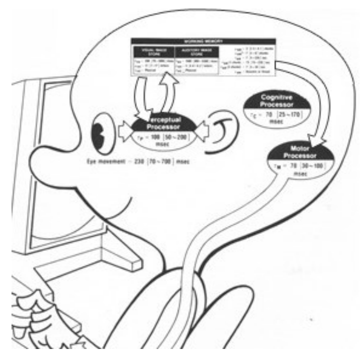
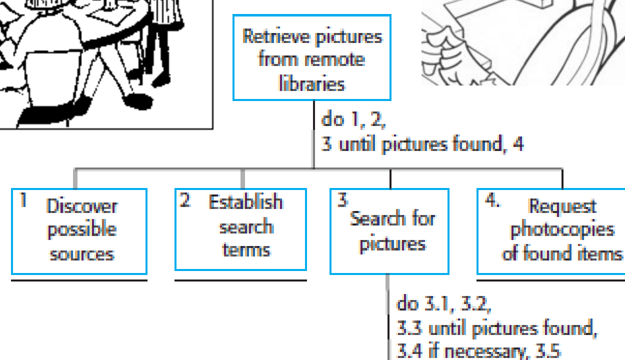
Universidade de Aveiro
Departamento de Electrónica,
Telecomunicações e Informática

Models for design





Beatriz Sousa Santos, 2021

- All engineering fields use models:
  - To evaluate – does the design have the needed characteristics?
  - To prescribe – directly contribute to the design
- Models are needed also in UI design
- Several types of models may be used throughout the design of user interfaces:
  - User models
  - Task analysis
  - Dialog notation
  - System models

I- User Models

- Models to obtain user requirements in their social and organization context

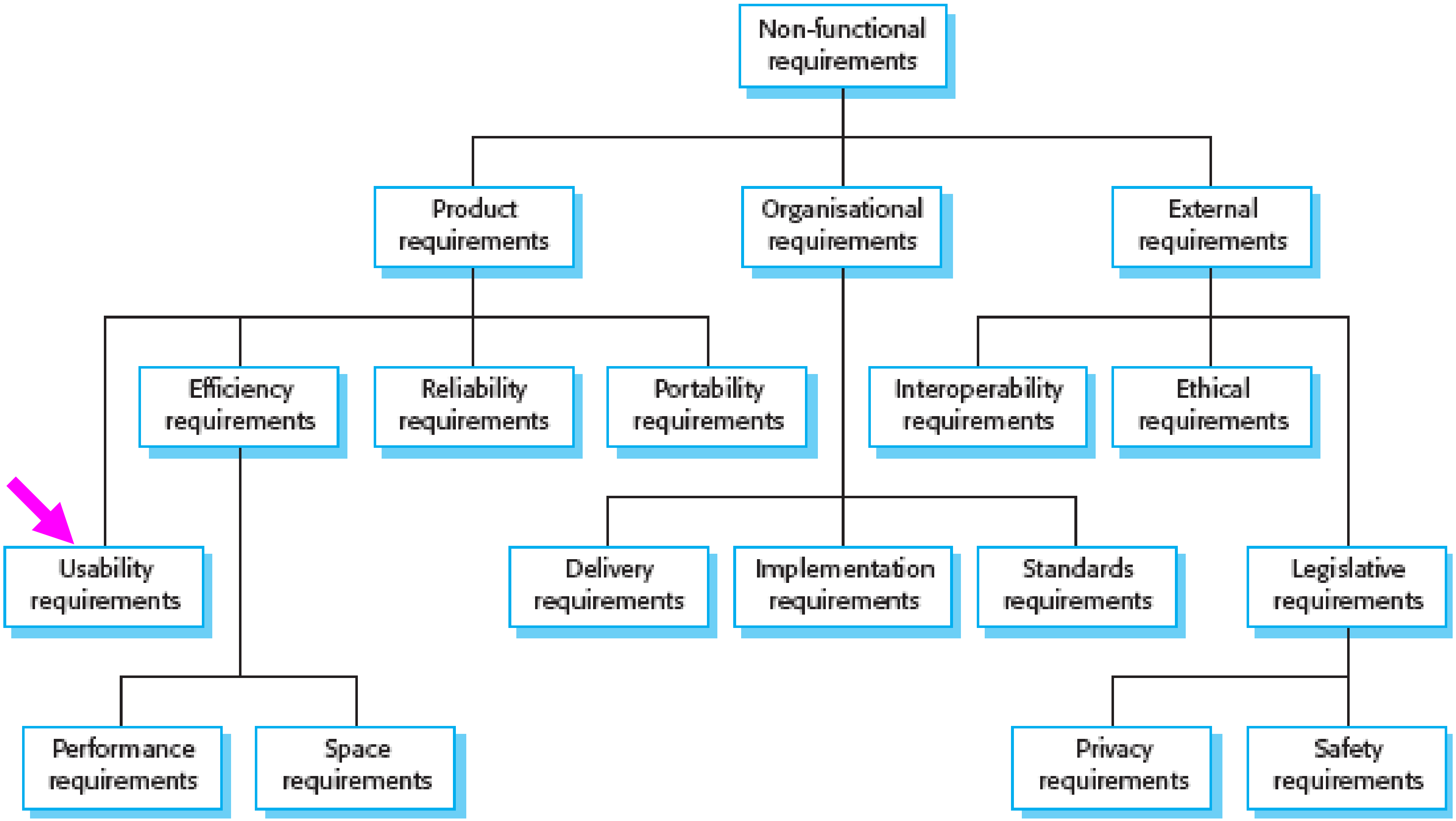
- User models- of the users' mental, perceptual and motor processes

- Personas- fictional characters based upon research in order to represent the different types of users

Models to obtain user requirements

- Obtaining user requirements is very important in S/W engineering
- But often focuses on functional requirements: what the system must do
- Overlooking non-functional requirements, as if it is:
    - acceptable
    - usable
- There are several models to capture user requirements

Non-functional requirements



(Sommerville, 2016)

Participatory design

- Users are involved as domain experts (e.g. business representatives and users) along the complete process and work together with developers to design a solution

  - It is work oriented and not system oriented
  - It is collaborative- users contribute to all phases
  - It is iterative – design is evaluated and reviewed in every phase

Participatory design

- Uses a set of techniques (that can be used in other contexts) to help transfer information from users to designers:
  - Brainstorming
  - Scenarios
  - Story boarding
  - Workshops
  - Paper and pencil exercises ...

http://infodesign.com.au/usabilityresources/participatorydesign
https://www.usabilityfirst.com/usability-methods/participatory-design/index.html

Personas

- Use a set of techniques (that can be used in other contexts) to help transfer information from users to designers:
  - Based on research
  - Represented as individual people
  - But represent groups of users
  - Explore ranges of behavior
  - Must have motivations

https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/personas

Cognitive models

- Represent the user in an interaction with the system; i.e. model aspects of user knowledge, intentions or processing

- The representation level varies from model to model, from:

  High level models...→ motor activity

- There are several types of cognitive models:
  - Object and tasks hierarchies

    (GOMS- Goals, Operators, Methods and Selection) ✓
  - Linguistic models
  - Physical and device

    (KLM- Keystroke Level Model) ✓

GOMS- Goals, Operators, Methods and Selections

- Proposed by Card, Moran and Newell, 1983
- A GOMS decomposition has the following elements:
- ***Goals***: what the user wants to attain
- ***Operators***: basic operations that the user has to perform to use the system; may affect the system or not (press a key or read a message)
- ***Methods***: possible decompositions of the goal into sub-goals (e.g. Select an option “Save” or press “ctrl S”)
- ***Selections***: rules to select the possible methods (taking into account the type of user and the system status)

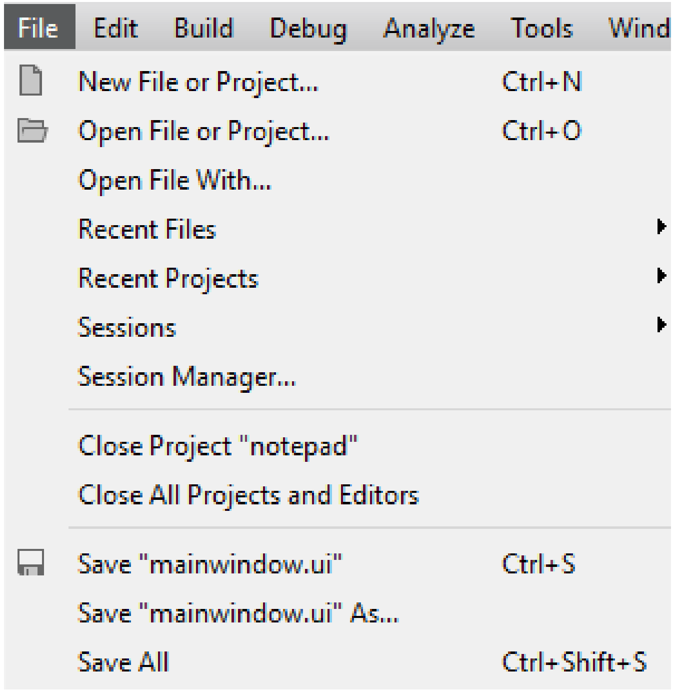
GOMS- Goals, Operators, Methods and Selections

- A typical GOMS analysis consists in decomposing a high level goal in a sequence sub-goals
- Selection rules must be adjusted to the user profile
- Analyzing the structure of the GOMS decomposition may give an **approximate measure** of :
  - Short Term Memory load (depth of the goal structure)
  - Time needed (a time for each operator)

Example: ‘save’ a file: using two common ways

- GOAL: **SAVE-A-DOCUMENT**

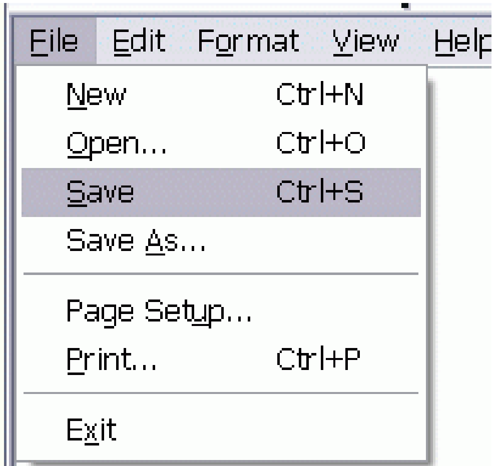
. [select GOAL: USE-SAVE-OPTION-METHOD
. . MOVE-MOUSE-TO-MENU-BAR
. . CLICK-OVER-FILE-OPTION
. . MOVE-MOUSE-TO-SAVE-OPTION
. . CLICK-SAVE-OPTION
. GOAL: USE-CTRLS-METHOD
. . PRESS-’CTRL’+’S’-KEYS]



User BSS:

Rule 1: **USE-CTRLS-METHOD** unless other rule applies

Rule 2: If has hand on mouse **USE-SAVE-OPTION-METHOD**



Another example: Copy an article from a journal

- Goal: Photocopy-paper
- . Goal: Locate-article
- Goal: Photocopy-page repeat
- Goal: Orient-page
- open cover
- select-page
- position-paper
- close-cover
- Goal: Press-copy-button
- Goal: Verify-copy
- locate-out-tray
- examine-copy
- Goal: Collect-copy
- locate-out-tray
- remove-copy (outer goal satisfied)
- Goal: Retrieve-journal
- open-cover
- remove-journal
- close-cover

Closure problem
(the user attains the goal before the task is complete)

The “closure problem”

The copies usually are available to the user before they remove the original from the photocopier and walk away!

To prevent this, the overall goal should be satisfied only after removing the original

In earlier ATMs the money was given before returning the card

... many users left the card:
their goal was getting money!

This was changed.

The “closure problem” in MultiBanco

In stores usually these are the following steps:

- Insert the card
- Insert the pin code
- Transaction approval -> audio signal
- Remove the card
- Receipt is handed to the client

At the ATMs the money is given (goal satisfaction) only after the card is removed by the client

These procedures help not to forget the card!

GOMS- Goals, Operators, Methods and Selections

- Capacities:
  - It has been used in cognitive model research
  - It may describe adequately how **experienced users** perform **routine tasks**
  - Associated to a device model allows time estimates

  Limitations:
  - It does not give information concerning user knowledge to estimate training or transfer times

KLM- Keystroke-Level Model

- Proposed by Card, Moran e Newell, 1980
- Predicts user performance based on motor system characteristics
- Models unitary interaction tasks (simple command sequences <20s) (e.g. change the font of a word, use search and replace)
- These tasks have two phases:
  - Acquisition (building the mental representation of the task)
  - Execution (using the system)
- KLM only models the execution phase (the user has already decided how to use the task)

KLM- Keystroke-Level Model

- The execution phase may be decomposed in 7 operators:

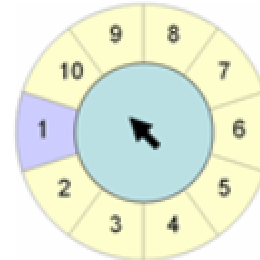
  - K- *Keystroke* (varies with typing skill)
  - B- *Button press of the mouse*
  - P- *Pointing at a target* (Fitts' law)
  - H- *Homming between mouse and keyboard*
  - D- *Drawing using mouse*

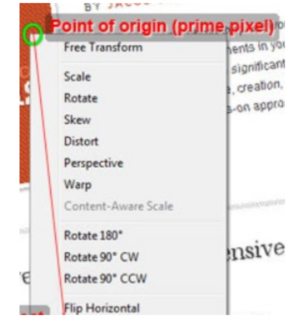
  motor

  - M- *Mentally preparing for physical action* → mental

  - R- *System Response* (often may be ignored) → system

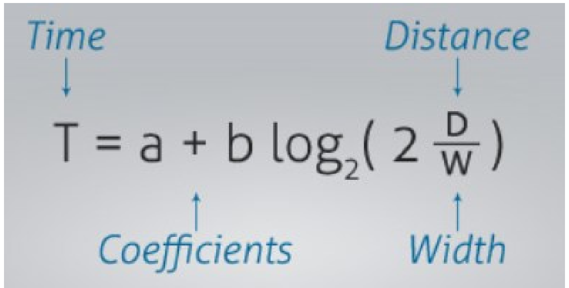
Fitts's law (1954)

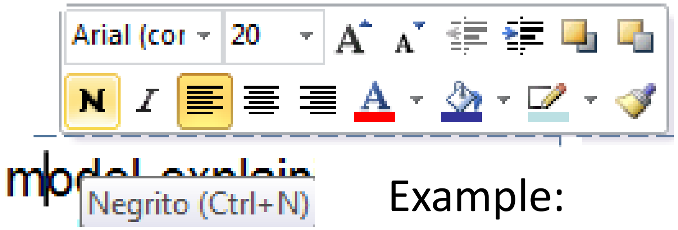




- Empirical model explaining <u>speed-accuracy tradeoff</u> characteristics of human muscle movement with some analogy to Shannon's channel capacity theorem

- Estimates the average time a user takes to select a target considering the distance (D) from the cursor and the Width (W) of the target:





Example:
Minimizing D

- The larger the target the easier to select (no fine control needed)

- The farther the target from the cursor the longer it will take

http://www.interaction-design.org/encyclopedia/fitts_law.html

Example

Using a mouse based editor – correct a character

Point at the wrong character, delete it, write a new character, return to the original place

1- Move hand to mouse → H [mouse]

2- Place the cursor after the error → PB [left]

3- Return hand to keyboard → H [keyboard]

4- Delete wrong character → MK [delete]

5- Write correct character → K [char]

6- Replace cursor → H [mouse] MPB [left]

Adding all times:

$T\ total = 2T_k + 2T_P + 2T_B + 3T_H + 2T_M$

This is an **estimate of the total time** the user will take to perform the task

Times for KLM operators:

(empirically established and may be vary for different types of users, e.g. naïve or experienced)

Operator	Remarks	Time (s)
K	Press key	
	good typist (90 wpm)	0.12
	poor typist (40 wpm)	0.28
	non-typist	1.20
B	Mouse button press	
	down or up	0.10
	click	0.20
P	Point with mouse	
	Fitts' law	$0.1 \log_2(D/S + 0.5)$
	average movement	1.10
H	Home hands to and from keyboard	0.40
D	Drawing – domain dependent	–
M	Mentally prepare	1.35
R	Response from system – measure	–

wpm = words per minute

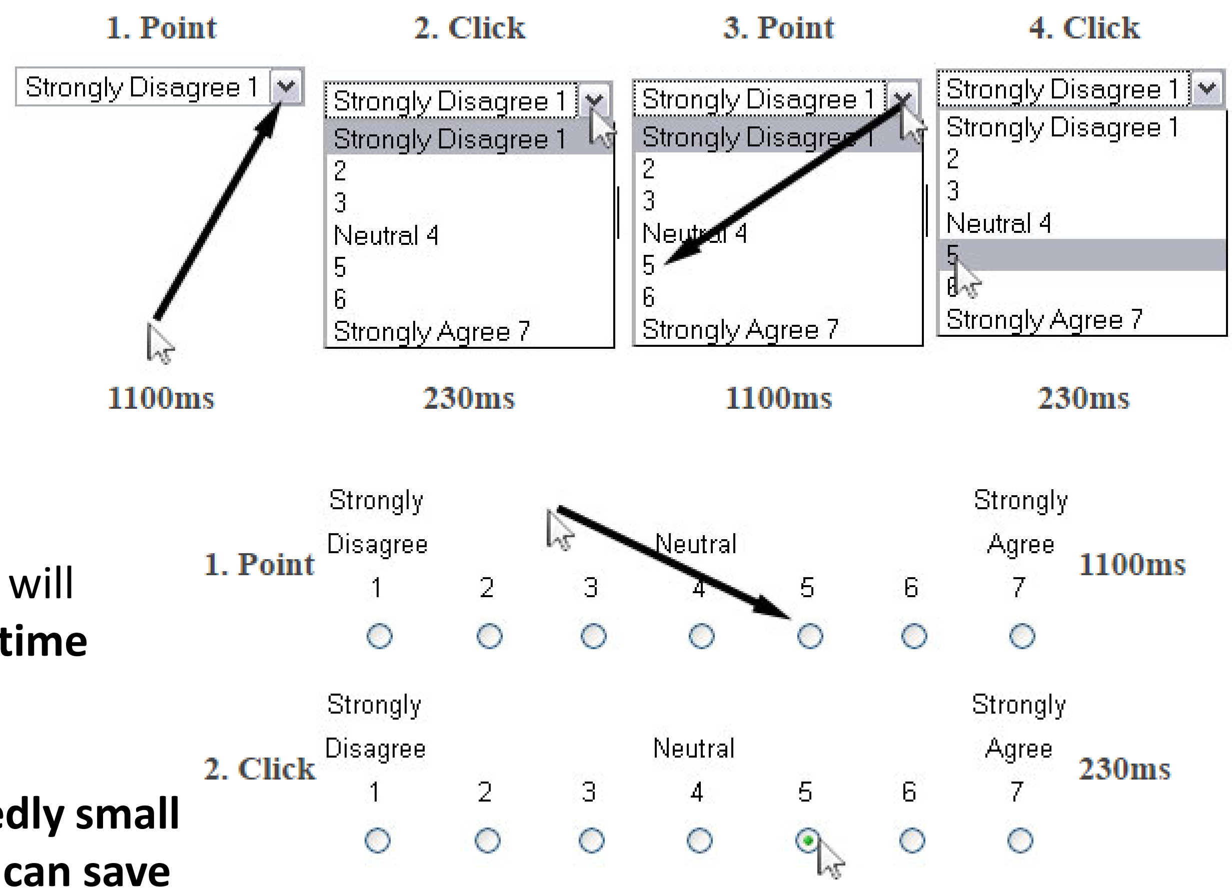
(Dix et al., 2004)

Another example: which survey alternative takes less time?

- Drop down list

  or

- Radio buttons?



- The radio button option will probably take ~**half the time**

- **If a task is done repeatedly small changes to an interface can save a lot of time!**

https://measuringu.com/predicted-times/

More examples:

Design A: drag the file into the trash can[29]	Design B: use the short cut “control + T”[30]
method encoding (operator sequence)[31]	method encoding (operator sequence)[32]
1. initiate the deletion (M) 2. find the file icon (M) 3. point to file icon (P) 4. press and hold mouse button (B) 5. drag file icon to trash can icon (P) 6. release mouse button (B) 7. point to original window (P)	1. initiate the deletion (M) 2. find the icon for the to-be-deleted file (M) 3. point to file icon (P) 4. press mouse button (B) 5. release mouse button (B) 6. move hand to keyboard (H) 7. press control key (K) 8. press T key (K) 9. move hand back to mouse (H)
Total time	Total time
3P + 2B + 2M = 3*1.1 sec + 2*.1 sec+ 2*1.35 sec = 6.2 sec	P + 2B + 2H + 2K + 2M = 1.1 sec + 2*.1 sec + 2*.4 sec + 2*.2 sec + 2*1.35 sec = 5.2 sec

This shows that Design B is 1 second faster than Design A, although it contains more operations.

https://en.wikipedia.org/wiki/Keystroke-level_model

- KLM has been extended to mobile

Paul Holleis, Friederike Otto, Heinrich Hussmann, and Albrecht Schmidt. 2007. Keystroke-level model for advanced mobile phone interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '07).

KLM- Keystroke-Level Model

- Placing times T_M is related to chunking and the type of user
- This model has an applicability limited to micro-dialog
- **It allows only approximate results**; thus reasonable estimates concerning the user are enough
- Can predict a skilled user's task time (error-free) to within10-20% of the actual time.
- Its main application is **alternative comparison**

II- Task analysis

- What it is and how it can be performed (more or less formally)

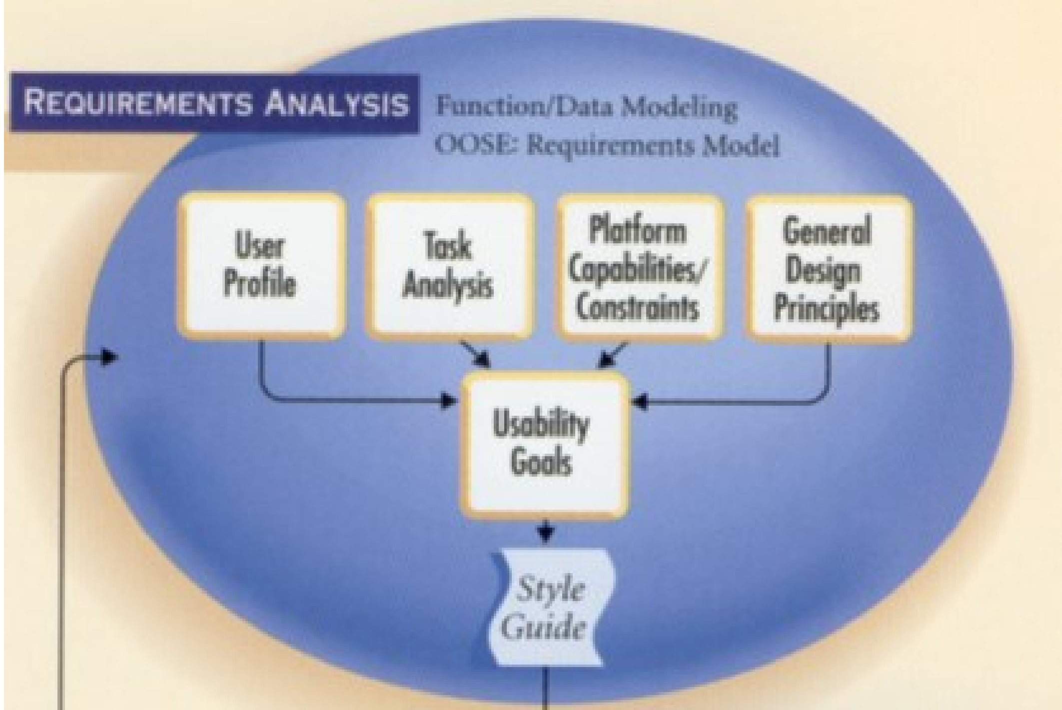
(Note that in Lab classes we started by doing it in an informal way…)

- Techniques

- Sources of information

- How to use it

(Mayhew, 1999)



Task analysis

- It is the analysis of how people perform their work
  - what they do
  - what they use
  - what they need to know

- Example: vacuum cleaning a house

  Get the vacuum cleaner

  Choose the adequate attachment

  Clean the rooms

  Empty the bag when it is full

  Put the vacuum cleaner and attachments away

- Users have to know about vacuum cleaners, rooms, …

Task Analysis approaches

- There are a lot of approaches and methods; we have used an informal way in the practical classes, but there are more formal methods

  – Task decomposition ✓

  – Knowledge based

  – Relation and entities based

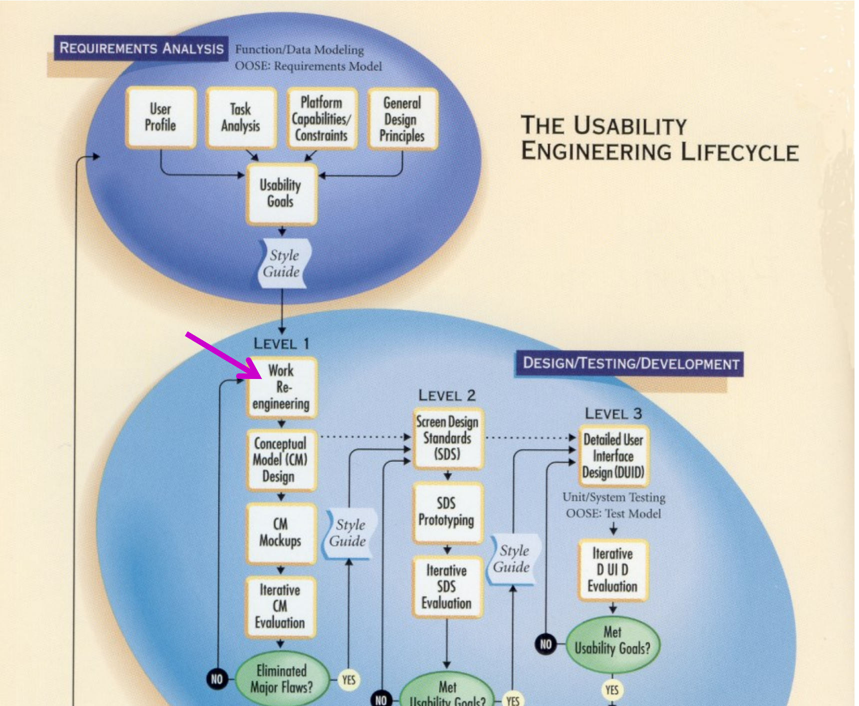
- Different approaches:
  - <u>Task decomposition</u> - divides tasks into subtasks that must be performed in a specific sequence
  - <u>Knowledge based</u> – considers what users need to know about the objects and actions involved in performing the task and how knowledge is organized
  - <u>Relation based</u> - is focused on actors and objects, relations among them and the actions they perform

Task Analysis

- Observation (of various types) is a fundamental tool

- It can be used to:
  - produce documentation and training materials
    (the observation of how existing
    systems are used is enough)

  - design new systems
    (work-re-engineering is
    usually necessary)



Task Decomposition

- Hierarchical Task Analysis (HTA) is one of the most used task analysis techniques and produces:

  - a task and sub-task hierarchy

  - plans with a sequence and execution conditions

Simple example: vacuum cleaning the house:

0. in order to clean the house
    1. get the vacuum cleaner
    2. fix the appropriate attachment
    3. clean the rooms
        3.1. clean the hall
        3.2. 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

Plan 3 could be more specific; what if it were varnishing the house?

- Where should the decomposition stop?

- The decomposition detail depends on the goals of each task analysis

- Example: in a factory what should be done in an emergency
  0. in a emergency
     1. Read the alarms
     2. Determine the corrective actions
     3. Execute the correction actions

- If the goal is
  – installing a monitoring system → expand 1 e 3

  – produce operation manuals → expand 2

- Where should the decomposition stop?

- A stop decomposition rule:

Stop if P x C < a specific value

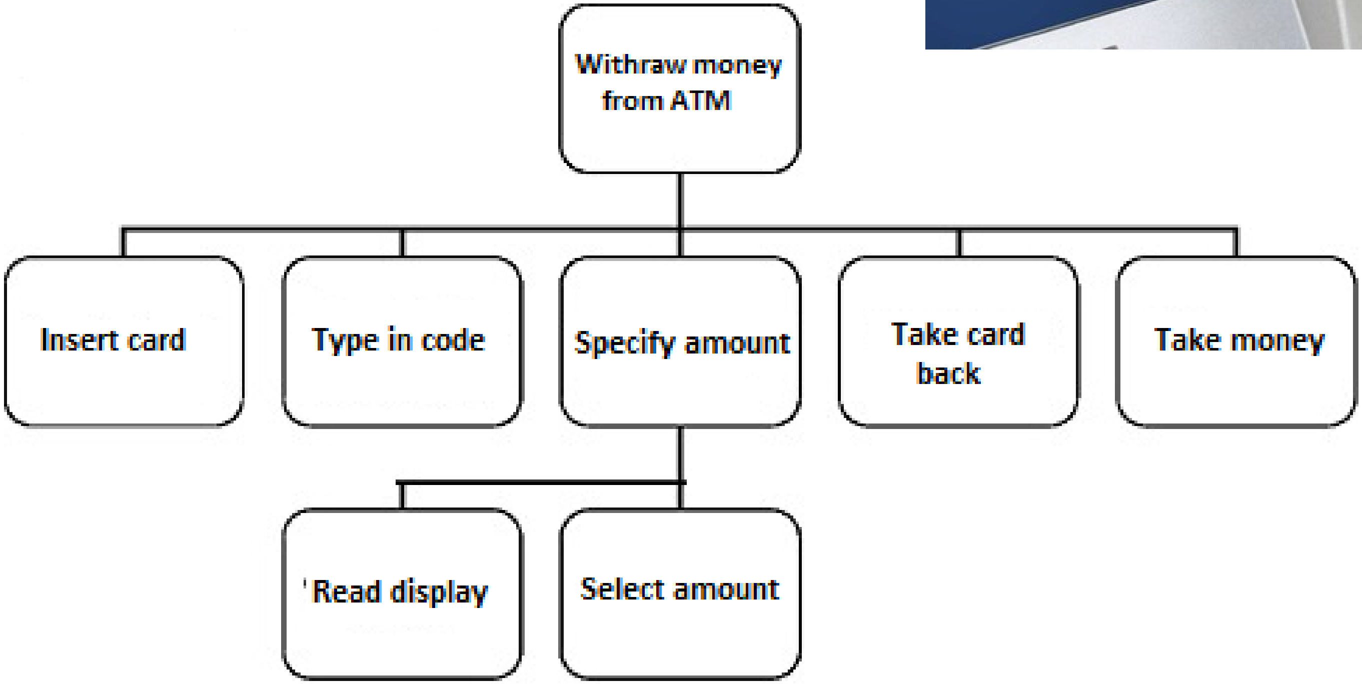
error cost

Probability of making an error

i.e.: simple tasks do not need decomposition unless they are critical!

Incomplete decomposition of the task: Withdraw money from an ATM

Can you fix and complete it?

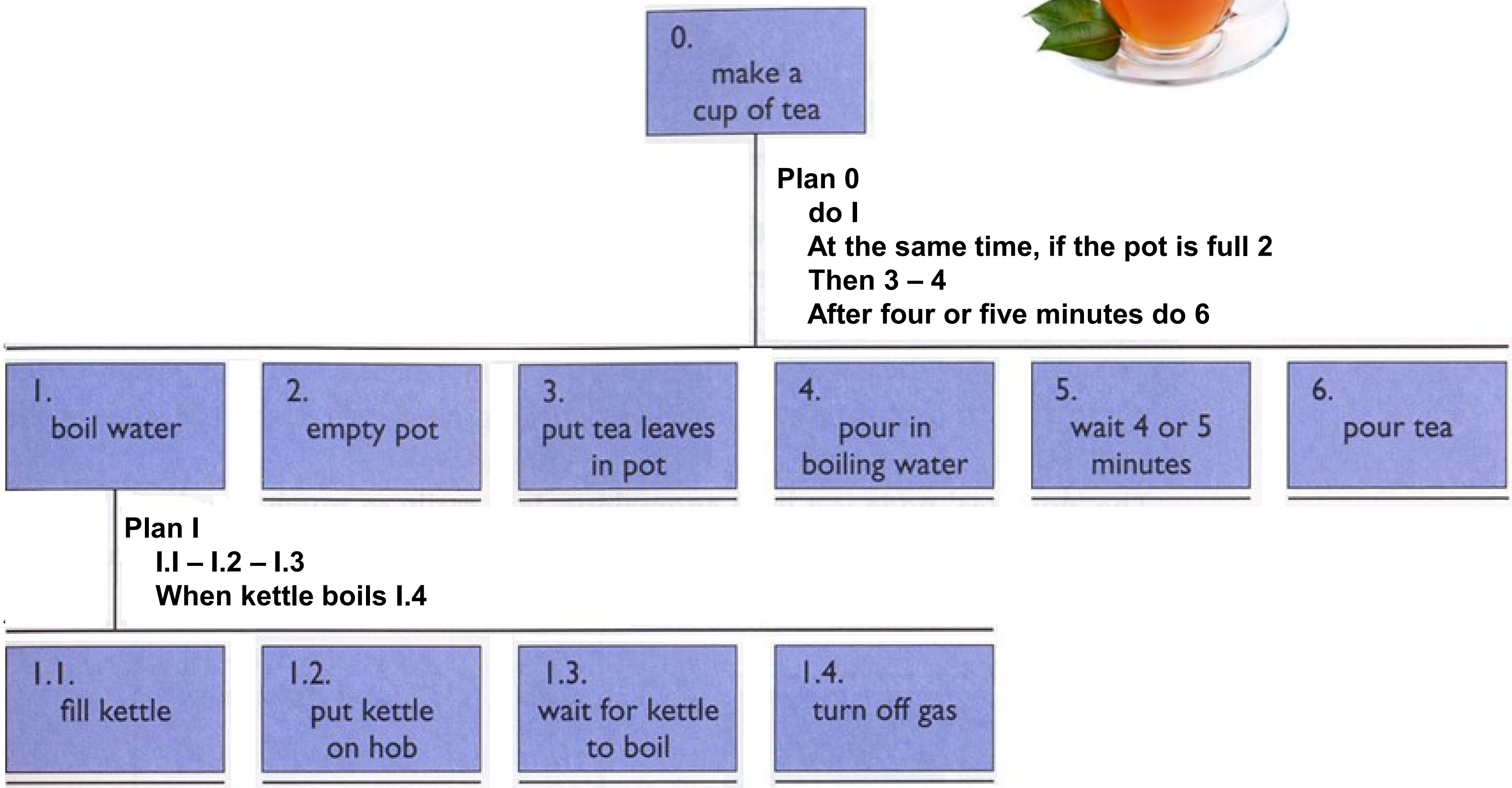


Another example:

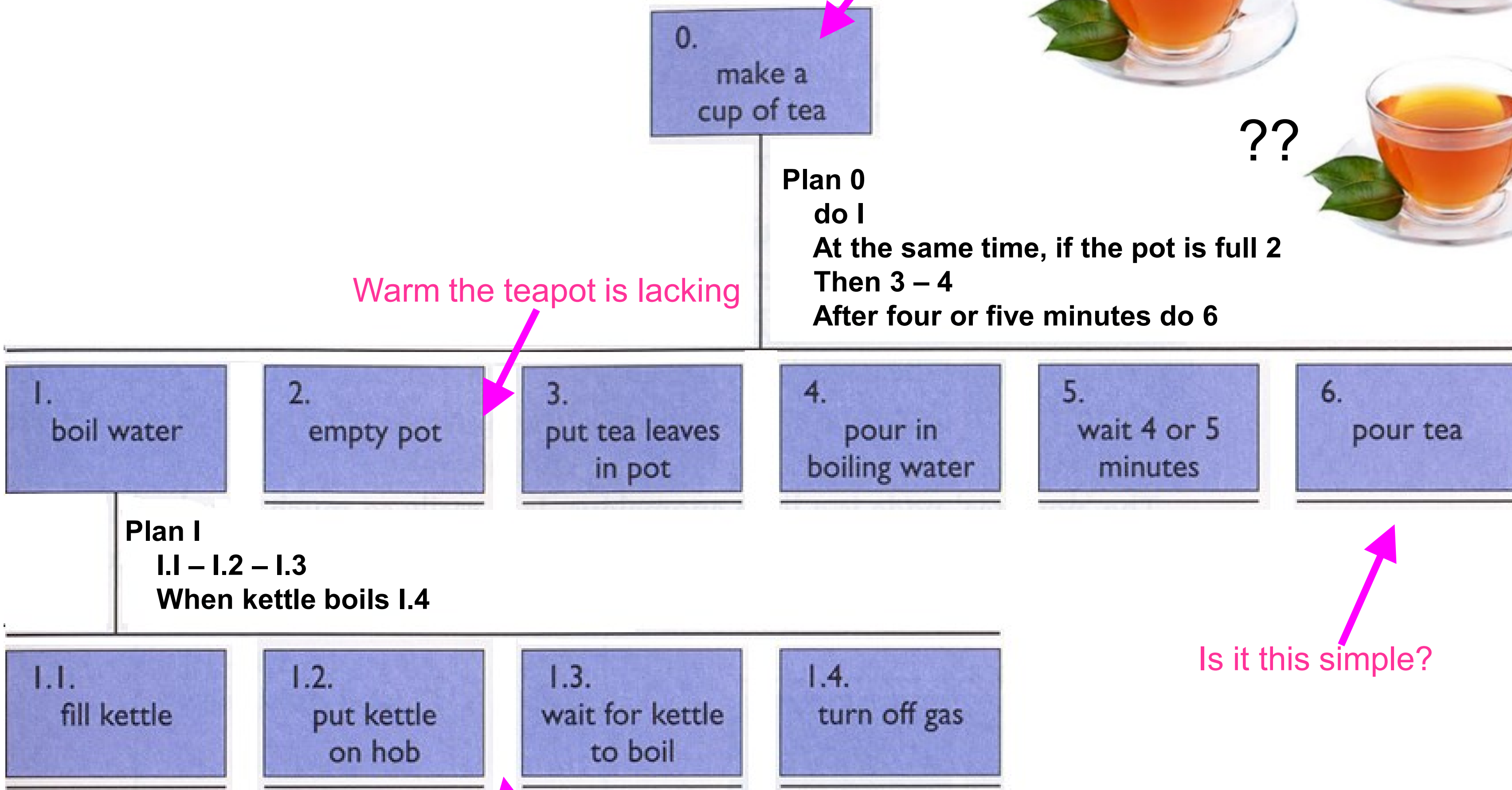
Preparing a cup of tea

Can you do a HTA describing this task?

HTA- Preparing a cup of tea
(graphical representation
of first approach)

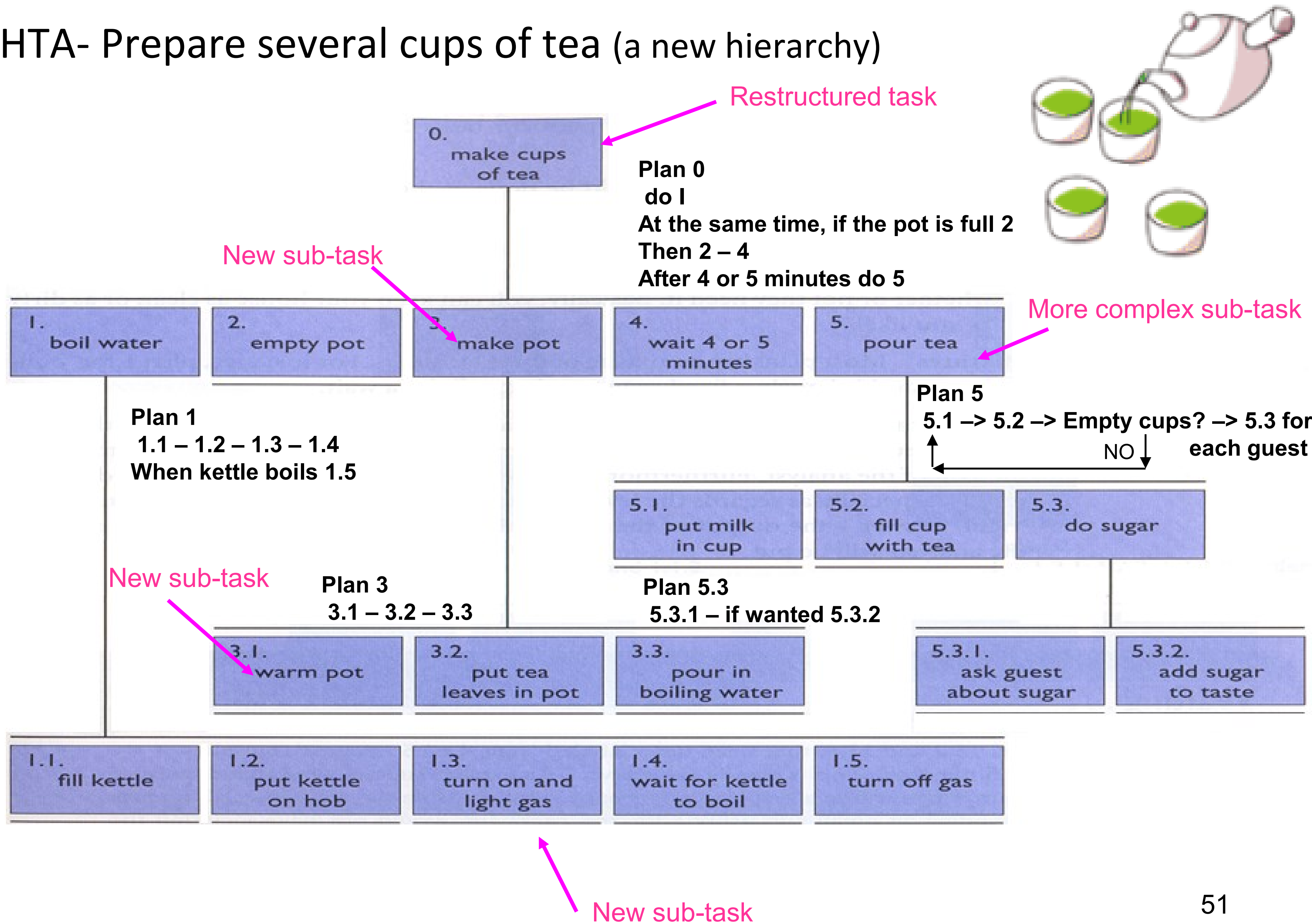


HTA- Preparing a cup of tea
(analysis of the of first approach)



??

HTA- Prepare several cups of tea (a new hierarchy)



Plan types in HTA

- Fixed sequence (plan 3 – prepare the teapot)
- Optional tasks (5.3 sugar?)
- Waiting for events (4- wait 4 or 5 minutes)
- Cycles (plan 5 – serve tea)
- Time sharing (1 and 2 prepare teapot, boil water)
- Random (vacuum cleaning rooms)
- Mix of several types

- The result of the analysis depends a lot on the experience of the analyst

- **Different analysts usually produce different results** (mainly at the detail level) varying with the goal of the analyst

Task analysis information sources

- The quality of task analysis results cannot be better than the original data

  “*garbage in garbage out*”

- The process of analysis in general triggers new questions, thus <u>several phases</u> of data collection and analysis are needed

- There are several types of information sources:

  – Documentation
  – Observation (expensive)
  – Interviews

Documentation:

- Manuals, instruction books, training documentation ... are very good information sources

- But they describe what people are <u>supposed to do</u>, not what they actually do

- System's manuals usually describe functionality, not how they are used

- Observation and user interviews should be performed based on this information

- Be careful with user interviews!!

Interviews:

- Interviewing domain experts is a good way of getting information about the task; should include:
  - General questions (e.g. a typical day)
  - Specific questions (e.g. why did you do that?)
  - Task decomposition (~ HTA)

Observation:

- It is always necessary to perform (formal or informal) observation to understand the tasks
- Reading documentation and observing users is a good starting point
- More observation should follow:
    - In the lab
    - In the field

    - Passive (only observation)
    - Active (questions, post-task walkthrough)

Using Task Analysis:

- May be used in:

  - Manuals and teaching materials

  - High-level system design

  - Detailed design of the system user interface

- In the first case users are observed while performing tasks using the system

- In the other cases task analysis contributes to the design of the new system

Main bibliography

- Alan Dix, J. Finley, G. Abowd, R. Beale, *Human-Computer Interaction*, 3rd ed., Prentice Hall, 2004
- Dan Diaper, Neville Stanton, *The Handbook of Task Analysis for Human-Computer Interaction*, CRC Press, 2003
  http://www.amazon.com/Handbook-Task-Analysis-Human-Computer-Interaction/dp/0805844333/ref=sr_sp-atf_title_1_1?s=books&ie=UTF8&qid=1396779216&sr=1-1&keywords=Handbook+task+analysis#reader_0805844333
- Ian Sommerville, *Software Engineering*, 10th ed., Pearson, 2016
  https://www.amazon.com/Software-Engineering-10th-Ian-Sommerville/dp/0133943038
- Lene Nielsen, "Personas" In: Soegaard, Mads and Dam, Rikke Friis (eds.). *The Encyclopedia of Human-Computer Interaction*, 2nd Ed. Aarhus, Denmark: The Interaction Design Foundation, 2014
  https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/personas