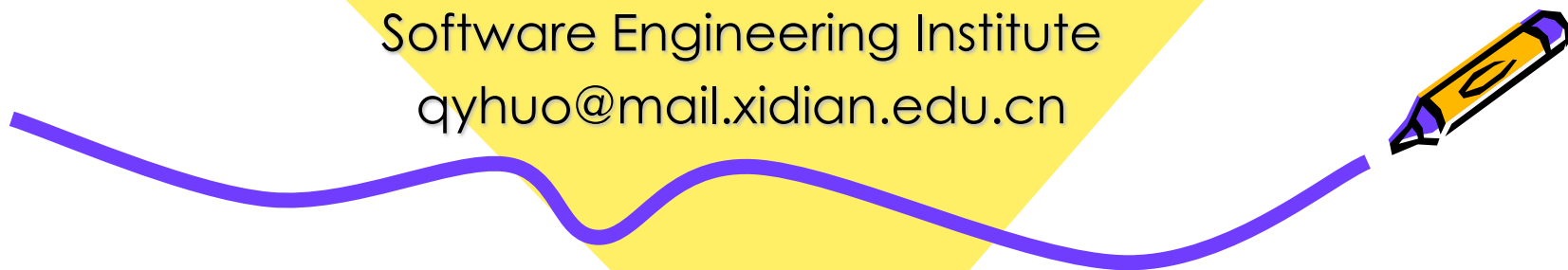




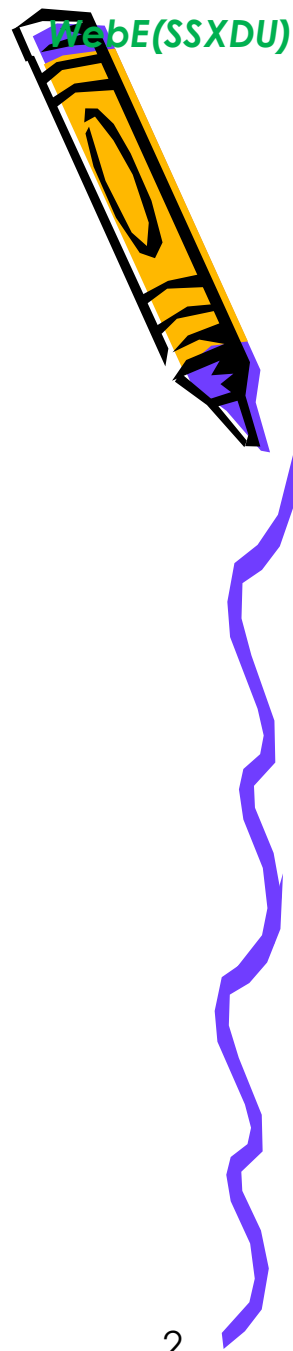
Web应用设计

Qiuyan Huo 霍秋艳
Software Engineering Institute
qyhuo@mail.xidian.edu.cn



Outline

- Web应用设计特性
- 交互设计
- 展示设计
- 内容设计
- 功能设计
- 总结与展望
 - Web Design Trends for 2018



WEB应用设计特性

超文本

- 节点、链接和锚 (anchor)
- 网和其它聚合体
- HTML:非线性
 - 在文本文档加标记→忽视节点的原子性，文档（节点）长度难以控制
 - 通过链接和锚标记等确定超文本结构→布局混杂
 - HTML缺乏软件架构的抽象机制
 - 很多其它媒体无法由链接来支持，只能通过链接目标来实现
 - 对节点和链接类型的自定义、反向链接、分离链接和节点的存储、不同锚等仍然无法由HTML支持



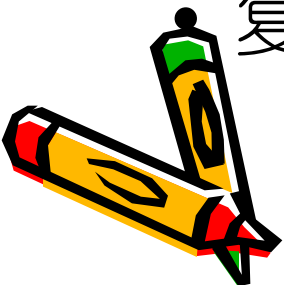
基于文档的Web应用

- 基本规则
 - 网应该是信息设计的核心；
 - 传统文档应该分解为原子节点；
 - 即使技术上不支持布局与内容、节点和网等的分离，也应该在概念上进行区分；
 - 所选技术应该支持如集中链接管理、内容管理等高级概念，至少是在设计时支持，应该优先考虑基于XML 的解决方法。

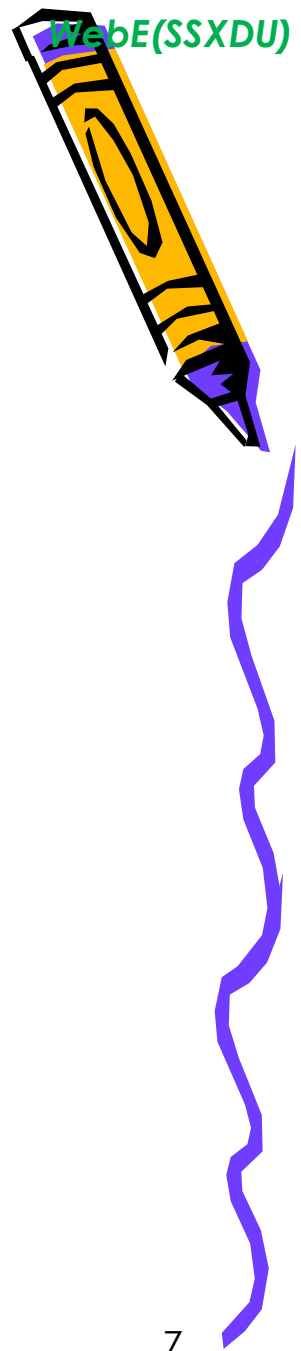


Web应用设计

- 将Web应用设计分为子任务
 - 区分Web应用的组件（即节点和链接）
 - 网（包含组件的整个Web应用）
- 轻量级
- 美学、内容和技术适当结合
- 人才是核心→从对最终用户来说很重要的问题开始设计
- 吸引用户→好的观感→展示设计
- 复杂性的提高→功能设计、内容设计



交互设计



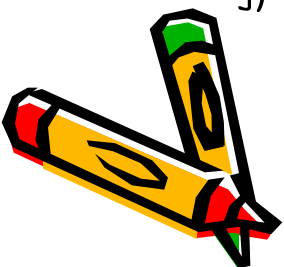
交互设计

- Web应用的可视性、动态性、功能性和技术性元素的交织
- 目的是很好地将这些元素综合并解决相互之间的冲突，以提供给用户感兴趣的、有吸引力、一致的和易于理解的体验。
- 人机工程学、人机交互界面、可用性工程、视觉传达设计、认知心理学、工业设计、用户体验设计等



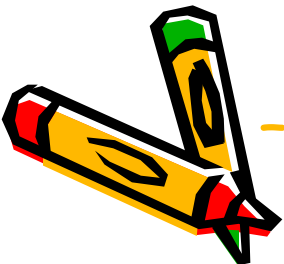
用户交互

- 交互：信息的互换、互动和反馈，即需要用户做出某种动作，如填写表单、点击链接或按钮等等。
- 交互设计是人工制品、环境和系统的行为，以及传达这种行为的外形元素的设计和定义，它首先旨在规划和描述事物的行为方式，然后描述传达这种行为的最有效形式。
- 人性化的设计是Web应用设计的核心，重点是如何根据人的心理、生理特点，运用技术手段，创造简单、友好的交互方式。
- 设计标准：可维护性、可重用性、可伸缩性、可扩充性和可持续性。



用户页面组织

- 集成方面：Web应用的应用页面经常需要展示大量的信息、信息上的操作，以及信息之间的关系。
- 衡量屏幕大小和节点原子性
 - 拆分？导航？滚动？
 - 复杂行为和可移植性？
- 根据导航语义、可移植性以及可用性
 - 节点一次性以HTML全部发送给用户
 - 节点一次以大HTML页面方式发送给客户，不包含脚本
 - 节点一部分发送给用户



导航设计

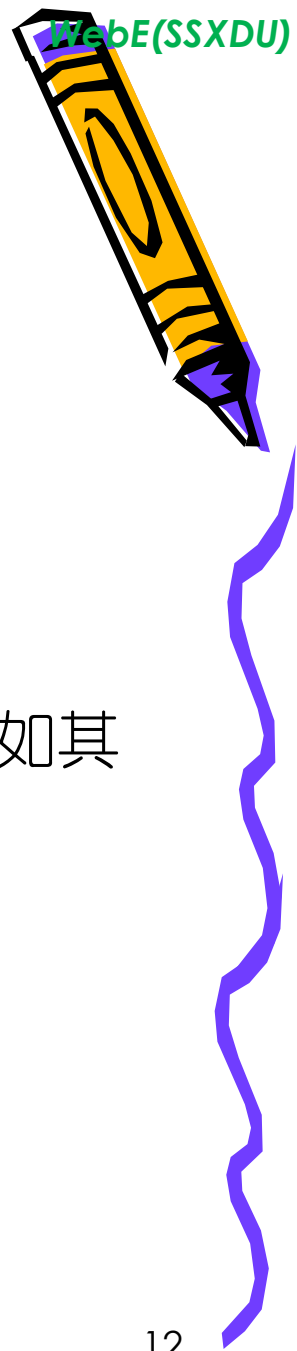
- 用户可以访问的元素和导航结构的设计
 - 元素变成节点，结构定义节点之间的关系
- 设计链接的代表——锚 (Anchor)
 - URL的可视化展示方式 <a>
- 导航及其方向：减轻用户的认知压力
 - 导航组织策略，用于确定整体导航结构
 - 方向援助策略，解决展示设计的交互方面的“我在哪儿？”和“我访问过哪儿？”等问题
 - 链接认知策略，关注链接关联有关的目的和结果



复杂活动的会话设计

- 用户参与才能触发一些扩展步骤的可视化
- 前向导航分为三种：
 - 导航步骤触发一个动作
 - 导航只调用一个页面，如一个表单的第二页
 - 导航步骤导向一个并不直接参与活动的节点，如其它信息等
- 业务过程和超文本
 - 通过简单HTML完成
 - workflow 管理的方法和工具

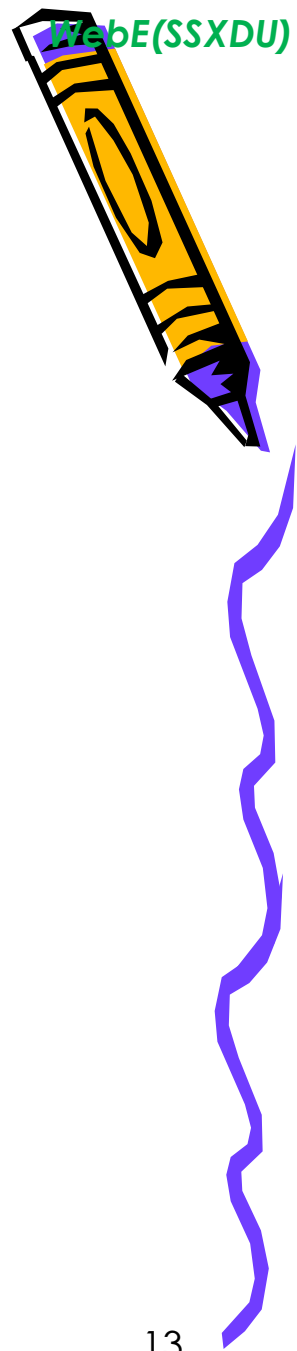
尽量将复杂任务和导航的可能性关联起来



交互设计原则

考虑与技术和架构的相互作用和相互影响

- 控件控制
- 表单控制
- 窗体控制
- 输入设计
- 提示信息
- 出错处理及出错页面的跳转



交互设计模式原理一：直截了当

- 页内编辑

- 单字段行内编辑：编辑一行文本，*Flickr*照片标题
- 多字段行内编辑：编辑更复杂的内容，*Backpackit*
- 覆盖层编辑：在覆盖层面板中编辑，*日历*
- 表格编辑：编辑网格中的项，*Google docs*
- 群组编辑：直接修改一组项，*iPhone*主屏
- 模块配置：直接在页面中配置设置



交互设计模式原理一：直截了当

- 利用拖放
 - 趣味瞬间（交互事件）
 - 拖放模块：重新排列页面上的模块，*Netvibes*
 - 拖放列表：重新排列列表项的顺序，*Backpackit 计划任务列表*
 - 拖放对象：改变对象间的从属关系，*Cogmap 重拍组织结构*
 - 拖放操作：在被放置对象上执行操作，*上传照片*
 - 拖放集合：通过拖放操作集合，*Laszlo 加购物车*



交互设计模式原理一：直截了当

- 直接选择

- 切换选择：复选框或基于插件的选择，*Gmail*
- 集合选择：跨越多页的选择，*Gmail* 跨页全选
- 对象选择：直接选择对象，*Laszlo* 的 *WebTop*
- 混合选择：切换选择与对象选择的组合，*Yahoo!Mail*



交互设计模式原理二：简化交互

- 上下文工具：费茨定律(Fitts's Law)

- 实时可见工具：把上下文工具直接放在内容中，*Digg*
- 悬停即现工具：在鼠标悬停时显示上下文工具，*Yahoo!Buzz*
- 开关显示工具：打开/关闭页面中上下文工具的总开关，*Basecamp*
- 级联递进工具：基于用户操作渐进显示工具，*Songza*
- 二级菜单：显示二级菜单，*Google Maps*



交互设计模式原理三：足不出户

- 为创造连续视觉感知的**心流**最高境界服务
- 覆盖层：在当前页面上显示一个可以代替新页面的、轻量级的迷你页面，Orbitz的DHTML日历选择器
 - 对话框覆盖层，Netflix购买确认
 - 详情覆盖层，Netflix鼠标悬停显示电影剧情简介
 - 反模式：捕鼠器(Amazon Buy Now)、激活/禁用不对称、过分炫耀(Borders online)、悬停与遮挡(Barnes & Noble 详情)
 - 输入覆盖层，American Express引导注册
 - 反模式：悬停与遮挡



交互设计模式原理三：足不出户

- 嵌入层：把信息或操作以嵌入方式显示在当前页面
 - 对话框嵌入层, *BBC主页*
 - 列表嵌入层, *Google Reader列表视图*
 - 详情嵌入层, *Roost详情嵌入住宅照片缩略图*
 - 标签页, *作为详情嵌入层*



交互设计模式原理三：足不出户

- 虚拟页面：借助动态内容显示及动画技术扩展页面虚拟空间
 - 虚拟滚动, *Yahoo!Mail* 动态加载邮件
 - 内置分页, *iPhone* 显示*Tunes*商店搜索结果分页
 - 滚动分页--传送带模式, *Yahoo!Underground*
 - 虚拟摇摄, *Google Maps*
 - 伸缩式用户界面 (ZUI) , *Hard Rock Cafe*



交互设计模式原理三：足不出户

- 流程处理：在同一个页面中实现既定流程，*Google Blogger*通过三个步骤删除评论，魔法原理
 - 交互式单页，*Gap*实时显示每种颜色有货的尺码
 - 嵌入式部件，*Gap* “Add to Gag” 处理
 - 对话框覆盖层，*Discover*的填写个人信息
 - 配置程序，*Apple*为顾客购买设计了一个单页配置程序
 - 静态单页，*eBay*提供的简化的 “Sell Your Item” 静态单页



交互设计模式原理四：提供邀请

- 静态邀请：通过可视化技术
 - 引导操作邀请：执行一个或多个主要任务，*Yahoo!Answers*、*Discover Card* 明确的1-2-3步骤
 - 漫游探索邀请：探索新功能，*Yahoo!Maps*

引导用户进入下一交互层次的提醒和暗示



交互设计模式原理四：提供邀请

- 动态邀请：响应用户在特定位置执行特定操作
 - 悬停邀请：鼠标悬停期间发出邀请，Gap商品详情
 - 预期功能邀请：使用熟悉的事物引出新事物，Flickr会在用户鼠标悬停于另一个用户头像上时显示一组操作菜单
 - 拖放邀请：引导用户拖放，Yahoo!Teachers Gobbler
 - 推论邀请：用户交互期间，Google Sketchup绘制3D对象
 - 更多内容要求：邀请用户查看更多内容，Yahoo!个人助理基于悬停邀请展示更多内容邀请；Yahoo!Games使用两级查看器表示还有更多游戏



交互设计模式原理五：巧用变换

- 变换模式
 - 加亮和减暗、扩展与折叠、自恢复式淡出、动画及聚光灯效果、亮盒效果、面板互换、翻转、传送带、折叠窗格、滑入与滑出
- 变换的目的
 - 增添魅力、增进沟通



交互设计模式原理六：即时反应

- 查找模式
 - 自动完成：Gmail用户输入地址时匹配
 - 实时建议：Google Suggest
 - 实时搜索：Amazon ZUGGEST
 - 微调搜索--多侧面浏览：动态筛选过滤实时调整搜索结果，Kayak左侧筛选右侧实时调整行程

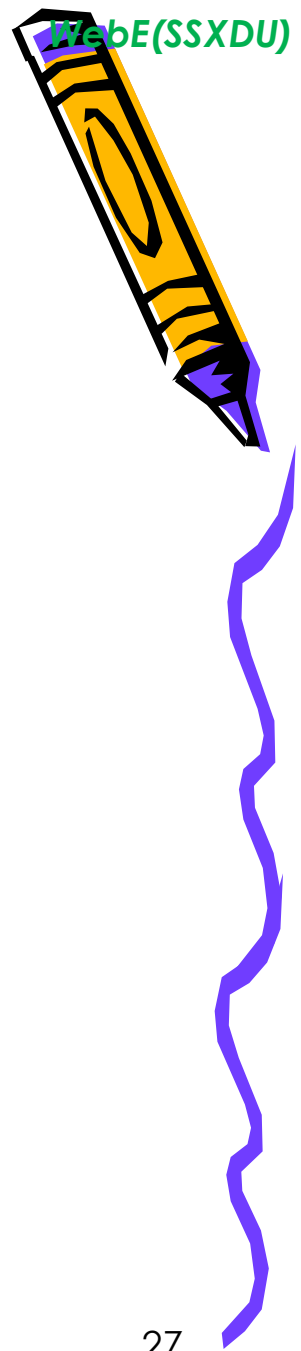
交互设计模式原理六：即时反应

- 反馈模式

- 实时预览：一份预防胜过十分治疗，wundrbar.com
- 渐进展现：必要时再给出提示，*Picnik*的创建账户
- 进度指示：冗长过程中反应性界面，*Forecast*
- 定时刷新：无须用户介入定时展示新内容，*Digg Spy*

与时俱进

展示设计



展示设计

- 定义视觉和某种意义上多媒体内容的表现形式
- 交互设计的延伸
- 关注用户界面的结构和组织形式
- Web应用的功能、对工作流程顺序和相关问题的描述
- 美学设计（也称为美术设计），描述Web应用的外观和感觉
- 根据人的特点，运用技术手段，创建简单、友好且令人满意的人性化的Web页面，是Web应用展示设计的重点



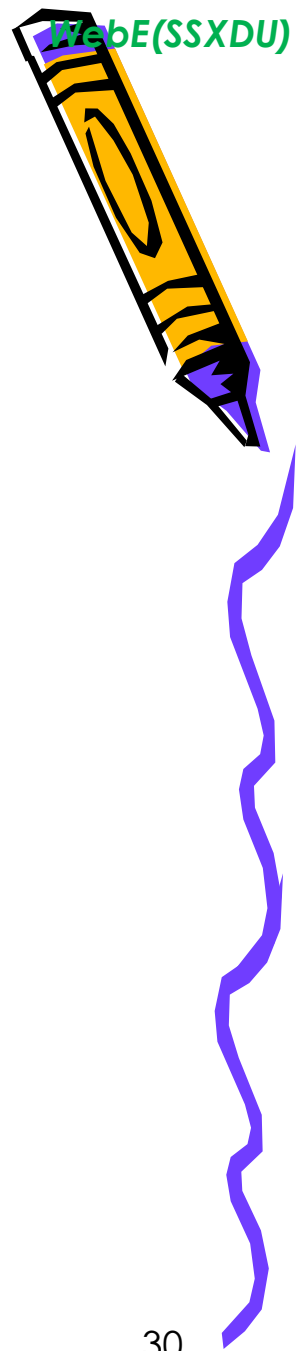
展示设计

- 好的展示设计
 - Web应用的多媒体内容从组件角度进行开发，并隐含地在网中进行定义
 - 应该针对各种文化、技术和上下文需求，具有很好的适应性



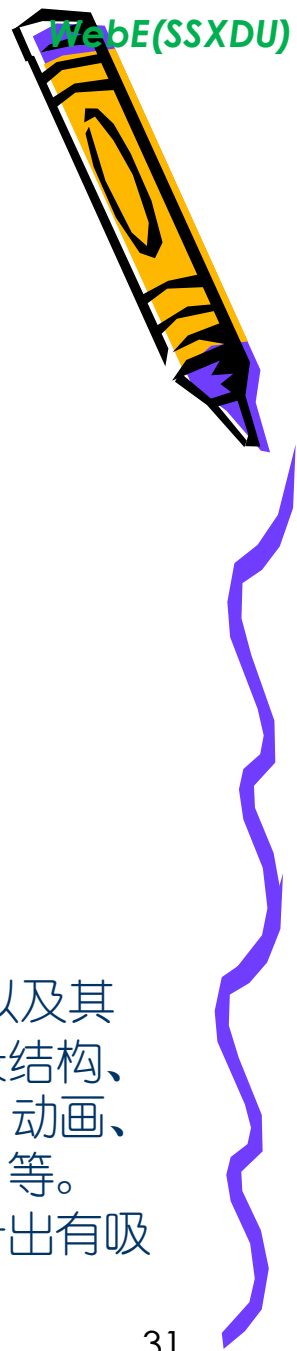
Web页面特性

- 导航更灵活
- 多层架构
- 页面包容
- 布局不同
- 分工明确
- Web页面更易变化



Web页面设计特性

- 新颖性
 - 信息性
 - 多维性
 - 娱乐性
 - 交互性
 - 多媒体使用
 - 动态性
 - 虚拟性
 - 艺术性
 - 适应性
 - 可伸缩性
- Web页面设计需要考虑各种组件要素、导航链接网、以及其它诸多要素，如用户的行为习惯、页面框架布局（目录结构、页面布局、页面尺寸等）、页面元素（LOGO、图片、动画、文字、点、线、面等）、美学（色彩、风格、创意等）等。
 - 只有认真规划，遵循一定的设计思路和策略，才能设计出有吸引力的Web应用页面。



用户行为习惯

- “一幅图胜过千言万语”
 - 吸引力or
 - 信息板块
- 尊重绝大多数用户的**习惯**
 - 大多用户浏览页面都是首先注意偏左上角的内容，用户的浏览顺序一般是从上到下，从左到右
 - 用户**匆忙**→**标题的代表性**



页面布局设计

- 合理的页面布局
 - 将Web页面上的文字、图片、色彩、音频、视频等展示元素合理安排和表现。
 - Web页面设计的要素协调安排，充分利用并有效分割有限的空间。
- 线框（Wireframes）



什么是线框?

- 线框描述Web应用内的一个页面（或多个页面）如何从概念上来看达到并行的目的。
 - 捕获那些应该显示在页面上的核心信息和导航元素以及这些元素的大概布置;
 - 一个没有图形设计或实际实例内容的页面表达，可用于表达内容的集成以及信息结构从Web应用内的一点来看像什么。



什么是线框？



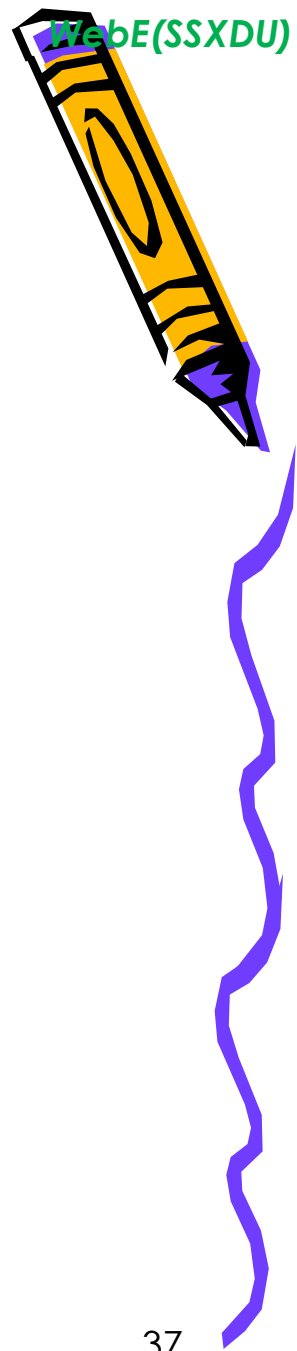
什么是线框？



Wirify將網頁轉換為線框的小工具

布局类型

- 国字型布局
- 拐角型布局
- 框架型布局
- 封面型布局
- Flash型布局
-



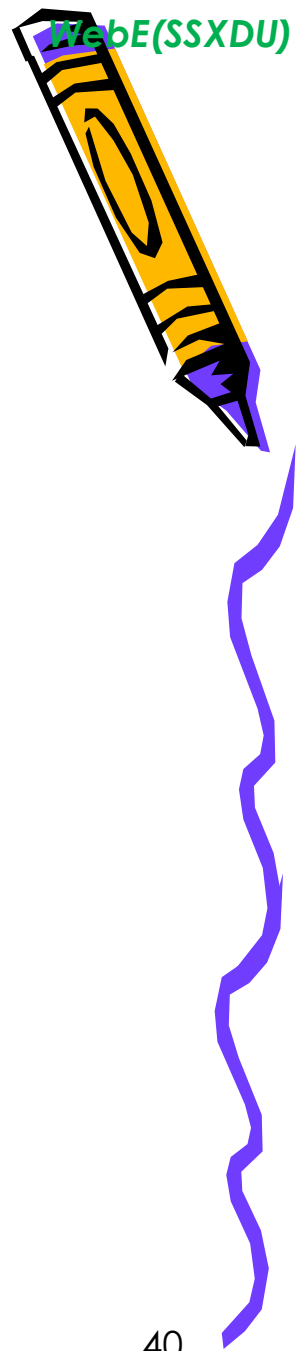
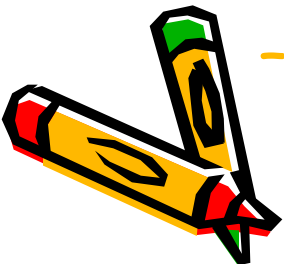
页面布局技术

- 表格布局
 - `<table></table>`: 最早且最广泛
- 框架布局
 - `<frame></frame>`: 对于导航或不动的窗口在浏览Web应用时只需加载一次 BUT...
- DIV+CSS
 - CSS用于控制Web页面样式
 - DIV元素用来为HTML文档内大块内容提供结构和背景的元素
 - 目前Web设计中**推荐**的方式



主页布局设计

- 追求画面美观效果（静态）
 - 适合内容不多的企业Web应用
- 追求内容丰富效果（动态）
 - 适合内容丰富的综合Web应用
- 设计原则
 - 机构名称、标识、品牌位置显著
 - 主题风格鲜明
 - 内容明确
 - 易用可读
 - 提供站内搜索



页面元素设计

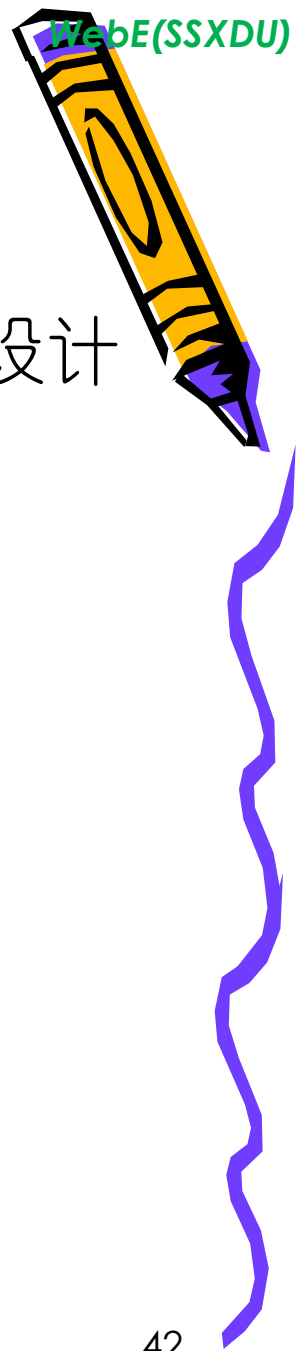
- 将LOGO、图标、文本块、图片等及其组合等页面元素遵循设计原则进行组织，在二维空间中进行布局。
 - 点、线、面、空间
 - 链接导航
 - 图片与动画
 - 文字
 - LOGO



6 reasons your logos fail

美学设计

- 美学设计一种艺术，是对页面设计和内容设计技术方面的补充。
- 目标：令人愉悦→吸引力
- 关注多方面
 - 文字
 - 更富创意的形式表达出深层的设计思想
 - 色彩
 - 唤起人类的心灵感知，色相、饱和度和亮度
 - 风格与创意
 - 具有代表性的独特面貌



展示设计原则

- Web页面展示

- 主题鲜明
- 统一
- 平衡
- 连贯
- 分割
- 对比
- 和谐
- 留白
- 无障碍设计

- Web页面的浏览速度和页面的适应性

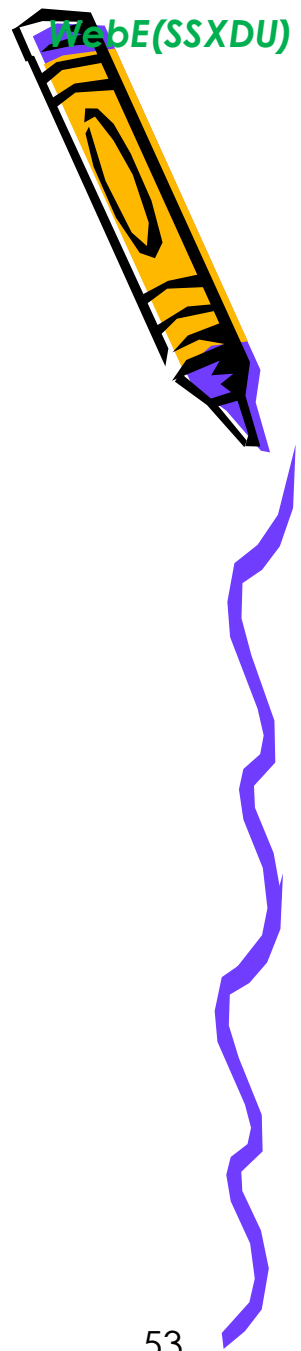
- 源代码优化
- 图像优化
- 表格优化
- 搜索引擎优化 (SEO)
- 目录结构清晰
- 避免使用框架
- 文字优化



展示设计工具

- Web页面制作和Web应用管理Web页面编辑器
 - 美国MACROMEDIA的Dreamweaver
 - 8款替代Dreamweaver的开源网页开发工具
 - 微软的FrontPage
 - 浏览器
- 可视化的Web应用的构建和管理
 - NetObject Fusion, Adobe 系列, CorelDraw 图形套件,



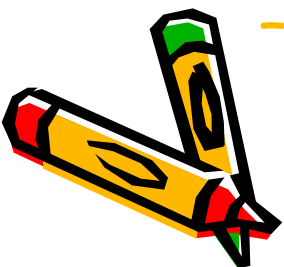


内容设计



内容设计

- “内容为王”
- 提供哪些可用内容（组件）及其组成和导航（网），为用户提供的视图
- 目标
 - 把通常以一组详细的内容对象表示的内容需求转化为Web应用具体的信息设计
 - 关注如何组织、访问和管理这些内容
 - 定义所有内容的布局、结构和作为Web应用的一部分进行展示的内容大纲
 - 建立内容对象之间的关系



信息设计方法

- 自底向上的信息设计
 - 仅涉及实际Web页面的设计和构建，并逐步把这些Web页面链接起来
 - 应用于小型的Web应用
- 自顶向下的信息设计
 - 强调Web应用内主要内容种类的总体组织、相互关系和结构
 - 信息架构



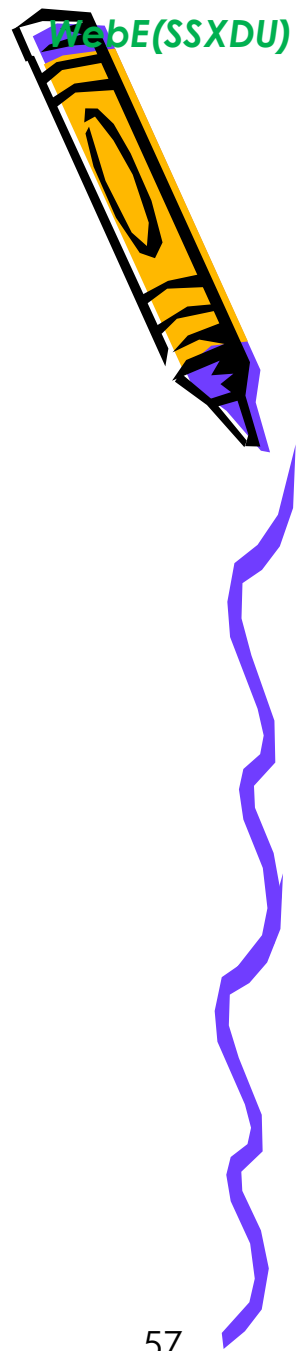
信息架构

- 信息架构 (Information Architecture, IA) 是信息空间结构的高层设计, 涉及组织、标识、导航和搜索的设计。
- 目的: 帮助人们在网络和Web环境中更成功地发现和管理信息, 有效地解决用户的信息需求。
- 主要任务是对信息进行分类、归类, 以及组织



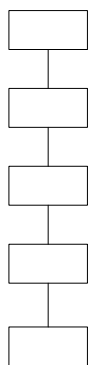
信息架构的特性

- 与多种动态数据进行组合
- 高层表示的规格说明
- 时间关系
- 链接的上下文和链接语义
- 内容与信息分离
- 信息和应用分离
- 应用和展示的分离

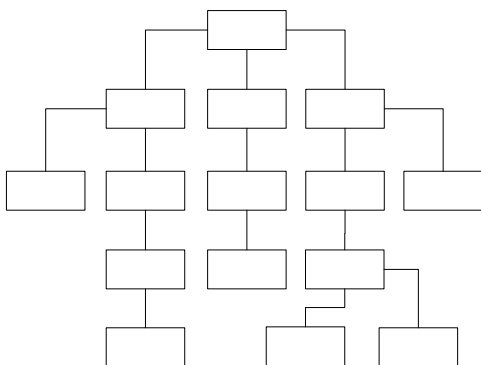


组织内容

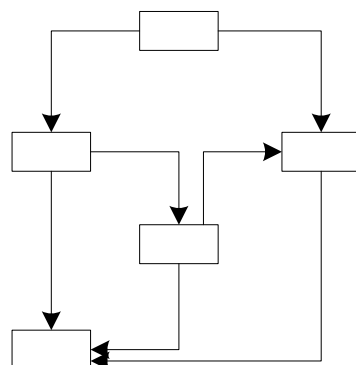
- 目的
 - 信息**可**获取
 - 信息**易**获取
- 组织方式
 - 线性结构、层次结构、网络结构、矩阵结构



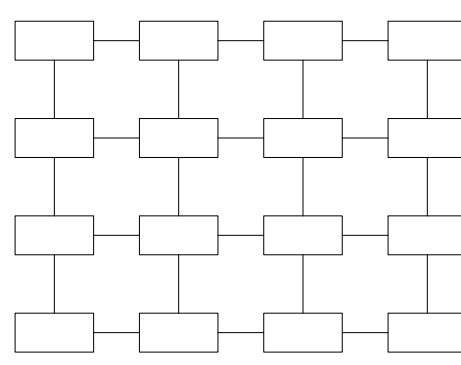
线性结构



层次结构



网络结构



矩阵结构

访问信息

- 影响因素
 - 使用户在任何给定的时间知道哪些导航选项是可用的机制；
 - 为用户提供说明他们在哪里以及他们正在看什么的界面机制；
 - 允许用户在信息结构内游走的导航机制。



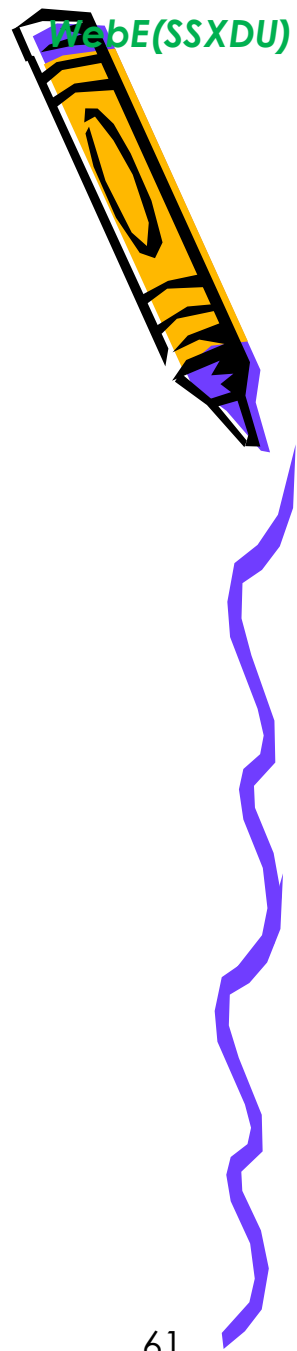
导航机制

- 全局导航
 - 对整个Web应用内容进行导航
- 局部导航
 - 全局导航的下层内容进行细分,一个内容域
- 语境导航
 - 语境:上下文相关的内容
- 补充导航
 - 在页面内容之外的方式向用户提供进入内容的途径

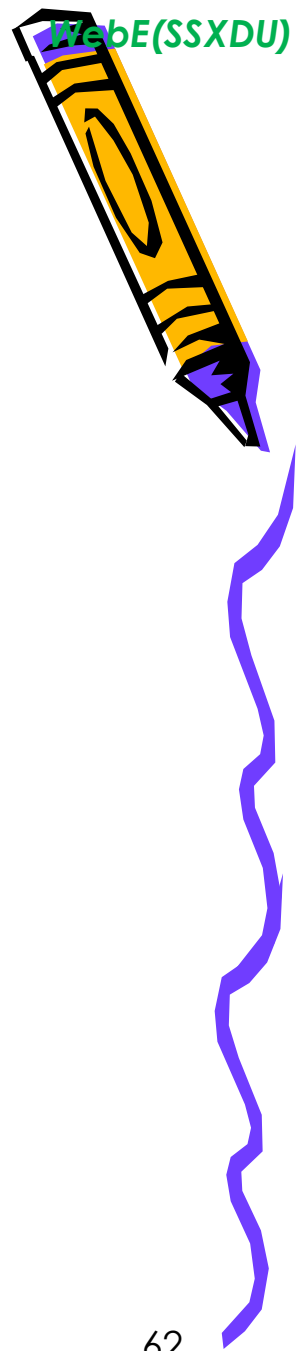


导航机制的原则

- 明确标记
- 全局链接
- 快捷方式
- 层级（Breadcrumbs）和轨迹
- 个性化
- 搜索机制



功能设计



功能设计

- 主要关注Web应用所支持的行为和功能
 - 如 workflow 支持、内容和界面自适应和（或）定制、订单录入、数据库处理、计算功能等
- 从用户功能设计开始，作为Web应用功能建模的延续
 - 以交互模型和功能模型来描述，如序列图、状态图和活动图等
- 和信息设计并行交织进行，两者之间相互依赖
- **技术对Web应用开发的影响**
 - 可扩充性、可伸缩性和可维护性
 - 组件之间的相互作用



集成

- 集成的三个功能层次
 - 数据层
 - 不同应用的数据之间可以进行转换和复制
 - 应用层
 - 时间和语义密切交织，依赖于**中间件**
 - 过程层
 - 根据基础架构单独对业务模型进行建模

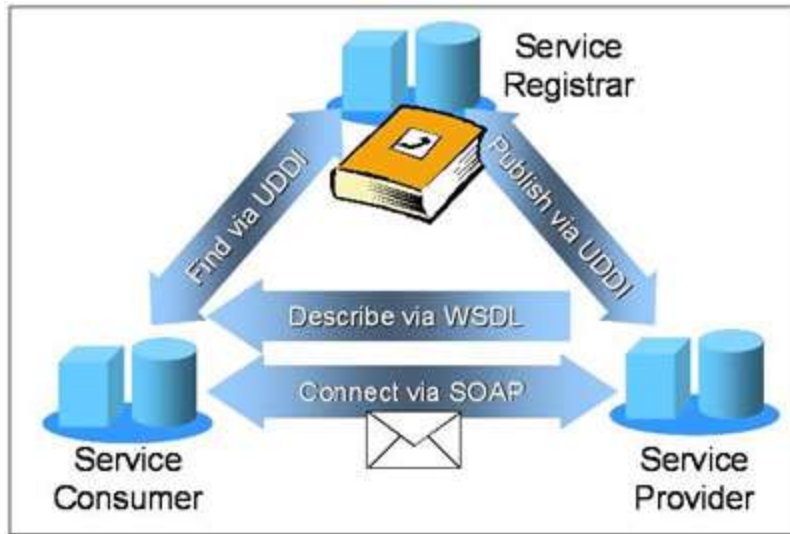


分布式Web应用

- 分布式软件访问远程Web应用，可以看作服务与服务交互。
 - 服务通过定义好的接口方式提供
 - 涉及企业内部应用、第三方应用或服务
 - 例：eBay不仅有单点认证，还支持微软的Passport；Google可以在外部应用中通过SOAP集成其搜索引擎。
 - Web服务
 - XML和SOAP、WSDL、UDDI、BPEL等协议

分布式Web应用

- Service-to-service communication



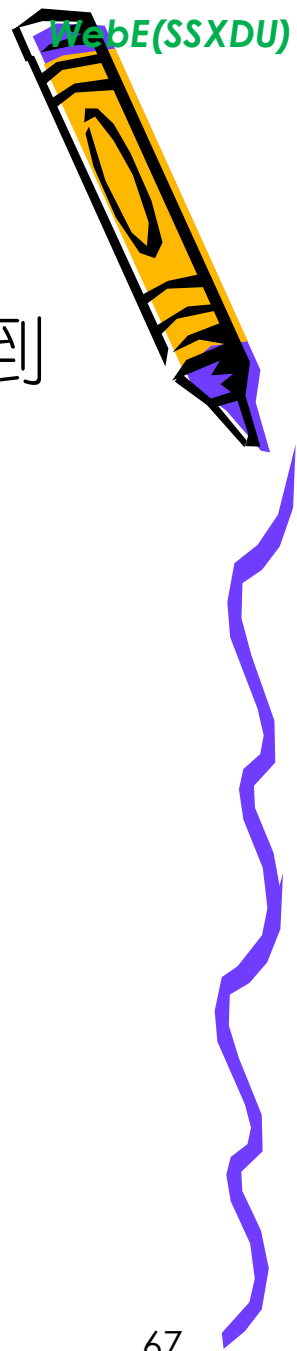
Basic Web Service Protocols

WS-Security	WSCI	BPEL	UDDI
	WS-Transaction		
	WS-Coordination		
	WSDL		
	SOAP		

Protocol Stack for Web Services

功能设计原则

- 以需求为导向，切合实际，操作简便，达到“实用、简单、好用、耐用”的要求。
- 常用原则
 - 先进性原则
 - 可用性原则
 - 实用性原则
 - 可扩展性原则



总结与展望



总结

- Web应用的立足之本——最大限度地满足客户的需要
 - 以用户为中心
 - 有机结合交互设计、**展示**设计、信息设计和功能设计
 - 采用可扩展的适应性好的技术

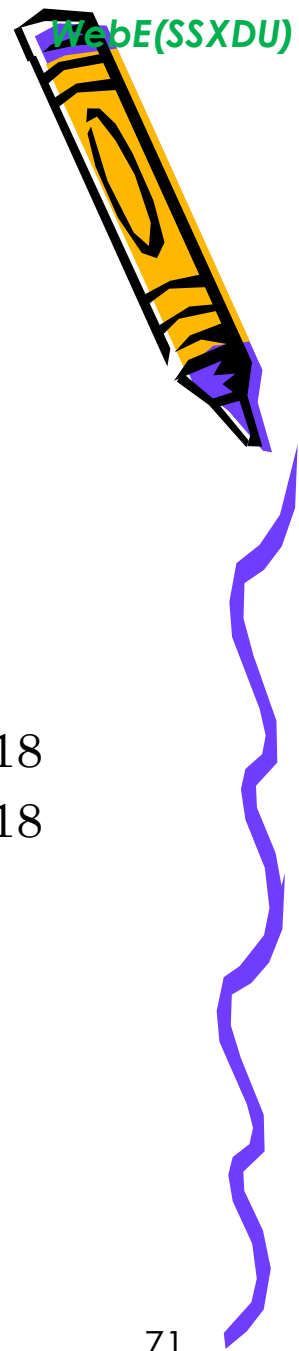


展望

- 上下文感知和设备无关性
 - 基于位置等上下文的服务
 - 移动设备
 - 采用平台无关的协议描述并结合Web服务
 - 尽量提供满足最小设备的应用展现，基于CC/PP (DIWG, W3c 2001c) 标准提供上下文描述并适合各种不同设备
- 重用性
 - 网由元素和链接组成不足以表达Web应用的软件和信息设计，需要特殊的聚合概念
 - 需要进一步设计出统一的设计符号
 - 新的组合概念，如基于事件的通信以及多维链接的超文本概念

Web Design Trends

基于HTML5的设计pptx



<https://www.theedesign.com/blog/2017/top-web-design-trends-2018>

<https://www.theedesign.com/blog/2017/top-web-design-trends-2018>

<https://webflow.com/blog/19-web-design-trends-for-2018>

TRENDS



Project Task: Task6

- Web应用设计
 - 结合本章内容，完成Web应用设计
- 展示的内容之一
 - ~2 slides

