Context- and Location-awarenss

CBU ICT Summer School 2009

Patrik Floréen, HIIT

Course owerview

- 2 lectures by Patrik Floréen on context-awareness
  - 4 hours
- 2 lectures by Petteri Nurmi on location-awareness
  - 4 hours
- Group exercises
  - 8-10 hours
- 1 ECTS points (+1 more with study diary)

Objective and outline of the lectures on context-awareness

- The objective of these lectures is to give an introduction to context-awareness, including context-aware applications and methods required in their design
- Context-awareness is part of what is called Ubiquitous computing or Pervasive computing

INTRODUCTION

Some important "classical" publications

- A. K. Dey and G. D. Abowd, "Towards a better understanding of context and context-awareness," The What, Who, Where, When, Why and How of Context-Awareness Workshop at the Conference on Human Factors in Computing Systems (CHI), 2000
- G. Chen and D. Kotz, A survey of context-aware mobile computing research. Technical Report TR 2000-381, Department of Computer Science, Dartmouth College, November 2000
- J. Pascoe, N. Ryan and D. Morse, "Issues in developing context-aware computing," in Proc. Intl. Symposium on Handheld and Ubiquitous Computing, LNCS 1707, Springer, 1999, 208-221
- B. N. Schilit, N. Adams and R. Want, "Context-aware computing applications," in Proc. Workshop on Mobile Computing Systems and Applications, IEEE, 1994, 85-90

Important journals and conferences

- Some journals:
  - IEEE Pervasive Computing
  - Personal and Ubiquitous Computing (Springer)
  - Pervasive and Mobile Computing (Elsevier)
- Some conferences:
  - Pervasive: International Conference on Pervasive Computing
  - Ubicomp: International Conference on Ubiquitous Computing
  - PerCom: Annual International Conference on Pervasive Computing and Communications
  - CHI:International Conference for Human-Computer Interaction
  - MobileHCI: International Conference on Human-Computer Interaction with Mobile Devices and Services

Activities in context-awareness at HIIT

- Research in this area is mainly undertaken in the Adaptive Computing, CoSCo and Ubiquitous Interaction research groups
- The course teachers are from the Adaptive Computing research group
- We have a project called PUPS: Personalized Ubiservices in Public Spaces from the Tekes Ubicom programme and participate in LUCRE: Local and User-Created Services in the ICT SHOK programme Flexible Services
- Some of our software: BeTelGeuse (betelgeuse.hiit.fi), Capricorn (universe.hiit.fi/capricorn)

MOTIVATION AND HISTORY

Ubiquitous computing

- Mark Weiser (1952-1999, Xerox PARC) presented a vision of ubiquitous computing ["The Computer for the 21st century," Scientific American, 1991]
- Anywhere, anytime, for everybody
- Computers are in the background; we do not notice them

Foundation for Weiser's vision

- Most successful technologies are those that recede into the background and become unannounced
- Technological development enables a new model of computation
- Compare:
  - Mainframe: many users per processor
  - PC: one user per processor
  - Ubicomp: many processors per user

Other similar buzzwords

- Pervasive computing
  - Spread out everywhere
- Proactive computing
  - Anticipating the user's needs
  - Predicting changes in environment and act in advance
  - In contrast to reactive computing
- Context-aware computing
  - Described in a moment
- Adaptive computing
  - Described in a moment
- Ambient intelligence
  - Used by the European Commission for the same things

Why context? (1/2)

- In human communication, situational info is implicit and increase the amount of information available
- The goals of using context-awareness it to make interacting with computers easier. Context-awareness can be seen as an enabling technology to help other applications
- Context must be derived to a large extent automatically. Most users would not even know what info is relevant

Why context? (2/2)

- The limiting factor of computing is often not processor speed or memory size, but the limits of human attention. The user must get out of the information overload and continual interruptions
- Especially in mobile computing, situation changes frequently and the user is preoccupied with other tasks and has not the possibility to provide lots of explicit input and has difficulties to notice or digest the output

History

- The first research investigation of context-aware computing was the Olivetti Active Badge mentioned in Weiser's article [Want et al., ACM Transactions on Information Systems, 1992]
- Schilit and Theimer [IEEE Network, 1994], and Schilit, Adams and Want introduced the term context-aware
- According to Pascoe, Ryan and Morse, the ancestor to all work on context services is Schilit's system architecture, described in his PhD Thesis [1995]

CONCEPTS

Formal definitions of context

- Tacit understanding of "context"
- Explicit definitions are often by examples of contexts or synonyms, but that is by many not regarded sufficient
- Longman Dictionary of Contemporary English: "The surrounding conditions in which something takes place"
- Next we look at the definitions of Dey and Abowd, and Chen and Kotz
- Sometimes, as in [Lieberman & Selker, 2000], context is everything but explicit input and output. Lieberman and Selker includes in context the internal state of the system, leading to a form of self-awareness
- In [Barkhuus, 2003], context-awareness is an application's ability to detect and react to environment variables

Context according to Dey and Abowd (1/2)

- "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves"
- Dey and Abowd include in context also explicit information given by the user, as it does not matter, for instance, whether a user's identity is detected implicitly or given explicitly
- This is the definition we have seen used mostly and use ourselves

Context according to Dey and Abowd (2/2)

- They talk about primary context (location - where, identity - who, time - when, activity - what) and secondary context (e.g. a person's email address that can be found on the basis of the primary context)
- "A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task"
- The system tries to answer the "why" and do some action accordingly

Context according to Chen and Kotz

- They want to make a distinction between what is critical (characteristics determining the behaviour) and what is relevant, but not critical
- "Context is the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user"
- They call the first case (the critical one) active context and the second (the relevant one) passive context

Types of context (1/3)

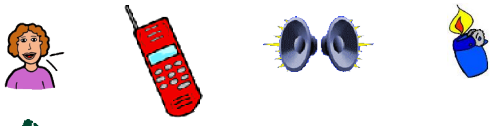
- Schilit, Adams and Want say that there are 3 important aspects of context:
  - Where are you?
  - Who are you with?
  - What resources are nearby?
- Gross and Specht [2001] list 4 dimensions of context:
  - Location
  - Identity (user's interests, preferences, knowledge and activity logs)
  - Time (e.g. working hours)
  - Environment or activity

Types of context (2/3)

- Mayrhofer [PhD thesis, 2004] lists aspects of context:
  - Geographical (e.g. country, street, building, floor, office)
  - Physical (e.g. lighting, noise level, temperature, acceleration, tilt)
  - Organisational (e.g. institution, department, group, project)
  - Social (e.g. family, friend, co-worker, married, single)
  - Emotional (e.g. heart rate, skin conductance)
  - User (e.g. profile, location, capabilities, role, access rights)
  - Task (e.g. documenting, programming, building a house)
  - Action (e.g. typing, reading, walking, sitting, talking)
  - Technological (e.g. connectivity, network bandwidth, network latency)
  - Time (e.g. time of day, weekday, week, month, season)

Types of context (3/3)

- There is also the time dimension of context
- Lieberman and Selker include explicitly the history of user-computer-environment interaction
- More broadly, one could consider context to include the full scale of past/historical, present/current and predicted/future context



Dourish interactional view on context

- Paul Dourish criticizes in his paper "What do we talk about when we talk about context" (2004) the representational definitions above
- His point is that context is a relational property between objects or activities
- Context arises from the activity, is dynamical, and is relevant to the particular setting, particular instances of action and particular parties to that action

Some other important concepts

- Adaptation is the process of changing the content, behaviour and/or appearance of an application or service towards a convenient user experience in reaction to a change of context
- Profile is a "persistent" set of data about an entity
- Personalisation is the appliance of adaptation based on personal profiles and preferences in order to make usage easier and the perception of information services more pleasant to users

MAKING USE OF CONTEXT

Nature of context data (1/2)

- Raw context data (physical and logical "sensors"): Sensor data, existing information and explicit usage settings, user and task models, state of equipment and user interaction, network status, …
- Context is both real and virtual
  - Example: virtual tour at tourist site
  - Virtual entities can break the laws of physics (to be remembered in conditional rules)
- Context is both explicit (given by the user) and implicit
- Context can be discrete or continuous
- Of all available data only some is contextually relevant; the context is different for different applications

Nature of context data (2/2)

- Context data can be incomplete, incorrect and inconsistent
  - Conflict resolution, prioritization, probabilistic reasoning
- Context data is heterogeneous
  - Combination is non-trivial
- Context is dynamic: it is changing (but often gradually and semi-predictably), especially in mobile applications
- Context information itself can be considered hierarchical, as raw context data can be further mapped into higher-level categories through context interpretation,use of ontologies, and so on
- This higher-level context data can then in combination with other context data be further used for inference

Context usage

- Classical view: input -> processing -> output
- Nowadays many systems are dynamical systems, so the output is also input, as noted by Lieberman and Selker [IBM Systems Journal, 2000]
- First we must acquire the context
- Then we can use raw context data to infer higher-level context
- Dey and Abowd have three context-aware categories:
  - Presentation of info and services
  - Automatic execution of a service
  - Tagging of context for later retrieval
- For automatic execution we can use rule-based systems, Bayesian networks, neural networks

Contextual presentation

- Contextual presentation:
  - Content adaptation based on user, device and network data
    - Information abstraction (e.g. compression)
    - Modality transformation (e.g. text to audio)
    - Data transcoding (e.g. GIF image to JPG image)
    - Data prioritisation (ordering according to relevance / interest)
    - Purpose classification
    - Text summarisation
  - Personalisation
    - Collaborative filtering (recommendation systems), which is a type of social navigation
    - Content-based personalisation

Context history and sharing

- Use can be made of context history
  - For instance, prediction according to what the user has done before in a particular situation
- Context can be shared, also context history can be shared
  - For instance, prediction according to what others have done in the same situation
- History information is also needed for detecting change and for being able to do learning
- So we have to store some context data, but on the other hand, we cannot store everything
- This is an open question: what to store and where, so that it can be retrieved efficiently, and useless or redundant information is discarded

Active and passive context awareness

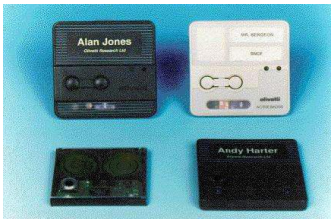
- Chen and Kotz give two ways to use context:
  - Active context awareness: an application automatically adapts to discovered context, by changing the application's behavior
  - Passive context awareness: an application presents the new or updated context to an interested user or makes the context persistent for the user to retrieve later
- This is opposed to the basic setting of customisation / tailoring, where the user specifies his own settings for how the application should behave in a given situation

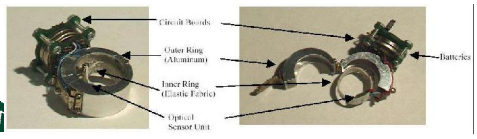
DEVICES

A multitude of devices can be used

- RFID-tags
- Mobile phones, wrist computers, palmtops, laptops
- Audio recorders, video cameras
- Robotic hands that “feel”
- PCs, servers, disposable computers…
- Example below: load sensing furniture

Examples: Active Badges / Sensors attached to finger





Types of sensors

- Mayrhofer lists in his thesis an abundance of sensors:
  - Light (brightness)
  - Vision, i.e. cameras and video cameras
  - Audio, i.e. microphones
  - Acceleration
  - Location: GPS, GSM cells, WLAN, Bluetooth, RFID
  - Orientation: gyroscopes, magnetic field or tilt sensors
  - Proximity: Bluetooth, WLAN, RFID, touch sensors
  - Environmental conditions: temp., humidity, air pressure
  - Force
  - Bio-sensors
  - Identity: iris scanning, fingerprint sensors, and other “biometric” systems; RFID, bar codes, infrared badges,…
  - State change: switches for lights on/off, …
- In addition: information sensors (not a device)

APPLICATIONS

Application domains

- Context-aware systems are developed for, e.g.,
  - Smart spaces
  - Smart everyday objects
  - Wearable computing
  - Mobile computing

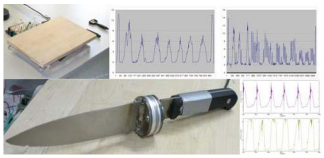
Smart spaces (1/2)

- Smart spaces are ubiquitous computing environments that encompass physical spaces: homes, offices,…
- Example smart space projects:
  - Adaptive House (University of Colorado at Boulder)
  - Aware Home (Georgia Tech)
  - Easy Living (Microsoft)
  - MavHome (University of Texas at Arlington)

Smart spaces (2/2)

- In smart spaces, context information comes from sensors (touch, audio, video etc.) placed in particular places or attached to objects
- Hidden Markov models, Bayesian networks etc. can be used for activity recognition in smart spaces
- This has been used for task prediction to select the remote control UI [Desai et al., 2002; Isbell et al., 2004]

Smart everyday objects



- A cup that knows its movements and whether there is contents in it [Gellersen et al., 2002]
- A cutting board and a knife that both can detect what is cut [Kranz et al., 2007]

Wearable and mobile computing

- Accelerometers, galvanic skin response and temperature sensors have been used to detect behaviour like walking, sitting, waving arms and climbing stairs
- Naïve Bayesian networks have been applied to audio data to recognize the moving behaviour and the kind of music a user is listening to [Korpipää et al., Pers. Ubiq. Comp. 7, 2003]

Mobile applications

- Users of mobile devices are likely to be occupied in other activities: precision is more important than recall [Jones & Brown]
- Rule of thumb: Mobile users must be able to access information in the time that they can hold their breath
- In mobile situations, the environment changes fast (⇒ dynamic context). However, the relevant context information is usually spatially and temporally close to the application host [Huang, 2002]

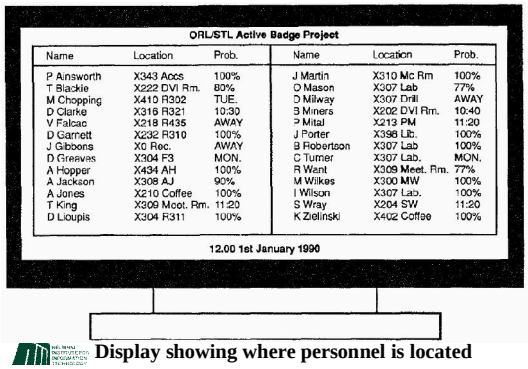
Active Badge (1989)

- Mark Weiser's vision
- Application area: How to locate a person, e.g.,
  - At a hospital: a doctor needed for an emergency
  - In an Office: a person missing from a meeting
- Solution 1: Pager or Beeper, requires person to call back
- Solution 2: **Active Badge** (Olivetti, AT&T Labs, Cambridge, ...)
  - Emits unique code every 15 s (IR signal)
  - Code picked up by sensor network in the building
  - Master station polls sensors for "sightings"
  - Thereby provides location of users

Active Badge, cont.

- www.cl.cam.ac.uk/research/dtg/attarchive/ab.html
- Initial application
  - A telephone receptionist is able to forward incoming calls
- Other
  - Automatic routing of phone calls
  - (Simple prediction of where a person is going)
- "Problems"
  - Technical: battery power, communication, reliability
  - Solution: short signals, every 15 s, automatically switches off
  - Social: privacy concerns
    - Calls forwarded to places they should not have been
    - Misguided management could misuse the data ...
    - Solution: Badge owners could just take it off and leave it on their desk

Active Badge, cont.



ORL/STL Active Badge Project

Name	Location	Prob.	Name	Location	Prob.
P Ainsworth	X343 Accs	100%	J Martin	X310 Mc Rm	100%
T Blackie	X222 DVI Rm.	80%	D Mason	X307 Lab	77%
M Chopping	X410 R302	TUE.	D Milway	X307 Drill	AWAY
D Clarke	X316 R321	10:30	B Miners	X202 DVI Rm.	10:40
V Falcao	X218 R435	AWAY	P Mital	X213 PM	11:20
D Garrett	X232 R310	100%	J Porter	X398 Lib.	100%
J Gibbons	X6 Rec.	AWAY	B Robertson	X307 Lab	100%
D Greaves	X304 F3	MON.	C Turner	X307 Lab.	MON.
A Hopper	X434 AH	100%	R Want	X309 Meet. Rm.	77%
A Jackson	X308 AJ	90%	M Wilkes	X300 MW	100%
A Jones	X210 Coffee	100%	I Wilson	X307 Lab.	100%
T King	X309 Meet. Rm.	11:20	S Wray	X204 SW	11:20
D Lioupis	X304 R311	100%	K Zielinski	X402 Coffee	100%

12.00 1st January 1990

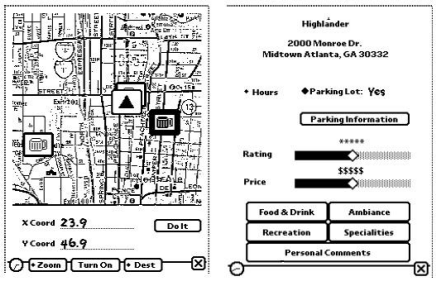
Display showing where personnel is located

Cyberguide (1997)

- www.cc.gatech.edu/fce/cyberguide/
- Mobile context-aware tour guide (indoors; outdoors)
- Georgia Tech
- Apple MessagePad

- Map of the entire research centre/campus
- Information in web pages stored locally (research demos)
- Communication (TCP/IP: web pages, email)
- Positioning by infrared beacons hanging in the ceiling (indoor); GPS (outdoor)
- Storing historical data (what has been visited, how long)

Cyberguide, cont.



Cyberguide map and information (outdoor)

Activity recognition

- Examples:
  - Detecting movement from accelerometers and other sensors (own work)
  - Powerline event detection (Patel, 2007)
  - Detecting human movement by air pressure sensing (Patel, 2008)

ARCHITECTURES

Terminology

- Architecture = overall structure, logical components, and the logical interrelationships of a system
- Library = generalized set of related algorithms (focus on code reuse)
- Framework = basic structure for a certain class of applications (focus on design reuse)
- Middleware = software connecting different components or applications, sits “in the middle” between applications and operating system
- Toolkit = large number of reusable components for common functionality (builds on frameworks)
- Infrastructure = well-established, pervasive, reliable, and publicly accessible set of technologies that act as a foundation for other systems
- Service infrastructure = middleware technologies that can be accessed through a network

Overview

- The architecture deals with issues like
  - Context gathering
  - Context representation of data between components
  - Context storage and sharing
  - Context lookup service or broker
  - Context inference
  - Relation to application logic and user i/o
- Morgan and Dourish [HCI 16, 2001] say that usually either blackboard approaches or widget based approaches are used
- Winograd [HCI 16, 2001] lists as models widgets, networked services and blackboards
- Component architectures, such as CORBA

Context gathering

- Raw context can be obtained from devices and databases (e.g. calendars)
- Semantics of gathered data important

Context representation (1/2)

- How to provide context data to other components
- Concerning the contextual model, i.e., the representation of context, different data structures have been used (see Chen and Kotz)
- Generally, context is described by a set of attributes

Context representation (2/2)

- Context is name/value pairs, with the rules for determining whether two values match
- Brown points out that attribute values like "none", "any" and "not-working" are needed, to be able to get all desired effects [Brown, Personal Technologies 2, 1998]
- Some use predicates for the context, e.g. Location(Chris, entering, room 3231), such as the Gaia project, which also uses logical reasoning with ontologies
- XML can be used to provide a structured representation

Context lookup service or broker

- The main methods to disseminate context information are:
  - Explicit query when the application needs information
  - Polling, i.e. synchronous retrieval of context information
  - Event-driven messaging: subscriptions and notifications
- More generally: the dissemination can be either 1-way or 2-way and either synchronous or asynchronous

Examples of middleware for context-aware systems

- RCSM [Yau et al., Pervasive Computing 1, 2002]
- Confab [Hong & Landay, MobiSys 2004]
- GAIA – for smart spaces [Campbell et al., Univ. of Illinois at Urbana-Champaign]
- CORTEX [Blair et al. at Lancaster University]

Context Toolkit

- Salber, Dey and Abowd [CHI'99, 1999] have a "context toolkit": a context widget gets a certain type of context info and a context server gathers the context about an entity (e.g. a person) from the available context widgets, behaving as a proxy to the context for applications
- A thorough description (70 pages) of this system is available in the special issue on context-aware computing of Human-Computer Interaction [2001]
- In this same issue, the article by J. I. Hong and J. A. Landay, ["An infrastructure approach to context-aware computing"] argues for a service infrastructure (next slides)

Benefits of a service infrastructure

- A greater variety of devices and applications can be used, because of independence from hardware platform, operating system, and programming language
- Sensors and services can be upgraded independently of one another and dynamically while the system is running, as the middleware layer separates them from each other
- Devices can be simpler, as they can share sensors, processing power, data and services

Challenges for a service infrastructure

- Designing data formats and network protocols to allow interoperability and supporting different types of sensors and context
- Building the basic services in the infrastructure
- Finding the right balance of responsibilities between devices and infrastructure: smart devices vs. smart infrastructure
- Scoping of sensor and context data to ensure security and privacy
- Scaling up gracefully for large number of sensors, services, devices and users

CIS

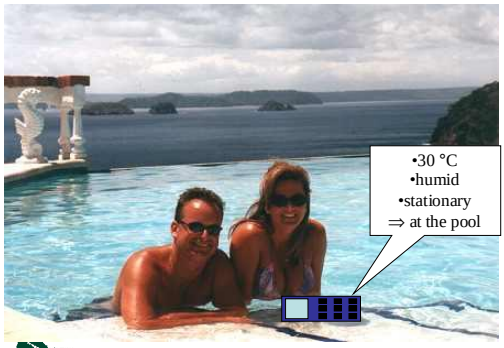
- Pascoe, Ryan and Morse say that the core of a supporting infrastructure for context-aware computing is a "context information service" (CIS), responsible for gathering, modeling and providing contextual data
- Their guidelines for a CIS:
  - Object-oriented contextual model
  - Shared access to resources
  - Supports extensible and reusable components
  - Supports layered service structure
  - Globally scalable
  - Platform independent

CONTEXT INFERENCE

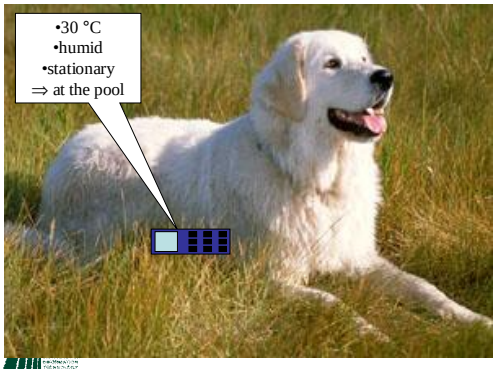
Context inference

- Context inference can be explained as the task of deducing new information that is relevant to the use of applications and users, from the various sources of context data
- Context inference is essentially a classification problem
  - Methods include decision trees, nearest neighbour methods, Bayesian networks, neural networks
- There is a trade-off between accuracy and computational complexity
- Learning and adaptation should happen on-line in contrast to many classical reasoning tasks that happen off-line

Context inference is difficult



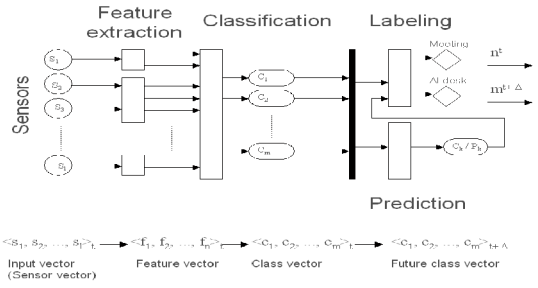
Context inference is difficult



Context prediction

- A good description is given in Rene Mayrhofer's PhD thesis [Linz, Austria, 2004]
- He describes a five-step context prediction model: sensor data acquisition, feature extraction, classification, labelling and prediction
- Time-series analysis is central for prediction: HMMs and dynamic Bayesian networks can be used
- For the classification, there is an abundance of methods, such as k-means, SOM, etc.

Mayrhofer's picture of the prediction process



Use of prediction

- Reconfiguration (e.g. preparing with loading libraries)
- Prioritisation
- Device power management
- Planning
- Early warning / alerts, e.g. for accident prevention

Learning

- Learning must be on-line and life-long (i.e. never stops)
- Use of explicit and implicit feedback can be made through reinforcement learning
- For prediction, use can be made of patterns and trends, as people are creatures of habits and also certain tasks are done in an established way, as well as of what other people have done (social navigation)

USER ASPECTS

User acceptance

- User satisfaction and technological acceptance can be achieved by providing the user with the information she wants at the right time and place
- In usability, a standard benchmark is that user/system tasks need to be 95% successful or error free [Rubin, 1994]
- We need systems that learn from the context and the user actions in different situations
- "There will be no pure context-aware applications, since concext-awareness on its own is not something a user needs… context-awareness is an enabling technology to help other applications perform better" [Brown et al., CHI 2000]

Atteniton

- Bardram and Hansen [2004] refer to studies of organisational and office work that show:
  - 90% of brief conversations are unplanned, and potentially interruptive
  - Only 55% of people who are interrupted continue their previous activity
- Oulasvirta et al. [2005] at HIIT have studied attention when using mobile devices

Usability

- ISO 9241-11 "Ergonomic requirements for office work with visual display terminals (VDTs) – Guidande on usability"
- Usability: extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use

Usability dimensions

- Effectiveness: accuracy and completeness with which users achieve specified goals
- Efficiency: resources expended in relation to the accuracy and completeness with which users achieve goals
- Satisfaction: freedom from discomfort, and positive attitudes towards the use of the product

Example usability measures

- Effectiveness:
  - accuracy: e.g., #errors
  - completeness: e.g., percentage of goals achieved, percentage of users succesfully completing the task
- Efficiency: effectiveness/effort with effort in human effort, time, money:
  - e.g., time to complete a task, #tasks / minute, cost of a task
- Satisfaction:
  - e.g., subjective ratings, # positive and #negative comments

Good advice (selection from Lund's usability maxims, 1997)

- Know thy user and you are not thy user
- Things that look the same should act the same. Things that look different should act different
- The information for the decision needs to be there when needed
- Error messages should actually mean something to the user
- Every action should have a reaction
- Everyone makes a mistake, so every mistake should be fixable
- Don't overload the user's buffer. Keep it simple, neat and organized
- The user should control the system and always know what is happening
- The more you do something the easier it should be to do
- Eliminate unnecessary decisions, and illuminate the rest
- The best journey is the one with the fewest steps
- You should always know how to find out what to do next
- Colour is information

User experience

- User experience design: overarching experience a person has as a result of their interactions with a particular product or service, its delivery, and related artefacts, according to their design
- Usability and user interface design are part of this

Evaluation of user experience

- Several methods, here only some, based on scenario, mock-up, prototype or product:
  - Brainstorming: persona (example abstract person), bodystorming = brainstorming "in the wild"
  - Self-reporting: questionnaires, interviews, diaries, experience sampling
  - Observation: in laboratory, Wizard-of-Oz, in the wild
  - Ethnographic methods (ethnomethodology)
  - Living Labs
    - William Mitchell, MIT: "Living Labs represent a user-centric research methodology for sensing, prototyping, validating and refining complex solutions in multiple and evolving real life contexts"

SECURITY, PRIVACY AND TRUST

Security

- Information security: the concepts, techniques, technical measures, and administrative measures used to protect information assets from deliberate or inadvertent unauthorized acquisition, damage, disclosure, manipulation, modification, loss, or use [McDaniel, 1994]
- Security in context-aware applications is not different from security generally
- Security has the following aspects:
  - Authentication: an entity is verified to be the entity it claims to be
  - Confidentiality: only legitimate receivers can get the data
  - Integrity: the data remains unchanged
  - Non-repudiation: a sender or recipient cannot falsely deny having sent/received the data

Privacy

- Information privacy is the claim of individuals, groups and institutions to determine for themselves when, how, and to what extent information about them is communicated to others [Westin, 1967]
- Most people are not aware of privacy issues, and even those who are, do not behave accordingly [Acquisti & Grossklangs, 2004]
- "Privacy concerns have killed many potential applications already, and will continue to do so" [Brown et al., CHI 2000]
- Privacy must be taken into account from the beginning [Meyer & Rakotonirainy, 2003], and so must security

Legal aspects of privacy

- In legal science, privacy refers to the protection of everyone's private life, honour and the sanctity of the home: data protection
- Transfer of private sensitive data requires the consent of the user, which legally hinders automatic data transfer or proactive actions
- Legal differences globally
  - EC Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data
  - EC Directive 2002/58/EC concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications)

Methods for privacy protection

- Prevent disclosure of unnecessary information (minimize amount of information disclosed)
- Access control and authentication
- Anonymity, i.e., dissociation of information from an individual [Duckham & Kulik, 2005]
- Pseudonymity, i.e. an individual maintains a persistent identity that cannot be linked to her actual identity [Duckham & Kulik, 2005]
- Obfuscation, i.e. deliberately degrading the quality of information about an individual in order to protect the individual's privacy, by abstraction or falsification [Duckham & Kulik, 2005]
- Privacy policies

Problems with privacy protection methods

- Anonymity and pseudonymity is a barrier to authentication and anonymity to personalisation
- Keep in mind vulnerability to data mining
- The user intervention for privacy policies is tricky as some users don't care, others don't know how and still some are experts
- Description of privacy for context-aware applications is difficult

Trust (1/2)

- Trust can be described as the firm belief in the competence of an entity to act dependably, securely and reliably within a specified context [Grandison & Sloman, 2000]
- System reliability is essential to trust
- System functionality and how it uses information must be understandable to the user
- The user must perceive to be in control of the application and the information
- There may be cultural differences in people's approach to trust

Trust (2/2)

- According to Wilhelm et al. [1998], the foundation of trust is based on four corner stones within a communications context:
  - Blind trust
  - Good reputation
  - Control and punishment
  - Policy enforcement

Other legal issues

- Who is responsible for a decision taken by a computer in a proactive action? If there are legal implications, a living person should be included in the process, as a computer cannot be held legally responsible
- Extensive processing of information is always related to IPRs
  - EC directive 96/9/EC on the legal protection of databases

PROBLEMS WITH CONTEXT-AWARE SYSTEMS

Desired characteristics

- Important for a successful ubiquitous computing project is a compelling application on relevant data
- According to Zipf's Principle of Least Effort [1949], people try to minimize their total future work, given their best estimates at the time [Rhodes & Maes, 2000]
- User is likely to be interested in a context "just ahead" [Jones & Brown, 2002]
- To get users to use a context-aware application, there must be some kind of add-on benefits. The user needs control and ability to opt out [Lamming & Newman, 1992]

Problems with context-aware systems (1/2)

- Pascoe, Ryan and Morse:
  - Resource hungry
  - High development cost
  - Computing environments are diverse
- Context-aware applications are built as monolithic stand-alone systems. A shared conceptual model is missing [Huang, 2002]
- Tools and standards are needed to facilitate development
- Brown et al. note that privacy concerns have killed many potential applications. “Debates on privacy tend to generate heat rather than light”
- Who should decide how to use the context info?

Problems with context-aware systems (2/2)

- Lamming and Newman points out that "unobtrusive" technologies are in fact often distracting to users
- Private observation: In US made applications, the importance of language is generally forgotten
- Context is difficult to gather and reason about
  - Users have different roles in different situations: this is difficult to detect and use

Where is the science in context-awareness?

- It is in the solutions and methods underlying context-aware applications
  - Example from Schmidt et al. (TEA project): they used Kohonen's SOM to cluster raw sensor data; then predictive Markov Chains
- Areas involved include at least:
  - HCI and UI design, usability
  - AI and machine learning
  - Data mining and probabilistic modeling
  - Software engineering, incl. architectures and agent technology

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

THANKS!

TKK