# G54MRT: Ubiquitous Computing

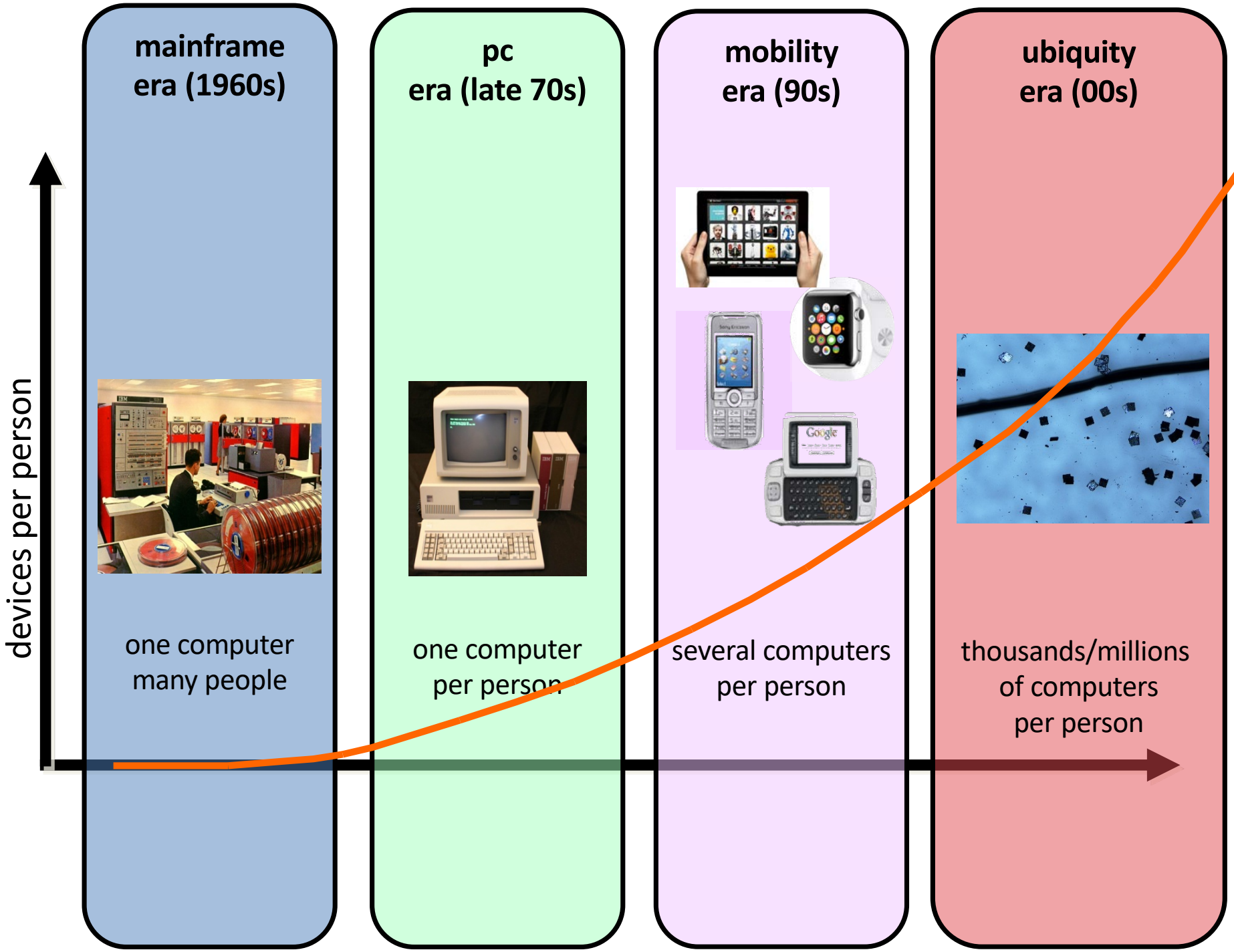
**Stuart Reeves** & Joe Marshall

*Topic: Ubicomp systems design*

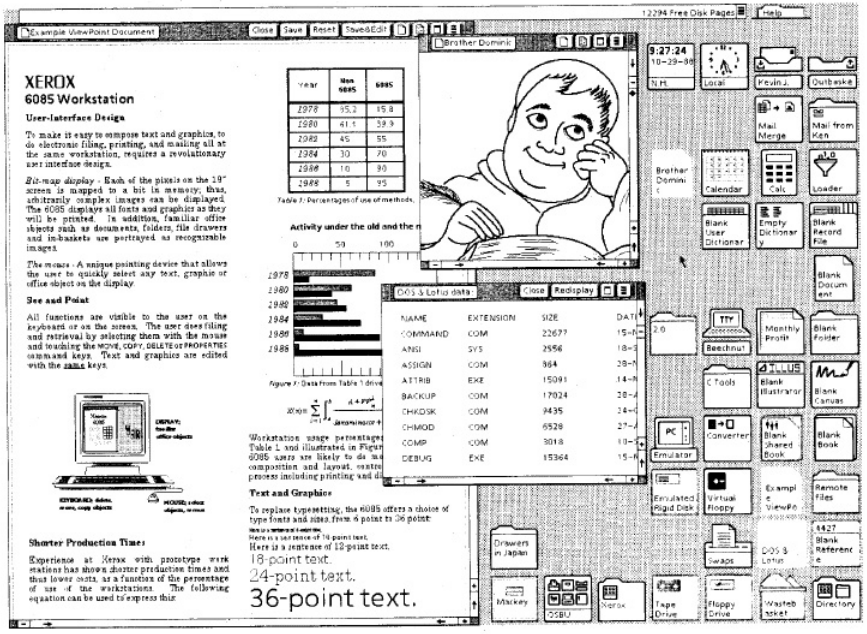
v2

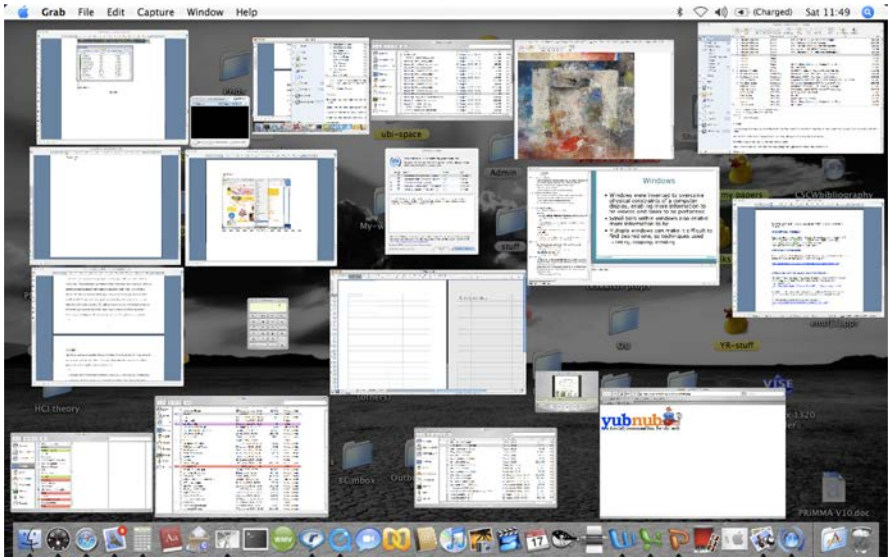
# Designing ubiquitous computing systems

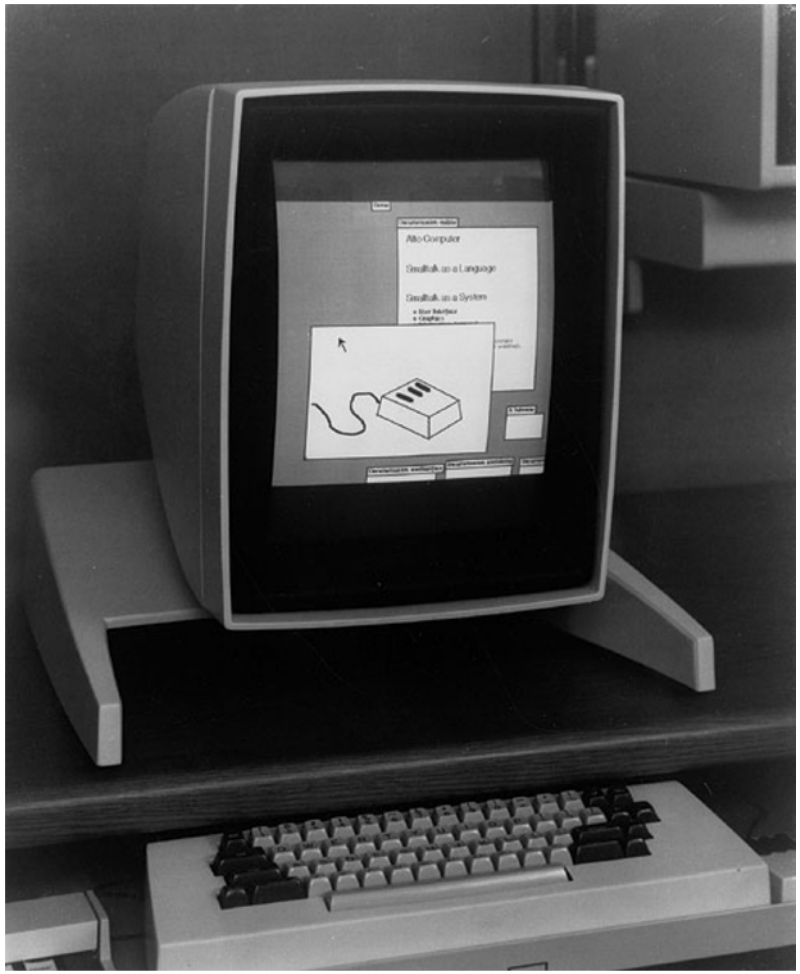
- Two lectures about ‘users’
- Goals for this lecture
  - Introduce basic interactive systems design ideas
    - Contrast between interaction design for ubicomp and pre-ubicomp systems
    - Understand core design challenges for ubicomp systems
  - Simple design framework for thinking about ubicomp systems design
  - Applying this to your (emerging) proposals
    - *Project proposal group discussions*
- Goals for next design lecture (12th March)
  - Case study of contextual / feedback relationship between sensor systems + human ‘users’
  - More sophisticated frameworks for thinking about ubicomp design


mainframe era (1960s)
pc era (late 70s)
mobility era (90s)
ubiquity era (00s)
devices per person
one computer many people
one computer per person
several computers per person
thousands/millions of computers per person

# How is design for ubicomp systems different?

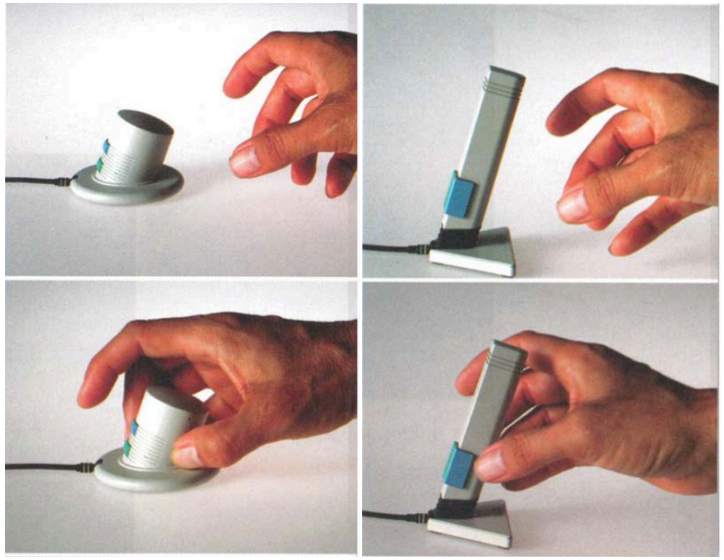


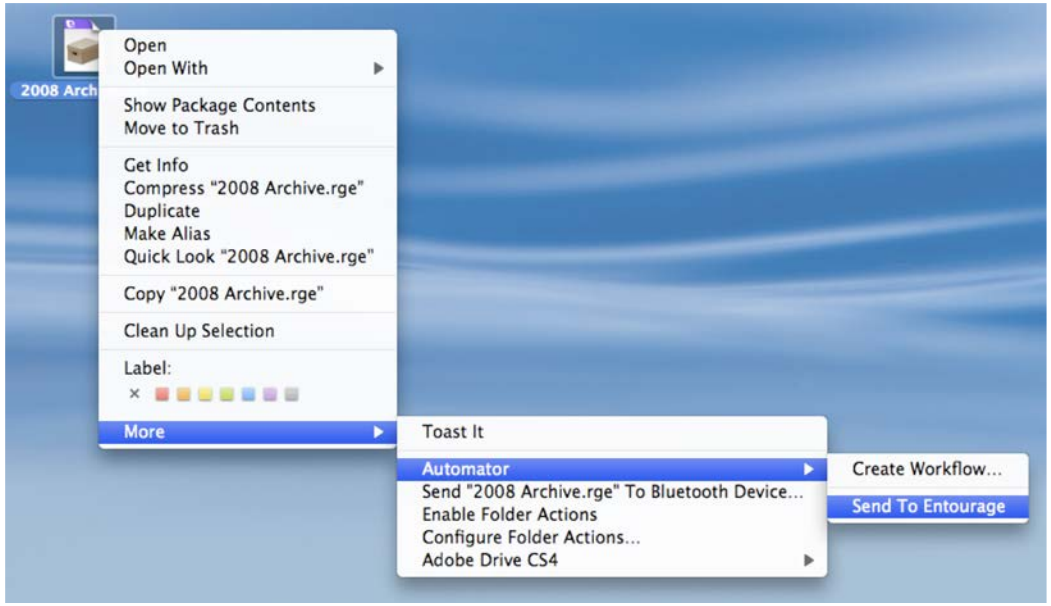
# Interaction design for the desktop

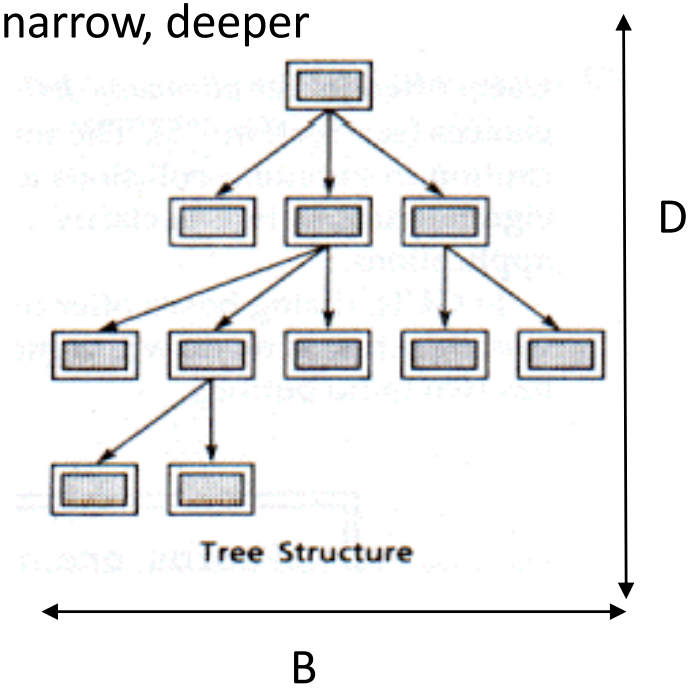
- Interaction design for traditional computation:
  - Predominantly input-device and graphically-oriented
- Leveraged things like:
  - Gestalt psychology (.. .. ..)
  - Cognitive models, hand-eye coordination systems



# Interaction design for the desktop

- E.g., how to construct menus?
- Breadth generally better than depth (Landauer & Nachbar 1985)
- Depth = number of levels
  - Ideally, no more than 3 to 4 levels
  - More levels leads to navigation problems
- Breadth = number of items per level
  - Ideally, no more than 8 items/level
  - More than this leads to scanning problems
- Best tree structure: a menu breadth that falls within G. A. Miller's (1956) 'seven plus or minus two'
- Broader, shallower menu trees = faster search than narrow, deeper



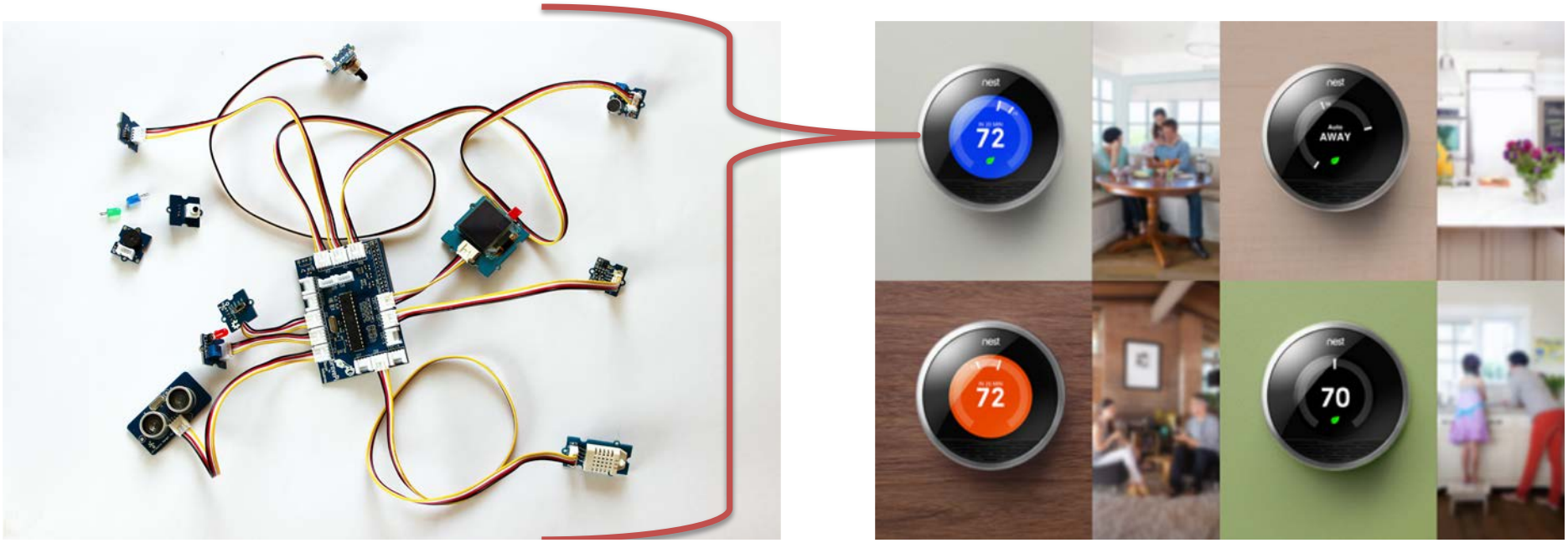
# Classic interaction design heuristics (Neilsen 1995)

- **Visibility of system status:** The system should always keep users informed about what is going on
- **Match between system and the real world:** speak the user's language, follow real world conventions
- **User control and freedom:** E.g., support undo and redo
- **Consistency and standards:** Follow convention, ensure words, actions, situations follow expectations
- **Error prevention:** Avoid the possibility of error
- **Recognition rather than recall:** Make options, actions, etc. visible
- **Flexibility and efficiency of use:** Cater for expert and novice users (e.g., via accelerators)
- **Aesthetic and minimalist design:** Only do what is needed
- **Help users recognize, diagnose, and recover from errors**
- **Help and documentation:** Do you really need it?

# IBM supermarket (2006)

# Key issues for designing ubicomp systems

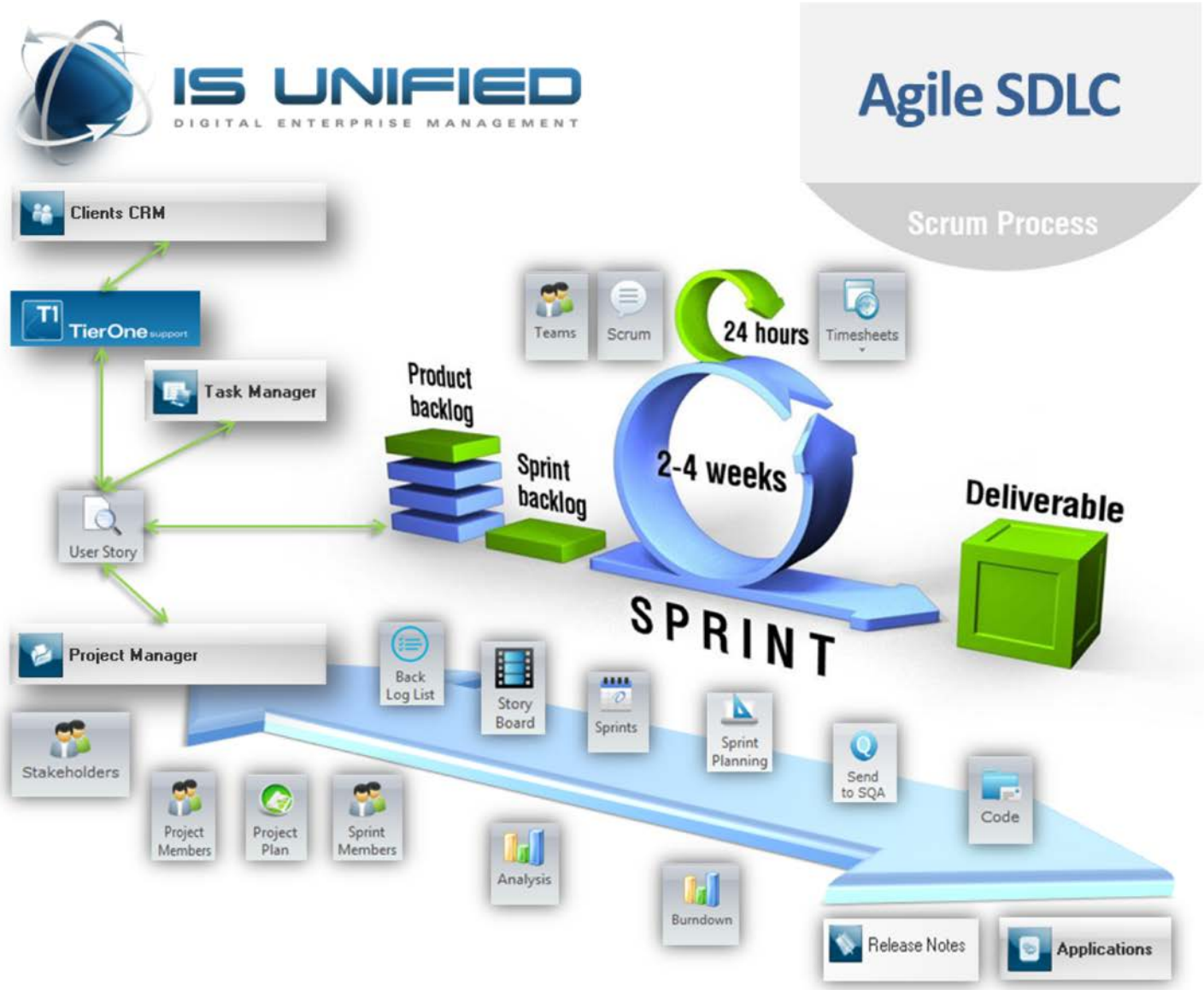
- Ubicomp systems (and their **sensors**) are **embedded** in the world
  - Ubicomp sensor data cannot be understood without **context of use**
  - Ubicomp sensor systems **feed-back** (mutually constitutive)



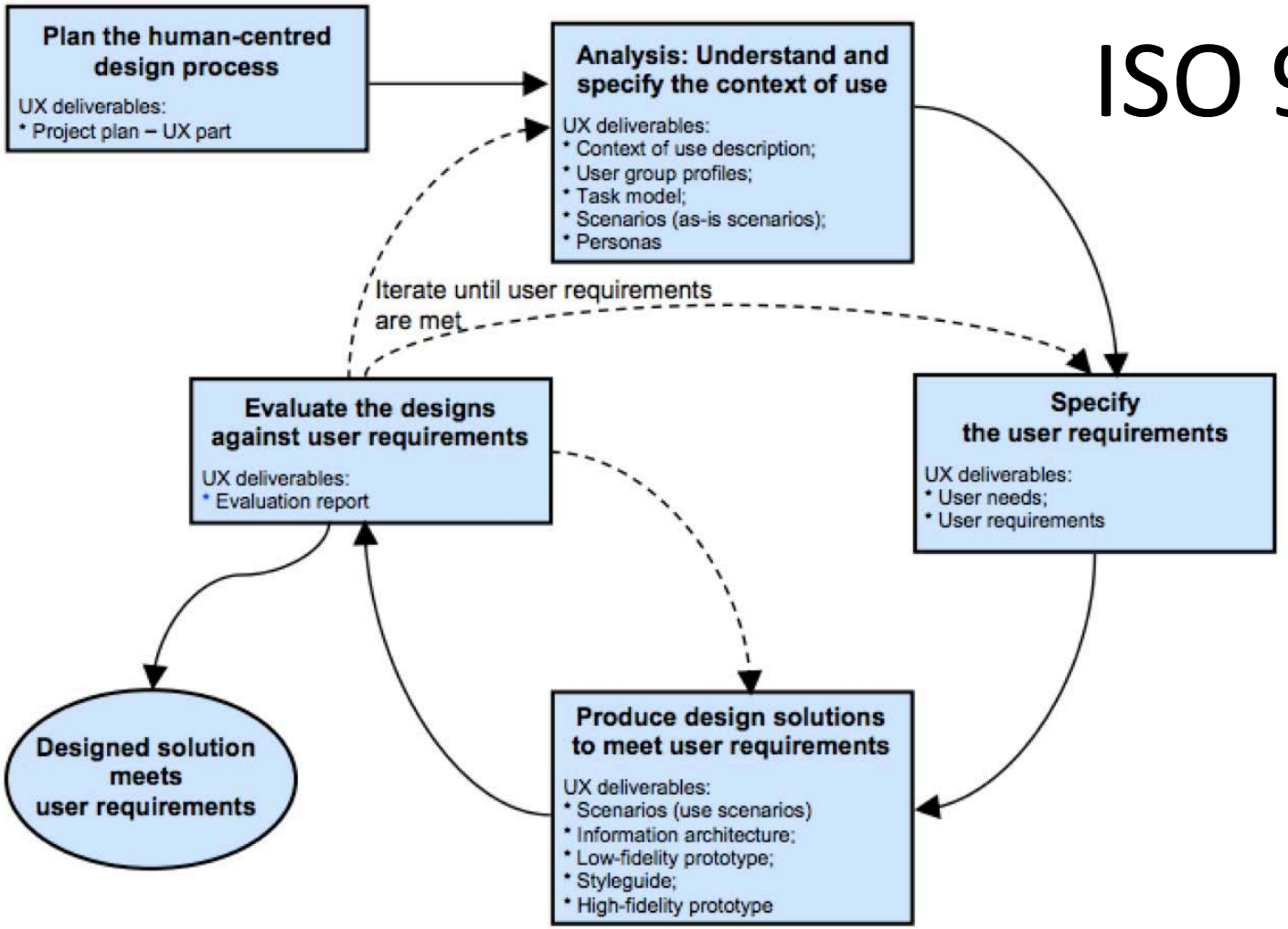
# Three *interactional* challenges for ubicomp systems design

- **Invisibility**: Ubicomp is invisible computing
  - Inaccessible or doesn't 'look like' a computer
  - No user interface? 'Displaced' user interface
- **Situation** / **context** of use
  - Embedded in everyday life / everyday practices
  - Becoming unwittingly implicated
- **Technological seams**
  - Unbroken and seamless service provision

# Why should you care about design?



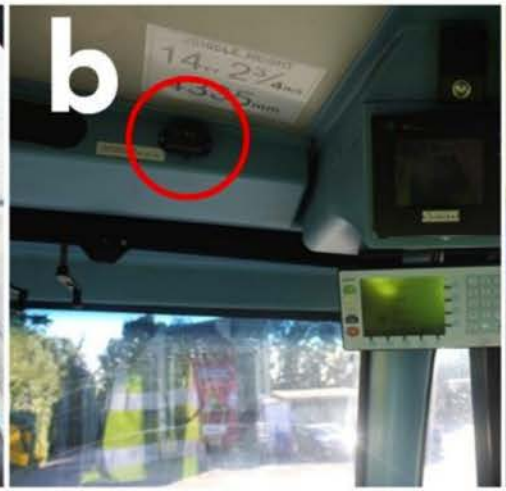
# Why should you care about design?

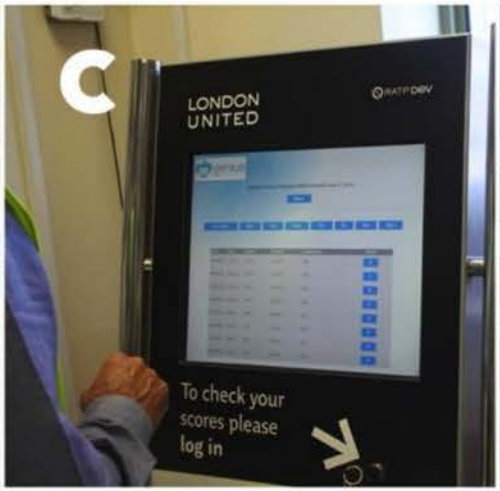



ISO 9241

# The reality: Connected products

# The reality: Living with sensing systems



# Who are the ‘users’?
# Simple framework for system design (from ISO standard)

- Who are the **stakeholders** in your system design? E.g.:
  - Who are the **direct** users?
    - Direct users interact with the system
  - Who are the **indirect** users?
    - E.g., use output of the system but don’t interact directly with it
  - Who is a **primary** user?
    - Interacts with system to achieve goals supported by system
  - Who is a **secondary** user?
    - Interacts with system to support or maintain it
- What **user needs** and **requirements** is your system serving?
  - User needs are (normally) independent of proposed system
  - Requirements respond to user needs
- What are potential **use cases** / **scenarios**?
  - Imagine step-by-step scenarios for different user types
  - From this what sensors might help characterise the needs of those users?

# Example:
# A room monitoring system

- How busy is a room?
- How many people are in the room?
- What does long-term usage of the room look like?

# Example: LabMon proposal

**Summary**

Use sensor data to detect how busy the A32 lab is. This could provide data for a web page to help me choose when I want to go to the lab to work.

**Technologies and sensor data**

I will develop a fixed sensor unit for the room in question (A32). I will use the motion sensor and noise sensors, possibly the ultrasound and light sensors, and maybe the temperature sensor.

…

# Example: LabMon proposal (stakeholders)

Use sensor data to detect how busy the A32 lab is.

- Student activity in the lab will be captured (e.g., via motion, sound, etc.)
  - Direct user
- Activity in the lab will be made available to lecturers so as to monitor lab attendance.
  - Indirect user, primary user
- The sensor boxes will be maintained by system administrators
  - Secondary user

# Example: LabMon proposal (user needs)

- Lecturers need to find out student attendance
- Lecturers need to see how accurate counts are
- Students need to be non-identifiable / anonymous for personal privacy
- Students need to understand how their data is being used
- Students need to know when they are and are not being tracked
- System administrators need to access the sensor boxes at appropriate times
- System administrators need to understand how upgrades may impact student and lecturer use of the system

# Example: LabMon proposal (scenario)

- A lecturer is planning the teaching for next year's module (which has labs)
- She wants to compare attendance across the whole semester to know how many lab assistants to recruit for the next year
- She opens up the LabMon web interface to find out
- She is presented with a graphical representation of the whole year
- Specifying her module, the web interface then shows a graph of attendance with on-screen guidance about accuracy of this data
- ...

- Thinking about the design will not *necessarily* change your core idea and what you implement technically
- It *should* change how you report the broader background / motivation and design for your system



# G54MRT Coursework 2 Final Report

*[This document provides a suggested structure for the final report]*

## Title:

Student ID:
Date:

## Summary

*[Brief summary of the project and the contents of the report]*

## Background and motivation

*[Describe the proposal idea, its motivation and the context which the system was designed for]*

## Related work

*[Discuss any literature presented in lectures / on Moodle that is relevant to your work and briefly why, e.g., design frameworks for sensing, ubicomp vision, ubicomp literature, etc. and any existing available / related applications / hardware / products. You can reserve critical reflection on this literature and your system for later, see below.]*

## Design

*[Describe the interaction design of the project where relevant. Diagrams, sketches, use cases etc. can help make this clear.]*

*[Describe the technical design of the project, note that this could include aspects that you haven't implemented but were still part of the design process. Again, system diagrams]*

## Implementation

*[Describe what was actually implemented. You might want to include screen shots, photographs of the prototype setup, sensor data examples, brief source code snippets etc.]*

## Testing

*[Describe testing processes you engaged in, e.g., data calibration / ground truth etc. Use graphs, tables of descriptive statistics etc. to show the results of testing]*



## Critical reflection

*[Critical consideration of the design, e.g., considering ethical implications of your system design, or issues discovered in project implementation, testing, etc.]*

## References

*[List any literature you have cited in the sections above.*
*http://www.nottingham.ac.uk/is/documents/about/inductionguides/references.pdf is a useful guide for how to cite literature.]*

## Appendix A – instructions

*[Put instructions related to running / operating your coursework's code etc. here]*

## [Appendix B – NOTE: You need to submit your coursework as zip file to Moodle – including this report and source code, test data files etc.]

# Group discussions:
# Apply this to your proposals

- Who are the **stakeholders** in your system design? E.g.:
  - Who are the **direct** users?
  - Who are the **indirect** users?
  - Who is a **primary** user?
  - Who is a **secondary** user?
- What **user needs** is your system serving?
- What are potential **use cases** / **scenarios**?

- *Be creative*
- *You are not committed to follow through*

- Python + sensors (double lecture) 20th Feb
- Proposals due 22nd Feb (Friday) @ 3pm

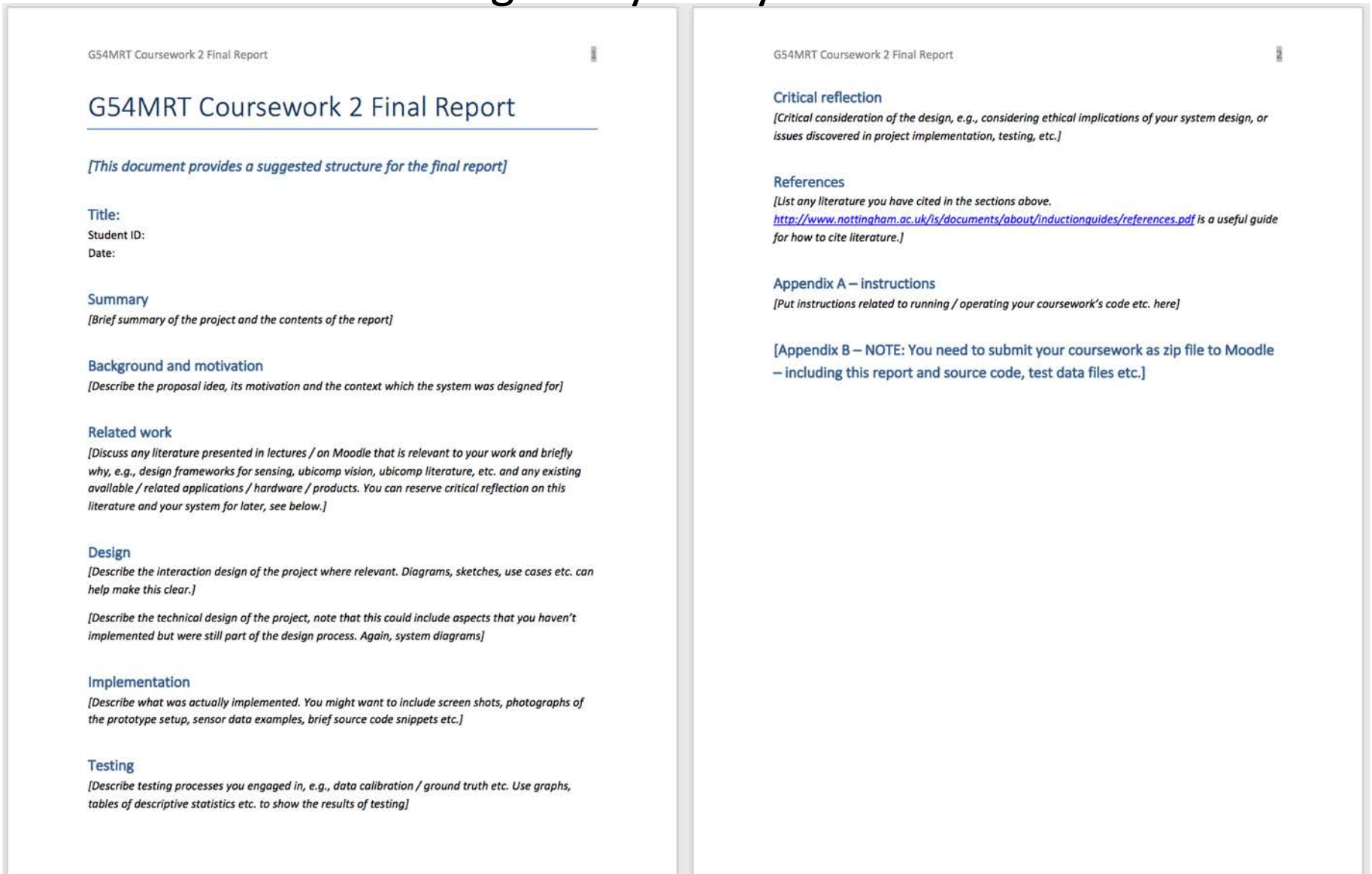
# Overview

- Two questions for ubicomp sensing systems:
  - How **context** can be important
  - The role of **feedback** in ubicomp systems
- Other ways to think about ubicomp sensing systems:
  - “Expected, sensed, and desired” design framework
  - Designing for seams in sensor systems

# Question: Who is the user?

- Who are the **stakeholders** in your system design? E.g.:
  - Who are the **direct** users?
    - Direct users interact with the system
  - Who are the **indirect** users?
    - E.g., use output of the system but don't interact directly with it
  - Who is a **primary** user?
    - Interacts with system to achieve goals supported by system
  - Who is a **secondary** user?
    - Interacts with system to support or maintain it
- What **user needs** and **requirements** is your system serving?
  - User needs are descriptions of what a user must be able / would like to achieve (*independent* of proposed system, never tech-driven)
  - Requirements respond to user needs; requirements are technical design responses to needs
- What are potential use cases / **scenarios**?
  - Imagine step-by-step scenarios for different user types; these elaborate context
  - From this what sensors might help characterise the needs of those users?

- Thinking about the design will not *necessarily* change your core idea and what you implement technically
- It *should* change how you report the broader background / motivation and design for your system





# G54MRT Coursework 2 Final Report

*[This document provides a suggested structure for the final report]*

## Title:

Student ID:
Date:

## Summary

*[Brief summary of the project and the contents of the report]*

## Background and motivation

*[Describe the proposal idea, its motivation and the context which the system was designed for]*

## Related work

*[Discuss any literature presented in lectures / on Moodle that is relevant to your work and briefly why, e.g., design frameworks for sensing, ubicomp vision, ubicomp literature, etc. and any existing available / related applications / hardware / products. You can reserve critical reflection on this literature and your system for later, see below.]*

## Design

*[Describe the interaction design of the project where relevant. Diagrams, sketches, use cases etc. can help make this clear.]*

*[Describe the technical design of the project, note that this could include aspects that you haven't implemented but were still part of the design process. Again, system diagrams]*

## Implementation

*[Describe what was actually implemented. You might want to include screen shots, photographs of the prototype setup, sensor data examples, brief source code snippets etc.]*

## Testing

*[Describe testing processes you engaged in, e.g., data calibration / ground truth etc. Use graphs, tables of descriptive statistics etc. to show the results of testing]*



## Critical reflection

*[Critical consideration of the design, e.g., considering ethical implications of your system design, or issues discovered in project implementation, testing, etc.]*

## References

*[List any literature you have cited in the sections above.*
*http://www.nottingham.ac.uk/is/documents/about/inductionguides/references.pdf is a useful guide for how to cite literature.]*

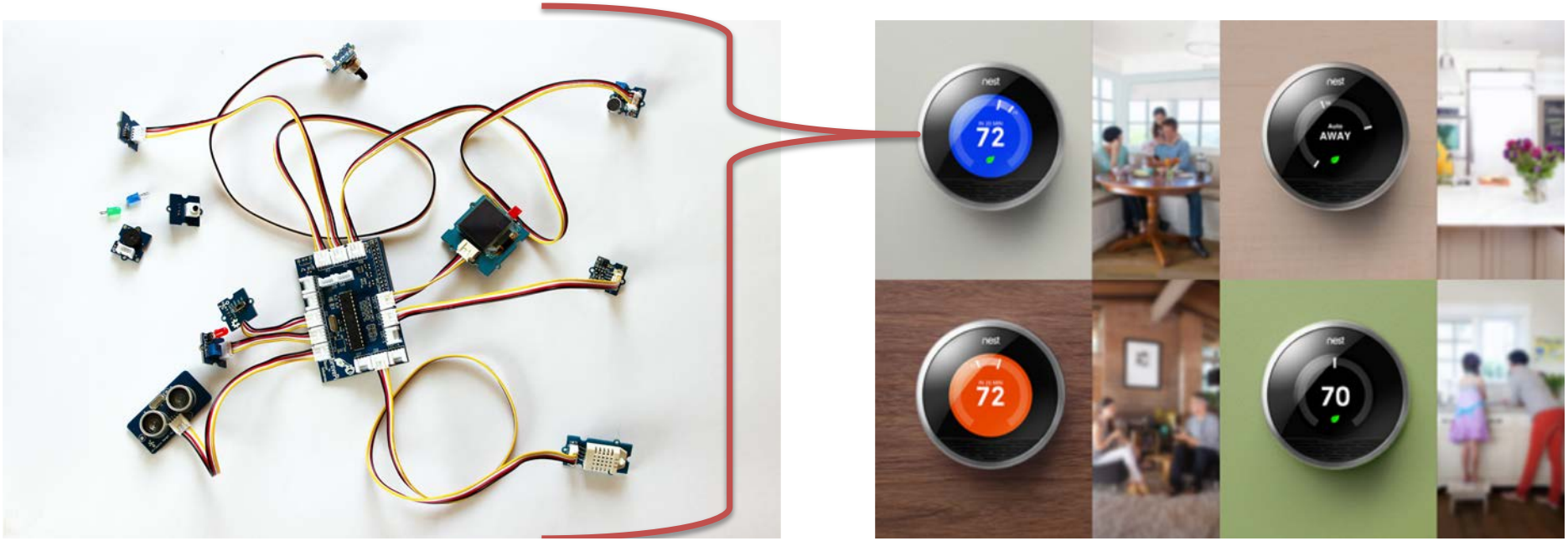
## Appendix A – instructions

*[Put instructions related to running / operating your coursework's code etc. here]*

## [Appendix B – NOTE: You need to submit your coursework as zip file to Moodle – including this report and source code, test data files etc.]

# Embeddedness: context & feedback

- Ubicomp systems (and their **sensors**) are **embedded** in the world
  - Ubicomp sensor data cannot be understood without **context of use**
  - Ubicomp sensor systems **feed-back** (mutually constitutive)
- This is why we want you to think about scenarios of use



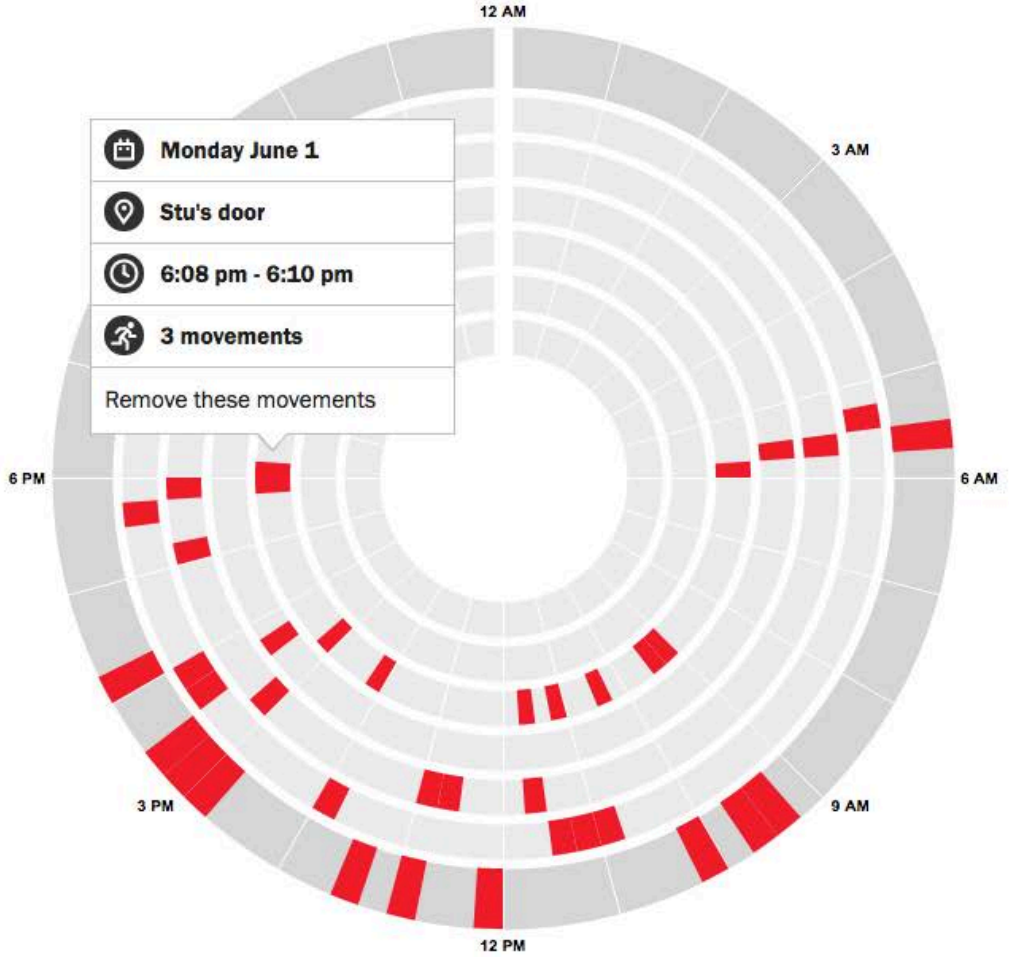
# Ubicomp systems & **context**

Senseboard **Door Stu's office** James

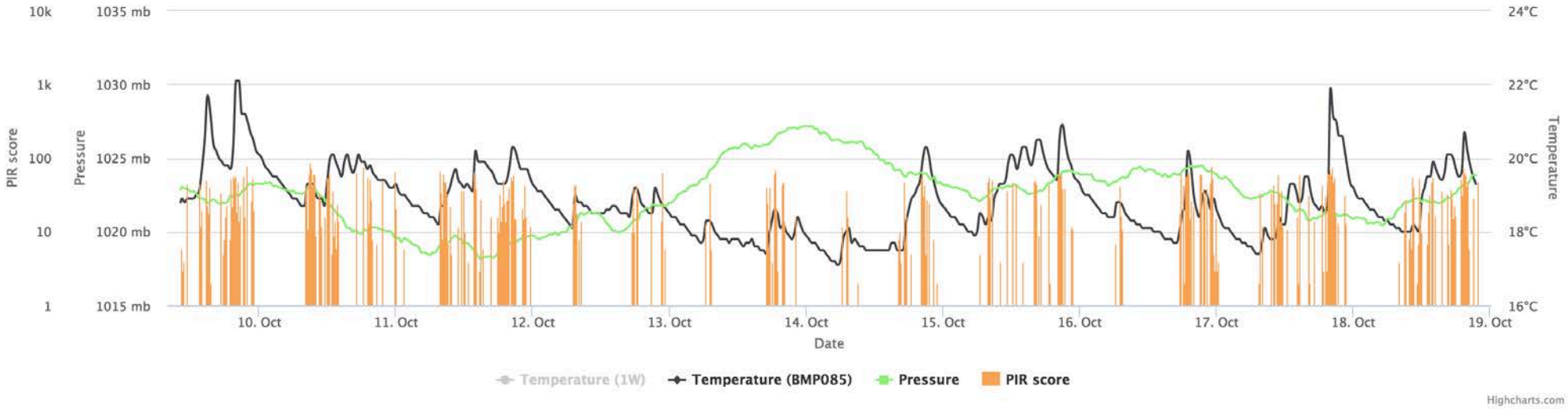
**7 DAYS** MONTH

Friday June 5 and the previous six days

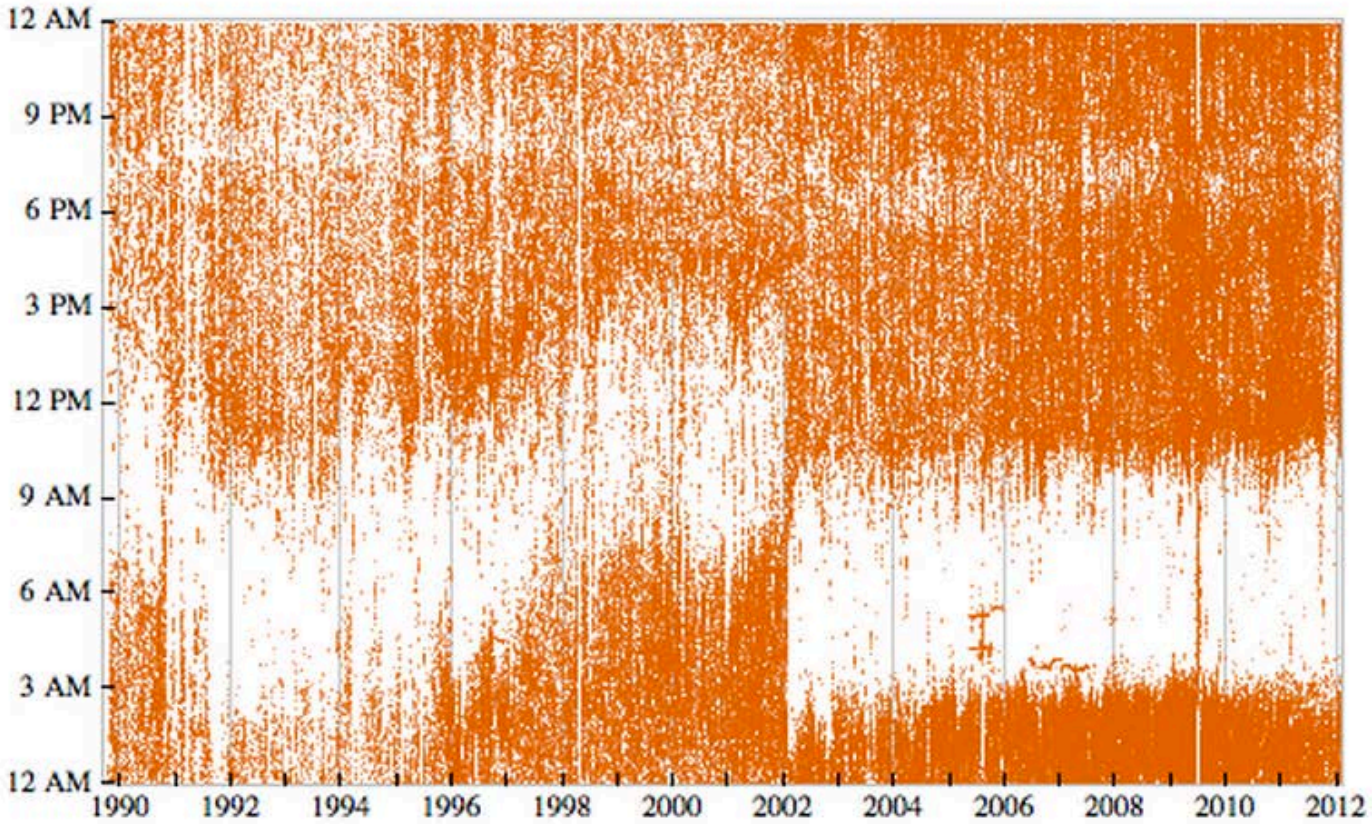
Last updated on October 2 at 4:34 PM




Stu's door (78)


10k
1k
100
10
1
PIR score
1035 mb
1030 mb
1025 mb
1020 mb
1015 mb
Pressure
24°C
22°C
20°C
18°C
16°C
Temperature
10. Oct
11. Oct
12. Oct
13. Oct
14. Oct
15. Oct
16. Oct
17. Oct
18. Oct
19. Oct
Date
Temperature (1W)
Temperature (BMP085)
Pressure
PIR score
Highcharts.com

http://blog.stephenwolfram.com/2012/03/the-personal-analytics-of-my-life/




The first thing one sees from this plot is that, yes, I've been busy. And for more than 20 years, I've been sending emails throughout my waking day, albeit with a little dip around dinner time. The big gap each day comes from when I was asleep. And for the last decade, the plot shows I've been pretty consistent, going to sleep around 3am ET, and getting up around 11am (yes, I'm something of a night owl). (The stripe in summer 2009 is a trip to Europe.)

# Data doesn't mean anything by itself

# What *does* sensor data mean?

- You are **encoding** human interpretations / human contextual understandings of data **into** your systems when you write the software that collects and makes interpretations of sensor data
- E.g., **you** decide the meaning of a PIR triggering
  - That meaning is what you are building into your systems

# Dey (2001) on context

- "Context is any information that can be used to **characterise** 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 applications themselves."
  - If some information is **relevant** to characterise the situation (user's task), then **that information is context**
  - E.g., weather data is probably not relevant to **characterise** lab context

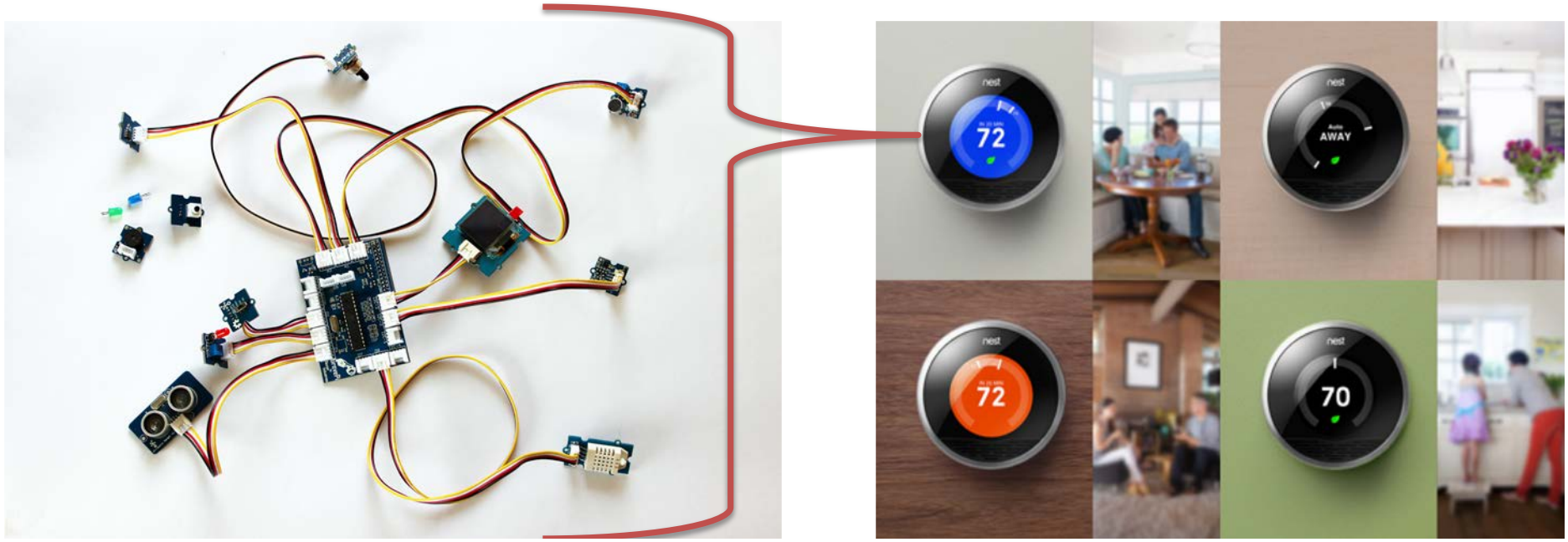
# Questions on context for your projects

- What interpretations *of your own* are you leveraging to make sense of your data?
  - E.g., for LabMon you might interpret lots of PIR signal and higher ambient temps as ‘labs’
- How similar or different is the system ‘interpretation’ to human interpretation?
  - E.g., maybe 80% of the time both the ground truth / human interpretation and LabMon ‘see’ labs

# Ubicomp systems & **feedback**

# Embeddedness: context & feedback

- Ubicomp systems (and their **sensors**) are **embedded** in the world
  - Ubicomp sensor data cannot be understood without **context of use**
  - Ubicomp sensor systems **feed-back** (mutually constitutive)



# How to Drive a London Bus

## Measuring Performance in a Mobile and Remote Workplace

- Drivewell: in-vehicle data recorder
  - Speed, stability / vibration, braking, turn rate etc.
- Traffic light system on-board gives real-time feedback (b)



**Figure 1. From left to right: a) Placing the *Drivewell* key fob onto the reader; b) The *Drivewell* indicator in the driver's cabin (circled red); c) The *Drivewell* computer terminal; and d) close-up of computer terminal screen showing Drivewell scores.**

# How to Drive a London Bus

## Measuring Performance in a Mobile and Remote Workplace

- Detailed breakdown provided to drivers: date of journey, vehicle, duration, ‘achievement’ (given as a percentage) and overall number of ‘events’ (or incidences of poor driving)
- Data monitored by supervisors
- ‘Poor driving’ can lead to sanctions etc.

# Findings (1)

- Accommodation and acceptance
  - Self-surveillance and ability to ‘balance’ driving:
    - to “get around reasonably quick but not too quick so it’s too uncomfortable. Have the score sort of above average”
  - Reports supporting revised routing e.g., to hit fewer potholes and get a better score
- Questioning the system
  - Ambiguity and skepticism over what the system ‘counts’ and what it does not

# Findings (2)

- Resistance and subversion
  - Modifying behaviour to scrape past employer sanction
  - Avoiding too much ‘green’ through braking hard
    - “if you stay in the 100s, then all of a sudden you drop down to the 80s, they’ll want to know why”
  - Managing expectations of the supervisor
  - Removing Drivewell key when no passengers onboard - carving a gap in the apparatus of surveillance
  - Removal of Drivewell key during anticipated problems e.g., large pothole

# Embeddedness: context & feedback

- Ubicomp systems (and their **sensors**) are **embedded** in the world
  - Ubicomp sensor data cannot be understood without **context of use**
  - Ubicomp sensor systems **feed-back** (mutually constitutive)


# Other ways of thinking about sensing systems

- "Expected, sensed, and desired" framework
  - For understanding how to sense movement (although applicable more generally)
  - "**Expected** movements are those that users naturally perform; **sensed** are those that can be measured by a computer; and **desired** movements are those that are required by a given application."
- Seams in sensing systems
  - Breaks or breaches in continuous service provision
  - Inevitable but maybe you can design for / around them?

# Expected

- **Expected: “natural movements for a given combination of user, interface, and environment”**
- LabMon:
  - Walking around the lab space
  - Sitting on the chairs
- Actions that are “naturally” performed - but what is natural? Expect the unexpected

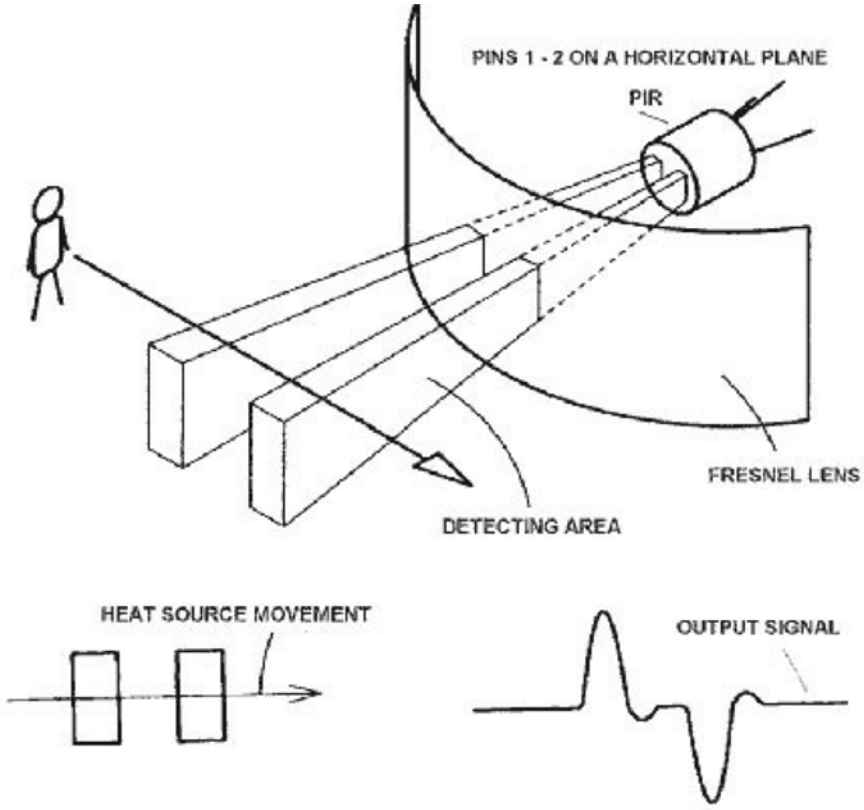

Getty Images, “Young man screaming into mobile phone taped to head, side view”

# Expected

- Physical form of interface might constrain possible movements etc.
  - E.g., maybe the introduction of a sensing system changes the form factor of an artefact

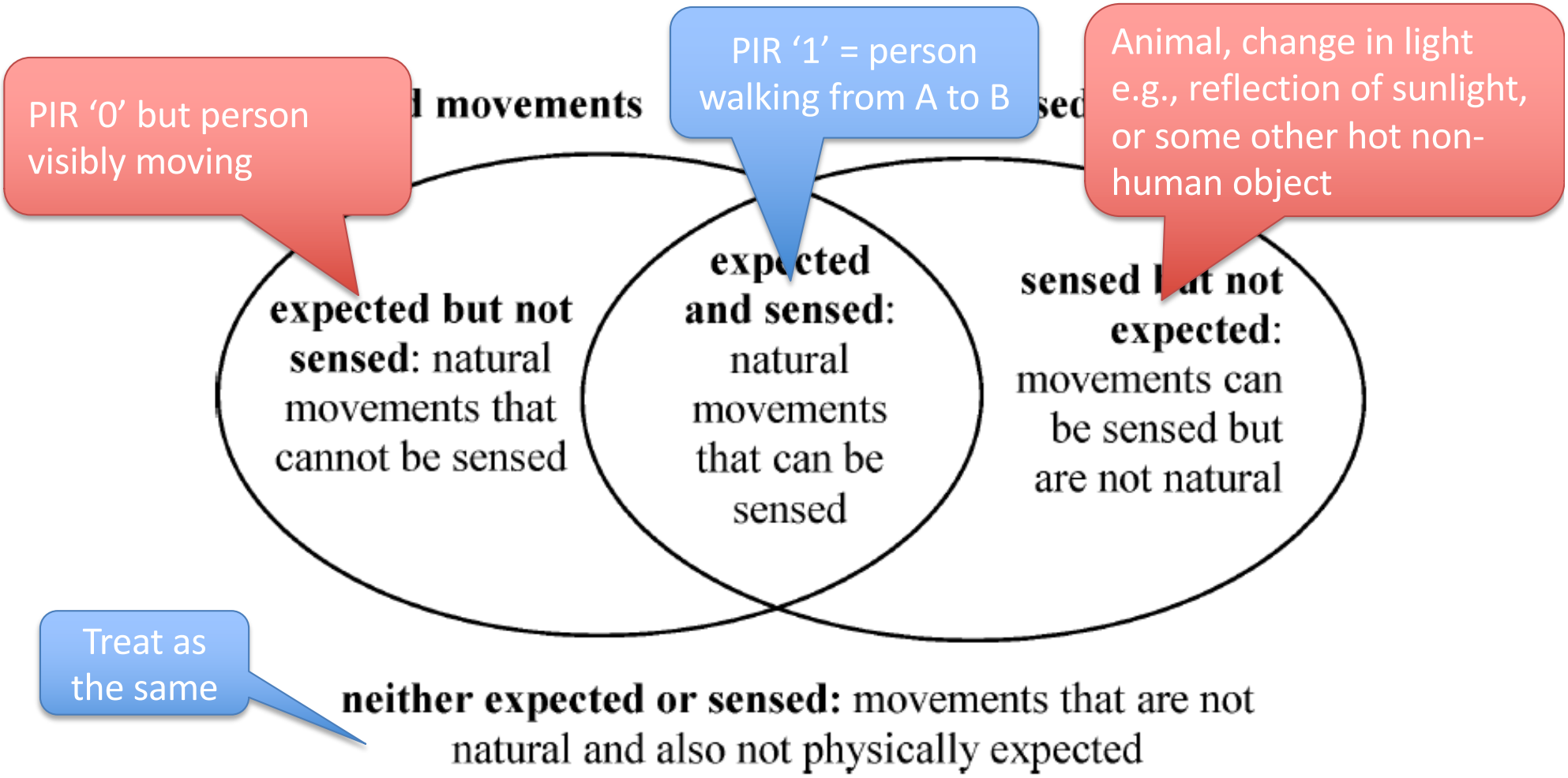
# Sensed

- **Sensed: "[movements] that can actually be measured by a computer"**
- LabMon:
  - *Changes* in heat emitted by a thing within the tolerances / range of the PIR
  - Compare with the description, "Walking around the lab space"



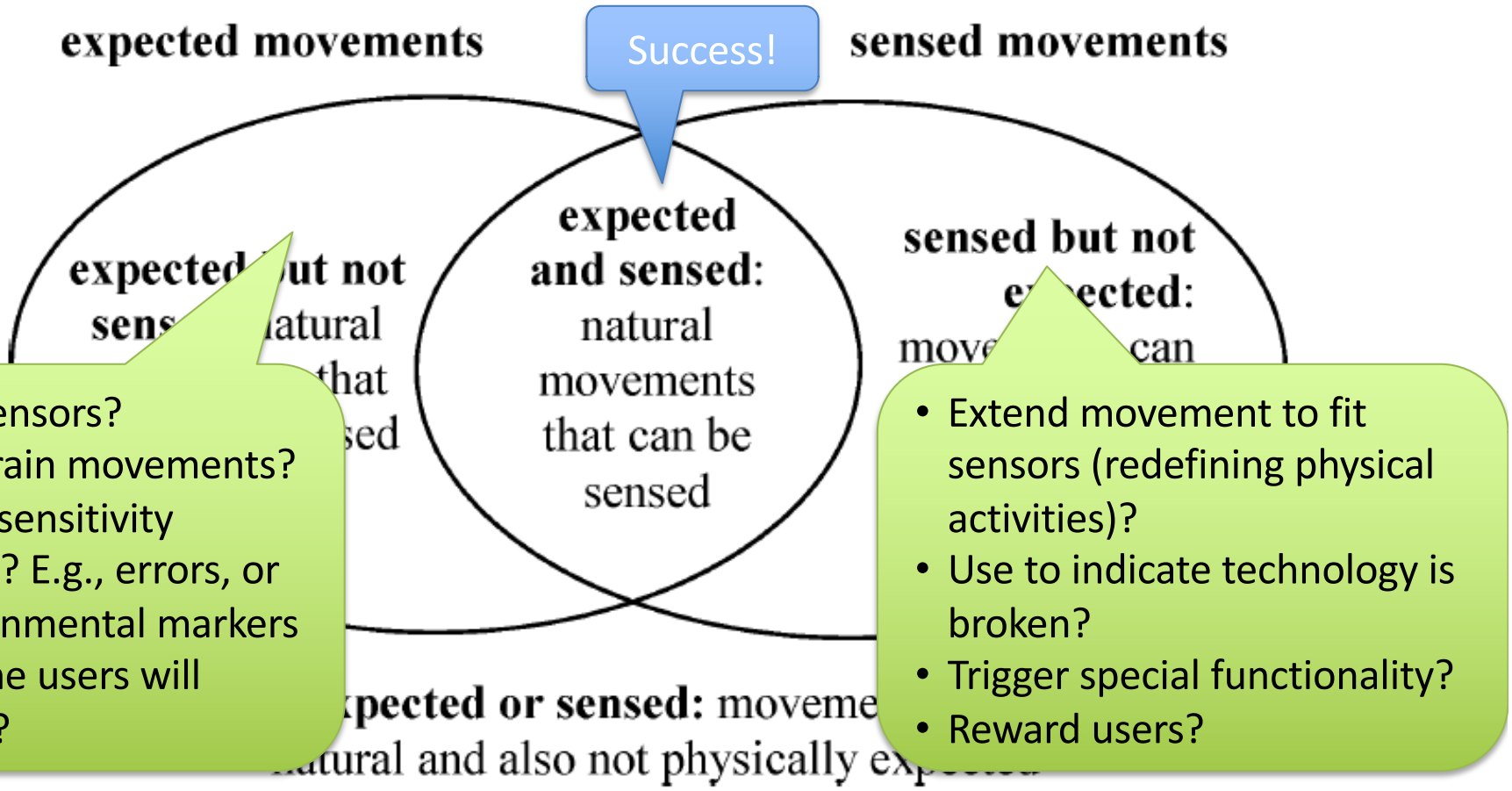
# Expected versus sensed

- Consider what happens in each zone



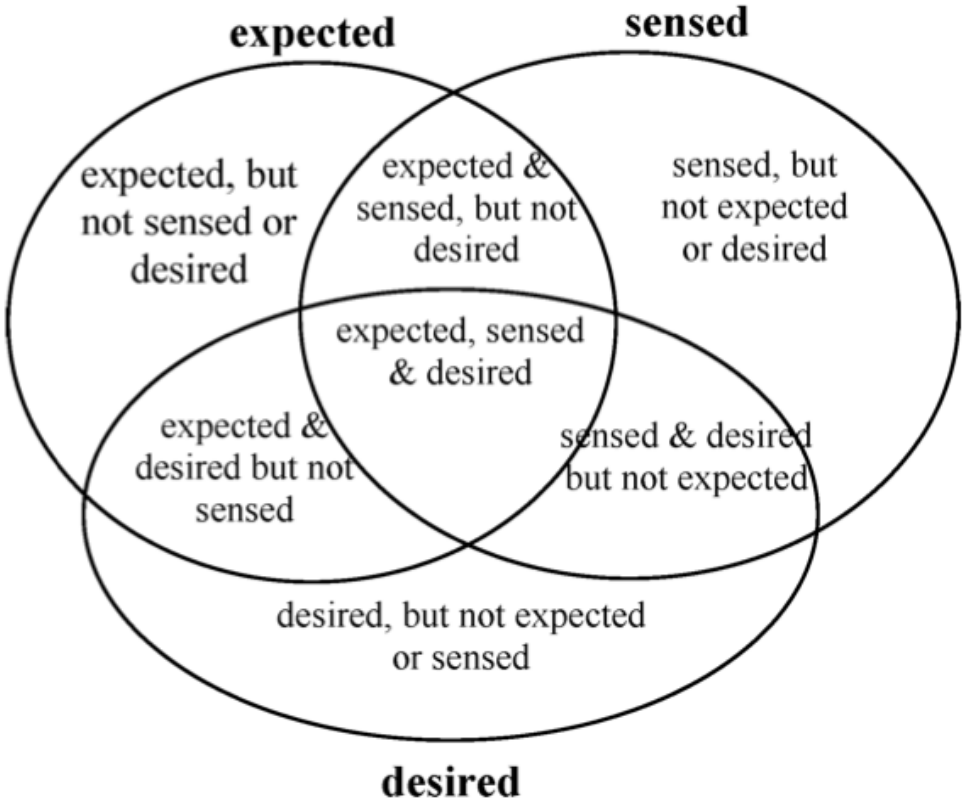
# Remedying problems

- Consider what happens in each zone



# Desired

- Desired movements are those we design for: ‘ideal’ uses / interactions
- Supported by the ‘ideal interface’



# The seams of ubicomp sensing systems

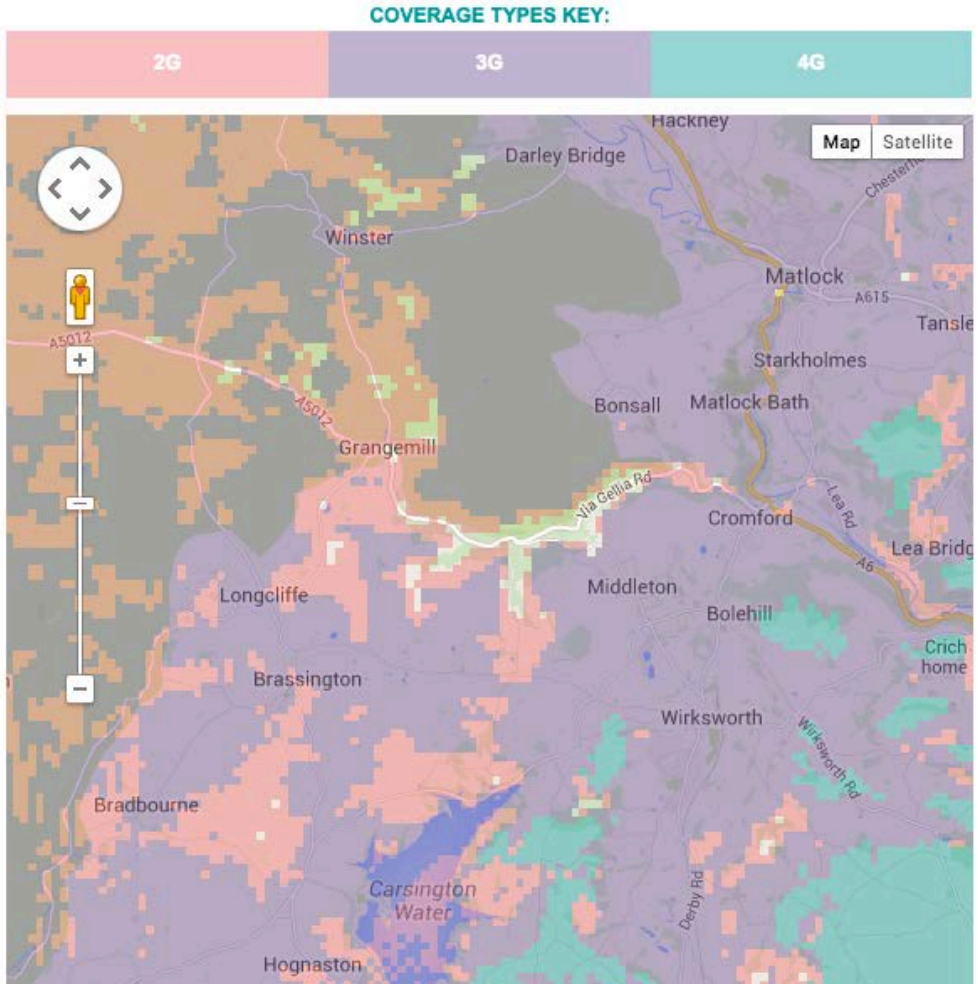
- What ubicomp could be:
  - Seamless and transparent integration of computation embedded / in the world
  - E.g., seamless interactions with sensor tech
  - E.g., seamless integration of sensed data with applications
  - E.g., seamless integration of comms networks
- Infrastructural integration
- Handling heterogeneity
- The vision of ubicomp interactions: effortlessness of user experience

# Seamless Cooltown



# Seams

- Seam in technology
  - A break, or gap in something that should be uniform
  - A ‘loss in translation’ between two things
  - E.g., wifi networks - designed in a way that assumes seamlessness



# Seamful design approaches

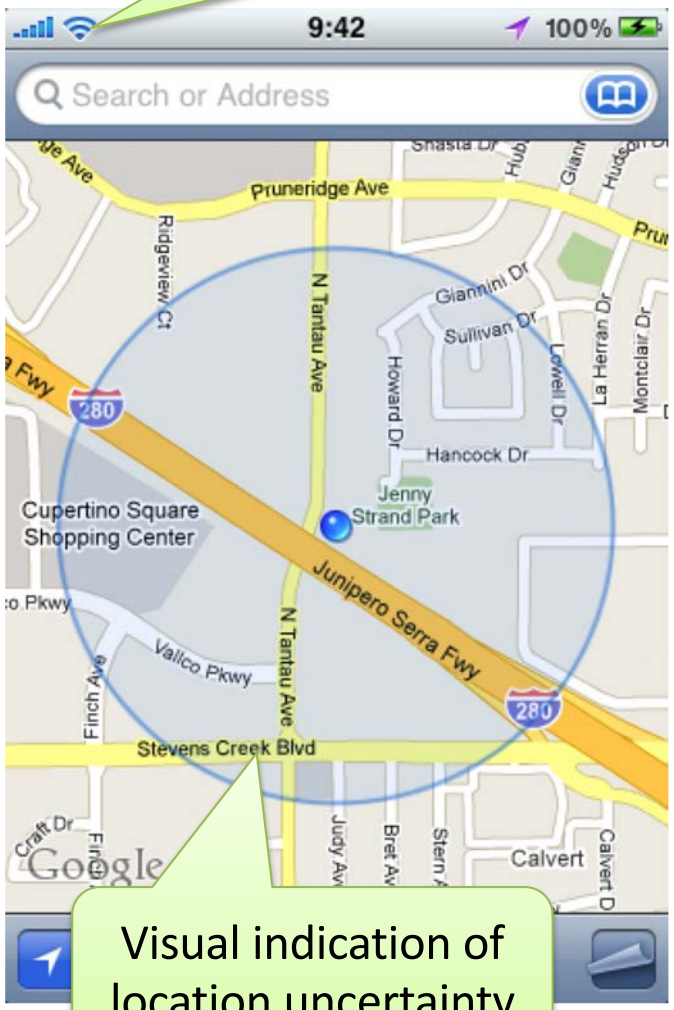
Potentially frustrating

- *pessimistic:* only show information that is known to be correct
- *optimistic:* show everything as if it were correct
- *cautious:* explicitly present uncertainty
- *opportunistic:* exploit uncertainty

Common and dangerous!

More interesting...

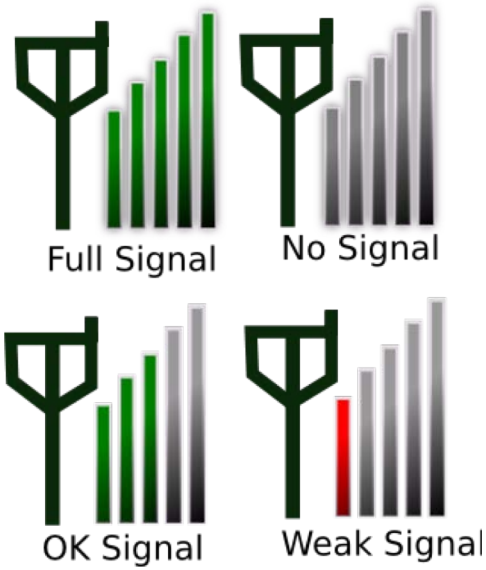
(Signal strength)




Visual indication of location uncertainty

# Seamful design in the world

- Seamful design exists out there already in places, e.g., phones
- Signal strength is exposed as a resource
  - For exploitation e.g., moving to high strength area
  - Provides resources for excuses too...



# Overview

- Two concepts for ubicomp sensing systems:
  - How **context** can be important
  - The role of **feedback** in ubicomp systems
- Other ways to think about ubicomp sensing systems:
  - “Expected, sensed, and desired” design framework
  - Designing for seams in sensor systems

# Reading

- Pritchard et al. (2015) How to Drive a London Bus: Measuring Performance in a Mobile and Remote Workplace
- Dey (2001) Understanding and Using Context
- Benford et al. (2005) Expected, Sensed, and Desired: A framework for designing sensing-based interaction
- Chalmers & MacColl (2003) Seamful and Seamless Design in Ubiquitous Computing