Just what you need: Simplifying electronic devices

Steve Hodges shodges@microsoft.com
Richard Harper r.harper@microsoft.com

Microsoft Research Cambridge

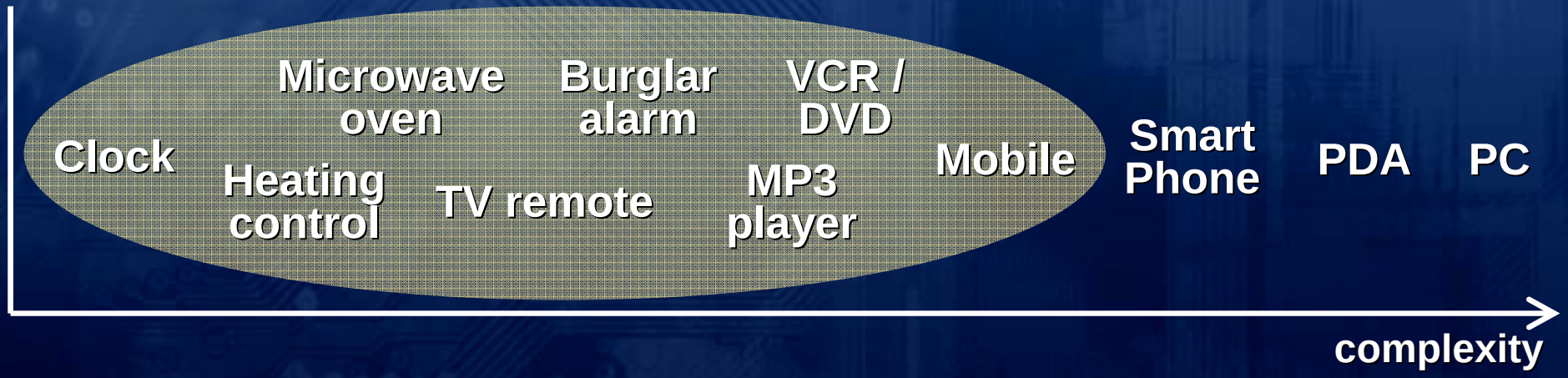
Simplifying electronic devices

- Motivation
- Previous related ideas
- Proposed approach
- Issues

Microsoft Research Cambridge

What do we want to simplify?

- Consumer electronic devices
- Powered by microprocessor but no ‘visible’ operating system
  - Capable of complex operation
  - Users not expected to modify applications



Microsoft Research Cambridge

Why do we want to simplify?

1. Intrinsic capabilities continually grow
   - More and more features are possible
   - Manufacturers feel the need to differentiate
2. Products physically hard to interact with
   - Interaction component cost dominates as. computational power cost drops
   - Miniaturisation limits space for interaction

Why can’t we just remove features?

3. Diverse consumer base expects more
   - Expectation of greater choice, more features and less cost

Microsoft Research Cambridge

Previous approaches: 1. Simplifying intrinsic capabilities

- Mobile phones
  - Menu systems
  - Speed dial shortcuts

- Remote controls
  - Sony dual-mode reversible remote

Previous approaches: 2. Enhancing physical interaction

- VCR/DVD players
  - Limited real estate on front-panel
  - Alleviate with remote control and on-screen displays

- iPod (especially shuffle)
  - Desktop computer used for setup
  - Device interaction limited

Previous approaches: 3. Extending consumer choice

- Mass customisation
  - Mass production of tailored products
  - Selection of options during manufacturing process
  - E.g. cars, bicycles, Dell computers

- In-use cosmetic upgrades
  - Snap-on cover
  - Downloadable ring tones and logos
  - Skins for computer applications

Microsoft Research Cambridge

Simplifying electronic devices

- Motivation
- Previous related ideas
- Proposed approach
- Issues

Microsoft Research Cambridge

Just what you need: Extending mass customisation

- Make aspects of operation configurable
  - Define the basic personality of the device
  - Map buttons and controls to common operations
  - Completely remove unwanted features, specify default settings
  - Allow personalisation of appearance
- Build more complexity into a device
  - Support more features across the consumer base
  - Remove complexity from any one user's perspective
- New *configuration* phase in product life-cycle
  - Can be done at point of sale or by consumer

Just what you need: Customisation is different from use

- Configuration is specified using a rich UI
  - Desktop computer is natural choice
  - Custom application or manufacturer's web site
  - Allows infrequent and one-time-only settings to be made easily
- Configuration is then transferred to device
  - Memory key, USB, Bluetooth, Zigbee etc.
- Frequently-changed settings on-device
  - e.g. washing machine programme selection

Microsoft Research Cambridge

Issues – extent of customisation

- What level of customisation is best?
  - Downloading cosmetic personalisation data
  - Tweaking infrequently set parameters
  - Mapping between controls and functions
  - Support for different personalities/modes
  - Full support for development of new applications
- What range of devices should be targeted?
  - Create a path to cheap ($10 retail) devices
  - User-centric look-and-feel across devices
- Can an generic framework be developed?
  - Support for different types of device
  - Agnostic to implementation technology

Example: outdoor sports watch

- Cosmetic personalisation
  - Custom text content and style, graphics...
  - Custom colours, alert sounds…
- Infrequent parameters
  - Analogue/digital; 12/24hr
  - Time-base for calculating average pace
- Control mappings
  - Which (& how many) items are displayed
  - How to control stopwatch, split times etc. (if at all)
- Personality selection
  - Hiker vs. runner vs. cyclist
- Complete programmability
  - How to switch between personalities on the device
  - Constructing a new personality for Triathlon competitors

Garmin
Forerunner 201

Issues – technical implementation

- Definition of configuration options
  - Custom-built application/web interface?
  - Fine-grained control language or coarser XML parameter list?
- Transfer of configuration information to device
  - Device may not be portable (or near a PC)
  - Hardware interface needs to be very cheap
  - Limited ability to handle protocols
- Representation and interpretation of configuration
  - Minimal ability to interpret on device
  - Options include compiled code, mapping tables etc.

Microsoft Research Cambridge

Issues – market forces

- Consumer electronics marketplace diverse
  - Much less standardisation than e.g. PC market
- Opens new opportunities for manufacturers
  - Differentiation for early-adopters
  - New features without necessarily burdening UI
  - Possibility to charge for software-only feature upgrades
  - A new ‘configuration’ after-market
- Need to understand what motivates them
  - Many demonstrate very bad design practice!
  - May commoditise hardware

Microsoft Research Cambridge

Issues – user experience

- Maintaining quality of configurations
  - Goal is to make devices *easier* to use!
  - Possibly offer a limited set of configurations
  - Interaction design critical!
- Device documentation
  - Hope to reduce the reliance on a manual
  - Manual can be printed at configuration time
- Device labelling
  - Familiar form ≠ familiar interface!
  - Legends can be created at configuration time

Microsoft Research Cambridge

Just what you need: Summary

- Addition of a ‘configuration phase’ to life-cycle
  - Leverage previous approaches to ease-of-use in more generic way
- Much more work to explore further
  - Need to consider very large range of possibilities
  - Build prototype devices and try them out

Microsoft Research Cambridge

Microsoft Research Cambridge