BUILDING MOBILE EXPERIENCES

Frank Bentley
Principal Staff Research Scientist
Motorola Mobility

University of Zurich Summer School 2011

Who am I?

- 10 years experience in Motorola Research
- 6 years of teaching mobile apps class at MIT
- Strong believer in user-inspired innovation
- Have brought several new services to market including TuVista, MotoBLUR, and StoryPlace.me

What is the class about?

- Investigating the interaction between people and mobile computing devices
- Creating compelling mobile applications and services that fit into daily life
- Rapid prototyping and evaluation on paper and through digital tools (e.g. App Inventor)

The Day

- 8:45 - 9:00 Check-in
- 9:00 - 10:15 Unique Aspects of Mobile, Rapid User Centered Design Cycle
- 10:15 - 10:45 Coffee break
- 10:45 - 12:00 Mobile Design
- 12:00 - 13:00 Lunch
- 13:00 - 14:45 Paper Prototyping and Usability Evaluation
- 14:45 - 15:15 Coffee break
- 15:15 - 17:00 Rapid Prototyping and Field Evaluations

Mobile Computing is Everywhere

An Internet-connected computer in 5 billion pockets…

Why study mobile computing?

- Changing lives of many
  - Farm Prices
  - Microfinance
  - Live Media Sharing
  - Constant Access to Information
- Mobile Applications Becoming Ubiquitous
  - iPhone App Store
  - Android Marketplace
  - Windows Marketplace for Mobile
- Ability to get your app out, even as a researcher

Question?

- What are some of your favorite mobile apps?

What's unique about mobile?

- Media Capture
  - Ability to share your world with others
  - Document life for yourself, anywhere
- Sensing
  - Location, accelerometer, compass, body sensing
  - Get content relevant to your context
  - Sense and track personal health
- Social Connectivity
  - Device tied to a person, almost always with them
  - Voice, text, email, chat, media sharing…

Top apps in the Android Market

App Name	Media Capture	Sensing	Social Connectivity
Google Maps		X	
Facebook	X	X	X
Pandora			
YouTube	X	X	X
Words with Friends			X
Advanced Task Killer			
Skype			X
Barcode Scanner	X	X	
Weather Channel		X	
Google+	X	X	X

* Excluding all variants of Angry Birds

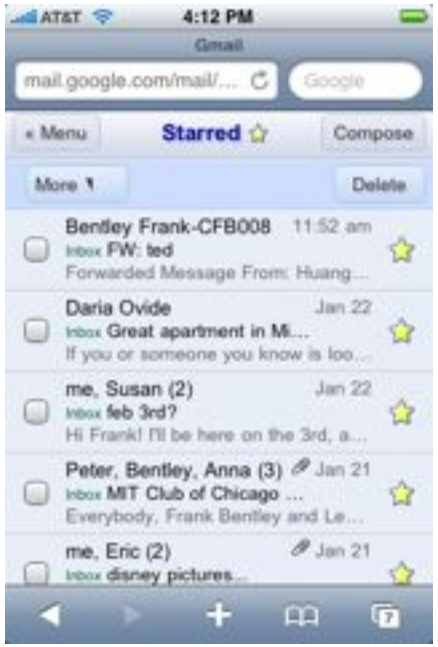
Apps from Motorola Research

App Name	Media Capture	Sensing	Social Connectivity
ZoneTag	X	X	X
TuVista	X	X	X
BLUR	X		X
StoryPlace.me	X	X	X
Motion Presence		X	X
Music Presence		X	X

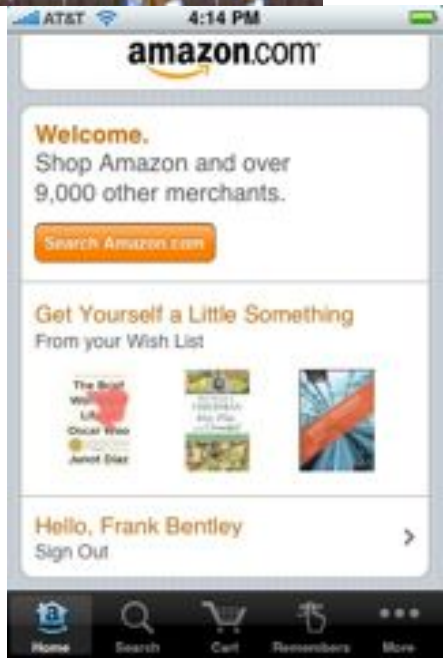
Mobile Ecosystem

Web	J2ME	Native
• No installation • Limited interaction with phone platform • Rendering issues on different handsets • Some reach to almost every device	• Works on most non-smart phones • More interaction with phone platform • Consistent Look and Feel • Lifecycle limitations	• Written for a particular OS • Deepest interaction with phone platform • Background Processes • More complex distribution/ testing

Mobile Web Applications



Native Applications



Path to consumer

- Web:
  - Go to a URL (but they must know/find that URL)
  - Instant and updatable totally from server side
- J2ME:
  - Carrier marketplace
  - No paths for updates unless it's a part of your app
- Native:
  - Platform App Store (Apple, Google, MS)
  - Updates can be placed in app stores / users notified on some schedule, but not instantly

Current Mobile Research Areas

- Location-Based Mobile Computing
- Persuasive Applications
- Social Networking / Web 2.0
- Extending Experiences
- Personal Networks/Wearables
- Companion Devices
- Enterprise

Location-Aware Computing

- How can location help make any mobile task more efficient?
  - Finding restaurants
  - Getting movie tickets
  - Knowing which bus to take
  - Tagging photos
  - Finding friends
  - Know where to sell their crops
  - Mobile tour guide/games

REXplorer

Mobile Persuasion

- How can mobile phones convince people to…
  - Eat healthy food
  - Work out
  - Use less energy
  - Help their friends and family
  - Volunteer
  - Be religious
  - Save the planet

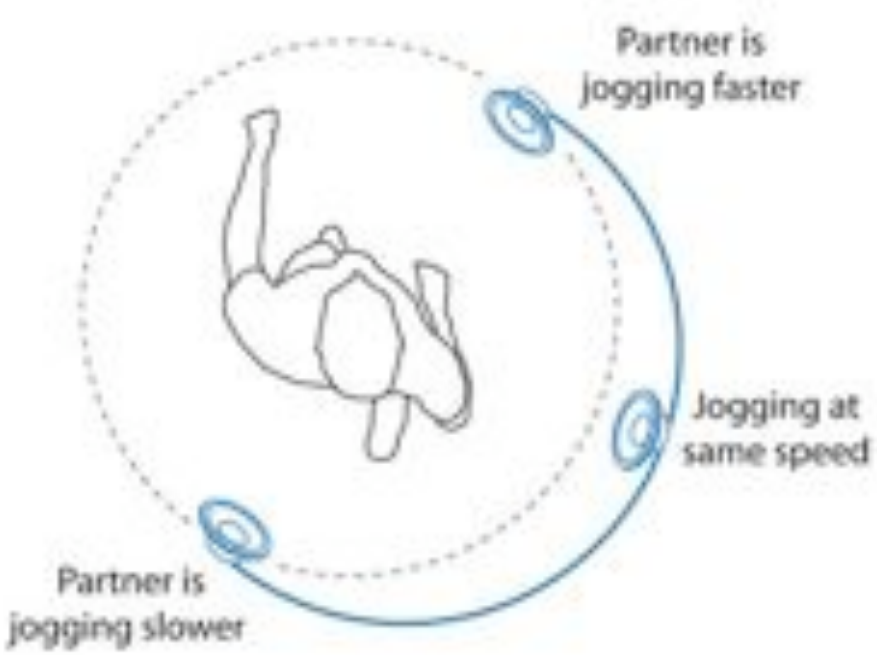
Social Networking / Web 2.0

- How can phones link in data from online communities?
  - Status in contacts app
  - Micro-coordination (helping plan and meet up)
  - See photos from friends
  - Syncing online calendars
  - Selling goods/services
  - Managing group finances / microfinance

Yahoo! OneConnect

Extending Experiences

- How can an experience on a phone augment an in-person experience?
  - Sports – replays/stats on phone
  - Concerts/Festivals – see other people's photos/ videos in real time
  - Working out – virtual workouts with others



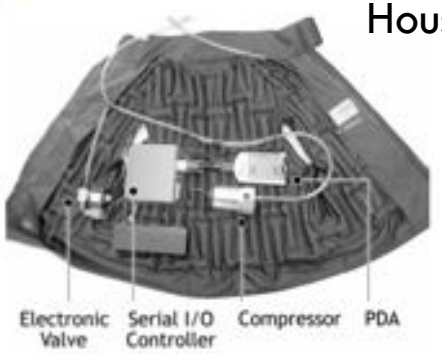
Jogging the Distance (Floyd Mueller)

Personal Networks/Wearables

- What can you do when you combine a phone with other sensors/actuators?
  - Step Counters
  - Pollution Sensors
  - Actuated Hugs

Houston (Intel Research)

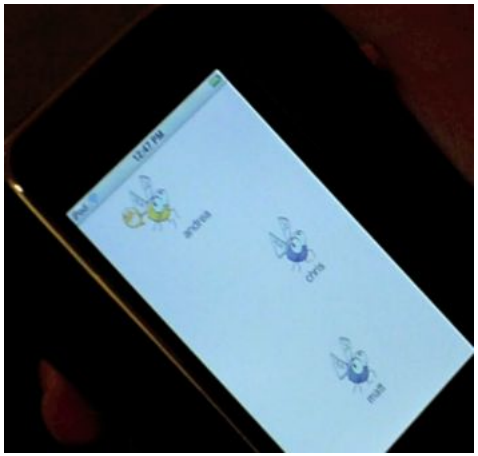
inAir (CMU)



Hug Over a Distance (Melbourne)

Companion Devices

- How can mobile devices be used to complement other screens (TV/Computer/Games)?
  - Displaying "private" information
  - Interactions off-screen to not disturb others
  - Different content for different people



BackTalk (Media Lab)

DIRECTV iPhone app

Enterprise

- How can mobile devices be used in a work context?
  - Routing people more efficiently
  - Getting additional information about products from web / checking inventory
  - Automatic check-in based on Bluetooth ID

RFID Reader

QR Code for Wikipedia

And many more…

- Check out latest proceedings of Ubicomp, Pervasive, Mobile HCI, etc. for more.

Mobile is different

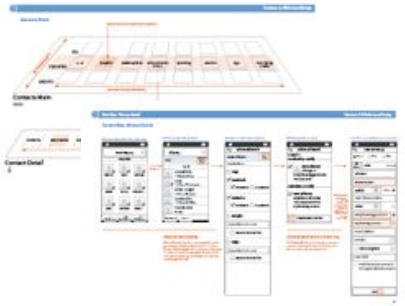
- Interactions happen in the world
- Interactions are generally private and not directly observable
- Interactions are short and spread throughout the day

- Requires different field methods than traditional web/desktop computing

Much more than just writing the code

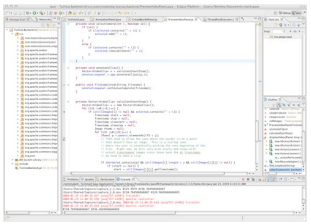
Observation

Ideation



Design

Build



Test

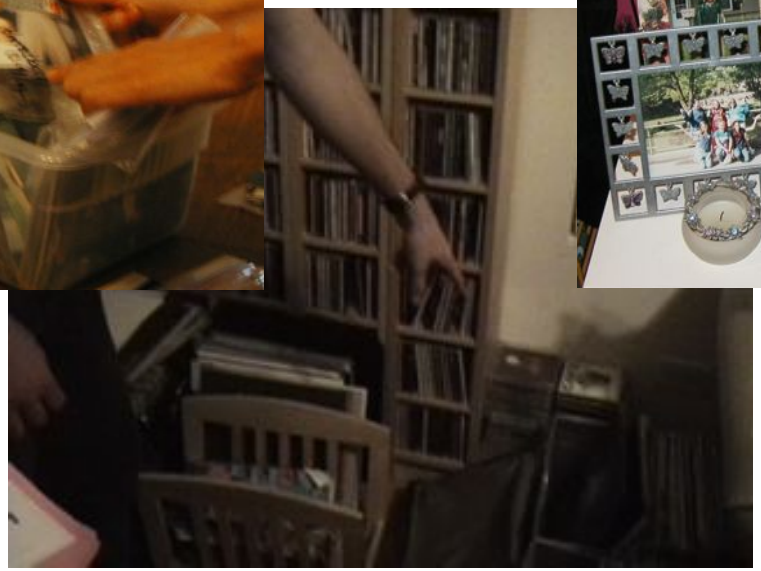
Rapid UCD Process

1. ethnographic-style investigations in a new space of interest
2. concept generation and prioritization
3. initial prototype implementation (days or weeks)
4. field test of new system
5. iterate
6. product decision
7. work with marketing, design, engineering, and sales teams to create product version

Can stop work at any time if value is not shown

Inspiration for design

- Learning from people
- In-situ observations of related practices
- Grounding new designs in real-world behavior

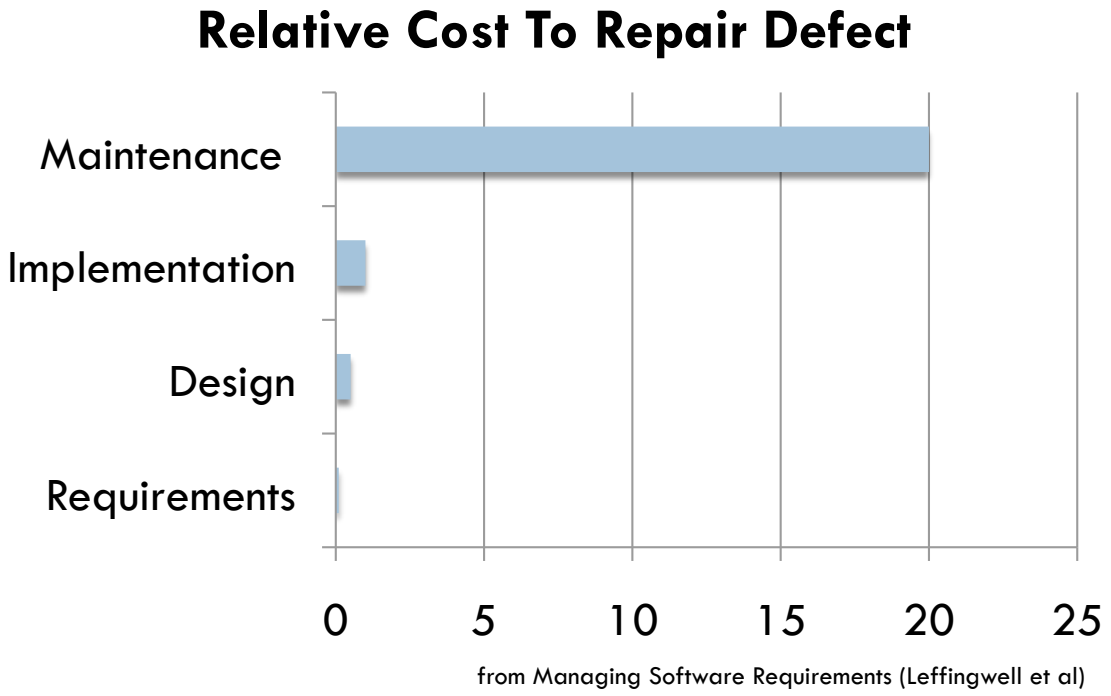


Coherent Design

- Design is a process from the beginning
- Ensure interaction flows for a user
- Especially important on a small-screen mobile device!

Building and Testing are Iterative

- Increasing fidelity with quick tests along the way
- From paper to working in the world
- Catch mistakes early when it's easy to change



Inspiration for Design

- Design of a new application/service should be grounded in daily realities
- Should work with how people think about each other, their environments, and the world

- Need to get out into the world to learn this…

Pick a topic area and explore…

- we have studied:
  - in-family communication (2002)
  - photo sharing (2003 & 2007)
  - music use (2005)
  - location sharing (2006)
  - social television (2007-2009)
  - trips to sporting events (2008)
  - intergenerational distance communication (2010)

Now…

- Introduce yourselves, discuss mobile interests
- Think of domain areas that you might be interested in prototyping today

Methods for understanding users

- Observation
- Home Tours
- Diary (and/or system) Logging
- Interviews

Methods: Observation

- Watch how users interact with a space/object/each other
- Good for observing many people, findings patterns
- Good when interaction times/locations are predictable
- But only get what, not why

- Used by students for grocery store behaviors, wayfinding in public spaces, booking travel

Methods: Home Tours

- Visit a home or work setting and see particular places of interaction
- Good for tasks which are very context dependent / rely on physical objects

- Used in Elder Communication study, music study (CDs), photo study (photos on display, use of computers, etc.)

Methods: Diary Logging

- Have participants keep a log (paper, voicemail, voicenotes, etc.) when they do particular things of interest
- Get data at time event happens. MUCH more reliable than recall some time later
- Can also be combined with system logging

- Used in almost all of our studies: Elder Communication, Location Sharing, etc.

Methods: Interviews

- Interviews complement direct observation
- Should focus on understanding current practice
- NOT future concepts, “would you like …,” etc.
- Generally semi-structured with probes that follow up in interesting areas of use

- Used in every study we’ve done

Examples: Music Usage

- Home tours, Interviews, Contextual Inquiry
- 12 participants
- Explored how they searched for, selected, played, and acquired music
- Findings: Satisficing, Ruts + Kicks, More like this
- Designs: Metadata Knob, Playtree, Music Presence

Examples: Location Sharing

- 5 participants
- Recorded phone calls for 1 week
- Interviews + Statistical data analysis of calls

- Findings: People already share basic location context, transition times mostly unknown, confirmation of context confirms availability
- Designs: Motion Presence, Contacts 3.0/MotoBLUR

Question?

- Which methods would you use to explore your domain of interest?
- How would you design the study?

- Try to refine your topic area into a concept before the coffee break

Coffee Break

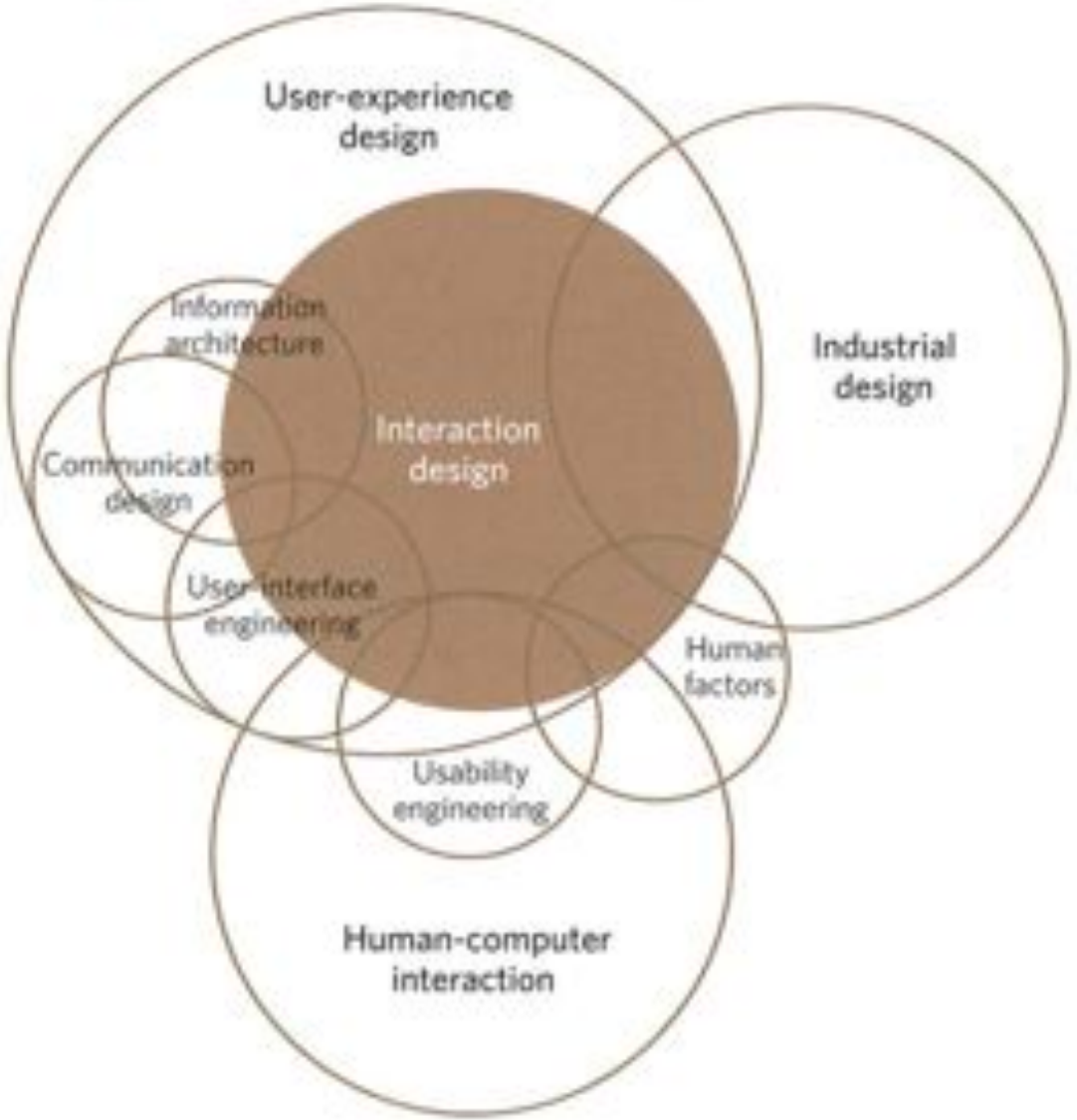
- Please be back by 10:45

Mobile Design and Prototyping

- Concept models
- Flows / User environments
- Screens and details

interaction design

interaction design



Source: *Designing for Interaction: Creating Smart Appliances and Clever Devices*, Dan Saffer

interaction design > interface design

Modeling

Concept model – at a high level, how does your solution solve your users' needs?

Interaction Framework – begins to describe how users will interact with your solution.

Structure…Flow…Process (how do the pieces begin to fit together?)

Design at the system level

aka "Interaction Design"

Affordances (what does this button do? What is its purpose?)

Design at the screen level

aka "Interface Design"

modeling

modeling: concept models

How do users think about your system?

What are the activities that they can do?

What are the different types of users that will use your system?

What are the different modes in which they might use it?

What are the different phases that they might associate with your system?

How is it different than what they used before?

Is there a familiar metaphor that will help users understand your system?

modeling: interaction framework

What should your system do?

Functionality should map to goals
Life Goals – beyond the system, but help explain why trying to accomplish end goals
Experience Goals – how someone wants to feel while using the product
End Goals – outcomes users expect from using the system

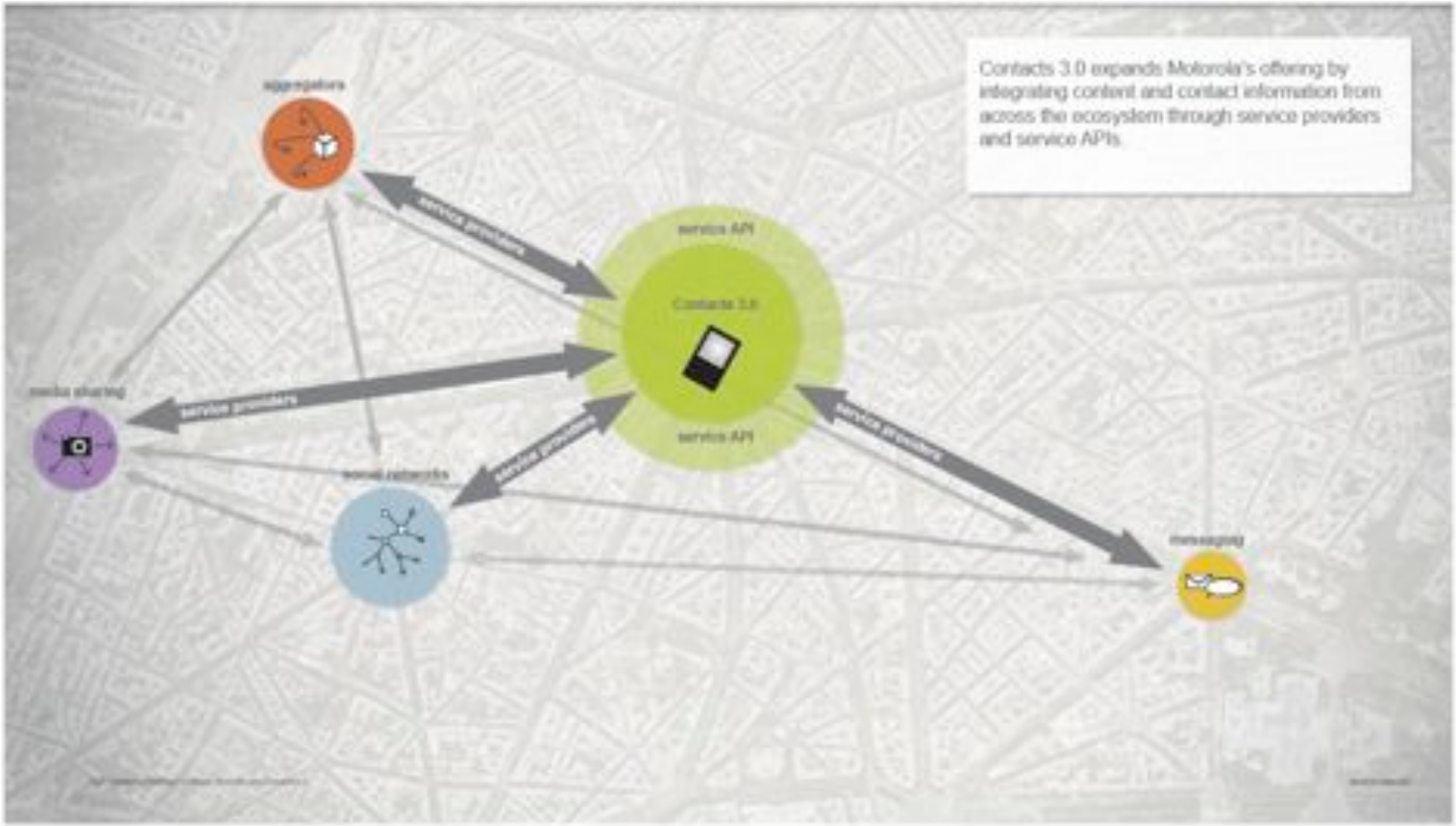
How should it work?

Cooper's Interaction Framework

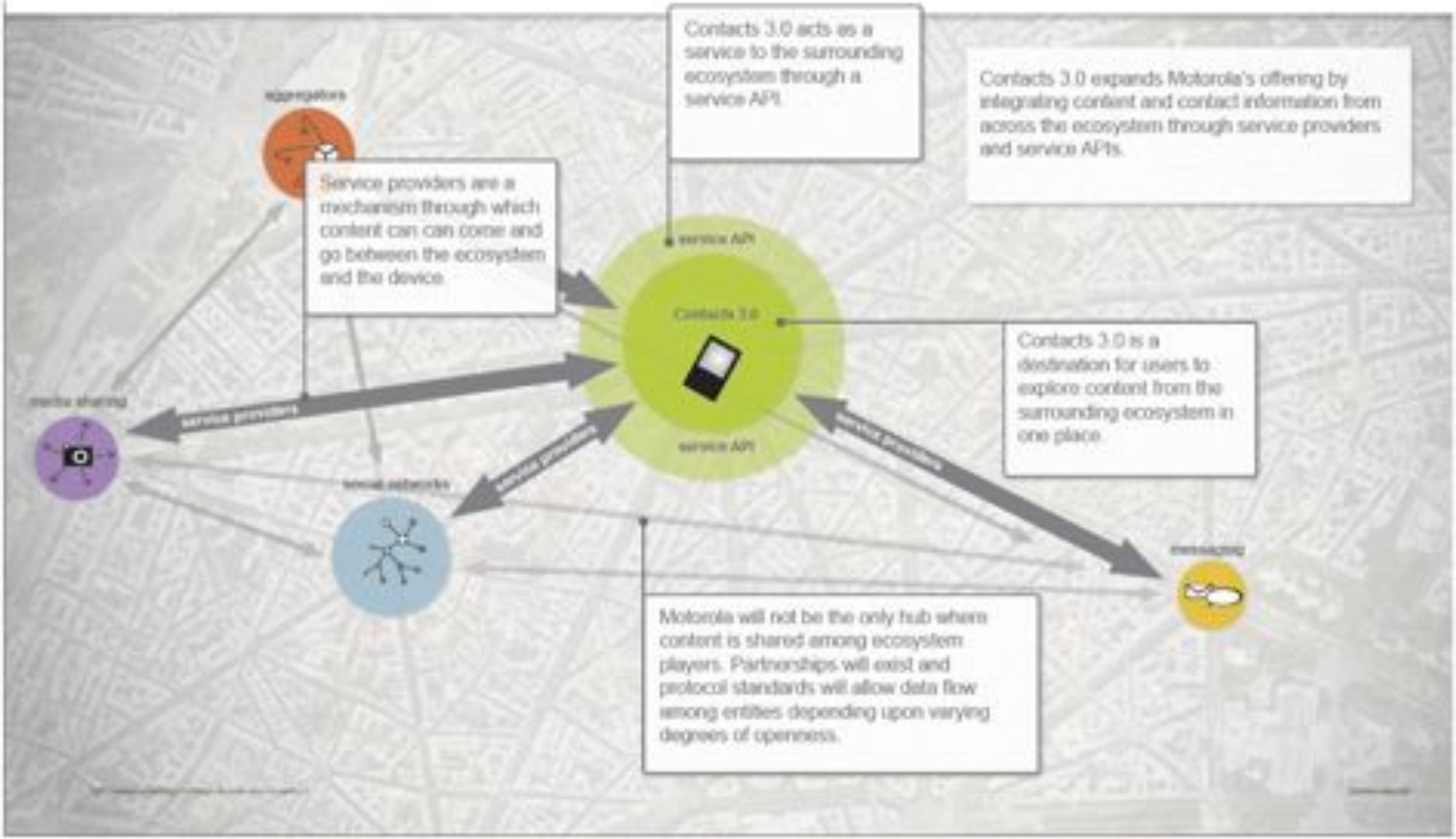
1. What is the form factor and input methods?
2. What are the different views?
3. What are the functional and data elements? (basically, parts of a larger key path)
4. What are the functional groups and hierarchy? (sequence, groupings of containers, functionality – based on form factor and input methods)
5. What does the interaction framework begin to look like?
6. What are the key path scenarios? (primary actions and pathways through the system, e.g. viewing and composing emails)

modeling: using metaphors

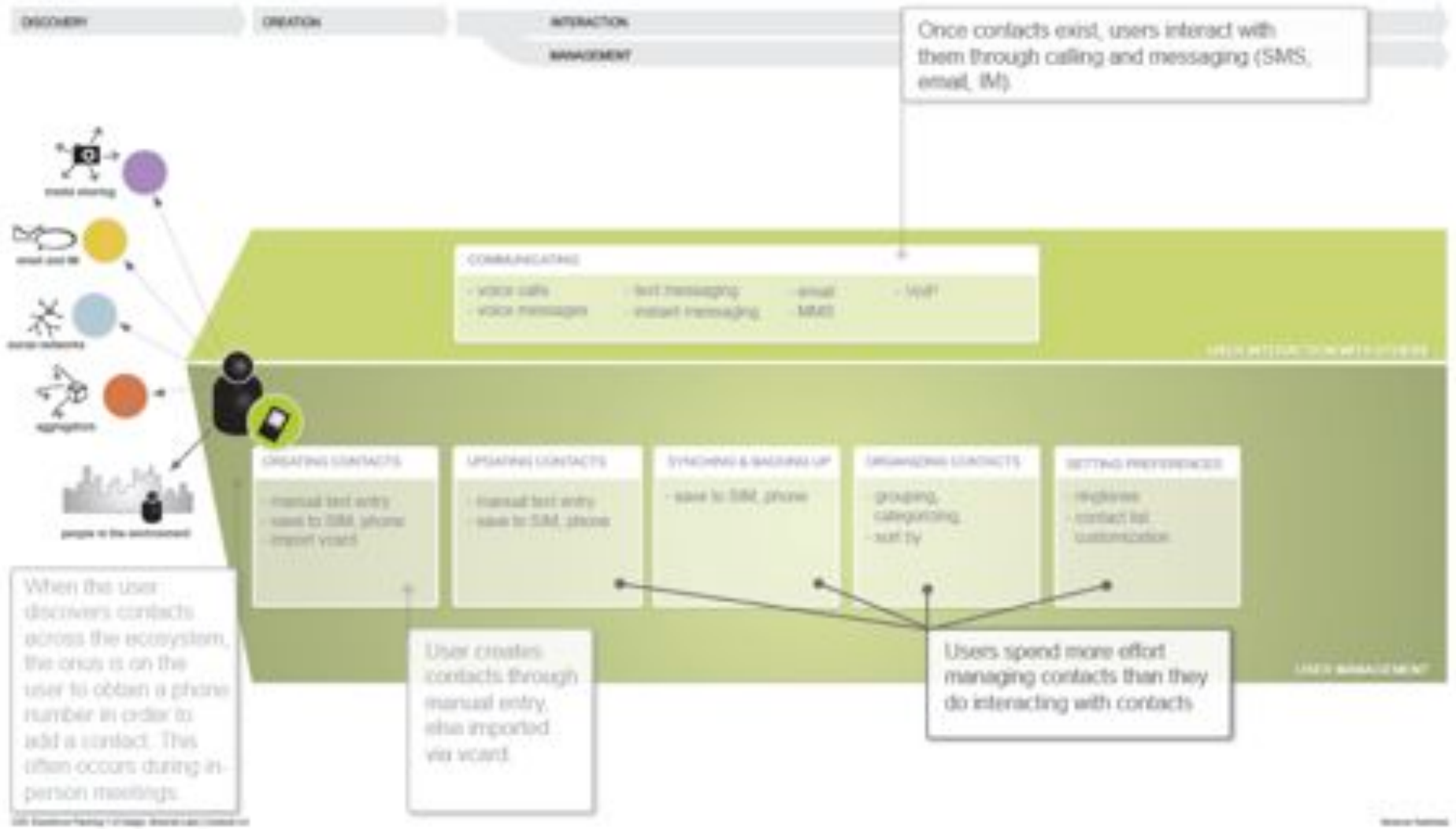
modeling: using metaphors



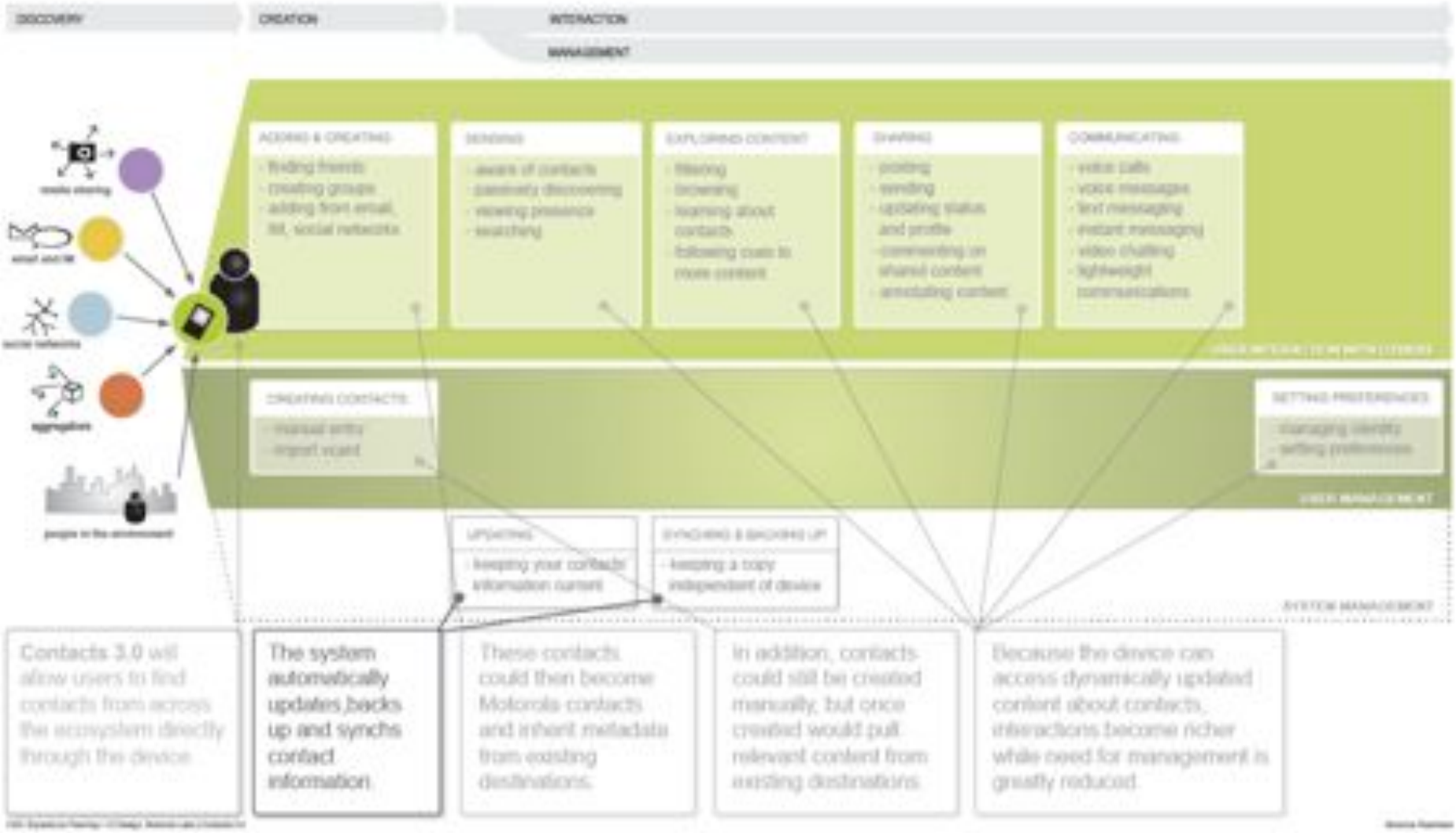
modeling: using metaphors



modeling: before + after



modeling: before + after



Concept Models

- Before and after
- Metaphor
- User/Task analysis
- Modes

- Create a concept model for your app/service

structure...flow...process

structure…flow…process: use cases + user flows

Developing the structure of the system

Its not a linear process – iterate back and forth between user flows + wireframes

Showing user experience flows vs system/business flows

Flows demonstrate users movement through time – how the user begins, ends and the clearly marked path that they take

Demonstrating cause + effect

Consistency is key in diagramming systems – there is no "right" visual language

Visual vocabulary – borrowing and adapting to make it your own

structure…flow…process: use cases

Use Case Catalog

Click here to access the
document on Lifeagora

Key
C = Core
I = Important
M = Must-to-have

Communicating	Priority
Call to Contact	C
Send Message (or reply to) a Contact (SMS, MMS)	C
Send Message a Group (SMS/MMS)	C
Email to Contact	I
Email a Group	I
Send (Lightweight) Communication to Contact	M
Send (Lightweight) Communication to Group	M
Instant Message a Contact	M
Instant Message a Group	M
Send micro note to a Contact	M
Send micro note to a Group	M
Send Message to a Contact through an online service (i.e. Facebook)	C
Post on someone's Wall on Facebook (or equivalent for another Service)	C
Request info from a contact (Location, Contact Info, etc.)	C
Comment on Contact's media	C
Learning	
View Contact's Status Message from online services (i.e. Facebook)	C
View Contact's Mood	M
View Contact's exact Location (cross-streets, address, dot on map)	I
View Contact's vague Location (City, State)	I
View Contact's user-defined Location (home, work, bag, etc.)	I
View Contact's time zone	I
View Contact's Distance from me (exact - 2 mi)	I
View Contact's Distance from me (general - same city, near, etc.)	I
View Contact's Media Presence (moving/not moving, location)	I
View Contact's preferred communication method	I
View Contact's Music Status	I
View Contact's (newest/recent) photos	C
View Contact's comment on a photo	
View Contact's recent posts/online activities (short, eg. Twitter)	C
View Contact's recent posts/online activities (long, eg. Blog)	C
View Contact's comment on a Blog post	
View Contact's Facebook Profile update	
View Communication History with Contact (Recent Calls, etc.)	C
View Business specific data (Hours, Locations, etc.)	I
View Contact's IM Status now	I
View Contact's Calendar availability	I
View Contact's Calendar	M
View Contact's contacts, network	M
View Contact's online profiles (Facebook, Myspace, LinkedIn)	C
View Contact's weather where they are (attach Widget to a Contact?)	I
View Contact's Blog Profile (blog, tumblr, etc.)	M
View basic Contact Information (Phone numbers, Email, IM, etc.)	C

Sharing	
Share My Info as Icon on the Social Dashboard	
Set my Status (make avail, OK push)	C
Set my photo (to appear to others' Contact lists)	C
Set my Mood (make avail, OK push)	M
Share (make avail, OK push) my exact Location	I
Share (make avail, OK push) my user-generated Location	I
Share (make avail) my Motion Presence	I
Share (make avail, OK push) my vague Location (City, State)	I
Share (make avail, OK push) my Time Zone	I
Share (make avail, OK push) my Calendar	I
Share (manage permissions for) my current media (what I'm watching, Blogs, Music?)	I
Share my preferred communication method	M
Share (make avail) my IM Status	I
Share my online identities (url to any online profiles that a user has)	C
Share myself as a Contact (my VCard)	I
Share my Contact's (my) Network	
Make an introduction to a Contact (like LinkedIn) (sending Contact Info (like sending VCard))	I
Send content to a contact (link to media, Blogpost, etc.)	C
Send content to a group (link to media, Blogpost, etc.)	C
Send Meeting/Event Invite	I
Share business specific data (if Contact is a business)	M
Searching/Finding/Browsing	**Priority**
Search for a Contact on Device	C
Search for a Contact off-device	
Search by Tag	
Browse Contacts	C
Filter/sort Contacts (Main by	C
Tag	C
Location	I
How (or recently) Playing (media)	I
Recency of communication	C
Recent Updates	I
Calendar (Upcoming)	
Function Specific activity (Now playing media)	
Group	
Favorites (suggested by Response.p)	C
Filter/sort Social Dashboard Content (Create a Social Web updates) by:	C
Group (from Contacts App)	C
Media Type (off-Device)	C
Recency (time based by default)	C
Communication Type (on-Device: calls, Messages)	C
Web (off-Device)	
Phone (off-or-Device)	
Web Service	C

Creating	
Create a Group (from Page)	C
Create a Group from a shared group (or Contact' send/share the Group with another member)	C
Create a Group from a Message (sending to multiple recipients)	C
Publish Group (notify Contact that they are in a Group) and give option to add groups	I
Create a Group from scratch	C
Create a Contact from Web Service (custom API's like Facebook, etc.)	C
Create a Contact from another Device	M
(including "sharing" and Tagging functions)	C
Create a Contact from scratch	C
Create a Contact from Group based Data	C
Request an introduction to a Contact (requesting Contact Info for another person)	I
Make Contacts (make Favorites)	M
Block a Contact	C
Block particular fields for a Contact	C
Block a Group	C
Disband a Group	C
Managing/Updating/Syncing	
Sync/update Contacts with Web Service	C
Sync/update Contacts with another Device (PC or second phone)	I
Save Contacts to SIM	C
Save Contacts to iPhone	C
Save Contacts to Service (WAP - Backup/Load/Sync, 3rd party?)	I
Customize/set preferences for Contacts (Detail)	C
Customize/set preferences for Contacts (Main)	C
Settings (p)	
Import Contacts from Online Service	C
Reconcile/exchange Contacts across information sources (main thing, links on Facebook to Jobs in Contact Info)	C
Set which content sources I want from a given Contact	I
Receive notification that a Contact has edited their Contact Info and there is a conflict (following auto-Sync)	I
Respond to conflict alerts following automatic sync/update with online sources (in background)	I
Set sync (Web service) option to manual	C
Configure automatic Sync	
Choose who to add from a given Service	C

structure…flow…process: use cases

Identifier	Use Case	Priority		
Communicating				
A	Call a Contact	C	x	x
B	Text Message (or reply to) a Contact (SMS/MMS)	C	x	x
C	Text Message a Group (SMS/MMS)	C	x	x
D	Email a Contact	I	x	x
E	Email a Group	I	x	x
F	Send Lightweight Communication to Contact	N	x	x
G	Send Lightweight Communication to Group	N	x	x
H	Instant Message a Contact	I	x	x
I	Start a Group chat from Group Detail	I		x
J	Send voice note to a Contact	N		x
K	Send voice note to a Group	N		x
L	Message a Contact on online service (i.e. Facebook)	C	x	x
M	Communicate via service specific methods (i.e. Post on someone's wall on Facebook)	C	x	x
P	Request information from a Contact (exact location; duration, contact info)	I	x	x
	Request information from a Contact (Introduction to another Contact)			x
Q		C	x	x

Task list (Use cases)

- Tasks are high level concepts of purposeful use
- Most revolve around end states the user would like to be in (e.g. “select music to play” “play desired music” “add music to collection”)
- Not individual requirements for a system
- Ideally, tasks come from observations, user needs

Task list (Use cases)

Make a task list for your application

Requirements list

- Functional requirements needed to accomplish tasks
- Can be user facing (visible) or system facing (hidden)
- Should exhaustively enumerate everything the application/system has to perform
- Prioritize list to determine what will be implemented / what can safely be omitted in early versions (common prioritizing is Core, Important, Nice to Have)
- Prioritize by use (Used by many, most, few) and expected frequency (Used often, sometimes, rarely/once)
- You'll rarely be able to implement everything or cleanly fit it into a design

Requirements list

- Develop requirements for each task…
- Select music to play:
  - **Select metadata attributes to search on**
    - **Search on Artist**
    - **Search on Album**
    - **Search on Genre**
    - **Search on Playlist**
    - **Search on Year**
  - **Search on combination of attribute/values**
  - **View values for the given attribute**
  - **Select an attribute**
  - **View songs matching the query**
  - **Play entire results list**
  - **Play starting at an item in the results list**

Requirements list:

Make a requirements list for your application

structure…flow…process: organizing info + user flows

Information architecture

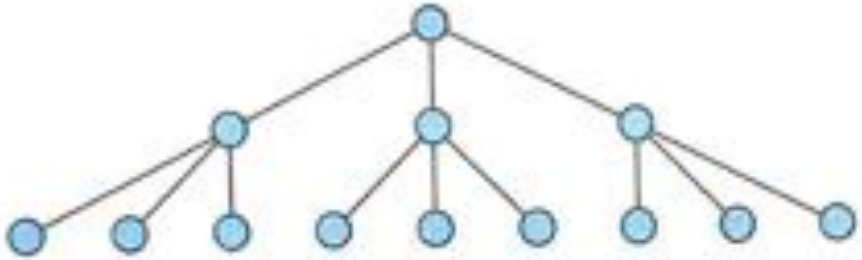
Especially important for heavy content (more at the interface level)

top down

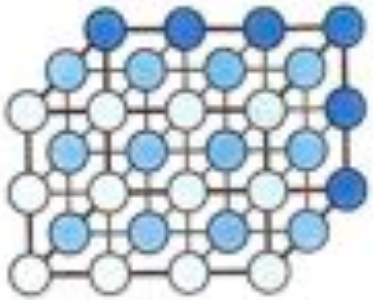
bottom up

structure…flow…process: organizing info + user flows

Architectural Approaches



Hierarchical parent and child relationships

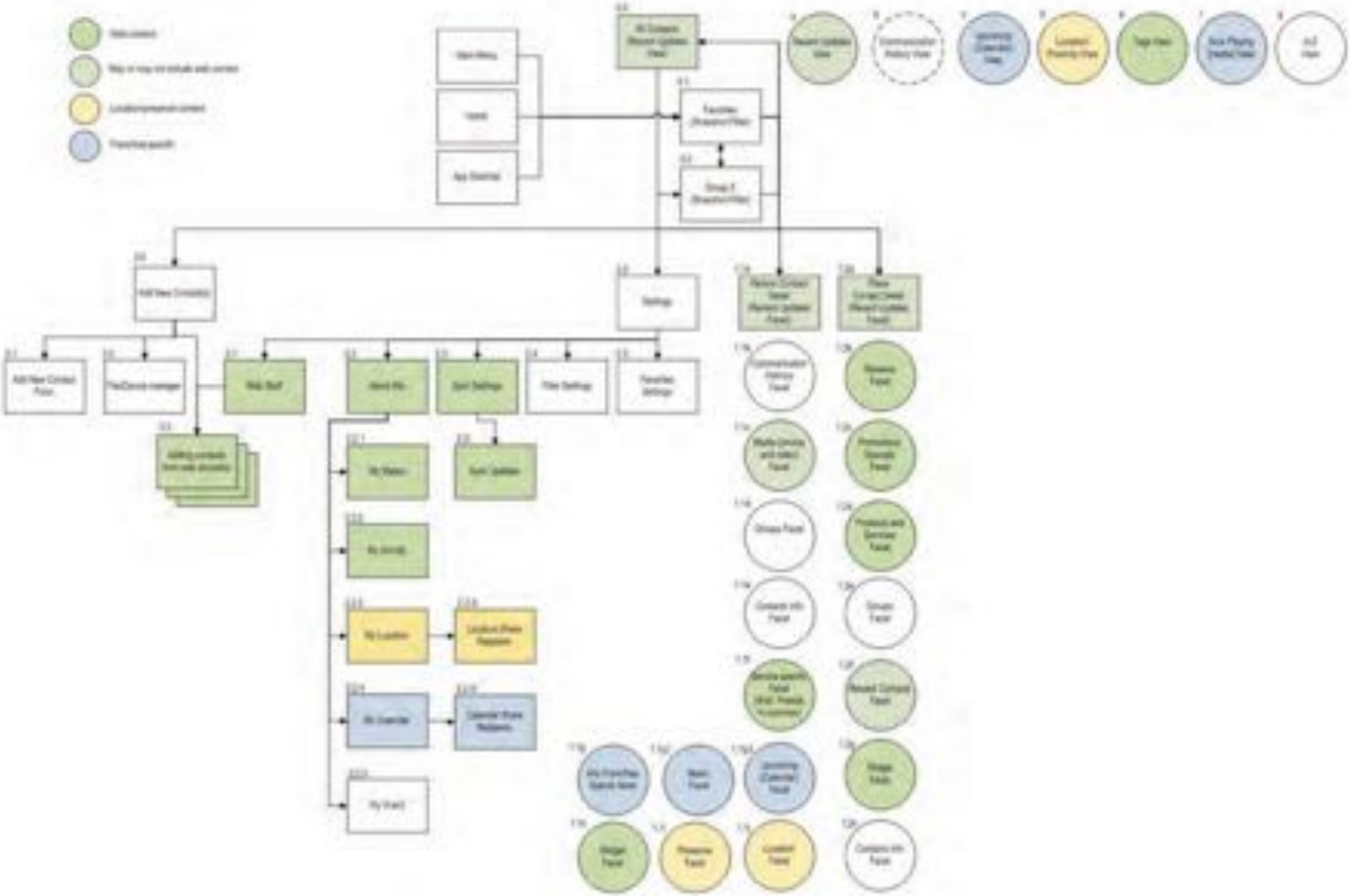


Matrix 2 dimensions (sometimes 3?)

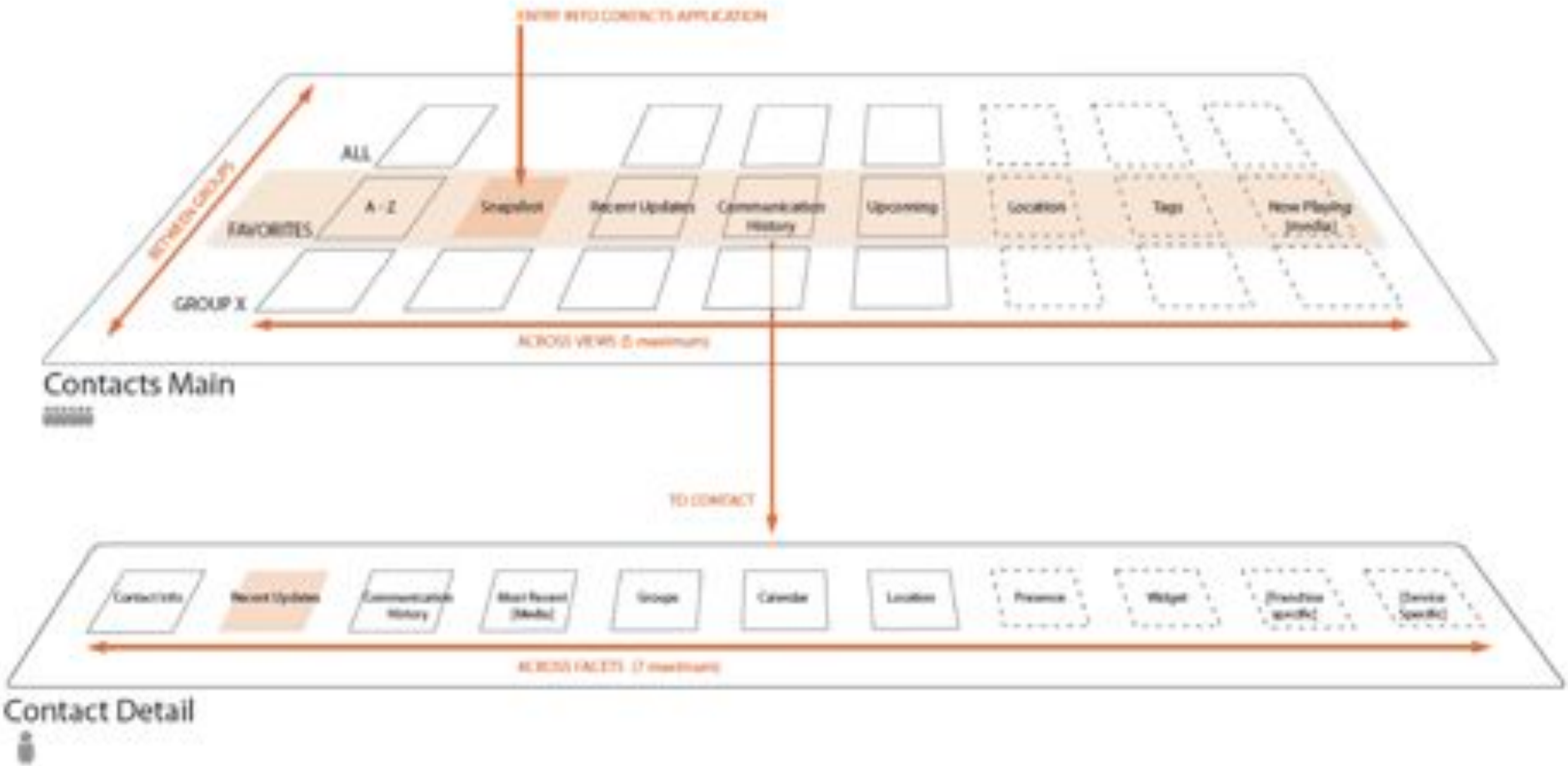
Organic no consistent pattern

Sequential step by step

structure…flow…process: information architecture



structure…flow…process: interaction model



Lunch Break

- Please be back by 1pm

Flows

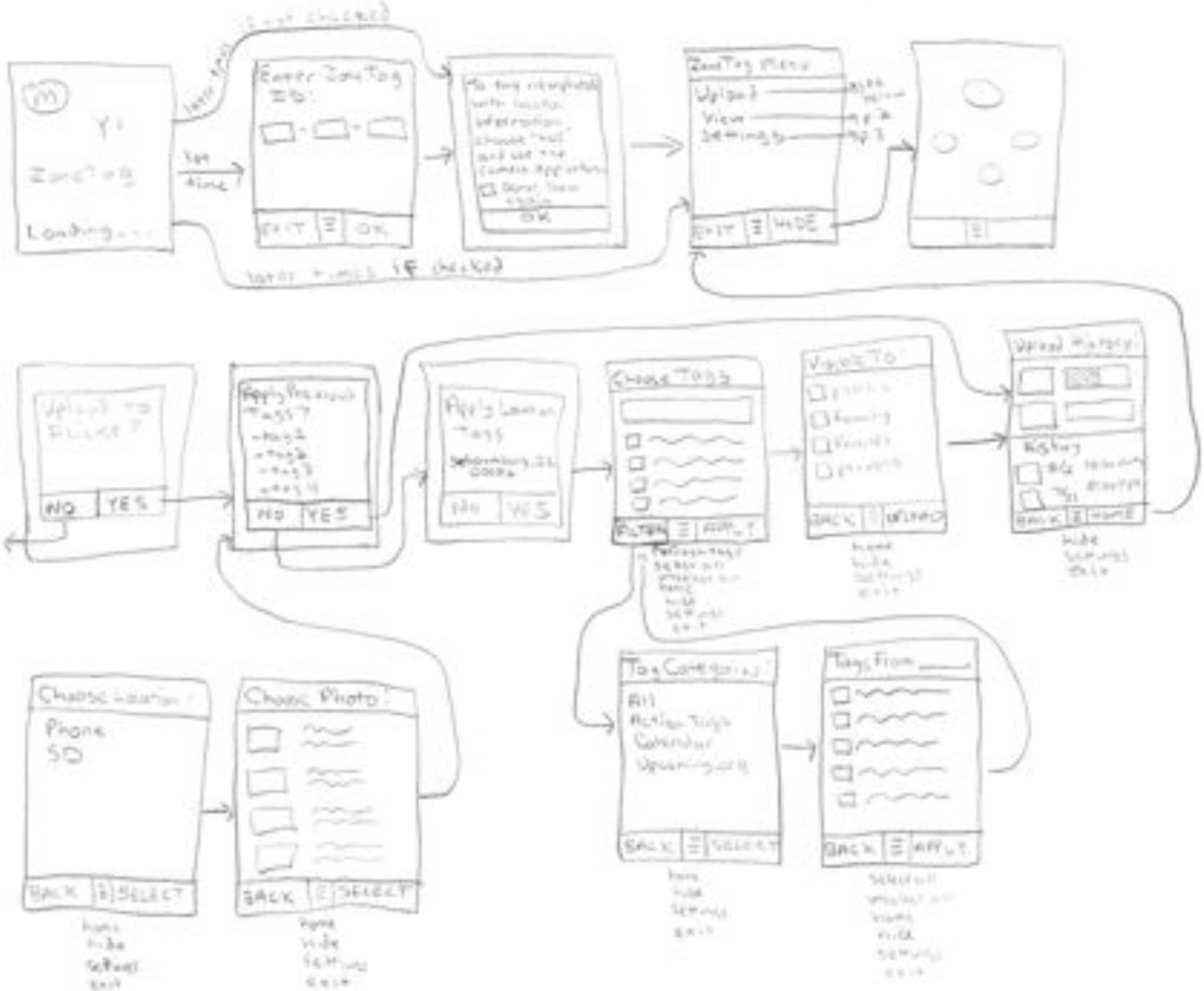
- Mapping out a flow of screens and user actions for each use case.
- Hand sketched, iterate as you go
- Make sure operations that are frequent are easy

ZoneTag UI Rev 2

open app

new photo

upload existing



affordances/interface design: interface design principles

Wodtke's 8 principles

1. Design for way-finding – where you are, where you can go, how to get there
2. Set expectations and provide feedback
3. Ergonomics design
4. Be consistent and consider standards
5. Provide error support – prevent, protect, and inform
6. Rely on recognition rather than recall
7. Provide for people of varying skill levels- intermediate is fine
8. Provide meaningful and contextual help and documentation

Blueprints (or user flows and wireframes) are just good thinking written down

Beware of easy-to-get, easy-to-remember answers.

affordances/interface design: screens

Anatomy of the Contact Detail screen (for a person - 1/6)

Back:

The next six pages will step you through the Contact Detail screen, explaining necessary widgets and interactions.

Section 1: Favorite, Block, and Status

Favorite/Block functionality

Facebook Status

affordances/interface design: screens

Anatomy of the Contact Detail screen (for a person - 2/6)

Section 2: Account Profile Images

Find Contact's (a) Account

The first photo in the row is the default profile photo for that Contact (what appears with incoming communications, such as the photo you see when Michael Ives calls).

When the first photo in the row (around the recent updates item 1) is selected is displayed.

Note: the profile image of the first person added to the Contact automatically becomes the default image for that Contact's.

The default photo can be changed via the Contact information (send it) on the media (send it) within the Contact Detail. This may seem redundant, but the reason for both is to keep what's familiar while providing as often some that can make more sense.

Account Profile

Users can scroll horizontally through multiple accounts if more than two accounts are linked to the contact.

When pressed, the hook for that service becomes visible (i.e. when the Facebook Profile Photo is pressed, the Facebook Feed (1 of) appears).

Section 3: Action Dropdown

Action One Item

If a phone number is stored for the Contact, the action (the icon will be 1 of if present a call will be placed to the Contact's #).

If there is no phone # for the Contact but the user has an email address or is a friend on Facebook, the action (the icon will be 'send message').

Action Dropdown

The Action Dropdown is lightly dynamic. The first option will always be '1 of' (it's pre-stored) or 'Send Message' but the remaining options will vary based on the facet that's available.

(see the Action Dropdown (view) section of this document for the rules around what options will appear, when.

affordances/interface design: screens

Anatomy of the Contact Detail screen (for a person - 3/4)

What is a Facet?

Facets are similar to Views. They share a common interaction (swiping), and information sets (Social Updates, Location, etc.). What makes a Facet different is that it is specific to a Contact, displaying content for one person only. A Facet is also visually unique in that it is situated on the bottom half of the screen. Each Facet contains a dynamic and focused set of information about a Contact that can come from local and/or Web sources. For example, the Communication History Facet is a log of all incoming communication from a Contact. This information is largely drawn on local on-device information and relies little on the Web. The Friends In Common Facet, on the other hand, is a list of Friends that the user and a Contact share on Facebook, is completely reliant on the web and the Facebook Service.

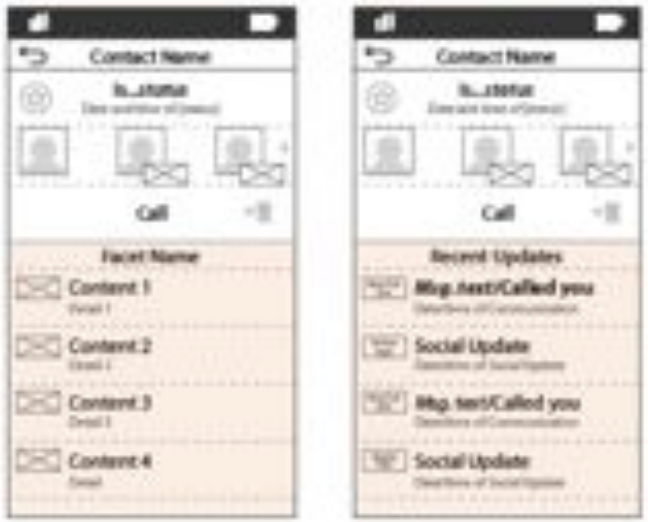
Facets are not to be confused with applications or like versions of applications. In some instances, such as the Calendar or Location Facet, the View may look similar to a device application, but it just provides a preview of information about a contact and does not replicate full functionality of the Calendar or Location application. It is possible to launch the full Calendar or Location application from those facets.

Basic Rules: 1. Facets should have a title at the top. 2. Facets are scrollable. 3. Every Contact Detail can have a maximum of 7 facets. 4. Facets are sticky. 5. Facets affect the what options are available in the Action Dropdown (see the Action Dropdown Detail for more information). Default facets (Standard Contact): Recent Updates, Communication History, Media, Groups, and Contact Information

Connected facets (Web Contact): Main (Service) Facet (1 per service), Additional (Service) Facets and Widgets

Future facets (Exemplar Contact): Calendar (Shared), Location and Presence

Section 4: Facets



Facet Basics

The following facets are covered in this design:

- Recent Updates (all web and device activity)
- Communication History
- Media
- Groups
- Calendar
- Location
- Presence
- Widgets
- Main (Service) Facet
- Additional (Service) Facet
  (i.e. Friends in common - Facebook)
- Contact Information

Recent Updates Facet

The Recent Updates facet logs all incoming Web and device activity. Updates that have not been viewed by the user are bolded. Updates that have not been viewed, it is not longer bolded.

The number of updates in this list will be determined by time or capacity.

structure…flow…process: a few web resources

First Principles of Interaction Design
http://www.asktog.com/basics/firstPrinciples.html

Views and Forms: Principles of Task Flow for Web Applications Part 1
http://www.boxesandarrows.com/view/views_and_forms_principles_of_task_flow_for_web_applications_part_1

Wizards and Guides: Principles of Task Flow for Web Applications Part 2
http://www.boxesandarrows.com/view/wizards_and_guides_principles_of_task_flow_for_web_applications_part_2

A visual vocabulary for describing information architecture and interaction design
http://www.jjg.net/ia/visvocab/

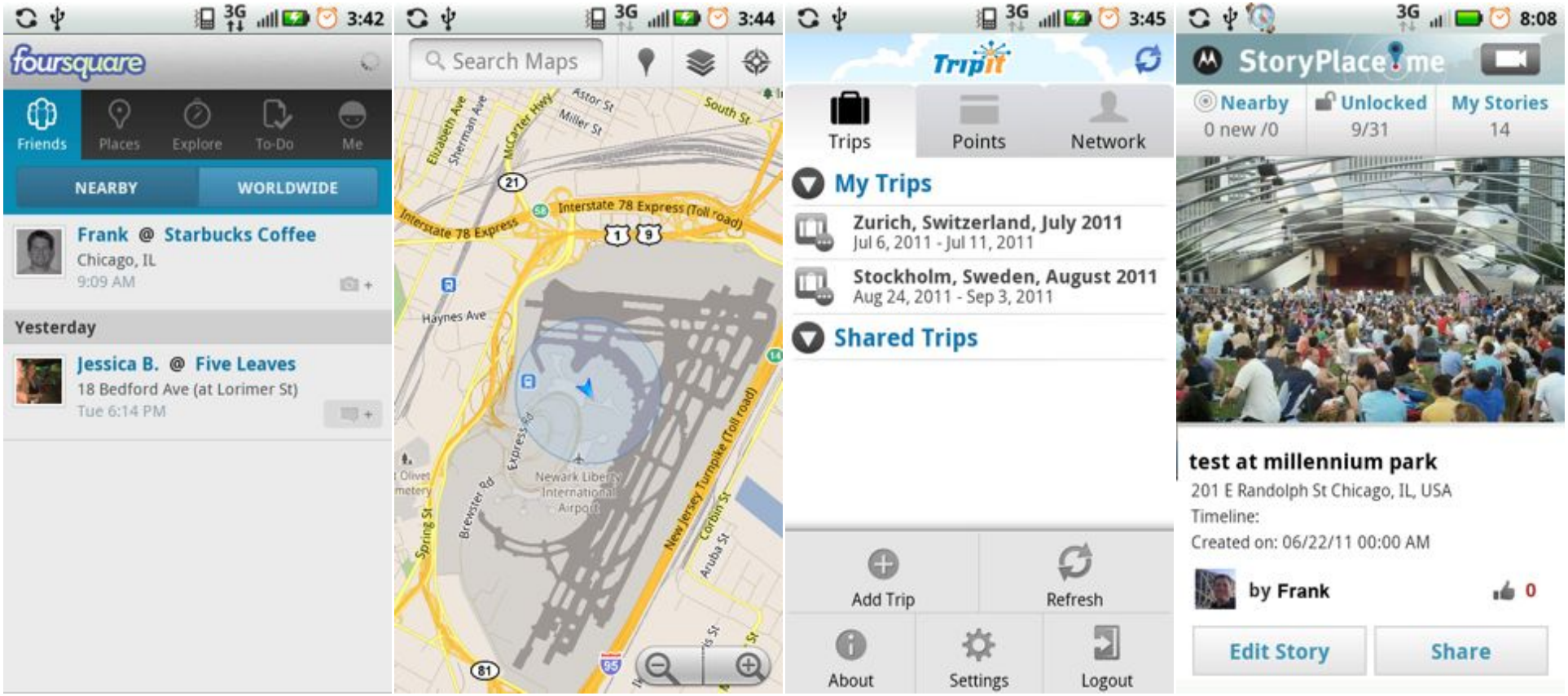
Mobile UI design considerations…

- Good default options, but easy to change
- Minimize need to scroll – multiple screens often better than one long scrolling screen
- Consistency – "Back" and Confirm actions always in the same place
- Shortcuts for advanced users – e.g. GMail's number shortcuts for delete, compose, etc.

Android UI Paradigms

- Dedicated hard-key for "back" – no need for navigation controls on screen
- Menu that can be activated on each screen with square menu items (large icon + text)
- Each screen is called an Activity, any Activity can launch any other Activity (e.g. bring up a Google Map, pre-composed email, web page, etc.)
- Tabbed interfaces common with large tabs at the top of the screen

Example Screenshots in Android



Heuristic Evaluation

- A list of common errors in user interfaces to check your interface against (sanity check)
- Simple way to evaluate interface without involving formal user study
- Can generally solve many initial usability issues
- No replacement for usability testing

Nielsen's Heuristics

- Visibility of system status
- The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
- Match between system and the real world
- The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
- User control and freedom
- Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
- Consistency and standards
- Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
- Error prevention
- Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
- Recognition rather than recall
- Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
- Flexibility and efficiency of use
- Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
- Aesthetic and minimalist design
- Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
- Help users recognize, diagnose, and recover from errors
- Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- Help and documentation
- Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Paper Prototypes

- Each screen should be a separate piece of paper
- Menus, dialog boxes cut out and overlaid on screens
- "Users" can interact with these prototypes by touching on areas where they would interact with the application.

- Very easy to iterate (it's only paper!)

Activity

- Create paper prototype for top few use cases
- Should derive from your User Environment Diagrams
- Should show enough detail on each screen for other students to navigate through interface and perform a few tasks

Usability Testing

- Best way to learn problems in interaction is to see it used
- Choose tasks that users would actually perform (don't ask someone to do something they never intend to do)
- Don't use the words in your interface to describe the task (e.g. "Filter" your contacts if the button is labelled "Filter")
- Use 5-7 users to catch majority of major flaws
- Tell user that interface is being tested, not them
- Have users "think aloud" verbalizing what is going through their heads, not reflections on what they are doing
- Don't help users (only ask them to keep talking or move to the next task upon success / failure)
- Determine ahead what constitutes a failure case, don't allow users to run amok in your UI aimlessly
- Watch for critical incidents

Usability Evaluation

- Choose 2-3 tasks (common or troublesome use cases)
- One person from your team acts as “computer” changing the screens as the “user” from a different project group points to navigate
- Another team member takes notes
- Participant “thinks aloud” while performing tasks
- Do not interrupt the participant while performing the task. Do not give away that they might be going down the wrong path.
- When the user finishes all tasks, feel free to ask them questions.

What a usability evaluation looks like…

Conduct a usability evaluation…

- One group member travels to group to your right
- Everyone else stays
- Pick two tasks for visiting "user" to accomplish
- Present these tasks to them in a neutral fashion (i.e. not using any words in your UI)
- One person places screens in place, writes in any input
- One person (or more) takes notes

- Between users, quick debrief and time for simple iterations

Coffee Break

- Please return by 3:15pm

on specs, phones are computers

- 1 GHz processor
- 512 MB RAM
- 16 GB storage
- 480 x 854 display
- WiFi, 3G, Bluetooth
- 70,000 applications

in use, they are not…

http://www.flickr.com/photos/c0t0s0d0/2371272918/


http://www.flickr.com/photos/mmewuji/2299648584/

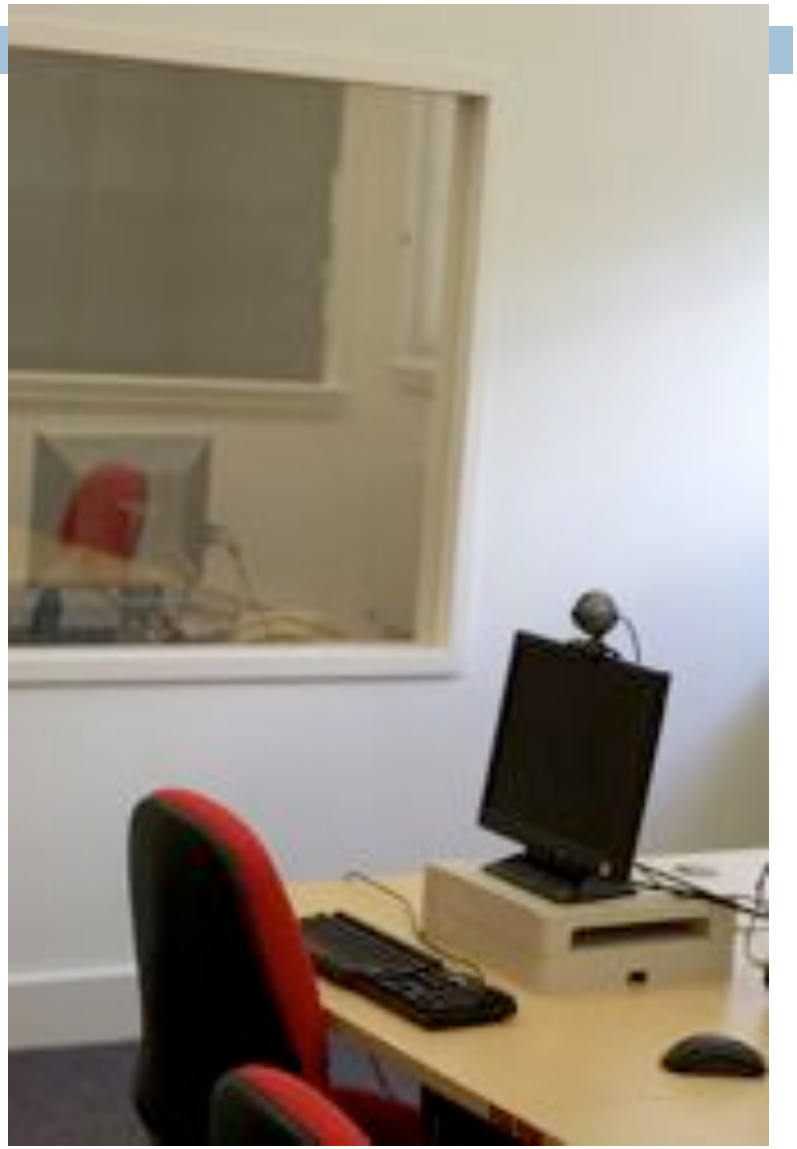

http://www.flickr.com/photos/abekleinfeld/3112840560/

- Location
- Social Connectivity
- Ubiquitous Media Capture

- Down Time
- Serendipitous Interactions
- Interaction with Environments

what the lab can tell you

- basic flow and interaction issues
- data integration issues (interfacing with contacts, web service accounts)
- understanding of prompts and icons
- etc.



http://www.flickr.com/photos/kaichanvong/2826327996/

what use in context can tell you

- How and when service will be used
- How interactions fit into daily life
- Tasks which system does not support
- Creative uses of service in the world

http://www.flickr.com/photos/nicaprio/4557139564/

http://www.flickr.com/photos/yourdon/3030246123/

Much more basic (and important!) questions!

storyboards vs. reality

www.dodgeball.com via internet archive

vs.

http://www.flickr.com/photos/francois/384850919/

new philosophy

- Get new concepts in the field as early as possible
  - Weeks after having idea!
  - Working functional prototype
- Field evaluations serve to:
  - Improve Concept
  - Mitigate Risk (kill unsuccessful concepts)
  - Build understanding of mobile interaction
- Make prototyping and field testing:
  - Cheap (incentives + your time)
  - Fast (~2 week implementation, ~3 week study)
  - Informative (interviews, home tours, diary logs, voicemails, photos, etc.)

principles for building prototypes

1. build only what you need
2. build the experience, not the technology
3. build it sturdy (enough)

Build only what you need

- The first prototype is built to answer specific research or implementation questions
- Build the minimum that you need to answer these!
  - Minimal configuration
  - Minimal graphics / custom screens
- If purpose is to test new experience, goal is to minimally prototype that experience
- If purpose is to test tech, goal should be to implement that new technology component

Build the Experience, Not the Tech

- If testing how a new experience will fit into people's lives:
  - Build something that creates that experience as simply as possible
  - Limit the development of large tech pieces until you have proven your concept
  - Mashup, don't reinvent

Build it study (enough)

- Application has to work in real world settings
  - Sturdy network code (retry as necessary, fail gracefully)
  - Sturdy database code (accept all types of character inputs)
  - Stable and responsive interface that does not (often) crash or crashes gracefully

principles for testing prototypes

1. social groups for social technologies
2. real contexts of use
3. primary device
4. field-based data collection

Social Groups for Social Tech

- If app/service requires communication among friends/family
  - Recruit social groups to participate in study
  - Find existing groups of friends and family
  - Can be difficult if they all need to have a certain phone type or carrier!

Real Contexts of Use

- Have participants use system in real settings
  - Throughout daily life for most apps/services
  - For apps tied to a place, can run small trials in a particular setting:
    - Restaurant
    - Stadium
    - Etc.
  - Directly observe users of have them keep detailed notes of their interactions combined with interviews with you

Primary Device

- For systems where people are using system over a period of time:
  - Use their own phones
  - Install app
  - Interacting with two devices creates new patterns of interaction that do not correlate with real use
  - Much easier now with Android and iPhone

Field-Based Data Collection

- Don't rely on self reports
- Instrument application/server to log interactions
- Have participants create voice notes or call a voicemail system to record details of their interactions
- Conduct interviews soon after participants use application and use prompts from logs/voicemails to elicit memories of particular interactions

Evaluating results of studies

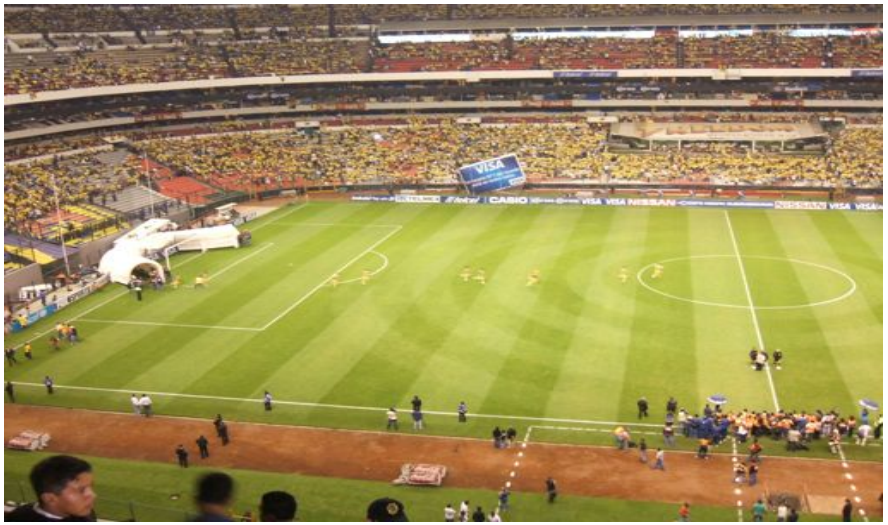
- When should you keep going with an idea?
  - Participants choose to keep your application installed on their phones
  - Participants find rewarding ways to interact with your application/service that bring them real benefit/ enjoyment
- When should you change course?
  - If users find little/no benefit to using service
  - If service does not fit into the way they live their lives

3 examples

- TuVista
  - Mobile sports video service
- Motion Presence
  - Mobile presence service to build awareness without sharing location
- Music Presence
  - Social system to receive metadata about music your friends are playing on your phone

example 1: TuVista

- How can mobile video enhance the fan experience at a live sporting event?
  - Additional Content / Replays
  - Social Interaction / Comments / User Generated Content



+

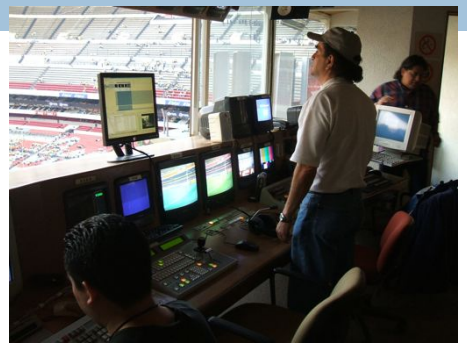
http://www.flickr.com/photos/garydenness/4614160324/

Detached and anonymous

Engaged and social

method

- Could follow traditional UCD process:
  - Interview each stakeholder
  - Co-design solutions
  - Iteratively build and test and improve

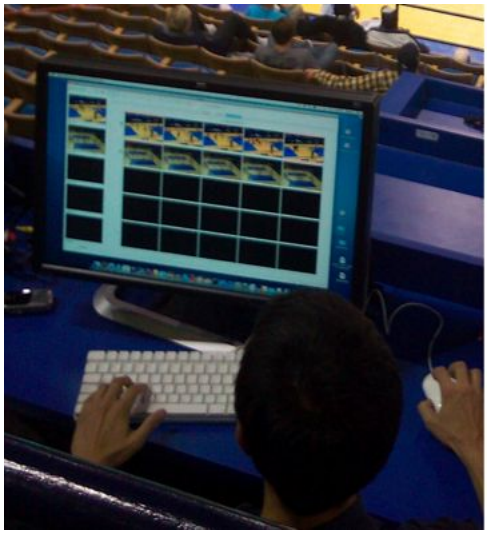


(Does anyone really know what the solution should be until we've tried?)

- Learn by doing:
  - Build initial rapid functional prototype
  - Test in real setting
  - Gather feedback
  - Learn from use + iterate

timeline

- Phase I prototype (Mexico City)
  - Implementation
  - Field Trial (Spring 2008)
- Learnings from use
  - Producing content
  - Fan experience
- Phase II commercial system
  - Used learnings to create new system
  - UCLA Trials (Spring 2009)
  - Paralympics World Cup (Spring 2009)
  - Product with Denver Broncos NFL team



build only what you need

- Off the shelf tech wherever possible (Final Cut Pro, video switchers, etc.)
- Simple website to upload content (videos, text bios, news, photos)
- Simple mobile app to view content, get live updates
- No chat, live stats integration, commenting, etc.

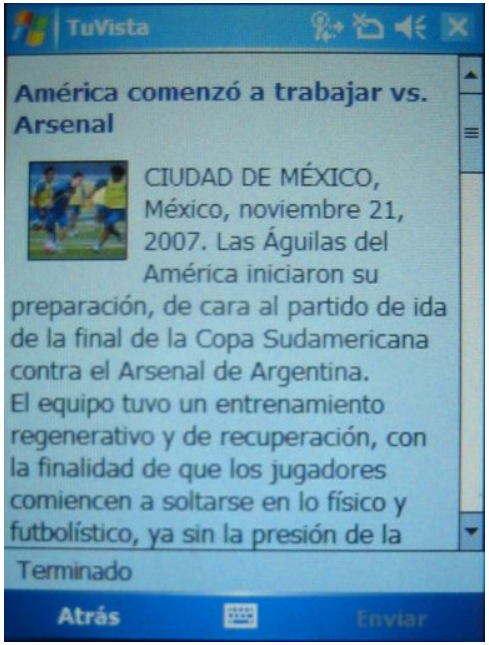
build the experience, not the tech

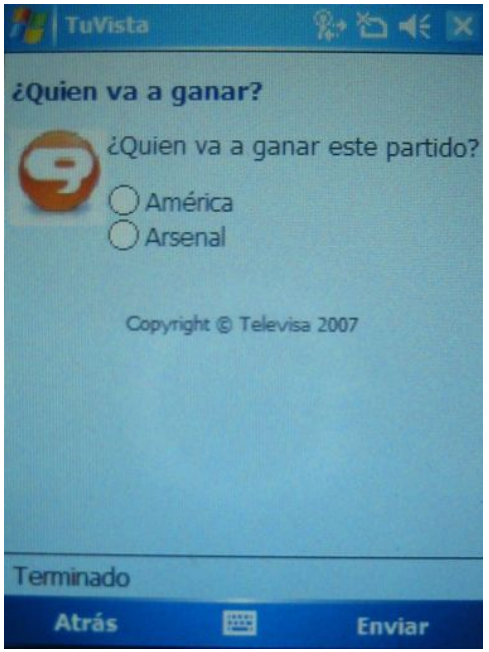
- Simple in-stadium server that could easily handle 100 users
- WiFi network in stadium, not 3G (no need to solve location-based access control issues)
- Used IP Multicast to deliver updated bundles (won't work on broader Internet)
- Video editing used existing tools (not quite fast enough but good enough)

build it sturdy (enough)

- Multicast message sent out repeatedly
- Client would periodically refresh content list
- Multiple wifi access points placed in stadium
- Used enterprise-class mobile devices (MC35)

mobile client





trial

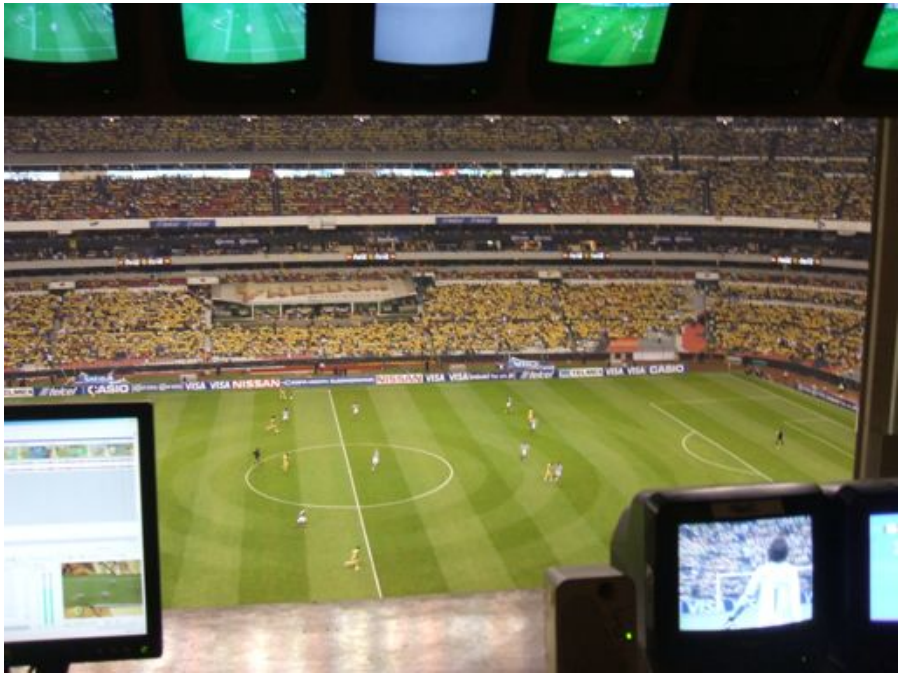
- South-American Cup semi-final at Estadio Azteca
- Researchers produced content
- 60 fans using system
  - Recruited from fans in restaurant and luxury box areas
  - Used a loaned device throughout game
  - Short interview and questionnaire after the game

social groups for social tech

- Given to groups of friends in restaurant
- Given to fans in private boxes with friends / colleagues

real context of use / primary device

- Used during semi-finals of South America Cup game at Estadio Azteca
- Was not primary device (initially thinking of renting devices as model)
- Later trials were on fans' own smartphones



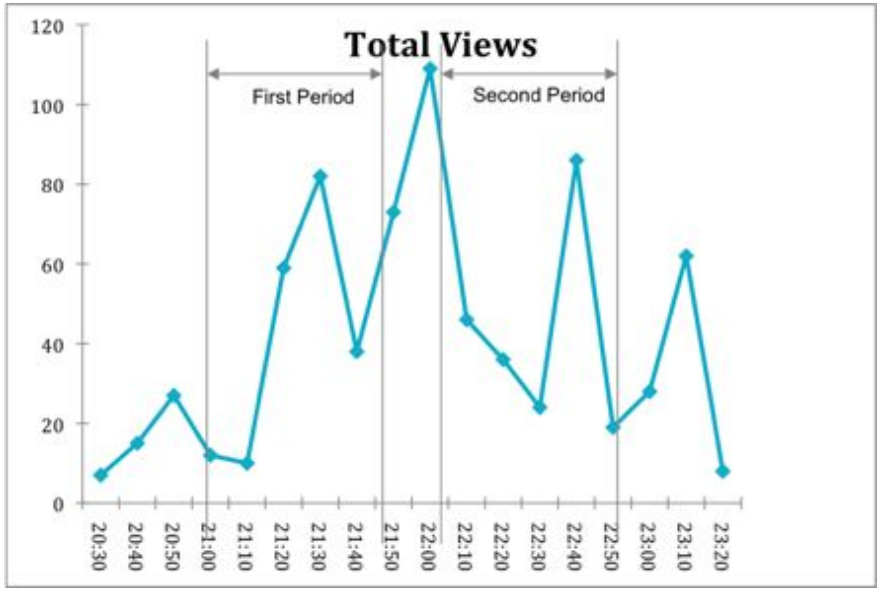
field-based data collection

- Observation during game
  - How and when use
  - With whom
- Interviews post-game
- Survey during game as a bundle
- Access logs from server for all user interaction with application

producing and consuming content

- Linear editing created bottleneck
- Views centered around breaks in the game, right

Time of Action	Editing Time (mins)
21:11	8
21:16	8
21:33	7
21:37	9
21:39	17
22:16	8
22:18	14
22:20	17
22:26	23
22:34	18
22:55	36

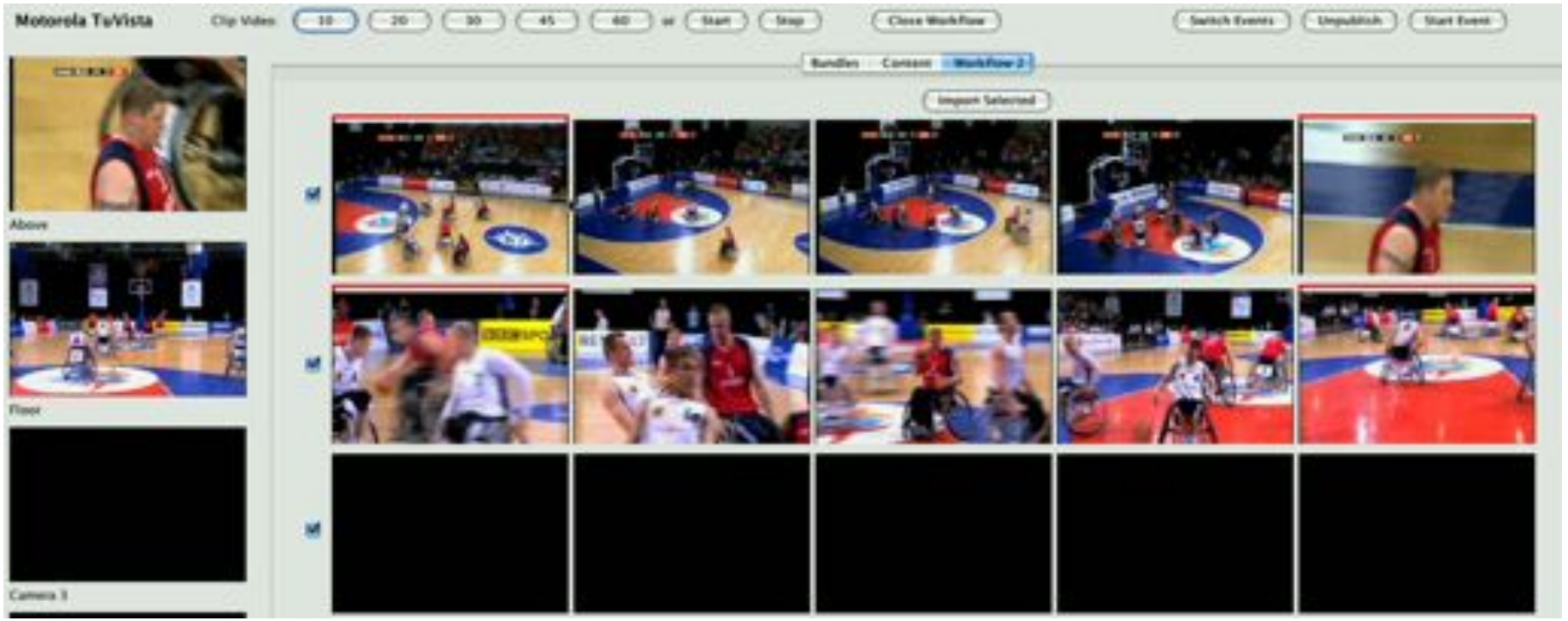


findings

- Video had to be produced much faster
- Multiple camera angles desired
- Commentary important
- Use was social in-person
- Most participants used other devices to capture photos/ video during game
- Overall concept very well received – move forward with feedback

TuVista Phase II

- Multiple camera angles
- Fast (< 30 second) video production flow
- Can mix in any audio source with video



TuVista Phase II

- Multiple views into content
  - Timeline
  - Player
  - Video

example 2: Motion Presence

- How can we get benefits of sharing location without sharing location data?
- How will people use location-derived presence information in daily life?
- What are the major privacy concerns of sharing location-derived presence information?
- How accurate is motion information derived from cell tower ID changes?

first prototype

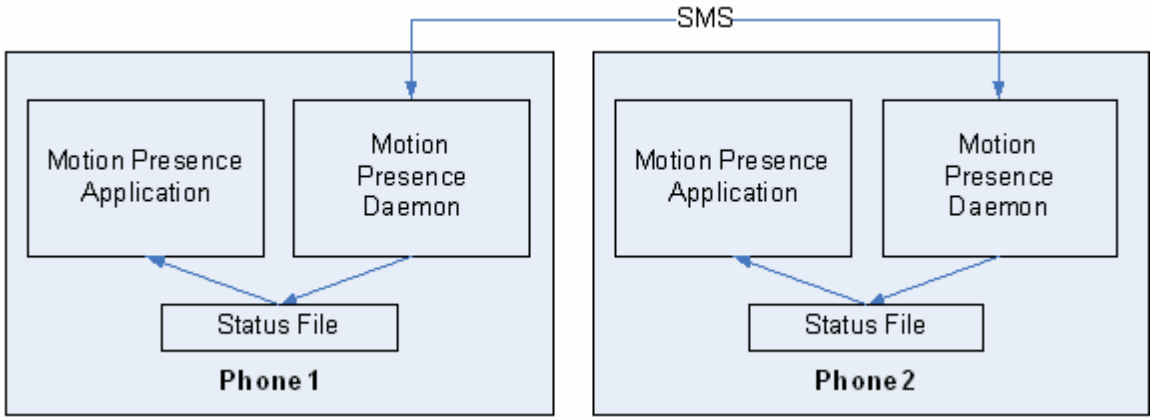
- Two screens
- Contact data read from file / not (user) editable

My Status

Friends Status

build only what you need

- No way for users to add/modify contacts
- No server (all P2P)
- No fancy graphics or images
- Just two screens

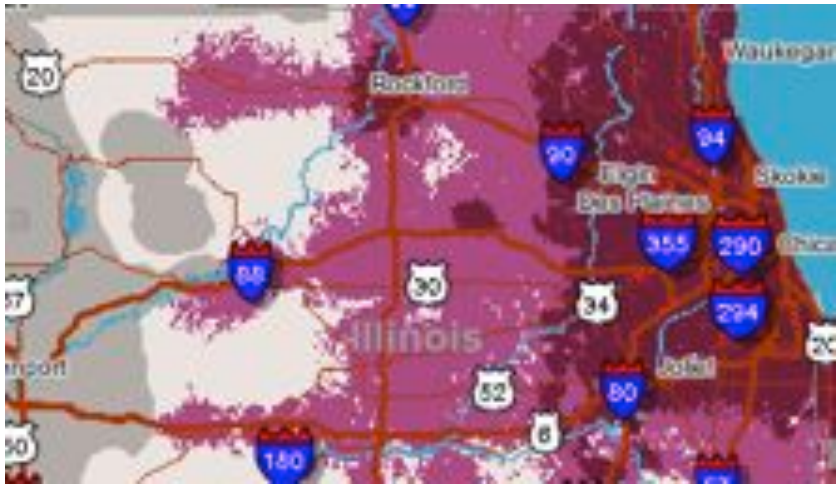


build the experience, not the tech

- Used SMS to send changes in motion state
  - Each user changed motion state ~15x/day
  - In a group of 4, this is 240 messages/day or 3,360/trial
- Phonebook app reads from a file, not from actual contact database
- Motion algorithm required processor to always be awake (and thus a LARGE battery)

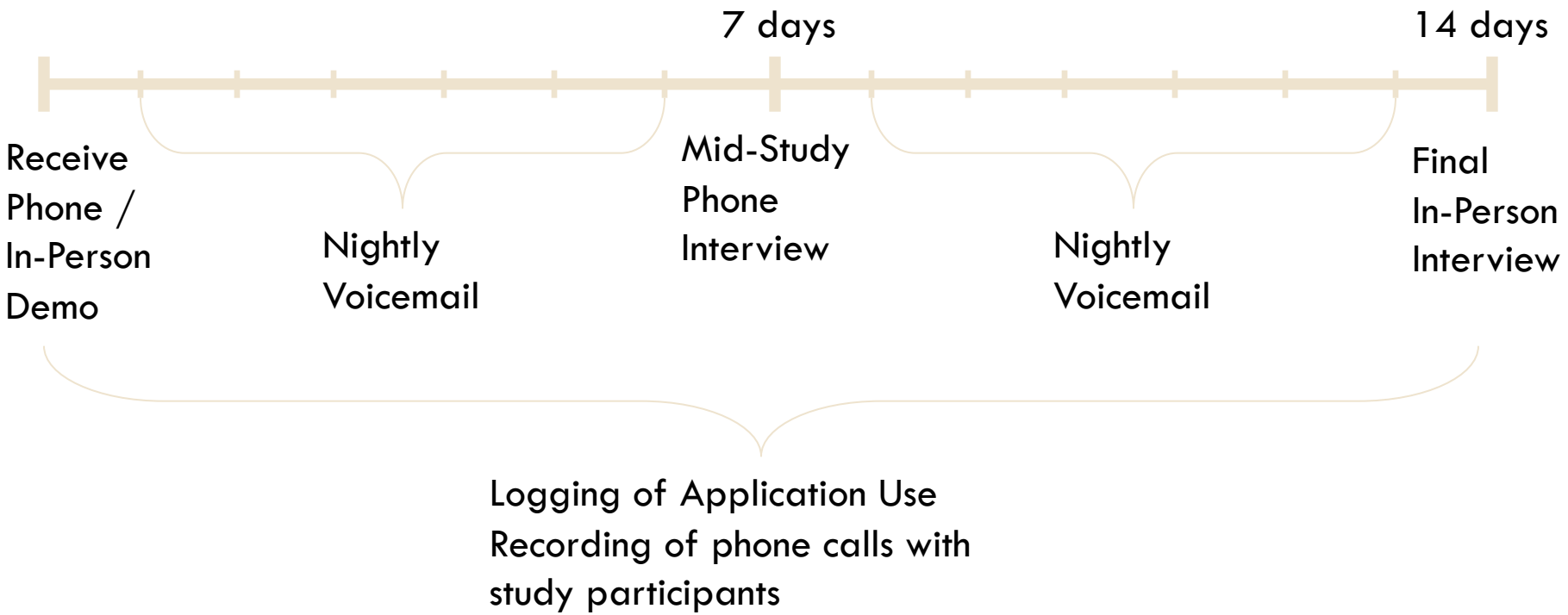
build it sturdy (enough)

- Motion background process had to handle going in and out of coverage (subway, elevators, etc.)
- Motion algorithm had to be right at least most of the time (few days of trial and error tweaking parameters)
  - Only a few reported cases of errors – large warehouse, stuck in really bad traffic
- Had to secure large battery so that it did not fall out (tape!)



coverage map from tmobile.com

methods



- Social Groups (3 couples, 1 group of 4 friends)
- Primary device for two weeks
- Used in daily life

social groups for social tech

Pseudonym	Age	Occupation	Gender
Alejandro	34	Mechanic	M
Beatriz	38	Admin. Assistant	F
Chris	40	Fundraiser	M
Dana	40	Interior Design	F
Ebony	46	Transportation Mgr	F
Farisa	47	HR Manager	F
George	23	Server	M
Harold	22	Loan Officer	M
Ian	23	Warehouse Profiler	M
James	23	Credit Analyst	M

real context of use / primary device

- Used in their lives for 14 days
- Transferred contacts and SIM card to new phone for duration of study
- Used at work, home, vacation, business trips, parties, out in the city

field-based data collection

- All interactions with application were logged
  - Open/Close
  - Click on a contact
  - Initiate phone call / text message
- All changes in motion state logged for all participants
- Participants recorded phone calls with study participants
- Participants left nightly voicemails about their use

learnings

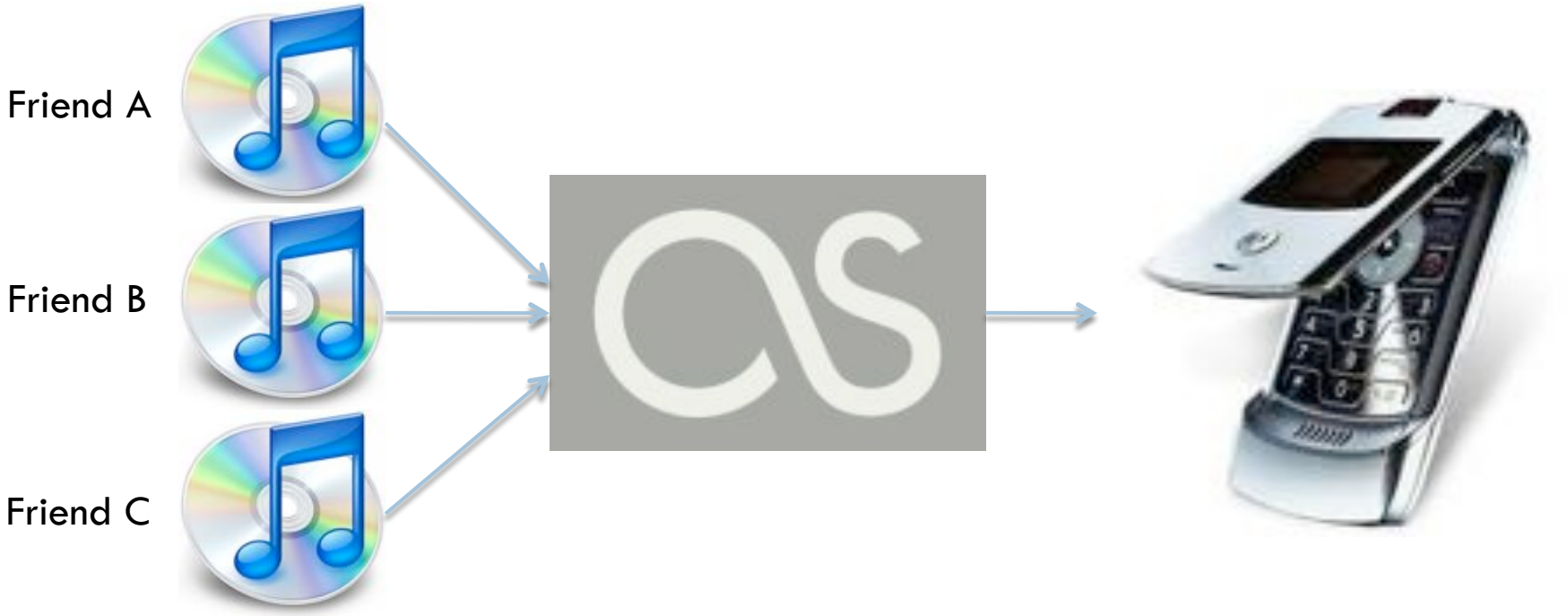
- Participants used motion information in unexpected ways:
  - Get more time at current activity
  - Reach a destination at the same time
  - Know if someone was busy to take a phone call
  - Know that someone was safe at work
  - See if a spouse had stopped at the grocery on the way home
  - Feel connected to lives of close friends and family
- Used multiple times a day and not just when about to call someone
- Algorithm worked!
- Power of mobile context in phone book

MotoBLUR

- presence in augmented phonebook

example 3: Music Presence

- How would people use knowledge of what friends are listening to if received on their mobile devices?



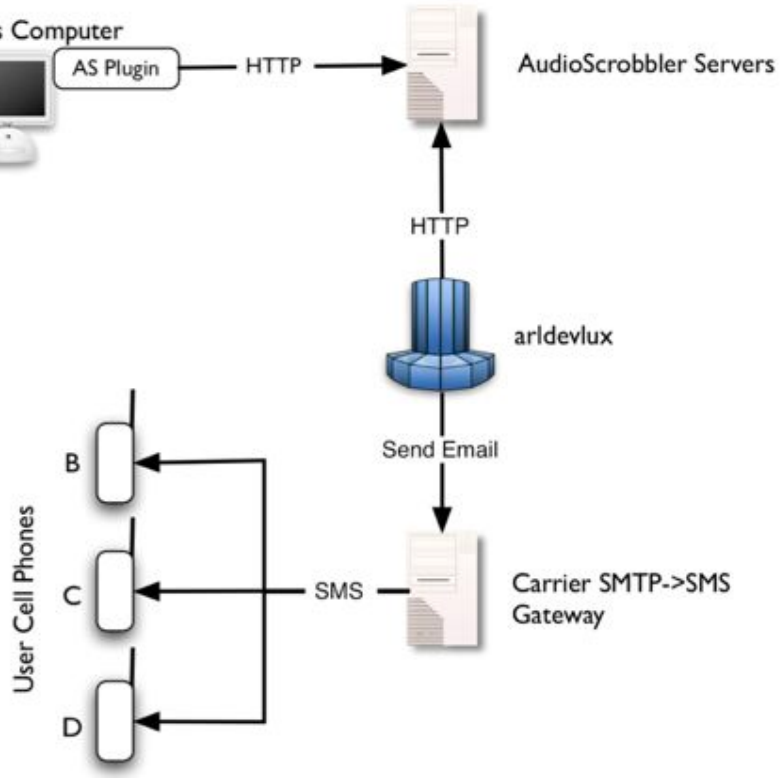
first prototype

build only what you need

- Nothing implemented on the phone
  - Used existing SMS inbox
  - Used existing SMS notifications (could apply different ringtone to messages from automated sender)
- Used existing Audioscrobbler (later last.fm) service
- few hours of implementation

build the experience, not the tech

- SMS sent to entire friend group every time you play a song (300 songs played in 5 days = 900 text messages)
- Server polled audioscrobbler APIs every 3 minutes to see if new music was played



build it sturdy (enough)

- Used existing SMS infrastructure for messaging (no need to rely on unreliable persistent connections)
- Used audioscrobbler service which was (mostly) reliable

social groups for social tech

Abigail

- Backstreet Boys
- Michelle Branch
- Blessid Union of Souls

Bianca

- Beck
- The Sea and Cake
- The Arcade Fire

Caroline

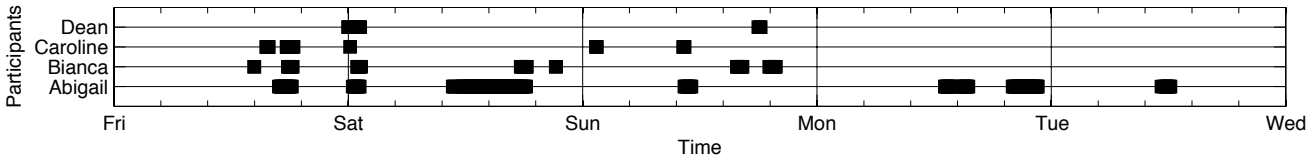
- The Beatles
- Dispatch
- Grateful Dead

Dean

- Aloha
- Saturday Looks Good To Me
- The Silver Mount Zion Orchestra & Tra La La Band

real context of use / primary device

- Used for music played on their personal computers
- Data about friends' music sent to their regular cell phone



field-based data collection

- Nightly voicemails
- In-home interviews at beginning and end
- Logs of all music played

Dean: "So when I see what Abigail listens to I think about junior high and how she's always so upbeat most of the time and she dances and does all this fun stuff."

Bianca: "[I'm] bored because nobody could go out and do things this weekend, so there's nothing for me to do now. But maybe if someone is listening to music I'll know they're home. … I was thinking if maybe they played music, I could call them because I know they'd be home."

learnings

- need for lightweight communication
  - thumbs up
  - thumbs down
  - exclamation point
- lightweight communication can escalate into richer communication types
- presence and mood can be inferred from music choice
  - home + playing music on friday night = bored, available
- rich communication through music choice
  - played songs with her name in them
  - played songs given to her by a friend

second iteration

- J2ME app
- Created in 2 weeks
- Consolidated presence info
- Lightweight communication added (thumbs up, down, !)
- Used HTTP polling for content delivery

stop and learn from the world

- Field studies are the best way to learn about use in context
- You don't have to wait for a final system to take something in the field
- The earlier you get data, the easier it is to change course

http://www.flickr.com/photos/jasonbachman/4177519542/

reflections

- Use of social / contextual systems in the world is often different than you would imagine
- Findings from a quick field study can inspire powerful, new concepts
- You can learn a lot from a prototype that does a little

guidelines

building

1. build only what you need
2. build the experience, not the technology
3. build it sturdy (enough)

testing

1. social groups for social technologies
2. real contexts of use
3. primary device
4. field-based data collection

Thanks to collaborators:
TuVista – Mike Groble
Motion Presence – Crysta Metcalf
Music Presence – Drew Harry

Questions on methods…

- Any questions on systems, methods, …

What to build?

- What are the core research questions you have right now?
- What can you build to answer them?
- What can be hard-coded/faked out for now?

How to test it?

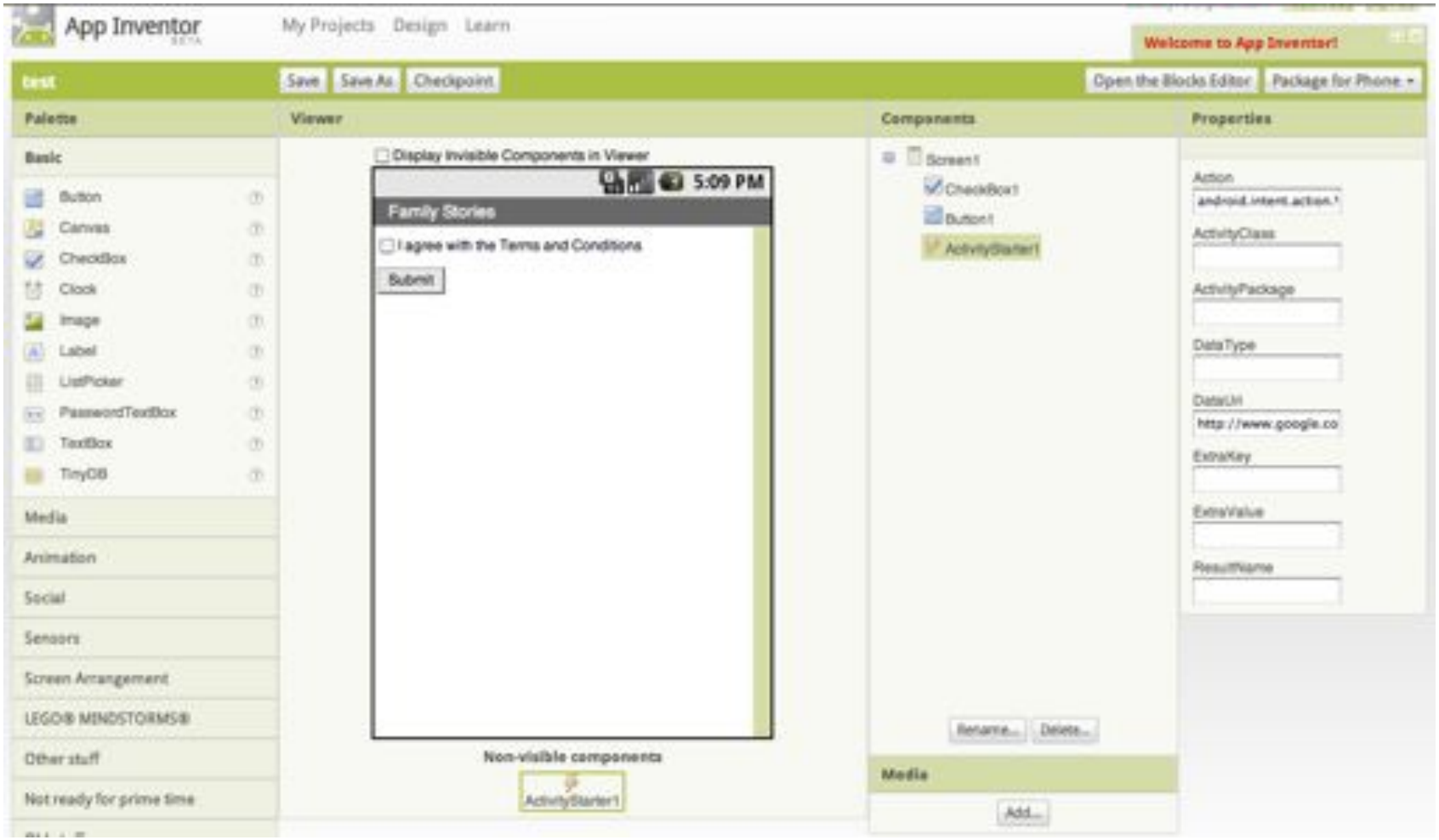
- Who would you want to use it?
- For how long? Where?
- How would you capture data?

Google App Inventor

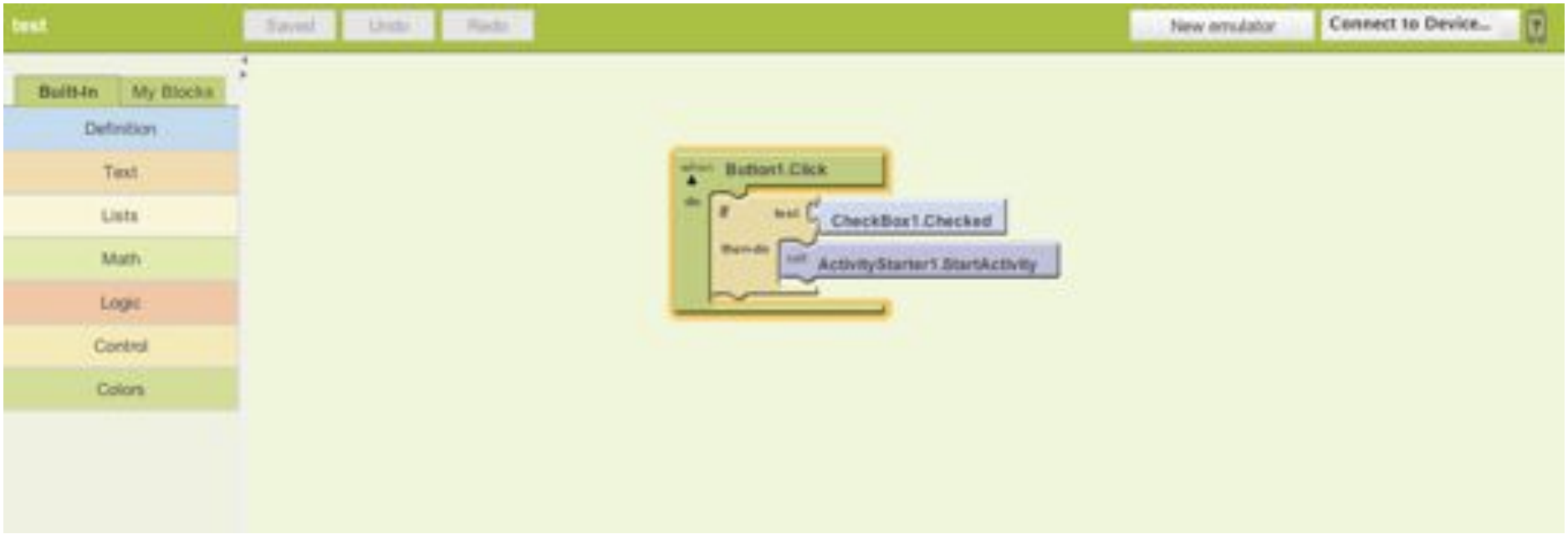
- Quick way to prototype new concepts
- Get started at: http://appinventor.googlelabs.com/learn/setup/

- In remaining time we have today, try to make at least one functional screen, load it onto your phone (or run in the emulator)

Laying out Components



Adding Logic



What App Inventor is good for…

- Quickly laying out a screen to see how components will fit together on actual device
- Simple usability evaluations (a higher fidelity paper prototype, with some actual working functionality)
- Making simple 1-2 screen applications

Thanks!

- Hope you learned some things today
- Touched on many different topics, full classes offered in many of these

- Feel free to follow up with any questions:

f.bentley [at] motorola [dot] com