

普适计算应用的快速设计与评估技术

唐 蕾¹ 周兴社² 於志文²

(长安大学信息工程学院 西安 710064)¹ (西北工业大学计算机学院 西安 710072)²

摘 要 普适计算的特点使得营造以人为本的信息服务环境成为可能。然而构建具有可行性和高满意度的普适计算应用却是一项需要投入时间和精力复杂工程。首先分析了构建普适计算应用面临的挑战;然后总结、比较和分析了设计方法和仿真测试技术等相关研究;最后展望了当前技术的研究难点和发展趋势。

关键词 普适计算, 开发方法, 原型工具, 仿真测试

中图分类号 TP311.5 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.08.006

Rapid Design and Evaluation for Pervasive Application

TANG Lei¹ ZHOU Xing-she² YU Zhi-wen²

(School of Information Engineering, Chang'an University, Xi'an 710064, China)¹

(School of Computer, Northwestern Polytechnical University, Xi'an 710072, China)²

Abstract Pervasive computing makes it possible to create an environment of focusing on the people. However, developers have spent much time and energy on building applications with feasibility and high satisfaction. In this paper, new challenges to the design and evaluation of pervasive applications were analyzed firstly, and then two hot technologies including development methodology and simulation testing were introduced and compared. The prospects for future development and suggestions for possible extensions were also discussed.

Keywords Pervasive computing, Development methodology, Prototyping toolkits, Simulation testing

普适计算的出现改变了人适应机器的被动式服务理念,在追求“以人为本”的当今时代背景下,普适计算成功地突破传统桌面计算模式下服务和交互方式的限制,成为信息时代中计算模式的一个里程碑。近年来,普适计算在医疗护理、军事、教学乃至生活辅助等领域得到了广泛应用,主要为以下方面提供了解决方案:

1) 模糊化信息空间与物理空间的界限,使得随时随地、透明地访问到合适的数字化服务成为可能;

2) 模糊化多种技术的领域限制,使得应用功能由单一向多能发展,执行空间由有限向无限拓展;

3) 模糊化人机交互的固态模式,使得存在于三维物理空间的语音、文字和图像等多模态数据扩展为具有语义的“时间-空间流”^[1]的接口表现。

各类新型技术的成熟发展,为构建复杂的普适计算应用提供了理论基础和技术支撑。然而构建具有可行性和高满意度的普适计算应用却是一项需要投入时间和精力复杂工程。可行性需要考虑应用所处环境的特殊性。普适计算环境的主要特点在于异构性的“无所不在”,这不仅指设备的种类繁多、彼此功能和能力各异,而且指设备间的网络通信和交互协议也是多样的。并且由于计算实体与用户在物理空间具有高度移动性,因此服务资源的可用性也将随之发生改变。技

术上的挑战,要求设计者需要将大量的时间放在底层技术的研究和实现上,这无形中增加了设计工艺的难度。

随着普适计算模式被广泛地应用在日常生活中,Weiser^[1]所提出的计算理念被重新诠释,目标受众的主观体验逐步成为应用的设计决策主导。“最大化效率,最小化干扰”是普适计算应用的核心目标,其强调了一切应用功能都需要以获得用户的高满意度为目标。例如,在健康看护应用中,提醒辅助是提高老年人服药依从性的一个重要手段^[2]。现有应用普遍具有上下文感知特性,能够自适应选择提醒发生的时间和方式。然而由于触发提醒会造成老年人注意力的转移^[3],因此不合理的实施提醒将会影响老年人的当前行为,甚至干扰正常生活。所以,在设计阶段忽略对用户需求的认知,即使提供再完美的功能,也会导致应用的可用性降低。在高满意度的需求下,构建普适计算应用依赖于设计者与用户的交互能力,这体现在了解用户与应用交互时彼此的行为等。然而,缺乏必要的设计方法指导和需求分析手段,设计者很容易忽视用户体验的重要性,导致应用的功能核心与交互界面脱节,后期维护成本增加。

传统的瀑布型开发方法难以解决普适计算复杂性和动态特征所带来的应用设计和评估问题。普适计算应用的设计重点在于以用户需求为导向,以自然的交互方式手段提供服

到稿日期:2013-10-24 返修日期:2014-02-17 本文受国家自然科学基金资助项目(61303041,61222209),国家重点基础研究发展计划(973)(2012CB316400)资助。

唐 蕾(1983—),女,博士,讲师,CCF会员,主要研究领域为普适计算、快速原型技术,E-mail: tanglei24@chd.edu.cn;周兴社(1955—),男,硕士,教授,主要研究领域为网络化嵌入式计算、普适计算;於志文(1977—),男,博士,教授,主要研究领域为普适计算、社会计算。

务。对于设计者来说,在花费较少时间和精力,能够根据创意概念正确地构建出一系列的装置,实现与用户的交互,获取真实的用户需求以修正或指导之后的设计,是非常有现实意义的。因此,设计者要打破设计常规,高速优质地在较短时间内,实现一个可供发展的设计方案并加以评价,这种工作方法便是快速设计与评估。

快速设计与评估技术为构建普适计算环境下的计算应用提供了方法指导与技术支持,然而,其强大的能力也使得它在实际应用中面临诸多的挑战。本文第1节强调传统方法在构建普适计算应用中所面临的挑战;第2节、第3节回顾现有的理论和方法,通过分析和归类,指出它们的特点和不足;第4节分析快速设计与评估技术在研究与实践中面临的问题,并讨论未来研究的若干方向;最后总结全文。

1 构建普适计算应用面临的挑战

以用户为核心的设计方法(UCD, User-Centered Design)提供了一整套工程实践方法,能够在应用设计、开发、维护阶段时刻高度关注并考虑用户的使用习惯、预期的交互方式等,以降低功能核心与用户体验脱节而带来的后期维护成本。然而由于在易变不确定的普适计算环境下,难以准确预测应用行为的触发条件(比如用户隐式输入)和执行结果(如环境对用户感官的影响),使得面向桌面计算模式的传统设计方法面临诸多挑战。

以典型的位置感知应用为案例,设计者需要充分理解用户情景中的位置语义,并建立位置情境与功能服务之间的唯一映射。使用传统的UCD方法,设计者仅能预测个体用户的特定位置实例,而无法捕获用户意图和简单位置信息之间的隐含关系,因此难以明确用户的需求。在对应用逻辑的设计层面上,设计者需要花费大量的精力来实现底层可用情境的建模和分析,同时仅采用if-then语句对自适应的功能服务进行全面实现,这会为实地部署带来较高的测试成本,甚至因设计不合理而造成对功能核心的反复修改。由此,在对普适计算应用的快速设计与评估工作中使用传统方法会面临以下挑战:

1) 认知使用者的状态,包括物理属性、技能和社会身份等方面,是UCD方法的首要任务,有利于更早地明确应用的非功能性需求,指导应用设计。在普适计算中,由于不同的用户场景为应用的设计决策带来很大影响,而实体高移动性带来的不确定性又会使得对环境、软硬件和使用者状态^[4]的刻画更加复杂,因此在初期设计阶段进行的用户分析工作将难以实施。

2) 将复杂计算应用分割为任务单元,以实现将任务按功能序列组合执行,这是以任务为中心(TCSD, Task-Centered System Design)的传统方法的设计思想,它有利于分析应用的功能需求和技术特性,实现用户目标。在普适计算中,任务内部混合了感知、决策、执行和通信等多种计算过程以及复杂物理过程,而且任务之间主要是以上下文感知的方式进行组合。这样,对任务的分析需要综合考虑物理与计算过程,而物理过程又是难以控制和不可预测的,这无疑增加了TCSD方法实施的难度。

3) 在普适计算中,由于底层设备的异构性以及应用领域的多样性,较其他计算应用,设计者需要采用更多的复杂计算

处理技术和理论模型来分析和处理底层数据,建立触发规则,识别行为模式以及在移动设备上实施部署,这些都无形增加了快速设计的难度。

4) 搭建测试环境有利于尽快地得到用户反馈。然而普适计算应用的执行涉及了在复杂多变的网络环境下多种异构设备之间的相互协作,因此实现系统全面部署的传统测试方法将会增加测试工作成本。

解决这些问题的关键在于对普适计算应用特性的解读,通过对功能实现、空间交互和用户体验视图的分析,设计支持普适计算应用特性的模型;提供设计方法学来支持对该模型的快速搭建和实现,这包括对模型元素的内容和关系的定义和对搭建步骤的指导;提供评估技术来对该模型描述的应用进行快速部署与评估。

2 普适计算应用的快速设计技术

快速设计技术是目前普适计算领域的研究热点^[5],近年来涌现了许多支持普适计算应用的开发系统及工具,这在PerCom、UbiComp等会议和权威期刊IEEE Pervasive Computing上均有报道。根据目标问题域的不同,现有的开发系统项目在侧重点上有较大差异,如有些强调传感器数据获取与聚合,有些强调上下文建模、聚合与推理^[6],有些强调透明的网络切换与应用自适应调整^[7],有些强调交互应用的设计与测试等。为应对快速设计在普适计算应用领域下的挑战,通过分析和归类,本文从快速设计的3个主要研究方向(应用模型、设计方法学、原型构建技术)分析现有理论和方法的特点及存在的问题。

2.1 应用模型

模型是对现实系统的简化,能够从不同层面对系统设计提供蓝图。通过建模,有助于加深对复杂系统或应用的理解,更好地在快速设计过程中对应用需求进行可视化刻画,并协助设计者正确地进行构造,以提高开发工作效率。在普适计算中,信息与物理空间的界限已模糊不可见,计算应用提供的服务受到众多不确定、难以预测的因素影响,比如用户意图、用户蕴含式交互、资源能力和计算环境等,因此系统需要根据自身的资源状态,采取灵活有效的策略来保证应用程序的可靠、稳定执行^[4]。随着应用在时空维度上的演化,这种对自适应的高度需求势必会給应用理解、设计及开发带来挑战。显然,对普适计算应用进行建模,建立简单易懂的模型和相应的符号语言,能够刻画物理与信息实体间的融合交互,为多团队协作提供依据与指导。

目前,UML等各种建模语言在描述系统内部功能方面都有很好的支持。然而普适计算应用的一个主要特性便是无所不在的交互,它强调多个实体在融合3C的环境中进行无缝连接,这使得交互来自于物理世界,反映在信息世界,决策在物理世界。在普适计算应用的整个执行过程中,计算和物理过程环环相扣、相互制约、相互影响,而用户状态(如偏好、物理特征、心理感受等)是这一切发生的主导因素。传统的建模方法主要面向具体的计算环境和设备,其接口、执行流程和实施行为的方式都是可以预测的,而在面向不确定的普适计算环境中,我们需要扩展这些建模方法,使之能够屏蔽设备的异构性和实体的高移动性,抽象出功能与非功能性需求,以帮助设计者快速、高效地描述计算应用。

2.1.1 应用运行环境

普适计算环境可以被看作是特定应用领域下的上下文信息集合,它记录了与用户任务相关的对象属性。因此应用模型需要处理此类异构多样、动态易变和非确定信息^[8],使其能够被统一、抽象地表达和在设计阶段被共享。通过分析现有上下文模型,本文总结了3类典型的模型结构。

1) 面向特殊应用的上下文模型,如麻省理工大学的AIRE项目、CoolTown项目^[9]等。

2) 基于经典理论,如ER库和UML图的上下文模型。

Henricksen等人以对象-角色模型为基础,针对如何统一标识4类异构上下文信息的问题,设计出图形上下文建模语言^[10],它具有直观、表达能力强的优势。Context Toolkit^[11]运用面向对象建模方法,构建多个可重用的插件作为传感器的抽象,它可屏蔽底层传感器的信息采集与原始上下文的融合解释过程,并向上层提供更易使用 and 理解的接口。

3) 基于本体的上下文模型。

由于本体支持对领域中的基本概念以及概念之间关系的语义化解释,因此基于本体的建模方式能够通过用户在用户、设备和服务之间共享上下文信息,来实现具有语义的互操作^[2]。最重要的是,它使得上下文推理变得可能,从而扩展了上下文的获取方式,大大提高了上下文的重用效率^[6,12-14]。Vanderhulst等人强调了本体模型在应用执行阶段的使用。在分布式架构的支撑下,他们将模型中服务和用户接口的概念直接映射为不同粒度的构件,以为应用执行提供更易用的上下文信息交换机制^[15]。

以上模型侧重于对上下文个体特征的领域性表述,包括属类(如资源、用户、设备或服务)、属性(如用户特征文档)和语义关系(如“has_a”,“is_a”)等。然而,它们缺乏对环境中多种上下文变化过程的刻画,无法为上下文约束条件提供设计合理性的验证指导。

2.1.2 应用的行为

为了适应高度动态的计算环境,普适计算应用需要实时响应上下文的变化,并实施适当的调整。依赖上下文模型,一些普适计算应用已经能够实现对用户位置、时间维度或其他有限环境因素的自适应^[16]。然而它们仅是采用基于条件判断的编程模式来定义不同环境下的实体及应用的执行方式。为了实施高效的应用自适应行为,降低因未知因素的影响而带来的重写修改代价,需要在应用的设计阶段预测与刻画其行为特征,即将环境与执行行为的相互制约性引入到应用模型中。

在大规模分布式系统中,事件机制能够很好地支持组件间低耦合性和异步交互方式。在融合随机物理过程和异步计算过程的普适计算应用中,事件机制能够有助于表述这种难以控制的变化。因此事件-条件-动作(ECA, Event-Condition-Action)规则被用于刻画这种受用户和环境因素主导的执行行为,并能够为其附加语义化信息。Vanderhulst将传统的ECA扩展为ECAA⁻¹规则,通过绑定上下文信息,允许对规则中的动作进行撤销操作^[15]。

面对高度的自适应需求,设计者需要构建一种灵活的应用逻辑组成机制,通过对不同功能模块的分离,降低模块之间、应用逻辑与环境之间的耦合度。在此背景下,构件化开发的思想被引入到普适计算应用模型中,用以指导对这类上下

文感知行为的刻画^[17-19]。作为典型实例,PCOM^[7]提出基于分布式构件的树状应用模型,该模型通过采用规约描述构件的需求(如执行平台等)和能力提供(如文本输入服务),实施基于需求-匹配机制的构件组合,以刻画灵活的应用行为。

上述模型将普适计算环境中多种计算实体的执行细节抽象为能力提供者,这使得普适计算应用可以被建模为一组具有交互意图的实体集合,有助于提高应用设计的效率和实现面向终端用户的可操作设计。然而,它们需要对各种不同构件实体进行能力标识和执行动作说明,并通常需要使用严格的规约来进行复杂的接口及事件定义,这无疑妨碍了设计者对应用模型的理解和使用。

普适计算应用的特性,要求其应用模型具备以下能力:

1) 能够建模存在于用户与系统之间的交互逻辑,包括使用的设备、用户的偏好等;

2) 能够将功能逻辑建模为可重构模块,包括计算过程、数据空间以及物理介质的信息化接口。

因此,面向普适计算的应用模型需要帮助设计者充分理解运行环境,包括操作对象和交互实体的物理属性和行为语义,同时提供有效的建模方法,支持对目标数据的快速访问、处理和对应用逻辑的高效设计。

2.2 设计方法学

传统的应用设计主要采用以功能驱动的方法,通过在初期一次性提取用户需求后,对需求映射下的功能模块进行单纯式的递增设计。在整个开发过程中,设计者通常假想“用户”角色,认为自身的经验或对市场调研的结果为用户所需,因此他们往往关注任务应该如何被执行,而不是用户希望如何执行。Gould在文献^[20]中指出一个成功的交互系统必须能够识别各种用户群,而且还可辨别各个用户所掌握的技能、经验以及他们的偏好。同样,一个成功的普适计算应用更是需要将交互的理念发挥至极致,因此对用户需求的解读在设计阶段更不可缺少,特别是在设计初期,这样可以确保此后各阶段中进行的递增式设计能够更好地适合框架,而且用户界面在外观、术语和概念上都能够保持一致。

2.2.1 以用户为中心的设计方法

UCD是指在设计过程中以用户体验为核心、强调用户优先的设计模式。Holleis结合UCD设计原则,对普适计算应用的开发过程给出了形式化的说明^[21]。首先需要进行初期的领域性研究,包括相关工作、感兴趣人群、用户期望及相关属性配置、应用的执行条件及场景和系统需求等。之后便是进行设计决策的工作,通过采用一系列递增式设计,将粗糙、低精度的纸质原型精化为可执行的应用实体或高精度的原型。在此过程中,评估的结果会作为触发重新设计的信号,不断推动原型的精化过程直至初期的需求被全部满足。接下来的工作便是开发具备完整功能的版本,包括设计窗口规范和图形化的具体布局。

由于UCD的良好设计理念,许多研究者以此为指导方法,开发出自己的应用原型^[22-25]。Fitton等人将UCD设计过程用于构建基于场景交互的Hermes普适计算系统^[26]。Fitton的研究工作以及后续对SPAM和Hermes2的开发体验都为以UCD驱动的普适计算应用开发提供了很好的实用经验。欧盟IST项目COGKNOW旨在运用UCD设计方法,指导开发认知设备与智能家庭技术框架,为独居的中度痴呆人

士提供4类日常生活保障^[27]。为了准确获取目标受众的需求和开展快速评估,COGKNOW采用群体调研方式,定义系统的功能需求,并将其映射为技术的实施细节,同时引入用户研究,不断修正功能需求,以期获得用户体验与技术实现的一致性。

考虑到在普适计算环境中实施用户实地评估所带来的高成本^[28],一些研究者尝试对UCD技术进行调整。Abowd提出了两类原型技术,其分别从对物理功能的评估和对用户交互体验的获取两方面来支持对普适计算应用的有效开发^[29]。以此为指导,研究人员开发了Personal Audio Loop^[30]项目,旨在为人们提供与语音相关的记忆辅助。为了快速测试PAL的有用性,研究人员构建了原型,以具体的实例模拟真实环境下的技术,来收集更多有关PAL的反馈信息。

UCD设计方法为实现普适计算“以人为本”的理念提供了可行之路,允许在功能核心与用户体验之间寻找平衡。然而,UCD方法对用户评估的强依赖性,使得在设计初期,即用户与低精度原型进行交互时,难以获得准确的反馈信息。特别是,由于用户往往对普适计算应用的交互模式不够理解以及对原型功能的实现程度要求很高,因此如何确定各开发阶段及其评估目标以快速获得需求,将成为采用UCD开发普适计算应用的一个待解决的问题。

2.2.2 以任务为中心的设计方法

考虑到在普适计算环境下,用户希望能够随时随地地访问服务,以执行其期望的计算任务,TCS D方法允许设计者从更高层的用户任务视角出发,灵活地编排应用的功能和提供的服务。卡内基梅隆大学研发的Aura项目^[31]便是将TCS D设计理念与普适计算结合的一个典型例子。它将用户工作抽象为任务,通过实现软硬件各层的主动行为,绑定用户意图与可用资源,使得用户即使是在移动的过程中,也能够不被中断地连续工作。Aura将任务的概念引入到对服务组合的实现中,这种思想在后续的研究中得到了充分的发展,并涌现出很多以任务为驱动的普适计算应用^[32-35]。与Aura等项目对任务的理解不同,Wurdel采用TCS D思想对基于组员协作的普适计算应用进行需求分析、设计与评估^[36]。特别地,他设计了协作任务模型(CTML, Collaborative Task Modeling Language),用于直观地描述任务的执行流程以及任务之间的调度。在TCS D的设计原则指导下,Wurdel使用CTML开展对单个任务、环境和任务间触发-响应关系的迭代分析,进而生成完整的、具有并发任务的执行图。

TCS D方法有助于令设计者追踪影响用户任务的因素并提供有针对性的服务支持,然而良好的设计应该是让用户在达成他们阶段性目标的时候,不影响他们终极的目标。目标不是任务,目标是终结条件,而任务是达到目标所需要的中间过程。目标是稳定的,任务是易变的。因此一些研究者开始面向具有更高抽象层次——任务目标的设计领域^[37]。针对需求分析阶段,Mylopoulos比较分析了面向对象和面向目标的设计方法。他指出:后者具有描述和评估对象间关系的结构,能够直观地跟踪系统功能和非功能性属性^[38]。面向普适计算环境,任务目标被理解为影响隐式人机交互的主要因素。在此背景下,研究者开始设计基于目标的应用交互行为。例如,在用户定义一个简单的目标(“我现在要看《新闻联播》节目”)之后,应用会自动执行相应的一系列动作来实现该目标。

尽管目前已存在相应的理论基础和架构支持,但是在开发方法学方面的研究却还处于起步阶段。特别是如何通过有效的研究手段来收集用户的阶段目标并提炼出终极目标,为用户提供最优设计,是目前有待研究者解决的科学问题。

2.2.3 以模型驱动的设计方法

模型驱动的开发方法(MDD, Model-Driven Development)强调通过对系统行为进行建模,带动整个系统的开发过程,包括需求分析、架构设计、构建、测试、部署和运行工作^[39]。正因为MDD提供了更加广泛的抽象理念以及更加标准的开发支持,研究者开始将MDD的设计思想移植到普适计算领域中,用来指导构建具有更高可用性和更易维护性的普适计算应用。Conejero以面向智能空间的应用开发为研究背景,重点选择MDD方法来指导构建与系统无关的测试过程,以期提高整个测试行为的可用性^[40,41]。与之前工作不同,Georgalas设计了基于MDD的集成平台,它允许设计者灵活地使用各类元模型来规范特定的技术配置文件,并利用模型转换机制,快速地生成与平台有关的代码实现。

近十年来对MDD的研究,使其在通用领域中已具备完善的设计理论和工具支持。针对普适计算,研究者的主要任务便是处理复杂易变的情境信息、用户与应用交互行为以及环境与应用之间的相互影响关系。因此MDD需要提供更多的元模型来刻画环境-人-应用之间的反馈过程。此外,掌握MDD的设计方法需要具备一定的技术背景,这使其难以满足普通用户定制个性化应用的需求。

2.3 快速原型方法及工具

原型来源于交互设计的范畴,它提倡设计者需要更加关注影响用户行为与习惯的各种因素。与传统软件设计关注于软件产品的功能和形态不同,交互设计的目的在于使用户在交互过程中获得良好的体验。因此,设计者往往需要根据创意概念构建出一系列装置,通过用户不断地与该装置进行交互,设计者可以获得第一手体验,从而验证想法,评估其价值,并为进一步的深入设计提供基础与灵感。这样的装置便被称为原型^[3]。

原型构建适合于描绘不同的设计状态与设计进程。根据原型的构建目的,其选择工具与技术的侧重点也有所不同。本文在PACT系统(People, Activity, Context, Technology)的基础上,从普适计算领域出发,结合多个研究项目的经验,给出3个观测方向,以帮助设计者更加了解原型的构建意图,从而决定使用何种构建技术来构建原型。面向快速设计过程,图1同时也指出了这3个方向的关系,即相互补充,不断融合。

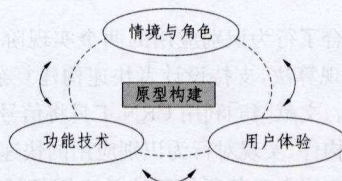


图1 选择原型的3个方向

1) 支持对情境与角色的原型构建

作为支持构建基于行为的应用原型工具,ActivityDesigner^[42]为设计者提供了创建、融合一类行为的场景机制,能够采用行为理论框架建模场景中提取的行为。设计者可通过ActivityDesigner提供的查询语言获取某类行为信息,并可以

此作为改变应用逻辑的条件,通过结合基于故事板的描述技术来支持原型的快速构建。目前 ActivityDesigner 已在设计健康监控应用等领域得到使用。

CogTool 旨在为设计者提供更加容易的方式来建立用户绩效模型。基于成熟的认知心理学理论,CogTool 结合 ACT-R(Adaptive control of thought-rational)计算引擎的预测能力,将 HTML 原型技术与 ACT-R 行为模型交互,产生一组能够预测用户绩效的按键序列,用于仿真用户与原型交互的真实体验。设计者利用 CogTool 提供的界面,能够快速构建基于 HTML 的原型,并利用 UI widget 描述任务流程。CogTool 简化了从任务编程到任务证明的模型处理过程,同时集成包含多种行为模型的 ACT-R 架构,使得设计者能够更容易地访问用户绩效模型。

为了提高原型的可用性,一个重要的任务便是建模角色的认知状态和行为驱动,这有利于在设计过程中为进一步的精细化提供可用的预测。Holleis 设计了 EIToolkit 原型工具^[21],它结合经典的按键层次模型,能够支持在缺少真实用户参与的情况下,对应用进行基于 UCD 的快速开发。

2) 支持对获取用户体验的原型构建

Reilly 等^[43]针对群组导航应用场景,设计了 3 类应用原型。在此基础上,他们探讨了用于有效评估原型可用性的方法。体验原型技术在于通过原型构建模拟所涉及的人、情境与设计物间的特殊局部关联以及整体上的联系来帮助深入理解现存的用户体验,因此 Reilly 将该原型技术用于研究者参与的真实场景中。

ActivityStudio^[44]提供了一种用于支持原型构建及实地测试的执行架构。其中 ActivityDesigner 采用高层的抽象技术,比如行为模型等,支持设计者快速创建原型,同时根据设计自动在 ActivityServer 上部署相应的计算结构,包括数据结构和测试模板,生成测试架构。在此框架下,ActivityStudio 设计了多种方式来支持原型的运行,同时提供了测试对话机制,以帮助设计者实时监控用户的行为以及终端设备的界面状态。通过这种方式,设计者能够从多个角度获得用户的真实体验,分析用户的行为模式,优化设计方案。

3) 支持对具有一定功能的原型的构建

VisualRDK 从屏蔽设备及服务器的目的出发,通过设计一类融合上下文信息的编程语言,支持对分布式普适计算应用的原型构建。开发者通过 VisualRDK 界面中提供的构件和简单的逻辑命令来设计程序结构,并采用其提供的解释器进行编译执行,能够以分布式的模式来监控数据和控制流的执行过程。

CRNT 融合了行为识别应用的两个实现阶段,通过提供识别方法和实现算法,支持设计者快速构建复杂的多模态上下文识别应用。文献^[45]利用 CRN 工具将信号分割、分类器等算法封装成构件,实现对行为识别应用的快速原型构建。

普适计算应用由于其所处环境及应用领域的多样性,需要考虑多种性能指标,这就要求在设计阶段解读其特点、需求及功能。目前的研究在支持原型构建开发方面做了大量的工作,但是在快速原型构建和评估方面还存在改进之处。例如,目前所支持的应用只能被动地反映信息,无法控制设备或环境,而且设计过程没有将计算和物理过程结合起来考虑,仅按照一类行为识别的实现流程进行模块封装。

3 普适计算应用的快速评估技术

在人类社会中,对正式融入人类日常生活之前的各类物品都需要开展不同程度的测试和评估,这往往造成实施完善的测试将消耗大量的人力与资源。例如,为了评估基于 Care-Net 的护理网络对老年人日常生活的辅助作用,Consolvo 等人花费 3 个月的时间将其部署在真实的环境中^[28]。同时,为了令参与者能够更好地提供准确的反馈,Consolvo 还需要对参与者进行细致的培训。由于普适计算环境具有异构性和动态性,为保证对应应用原型化的有效性,需要将设计场景从底层复杂的计算环境中抽离出来,进而实施快速部署与功能测试。以前的解决方案是在设计过程中生成与平台相关的应用版本,并利用架构实现其在计算设备上的部署和通信协作。测试者需要了解测试架构的搭建方法,并且完成对应应用功能的重构,这无疑为测试者带来了新的负担。

3.1 基于仿真测试技术的快速评估

普适计算系统是一类特殊的自治计算系统,它强调对人类及其所处环境的监控以及自适应调整。由于普适计算应用追求提高人类工作及生活的效率,并降低与计算机交互而带来的不舒适感,其运行环境通常为人类日常生活的一个地理映射,比如办公室或家居。在此背景下,仿真技术能够允许设计者多次重现设计场景以及复杂的情境信息,来评估普适计算应用对情境信息的反应效果,这也为管理和解决此类自适应问题提供了技术支持。

在快速设计过程中,原型是为了更好地理解应用的运作,及时进行用户测试并更新设计。测试贯穿于整个原型的构建过程,在初期,即在功能实现不完整的情况下,为保证获取真实的用户体验,仿真测试技术能够对场景进行仿真识别和简单的仿真控制,比如仿真传感器数据、上下文信息或系统智能行为,这可以降低原型开发部署的成本。

Reilly 等人将 Wizard of Oz(WOz)技术用于对群体交互应用中用户位置以及交互信息^[43]的模拟。其中,每位测试人员跟随一位参与者,他利用蓝牙通信获取该参与者的交互行为,并利用对讲机为另一组测试群提供最新的位置和交互信息。DART 原型工具^[46]能够允许设计者快速地将多媒体技术集成至普适计算应用中。在其提供的应用示例——参观奥克兰公墓中,WOz 构建技术被用来模拟位置传感器的探测和通讯功能,可以实现对物理与信息世界的融合,便于对依赖于底层感知、通信技术的设计问题进行早期的实地测试。为了获得参与者的实时位置,Topiary 为测试人员提供了特定的仿真操作界面,使其能够简单地模拟位置情境信息的产生。此外,参与者的操作界面也能够实现场景的重构,方便其对应用执行的结果进行快速评估^[47]。

除了使用 WOz 技术来仿真传感器数据,其他研究者从分析传感数据的来源、特征和反馈影响入手,模拟真实传感器数据的输出^[48-51]。为了验证系统的可扩展性,Sanmugalingam 在 QoSReAM 中间件^[52]上部署了基于事件的仿真模型,用于模拟物理空间中用户的移动行为。为了更好地建模人类移动过程中的场景,即排队等待、直线移动和躲避障碍物等,该模型引入经典的连续力学模型、离散队列模型和传感器模型等,来模拟在加速度以及位置信息未到达一定阈值时用户在虚拟空间下的持续移动行为。

3.2 仿真环境下的快速评估

为保证应用的可行性,同时降低开发和部署的成本,搭建有效的仿真测试环境是另一种比较合理的解决方案。通过研究对设备能力、数据事件流、实体的仿真实现,与平台的底层实现集成,可以有效地检验应用的性能以及为之后真实环境下的部署提供可行性方案。UbiWise 为设计者提供了一个基于 3D 模型的视图,其能够在仿真模型的基础上抽象出环境中的时空物理量^[53]。UbiWise 绑定虚拟设备的执行动作与用户的键盘、鼠标动作,利用消息通信机制实现环境视图与设备视图之间的交互,以仿真设备的行为和用户的交互体验。基于 Half-Life 游戏引擎,TATUS 采用 TOSSIM 仿真器来模拟传感器数据,并通过消息机制控制 3D 仿真环境下实体的自适应行为^[54]。以 TATUS 测试平台为基础,McGlinn 等人采用 SimCon 仿真工具,为面向智能建筑应用的设计与评估提供可视化支持。特别地,设计者能够自由地部署传感器的位置,来分别评估在静态与动态部署下建筑物对信号的影响或者信号强度对受控应用的影响。与此不同,UbiReal 主要关注普适计算应用对环境的反馈影响^[55]。它能够模拟环境中物理元素(比如温度)在时间维度上的变化过程,因此,设计者可以通过定义特定的情境实例和相应的规则,来描述和测试应用场景的变化结果。

完整的测试是由一系列基本步骤组成,而在不同的迭代设计阶段,测试工作具有不同的目标,因此在构建测试原型时需要明确需求。Driver 在分析近年来普适计算领域内的测试研究工作后,归纳了 5 类评估指标:可用性、无干扰性、功能可行性、接受度和可迭代性^[56]。这些指标能够在一定程度上帮助测试者针对性地选择评估公式,从而得到有效的评估结果。尽管目前的快速评估技术能够为收集测试数据和分析结果数据提供有效的解决方案,但是它们缺乏对应用部署的支持。测试者需要了解技术的实现细节以及仿真器或环境的搭建方法,这同样增加了测试者的负担。

4 研究发展的难点与热点

普适计算应用由于其所处环境及应用领域的多样性,需要考虑多种性能指标,这就要求在应用的设计阶段对其特点、需求及功能进行解读。现有研究在支持原型构建开发方面做了大量的工作,但是在快速设计和评估方面还存在改进之处。

4.1 刻画普适计算应用特性的模型

针对普适计算应用的本质特性,良好的应用模型需要具备以下特性。

1) 具备分布性和可扩展性

模型需要描述物理空间(设备)和信息空间(数据、控制流)的分布特征,扩展传统的编程方法,通过提供一套语义规范来屏蔽环境和设备的复杂异构性,支持对远程资源的访问控制,同时设置接口来说明资源的差异性,便于应用在执行阶段发现和选择合适的资源。同时,为了能够在分布易变的普适计算环境下实现数据的跨平台重用,应用模型需要解耦数据和功能,实现与数据无关的应用演化设计。

2) 具备感知性和决策性

模型需要描述上下文的发现和表示,以及上下文与场景之间的映射关系。同时,作为感知发生的前提,模型同样需要具备发现和通知上下文变化的能力,比如用户位置的变化和

执行设备的能力变化等。

3) 具备作用性

模型需要描述场景的作用域和多种可预测上下文的变化。同时该模型需要具备预测性,即可描述整个执行环境的变化。

4) 支持资源发现、松耦合通信以及抽象的人机交互机制

为了更好地实施“不可见计算”,普适计算应用需要持续地对高度动态的计算环境进行辨析和自身状态的调整,从而为用户提供满足其期望的功能。普适计算强调应用需要从用户交互意图出发,时刻跟随其需求,提供能够满足用户对功能和非功能性需求的服务。因此,应用模型需要通过建立抽象人机交互机制,提取用户的交互目的,并建模其产生的交互行为与环境的映射关系,使其具备上下文感知与反馈特性。

4.2 开发方法及快速原型技术

一个好的开发方法能够允许设计人员快速准确地解读用户需求,构建适应用户特点和任务的普适计算应用,而不是局限于开发一类面向特定环境,使用特定设备进行人机交互的应用。它将确保在开发各阶段中进行的递增式设计能够更好地适合框架,而且用户界面在外观、术语和概念上都能够保持一致。

作为探索与表达设计的重要媒介与手段,原型的构建贯穿于整个设计流程与评估中。在原型构建中,需要通过不断地调查、测试和调整设计来创造复杂的结构。而在此过程中,根据不同的设计阶段,原型的构建意图不同,且适合描述的原型也需要相应发生更改,因此需要研究多种类型的原型构建技术。

1) 原型及构建技术的选择

在不同的设计阶段,对每种原型的精度定义上有不同的侧重点,比如低精度原型以其简单、价格便宜与快速构建的特点,能够适应早期开发阶段的需求。因此需要针对普适计算应用的设计意图(用户感官、功能等)以及构建阶段,研究精度的定义和评价指标,并以此标准来衡量每类原型的精度是否满足应用设计的需求。同时需要研究对不同原型技术的选择,并进行技术融合以实现当前阶段的原型构建方案。

2) 计算与物理过程协同构建技术

普适计算环境的终极目标是实现物理空间与信息空间的完全融合,原型是实现信息空间与物理空间融合的纽带。因此在原型的构建中,需要研究其所处场景下的计算与物理过程,并在原型中得以体现。

3) 获取和描述物理上下文机制

存在于物理空间中的上下文信息是普适计算应用的一个核心要素,其获取及访问也是原型构建需要研究的一个重要方面。针对面向特定场景下的多种上下文信息,不同原型需要不同技术来实现不同上下文的仿真或实地获取。同时,原型需要提供友好的界面来支持上下文的抽象建模机制以及查询语言。

4) 集成多种典型数据分析处理模块

由于物理设备的异构性以及应用领域的多样性,普适计算应用需要采用多种复杂的计算处理技术和理论模型来分析和处理领域数据。因此将复杂的算法或处理技术封装为可扩展模块提供给设计者,将是原型构建乃至平台研究的一种发展趋势。

4.3 基于应用模型的快速验证

在普适计算环境下,应用的行为实质上由一系列情境驱动的 ECA 规则所决定。在真实环境下存在的不可预知的情境演化,为设计完备的应用逻辑带来了挑战^[57]。针对普适计算应用,本文认为模型验证的主要目的是检验设计对上下文感知行为刻画的正确性。

设计人员往往只关注于对一种特定类型上下文的指派设计,而忽略了场景在真实环境下执行时可能出现的情境约束,即多种上下文之间存在的制约关系。这将导致在由应用执行而引发的上下文变化过程中,基于预测的约束条件可能无法被满足。为了帮助设计人员合理地设置应用逻辑中的约束条件,需要验证设计场景的可行性,即在存在上下文变迁过程的执行环境中,验证条件的可满足性。

4.4 以用户为中心的评估技术

快速评估贯穿于整个原型的构建过程,在初期,即在实现的功能不完整的情况下,为保证获取真实的用户体验,适当地采用仿真技术,对场景进行仿真识别和简单的仿真控制,可以降低原型开发部署的成本。由于在不同阶段,评估技术有不同的需求(可用性、有用性、接受度等),因此需要对多种技术进行融合使用。

在快速设计后期,当系统功能开发完毕,需要进行原型的实地测试。然而由于部署过程是一项困难且耗时的工程,因此支持将原型快速部署在现场环境的技术将成为研究重点。为此,需要研究原型的评估框架及其自动部署机制。

4.5 快速开发平台及图形化工具

Banavar 在文献[58]中提出,开发人员面临的主要挑战在于缺乏支持完整方法学的软件系统平台。为了应对此挑战,需要分析支持普适计算应用开发的具体需求,将基于 UCD 的快速设计技术、原型构建技术与快速评估技术结合起来,提供易用的图形化抽象工具库,使设计者在关注用户需求和体验的情况下能够快速地进行设计及开发具有高满意度的普适计算应用。

结束语 步入 21 世纪,Weiser 所提出的普适计算理念已被重新诠释,目标受众的主观体验逐步成为计算应用的主要评判标准。在此背景下,普适计算应用设计的重点便需要以用户需求为导向,而且能够以用户可接受的方式有效地执行功能服务。近年来,UCD 设计理念、交互式原型系统和仿真环境的引入,进一步提高了设计的高效性和用户的满意度,促进了学科交叉融合的研究。目前,国内外关于普适计算应用设计与评估领域的综述性文献还很少见,我们将该领域的研究进展进行归纳总结,探寻对构建普适计算应用的有效支持,希望能对我国该领域的发展起到抛砖引玉的作用。

参 考 文 献

[1] 徐光佑,陶霖,等. 普适计算模式下的人机交互[J]. 计算机学报, 2007,30(7):1041-1053
[2] 郑笛. 基于上下文感知服务的构件化中间件关键技术研究[D]. 长沙:国防科学技术大学,2008
[3] 华梅立. 交互设计中的原型构建研究[D]. 无锡:江南大学,2008
[4] Satyanarayanan M. Pervasive computing: Vision and challenges [J]. Personal Communications, IEEE, 2001, 8(4):10-17
[5] Davies N, Landay J, et al. Guest Editors' Introduction: Rapid Prototyping for Ubiquitous Computing [J]. IEEE Pervasive

Computing, 2005, 4(4):15-17

[6] Gu T, Pung H K, et al. Toward an OSGi-based infrastructure for context-aware applications [J]. Pervasive Computing, IEEE, 2004, 3(4):66-74
[7] Becker C, Handte M, et al. PCOM: a component system for pervasive computing[C]//PerCom 2004. Orlando, FL, USA: IEEE, 2004:67-76
[8] Grimm R, Davis J, et al. System support for pervasive applications[J]. ACM Transactions on Computer Systems (TOCS), 2004, 22(4):421-486
[9] Kindberg T, Barton J. A Web-based nomadic computing system [J]. Computer Networks, 2001, 35(4):443-456
[10] Henricksen K, Indulska J. Developing context-aware pervasive computing applications: Models and approach[J]. Pervasive and Mobile Computing, 2006, 2(1):37-64
[11] Dey A K, Abowd G D, et al. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications[J]. Human-Computer Interaction, 2001, 16(2):97-166
[12] Ye J, Coyle L, et al. Ontology-based models in pervasive computing systems[J]. The Knowledge Engineering Review, 2007, 22(4):315-347
[13] Chen H, Perich F, et al. SOUPA: standard ontology for ubiquitous and pervasive applications[C]//MobiQuitous 2004. Cambridge, MA, USA: IEEE, 2004:258-267
[14] Preuveneers D, Van den Bergh J, et al. Towards an extensible context ontology for ambient intelligence[M]. Ambient Intelligence. Springer, 2004:148-159
[15] Vanderhulst G. Development and Deployment of Interactive Pervasive Applications for Ambient Intelligent Environments[D]. Hasselt: Hasselt University, 2010
[16] Baldauf M, Dustdar S, et al. A survey on context-aware systems [J]. International Journal of Ad Hoc and Ubiquitous Computing, 2007, 2(4):263-277
[17] Augustin I, Yamin A C, et al. ISAM, joining context-awareness and mobility to building pervasive applications[J]. Mobile Computing Handbook, 2004, 1:73-94
[18] Jang S. Implementation of Context-Aware Application Model in Smart Environments[D]. Gwangju Institute of Science and Technology, 2005
[19] Urbanski S, Handte M, et al. Experience using processes for pervasive applications[C]//Proc. Informatik. 2006:60-64
[20] Gould J D, Boies S J, et al. The 1984 Olympic Message System: a test of behavioral principles of system design[J]. Communications of the ACM, 1987, 30(9):758-769
[21] Holleis P. Integrating Usability Models into Pervasive Application Development[D]. Munich: Ludwig Maximilian Muenchen Universitaet, 2009
[22] Arnstein L, Hung C Y, et al. Labscape: A smart environment for the cell biology laboratory [J]. IEEE Pervasive Computing, 2002, 1(3):13-21
[23] Schmidt A, Terrenghi L, et al. Methods and guidelines for the design and development of domestic ubiquitous computing applications[J]. Pervasive and Mobile Computing, 2007, 3(6):721-738
[24] Abowd G D, Mynatt E D, et al. The human experience[J]. Pervasive Computing, 2002, 1(1):48-57

- [25] Consolvo S, Arnstein L, et al. User study techniques in the design and evaluation of a ubicomp environment[C] // UbiComp 2002, Göteborg, Sweden; Springer, 2002; 73-90
- [26] Fitton D, Cheverst K, et al. Rapid prototyping and user-centered design of interactive display-based systems[J]. *Pervasive Computing*, IEEE, 2005, 4(4): 58-66
- [27] Davies R, Nugent C, et al. A user driven approach to develop a cognitive prosthetic to address the unmet needs of people with mild dementia[J]. *Pervasive and Mobile Computing*, 2009, 5(3): 253-267
- [28] Consolvo S, Harrison B, et al. Conducting in situ evaluations for and with ubiquitous computing technologies[J]. *International Journal of Human-Computer Interaction*, 2007, 22 (1/2): 103-118
- [29] Abowd G D, Hayes G R, et al. Prototypes and paratypes: Designing mobile and ubiquitous computing applications[J]. *Pervasive Computing*, IEEE, 2005, 4(4): 67-73
- [30] Hayes G R, Patel S N, et al. The personal audio loop: Designing a ubiquitous audio-based memory aid[C] // Mobile Human-Computer Interaction-MobileHCI 2004. 2004; 168-179
- [31] Garlan D, Siewiorek D, et al. Project aura: Toward distraction-free pervasive computing[J]. *IEEE Pervasive Computing*, 2002, 1(2): 22-31
- [32] Truong K N, Huang E M, et al. CAMP: A magnetic poetry interface for end-user programming of capture applications for the home[C] // UbiComp 2004; Ubiquitous Computing. 2004; 143-160
- [33] Messer A, Kunjithapatham A, et al. Interplay: A middleware for seamless device integration and task orchestration in a networked home[C] // PerCom 2006. Pisa, Italy; IEEE, 2006; 296-307
- [34] Ni H, Abdulrazak B, et al. CDTOM: A Context-driven Task-oriented Middleware for Pervasive Homecare Environment[J]. *International Journal of UbiComp*, 2011, 2(1): 34-53
- [35] Rosi A, Bicocchi N, et al. Environmental Monitoring and Task-Driven Computing[J]. *Pervasive Computing*, IEEE, 2010, 9(4): 48-50
- [36] Wurdel M, Sinnig D, et al. Task-Based Development Methodology for Collaborative Environments[C] // HCSE/TAMODIA 2008. Pisa, Italy; Springer, 2008; 118-125
- [37] Saif U, Pham H, et al. A case for goal-oriented programming semantics[C] // UbiComp Workshop 2003. Seattle, WA, USA; Springer, 2003; 74-83
- [38] Mylopoulos J, Chung L, et al. From object-oriented to goal-oriented requirements analysis[J]. *Communications of the ACM*, 1999, 42(1): 31-37
- [39] Mellor S J, Clark T, et al. Model-driven development: guest editors' introduction[J]. *IEEE Software*, 2003, 20(5): 14-18
- [40] Conejero J M, Clemente P J, et al. A model-driven approach for reusing tests in smart home systems[J]. *Personal and ubiquitous computing*, 2011, 15(4): 1-11
- [41] Georgalas N, Economou D. Model-driven development of pervasive applications using context and ontologies; an m-commerce case study[J]. *Electronic Journal for E-Commerce Tools & Applications*, 2009, 3(1): 182-213
- [42] Li Y, Landay J A. Activity-based prototyping of ubicomp applications for long-lived, everyday human activities[C] // CHI 2008. Florence, Italy; ACM, 2008; 1303-1312
- [43] Reilly D, Dearman D, et al. Evaluating early prototypes in context: trade-offs, challenges, and successes[J]. *Pervasive Computing*, IEEE, 2005, 4(4): 42-50
- [44] Li Y, Landay J A. Into the Wild: Low-Cost UbiComp Prototype Testing[J]. *Computer*, 2008, 41(6): 94-97
- [45] Bannach D, Lukowicz P, et al. Rapid prototyping of activity recognition applications[J]. *Pervasive Computing*, IEEE, 2008, 7(2): 22-31
- [46] MacIntyre B, Gandy M, et al. DART: a toolkit for rapid design exploration of augmented reality experiences[C] // UIST 2004. Santa Fe, NM, USA; ACM, 2004; 197-206
- [47] Li Y, Hong J I, et al. Topiary: a tool for prototyping location-enhanced applications[C] // UIST 2004. Santa Fe, NM, USA; ACM, 2004; 217-226
- [48] Sanmugalingam K, Coulouris G. A generic location event simulator[C] // UbiComp 2002. 2002; 95-122
- [49] Huebscher M C, McCann J A. Simulation model for self-adaptive applications in pervasive computing[C] // DEXA Workshops 2004. Zaragoza, Spain; Springer, 2004; 694-698
- [50] McGlinn K, O'Neill E, et al. SimCon: A Tool to Support Rapid Evaluation of Smart Building Application Design using Context Simulation and Virtual Reality[J]. *Journal of Universal Computer Science*, 2010, 16(15): 1992-2018
- [51] O'Neill E, Lewis D, et al. A simulation-based approach to highly iterative prototyping of ubiquitous computing systems[C] // SimuTools 2009. Rome, Italy; Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2009; 1-10
- [52] George H N, Coulouris G. Location Information Management[C] // Ubicomp 2001. Atlanta, GA USA; Springer, 2001; 35-41
- [53] Barton J J, Vijayaraghavan V. UBIWISE, a simulator for ubiquitous computing systems design[R]. HPL-2003-93. 2002
- [54] O'Neill E, Klepal M, et al. A testbed for evaluating human interaction with ubiquitous computing environments[C] // TRIDENTCOM 2005. Trento, Italy; IEEE, 2005; 60-69
- [55] Nishikawa H, Yamamoto S, et al. UbiREAL: Realistic smart-space simulator for systematic testing[C] // UbiComp 2006. Orange County, CA, USA; Springer, 2006; 459-476
- [56] Driver C, Linehan E, et al. Analysis of the Evaluation of Application-Led Research in Pervasive Computing[R]. TCD-CS-2006-26. Trinity College, 2006
- [57] Achilleos A, Yang K, et al. Context modelling and a context-aware framework for pervasive service creation: A model-driven approach[J]. *Pervasive and Mobile Computing*, 2010, 6(2): 281-296
- [58] Banavar G, Beck J, et al. Challenges: an application model for pervasive computing[C] // MobiCom 2000. Boston, Massachusetts, USA; ACM, 2000; 266-274