Context-Awareness, Disappearing and Distributed User Interfaces

Experience, *Open Issues and Research Questions*

Albrecht Schmidt
Lancaster University, UK
albrecht@comp.lancs.ac.uk

Disappearing UI

- User Interfaces that are not seen as an “access point” to computing/control
- Interaction that is perceived as the normal way to get something done
- The situation/what the user does already describes the input to the system
- Term: Implicit HCI
- Anticipatory systems – do they need to be smart? Can they be smart enough?
Distributed UIs

- User Interfaces are everywhere - at least a part of them
- Alternative UI – same function is offered in different UI components
- Compound UI – a function is distributed over several UI components
- Distribution of UIs over physical space and modalities
- Application-to-Application transition – what is an application?

Experience I

TEA-Phone

- Mobile Phone that recognizes context using sensors and reacts accordingly
  - In pocket, in users hand, …, meeting
  - User specifies the reaction given a certain context
  - See http://www.teco.edu/tea/
- Problem: how to evaluate with “real people” in a non generated situation?
  - How often does it happen the phone rings when it is in your hand?
  - How much added value is necessary to make something successful – how to find this out?
- How to recover if you get it wrong? (and you will get it wrong sometimes!)
Experience II
TEA-Wearable

- Distribution of sensing technology over the body
- Contexts
  - walk
  - go up / down stairs
  - run
  - stand
  - sit
- Implicit input to a wearable computer
- UI dilemma – stability vs. plasticity
- Validity in time?
- Order of contexts (which is more important than the other?)

Experience III
Proactive Environment

- Calendar application
- Used contexts
  - People, time, activity
- Different view
  - Work, artistic, leisure
- Distribution of sensors
- Building the application vs. building the sensor network

Problems:
- Debugging?
- How to split work?
- Separation of concerns?
- What are components?
- How to match context? (between provider and consumer)
Experience IV
Context Phone Book

- A phone book on a mobile phone that includes information about
  - Connection state (phone off, answering machine, busy)
  - Whereabouts (home, at work, on the road)
  - Acceptance for a call (red, yellow, green)
  - Integration with making the phone call
  - See [http://www.comp.lancs.ac.uk/~albrecht/contextphone/](http://www.comp.lancs.ac.uk/~albrecht/contextphone/)

Problem:
- Blur between applications for the user but not for the developer!
- usefulness vs. privacy (level of abstraction)

Experience V
Aware Goods

- Recording context in a supply chain
  - Temperature
  - Moisture
  - Magnetic field
  - Shock, Vibration
  - See [http://www.comp.lancs.ac.uk/~albrecht/awaregoods/](http://www.comp.lancs.ac.uk/~albrecht/awaregoods/)

- Context is a proof for the insurance
  - Security
  - Integration, Standards?
Context

- Abstract representation of the situation
- Facts that matter for an application/user/device
- Often inherently connected to time and location → matter most at a certain place and at a certain time
- IMHO up to not very well understood

- Context ≠ Event
- Context ≠ Location

Describing Context

- Context = (description, unit, domain, probability) that is bound to
  - time of origin,
  - location of origin
  - distribution-rules

- Ist it just “Knowledge Representation”?
- Natural Language, e.g. Ontologies?
- Dimensions are useful? What are they?
- Describing the whole world?
- How to restrict the domains and keep it open at the same time?
Architecture(s)

- Separation of concerns
  - Context producer
  - Context consumer
  - (Context abstractor)
- “garbage collection” for contexts
- Distribution, Communication models
- Support for the programmer
  - Push vs. pull

Distribution

- Local and temporal validity of context?
  - Application dependent
  - Sensor dependent
- Of special interest
  - Co-location
  - Simultaneous (close in time)
- Ordering
Perception / Sensing

- Perception (if not trivial) is not 100% reliable
- New cost measures for perception
  - Calibration
  - Power
  - Constraints on physical design (e.g. window for a light sensors, opening for a microphone)
  - Setup time
  - Reliability
  - Unobtrusiveness – acceptability
- Value for cost

Debugging the Systems
Simulation

- Can I debug the sensors system without the application?
- Is the sensor infrastructure exchangeable?
- Can I debug the context-aware application without the sensor infrastructure?
- Can I simulate the sensor infrastructure

- Figure out if context usages makes sense for an application using an optimal sensor network (e.g. simulation of context sensing by humans)?
Evaluation

- “Lesson 4: Only if building something allows you to explore its full design potential”
  (from the talk by Roy Want: http://www.inf.ethz.ch/vs/events/dag2001/abstracts/roy.html)

- So how to evaluate something (e.g. fragile prototypes) that only works under lab conditions but is designed for other user groups?

- Is the “living lab” approach valid? – Is it still science?, How to deal with the bias? If not is there another way?

- Often Ubicomp systems are build to provide the user with a certain experience or to make a task more pleasant and not necessarily more efficient! This is often highly subjective! Similar issues are found in design and art.

- Is the evaluation “of parts of a Ubicomp system” with the methods used in other field (HCI, networks, AI etc) enough/useful/appropriate?

Cross-Field Developments

- “Interwoven Engineering”
  - Hardware
  - Communication
  - Software
  - User Interface
  - Design

- Often issues are not new in each of the fields but still the combination is hard and potentially more than the sum

- Sometimes researcher in Ubicomp have a really hard problems from other subjects (e.g. AI speech recognition) as a minor sub-problem in their system (and sometimes don’t realize it).

- Is it possible to define useful interfaces between these fields – how to work together?

- Moving borderlines – e.g. when the software works it is rebuild in hardware