

欧洲图形国际会议文献

下 册

国家机械委武汉计算机外部设备研究所

一九八七年六月

879

0

目 录

知识引导交互作用的图形程序设计	1
图符命令 (ICDNS) 在应用程序中使用	9
HuTwINDoms一种用户接口开发的方法	21
用于射线投射显示的 CSG模型图象的生成	31
一种垂线缓冲器矢量曲面浓淡模型	43
NEM一种实现演员和物体动画的语言	51
一种三维单元帧缓冲器	62
PIPEMATIC: 一种密集管道结构设计的分布式系统	67
触摸屏与计算机图形写在科学馆中的应用	79
一种三维图形的交互涂画技术	83
ProLog中的GKS询问功能	95
GKS向布式结构扩展	101
轴面向目标的图形标准发展	109
GKS机器的概念	116

知识引导交互作用的图形程序设计

美国 新罕布什尔大学计算机科学系

Lee Tibbert, R Daniel Bergeron

摘 要

本文描述了一种将知识工程技术与一个交互式图形程序结合起来的方法。并且举出了一个交互地定义计算机系统的事例来说明此方法，在程序中加进一个基于规则的知识工程模式，而这种模式又考虑到了数据库一致性和通过一种对话引导用户的功能。

1. 导 言

知识工程作为一个功能强价格合理的程序设计工具，它的出现将促使人们对交互式图形程序的传统设计作出重新评价。把一个基于知识的系统结合到一个交互式程序的设计中去是向着真正用户友好系统发展的重要一步。应用程序员应能象过去运用分析模块那样很容易地把基于知识的模块应用到应用程序交互过程中。成功地把分析模块和图形模块结合起来，直接说明了能够从逻辑上将这些功能分解开来。同样，在交互处理中，能否有效地使用知识工程方法，则需要认真估价交互式对话的程序设计以及这样的知识库可能对这些对话产生怎样影响的。

本文描述了一种将知识工程技术与一个交互式图形程序结合起来的设计方法，并且用一个针对实际应用而实现的典型程序即一个定义计算机系统的接口说明此设计方法。在程序中加过了一种基于规则知识工程模块，且又考虑到数据库一致性保护和通过对话来引导用户能力。

2. 举例问题的说明

在笔者举例问题中涉及到计算机机房描述以及打算配置进去的详细说明和解释，规格可由经销商。现场服务处或负责安装的计算机服务指导来提出。我们假设有这么一个知识工程的程序，它可以分析出根据一整套专家规则产生说明，分析内容包括一向被人认为会引起麻烦的条件测试，以及完成表面配置，诸如电缆长度，电源要求等等。这样的配置模块很有作用，但计算量却大得无法有效进行交互式分析，然而，这些规则说明却又明显地有助于用交

互式来解决。那么为了提高整个系统的智能程度,这样的配置模块设计了一个图形接口应该考虑哪些条件呢?因为配置应用程序是一个“智能”程序,故改善而不是破坏智能的外部特征是最重要的。

利用常规的主干型方案方法能实现直接的交互式功能描述模块 LAYOUT 机房设计程序采用了这种方式,且很适用于计算机配件的安装,作为部份的物理世界且又不能变化的约束条件是很难编成代码写出图形接口程序里面,例如,虽然机房必备之用具的范围能够很容易地写到程序中去,可多数现实世界的约束条件并非一成不变,那么也就不能直接转换成代码、相应地,人们常常希望扩充基本交互程序以具备在配置程序应有的某些智能,如果没有这些,也就失去了功能描述动作的交互特点。

交互接口如果能以近乎实时响应速度查找出所谓明显的错误,那么就会提高整个设计的智能,诸如“Y型磁盘必须在距Cpu x的15吋内”之类的孤立一级规则几乎不需要什么处理,且也能很容易加到交互式功能描述模块中,而涉及全部电源要求或者总线负载规则处理量就要大得多,则必须安排在配置程序中,对规则作这样的划分通常在很大程度上与用户安排有关:明显的越权和疏忽要在定义的时候立即标识出,而更细小的错误会引起更加多的麻烦,可延到配置分析时处理。

虽说,与规定有关的配件都简单,但其更换频繁,因而使用者难以全部记住,因此交互程序应能直接访问这些规则。建立一个智能定接口的第一步就是利用这个规则去定义出数据库完整性约束条件。

一个更智能化的接口应能引导设计人员完成功能描述过程,要求有:

- (1) 一种其中每一个信息都重要的知识。
- (2) 标明如何运用图形程序获得这些信息的规则。
- (3) 相关的工作。

交互式计算机图形学、数据库和人工智能是当今程序设计者最有用处的三大有力工具,随着用户需要越来越高,系统越来越复杂,完善。在这些领域中,综合两种或多种技术运用也就变得越来越普遍。在这一节里,我们简要地评论一下几种流行的系统。

3.1 人工智能(AI)与图形学

依据人工智能和计算机图形学原理,一个共同基础正在形成,斯蒂芬(Stevena)曾描述过一个功能完善的计算辅助教学程序,广泛利用图形来表演轮船制造操作。布朗大学也做了一个靠自动决策程序提供图形能力的计划,从重建国家航天局空间计划和配置计算机系统的人工智能程序同样能看到图形学的潜力,自动改变零部件尺寸的图形设计方法之研究把图形学引向人工智能。

3.2 数据库和图形学

尼尔曾对近来的数据库——计算机图形一体化工作作出评论。为了有助于数据库信息的图形显示,好些个这样的系统正开始运用人工智能技术。格立特和富立叶曾在一个交互式程度很高的图形环境中使用过这样的方法。他们的系统可以动态刷新因数据库交互式刷新所引起的图形显示。其连续赋值限定刷新(CEQUS)采用了类似基于知识的系统中使用的生成系

统。

4. 讨 论

知识库至少在下列三种情形中特别有作用：

- (1) 利用加强数据库完整性约束条件提供数据库保护。
- (2) 在丢失一个重要信息和无明显非法动作存时提供查询。
- (3) 回答设计者之询问“下一个操作”？

4.1 数据库接口

一般交互式图形数据采集有赖于有着静态关系的提示符/菜单/帮助文件加以引导的用户启动操作。问题约束条件定义在代码中且分布在整个程序中，传统的数据库可简单地看作是一个信息栈。建立数据库是为了把约束条件检查集中起来，防止违反预定的约束条件。数据库实质是处于再激活状态。这样的数据库很有作用，但如果把它们改成为知识库，其作用更大。数据库与知识库的主要区别在于：数据库具有事实和联系，而知识库却有由解决一个问题组织起来的事实。在笔者建立的知识库中有一个表示法，它告诉知识库中包括哪些内容以及如何建立这些事实。

4.2 知识引导和交点

Learn 程序用计算机辅助指令提出了一个实例，说明交互适用于用户的熟练程度，在许多设计程序中，考虑到基于知识的程序成分影响和引导图形程序控制流很有帮助。在用户希望交互对话和因一个疏忽在其后来的设计处理中查找变得很困难时，都特别要强调这种能力。LAYOUT程序举出了一个常见的程序例子。这个举例中，设计或者说明一个环境，必须将从用户那里得到信息，隐含和明显的约束条件之数目常常多出设计者立即答出的数目，因为实际中，在必须根据新的要求，对现有设计进行修改而又不得违反原始约束条件时，人的记忆量不足就显得特别突出。

在这些情况下，如果设计者能列出使用一种系统的交互对话的直接重要性的详情，然后再切换到协议方式去就很有益处，因为在这种新方式中，由程序引导交互，把设计者的注意力集中在这方面的专家所指定的主要问题上。执行知识引导交互对话的协议程序，利用了数据库的当前状态和一组要求用户交互的相关规则，对这些规则要加以组织以描述一种信息采集序列的规范，这个规范象一个安在后台机上的工具随用随调，对用户不加约束，使用户做他们想做，要做的事情，且在用户需要更多的引导和结构时又给用户以支持。

针对用户兴趣和熟练程度之明显区别，可用知识引导交互来把握交互对话，经销商和现场服务项所需要细节的程度可能不尽相同，但两者都得使用同一数据库和同一程序，这可以用硬代码做，但交互作用的知识库则更加灵活。比如说，一个知识库包含了专家的经验，或许在一次会话中只用到其中一个，某个人对一个程序中经常使用的部份或许是内行，但却不熟悉程序的其它部份，正如[Sing 83]所述，菜单抓住了操作者的注意力，限制了选择，却使得一个知道该做什么的内行无所适从，允许知识库程序决定下一步操作的程序交互

结构，同样给专家提供了直接执行一条命令的能力，而不需要列出更多的菜单。

总之，知识库提供了一次激活的用户接口且又能动态地调用这些库，依其内部规则改变程序流向和数据库现行状态，诸如，它们提供了引导交互的基准。

5. 设计方法

为了生成一个灵活的能利用一个知识库的交互程序，有必要强调两主要方面的设计——交互处理器和知识库，由交互处理器组织显示屏上预定义区可见的内容，包括菜单，提示符，命令名，题头和草图区，而知识库介于交互处理器和数据库之间，如图一所示。

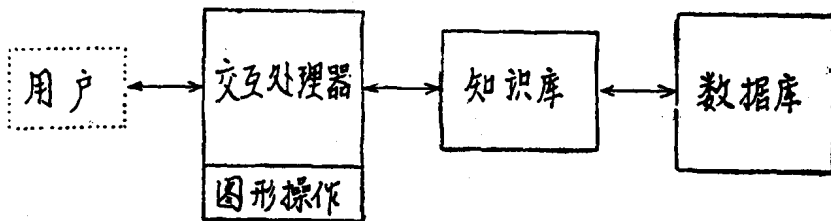


图 1.

5.1 知识库

从PROLOG语言可知，针对操作的成功或失败，建立了相应的知识库操作，而且人们又扩充了这种功能，以致于当一个操作失败时，其原因可反馈回来，知识库本身就是一组生成规则（在[Barr 83, Stef 82]中都有知识库系统的出色讨论）。

知识库有四个显著特点：

- 一组有关数据库的事实
- 一组有关交互程序事实
- 一组规划
- 一个推理规程

数据库事实：

知识库必须能访问数据库的两种特殊事实，一方面，它必须能取数据库中的信息（在图形世界中，这样的信息可简单分为两类，一类是表示物体的信息，另一类是赋予该物体的值），其次，知识库必须有关于数据库信息的知识，即，数据库表示物体的种类以及如何存取这些物体的信息，

交互程序事实：

知识库必须具备关于基本操作的知识，这些操作可由交互处理器（用户系统的命令）调

用, 知识库懂得如何调用这些命令即可, 其含义可以置在一边。

规则:

规则是知识库的主要手段, 大致上讲, 它们提供了两种基本类型的约束条件: 一致性与完整性。一致性规则禁止可能会违反某项规则面对数据库更新, 以维护数据库的完整性。完整性规则在引导方式中调用, 支持用户完成信息采集过程。为了引导用户键入其需要的数据, 这些规则把数据库状态 (诸如是否存在具有某一属性的物体) 与交互处理器的命令联系起来, 每一个规则包括一个“响应”字段, 以备万一在一次失效操作中作为处理规则之结果返回到交互处理器。

推理规程:

推理规程是知识库中的代码, 其作用就是翻译规划, 以决定完整性和一致性是否受到破坏以及 (通过规则之 响应字段) 向交互处理器汇报规则处理的结果。

多数情况下, 知识库仅对用户命令作出响应并基本上处于操接状态, 其工作就是使用督察刷新的完整性规划防止无效刷新, 除了报告出错信息外, 它只改变程序控制流向, 无其它任何动作, 仅当用户选定了引导方式时, 知识库才能运行, 这时它访问由编程人员提供的含有一致性规划的优先级表。这个优先级表可写在一内部文件中且可以刷新。如, 显示局部生成代码的变化, 表尾是一个“尾部”条件, 条件一满足就附上约束条件表空的响应返回到最上层, 一旦进入引导方式, 知识库搜索数据库查找第一个未满足条件, 然后就反馈满足那个条件的命令, 并作出相应的响应。

这样的组织形式具有相当大的灵活性, 不需要重新程序设计就能修改知识库规则, 加进新规则或改正旧规则也根本不需要动知识库。如果在新的规则上有了新的基物体或程序命令, 仅改变了交互处理器或图形程序即可。例如, 假设程序中设有定义划实践的手段且又希望有, 那么在程序中加上图形操作就是必要的。

5.2 交互处理器

交互处理器接到从知识库中来的命令, 就直接转换到那个状态, 并显示响应符号或者指示用户操作要领, 为了做到这点, 要确定好哪些提高是在任何状态下都可见的, 此外, 还要认真做好模块化。当你可以从其它任何状态进入到这一状态以及从这一状态退出到任何其它状态时, 必须特别注意显示的记录情况, 在多数传统的交互程序中, 一个模块常常利用它从调用者承接下来的有关显示环境的知识, 这样的承接可以与当前结构无关。

系统的图形程序提供了一组主干型的嵌套调用, 这些调用中的每一个基本上都是稍有区别的基本命令翻译循环程序的复制品, 为了做到在知识库指导下从一种给定状态转换到其它任何状态, 我们把主干型结构拉平成一级并取代一个有限状态机, 现行状态和从用户或知识库那里的键入决定一个动作以及下一个状态的转换。这方法长期以来一直被人建议用来取代主干型结构, 在正常程序中大量状态的出现使得编码困难了许多。但实际中, 一个清晰的主干型树可直接转换成一个等价的有限状态程序而且可以算法化, 叶节点有意义、内节点指导查找, 最烦的事情是在任何时候都得记住“双亲”节点。

采用有限状态机结构来考虑知识库引导交互，会产生意想不到的收益，程序一旦具有直接从状态到状态转换，而不需要中间状态转换的能力，自用户利用这种能力就容易多了，和屏幕上一一直有的Help键一起的还有一个Command键、正是由于Command命令调用文本分析程序作代价，用户可以得到一个比菜单系统更方便的系统和一种多面性的直接命令语言。

6. 程 序

如上所述，我们已建立了现实世界中以约束条件来配置计算机组件的主要概念，被我们称之为XGRAPH程序，在一台配有有Unix操作系统的VAX780驱动的AED512图形设备上运行。图形部份用C语言写成。采用了乔治，华盛顿大学的CORE图形系统版本。知识库用佛兰兹的LISP语言写成且已开始转换成PROLOG语言工作。XGRAPH无论在用户启动或者引导方式中都可开启，还能在这两者之间转换，除非调用它，否则，引导方式的存在仅只影响目标码所占空间的大小。

下例几幅图说明了基于知识的程序之特征，图2表示了一个约束条件违例的图形响应，为使现场可打开机壳门，每一个单元需要定义一个保留区，屏幕中间的“+”表示了一个用户打算移动的磁盘单元的位置，由于磁盘的新的服务区占去了VAX所需用的地方，数据库可能就只得挤一点了。

一进入引导方式，就会出现图3所示的情况，在此用户还远未定义一个门，规计中之一就会检测出这一疏忽且还要指示交互处理器转向“add door”命令。

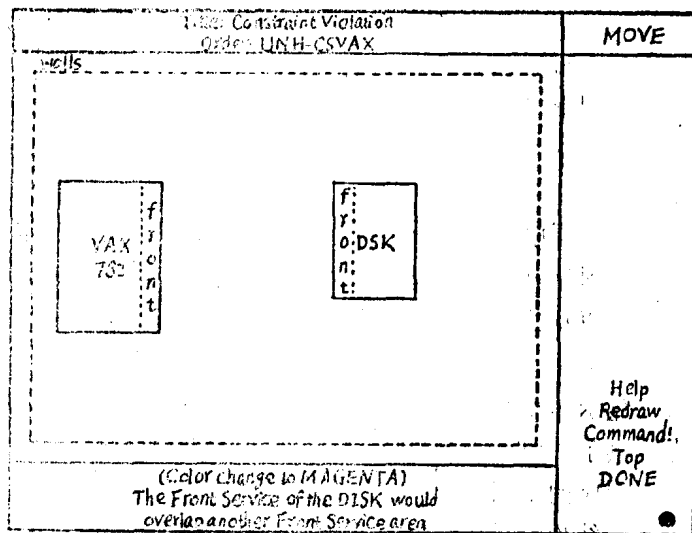


图 2.

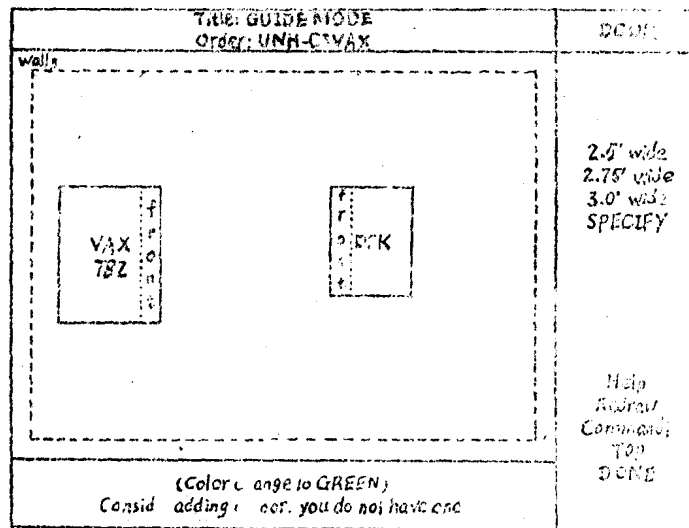


图 3.

7. 结束语和今后的方向

上述的XGRAPH程序利用采集有用的信息的交互图形提供了一种知识工程技术的具体例子，XGRAPH采用了知识工程完成两个主要操作。

1. 提供数据库完整性约束条件。
2. 提供一种知识引导的交互对话。

应用程序中很容易加入数据库完整性约束条件，而对程序其它结构不会有任何特殊作用。然而，要想提供知识引导的交互作用法，则对程序中交互处理那部份的设计有较高的要求。一种有效的设计方法是把交互处理器做成一个输入事件决定状态转换的有限状态机，其输入无论从用户或者是知识库来都行，用这种办法，知识库就可完成交互处理器到任意状态的转换。

或许，最有意义的下一步是把知识引导交互的概念应用到一个完整的用户接口管理系统UIMS中去，正如布克斯顿所描述的那样，其目的是多个显示管理，交互处理器以及知识库执行自动化。

参 考 文 献

- [Barr81] Barr, A. and Feigenbaum, E.A., Handbook of Artificial Intelligence

- vols. 1-3, esp. vol. 1 pp. 190-200 William Kaufman, Los Altos, Ca. 1981-1983.
- [Berg73] Bergson, D., Bono, P. and Foley, J.D. Graphics programming using the CORE system. *Comput Surv.* 10,4 (Dec. 1973). 389-443.
 - [Buxt83] Buxton, W., Lamb, M.R., Sherman, D., Smith, K.C., Towards a Comprehensive User Interface Management System. *Computer Graphics*, 17,3 (July 1983). 35-42.
 - [Cloc81] Clocksin, W.F. and Mellish, C.S., *Programming in PROLOG*, Springer Verlag, Heidelberg, 1981.
 - [Fein82] Feiner, S., Nagy, S., and Van, Dam A., An Experimental System for Creating and Presenting Interactive Graphical Documents. *ACM TOG*, 1, 1 January 1982), 59-81.
 - [Fole82a] Foley, J.D. the design and implementation of user-computer interfaces. *Siggraph 82 Tutorial Notes 7 ACM*, New York, N.Y., July 1982.
 - [Fole82b] Foley, J.D and Van Dam, A. *Fundamentals of Interactive Computer Graphics* Addison-Wesley, Reading MA, 1982.
 - [Garr82] Garrett, M.T., and Foley, J.D., Graphics Programming using a Database System with Dependency Declarations. *ACM TOG*, 1,2 (April 1982). 109-128
 - [Goss83] Gossard, D. Panel on Solid Modeling *Siggraph 83 Computer Graphics*, July 1982 pp. 163-164.
 - [Haye83] Hayes-Roth, F., Waterman, D.A., and Lenat, D.B., eds., *Building Expert Systems*, Addison Wesley, Reading, Ma., 1983.
 - [Kern79] Kernighan, B.W., and Lesk, M.E., *LEARN Computer-aided instruction on UNIX*, 1979.
 - [McDe80] McDermon, J., An Expert in the Computer Systems Domain. *Proc. Nat. Conf. Art. Intell. I.* pp. 269-271.
 - [Neal83] Neal, D.B Tutorial on Graphics and Data Bases. *Siggraph 83 Tutorial Notes 20*, ACM, New York, N.Y., July 1983.
 - [Newm68] Newman, William M., *A System for Interactive Graphical Programming*, AFIPS SJCC 1986, Thompson Books, Washington, D.C., pp47-54.
 - [Scar83] Scarl, E., MITRE Co., personal communication work in progress.
 - [Sing83] Singh, B., Beatty, J.C., Booth, K.S., Ryman, Ryman, R. A Graphics Editor for Benesh Movement Notation. *Computer Graphics* July 1983. pp.51-62.
 - [Stef82] Stefik, M., Aikins, J., Balzer, R., Benoit, J., Birnbaum, L., Hayes-Roth, F., and Sacerdoti, E. *The Organization of Expert*

图符命令 (ICONS) 在应用程序中的使用

英国国际计算机有限公司

H 山姆

伍德格特

摘 要

本文讲叙了在用户和应用程序之间用图形符号作为通讯操作命令的使用, 文中详论的例子是“电子数据表计算机“操作命令系统”, 该系统是按照“定点和启动”命令方法用图符就可驱动整个设计操作程序中用到的83个图符, 且还有一个“图形符号生成器”可用来扩充图符, 同时, 还讨论了用户体验。

1. 介 绍

“用户友好”是计算机应用程序设计人员的梦想, 而摆在用户面前的事实却常常是应用程序“既难用又麻烦。阻碍(非计算机)专业人员广泛运用计算机的最大障碍之一就是克服计算机专业人员所写程序中固有的人——机通信问题。

对于编写编写程序人, 或者有经验的用户来讲, 一切都可以相当清楚, 面对不熟悉这些程序的人, 却全都是含糊不清, 信息丢失, 意思误解会使得上机者沮丧地停机。因此, 一个软件产品要想应用成功, 首当其冲的是软件设计者和用户之间要有一定程序上的技术转让, 从早先人们学习使用计算机程序的方法研究可知, TV游戏建立在直接视觉刺激上(通常为圆形)不需要说明即可, TV游戏用户不需训练就达到从计算机获取知识的目的。人们也常常注意到日常生活中图符可得到广泛应用(如路标, 服务场所, 汽车检验等)。工业上许多行业已经有了图形符号通信协定, 英国标准协会所做的图形符号表较成功, 至少有28个标准。

从这些早先观察所得到的另一个结论是, 一个应用产品不仅要完成一项预定的工作, 而

Systems, a Tutorial. Artif. Intell. 18.2 (March 1982). 135-173

(also as Chapter 4 in [Haye83]).

[Stev83] Stevens, A., Roberts, B., and Stead, L., The Use of a Sophisticated Graphics Interface in Computer-Aided Instruction, IEEE CG & A, Mar-Apr 1982, pp 25-30.

彭 卫译

王世运校

且还必须，在某种程序上引起用户兴趣，把它们逐步地从初学者训练成行家，因此，开始设计程序时就得认真考虑人——机接口问题，而不是之后加进去。

从TV游戏到一个更加专业化阶段的发展，人们注意到计算机键盘自然用在输入数字和文本上，不是用在(编码)命令或在一个问题真值表中移动数据，所谓自然的方法(在本文涉及系统中)就是定点(认定部份数据)和使用橡皮和铅笔(添加、删除及更改)。这些观察表明用计算机图形学来模拟这个方法会有收益。

1.1 使用图形工作站

一九八〇年刚开始有这方面的试验，“单用户工作站”渐渐有了市场，且还得出两个重要功能，即

- 局部驱动高速交互式图形

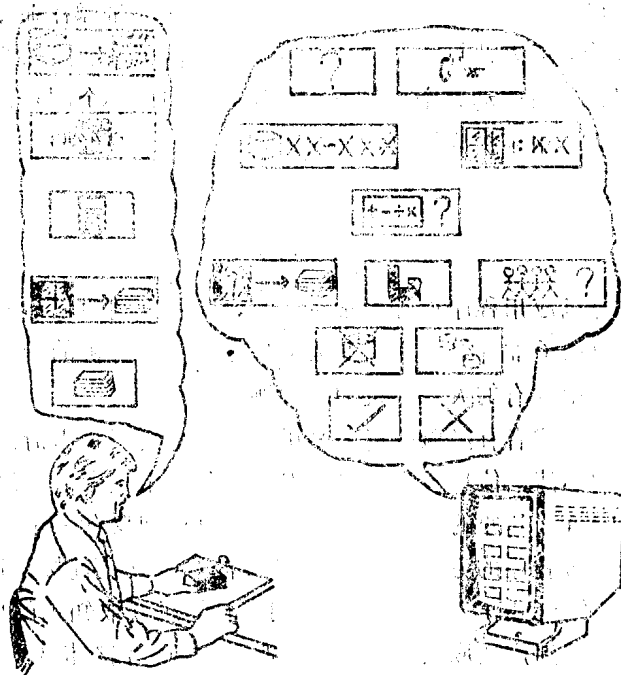


图1 图符通讯

用户

取文件
在窗口中
系统菜单
送往打印机
打印

计算机

询问?
口令?
哪个文件?
哪个窗口?
什么计算?
把什么送往打印机?
需要多少份?
打印多少份?
取消或装载结果文件?
让可或是否指令?

通过Puck和老鼠来“定点和驱动”

早先的计算机程序中使用图符的经验表明,人们识别符号的速度比识别相应(编码)字符集的速度快30倍,定点比相应键盘输入快6倍,且图形响应比一般字符响应效果更好些。

后来人们决定用图符命令来驱动一个完整的计算机应用设计,概念上讲,这样的系统可以用一种图形语言来处理人机对话,图1就表示了一种典型的人——机信息交换。

早期选为实际研究的应用程序是ICL PERQ计算机上的电子数据表计算器,之所以选择这种应用程序,是因为所涉及的计算广为大家熟悉,各类非计算机职员都用过电子数据表计算器,且认为在找一个经验丰富的测验续者时,这些代理商会有帮助。

由于其它(非图形)计算机上运行许多同种程序,直接比较用户的工作可作为对此方法效率评价。

下面的几节讲述了我们开发的图形符号命令系统,并举出了一个简单例子,描述用户对该系统的反应,首先讨论了硬/软件环境。

2. 硬/软件环境

工作环境包括计算机硬件(由它提供“定点和启动”手段,基本窗口和滚动功能)以及被驱动的应用程序。

人们会觉得由于图符,应用程序接口之内在特性,现有的电子数据表程序不能用了,而且整个程序还得重写,讲述这种程序不是本文的目的,2、3段给出了单个图符的功能性的大致描述。

2.1 计算机硬件

以交互图形为重点的单用户工作站的出现促进了本试验的进行,同时又是本试验的限制条件。

所用机器有ICL PERQ计算机,包括:

——1兆字随机存贮器

——1024×708线的光栅屏幕

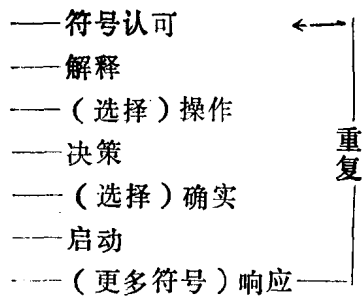
——键盘(ASCII字符码)

1024×708线的屏幕具有可寻址300,000个像素的好分辨率。处理器自动刷新(60次/秒),图形输入板带的三个按钮盘提供了硬/软件的按钮与屏光标的联系,出了“定点和启动”功能,机器本身的基本软件有很大的灵活性,可采用各种各样的光标设计。

2.2 “定点和启动”原理

本文所谈的“定点和启动”原理是闭控制循环的一部份,在该循环中,“定点和启动”以图形信息的形式触发从计算机来的响应,反之,该信息又刺激了用户下面来的定点和启动。

图形符号提出了人——机通信媒介,因此符号就成了一个链式的连接,其中包括



因此，用图形符号命令驱动的应用程序的基本原理建立在这么一个理论上“计算机显示的符号可由操作者来识别，对此，操作者不是做一个变换的选择，就是加进一些数据，反过来，这些选定的符号又触发执行相应动作的应用程序，在某些实例中，引入确实和检验信息的做法会减少给出错误命令的危险，图2就表示了这么一个序列。

如同一般情况那样，同时显示多个符号时，（来自计算机）的确实请求可用强度的关联符号来表示。当需要后继选择命令时，就以“闪烁”表示转换的符号，即黑白交替表示有效选择的符号——一旦作出选择，被选定的符号就呈现出认可决定的转换颜色。

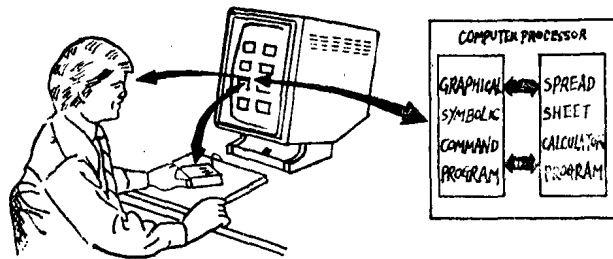


图2. “指示与启动”原理

2.3 应用程序

“电子数据表”计算器是选为试验的应用程序，因为在所有计算机应用程序中，电子数据表计算机应用最广，对本实验来讲，它还引起的面向非专业计算机人员的用户的兴趣，譬如主要事务并不需要大量使用计算机的商业科学用户，这类的专业人员或许对计算程序设计一无所知。

电子数据表是十分有名的程序，简而言之，就是包括安排在单个“格子”里的数据操作，这些“格子”排在“行”和“列”，“格子”之间存在着用户定义的算术关系。“电子数据表”之所以能普及推广，是因为它具有通用性，能解决各种问题，易懂易学。商业上取得成功的电子数据表必须有一定的复杂编辑和操作功能，利用现有程序特点，需要83种不同的图符命令，由此可看出复杂功能的范围。

应用程序ICL FERQ CALL的容量为250列，60行，其15,000个格子，格子大小可变，最大为128字符，可用各种数学表达式来描述单个格子间的关系。”

见PERQ CALL用户手册(2)

2.4 开窗与滚动

从屏幕相应尺寸, 及电子数据表最大尺寸可知, 不能同时显示所有的电子数据表。

另外, 为需要同时观察图符命令和数据, 屏幕得分成命令区和工作区, 工作区本身又可划分以显示电子数据表的不同部份。如图3所示, 在该例中, 有八十个分离的窗口, 两个用

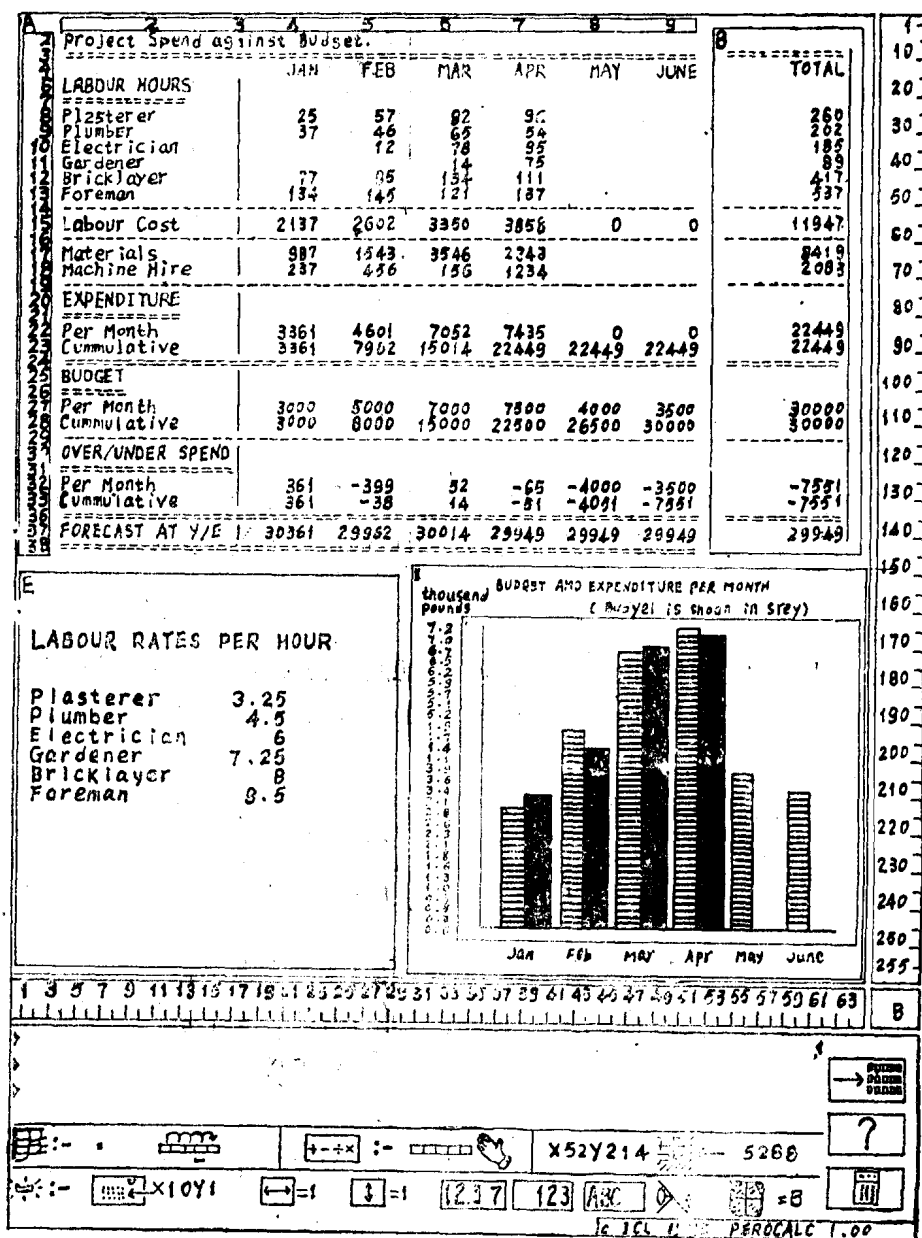


图3 屏面分成单独窗中的 PERQ CALL OUTPUT 实例

于滚动（水平和垂直），主表格，劳动率，直方图各一个，命令，数据录入区用两个窗口。

图3中，工作区内的格子由列，行序号来查找，在某一列（行）指定光标即可滚动这些格子，按下Puck键即可启动这些格子。图符在这类简单应用中用不上，在多个窗口齐步滚动时要用图符定标。

滚动图符如图4所示。箭头表示水平或垂直方向滚动，数字空格处表示要同步滚动行或列，举例中的○表示无同步滚动。

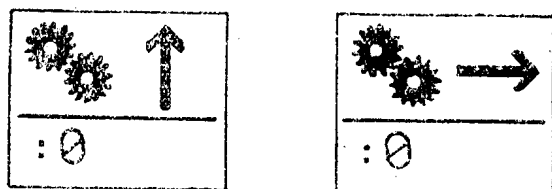


图4. 滚动多个窗口的符号

3. 图形符号命令

共有83个不同的图形命令，每一个都具有特定意义和精确含义，早在设计该系统时，我就认定直观识别与确定含义联用是至关重要的，因此在把每一个符号做成都是含义清楚方面作出了许多考虑。

图5表示了一些例子

在图5举例中，一见到钥匙（图5a）就使人联想起保密的概念“被保护”和“未保护概念一目了然，“+”号（曾解释过）显然是“加”的意思，容易记忆。

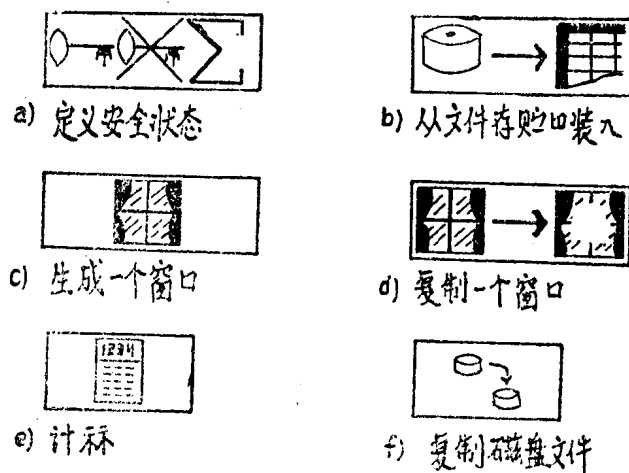


图5. 图符命令例子

在对话方式使用时，“从文件库取文件”（图5b）会询问“哪一个文件”和“哪一个窗口”（见图1）“生成一个窗口”（图5c）和“拷贝一个窗口”（图5d）含义明显，表示“计算”（图5c）和“拷贝盘文件”（图5f），这些功能对复制文件将换或文件保护时使用。

经验表明，即使有些符号开始不认识，一旦定义便会记住。

每个符号有两种使用方式

- 信息方式，表示选择或处理状态。
- 命令方式，启动时，由它预置一特殊程序功能。

3.1 命令结构

为显示那些所执行的功能有用的符号，83个图符在一个命令结构中呈主干型。

图6表示最初的选择，选定任何符号，原命令区的内容就会被一组序号较低的代表该选择对应的功能的符号刷新。

由图6可知，符号代表的功能可分成一簇簇的，且是与运行某一应用程序的对应的簇，即，

- 初始化
- 准备（如，清除工作区）
- 生成一个工作区（如，开窗）
- 生成一个工作表
- 定义保护级别
- 一般事务处理（取/存文件等）
- 计算
- 输出
- “帮助”文件等

“帮助”符号就是显示该符号（和其它相关信息）的含义，为驱动某个应用程序，有些符号可用来请求下面的符号以给出全部83个符号。

在PERQ CALL用户手册(2)中有关于这些符号和其详细使用说明的图解。

4. 用户体验

有些观察结果是用户在系统上完成的，这样就可判定系统的使用性和效率，人们觉得因熟练程序，知识和兴趣大小，再怎样严格控制测试用户使用系统的经验不合乎实际。

但是仍可提出如下三条标准

- i) 指定单个用户上机，根据他们自己进度熟悉系统，观察一段时间，就有底了。
- ii) 选定有经验的用户，给他们用这系统解决一个规范问题，且请他们分析在一般键盘驱动系统上同一问题的情况，还要测定相应次数，为比较逐个阶段的过程，可对这个方法录像
- iii) 请所有使用该系统的用户填上意见征询表，取得他们对该系统使用的反映，与测定的数据一起分析。