

91-93

面向对象的图形用户接口设计

房亚明 董继润^Y 李景州 TP311

(山东大学计算机系 济南250100)

摘要 From a standpoint different with a traditional user interface(UI) method, this article describes a method how to fuse Object-Oriented idea into UI design, and uses the analysis results of Object-Oriented Analysis guiding design. Our approach integrates methods and specification from software engineering with graphical user interface design, and gives a fresh idea in designing UI.

关键词 Object-oriented analysis, View, Dialogue net, Object model.

软件工程的发展与用户接口技术有着密切的联系,但用户接口(UI)的考虑侧重于用户的观点,软件工程(SE)的考虑则侧重于系统的观点。围绕用户接口的设计,Anthony I. Wasserman 提出了用户软件工程(USE),但很少把 UI 与 SE 方法联系起来以指导 UI 的设计。结构化分析作为需求分析的一种确定的方法,也没有指出在 UI 的设计中如何使用这些基本的模型。同样,面向对象的分析方法,虽然许诺支持由分析到设计的平稳过渡,但在这方面也未给予支持。另外,原型法通过对原型的构造和评价能提高用户和开发者之间的通讯能力,但用原型作为对应用规格说明和 UI 的早期评价亦有其缺点,原型代替不了详细的文档。因此,在与马来西亚金狮集团合作开发的面向数据库应用的 OO-CASE 中,对用户接口的设计,我们作了许多考虑:(1)用 OO 分析指导设计;(2)分析模型与 UI 设计模型一致,增加可重用性,从而使得信息系统中的图形用户结构是基于应用的对象模型上来开发的。(作为一个交互的信息系统至少包括:人/机对话,数据库,交易(事务处理)集合)。我们所讨论的面向对象对话结构,总的说就是:对话提供给用户一种方法,按照用户所关心的对象去导航、操作。

和面向对象的 UI 不同的是,功能结构的用户接口,是一组功能,一般是通过菜单和表格来访问,而面向对象的结构,遵循对 UI 设计的范型进行直接操纵。(就象可视化的 DBMS 中,用户直接操纵 E-R 图,形成查询。)由于 UI 的结构不是直接建立在任务的功能层次上,当任务有所改变时,最终的用户接口有较大的灵活性,只要对象的前后联系,上下环境未改变,很容易对当前任务作出修改。为了对所描述的具有灵活性的对话建模,需要一种适于进行 GUI

对话描述的建模技术,我们采用的是对话网结构(Dialogue Net),而不是转换图。

一、设计方法概览

我们所考虑的方法遵循把软件工程的建模技术同原型的反复评估结合起来。在系统分析的过程中,建立起确定的任务模型(task model)及初步的对象模型(object model)。任务模型可用数据流图来表示。图1表示了该方法的概览,对第二步应用规范说明是细化对象模型(描述属性、方法),除此之外,确

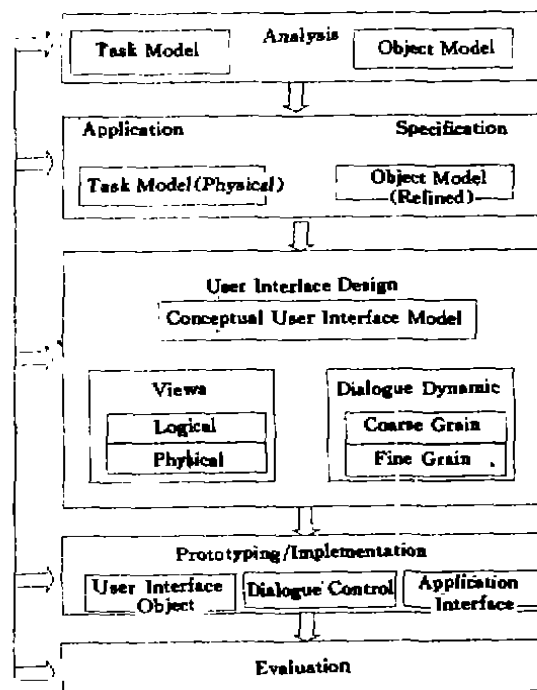


图1 方法概览

定每个类的实例数目,相关实例的实例关系的数目。为了找出对象的方法,系统功能要在用户和计算机之间进行划分,细化任务模型。用户接口设计主要基于对象模型,并且要考虑到任务需求说明及用户的特征。用户接口的概念模型则建立在应用规范的基础上。

为了表示对象,定义了用户视图。视图可从两个方面去理解,逻辑视图定义属于视图的对象模型的一部分,物理视图定义屏幕外观、GUI交互对象间的关联。在对话规范中我们定义了两层:粗粒度对话(coarse grain dialogue)定义视图序列及响应用户输入的对象方法的调用;细粒度对话定义交互对象的动态行为。

在上述模型的基础上,建立原型,由用户和开发者反复评估。

二、视图构造和对话设计

概念对象的表示及与其交互需要真正的屏幕表示,我们称为视图(views),逻辑视图是概念模型的一部分,一个视图一般包括一个用户接口概念对象的属性、方法。有的方法、属性也可能从其它对象引入,视图的物理表示,被选择的用户接口对象平铺放在一个窗口或子窗口中,交互对象的选择是根据属性类型及值域方法通过菜单项、按钮、直接操纵(双击、拖拽)等调用。

我们区分几种类型的视图,其中一些是特定于某种对象类型的。浓缩视图(condensed view)是对象的抽象表示(象图标),该视图适用于任何对象类型,容器对象的内容通常表示为一组图标或是一个列表(表视图)。其它一些视图类型(如 tree)要依赖于它的数据结构,数据视图是显示数据对象的属性方法。工具对象的属性方法参数,称作参数视图。

初始的视图和对话设计遵循用户接口概念模型,为每个对象引入视图,包括该对象的所有属性、方法以及访问相关对象的工具,通过引入其它对象的属性、方法,删掉不需要的属性、方法,把一个视图分成子视图(如果窗口显得拥挤)进行完善,细化设计。如果一个实例有较少的实例联系,可用列表选择对象。

初始的应用窗口是一组以图标表示的容器对象平铺显示在窗口中,非数据对象的一些方法在菜单项中,视图为根据用户任务构造对象模型提供了灵活性,从对象模型直接得到视图只是问题解决的第一步。一般,用户完成任务需要比对象属性更多的信

息。同样,对话设计中明确的模型,也为独立于对象模型构造对话提供了一种方法。

三、对话范型说明

对视图序列(即粗粒度对话)的描述需要图形技术提供一个可视化的概貌。目前用户接口管理系统中采用的文本对话规范描述无法表达粗粒度的对话(由对象模型得到的视图序列)。目前,描述这种对话的形式化方法有两种。一是状态转换图,一直被用于表格和菜单接口的设计,但这种方法不适用于具有许多并行窗口的图形用户接口。尽管有许多扩展的状态转换图希望能处理并行性,但都没有使用户能形象地看到并行视图之间的控制流。二是我们采用的一种类似于 Petri nets 的方法规范描述对话——对话网(图2),它提供了一种比状态转移更自然的方法处理并行性,并且具有分层构造,对话模式(宏)即为动态创建视图的对话描述。

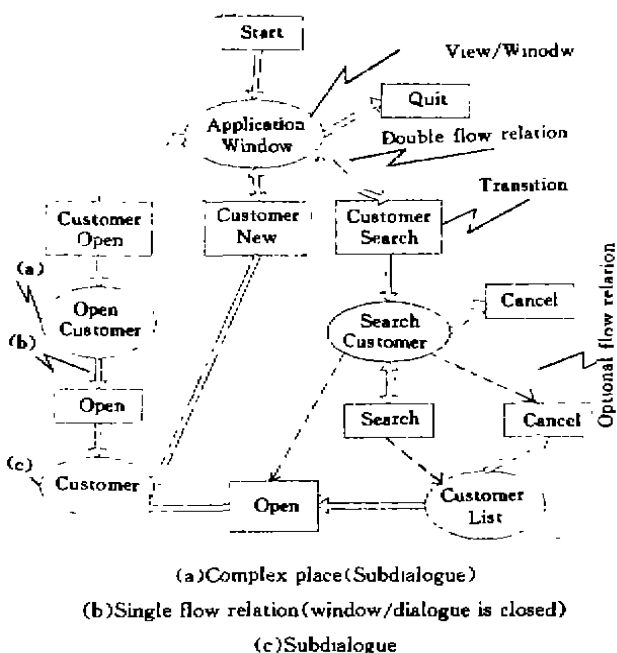


图2 给定了顾客选择对话的对话网

视图的动态可见度,由模型中的位置标记来确定。也就是说一个视图被映射到屏幕上当且仅当对话网中相应的位置被标定。转换(图中方框所示)模拟了对话的步骤。图2中,一个特别的 start 转换在对话开始时被激发。一个转换被激发,如果输入位置被标记(即 open),而输出位置未标记(即 close)。转换激发后,输出位置变为标记,输入位置变为未标记。

转换两侧的标记(input,output)是激发的先决条件。图2中,start 转换,在对话开始打开一应用窗口,只有双向关系才能脱离该窗口控制,所有整个对话期间,该窗口保持打开状态,通过转换 Customer Open, Customer New, Customer Search, 各自的视图及子对话被打开, Open Customer 视图让用户通过顾客号码打开一个顾客。

Search Customer 视图通过属性查找给出满足一定条件的顾客名单。图中的选择流关系不影响激发条件,如图中的较低位置的 cancel 转换可激发而不管 Search Customer 视图是否标记(open),即若 Search Customer 对话框以前未关闭,那么 cancel 转换关闭顾客名单(Customer List),然后关闭 Search Customer 对话框。

象图中所示,对话网是分层构造的,当图中 Complex place 的子对话被标记时,一个子对话网开始。进一步细化规范,要对转换指明事件和条件,这被看作激发的附加条件,指明转换被激发后执行的动作。图3指出了细化后的 Customer New 转换的规范。

```
event: Filemenu, Newselect/视图描述静态
condition: Currentselection=Customer
action: Number:= / 对话描述动态
GenerateCustomerNo()
```

图3

对话网主要用来描述粗粒度对话,而细粒度对

话仍是通过文本规范,易于描述和理解。

三、工具支持

为了在实际应用中引入一种方法的具体建模技术,工具的支持是必不可少的。

·用户接口的概念模型的产生:从分析任务的交互过程中确定主要对象,然后确定各自的容器对象和应用对象及其访问关系。对数据对象,从应用说明中得出属性和方法。当然模型的初步建立需要通过指明数据对象间的访问关系,及为组织对话引入新对象以进一步细化。

·对话模型的产生:根据访问关系为每个概念对象、对话转换假定一个视图。通过增加对话框来进行细化。

·从逻辑视图产生物理视图:逻辑视图在对象模型的基础上确定。物理视图由逻辑视图而来,是基于属性类型和值域通过规则(交互对象的选择和布局)来确定。

·可执行的对话的产生:从形式化说明的转换,可产生可执行的对话。

结论 为了精简窗口及导航的数目,设计应通过对对象层次的树形表示来细化。比较菜单表格的 UI 设计,建模技术对表示 UI 设计是有益的。我们采用的 OO-GUI 设计,其复杂性有所增加,但对话网能提供一种精确的规范说明。虽然完整的工具支持我们尚在开发阶段,但我们对采用面向对象的分析,并用分析结果指导 GUI 设计抱有极大信心。

(上接第58页)

信号的处理方法,来设计微内核结构下的信号机制,则发现有些信号不能被检测到。如当一个进程不调用系统调用时,发送给它的信号将永远检测不到。本文介绍的微内核结构下的信号机制,它的要点就是在信号发送时进行信号处理,系统调用返回前进行信号处理,增加一个定时线程,定时处理系统中有待处理的信号的进程,而信号的信号结构可在共享区中通过信号结构数组来实现。上述设计思想已在国家的“八五”重点攻关课题“系统软件平台”中的操作系统 COSIX V2.0 中实现,并通过了鉴定。当然,该系统在性能优化、可靠性、功能完备等方面还有很多工作要做。

致谢 COSIX V2.0 是中软总公司、国防科工委指挥技术学院、中科院软件所、海军计算技术研究所等单位共同研制完成的。作者对以孟庆余教授、周明

德教授为首的课题组的同志们所给予的帮助和合作表示感谢,并特别感谢殷树勋高级工程师和韩乃平同志,方案的设计和实现与他们的许多建设性意见和帮助是分不开的。

参考文献

- [1] Joseph Boykin, Programming under Mach, U. S. A.: Addison-Wesley Publish Company, 1993
- [2] Mike Accetta, Mach: A New Kernel Foundation for UNIX Development, U. S. A.: In Proc. of Summer Usenix, 1986
- [3] Golub, Unix as an Application Program, U. S. A.: In Proc. of the Summer Usenix, 1990
- [4] 周明嵩等, 计算机环境的可移植操作系统界面 POSIX.1, 电子工业出版社, 1990
- [5] 杨学良等, UNIX SYSTEM V 内核剖析, 电子工业出版社, 1990