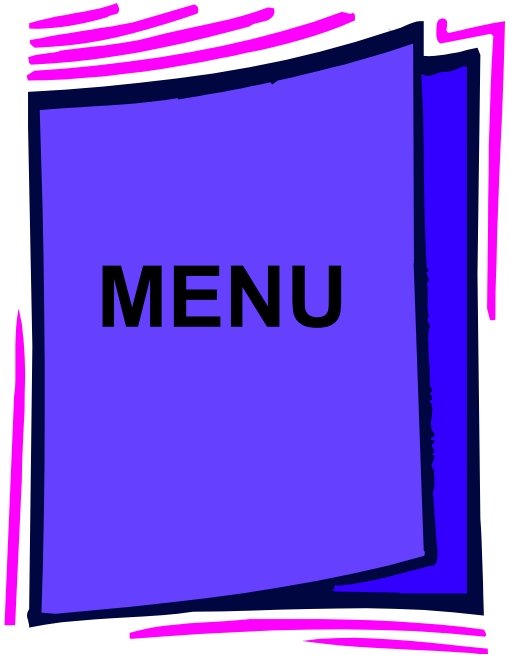
CS3724 Human-computer Interaction

Systems Analysis

Copyright © 2004 H. Rex Hartson and Deborah Hix.

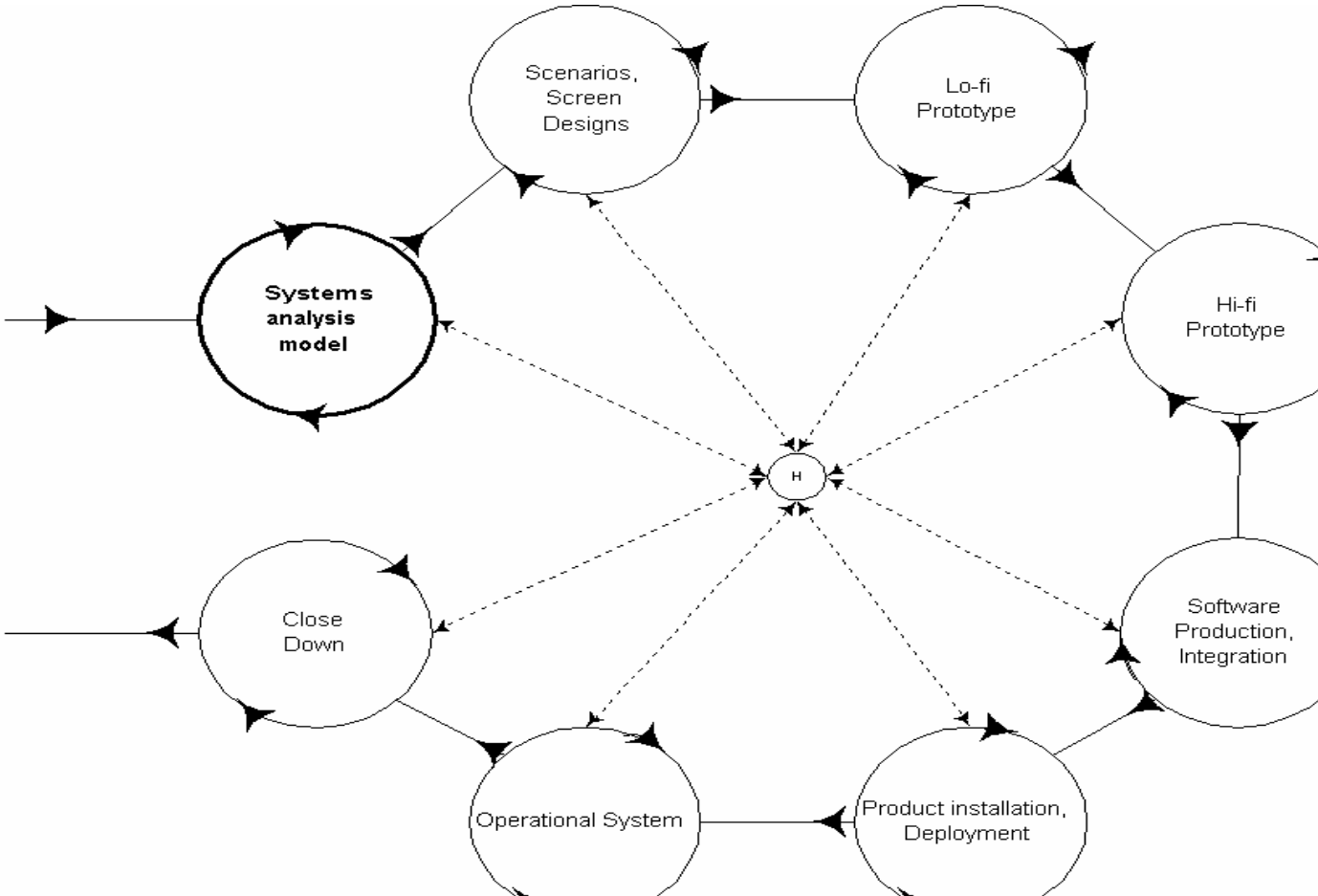
Topics

- Ethnographic field studies
- Product concept statement
- Business process model
- Needs analysis
- User class definitions
- Task analysis
- Usability goals
- Constraints



Introduction to Systems Analysis

- Revisiting the usability engineering life cycle

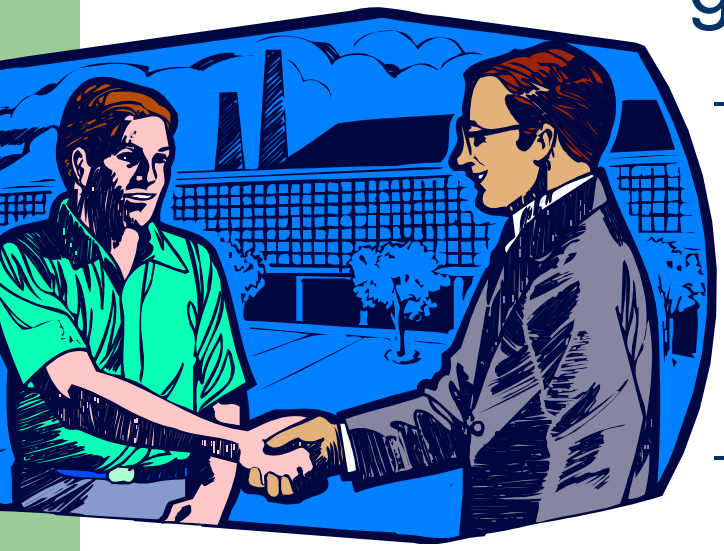


Ethnographic Field Visit

- In anthropology and sociology, ethnography is:
  - Participating, “overtly or covertly, in people’s daily lives for an extended period of time, watching what happens, listening to what is said, asking questions”

    [Hammersley & Atkinson 1983, as quoted by Shneiderman, p. 107]

Ethnographic Field Visit



- For user interaction requirements gathering:
  - UI designers limit study to days or even hours, but have to obtain needed data
    - “Quick and dirty ethnography”
  - Cannot obtain needs information by just brainstorming in your own office
  - Cannot substitute market research for ethnographic studies

Ethnographic Requirements Gathering

- Process for UI ethnography includes:
  - Preparation for field study
    - Start with “brainstorming” of user task statements
    - Understand organization’s policies & culture
    - Check out their website
    - Know current system & history
    - Prepare script of questions for interview
    - Select appropriate users to observe and/or interview
    - Obtain permission to observe and/or interview

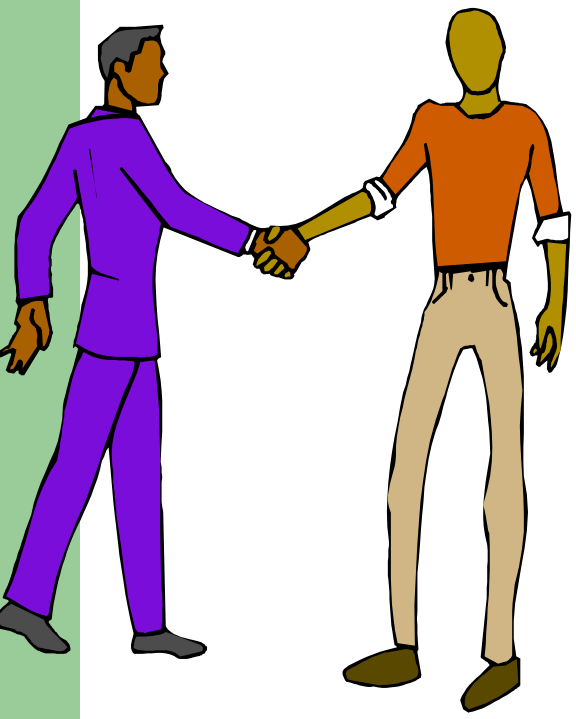
Ethnographic Requirements Gathering

- Perform field study
  - Establish rapport with managers and clients
  - Observe and/or interview users in workplace
  - Collect quantitative and qualitative data
  - Collect artifacts (e.g. paper forms) as available
  - Follow leads from visits, if any
  - Document and characterize user classes
  - Document work flow/user task analysis
- Keep focus of activities user centered!

Ethnographic Requirements Gathering

- Cannot just ask “What objects do you interact with?”
- Explain to users why you are there
- Have to “tease out” needed information
- Have initial questions scripted in advance
- Be ready to modify, explore, be ready to modify, explore, branch ou

Ethnographic Requirements Gathering



- Seems easy, but it's not
  - Hidden traps, surprises (e.g. what to wea different perceptions of managers vs. users, different use of language/technica terms)

Ethnographic Requirements Gathering

- Equally important as data collected: rapport/relationships with client, users established during process
- What if client is reluctant to give access to users?
  - Ask for a couple of hours
  - Establish necessity for usability

Ethnographic Requirements Gathering

- **Caution**: Difficult for users to tell developers what they want or need
  - Do not expect users to do design!
  - Important to observe users in their typical work environment

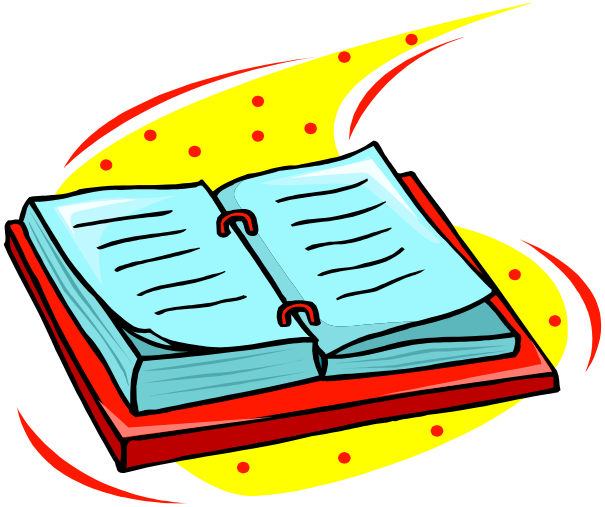
Developer: I try to tell users what they need, but they don't want to listen to me

Ethnographic Requirements Gathering – In Sum

- The “User Interface Requirements Detective”
  - Goal of ethnography for UI designer is to discover, extract, and collect the “clues” needed to ensure usability of design

Introduction To Example System

- Calendar System
  - Simple automated version of a paper calendar
  - Goal is to learn the development process, not to produce a marketable calendar product
  - Working assumptions: some boundaries (e.g., hardware) set by management, customers, marketing, etc., there is a need for this product



Example: System Analysis

- Goal:
  - To make a fast tour through the process of determining basic user and system requirements
- Activities:
  - A sampling of product concept statement, needs analysis, business process model, user class definition, task analysis, usability goals, and constraints

Product Concept Statement

- Product concept statement-brief descriptive summary of product, typically 50-75 words
- Mission statement for a product, to help focus product development
- Writing a good product concept statement is not easy and is not done once, highly iterative

Product Concept Statement

- Answer the following questions:
  - What is the product name?
  - Who are the product users?
  - What will the product do?
  - What problem(s) will the product solve?

Product Concept Statement

- A possible product concept statement for Calendar:
  - Our calendar will have automated support for scheduling appointments, to improve customer satisfaction.
    - Too vague

Product Concept Statement

- A better product concept statement (47 words):
  - The Calendar will allow a broad variety of users to schedule and manage appointments. These users can range from professionals using the system to run an office to casual users keeping track of personal information. Automated support will reduce scheduling effort and increase awareness of appointments.

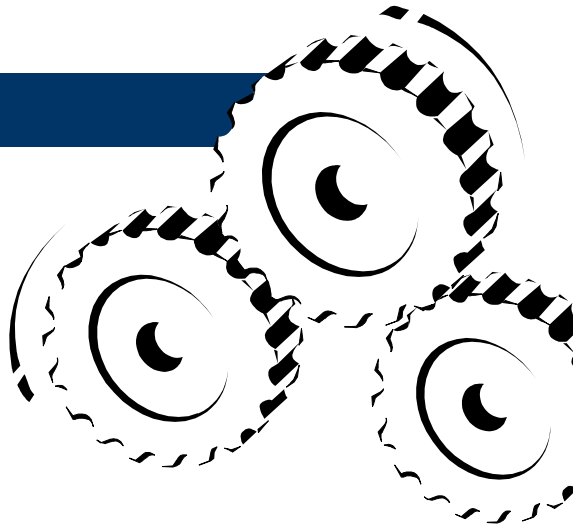
Product Concept Statement

- Example of being more specific
  - ‘Automated support will reduce scheduling effort and increase awareness of appointments.’
    - Reduce scheduling by supporting recurring appointments.
    - Increase awareness by giving alarm (visual and/or audible)

Team Exercise – Product Concept Statement

- Overall goal: On-going exercise in developing the interaction design for a specific Web application: ***A public ticket buying kiosk***
- Exercise goal: Write a concise product concept statement for your ticket kiosk system

Team Exercise – Product Concept Statement



- Activities:
  - Assemble in project teams
  - Talk about your approach to the kiosk
  - Write a product concept statement
  - Iterate and polish it
- Deliverables
  - Your “final” product concept statement, hand-written on plastic overhead
  - Schedule: Due yesterday!

Example: Needs Analysis

- Goal of system: *manage appointments*
- Assumption: Some boundaries set by management, marketing, customer, etc. (e.g. hardware); determination made that product is novel, market not yet saturated
- Features
  - Appointment means information on:
    - Date
    - Time
    - Place

Example: Needs Analysis

- Appointment description
  - *Manage* means
    - Add new appointment
    - Delete existing appointment
    - Modify existing appointment
  - Plus, need ability to view/display appointments

(Task=user, function=system)

- Follows from talking with client, users; not just developers' ideas

Example: Needs Analysis

- After observing users, someone thinks of "alarm" idea (the needs don't come all at once, up front)
  - Do we want to actively inform of appointments (maybe ask or observe users)
  - Decision: Yes, very useful; a way to beat paper
  - Iterate and revise needs
- New feature: Active reminder (increased functionality)

Business Process Modeling

- Understand application domain
- Important for non-UI software, too
- Goal is to capture
  - What gets done to run business
  - Who does what and how it gets done
  - How it relates to other things that get done

Business Process Modeling

- How to capture it
  - Look for both computer-supported and non-computer tasks
  - Gather and study work artifacts (e.g., paper work, tickets, slips)
  - Describe work flow, task flow, data & document flow
  - Flow charts are good (e.g., tasks are flow lines to/from people (users) and data objects

Team Exercise – Business Process Model

- Goal: a one-page diagram illustrating high-leve business process (obviously an over-simplification) for your ticket kiosk operation
- Activities:
  - Sketch out a diagram showing business roles, information flow, information repositories, transactions, etc.
- Deliverable: one sheet plastic overhead
- Schedule: Now!

User Class Definition

- User classes are about roles, not individuals
  - "Know thy user"—and it is *not* you!
    - Important to have representative user(s) on development team and/or have access to representative user(s)
  - Most of system analysis (e.g., task analysis, usability goals, usability specifications) is done for each user class

User Class Definition

- User knowledge of application/work domain
- User knowledge of computers
- Training and application-related experiences
- Frequency of use
- User goals
- Job- or task-related factors (e.g., job description, location, level of responsibility)

User Expertise Level

- Expertise levels don’t necessarily define user classes, but can occur within user classes
  - **Novice or first-time user**: may know application domain but not specifics of application
  - **Intermittent user**: uses several systems from time to time; knows application domain but not details of different applications

User Expertise Level

- **Experienced user**: “power” user, probably uses application daily and knows both application and task domain very well

- Design may have to account for each of these expertise levels
- Remember: experienced users for some systems are novices for others

These are not the specific user class types you should identify for your project!

Example: User Class Definition

- What are characteristics of users of Calendar system?
  - General Characteristics
    - Busy people
    - Keep schedule for self and others
    - Professional and personal use
    - Calendar is very small part of job
    - Need ‘transparent’ tool
    - High general skill level, literate

Example: User Class Definition

- Domain skills
  - Know how to use calendar
- Computer skills
  - Broad range
  - At least some typing skills
  - Familiar with GUI/mouse

Example: User Class Definition

- Conclusion
  - Keep it simple
  - Usability as important as functionality (or more)
  - Try to get functionality greater than paper calendar
  - Minimize typing
  - Users must learn quickly
- User class characterization matrix
  - First, decide on most appropriate set of parameters for your domain and context

Example: User Class Characterization Matrix

	User class A	User class B
Computer knowledge		
Domain knowledge		
Complexity of domain content		
Discretionary or captive?		

Example: User Class Characterization Matrix

	User class A	User class B
Receptive/ resistant?		
Usage frequency, duty cycle		
With whom do they interact (outside system)		

Example: User Class Characterization Matrix

	User class A	User class B
What information is exchanged (outside system)?		
Training?		
Culture of work context?		

Culture of Work Context

- Culture of work context is the overall flavor, philosophy, ambience, and environment of user's work
  - It's about thought processes and mind set, policies, terminology
  - Example: steel mill floor is about noise, dust, hot temperatures, safety concerns, making iron and steel
  - Doctor's office is very different culture

Team Exercise – User Class Definition

- Goal: Define characteristics of your major user classes
- Activities: fill out form on plastic overhead
- Schedule: Do it now, in class

Task Analysis

- What developers **observe** that users need, *not* what developers **think** that user need

- Task analysis gives inventory of user needs; feeds scenarios in design
- Task analysis is probably the most overlooked and shortchanged activity in the whole user interaction development process

Task Analysis

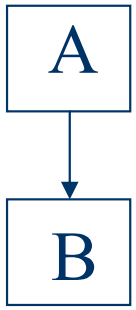
- Hierarchical task analysis (HTA)
  - Structured, organization, and relationships of tasks users perform with system
  - Not timing, precedence, order of task performance, work flow, etc.
  - Only what users *can* do, not *must* do

Task Analysis

- Hierarchical task decomposition (key ideas)
  - Task names: <action object>
    - Examples: add appointments, configure parameters
  - User-centered wording, not system centered
    - Example: view appointment, not display appointment

Task Analysis

- Hierarchical task decomposition (key ideas)
  - Hierarchical relationships

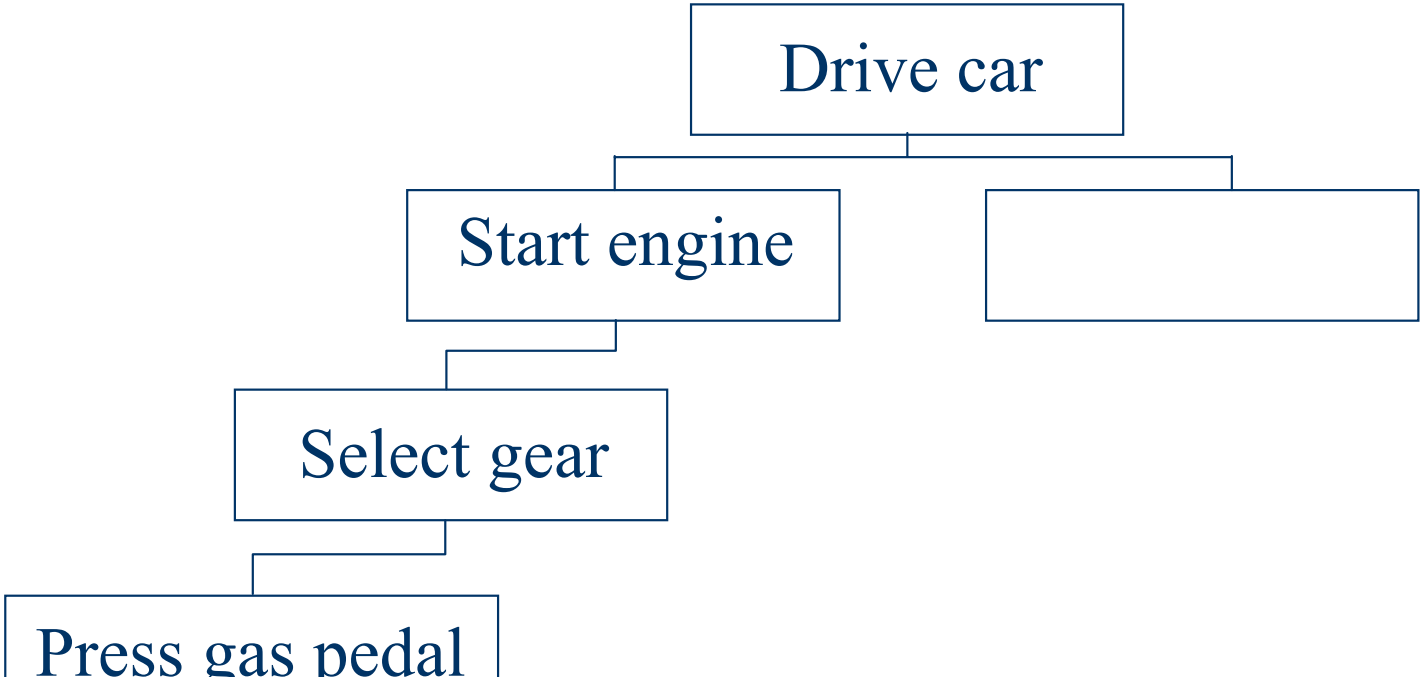


means A is a super-task of B, B is a sub-task of A

  - Semantics: Doing B is part of doing A (a "litmus" test for this characteristic
  - Example: Task a is filing out form; task B is filling out name field

Task Analysis

- Hierarchy does *not* show sequencing
  - **Incorrect** attempt at hierarchical relationship:



Example: Task Analysis

- What tasks will users perform with this system?
  - For highest-level tasks, start with goal of system: ***Manage appointments***
  - *Appointment* means information on: Date, time, place, appointment description
  - *Manage* means
    - Add new appointment
    - Change/delete existing appointment
    - Plus ability to view appointments
  - This all follows from user interviews/observations, not just developer's ideas

Example: Task Analysis

- What tasks will users perform with this system?
  - Tasks are performed by user (e.g., view)
  - Initial list of major sub-tasks
    - Add new appointment
    - View existing appointments
    - Modify existing appointments
    - Delete existing appointments
    - Set alarm
    - View calendar

Example: Task Analysis

- Task analysis iterated
  - As thinking about viewing appointments, realize the need for different levels or scopes of view
    - For example, by month, by week, by day, by hour
    - Implication: add “control view” task to list

Example: Task Analysis

- Also discovered need to *search* appointment database to retrieve by content
  - Implication: add to needs, tasks, functions, requirements
  - Note: from here on, “requirements” means interaction design requirements

  (but cannot separate entirely from system, functional requirements)

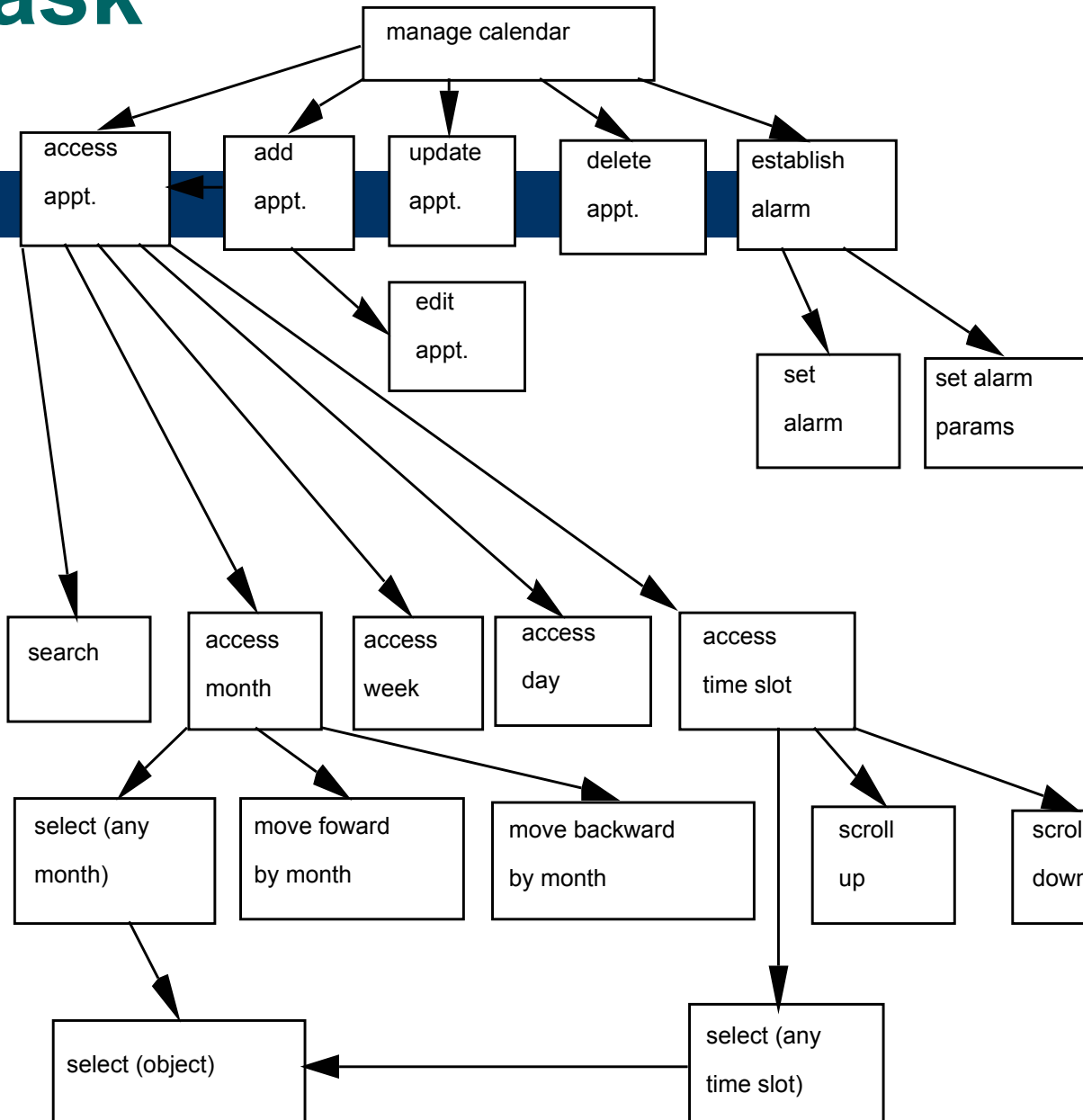
Example: Task Analysis

- Another example of iteration:
  - Alarm feature will lead to user tasks set parameters)
  - Decision: for now, hard wire for 10 minutes before appointment; no use tasks

Good example of early simplistic design decision; needs iteration

Example: Task Analysis

- Example of possible quasi-hierarchical user task structure for Calendar
  - Structure diagram is accompanied by brief description of what each box means



Usability Goals

- Usability evaluation design driven by usability goals
- Usability goals driven by business objectives
- Determine usability goals in terms of
  - User classes
  - User task content, special tasks
  - Walk-up-and-use learnability
  - High performance for expert users
  - User errors
  - User satisfaction

Usability Goals

- Example usability goals for Calendar
  - Fast walk-up-and-use for simple tasks
  - High learnability for more advanced tasks
  - Low error rate for rescheduling appointments
  - Increased effectiveness of calendar by helping users avoid missed appointments

Usability Goals

- High-level objectives in terms of usability and design of user interaction
  - Reflect real use of product in real world
  - Determine what is important to organization and to users
  - Example: Learnability for new users, power performance for experts, avoiding errors
  - Usability goals can be market driven

Team Exercise – Task Analysis, Usability Goals

- Goal: Simple task analysis and usability goal statement for your kiosk system
- Activities:
  - Sketch a simple HTA diagram
  - Write down a few usability goals for your kiosk
- Deliverable: HTA diagram, usability goals statement, each on a sheet of plastic overhead
- Schedule: Now

Constraints

- Cost and budget
- Schedule and development time
  - What restrictions do budget and schedule impose on product scope?
- Size and/or weight
  - Will product be on portable or mobile equipment?
- Integration with existing or other developing systems
- Security or privacy issues

Constraints

- Example constraints for Calendar System:
  - Prod will be used in wide variety of environments, from factory floors to open offices to homes
  - Product will run on wide variety of platforms, but mostly PCs, laptops with no special devices
  - Budget is highly limited
  - Schedule is one semester!