一个图形UIMS支持的用户接口描述方法及自动生成技术

黄涛 王裕国

(中国科学院软件研究所)

THE SPECIFICATION METHOD AND AUTOMATIC GENERATION TECHNIQUE FOR USER-INTERFACES SUPPORTED BY A GRAPHICAL UIMS

Huang Tao and Wang Yuguo

(*Institute of Software, Academia Sinica*)

ABSTRACT

This paper introduces the specification method for user-interfaces in a graphical User Interface Management System—GUIDE-GKS System. And the rapid prototyping and automatic generation technique for user-interfaces based on the specification information also are described. Among those, on the basis of the idea of the user's conceptual model, it presents a new kind of "guide" style specification technique, with which the user-interface designer can learn to use the system more conveniently, succinctly and effectively. This specification technique also enhances the guarantee of correctness for the process of the user-interfaces' specification, and encourages the creation of good designs for user-interfaces by making tasks or activities that result in such designs easy to accomplish.

摘 要

本文介绍了一个图形用户接口管理系统——GUIDE-GKS 系统中的用户接口描述方法，以及利用这些描述信息而进行的用户接口快速原型和自动生成技术。其中，基于用

1989 年 11 月 17 日收到，1990 年 1 月 23 日定稿。

户概念模型的思想，它提出了一种新的“引导式”描述技术，该技术使得接口设计者对系统的学习使用更为方便、简洁、有效，增强了接口描述过程的正确性保证，并易于导致好性能接口的开发。

§1. 引 言

用户接口管理系统(User Interface Management System, 简称UIMS) 是一个集成化的软件环境，它根据用户接口设计者提供的接口规格描述，自动生成某交互式应用系统的用户接口，并对接口的整个软件生命期进行集中统一的支持、控制和管理。

自从Newman 第一次提出这一基本概念以来[1], 对交互式系统用户接口友善性的要求不断提高，用户接口的设计复杂性及研制费用也相应随之提高，因此人们对用户接口管理系统的研制日益重视，近几年来，国际上已为此而相继召开了几次专题讨论会[2][3][4], 开始形成了一个逐渐加深和扩大的研究趋势。目前，在国外已有几个这样的系统被实现[5][6][7][8][9][10][11][12]。

UIMS 的探讨和研制在我国刚刚起步，并正在逐渐展开。GUIDE-GKS 系统(Graphical User Interface Design Environment, 简称GUIDE, 又含有其中用户接口的描述主要采用“引导式” (guide) 技术之意，它是在Graphical Kernel System(简称GKS, 一个国际标准图形软件系统) 之上实现其图形功能的), 是我们在SUN 3/60 图形工作站上、并在UNIX 操作系统支持下自行设计实现的一个图形UIMS, 它主要面向计算机辅助设计(CAD)、计算机辅助教学(CAI) 等交互式应用系统图形用户接口的设计和管理。GUIDE-GKS 系统采用外部控制UIMS 结构，在描述交互式对话的方式上，原则上是基于状态转换网络与事件方法相结合的模式，但在接口描述的具体技术和过程上，提出了一种新的用于用户接口描述的“引导式” (guide) 描述技术，并提供一种与之相兼容的形式化语言描述作为辅助手段。

下面我们大致介绍一下在GUIDE-GKS 系统中，图形用户接口的描述及自动生成机制。首先，我们应明确该系统所具有的一般的用户概念模型。

§2. GUIDE-GKS 系统一般的用户概念模型

为了开发一个应用系统，设计者首先需要有一个该应用系统的用户模型。因此，在开始设计GUIDE-GKS 系统时，我们首先必须明确的是，用该系统所能生成的交互式图形用户接口、以及一起被集成的整个应用系统应表现的一般的用户概念模型[3], 这种概念模型决定了UIMS 的实际功能、以及它所能产生的用户接口的一般风格。这样，接口设计者在使用UIMS 为某应用系统生成一个用户接口时，他就必然应该先有一个具体的用户模型，此模型可以是抽象表达的，也可能只存在设计者的头脑之中。

为了确定应有的概念模型，应确定应用系统对命令输入的响应的特色、应用系统状态的概念、以及它对用户执行过程的影响，而命令和状态的关系则是：命令促使状态的改变。

GUIDE-GKS 系统，主要是面向CAD、CAI 等领域内交互式图形用户接口的开发。

其一般的用户概念模型可概括成为分层命令(菜单) 结构。图1 所示为该模型的有向树表示。

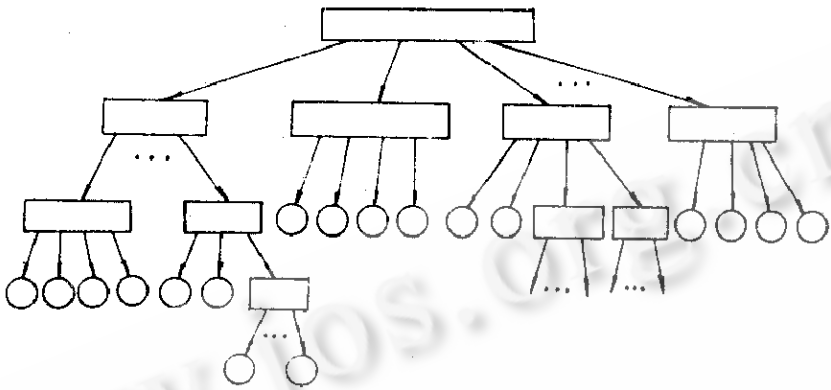

图1 分层命令(菜单) 结构 一般用户概念模型的有向树表示

在图1 中，各非叶子结点表示应用系统在某一时刻能提供的、可供选择的命令(菜单) 集，它定义了应用系统的某一个状态。用户对某命令(菜单项) 的选择执行或系统的内部运行，导致系统从一个状态转成另一个状态，在有向树中则表现为从一个结点进入到它的某一个子结点。每个状态都有一个exit 项，当选择它时系统返回到它的父结点。叶子结点中没有可供选择的命令，它只表示某个原活动的执行状态，该活动完成后，系统自动返回到该叶子结点的父结点。

在有向树中，每条有向树枝可看成是某事件的发生，其所指向的结点即为对该事件的处理器。当某事件发生时，即用户向应用系统发出了某条命令，或系统经过某一类内部运行之后，若沿此有向树枝进入的结点是叶子结点，则系统只对该事件处理，这样的活动我们称之为原活动；若沿此有向树枝进入的结点是非叶结点，则系统除对该事件进行处理外，还等待用户再次发出命令、或等待系统的某类内部运行，之后可能产生新的事件并进行以后相应的处理。

与此一般用户概念模型相对应，一个具体的用户接口模型应具有图2 中确切的内容。

GUIDE-GKS 系统将用户接口从应用中分离出来，完成应用的全部输入输出(图形和非图形)、应用的外观表现、涉及图形编辑管理使用的一些相关功能、以及应用语义信息映射的图形视见表示等等。

鉴于GUIDE-GKS 系统能够自动生成的交互式图形用户接口采用上述的一般用户概念模型，在系统本身的总体功能控制结构的设计中，我们也采用这种分层命令(菜单) 控制结构，并尽量使各子功能模块的使用也基本上采用这种控制结构，这样，不仅能使系统功能便于实现，也使设计用户接口时所使用的交互式接口与以后生成的接口的风格一致。

- 非叶结点
  - 命令(菜单项)
    - 名字，图形表示索引;
    - 本项性质(如是否继续分层);
    - 与其它命令(菜单项) 之间的联系
    - (如它可激发的命令集(状态));
  - 命令层(菜单)(状态)
    - 命令方式(如菜单、功能键、字符命令、应用内部运行判断等);
    - 菜单(命令集) 的布置、表示及特性;
    - 接口状态的外观，静态图索引及描述;
    - help 信息;
    - 标准组合子例程要求及对应的应用语义子例程联系;
- 叶子结点(原活动)
  - 对应的应用子例程;
  - 该原活动的反馈表现，静态图描述及索引;
  - 对输入、输出(图形、字符、数据) 的要求;
  - 基本控制结构要求(如sequence, case, circle 等);
  - undo 对应的应用语义子例程联系;
  - 输入
    - 数量，类别，工作站要求，设备的选择;
    - 提示方式，应答方式，表现形式;
    - 初始值要求，输入数据限制条件;
    - 接收信息的接口/ 应用公用信箱的协议指定;
  - 输出
    - 输出量，类别，风格(包括颜色、线型等等);
    - 输出数据的来源指定;
    - 交互式绘制时的数据交互输入方式;
    - 输出信息的接口/ 应用公用信箱的协议指定;

图 2

§3. 用户接口的设计描述

现有的各种UIMS 的工作，都是基于要创建的用户接口详细描述之上的[4]，这些描述包括屏幕布置、用户与计算机之间的对话、以及用户接口与应用程序其它部分的界面等等；而描述的程度则是相当详尽的，例如接口设计者必须说明菜单的位置、菜单项的次序、用户与用户接口之间的各种对话细节等等。

虽然现存的UIMS 已显著地提高了接口设计者的生产效率，但是，为了根本上使接口设计者的生产率再进一步有所提高，同时使得UIMS 本身易于被学习和使用，这要求UIMS 呈现一种新的类型，它的工作将使用要创建的用户接口的一种更高层的描述[4]，即根据被提供的任务模型、应用功能以及用户特性等等，只是基于这样一些用户接口应该做什么的、一个非常高层次的说明，UIMS 将自动进行用户接口的详细设计和实现。

但是，这种用户接口的自动设计要求超出了目前的技术状况，这在自动编程技术方面和用户接口设计知识的整理方面，需要有进一步的探索、研究和进展，而设计实现这种新类型UIMS 也是非常复杂和困难的。因此，目前还没有这种UIMS 被实现过。

现有UIMS 提供的各种用户接口详细描述，无论是BNF 描述[13]、状态转换图描述，还是上下文无关文法描述、基于事件的描述，其形式实际上大都先是确定了一种与该技术相对应的规格化描述语言(Formal Specification Language)。为了能对用户接口的各方面进行详尽的设计描述，这个语言体系一般会变得庞大且复杂，这带来的最大问题之一是使得UIMS 难于学习和使用，尤其对于UIMS 初学者来说这点更甚；再者，当接口设计者使用这些形式化描述语言进行用户接口描述时，在描述工作繁杂的情况下，一般会经常发生描述错误，有时一些错误难以查出，而且一般也不可能得到描述的同时隐在错误的即时反馈。

正是基于以上各因素的考虑，在GUIDE-GKS 系统的设计实现中，我们对用户接口的描述采用了一种新的方法，称之为“引导式”(guide) 的描述方法，这种方法除满足用户接口描述的一般要求外，还具有其特殊的优点：①用户接口设计者很快便能掌握其使用，即易于学习，而且使用起来非常方便、简炼、清楚、易懂，不必象对形式化描述语言那样，记忆繁多刻板的规则和格式，故效果上提高了设计者用户接口开发的效率；②描述信息易于被用来对用户接口进行快速原型、自动生成以及修改等工作；③增强了设计者对用户接口描述的正确性保证，因为它确保了系统所接收的、设计者对用户接口每项描述的正确性，一旦发现描述上的错误，立即通知设计者，并且便于设计者对错误的即时改正；④利于系统为设计者提供某些缺省值(default values)，引导设计者设计出正常的和好性能的用户接口，也增强了开发出的各用户接口的一致性。

3.1 “引导式”(guide) 描述方法

“引导式”描述方法的基本思想就是，系统基于一般的用户概念模型，根据相对应的正确的规则来提问或提示，设计者只需输入一些简单的回答或进行某些选择，便可完成对设计的描述。

在GUIDE-GKS 系统中，交互式图形用户接口的描述按其中的描述性质可分为两类：一类是词法描述性部分，说明接口的外观特性、输入输出的目的和内容、以及所运用的交互技术和显示技术，在具体用户接口概念模型有向树表示中，对应于该部分描述的是各非叶结点和叶结点内容；另一类是语法描述性部分，说明接口的交互会话控制结构，

在具体用户接口概念模型有向树表示中，与该部分相对应的是各结点之间的连接关系。

在GUIDE-GKS 系统中，采用“引导式”描述方法对用户接口的语法描述性部分进行描述时，系统根据分层命令(菜单) 结构的一般用户概念模型，以一定的次序向设计者提问，接口设计者则根据从用户处得到并确定了的、具体的接口用户模型，做出相应的一些简单的确定或回答，系统便自动在内部建立一个满足其要求的接口描述。由于这个过程是由系统引导设计者完成的，因此可保证用户接口交互会话控制结构的语法正确性。在对用户接口词法描述性部分描述时，形式上也表现为与生成的用户接口一致的接口风格，逐层深入，由概括到具体地提示设计者。例如，如果设计者说明某项活动要求输入操作时，系统则进入要求设计者说明输入的过程，这要求设计者指明哪种类型的输入操作? 在哪台工作站上? 用什么物理设备? 又根据输入操作的类别，要求设计者指明各种可能交互方式的哪一种应被选用? 以及还有关于交互方式的诸多细节指定等等。系统的提示通常采用菜单等形式，设计者只需做出简单的选择即可，非常方便；要求设计者输入简单的数据时也提供提示信息，且可进行立即检查，还可以使用适当的缺省值(default values)。由于在这种描述过程中，系统对设计者输入的描述信息的检查是解释性的，这使得一旦接口设计者输入了错误的描述信息，便可立即得到系统的反馈，使错误可马上得以更正，确保系统接收的描述信息的词法正确性。

采用“引导式”描述技术进行接口描述的过程，实际上是对具体用户概念模型的有向树表示的一个遍历过程，在这过程中进行交互会话控制结构描述及词法描述，这个遍历过程一结束，便完成了用户接口结构和内容的完整描述。

在“引导式”描述的同时，GUIDE-GKS 系统将所确定了的描述信息自动进行内部组织，由于有些相关性质的信息是被集成在一起进行描述的，因此系统内形成一块块描述信息类，这些描述信息块之间的内部组织结构，则与分层命令控制结构的用户概念模型相对应，这样，接口描述信息就能很方便地被用来进行用户接口的快速原型、代码自动生成以及修改等工作。

3.2 “引导式”描述和与之相兼容的传统形式化语言描述的交叉描述方法

交互式图形用户接口在GUIDE-GKS 系统中的描述，主要是依靠“引导式”描述技术来进行的。但是，当接口设计者一旦变得很熟悉本系统的描述过程，即成为这方面的“专家”(expert) 后，有时可能会觉得这个描述过程较为冗长，他并不需要系统提供如此多的提示信息，而是希望尽快地表达出所有的接口有关描述信息。为此，GUIDE-GKS 系统另外还增有接口的形式化描述语言(Formal Specification Language) 机制供接口设计者采用。

下面列出的是这个形式化描述语言的部分扩展BNF 表达，其中:

{ }: 表示可以出现一次或多次;

[ ]: 表示至多出现一次;

|: 表示“或者”。

```
<interactive_control_structure> ::={<state_node> | <primitive_action_leaf>}
<state_node> ::=STATE_NODE<name>
                    [ICON→<icon module name>]
                    LAYER→<layer module name>
                    INTO_OR_PERFORM→<node_and_leaf_sequence>
                ENDn_NODE
```

```
<node_and_leaf_sequence> ::=<state node name>
                           | <primitive action leaf name>
                           | <node_and_leaf_sequence>; <state node name>
                           | <node_and_leaf_sequence>; <primitive action leaf name>
<primitive_action_leaf> ::=PRIMITIVE_ACTION_LEAF<name>
                         [ICON→<icon module name>]
                          PERFORM→<primitive action module name>
                        END_LEAF
<LAYER MODULE> ::=LAYER MODULE<name>
                         TYPE→<layer_type>
                         [MENU_POSITION→<point>]
                         [SYSTEM_TRIGGER→<application sub_routine name>]
                         [INITIAL_GRAPH→<static_graph module name>]
                         [HELP→<help text file name>]
                           GRAPH_EDIT_MARK→YES|NO
                           GRAPH_ASPECTS_MARK→YES|NO
                         [<graph_edit_management>]
                    END_MODULE
<layer_type> ::=MENU|KEY_CHARACTER|.....|FUNCTION_BUTTON|SYSTEM
<ICON MODULE> ::=ICON MODULE<name>
                       {<graphical_element>}
                  END_MODULE
<STATIC_GRAPH MODULE> ::=STATIC_GRAPH MODULE<name>
                              {<graphical_element> | <icon_transform>}
                          END_MODULE
   ...                  ...                ...                ...
<PRIMITIVE_ACTION MODULE> ::=PRIMITIVE_ACTION MODULE<name>
                         [INPUT→<inputs>]
                         [PRESENT_GRAPH→<static_graph module name>]
                         [APPLICATION→<application sub_routine name>]
                         [UNDO_APPLICATION→<application sub_routine
                         name>]
                         [OUTPUT→<outputs> |{<outputs>}]
                         CONTROL_STRUCTURE→<sub_control_structure>
                         [BRANCE_NUMBER→<num>]
                         [BRANCH_VARIETY→<common mail_box name>]
                         [CIRCLE_VARIETY→<common mail _box name>]
                              END_MODULE
<sub_control_structure> ::=SEQUENCE|BRANCH|CIRCLE
<inputs> ::=<input module name> | <inputs>; <input module name>
<INPUT MODULE> ::=INPUT MODULE<name>
                       <locator> | <stroke> | <valuator> | <pick> |.....| <string>
                  END_MODULE
<locator> ::=LOCATOR RECORD
              WORKSTATION→<in_workstation>
              DEVICE→<in_device>
```

```
        INITIAL_POSITION→<point>
        PROMPT_AND_ECHO→<locator_prompt_and_echo_type>
        ECHO_AREA→<echo_area>
        LOCATOR_POSITION_STROKE→<common mail_box name>
        LOCATOR_PROCESS_NUMBER→<num>
    END_RECORD
...          ...          ...          ...
...          ...          ...          ...
```

GUIDE-GKS 系统中接口形式化语言描述的模块化结构，使得接口的设计描述具有一定的面向目标的设计风格，而且具有很大的灵活性。由于形式化语言描述中的每个模块结构，对应为“引导式”描述过程中接口的一类描述信息块，而且这两种描述方式相应的信息内部组织也完全相同，所以，在GUIDE-GKS 系统中进行接口描述时，可以交叉使用这两种技术。例如，接口设计者在“引导式”描述过程的某一个状态下，当不需要系统提供过多的提示信息，而希望熟练地尽早描述完某一部分时，他可要求系统转到用形式化语言进行描述的状态，来对某一类描述信息块相对应的模块结构进行描述；当他又需要系统来引导他进行接口描述时，他可又要求系统转回到“引导式”接口描述的状态，来继续完成接口的描述工作。又如，接口设计者可以先用形式化描述语言的方法，完成接口的交互会话控制结构的描述，然后又依此结构进入到“引导式”描述的状态下，完成对接口其它性质的描述，在这个过程中又可出现上例的那些情况，如此等等。这样，在GUIDE-GKS 系统中，接口设计者便能更加方便、灵活地完成用户接口的设计描述。

§4. 用户接口的快速原型(prototyping)和修改

用户接口的描述信息由设计者输入进GUIDE-GKS 系统后，被组织存贮在用户接口设计描述信息库中，只要它们被装载到按一定结构组织的数据结构中，则可进行该用户接口的快速原型(prototyping)。这时是采用表驱动(table-driven)的方式，即由一相当于驱动程序地位的程序来启动，依靠系统自动建立的一个适当的运行环境，来驱动这些数据结构，在这些结构中不但有接口的交互会话控制结构描述，也有词法性描述，在驱动过程中还可能激发到按描述说明的交互技术和显示技术，因此，在终端前的设计者和用户看来，这样就具有一定的所设计用户接口的大致表现和风格了，进而可分析评价其是否满足要求，以考虑进一步的改进。

如果在快速原型过程中，设计者想要对用户接口的描述做某些修改，GUIDE-GKS 系统则提供一定机制，使得设计者可以中途挂起快速原型过程，而转入对用户接口描述进行修改的功能模块，修改完成后，又可转回到原来被挂起的地方继续进行快速原型。由于修改只是在用户接口的描述信息之上进行的，并没有改变快速原型的驱动控制程序，因此用户接口被挂起修改后，并不会影响快速原型过程的立即可继续运行。

为便于接口设计者对用户接口的局部修改，GUIDE-GKS 系统除提供有可按名在用户接口结构描述中进行定位的功能之外，还有更为便利的机制。它可根据接口的设计描述，在屏幕上显示出一个类似于图1 的用户接口控制结构模拟图，这样，接口设计者便可很方便地在该图上指定应修改的相应部位(例如用mouse)，进行如删除某结点描述内容、改变结点之间的连接关系、增加结点描述以及重新描述某结点信息内容等等，如此使得

修改更为直观。而在修改描述的具体过程中，还可以以“引导式”风格来进行。

§5. 用户接口的自动生成及与应用的集成

用户接口的自动生成及与应用部分集成的全过程如图3所示。

GUIDE-GKS系统对其自动生成的交互式图形用户接口的运行时支持如图4所示。

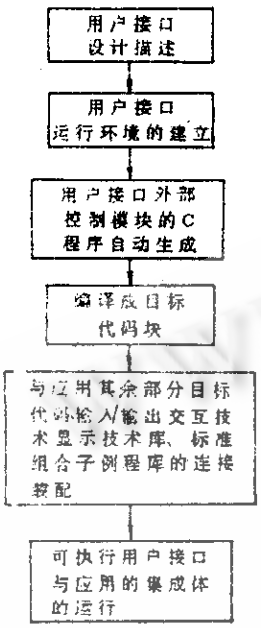

图3　GUIDE-GKS系统中用户接口自动生成流程图

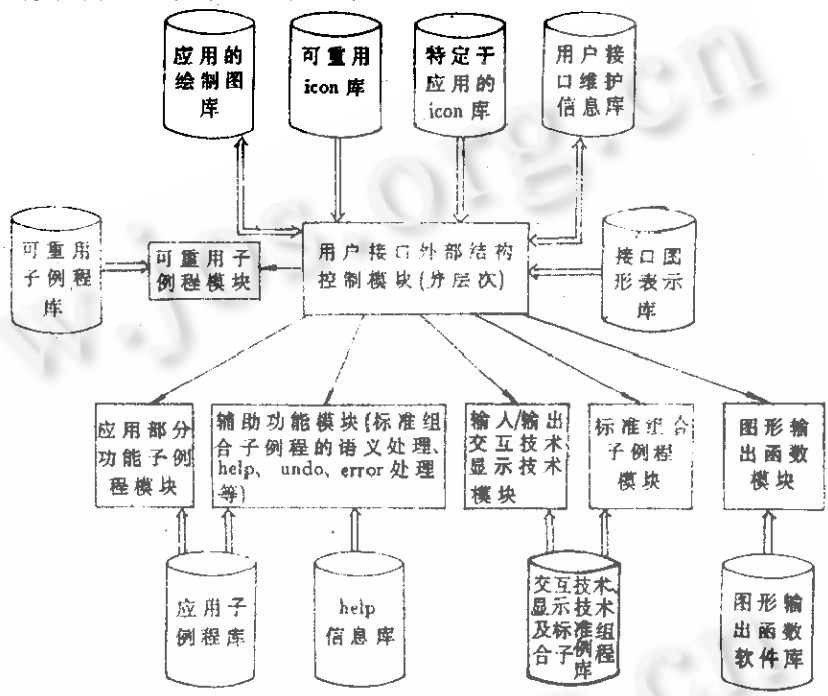

图4　GUIDE-GKS系统支持下应用的用户接口运行时各功能模块及库存之间的关系

§6. 结束语

GUIDE-GKS系统的实现符合UIMS的一般设计要求，它在很多方面确实具有不少的优点，是一个功能较强的用户接口管理系统。但是，它还存有不足之处，其中最主要的就是它的应用面的局限性，即它所能生成并支持的交互式图形用户接口，都具有分层命令控制结构的一般用户概念模型。当然，在它所面向的应用领域内，这些接口控制结构类型是能够基本上满足要求的。

我们考虑GUIDE-GKS系统以后的进一步发展是，如果合理运用人工智能领域中的诸如框架理论及其推理等技术，通过将系统设计者持有的各种一般用户接口概念模型，在系统中恰当地表达，同时配置相应于这些表达的解释过程以及描述信息转换机制，这样便可实现系统的多用户概念模型性质。从这点来看，以后的GUIDE-GKS系统又是一

个“用户接口管理专家系统”，系统中可表达的各种一般用户接口概念模型，即对应为各类接口设计、生成和管理经验，接口设计者利用系统设计生成一个用户接口时，便是使用其中某些经验来“引导式”地进行接口描述，然后由系统自动生成用户接口并进行维护和管理。由于在“引导式”接口描述过程中有效地使用了推理等技术，就使得接口的设计更为方便，且更易于导致好接口的实现；又由于可有各种一般的用户接口概念模型的预先选择，故在更多方面应用领域内，对于更多种交互会话控制结构和风格的用户接口，其描述、自动生成和管理得以支持。

参考文献

[1] Newman. W., "A sytem for Interactive Graphical Programming", Proceeding of AFIPS Spring Joint Computer Conference, 1968, PP.47-54.

[2] Thomas. J. and Hamlin. G. eds, "Graphical Input Interaction Technique [GIIT]", Rpt. on workshop held at Battelle Seattle Conf. center, June 1982, Computer Graphics, Vol. 17, No. 1, 1983.

[3] Pfaff, G. E. (ed.), 《User Interface Management System》, Springer-Verlay, Berlin, 1985.

[4] ACM SIGGRAPH Workshop on Software Tools for User Interface Management, Computer Graphics, Vol. 21, No. 2, April, 1987, pp.73-145.

[5] Green, M., "The University of Alberta User Interface Management System", Computer Graphics, Vol. 19, No. 5, 1985, PP. 205-213.

[6] HILL, R. D., "Supporting Concurrency Synchronization in Human-Computer Interaction: The Sassafras User Interface Management Systems", ACM Trans. Graph., Vol. 5, No. 3, 1986, PP.179-210.

[7] Olsen, D., et al., "SYNGRAPH: a Graphical User Interface Generator", SIGGRAPH'83, Computer Graphics, Vol. 17, No. 3, July, 1983, PP.43-50.

[8] R. J. K. Jacob, "Executable Specifications for a Human-Computer Interface", Proc. CHI'83, Dec. 1983, PP.28-34.

[9] W. Buxton, et al., "Towards a Comprehensive User Interface Management System", Computer Graphics, Vol. 17, No. 3, July, 1983, PP. 39-42.

[10] Stephen P. Guest, "The Use of Software Tools for Dialogue Design", Int. Journal of Man-Machine Studies 16, (1982), PP.263-258.

[11] D. J. Kasik, "A User Interface Management System", Computer Graphics, Vol. 16, No. 3, July, 1982, PP.99-106.

[12] Gurminder Singh and Mark Green, "AUTOMATIC GENERATION OF GRAPHICAL USER INTERFACE", 《Graphics Interface'86, Vision Interface'86》, PP.71-76.

[13] 陈华生、王靖宜、宋勇，“交互系统用户接口的描述方法”，计算机研究与发展，Vol. 25, No. 4, 1988, PP. 18-24。