HCI 2025

Analyzing and Synthetizing

Human Computer Interaction

Luigi De Russis

Academic Year 2025/2026

Hall of Fame or Shame?

User Needs Edition

Beware: we are missing the general context here!

- Users need a faster horse
- Users need to have financial help
- Users needs a way to move faster from one place to another
- Users need to have more tools
- Users need to practice more with the appropriate tools
- Users need to be able to run faster

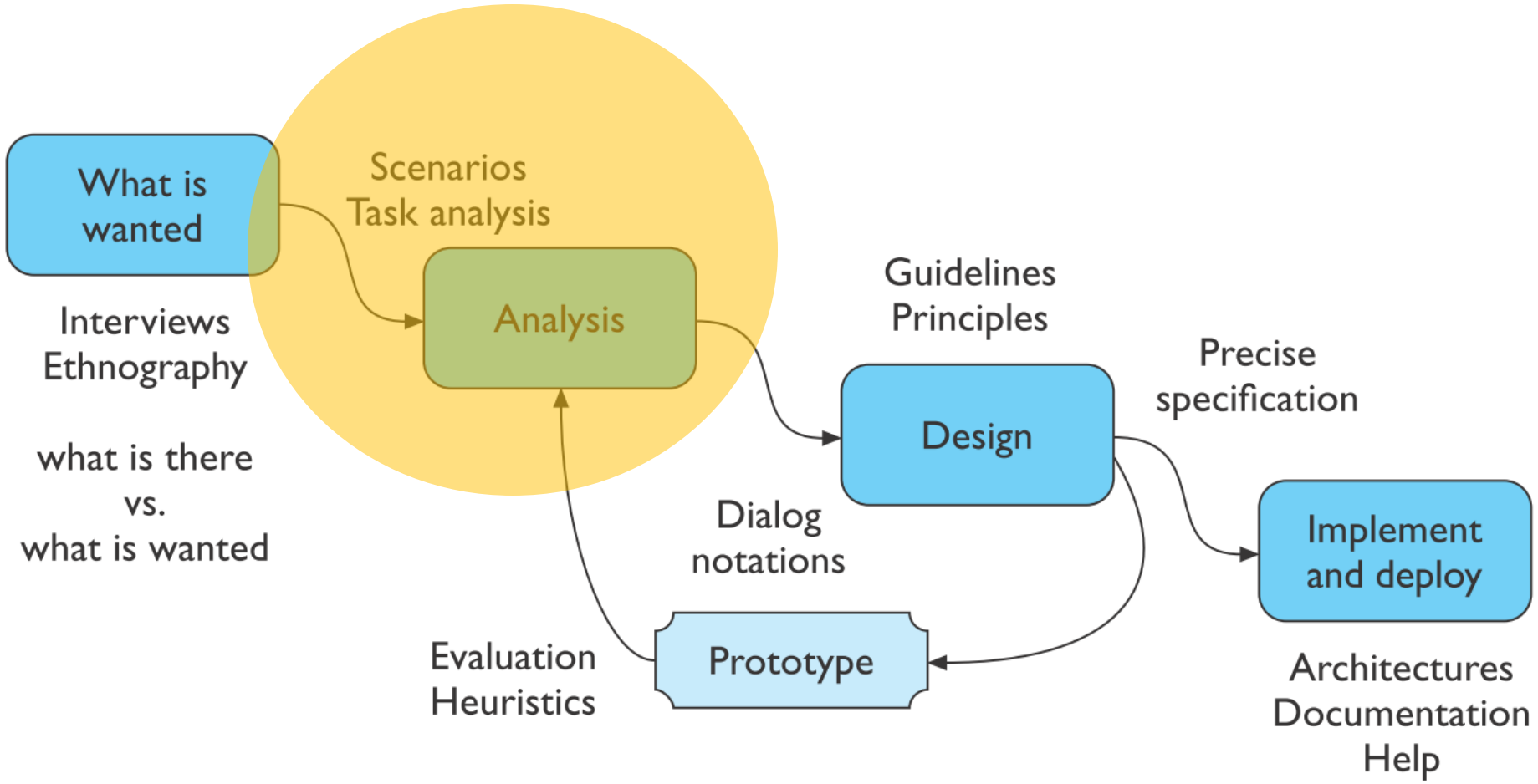
From Assignment 1:

"Needs are human emotional or physical necessities. […]

Needs are verbs (activities and desires with which your user could use help), not nouns (solutions). […]

It can be helpful to use the phrases 'needs a way to' or 'needs to be able to' in your list of user needs."

Human-Centered Design Process



Goals

- Create **design goals**
  - As an intermediate representation before the user interface design
- Make the user needs' analysis **explicit**
- Think about the *interplay* between the activity that someone has and the interface we offer
- **Represent** and synthetize the results of the analysis and the design goals

Tasks and Their Analysis

How people perform their activities

Tasks

- Task: the structured **set of activities**/high-level actions required to **achieve** a user goal
  - It says what a person *wants to do*, not how, while describing a *complete* goal
- Often, given a domain, you have a **mix** of tasks with different **complexity**
  - Simple tasks – common or introductory
  - Moderate tasks
  - Complex tasks – infrequent or for power/extreme users

Task Analysis

- Task analysis is the study of the way people perform their activities
- Aim is to determine:
  - what they **do** (steps)
  - what things they **use** (artifacts)
  - how well they **succeed** (goals, pain points)

Sample Task: To Clean The House (I)

- **Steps**:
  - get the vacuum cleaner out
  - fix the appropriate attachments
  - clean the rooms
  - when the dust bag gets full, empty it
  - put the vacuum cleaner and tools away
- Must know and use different **artifacts**:
  - vacuum cleaners, their attachments, dust bags
  - cupboards, rooms
  - …

Sample Task: To Clean The House (II)

- **Goals**:
  - Here your *point of view* comes in
  - Removing dust? -> narrow goal
  - Tidying up the house after a party?
  - Hosting people for the dinner?
  - Having a satisfying evening? -> wide goal

Sample Task: To Clean The House (III)

- **Pain points**:
  - Narrow version: Why I need to empty the dust bag?
  - Broader version: Why I need a vacuum cleaner to have the house cleaned up?

What is a Tasks?

- «A **task** is a **goal** together with some ordered set of **actions**.» (Benyon)

Goal

- A state of the application domain that a work system (user+technology) wishes to achieve.
- Specified at particular levels of abstraction.

Task

- A structured set of activities required, used, or believed to be necessary by an agent (human, machine) to achieve a goal using a particular technology.
- The task is broken down into more and more detailed levels of description until it is defined in terms of actions.

Action

- An action is a task that has no problem solving associated with it and which does not include any control structure.
- Actions are 'simple tasks'.

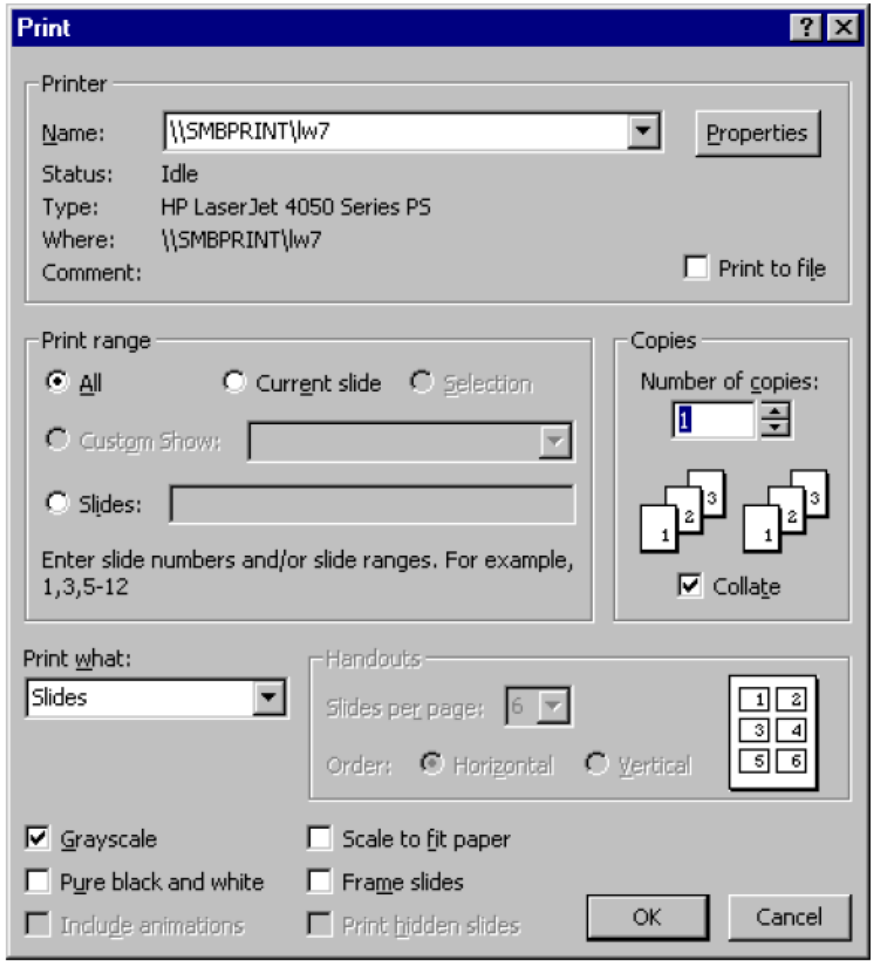
What You Learn with Task Analysis

- What your users' goals can be; what they are trying to achieve
- What users actually do to achieve those goals
- What experiences (personal, social, and cultural) users bring to the tasks
- How users are influenced by their physical environment
- How users' previous knowledge and experience influence:
  - How they think about their work
  - The workflow they follow to perform their tasks
  - The pain points they experience to perform the tasks

Why Is It Useful?

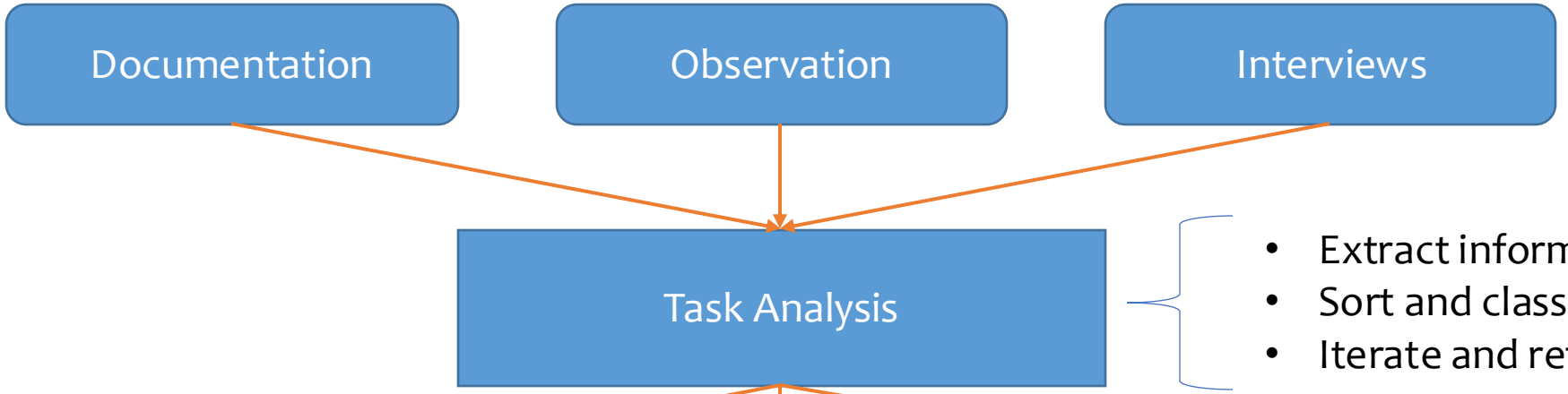
- Task analysis is the process of learning about ordinary users by observing them in action to understand in detail **how** they perform their **tasks** and achieve their intended **goals**
- Tasks analysis helps in:
  - Identifying the tasks that your application **must support**
  - Refining or re-defining your app's navigation or search
  - Application requirements gathering
  - Developing your content strategy and app structure
  - The initial stages of Prototyping
  - Performing usability testing

Example



- Tasks are used to plan for the layout of the application window
- Proximity and Boundaries reflect the decomposition of tasks
- Order of tasks is not mandatory

Where It Fits



Manuals and Documentation

- Conceptual Manuals (from KBA)
- Procedural (how-to) Manuals (from HTA)

Requirements ans systems design

- Focus on system usage rather than system features
- Suggests candidates for automation
- Unconvers user's conceptual model

Detailed interface design

- Taxonomies → menu layout
- Objects/actions → interface objects
- Task frequency → default choices

Characteristics of Task Analysis

- Task analysis is easier when you have well-defined **workflows** (e.g., planning a trip somewhere)
  - or **repeated activities**, such as scheduling
- Challenge:
  - We **do not** design tasks, but interfaces
  - Tasks and objects do not map 1:1
    - e.g., a web app has multiple tasks
  - People use the same interface and application to achieve slightly different results or do things differently one another

[Some] Techniques for Task Analysis

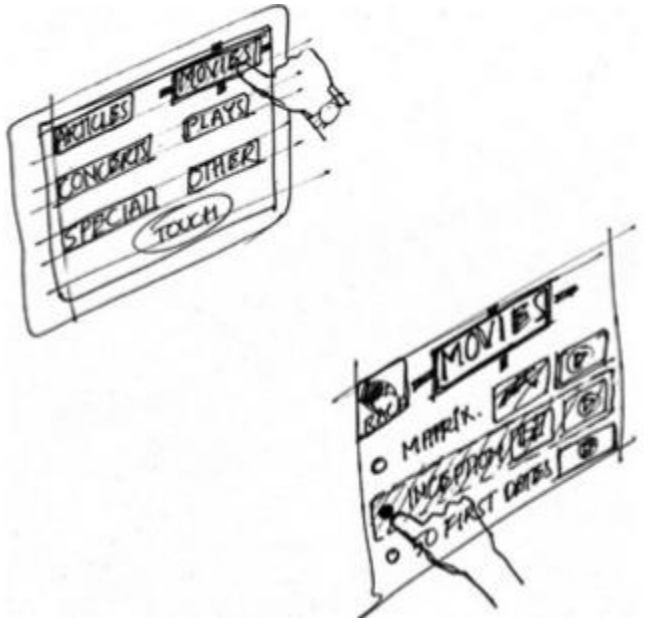
- **Task decomposition** – Splitting tasks into sub-tasks and their ordering
- **Knowledge-based techniques** – Any information and instructions that users need to know, and how that knowledge is organized
- **Entity-relationship-based analysis** – Identify actors, objects, relationships and their actions
- **Ethnography** – Observation of users' behavior in the use context
- **Protocol analysis** – Observation and documentation of actions of the user. This is achieved by authenticating the user's thinking. The user is made to think aloud so that the user's mental logic can be understood.

Sketches

Quick drawings to convey a part of the interface, or a feeling about a device

Sketch

- An individual drawing showing
  - A single user interface screen
  - A drawing of an artifact part of the system
  - The shape of an interaction object



- Gives a static view of a possible interaction
- Helps setting the interaction context
- Often, part of a longer representation (e.g., a storyboard)

Scenarios

Possible sequences of actions for reaching user goals

Scenario

- Scenarios are stories for design: rich stories of interaction
- Description of how the user engages the interactive system to solve a specific task
- Formats:
  - Written summary, Use Case
  - Graphical sketches (→ Storyboard)
  - Flowcharts, Transition Diagram…

Level of Details In Scenarios

- **Stories**
  - From needfinding
  - Used for understanding what people do and what they want
- **Conceptual** (abstract) **Scenarios**
  - Used for generating ideas and specifying requirements
  - Abstracts tasks from stories
  - No reference to technology
  - May lead to different concrete scenarios
- **Concrete Scenarios**
  - Used for envisioning ideas and evaluation
  - One possible solution to a Conceptual Scenario (may try many alternatives)
  - Shows how technologies are used in the user context
  - Key design features are included
- **Use Cases**
  - Used for specification and implementation (→ software engineering)

Storyboards

Comic book – like representation of user scenarios, with emphasis of how the system supports users in the development of the task

Storyboard

- "A graphical depiction of the outward appearance of the intended system, without any accompanying system functionality"
- A hand-drawn comic that features the execution of a task (like a concrete scenario)
- With a few panels (sequence of sketches) it conveys what a person may accomplish
  - Always include people
- They communicate **flow**, showing what happens **at key points** in time
- No artistic skills are required
  - Not about "nice pictures"
  - About communicating ideas

What To Find In a Storyboard

- Illustrate a goal (for the task)
- How a task unfolds (people interacting among themselves and with devices)
  - Repeated for all significant steps
- At the end, how they accomplish their goals (satisfactory outcome)

- Storyboards are **all about tasks**

Example

This storyboard illustrates how the app had already downloaded the daily recipe to the user's smartphone, so he could look it up and check the shopping list while on the underground, before shopping for ingredients and making a healthy meal.

http://alexmevissen.com/2014/07/16/storyboarding/

Example

This storyboard illustrates how the app can show the user that a home cooked meal can be quicker than ordering food delivery, using left over ingredients in the fridge.

http://alexmevissen.com/2014/07/16/storyboarding/

Storyboards Should Convey…

- Setting
  - People involved
  - Environment
  - Task being accomplished
- Sequence
  - What steps are involved?
    - Not the detailed UI
    - What role the UI plays in helping users achieve their goal?
  - What leads someone to use the system?
    - The "trigger" for the task
  - What task is being illustrated?
- Satisfaction
  - What's the motivation for the user?
    - The end point to reach after all the steps
  - What's the end result?
  - What need are you "satisfying"?

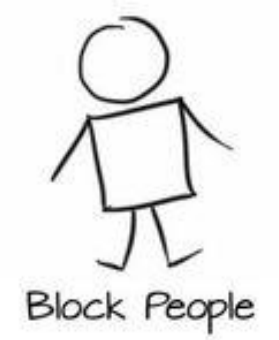
Handling Dynamicity In Storyboards

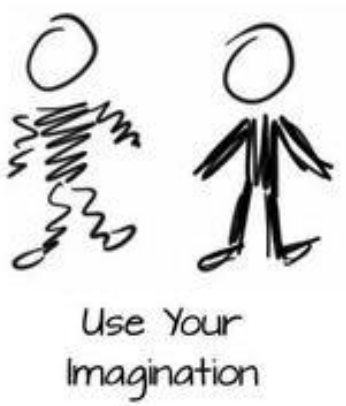
- Traditional storyboarding
  - “Comic book” conventions: actors, speech bubbles, background
  - Notes attached to each scene explaining what is happening
- Scored storyboards
  - When the user interface is highly dynamic, or contains specific media elements
  - Add specific annotations focusing on movement, colors, sounds, …
- Text-only storyboards
  - When the interaction behavior is too complex to compact into an illustration, use a longer text description

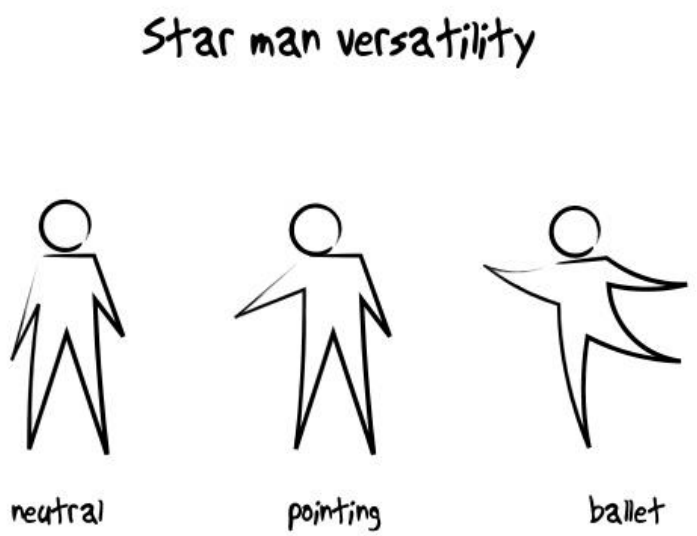
Why Hand-drawn?

- Quick
  - No need to spend time in graphics tools (they would "push" you to focus on details, too
  - Able to experiment different scenarios
- Imprecise
  - Users feel free to express more comments and suggestions w.r.t. a more "polished" version
  - Focus on the content (the graphics is obviously ignored)
  - No distraction by fonts, colors, icons, …

~~Drawing~~ Sketching People







Benefits of Storyboards

- Emphasize how an interface accomplishes a task
- Focus the conversation and feedback on user tasks
- Get everyone on same page about the app's goals
- Avoid nitpicking about user interface details (buttons, etc.)

References

- Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale: Human Computer Interaction, 3rd Edition, Chapter 15 “Task Analysis”
- David Benyon: Designing Interactive Systems, Chapter 11 “Task Analysis”
- http://www.usabilitybok.org/task-analysis
- https://www.usability.gov/how-to-and-tools/methods/task-analysis.html

Acknowledgements

- Some icons from https://icons8.com

- Some material by
  - http://www.inf.ed.ac.uk/teaching/courses/hci/0708/lecs/tasks.pdf
  - https://www.tutorialspoint.com/human_computer_interface/design_process_and_task_analysis.htm
  - https://www.slideshare.net/alanjohndix/hci-3e-ch-15-task-analysis

- Thanks to Fulvio Corno, past teacher of the course, for his work on some of these slides

License

- These slides are distributed under a Creative Commons license "**Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**"

- **You are free to:**
  - **Share** — copy and redistribute the material in any medium or format
  - **Adapt** — remix, transform, and build upon the material
  - The licensor cannot revoke these freedoms as long as you follow the license terms.
- **Under the following terms:**
  - **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
  - **NonCommercial** — You may not use the material for commercial purposes.
  - **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
  - **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.
- https://creativecommons.org/licenses/by-nc-sa/4.0/