Web前端 / 数据库 / 综合技术　（日）技术评论社 编

For All Web Application Developers　ウェブDBプレス

WEB+DB PRESS

WEB+DB PRESS 中文版 01

UI设计实践

提高用户满意度的
设计、实现和验证方法

使用D3.js [边做边学] 数据可视化

Web支付入门

PayPal、WebPay、iOS/Android
应用内支付的实现方法

Gradle

让Java的构建更高效

Serverspec

构建测试驱动基础设施架构

Fluented+FnordMetric

数据的收集和可视化

扁平化设计

摆脱拟物化隐喻表现的桎梏

人民邮电出版社
POSTS & TELECOM PRESS

寻人

图灵社区：iTuring.cn
热线：(010)51095186转600

分类建议 计算机/软件开发

人民邮电出版社网址：www.ptpress.com.cn

ISBN 978-7-115-38451-5

定价：20.00元

TURING

Web前端 / 数据库 / 综合技术　　（日）技术评论社 编

For All Web Application Developers　ウェブDBプレス

WEB+DB PRESS 中文版 01

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

WEB+DB PRESS中文版. 1 / 技术评论社编 ; 匿名译
. — 北京 : 人民邮电出版社, 2015.2
ISBN 978-7-115-38451-5

Ⅰ. ①W… Ⅱ. ①技… ②匿… Ⅲ. ①网页制作工具－
程序设计 Ⅳ. ①TP393.092

中国版本图书馆CIP数据核字(2015)第013458号

内 容 提 要

*WEB+DB PRESS*是日本主流的计算机技术杂志，旨在帮助程序员更实时、深入地了解前沿技术，扩大视野，提升技能。内容侧重于Web开发的相关技术。

本期的主题分为3个特辑：UI设计实践、Web支付入门和数据可视化。特辑1结合Cookpad网站详细介绍UI设计方面的实践知识。特辑2以在Web上使用最为广泛的信用卡支付为核心，介绍将信用卡支付整合进自己的网站或智能手机应用所必需的知识和方法。特辑3则介绍如何使用Web技术进行数据可视化。

本书适合各行业Web应用开发者阅读。

◆ 编　[日] 技术评论社
责任编辑　乐　馨
执行编辑　高宇涵
责任印制　杨林杰
装帧设计　broussaille私制
◆ 人民邮电出版社出版发行　北京市丰台区成寿寺路11号
邮编　100164　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京天宇星印刷厂印刷
◆ 开本：787×1092　1/16
印张：11.25
字数：303千字　2015年2月第1版
印数：1-3 500册　2015年2月北京第1次印刷
著作权合同登记号　图字：01-2014-0504号

定价：20.00元

读者服务热线：(010)51095186转600　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第0021号

WEB+DB PRESS

目　录
CONTENTS

分栏目录
CONTENTS BY TOPICS

编委点评 吴多益（百度）

看惯了中英文的技术文章，偶尔看看日文的杂志也挺有意思，既有许多相同之处，也有不少日本特色。

从编排结构上看，这本杂志由三个特辑和许多不同方向的专题文章组成，整体技术氛围比较强，但或许是为了考虑大部分初学者，很多文章都比较入门。

我们先从三个特辑开始介绍。

第一个特辑是“实践UI设计”，我在第一眼读到标题时并没有多少期待，因为这是一个很虚的话题，恐怕难以写好，不过在读完第一页后却意外地发现写得很不错。作者一开始就明确指出UI最重要的是帮助用户达到目标，而不是弄得多么绚丽，因此花了很大篇幅来谈产品设计与用户需求分析，几乎毫无保留地介绍了自己的设计流程、方法、工具及注意点，还有验证方法、内部前端UI框架等，整套流程做得非常专业，很值得借鉴，这也是整本杂志中我最喜欢的文章。

第二个特辑“Web支付”本来是个很有价值的话题，因为随着现在各种垂直电商网站和创业项目的增多，肯定有不少网站需要具备支付功能的，但可惜这篇文章谈的是日本的情况，和国内差距较大，所以只能当成技术文章看看了，实战价值不高。我曾经了解过一个收银台系统的实现，发现比想象中要复杂得多，一方面要和支付宝、网银、点卡、信用卡等系统进行对接，另一方面还要考虑各种促销活动导致的特殊逻辑。

第三个特辑的内容是基于D3实现数据可视化，其中的3篇文章分别介绍了可视化的3个应用场景——图表、地图和关系图。在内容丰富程度上做得不错，但D3独特的设计理念使得它的门槛很高，之前没学过D3的读者估计看不懂文中的代码，所以只能大概了解D3能做什么，后续还得花很多时间去学习。

说完了三个特辑，接下来谈谈连载的文章，这些文章的跨度很广，从前端、移动端到后端都有涉及，对于开拓视野很有帮助。

介绍Gradle的文章很简洁明了，基本上把常见功能都说清楚了，对于之前不了解Gradle的读者很有帮助。

介绍Android Studio的文章所用的软件版本比较老，是一年半以前的，这个工具在2014年12月9日已经发布了正式版。不过，本文主要介绍的是如何使用Gradle和IntelliJ IDEA，所以受版本影响不太大。

介绍serverspec的文章令我眼前一亮，这个工具我之前没听过，看作者名字是个日本人，它的特点是将服务器环境的搭建也纳入到单元测试中，比如搭建完后测试某个端口是否开启等，看起来还挺有用的，有了它就可以将环境部署脚本也纳入持续集成，从而保证质量。

介绍Coro的文章虽然花了很大篇幅来谈爬虫的实现，但其实还是远远不够，作者忽略了很多抓取中的重要技术，比如广度优先还是深度优先策略、网页更新策略、表单抓取、Ajax抓取等，所以看起来这篇文章的主要目的还是推广Coro这个协程框架。

介绍PHP中使用AOP的文章看起来很非主流，因为PHP本身并没有注解（annotation）的语法，所以就只能在注释中进行配置，看起来很奇怪，感觉还不如直接用Java。

在JavaScript专栏里介绍了Web Components，尽管这个技术推出有一段时间了，但国内似乎关注的不多。从我个人的使用体会来看，这个技术对于传统的前端开发是颠覆性的，可以说是目前最好的前端模块化方案，感兴趣的读者推荐读一下Polymer Designer的源码，体验一下完全不同的前端开发方式。不过这篇文章的写作时间较早，其实在2014年7月16日正式发布的Chrome 36中已经正式支持Object.observe和HTML Imports了，这使得Chrome在所有浏览器中最先全面支持Web Components。

在关系数据库的专题里介绍了图的基本概念，然后总结了几种数据库中存储图数据的优缺点，然而最终并没有很好地解决作者开始提出的几个问题。对于复杂的图算法，我个人觉得最好还是使用专门的数据库，比如Neo4j，它内置支持文中提到的Dijkstra等算法，所以实现起来很容易。

在Java专题里谈的是缓存，感觉这篇文章主要是介绍概念，实战价值不大，因为本文代码中实现的缓存太简单了，作者最后应该推荐一些真正的解决方案，比如简单的Guava cache，或者完善的Ehcache/JCS，以及分布式数据的Hazelcast/Redis等。

Ruby专栏介绍的是实时数据采集和展现，在这方面另一个比较流行的组合是Elasticsearch + Logstash + Kibana，我觉得它的功能更为强大，更适合用来分析日志。

在HTML5 canvas游戏开发经验中介绍了一个自己开发的库tofu.js，据说能解决一些Android机器下的性能问题，不过我到github看了一下发现只有38个Star，而且都一年多没更新了，所以看起来不靠谱。

最后，发现这本杂志里到处都有Ruby的影子，看来日本技术人员还真喜欢这个语言。

UI/UX未来志向

预测未来之走向，知晓当下之所需

第3回

明治大学　综合数理学部　先进媒体科学专业专职讲师
渡边惠太（WATANABE Keita）　译/王凤波
URL http://persistent.org/　mail watanabe@gmail.com　Twitter @100kw

扁平化设计

挣脱拟物化隐喻表现的桎梏

2013年6月，苹果公司宣布iSO 7的用户界面（UI）将改为采用扁平化设计（Flat Design）。

将现实世界真实物体的质感融入到用户界面中的视觉设计手法被称为拟物化（Skeuomorphism）。众所周知，iSO 6及其之前版本采用的都是拟物化设计。

而扁平化设计正如它的名称一样，平面的视觉印象是其最显著的特征。Windows 8就在iSO 7之前率先采用了这种扁平化设计的用户界面。

但是，到底为什么突然两个OS都采用了扁平化设计呢？由拟物化设计向扁平化设计的转变，并不像表面上看起来的那么简单，而是具有深远意义的，乃至需要我们不得不从电脑的本质上进行深入的分析和探讨。这一次，我们就从幕后的隐喻表现手法和电脑的本质即元媒体这两个角度，来深入地探讨一下由拟物化设计向扁平化设计转变的意义。

隐　喻

拟物化设计的根本在于隐喻（Metaphor）。迄今为止，用户界面可以说基本上是采用隐喻的表现手法来实现的。按钮、页面、滚动条、文件夹、桌面、垃圾箱等，这些都是以现实世界中的实物为模型进行模仿的隐喻表现形式。按钮模仿真实机器上的按钮，页面模仿真实书本的页面。

隐喻的使用使得人们在日常生活中用到的知识在电脑中也能够得以继续应用，即使不懂电脑的人，也能够通过隐喻的表现形式结合自己的生活常识推测出它的功能和作用。

可以说在UI的历史发展长河中，隐喻表现手法的应用具有深远的意义，它向广大的普通用户展示了“电脑是什么”以及“电脑能做什么”。Mac OS或者iOS 6及其之前的版本都积极地采用了拟物化设计，其原因不仅仅是为了美观，更是为了让用户能够由此推测出它的功能到底是什么。

元媒体

对于电脑而言，隐喻表现手法如此重要还有其他的原因。那就是因为电脑是一种元媒体（Meta-media）。元媒体的概念是由被称为电脑之父的阿伦·凯（Alan Kay）提出的。元媒体的概念表明它既可以成为一种道具或者媒体，也可以成为其他任何一种东西，能够实现人们见所未见闻所未闻的表现形式和功能。

电脑是一种元媒体，可以说“无论什么都是它的装置”。也正因为如此我们才需要对它具体是什么加以定义，这是电脑的本质要求。而隐喻则是进行定义的有效方法。也就是说，原本作为计算机的电脑，变成了文档编辑装置，变成了乐器，变成了绘画装置……这本身就是一种隐喻。不仅如此，在电脑软件和应用程序的领域中也使用了隐喻。

使用隐喻对元媒体进行定义的局限性

但是，使用隐喻对元媒体进行定义也是有局限性的。使用隐喻手法实现的文档编辑装置或者乐器等，其实都是迄今为止我们现实文化生活中真实存在的物品的一种替代品。而实际上在电脑中能够表现出比现实世界更多、更丰富的东西。对此，隐喻的表现手法就存在局限性了。对于作为元媒体的电脑而言，我们当然是想发挥其真正价值的，但如果使用隐喻表现手法的话，无论到何时所实现的都只能是现有文化和概念的替代品而已。

因此，挣脱隐喻表现手法的束缚，进而实现自由的表现方式在今后将变得尤为重要。由此我

们可以进一步认为以隐喻为根本的拟物化设计也将会变得步履维艰。

由于隐喻自身的局限性，其缺陷已经开始部分地显现出来。比如，无论是Windows还是Mac，目录都是采用“文件夹”的表现形式，文件夹中又可以无穷无尽地嵌套子文件夹。而这种形式在现实世界中是不可能存在的，从而导致人们对于计算机文件结构的理解产生了误区。也有人认为如果用户理解了计算机的原理，那么也许就不再需要什么隐喻或者拟物化了。

另外，对于世界上根本不存在的东西，如果使用拟物化的表现手法，则存在着不可逾越的壁垒。比如在iPhone或者Android这个新的平台上，庞大的用户群体从各种不同的生活场景自由随意地访问网络，孕育出该平台特有的创意或服务的可能性越来越大。在这里，存在着很多现有概念难以解释的价值形式，没有可供参照的隐喻手法，使用拟物化的设计方法是行不通的。

简而言之，我们可以认为由于以下两点原因，导致了拟物化设计已经被数字领域的原生设计手法，即扁平化设计所取代。

- 由于隐喻自身的局限性，可能会导致理解上的误区。
- 对于世界上根本不存在的新的服务形式，原本就难以用隐喻来实现。

扁平化设计的世界

综上所述，要充分运用并发挥元媒体这一表现形式的自由性和灵活性特点，就要采用扁平化设计。在扁平化设计的世界里，没有必要考虑如何去表现物理层面的制约因素，也没有必要考虑如何去表现文化因素，采用扁平化设计能够充分发挥计算机的性能和设备的特性，这是一个能够自由进行设计的世界。

例如，因操作快捷而为人们所熟知的Todo软件中有一个叫作Clear[①]的应用，采用的就是扁平化设计。如果没有人告诉我们这是一种Todo软件，我们根本无从知晓它到底是什么，或者应该把它比作什么。但是它的操作性却非常自由。迄今为止，它那令人愉悦的快捷操作是无可比拟的。通过宣传视频[②]大家也能够感受到这一点。

不使用隐喻能创造出直觉来吗？

虽然说扁平化设计是一种不受约束的自由的设计方法，但是也并不是说它已经真正地达到了绝对自由的境界。

它的制约因素首先体现在计算机的性能和设备层接口之上。例如，多点触控和非多点触控环境下的设计方法当然会有所不同。

另一个设计上的制约因素则体现在人类的知觉、认知和身体特性之上。例如UI的动作方式或页面的隐藏方式对人类的知觉而言都是最基本的要素。iOS 7中使用视差效果[③]来表现纵深的手法，可以说就是利用了人类知觉特性的一种设计吧。

如果人们发现一种交互式的设计方法，能够将这种设备特性和人类特性进行相互融合的话，那么必将促进用户界面发生进化，创造出下一代智能手机的原生用户界面。经过一定的积累和沉淀之后，就会形成一种与之相应的世界观（规则），从而人们也就会逐渐地明白如何根据这种设计驾驭自己的直觉了。

毫无疑问，今后考虑了上述要素的UI技术在经过一定的积累和沉淀之后势必会彰显其重要性。但是，我想更为重要的应该是应用程序自身的定义能力和宣传能力。

应用程序设计的重点始终在于该应用是做什么的，要实现什么样的价值。但是，对于该应用具体是什么，在使用扁平化设计去表现的时候，因为没有了隐喻手法，所以相应地要比拟物化设计困难得多。因此，对于这种不容许使用隐喻表现手法的扁平化设计而言，定义能力就会变得尤为重要。定义能力就是指具体问题是什么以及应该如何解决，即问题和解决方法的明确程度。

另外，UI如何运行，遇到问题如何解决，如何体验操作快感等，更加需要演示和宣传来传达给用户。

UI设计的未来

这一期，我们从隐喻和元媒体的角度对于拟物化设计和扁平化设计进行了探讨。UI设计依赖于采用怎样的隐喻手法，需要考虑应用程序要解决什么样的问题以及要在怎样的范围内解决该问题。迄今为止，隐喻手法是UI设计方法论的核心，而扁平化设计从某种意义上而言，将是UI设计所面临的一种挑战。也就是说，在UI的表现形式和用户体验规模进一步拓展的同时，在不使用隐喻手法的前提之下，UI设计人员能够将多少价值传达给用户将是问题的关键所在。

① http://www.realmacsoftware.com/clear

② Clear for iPhone (Coming Soon!)-Official Video
http://www.youtube.com/watch?v=S00H-rz7fGo

③ 指随着观测者的移动而产生的重叠在一起的移动变化。

特辑 1

UI设计实践

提高用户满意度的设计、实现和验证方法

本特辑将要通过Cookpad网站*介绍UI设计方面的实践知识。下面会以Cookpad设计的UI为例，将开发步骤划分为建立假设、进行开发和验证效果三个部分，由工作在第一线的工程师们向大家详细讲解UI设计的流程、注意事项等进行UI设计必须要了解的知识。

* Cookpad是日本最大的在线食谱分享网站。——译者注

第1章

开发人员所追求的UI设计

明确为用户提供的目标

文/五十岚启人（IGARASHI Hiroto） Cookpad股份有限公司
译/卫昊

本特集主要通过Cookpad公司的设计师们常用的UI设计方法，讲解一些设计师和开发人员必须了解的基础知识。

作为特辑的开篇，本章将介绍开发人员对待UI设计应该持有的态度，即开发人员为了使自己开发的服务获得成功，应该如何理解UI设计，以及作为一名开发人员，UI设计能使自己得到怎样的成长。

UI设计的目的

UI是你开发的服务与用户之间的连接点。用户通过UI使用你的服务，评价你的服务。而UI设计就是设计服务与用户之间的关系，这是一个非常有意义的过程。

说到UI设计，大家可能会联想到绚丽多彩、富有魅力的图形图像。图形图像固然可以提高用户对服务的印象分，使用户更加愉悦地使用我们的服务。但是，图形图像只不过是UI设计中的一部分，即便没有这方面的专业技巧也完全可以设计出漂亮的UI。

实际上，UI设计最重要的目的是：使用户认识到自己通过服务能达成什么目标，并指引用户以正确的方式使用服务所提供的功能来顺利达成这些目标。为此，我们不仅要针对用户操作的画面进行设计，还必须要从用户的整体体验出发进行UI设计。

开发人员必须了解的基础知识

在我所效力的Cookpad公司，开发人员都被称为“造物者”，受到别人的尊敬。除了开发产品，我们还要深入思考这些产品能使用户获得何种价值和体验。

这是因为一个无法明确用户能获得何种价值的服务，最终在商业上也会变得毫无价值，而且浪费开发以及UI设计的资源。

设计用户通过服务可以获得的价值

那么，就用实例来确认一下UI设计和服务价值之间的关系吧。

Cookpad一般被认为是一种“寻找食谱的服务”，但这并不是用户通过Cookpad可以获得的最本质上的价值。“寻找食谱”不过是Cookpad上的功能之一，其本质上想向用户提供的价值则是“发现今天想吃的东西”。

比如从搜索功能上也能看出这点。除了Cookpad以外，在搜索引擎上寻找食谱也是可以的。但是，Cookpad的搜索功能想提供的从来不仅仅是寻找食谱，而是找到今天想吃的东西这一价值，我们需要以此为目标来进行UI设计。

在开发服务时，理解自己开发的服务要向用户提供何种价值是十分重要的。能否在此基础上进行UI设计，是决定UI优劣的关键性因素。

在平时就去体验各式各样的服务

那么我们到底该如何通过服务向用户提供最好的体验和UI设计呢？

对开发人员和设计师来说十分重要的一点是：在平时就去体验和了解由不同人提供的、各式各样的服务。这样自己在进行设计时，就可以选择同样的手法。

◆ 体验其他Web服务提供的用户体验

对开发人员来说，体验和自己所要开发的服务相似的，或者自己感兴趣的各种服务和应用程序是唾手可得的事情。开发人员可以先列出一张清单，上面写上与自己想要开发的服务、在工作中正在开发的服务相似的数十个服务，然后分别使用、学习。

◆ 体验现实世界中的各种服务

可能的话，体验现实中世界各地的、由不同的人提供的各种服务，对提高自己所开发服务的品质一定有着很大的帮助。

例如，著名的设计公司IDEO为了再次设计医院急诊室的工作流程，就去观察美国有名的纳斯卡赛车（NASCAR），并从赛车修理站获得了灵感①。乍一看二者毫无关系，但运送至急诊室的急救病人和进入修理站的车手一样，都是为了恢复正常，而变成了“任人摆布的状态”，在这一点上他们有着很相似的体验。

◆ 体验各种事物时需要注意的事项

在体验其他人下工夫开发的服务或UI设计时，我们需要意识到下面两个问题，这样才能够更加深刻地体会到对方的用心良苦，然后带着尊敬去“盗用”对方的成果。

- 注意好的地方，而不是差的地方。
- 发现并观察与自己性质不同的地方，而不是相同的地方。

世界上有许多人在精心打造并努力提升服务的品质。让我们体验不同的服务，总结它们的特点并好好领会其中的精华吧。随着经验的积累，你也会变得更加注重用户的体验。

① 在Diamond Online的报道“仅拥有600名员工，却和苹果、谷歌并肩的世界上最具革新性的公司——IDEO的思考方式”中提到过这件事。http://diamond.jp/articles/-/36808?page=3

如何明确为用户提供的目标

前一节叙述了在进行UI设计前，我们必须明确所开发的服务向用户提供何种价值。下面将讲解从整理服务的价值到进行UI设计前的这个阶段。

制作产品定义文档

在头脑中确定了服务应向用户提供的价值之后，为了明确服务的目标，我们还要将其落实成“产品定义文档”。

产品定义文档是非常有力的工具，它能与一起开发服务的伙伴共享目标，或让自己在一人开发时不偏离服务的方向。

产品定义文档只要包含下面的要素，以何种形式书写都没有关系。

- 什么样的人使用这项服务？
- 用户使用这项服务是为了解决何种问题？

产品定义文档根据人和团队的不同，有着各式各样的叫法、书写方式以及定义方法，下面介绍几个Cookpad使用的框架方法。

❶ Emotion Oriented Goal Sheet（EOGS）

这是在Cookpad使用时间最长的一个框架。我们会列出主要的利益关系者，以他们的根本需求作为出发点，找出他们的共同目的，制定服务目标。因为拥有明确判断何为成功的标准，同时能够提供数字上的参考，所以这种框架在商业上也可以使用（图1）。

❷ 根据用户故事定义服务

这是在开发较小规模的服务或添加新功能时可以使用的框架。这个框架不以“能够寻找食谱”这种功能性的描述来定义服务，而是采用“互联网用户每日为吃饭烦恼时，能够决定想吃的

东西并知道如何制作”这种形式，制作以解决用户问题为核心的用户故事模板（图2）。

❸ 价值假说

价值假说和❷很相似，但是更加关注用户遇到的问题。如图3中的模板所示，这个框架从用户的角度来定义服务的价值。

❹ Goal Directed Design

这是通过创建虚拟的人物角色，并归纳这个虚拟用户的各种情感来定义产品的方法。这个方法将在第2章中以实例进行详细的讲解。

◆苹果和微软推荐的方法

在Cookpad公司内部，不同的服务规模和不同的团队都会有各种不同的框架方法，那么不同的企业当然也会有各种各样的方法。例如苹果和微软发布的文档中就推荐了如表1所示的方法。

要想检验这些框架方法是否好用，将一个已经成功的服务或产品代入它们来一一验证也许是个不错的选择。

◆修改产品定义文档

产品定义文档的内容根据服务的状态和阶段会发生很大的变化。Cookpad最初也不是把“决

▼图1 EOGS的例子

▼图2 使用用户故事表的例子

作为一个手机用户，当我在制作料理的时候，
希望可以在不弄脏手的情况下浏览食谱

▼图3 使用价值假说表的例子（按人气搜索食谱）

用户	寻找食谱的用户
需求	希望可以尽早决定今天要做的料理
课题	食谱太多了不知道该如何决定
产品特征	可以搜索人气食谱的话将会很有价值（价值）

▼表1 其他定义方法

事例	URL
苹果提出的公司内部应用程序开发的例子	http://www.apple.com/jp/business/accelerator/plan/define-your-app.html
微软提出的Windows商店应用开发的例子	http://msdn.microsoft.com/zh-cn/library/windows/apps/hh465427

定每天该吃什么”作为核心的服务目标，而是以“能够轻松刊登食谱”作为其核心价值。

为此，Cookpad十分重视以传统地刊登食谱为目标的用户UI设计。不过也正是因为Cookpad在服务初期明确的这一核心价值，现在才有可能收集到如此之多的食谱吧。

撰写达成目标的脚本

明确了向用户提供的价值之后，我们还要为了确保用户可以顺利到达服务的“终点”（即使用服务达成某种目标）而进行架构设计。

◆导致用户“迷路”的原因

主要有两个因素会导致用户在前进时“迷路”。

❶ 无法认识到该以怎样的步骤达成目标，无法理解步骤间的关联性
❷ 个别步骤无法如愿以偿地使用

UI设计的主战场在第2个因素上，但如果因素1有理论上的破绽，那么无论在因素2上下了多大的工夫，也不会提升用户对服务的评价。以公交车为例，最近，电子票的引入使得我们在坐公交车时付钱买票变得非常简单。但是，根据地域的不同，有着前门上车、后门上车、上车付钱、下车付钱等各种方式，乘客必须在公交车到达的一瞬间，根据车体的外形区别、判断它们。公交车原本是达成“移动到别的地方”这一目标的手段，结果却给使用者带去迷惑和不安，让他们犹豫是否还要使用这个手段。

◆撰写脚本的步骤

为了让用户可以顺利到达服务的“终点”，让我们来撰写一个脚本范例，描述服务的使用步骤。

- 将条目列在纸上
- 在纸上画出简单的画面迁移图

以刚才的公交车为例，按照共同的目标，将用户划分为等车、乘车等阶段，并列在纸上。讨论一下每个步骤中哪些事情是必要的，步骤的顺序是否难以明白，然后在纸上画出简单的模型。这个方法有很多具体做法，在第2章中将配合实例进行讲解。

可能的话，我们还要与他人交换意见，按照之前定义的“产品定义文档”编写脚本，然后再进行各步骤的UI设计，并且为了用户可以如愿以偿地完成各个步骤而进行相应的调整。

提供具有一致性的UI设计

前面讲述了实际进行UI设计需要的各种前提和准备。具体的UI设计方法和技术将在第3～5章中介绍，但它们都以如下几条内容作为基础。

为了用户不会迷失目标而进行UI设计

要想让用户可以使用服务达成目标，排除途中会迷惑用户的因素是十分必要的。为了使用户不会受达成目标以外的因素影响，进行UI设计时常常需要确认如下内容。

- 服务是否与操作系统的标准UI，或者与提供相似价值的知名服务的UI相似，用户能否凭借已有经验直接使用我们的服务
- 是否统一了服务中图标与文本的显示和使用方法，是否可以预测用户使用UI的结果

向UI设计中增加新内容时，最需要注意的就是设计的一致性。没有一致性的UI设计会导致用户陷入混乱，注意力被分散。

而且，我们也需要慎重考虑用户是否真的需要这个UI或功能。开发人员经常会自作主张地添加一些不必要的功能和信息。总之，我们要慎重考虑新添加的UI或功能能否对用户达成目标提供帮助，仅仅保留绝对不能缺少的内容才会提高我们服务的品质。

使用用户可以理解的语言

UI设计中开发人员很容易将重点放到图形和布局设计上，但是配合UI使用的语言也需要十分注意，特别是如下几点。

- 一致性的书写和表达方式
- 使用用户立场的语言(例如：加入会员需收费→加入会员需要交纳手续费)
- 使用服务对象(即用户)熟悉的语言

99%的引用和1%的原创

有时我们在开发新的服务时，会精心设计并准备新服务专用的UI，而这正是最需要我们慎重的时候。

旧的UI经过长时间的使用，积累了各种知识和用户的反馈，是用户体验的宝库。学习和再利用这些经验，可以防止开发人员“重新发明轮子”，从而有效率地进行UI设计。参考类似的作品，你的服务中有99%的部分可以再利用已经存在的UI，最后再加上1%精心设计的、具有你自己服务特征的UI就可以了。在UI设计上，引用成功者的成果并不是一件可耻的事。

学习各操作系统的设计指南

各个操作系统都会向开发人员提供一套标准的UI组件，让他们在开发服务时可以重复利用。而且，各个平台都会将各自UI组件的使用方法整合为“设计指南”并公开，开发人员可以从中学习到各个操作系统的UI设计模式以及“文化和理念”(Tone and Manner)。

具有代表性的设计指南如表2所示。在Web上搜索它们后就可以免费阅读。这些指南毫不吝啬地公开了各公司的顶级设计秘籍，我建议各位读者务必浏览一下。在和其他设计师、开发人员一同进行开发时，学习过各个操作系统中UI组件的名称，还可以使相互间的沟通变得更加简单。

而且，在进行非各操作系统原生应用程序的Web开发时，通过学习各操作系统的理念，吸收各个操作系统的优点，我们就可以设计出不针对特定操作系统用户的通用性的设计。

总 结

对于想尽早开始UI设计的读者来说，本文到现在为止似乎都在“绕远路”。可是，开发人员应该抑制自己急于开始设计的心情，首先从自己开发的服务着眼，认真思考、整理，这才是使UI设计成功的最为重要的一点。

既然UI设计前需要做的准备都已经完成了，下一章我们将配合实例解说UI设计的具体步骤和技巧。

▼表2 具有代表性的设计指南

名称	URL
iOS Human Interface Guidelines	http://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/
OS X Human Interface Guidelines	https://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/OSXHIGuidelines/index.html#//apple_ref/doc/uid/TP30000894-TP6
Android Design	http://developer.android.com/design/index.html
Windows商店应用程序的UX指南	http://msdn.microsoft.com/zh-cn/library/windows/apps/hh465424.aspx
Windows用户体验交互指南	http://msdn.microsoft.com/zh-cn/library/aa511258.aspx

第2章

为了UI设计而进行的用户体验设计

找出用户想要达成的目标

文/伊野亘辉（INO Noriteru） Cookpad股份有限公司 译/卫昊
URL http://cookpad.com/ mail noriteru-ino@cookpad.com Twitter @memocamera

考虑何为好的设计

在设计服务的UI时，首先我们必须好好考虑“究竟何为好的设计”。无论使用何种服务，人们总是有意识或无意识地想要达成某种目标。这个目标可能是“获得有用的信息”，可能是“轻松确定每天的菜单”，也可能是“寻求他人对自己的认同感”。

好的服务就是可以帮助用户通过行动达成目标的服务。明白了用户的目标后再进行服务的UI设计是非常重要的。

本章将从对用户目标的调查、设计，重要功能的选取这两方面出发，结合Cookpad的iPhone应用开发实例来介绍UI的设计方法（图1）。

▼图1 使用本章介绍的方法开发的Cookpad应用

用户想要达成何种目标

用户使用服务到底是想达成什么目标呢？我们又该如何发现这个目标呢？有几种方法可以帮助我们明白这些问题。

最简单的方法就是直接听取用户的意见，观察他们的行动，这种方法一般被称为用户调查。以调查得到的信息为基准进行开发，自然而然就能知道如何才能开发出好的服务。

在服务开发初期，精心地进行用户体验设计，明确开发小组中大家应该完成的目标是非常重要的。如果能做到这点，可以说服务开发已经有了一个成功的开始。

用户体验设计的步骤

用户体验设计分为如下几个步骤。第1章提到过的产品定义文档就是通过前5个步骤实现的，并且最终记入用户体验设计书。

1. 找到作为调查对象的用户
2. 对用户进行调查
3. 挖掘用户的目标
4. 创建人物角色（Persona）
5. 制作用户体验设计书
6. 撰写脚本
7. 从脚本中挑选出重要的功能
8. 进行UI设计