

1. Context, context-awareness; history and evolution of context-aware computing applications

Definition of a context

Definition by enumeration (weather, time, position)

Broad definitions (all relevant aspects of surroundings)

Definitions either too general or too narrow (needs to be determined case by case)

Examples of context - identity, temporal, social, activity

Aspects of context: physical surroundings, digital information, users' activity (*Brynskov*)

Context-awareness - application reacts to changing context

Passive context-awareness: presents the context (nearby gas stations)

Active context-awareness: automatic adaption by changing applications behaviour

History

Active badge (1992) - context awareness with regards to call forwarding

Tourist guides (cyberguide 1997)

Mostly academical for a long time

HyCon (2003) - too much equipment, model your own context

Adopted by the public when sensors became available in **smartphones**

2. Software architectures and frameworks for context-aware applications

Architectures

widget infrastructure (w2s21)

widget, interpreter, aggregator, service, and discoverer (w2s23)

Inspired by GUI programming

Refer to our own project

Provides good structure

blackboard systems (w2s28)

event heap and context memory (w2s29)

Active Badge

Easy to understand (but no structure)

service infrastructure (w2s34)

Client server

Offload computation to external entity

Platform independence, easier maintenance, resource sharing → simple devices

Discuss our project w.r.t. architectures

Framework vs architecture - frameworks are reusable and adaptable (w2s53)

Case examples

HyCon (HyperContext) - service oriented

The Context Toolkit (w2s56) - widget based

3. Sensing technology and mobile sensing concepts

Stationary vs mobile sensing

Stationary: sensing environment, specialized hardware, dedicated sensing, high cost

Mobile: user centric, low cost, available

Type of sensing (w3s16)

Participatory sensing

Opportunistic sensing

Design mobile phone sensing

Collect raw data

Infer context

Expose higher level context to user (or perform action)

Resource usage

Tradeoff between battery and accuracy

Sensing modes (collect raw data)

Event based sensing

Continuous sensing (using a lot of resources)

Duty cycling (what happens during sleep) (adaptive duty cycling)

Context inference

Tell what we did in the project

Use collected data with machine learning

Feature extraction

4. Machine learning for context inference

Inference pipeline: Sensing → feature extraction → classification

1. **Get data**
2. **Understand data and goal**
3. **Pick a learner and parameterize it**
4. **Understand results**

Building a classifier

1. Collect raw data
2. Design features
3. Format feature data to ARFF
4. Choose a learner
Learners (decision tree, naive bayes)
5. Choose parameters for learner
6. Reason about output

Accuracy

Confusion matrix

Bagging and boosting

5. Applied context inference, using sensing, sensor fusion, and machine learning

What is sensor fusion?

Exemplify - accelerometer and gyroscope, accelerometer and microphone

Example - google gesture recognition

What is machine learning?

Learn the computer to recognize context from raw sensor data (features)

Wu et al. (Sensor fusion for context understanding)

Determine who has the focus of attention with camera and microphone

Widget architecture for sensors (blackboard between mediator and widgets)

How we used Machine Learning

Building a classifier

1. Collect raw data
2. Design features
3. Format feature data to ARFF
4. Choose a learner
Learners (decision tree, naive bayes)
5. Choose parameters for learner
6. Reason about output

6. Advanced context-aware applications and systems

HyCon - HyperContext framework

Used by tourist guides and in education

Predates the smartphone

Hypermedia is non-linear media (video, images, hyperlinks)

Makes link between content and context

Focus on browsing, searching, annotating (tagging), linking

Browsing with your feet

Indirect representational navigation (vs direct physical navigation)

UrbanWeb

Lightweight framework for supporting context-aware urban computing

No boilerplate object model

Makes it easy to utilize context information in applications

Tagging model and context matching is the core