件计算平台的界面;开发适应多种用户和使用环境的界面。

对开发适应多平台界面方面研究,代表性的工作有:

- **UsiXML** 是由 Katholieke Universiteit Leuven 开发的一种基于 XML 的标记语言,它可以描述不同设备、交互通道的用户界面^[21,22].UsiXML 依赖 CAMELEON 参考框架定义的 4 层抽象,采用多个元模型描述用户界面的不同方面,用一个转换模型来定义不同模型之间的转换关系,完成从 CTT 任务模型到最终用户界面之间的转换;
- **MARIA** 是一种基于模型的用户界面语言,它描述了如何在 CTT 任务模型的基础上为不同的设备、平台开发不同的用户界面^[23].MARIA 分别从 4 个方面去抽象描述平台和资源相关的交互信息:数据模型、事件模型、支持 Ajax 脚本、用户界面元素的动态集合。

上述研究主要是在现有任务模型所描述的交互语义基础上,结合软硬件平台属性,逐步转换生成适用于特定平台的用户界面.与本文工作相比较,主要区别是在模型转换过程中针对的使用上下文不同:前者研究的重点是如何针对设备、交互通道和开发平台的多样性,开发适应于各种技术异构性的界面;而本文所针对的上下文是对任务执行逻辑有影响的因素,如用户、环境等.在任务建模过程中,将使用上下文的特征进行了分析和处理,在后续的工作中只需按照相应的转换流程,就可以开发出具有一定可用性的界面。

对开发适应多用户和使用环境界面方面研究,代表性的工作有:

- **Halbrügge** 等人针对普适计算环境中用户与系统交互过程中的错误,提出一种基于目标记忆理论的认

知模型与 MBUID 开发框架相结合的系统^[24],依据用户界面元素特征来预测任务执行中的错误,使设计的用户界面尽可能避免用户错误操作;

- DynaMo-AID 动态任务模型^[25],它在 CTT 模型的基础上增加了上下文相关的决策点,依据运行时刻的使用上下文信息来选择子任务,动态生成对话模型;
- Klug 等人提出了一种可执行的任务模型 Executable Task Model^[26],为每个任务增加输入/输出端口,实现任何两个任务之间的信息交换操作,使得在动态环境中任务组织更加灵活;
- 为了保证系统的可信性,Martinie 等人提出了 HAMSTERS 方法^[27],针对在使用软件过程中可能出现的异常情况及其应对策略进行分析,将由系统或者人类导致任务失败时所需要执行的活动序列整合到任务模型中;
- Gaulke 等人提出一种增强执行规则的任务模型 Rule-Enhanced Task Model^[28],在 CTT 任务模型的基础上增加了规则语言,将领域模型与任务模型紧密结合,在任务模型中嵌入业务逻辑,更精确地描述了任务的执行过程。

上述工作对 PCTBTA 方法具有启发意义,但以上研究仅仅是针对实际工作中某一个具体问题,对现有任务模型在某一个方面的改进,每个工作都有各自的侧重点。而 PCTBTA 方法则更加全面、更加系统地描述了普适计算环境中基于任务模型的界面生成过程,包括任务分析的思想、任务建模、任务表示符号以及从抽象描述到界面模型的转换过程,相对于上述工作具有更广泛的应用领域。

7 结束语

为了提高 MBUID 生成用户界面的可用性,本文提出一种基于 PCTBTA 任务模型的用户界面开发框架,探索一种更加适用于建模普适计算环境中用户任务的方法,并且为这种概念化方法到用户界面原型的转换提供了技术支持。与其他任务模型相比较,PCTBTA 任务模型的优势主要表现在:

- (1) 描述能力更强:不仅能够描述一般情况下用户与系统的交互过程,也能够描述不同使用上下文中用户、环境与系统的交互过程;
- (2) 有较好的可复用性和可扩展性:在 PCTBTA 方法中,设计者可以将那些常见的、反复出现的任务集合定义为一个 PCTU,它不仅包括常见任务的重复使用,也包括对诸如用户及使用上下文、领域知识、设计知识等的复用。当有新的用户需求,或者有新的使用情境,亦或有新技术的出现,设计者可以添加新的 PCTU。这种方法提供了分析内容的扩充机制;
- (3) 技术无关性:PCTBTA 方法为设计者提供了一套带有语义的任务描述符号,使得设计者在设计过程中,只需要对用户任务的描述,而无须考虑复杂的开发过程。

随着人们生活的物理空间和信息空间的不断融合,用户与系统的交互范围不仅限于单用户和单系统、多用户和单系统,而是单用户和多系统、多用户和多系统之间的交互。在这种多维度交互空间中,用户的任务也变得更加复杂,任务的执行需要跨越多个计算环境。本文后续的工作主要集中在:如何在原有理论上,结合不同设备不同系统的特征,动态组织环境资源来完成相应的任务,并且始终保持任务执行的连续性,让用户在普适环境中与各个设备实现无缝的交互。

References:

- [1] Eisenstein J, Vanderdonck J, Puerta A. Applying model-based techniques to the development of UIs for mobile computers. In: Proc. of the 5th ACM Int'l Conf. on Intelligent User Interfaces (IUI 2001). 2001. 69-76. [doi: 10.1145/359784.360122]
- [2] Paternò F, Santoro C, Spano LD. Exploiting Web service annotations in model-based user interface development. In: Proc. of the 2010 ACM SIGCHI Symp. on Engineering Interactive Computing Systems. 2010. 219-224. [doi: 10.1145/1822018.1822053]
- [3] Delgado A. Reusing UI elements with model-based user interface development. Int'l Journal of Human-Computer Studies, 2016, 86(C):48-62. [doi: 10.1016/j.ijhcs.2015.09.003]

- [4] Sayih M, Brüggemann KA, Dimitrov L. Development of model-based user interfaces with XML technology. In: Proc. of the IEEE 7th Int'l Conf. on Intelligent Computing and Information Systems (ICICIS 2015). 2015. 321–327. [doi: 10.1109/IntelCIS.2015.7397241]
- [5] Weyers B. Formal Description of Adaptable Interactive Systems Based on Reconfigurable User Interface Models. *The Handbook of Formal Methods in Human-Computer Interaction*. Springer Int'l Publishing, 2017. 273–294.
- [6] Calvary G. A unifying reference framework for multi-target user interfaces. *Interacting with Computers*, 2003,15(3):289–308. [doi: 10.1016/S0953-5438(03)00010-9]
- [7] Taylor MM. Editorial: Perceptual control theory and its application. *Int'l Journal of Human-Computer Studies*, 1999,50(6):433–444. [doi: 10.1006/ijhc.1998.0262]
- [8] Paternò F. *Model-Based Design and Evaluation of Interactive Applications*. London: Springer-Verlag, 2000. 73–77. [doi: 10.1007/978-1-4471-0445-2]
- [9] Wasserman A. Extending state transition diagrams for the specification of human-computer interaction. *IEEE Trans. on Software Engineering*, 1985,11(8):699–713.
- [10] Paternò F, Mancini C, Meniconi S. ConcurTaskTrees: A diagrammatic notation for specifying task models. In: Proc. of the Human-Computer Interaction (INTERACT'97). Springer-Verlag, 1997. 362–369. [doi: 10.1007/978-0-387-35175-9_58]
- [11] Li JN, Hua QY, Ji X, Zhang MJ. Task analysis approach based on perceptual control theory. *Journal of Beijing Institute of Technology*, 2016,25(2):278–286.
- [12] Zhang WB. The research of task model based user interface automatic generation [MS. Thesis]. Ji'nan: Shandong University, 2007 (in Chinese with English abstract).
- [13] Akiki PA, Bandara AK, Yu Y. Adaptive model-driven user interface development systems. *ACM Computing Surveys*, 2015,47(1): 1–33. [doi: 10.1145/2597999]
- [14] Ragosta M, Ragosta M, Fahssi R. Extending procedural task models by systematic explicit integration of objects, knowledge and information. In: Proc. of the European Conf. on Cognitive Ergonomics. ACM Press, 2013. 23.
- [15] Fleischmann A, Lippe S, Meyer N, *et al.* Coherent task modeling and execution based on subject-oriented representations. In: Proc. of the Int'l Conf. on Task MODELS and Diagrams for User Interface Design. Springer-Verlag, 2009. 78–91. [doi: 10.1007/978-3-642-11797-8_7]
- [16] Nielsen J. *Usability Engineering*. Academic Press, Inc., 1993.
- [17] ISO 9241-11. Ergonomic requirements for office work with visual display terminals (VDTs)—Part 11: Guidance on usability. 1998.
- [18] Van Welie M. Task-Based user interface design [Ph.D. Thesis]. Amsterdam: Vrije Universiteit, 2001.
- [19] Shneiderman B, Plaisant C. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 4th ed., New York: Pearson, 2004. 15–33.
- [20] Miñón R, Paternò F, Arrue M, Abascal J. Integrating adaptation rules for people with special needs in model-based UI development process. *Universal Access in the Information Society*, 2016,15(1):153–168.
- [21] Limbourg Q, Vanderdonck J, Michotte B. USIXML: A language supporting multi-path development of user interfaces. In: Proc. of the Engineering Human Computer Interaction and Interactive Systems (EHCI 2004). Springer-Verlag, 2004. 200–220. [doi: 10.1007/11431879_12]
- [22] Desruelle H. Accessible user interface support for multi-device ubiquitous applications: Architectural modifiability considerations. *Universal Access in the Information Society*, 2016,15(1):5–19. [doi: 10.1007/s10209-014-0373-0]
- [23] Paternò 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):433–486. [doi: 10.1145/1614390.1614394]
- [24] Halbrügge M, Quade M, Engelbrecht KP. Predicting user error for ambient systems by integrating model-based UI development and cognitive modeling. In: Proc. of the ACM Int'l Joint Conf. on Pervasive and Ubiquitous Computing. 2016. 1028–1039.
- [25] Clerckx T, Luyten K, Coninx K. DynaMo-AID: A design process and a runtime architecture for dynamic model-based user interface development. In: Bastide R, ed. Proc. of the 2004 Int'l Conf. on Engineering Human Computer Interaction and Interactive Systems (EHCI-DSVIS 2004). Berlin, Heidelberg: Springer-Verlag, 2005. 77–95. [doi: 10.1007/11431879_5]

- [26] Klug T, Kangasharju J. Executable task models. In: Proc. of the Int'l Workshop on Task MODELS and Diagrams. ACM Press, 2005. 119–122. [doi: 10.1145/1122935.1122958]
- [27] Martinie C, Palanque P, Barboni E. Task-Model based assessment of automation levels: Application to space ground segments. In: Proc. of the IEEE Int'l Conf. on Systems, Man and Cybernetics. IEEE Press, 2011. 3267–3273. [doi: 10.1109/ICSMC.2011.6084173]
- [28] MGaulke W, Ziegler J. Rule-Enhanced task models for increased expressiveness and compactness. In: Proc. of the ACM Sigchi Symp. 2016. 4–15.

附中文参考文献:

- [12] 张文波. 基于任务模型的用户界面自动生成研究[硕士学位论文]. 济南: 山东大学, 2007.



李娟妮(1979—),女,山东潍坊人,博士,讲师,主要研究领域为人机交互,软件工程.



陈锐(1979—),男,工程师,主要研究领域为人机交互,推荐系统.



华庆一(1956—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为人机交互,软件工程.



苏荟(1977—),女,讲师,主要研究领域为时间序列分析.



吴昊(1972—),男,博士,高级工程师,主要研究领域为人机交互,计算机网络.



周筠(1984—),女,博士,副教授,主要研究领域为人机交互,教育技术.