ESTABLISHING REQUIREMENTS
TASK CENTERED DESIGN

CPSC 544 FUNDAMENTALS IN DESIGNING INTERACTIVE COMPUTATION TECHNOLOGY FOR PEOPLE (HUMAN COMPUTER INTERACTION)

WEEK 6 – CLASS 11

© Joanna McGrenere and Leila Aflatoony
Includes slides from Karon MacLean and Jessica Dawson
TODAY

- Task centered design
  - Task description: task example and analysis
- Requirements and metrics
- In class activity
  - Task analysis and description
- Discussion of reading
TASK CENTERED SYSTEM DESIGN

The User
a pretend person who will mold themselves to fit your system

vs.

Mary
a real person with real constraints trying to get her job done
TASKS ARE CENTRAL TO HCI

Recall Cooper’s 3 levels of goal:

Figure 3-4: The three types of user goals

Tasks!
Classic HCI is all about tasks.
in HCI, establishing requirements typically begins with establishing tasks:

- **task examples** describe tasks and users

together with design prototypes, task examples are also a good way to evaluate initial designs at low cost and effort.

**USERs -> this term we are using PERSONAS**

So in this lecture, when you see User, think Persona.
IDENTIFICATION

• contact real people who will be potential users of system
  • identify specific end users: prototypical categories & extremes
• spend time with them discussing how the system might fit in
  • who would be willing to talk to you about this?
  • if you can’t get them interested, who will actually buy/use your system?
• learn about the user’s tasks (and potentially model w/task analysis)

**task examples:** articulate concrete, detailed examples of tasks they perform or want to perform that your system should support

• routine
• infrequent but important
• infrequent and incidental
IDENTIFICATION

if there are no real users or tasks...

• think again, there probably are!

Jeff Hawkins, the inventor of the Palm Pilot (first popular handheld personal organizer), was said to have carried a small block of wood around in his shirt pocket … As various everyday situations arose, he would take out the block of wood and imagine how he would use the device.¹

…the same technique can be used to evoke a response from expected end-users.

if all else fails...

• describe your expected set of users, and expected set of tasks
• later, you will need to verify or modify your assumptions.

¹see Sato and Salvador, Interactions 6(5)
GROCERY LIST TASK EXAMPLE

Lisa is doing the weekly menu planning for her family of 4. She chooses a set of recipes that suit the season, available prep time, current individual dietary eccentricities and her own preference at that moment.

Many of this week’s choices are regulars. She creates a shopping list of ingredients, ordered by where they can be found in the grocery store. Her husband Nick, who does the actual shopping and is more familiar with the store, supplies “feedback” on any errors she makes.

When a recipe requires an ingredient that was already needed for an earlier day’s meal, it is incremented. After getting through the week’s meals, she adds a few regular items like milk, bread, cereal and juice. After Nick has left with the list, she realizes she’s forgotten to check the pantry for staples like flour and rice.
KEY POINT TO NOTE

Task examples are interface independent, or as independent as possible.

What does that mean?
PEOPLE MAKE MANY KINDS OF GROCERY LISTS

lists say a lot about the task and the people who make the lists.
AND LISTS ARE JUST ONE OF THE WAYS THAT PEOPLE VARY IN HOW THEY DO THEIR MEAL PLANNING.

- plan ahead, or last minute?
- alone, or as a family?
- does list maker also shop?
- improvisation allowed?
REQUIREMENTS & METRICS

DESCRIBE THE PROBLEM WE ARE SOLVING
→ WHAT IS REQUIRED TO FIX IT
→ ARRIVE SYSTEMATICALLY AT A SOLUTION APPROACH

• what: functional vs. usability requirements
• when and how
• three steps to requirements
  1. ways to describe tasks and users
  2. identify focus users and metrics
  3. set level of support
WHAT ARE REQUIREMENTS?

functional requirements: what the interface must do

• usefulness -- scope, features…

non-functional requirements: constraints that development must live in:

• delivery time, maximum cost, delivery platform, supportability, sustainability…

• usability / user experience: what it should be like to use (a primary focus of HCI design)

ALL: clear, specific, defined in a MEASURABLE way
“USABILITY” FOCUS IS RARELY PURE

meeting usability requirements will/should often influence functional requirements. e.g.

- needed *speed* of response from system to user
  → implementation platform

- desired user context – e.g. on-the-go
  → must work on mobile platform

- use in distracted environment – e.g. hospital tool
  → voice output, speech input
  → visual display requirements to support memory chunking
FUNCTIONAL VS. USABILITY REQUIREMENTS:

The following list of requirements includes functional, usability and user experience requirements that could be defined for a movie theatre website. Which requirement is the best example of a functional requirement?

a) users must be able to buy electronic tickets in less than 3 distinct steps.

b) users must feel that the buying of electronic tickets is easy.

c) users must be able to buy electronic tickets using the website.

d) users must be able to learn how to buy electronic tickets on their first attempt.

functional: usefulness not usability specifies a feature or capability
CAN REQUIREMENTS CHANGE?

• requirements should be stable
  if based on good data
• but not rigid
  they may shift over time, in particular as design reality dictates
  what is possible / feasible given other constraints.
REQUIREMENTS
ESTABLISHED PRIMARILY
DURING PRE- AND
EARLY DESIGN

Data gathering
quantitative, qualitative

Task model

describe
• user
• task
• problem

other knowledge
of system, task, users

Needs: design
independent requirements

early design:
alternative 1
alternative 2
alternative 3
alternative 4

chosen design
approach: specifications

try out concepts
on user/task
descriptions

try out concepts
on user/task
descriptions
**Understand USERS:**
- who they are
- their key tasks

**Understand DESIGN:**
- design space and risks
- choose design approach

**REFINE Design:**
- by element
- considering task
- varied contexts

**CONFIRM & debug:**
- performance in real use

**Examine existing:**
- user tasks & objectives
- contexts
- interfaces

**Make use of:**
- requirements
- task examples, analysis
- real & virtualized users
- technology options
- company IP

**Evaluate w/:**
- observation
- interview/quest
- participatory interaction
- task walk-throughs

**Evaluate w/:**
- usability testing – controlled, uncontrolled
- heuristic evaluation

**Make use of:**
- graphical design
- interface guidelines
- style guides
- real & virtualized users

**Evaluate w/:**
- observation – many kinds
- ethnography
- interviews, questionnaires
- task examples
- task analysis

**PRODUCTS**
- user and task descriptions
- design requirements

**MATERIALS / METHODS**
- low fidelity prototyping methods

**GOALS**
- MEDIAL DESIGN
- LATE DESIGN
- ALPHA/ BETA SYSTEMS OR COMPLETE SPECIFICATION

**Big Picture – WHEN do requirements happen?**

K MACLEAN - DERIVED FROM VERSION BY SAUL GREENBERG (U CALGARY)
THREE STEPS TO REQUIREMENTS

1. **identify (and model) the human activity**
   which the proposed interactive system will support:
   task, goals, conditions; current problems and strengths

2. **identify all the users & other stakeholders**
   who perform or will perform the activity:
   groups, capabilities, motives, needs

3. **set focus and levels of support**
   which the system will provide (the system’s usability):
   constraints on the product’s performance, to support specific user-stakeholders you have targeted.
IDENTIFY (AND MODEL) THE HUMAN ACTIVITY

what outputs might you have at the end of this?

• goals
• task descriptions, task examples
• task models; normal steps and process; common breakdowns
• lists of problems
• things that work well
  • together describe our situation of concern
THE SITUATION OF CONCERN IS ...

altogether the context for the design problem

- something is wrong that we want to change, or
- something could be improved upon

some course of action is required

- that will result in a change that resolves the situation

⇒ this course of action (i.e. the solution) is what needs to be specified in requirements.

Situation of concern becomes clarified through data collection (Empathize) and other means (e.g., literature review) into a concrete Problem Definition.
EXAMPLE: SCHEDULING MEETINGS

situation of concern:

• hard to learn everyone’s schedule & find a common free time
• participants respond slowly or incompletely to request
• complicated to respond in adequate detail
• individual schedules change → time no longer available
• shared calendars: privacy and system incompatibility

result: too much iteration; non-convergent

course of action:

• ideas?

Online scheduler is one obvious one which has taken hold; others?
ACTIVITY – PART I  [10 MIN]

activity goal: practice breaking down and analyzing a human activity to start to generate requirements.

human activity: scheduling meetings

Task: schedule a meeting between project team members.

1. what steps are involved in this task?
   • try stating goals for the steps.
   • remember design/interface independence!
     these goals should apply for …
     Doodle poll, email coordination, in meeting, etc.

2. how can these steps go wrong?

Create a diagram of this task to help answer these questions
SCHEDULING
EXAMPLE, MY LIST...

some possible task goals:
• identify who needs to be @ meeting
• find common empty spaces in calendars
• identify a subset of empty spaces to suggest
• choose one » tell everyone
• receive confirmation that everyone still avail
• if no, iterate
• identify location

NEXT, break one of these down (many possible ways)
• find common empty spaces in calendars:
  1. ask all to communicate avail during a block; OR suggest times, get responses
  2. examine, manually or automatically
  3. find common openings, if any
  4. if no, iterate with different time blocks or suggestions

How might this go wrong?
• what if people respond very slowly?
• what if people respond incompletely?
• what if there are no solutions?
A FLOW DIAGRAM OF MEETING TASK

Flow Diagram
(from meeting creator perspective)

Advantages:
- Captures flow of information between different people, some artifacts
- Loops and other structure is implied
- A different view, which might reveal problems not encountered in other diagrams

Limitations:
- Only from one perspective (task flow might look different from invitee perspective)
- Highlights certain problems, but might hide others (diagramming other aspects may be valuable)
- Breakdowns point to places where more support for iteration on times/locations may be necessary (sequence diagram might help)

Invitee
- Responds to invites

List of possible times
(email, phone, paper, software)

List of available times
(email, phone, paper, software)

May have no available times

Meeting creation
(email, phone, paper, software)

Meeting reminder
(email, phone, paper, software)

Meeting room reservation
request for specific time
(email, phone, in-person)

Meeting room confirmation
(email, phone, in-person)

Meeting room confirmation
(email, phone, paper, software)

Meeting confirmation
(email, phone, paper, software)

May not be available

May not be available to attend

Space manager
- Manages meeting room availability

Check availability

Meeting calendar
DEPENDENCIES AMONG TASKS; TASK OBJECTS

“task objects” are resources required by tasks

- artifacts (files, lists, databases)
- people (special expertise, authority, or knowledge)
- other processes, equipment or tasks

Most tasks focus on a single resource

- task cannot be accomplished without the resource
- once resource is available the task can be completed

(Higher-level) processes have multiple dependencies

- the focus shifts as different tasks are performed
- activity is suspended when resources are not available
SIGNS OF TASK DEPENDENCIES?

• joint use of task objects by different tasks
e.g. access to shared files or databases

• communication between people
  may be direct (phone call) or indirect (memo)

• synchronization
  with real-world physical and mechanical processes

• suspension
  blocking when resources (information, people, real-world processes) are not available
ACTIVITY – PART 2
TASK DEPENDENCIES [5 MIN]

1. What “task objects” (i.e. resources, parts of process, even things generated by the process) might be required to meet this task?

2. Are there dependencies on these objects that could lead to conflicts or breakdowns?

Use your diagram to identify, or draw an entirely new diagram if required!
SCHEDULING EXAMPLE, MY LIST

*OBJECTS for Scheduling task?*
conflicts in their use **suggest dependencies**…
• calendars
• communication mechanisms (email, phone, cooler)
• “leader” – meeting leader, secretary, program

*Other signs of task dependencies?*
• can’t find time until have heard from all participants
• participants can’t give feedback on times until told their choices
THREE STEPS TO REQUIREMENTS

1. identify (and model) the human activity
   which the proposed interactive system will support:
   task, goals, conditions; current problems and strengths

2. identify all the users & other stakeholders
   who do or will perform the activity:
   groups, capabilities, motives, needs

3. set focus and levels of support
   which the system will provide (the system’s usability):
   constraints on the product’s performance, to support specific user-
   stakeholders you have targeted.
THE USER

need to understand general human needs:

• physical and cognitive abilities
• social and cultural environments
• use models of human behavior to test ideas

need to understand specific human needs:

• individuals have different skills and requirements
• they have responsibilities and authority in organizations
• they are expected to have a certain level of training
• they have specific access to tools and resources

Do not assume the user is “like you”, or “normal”
ACTIVITY – PART 3
USER NEEDS [5 MIN]

1. Identify potential users in this situation (are they all the same? different?)

2. Brainstorm a list of general human needs or needs specific to these users that could apply to this task?

For developing Personas – What are the relevant variables? What can their range of values be?
OBJECTS for Scheduling task?
conflicts in their use suggest dependencies…
• calendars
• Comm. Mech. (email, phone, cooler)
• “leader” – mtg leader, secretary, program

Other signs of task dependencies?
• can’t find time until heard from all participants
• participants can’t supply time until told when to look

The USER
examples of general needs:
• social / cultural environments (are people more comfortable with email, telephone or just running into each other?)
• are some users more overwhelmed with information than others, more than they can humanly process?

examples of specific needs:
• do all have laptops? are some reliant on mobile devices?
• is there variation in how responsive they are?
• do they have control over their time – i.e. are they permitted to decide what meetings they should / should not go to?
THREE STEPS TO REQUIREMENTS

1. identify (and model) the human activity which the proposed interactive system will support: task, goals, conditions; current problems and strengths

2. identify all the users & other stakeholders who do or will perform the activity: groups, capabilities, motives, needs

3. set focus and levels of support which the system will provide (the system’s usability): constraints on the product’s performance, to support specific user-stakeholders you have targeted.
FOCUS: WHICH USERS WILL YOU SUPPORT?

usually the intersection of

• greatest need
  and thus opportunity for business

plus

• feasibility
  we can identify a way to help

For example: you might choose NOT to support student users who do not have mobile devices. Why?

How does this relate do the different types of Personas?
METRICS: HOW WE KNOW IF WE HAVE SUCCEEDED

**quantitative metrics:** measured (countable) indicators of people’s use of the interface:

- speed of performance
- incidence of errors
- ease of learning the system
- user satisfaction
- …

**qualitative metrics:** descriptive accounts that shed light on quantitative measures:

- really important
- e.g. *stories* about user impressions and frustrations, and their *change* pre- and post design; critical incidents; product “buzz” …
CHOOSING USABILITY TARGETS

choice of usability metrics affects the solution:

• prioritize most important facets based on design goals:
  * e.g. is speed most important, or is it very bad to make errors?
• ease of learning can be important, especially for novices

levels of performance need to be quantified:

• must know baseline performance first (pre-redesign)
• then establish realistic target levels
• make sure we can measure the changes → iterate
PUTTING IT ALL TOGETHER
STATING REQUIREMENTS

no single right way to write requirements; lots of companies have specific methods, tools, etc.

one approach – list out and then prioritize each of:

1. **supported activities** (tasks and steps)
   tasks and processes involved that support the activity.

2. **user(s)**
   who does the task and what are their characteristics?

3. **level of support**
   what usability properties are important?
putting it all together
example – meeting scheduling

1. supported task
   • locating a jointly available meeting time

2. users
   • people with tight schedules who need to participate in meetings of 3 or more participants
   • people with “frequent” online access

3. level of support
   • users can provide all requested information with 1 minute of their time
   • require no iteration
   • respect privacy (e.g. posting shared calendars)
WHERE DOES TECHNOLOGY COME IN?

Tools support tasks

- new tools should **improve performance** of a task
- tools are often **specific** to the tasks they support
- tools must be acceptable / desirable to users

Systems support processes

- systems have to support **links** between tasks
- often tasks are **automated** using technology
- tasks have to be **supported** in a consistent manner
- desirable to **reduce dependencies**
- desirable to **reduce task complexity**

one way (of many) to use your task description…help you find solutions!
DISCUSSION ON REQUIREMENT READINGS [10 MIN]

Get into group of 3-4 answering the following questions:

• What surprised you? or
• What you disagreed with?
• Others?
ON DECK...

Next class (Tuesday) …

• Readings (as posted) and researcher journal

• Third project milestone: Ideate
  • due on Tuesday Oct 24th
THE FORM OF THE SOLUTION (FINALLY, THE DESIGN!!):

But, design starts by adding constraints

- cost (time, money, expertise)
- compatibility with specific hardware or software
- market pressures (standards, “look and feel”)
- many of these don’t come from the designers!

multiple levels in the description (and prototypes)

- the social/cultural/physical environment
- the user interface
- the application software
- the operating system
- system resources (storage, networking, peripherals)
OTHER FACTORS IN CHOOSING A SOLUTION

existing intellectual property

• technology owned or licensed by the organization
• unique skills or knowledge in the organization
• market share or reputation

innovation

• technology becomes obsolete quickly
• R&D requires time and effort
• often incremental improvements are good enough
• significant changes may be required sometimes