GI——一个图形交互用户接口生成器

柳西玲 孙智勇

(清华大学计算机系) (中国软件公司)

GI—A GENERATOR OF GRAPHIC INTERACTIVE USER INTERFACE

Liu Xiling

(*Tsinghua University*)

Sun Zhiyong

(*The Software Company of China*)

ABSTRACT

User interface management system (UIMS) is one kind of software development tool. According to the characteristic of the user interface of VLSI/CAD software, we design and implement a UIMS–GI, which supports graphical interaction. It is based on the simplified event-respond model and its main aim is easy used. It produces the user interfaces of the application programs according to the user's interface descriptions. GI supports menu, keyboard and window input and serves the application programs in the parallel mode, so it not only has good interactive characteristic, but also the entirety and independence of the application programs.

摘 要

用户接口管理程序UIMS是一类软件开发工具。GI是根据VLSI/CAD软件用户接口的特点而设计和实现的一个图形交互式UIMS。它基于简化事件响应模型，以使用简便为主要目标，由用户说明而自动生成相应的应用程序用户接口。它支持菜单、

1989 年8 月2 日收到，1989 年11 月20 日定稿。

键盘、窗口等输入方式，以异步并行方式为应用程序服务。它不仅有较好的交互性能，还保持了应用程序的独立和完整性。

§1. 引 言

计算机的广泛应用，要求软件具有友好用户界面的呼声日益高涨，现在对于软件的实用化、商品化而言，友好的用户界面不再是无足轻重了。方便用户几乎成为实用软件的第一目标，而图形工作站的出现更进一步推动了这个趋势。因为图形方式的交互接口有主观、易学、方便等优点，越来越多的得到用户欢迎。然而开发友好的用户界面，工作量较大，有时高达整个软件开发的20%-30%，实际上各应用程序的用户界面在功能上却十分相似而互不相容，在这种情况下提出了UIMS 的概念，自1982 年在ACM/SIG/GRAPH 的GIIT 讨论会上正式提出UIMS 这个术语以来，它已成为近十几年来软件工具研究的热点之一，而且已开始出现一批试验成果[1-5]，其效果很令人鼓舞。

UIMS 的提出是为了提供一套构成和控制用户与计算机资源之间交互对话的软件工具，它的主要优点是能更快更可靠的应用于软件开发，生成不同领域的一致性用户接口，并使所生成的接口易于修改。这对VLSI/CAD 应用软件的实用化、商品化也极为重要。目前VLSI 生产的飞速发展，开始形成由设计、生产各步骤的应用软件集成为一体化的综合设计自动化(或半自动化) 系统。其中不少应用软件是早已开发好，有些可能尚待去开发，但对于这样一种集成性高的系统，面临如何去组织一个统一的屏幕设计，即友好用户界面的问题，目前许多工作站所提供的各种窗口或图形的低层软件还不能满足它们的要求。GI 是以X_window 作为低层软件支撑，在输入方式上增加菜单描述种类及操作的一个用户接口生成器。它将图形子程序包作为一种特殊应用程序相接，从而使它具有独道之处，使用简便、程序紧凑是它的最大优点。

§2. GI 的目标

2.1 GI 的软硬件环境

GI 的硬件环境是各种工作站，带有高分辨率图形终端和鼠标器。X_window 是它的软件支撑，这是近几年流行的软件，它的第十一版本将要成为一个低层窗口软件的国际标准，因此立足于它，将会使GI 具有更好的通用性，另外它的层次窗口树和图形操作管理上的输出功能基本能满足VLSI/CAD 应用程序的输出要求，因此在GI 设计时只需重点放在增加输入管理的功能上。

2.2 GI 的具体目标

a. 菜单输入管理:

在形式上支持固定菜单和pop-up 菜单两种；在内容上支持字符串、图形及用户

自定义符号；在操作上支持移动、滚动和换页。

b. 键盘输入管理:

允许回显操作和回送操作，回显操作又分为系统回显和用户回显两种，后者对用户每键入的回显内容由应用程序自定。回送操作又分为立即回送和行回送两种。

c. 窗口操作管理:

支持图形窗口、文本窗口和状态窗口三种操作。对图形窗口提供显示图形、输入点、线、矩形、多边形操作。对文本窗口提供字符串显示、自动滚屏、移动光标、输入字符串等操作。对状态窗口只提供文本输出操作。

d. 屏幕状态管理:

支持用户对各窗口、菜单在屏幕上状态的调整，改变各I/O 单元的工作方式。

e. 与应用程序间的通讯:

保证与应用程序连接方便，并能把GI 接收到的外部输入传递给应用程序。这要求有简洁的描述语言让用户可以表达接口形式定义。

总之要GI 方便使用，并有较好的响应速度，如光标不能太迟于鼠标。同时应充分利用X_window 的功能，精炼它的结构。

§3. GI 的结构

GI 的基本结构如图1:

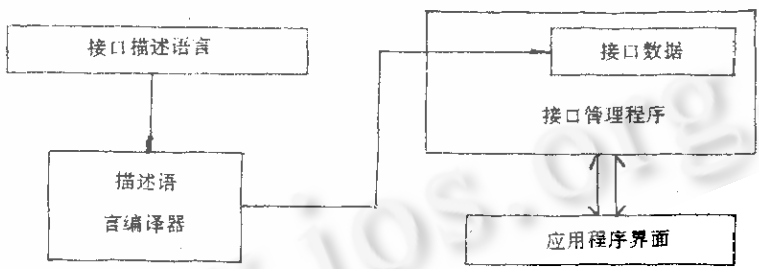


图1　GI 的结构

GI 可分为四大部分：描述语言编译器GIC, 接口管理程序SM, 应用程序界面TOOL, 输出子程序包。其中输出子程序包可以是某种低层图形软件包，如GKS、X_window 之类。只要协商好窗口软件资源的使用划分，保证运行中不冲突，GI 可将它看作为一个应用程序处理，因此我们本文主要讨论前三部分的实现和设计。

描述语言编译器是将用户对接口要求转换成接口管理程序中所需的接口数据。

接口管理程序是GI 的核心，它根据用户的要求去产生相应的接口，并对其进行协调管理，外部对接口的各种操作都是由接口管理程序接收响应的。

应用程序界面是一个函数包，它为应用程序提供控制和使用接口管理程序的方便，即通过它，应用程序可启动或停止接口管理程序，改变接口管理程序的状态，读取接口管理程序从外部读入的各种数据。

具体地说，GI 的使用步骤如下:

(1) 用接口描述语言描述用户对接口的要求。

(2) 将由描述语言编译器产生的接口要求与接口管理程序的基本部分构成完整的接口管理程序。

(3) 应用程序再调用界面子程序形成控制、使用接口管理程序的模块。

以上三部分中，编译器是比较独立的部分，与另外两部分只是通过接口要求(数据)发生联系。但接口管理和界面子程序之间关系密切，二者间有各种数据需传递，它们设计的好坏不仅影响GI 性能，还会影响到应用程序与GI 之间的关系。

§4. GI 的接口描述语言

接口描述主要对接口两方面性质进行说明，一是说明接口的外部形状(包括大小、位置坐标、颜色等)，二是对接口所需动作过程的说明，也就是对外界输入接口必须做相应的响应。通常这种描述按输入输出单元进行分类说明，分类的详细程度反映语言的描述能力强弱，但分类过细也会使语言复杂，难于记忆掌握。GI 面向VLSI/CAD 应用软件，各应用程序的接口风格是一致的，也就是生成接口统一格式，可使描述语言简洁方便，它对接口按输入单元进行适当分类，对一些细节采用缺省方式，将它们分成三大类: 菜单输入、键盘输入以及窗口输入。其中菜单输入又分为字符、图形、二维三个子类，通过鼠标选择。对窗口输入又分为字符、图形二个子类，而且对窗口输入为不使应用程序难于控制，只指定初始输入方式，应用程序允许重新对它设置，这样GI 无需管理窗口输入的全过程，将它交给应用程序管理，这一方面可使GI 管理减轻，同时不会使得应用程序被动而不方便，因为窗口输入由应用程序管理更自然些。

描述语言的BNF 范式如下:

<接口描述> ::=< 单元类> : < 单元名>;
< 接口外形描述>;
< 类格式>;

< 单元名> ::=< 字符串>

< 字符串> ::=< 英文字母串>

< 单元类> ::=< 菜单类> | < 键盘类> | < 窗口类>

< 菜单类> ::=< 字符菜单> | < 图形菜单> | < 二维菜单>

< 窗口类> ::=< 图形窗口> | < 字符窗口>

< 字符菜单> ::=C_menu_name

< 图形菜单> ::=G_menu_name

< 二维菜单> ::=D_menu_name

<键盘类> ::=B_input
<字符窗口> ::=C_window
<图形窗口> ::=G_window
<接口外形描述> ::= Z_START=<x 方向坐标>;
Y_START=<y 方向坐标>;
WIDTH=< 宽度>;
LENGTH=< 高度>;
FORECOLOR=< 前景色>;
BACKCOLOR=< 背景色>;
<类格式> ::=< 菜单格式> | < 键盘格式>
<键盘格式> ::=PROMPT= “提示符”;
MODE=< 回送方式> | < 回显方式>;
MAXCHAR=< 行最大长度>;
<菜单格式> ::=FORM=(“字符串1”, 动作1, ……, “字符串n”, 动作n);
MODE=< 排列方式> | < 激活方式> | < 滚动方式>;
DISPLAY=< 显示项数>; [< 显示列数>];
< 显示项数>, < 显示列数> ::= 整数;
< 排列方式> ::=C
< 激活方式> ::=A
< 滚动方式> ::=P
< 字符串1>, < 字符串2>, ……, < 字符串n> ::=string
< 动作1>, < 动作2>, ……, < 动作n> ::=string
< 回送方式> ::=S
< 回显方式> ::=D
< 提示符> ::=string
< 前景色>, < 背景色> ::= 黑| 红| 兰| 黄| 白
< 宽度>, < 高度> ::= 整数
<x 方向坐标>, <y 方向坐标> ::= 浮点数

从以上范式看出描述语言的简洁，这使它能在满足用户要求下，较快地开发出来，另外GI还允许以上三大类输入中任何类的单元成组输入，这也增加了功能，同时方便使用。

§5. GI 接口管理程序 SM

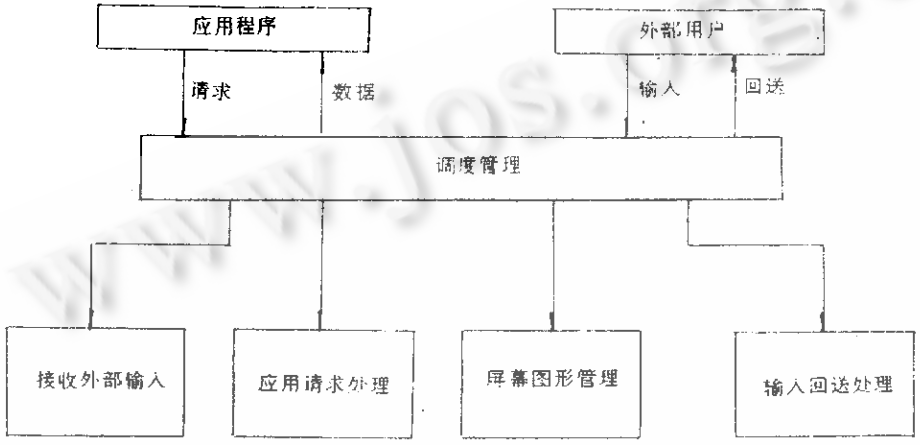
SM 是GI 的核心，它的功能是：由接口描述语言产生的接口数据显示相应的接口图形，管理用户的外部输入，并向应用程序传送这些外部输入；以及响应应用程序发出的请求。

5.1 SM 的结构

从外部特征看，SM 有两个输入流和两个输出流，从功能特征看，它具体完成四类工作:

· 接收外部输入(鼠标或键盘输入)。
· 接收应用程序请求。
· 屏幕图形管理。
· 给应用程序传送数据。

它的结构如下:



调度管理根据输入流的来向以及接口当前所处状态，选择决定由下面四类工作之一去完成。它采用简化的事件处理模型，在下节中将详述这种模型。

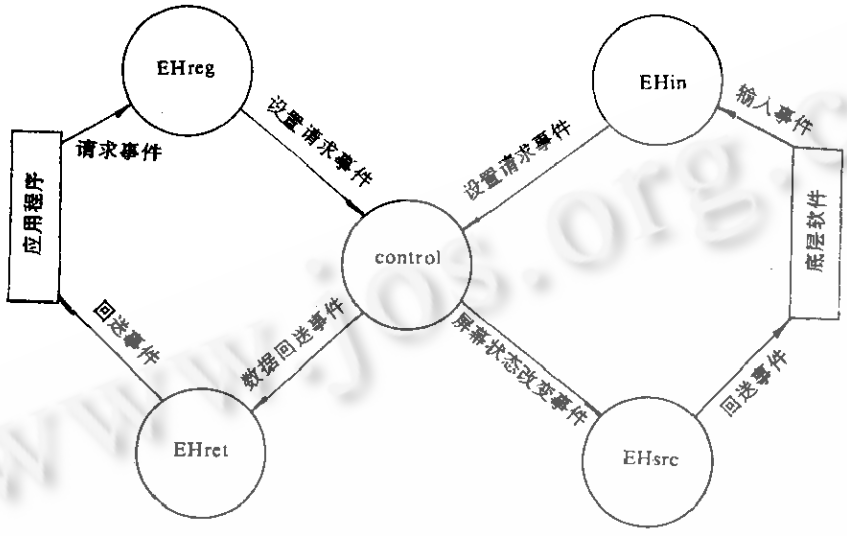
5.2 SM 的管理模型

SM 管理模型可抽象为事件处理模型，在该模型中有事件E, 事件处理器EH, 状态T 三种元素: S=(E, EH, T)。

通俗地讲，事件是GI 中的各类数据，事件处理器是对数据进行各种加工。事件可用一个三元组来定义: E=(eh, type, value), 其中eh 是该事件发向的事件处理器，type 是该事件的类型，value 是该事件的值。而事件处理器又可用一个二元组来定义: EH=(Q, m), 其中Q 是该事件处理器的输入队列，m 是事件处理器的变换函数，即处理器的操作。

状态是指GI 在某时刻的状况。它由各输入输出单元的状态组成，即$T = \{T_i\}$, 其中每个T_i 的值域是TV={ACTIVE, DEACTIVE, PAUSE, STOP, POPUP, COVER}, 其中ACTIVE 表示相应的输入输出单元处于活动状态，可接收输入; DEACTIVE 表示相应输入单元处于非活动状态，暂不能接收输入，但屏幕上有显示; PAUSE 表示相应的单元还未取走输入数据; STOP 表示相应单元不在屏幕上; POPUP 表示相应单元处于弹出状态(只有菜单可能); COVER 表示相应单元被覆盖。

针对GI 四种工作，可设它们是四个事件处理器，所以EH={EHinput, EHrequire, EHscreen, EHreturn}，而事件可分为菜单输入、键盘输入、窗口输入、设置请求、状态改变、数据回送等类，GI 的状态过程图如下:



此模型是用事件和状态相结合来驱动程序的，从状态过程图看出，所有事件先送往control 这个调度处理器，由它根据一定规则，选择一个事件发送到相应的事件处理器，等该事件处理完后，调度处理器再选下一个事件发出，这时引入调度处理器，把原来并行事件结构改为串行事件处理结构，这样调度处理器的选送事件原则，也就是调度原则很重要，它会影响执行效率。在SM 中，主要采用两条调度原则:

· 所有事件是先登记后处理。这原则保证了事件不丢失。

· 外部输入事件优先处理。具体地说，对事件分成两个队列，即外部输入事件队和应用程序请求队，在处理外部输入队列时一次将队列全部处理完，而处理应用程序请求队列时一次只处理一个事件，这样自然保证了外部输入的优先级，又使管理简便。

5.3 SM 的操作处理

输入处理、屏幕管理、请求接收和数据回送是SM 的四大操作。输入处理的工作量最大，与底层软件联系最密切，它按菜单输入、键盘输入、窗口输入三类处理。只是在窗口输入中有不同的橡皮筋提示，这是VLSI 的特殊要求。

屏幕管理与底层图形软件有关，主要能实现I/O 单元的图形显示，POPUP 菜单实现，由于立足于X_window 之上，它有很强的图形和窗口管理，因此这种操作实现比较简单，特别是I/O 单元的覆盖管理。

请求接收主要接受应用程序发来的各种请求，对它们进行分析、执行时，或去改变GI 状态，或去调用屏幕管理改变屏幕内容。

数据回送的任务是将用户输入的数据存放到共用数据区中。由于GI和应用程序是异步并行工作，有可能出现在一个数据还没被应用程序取走之前，用户又有新的输入，为保证不丢失数据，就要建立必要的缓冲机构，GI采用的是一级缓冲，即为每一个I/O单元提供一个缓冲区。由于外界输入速度与计算机速度相比慢很多，通常这种处理可以满足要求，如引入多级缓冲将会影响GI的工作效率。

§6. GI与应用程序的界面

GI的最大特点是与应用程序处于并行地位，一般UIMS与应用程序之间有三种连接方式：内部控制方式、外部控制方式、并行方式。并行方式在实现上困难多，但在操作和使用上有前两种控制方式所不具备的优点。因为在内部控制方式时，应用程序运行与接口运行串行工作，这大大降低了接口操作的灵活性，而外部控制方式要求应用程序分成几部分插入，应用程序不能自己组成一个统一整体，它要靠UIMS连接，使得应用程序独立性差，编写、维护困难。并行方式可克服这些缺点。

并行方式在管理上要复杂一些，它会涉及到进程的中断、同步信号传输、共享内存等问题，实质上要解决二者间界面基础，GI的界面是建立在管道、共享内存和信号机构之上。管道是为传送简单数据和同步控制所用，共享内存是为大量数据传递迅速所用。

具体地说，对应用程序而言，它所了解的界面只是一个函数库，它用调用函数的方式与GI相连，这个函数库由三类函数组成：输入函数、控制函数、设置函数。

输入函数用来读取SM传给应用程序的输入数据，它包括：

m_read (name, string, type, mode); 读取菜单输入

d_read (name, string, type, mode); 读取命令行输入

g_read (name, x, y, count, button, type, mode); 读取图形窗口输入

t_read (name, ll, cc, line, type, mode); 读取文本窗口输入

控制函数用来控制GI的工作状态，如启动、停止等，它包括：

start( ); 启动GI

stop( ); 停止GI

sigrequire(opr, obj); 改变输入单元obj的状态

opr是改变状态的操作

设置函数用来改变各输入单元的参数：如窗口提示符、输入方式等，它包括：

m_set (name, attribute, mode); 菜单设置

d_set (name, attribute, mode); 命令设置

g_set (name, attribute, mode); 图形窗口设置

t_set (name, attribute, mode); 文本窗口设置

下面给出一小段程序例子是应用程序调用函数的使用：(C语言编程)

```
main ( )
    {
     ... ...
```

```
... ...
start ( );   启动GI 运行
sigreguire (ACTIVE, "menul");   激活菜单menul
sigrequire (ACTIVE, "window 1");   激活窗口window 1
m_read ("menul", choice, WAIT);   读取菜单menul
if (strcmp(choice, "rectangle")==0)
  {
    g_set ("window 1", INPUT, REC);   设置窗口window 1 是矩形输入
    g_read ("window 1", x, y, c, button, WAIT);   从窗口window 1 中读取一个矩形
  }
if (strcmp(choice, "polygon")==0)   设置窗口是多边形输入
  {
    g_set ("window 1", INPUT, POLY);   从窗口读取一个多边形
    g_read ("window 1", x, y, c, button, WAIT);
  }
......
......
  }
```

以上例子看出，从应用程序的角度看，与GI 连接很简单，并不需要了解GI 的工作细节。

§7. 结论

GI 目前已实现在HP 工作站上，并已有应用程序相连的实例，它的编译器是利用UNIX 系统的YACC 和LEX 工具实现，这样开发速度快又便于今后描述语言的修改、增删。其他程序全部用C 语言编译。

参考文献

[1] Balbir S. Barn, "Graphical Interaction Management", Computer Graphics Forms, July, 1987.

[2] Dan R. Olsen Jr., "SYNGRAOH: A Graphical User Interface Generator", Computer Graphics Form, June, 1983.

[3] Dan R. Olsen Jr., "MIKE: The Menu Interaction Kontrol Environment", ACM Transaction on Graphics, Oct. 1986.

[4] Mark Green, "A Survey of Three Dialogue Models", ACM Transaction on Graphics, July, 1986.

[5] Ralph D. Hill, "Supporting Concurrency Communication and Synchronization in Human-Computer Interaction", ACM Trasaction on Graphics, July, 1986.

[6] Robert-J. K. Jacob, "A Specification Language for Direct Manipulation User Interfaces", ACM Transaction on Graphics, Oct. 1986.

[7] Robert W. Scheifler, "The X Window System", ACM Transaction on Graphics, April, 1986.

[8] Strubble H. J., "Role, Model, Structure and Construction of a UIMS Working Group Report", Computer Graphics Forms, Mar. 1983.
[9] W. Buxton, "Towards a Comprephmensive User Interface Management System", Computer Graphics Forms, June, 1983.
[10] Domain/Dialogue User's Guide.

第三届全国青年计算机会议(NC YC'91)通知

会议将于1991年5月23日至26日在湖南省大庸市召开，参加会议的对象为年龄在35岁以下的计算机工作者。

这次会议是中国计算机学会计划召开的一级学术会议。学会委托国防科技大学计算机系兼研究所承办。参与协办的单位有湖南省宇航学会、湖南省计算机学会、湖南大学、中南工业大学和国防科技大学的四个系。

这次会议还将出版论文集，评选优秀论文，并给予奖励。此次会议愿以多种形式为国内外各公司、企事业单位提供产品的广告宣传、产品展览及产品的专题报告等服务，也热烈欢迎关心青年成长和我国计算机事业的个人、团体和企事业单位提供赞助。联系地址：410073 湖南长沙国防科技大学计算机系杨涛收。愿征订论文集的单位和个人请与NCYC'91组委会联系。凡愿参加本会议的同志(无须提交论文)，可向NCYC'91组委会索取详细材料。

联系地址：410073 湖南长沙国防科技大学计算机系吴涛收。

第四届全国知识工程研讨会征文通知

由中国计算机学会软件专业学会智能软件学组主持，长沙国防科技大学承办的第四届全国知识工程研讨会将于1991年10月在湖南长沙召开。会前由中国地质大学出版社正式出版会议论文集《知识工程进展1991》。现将有关事项通知如下。

一、征文内容：①基于知识的系统：特别是在工程设计、决策规划、视觉、听觉研究、实时控制及教学辅导等领域的专门问题研究和开发技术；②知识工程中的软件工具：各种开发工具、方法评价、性能分析工具等；③知识获取与机器学习：知识获取方法，各种学习方法和策略；④推理：自动演绎、不确定性推理、非单调推理、定性推理等；⑤自然语言理解与生成：自然语言分析模型和分析方法；⑥人工智能语言：人工智能语言的理论、设计、实现和分析；⑦并行及分布式知识处理：结构分析、分布处理环境、分布式语言；⑧认知模型；⑨基础研究：知识表示方法、搜索算法、理论计算机科学在知识工程中的应用等。

二、征文注意事项：按《知识工程进展1988》文体的格式；一式两份于1991年4月30日前寄北京大学计算机系张宁收，邮政编码100871，以邮戳为准。

中国计算机学会软件专业学会智能软件学组

附：第四届全国知识工程研讨会程序委员会名单

主席：石纯一 委员：孙怀民 何志均 胡运发 石纯一 刘大有 刘尊全 刘椿年 陆汝钤 许卓群 陶森兰 陈世福 陆汝占 李应谭 王树林 冯玉琳 张一立 何华灿

第四届全国知识工程研讨会：

主席：许卓群 秘书：张宁 李强