OSF
Motif
图形开发工具

微机接口与应用系列丛书

OSF/Motif

标准图形界面开发工具

参考手册

杨　华　周小东
黄　勇　赵　生　编写

宇航出版社

（上册）

微机接口与应用系列丛书

OSF/Motif 标准图形界面开发工具

——风格指南
——程序员指南

（上册）

杨　华　周小东
黄　勇　赵　生　编写

罗　坤　审校

学苑出版社
1993.

(京)新登字 151 号

内 容 提 要

本书是为 OSF/Motif 程序员而编写的，内容循序渐进，结构清晰，图文并茂，并附有大量编程实例。

欲购本书的用户，请直接与北京 8721 信箱联系，电话 2562329，邮码 100080。

微机接口与应用系列丛书

OSF/Motif 标准图形界面开发工具(上)

编 [illegible]：杨 华 周小东 黄 勇 赵 生
审 校：李 坤
责任编辑：甄国宪
出版发行：学苑出版社 邮政编码：100032
社 址：北京市西城区成方街 33 号
印 刷：长空印刷厂
开 本：787×1092 1/16
印 张：16.625 字数：848 千字
印 数：1～3000 册
版 次：1993 年 11 月北京第 1 版第 1 次
ISBN 7-5077-0803-9/TP・14
本册定价：95.00 元(2 册/套)

学苑版图书印、装错误可随时退换

前　言

OSF/Motif widget 系统是开放软件基金会推荐使用的标准图形界面开发工具。它建立在基于 X Windows 系统的 Xt Intrinsics widget 之上，故它能方便和快速地访问 X Window 系统的低层函数，并支持用户定义的或扩充的 widget，允许用户混合使用 Motif，Xt Intrinsics，X Window 的过程和函数。使用 OSF/Motif widget 系统用户能灵活而简炼地构造复杂的应用程序。

由于 X Window 只提供机制，而不提供方法，故仅使用 X Window 构造程序时，需进行繁琐的设置工作。OSF/Motif 不仅提供机制，而且提供方法，用户只需提供极少的外观控制参数即可完成应用程序的设计，这样大大简化了应用程序的开发工作。如果说应用程序是一台机器，则 X Window 为构造该机器提供原材料，Xt Intrinsics 提供零件，而 Motif 则提供部件。

使用 OSF/Motif 设计的菜单界面非常美观。OSF/Motif 拥有充分的缺省资源，缺省资源的设置值以大多数人的审美为基础，符合美学的原理。

OSF/Motif 缺省资源可以由用户设置的资源代替，故用户可以灵活地控制 Motif 界面的外观。

OSF/Motif 菜单调用机制由系统控制，呈树状管理，故响应迅速。

凡支持 X Window 的硬平台都支持 OSF/Motif，这些硬件平台包括 386、486 微机及工作站。

本套书为 OSF/Motif 程序员而编写，内容循序渐进、结构清晰、图文并茂并含有大量编程实例，是设计 Motif 用户界面的理想工具。

本套书分为上下册，内容包括 Motif 风格指南、程序员指南和参考手册

参加本书编写的还有徐青、柳文德、钟佳惠、吴玉奇、尹文强、赵枫、刘文品、叶佳琪，在此一并感谢。

目　录

第0章 引 言

OSF / Motif 风格指南提供一种规格说明框架以指导应用程序开发人员、Widget 开发人员、用户界面系统开发人员以及窗口管理开发人员设计并实施与 OSF / Motif 用户界面相一致的新产品．这份风格指南也严格地与 Microsoft Windows，表示管理器以及公共用户访问（CUA）保持一致．

通过从各种当前的行为模式中抽取公共元素，该风格指南可在新产品中建立起一致的行为．这本风格指南期待着把图形用户界面演化为可用的新技术，也希望把图形用户界面演化为 Motif 用户界面的用法．在内格指南变为稳定可靠之前，要化额外的时间补充它们．

有关把这些实施思想编写为应用程序、Widget 或者窗口管理程序的细节请参阅 OSF / Motif 文档集中的其它卷．

读者

这本书是为四种读者而写的．下边向每一类读者推荐这本指南中他应读的有关章节．我们建议通读整个"风格指南"以熟悉所有用户界面设计概念并保证没有漏掉什么内容．

- 应用程序设计人员：
  应该熟悉第一、第四以及第六章的内容。
- 工具（Widget）设计人员：
  应该熟悉第一、第二、第三、第四、第六、第七章的内容．
- 用户界面系统设计人员：
  应该熟悉本指南的全部内容．
- 窗口管理程序设计人员：
  应该熟悉第一、第四、第五、第六章的内容．

本指南包括的内容

本指南分为七章及二个附录．

- 第一章描述一般用户界面设计原则．每一个人都应该读这一章。
- 第二章描述输入及导向模式．新Widget设计人员以及用户界面设计人员都应该读本章．
- 第三章描述选择和活动模式．新Widget设计人员和用户界面系统设计人员都应该读这一章．
- 第四章描述用户界面部件选择、布局以及交互作用．每一个人都应读这一章．
- 第五章描述窗口管理程序设计．窗口管理程序设计人员以及用户界面系统设计人员应该读阅这一章．

- 第六章介绍并简要描述与用户界面设计有关的国际性和地方性概念和问题.
- 第七章为前边几章（前六章）描述的概念提供一些参考信息，第七章还提供与部件有关的详细信息．新 Widget 设计人员应阅读本章，本章也可以作为每一个人的参考信息.
- 附录A给出省缺键盘和鼠标连接.
- 附录B说明本"风格指南"所描述的OSF／Motif工具（Widget）和部件之间的对应关系.

第一章　用户界面设计原则

1.1　简介

用户界面是应用程序和应用程序的用户之间的接口．用户界面的主要目标就是使用户有效、满意地使用应用程序．遵循这本"风格指南"所提出的指导可以建立性能好、使用方便的应用程序，这些指导性信息适合于所有种类的应用程序，从电子数据报表、字处理程序到 CAD 工具无所不能．

为了有效地工作，用户界面允许用户简单、自然地与应用程序进行交互工作．成功的用户界面设计人员在应用程序的设计中应时刻把用户牢记心中．为了把用户记在心中，遵循以下两条原则：

- 了解用户
- 给用户以权力

应用程序的用户感到最主要的就是把工作干好．所以就应该把用户界面设计成使用户能迅速、容易地完成任务．用户想要控制应用程序（精通应用程序）．为了使用户掌握应用程序的基本要点，就要使应用程序尽可能简单．同时，还可包含一些和应用程序交互工作的先进方法，一些便于用户使用的捷径．一些用户又好奇又好探索，当这些用户使用并掌握了应用程序时，也就找到了捷径．这些捷径并不必象常规的交互方式那样直观．

这章讨论以下一些指导信息，遵照它就能产生一致而又易用的用户界面．由于用户应用程序、部件、窗口管理程序、用户响应等的特殊性，并不是一切都必须应用所有这些原则．

- 采纳用户的观点
- 让用户控制
- 使用真实世界中的一些比喻
- 使界面保持自然
- 命名界面保持一致（兼容）
- 让应用程序动作和用户通讯
- 避免普遍犯的设计错误

1.2　采纳用户的观点

有效的设计工作是从采纳用户的观点开始的，这点通常是很难做到的．应用程序设计人员想把应用程序看成一些功能的实现．而用户是根据应用程序的界面来看待应用程序的．

好的程序设计就是要以了解用户的工作为基础．设计得很好的应用程序可以解决用户的问题，使用户的工作易于完成，并经常向用户提供一些新的处理问题的能力．了解用户的工作的两个最有效的方法就是在设计工作中把用户联系进去，再一个就是使自己充当用户。

来自用户方面的信息不但可以帮助决定适当的功能，还可以帮助决定表示这些功能使用的方法．在设计的过程中，应尽可能早地把用户涉及进去，这是因为当设计工作有了很大的进展且调度安排结束时，改变设计的可能性就减少了。

不需要涉及用户的工作原型．事实上，在写规则说明时，甚至就可以涉及用户的问题了．在这一步，可以观察一下用户的工作，以便于了解应用程序使用的环境．同这些用户谈一谈他们的工作，谈一谈他们当前使用的工具以及他们对新工具的要求目标．

例如，如果你打算设计一种软件来产生并显示会议期间要使用的图表和图形，这时，你就可能要参加各种用户现场会议，看一看当前在会议中如何使用图表、图形，可能还要访问一下会议的参加者以知道他们在新的工具中想看见什么．一旦有了应用程序的工作原型，就邀请用户测试这个原型以检查一下你设计的界面是否满足为原型建立的目标．

在实际的现场中试用应用程序．使用应用程序的人可以提供一些指示用户界面问题的批评意见．获得使用应用程序的经验是很困难的，也很浪费时间，但很值得去做．在为应用程序建立界面之前，使用一下类似的应用程序，甚至使用一些竞争产品以帮助你懂得用户的任务．

1.3 让用户控制

用户想要控制，也需要控制他们所使用的工具以完成他们的工作．当应用程序灵活方便时，用户能控制这些工作并且逐步揭开其中的秘密．

1.3.1 使用户界面保持灵活

为用户提供访问应用程序功能及完成其任务的多种方法以增加用户控制的意识．界面的灵活性使用户能按自己的准则选择最好的办法去访问他们所需的功能．这些准则可以是经验水平、个人爱好、特定场合或者简单的习惯．例如，用户可以通过下拉一个菜单访问一种功能，直接操作对象，按下一个助记键或者控制键盘加速程序来访问某功能。

应用程序也应该是可配置的．允许用户作一些配置设置，允许用户选用个人的爱好以增加他们控制应用程序的意识，鼓励用户在了解你的产品和懂得产品如何工作方面发挥积极的作用．为了有效地工作，应用程序的可配置性要求易于存取访问。

1.3.2 使用顺序（逐渐）指示的特点

要把应用程序设计成首先出现必须和普遍使用的功能并按逻辑次序出现．使得较复杂和不经常使用的功能不易察觉，但仍然可以使用．例如，用对话框（DialogBox）隐藏一些不经常访问的设置。

决定应用程序功能的布局并不容易．从实现的观点看来，所有功能都很重要．然而，经常是相当少的功能被大量使用．要保证这些重要的功能成为界面表示的主要特征．要记

住，只有把其它功能隐藏起来时，这些重要功能才突出地表现出来。

1.4 使用真实世界中的一些比喻

好的用户界面允许用户使用真实世界中经历过的技艺。例如，按下控制按钮(PushButton)以及滑动比例尺。这些例子使用户很容易推理如何使用用户程序。当设计一个新的程序部件时，考虑如何把该比喻合并到新的程序部件中去的。同样可以把真实世界中的这些比喻扩大到程序部件组中去（特别是当建立代替机械用户界面的基于计算机的用户界面时）。

1.4.1 允许直接操作

用户要求能直接操作用户界面及应用程序的元素。例如，用户要求能直接用ScrollBar（滚动条）滚动文本，而不是用键盘发出的命令去滚动文本。直接操作模拟真实世界，在这里用户借用一些工具完成物理对象中的任务。用户通过直接操作类似于真实世界控制的图形部件控制应用程序，而不是在命令行中输入命令控制应用程序。直接进行操作可以减少用户需要记忆的信息量。

直接操作可以把某个动作连接到从某个部件可观察到的响应中。如果使用直接操作，用户就可立即看到每一个动作的结果。

直接操作模式是一种对象动作模式。也就是说，首先选择一个对象或者对象组，然后在所选择的对象上完成一种动作。对象动作模式可以使用户看到在完成某个动作之前，对象上什么元素在起作用。也允许在所选用的元素上连续完成多个动作。

尽管在应用程序中允许直接对对象进行操作是非常重要的，但也必须提供一些方法以使得只使用键盘的用户与你的应用程序交互工作。一些高级的用户可以使用这些方法以便更快完成某些任务。

1.4.2 提供快速响应

使应用程序尽可能快地对输入作出响应。立即得到可见的响应对进行直接操作是非常重要的。当使用程序部件时，立即提供应用程序的响应以及相称的部件动作。延迟、不相称的响应、或者不一致的响应都会使本可设计得很好的应用程序不适用。因为性能问题使用户难于集中精力完成手头的任务。

1.4.3 象输入那样提供输出

直接操作的另一个特点就是把应用程序一部分输出或者应用程序自己的输出也作为输入使用。例如，如果三个动作产生了一个文件名表，另一个动作就可以选择使用它们。

用户通过对对象进行定位并恰好与这些对象吻合去操作对象而不是输入对象的名字对对象进行操作。要把应用程序设计成只有不存在一个对象时，用户才需要输入对象的名字。设计得很好的应用程序会大大减少用户完成任务需要记忆的信息量。

1.5 使界面保持自然

可以扩大让用户自行控制和使用真实世界比喻去安排应用程序的概念以使任务自然地进行；如果实施了这种策略，用户就能更快地完成任务.

每一种屏幕对象都必须有一种清晰明了的样子以使用户容易认识、便于理解．同时，界面的形式在图形上要统一;要保证使用任何分辨率的屏幕、界面都保持兼容一致且有一个吸引人的外观.

1.5.1 使导向容易

通过提供整个工作区域的直接表示以及在该工作区域中移动的机制使得导向比较容易、方便、迅速地在工作区域内移动给用户一种驾驶应用程序的感觉．例如，ScrollBar（滚动条）就是一种最有效的方法，它可以指示相对于整个区域的当前画面位置．ScrollBar 除了可以提供位置的反馈信息之外，它还可以使用户在整个区域内移来移去.

按照元素的用处把它们安排在屏幕的不同位置上，一种优化的安排屏幕的方法能帮助用户决定要处理的工作,减少可能出现的错误．按照用途安排屏幕的最好方法就是让用户参与安排屏幕的过程，屏幕对象的安排要简单且有条理.

减少鼠标移动以简化用户的动作．例如，把次级 DialogBox（对话框）放在其父 DialogBOx 附近，以使得当第二 DialogBox 出现时，鼠标光标在缺省的 PushButton 上，但要除去用户需要看原 DialogBox 中内容的情况．减少鼠标移动可以使界面自然，这是由于从用户的观点来看，工作是由思想、意图和任务(有些任务是预定的,有些则在工作过程中在明朗化)构成的,而这些思想、意图和任务都与所需结果相关．当用户必须作一些不必要的鼠标移动，打开以及关闭 DialogBox（对话框）或者寻找命令时，其思路就被打断了.

1.5.2 提供自然的色调和颜色

为了引导用户的注意力就要减少屏幕对象之间的对比度．适当使用对比度可以帮助用户按窗口的背景区分屏幕对象．亮背景上很黑的屏幕对象，黑背景上很亮的对象，所有命令具有鲜艳的颜色都会引起用户的注意力．如果屏幕上很多对象都有强烈的对比或者鲜明的颜色，用户就很难知道先看什么，这是因为所有这些对象在引起人们注意力方面是完全相同的.

把颜色作为使界面丰富的一个方面，也就是说，用它使屏幕对象具有另外的区别．当然，这些区别还有屏幕对象的形状和大小．例如，在世界上很多地方，停止符号都是红色的八边形，所以就可通过其颜色、形状识别停止符号.

1.6 让界面保持一致

本《风格指南》的主要目的就是要保证界面的一致性．一致性对于各应用程序之间及单个应用程序内部都很重要．一致性帮助用户把所熟悉的技能应用到新的场合中．用户可

以把从一个应用程序学到的知识应用到其它的应用程序，这样就可以减少他们需要学习的信息量和以后要重新回忆的信息量．应用程序中的一致性有利于探索一些新的功能．这些新的功能看起来会很熟悉、舒服且很合适．本《风格指南》中的指导信息能使你的应用程序和市场中的其它应用程序完全一致（兼容），使你的应用程序在市场上获得成功．

应用程序内部保持一致包含以下几个方面:

- 类似的部件操作类似且有类似的用途．
  例如，由于PullDown（下拉）、Popup（弹出）、Option Menu（选择菜单）有相同的部件，它们的操作和用途也应该是相同的．在 4.1 节中描述有关选择适当部件的问题．部件交互在每一个部件的参考部分及 4.3 节中描述．
- 相同的动作应该得到相同的结果．
  例如，推ScrollBar（滚动条）中的顶部箭头总应该使ScrollBar上移．交互在4.3节中描述．
- 不应该因环境而改变部件的功能．
  例如，按动按钮总应该完成相同的动作．请注意，尽管动作相同，但动作的结果可以与环境有关．文件编辑器中的按钮可以开始编辑许多文件中的一个文件．该按钮无需总是编辑相同的文件，而是相同的动作会编辑所选择的文件．交互工作在 4.3 节中描述．
- 部件的位置不应随环境而改变．
  一般不应增加或者删除部件的内容．否则,就很难找到所需要的功能．为此，可使不需要的部件失去功能，通过不强调不需要的部件的标签而指示它们（不需要的部件）．在 4.2 节描述部件布局．
- 鼠标光标的位置不应随意改动．
  鼠标光标的位置应该通过直接操作决定而不应该由应用程序任意定位．由应用程序对鼠标光标进行定位会使用户失去对光标的跟踪．随意改动光标也会给图形输入板指点设备带来一些问题（使该设备依赖于绝对光标定位）．第二章描述了输入模式．

应用程序中的一致性增加用户的控制意识．一个应用程序的经验容易用到另外一个应用程序中，搞好手中的任务而不是学习新的应用程序变成了计算机会话的焦点．当应用程序用一种和其它应用程序一致的方式工作时，用户就会立即对他们学会新的应用程序的能力感到有了信心．当用户试用新的功能时，由于这些功能尽管很新，但对他们很熟悉，他们就会格外高兴．

应用程序内部一致还包含以下几点:

- 部件看起来很熟悉．
  这并不意味着这些部件看起来完全相同，但这些部件的布局却是相同的．元素的外观，例如颜色、尺寸、衬边的厚度对应用程序的内部可操作性是无关紧要的．每一个部件的设计及布局在参考部分及 4.4 节中描述．
- 交互工作的方式令人熟悉．

当在应用程序中使用不同的交互工作方式时，就会把用户搞糊涂，同时用户要集中精力完成应用任务也是很困难的．这一点也适合于部件行为、输入方法、选择模式和键盘导向．交互工作方式在 4.3 节中描述．

- 以熟悉的方式组织部件．

  用户要能够迅速地为每一个任务找到适当的部件．一致性指南信息对于用户组织部件很有帮助．4.2 节描述应用程序布局．

1.7 让应用程序动作和用户通讯

有效的应用程序可以让用户知道应用程序中发生了什么事，但却无须知道完成这些事情的细节．用户和应用程序之间适当进行通讯增加了用户的满意感．应用程序和用户进行通讯有三条原则：提供反馈信息,预见错误，并提供一些警告信息．

1.7.1 给用户一些反馈信息

反馈信息使用户知道计算机已经接收到了他们输入的信息．一旦用户以某种方法使部件或者菜单项醒目显示而选中了某部件或者菜单项时，就向用户提供反馈信息．另外，如果某些操作要花好几秒钟才能完成，就应该让用户知道计算机正在进行这种操作，通过提供一条消息或把光标变为工作光标就可以达到通知用户的目的．

1.7.2 预见错误

预见可能会发生的那些错误．通过预见一些错误，就可在程序设计中避免这些错误，使程序支持出错恢复，并提供一些消息通知用户采用适当的纠正动作．例如，避免过多错误信息的一种方法就是当不能使用这些错误信息时使界面部件变暗淡．与环境相关的邦助信息,有助于用户理解应用程序,减少错误和恢复错误．邦助信息的正文要使用日常用语，简洁，明了．邦助信息应能够随时获得和删除．

用户在学习如何使用软件时，如果使用自然、错误很小的方法就感到非常舒服．取消(undo)功能支持通过细微的错误及减少错误的代价进行学习．取消功能可以使用户重新跟踪以前的动作，即发扬了探索精神，也积累了经验．

1.7.3 使用显式破坏动作

显式破坏意味着，当某个动作有不可改变的消极后果时，就应该要求用户采取显式的动作去完成这个动作．例如，在 SavePushButton 上接动鼠标按钮就可以保存一个工作单，但如果要消除一个工作单时，不但要在 ErasePushButton 上接动鼠标按钮，而且还要回答诸如"Arc you surc you want to crase this worksheet? (能保证要清除这个工作单吗?)"这样的问题．

警告信息可避勉由于用户不当心造成的破坏性操作，从而允许这些操作存在于应用程序的控制之中．警告信息也鼓励用户大量实验而不要害怕失误．可以引起严重错误或者不可恢复性丢失数据的操作应该向用户警告后果,并要求用户显式地给于肯定．

1.8 避免普遍存在的设计错误

获得好的设计过程提出了很多挑战性问题，同时也出现了一些可能的错误。以下的指导信息可帮助你避免犯普遍的错误。

·注意细节

应用程序的细节反映了你在该程序中使用的技巧。一个精心设计的界面的各个细节都能方便用户的工作，使他们满意。例如，把两个相关并搭接的对话框对齐，使用户易于在一个连续的动作中激活新的设置。使菜单项和对话框标签的大写字母一致化，是减少用户在文字上的分心的一个设计细节。

·不要过早结束设计

常见的一个设计错误是，过早地假设某项设计已结束。在工作安排紧张及某项设计的不足部分难以确定时，这种倾向会加剧。要尽早开始设计，并且要使重新设计成为可能。应用程序的第一次设计不是最终结果，而是可以用来察看界面设计问题的真实模型。

·叠代设计

界面设计最好叠代进行。实现，返馈，评估和修改的开发循环可以提早发现和修改无效的设计，从而避勉错误。

·从真实模式型始

不要企图把现有软件移植到一新形式的界面下来转换软件。因为直接操作，需要进行广泛的考虑和检查。需要重新设计功能的层次结构和表示。

·隐藏实现细节

用户界面要隐藏内部的软件，而给用户提供一个一致的界面。一个好的用户界面不允许用户看到应用程序的实现细节，使用户不必关心应用程序的内部机制。

第二章　输入和导向模式

2.1　简介

各模式之间的一致性能便于用户对系统的控制，尤其在系统中或各应用程序之间使用了一致的模式后，人们更会深刻地感觉到这一点．本章描述是 OSF／Motif 用于在窗口和各部件之间进行移动以便与这些部件进行交互的模式．

- 键盘聚焦模式，能够确定屏幕中哪一部分接受键盘事件．
- 输入设备模式，它描述了不同类型的设备（如键盘和鼠标）怎样与应用程序进行交互．
- 导向模式，它确定怎样将键盘焦点在各部件之间移动．

对于系统和应用程序的一致性来说，激活（activation）和选择（selection）模式也是很重要的，它们将在第三章进行详细讨论．

2.2　键盘聚焦模式

典型的工作空间可以包含许多窗口，其中每一个窗口都要从键盘（或鼠标或二者兼而有之）中接受输入，收到键盘事件的窗口就拥有输入焦点．实际上，当键盘输入被定向到窗口中时，它实际上是由该窗口中的某些控制机制接受了．键盘聚焦模式能够确定工作空间中的哪个窗口，窗口中的哪一部分接受每一键盘输入．键盘聚焦有时侯指的是输入聚焦．

具有键盘焦点的窗口必须以某种方式醒目显示，通常是改变阴影或窗口边界的颜色．具有键盘焦点的部件也必须用位置光标（location cursor）进行醒目显示．位置光标通常是在键盘聚焦的那部分周围画一方框，但也有其它一些方法，如将列表元素加轮廓或改变被画区域的背景等．位置光标将在第 2.4.2.1 节详细讨论．

为避免冲突，窗口管理程序必须使得在同一时刻只有一个窗口拥有键盘焦点．每一个应用程序窗口在同一时刻也只能有一个部件拥有键盘焦点．

键盘聚焦模式是由聚焦策略（focus policy）定义的，聚焦策略是一特殊机制，它能在窗口及各部件之间移动焦点．本节仅讨论聚焦策略，至于它对窗口管理程序、应用程序及各部件的影响将在遇到时详细讨论．仅从这一点上，我们就能够强调：窗口管理程序、应用程序和新的部件都必须同时支持隐式和显式聚焦策略．第五章讨论的是怎样使用隐式聚焦策略和显式聚焦策略在窗口之间移动焦点．

2.2.1　隐式聚焦

在隐式聚焦策略中，键盘聚焦位于用户移动鼠标光标的那个窗口中，在隐式聚焦模式