Benyon text
Chapters 3
& 11.1-11.3

Techniques for Design: Scenarios and Task Analysis

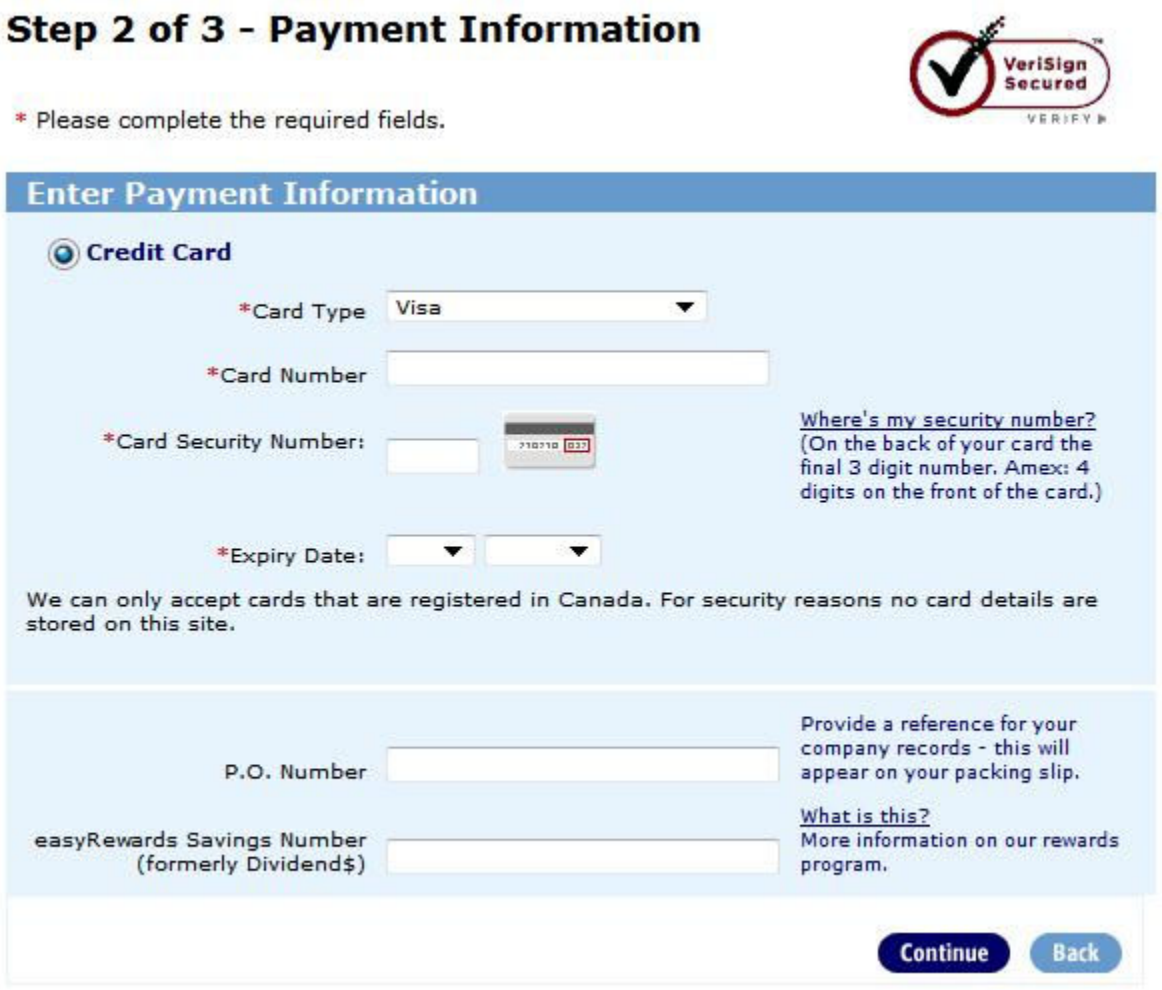
Human Computer Interaction
CSCI 4620 U/G | SOFE 4850
Dr. Christopher Collins

Acknowledgement: Parts of these lectures are based on material prepared by Ron Baecker, Ravin Balakrishnan, John Chattoe, Ilona Posner, Scott Klemmer, and Jeremy Bradbury.

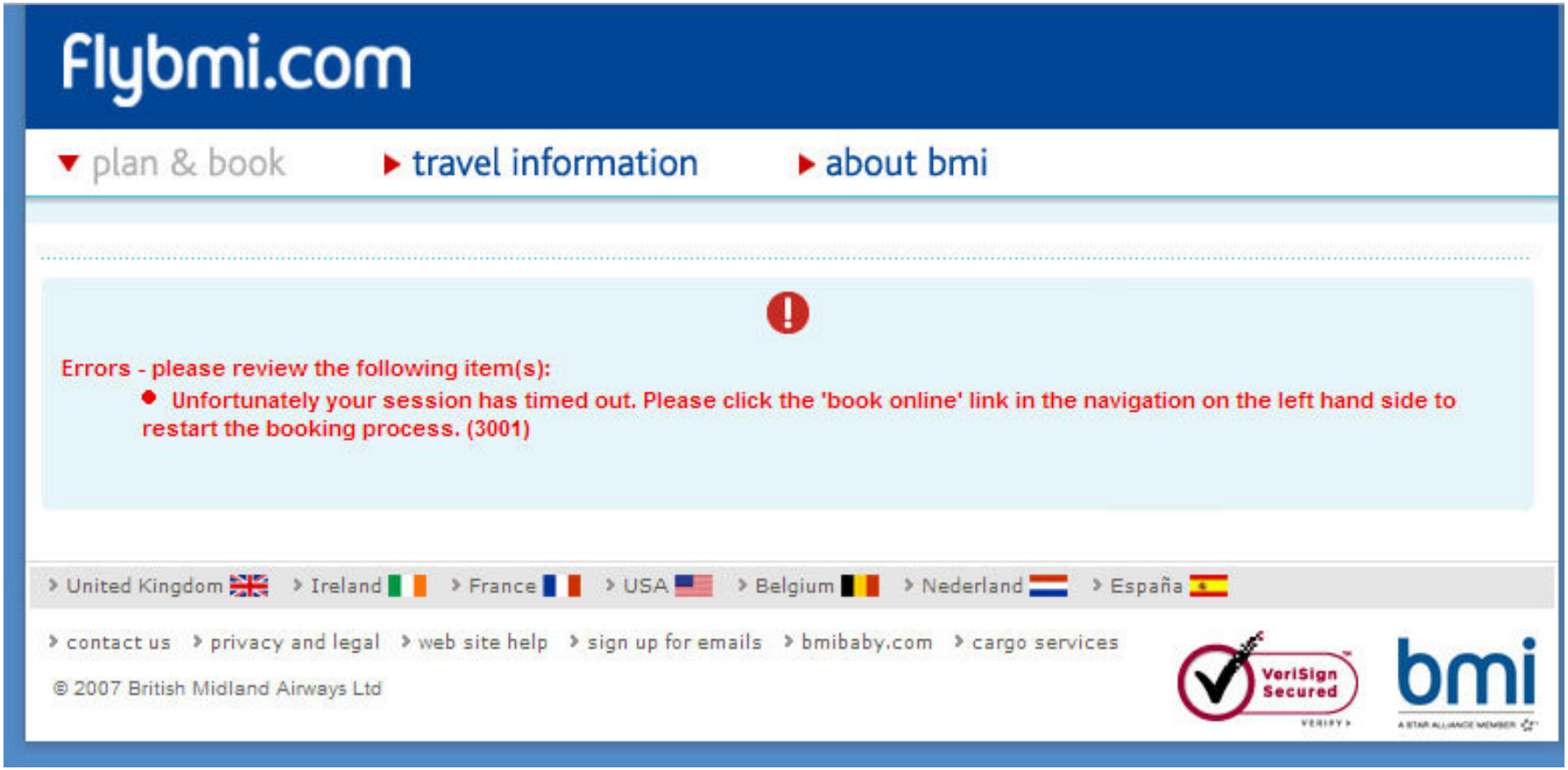
Neilson's Usability Heuristics

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and Standards
- Help users recognize, diagnose, and recover from errors
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help and documentation

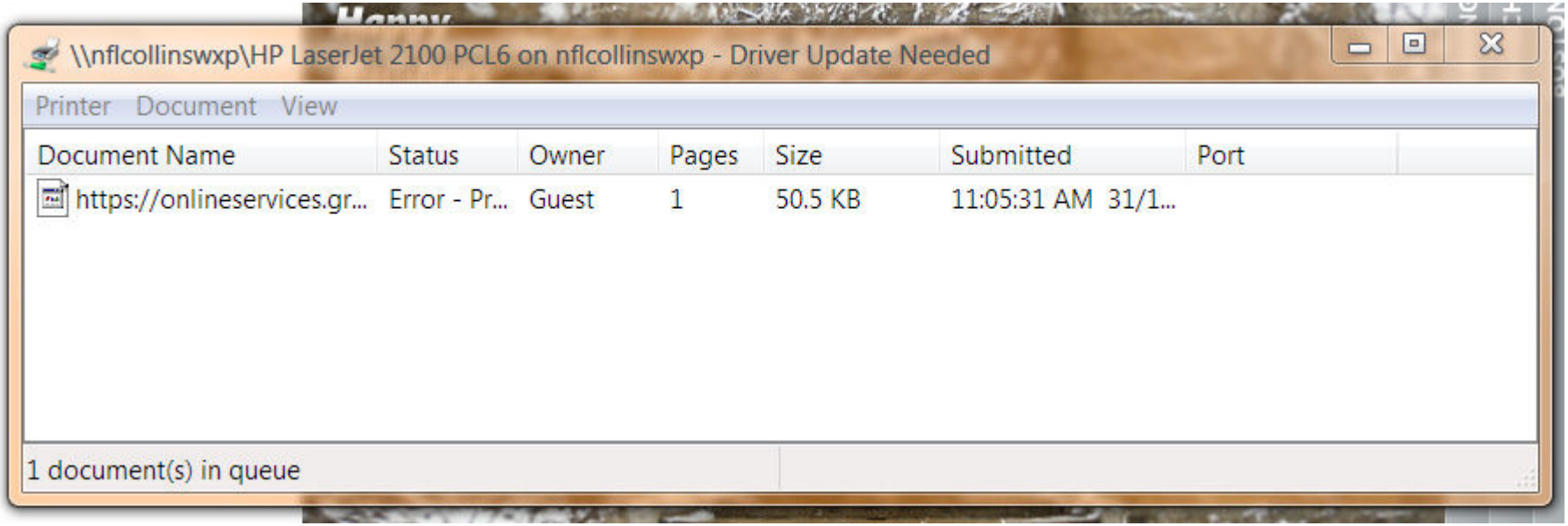
Purchasing Sequence



Website Timeout



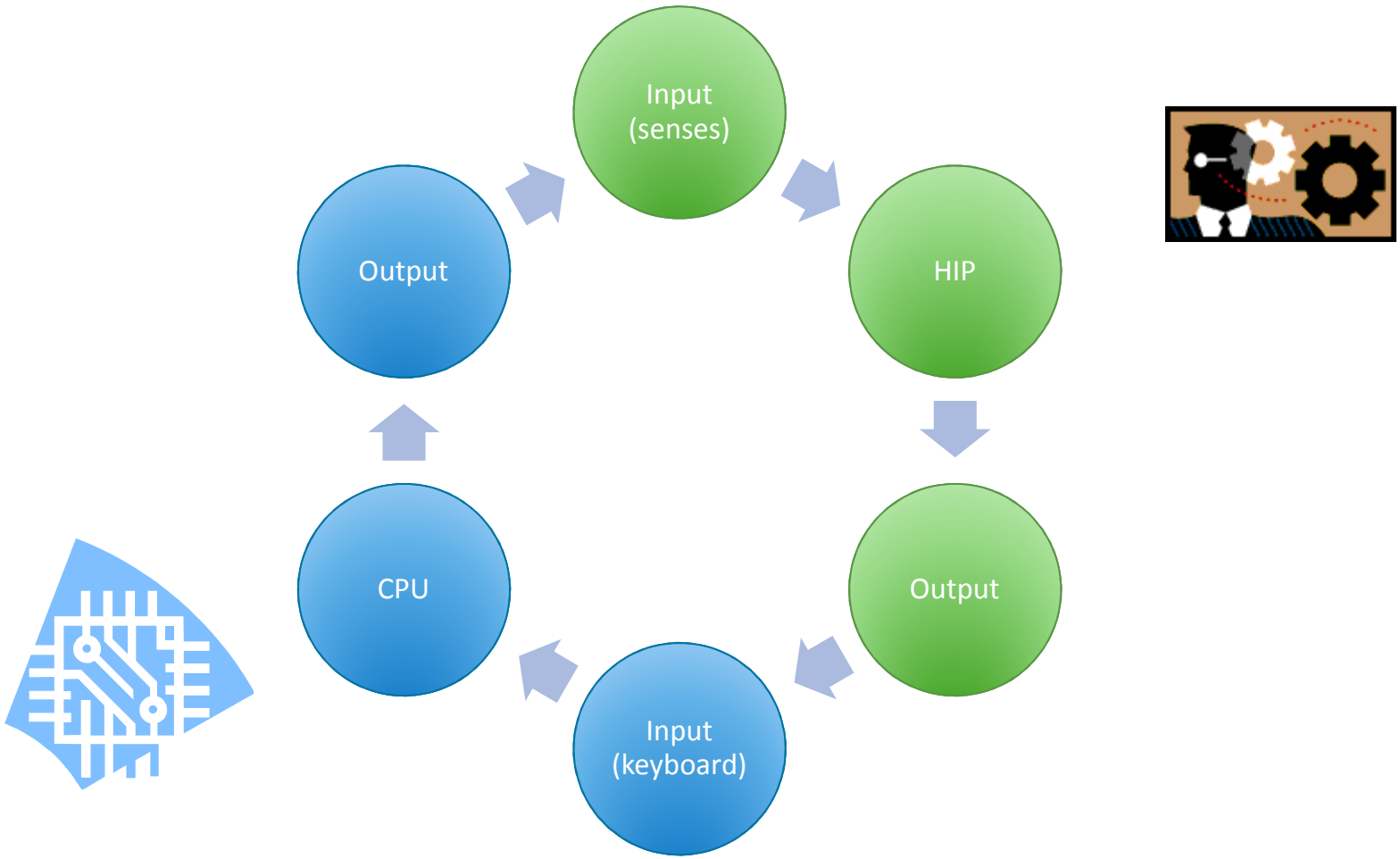
Printer Error



Last Time

- Models of Interaction
- Personas

Human Information Processing

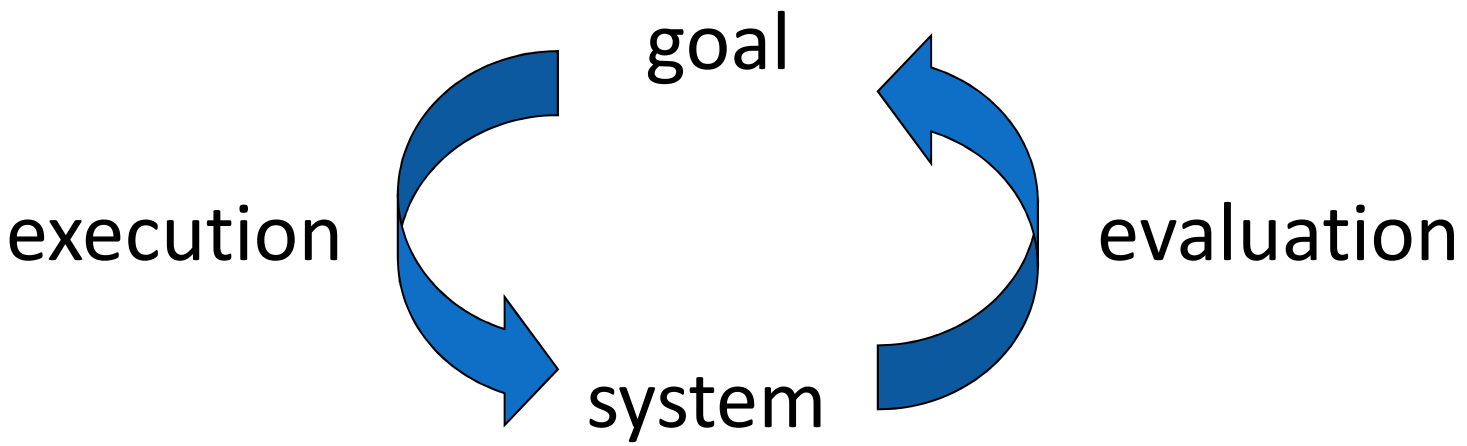


Norman's Model of Interaction

- A 7 stage model of the execution-evaluation cycle:
  1. Goal is established
  2. Intention is formed
  3. The action sequence is specified
  4. The action is executed
  5. The system state after the action is perceived
  6. The perceived system state is interpreted
  7. The system state is evaluated based on the original goals and intentions (see stages 1 & 2)

Modeling interaction

Norman's Model of Interaction



(Source: Dix, Finley, Abowd, Beale*, "Human-Computer Interaction")*

Techniques for Design

- Formal methods:
  - Stakeholders
  - Personas
  - Scenarios
    - Stories, conceptual scenarios, concrete scenarios, use cases
  - Task Analysis
- For describing:
  - *the current situation* (to motivate design)
  - *the proposed design solution* (to drive implementation of prototypes, and test for errors)

Explore Personas

- http://www.aegis-project.eu/index.php?option=com_content&view=article&id=63&Itemid=53
- Or http://bit.ly/dvxsr7

Announcements

- Midterm exam (in class) Oct 17
- Remember: J 123A has open study hours!

Today

- In today's lesson we will review techniques for motivating design:
  - Scenarios
  - Task analysis
  - Requirements

Understanding

Understanding

- Understanding is concerned with what the system has to do, what it has to be like and how it has to fit in with other things; with the requirements of the product, system or service.
- Designers need to research the range of people, activities and contexts relevant to the domain they are investigating so that they can **understand** the requirements of the system they are developing.
- They need to **understand** the opportunities and constraints provided by technologies.

Techniques for Design

TASK ANALYSIS

Task Analysis Goals

- To uncover existing:
  - user characteristics
  - work practice
  - system usage
  - work environment
  - tasks performed
  - language used
  - objectives for usage of a product
  - qualitative data: emotions, mental models
  - quantitative data: how, how often, how long

Task Analysis - Users

- Learn Stakeholder Characteristics
  - Task experience and domain knowledge, e.g., by radiologists, truck drivers, call center workers
  - Computer literacy, e.g., systems & application experience
- Understand Users' Conceptual Model
  - Task structures and organizational patterns, e.g., order taking, order entry, shipping, billing
  - Artifacts or objects used in tasks, e.g., files, forms
  - Organization of artifacts, e.g., in a digital library, page->section->chapter->book->library

Task Analysis - Work

- Understand Work Flow Patterns
  - Who performs which tasks and how often
  - Communication patterns among workers
- Relationships Between Tasks & Artifacts
  - How specific forms and files are used in order entry
- Use of Information in the Environment
  - Things perceived visually, e.g., materials on hand
  - Things perceived acoustically, e.g., conversations of co-workers, opening of door

Task Analysis - Technology

- Use of other Technologies,
  - e.g., phones, voice mail, fax
- Observational Methods used:
  - Note-taking
  - Audio recording
  - Video recording
  - Think-aloud protocols

Formal Techniques

- Personas
- Scenarios
- Use Cases
- Hierarchical Task Analysis

Techniques for Design

SCENARIOS

Scenarios

- A scenario is a description of people using artifacts within environments carrying out tasks or activities
- Scenarios summarizing field study work focus on current state of users activities and exclude the proposed system
- Scenarios are typically expressed in plain English
  - But they can be done graphically, i.e., with storyboards

Scenarios

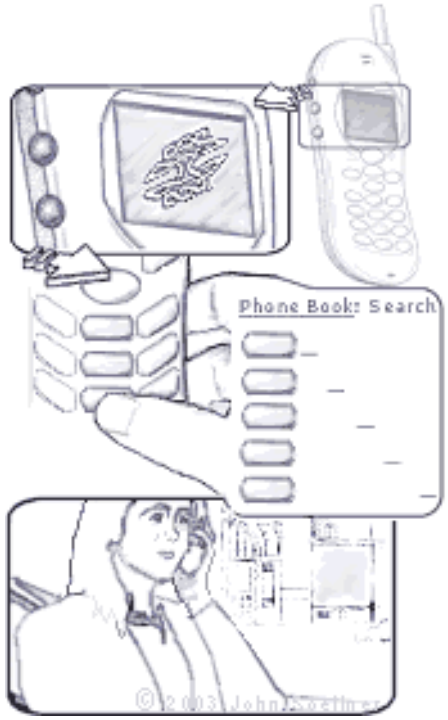
- Example:
  - Bob is a project manager in an advertising company, who often works from home. One day he forgot his day-timer at home and had meetings with clients across town. While attending an important meeting he realized that he was about to miss his lunch appointment, but he didn't have the phone number to cancel his scheduled lunch.

Day-in-the-life scenarios

- Characterize what happens when people perform typical tasks
- Can be acted out as a form of walkthrough
- Can be shown to study participants for confirmation of accuracy

Storyboard Examples

- Go online and find examples of sketched storyboards.
- Post links to “storyboard” thread in general discussion.



Persona in Scenario

Bjorn

- Age 32
- Office worker (ad account manager)
- No children, lives alone
- Dog died (used to walk it for exercise)
- Starting to put on weight
- Used to play football at university, much less active now
- Active social life
- 'I want to stay fit, but on my own time and fitting in to my own schedule'

1. Home from work, he was meant to go out the previous evening but got invited out to a dinner party instead. This evening is now free, so he decides to go for a run.
2. He's in his living room and sets up for his run. This involves:
   - route choice
   - exercise level, eg easy jog or hard run (specific pacing feedback choice, eg within PB)
   - music choice
   - disturbability status (eg, open to contact/running partner)
   - weather
   - (warm up/stretching?)
3. He gets changed and leaves the house, the handover is transparent from living room companion to mobile device-based companion and is aware of all Bjorn's choices regarding run setup.
4. Just as he's about to begin, the sun breaks through the clouds and Bjorn decides he'd rather go for a longer run than initially selected in his living room; this change must be facilitated through his mobile companion device. Selective rather than creative process (eg, chose run three on route 2).
5. He starts running hard.
6. Asked whether he's warmed up as he's running above a warm-up rate.
7. He slows down to a more gentle jog and reaches his start point.
8. A touch of the device indicates he's starting his run.
9. Music begins.
10. Pace-setting tactile feedback begins.
11. Midway through run he's informed that Julie is also running in the woods and has set her HFC at open to running partners (this is a closed list of the pre-set social network that Bjorn belongs to).
12. He slows down and runs on the spot and sends her a greeting, asking if she'd like to join him; she says yes
13. She catches up and the companion automatically reconfigures his pacing settings to match hers.
14. After a circuit they part ways and Bjorn heads home.
15. On entering the house Bjorn warms down and stretches which induces a brief summary on his mobile device whilst the detailed data from his run is transparently transferred to his home network.
16. He walks into the kitchen to grab a glass of water and plan what to make for dinner. His home companion notes that he went for a long run today so he must be hungry, and suggests some recipes based on what he has in his fridge: 'how about the steak, it goes out of date tomorrow'. Nothing takes his fancy so he asks the companion to search online whilst he has a shower. Takes shower, comes down and is presented with some new recipes and the fact that Julie called and asked him for a drink that night.
17. At a later time he asks for an overview of his past three months' exercise. His companion notes that his heart rate is recovering quicker which suggests he's getting fitter, but for the past two weeks he's not been running for as long.

Figure 3.6 The Bjorn persona for HFC scenario

Types of Scenarios

- Stories are the real-world experiences of people.
- Conceptual scenarios are more abstract descriptions in which some details have been stripped away.
- Concrete scenarios are generated from abstract scenarios by adding specific design decisions and technologies and once completed these can be represented as use cases.
- Use cases are formal descriptions that can be given to programmers.

Scenarios at different stages

- Many stories will be represented by a few conceptual scenarios. However, each conceptual scenario may generate many concrete scenarios.
- Several concrete scenarios will be represented by a single use case.
- Designers abstract from the details of stories to arrive at conceptual scenarios.
- They specify design constraints on conceptual scenarios to arrive at concrete scenarios.
- Finally they formalize the design ideas as use cases.

Stories

- Stories are the real-world experiences, ideas, anecdotes and knowledge of people.
- These may be captured in any form and comprise small snippets of activities and the contexts in which they occur.
- This could include videos of people engaged in an activity, diary entries, photographs, documents, the results of observations and interviews and so on.
- People's stories are rich in context.
- Stories also capture many seemingly trivial details that are usually left out if people are asked to provide more formal representations of what they do.

Story: Example

I needed to make an appointment for Kirsty, my little one. It wasn't urgent – she had been having a lot of bad ear-ache every time she had a cold – but I did want to see Dr. Fox since she's so good with the children. And of course ideally it had to be when Kirsty was out of school and I could take time off work. I rang the surgery and the receptionist told me that the next appointment for Dr. Fox was the next Tuesday afternoon. That was no good since Tuesday is one of my really busy days so I asked when the next one was. The receptionist said Thursday morning. That meant making Kirsty late for school but I agreed because they sounded very busy – the other phone kept ringing in the background – and I was in a hurry myself. It was difficult to suggest a better time without knowing which appointments were still free.

Conceptual scenarios

- Conceptual scenarios are more abstract than stories.
- Much of the context is stripped away during the process of abstraction and similar stories are combined together.
- Useful for generating design ideas and for understanding the requirements of the system.
- Once the designer has accumulated a collection of stories, common elements will start to emerge.

Abstraction

- Classification and aggregation: moving from the details of specific people undertaking specific activities in a specific context using a particular piece of technology to a more general description that still manages to catch the essence of the activity.

Aggregation

- Aggregation is the process of treating a whole thing as a single entity rather than looking at the components of something.
- In most domains, for example, one would aggregate a screen, processor, disc drive, keyboard and mouse and treat this as a single thing – a computer – rather than focusing on the components

Classification

- Classification is the process of recognizing that things can be collected together, so that dealing with the class of things is simpler (more abstract) than dealing with the individual things.
- There are no set ways to classify things, so the analyst has to work with the stories that have been gathered and with the users themselves to decide which things belong together and why.

Conceptual Scenario Example

Booking an appointment: People with any degree of basic computer skills will be able to contact the doctors' office at any time via the Internet and see the times which are free for each doctor. They can book a time and receive confirmation of the appointment.

Conceptual Scenario Example

- Little or no specification of precise technologies or how the functions will be provided.
- Could be made
  - more abstract by not specifying that the Internet should be used
  - more concrete by specifying that the booking should be made from a computer rather than a mobile phone.
- Finding an appropriate level of abstraction at which to describe things for a given purpose is a key skill of the designer.

Concrete scenarios

- Each conceptual scenario may generate lots of concrete scenarios.
- Most useful in design (not in understanding the context)
- When designers are working on a particular problem or issue they will often identify some feature that applies only under certain circumstances.
- At this point they may develop a more specific elaboration of the scenario and link it to the original.
- Thus one reasonably abstract scenario may spawn several more concrete elaborations which are useful for exploring particular issues.

Concrete scenarios

- Notes can be added to scenarios that draw attention to possible design features and problems.
- Concrete scenarios also begin to dictate a particular interface design and a particular allocation of functions between people and devices.
- Concrete scenarios are particularly useful for prototyping and envisioning design ideas and for evaluation because they are more prescriptive about some aspects of the technology.
- However, there is not a clean break between conceptual and concrete scenarios.
- The more specific the scenario is about some aspects, the more concrete it is.

Concrete scenarios: Example

In the example below, decisions have now been taken concerning drop-down menus, the fact that the next two weeks' details are to be shown, and so on. However, the notes following the scenario show that there are many design decisions still to be taken.

Booking an appointment/01

Andy needs a doctor's appointment for his young daughter Kirsty in the next week or so. The appointment needs to be outside school-time and Andy's core working hours, and ideally with Dr. Fox, who is the children's specialist. Andy uses a PC and the Internet at work, so has no difficulty in running up the appointments booking system. He logs in [1] and from a series of drop-down boxes, chooses to have free times for Dr. Fox [2] displayed for the next two weeks [the scenario would continue to describe how Andy books the appointment and receives confirmation]

Notes to booking an appointment/01

1. Is logging in necessary? Probably, to discourage bogus access to the system, but check with the doctor.
2. Free times can be organized by doctor, by time of day, or by next available time. Drop-down boxes will save screen space but may present problems for less experienced users or those with poor eyesight.

Techniques for Design

FORMAL SCENARIOS: USE CASES

Use Cases

- Focus on human-computer *interaction*
- Sequence of steps (the “normal course”)
  - Alternative paths
- In narrative style (as a story), with a high level of details, a use case is called a *scenario.*

Use Case: Arrange a Meeting

1. The users chooses the option to arrange a meeting.
2. The system prompts user for the names of attendees.
3. The user types the list of names.
4. The system checks that the list is valid.
5. The system prompts the user for meeting constraints.
6. The user types in meeting constraints.
7. The system searches the calendars for a date that satisfies the constraints.
8. The system displays a list of potential dates.
9. The user chooses one of the dates.
10. The system writes the meeting into the calendar.
11. The system emails all the meeting participants informing them of the appointment.

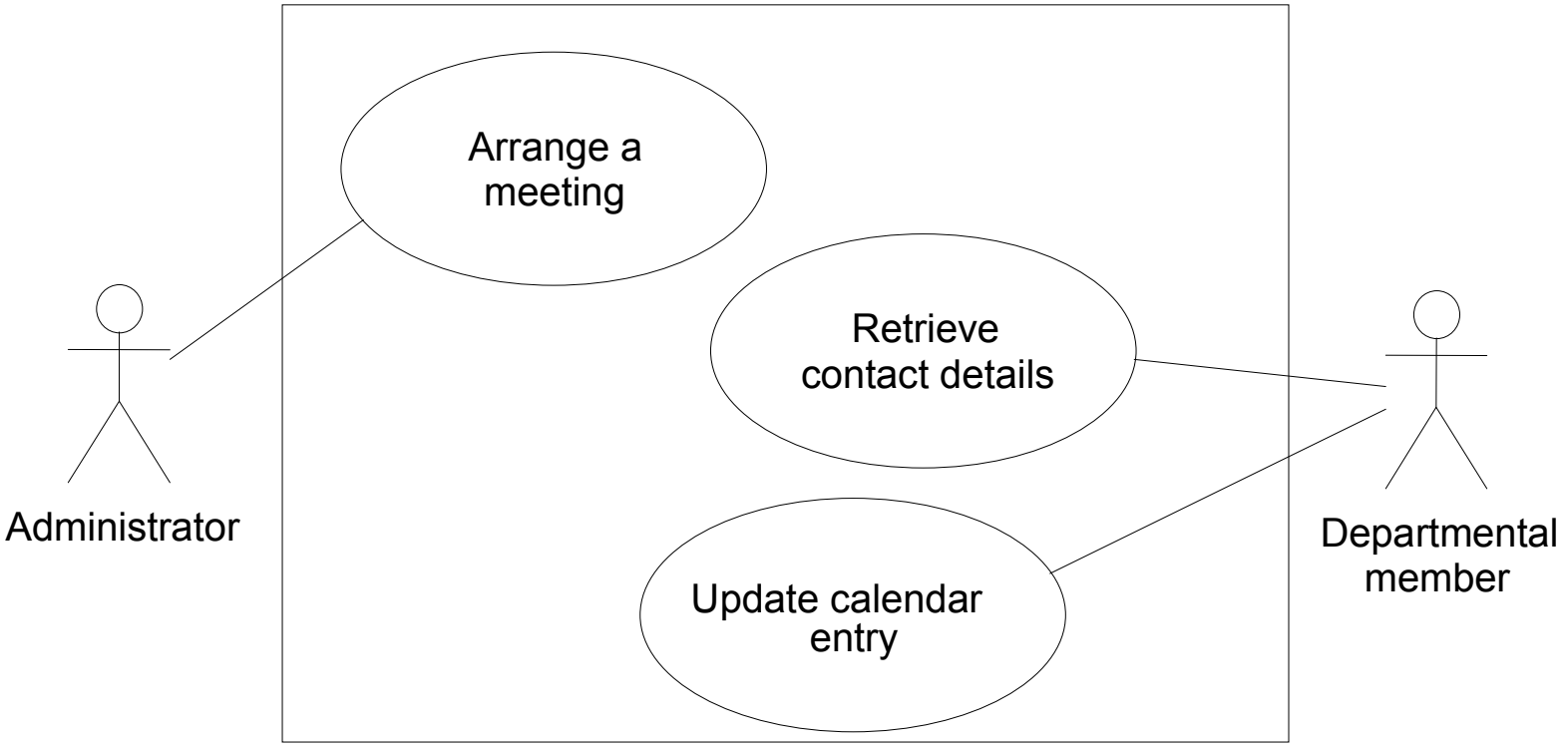
Arrange a Meeting w/ Alternatives

1. The users chooses the option to arrange a meeting.
2. The system prompts user for the names of attendees.
3. The user types the list of names.
4. The system checks that the list is valid.
    1. If the list is not valid: The system displays an error message.
       The system returns to step 2.
5. The system prompts the user for meeting constraints.
6. The user types in meeting constraints.
7. The system searches the calendars for a date that satisfies the constraints.
8. The system displays a list of potential dates.
    1. If no potential dates are found: The system displays a suitable message.
       The system returns to step 5.
9. The user chooses one of the dates.
10. The system writes the meeting into the calendar.
11. The system emails all the meeting participants informing them of the appointment.

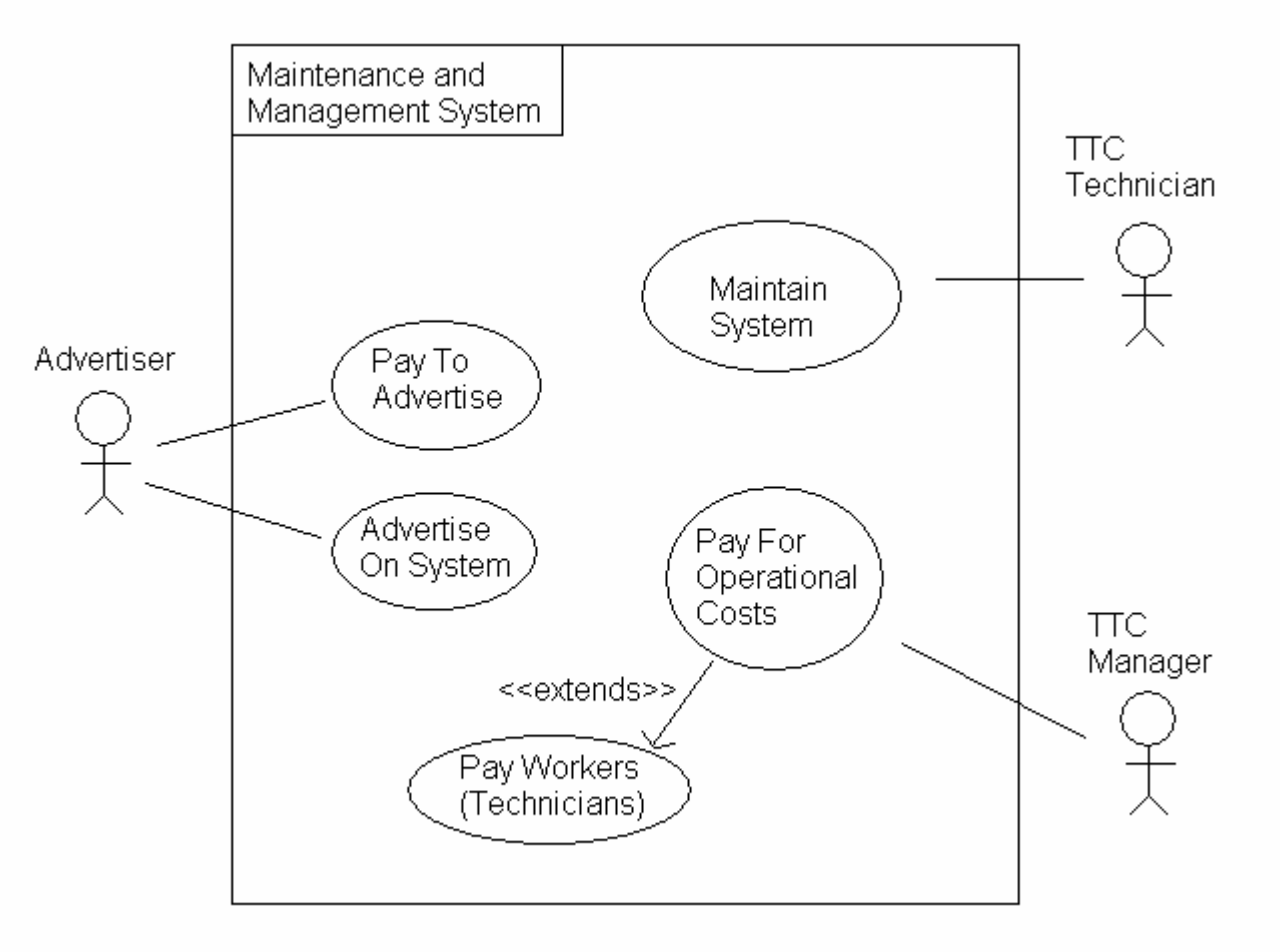
Use Case Diagrams

- Describe a use cases graphically
- Involves one or more ‘actors’ interacting with the system (in the box)
- Each use case may be associated with more than one actor

Use Case Diagram for Shared Calendar



Use Case Diagram for “WheelConnect”



Essential Use Cases

- More general than scenarios
- Avoid the use case assumption that any technology already exists
- 3 part narrative:
  - A name that expresses the overall user intention
  - A stepped description of user actions
  - A stepped description of system responsibilities

Essential Use Case for Shared Calendar

arrangeMeeting

USER INTENTION	**SYSTEM RESPONSIBILITY**
arrange a meeting	
	request meeting attendees & constraints
identify meeting attendees & constraints	
	search calendars for suitable dates
	suggest potential dates
choose preferred date	
	book meeting

Source: Constantine and Lockwood (1999) also in PRS

Essential Use Cases

- In comparison to a scenario-style use case, the essential use case:
  - Doesn’t say anything about specific technology, e.g. typing a list of names
  - Generalizes to any suitable system, e.g. selecting names from a list

Activity: Make an Essential Use Case

Locate a library book

1. The system prompts for user name and password.
2. The user enters his or her user name and password.
3. The system verifies the user's password.
4. The system displays a menu of choices.
5. The user chooses the search option.
6. The system displays the search menu.
7. The user chooses to search by author.
8. The system displays the search author screen.
9. The user enters the author's name.
10. The system displays the search results.
11. The user chooses the required book.
12. The system displays the details of the chosen book.
13. The user notes the location.
14. The user quits the catalogue system.

Essential Use Case

locateBook

USER INTENTION	**SYSTEM RESPONSIBILITY**
identify self	
	verify identity
	request appropriate details
offer known details	
	offer search results
note search results	
quit system	
	close

HIERARCHICAL TASK ANALYSIS

Textual HTA description

Hierarchy description ...

0. in order to clean the house
    1. get the vacuum cleaner out
    2. get 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 vacuum cleaner and attachments away

... and plans

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

Note: only the plans denote order

Generating the Hierarchy

1. get list of tasks
2. group tasks into higher level tasks
3. decompose lowest level tasks further

Stopping rules

- How do we know when to stop?
- Is “empty the dust bag” simple enough?
- Purpose: expand only relevant tasks
- Motor actions: lowest sensible level

Tasks as Explanation

- imagine asking the user the question:

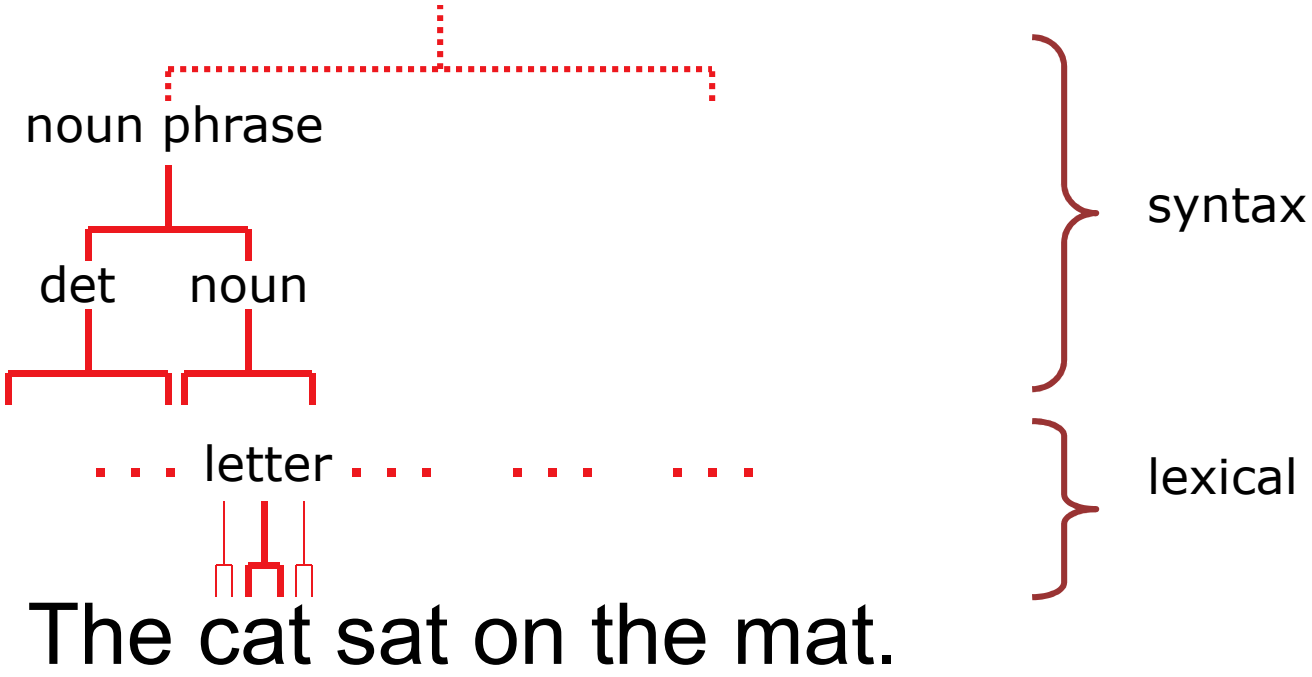
  what are you doing now?

- for the same action the answer may be:

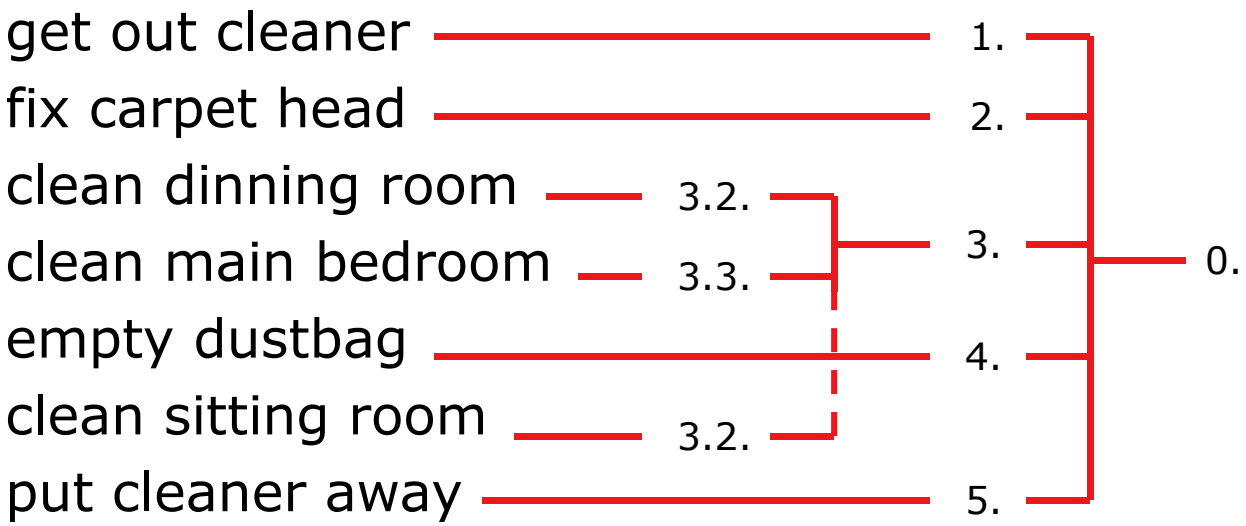
  typing ctrl-B
  making a word bold
  emphasising a word
  editing a document
  writing a letter
  preparing a legal case

HTA as Grammar

- can parse sentence into letters, nouns, noun phrase, etc.

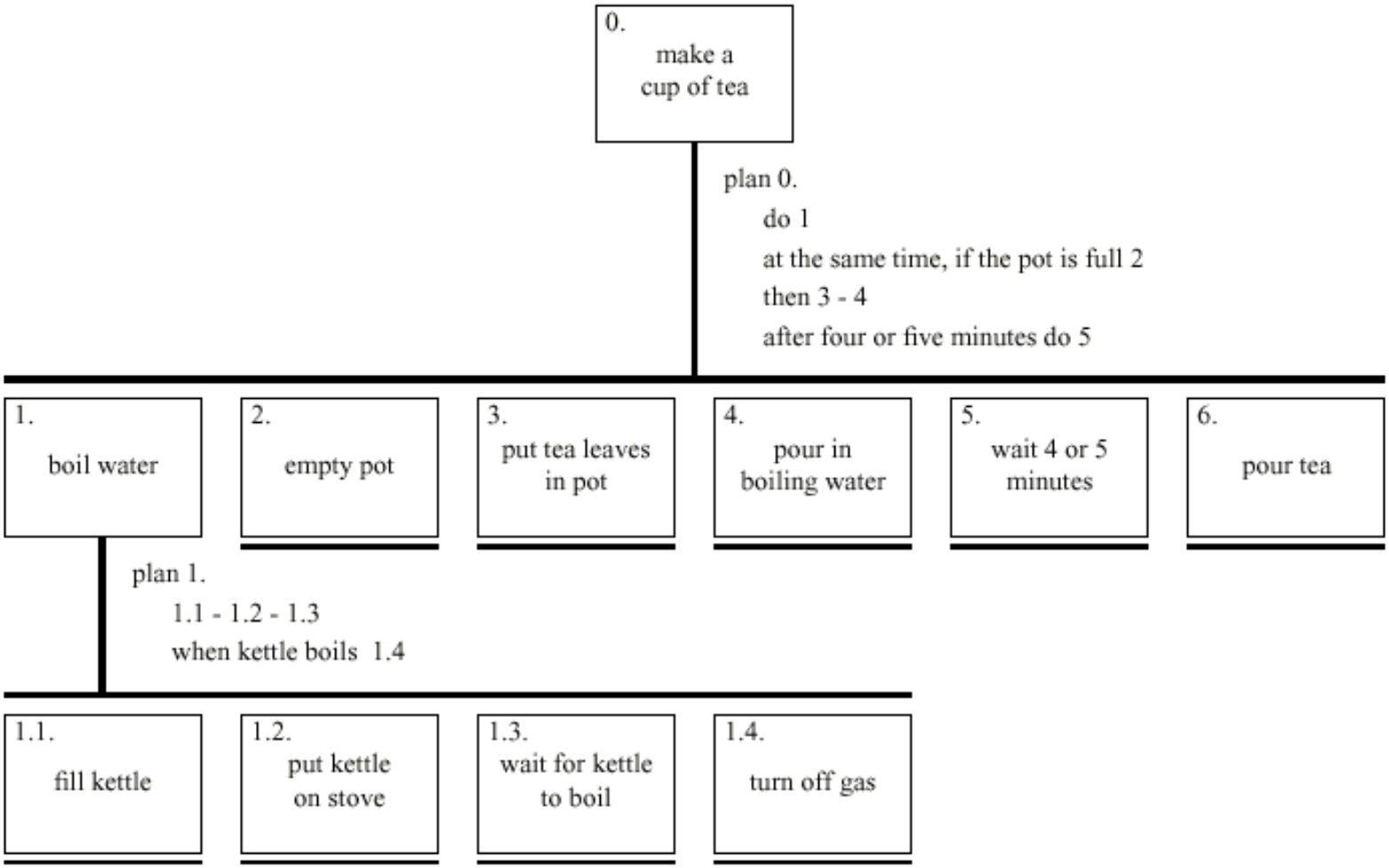


parse scenario using HTA



0. in order to clean the house
    1. get the vacuum cleaner out
    2. get 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 vacuum cleaner and attachments away

Diagrammatic HTA



Refining the description

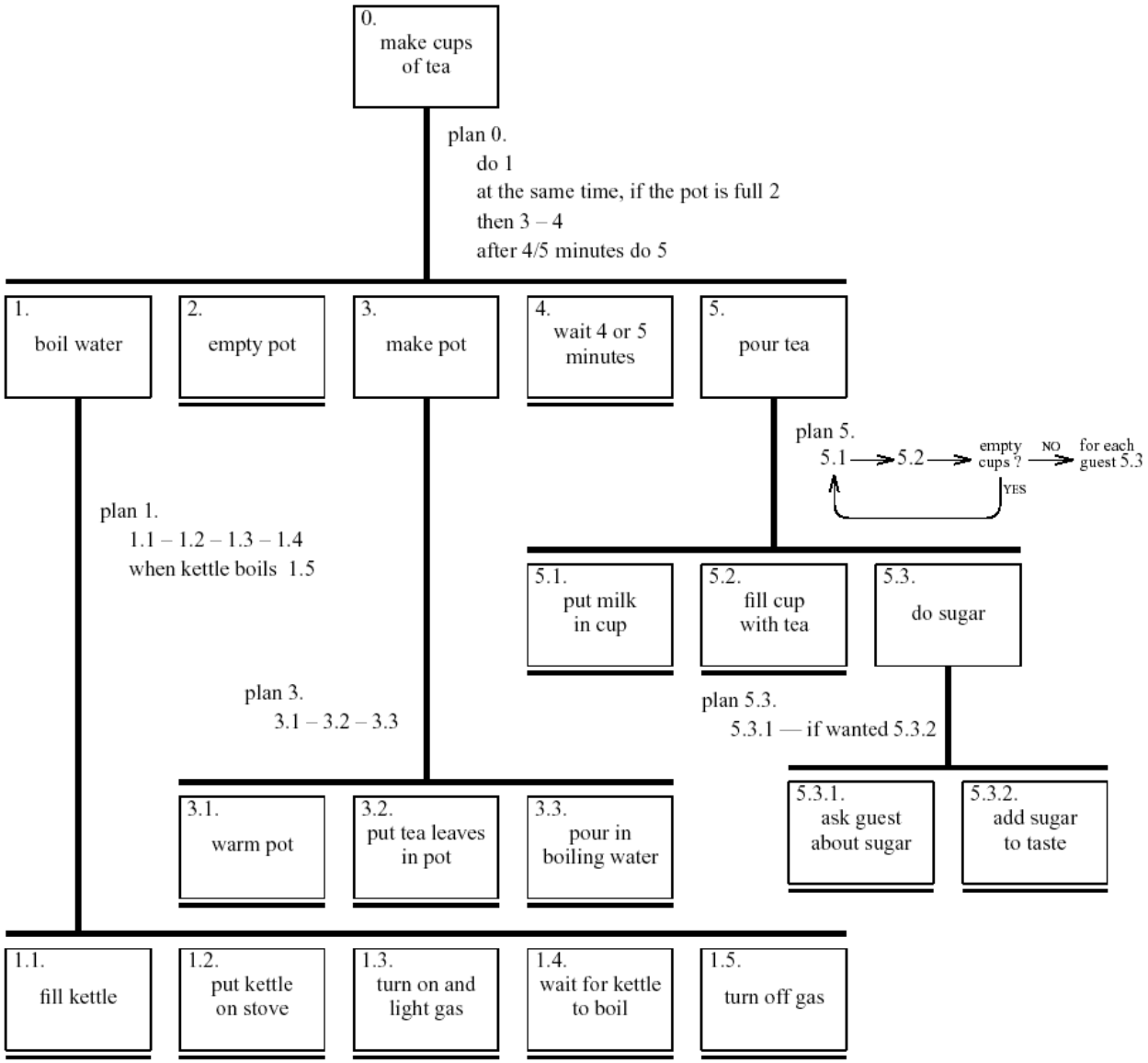
Given initial HTA (textual or diagram)

How to check / improve it?

Some heuristics:

paired actions	e.g., where is `turn on gas'
restructure	e.g., generate task `make pot'
balance	e.g., is `pour tea' simpler than making pot?
generalise	e.g., make one cup ….. or more

Refined HTA for Making Tea

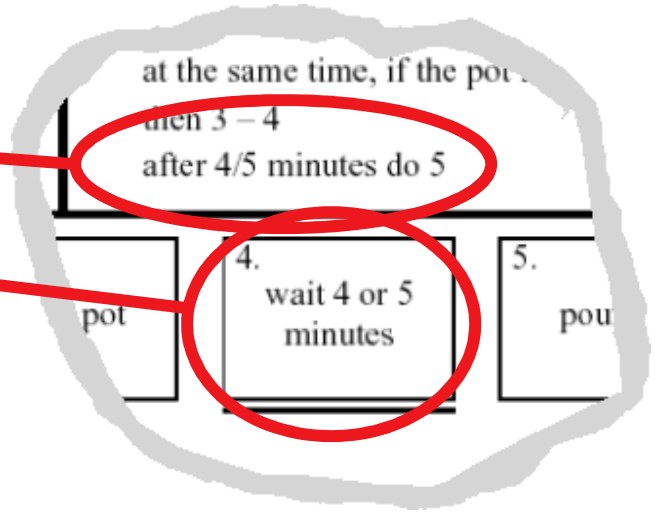


Types of plan

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

waiting …

- is waiting part of a plan?
  … or a task?
- generally
  - task – if ‘busy’ wait
    - you are actively waiting
  - plan – if end of delay is the event
    - e.g. “when alarm rings”, “when reply arrives”
- in this example …
  - perhaps a little redundant …
  - TA not an exact science



Hierarchical Task Analysis

0. In order to borrow a book from the library
   1. go to the library
   2. find the required book
      2.1 access library catalogue
      2.2 access the search screen
      2.3 enter search criteria
      2.4 identify required book
      2.5 note location
   3. go to correct shelf and retrieve book
   4. take book to checkout counter

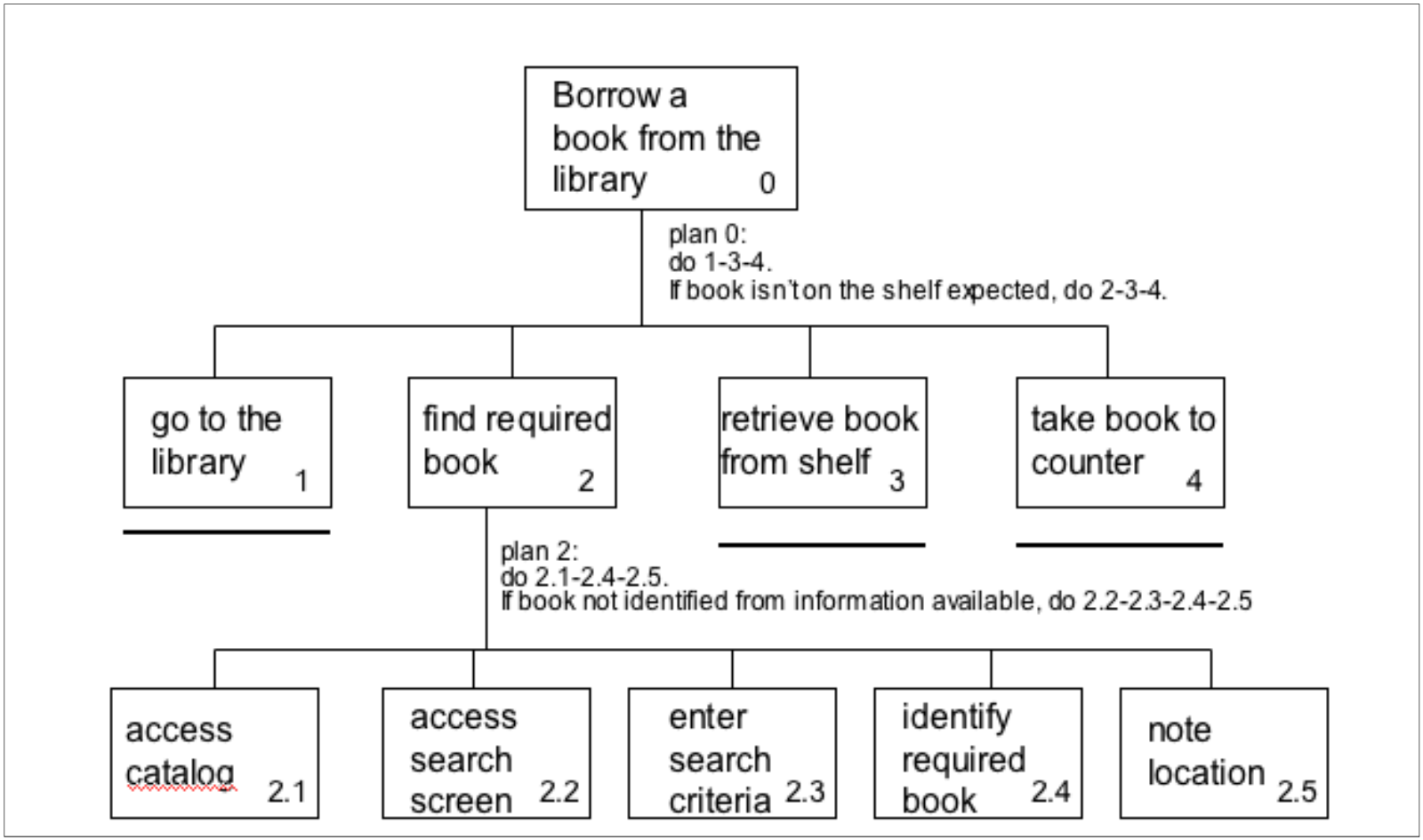
Hierarchical Task Analysis (HTA)

plan 0: do 1-3-4. If book isn't on the shelf expected, do 2-3-4.

plan 2: do 2.1-2.4-2.5. If book not identified do 2.2-2.3-2.4.

HTA Plan

HTA Graphical Representation



Techniques for Design

REQUIREMENTS

Requirements Categories

- stakeholder needs
- environmental requirements
- functional requirements
- technical requirements, and
- usability requirements

Stakeholder Needs

- Characteristics the user must possess (e.g. must know how to use a web browser)
  - Start from what users do, how they do it (field studies)
- High level needs
  - recall Maslow's Hierarchy
  - May need satisfaction, comfort, reliability, social connections, safety

Environmental Requirements

- Context of use
- Physical environment
- Social environment
  - e.g. requires login by three people who are friends
- Organizational environment
  - e.g. requires a professor and a student
  - e.g. requires approval of the manager to be activated

Functional Requirements

- What the new system is to do in general terms
- What specific capabilities are therefore required

Technical Requirements

- Technical requirements, constraints, assumptions
  - Price, size, weight, etc.
  - Compatibility with other technologies, adherence to standards
  - Data requirements

Usability Requirements

- Ease of learning
- Ease of use
- Protection against “errors”
- Specific heuristics which are important

Note: List *specifics* for the project/technology

Evaluating Requirements

- Measures of success
  - Absolute, objective, quantifiable, measurable,
    - "Productivity" improvement of 10% within 1 year
    - Error-free performance in 1 hour without use of manual
  - Subjective
    - Satisfaction expressed by 95% of operators after 6 months
  - Relative to current method, e.g., alternative or no technology

Evaluating Requirements

- Priorities, tradeoffs, & constraints
  - High-end vs. low-end
    - in functionality and price
  - General-purpose vs. special-purpose
  - Ease of use and ease of learning
  - Power and simplicity
  - High-speed and error-free performance

Summary

- Today we introduced:
  - Scenarios
  - Task Analysis
  - Requirements

Your Action Items

- Work on group project part 2b
- Prepare for midterm
- Read assigned textbook chapters

Homework Activity

- Develop a persona which describes a member of a primary stakeholder group for computer-based solutions to assist *group study*
- Using your persona, write a scenario for your persona to achieve the task of *scheduling a group study meeting*
- Post your persona and scenario to *Blackboard discussion Activity: Personas*

Ongoing Course Evaluation

- Please complete the Lecture 9 daily feedback form!