

基于感知控制的代理体系结构模型*

华庆^{1,2+}, 刘庆芳^{2,1}, 蔚 娣^{2,1}, 王小文^{2,1}

¹(中国科学院 软件研究所 计算机科学国家重点实验室,北京 100190)

²(西北大学 信息科学与技术学院,陕西 西安 710127)

Perceptual-Control-Based Agent Architecture Model

HUA Qing-Yi^{1,2+}, LIU Qing-Fang^{2,1}, YU Di^{2,1}, WANG Xiao-Wen^{2,1}

¹(State Key Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

²(School of Information Science and Technology, Northwest University, Xi'an 710127, China)

+ Corresponding author: E-mail: huaqy@nwu.edu.cn

Hua QY, Liu QF, Yu D, Wang XW. Perceptual-Control-Based agent architecture model. Journal of Software, 2009,20(Suppl.):76-83. <http://www.jos.org.cn/1000-9825/09010.htm>

Abstract: This paper presents a perceptual-control-based agent architecture which enable to construct interactive software system with high usability. Adopting this architecture, the usability requirements can be modeled by adding new architecture levels, in order to realize the orthogonal relationship between different features of usability. And it adopts perceptual-control theory to match the nonlinear relationship between the user-interface and application core. Comparing to the other architectures, systems adopting this architecture can present users with user-interface elements at the user-task level, and permit users to operate the system at the user-task level. The users, therefore, can naturally complete their tasks and reach their goals. Furthermore, the systems adopting the architecture can easily extend the usability dynamically.

Key words: perceptual control; agent; architecture; usability

摘 要: 提出一个基于感知控制代理的体系结构模型,旨在建模高可用的交互式软件系统.该模型通过增加新的体系结构层次来显式地建模可用性需求,以实现不同可用性特征之间的正交关系,并采用感知控制理论来匹配用户界面和应用核心之间的非线性关系.与现有的体系结构模型相比,该模型不仅使系统能够呈现用户任务层次的界面元素,而且允许用户在任务层次上对系统进行操作,从而使用户以更自然的方式控制系统来完成任任务,实现目标.此外,采用该模型的系统还可以实现可用性的动态扩充.

关键词: 感知控制;代理;体系结构;可用性

无处不在的计算使得当前的软件可用性由第一代逐渐过渡到第二代.第一代可用性主要集中在给用户提供正确的“观与感”,使得用户可以对系统进行控制,并看到系统的输出结果,由此导致用户界面和应用核心的分

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2006AA01Z328 (国家高技术研究发展计划(863)); the Open Foundation of State Key Laboratory of Computer Science, the Chinese Academy of Sciences under Grant No.SYSKF0704 (中国科学院计算机科学国家重点实验室开放基金)

Received 2008-09-20; Accepted 2009-04-09

离,出现了MVC,PAC等体系结构,但其界面是领域的直接表示;今天的第二代可用性所关注的问题是给用户提供一个以用户任务为导向的界面,使用户能在任务层次上与系统进行共享控制^[1]。例如,数码相机能提供用户想要的照片显示(输出形式),并允许用户直接操作于该显示(如增减亮度或对比度,系统将其转换为对光圈和快门的控制)。然而仅仅依靠对话独立性,即简单地将用户界面和应用核心相分离,只能表示“观与感”相关的可用性。因此,有必要增加新的体系结构层次来显式地表示新型的可用性需求。当今很少有有效的方法或模型支持软件体系结构层次的可用性设计^[2]。

今天多数系统的界面直接显示应用领域的业务逻辑,或者系统直接按照固定模式代替用户完成任务。前者,直接呈现应用领域状态,用户必须自己计算任务模型与领域模型之间的复杂关系,并控制领域层次的输入变量来完成的任务,如传统相机需要用户控制应用领域变量(光圈和快门);后者,利用自动代理取代用户进行控制,通常无法满足用户意图的需要,如自动相机在目标层次上利用软件代理提供命令,不允许用户对照片特征(如对比度、亮度、聚焦等)进行控制。

当前采用的软件体系结构是按照“观与感”划分系统的层次与构件,满足了构建第一代可用性的需要,而共享控制与“观与感”之间是正交关系^[3],即两者之间相互独立。因此,这些体系结构很难同时建模不同的可用性需求,必然会造成有些可用性分散在体系结构的多个层或构件中,增加了构件之间的耦合度,实现时难以跟踪和维护。许多可用性需求是在软件生命周期的后期(如测试阶段和评估阶段)发现的^[4],已有的体系结构模型并不能有效地支持迭代设计过程中的可用性动态修改和扩充。

本文提出一个基于感知控制的代理模型,旨在增强软件体系结构的可用性,特别是与共享控制相关的新型可用性。该结构模型在界面和应用核心之间增加新的层次——感知控制代理层,利用与“观与感”相独立的感知代理来显式地建模上述新型可用性需求,并可以动态增加代理单元来实现动态扩充的可用性需求。此外,该模型利用用户任务数据总线和应用领域数据总线隔离用户界面和功能核心的相互影响,通过感知控制代理单元来匹配用户界面和功能核心之间的非线性关系。

1 相关研究

交互范型的发展驱动了交互式软件体系结构研究。如,基于语言的Seeheim^[5]模型易于用于解释交互式系统设计过程和软件体系结构的组织,但Seeheim模型存在着过分强调对话的语法性质,而忽略了对话的动态性质,易于引起体系结构失配,难以满足GUI直接操作信息对象的需要。Arch模型^[6]是一种整体分解的风格,应用语义和表示之间的距离仍然存在。MVC^[7]从系统的功能角度把软件系统分成了显示层、控制层和模型层,这样的设计可以保证系统的功能性而不能很好地解决软件的可用性。这些模型及其衍生物的主要特征是强调“对话独立性”:用户界面和功能核心构件的设计决策应当从应用语义上分离,在低级功能层次上给用户提供一个“观与感”的图形用户界面。

20世纪90年代前后,研究者已经发现从应用语义上实现对话独立性是困难的,界面与功能核心之间对话控制和信息通信的粒度并非仅仅依赖于领域的特征,导致了界面和功能构件必须包含某种形式的计算^[8],或者两者之间需要实现一个有关控制策略和信息交换的协议^[9]。然而,从系统内部不易实现这样的计算或协议,因为建模它们的信息来自于系统外部的使用上下文^[10]。这导致了对话控制和通信的实现一直是一种艺术或技巧,依赖于设计者的知识和经验。传统的软件体系结构面对今天的可用性,暴露出来很多不足,如可用性不易跟踪、可用性难以修改等。

针对上述问题,已有学者进行了初步研究。如,Len Bass和Bonnie E. John列举了多个可用性场景,并且每个可用性场景都提供了与之相对应的架构模式^[11]。这些架构模式只支持需求分析阶段已明确的可用性需求,灵活性较差。而多数可用性需求是在开发后期才发现的,因此这些架构模式也不能有效地支持迭代设计过程中可用性的动态扩充。又如,李光俊等人通过增加面向方面构件来建模动态变化的可用性需求^[12],但这只是在已有体系结构上的修补,没有摆脱传统的软件体系结构所面临的可用性问题。以上研究仅仅是对已有体系结构的简单改进,并没有改变结构层次,面对不断变化、不可预知的人类需求则显得无能为力。而我们提出的基于感知控制代理

的体系结构可以满足第二代可用性所需的动态变化。

2 感知控制理论的分析和研究

感知控制是人类面对复杂的外界环境,为达到预期目标而采取的行为方式.感知控制理论是一种一般的心理学理论,其显著特征是“所有的行为都是感知控制”^[13],它认为一切生命都是层级地组织起来的负反馈系统,生命行为在所有时间里都是按特定目的对某种变量的控制.它不同于传统的行为理论,认为所有的行为都是刺激所引起的;也不同于认知心理学理论,认为所有的行为都是由计划决定的.感知控制是个体与外界环境交互的 3 种方式之一,其核心是控制,个体通过主动控制自身和环境的状态来使系统满足参考状态,并维持此参考状态,使外界环境对系统的干扰降到最低.感知控制系统在控制过程中主动感知自身当前的状态,并将当前状态与参考进行比较,得出两者之差,然后通过执行动作,调整自身和外界的状态,使其逼近,直至达到参考状态.

基于感知控制的任务分析^[14]与人机界面设计直接相关,它描述了显示变量和影响这些变量的动作,告诉用户界面设计人员应该将哪些变量显示在用户界面上,以及给用户哪些输入,即用户如何与系统交互.它将用户任务抽象为 4 部分(如图 1 所示):a) 显示变量:是系统提供给用户的可以控制的物理变量或信息化变量.b) 任务目标:存在于用户的大脑中,表示用户期望系统达到的意图状态,即用户期望显示变量达到的目标状态.c) 输入动作:是用户可作用于系统的动作,用于控制系统使其状态达到期望状态.d) 外界干扰:是环境对系统状态的干扰作用,通常是由于多用户系统中其他用户的动作.

本文将感知控制理论应用于软件代理,将软件代理看作是具有感知控制能力的独立个体,在此基础上提出了代理逻辑结构(如图 2 所示),此结构不同于传统的体系结构模型.在传统的体系结构模型中,界面元素是应用领域模型的直接表示,它们之间是简单的线性关系.然而用户任务和对象模型之间是复杂的非线性关系,很难用精确模型来建模.代理逻辑结构应用了感知控制理论,通过增加新的感知代理层次来匹配上述的非线性关系,更加贴近于人类思维方式,有助于设计面向用户任务的软件系统.

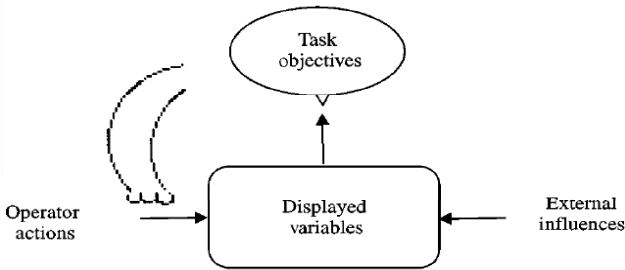


Fig.1 Perceptual control analysis of task^[14]

图 1 基于感知控制的任务分析模型^[14]

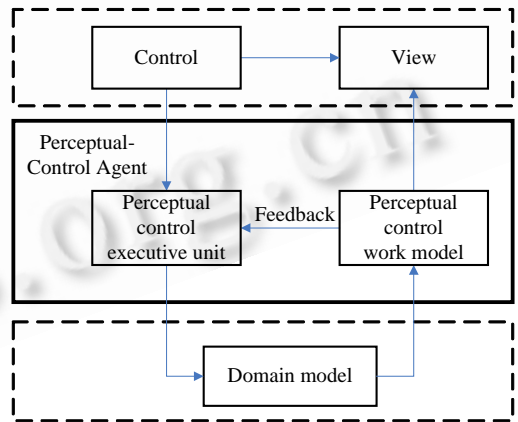


Fig.2 Logical structure of agent

图 2 代理逻辑结构

在此代理逻辑结构中,界面元素是用户任务模型^[15]的直接表示,并非应用领域模型的直接表示,用户可以在任务层次上与界面元素交互(即给感知控制代理设置参考状态).此结构的软件代理层包括感知执行器与感知控制工作模型.感知执行器用于匹配任务目标(高级功能层次)和应用领域(低级功能层次)之间的非线性关系;感知控制工作模型将自动分析领域模型的运算结果,同时与用户期望进行比较,并将比较结果反馈到感知执行器,以进行迭代运算.通过负反馈机制加速系统状态达到参考状态的过程.在任务执行过程中,那些给用户带来沉重负担的记忆与计算工作将被委派给系统,用户只需在任务层次操作于系统输出即可达到任务目标.

3 基于感知控制代理的体系结构模型概述

基于感知控制代理的体系结构利用了感知控制理论、软件代理理论和数据驱动协调理论.该体系结构可以使用户界面和功能核心在语义上分离,以隔离用户界面和功能核心之间的耦合.同时,在用户界面和功能核心之间增加代理单元.代理单元不仅是连接用户界面和功能核心的桥梁,而且还可以匹配用户界面语义和功能核心语义之间的非线性关系.代理单元内部存储了用户任务的历史状态,利用此历史状态可以帮助实现系统的可用性^[16],如“支持撤销”^[17].

在基于感知控制代理的体系结构中,按照模块内部所处理语义的层次不同从上到下分成3个层次、两条总线:用户界面层、用户任务数据总线、感知控制代理层、应用领域数据总线、功能核心层(如图3所示).

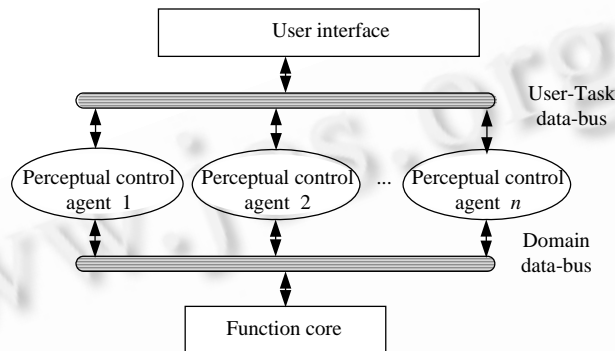


Fig.3 Agent-Based architecture model

图3 基于代理的结构模型

- 用户界面层.该层将给用户提供更接近大众心理模型的交互界面(如图形表示、直接操作、隐喻等),允许用户在任务层次上对系统进行操作控制.该层除了采用标准的图形控件之外,还将在主窗口内给用户提提供隐喻物理世界信息的交互式图形界面元素,并允许用户直接操作.利用用户界面工具箱(如OOIGT^[18])可以快速构建以用户任务目标为导向的交互式图形用户界面.

- 感知控制代理层.该层可以看作是一个有自治控制逻辑的独立智能体,它既是领域匹配构件,又是界面与功能核心的连接构件.该层与上、下层的通信是基于数据驱动的协调模式,即与用户界面层的通信通过用户任务数据总线完成,它与功能核心的通信通过应用领域数据总线完成.该层利用了感知控制机制,在内部设有感知控制单元和执行单元,可以感知用户界面产生的用户任务期望和功能核心系统状态,根据领域转换规则进行计算,以匹配用户任务描述和功能核心状态之间的非线性关系,然后将结果数据更新到相应的数据总线.该层内部存储了用户期望和系统状态的历史数据,有助于实现系统的可用性^[16].

- 功能核心层.该层按照传统软件工程思想设计,在低级功能层次上提供服务,旨在实现功能性需求.该层与代理层的通信通过向应用领域数据总线传递数据进行.对于已有的系统,可以复用此层已有的设计,而不必重新构建此层,只需增加其他层,即可构建高可用性的软件系统.

- 用户任务数据总线 and 应用领域数据总线.这两个数据总线是一个共享数据区.此共享区是一种公共的可扩展的数据结构,这里的数据结构通常是带有通知机制的“活动值(active value)”^[19],即数据结构将数据的变化实时地通知到已经注册到总线通知队列中的组件,这些组件通常是对此总线中某些数据敏感的组件.

- 用户任务数据总线.该总线存储了用于描述用户任务目标的数据.在运行时,用户界面只与此总线交互,使用户界面和功能核心的耦合性降到最低.内部的数据包括:(1) 期望值:此单元存储了高级功能层次的任务目标,即用户期望系统达到的某种状态,此目标与用户任务直接相关.(2) 更新值:系统为了实现用户目标(即期望值)而达到的当前实际状态,此状态也是高级功能层次上的系统状态描述,与用户目标直接相关.

- 应用领域数据总线.该总线存储了应用领域的状态,将功能核心层与其他层彻底隔离.内部的数据包

括:(1) 领域输入:系统的实际状态,是低级功能层次的状态描述.(2) 领域输出:感知控制代理期望系统达到的低级功能层次的状态.

这种体系结构模型有以下优点:(1) 用户界面设计专家可以专注于分析、设计以用户任务目标为导向的界面,而不必考虑工程领域的具体实现;工程领域的专家可以专注于应用领域的实现,而不必受限于用户的任务;最后通过代理,将用户界面和业务领域进行匹配.(2) 通过代理可以实现用户任务和应用领域功能之间的不同粒度、不同功能层次的转换.(3) 对一个已有的系统来说,不需要完全抛弃已有的应用领域的设计,只需在现有系统的基础上增加代理层、新的用户界面层,即可给用户新的面向用户任务的交互界面,从而复用原有系统中的应用领域模块.(4) 用户界面模块和应用领域模块之间不直接通信,它们都通过总线发送请求或提供服务,这种隐式事件调用风格的机制容易实现两个模块之间的强内聚、松耦合.(5) 有助于动态增加代理单元,实现新的可用性,而不必修改系统原有的结构.

4 感知控制代理层详细设计

感知控制代理模块由执行单元、反馈单元、感知控制单元、期望值历史单元组成(如图 4 所示).

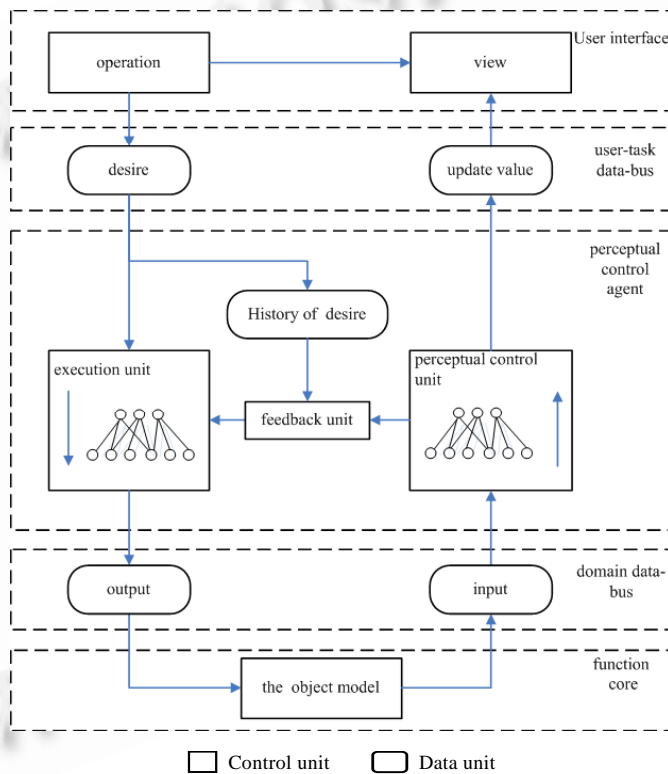


Fig.4 Perceptual-Control-Based agent architecture

图 4 基于感知控制代理体系结构

感知控制代理模块负责将用户任务数据总线中的用户任务状态(即期望值)转换为功能核心可接受的应用领域数据,然后更新到应用领域数据总线上,同时将功能核心的系统状态聚合为用户任务状态,完成从用户任务描述到功能核心状态的非线性转换.代理内部有期望值历史单元,负责存储用户任务(即期望值)的历史状态数据,此数据可以用来与当前最新用户任务完成情况作比较,以决定执行单元是否继续迭代计算新的功能核心的值,从而改变功能核心的状态,最终使系统的状态达到用户期望的任务目标.当用户的期望不能得到一组确定的系统状态数据时,执行单元将先根据经验数据进行计算求解,求解的结果如果能满足用户期望则算法停止;如果

不能达到用户的期望,本次求解的结果将作为下一次迭代的一个约束条件,并重新启动算法,直到用户任务状态(即期望值)被满足或基本接近为止.此求解过程用负反馈原理实现非线性转换.

- 执行单元.如果期望值改变,或由于应用领域输入的改变导致系统当前的状态不能满足期望值,则此单元的处理逻辑将被启动.如果是第 1 种情况,则此单元将根据用户期望启动一个新的变换处理逻辑,用于将期望值分解、变换到功能核心状态,但当系统状态参数多于用户任务描述参数时,仅根据期望值不能得到确定的功能核心状态,则执行单元通常是根据经验值给定一组值或任意给定一组值.对于第 2 种情况,即当感知控制代理单元感知到系统状态不能满足用户的期望时,则进行正向的聚合求解,然后将结果更新到用户任务总线上,同时感知控制代理单元应当通知反馈单元,以便启动反馈处理逻辑.反馈处理逻辑的工作原理是比较更新值和期望值历史,如果两者有偏差,则通知执行单元进行下一次迭代,迭代的前提是将本次比较的结果作为限制条件添加到原有关系表达式队列中.

- 感知控制单元.此单元主动感知应用领域数据总线中当前系统的状态,通过正向函数求解,由系统状态得出当前任务状态(通常此状态是一组确定的值),并将此任务状态更新到用户任务数据总线上,同时将当前任务状态传递到反馈控制单元,以便进行反馈处理.

- 反馈单元.此单元是进行反馈处理的关键单元.此单元接收到感知控制单元传递的当前任务状态后,与本单元中保存的最新的期望值进行比较,并将比较结果生成关系表达式,然后将此关系表达式传递给执行单元,以便执行单元进行下一次迭代.通过此种负反馈控制机制加速系统达到用户期望状态的过程,同时也可以维持系统状态,使外界对系统的影响降到最低.

- 期望值历史单元.此单元按照用户的操作顺序存储了最近几次用户的期望状态,每个状态都有时间戳,记录状态设置的时间.反馈控制单元用期望值历史单元的数据与当前系统状态进行比较,比较的结果作为执行单元处理逻辑的运算依据.期望值历史单元存储的历史状态有助于实现可用性(如 Undo/Redo).

5 案例研究

Len Bass和Bonnie E. John总结了 27 个与软件体系结构密切相关的可用性场景^[17].其中,“聚合数据(aggregating data)”可用性场景描述为:系统允许用户在一个或多个动作中对任意的数据组合进行操作和控制^[17];“支持撤销(supporting undo)”可用性场景描述为:系统允许用户返回本次操作之前的系统状态,并允许用户撤销之前的操作^[17].

在实现了“聚合数据”可用性场景的控制系统中,一个操作可以控制多个变量,用户仅需在任务层次上进行控制,而不需要计算任务目标和领域变量之间的对应关系,系统自动匹配任务目标和领域变量之间的关系.而在传统的控制系统中,一个操作只能控制一个领域变量,用户要达到任务目标,需要自己计算任务目标和领域变量之间的对应关系,并对多个领域变量进行控制.

例如,在应用领域只提供了低温水流和高温水流的情况下,用户要得到一定温度的混合水流.传统的控制界面直接给用户提供单个低温水流和单个高温水流的流量控制操作(如图 5 中的“下”所示),用户只能先根据目标水流的水量和温度来计算两种水流的各自流量,再分别去控制两种水流流量的大小.

采用基于感知控制代理体系结构,我们已经实现了“聚合数据”可用性场景.在系统中给用户提供的任务层次的控制操作,即用户可以直接控制结果水流的大小和温度(如图 5 中的“上”所示).当用户控制水流大小时,系统根据用户的输入自动调节低温水流的大小和高温水流的大小;温度控制亦然.系统内部的代理单元可以实现界面输入和功能核心之间的非线性转换,系统可以动态添加代理单元来实现不断变化的可用性.

系统分为 3 层(如图 6 所示):a) 用户界面层:是用 JOOIGT(OOIGT^[18])的 Java 版本实现的,允许用户直接控制窗口中图形界面元素来调节流量和水温.b) 功能核心层:控制两个变量,表示两个固定水温的水流大小(一个是高温,一个是低温,且温度已知).c) 感知控制代理层:由多个并列的感知控制代理组成,代理用来匹配界面层和功能核心层的非线性关系(实现“聚合数据”可用性^[17]).功能代理中的执行单元用来将用户界面的任务层次期望(流量、温度)转换为功能核心低级物理层次的控制变量(高温水流、低温水流),反之亦然,功能代理实现了系统

的基本功能需求;Undo/Redo代理存储了用户界面任务(流量、温度)的历史状态,系统可以恢复到指定的某个历史状态,实现撤销功能(实现“支持撤销”可用性^[17]),Undo/Redo代理仅通过与用户任务数据总线与应用领域数据总线的通信来完成撤销和重做,并不与其他代理单元通信,以此来保证各单元的独立性。

如果系统需要实现其他新的可用性需求,可以增加新的代理,而不必修改已有的结构和构件.用户界面与控制代理层的通信通过向用户任务数据总线更新数据来完成,而功能核心层和感知控制代理的交互通过向应用领域数据总线传递数据来实现。

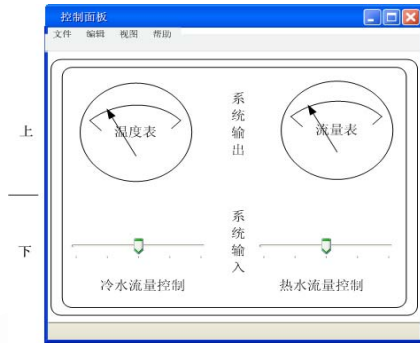


Fig.5 Comparison of two flow-controls
图 5 两种水流控制界面对比

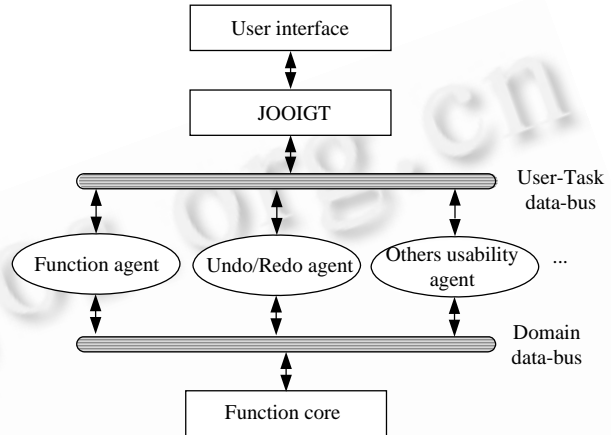


Fig.6 Architecture of flow control system
图 6 水流控制系统结构

6 结束语

本文提出了一个基于感知控制代理的体系结构,通过增加感知控制代理层,可以匹配用户界面和功能核心之间的非线性关系.这样的体系结构不仅可以帮助建立高可用性的系统而不改变系统原有的功能性,而且还支持动态增加新的可用性代理,以适应可用性需求的不断变化.通过用户任务数据总线,隔离用户界面变化对系统结构以及其他组件的影响;通过领域数据总线,隔离了可用性变化对功能核心的影响,使构建的系统更灵活、更稳定。

我们当前的工作是应用基于感知控制代理的体系结构来实现其他更多的可用性场景,同时实现基于感知控制代理的体系结构的通用框架,以及提供界面层和感知控制代理层的开发工具箱,以支持用户迅速构建符合此体系结构的软件系统。

References:

- [1] Gentner D, Nielson J. The Anti-Mac interface. *Communications of the ACM*, 1996,39(8):70-82.
- [2] Folmer E, Bosch J. Architecting for usability: A survey. *The Journal of Systems and Software*, 2004,70(1-2):61-78.
- [3] Bass L, Clements P, Kazman R. *Software Architecture in Practice*. Boston: Addison-Wesley, 1998.
- [4] Dix A, Finlay J, Abowd G, Beale R. *Human-Computer Interaction*. 2nd ed., Pearson Education Limited, 1998.
- [5] Green M. A survey of three dialogue models. *ACM Trans. on Graphics*, 1986,5(3):244-275.
- [6] A metamodel for the runtime architecture of an interactive system: The UIMS tool developers workshop. *SIGCHI Bulletin*, 1992, 24(1):32-37.
- [7] 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.
- [8] Hartson R. User-Interface management control and communication. *IEEE Software*, 1989,6(1):62-70.
- [9] Coutaz J, Balbo S. Applications: A dimension space for user interface management systems. In: *Proc. of the ACM CHI'91*. 1991.

[10] Hua QY, Wang H, Muscogiuri C, Niederee C, Hemmje M. From conceptual modeling to architectural modeling—A UCD method for interactive systems. *Journal of Asian Science-Information and Life*, 2004,3(2):187–196.

[11] Bass L, John BE, Kates J. Achieving Usability through Software Architecture. Technical Report, CMU/SEI-2001-TR-005, Carnegie Mellon University, 2001.

[12] Li GJ. Research on SAEU: A usability-oriented architecture model and software development environment [MS. Thesis]. Xi'an: Northwest University, 2008 (in Chinese with English abstract).

[13] Farrell PSE, Hollands JG. Perceptual control and layered protocols in interface design: I. Fundamental concepts. *Int'l Journal of Human-Computer Studies*, 1999,50(6):489–520.

[14] Marken RS. PERCPLATe: Perceptual control analysis of tasks. *Int'l Journal of Human-Computer Studies*, 1999,50(6):481–487.

[15] Gentner DR, Grudin J. Design models for computer-human interfaces. *IEEE Computer*, 1996,29(6):28–35.

[16] Folmer E, Bosch J. Case studies on analyzing software architectures for usability. In: Proc. of the 2005 the 31st EUROMICRO Conf. on Software Engineering and Advanced Applications. 2005. 206–213.

[17] Bass L, John BE. Linking usability to software architecture patterns through general scenarios. *The Journal of System and Software*, 2003,66(3):187–197.

[18] Hua QY, Ge W, Yu B, Hao KG. OOIGT: An object-oriented interactive graphics toolkit. *Chinese Journal of Computers*, 1998, 21(2):154–161 (in Chinese with English abstract).

[19] Hill RD. The abstraction-link-view paradigm: Using constraints to connect user interfaces to applications. In: Proc. of the CHI'92. New York, 1992. 335–342.

附中文参考文献:

[12] 李光俊.面向可用性的软件体系结构模型 SAEU 开发环境研究与实现[硕士学位论文].西安:西北大学,2008.

[18] 华庆一,葛伟,鱼滨,郝克刚.一个面向对象的交互式图形工具箱 OOIGT.计算机学报,1998,21(2):154–161.



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



蔚娣(1984—),女,硕士生,主要研究领域为人机交互,软件工程.



刘庆芳(1981—),男,硕士生,主要研究领域为人机交互,软件工程.



王小文(1984—),男,硕士生,主要研究领域为人机交互,软件工程.