Usability in Pervasive Systems

307CR INTERACTIVE PERVASIVE SYSTEMS

Usability introduction/recap

- The majority (if not all) of you have studied usability in some depth during your degree.
- What do you remember about it?
  - Looks at the user and how to design systems that are accessible and easy to interact with.
  - PACT analysis
  - User studies
  - Quantitative/Qualitative data
  - Heuristics and Principles

Pervasive systems recap

- What do we mean by "*pervasive systems*"?
- Why do we make pervasive systems?
- What do/don't we like about them?
- What requirements do pervasive systems have?
  - Phenomena | Sensing | Processing | Communication | Visualisation
- How could these requirements affect usability?

Why is usability important?

- Why is usability in general important? How does it inform design?
- Why might usability be particularly important in relation to pervasive systems?
  - What are pervasive systems designed for?
  - Where are they used?
  - What do they do?
  - Who uses them?
  - What range of interfaces do we have?

…

- Pervasive systems largely deal with the collection and visualisation of data.
  - Therefore the way in which this data is presented to the user is important.
- Passive / Active pervasive systems.
  - Do all pervasive systems need interfaces?
  - What about automated context-aware systems?

Heuristics

- Due to their practical nature, the majority of pervasive systems are designed using a heuristic approach.
  - Heuristic approach is highly focused on practical development
  - Does not necessarily mean it is the most efficient approach (but better than nothing!)

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

Jacob Neilson's 10 usability heuristics of interface design

- (Do these all apply to pervasive system design?)

Different types of interfaces

- Visual feedback of the system state and/or data values
  - E.g. This could be as simple as using LED's
- Sensory feedback
  - Utilising some form of sensory feedback mechanism to inform the user of change within the system.
  - E.g. Using vibration in wearable systems
  - E.g. Audio feedback

User vs phenomena driven design

- Pervasive systems are almost entirely built to purpose. Meaning they have a bespoke job to do (such as monitor state change in a pre-determined phenomena).
  - These systems are generally designed from the phenomena forwards.
  - Temp >> sensors >> processing >> communication >> visualisation >> user.
- User centred design focuses the development process on the user and their abilities/requirements. Helping to ensure a purposeful and functional system.
  - These systems are designed from the user backwards.
  - User >> Visualisation >> Communication >> Processing >> Sensors >> Temp

- Our system and data can affect the user in a range of ways
- How?
  - Sensors may only be able to provide a certain range of values
  - Data collection and processing times
  - Communication cost (in power) and frequency (time)
  - Format for data collected
  - Accuracy
  - Types of interfaces available

- User requirements can affect our system on the lowest levels.
- How?
  - How often does the user sensibly require state change notification?
  - What response time is considered acceptable?
  - What information does the user need?
  - Does the user need to interact with the system / phenomena?
  - Can the user interact with the system to alter conditions?

…

- How can these processes effect various aspects of the system?
- How do they effect different members of your team?
  - [TASK] What roles might you expect to have within a pervasive system team?
  - What would they be responsible for?
- How does the type/format of data collected impact our system interface?

Examples of different Pervasive Computing interfaces

Powerglove

UbiBag

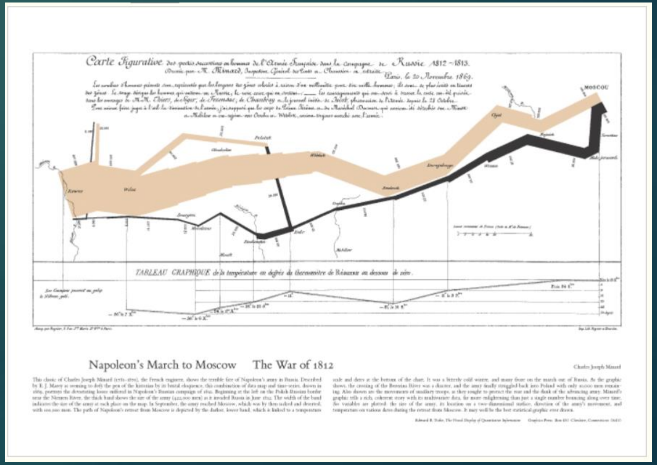
Pervasive life

More examples

- You could also look at…
  - Ring protect doorbell and security system
  - Microsoft HoloLens augmented reality
- How do these systems work?
- How does the user interact with them?
- What kind of technologies are involved?

Visualising quantitative data.

- What link do pervasive systems have to quantitative data?
- Edward Tufte is considered one of the leading experts in the visualisation of data.
- Pervasive systems can potentially collect vast amounts of data, why is it important this is designed for?
    - What can we do about it?



Front end development

- How important is this for pervasive systems?
- Should you employ a dedicated front end designer/developer?
  - What are the advantages to this?
  - Why can't someone else in the team just 'bang something together'?
- Why is it important for front end developers to understand the rest of the system?

Cloud computing and social systems

- Given the large volume of data pervasive systems can collect, is it worth utilising cloud computing practices/technology?
- What are the advantages/disadvantages to implementing cloud computing within our pervasive system?
- Can we utilise our users (or their technology) to create ad-hoc pervasive systems that give low-level detail?
  - If yes, how?

…

- [Task] How can we utilise the technology commonly found on/around people to create a pervasive system?
  - This can make use of:
    - cloud computing principles
    - social media platforms
    - mobile technology

Summary

- Given that pervasive systems are designed to deliver data to an end user, it is important that we consider how best this can be achieved.
- Pervasive computing is highly technical and we often get caught up in a phenomena driver approach to designing new systems.
- Don’t be afraid to focus on the user and hypothesise how the system will work at the beginning (although you’ll need to make it eventually!).

…

- Get used to rapidly prototyping pervasive solutions. You won't need to implement them so use your imagination.

- Think about user requirements and how this impacts your system at the lowest levels.
  - Such as sampling rate (how quickly does your user really need data?)

- Being able to understand how the system works will alter your front-end design, so if you're not sure ask a team member.