

# 复习内容

状态 未填写

创建时间 未填写

---

## 一、概述

### 什么是人机交互（HCI）？

- 人机交互是一门学科，其关注点为：设计、评估与实现供人类使用的交互式计算系统，并研究与这些系统相关的主要现象。
- 本质目标是解决“终端用户与技术之间的最后一公里”

### 人机交互研究什么？哪些工作属于 HCI

#### HCI 的研究内容（四个维度）

##### （1）人（Human）

- 人类信息处理
- 语言、交流与认知
- 人因工程 / 工效学

##### （2）计算机（Computer）

- 输入 / 输出设备
- 对话技术、对话类型
- 计算机图形学

##### （3）使用上下文（Context）

- 应用领域
- 社会组织与工作方式
- 人机匹配与适应

#### (4) 开发过程 (Development)

- 设计方法
- 实现技术与工具
- 评估技术
- 示例系统与案例

### 哪些工作属于人机交互?

- 交互式系统的界面设计
- 可用性评估
- 用户研究
- 人因工程
- 用户体验研究

## 三、人机交互与相关领域的关系【对比题】

### HCI、用户体验 (UX)、交互设计 (IxD)

概念	侧重点	特征
人机交互 (HCI)	学术 + 工程	偏计算机科学、研究
用户体验 (UX)	整体感受	覆盖使用前中后
交互设计 (IxD)	产品与服务	面向交互流程

### 关于用户体验的关键观点 (非常爱考)

- 用户体验是人与生俱来的感受能力
- 不能设计用户体验，只能为用户体验而设计

## 二、HCI历史

### 一、人机交互发展的总体规律

#### HCI发展的基本观点

- 新的界面变革并不会完全取代旧界面
  - 新界面通常包含上一代界面（作为特例）
  - 旧的交互方式仍然有存在价值
- 原因：
  - 以前的用户从未消失
- 学习历史的目的：
  - 利用已有技术实现新的交互方式

### 二、人机交互的主要发展阶段

#### 四个主要发展阶段

##### (1) 批处理阶段 (Batch Processing)

- 特点：
  - 一次只能一个用户使用
  - 使用机器语言（0/1）
- 问题：
  - 不符合人的认知习惯
  - 易错、效率低
  - 只有少数专业人员能使用
- 代表系统：
  - ENIAC (1946)

---

## (2) 联机终端 / 命令行阶段 (CLI)

- 时间：20世纪50年代
- 特点：
  - 一维界面
  - 回车后不能修改命令
- 问题：
  - 命令严格
  - 学习成本高
- 改进方式：
  - 命令缩写降低负担

---

## (3) 图形用户界面阶段 (GUI)

- 关键事件：
  - 1962: Sutherland 的 Sketchpad
  - 1964: Engelbart 发明鼠标
- 主要特征：
  - WIMP 界面
  - 二维半界面
  - 直接操纵 (Direct Manipulation)
- 重要观点：
  - GUI 并非天然优于 CLI
  - 可用性差异更多来自设计态度而非交互方式本身

---

## (4) 多模态 / 智能交互阶段

- 特点：

- 多媒体界面
  - 多通道交互（并行）
  - 虚拟现实、语音交互、脑机接口
- 趋势：
    - 从“命令”走向“自然交互”

### 三、过去 → 现在 → 未来

#### 未来的人机交互趋势

- 图形界面受到批评：
  - “one ear, one finger, one eye”
- 发展方向：
  - 多媒体界面
  - 多通道交互
  - 虚拟现实、语音、脑机交互
- 总体趋势：
  - 没有命令的用户界面
  - 更自然、更便携、更个性化

### 三、设计原则和目标

#### 一、执行—评估周期模型 (EEC)

##### 交互框架的作用

交互框架用于：

- 提供理解或定义事物的结构
- 帮助设计者结构化设计过程
- 识别设计过程中的主要问题

- 明确问题所涉及的领域
- 

## 执行 / 评估活动周期 (EEC)

- HCI 中最有影响力的框架之一
  - 从用户视角分析人机界面问题
  - 定义了交互活动的四个核心组成部分：
    - 目标 (Goal)
    - 意图 (Intention)
    - 执行 (Execution)
    - 评估 (Evaluation)
- 

### 目标 (Goal) vs 意图 (Intention)

- 一个目标可以对应多个意图
  - 例：
    - 目标：删除文档中的一段内容
    - 意图1：通过菜单删除
    - 意图2：通过按钮删除
- 

### EEC 七阶段模型

- 共 7 个阶段
    - 1–4：执行阶段
    - 5–7：评估阶段
  - 每一次循环代表用户完成的一个动作
- 

## 二、执行隔阂 & 评估隔阂 【必考·简答 / 论述】

### 执行隔阂 (Gulf of Execution)

- 含义：
    - 用户为达成目标而制定的动作
    - 与系统允许的动作之间的差别
  - 本质问题：
    - 用户不知道“**我能做什么**”
- 

## 评估隔阂 (Gulf of Evaluation)

- 含义：
    - 系统的实际状态
    - 与用户**预期状态之间的差别**
  - 本质问题：
    - 用户不知道“**系统现在在干什么**”
- 

## 两个隔阂的意义

- 帮助设计者思考：
  - 用户如何确定哪些操作是被允许的
  - 用户如何判断系统是否处于期望状态

## 三、可用性目标 (相对客观)

### 可用性目标的作用

- 为交互设计人员提供评估产品和用户体验的具体方法
- 

### 五个可用性目标

1. 易学性 (Learnability)
2. 易记性 (Memorability)
3. 高效率 (Efficiency)

4. 效用性 (Utility)

5. 安全性 (Safety)

## 四、用户体验目标

### 用户体验目标的含义

- 覆盖一系列：
  - 情绪
  - 感受
  - 心理体验
- **高度依赖使用情境**
- 同一产品在不同时间、地点 → 不同体验
- 游戏、娱乐产品常牺牲可用性换取体验

## 五、简易可用性工程（重点）

### 1 8 简易可用性工程的特点

- 以人为中心
- 强调评估与迭代
- 工业界证明投入产出比极高 (IBM、Microsoft)

---

### 1 9 四项关键技术（大纲重点）

1. 用户和任务观察
2. 场景 (Scenario)
3. 简化的边做边说 (Think Aloud)
4. 启发式评估

## 六、设计原则【必考大题】

### 三类设计原则体系

- 一般设计原则 (Alan Dix)
  - Shneiderman 八大黄金法则
  - Nielsen 十项启发式原则
- 

## 八大黄金法则

1. 尽可能保证一致
  2. 符合普遍可用性
  3. 提供信息丰富的反馈
  4. 明确结束状态
  5. 预防并处理错误
  6. 支持撤销
  7. 支持内部控制点
  8. 减轻短时记忆负担
- 

## Nielsen 十项启发式原则

- 系统状态可见
  - 贴近现实世界
  - 用户控制权
  - 一致性与标准
  - 防错
  - 识别优于记忆
  - 灵活高效
  - 简约美观
  - 错误恢复
  - 帮助与文档
- 

## 黄金法则和启发式原则对应

一致性 → 一致性和标准  
反馈 → 系统状态可见  
防错+纠错 → 避免出错 + 错误恢复  
撤销/掌控 → 用户控制权  
内部控制点 → 用户控制权  
减轻记忆负担 → 识别而非记忆  
普遍可用性 → 灵活性和高效性  
结束状态 → 简约设计（弱）  
帮助文档 → Nielsen 独有

## 四、需求

### 人物角色 (Personas)

#### 什么是人物角色

- 不是真实的人
- 基于真实用户行为和动机
- 是用户的综合原型
- 用来在整个设计过程中代表真实用户

---

#### 人物角色的作用（三个设计问题）

1. 避免“弹性用户”
2. 防止“自参考设计”
3. 不被边缘情况主导

---

#### 人物角色构造的核心问题

- 谁将使用系统？
- 属于哪类人群？
- 使用动机与目标？
- 与软件的关系？
- 期望系统提供什么支持？

## 用户特性

### 按使用经验划分的用户特性

用户按经验分为三类：

用户类型	典型特性	设计关注点
新手用户	易挫败、不熟悉系统	易学性、引导
中间用户	使用最频繁、最稳定	效率 + 支持
专家用户	熟练、追求效率	快捷方式、高级功能

## 任务分析 (HTA)

### 任务分析是什么

- 记录人们如何完成任务
- 用于理解现有流程
- 建立需求

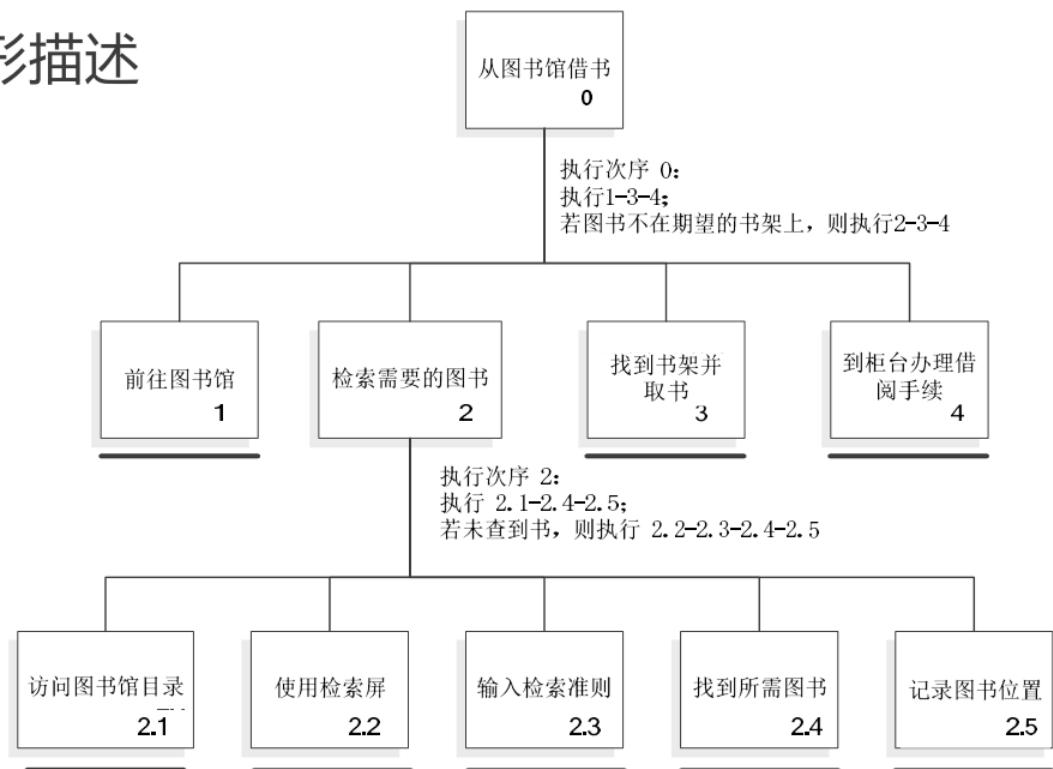
---

### 层次化任务分析 (HTA)

- 将任务不断分解为子任务
- 定义执行次序

- 说明实际操作流程
- 终止点
  - 任务包含了复杂机械响应的地方
    - 如鼠标移动
    - 此时分解没有价值
  - 涉及内部决断的地方
    - 若决断和查找文档等外部动作相关，则分解
    - 若为纯粹认知性，则终止

## ■HTA的图形描述



## 五、设计

### 一、设计框架

#### 什么是设计框架

设计框架是将需求转化为具体交互方案的整体结构方法，用于指导：

- 功能如何组织

- 界面如何布局
  - 用户如何完成任务
- 

## 交互式系统的设计框架

### (1) 定义外形因素与输入方法

- 决定系统运行环境：
    - Web / 移动 / 桌面 / 穿戴
  - 决定主要输入方式：
    - 触控 / 键盘 / 语音 / 手势
  - 依据：
    - 用户特性
    - 使用场景
  - 输入方式是设计框架的一部分，而不是实现细节
- 

### (2) 定义功能和数据元素

- **数据元素：**系统中被操作的对象
    - 例如：文件、照片、订单
  - **功能元素：**对数据的操作
    - 例如：创建、删除、编辑
  - 功能 ≠ 按钮，功能是“做什么”，不是“怎么点”
- 

### (3) 决定功能组合层次（信息组织）

- 哪些功能经常一起使用？
- 是否存在先后顺序？
- 是否需要分组或分层？

常见原则：

- 一起用的放一起
- 有顺序的按流程组织
- 避免跨页面频繁跳转

---

#### (4) 勾画大致设计框架（低保真）

- 使用：
    - 方块图
    - 草图
  - 表示：
    - 区域
    - 功能分布
  - 不画细节、不选颜色、不选字体
- 

#### (5) 构建关键场景剧本

- 使用人物角色（Persona）
  - 描述完成关键任务的完整过程
  - 关注：
    - 主路径（Key Path）
- 

#### (6) 用验证性场景检查设计

- 检查：
    - 其他操作方式是否合理
    - 异常情况是否可处理
    - 边缘情况是否被忽略
- 

## 二、细节可能会考的地方

### 功能 vs 控件（非常容易考混）

- 功能 ≠ 按钮
  - 功能 = 用户想完成的事情
  - 控件只是功能的一种实现方式
- 

高保真 ≠ 好设计

- 高保真原型：
    - 接近最终外观
  - 但：
    - 不能替代结构思考
- 

## 设计是迭代的，不是线性的

- 设计框架不是“一次走完”
  - 场景 → 发现问题 → 回退调整
- 

## 三、四种策略的区别 & 分别什么时候用

### 总览表

策略	核心思想	解决什么问题	什么时候用
删除	不该有的就不要	功能过多	几乎没人用
组织	把东西放对位置	混乱	结构不清
隐藏	暂时不展示	认知负担大	低频功能
转移	换地方/换人做	平台不合适	复杂任务

---

### (一) 删除 (Delete)

#### 是什么

- 删掉不必要的功能或内容

#### 解决的问题

- 功能堆砌
- 学习成本高

#### 什么时候用

- 功能几乎没人使用

- 功能不完整、体验差
  - 为“以后可能有用”而保留的功能
- 

## (二) 组织 (Organize)

是什么

- 对功能和内容进行分组、分层、排序

解决的问题

- 信息杂乱
- 用户找不到东西

什么时候用

- 功能本身有价值，但结构混乱
- 页面内容过多

常用方法：

- 分区
  - 层级结构
  - 7±2 原则
- 

## (三) 隐藏 (Hide)

是什么

- 不常用功能暂不展示
- 需要时再出现

解决的问题

- 初学者认知负担过重

什么时候用

- 功能对部分用户有用
  - 但不是所有人都需要
  - 专家功能 / 配置项
-

## (四) 转移 (Transfer)

是什么

- 把复杂任务转移到：
  - 更合适的平台
  - 或让用户在系统外完成

解决的问题

- 当前平台不适合完成复杂操作

什么时候用

- 屏幕小 (手机)
- 输入受限
- 操作复杂

# 六、评估

## 一、评估范型

四种评估范型

范型	特点	关键词
快速评估	快、非正式	早期
可用性测试	量化、受控	实验室
实地研究	真实环境	观察
预测性评估	用户可不在场	专家

## 二、观察用户 (第七章·方法题核心)

为什么要观察用户？

- 用户：
    - 说的不一定是真的
    - 会忽略细节
  - 观察能发现：
    - 意料之外的行为
    - 使用环境对交互的影响
- 

## 两种观察方式

类型	特点
实验室观察	可控、可比较
现场观察	真实、复杂

---

## 边做边说（Think-aloud）

- 用户在操作时说出想法
  - 优点：
    - 理解思考过程
  - 缺点：
    - 不自然
    - 可能影响行为
- 

## 观察的局限

- 看得到行为
  - 看不到原因
- 👉 常与访谈结合

## 三、询问用户 & 专家（第八章·对比题高频）

### 访谈（Interview）

## 类型

- 非结构化
- 半结构化（最常用）
- 结构化
- 集体访谈（焦点小组）

## 原则

- 避免诱导性问题
  - 问题清晰、简短、中性
- 

## 问卷调查（Questionnaire）

### 适合

- 大规模
- 定量分析

### 常见题型

- Likert 量表
- 语义差异度

### 典型问卷

- SUS（系统可用性量表）
  - NASA-TLX（工作负荷）
- 

## 认知走查（Cognitive Walkthrough）

### 是什么

- 专家模拟用户完成任务
- 逐步检查是否易学

### 关注三个问题

1. 用户知道该做什么吗？
2. 能发现正确操作吗？
3. 能理解反馈吗？

## 优缺点

- 不需要用户
  - 只适合评估易学性
- 

## 启发式评估 (Heuristic Evaluation)

- 基于 Nielsen 十项原则
  - 由专家完成
  - 关注问题严重性分级：
    - 表面
    - 次要
    - 主要
    - 灾难性
- 

## 四、用户测试

### 什么是用户测试？

| 在受控环境中，让典型用户执行典型任务，获取客观性能数据的评估方法。

---

### DECIDE 评估框架

步骤	含义
D	决定目标
E	发掘问题
C	选择方法
I	明确实际问题
D	处理道德问题
E	解释和表示数据

---

## 用户测试的关键要素

- 参与者：
    - 具有代表性
  - 任务：
    - 与目标相关
    - 5–20 分钟
  - 数据：
    - 定量 + 定性
- 

## 常见定量指标

- 任务完成时间
  - 错误次数
  - 成功率
  - 求助次数
  - 满意度评分
- 

## 实验设计中的经典对比

设计	特点
组间设计	无顺序效应
组内设计	消除个体差异
配对设计	控制关键变量

# 七、认知

## 一、人类处理机模型

## 三个处理器

处理器	作用
感知处理器	接收视觉、听觉信息
认知处理器	在工作记忆中处理信息
动作处理器	执行动作

## 模型的局限

- 只关注：
  - 单人
  - 单任务
- 忽视：
  - 多人协作
  - 环境影响
- 引出：
  - 分布式认知
  - 外部认知

## 二、格式塔心理学

### 格式塔心理学的核心思想

| 整体不同于部分之和

用户会：

- 主动寻找“好”的结构
- 自动分组和组织信息

## 五个格式塔原则

### (1) 相近性原则

- 空间上接近 → 被视为一组
- 应用：
  - 相关控件放一起

### (2) 相似性原则

- 外观相似 → 被视为一组
- 应用：
  - 功能相近用相同样式

### (3) 连续性原则

- 共线、同方向 → 被组合
- 应用：
  - 对齐增强理解

### (4) 完整与闭合性原则

- 人会补全不完整图形
- 应用：
  - 留白实现分组

### (5) 前景与背景

- 前景/背景可互换
- 应用：
  - 注意对比，避免歧义

---

## 三、记忆模型

### 三种记忆 (Atkinson & Shiffrin)

类型	特点
感觉记忆	极短 ( $\approx 1$ 秒)

短时记忆 / 工作记忆	≈30 秒，容量有限
长时记忆	容量几乎无限

## 7±2 原则（考试一定会出现）

- 短时记忆容量有限
- 启示：
  - 减少记忆负担
  - 用识别代替回忆
  - 利用分组（Chunking）

## 四、界面类型（结合发展历史）

### 命令行界面（CLI）

- 优点：
  - 高效
  - 节省资源
- 缺点：
  - 学习成本高
- 适合：
  - 专家用户

### WIMP / GUI

- Window / Icon / Menu / Pointing
- 特点：
  - 少记忆
  - 多识别
  - 上下文可见

## 多媒体界面

- 文本 + 图像 + 视频 + 音频

- 优点：

- 提高参与感

- 问题：

- 何时用音频？

- 何时用动画？

---

## 新型界面（了解即可）

- VR / AR
- 信息可视化 / 仪表盘
- 触摸 / 手势
- 实物界面
- 可穿戴计算
- 脑机接口

# 八、模型和理论

## 一、击键层次模型 KLM

### KLM 是什么？

- GOMS 的具体化版本
- 只预测时间
- 面向：
  - 无错误的专家用户

---

### KLM 的基本操作符（必须认识）

操作符	含义
K	键击
P	指向目标
H	手移到设备
D	拖动
M	心理准备
R(t)	系统响应时间

## KLM 的时间预测公式

|  $T = \sum$  各操作符时间

例子中常见写法：

▼  
1  $T_{execute} = T_k + T_p + T_h + T_d + T_m + T_r$

## KLM 的局限

- 不考虑：
  - 错误
  - 学习
  - 疲劳
  - 接受度

只能用于“理想情况”比较

## 五、Fitts 定律

Fitts 定律解决什么问题？

- 预测：
    - | 用户指向某个目标所需的时间
  - 与：
    - 目标大小 (W)
    - 目标距离 (A)
  - 有关
- 

## Fitts 定律核心公式

### 困难指数

$$ID = \log_2(W/A)$$

### 运动时间

$$MT = a + b \cdot ID$$

---

## 定律结论（

- 目标越大 → 越容易点
  - 距离越近 → 越快
  - 同样 ID → 同样难度
- 

## 对界面设计的启示

- 增大可点击区域
- 缩短指针移动距离
- 利用屏幕边缘（无限目标）
- 饼形菜单优于线性菜单

### 经典案例

- Mac OS 菜单在屏幕顶端
- Windows 菜单在窗口内部

## 九、以用户为中心

倾听用户的意见（不等于用户做决定）

局限性（可能无法带来超出用户期望的产品）