

可视化、智能化 用户界面管理系统的模型及实现^{*}

吴刚 董金祥 何志均 陈刚

(浙江大学人工智能研究所 杭州 310027)

摘要 本文描述了用户界面开发工具 ZDIS (ZheDa interactive system) 系统的设计思想、体系结构和实现技术. ZDIS 系统包括 5 部分: 基于事例学习的自适应环境、基于直接操纵技术的可视化操作环境、基于事件模式的任务调度器、基于面向对象技术的交互类集和运行支持工具集、基于形式文法的界面描述语言. ZDIS 系统支持并发、多线索的用户界面机制, 并提供了动态用户界面的构筑方法, 此外在用户界面的测试、评价方法上也进行了有益的探索.

关键词 用户界面管理系统, 直接操纵, 协同过程, 多线索.

中图分类号 TP311

自 80 年代初用户界面管理系统 UIMS 概念^[1]提出以来, 新的观念、新的技术不断引入这一领域, 从最初的 Seeheim 模型^[2]到现在的 EIS 模型^[3]等, 传统的以字符应答方式为主的人机交互技术逐步让位于以可视技术为核心的图示化、智能化用户接口形式, 并把各媒体形式溶于一体, 构筑这种形式的用户界面主要要解决以下几个问题^[4]: (1) 如何用较自然的方式去构造人机界面; (2) 如何产生适应于用户意旨、知识和技能的更为精确、含义鲜明、语意透明的高效率模型; (3) 如何支持并发、多线索的应用环境; (4) 动态用户界面的生成和界面的评测机制.

问题(1)的解决方法, 目前普遍采用基于直接操纵技术的可视化操作环境. Shneiderman 最早提出了直接操纵技术 DM (direct manipulation) 的概念^[5], 它的 3 个基本特征是: 兴趣对象的持续显示、以可视动作来代替复杂的句法、提供快速可视可逆操作. 它对于兴趣对象所产生的影响立即可见.

采用 DM 技术, 解决了设计原型的直观显示问题, 提供了快速原型的方法, 它的精髓即“所见即所得 (WYSIWYG)”. 第 3 节详细介绍了 ZDIS 系统的可视界面编辑器中如何使用 DM 技术的.

* 本文研究得到国家 863 高科技项目基金资助. 作者吴刚, 1968 年生, 博士生, 主要研究领域为人机工程, 计算机图形学, 人工智能. 董金祥, 1945 年生, 教授, 主要研究领域为计算机图形学, CAD, 计算机集成制造, 人工智能. 何志均, 1923 年生, 教授, 博士生导师, 主要研究领域为人工智能, 计算图形学, 集成技术, 计算机辅助设计. 陈刚, 1973 年生, 博士生, 主要研究领域为并行工程, 人机工程, 人工智能.

本文通讯联系人: 吴刚, 杭州 310027, 浙江大学人工智能研究所

本文 1996-04-08 收到修改稿

问题(2)的解决方法,已经提出了许多用户界面定义方式,主要可分为3类^[6~8]:语义转换网络、形式化描述方法和事件描述方法.事件描述方法由于它的表达能力非常强,可支持多线控制结构而在目前普遍采用.作为这一问题的另一个方面就是如何使界面能主动去捕捉用户的操作习惯、操作意图,因此引入了自适应界面的概念,它包括3方面的内容:建立知识模型,以决定界面什么时候转换;能对用户需求的变化作出响应;保证改变的数量和类型不会引起界面设计的不一致.第2节介绍了ZDIS系统在这一方面的处理方法.

问题(3)的解决方法,当前有许多途径,ZDIS系统采用事件驱动的控制策略,通过X-Window的事件和通讯机制以及UNIX环境下的轻量级进程加以实现的.ZDIS系统引入了任务调度器的概念,它提供了基于分布式设计环境下的任务的并发、多线索调度策略和数据共享方法,在第4节中描述了它的实现机制.

问题(4)的解决方法,就要求采用面向对象方法来构筑用户界面的交互类集和运行支持工具集,OO方法所提倡的抽象数据类型、封装、继承以及多态性特征,使得作为用户界面基本元素的对象能够重用,同时带来了设计上的灵活性和可扩充性,在第5节中介绍了ZDIS系统的交互类集和运行支持工具集的设计.

1 ZDIS 系统的体系结构

ZDIS系统是一个面向CIMS集成应用环境的用户界面开发工具,它构筑在X-Window的Motif或OpenLook上,其体系结构如图1所示.

从图1可以看出ZDIS的工作流程如下:

(1)设计人员通过对用户应用领域的交互模型的建模,来确定整个交互系统的风格和模式,其中数据源是界面模式库和建模知识库,这2个工具还没有完全实现,而是由设计人员根据库的格式直接定义的;另一个是模式抽取工具由已生成的用户界面提取出来的,这个已实现.

(2)对界面原型框架,通过可视界面编辑器,完成布局的细化、事件的连接、用户接口的定义等各项工作,其中数据源是交互类集、运行支持工具集以及抽象事件集.

(3)对生成的界面进行模拟、测试、评价,可返回(2)进行修改.

(4)界面代码生成,ZDIS提供了2种途径,①C/C++语言,②界面描述文件.前者与应用程序联编,后者通过调用接口装入.

(5)自动装配,把应用程序、界面生成代码或描述文件与系统底层库装配成交互式应用环境.

(6)用户通过交互设备与任务调度器进行数据和控制的交互.其中有一运行开关,以决定用户是否需要通过自适应工具集进行界面的调整,当界面定型以后,就不需设置该开关,以提高运行效率.

2 自适应界面实现技术

自适应界面是通过对用户交互过程中各交互对象使用频度、空间和时序特性的统计,以及交互过程的物理记录(如出错频率、求助次数等),基于用户概念模型、领域概念模型和交

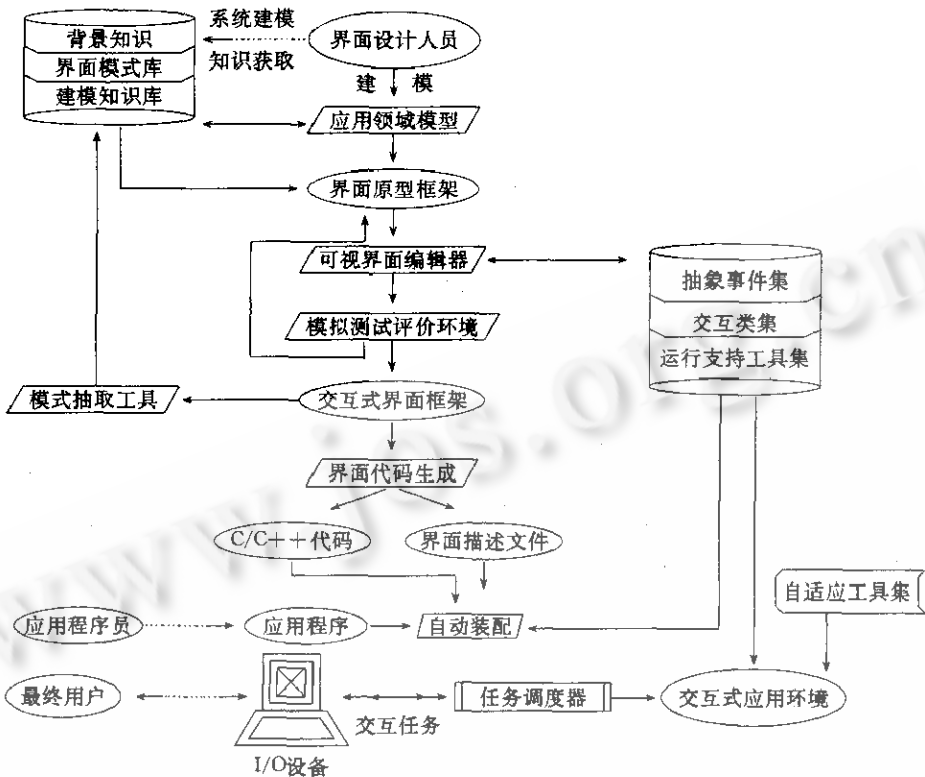


图1 ZDIS系统的体系结构

互模型,推导出用户交互能力、熟练程度和交互习惯,根据界面知识库当中定义的自适应策略对用户界面进行调整,从而使交互环境成为更灵活、高效的一种界面形式。

ZDIS 的自适应模型如图 2 所示。其中用户模型是评价用户的操作能力、操作习惯的基础,它是基于历史的记录分析建立的,用户被分为生手、初级、一般、熟练和专家 5 级,根据该模型和交互的历史流水帐,推理机可测算出用户的相应级别,从而触发转换开关,改变界面的交互模式、帮助模式等内容。领域模型

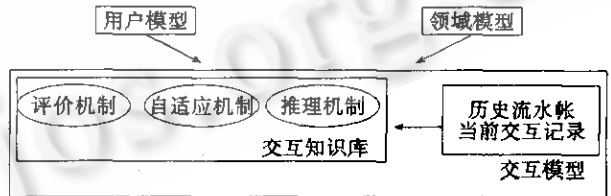


图2 ZDIS的自适应模型

描述了应用任务功能的划分和相应的系统对话和表现。交互模型通过上面 2 个模型,基于交互知识库中的知识对交互历史流水帐和当前交互记录进行分析、评价,从而触发自适应机制。下面是其中的一条交互记录示例:

MdP1T3MdP2—— 移动鼠标按下按键 1 过了 30s 后按下按键 2。

在 ZDIS 系统中,界面模式是一个(L,S,A)的三元组,其中 L 代表界面布局,S 代表交互风格,A 是交互对象属性值。界面知识库定义了界面设计的评价原则、自适应机制和推理机制,它是一个产生式系统,用〈对象,特性,取值〉来表示事实,如〈Button, Background, Red〉,因此,问题状态的描述就可以分解为交互任务中对象的描述及对象间内在关系的描述,主要谓词有 PLINK,FRQ,MOVE 等,下面是 ZDIS 中常用的一条规则示例:

如果 2 个按键在时序上相邻使用的频度大于 70%，则应把它们放在空间相邻的位置上
 $PREMSIZE;FRQ((PRESS_B1, PRESS_B2, SEQ), >70)$

$ACTION;PLINK(B1, B2)$

其中 $FRQ(A, N)$ 表示事实 A 的频度范围 N , $PLINK(A, B)$ 表示邻接放置 A, B ; 因此自适应过程可用下面的形式化方法加以描述:

设 $M=(L, S, A)$ 是界面模式集, $F(M, D)$ 是界面知识库中的谓词, 它主要根据当前模式 M 的交互属性值 A 的统计数据集 D 来改变界面的屏幕布局和交互风格, $V(M)$ 是界面模式的评价函数, 初始模式 $m_0 \in M, P_n = \bigcup_{i=0 \dots n} m_i$ 是从 m_0 到 m_n 的模式序列, 对于当前模式, $m_i = (l_i, s_i, a_i), l_i \in L, s_i \in S, a_i \in A, \exists P_i = (m_0, m_1, \dots, m_i)$, 且满足:

(1) $V(m_i) > V(m_{i-1})$

(2) $\exists f_i \in F, d_{i-1} \in D, f_i(m_{i-1}, d_{i-1}) \vdash m_i$

统计数据集 D 的主要内容包括对象使用频度、相邻使用特性、操作对象相识性、操作移动距离等; F 中主要使用的规则是: (1) 时序操作相邻的交互对象应放在空间相邻的位置上; (2) 使用频度最高的交互对象应放在最易操作的地方。

这些规则在国外一些著名的 CAD 软件如 Euclide, CADD5 等都有所体现, 经 ZDIS 构造出的 CAD 系统 ZD-MCAD 也采用了这一方法。

在 ZDIS 系统中还引入了设计约束, 主要分为 3 种: (1) 基于原则和经验的常识性约束, 这些约束以具体的指标来对界面设计进行评价和指导; (2) 基于对象间关系的关联性约束, 这些约束定义了对象空间位置的关联和对象间蕴含的从属关系; (3) 基于操作对象和操作序列的控制性约束, 这些约束是指对于交互对象在某一模式下合法操作的辨识以及合法操作序列的检查。

3 可视界面编辑器的实现方法

ZDIS 系统的可视界面编辑器采用 DM 技术和可视技术, 因此使用 ZDIS 构造界面, 不需编界面程序。具有如下特点: (1) 编辑器的所有动作均以所见即所得方式给出; (2) 生成界面的外观和联接动作均可实时模拟; (3) 提供回溯机制, 支持 Undo 和 Redo 操作; (4) 支持剪裁板技术; (5) 支持约束关系, 包括对象空间依赖关系及分组技术的使用。

(a) 对象类编辑器

在 ZDIS 中交互对象可以复合与分组, 从而实现属性的继承和同质, 还可以定义对象间的相互约束和相互依赖关系。交互对象被分为 4 类:

(1) 顶层 Shell 类, 负责与环境的交互;

(2) 包容 Composite 类, 提供子对象的复合机制;

(3) 基元 Primitive 类, 定义了十几种最基本的交互对象, 如 Window, Menu, Button 等;

(4) 标准组合件 Standard Widget, 如各类对话框的外壳、命令盒等, 用户可定义新类加入系统。

(b) 菜单编辑器

菜单是用户界面使用频度较高的构件, ZDIS 提供了完备的菜单构造手段, 从操作形式

上看可分为:弹出式、下拉式、静态式、粘连式;从控制结构看可分为单一菜单、线性结构菜单、树结构菜单、非循环网状菜单和循环网状菜单 5 种。

(c) 图符编辑器

提供了可嵌入中文的彩色图符编辑,支持多种文件格式和编辑手段。

(d) 在线帮助编辑器

ZDIS 提供了超文本自动生成器 MMHG^[9],用以生成帮助文档,它的结构如图 3 所示。MMHG 中的文本的组织参照 SGML 标准,SGML 是一个开放的国际标准,具有描述文字、引用图形、图象、语音及相互关系的能力,应用系统可约定语种(包括汉语),定义文档结构和控制代码集。

(e) 中西文编辑器

基于 UNIX 操作系统下的 X-Window 环境,提供了中西文编辑器 CCVI,它采用外挂式输入方法,支持常用的编辑操作。

(f) 事件联接编辑器

事件联接编辑器负责可视界面编辑器与应用程序接口的定义与描述,它是基于事件驱动的,事件来源于人机交互过程。最基本的事件可分为 2 类:交互对象触发产生的事件(Object Action),如选中菜单项、按下按键等;硬件设备产生的事件(Physical Event),如按下特定的控制键,特定区域的鼠标动作等。对基本事件的形式化描述如下:

```
Event ::= <Object Action> | <Physical Event>
Physical Event ::= {[Modifier_List] Basic_Event_type}
Modifier_List ::= <[!][~]{Modifier}> | "None"
.....
```

与事件相联接的操作有 4 个来源:(1)标准的系统函数,如 UI_Button_Enter, UI_Put_Confirm 等;(2)应用程序函数及其参数的定义;(3)直接输入的 C/C++ 代码;(4)直接的交互对话过程,ZDIS 支持动态交互对话,每一个交互对话框都是可视界面编辑器生成的子对象,其顶层是 DialogShell。

ZDIS 对事件的执行是通过任务调度器来处理的。经过事件联接器联接的对象,形成了一颗事件树,它提供了任务调度器的入口。

4 任务调度器的实现机制

ZDIS 系统对并发、多线索环境的支持是通过任务调度器来实现的,它支持任务间的互斥、并发、延迟、等待和异步等操作,并支持任务间的数据共享,它是建立在 X-Window 通讯机制和 Unix 的轻量级进程基础上的。

下面给出一些概念的定义^[3,10]。

定义 1. 协同过程 Coroutine. 协同过程是一段可挂起和重新激活的程序。它包括一个当

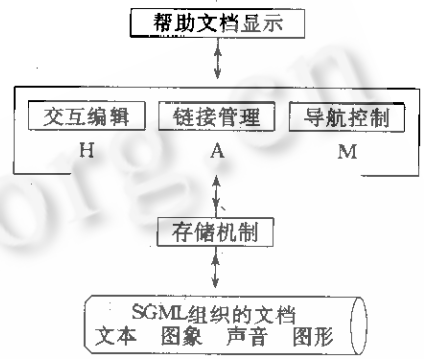


图3 MMHG的体系结构

前程序指示点、当前运行状态和局部堆栈,它有激活、阻塞和完成3种状态.

定义2. 任务 Task. 任务是一个拥有独立的控制线索和状态信息,由任务标识符定义的程序段. 它是由相互通讯的 Coroutine 组成,同一任务下的 Coroutine 是同步执行的.

Task 是 ZDIS 任务调度器的核心单元,它是通过轻量级进程实现的. 它的调度策略采用基于优先级的非抢占 Non-Preemptive 方法,图4是任务调度器运行示意图.

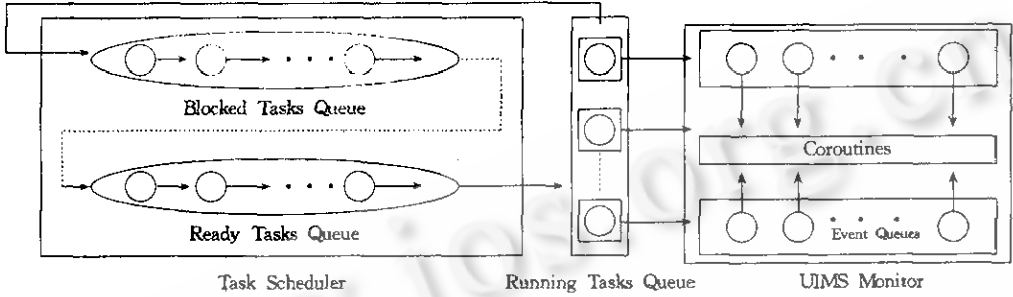


图4 任务调度器运行示意图

Task 间通讯采用 X-Window 的 Selection 机制实现的,数据共享采用 X-Window 提供的 CutBuffer,图5是它的结构框图.

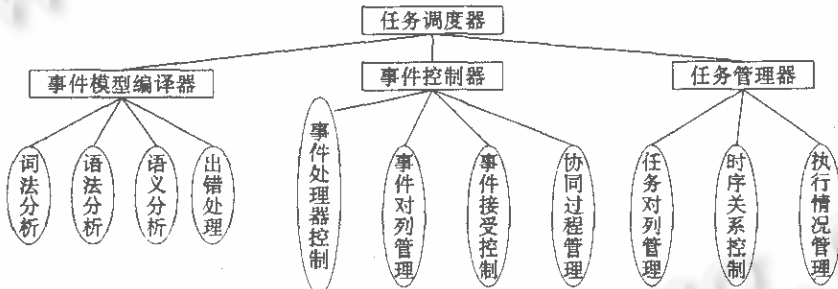


图5 任务调度器的结构框图

5 交互类集和运行支持工具集

交互对象类是 ZDIS 处理的基本单元,它可用 $T_{class} = (T_{relation}, T_{attribute}, T_{method}, T_{event})$ 四元组来表示,其中 $T_{relation}$ 是它在类层次树中的位置,它与其它类的关系和一些说明信息; $T_{attribute}$ 定义了它的基本属性; T_{method} 定义了它的类方法; T_{event} 定义了所处理的事件和相应的事件处理器. 具体介绍可参阅第3节.

运行支持工具集提供了大量的辅助工具,它包括以下几个主要方面:(1)自动演示功能,它可自动记录用户的所有交互活动,并以幻灯片的形式加以重放;(2)有向图、表、树的自动生成工具,它可按照界面知识库中定义的生成原则,生成相应的较美观的显示外观,这已在 MMHG 中加以使用;(3)版本管理功能,界面设计是一个不断精化的过程,ZDIS 支持设计过程中的版本更新和回溯;(4)系统管理工具,包括运行日志、系统环境编辑功能.

6 界面描述语言

ZDIS 提供了交互对象描述语言和交互对话描述语言,它具有第4代语言的特征——高

度的非过程化和直观性. 交互对象描述语言用 BNF 范式定义, 主要定义用户 Instance 的属性和方法, 通过一个描述转换器构造出相应子例的数据结构, 并负责创建、修改和删除, 支持动态装入, 下面是它的一个示例:

```

INSTANCE          ins_name
%%{
  NAME            name
  PARENT          parent_name
  GEOMETRY       geometry_attr
  GRAPH          graph_attr
  .....
  METHODS        { {
                    WHEN          condition
                    HANDLER       ev_handler
                    ARG_TYPE      args
                    .....
                  } }
}
}%%

```

交互对话描述语言是 UIMS 中最为复杂, 也是研究最多的领域.^[6~8]ZDIS 采用事件—进程方法来描述交互对话, 并把简单的时态逻辑引入了描述当中, 这样可提供并发控制机制, 方便实现进程间的同步、异步操作. 用户界面的对话功能

是通过其事件处理器的模式集合来描述的, 事件处理器的模式可用(N, M, R, P, Q)五元组来表示, N 代表事件处理器的名字及其驻留对象; M 代表事件处理器所处理的事件类型集; R 代表事件处理器的局部数据区; P 代表事件处理函数集; Q 代表事件处理队列. ZDIS 提供了基于物理输入事件的虚拟输入事件描述 VIE(virtual input event), 它包含了基本事件信息、时间标签、主机号等内容, 基于此提供了事件的定义文件 EDF(event definition file), 它包括了事件间的 *, +, ∩, ∪ 等操作. 界面描述语言的处理流程见图 6.

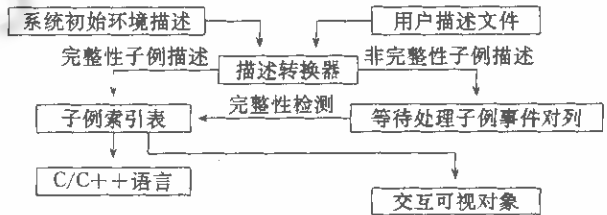


图6 界面描述语言的处理流程图

7 应用实例

ZDIS 系统的开发平台基于 UNIX 环境下的 Motif(或 OpenLook), 利用它我们开发了 863 项目“集成化 CAD/CAPP/CAM 系统”的应用交互环境, 由于涉及方面较多, 重点介绍一下它的自适应技术的应用.

在该系统中, 分别为初学者和专家级建立了相应的用户模型. 根据用户设计中的出错率, 寻求帮助次数以及复杂命令使用次数确定用户的级别, 从而提供相应的操作模式: 命令式、菜单式、宏语言式; 以及相应的帮助模式: 详细、概略、浏览; 并根据按键使用特性, 自动调整菜单排列和热菜单区中的内容.

“图视化、智能化用户界面开发工具 ZDIS 系统”作为国家 863 高科技研究课题, 从 1994 年开发完成至今, 其成果已在 20 多个不同的科研项目和商品化软件中使用, 并于 1995 年获

得浙江省科技进步 3 等奖. 本文是对整个系统的实现机制的综述性介绍.

参考文献

- 1 Thomas J J. Architecture for a user interface management system. *Eurographic Seminars: User Interface Management System*, 1985. 81~85.
- 2 Green M. The university of alberta user interface management system. *Computer Graphics*, 1985, 19(3):205~213.
- 3 Frankowski E N, Wood W T, Larson J. Concurrency and multi-threaded interaction in the task-script user interface model. *Engineering for Human-Computer Interaction*, 1993. 359~379.
- 4 Foley J, Kim W C, Kovacevic S. A knowledge-based user interface management system. *CHI'87 Proceedings*, 1988. 67~72.
- 5 Shneiderman B. Direct manipulation: a step beyond programming languages. *IEEE Computer*, 1983, 16(8):57~69.
- 6 Green M. A survey of three dialog models. *ACM Transactions on Graphics*, 1986, 5(3):244~275.
- 7 Bos J. Abstract interaction tools: a language for user interface management system. *ACM Transactions on Programming Languages and Systems*, 1988, 10(2):199~204.
- 8 Robert, Jacob J K. A special language for direct manipulation user interface. *ACM Transactions on Graphics*, 1986, 5(4):238~317.
- 9 吴海腾, 董金祥, 吴刚. 超文本自动生成器 MMHG 的设计与实现. *计算机工程*, 1995, 21(5):252~255.
- 10 Buhr P A, Strooboscher R A. The μ system: providing light-weight concurrency on shared-memory multiprocessor computers running UNIX. *Software-Practice and Experience*, 1990, 20(9):929~964.

THE MODEL AND IMPLEMENTATION OF A GRAPHICAL INTELLIGENT USER INTERFACE MANAGEMENT SYSTEM

WU Gang DONG Jinxiang HE Zhijun CHEN Gang

(*Institute of Artificial Intelligence Zhejiang University Hangzhou 310027*)

Abstract This paper presents the design thoughts, architecture and implementations of ZDIS (ZheDa interactive system) system. The system consists of five main components: an adaptive environment based on case study, an visualization design environment by direct manipulation, a task scheduler based on event model, a set of object-oriented widgets and run-time support tools, a scripts using formal grammar. ZDIS supports concurrency and multi-threaded interaction. It also supports dynamic user interface and evaluation environment.

Key words User interface management system, direct manipulation, coroutine, multi-thread.

Class number TP311