



Anna Dickinson

adickin2@inf.ed.ac.uk

## The Typical User



The Typical Computer User



**THERE YOU ARE, THE TYPICAL SMARTPHONE USER**

**Demography**

- Typically 25-34 years old
- Lives in Budapest or in a chief town of a country
- Has university, college education
- Economically active
- Is AB status

**Innovator**

On the basis of innovation skills, he/she belongs to the "innovator" and "early adopter" segment. As long as possessing an iPhone, it's in the top that he/she belongs to the "innovator" category.

**Active on the social media sites**

More active on the social media sites than the average internet users, recommends, shares personal or brands, shopping related content more often. Online shopping or the use of online methods of payment has a higher ratio among the typical smartphone users.

© 2014 HUNGARIAN TELECOMUNICATIONS AUTHORITY (HAT)

## Suspect it!

- Suspect the word "typical"
- Suspect the word "normal"
- Suspect the word "ordinary"
- If you design for what the technology community often considers "normal", you're narrowing your users right down to a small, privileged group.
- If you're going to do it, at least do it because you have thought about it and decided that's what you want.
- Better yet, don't do it!



Thank goodness for remote controls...

## Height

How tall are you?

How tall is the person next to you?

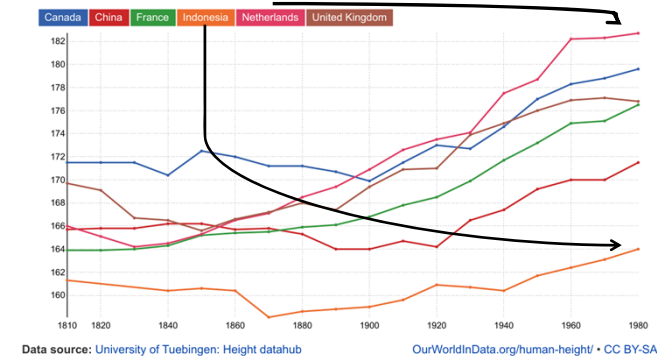
Take 10 minutes to record the height of everyone in the room.

Now average it.

(You might like also to separate it into men and women – you'll probably find there's a different average)

### Average height of men for selected countries, 1810 to 1980

Height in centimeters by year of birth.



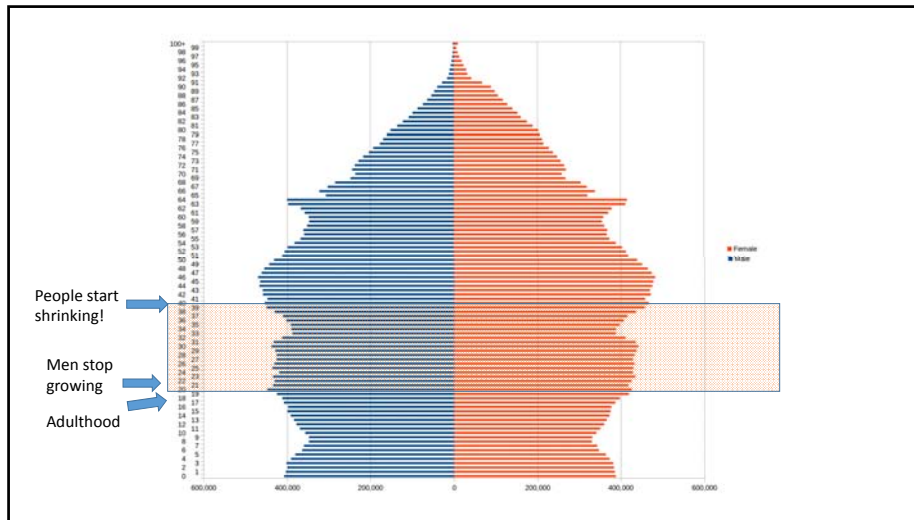
And how about...?



Students aren't average!

- Height changes with age
- Within countries where the research has been done, height is correlated with education, income and intelligence

→ Average student height is rarely a good measure of average *human* height.



## UK

- 64.1 million people (5.3 million in Scotland)
- 2014 (ONS) – 19% of population is 0-15, and 18% is over 65.
- But there are huge differences between people who are “working age”
- And: over 2 million people in UK registered with sight loss (NHS and RNIB) – 3% of the population
- 11 million people have hearing loss (1 in 6 people in UK)
- 12 million people have a disability – this includes 7% of children and 16% of the working age population.
- 6.5 million have a movement disability.
- 2.8 manual dexterity
- 2.2 million communication
- 2.5 memory/ concentration/ learning
- 4.1 physical co-ordination

## Navigation

A system to introduce tourists to a new city

System to navigate people through areas with lots of traffic and busy roads.

A system for business people in an unfamiliar city

A system safe to use in areas of (relatively) high crime

A navigation system for people with cognitive impairments (e.g. short-term memory difficulties)

Think about that. Did you learn anything?

## Spacebook

Speech-driven

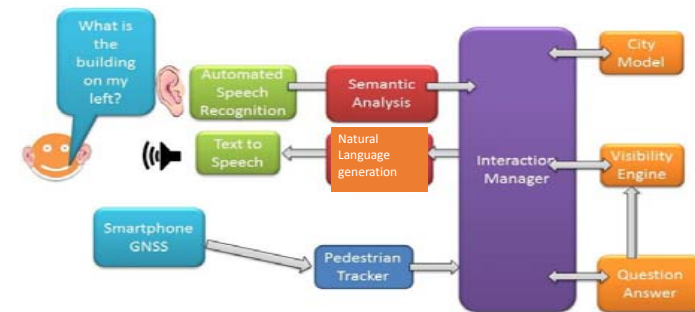
Hands-free

Eyes-free

Natural interaction



## SpaceBook: Essential Ingredients



- Interaction manager
  - 1: Dialogue control
  - 2: Responded to user requests
  - 3: System-initiated navigation actions
  - 4: Responses to user-initiated question/ answer actions
  - 5: Point of Interest push actions
- The Visibility Engine
  - Identified the entities that were in the user's vista space.
  - Basically a 2.5 dimensional representation of the city that allowed the system to judge what the user could see.

## Evaluation (the important bit)

42 participants (24 younger and 18 older).

Recruited through the university careers service

(mean: 23, range: 16-40)

Recruited from local over-50s organisations

(mean: 62, range: 52-76)

All participants were native English speakers and fit enough to walk for 90 minutes over uneven and steep terrain.

## System Variants

Evaluation aimed to: assess the contribution of **Visibility Engine** and the **Multi-threading Interaction Manager** to SpaceBook functionality (push and pull).

System	Visibility Engine	Multi-threaded Interaction Manager	
I	Yes	Yes	System with single-threaded IM had no prioritization. Less responsive to users
II	Yes	No	
III	No	Yes	

System w/out VE depended on proximal landmarks – so 'breadcrumb trail', step by step navigation

17

## Evaluation Route

- Significant route-specific landmarks (e.g. Bank of Scotland dome, the Hub).
- Mix of busy and quiet streets.
- Gradients (up and down hill).
- Points of interest, e.g. National museum, NLS, Scott Monument.

18



## Issues: Navigation

At best, approx. 50% of user utterances understood by SB – environment noisy (weather, traffic, passers-by...)

Cues delayed – esp. in Systems I & II (both of which had VE)

Landmark descriptions popular – more needed

Landmark visibility could be problematic. e.g. The Hub

91% of the navigation tasks were successfully completed; 98% for Leg 3

## The Hub?

“Can you see The Hub in front of you about 130 metres away? It has a tall clock tower.”

22

## Results Overview

Confidence and control

Interaction ease

Information content

Age effect

Average moving speed

Total moving time

Total task completion time

Stationary time

Distance travelled

Data very widely spread e.g. B18 and 20 walked close to 4 miles, whereas B12, 19 and C02 walked just over 1½ miles.

B20 took 70 minutes, but B15 completed the tasks in 27 minutes

Few statistically significant outcomes for differences between systems or groups.  
But...

23

### Differences between older people and younger people...?

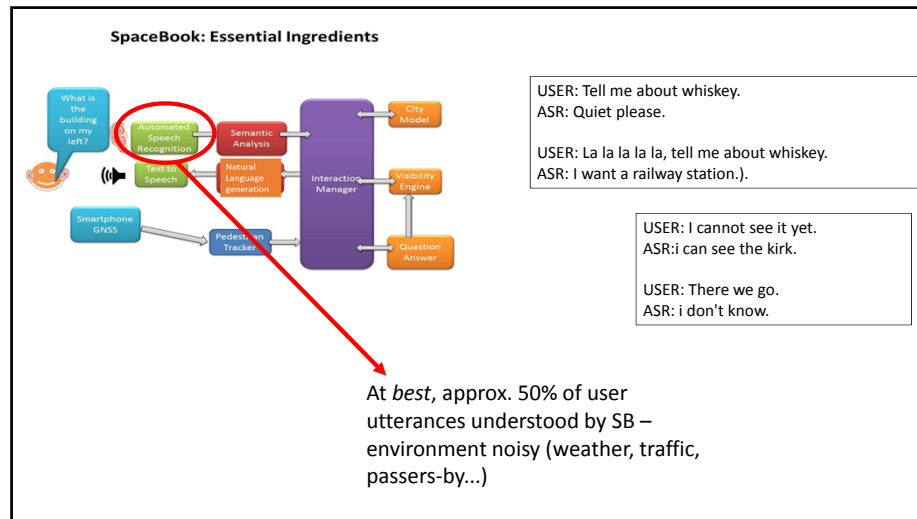
I found the task easy

I felt in control of my journey

I felt confident I was going to reach my destination

I felt confident I was going to reach my destination





## Concluding rant

**Evaluation isn't the add on to confirm everything's OK – it's a fundamental part of system development. It doesn't matter how clever or pretty the system is if people cannot use it.**

You might be making a system that is supposed to confuse people or annoy them – or one that they need to struggle to overcome (games are often those kinds of systems) but again, do it deliberately. Do it because this is what you want, don't do it because you are so focused on the shiny bits that you don't evaluate – or possibly even worse, that you do evaluate but hear what you expect rather than what the users are telling you.

Shiny isn't enough!

**If people cannot use a system, it doesn't work.**

How do you know if people can use it?

You test it. You listen. You make changes. Then you test it again.

There is an argument that says: if your evaluation finds no problems, you're not testing hard enough.

Is that as simple as it sounds?

- Of course not.
- Evaluation has its own problems – some people can't give you feedback using some methods (think-aloud, diaries) or you need to decide what they mean ("I can't use this!" / "That doesn't make sense!") and how to fix it.
- You need to enable people to be critical of your system.
- Secret: we all think we've built something perfect. None of us have (or almost none). Let your user break it – give them all the tools to break it, and *listen* to their sighs and grunts and silences as well as what they say.
- Let them break the system, and then make that system *better*.