Task Analysis

Human Computer Interaction

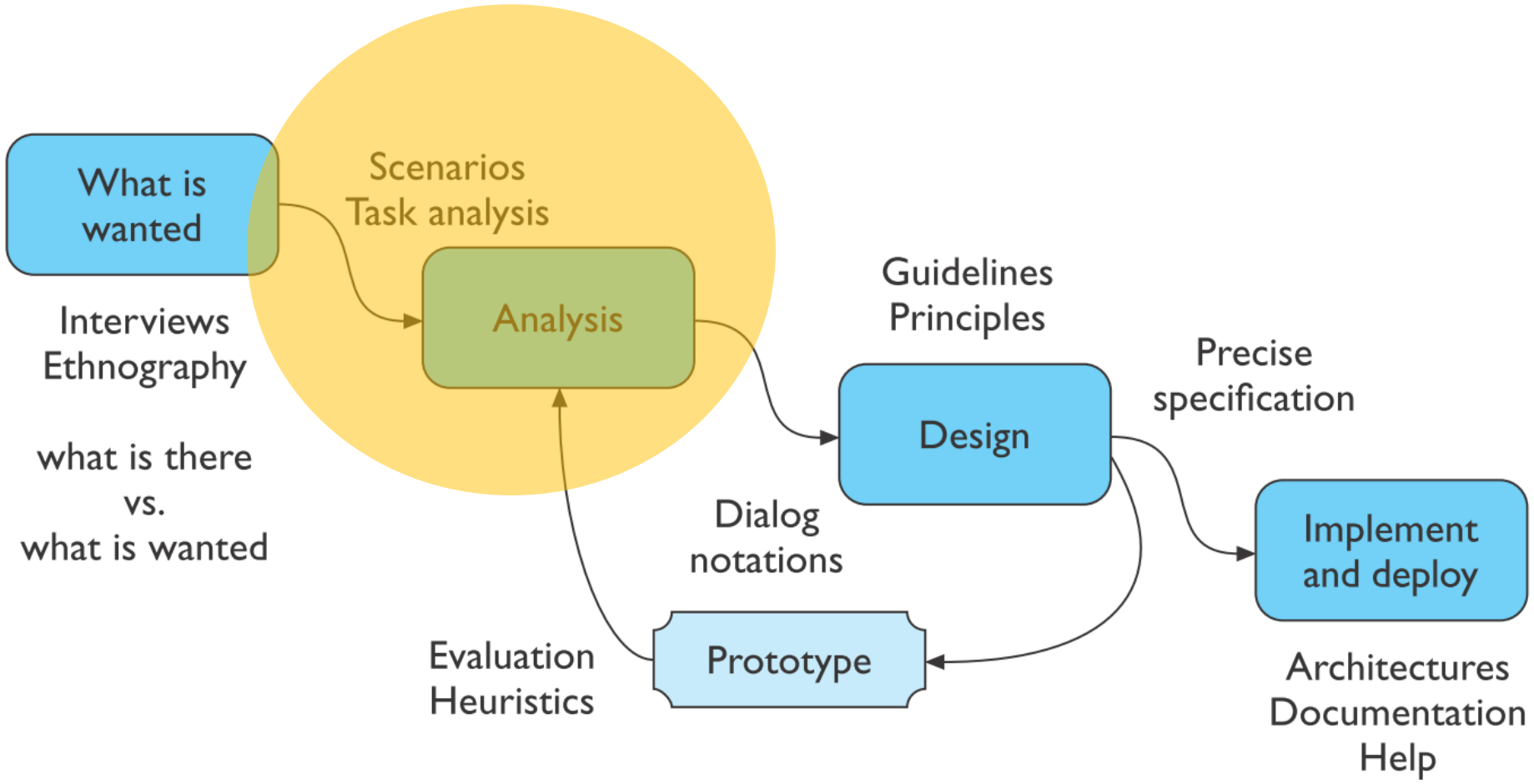
Fulvio Corno, Luigi De Russis

Academic Year 2020/2021

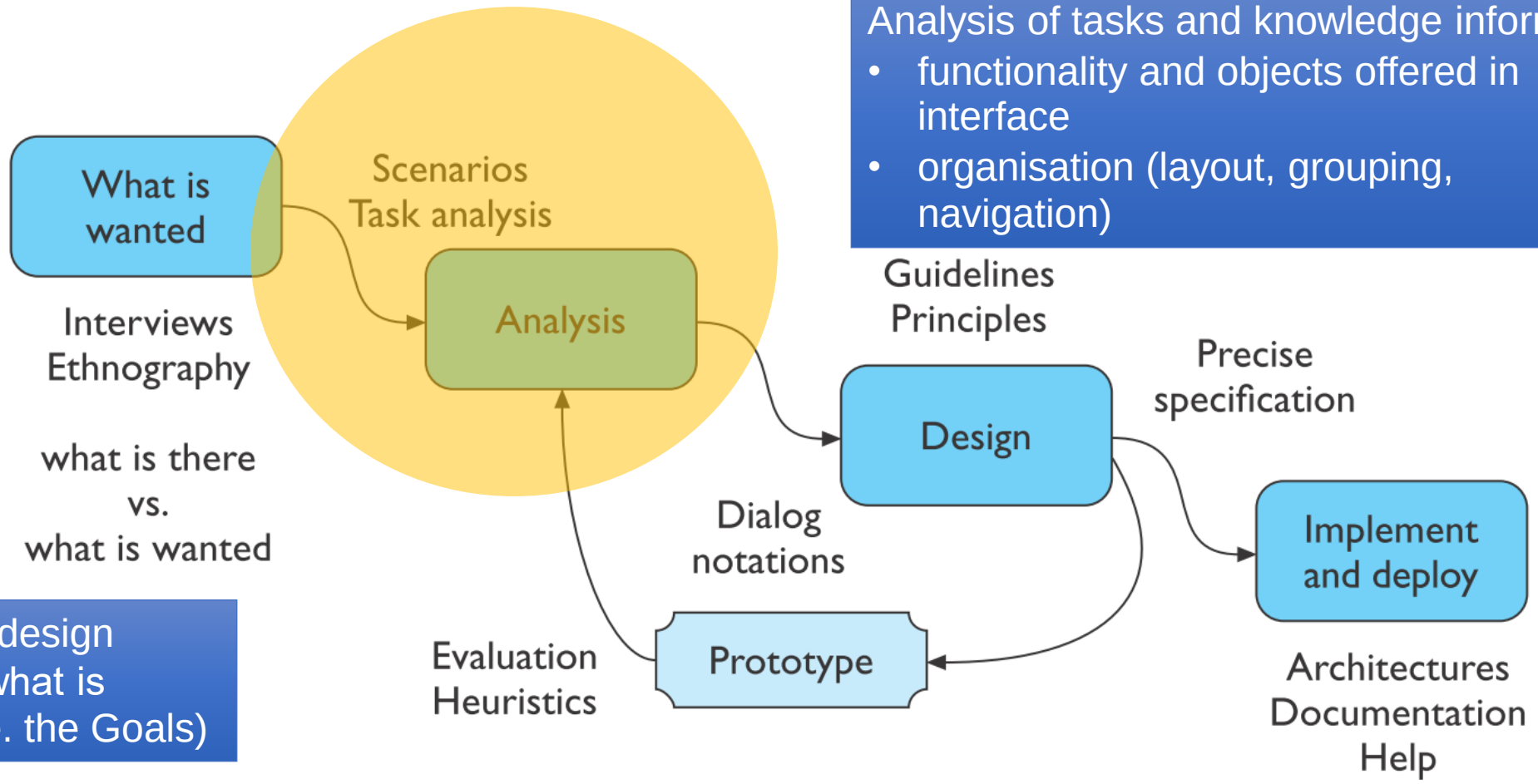
Summary

- Task Analysis
  - Purpose and structure
  - Goals, Tasks, Actions
- Hierarchical Task Analysis (HTA)
- Knowledge-based Analysis

Human-Centered Design Process



Human-Centered Design Process



Analysis of tasks and knowledge informs:

- functionality and objects offered in interface
- organisation (layout, grouping, navigation)

Interaction design driven by 'what is wanted' (i.e. the Goals)

Task Analysis

- Task Analysis is the study of the way people perform their jobs.
- Aim is to determine:
  - what they **do**
  - what things they **use**
  - what they must **know**

Example

- Task: to clean the house
  - 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 about:
  - vacuum cleaners, their attachments, dust bags,
  - cupboards, rooms etc.

Example

- For example, a person preparing an overhead projector for use would be seen to carry out the following actions :
  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'

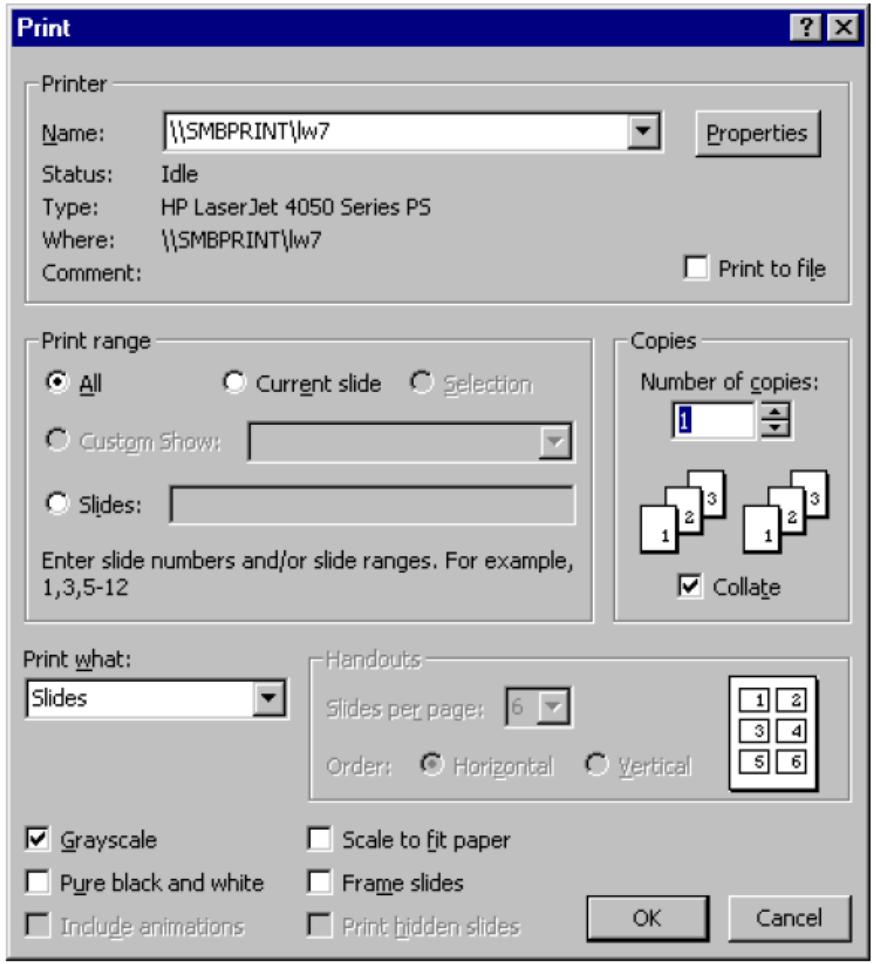
What you learn with Task Analysis

- What your users' goals are; 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

Why is it useful?

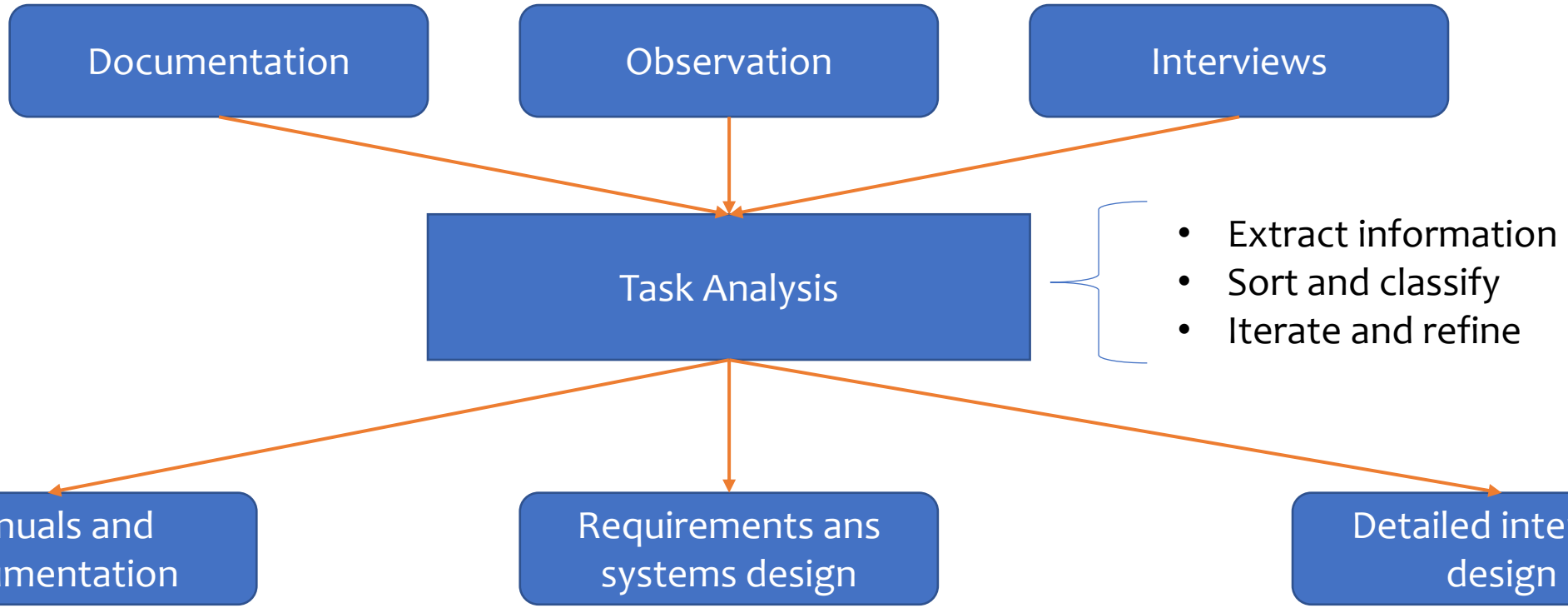
- Task analysis is the process of learning about ordinary users by observing them in action to understand in detail **how** they perform their **tasks** and achieve their intended **goals**.
- Tasks analysis helps in
  - Identifying the tasks that your website and applications **must support**
  - Refining or re-defining your site's navigation or search
  - Website requirements gathering
  - Developing your content strategy and site structure
  - Wireframing and 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



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

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

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

[Some] Techniques for 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

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

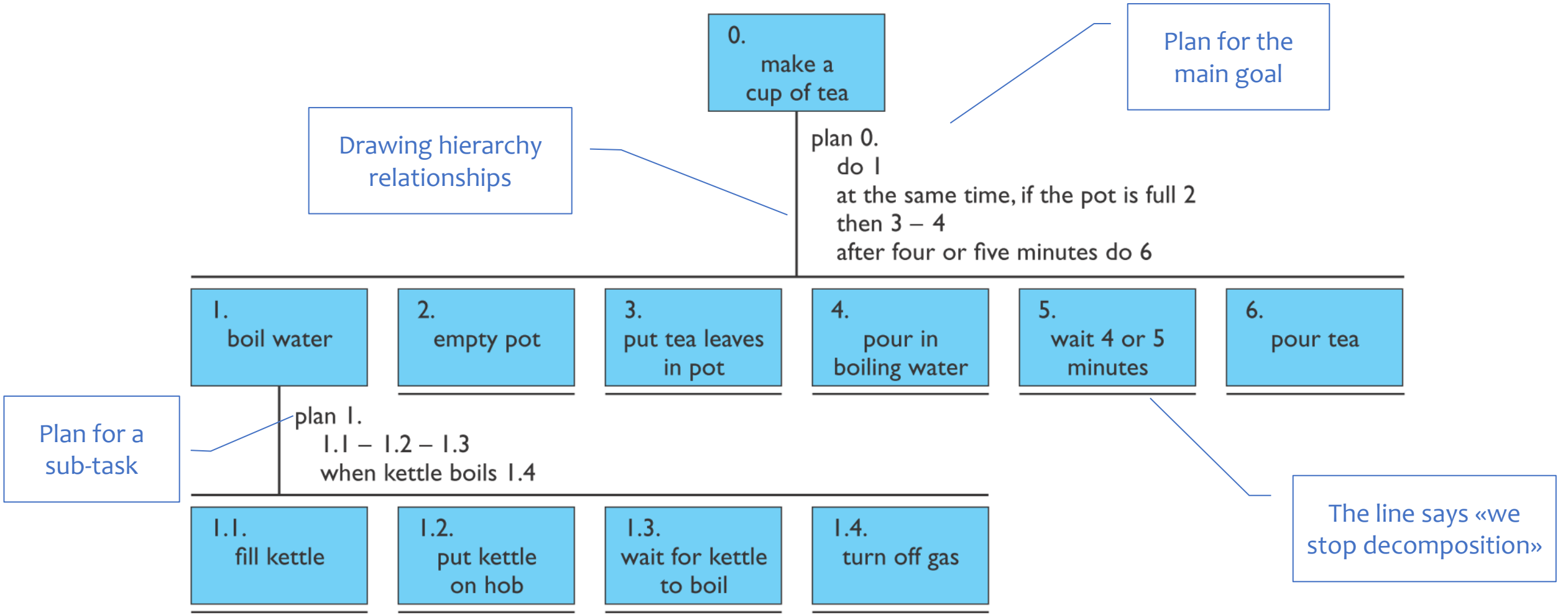
- Not all tasks are mandatory
  - E.g., task 4 is needed only if the bag is full.
- The order or operations may be free
  - E.g., the rooms may be cleaned in any order
- Could be further refined with additional knowledge or context
  - E.g., Plan 3: do 3.1 every day
    3.2 once a week
    when visitors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

Expanding the hierarchy

- 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

- 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)

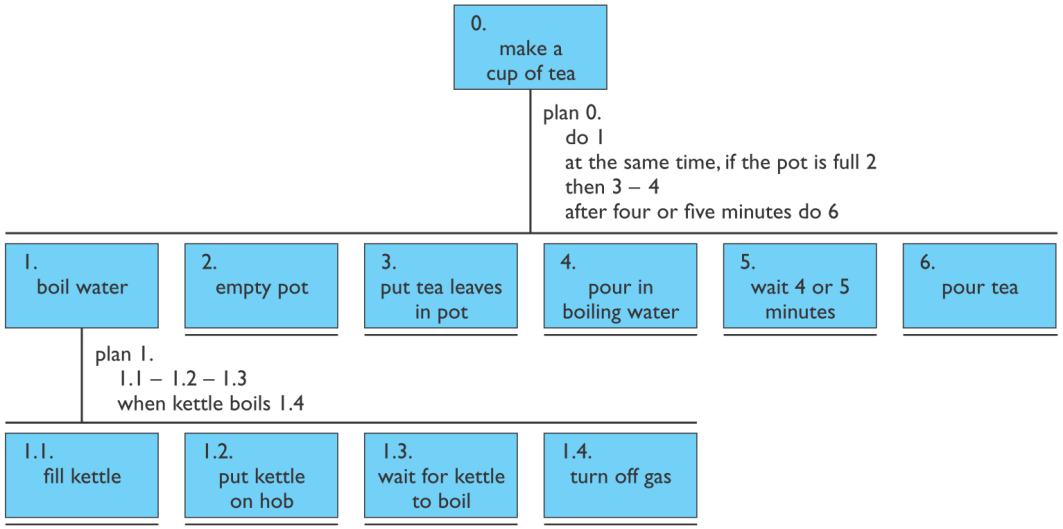
Example


0.
make a
cup of tea
plan 0.
do 1
at the same time, if the pot is full 2
then 3 – 4
after four or five minutes do 6
1.
boil water
2.
empty pot
3.
put tea leaves
in pot
4.
pour in
boiling water
5.
wait 4 or 5
minutes
6.
pour tea
plan 1.
1.1 – 1.2 – 1.3
when kettle boils 1.4
1.1.
fill kettle
1.2.
put kettle
on hob
1.3.
wait for kettle
to boil
1.4.
turn off gas
Plan for the main goal
Drawing hierarchy relationships
Plan for a sub-task
The line says «we stop decomposition»

Tasks as explanation

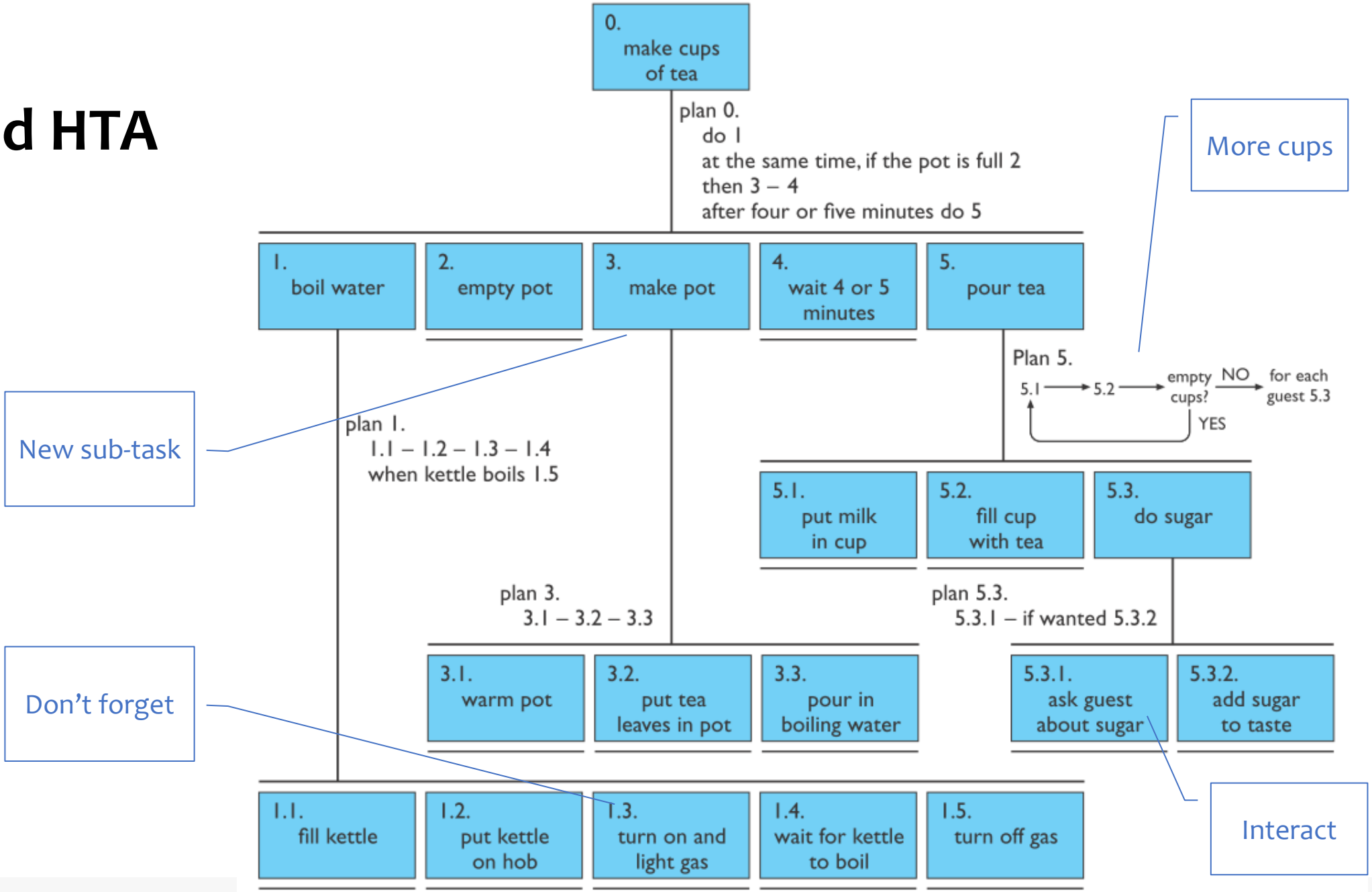
- Imagine asking the user the question:
  - **What are you doing now?**
- For the same action the answer may be:
  - typing ctrl-B
  - making a word bold
  - emphasising a word
  - editing a document
  - writing a letter
  - preparing a legal case

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

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
  - Organisation (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*

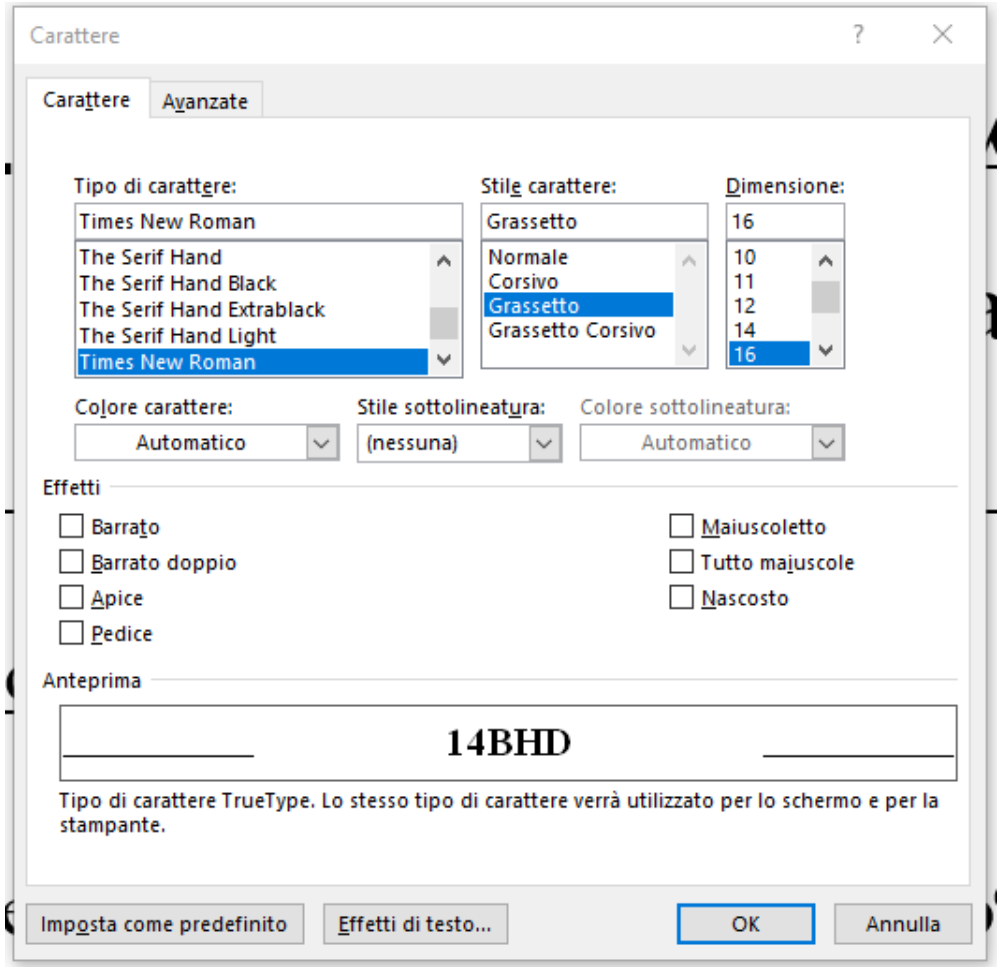
- Both views are correct, both are useful in different contexts

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

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

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/