交互式用户界面的形式化描述与性质验证*

朱 军[1] 张 高[1] 华庆一[2] 戴国忠[1]

[1](中国科学院软件研究所计算机科学开放研究实验室 北京 100080)

[2](西北大学计算机科学系 西安 710069)

摘要 随着人机交互技术的发展,计算机和用户之间的接口越来越自然,但用户界面管理系统内部的复杂度却大大地增加了.目前提出的新一代用户界面的模型大都停留在概念模型阶段,缺乏对模型的严格描述和证明.该文结合对基于自然交互方式的用户界面的研究成果,归纳出了一个交互式用户界面的通用模型.为了保证系统设计的正确性,文章讨论了如何使用形式化描述语言LOTOS(language of temporal ordering specification)和基于动作的时序逻辑ACTL(action based temporal logical)对系统进行描述与验证,这有利于人们对交互式用户界面的动态行为进行研究、评估与定义.

关键词 交互式用户界面,形式化方法,模型检查,时序逻辑,LOTOS,ACTL.

中图法分类号 TP311

建立以人为主、基于自然交互方式的用户界面是下一代用户界面的发展趋势.多媒体用户界面和多通道用户界面由于比较好地提高了用户和计算机之间的通信频带,从而成为下一代用户界面的主要模式.随着用户界面技术的发展,计算机和用户之间的接口越来越自然,但用户界面管理系统内部的复杂度却大大地增加了.为了适应用户界面技术的发展,国内外的专家提出了许多下一代的用户界面模型和原型系统[1,2],但这些模型和原型系统大都还停留在概念设计阶段,模型缺乏严格的描述和关键性质的定义与证明.由于模型对用户界面的具体实现起着重要的指导作用,所以保证其正确性和合理性就显得尤为重要.

本文结合国内外提出的下一代用户界面模型的特点,提出了一个交互式用户界面的通用模型,并利用形式化方法对其进行定义与研究.形式化方法能对系统进行精确的并且无二义性的描述,并能在此基础上进行推理,从而可以对系统应具有的关键性质进行验证,以保证它的正确性.近年来,形式化的方法在用户界面领域中的应用越来越受到重视,为了保证系统开发的正确性,本文采用形式化的方法对模型进行了准确的描述和性质定义,从而为系统验证提供了有效的手段.

目前,在用户界面领域中采用的形式化技术主要有用户动作标记UAN(user action notation)、Z语言(Z notation)、时序描述语言LOTOS(language of temporal ordering specification)以及Petri网等.UAN适合从用户的角度出发对界面所应完成的任务进行有效的描述[3],基于类型集合理论和一阶谓词逻辑的Z在刻画交互对象自身的特征与性质方面具有明显的优势[4],而基于进程代数的LOTOS更适合描述各个界面对象间的交互行为[5].此外,Petri网也被用来描述用户界面的交互过程[6].

本文采用形式化描述语言LOTOS描述用户界面系统的通用模型.LOTOS是由国际标准化组织开发的一种形式化规格说明语言[7],它提供了一组符号用于定义系统外部可见行为之间的时序关系.在LOTOS中,并发系统被看做是具有内部动作的一系列进程(process).每个进程可以通过定义在事件端口(gate)的可见动作与其他进程进行交互.通过定义进程的一系列可观察的动作可以描述一个进程所具有的外部行为特征,而整个系统

---

* 本文研究得到国家自然科学基金和国家863高科技项目基金资助.作者朱军,1972年生,博士,主要研究领域为面向对象技术,形式化方法.张高,1971年生,博士,主要研究领域为人机交互,用户界面.华庆一,1956年生,教授,主要研究领域为图形用户界面,软件可视化.戴国忠,1944年生,研究员,博士生导师,主要研究领域为计算机图形学,用户界面.

本文通讯联系人:朱军,北京 100080,中国科学院软件研究所计算机科学开放研究实验室

本文1998-07-10收到原稿,1998-09-02收到修改稿

的行为可由一个动作树来表示,树中的每一个路径代表着进程可能执行的一个可观察动作序列.使用 LOTOS,可以用数据代数(data algebra)表示系统状态,用进程(process)来描述系统行为.而交互式用户界面有两个显著特点——外观(appearance)和行为(behavior).外观是用户所观察到的系统当前状态的表示,行为描述了界面如何对用户或应用产生的事件进行反应,用户可以通过交互行为改变系统内部状态.由此可见,LOTOS 比较适于描述交互式用户界面.

另一方面,我们采用基于动作的时序逻辑 ACTL(action based temporal logical)定义用户界面系统的一般性质.ACTL 是一种基于动作的分支时序逻辑[8],它的解释域为标号转移系统 LTS(labeled transition system).在这类系统中,状态间的转移带有标号,且标号是用来表示引起状态转移的动作的.通常人们认为时序逻辑是适于并发反应式系统的性质定义的,而用户界面系统中状态的转移通常由用户的动作产生,所以我们采用基于动作的时序逻辑,这使得对于系统性质的定义更加直观.一般而言,使用 LOTOS 描述的系统的语义模型,能够由一个标号转移系统表示.这样,利用计算机辅助工具,我们可以对 LOTOS 描述的系统模型进行检查,验证其是否具有 ACTL 定义的性质,从而保证系统在某些方面的正确性与可靠性.

目前,下一代用户界面系统的设计趋势是能够支持多通道并行输入与输出、并行对话、并发控制与多视图数据显示,这就意味着必须以并行方法设计用户界面系统.面向对象方法的应用使这种并行性得以实现.我们提出的交互式用户界面的通用模型就是基于这一特点提出的多代理(multi-agent)交互模型.这是一种典型的反应式系统,它将用户或应用产生的事件传递给各个相关的对象,由这些对象合作完成相应的任务.由此可见,该模型是非常有利于使用 LOTOS 与 ACTL 对其进行描述与验证的.

1 用户界面的抽象模型

为了便于进行形式化描述与验证,首先需要给出一个关于用户界面的系统模型.由于直接操纵式用户界面与多通道用户界面的采用,目前大多数用户界面的设计都采用了面向对象方法.基于此类方法的用户界面体系结构主要有 MVC(model,view,controller),PAC(presentation,abstraction,control)和 ALV(abstraction,link,view)等[9~11].这里,为了有效地进行形式化描述,我们在一个更高的层次上给出一个关于用户界面的通用模型.如图 1 所示,根据对话分离原则,一般的交互式系统分为应用核心(application core)与用户界面系统两部分,而用户界面系统是由多个交互对象 UIO(user interface object)组成的具有一定层次的网状结构.底层的 UIO 负责管理交互设备,接收用户动作,显示用户信息;高层的 UIO 负责管理底层的 UIO,并与应用核心进行信息交换.

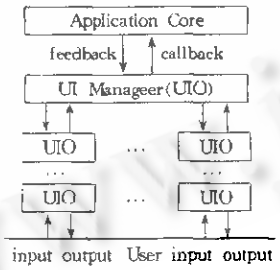

图1 交互式系统用户界面抽象模型

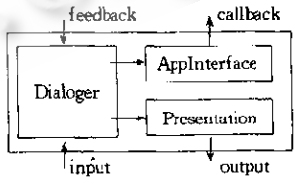

图2 用户界面交互对象UIO概念模型

可以看出,在这种模型下,交互对象 UIO 应当是一种重用性较高,具有相对独立功能的对象构件.它不仅具有自己的内部状态,还具有和其他对象交互的能力.一般而言,交互对象 UIO 具有一定的内部结构,如图 2 所示,它主要由 3 部分组成.

(1) 表示元素 Presentation.一般的 UIO 都具有表示其内部状态的外观,通常由表示元素维护.外观的变化反应了系统内部状态的变化.

(2) 应用接口 AppInterface.为了保证交互式系统中用户界面与应用核心分离后两者间的有效联系,UIO 中应含有能产生回调的应用接口.

(3) 对话元素 Dialoger.此部分主要用于接收来自外部对象的消息,并产生相应的动作.这些消息可能由用

户动作产生,也可能由应用核心或其他交互对象 UIO 发出. Dialoger 对这些消息进行处理,根据情况驱动表示元素改变外观,也可回调应用核心与其他 UIO 进一步对其加以处理.

UIO 可以通过 4 种端口(gate)与外界交互. 一方面,它从 input 端口接受来自用户的输入,从 output 端口输出内部表示以改变界面外观;另一方面,它可以从 feedback 端口接收来自应用核心的信息,也可从 callback 端口请求应用核心进行计算处理. 采用 LOTOS,我们可以对 UIO 的交互行为严格地进行描述,如下所示.

```
process UIO [input,output,feedback,callback]:(input_data:InputData,feed_data:FeedData):noexit:=
(
input? id:InputData;callback! AppInterface(id);
UIO[input,output,feedback,callback](id,feed_data)
[]feedback? fd:FeedData;output! Presentation(fd);
UIO[input,output,feedback,callback](input_data,add(fd,feed_data))
)
endproc
```

UIO 在接收到 input 事件后,将产生 callback 事件,并依此递归下去. 同样地,它在接收到 feedback 事件后,将产生 output 事件,同时将 fd 加入反馈信息集合,并依此递归下去. 这里使用数据代数方法定义进程参数相关的数据类型,与 callback 事件相关的数据值由 AppInterface(id)产生,而与 output 事件相关的数据值由 Presentation(fd)产生.

2　用户界面性质的定义

从用户界面的系统模型可以看出,整个用户界面可以视为由多个交互对象 UIO 组成的带有层次结构的系统.这个系统介于整个交互式系统的应用核心与用户之间,为两者的交互提供友好的方式. 考察用户界面系统中 UIO 的交互性质是为了检查用户界面的完整性以及刻画它对用户及应用核心产生的事件反应时的行为特征. 我们特别关注的性质有两个方面,下面给予详细介绍.

(1) 基本图形交互对象 UIO 的一般特点.这有助于发现 UIO 的通用特征,从而可以完善 UIO 模型. 显而易见,对于 UIO 而言,它应该在任何条件下都能够对来自用户或应用的输入事件作出相应的反应,进行内部处理并产生相关的输出事件. 这一性质可由下述两条 ACTL 公式定义.

$$AG([input]E[true\{true\}U\{callback\}true]).$$

$$AG([feedback]E[true\{true\}U\{output\}true]).$$

上述公式的含义是,在任何路径(A 算子)和状态中(G 算子),若 UIO 接收到 input 或 feedback 事件,则其内部进程间总存在(E 算子)一条路径,使 UIO 产生 callback 或 output 事件.

(2) 用户界面系统的一般性质.这是为了检查用户界面体系结构在设计上的逻辑完整性. 在应用系统中,若用户界面涉及到的 UIO 数目较多,且 UIO 之间的协作关系较强,则系统的复杂度将显著增加. 为了保证系统的可靠与稳定以及设计的正确性,需要检查所设计的系统是否满足用户界面的一般性质以及是否能够达到预期的效果. 我们主要从系统和用户两个角度来考虑:一方面,作为交互式系统,其真正的功能处理部分在应用核心中,用户界面只是按一定的方式去理解用户动作的意图,因此,用户产生的有效命令必须能正确到达应用核心;另一方面,作为与用户交互的界面系统,它必须将应用核心对命令的处理结果或其他一些信息以一定的方式显示出来,使用户对系统内部状态可见. 下面,我们给出这两个性质的定义.

· 可达性

可达性是指,当某一个用户动作发生时,系统能否达到指定效果. 这一性质允许我们验证用户的交互动作能否引起应用核心的相应功能处理部分对其产生反应. 其定义如下:

$$AG([user\_action]E[true\{true\}U\{Application\_function\}true]).$$

这一 ACTL 公式的含义为:对于所有可能的将来(A 算子)以及所有可能的状态(G 算子),如果用户动作 user_action 发生,那么系统在以后的时序演化过程(U 算子)中至少存在(E 算子)一条路径,即一组不加任何限

制(true{true}算子)的转移,使得最终导致动作 Application_function 发生.

• 可视性

可视性是指,用户动作能够引起系统用户界面外观的变化.从用户的角度看系统交互特性,是为了检查界面能否对用户动作产生的事件作出反应,并引起界面外观的改变.利用这一性质,我们可以验证某一用户动作能否在用户界面上产生反馈信息.其定义如下:

$$AG([user\_action]E[true\{true\}U\{User\_interface\_appearance\}true]).$$

对于一给定用户动作 user_action,我们可以使用上述 ACTL 公式确认,在以后的时序演化过程(U 算子)中是否至少存在一条(E 算子)路径,使得系统产生一个事件 User_interface_appearrance 改变相应的用户界面外观,从而表明用户的输入已被系统接收.

3 模型检查方法与实例

3.1 模型检查

模型检查是检验以有限状态机为基本模型的系统的一种常用方法.对于并发反应式系统,人们常以标号转移系统作为其抽象模型,并使用与之相应的时序逻辑公式定义其性质,从而进行验证工作.对于用户界面系统,通常所要验证的是当系统从某一状态转移到另一状态时,是否满足转移路径上定义的动作公式以及所到达的状态是否满足相应的状态公式.

如图 3 所示,为了对用户界面系统进行有效的验证,一方面,我们使用 LOTOS 对用户界面进行严格的形式化描述,并将其转化为一有限状态机表示的标号转移系统 LTS.其中,LOTOS 中每一个进程对应于界面模型中的一个 UIO 对象,而定义在其端口上的动作对应于一个转移事件.另一方面,通过提炼出所要检验的用户界面系统性质,并使用 ACTL 对其进行严格的逻辑定义.这样,利用基于时序逻辑的模型检查工具,我们就可以对用户界面系统模型进行严格而有效的验证.

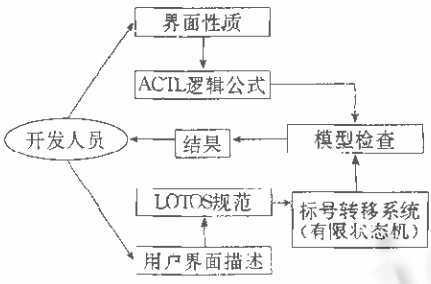

图3 用户界面模型检查示意图

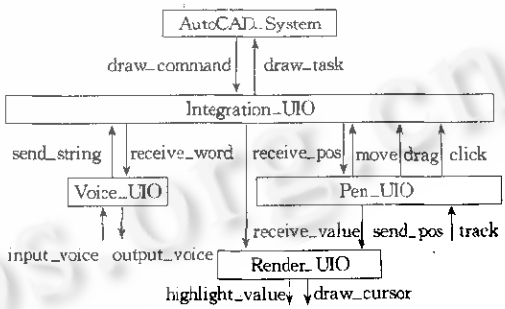

图4 一个多通道CAD系统的UIO模型

3.2 一个多通道用户界面系统的验证

现在,我们以一个 CAD 系统的用户界面为例[12],介绍如何使用形式化方法对其进行描述与验证.此用户界面系统支持多通道输入方式,它通过语音输入作图命令,并通过光笔确定图形坐标,经过任务整合后,由系统根据当前状态完成作图任务.为了在系统开发的早期过程中判断系统是否设计得完善以及是否存在问题,我们对系统所应具有的性质进行了定义与验证,这些性质是此用户界面系统达到其功能要求的必要条件.具体步骤如下:

步骤 1. 建立 CAD 用户界面系统的 UIO 模型.

首先,我们根据系统的需求使用面向对象方法分析系统,并将其分解为由多个交互对象 UIO 组成的对象系统.如图 4 所示,支持多通道输入方式的 CAD 用户界面系统主要由 4 类 UIO 组成,它们是通道整合对象 Integration_UIO;语音处理对象 Voice_UIO;光笔处理对象 Pen_UIO 和涂刷显示对象 Render_UIO.它们之间及其与 CAD 系统核心的交互关系如图 4 所示.

步骤 2. 使用 LOTOS 对用户界面模型进行形式化描述.

在建立了系统的UIO模型之后,我们进一步需要使用LOTOS对每一个UIO对象进行严格的形式化描述,通常这些UIO对应于LOTOS中的进程,其交互动作可表示为事件端口.举例如下,语音处理对象的LOTOS描述为:

```
process Voice_UIO[input_voice,output_voice,send_string,receive_word]:
                    (input_voice:Voice,receive_word:String):noexit:=
(input_voice? id:Voice;send_string! AppInterface(id);
    UIO[input_voice,output_voice,send_string,receive_word](id,receive_word)
[]receive_word? fd:String;output_voice! Presentation(fd);
    UIO[input_voice,output_voice,send_string,receive_word](input_data,add(fd,receive_word))
)
endproc
```

步骤3. 使用ACTL定义用户界面的交互性质.

从数学角度而言,系统的性质可以由一些数学公式所定义,即这些公式定义了系统成立所需满足的必要条件.对于系统性质定义得越多,则表示对系统的约束越严格.所以,如何准确地提炼并定义系统的性质是对系统进行有效验证的先决条件.对于此CAD系统,我们可使用如下ACTL公式检验系统的可视性.

- 光笔的动作将在显示器上画出光标位置

$$AG([track]E[true\{true\}U\{draw\_cursor\}true]).$$

- 语音与光笔同时输入可产生命令,使界面元素刷新

$$AG([track\ \&\ input\_voice]E[true\{true\}U\{highlight\_value\}true]).$$

步骤4. 采用基于LOTOS的时序逻辑验证工具对系统模型进行验证.

这里,我们选用基于LOTOS的集成工具环境MiniLite[13]对上述CAD用户界面系统的UIO模型进行性质检验.这一工具的工作原理如图3所示,它将所输入的表示系统模型的LOTOS描述自动转化为与之对应的标号转移系统,然后在其上进行推理,检验所输入的ACTL时序逻辑公式是否成立.当公式为真时,表明模型满足所检验的系统性质.

步骤5. 根据错误情况修改系统的UIO模型,重复上述步骤,直至它满足所有性质.

4 结　论

本文提出了一个交互式用户界面的通用模型,同时,利用形式化描述语言LOTOS对该模型进行了描述,并使用基于动作的时序逻辑ACTL定义了它的一些基本特性.基于这一方法,我们对一个支持多通道输入方式的CAD用户界面子系统进行了系统模型验证,也对其中采用的用户界面对象进行了基本性质验证.实践表明这一方法是可行且有效的,它能够在系统设计的早期对系统行为进行模拟并发现错误,这有助于缩短开发周期,保证开发质量.文中提出的验证方法为进一步证明其他性质提供了指导.我们今后将对用户界面系统中其他更为复杂的性质,如通道整合的正确性、错误操作的可恢复性等作进一步的定义和验证.

参考文献

1 李茂贞,戴国忠,董士海.多通道界面模型与关键技术.计算机科学,1998,24(1):1~4
(Li Mao-zhen, Dai Guo-zhong, Dong Shi-hai. Multimodal interface model and key technology. Computer Science, 1998,24(1):1~4)

2 李茂贞,戴国忠,董士海.多通道界面软件结构模型及整合算法.计算机学报,1998,21(2):111~118
(Li Mao-zhen, Dai Guo-zhong, Dong Shi-hai. Software model and integration algorithm of multimodal interface. Chinese Journal of Computers, 1998,21(2):111~118)

3 Hartson H R, Gray P D. Temporal aspects of tasks in the user action notation. Human-Computer Interaction, 1992,7(1):1~45

4 Duke D J, Harrison M D. Abstract interaction objects. Computer Graphics Forum, 1993,12(3):25~36

5 Johnson C W. Using temporal logic to prototype interactive system. In: Diaper D, Gilmore D, Cockton G *et al* eds. Proceedings of the Human-Computer Interaction (INTERACT'90). North Holland: Elsevier Science Publishers, 1990. 1019～1021

6 Palanque P, Bastide R. Petri net based design of user-driven interfaces using the interactive cooperative object formalism. In: Paterno F ed. Proceeding of the 1st Eurographics Workshop on Design, Specification and Verification of Interactive System. Berlin: Springer-Verlag, 1995. 383～401

7 Bolognesi T, Brinskma E. Introduction to the ISO specification language LOTOS. Computer Networks and ISDN Systems, 1987,14(1):25～59

8 DeNicola R, Fantechi A, Gnesi S *et al*. An action based framework for verifying logical and behavioural properties of concurrent systems. Computer Networks and ISDN Systems, 1993,25(7):761～778

9 Krasner G E, Pope S T. A cookbook for using the model-view-controller interface paradigm. Journal of Object-oriented Programming, 1988,3(1):26～49

10 Hill R. The abstraction-link-view paradigm: using constraints to connect user interfaces to applications. In: Proceedings of the Human Factors in Computing System (CHI'92). New York: ACM Press, 1992. 335～342

11 Coutaz J. PAC, an object oriented model for dialog design. In: Bullinger, Shackel eds. Proceeding of the Human-Computer Interaction (INTERACT'87). North Holland: Elsevier Science Publishers, 1987. 431～436

12 Li Mao-zhen, Zhang Gao, Dai Guo-zhong. A primitive-based architecture of multimodal interface (PBM-MMI). In: Proceeding of the 1997 IEEE International Conference on Intelligent Processing Systems. Beijing: International Academic Publishers, 1997. 858～862

13 Bolognesi T, Lagemaat J, Vissers C. LOTOSphere Software Development with LOTOS. Boston, London: Kluwer Academic Publishers, 1995. 1～43

The Formal Specification and Property Verification of Interactive User Interface

ZHU Jun[1] ZHANG Gao[1] HUA Qing-yi[2] DAI Guo-zhong[1]

[1](*Laboratory of Computer Science Institute of Software The Chinese Academy of Sciences Beijing* 100080)

[2](*Department of Computer Science Northwest University Xi'an* 710069)

Abstract With the development of human-computer interactive technology, the interface between computers and users is more and more natural, the management of user interface is becoming more and more complex. Now, the available models of next generation user interface are almost conceptual. The formal specification and property verification to them are necessary. Based on the results of the research on natural interactive user interface, the authors give out a general model of interactive user interface. In order to guarantee the correction of the design of system, it needs to specify and verify it rigidly. The formal description and verification by use of LOTOS (language of temporal ordering specification) and ACTL (action based temporal logic) are given out in this paper. These allow people to study, evaluate and define the dynamic behavior of current user interfaces.

Key words Graphical user interface, formal method, model checking, temporal logic, LOTOS (language of temporal ordering specification), ACTL (action based temporal logic).