交互设计语言 INTERACTION DESIGN LANGUAGE

Designing the Beauty of Interactions

下册

与万物对话的艺术

罗涛 著

清華大學出版社

交互设计语言——与万物对话的艺术

罗涛 著

清华大学出版社
北 京

内 容 简 介

交互设计语言是一门将复杂抽象的交互逻辑，用视觉化的形式准确呈现出来的语言。它有助于交互设计师思考丰富的交互形式和细节，并与其他设计师和前端开发人员有效地进行沟通。

本书适合交互设计从业人员、软件前端开发人员、产品经理阅读，也可供交互设计相关专业的学生、教师以及任何对交互设计感兴趣的读者学习参考。

图书在版编目(CIP)数据

交互设计语言 : 与万物对话的艺术 / 罗涛著. — 北京：清华大学出版社，2018
ISBN 978-7-302-51533-3

Ⅰ. ①交… Ⅱ. ①罗… Ⅲ. ①人机界面—程序设计 Ⅳ. ①TP311.1

中国版本图书馆 CIP 数据核字(2018) 第 252955 号

责任编辑： 张 敏
封面设计： 朱子健
责任校对： 胡伟民
责任印制： 丛怀宇

出版发行： 清华大学出版社
网 址：http://www.tup.com.cn，http://www.wqbook.com
地 址：北京清华大学学研大厦 A 座 邮 编：100084
社 总 机：010-62770175 邮 购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者： 北京亿浓世纪彩色印刷有限公司
经 销： 全国新华书店
开 本： 170mm×230mm **印 张：** 29 **字 数：** 372 千字
版 次： 2018 年 12 月第 1 版 **印 次：** 2018 年 12 月第 1 次印刷
定 价： 129.00 元（上、下册）

产品编号：077871-01

目

录

第5章 符　　号

第6章 框 架

第 7 章 实 践

第 8 章 终章 设计魔法

第5章 符号

所谓符号是相对于某人，在某个方面能代替（代表、表现）其他事物的某种东西。

——皮尔士

本章将要细致阐述的交互设计语言符号体系，是描述“操控力”的一系列符号和它们之间的组合规则。我们先讨论核心概念及其特征，然后通过四个小节阐述符号体系的不同部分。每个符号我们都会用日常生活的实际案例进行说明。但由于交互设计语言将交互的结构拆分得比较细致，因此从理论上会提出一些实际产品中没有的交互方式。对于没有实际案例的情况，我们专门制作了交互式原型供参考。

任何一门语言都是一个符号体系，不同语言中使用的符号有很大区别。符号体系是由一系列的符号和符号之间的组合规则构成的。每一个符号都有自己的含义，代表了某种意义。通过这个系统中各个符号的排列组合，我们可以表达各种不同的意思。原则上来说，一个符号可以代表不同于它本身的任何东西。例如“!”这个符号，在自然语言中表示“感叹”的情绪，而在编程语言中表示“否定”（“==”表示“等于”，“!==”表示“不等于”）。我们也完全可以约定用“!”符号表示“天花板漏水了”，毕竟它的形状看上去也像水滴从高处落下来的样子。知道这个约定的人再次看到“!”，就能够将它和“天花板漏水了”联系起来。

符号与含义之间的匹配关系是一种约定俗成的规范，并不一定需要有客观上的联系或者逻辑上的支撑。单一符号本身也许并没有什么价值，因为它能够指代的东西并没有限制。一个符号的价值体现在它与同一体系中其他符号的差别，当所有不同的符号作为一个整体时，才共同构成了符号体系的意义。这些符号组成的整体，代表了被这门语言所描述事物的方方面面，也体现了这门语言所独有的视角。

正如其他符号体系一样，替换掉交互设计语言中的某个符号并没有意义。每个符号都代表了“操控力”中或大或小的某个概念，这些符号和它们的组合规则作为一个整体，描述了“操控力”视角下的交互结构。这些符号并没有受限于某个技术平台和设备，而是在“操控力”的视角下对交互方式的抽象规律进行描述。通过了解这些符号，你可以实现常见的交互效果，也会发现一些未曾深入思考的交互方式和细节。这些符号的组合规则鼓励人们尝试新的组合方式，而每一种组合代表了一种独特的交互方式。因此我们经常会发现可能未曾见过，但理论上确实存在并且可以被实现的交互方式。作为约定俗成的规范，交互设计语言的符号体系借鉴了一些为人所熟知的符号，并尽量形象化地表示其含义。

5.1 核心概念

5.1.1 联动关系

为了精确地描述“操控力”这个抽象的概念，我们首先需要将涉及的对象确定下来。既然我们讨论的是“操控力”，那么就有操控的主体、被操控的客体和它们之间的操控关系。在交互设计语言中，我们称之为“控件(Control Item)”“物件(Widget Item)”和“联动(Mapping)”。“控件”和“物件”本质上都是一个物件（Item），是由物质、能量和信息组成的（在本书中我们用带引号的“物件”表示“操控力”中被操控的客体，用不带引号的物件表示任何一个物件）。物件可以是我们的指尖、一支笔、一节电池、一个图标、一个滚动条、邮件列表，或者列表中的一个条目等。

用“控件”和“物件”进行区分是为了表示它们之间主动与被动的角色差异。在一个操控关系中，被操控的“物件”可以在下一个操控关系中成为主动的“控件”。鼠标控制光标，然后光标控制图标在桌面上的位置，这个过程就是一个很好的例子，参见图 5-1。在不同的操控关系中，光标可以是“物件”也可以是“控件”。而我们的身体，尤其是双手通常是“控件”，毕竟是我们在操控某个对象。

图 5-1 鼠标、光标、图标之间的控制关系

“联动”是描述“操控力”的关键，其中的内容描述了“控件”和“物件”之间的操控关系。为了更好地理解“联动”，我们可以将其看作数学中的函数 $y=f(x)$。“控件”是 x，“物件”是 y。或者将其看作 Origami Studio 中不同模块之间的连线，允许左侧的模块控制右侧的模块。再或者更形象地将其看作自行车脚踏板轴心上的齿轮和后轮齿轮之间的链条，将脚的力传递到后轮上，参见图 5-2。这些比喻虽然都不能完全准确地表达“联动”的含义，但可以让我们从不同的侧面对“联动”有一个形象的认识。稍后会详细地阐述“联动”的细节。

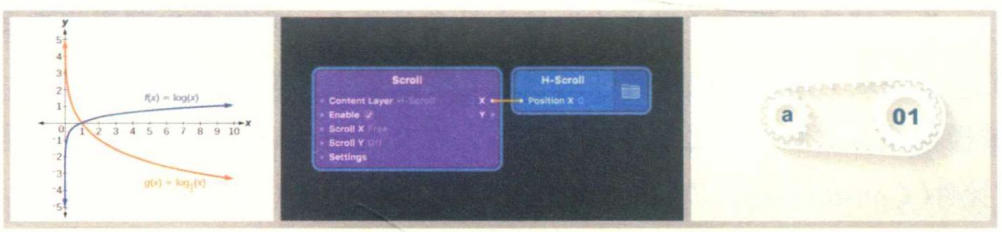

图 5-2 数学函数、Origami Studio 模块间的连线、齿轮联动

“控件 - 联动 - 物件”是交互设计语言中描述“操控力”的基本框架，我们称之为“联动关系”。**当我们讨论一个交互的时候，要首先明确“控件”和“物件”是什么，然后才能准确地描述它们之间的“联动”，从而清**

晰地阐述在这个交互中我们所拥有的“操控力”。直观上容易看成一个整体的交互方式，其中可能包含不止一个联动关系，只不过因为我们熟悉之后便不再注意其中的细节了。当用指尖操控智能手机屏幕上的列表时，指尖是“控件”而列表是“物件”，这里只有一个联动关系。但刚刚提到的鼠标控制光标、光标控制图标的交互方式中就有两个联动关系。而鼠标控制光标、光标控制滚动条、滚动条控制页面位置的交互方式中则有三个联动关系，参见图 5-3。

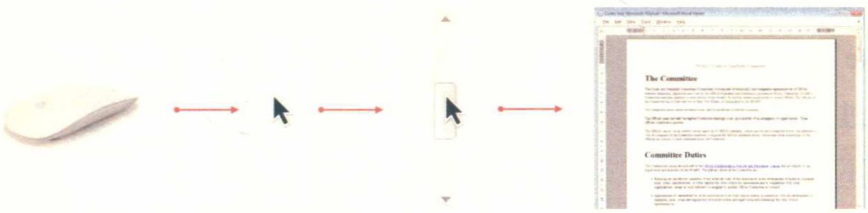

图 5-3 鼠标、光标、滚动条、页面之间的控制关系

在明确了“控件”和“物件”的情况下，我们可以开始分析它们之间“联动”的细节。首先，“控件”和“物件”作为“物”有着很多的属性，例如大小、旋转角度、位置、颜色、重量等，参见图 5-4。但并不是所有的属性都必须参与到某个联动关系中。例如在鼠标控制光标的联动关系中，是鼠标在桌面上的位置控制光标在屏幕上的位置。但是鼠标的旋转角度就对光标的位置没有什么影响，鼠标的位置对于光标的大小也没有什么影响（在 Windows 操作系统中）。而鼠标和光标各自还有颜色、透明度等属性，这些属性相互之间都没有控制与被控制的关系。

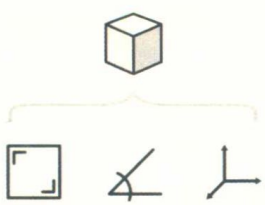

图 5-4 物件的各种属性，如大小、旋转角度、位置等

一个联动关系将“控件”的属性和“物件”的属性以某种方式对应起来，就可以理解为“控件”的属性在操控“物件”的属性。“控件”与“物件”之间的联动关系归根结底是属性与属性之间的对应关系。当两个属性之间有对应关系，可以通过改变“控件”的属性来改变“物件”的属性，我们就能感受到“操控力”。如果两个属性之间没有对应关系，则感受不到任何“操控力”。

一个交互方式中也可能同时涉及“控件”和“物件”的多个属性，从而提升我们对“物件”的“操控力”。“双球鼠标（two-ball mouse）”就不仅能像传统鼠标那样控制光标在屏幕上的位置，同时也能通过鼠标的旋转控制光标在屏幕上的旋转。现在的 Mac 机中，鼠标的位置不仅能够控制光标的位置，也能控制光标的大小这个属性。快速来回移动鼠标就能让光标暂时变大，目的是让用户在复杂的界面中快速找到光标的位置，参见图 5-5。

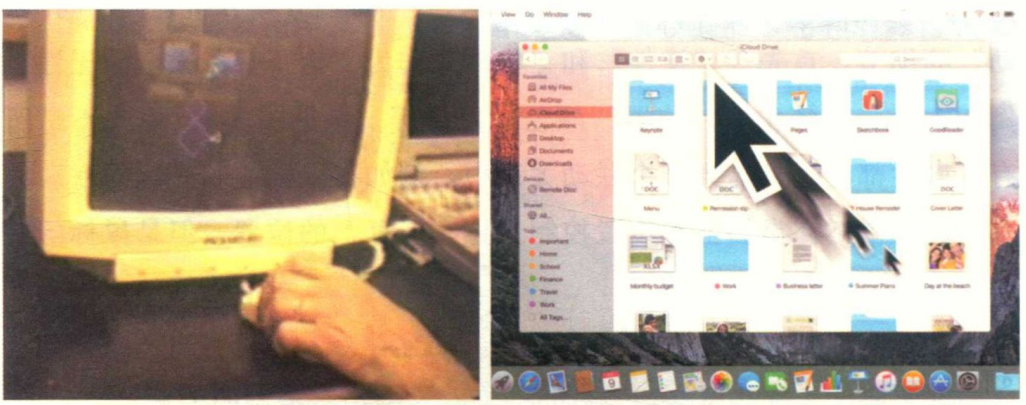

图 5-5　光标的旋转角度和大小也可以被操控

属性是客观存在的，某个属性不会因为没有被交互技术利用到就不存在。技术的进步可以帮助我们建立起更多属性之间的对应关系，让我们拥有更多的“操控力”。如果我们的设计思路只是局限在当前技术所支持的属性，那么可能会错过一些很有用、操作起来也很直观的属性。

以大家每天都要使用的智能手机为例，我们来分析一下手指这个“控件”在屏幕上都有哪些属性。

首先最直观的就是手指触碰到屏幕与否，以及手指在屏幕上的触碰位置。人们通过对这两个属性的控制，实现常用的“点击”“双击”“长按”“拖曳”“甩动”等操作。除了这两个属性以外，另一个很容易想到的属性是手指对屏幕的压力。从 iPhone 6s 开始，iPhone 开始利用手指对屏幕的压力这一属性，使得传统的触屏交互方式多了一个维度。人们可以通过对按压屏幕力度的控制实现不同的交互效果，参见图 5-6。

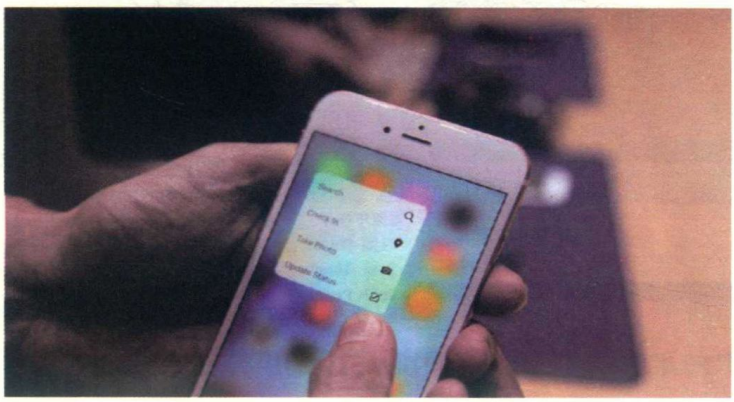

图 5-6 智能手机的使用场景

在这个常见的使用场景中，还有另一个很容易被忽略的属性，那就是手指与屏幕之间的距离。当我们的手指没有触碰屏幕但处于屏幕上方的时候，手指与屏幕之间的距离也是一个我们可以自由控制的属性。大部分手机并不支持这个功能，但并不意味着我们就不能在交互方式中利用这一属性。虽然还处在实验的阶段，但已经有不少人在探索如何利用距离这个属性创造新的交互方式。例如向上快速甩动一个列表之后，列表会向上滑动，这时手指和屏幕的距离越大，列表滑动的速度就会越快；手指和屏幕的距离越小，列表滑动的速度就会越慢，直到手指触碰到屏幕时速度降为零。三星的一些手机能分辨手指在屏幕上“悬浮”和“远

离”这样简单的距离差别，韩国科学技术院（KAIST）中就有人利用这个差别研究特殊的交互手势，参见图 5-7。虽然相关的技术还有待改善，但这说明在我们熟知的场景里，可能还有很多属性没有被我们利用。如果我们合理地利用各类属性并建立联动关系，就能够设计出直观且高效的交互方式。

图 5-7　韩国科学技术院利用距离的研究案例

5.1.2　“状态”与“状态变化”

仅仅确定了联动关系中相互关联的属性还不足以清晰地描述属性之间的对应关系。为了精确地描述属性之间的对应关系，就需要引入交互设计语言的另外两个核心元素：“状态（State）”和“状态变化（State-Change）”。

任何一个属性都是由一系列状态组合而成的，属性的某个状态是该属性中某个特定的值。例如光标在屏幕上的位置是光标的一个属性，光标在屏幕正中间的这一点就是位置属性的一个状态；音量是一个属性，

56% 的音量大小就是一个状态；透明度是一个属性，80% 的透明度就是一个状态；手指在屏幕上的压力是一个属性，手指触碰到屏幕但完全没压力就是一个状态，参见图 5-8。

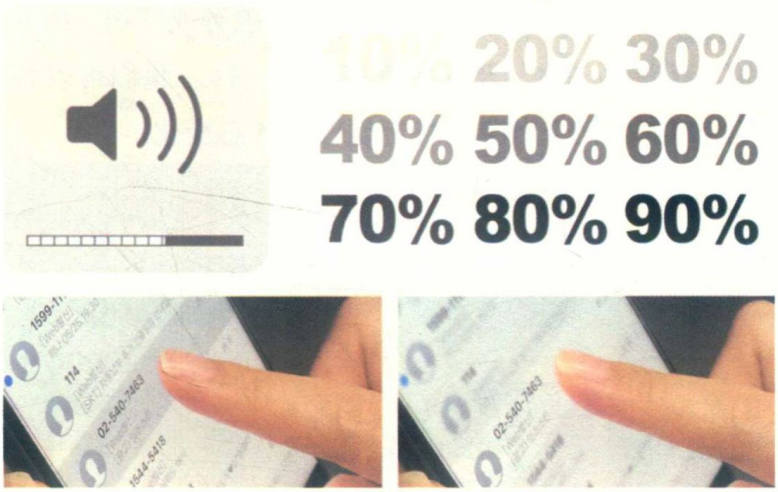

图 5-8 各种属性具体状态的案例

虽然不同的属性有很大差别，但是我们可以将任何一种属性都看成一组状态的集合。这样的抽象能够帮助我们讨论联动关系的本质和普遍规律，对交互规律的讨论因此不会被某个属性的具体特征干扰。同时，这也允许我们将两个看似毫无关系的技术平台中的交互规律联系起来，从而迅速地找到解决问题的办法（我们将会在第 7 章“实践”中看到具体的案例）。弄清楚联动关系的规律后，在进行具体设计时依然需要充分考虑属性的具体特征，尤其是对用户造成的影响。不过在讨论抽象的“操控力”和联动关系时，我们仅将属性看作一组状态的集合。

“状态变化”，顾名思义，就是状态所发生的变化。这个变化发生在一组状态集合允许的范围之内。我们所有的交互手势、操作和行为，都可以被理解为某个属性的状态变化（或某几个属性的状态变化组合）。例如，手指在屏幕上滑动是手指在屏幕上位置状态的变化，被滑动列表

的位置状态也发生了相应的变化；单击鼠标的行为是鼠标按键从抬起这个状态，变化到按下这个状态的过程，也是一个状态变化；单击窗口最小化图标后，窗口从现有的大小缩小到任务栏中图标的过程也是一个状态变化，参见图 5-9。任何的“事”都可以被理解成“物”的状态变化，不同的“事”牵涉的“物”及其状态变化不一样。我们可以将所有的操作行为都可以理解为某个对象的特定属性的状态变化。

图 5-9　Mac 系统中窗口缩小的状态变化过程

对于“状态变化”这个概念，一些理论和工具将其理解为“事件（Event）”。Alan Dix 早在 20 世纪 90 年代就探讨过用“状态（Status）”和“事件（Event）”描述交互系统的方式。Buxton 提出了“3-state model”，结合“事件”的概念一起解释鼠标的交互方式。原型工具 Framer 中更是明确地使用“状态”和“事件”作为制作交互原型的基本概念，并设置了专门的快捷按钮帮助用户输入相关代码，参见图 5-10。