Ubiquitous Computing

Daniel Popescu

October 3, 2006

Ubiquitous Computing: Motivation

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it - Mark Weiser The Computer for the 21st Century, Scientific American, Sept 1991, pp. 94-104.

Example: Writing, electricity,

Ubiquitous Computing Motivation:

- Desire to break the human away from desktop-bound interaction.

The Ubiquitous Computing Vision

- Vision of people and environments augmented with computational resources that provide information and services when and where desired
- Assisting everyday life (not overwhelming it)
- New types of application will emerge from this environment

The Ubiquitous Computing Vision

- Vision of people and environments augmented with computational resources that provide information and services when and where desired
- Assisting everyday life (not overwhelming it)
- New types of application will emerge from this environment

Past to present

Main interaction themes

- natural interfaces
- context-aware applications
- automated capture and access

Scale

Scaling is major issues for ubiquitous computing

- Devices
  - PDA
  - Tablet PC
  - Electronic Whiteboards
- Distribution of computing
- People
- Time

Computing with Natural Interfaces

- Replace the GUI paradigm
- Humans speak, gesture, and use writing utensils to communicate with other humans and alter physical artifacts
  - high learnability
  - general ease of use
  - usable for people with disabilities
- Past Research
  - speech-related interfaces
  - pen-based / free-form interaction
  - graspable / tangible user interfaces

Challenge: First-Class Types in Interactive Development

What kind of operations could be done with the natural data type *ink*?

- recognition techniques from ink to text

but also useful are:

- uninterpreted ink data
- structural gestures

Therefore, Abowd and Mynatt propose that audio, video, ink and sensor input need to become first-class types.

These first class types should be accessible through programming toolkits.

Challenge: Error-Prone Interaction for Recognition-Based Interaction

Even humans make mistakes when they deal with e.g. recognizing hand-writing

Assumption: It is likely that computer handwriting recognition will never be perfect.

Therefore, research should deal with identifying the best ways to deal with errors

- *Error Reduction*: Improving recognition technologies
- *Error Discovery*
  - thresholding of confidence measure
  - historical statistics
  - explicit rules
- *Reusable infrastructure for error correction* Library

Context-Aware Computing

Ubiquitous computing application need to be *context-aware*, *adapting* their behavior based on *information sensed* from the physical and computational environment.

Historical Examples:

- Location
  - car navigation
  - handheld "tour guide"
- Recognizing Individual Objects
  - Tags
  - Vision-Based

Design Challenge: To create a believable experience with context-aware interfaces, responsiveness of the interface is required. (Example: "seeing inside" walls)

Defining Context

Abowd and Mynatt provide the *five W's* to define the minimal requirements of context.

- *Who*: particular user, people in the environment
- *What*: interpretation of the user's current activity
- *Where*: Location
- *When*: index captured record, duration of an activity, relative changes in everyday life
- *Why*: Why is a person doing something (e.g. start by measuring heart rate, temperature)

One Context Challenge: Context Fusion

A single source for context information does not exist. (e.g. GPS does not work for indoor location services)

Therefore, handing off of sensing responsibility between different context services is needed.

Benefits of multiple context sensors:

- reduce noise in the signal
- provide better coverage

Automated Capture and Access to Live Experiences

One common daily activity is

- listening to and recording events that surround us
- trying to remember and access those events

$\Rightarrow$ Automated tools could remove the burden of recording for the human.

Examples:

- Tivoli system (team meetings)
- Classroom 2000 (university lectures)

In all these systems capturing is moved aways from traditional interface devices.

Challenges in Capture and Access

Capture - Move capturing to new domains:

- Recoding special events like vacation, birthday, ...
- Capture design decision rationale
- Capture construction of a building
- Derive information instead of only recording raw data

Access:

- Automatic Theme Detection in data
- Synchronization of multiple captured streams
- Privacy concerns in public capturing settings

Future: Everyday Computing

Motivation: Support the informal and unstructured activities typical of much of our everyday lives.

Everyday Computing requirements

- They rarely have a beginning or end (no closure for task)
- Interruption is expected
- Multiple activities operate concurrently (e.g. monitoring a background activity)
- Time is an important discriminator (e.g. regaining working state since last interaction)
- Associative models of information are needed (e.g. information like emails have different viewpoints, information are connected to the presents of people)

Challenges in Everyday Computing

- Continuously changing user context
- Design a continuously present computer interface
  - Information Appliance - sitting dumbly in the background
  - Continually worn interfaces
  - sensor networks
- Presenting information at different level of the periphery of human attention
- Connecting events in the physical and virtual worlds
  - Virtual Space: E-Mails, Web,
  - Physical Space: Books, Face-to-face interactions

Challenges in Ubiquitous Computing

- Evaluating Ubiquitous Computing
- Social Issues for Ubiquitous Computing

Evaluating Ubiquitous Computing

Evaluation of Ubiquitous Computing systems is difficult, because

- No reliable system exist to evaluate
- A compelling story is often missing
  - The technology must serve a real human need
- System might not have been built
  - *Feasibility study* of cutting-edge applications is needed to avoid wasted engineering effort.
- Evaluation cannot be performed in a lab, since everyday routines should be observed.
  - Classroom 2000
  - Aware Home
- Task-Centric Evaluation Methods are inappropriate

Social Issues for Ubiquitous Computing (1)

Ubiquitous Computing Infrastructure makes it easy to spy on people.

Since information can be anywhere in the system, security is important.

Lack of Knowledge. Users are afraid that information gets recorded "behind their back," since the computing devices become invisible.

Solution Ideas:

- Provide indicator for recording devices
- Allow recorded persons to set their visibility

Challenge:

- Find acceptable policies for erasing or forgetting captured information.

Social Issues for Ubiquitous Computing (2)

Recording a lecture can have impact on the attendees.

- Attendees are aware of the capturing and are afraid of saying something that they would regret.

Weaknesses of the Presented Paper

- Toolkit support is presented as an important tool, but it is feasible to build a generic ubiquitous computing toolkit?
- Maybe wrong assumption: Why is a pen a more natural interface than a keyboard?
- The paper is a survey paper, but the presented examples are mostly work of the authors.

Strengths of the Presented Paper

- Good overview of research field (Past, present and future)
- Research Area driven by user needs
- Strong emphasis on evaluating the usefulness
- Relevance to Embedded Systems

Relevance to Embedded Systems

Clear formulation of requirements for embedded systems.

- Suggestions for concrete needed toolkits

Ubiquitous computing was especially created to *drive the development* of devices and infrastructures. These devices are *heterogeneous embedded systems*, since a motivation was the desire to break the human *away from desktop-bound interaction*.

Thank You

Thank you for your attention.

Discussion