MMI 2: Mobile Human-Computer Interaction

Design Process and Prototyping

Prof. Dr. Michael Rohs

michael.rohs@ifi.lmu.de

Mobile Interaction Lab, LMU München

Lectures

#	Date	Topic
1	19.10.2011	Introduction to Mobile Interaction, Mobile Device Platforms
2	26.10.2011	History of Mobile Interaction, Mobile Device Platforms
3	2.11.2011	Mobile Input and Output Technologies
4	9.11.2011	Mobile Input and Output Technologies, Mobile Device Platforms
5	16.11.2011	Mobile Communication
6	23.11.2011	Location and Context
7	30.11.2011	Mobile Interaction Design Process and Prototyping
8	7.12.2011	Evaluation of Mobile Applications
9	14.12.2011	Visualization and Interaction Techniques for Small Displays
10	21.12.2011	Mobile Devices and Interactive Surfaces
11	11.1.2012	Camera-Based Mobile Interaction 1
12	18.1.2012	Camera-Based Mobile Interaction 2
13	25.1.2012	Sensor-Based Mobile Interaction 1
14	1.2.2012	Sensor-Based Mobile Interaction 2
15	8.2.2012	Exam

Review

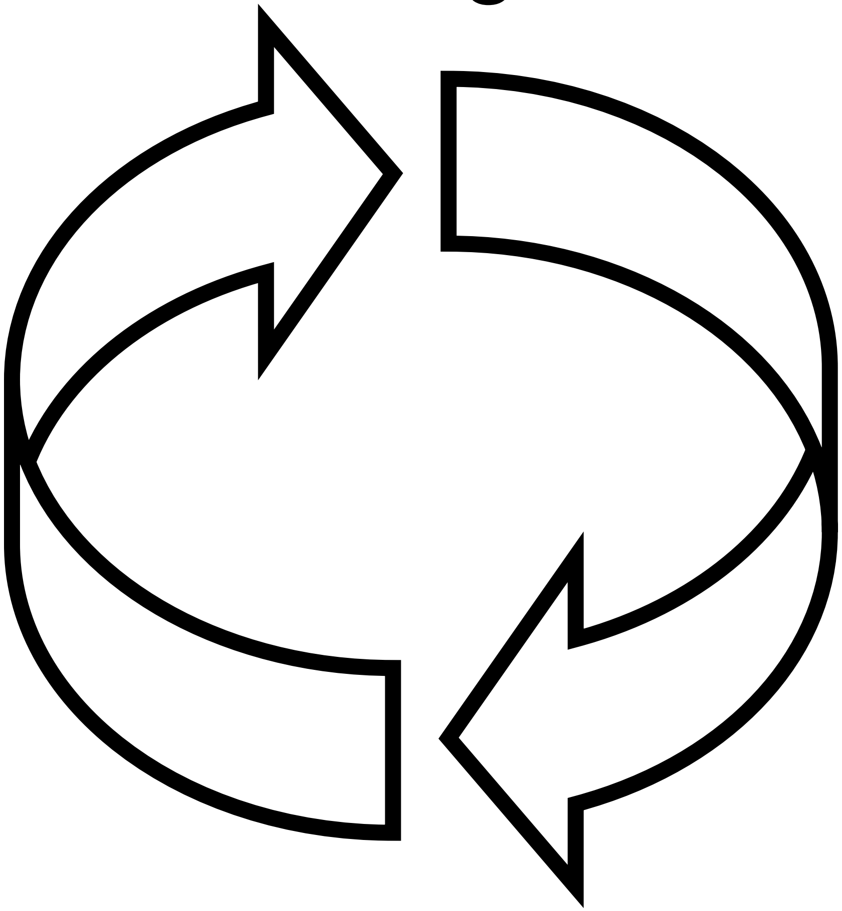
- Name wireless communication technologies
- What technology for transmitting sensor data?
- What technology requirements for voice calls?
- Design goals of Bluetooth?
- What is a Piconet? Device roles?
- How are connections established in Bluetooth?
- How is power saved in Bluetooth?
- What is SDP?
- What are examples of cell-based systems?

Preview

- Design Process
- Users
- Requirements
- Initial Design Techniques
- Scenarios
- Prototyping

DESIGN PROCESS

Iterative Design: DIA Cycle



Design

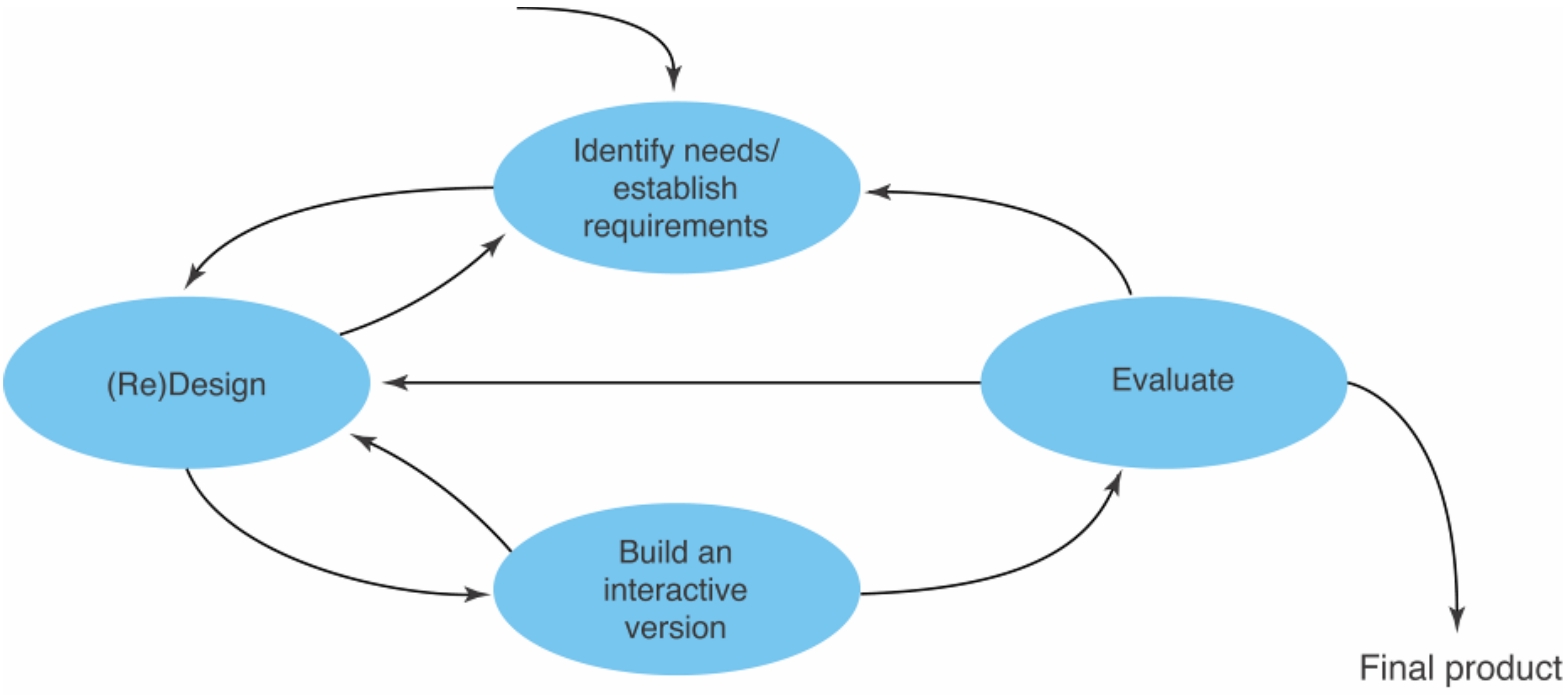
Analyze

Implement

Basic Activities of Interaction Design
(Preece et al.)

- Identify needs and establishing requirements
  - Target users
  - Needs and wants of target users
- Develop alternative designs
  - Suggesting ideas for meeting the requirements
  - Conceptual design and physical design
- Build interactive versions of the designs
  - Paper-based prototypes
  - Role-playing users
- Evaluate designs
  - Determining usability and acceptability of the product
  - User involvement throughout the process

Basic Activities of Interaction Design
(Preece et al.)



Identify needs/ establish requirements
(Re)Design
Evaluate
Build an interactive version
Final product

Source: Preece, Rogers, Sharp: Interaction Design

The Usability Design Process (Gould et al.)

- Steps in designing usable systems:
  - Define the **problem** the user wants solved
  - Identify **tasks** users must perform
  - Learn **user** capabilities
  - Set specific **usability targets**
  - Sketch out user **scenarios**
  - Design, build, and test **prototypes**
  - **Repeat** until targets are met or deadline is reached
  - **Deploy** system
  - Measure acceptance

Focus on Users

- Decide **who** the users will be
- Decide **what** they will be doing with the system

- "You can't figure out what people want, need, can do, and will do without talking to them."
- Find real people interested in your planned system (otherwise there's a problem)
- Methods
  - Talk with users
  - Visit user locations, observe (and videotape) users working
  - Have users think aloud, try it yourself
  - Use surveys and questionnaires

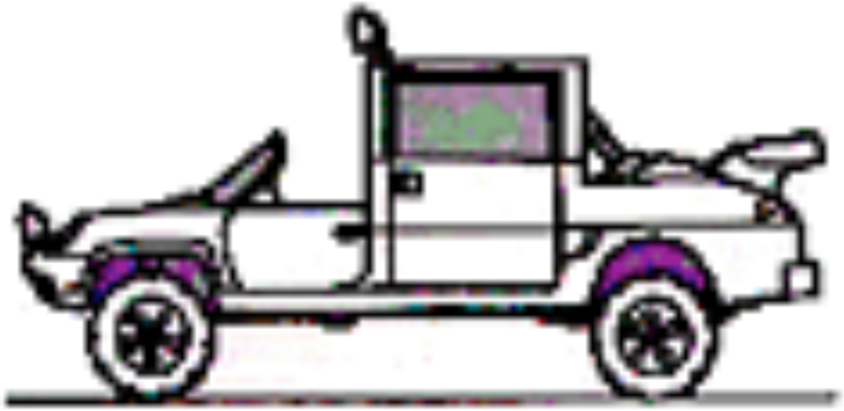
User Profiles or “Personas” (Cooper, 1998)

- Short profiles of typical users
  - Prototypical user for a specific user group
  - Fictitious individual with concrete characteristics
- Building personas
  - Often built from interview results
  - Synthesize fictitious users from real user characteristics
  - Develop multiple personas for different types of users
- Bring them to life
  - With a name, characteristics, experience, personal background, environment they are located in, goals, tasks, skill levels, etc.
- Base design decisions on the needs of the personas

Personas Example

(Cooper, About Face, Chapter 5)

Building a car that pleases everyone



Building a car based on three personas (representing larger groups)

Marge, *mother of three*

Marge wants safety and room for many passengers. A minivan meets her needs.

Jim, *construction worker*

Jim wants cargo space and the ability to carry heavy load. A pickup truck meets his needs.

Alesandro, *software engineer*

Alesandro wants sporty looks and speed. A two-door sports car meets his needs.

Personas Help to…

- Communicate user characteristics to developers
  - Consistent, coherent understanding of target group
  - Features prioritized by how well they match personas' needs
  - Evoke empathy for the target users represented by personas
- Avoid elastic users
  - Constantly changing target user to justify differing design choices
  - Clear focus towards the intended users
- Avoid self referential design
  - Developers / designers designing for themselves
- Avoid designing for "edge cases"
  - Should not be central to design (prioritize!)

Example Persona: Bob

Bob is 52 years old and works as a mechanic with an organisation offering road service to customers when their car breaks down. He has worked in the job for the past 12 years and knows it well. Many of the younger mechanics ask Bob for advice when they meet up in the depot as he always knows the answer to tricky mechanical problems. Bob likes sharing his knowledge with the younger guys, as it makes him feel a valued part of the team.

Bob works rolling day and night shifts and spends his shifts attending breakdowns and lockouts (when customers lock their keys in the car). About 20% of the jobs he attends are complex and he occasionally needs to refer to his standard issue manuals. Bob tries to avoid using the manuals in front of customers as he thinks it gives the impression he doesn't know what he's doing.

Bob has seen many changes over the years with the company and has tried his best to move with the times. However he found it a bit daunting when a new computer was installed in his van several years ago, and now he has heard rumors that the computer is going to be updated to one with a bigger screen that's meant to be faster and better.

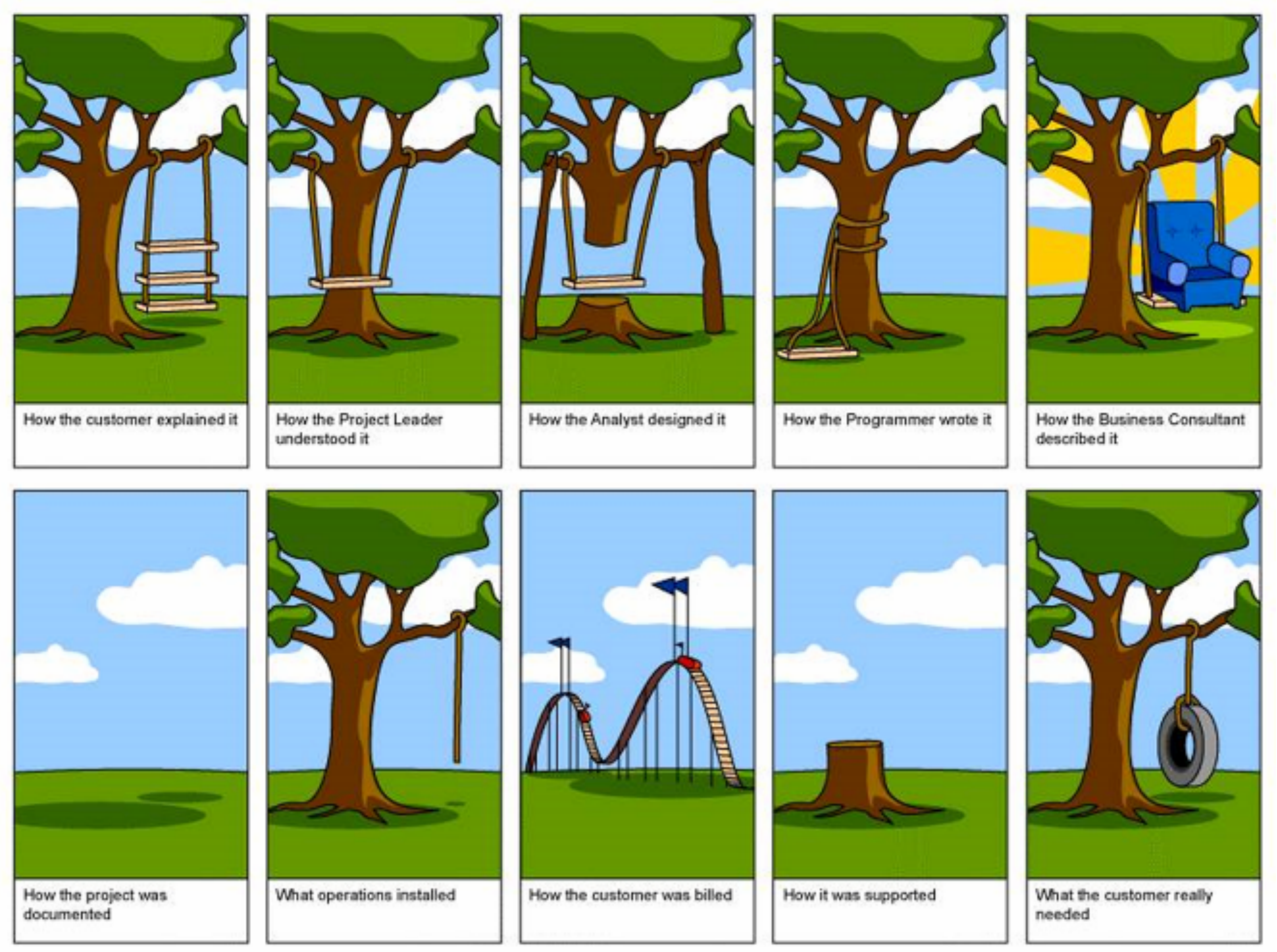
Bob's been told that he will be able to access the intranet on the new computer. He has heard about the intranet and saw once in an early version on his manager's computer. He wonders if he will be able to find out want's going on in the company more easily, especially as customers' seem to know more about the latest company news than he does when he turns up at a job. This can be embarrassing and has been a source of frustration for Bob throughout his time with the company.

Bob wonders if he will be able to cope with the new computer system. He doesn't mind asking his grandchildren for help when he wants to send an email to his brother overseas, but asking the guys at work for help is another story.

Source: http://www.steptwo.com.au/papers/kmc_personas/

Getting the Requirements Right

Major cause of project failure: unclear requirements


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Source: Preece et al.: Interaction Design

Gathering Data

- Researching similar products
  - State-of-the-Art
  - Sets level of user expectation
- Interviews
  - Good for exploring issues and getting new perspectives
  - Props, e.g. sample scenarios, paper prototypes
- Focus groups
  - Group interviews
  - Multiple viewpoints, highlighting areas of conflict
- Questionnaires
  - Large number of particiapnts
- Diaries, cultural probes, experience sampling

Task-Centered Design (Lewis and Rieman)

- Task-centered design
  - Real, complete, representative tasks
- Traditional requirements analysis
  - Abstract, partial task elements
- Choosing concrete representative tasks
  - Real tasks described by real users
  - Referenced in a few words, expanded if needed
- Tasks should cover main system functionality
  - Don’t include features that don’t support any task
- Examples
  - For a mobile calendar: “Enter a meeting with a colleague.”
  - For a barcode reader: “Link to Web page from the printed ad.”

Task-Centered Design (Lewis and Rieman)

- Using tasks in design
  - Produce scenarios from sample tasks
  - Represent scenarios as storyboards and step through sample tasks
  - Tasks are design-independent, scenarios are design-specific
  - Settle design arguments by looking at task context
- Risk of leaving something out
  - Treat sample tasks as examples
  - Look beyond examples
  - Judge what will be common and what will be uncommon
  - System can do at least some real task and has a good chance of working with other tasks

Scenario-based Design

- User interaction scenario
  - Informal narrative description of user activity and experience when performing a task
  - What a user would have to do and sees in performing a task step-by-step using a given system
- Scenarios are design-specific
  - How would a task be performed in a particular design
  - Task is design-independent

Scenario Perspective

- Use-oriented and user-oriented perspective
  - Represent, analyze, plan impact of system on activities
  - Open up design process to intended users
  - Accessible to all stakeholders
- User's view of what happens, how it happens, and why
  - User motivations toward the system
  - User actions taken
  - User's reasons why actions were taken
  - Results in terms of user's motivations and expectations

Futuristic Scenario: Smartphone

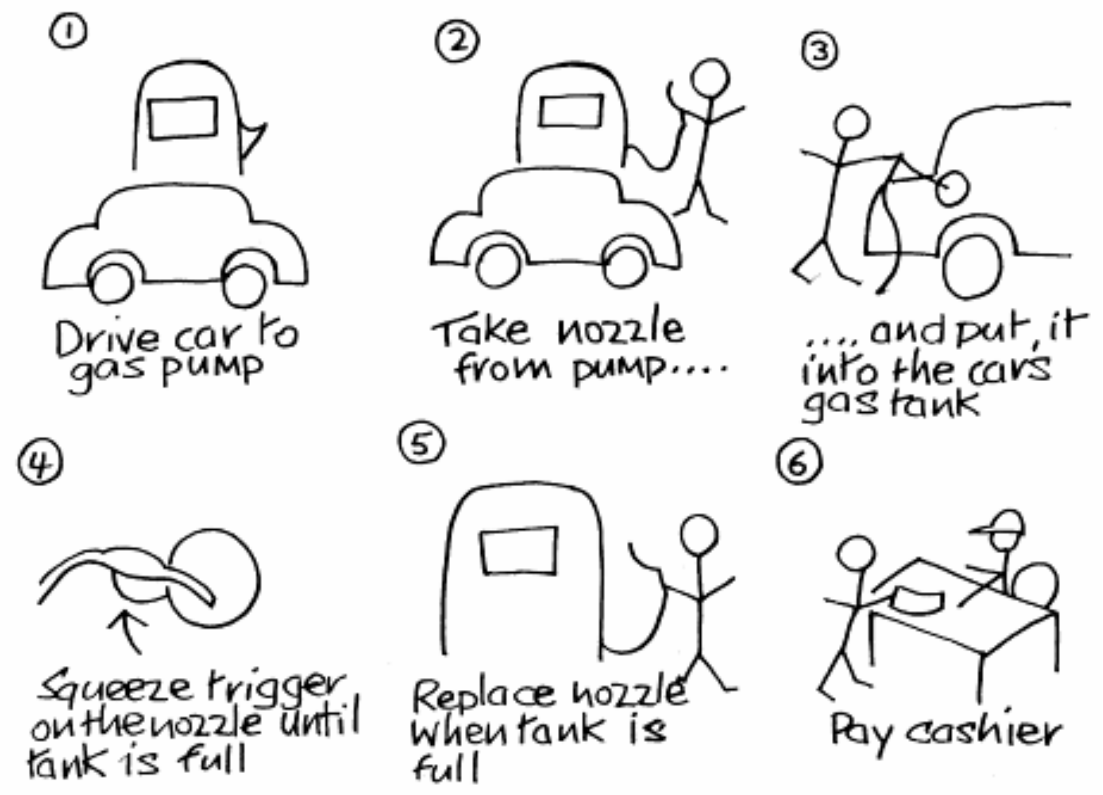
- “A businesswoman is traveling from San Francisco to Paris on a business trip. On her way to the airport she narrowly misses a traffic delay. She avoids the traffic jam because her Smartphone beeps, then sends her a text message warning her of the traffic accident on her normal route from her office to the airport.

  Upon arrival at the airport, the location-sensitive Smartphone notifies the airline that she will be checking in shortly, and an airline employee immediately finds her and takes her baggage. Her on-screen display shows that her flight is on time and provides a map to her gate.”

Preece, Rogers, Sharp: Interaction Design, 2002

Representations of Scenarios

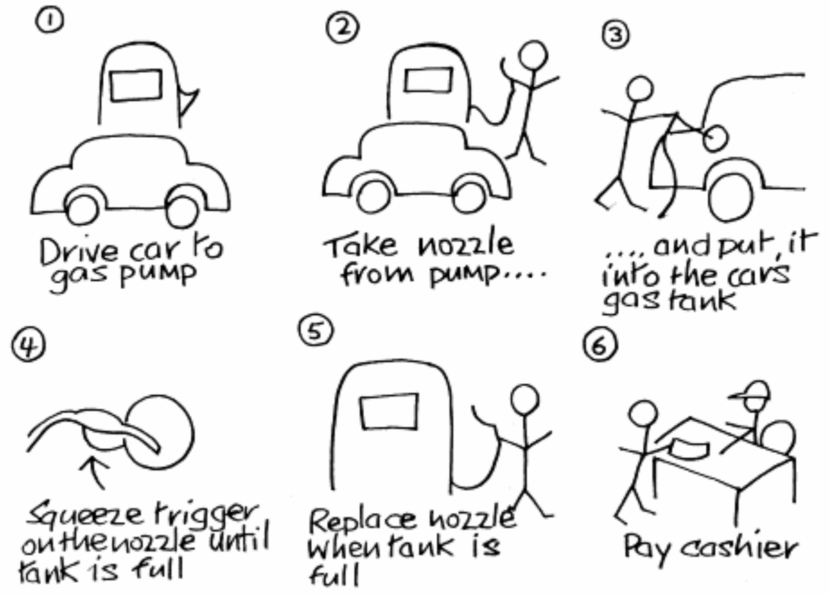
- Text
- Storyboards
- Video mock-ups
- Scripted prototypes
- Physical situations


1
Drive car to gas pump
2
Take nozzle from pump....
3
...., and put it into the car's gas tank
4
Squeeze trigger on the nozzle until tank is full
5
Replace nozzle when tank is full
6
Pay cashier

Example storyboard

- Different levels of detail possible
- Expanding scenarios if needed

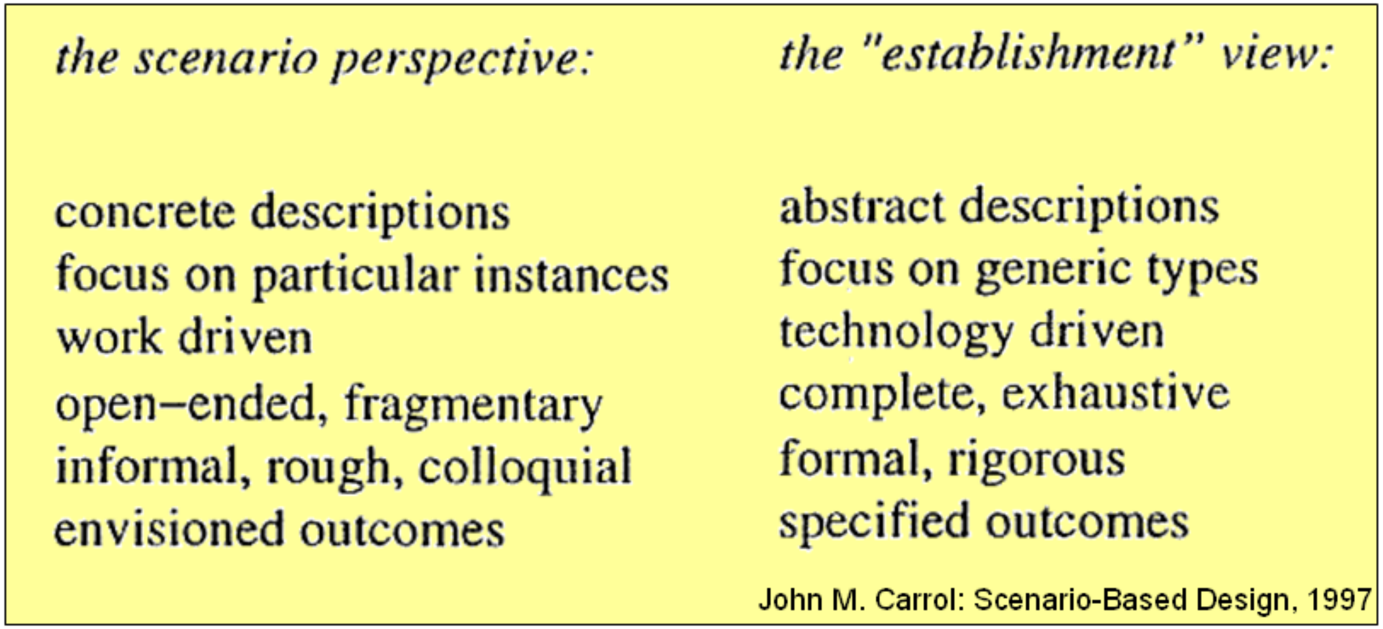
Initial Design Techniques: Storyboarding


1
Drive car to gas pump
2
Take nozzle from pump....
3
...., and put it into the cars gas tank
4
Squeeze trigger on the nozzle until tank is full
5
Replace nozzle when tank is full
6
Pay cashier

- What?
  - Sequence of single images
  - Like visual outline of a movie
  - Illustrates interaction
- Why?
  - Describes task showing environment, user, and computer
  - Or describes UI as series of screen images
  - Helps working out interaction details
  - Great at-a-glance overview of interaction
  - Helps developing usage scenarios
- When?
  - After describing a task, storyboard it, then take back to user

Scenario Benefits

- Explore design space with low degree of commitment
  - Concrete and detailed enough to guide design
  - Infer, reason about, discuss design implications
  - Keeps future system use in view
  - Describes particular instances



the scenario perspective:	*the "establishment" view:*
concrete descriptions	abstract descriptions
focus on particular instances	focus on generic types
work driven	technology driven
open–ended, fragmentary	complete, exhaustive
informal, rough, colloquial	formal, rigorous
envisioned outcomes	specified outcomes

John M. Carrol: Scenario-Based Design, 1997

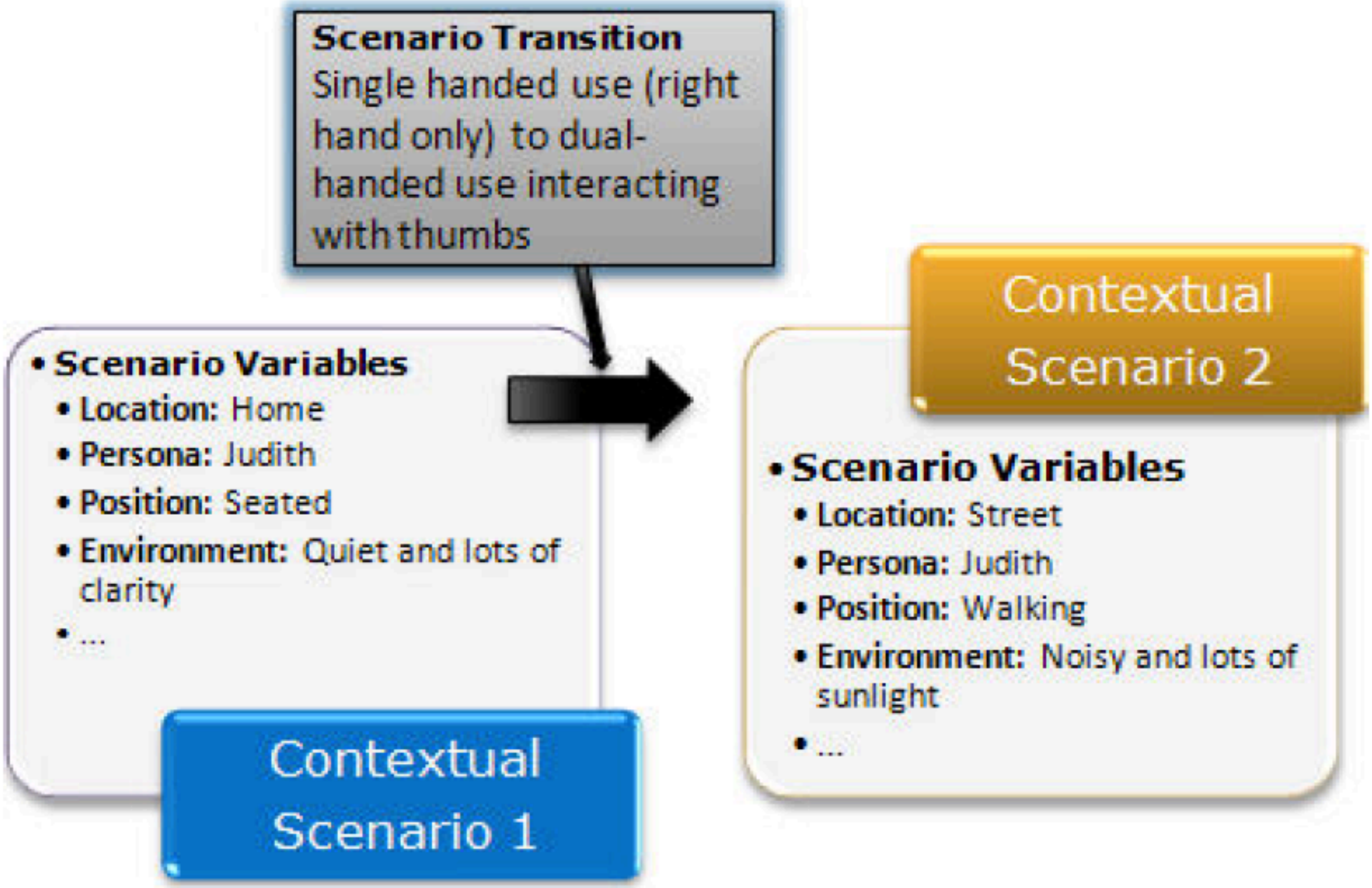
Scenarios in the Development Lifecycle

- Envisionment
  - Scenarios encourage creative thinking by leaving details out
- Requirements analysis
  - Scenario description of current state
- User-designer communication
  - Users create scenarios illustrating design questions
  - Joint creation of overview scenarios and scenarios for particular tasks
- Design rationale
  - Rationale explains design with respect to scenarios
  - Identify design tradeoffs by contrasting scenarios
    - E.g. query scenario vs. browsing scenario
- Documentation

Mobile Design Specifics

- Usage situation / external context very important
- Traditional methods have limited success in guiding mobile design
  - Ethnographic sessions, user observation
- Do not capture specific problems of mobile design
  - E.g., changing context, small displays
- Pragmatic and ethical/privacy issues
  - Not feasible / too intrusive if designer follows user all the time
- Possible methods, but do not capture usage context
  - Interviews, questionnaires, simulations
  - Paper prototyping
  - Wizard of Oz evaluation

Conceptual Framework for Generating Scenarios & Personas (de Sa, Carrico)



Scenario Transition
Single handed use (right hand only) to dual-handed use interacting with thumbs

- **Scenario Variables**
  - **Location:** Home
  - **Persona:** Judith
  - **Position:** Seated
  - **Environment:** Quiet and lots of clarity
  - ...

Contextual Scenario 1

Contextual Scenario 2

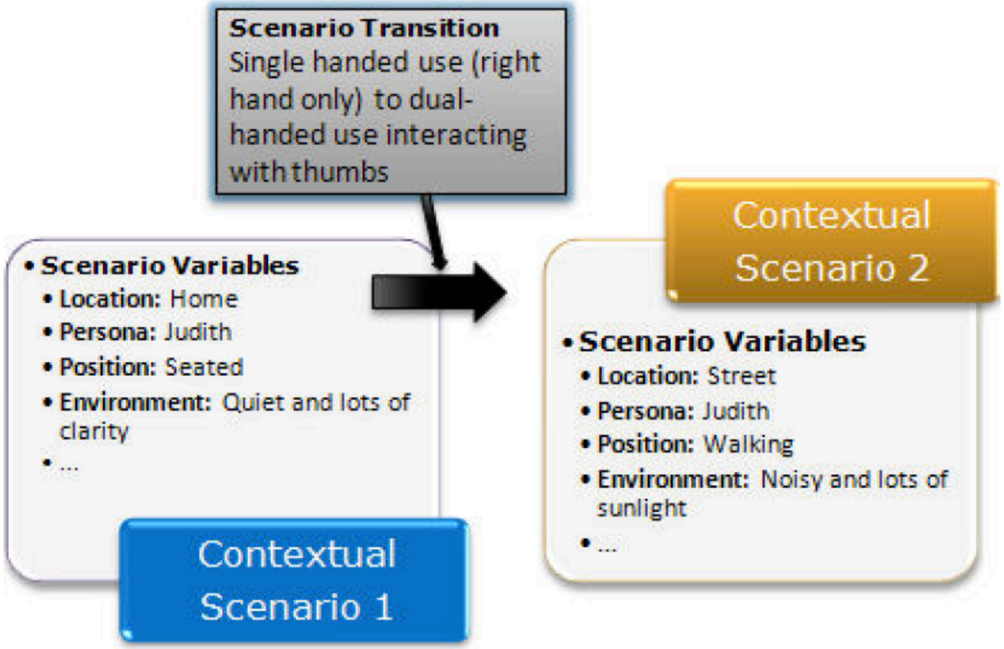
- **Scenario Variables**
  - **Location:** Street
  - **Persona:** Judith
  - **Position:** Walking
  - **Environment:** Noisy and lots of sunlight
  - ...

de Sa, Carrico: Lessons from Early Stages Design of Mobile Applications, Mobile HCI 2008

Conceptual Framework for Generating Scenarios & Personas (de Sa, Carrico)

- Highlights relevant factors for design task

- Contextual scenarios
  - Described by variables
- Scenario transitions
  - Context changes
- Scenario variables
  1. Locations and settings
  2. Movement and posture
  3. Workloads, distractions, and activities
  4. Devices and usages
  5. Users and personas



Scenario Transition
Single handed use (right hand only) to dual-handed use interacting with thumbs

- **Scenario Variables**
  - **Location:** Home
  - **Persona:** Judith
  - **Position:** Seated
  - **Environment:** Quiet and lots of clarity
  - ...

Contextual Scenario 1

Contextual Scenario 2

- **Scenario Variables**
  - **Location:** Street
  - **Persona:** Judith
  - **Position:** Walking
  - **Environment:** Noisy and lots of sunlight
  - ...

de Sa, Carrico: Lessons from Early Stages Design of Mobile Applications, Mobile HCI 2008

Conceptual Framework for Generating Scenarios & Personas (de Sa, Carrico)

- Contextual variable: locations and settings
  - Lighting (e.g., well lit, medium lit, dark; natural / artificial lights)
  - Noise (e.g., noisy, quiet)
  - Weather
  - Obstacles (users has to avoid them, interruptions)
  - Social environment (presence of others influences behavior)

- Contextual variable: movement and posture
  - Sitting
  - Standing
  - Walking

de Sa, Carrico: Lessons from Early Stages Design of Mobile Applications, Mobile HCI 2008

Conceptual Framework for Generating Scenarios & Personas (de Sa, Carrico)

- Contextual variable: devices and usages
  - Single-handed
  - Dual-handed
  - Stylus / finger / keyboard/ numeric keypad
- Contextual variable: workloads, distractions & activities
  - Criticality (degree of attention & concentration)
  - Cognitive distractions (e.g. phone ringing, resumption)
  - Physical distractions (real-world activities)
  - Activities (may introduce specific challenges)

de Sa, Carrico: Lessons from Early Stages Design of Mobile Applications, Mobile HCI 2008

Conceptual Framework for Generating Scenarios & Personas (de Sa, Carrico)

- Contextual variable: users and personas
  - Movement impaired
  - Visually impaired
  - Heterogeneity (e.g. differences in age, finger size, handedness, cultural background, profession)

- Exercise: Create scenario for mobile clothes shopping app, helps users to navigate from shop to shop
  - Contextual scenarios?
  - Scenarios transitions?
  - Scenario variables?

de Sa, Carrico: Lessons from Early Stages Design of Mobile Applications, Mobile HCI 2008

Applying the Generated Scenarios

- Generated scenarios help to understand how users are affected by social and physical context

- Put actual users in scenario contexts
- Questionnaires with contextual questions
  - gather contextual data without following users
- Experience sampling method (ESM)
  - diary of experiences, triggered at random times
- Probes
  - tools to document, reflect, express thoughts on environment and actions

de Sa, Carrico: Lessons from Early Stages Design of Mobile Applications, Mobile HCI 2008

Experience Sampling Method (ESM)

- Diary of experiences
- Users fill out a questionnaire in response to alert
  - Several times a day, long-term
  - Users are given paging devices, which alarm randomly
  - Users note down what they feel/think based on given questions
- Experiences are sampled “in situ” (in the actual situation)
  - Mood, time use, social interactions, etc.
  - No need for recall, no cognitive bias
  - Capture infrequent events
- Self-reporting
  - Researcher not present when user samples experience
  - Interview at beginning / end, but not

Exercise: How to alert and capture experience? What to ask?

Consolvo, Walker: Using the Experience Sampling Method to Evaluate Ubicomp Applications. IEEE Pervasive Computing, 2003.

Mobile Probes

- Diaries
  - Allow users to document feelings and activities over time
  - Design ethnography method: capture activities in context, understand needs and motivations, gather requirements
- Self-photography
  - Similar to diaries, but users take photos
  - The world through the eyes of the user
- Probes
  - Tools that allow people to document, reflect, express thoughts
  - Communication link between users and designers
- Mobile probes
  - Camera phone with questionnaire app to capture context
  - Example: Shopping context

Hulkko, Mattelmäki, Virtanen, Keinonen: Mobile Probes. NordiCHI 2004.

INTERVIEWS & QUESTIONNAIRES

Four Key Issues

1. Setting goals
   - Decide how to analyze data once collected
2. Relationship with participants
   - Clear and professional
   - Protect privacy
   - Informed consent form when appropriate
     - Signed agreement between evaluator and participant
3. Triangulation
   - Use more than one approach
   - Use different perspectives to understand a problem or situation
4. Iterate
   - If questions reveal that goal was not sufficiently refined: refine goal, repeat

Data Recording

- Notes, audio, video, photographs

- Notes plus photographs
- Audio plus photographs
- Video

Interviews

- Unstructured
  - Not directed by a script
  - Rich but not replicable
- Structured
  - Tightly scripted, often like a questionnaire
  - Replicable but may lack richness
- Semi-structured
  - Guided by a script but free to explore interesting issues in more depth
  - Good balance between richness and replicability

How to Ask Questions

- Clear and simple, not too broad
  - “How do you like the UI?” is too general!
- Affording logical, quantitative answers
  - Bad questions give unusable or wrong answers
  - Open vs. closed questions
- Users don’t always answer truthfully
  - Lack of knowledge, bad estimates, embarrassment
  - So formulate questions carefully, maybe indirectly
- No leading questions!
  - For initial input, do not focus on presenting your design ideas, but on learning about the task

Running the Interview

- Introduction
  - Introduce yourself, explain the goals of the interview, reassure about the ethical issues, ask to record, present any informed consent form
- Warm-up
  - Make first questions easy and non-threatening
- Main body
  - Present questions in a logical order
- A cool-off period
  - Include a few easy questions to defuse tension at the end
- Closure
  - Thank interviewee, signal the end, e.g., switch recorder off

Enriching the Interview Process

- Props: devices for prompting interviewee
  - Examples: prototype, scenario

Interview Results

- User profile
  - Target user of system
- Persona
- Task profiles
  - Core functionality of system
- Contextual factors

- If existing system involved
  - Problem areas with the system
  - Ideas for improvement

Questionnaires

- Can be administered to large populations
  - Paper, email, and the web used for dissemination
- Provide clear instructions on how to complete the questionnaire
- Decide on whether phrases will all be positive, all negative, or mixed
- Presentation consistency:
  - Yes/No always in same position
    - yes ( ) no ( )
  - All positives always at same side
    - Bad |--|--|--|--|--|--| Good
  - Avoid conflict with existing scoring systems
    - 1-6 or A-F for grades

Observation

- Skilled behavior is hard to explain
  - (→ procedural memory)
- Observation
  - Spending time with the users as they do tasks in natural setting
- Follow the user "like a shadow"
  - Make notes
  - Observe activity
  - Ask questions (but not too many)
- Level of involvement
  - No involvement (outside observation)
  - Full involvement (participant observation)
  - Or anything in between

Characteristics of Observation

- Captures aspects that other techniques do not
- Captures context of tasks
  - Work context can be an important factor of usability, e.g., interruptions
- Requires a lot of time and commitment
- Results in a lot of data that needs to be analyzed

Kinds of Observation

- Direct observation in the field
  - Structuring frameworks
  - Degree of participation (insider or outsider)
  - Ethnography
- Direct observation in controlled environments
- Indirect observation: tracking users' activities
  - Diaries
  - Interaction logging

Structuring Frameworks to Guide Observation

- The person. Who?
  The place. Where?
  The thing. What?

- The Goetz and LeCompte (1984) framework:
  Who is present?
  What is their role?
  What is happening?
  When does the activity occur?
  Where is it happening?
  Why is it happening?
  How is the activity organized?

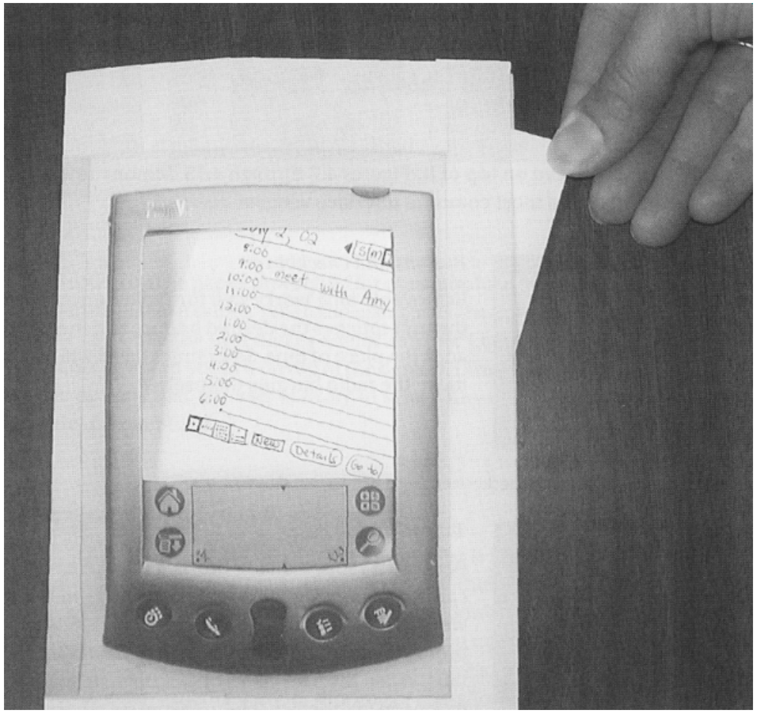
Ethnography

- Ethnography is a philosophy with a set of techniques that include participant observation and interviews
- Ethnographers immerse themselves in the culture that they study
- Degree of participation can vary
- Analyzing video and data logs can be time-consuming
- Collections of comments, incidents, and artifacts

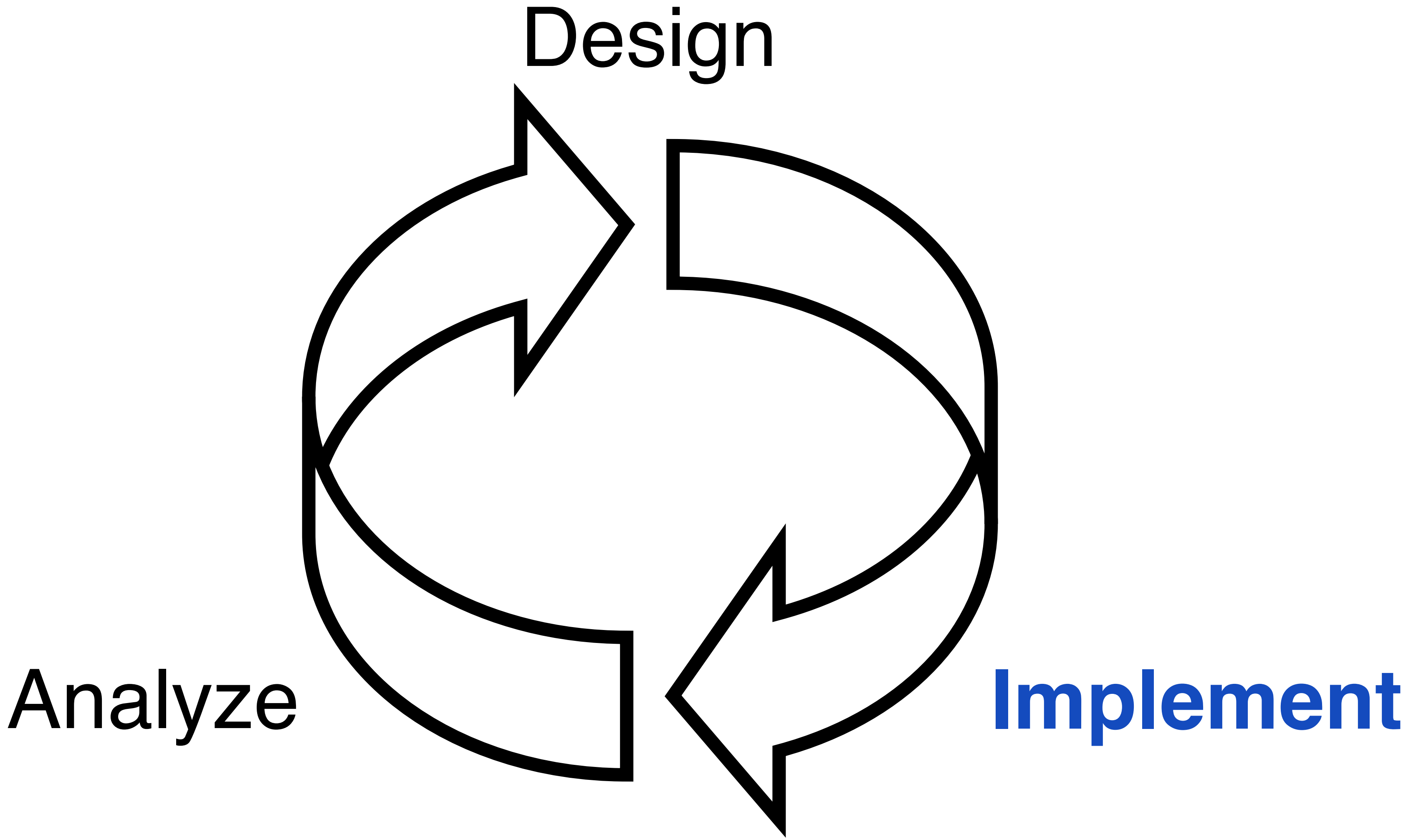
Data Interpretation and Analysis

- Analysis
  - As soon as possible
  - Initial interpretation to get the big picture
  - Deeper analysis on selected aspects
- Representation of analysis result
  - Textual description
  - State charts, work-flow charts, entity-relationship diagrams
  - “Design principle” for your application → projects

PROTOTYPING

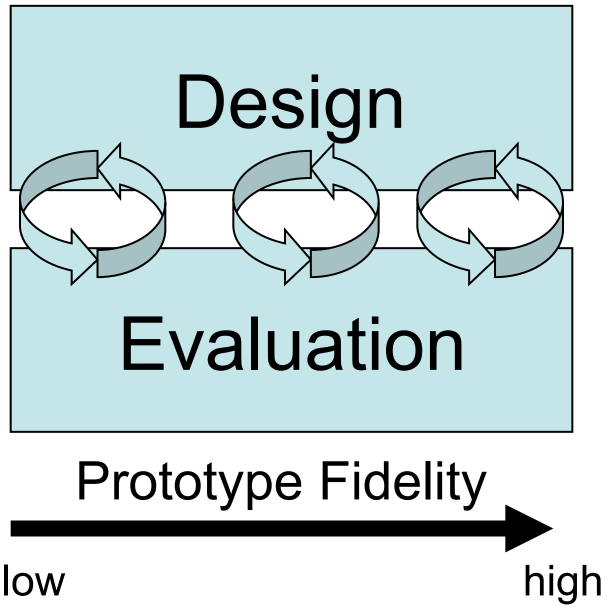


DIA Cycle: How to realize design ideas?



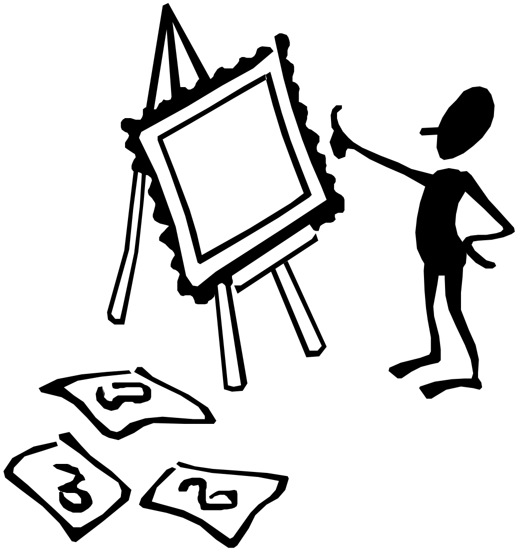
From Ideas to Implementation: Prototyping

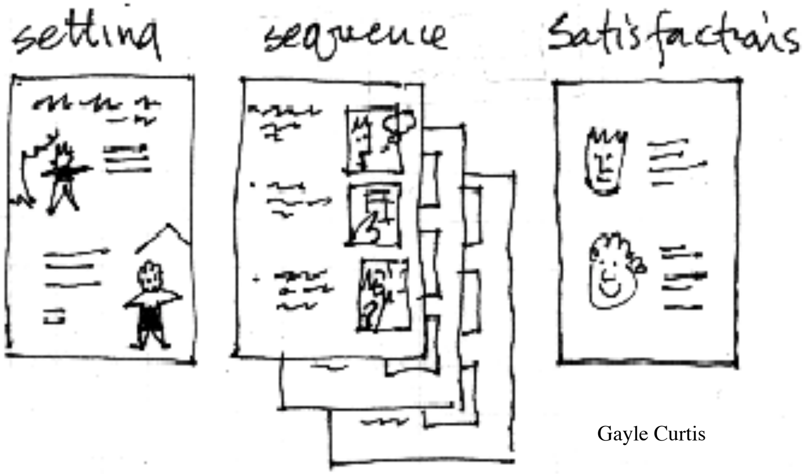
- Building a scaled-down version of an interactive system to collect information to guide its further design
  - Invaluable for iterative design
- Get early feedback on emerging designs
  - After initial requirements analysis, scenarios
- Continuous input for design decisions
  - During all design phases
- Prototype appropriate for
  - Audience
  - Design phase
  - Design question



Low-Fidelity Paper Prototypes

- First prototype, quick and cheap
- Paper and pencil mockup of user interface
  - Rough sketches of the main screens and dialogs
  - Textual description of interface functions and relationships between screens
- Goals
  - Brainstorming
  - Expert review of interaction flow
  - First user feedback
  - User tests





Gayle Curtis

Paper / Post-it Prototype Process

Collaboratively creating the prototypes

Reviewing the prototypes

Source: http://www.pocketpcmag.com/_archives/may03/e_prototyping.asp

Building a Low-Fidelity Prototype

- Assemble material
  - Paper: large heavy paper for designs, cards for note taking
  - Adhesives: tape, glue sticks, correction tape
  - Markers: colored pens and pencils, highlighters, fine liners
  - Post-it notes
  - Transparent sheets for user input

- Wooden device frame with slots for cards
  - More durable than paper
  - Can be put in pocket
  - Can serve as frame for taking notes ("in situ prototyping")

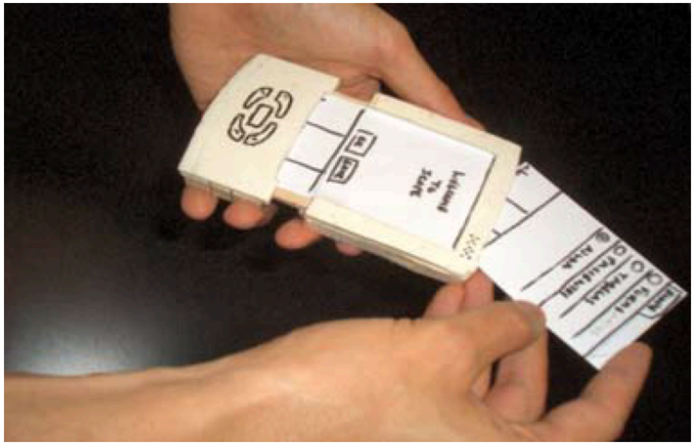
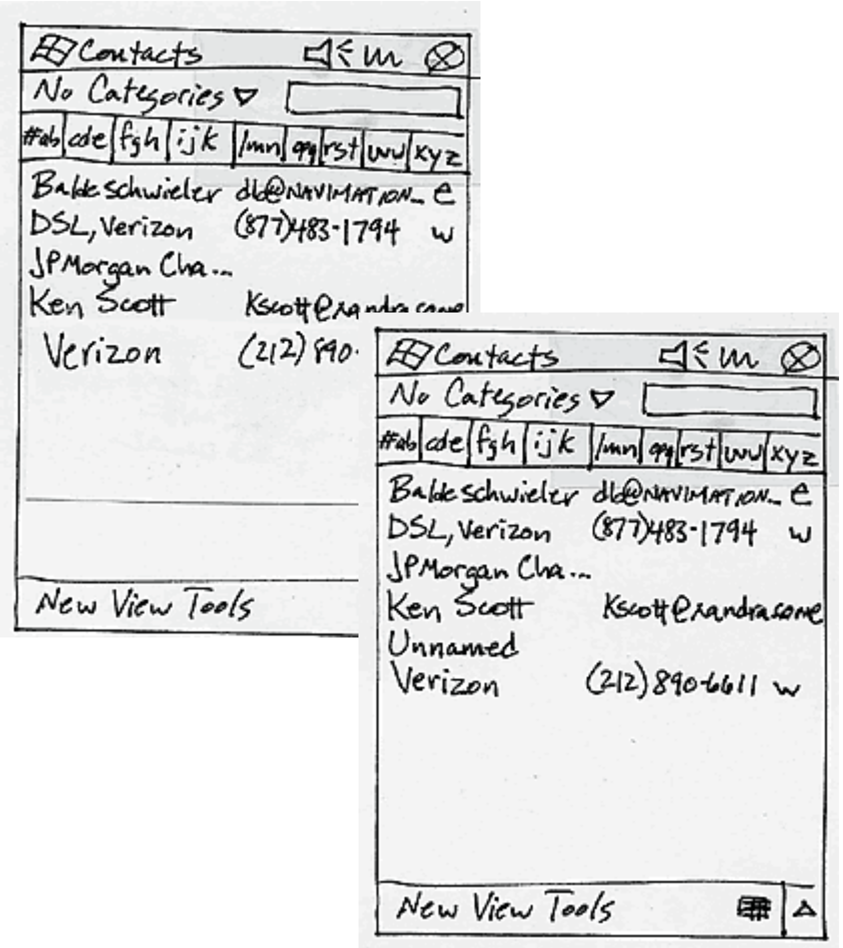
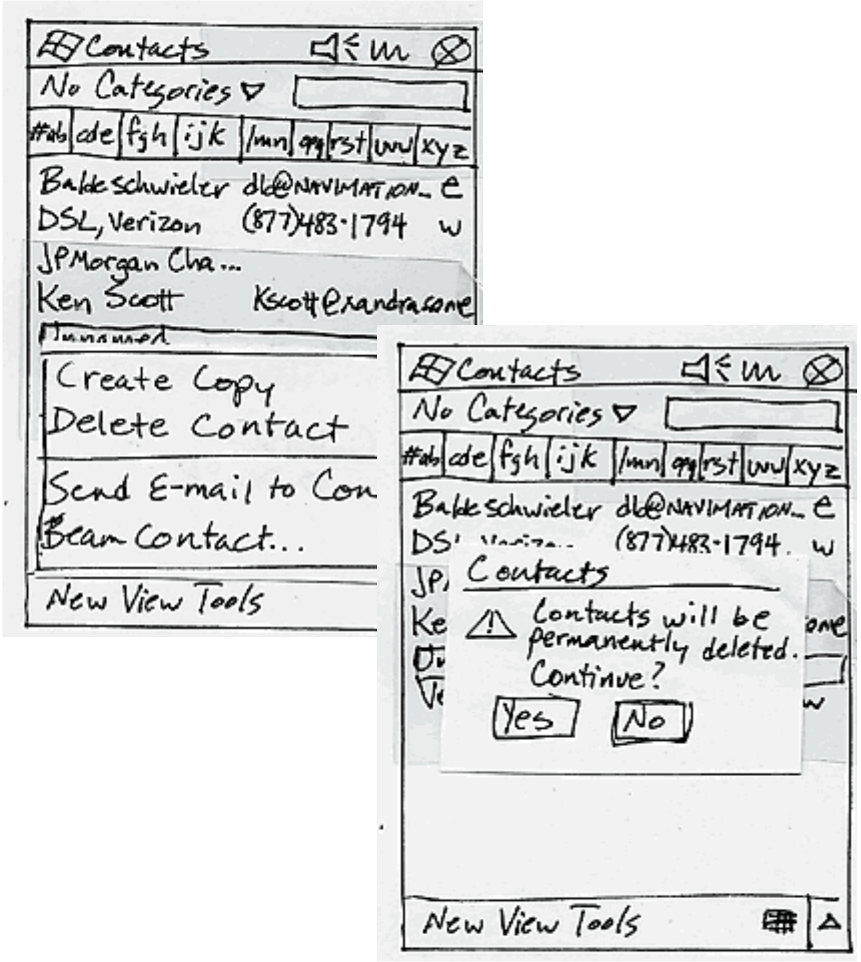


Image source: de Sa, Carrico: Lessons from Early Stages Design of Mobile Applications, Mobile HCI 2008

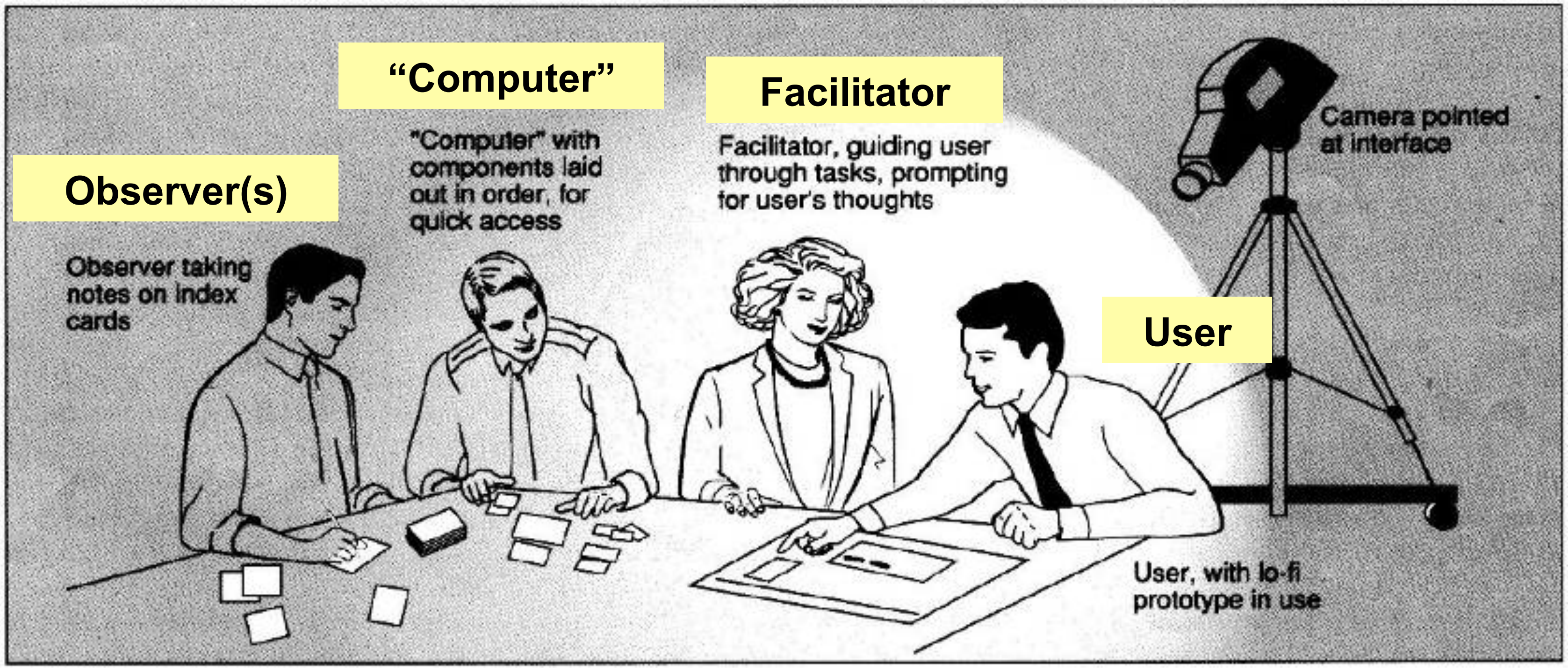
Paper Prototype Examples


Contacts
No Categories ▽
#ab cde fgh ijk lmn opq rst uvw xyz
Baldeschwieler dl@NAVIMATION e
DSL, Verizon (877)483-1794 w
JPMorgan Cha...
Ken Scott Kscott@xandra.com e
Unnamed
Verizon (212)890-6611 w
New View Tools


Contacts
No Categories ▽
#ab cde fgh ijk lmn opq rst uvw xyz
Baldeschwieler dl@NAVIMATION e
DSL, Verizon (877)483-1794 w
JPMorgan Cha...
Ken Scott Kscott@xandra.com e
Unnamed
Create Copy
Delete Contact
Send E-mail to Con
Beam Contact...
New View Tools
Contacts
Contacts will be permanently deleted. Continue?
Yes
No

Source: http://www.pocketpcmag.com/_archives/may03/e_prototyping.asp

Low-Fidelity User Testing



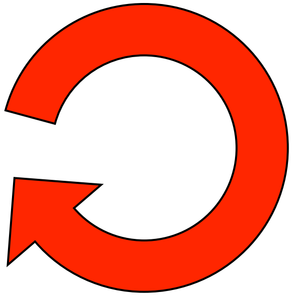
"Computer"
Facilitator
Observer(s)
User
"Computer" with components laid out in order, for quick access
Facilitator, guiding user through tasks, prompting for user's thoughts
Camera pointed at interface
Observer taking notes on index cards
User, with lo-fi prototype in use

Marc Rettig: Prototyping for Tiny Fingers

- Select users
- Prepare test scenarios, drawn from task analysis
  - familiar data, realistic tasks
- Practice
  - team members know their roles, no "computer" delays

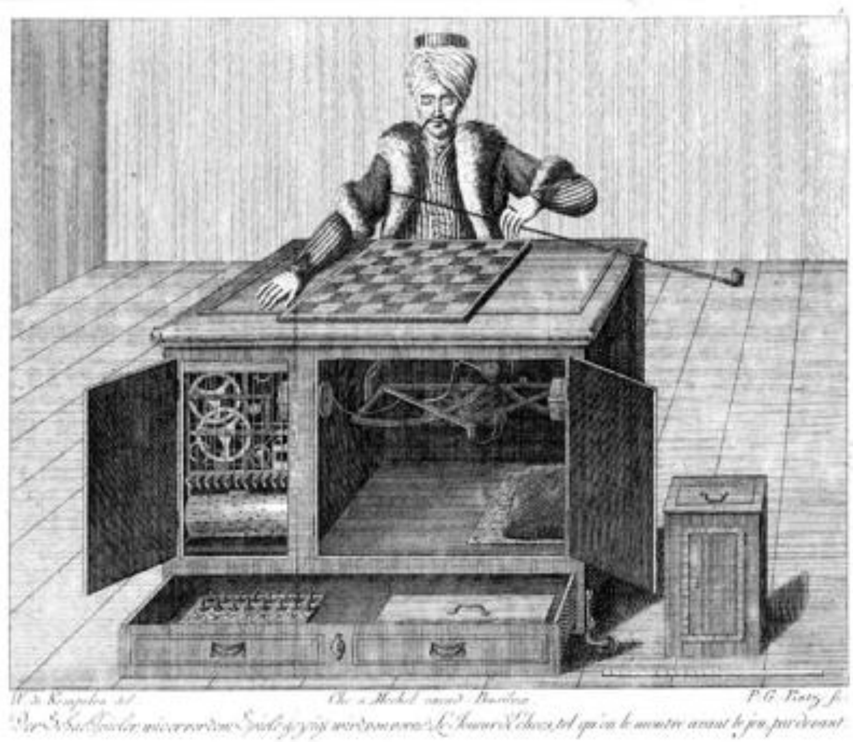
Low-Fidelity Prototype Revision

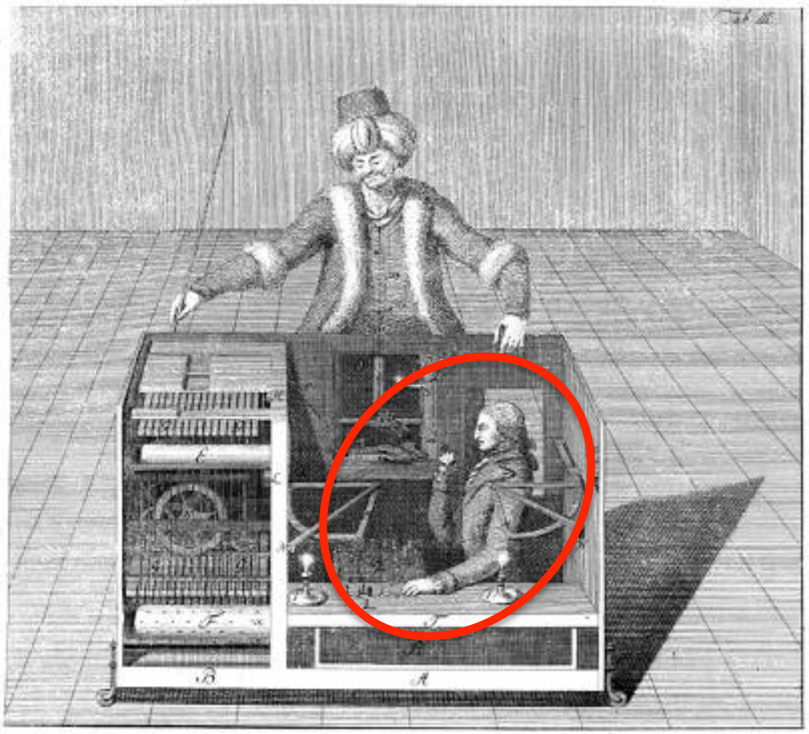
- Evaluation of test results
  - Arrange paper prototype on table
  - Pile note cards next to component
- Summarize and prioritize problems
  - Written report on findings
- Prototype refinement
  - Agenda for meeting to discuss design changes
  - Attach post-it notes with changes to each component



Wizard of Oz …

- The first “Chess Computer”
  - “In 1769, Hungarian nobleman Wolfgang von Kempelen astonished Europe by building a mechanical chess-playing automaton that defeated nearly every opponent it faced.”

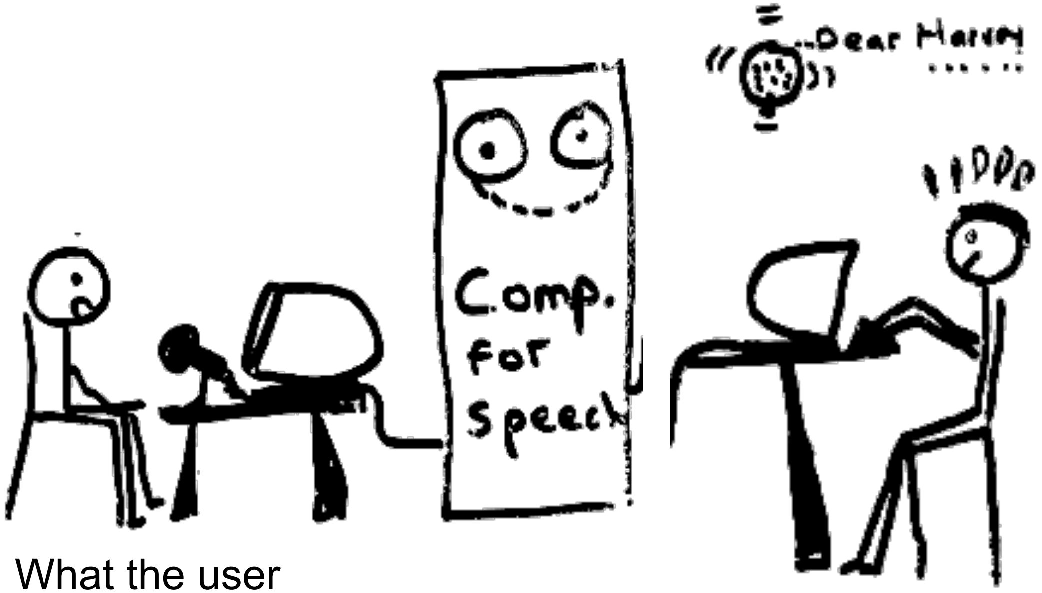




Source: http://collabor.f4.fhtw-berlin.de:8888/mmgestalt07s/topics/Beispiel/

… Wizard of Oz

- Method for testing a non-existing system
- Human “wizard” simulates system responses
  - Interacts with user via a simulated software user interface
- Useful for adding complex vertical functionality
  - Speech and gesture recognition, language translation



What the user sees

The Wizard

Mockup Tools for Mobile Devices

- Graphical sketching and interface builder tools
  - E.g., Pencil (pencil.evolus.vn/) open source project
- Predefined templates for inclusion in graphics programs
  - Widgets, icon sets
  - E.g., Android UI stencils (code.google.com/p/android-ui-utils/)
  - E.g., Balsamiq (mockupstogo.mybalsamiq.com/ projects/android/Android+Controls)

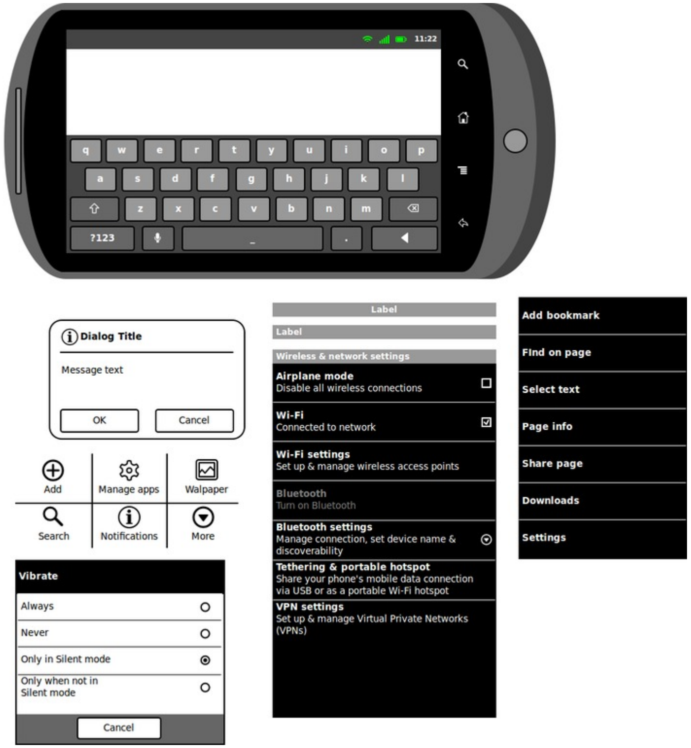


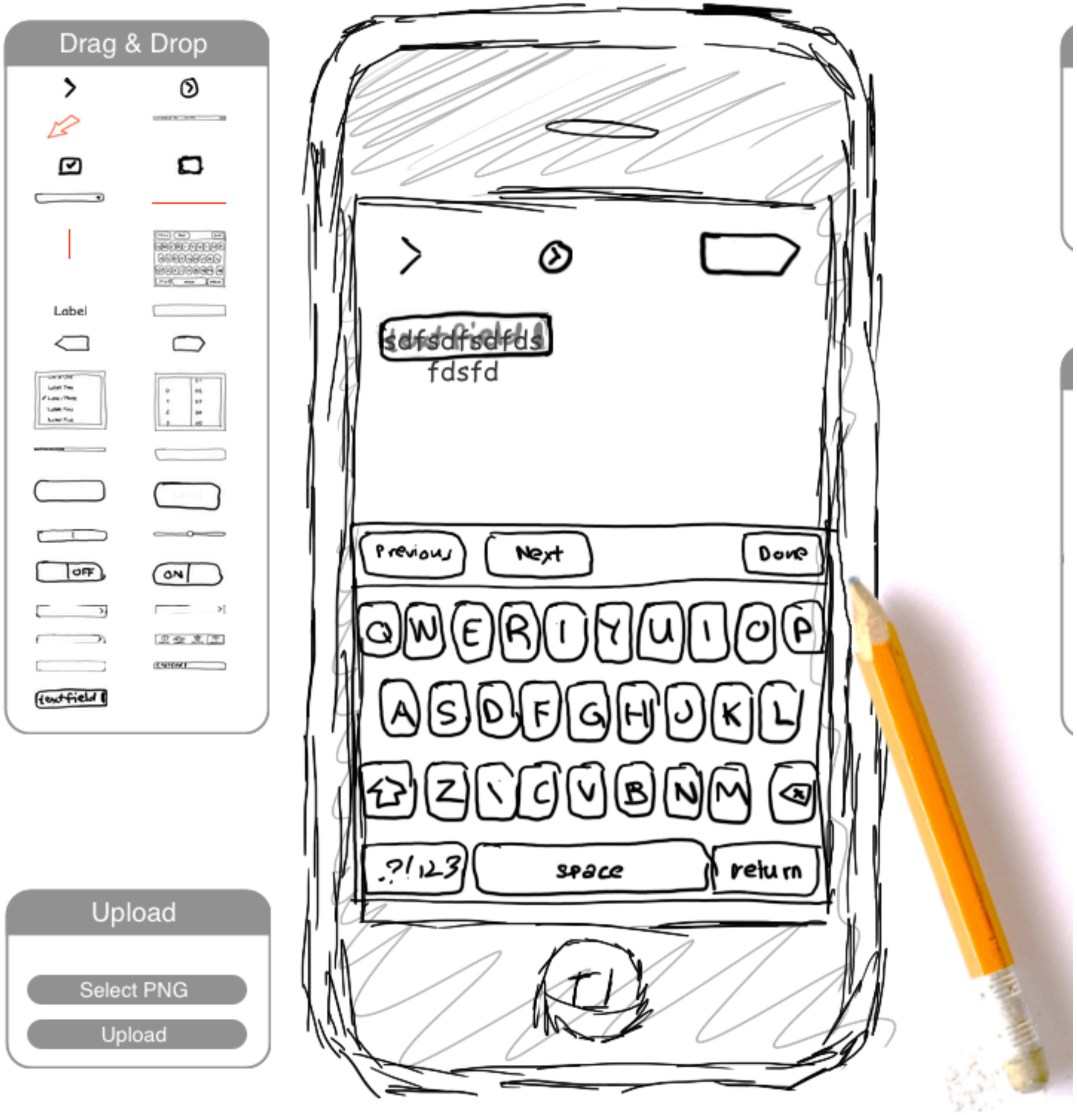
Image source: http://mockupsgallery.com/ android-stencil-for-nexus-one-phone

Mockup Tools for Mobile Devices

- Lots of tools for Android and iOS
  - http://iphoneized.com/2009/11/21-prototyping-mockup-wireframing-tools-iphone-app-development/
- Android App Developers GUI Kits, Icons, Fonts and Tools
  - http://speckyboy.com/2010/05/10/android-app-developers-gui-kits-icons-fonts-and-tools/
- iPhone & iPad GUI Kits
  - http://speckyboy.com/2010/04/30/iphone-and-ipad-development-gui-kits-stencils-and-icons/
- Discussion
  - http://stackoverflow.com/questions/4253197/gui-design-mockup-design-tool-for-android-application

Example: iPhone Mockup

iphonemockup.lkmc.ch



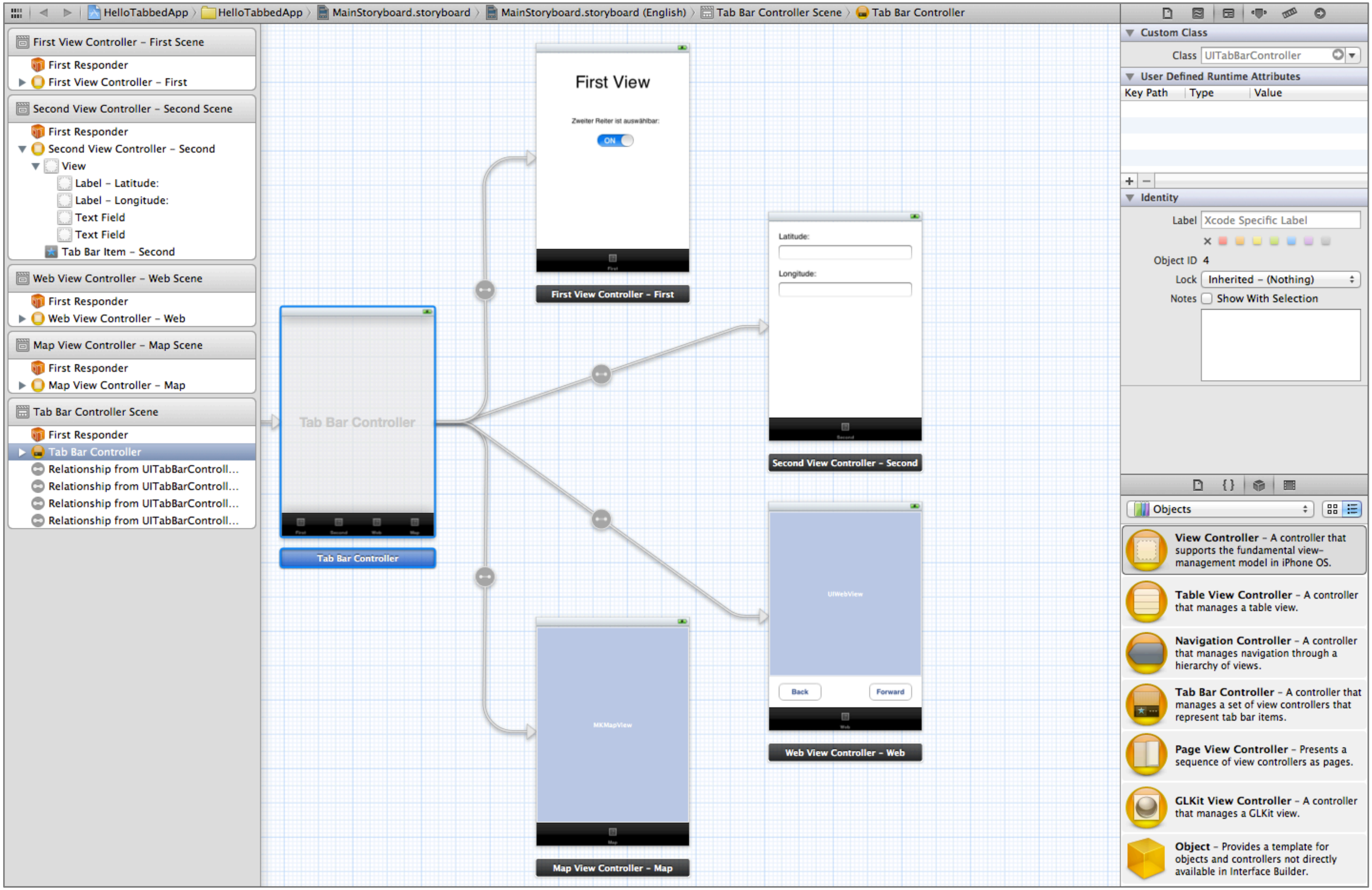
User Interface Builders

- Graphical tools to define UI of real software application
- Usually part of integrated development environment (IDE)
  - Xcode, Visual Studio, Eclipse, NetBeans
- Pro
  - Finished design can be used for final implementation
  - Real look & feel
  - Vertical functionality can be added easily
- Con
  - Limited to specific system and its toolkit (windows, buttons,…)

Example: Mac OS X Interface Builder

- Graphical tool to create UIs for Mac OS X applications
  - Part of Apple's standard Xcode IDE for Mac OS X (free)
- Visually place widgets (buttons, menus,...) in a UI
  - Static layout
- Define the connections between widgets and custom classes of the application that is being written
  - Dynamic behavior
- UI can be tested inside Interface Builder
  - Without compiling or writing any code
- Implementation process
  - Start with the user interface, not the application functionality
  - IB generates source code skeleton that can then be filled in

Example: Storyboards in Xcode, iOS5



HelloTabbedApp
HelloTabbedApp
MainStoryboard.storyboard
MainStoryboard.storyboard (English)
Tab Bar Controller Scene
Tab Bar Controller
First View
Zweiter Reiter ist auswählbar:
ON
First
First View Controller – First
Latitude:
Longitude:
Second
Second View Controller – Second
Tab Bar Controller
First
Second
Web
Map
Tab Bar Controller
UIWebView
Back
Forward
Web
Web View Controller – Web
MKMapView
Map
Map View Controller – Map

First View Controller – First Scene
- First Responder
- First View Controller – First

Second View Controller – Second Scene
- First Responder
- Second View Controller – Second
  - View
    - Label – Latitude:
    - Label – Longitude:
    - Text Field
    - Text Field
  - Tab Bar Item – Second

Web View Controller – Web Scene
- First Responder
- Web View Controller – Web

Map View Controller – Map Scene
- First Responder
- Map View Controller – Map

Tab Bar Controller Scene
- First Responder
- Tab Bar Controller
- Relationship from UITabBarControll…
- Relationship from UITabBarControll…
- Relationship from UITabBarControll…
- Relationship from UITabBarControll…

Custom Class

Class: UITabBarController

User Defined Runtime Attributes

Key Path	Type	Value

Identity

Label: Xcode Specific Label

Object ID: 4

Lock: Inherited – (Nothing)

Notes: □ Show With Selection

Objects

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

Tab Bar Controller – A controller that manages a set of view controllers that represent tab bar items.

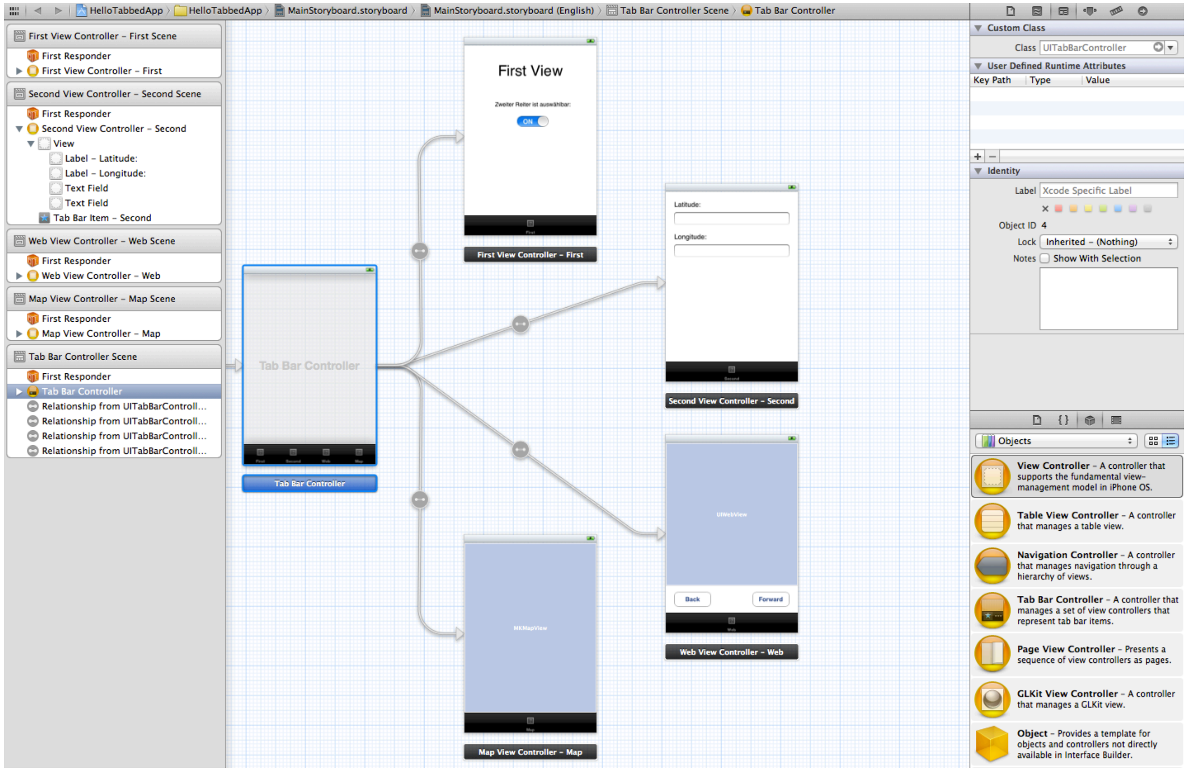
Page View Controller – Presents a sequence of view controllers as pages.

GLKit View Controller – A controller that manages a GLKit view.

Object – Provides a template for objects and controllers not directly available in Interface Builder.

Storyboards = Scenes + Segues

- Scene: A single screen of content
- Segue: Transition between scenes
- One storyboard per project, contains all scenes


HelloTabbedApp › HelloTabbedApp › MainStoryboard.storyboard › MainStoryboard.storyboard (English) › Tab Bar Controller Scene › Tab Bar Controller
First View Controller – First Scene
First Responder
First View Controller – First
Second View Controller – Second Scene
First Responder
Second View Controller – Second
View
Label – Latitude:
Label – Longitude:
Text Field
Text Field
Tab Bar Item – Second
Web View Controller – Web Scene
First Responder
Web View Controller – Web
Map View Controller – Map Scene
First Responder
Map View Controller – Map
Tab Bar Controller Scene
First Responder
Tab Bar Controller
Relationship from UITabBarControll...
Relationship from UITabBarControll...
Relationship from UITabBarControll...
Relationship from UITabBarControll...
First View
Zweiter Reiter ist auswählbar:
ON
First View Controller – First
Tab Bar Controller
Tab Bar Controller
Latitude:
Longitude:
Second View Controller – Second
UIWebView
Back
Forward
Web View Controller – Web
MKMapView
Map View Controller – Map
Custom Class
User Defined Runtime Attributes
Key Path
Type
Value
Identity
Objects

Class UITabBarController

Label Xcode Specific Label

Object ID 4

Lock Inherited – (Nothing)

Notes □ Show With Selection

View Controller – A controller that supports the fundamental view-management model in iPhone OS.

Table View Controller – A controller that manages a table view.

Navigation Controller – A controller that manages navigation through a hierarchy of views.

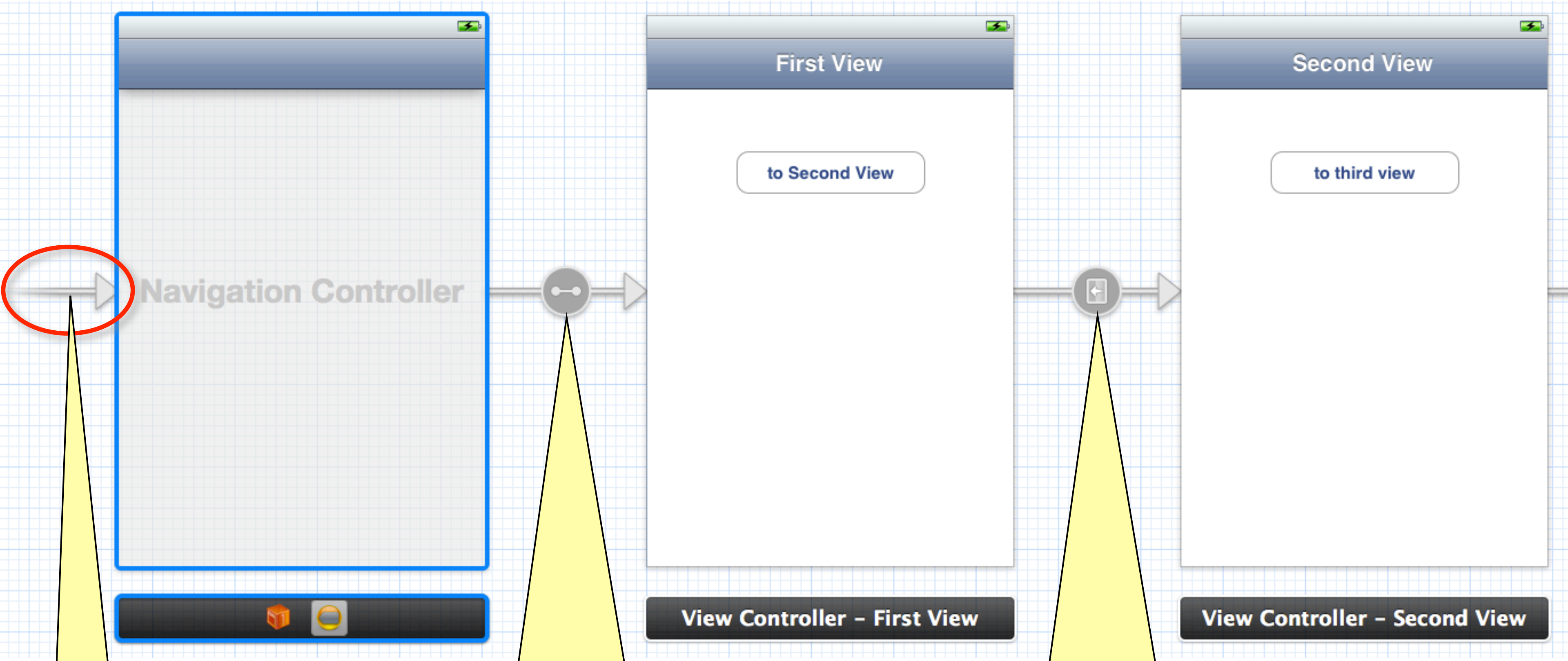
Tab Bar Controller – A controller that manages a set of view controllers that represent tab bar items.

Page View Controller – Presents a sequence of view controllers as pages.

GLKit View Controller – A controller that manages a GLKit view.

Object – Provides a template for objects and controllers not directly available in Interface Builder.

Scenes and Segues



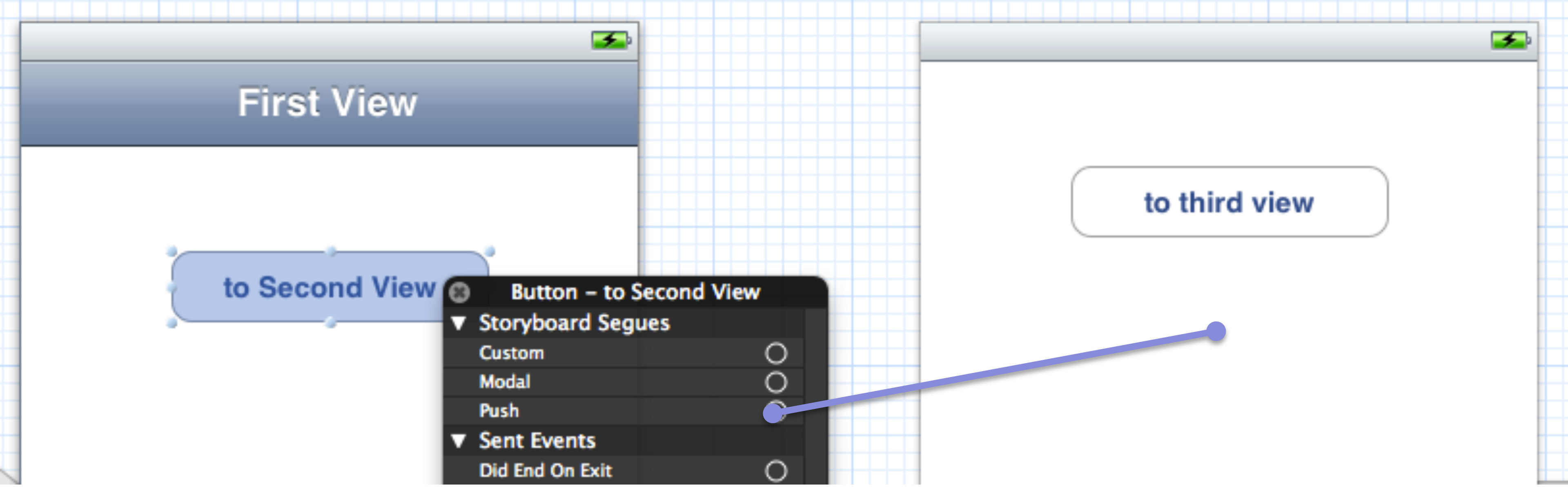
Point to initial scene

Relationship link between navigation controller and root view controller

Push segue between root view controller and second level view controller

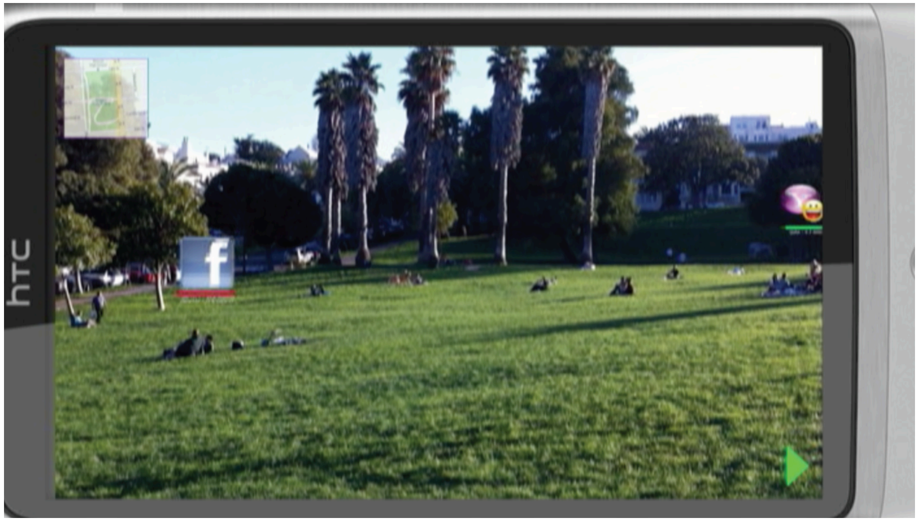
Creating Segues

- Ctrl-click element that invokes second scene (e.g. button)
- Or right-click this element



Mobile AR Prototyping

- "Low-fidelity" AR prototyping
- Take video, edit video off-line, show video to users in the context ("guerilla user test")



de Sa, et al.: Mobile Augmented Reality Prototyping, CHI EA 2011

Summary

- Prototyping enables rapid feedback on design ideas
- Different kinds for different purposes and design stages
- Choose appropriate prototype for question to answer
- Low-fidelity vs. medium-fidelity vs. high-fidelity
- Many approaches, methods, and tools