

Analyzing

User Experience Design

Alberto Monge Roffarello

Academic Year 2023/2024

Administrativa

- **L'Assignment 1** è stato pubblicato
 - Deadline: 30 ottobre
 - Durerà due laboratori: usateli per ottenere il maggior numero di feedback possibile.
- Il laboratorio di martedì (17 ottobre) è pensato per **scegliere** il vostro dominio di interesse e iniziare a **prepare** le domande per le vostre interviste

Hall of Fame or Shame?

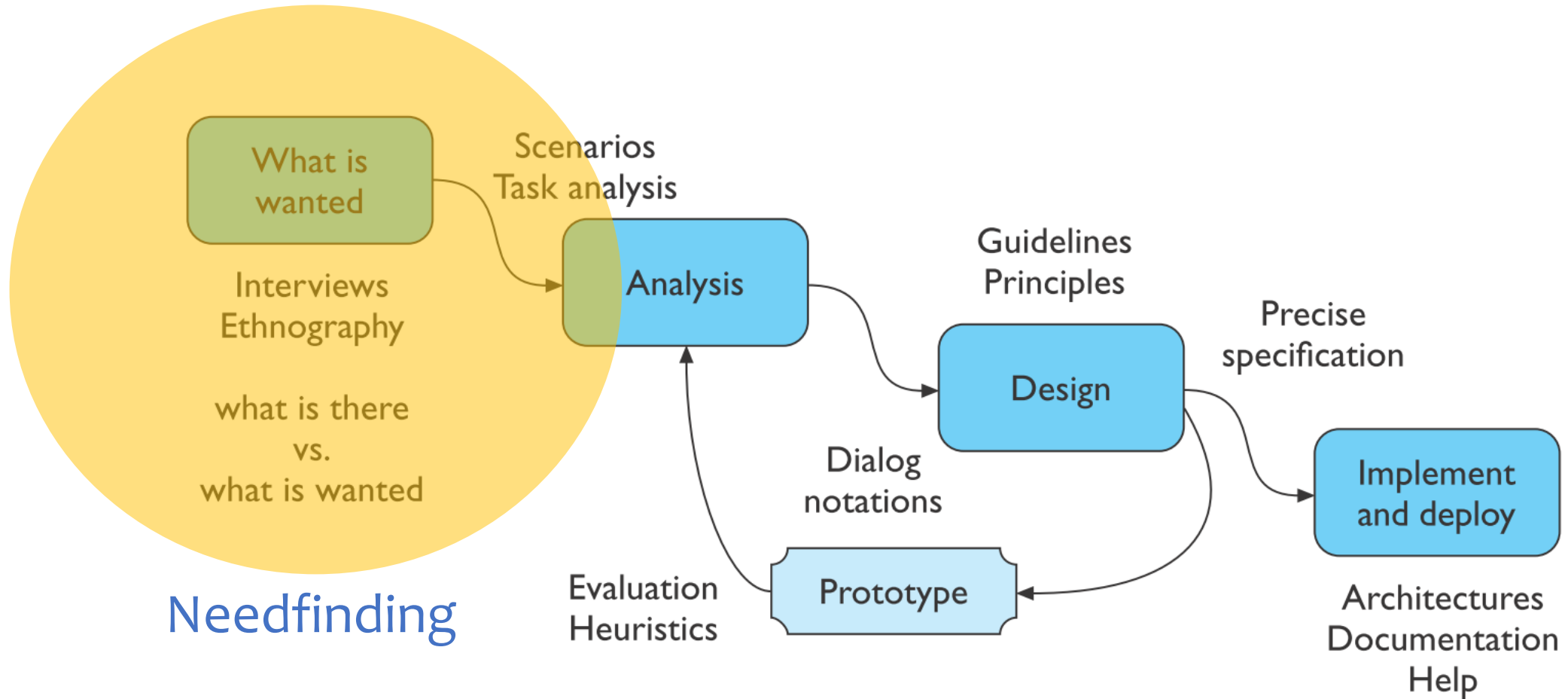


Hall of Fame or Shame?



<https://www.youtube.com/watch?v=e2RoNSKtVAo>

Human-Centered Design Process



What Are Needs?

- “Human emotional or physical necessities”
- “Gaps in a system”
- They are **verbs**: activities and desires with which your user could use help
 - not nouns (solutions)
- Often, it is helpful to use the phrases ‘needs a way to’ or ‘needs to be able to’ to describe user needs
- They emerge directly from user traits or from contradictions between two traits
 - such as a disconnect between what they says and what they do

What is Needfinding?

- Needs: **gaps** in a system
- Needfinding: discovering opportunities by **recognizing** this gaps



What is Needfinding?

- Figure out the story of **what** and **why**...

- ... and tell a new one!



Main Needfinding Questions

- Needfinding = **Finding Potential User Needs**
 - What do users need?

- That also requires
 - Who are the users?
 - How are they doing it, now?
 - What is the context in which they are doing it?
 - Can't we just ask them?

A Sample Domain

- Theme: Transportation
- Specific domain:

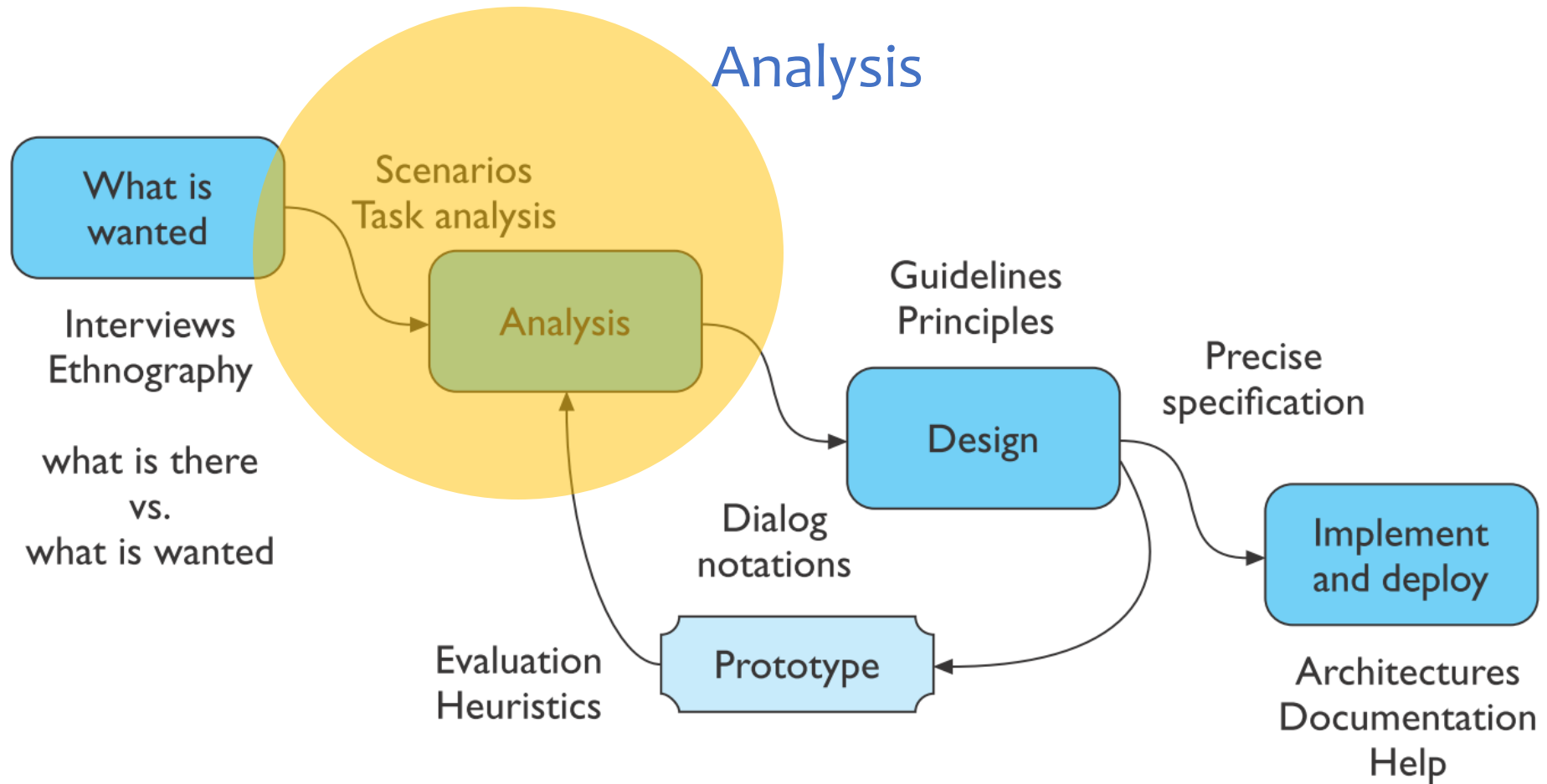
Target user(s)

- Immediate users:
- Domain experts:
- Lead users:
- Extreme users:

Planning an Interview

- To whom?
- **Direct 1:1 Interview** vs. Focus Group vs. Survey
- General Structure
- Questions
 - How many?
 - Which ones?
- Interacting
 - Follow-ups?
- Recording
 - Paper / Audio / Video / Pictures ?

Human-Centered Design Process



Analyzing and Synthetizing

- 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

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)

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?

Another Example of Task (with Steps)

- A person preparing an overhead projector for use would be seen to carry out the following steps:
 1. Plug in to main and switch on supply.
 2. Locate on/off switch on projector.
 3. Discover which way to press the switch.
 4. Press the switch for power.
 5. Put on the slide and orientate correctly.
 6. Align the projector on the screen.
 7. Focus the slide.

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'.

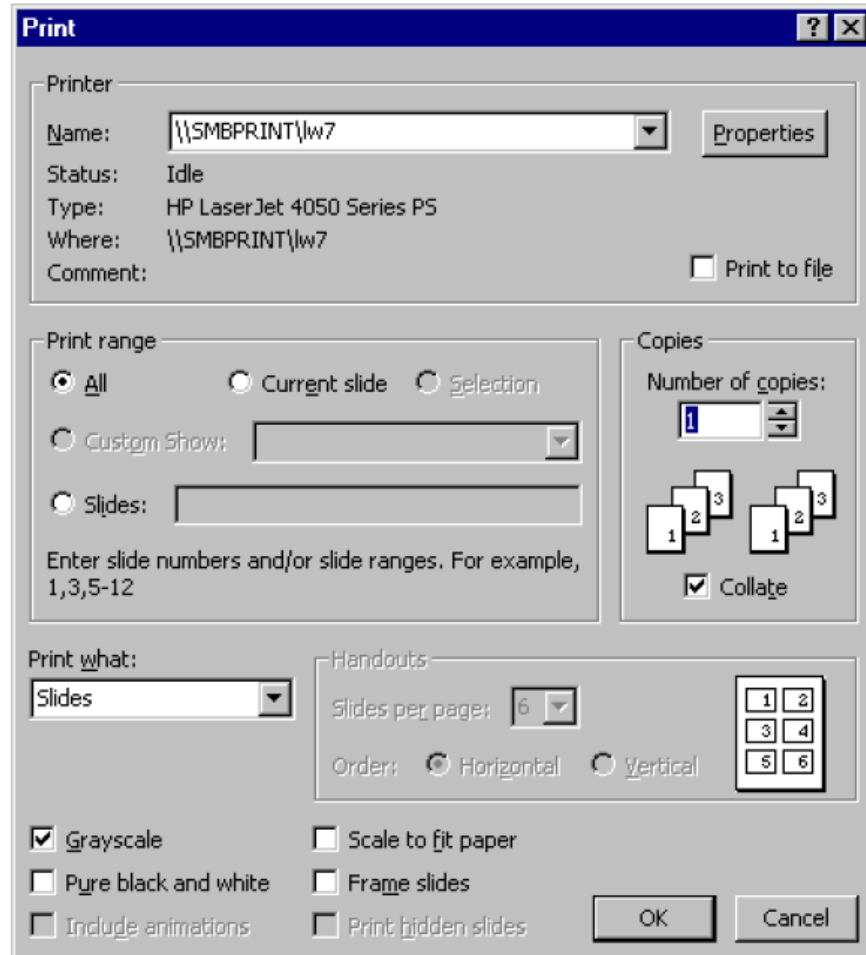
What You Learn with Task Analysis

- 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?

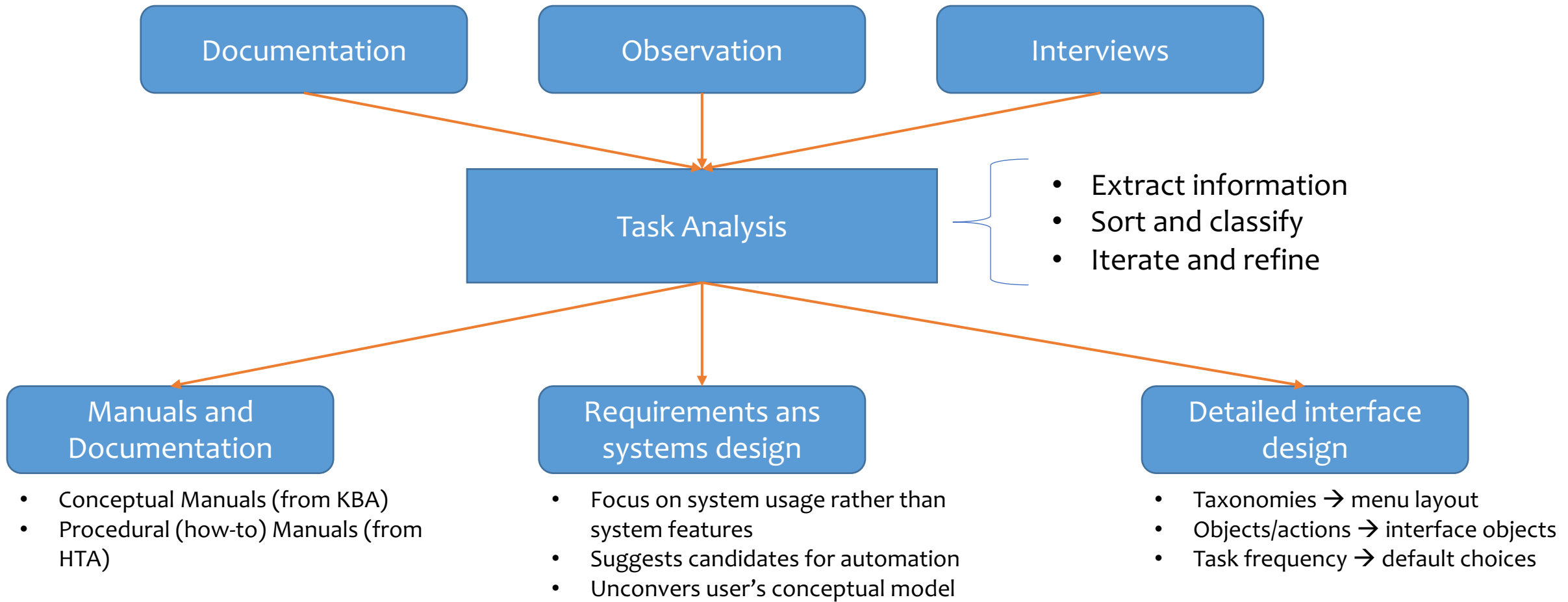
- Task analysis is the process of learning about ordinary users by observing them in action to **understand in detail how they perform their tasks and achieve their intended goals**
- Tasks analysis helps in:
 - **Identifying** the tasks that your application **must support**
 - Refining or re-defining 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



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

General Approach

- Observe
- Collect unstructured lists of words and actions
- Organize using notation or diagrams

[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.

Hierarchical Task Analysis

A Task decomposition method

Hierarchical Task Analysis (HTA)

- One possible method for Task Decomposition
- Hierarchical Task Analysis is the procedure of **disintegrating tasks into subtasks** that could be analyzed using the logical sequence for execution
- This would help in achieving the goal in the best possible way

“A hierarchy is an organization of elements that, according to prerequisite relationships, describes the path of experiences a learner must take to achieve any single behavior that appears higher in the hierarchy”.
(Seels & Glasgow, 1990, p. 94)

Example HTA: How To Clean a House

0. in order to clean the house
 1. get the vacuum cleaner out
 2. fix the appropriate attachment
 3. clean the rooms
 - 3.1. clean the hall
 - 3.2. clean the living rooms
 - 3.3. clean the bedrooms
 4. empty the dust bag
 5. put the vacuum cleaner and attachments away

Plan 0: do 1 – 2 – 3 – 5 in that order.
when the dust bag gets full do 4

Plan 3: do any of 3.1, 3.2 or 3.3 in any order
depending on which rooms need cleaning

- A **hierarchy** of **tasks** and **sub-tasks**
 - Indentation and numbering denote the levels
- A set of **plans** describing in what **order** and under what **conditions** subtasks are performed
 - Plans are labeled by the task they describe

Notes

0. in order to clean the house
 1. get the vacuum cleaner out
 2. fix the appropriate attachment
 3. clean the rooms
 - 3.1. clean the hall
 - 3.2. clean the living rooms
 - 3.3. clean the bedrooms
 4. empty the dust bag
 5. put the vacuum cleaner and attachments away

Plan 0: do 1 – 2 – 3 – 5 in that order.
when the dust bag gets full do 4

Plan 3: do any of 3.1, 3.2 or 3.3 in any order
depending on which rooms need cleaning

- Not all tasks are mandatory
 - E.g., task 4 is needed only if the bag is full.
- The order or operations may be free
 - E.g., the rooms may be cleaned in any order
- Could be further refined with additional knowledge or context
 - E.g., Plan 3: do 3.1 every day
3.2 once a week
when visitors are due 3.3

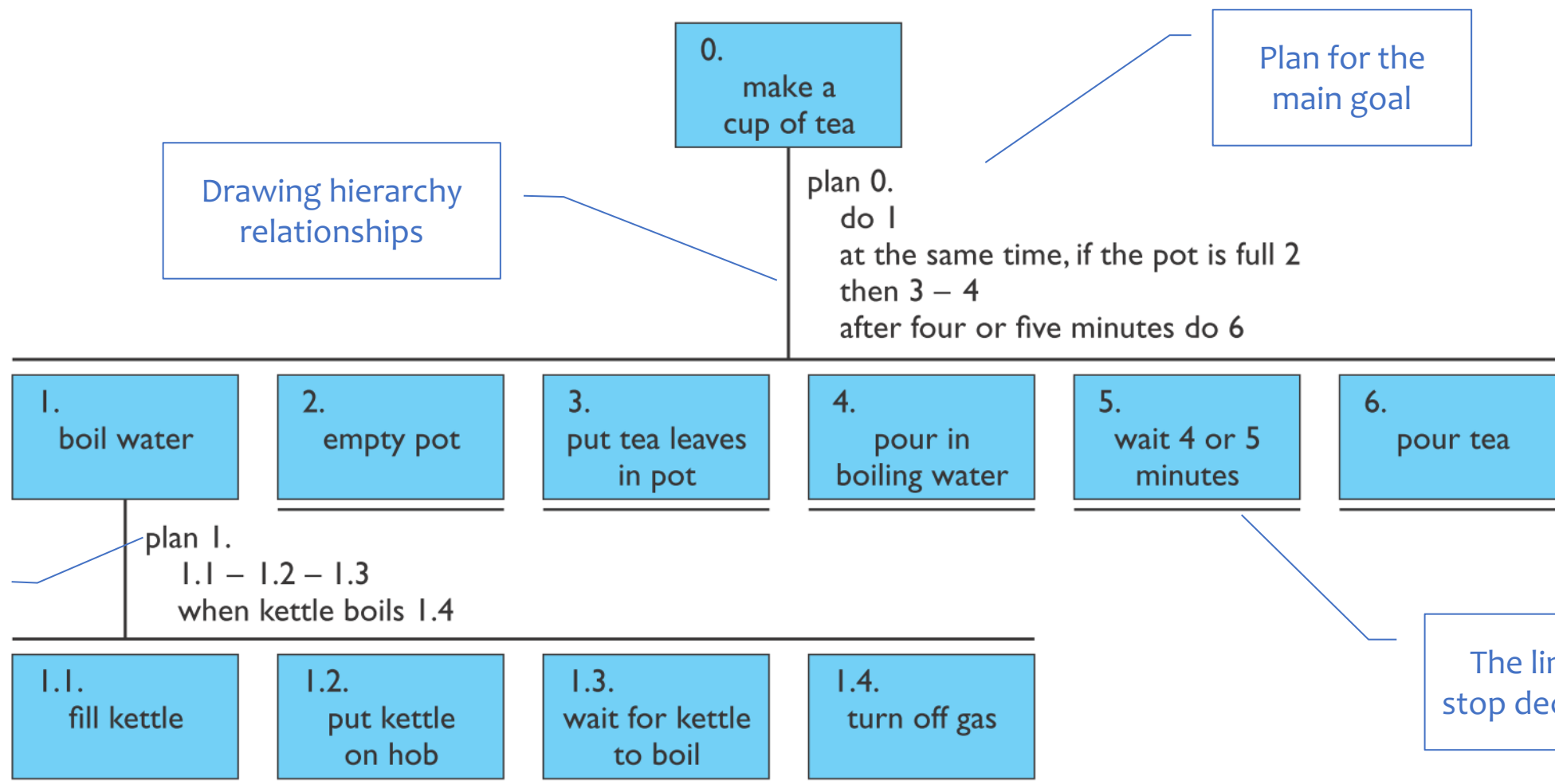
Expanding the Hierarchy (I)

- Each task is de-composed in sub-tasks, iteratively and recursively
 - Answer to the question: «what subtasks must be accomplished in order to perform the main task?»
 - The answer will come from direct observation, expert opinion, documentation, ...
- Procedural task knowledge elicitation techniques:
 - Observation, re-enactment
 - Ask about procedures and triggers (pre-conditions)
 - “What happens if X goes wrong?”
 - Sorting steps into appropriate orders

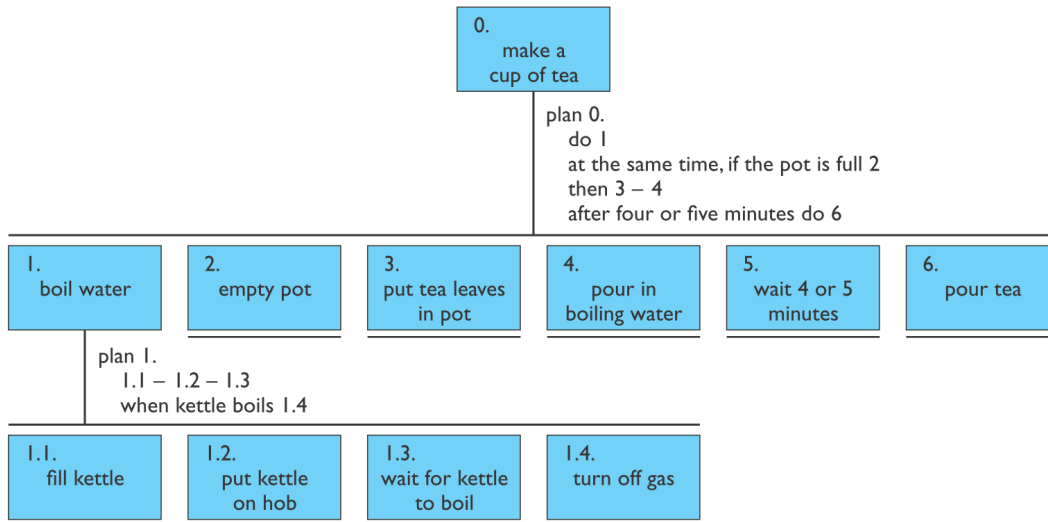
Expanding the Hierarchy (II)

- When is this process stopped?
 - Depends on the intended usage of the HTA (design vs. documentation vs. troubleshooting vs. ...)
 - Expand only **relevant** tasks
 - “Simple” tasks should be **obvious** to the users, and they should not contain hidden **risks** of failure
 - **Motor** actions are the lowest level (not always needed)

Example

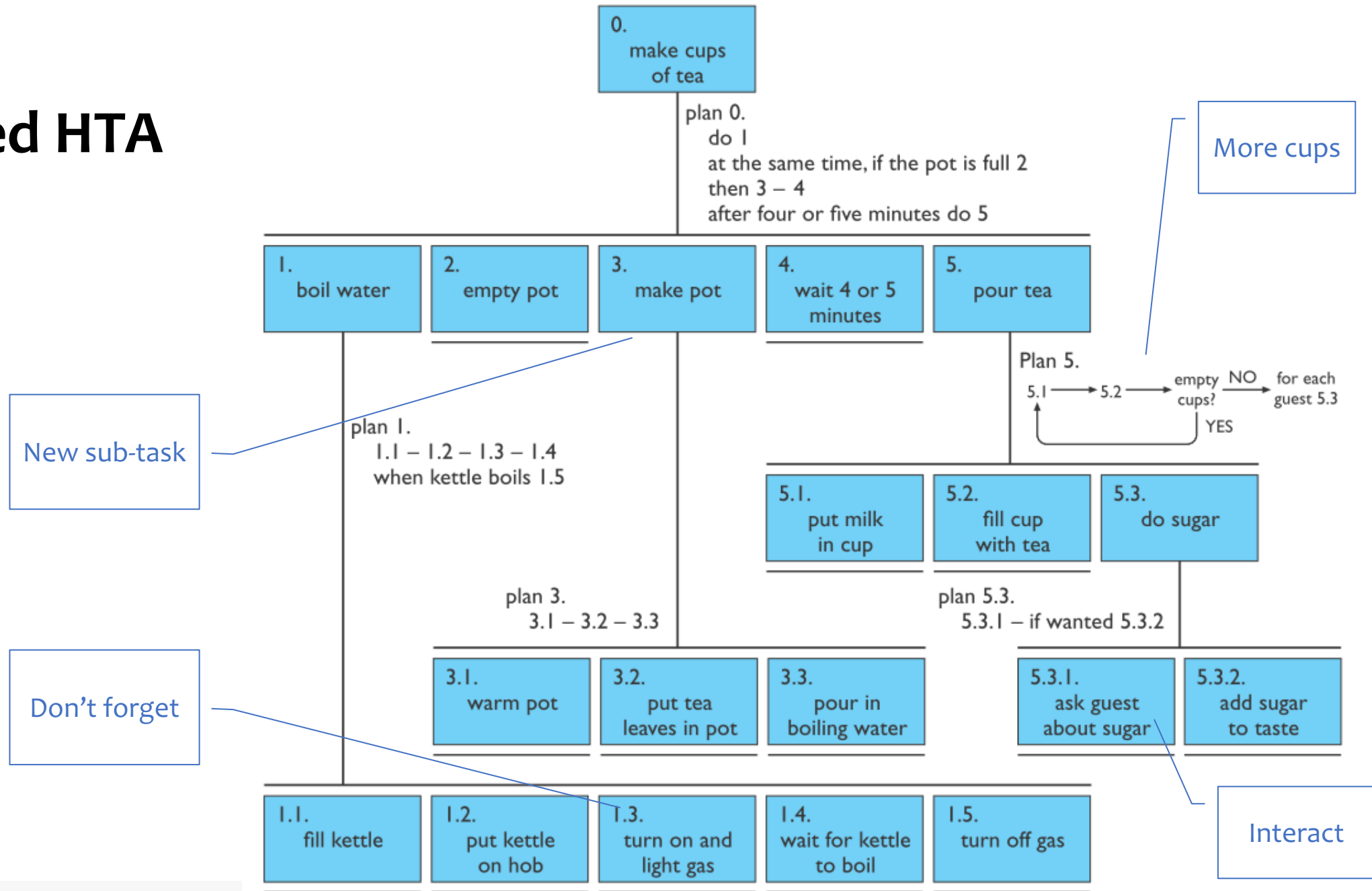


Refining the HTA



- Checking matched actions
 - Turn “off” without turning “on”?
- Restructuring
 - “Make pot” might be a meaningful task and group related actions
- Balancing complexity
 - Is “pour tea” simpler than “make pot”?
- Generalizing
 - If we want to make one or more cups?

Modified HTA



Main Constructs To Define Plans

- **Fixed sequence** - 1.1 then 1.2 then 1.3
- **Optional tasks** - if the pot is full 2
- **Wait for events** - when kettle boils 1.4
- **Cycles** - do 5.1 5.2 while there are still empty cups
- **Time-sharing** - do 1; at the same time ...
- **Discretionary** - do any of 3.1, 3.2 or 3.3 in any order
- **Mixtures** - most plans involve several of the above

Tasks as Explanation of Goals

- Imagine asking the user the question:
 - **What are you doing now?**
- For the same action, the answer may be:
 - Typing ctrl-B
 - Making a word bold
 - Emphasizing a word
 - Editing a document
 - Writing a letter
 - Preparing a legal case

Knowledge-based Analysis

Knowledge-based Analysis

- Aim to understand knowledge required for a task
 - Provide training material, how-to manuals
 - Take advantage of common knowledge across tasks
 - Organize information and Navigation in the application
- Focus on:
 - Objects used in task
 - Actions performed

Capturing Knowledge

- Use taxonomies
 - Represent levels of abstraction
 - Organization (grouping) depends on purpose

Example: Taxonomy of Car Controls

motor controls

steering *steering wheel, indicators*

engine/speed

direct *ignition, accelerator, foot brake*

gearing *clutch, gear stick*

lights

external *headlights, hazard lights*

internal *courtesy light*

wash/wipe

wipers *front wipers, rear wipers*

washers *front washers, rear washers*

heating *temperature control, air direction, fan, rear screen heater*

parking *hand brake, door lock*

radio

numerous!

How To Generate Concepts?

- Declarative knowledge elicitation techniques:
 - Established convention, existing documentation
 - Asking users to list objects; card sorting
 - Structured interviews, listing nouns and verbs
- Group concepts according to general-specific relationships

Modeling Depends On Your Goals

- Functional decomposition (what they do)

wash/wipe wipers *front wipers, rear wipers*
washers *front washers, rear washers*

- Positional decomposition (where they are located)

wash/wipe front *front wipers, front washers*
rear *rear wipers, rear washers*

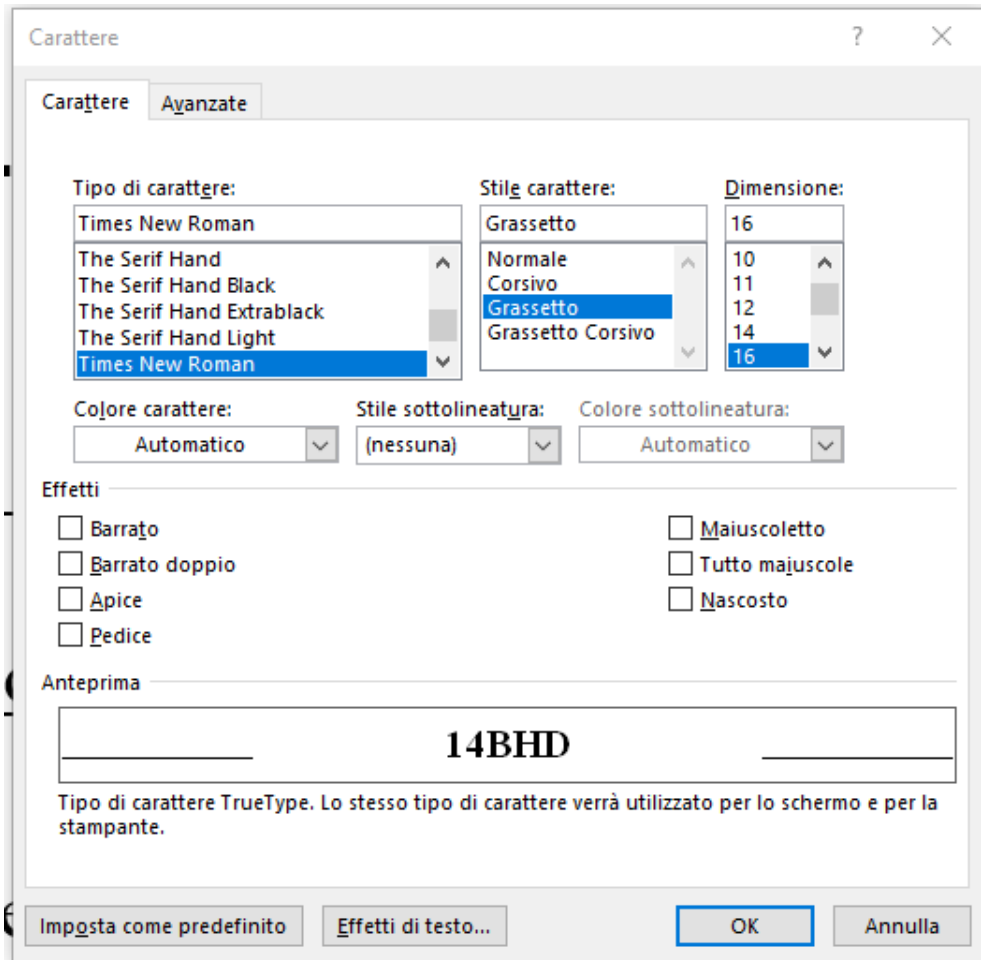
- Both views are correct, both are useful in different contexts

Task Descriptive Hierarchy

- More complex formalisms (not studied here) aim at capturing several (all?) points of view in a unique model

wash/wipe **AND**
function **XOR**
wipe
front wipers, rear wipers
wash
front washers, rear washers
position **XOR**
front
front wipers, front washers
rear
rear wipers, rear washers

From Concept Taxonomies To User Interfaces



- A Typeface is described by Font, Style, Size, Color
- Different “effects” may be applied
- Moving in the dialog window will explore different related concepts in the taxonomy

Tasks: Exercises

Goal

- Reflect on (good vs. bad) tasks
- Experiment with task analysis

All About 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, and describe 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

Example of Good Tasks

- Service/App: Uber
- Simple task: signaling for a ride
 - *Is it a task? Why is it simple?*
- Moderate task: reach out to the driver to get a forgotten object
 - *Is it a task? Why is it moderate?*
- Complex task: become a driver for Uber
 - *Is it a task? Why is it complex?*

Example of Bad Tasks

- Service/App: Uber
- Open the app and tap on “Travel”
 - *Is it a task? Why is it bad?*
- Go into your account settings, check the messages, and then send a present
 - *Is it a task? Why is it bad?*
- ...

Example of (Good) Tasks

- Service/App: Glovo/JustEat
- Simple task: _____
- Moderate task: _____
- Complex task: _____

Task Analysis

- Pick **one** of the “good tasks” discussed so far
- Use the Hierarchical Task Analysis to decompose it
 - In terms of steps and plans
- You can use draw.io to create the diagram
- We will see a few of them before the end of the class
 - E.g., by sharing the result in the Telegram group

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>
- Most of the slides are adapted from those used in the "Human Computer Interaction" course of Politecnico di Torino
 - <http://bit.ly/polito-hci>





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/>

