Context-Aware Computing

IAT351

Week 10 Lecture 1

5.11.2012

Lyn Bartram

lyn@sfu.ca

SFU

SCHOOL OF INTERACTIVE ARTS + TECHNOLOGY

Administrivia

- Assignment 4 details
- Mahshid has put sample working code and instructions up
- Assignment 4 25% catchup
- Final projects are in review, looking good

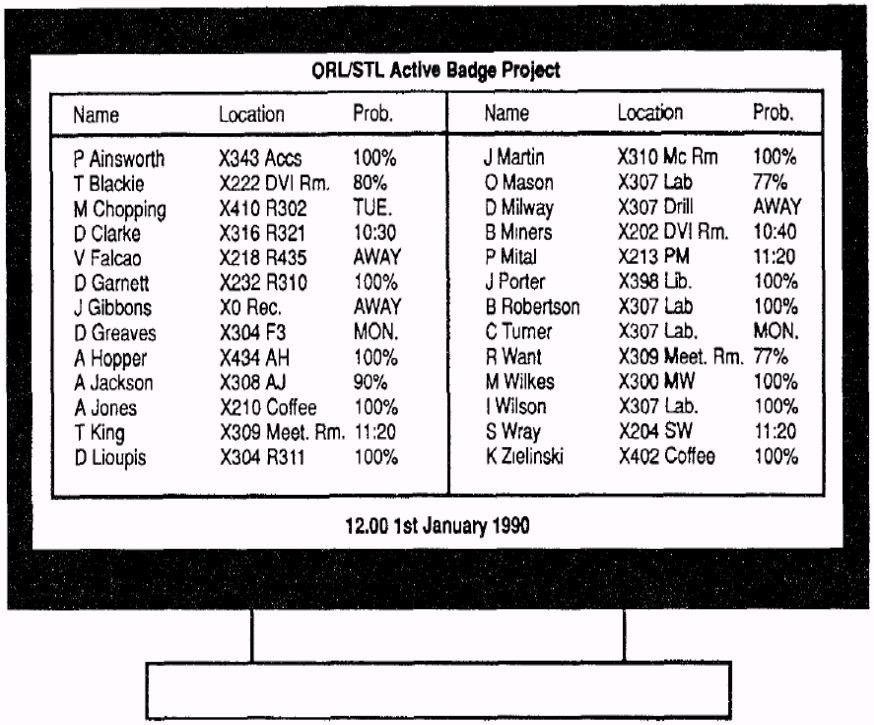
- Final presentation schedules will be randomly assigned

Context-Aware Computing

- Context: situational elements relevant to interaction between user, application, environment
- Context-awareness: situationally appropriate; apps adapting to context, increasing value to users
  - Using sensors and actuators to improve human-computer interaction and (computer-mediated) human-human and human-environment interaction
- Examples: tour guide, reminders, diary retrieval

Olivetti Active Badges

- Problem: **locating** researchers
- Solution: badge tied to identity, tracked as researcher moves in building



Assistant sees this view

- knows where researcher is
- can forward call

[Want and Hopper, 1992]

Active Badges

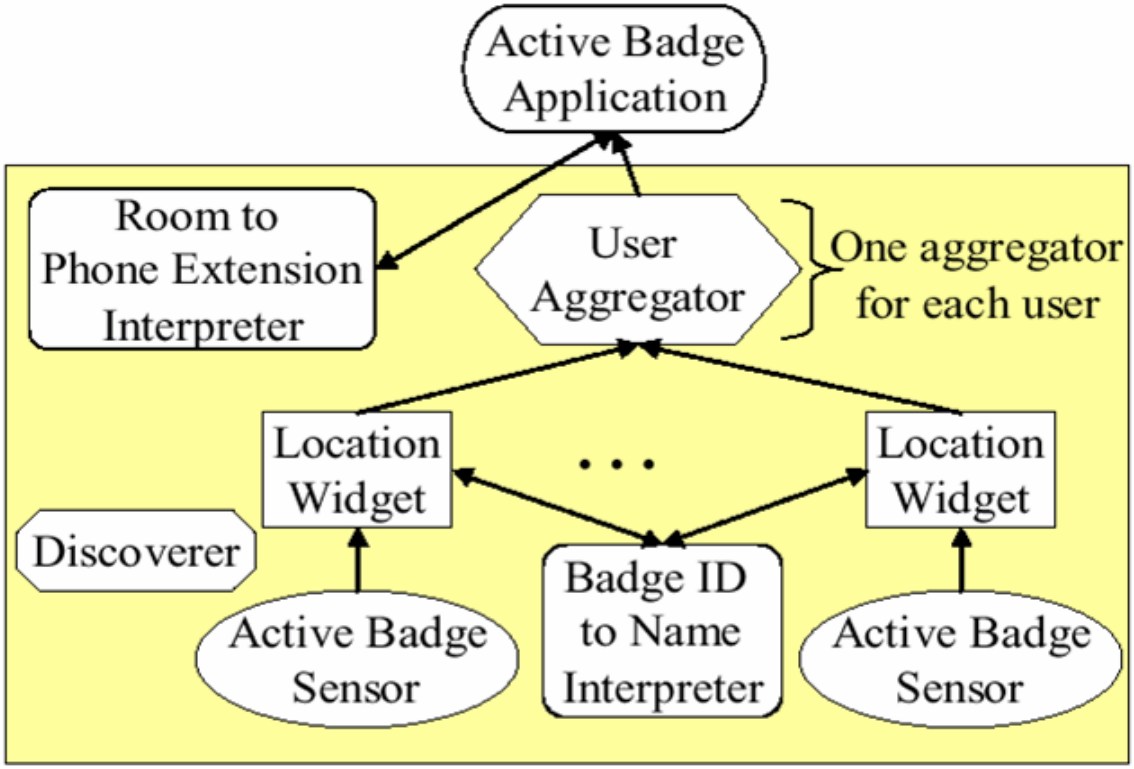


Figure 2. Architecture diagram for the Active Badge call-forwarding application.

People Issues

- Avoiding embarrassing situations
  - Active Badges + bathrooms
  - Inconvenient phone forwarding
- Avoiding dangerous situations
  - Need to take into consideration cost of mistake
  - Smoke alarms when cooking
  - Lights that turn off when you're still there
  - Woman locked in "smart toilet stall"
- Will adding more context really help here?

People Issues

- Making it predictable and understandable
  - Setting preferences
  - "I want my cell phone to ring except in theaters and when I'm in a meeting unless…"
  - Why the heck did it do that?
- Privacy
  - What does the computer know about me? What do others know about me?
  - Capturing/collecting lots of information about people, places and devices
  - People uncomfortable when don't know what is being collected and how it's used

What's the Context

Shop in Indonesia?
Buying something?

Being a tourist?
Getting a good bargain?
Making a video?

Operational Definition of Context

"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves." [Dey and Abowd, 2000]

- Emerged from point of view of developer/computer scientist
- Determined in absence of all but basic infrastructure
  - Networking but no services

Context-Aware Infrastructures

- Context Toolkit: privacy, uncertainty, end-user support (2001)
- Java Context-Awareness Framework: quality of context and context modeling (2009)

- PersonisAD: context models of entities and links between them, support for *scrutability (2002)* – user can examine

- Nexus: context modeling, representation, and spatiotemporal processing
- Context Fabric: privacy, and modeling of context entities
- Context Spaces: context modeling, inferencing of security and probabilistic reasoning

Dey (CMU) – 10 years of context-aware computing research

- Driven by a single problem
- Reminder to buy milk
- When to deliver: not time/location specific
- How to deliver: appropriate modality

We can build it now but none of you would use it

Context Toolkit

- 1997: difficult to build simple context-aware applications
  - No abstractions for acquiring and using context from sensors or controlling actuators
  - Context coming from a number of distributed sources
  - No principles for designing applications
- Aggregated sources and distributed service deliveries

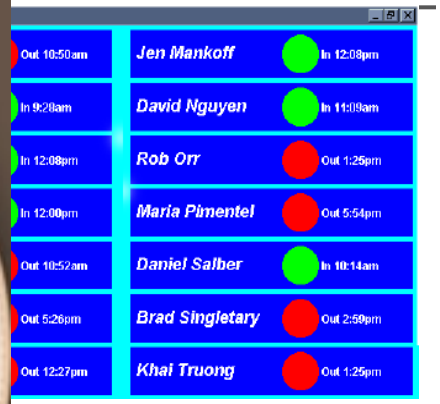
Context-Aware Applications

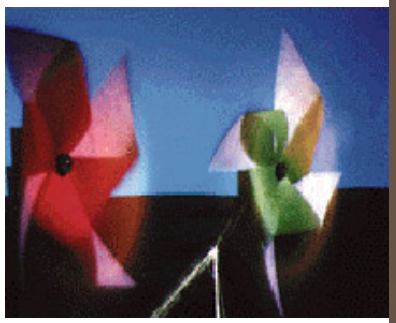
- What is a context-aware application?
  - App that uses context to perform some behavior/service for its user(s)
- "A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task."

Context-Aware Features

1. Presentation of information and services
   - Tour guide, Active Badges
2. Automatic execution of services
   - Smart homes (turn off lights, adjust temperature)
3. Tagging of context to information for later retrieval
   - Digital camera meta-data (time, location)

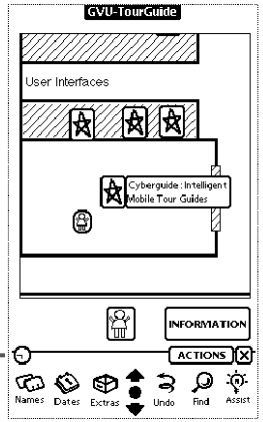
Directly displaying context

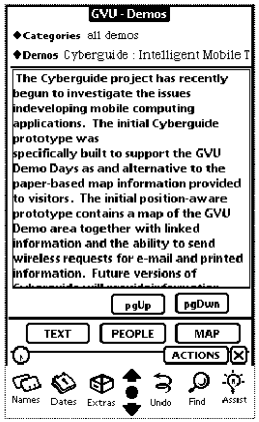




Execute/Adapt Services

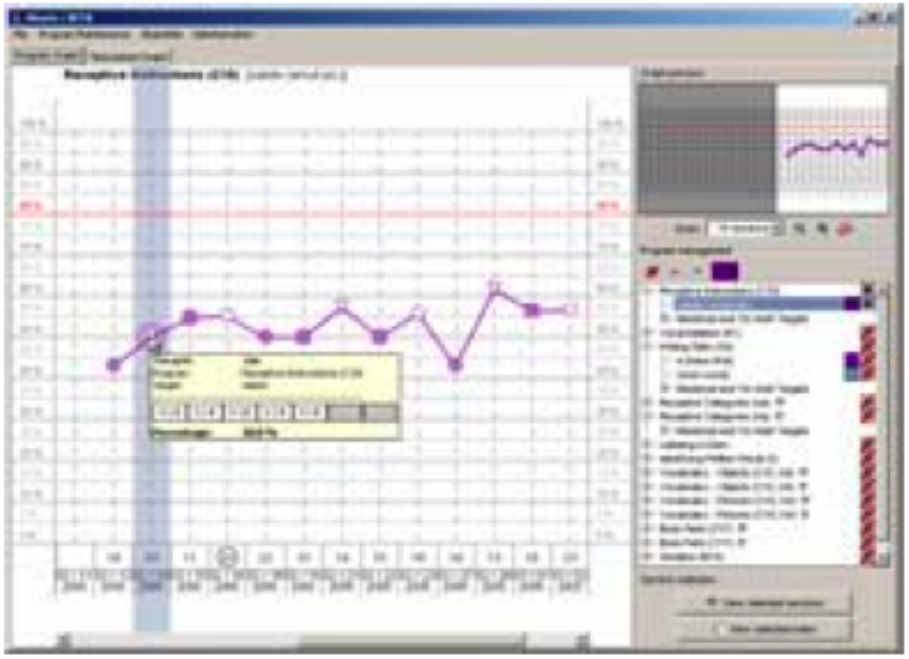
- Select and perform a relevant service from multiple services
- Change how a service behaves or is executed





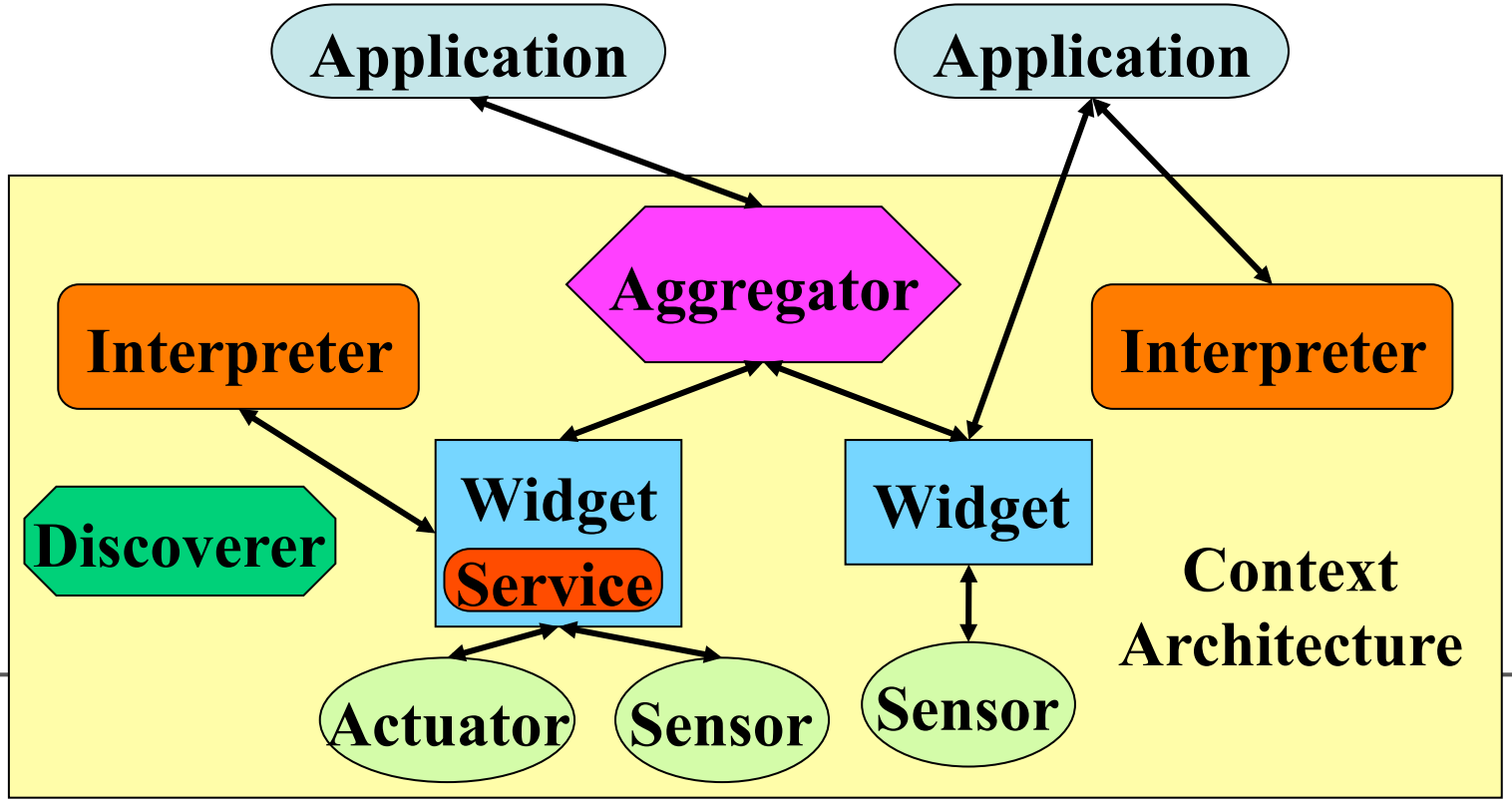
Tag Information

- Use context to aid in context-based retrieval



Thesis: Context Toolkit

- Context Toolkit to support programmers in building context-aware applications more easily



Context Toolkit Infrastructure

- Context Widgets – Basic building block, provides abstraction of sensor details.
- Context Servers – Abstraction to provide aggregation of context data from context widgets
- Context Interpreter – Takes input from context widgets and outputs it in a way that a set of applications can use.

Context Widget

- Abstracts sensor details.
- Provides a common paradigm for context-aware applications.
- Allows reuse of code and persistence.
- Tailors sensor information to the needs of the application.
- Have attributes and behaviors.

Aggregators

- Widgets + ability to aggregate context info
- Collect sensory info about an entity (person, place, thing, *etc*.) from multiple sources into one widget
- Hide more complexity about context-sensing mechanisms by combining multiple sensors
- Enable maintainability and efficiency

How are they used?

Applications use context widgets as follows:

- Applications subscribe to widgets that they need data from
- a widget could be local or remote.
- Widgets perform callbacks to the application when something interesting happens, passing data back to the application in a form that the application requires.
  - “callbacks” = event-listener model

So if the technology is here….are we ready?

- Just making it easier to build applications is only part of the problem
  - Once you know what context to capture and how to use it, easier to execute
  - But how do you figure this out?

- But, we still don't have widespread deployment
- Lots of problems to be addressed on the user side
- Commercially available apps focusing on location-based services – with two notable exceptions…(later)

What are the Impediments?

- Cost and difficulty in deploying infrastructure
- Applications are of limited value
  - Not modeling the right tasks
- Applications lacking in usability
  - Apps lacking in support for end-user control
  - Apps lacking in intelligibility

Usability is Key

- Dourish, Abowd and Mynatt, Bellotti and others: lack of control in these environments

- Information collected, synthesized and used implicitly

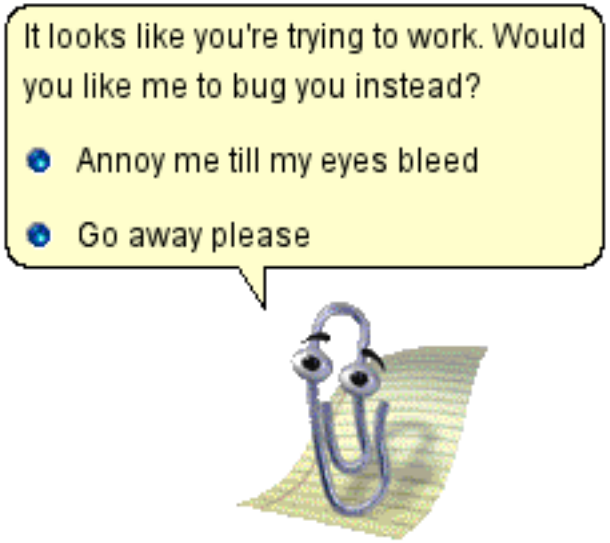
- How do I know what's going on? (intelligibility)
- How do I change what's going on? (control)
- Who gets this information? (privacy)
- Is this another way to SPAM me? (overload)

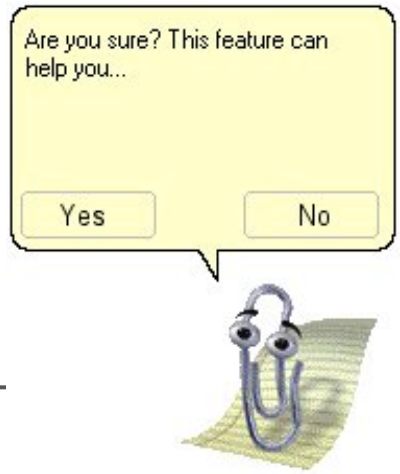
Dey: Key principles

- End-User Intelligibility and Control: decide what actions an environment should take and understand environment state; help the user feel in control
  - CHI 04, Pervasive 06, Ubicomp 06, CHI 07, Ubicomp 07
- Privacy: user decides what to release and when
- Calmness: use peripheral displays and unintrusive modalities to present information without overwhelming
  - CHI 03, CHI 04, Pervasive 04, UIST 04, UIST 05, CHI 06

Challenges

- Leveraging real human context a you’ re not really getting at huma
  - Clippy

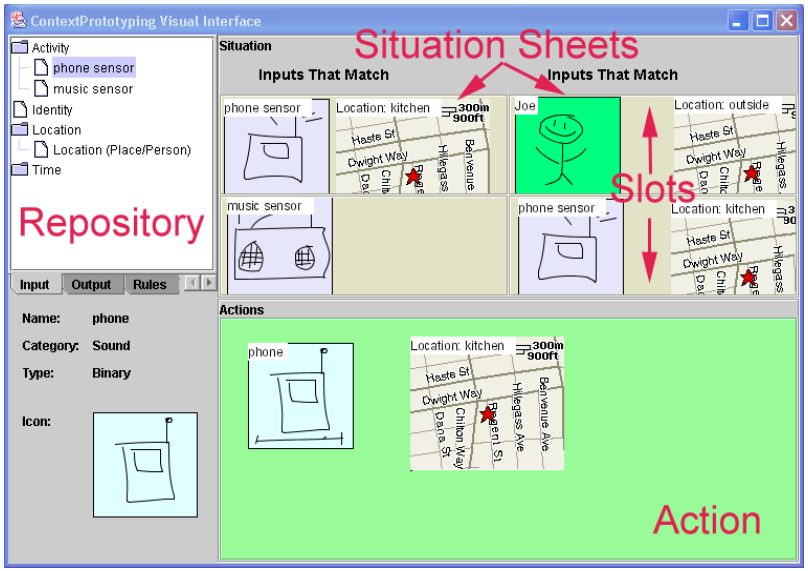


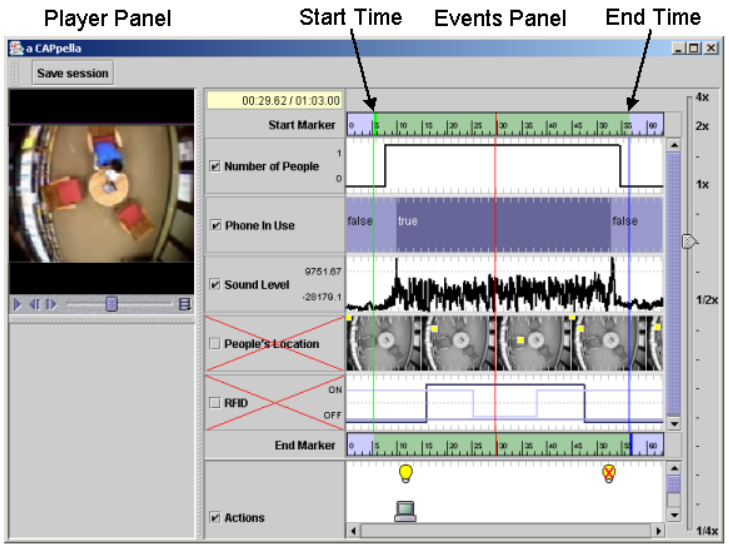


End User Control and Intelligibility

- Control: real situations are dynamic and cannot always be planned for *a priori*
  - Need to allow constant adaptation
    - Home: new people, new organization, new roles
    - Programmer can't possibly hope to cover these situations
    - Must support end-users in maintaining control
    - Need support for this

Control of Smart Homes





End User Control and Intelligibility

- Intelligibility: users will reject systems they do not understand, particularly when these systems make mistakes
  - Amazon recommender, Clippy
  - Complex systems need to describe their behavior to some degree

Discussion

- If you were designing an application and you wanted to take advantage of context, would this framework be helpful?
- Example: cell-phone restaurant locator
- What are the entities?
- Relevant Characteristics (context): ?
- How Does this help the designer and user?

Implications of Representable Context

- Context is:
  - Form of information that can be encoded
  - Delineable: in advance define what contexts are relevant for the application
  - Stable: determination of relevance of potential context in an activity can be made once, reused
  - Separable from *activity*

Dourish's Context

- Previous approaches to context are **representational**: "what is context and how can it be encoded?"
- Alternative approach uses **interactional** model: "how and why, in the course of their interactions, do people achieve and maintain a mutual understanding of the context for their actions"

- Miminum set for reliable behaviour?

Context can be encoded

- Alternative:
  - You cannot bundle up all the context
  - Objects can be *contextually relevant*

- **Dey**: relevant info about entities (people, exhibit, interface, …) is context

- **Dourish**: all those things might be contextually relevant, but they do not fully describe the context

Context is Delineable

- Alternative:
  - Scope of application's contextual features is defined dynamically

- When contexts X, Y, Z come into play, feature A can be engaged
- simplistic

Context is Stable

- Alternative
  - Context is an occasioned property
  - Particulars of situation and activity matter

- example:

-**Dey**: Relevance of user's proximity to an exhibit is always relevant

- **Dourish**: highly dependent on the current situation

Case Study: Web Apps

- Do they match up with our discussion of context?
- How effective are they?
- What are the problems?
- What can they learn from the views of context discussed here?
- **Dey**: Context can be represented and processed
- **Dourish**: Context is emergent. Applications should help users produce new meanings and contexts

Case study: context-aware mobile tools

- Killer app? The Virtual Personal Assistant
  - Android Friday™
  - iPhone Siri™
- Mashup of existing disparate tools, information, activities
- Infrastructure of services and APIs

Android™ Friday™ Personal assistant

Siri™


AT&T 3G 10:07 AM
“Where can I find a sandwich”
I found a number of Sandwich restaurants fairly close to you:
NW 1st Avenue
The Peoples' Sandwic…
83 reviews
0.3 miles
East Burnside
Slow & Low
5 reviews
0.9 miles
West Burnside
Subway
4 reviews
0.9 miles
AT&T 10:12 AM
“Where can I get a beer”
I found a number of bars fairly close to you:
SW Ankeny Street
Tug Boat Brewing Co
102 reviews
0.1 miles
SW 2nd Avenue
XV
53 reviews
0.2 miles
SW Ankeny Street
Berbati's Pan & Resta…
41 reviews
0.2 miles
West Burnside Street
AT&T 3G 9:59 AM
“When did Abraham Lincoln die”
I found this for you:
Input interpretation
Abraham Lincoln | date of death
Result
Saturday, April 15, 1865
Date formats
04/15/1865 (month/day/year)
Time difference from today (Friday, October…
146 years 5 months 29 days ago
7643 weeks 6 days ago

Siri Helps you Do Things

- Multiple-criteria vertical search and browse
- combining multiple sources of information
- with integrated transactions
- and social communication

©Tom Gruber, Siri, Inc. 2009

Employing the services of...

Contacts

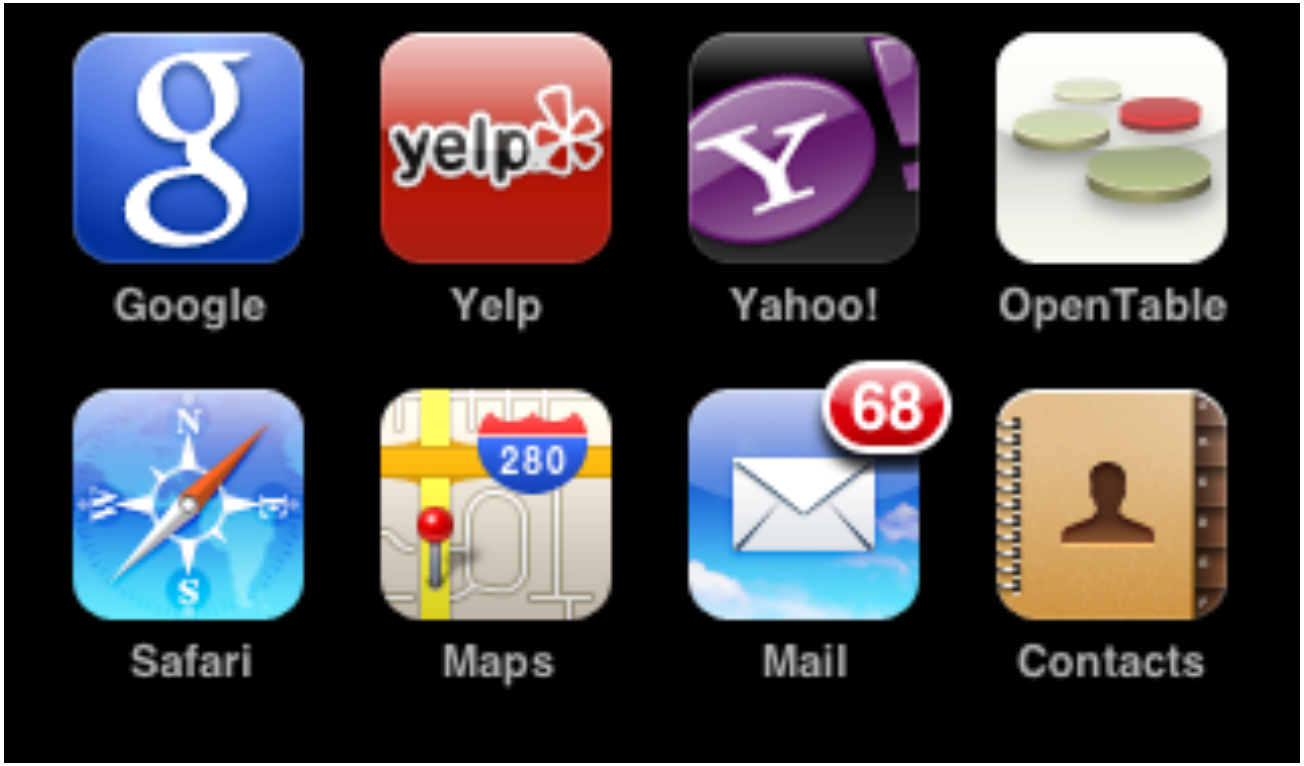
Email

Maps

and many more...

©Tom Gruber, Siri, Inc. 2009

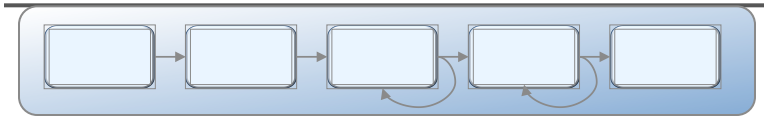
Lots of Steps, Lots of Apps



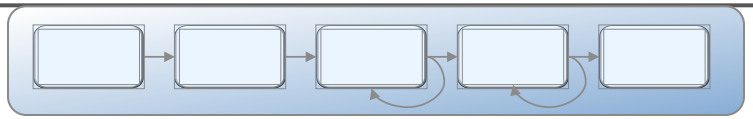
Apps required to do meal planning flow

©Tom Gruber, Siri, Inc. 2009

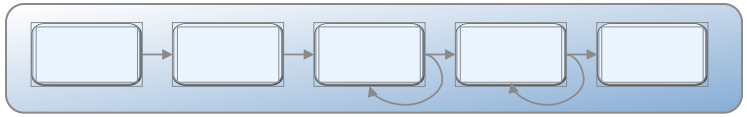
And Lots of Tasks



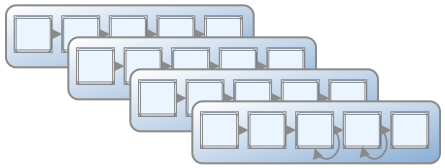
Plan a Meal



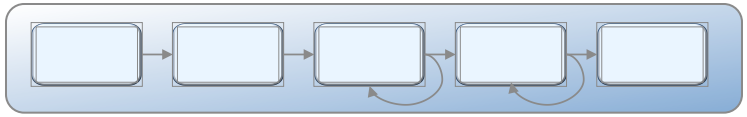
Find a store



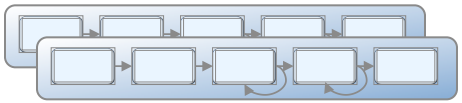
Find something to do



Plan a trip

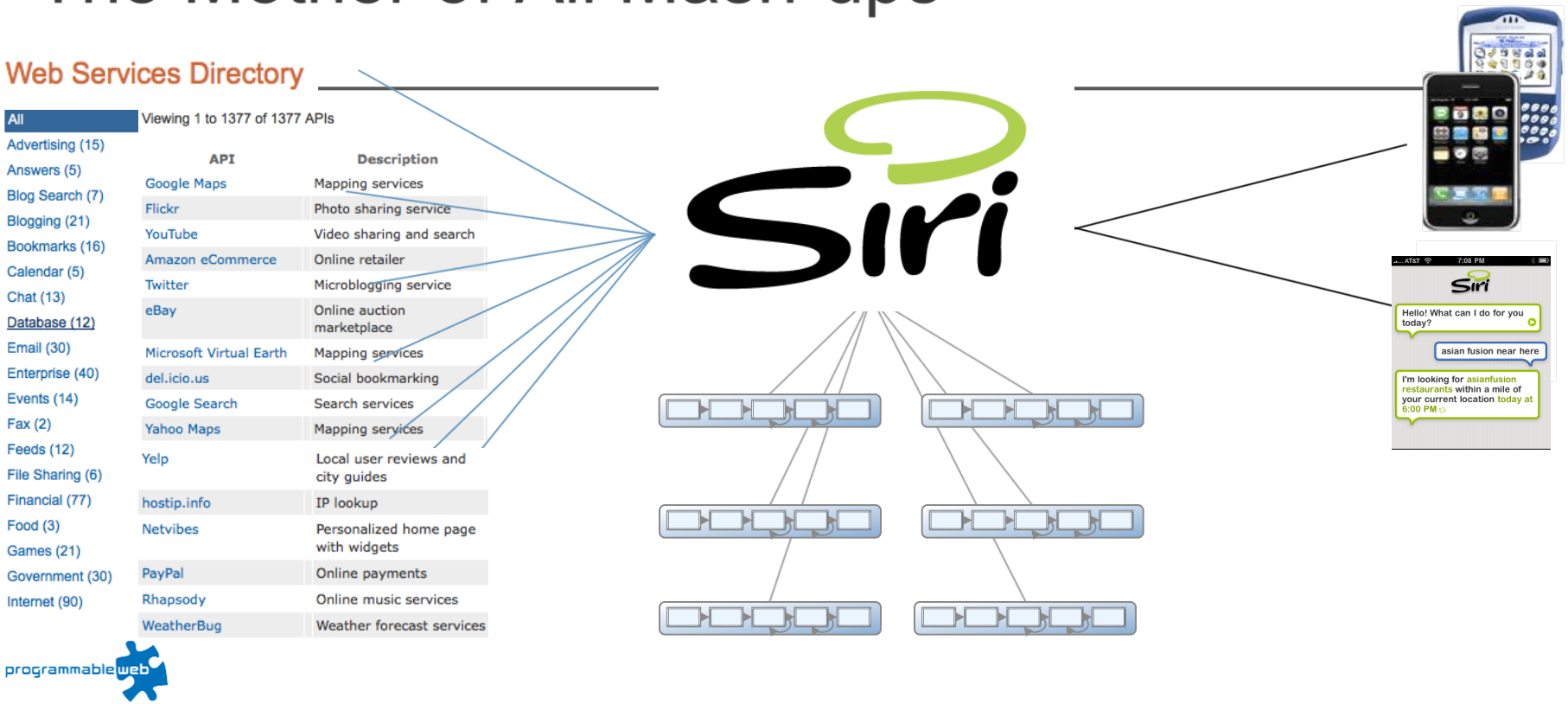


Go to the movies



. . .

Service Delegation: The Mother of All Mash-ups



Web Services and APIs | **Domain & Task Models** | **Guided Dialog**

©Tom Gruber, Siri, Inc. 2009

Siri:

- **Does Things for You**
  focus on task completion
- **Gets What you Say**
  intent understanding via conversation
- **Gets to Know You**
  learns and applies personal information

Understanding Intent in Context

- Location Context

... near **home**

... in **downtown** [San Jose]

- Task Context

... [serving] **burgers**

... [described as] **romantic**

- Time Context

... **this weekend**

... **tomorrow night**

- Dialog Context

... hotels **in Boston**

... **weather** [in Boston]

Siri:

- **Does Things for You**
  focus on task completion
- **Gets What you Say**
  intent understanding via conversation
- **Gets to Know You**
  learns and applies personal information

Development Evolution

Doing Things For You

- Service APIs
- Data feeds
- Auth Standards
- SW & Data Commons
- Faceted Search Domains
- Recommendation Services

Getting What You Say

- Location
- Speech
- Semantic NLP
- Linguistic NLP
- Date/time
- Conversational UI
- Social Contexts

Getting Personal

- Contacts
- Calendars
- Explicit Preferences
- Profile
- Favorites
- Learned Preferences

Today — Tomorrow — Future

Implications on Design

- Predefined contexts will likely fail
- What are the critical features?
- What can we live with?
  - Highly context dependent

- Cost of being wrong
- Overhead of recovery
- Support evolution of meaning through practice