Mobile Information Systems

Lecture 06: Context

© 2015-20 Dr. Florian Echtler
Bauhaus-Universität Weimar
<florian.echtler@uni-weimar.de>

Key issue: context (recap)

- Unpredictable usage context
- Different viewpoints:
  - Environment (motion/noise/brightness)
  - Geometric (position/location)
  - Social context (acceptable behaviour, snooping)
  - Activity context (what the user is doing)
    - Physical/virtual activities
  - *Device context (what the device can do)*
- *Context recognition*

Context: device

- Wildly varying capabilities
  - Screen size: 1” (smartwatch) – 12” (large tablet)
  - Connection: speed (EDGE – LTE), price, …
  - Memory, energy supply, CPU power, sensors, …

→ Information *adaptation*?
  - Server-side or device-side
  - Static or dynamic

Device – server-side adaptation

- “Browser switches” (static)
  - server checks HTTP headers delivered by browser:

```
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:30.0)
Gecko/20100101 Firefox/30.0
Accept: text/html,application/xhtml+xml,
application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-gb,en;q=0.5
Accept-Encoding: gzip, deflate
```

  - select primary representation based on *User-Agent*
  - select compression, language etc. based on *Accept*
- Requires multiple versions of content (or at least multiple layouts for dynamic content)

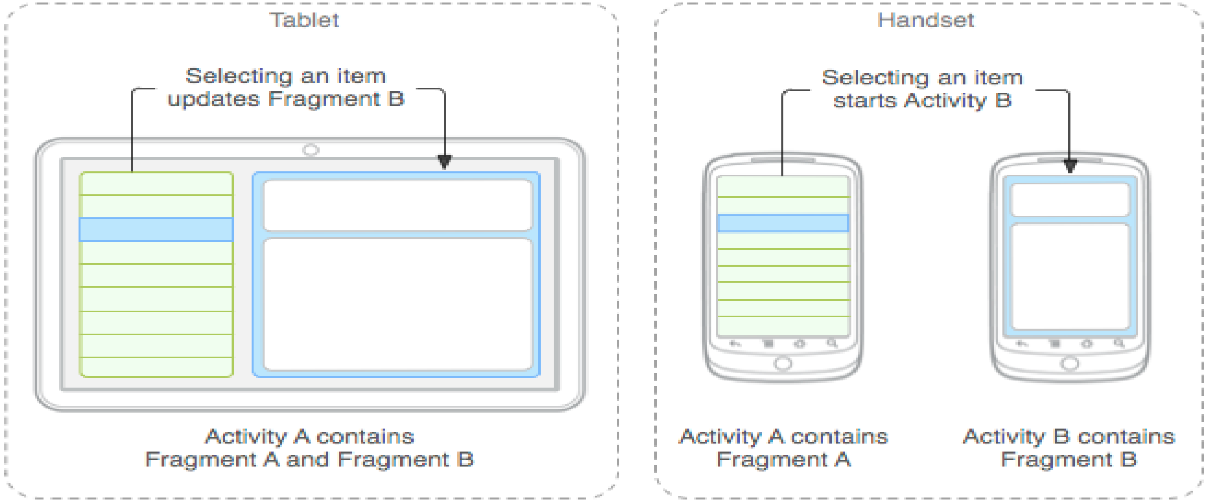
Device – server-side adaptation (2)

- Syntactic transformations (dynamic)
- Does not consider content/semantics and other aspects of context
  - HTML → WML gateways (outdated)
  - “Compilation“ via XML+XSLT
  - Compression proxy, e.g. using Opera Mobile
  - Content management systems

Device – device-side adaptation

Image source (FU): https://developer.android.com/guide/components/fragments.html

- Dynamic layout, e.g. for …
  - Web pages: using CSS3 “flex boxes”
  - Apps: using Android *fragments*



Device – resource substitution

- Adaptation via resource substitution
  - Perhaps: reduction of the data quality necessary

Substitute →→ for …	CPU	Communication	Memory
CPU		Migration of compu-tations to the server/ uncompressed data	Caching and reuse of temporary results
Communication	Local execution of calculations/ compression		Local data storage
Memory	Data compression and compact data structures	Data management on the server only	

Context: recognition (recap)

Image source (FU): http://www.gettyimages.com/gi-resources/ub/unfinishedbusiness/index.html

- Example: automatic meeting detection

  → Disable audible notifications,
  send all calls to voicemail
- Problem: what if it fails?
  - False positive: user misses important call
  - False negative: phone plays embarrassing ringtone in meeting
- Must be very, very accurate to earn user trust

Context: recognition (2)

- Aspects of context (ordered by complexity):
  - Device (e.g. screen size & resolution)
  - Virtual activities (e.g. using foreground task)
  - Position (e.g. using GPS)
  - Location (e.g. using geocoding)
  - Environment (e.g. using light sensor/mic)
  - Physical activities (e.g. using accelerometer)
  - Social (e.g. using ???)

Context recognition – sensors

- Hardware sensors (typical)
  - Microphone (obviously :-)
  - Touch screen (see lecture 4)
  - Motion sensor (IMU, see lecture 4)
  - Position sensor (GPS/GLONASS, see lecture 2)
  - Proximity sensor
    - Turns screen off when phone on ear
  - Brightness sensor
    - Controls screen brightness

Context recognition – sensors (2)

- Hardware sensors (less common)
  - Camera (e.g. as heartbeat sensor)
  - Eye tracker (see lecture 5)
  - Fingerprint sensor (see lecture 5)
  - Temperature sensor
    - Often built into battery – why?
  - Pressure sensor
    - Helps GPS with height measurement

Context recognition – sensors (3)

- Software sensors → aggregate/convert/ interpret data from hardware sensors
  - Location sensor
    - GPS + WiFi + cell location
  - Orientation sensor
    - Filtered IMU data
  - “Attention” sensor
    - Processed eye tracker data
  - “Network sensor”
    - WiFi/IP address/cell information

Context recognition – sensors (4)

- Issues when using sensors:
  - Power consumption
    - Significant impact on runtime
    - Device can get uncomfortably warm
    - Mitigation:
      - adjusting sample rate
      - turn off sensors when not needed
  - CPU load
    - When using multiple data streams at high frequency (e.g. camera + IMU + GPS) → even powerful multi-core CPUs can get high load → higher power consumption

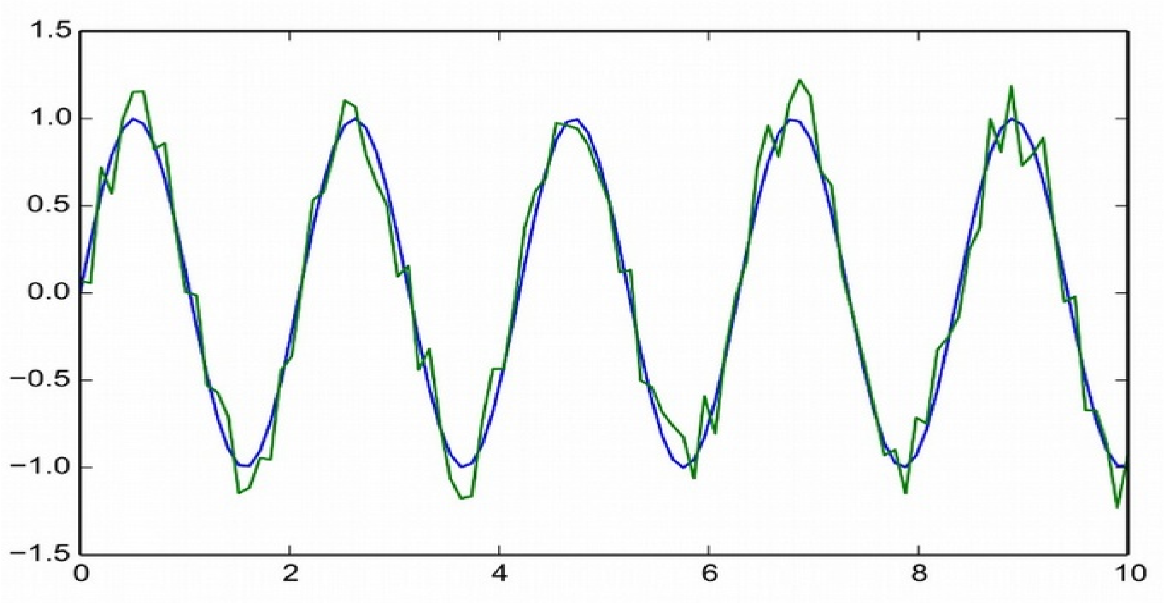
Context recognition scenario

Image source (CC): https://en.wikipedia.org/.../File:Cell_phone_use_while_driving.jpg

- Car driver scenario →
- How would you recognize this context?
  - GPS motion > ~ 20 km/h
  - On/near road
  - Constant vibrations
  - Unimanual usage
- Consequences?

Side track: signal processing

- Basic time series (blue)
  - X axis = time (evenly spaced)
  - Y axis = values/ measurements/ samples
- Data almost always noisy (green)

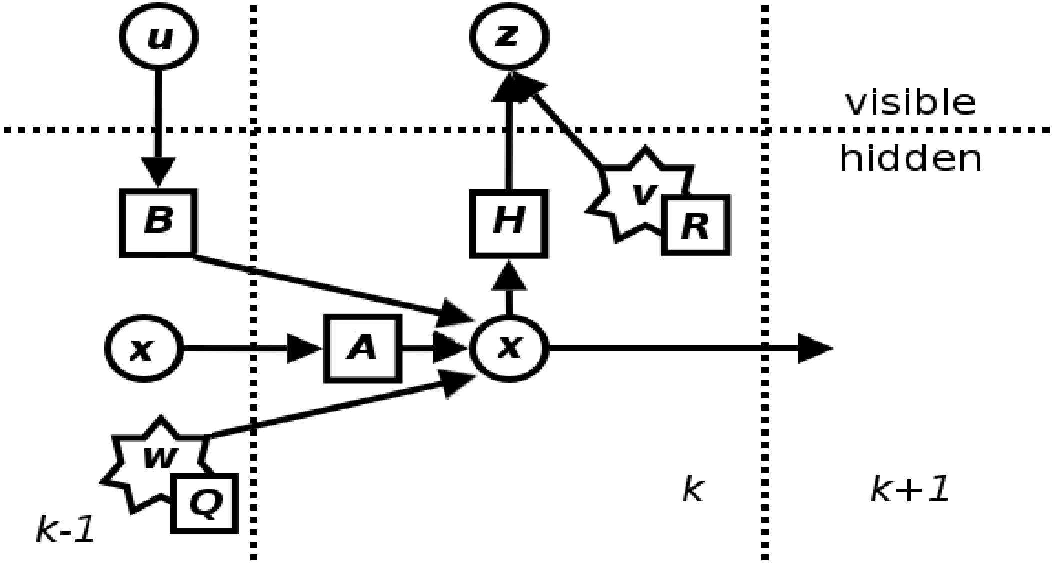


Side track: signal processing (2)

- Noisy data – two types of noise:
  - Process noise: physically present in the measured process, e.g. hand tremor
  - Measurement noise: occurs in sensor/equipment, e.g. thermal noise
- Both unwanted → filter out
  - Magnitude of noise sometimes important
- However: both part of *hidden* state

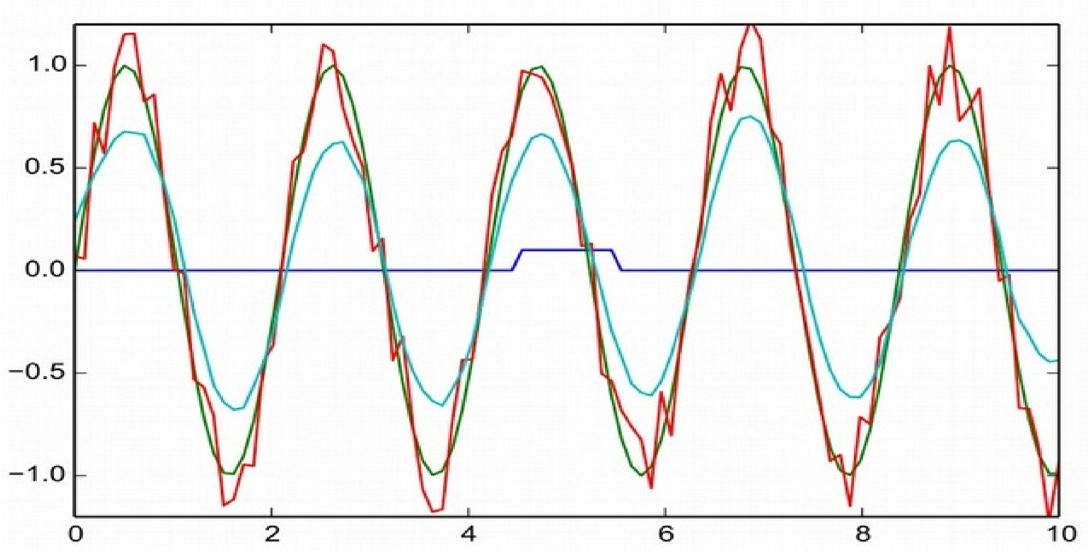
Side track: signal processing (3)

- Noise model (from Kalman filter):
  - X = “true” system state, A = transfer function
  - U = control vector, B = control function
  - W = process noise
  - Measurement …
    - *Function H*
    - *Noise V*
    - *Result Z*
  - “Timeslice” k



Side track: signal processing (4)

- Filter *kernels (blue)*
  - 2nd function multiplied “on top” of time series (red)
  - Sum of products = new function (cyan)
  - E.g. square kernel → moving average
- *Convolution*
  - Point-wise multiplication of 2 functions



Side track: signal processing (5)

- Characterizing signals
  - Simple approach: mean & standard deviation
  - Mean = average over certain time window
  - Standard deviation (aka standard error)
    ~ average difference from mean value
- Very different signals can have very similar values for mean & standard deviation!

Side track: signal processing (6)

- Further reading: DSP Guide (chapters 1-9)
  http://www.dspguide.com/pdfbook.htm

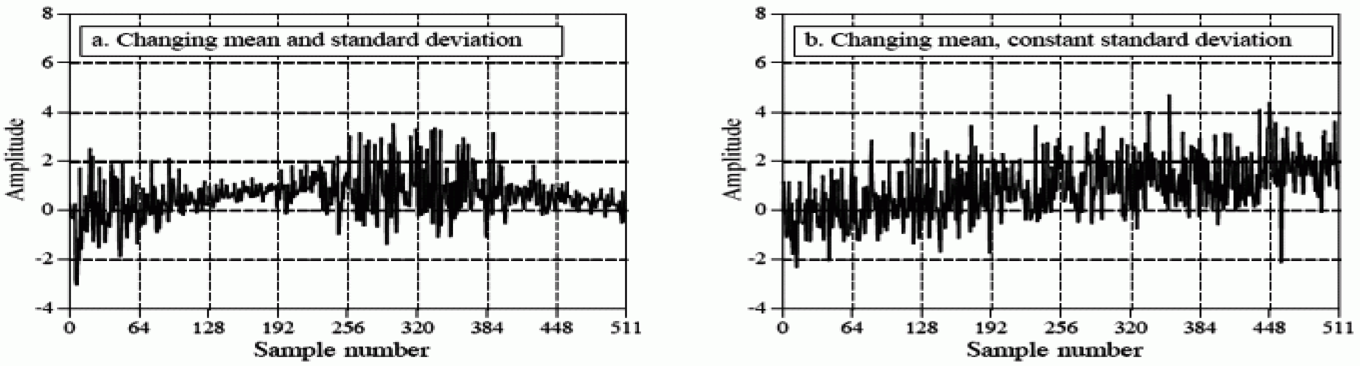
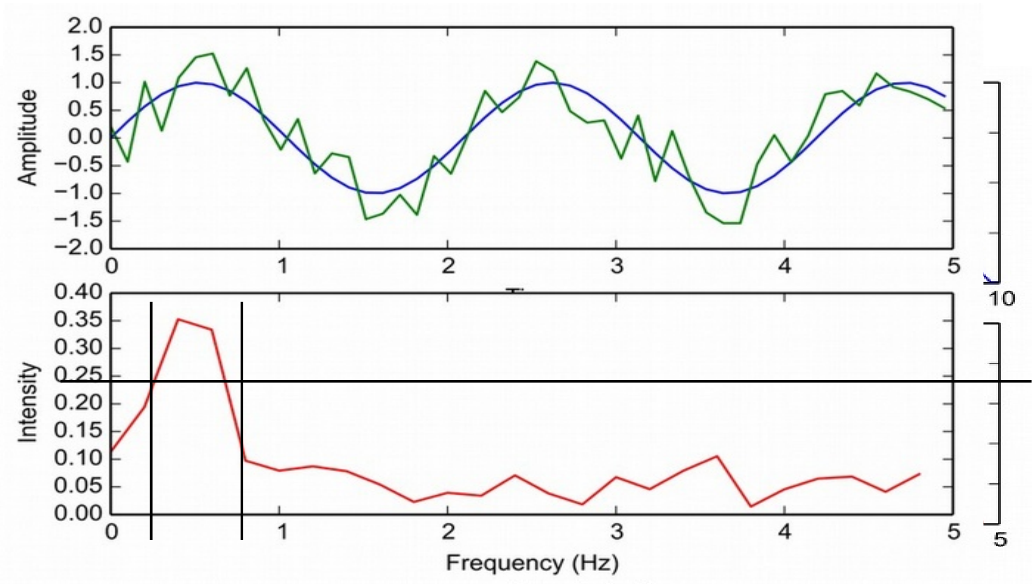


FIGURE 2-3
Examples of signals generated from nonstationary processes. In (a), both the mean and standard deviation change. In (b), the standard deviation remains a constant value of one, while the mean changes from a value of zero to two. It is a common analysis technique to break these signals into short segments, and calculate the statistics of each segment individually.

Side track: signal processing (7)

- *Fourier* transformation (DFT, FFT)
  - Time series (*amplitude* domain) can be represented as sum of sin/cos functions (*frequency* domain)
  - Transformed data is new function, can be convoluted, filtered, etc. again
  - Analysis of certain aspects easier than on original function



Context: implications

- Available/permitted attention level ("channel bandwidth")
  - Audiovisual attention
  - Input speed/precision
  - Timespan for interaction
- Available/permitted operations
  - Bimanual operation not possible?
  - Unobtrusive operation required?
  - Legal restrictions?

Context: implications (2)

- Possible reactions?
  - E.g. disable phone while driving?
  - E.g. enlarge buttons while walking?
  - E.g. dim screen while in a crowd?

→ exceptions always possible & required

→ even more difficult to recognize!

The End


SCHEDULE A MEETING WITH DILBERT AND ALICE FOR NEXT TUESDAY AT TEN.
DONE
Dilbert.com DilbertCartoonist@gmail.com
NEVER MIND. MY PHONE TOOK CARE OF IT.
12-16-11 ©2011 Scott Adams, Inc./Dist. by Universal Uclick
AWKWARD.