Telecooperation Lab
Prof. Dr. Max Mühlhäuser

TK3: Ubiquitous Computing

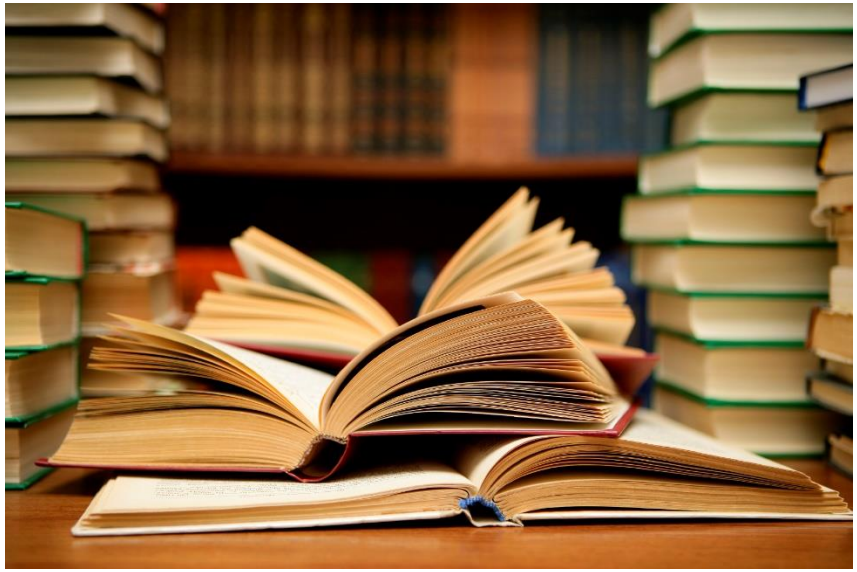
Chapter 3: Context-aware Computing

Part 1: Basics

Lecturer: Dr. Immanuel Schweizer

Sources

- http://cgi.csc.liv.ac.uk/~trp/Teaching_Resources/COMP327/327-Lecture7-ContextAware.pdf
- www.intel.com/education/highered/wireless/lectures/lecture_14_pervasive.ppt



Pervasive Computing

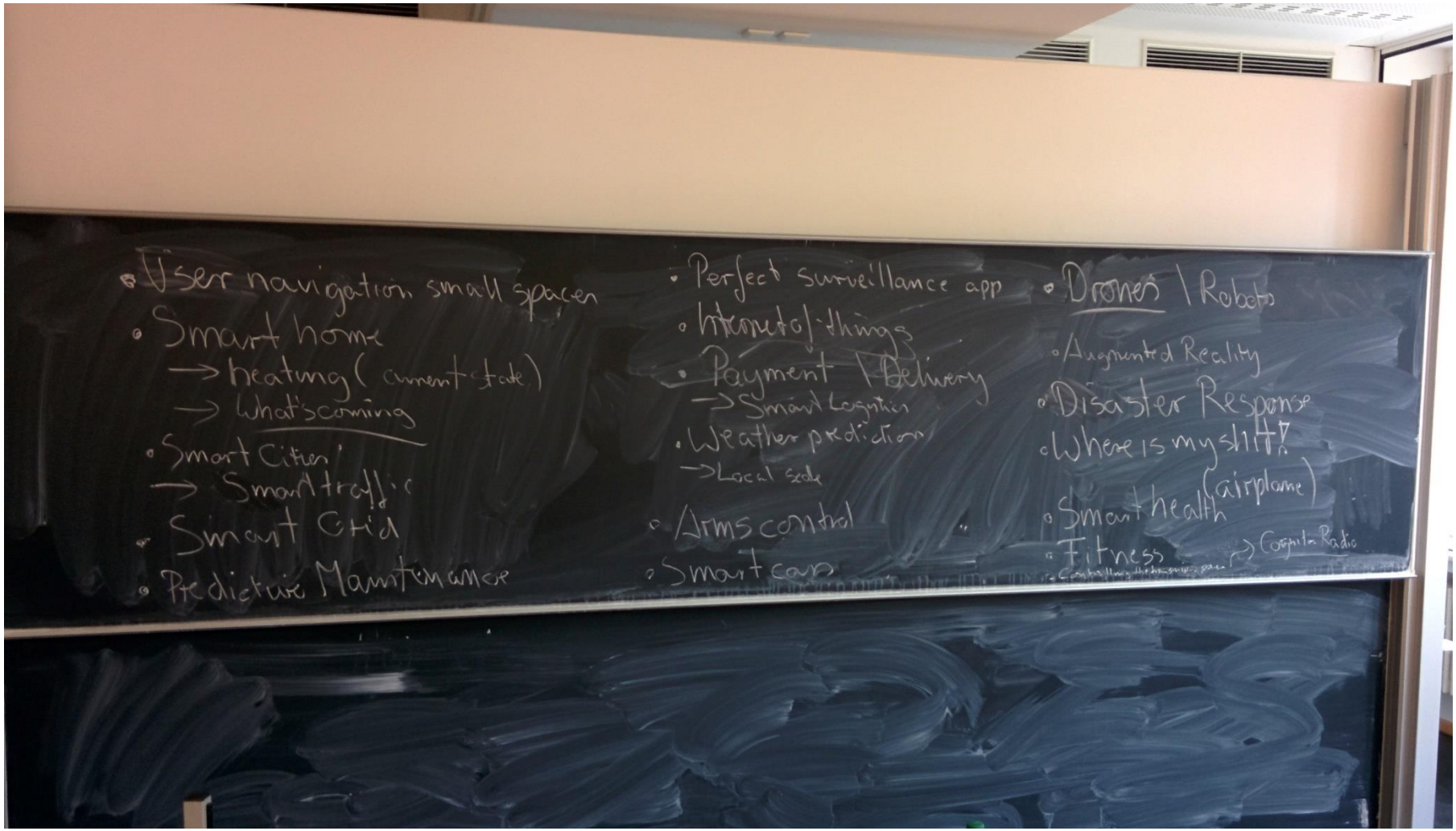
TECHNISCHE UNIVERSITÄT DARMSTADT

- Pervasive computing is:
  - An environment in which people interact with embedded (and mostly invisible) computers and in which networked devices are aware of their surroundings and peers and are able to provide services or use services from peers effectively
  - The creation of environments saturated with computing and wireless communication, yet gracefully integrated with human users

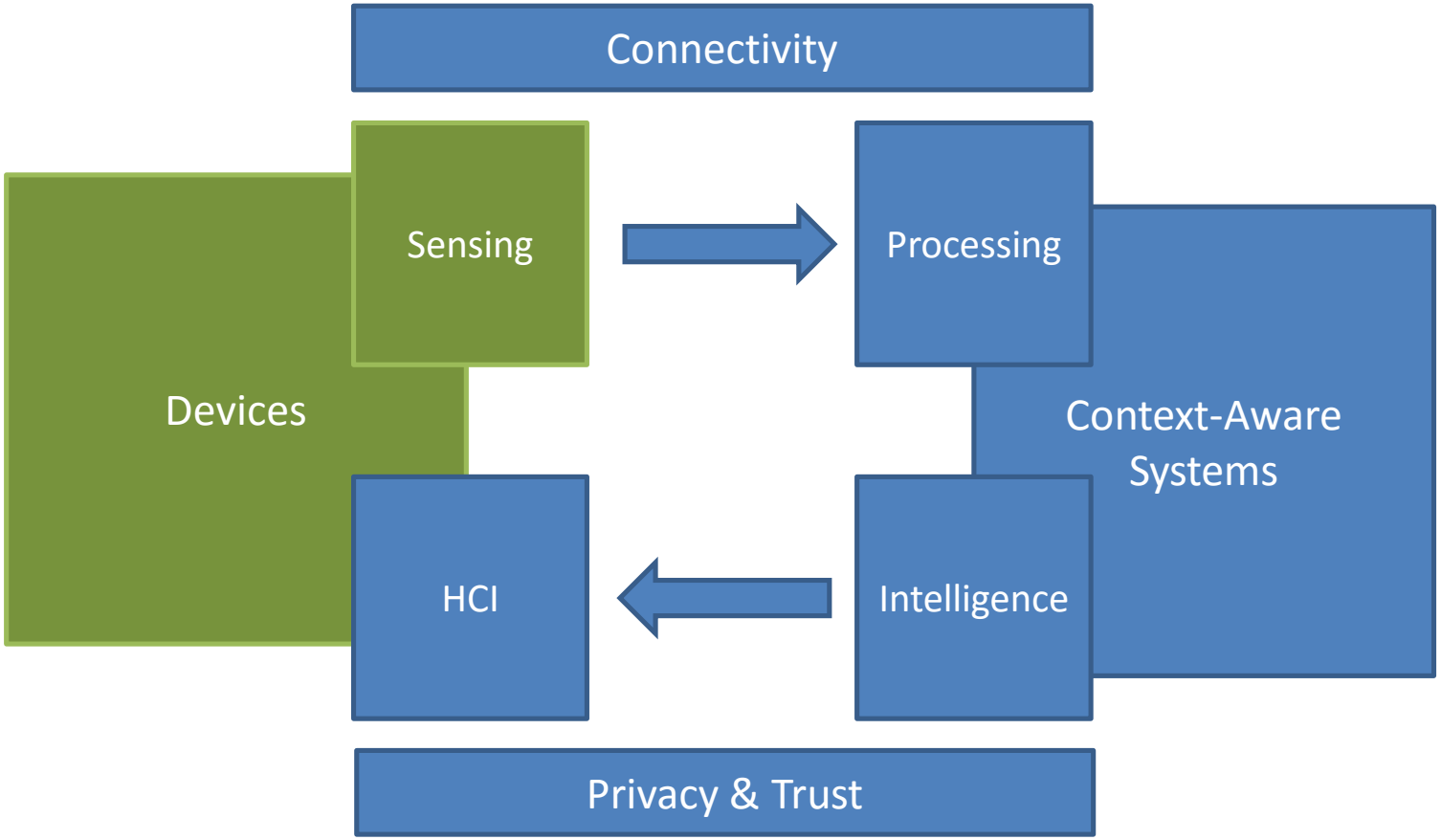
System Properties

1. Computers need to be networked, *distributed* and transparently accessible
2. Computer *Interaction* with Humans needs to be more *hidden*
3. Computers need to be *aware* of *environment context*
4. Computers can operate autonomously, without human intervention, be self-governed
5. Computers can handle a multiplicity of dynamic actions and interactions, governed by intelligent decision-making and intelligent organisational interaction. This entails some form of artificial intelligence.

Ubiquitous Computing



Simple Architecture



Context Aware Systems

TECHNISCHE UNIVERSITÄT DARMSTADT

- Systems that are aware of their own situation in the *physical*, *virtual*, and *user* environment
  - Examples?
- Typically adapt their operation or goal based on contextual cues from the environment or the user's actions
  - Implicit behavior, rather than explicitly directed by the user
  - Lessen the (cognitive) load on the user
  - Requires ability to not only sense environment, but to determine what is relevant to the system's task(s)
    - May require additional processing to convert raw sensor data into relevant information
    - May also need a representation of the relevant knowledge model used to comprehend contextual cues and direct behavior

Why use context?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Humans use context for adapting their behavior to the current situation (e.g. time of day, location, surrounding people, …)
- *Context-Aware Computing (CAC):* let computers do the same
- Goal: Build (enable) applications, smart environments, … that:
  - Adapt functionality to 'situations' (render more appropriate service, …)
  - Adapt interactivity to 'situations' (reduce cognitive load of users, …)
- Re. interactivity: how to do it?
  - Proactivity
    - Set up environment according to user's preferences or usage history
    - Auto-completion of forms (location, time in HEAG timetable)
    - Reminders
  - Search and filter information according to the user's current needs
  - Avoid interrupting the user in inappropriate situations
  - … and in smart environments e.g.:
    - Turn devices on/off, start applications, … depending on location, time, situation (lecture, meeting, home cinema, …)
    - Discover and use nearby interaction devices

What is context?

TECHNISCHE UNIVERSITÄT DARMSTADT

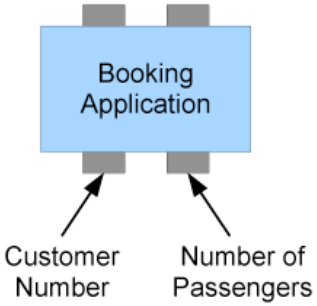
- Context (Dey 99) (defines *context* by means of *situation* [def?]: scientifically doubtful)
  - Context is **any information** that can be used to **characterize the situation of an entity**. An entity is a **person**, **place or object** that is considered **relevant** to the interaction between a user and an application, including the user and application themselves.
  - Context-aware System:
    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

- Our more precise notion, replacing 'situation':
  Context Aware Computing is the **ability** of a computer-based system
  - To (self-)**adapt its behavior** (i.e. functionality and/or interactivity)
  - To **physical conditions** of operation a/o **human factors**
    - **Human?** user/s & 3rd parties (group, community, org-chart), factors? characteristics, behavior beyond immediate 'provision of input data'
      - Maybe in relation to past conditions / behavior (learned, modelled, …)
      - Usually based on aging (time dependent), probabilistic, and error-prone data

What is context?

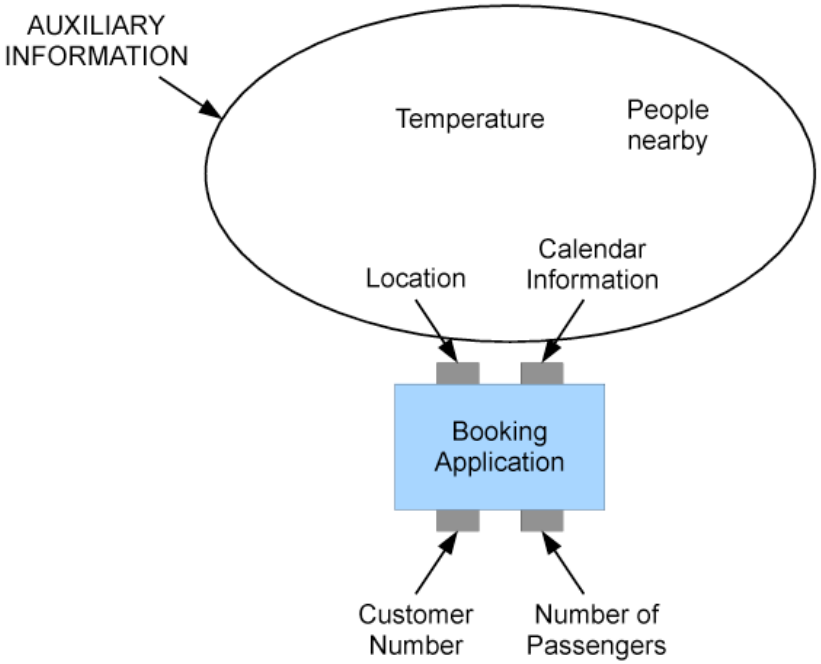
TECHNISCHE UNIVERSITÄT DARMSTADT

- Example: (Train) booking application requires
  - User input: booking details, such as customer#, # of passengers



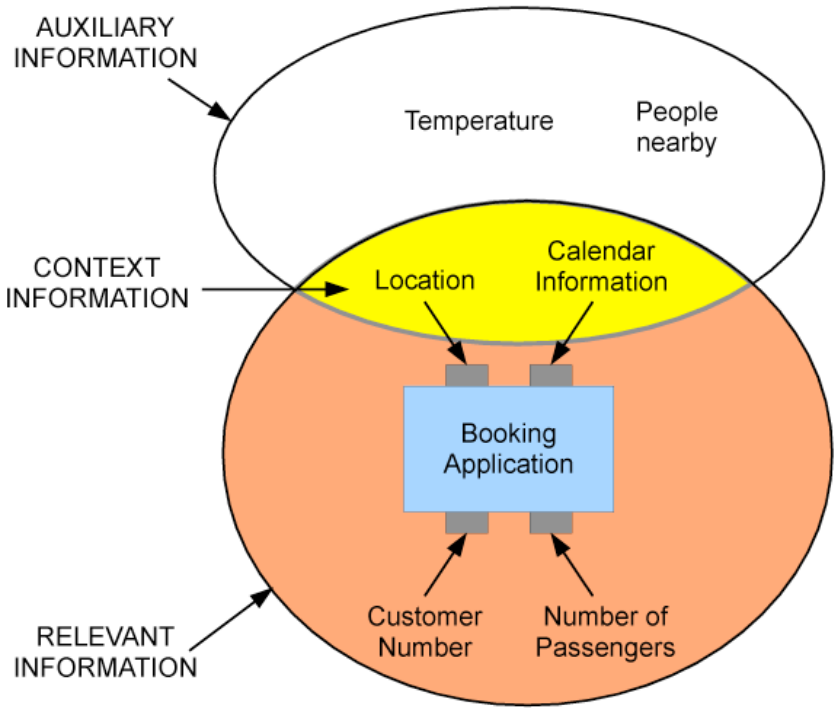
What is context?

- Example: (Train) booking application requires
  - User input: booking details, such as customer#, # of passengers
  - **Auxiliary information:** location, calendar information, temperature, people nearby, …



What is context?

- Example: (Train) booking application requires
  - User input: booking details, such as customer#, # of passengers
  - **Auxiliary information:** location, calendar information, temperature, people nearby, …
  - Only some of this information is relevant



Definition

TECHNISCHE UNIVERSITÄT DARMSTADT

- Context (Hartmann et al.):
  - Context characterizes the actual situation in which an application is used (weak, not part of definition: “situation” itself undef.)
  - We only refer to information as **Context** that can actually be processed by an application (**relevant information**), but that is not mandatory for its normal functionality (**auxiliary information**) or for which input is taken from 3rd source (real world, foreign-source data, data obtained from software instrumentation) (**information from auxiliary source**)
- Context-aware system: system that adapts itself to context

Classifying CA systems

TECHNISCHE UNIVERSITÄT DARMSTADT

- Various ways of classifying CA software:
  - Proactive Triggering
    - e.g., performing some interaction based on environmental perceptions
  - Streamlining Interaction
    - Reducing irrelevant information - e.g. travel guide for current location
  - Memories for past events
    - Contextual retrieval, e.g. based on spatial or feature-based cues
  - Reminders for future contexts
    - e.g., tagging details regarding current context for future access
  - Optimizing patterns of behaviors
    - e.g. changing interface based on situation (from screen to text-to-voice)
  - Sharing Experiences
    - Social networking based on shared contexts

Classifying CA systems

TECHNISCHE UNIVERSITÄT DARMSTADT

- Passive context-aware systems
  - New context is presented to the user, to inform them of change.
    - User can then explicitly determine if the use of an application should change
    - Examples include
      - Changing information regarding environment (weather meters)
      - Context-based tagging (e.g. by camera)
- Active context-aware system
  - Behavior of the applications change automatically
    - Examples include
      - Task filtering (e.g. information filtering, based on current wireless network speed)
      - Context-base task activation (e.g. routing)
      - Content adaptation (e.g. for people with disability)

Context Architectures

- A basic context-aware model consists of four components:
  1. Current context capture (context determination)
  2. Goal context creation
  3. Adaptation of the current context to goal context
  4. Context management

I. Context Determination

TECHNISCHE UNIVERSITÄT DARMSTADT

- Acquisition of environmental cues
  - May require configuration, e.g. frequency of samples
- Acquisition of user contexts
  - e.g. user identity, stereotype recognition, preference retrieval
- Encapsulation and Abstraction
  - Convert raw data into meaningful information
  - Harmonize heterogeneous sources into homogenous data
- Filtering of environmental context to user context
  - Identify what is relevant for further use!

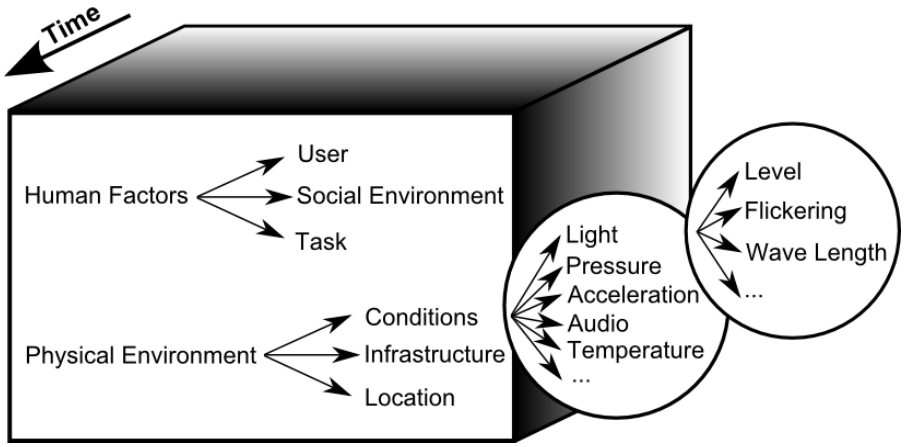
Context Creation

TECHNISCHE UNIVERSITÄT DARMSTADT

- New contexts can be created based on sensor data (captured by the device or nearby sensors)
  - Lower-level raw contexts may need to be processed into higher level contexts
    - Raw data may need to be scaled or transformed
      - e.g. electrical signal on a temperature gauge should be converted into a Celsius value...
      - ...or absolute geo-location position should be converted into an address or identification of a building
  - Often this abstraction is more useful
    - e.g. “ this photo was taken at my parents home last christmas”

Examples

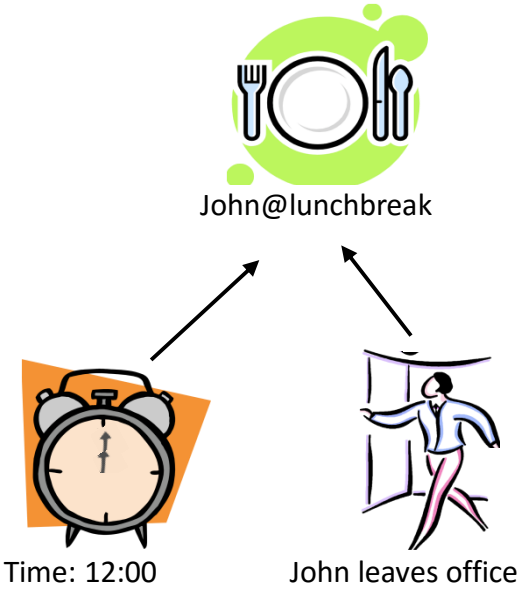
- Physical environment
  - Location
    absolute or relative position
  - Infrastructure
    surrounding resources for computation, communication
  - Physical conditions
    noise, light, pressure



- Human Factors
  - Information on user:
    knowledge of habits, emotional state, biophysical conditions
  - User's social environment
    co-location of others, social interaction, group dynamics
  - User's tasks
    current activity, ongoing tasks, general goals

Context Types

- Sensed context
  - Physical sensors or virtual sensors (applications)
  - Examples: temperature, Outlook calendar entries
- Inferred or derived context
  - Combination of context data to gain new information (“higher level context”)
  - Examples: Activity (e.g., “being in a meeting”), symbolic location (e.g., “S202|A124”)



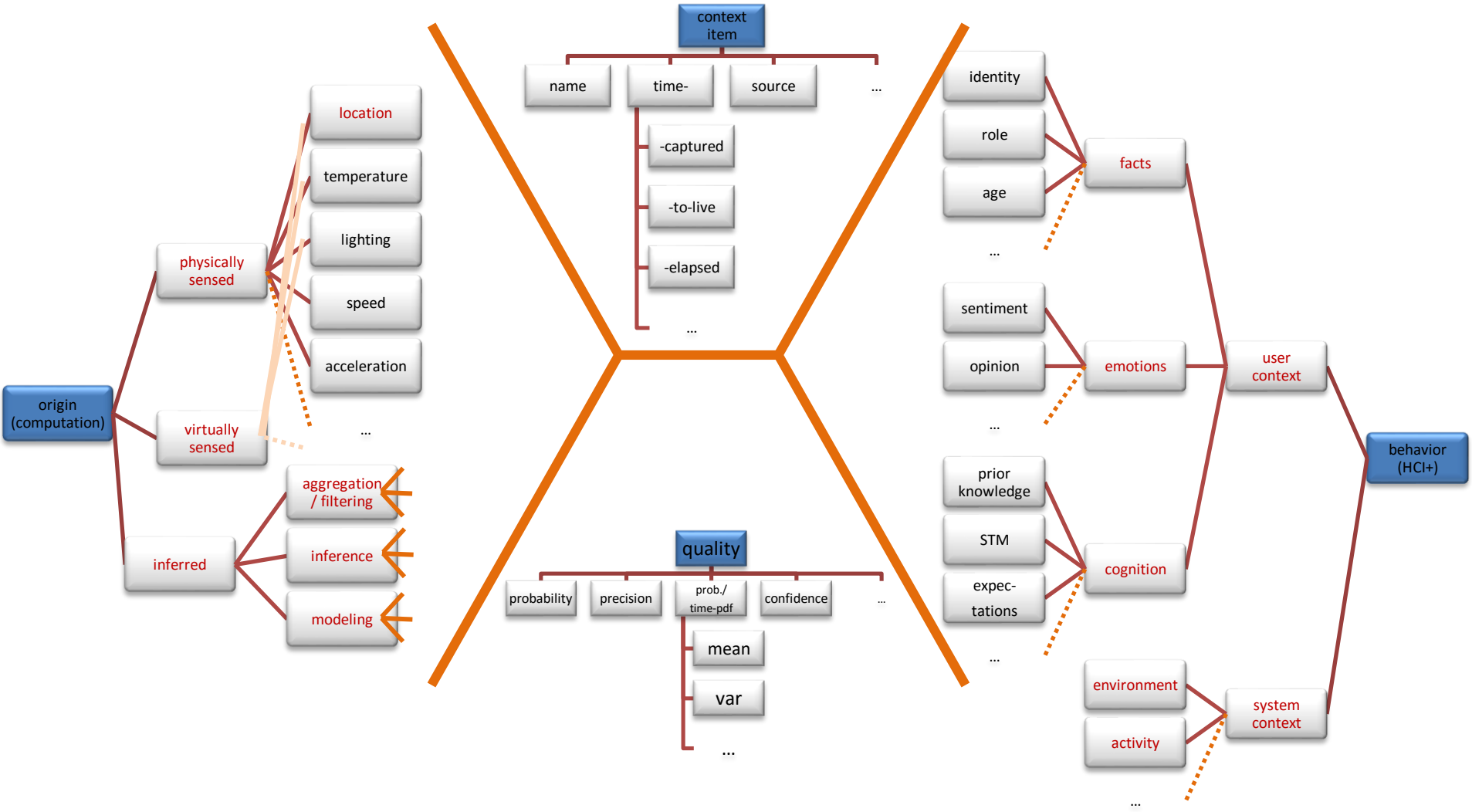
Context Type	Sensors	Examples
Sensed context	Physical sensors	Temperature
	Virtual sensors	Outlook
Inferred Context	Logical Sensors	Activity

Context Creation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Some contexts can be derived from other contexts
  - By combining several contextual cues, a better understanding of the context can be determined, than considering the cues in isolation
  - The same cues might mean different things
    - A user in front of a building may be admiring the architecture, or waiting for a friend
- Good for implicit HCI - iHCI systems
  - e.g. knowing the current location, current time, and having access to the user's calendar, an app can infer the user's context
    - i.e. waiting for a flight, in a meeting, or even en-route, but running late.
- Cues may be heterogeneous in format
  - Currently over 100 location coordinate systems are in use
  - Need mechanisms to resolve heterogeneity within open environments

Categories of Context



Example: Smart Kitchen

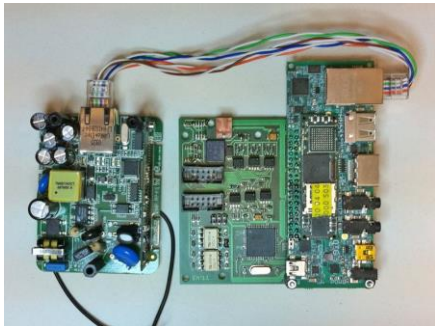
- Detect cooking activities
- Why?
  - Recipe support: **guide** the user through the cooking process, **adapt** when the user deviates from the plan
  - Nutrition: determine the nutritional value (Vitamins, etc.) taking the cooking process into account

Smart Kitchen: Sensed Context

- Kinect sensors (2x)
  - Depth camera
  - RGB camera
  - Microphone array

- Custom electronics
  - Using the Arduino prototyping platform
  - Detection of open / closed doors

- 3D wireless accelerometers (WAX) attached to knives, spoons, etc.
- Smartphones (gyro, accelerometer, compass) worn by users
- Smart kitchen appliances: e.g., Smart Blender

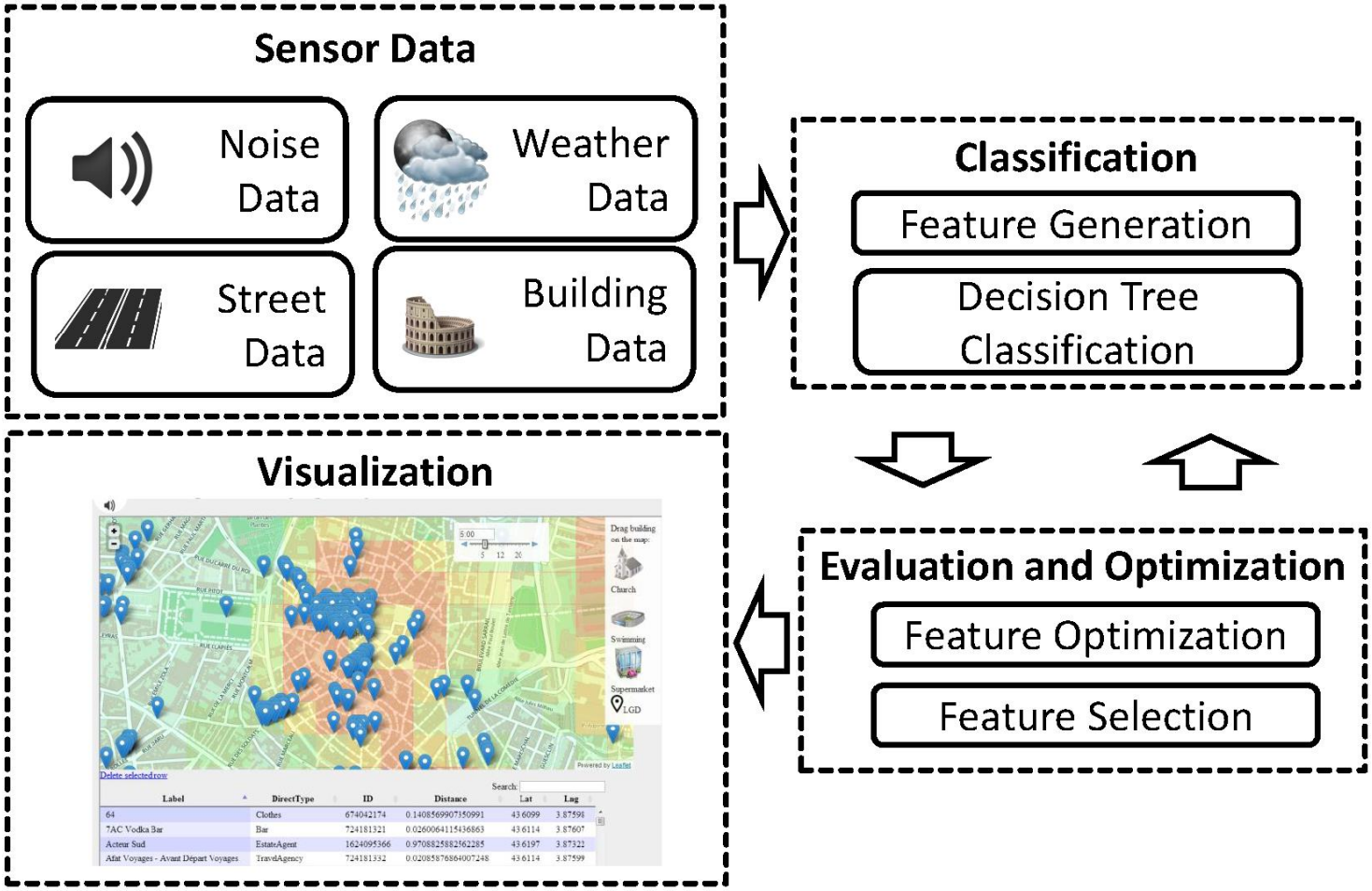


Context Processing

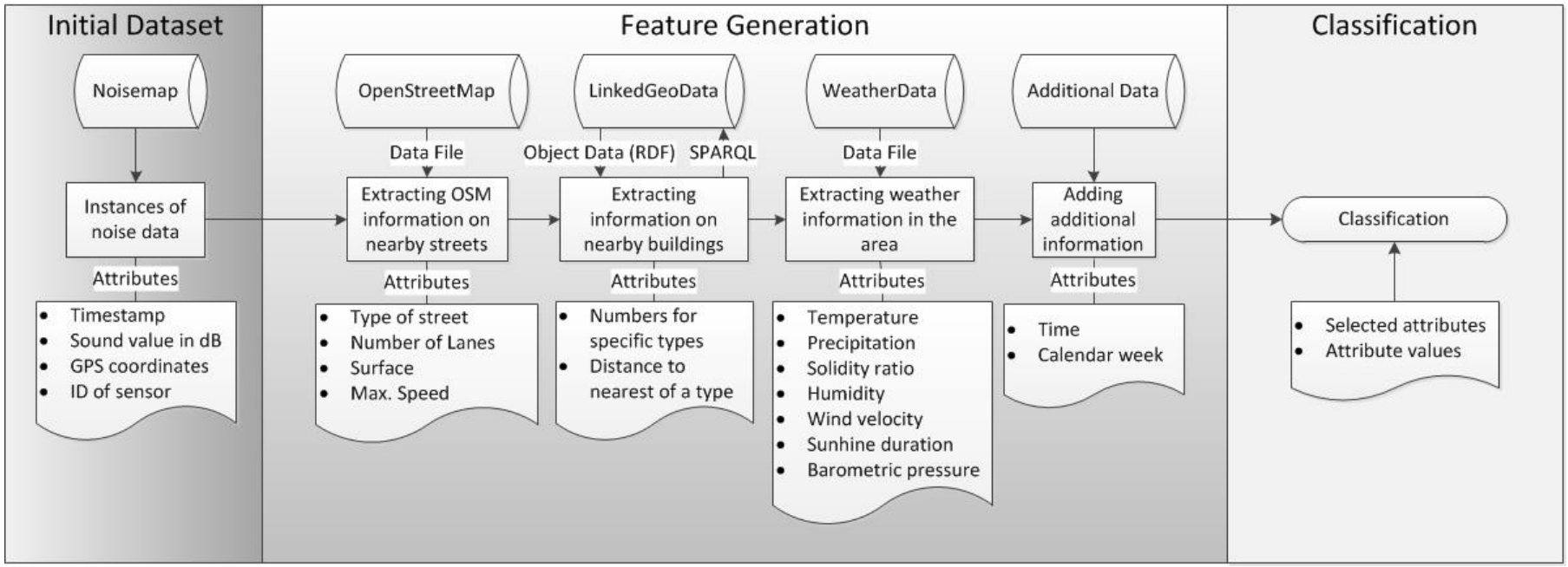
TECHNISCHE UNIVERSITÄT DARMSTADT

- Basically two steps
  1. Identify existing sensors / install new sensors
  2. Analyze sensor data to derive „higher level“ context
- Sensors:
  - **Almost endless list of sensing options:** distance, acceleration, strain, torque, speed, flow, pressure, color, humidity, liquid level, gas, weight, vibration, force, position, light, proximity, magnetometer, surface tension, optosensors, pH meter, rain, microphone, radiation, electricity, temperature, ...
  - Often tightly integrated: e.g., InvenSense 9-Axis (gyro, accelerometer, compass) motion tracking chip: dimensions = 4mm x 4mm x 1mm
- Analysis
  - Signal processing, feature extraction
  - Machine learning or heuristics

Example: Pipeline

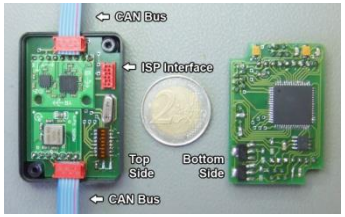


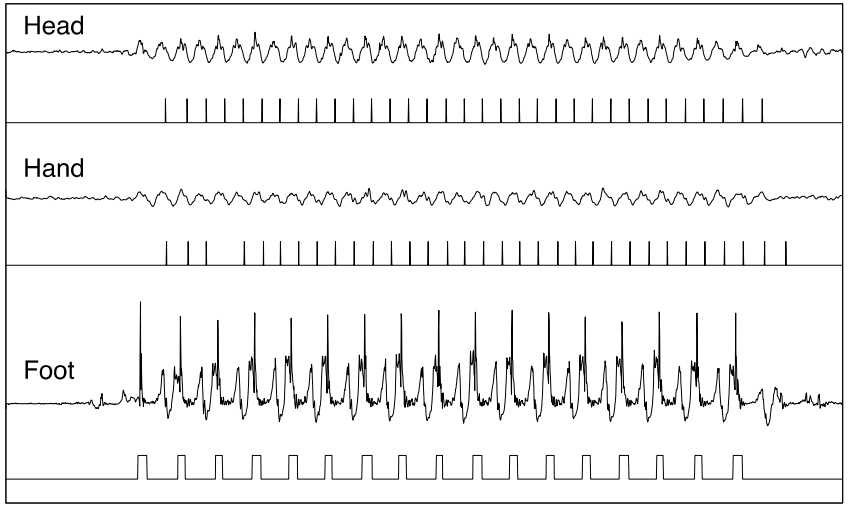
Example: Feature Extraction



Example: Step Detection

- Inertial sensor
  - 3D accelerometer
  - 3D gyroscope
  - Worn on head, hand, or foot





- Signal processing (hand)
  - Calculate the amount of acceleration: from x, y, z acceleration signal

$$\sqrt{x^2 + y^2 + z^2}$$

  - Highpass filter
    - Butterworth 10th order
    - Remove static and low-frequency components
  - Rectify signal
  - Moving-average filter (0.1 s window size)
  - Hysteresis threshold obtains the binary step signal

E. Aitenbichler, F. Lyardet, A. Hadjakos, M. Mühlhäuser: Fine-Grained Evaluation of Local Positioning Systems for Specific Target Applications

Dealing with Uncertainty

TECHNISCHE UNIVERSITÄT DARMSTADT

- Has to be handled in three areas:
  - Sensing context information
  - Inferring context information
  - Using context information
- How to determine uncertainty of sensed context
  - May be reported by sensor (e.g. biometric authentication devices give a measure for the confidence in reported data)
  - Specified relevance function takes freshness of context data into account. Validity decreases with distance to acquisition event
- How to determine uncertainty of inferred context
  - Most widely used reasoning strategies are: probabilistic and fuzzy logic and Bayesian networks
- How to use uncertain context information
  - Specify required confidence level (e.g. for authentication)
  - Only regard the context value with maximum probability as valid

Sensor Data Analysis

TECHNISCHE UNIVERSITÄT DARMSTADT

- Segmentation
  - Sensors usually produce data continuously
  - Choose time-granularity to predict the context
  - Detect segments in the data signal, e.g., using local thresholds
- Raw sensor data
  - Contains redundant information
  - May confuse the classifier
- Features
  - Calculated based on the raw sensor data
  - Usually domain specific
  - Examples: mean, variance, standard deviation, zero crossings, FFT coefficients, integration, ...
- Machine Learning
  - Training set consisting of features and class labels
  - Method learns to predict the class label given the features
  - Methods: k-NN, SVN, decision trees, neuronal nets, ...

Context Architectures

- A basic context-aware model consists of four components:
  1. Current context capture (context determination)
  2. Goal context creation
  3. Adaptation of the current context to goal context
  4. Context management

2. User Context Acquisition

TECHNISCHE UNIVERSITÄT DARMSTADT

- Determine user's goal context
  - Could be based on a list of the user's application tasks or desired goals
    - e.g. todo list
- Policy or constraint acquisition
  - Based on user's tasks, what should be mediated based on the environmental context (from point 1)?
- Encapsulation and abstraction of user contexts

3. Context Processing

TECHNISCHE UNIVERSITÄT DARMSTADT

- Determine any synergies or compositional implications of the contextual cues
  - An application may govern the use of multiple environmental contexts from different sources
- Mediate and interoperate between difference contextual representations
  - Identify a suitable representation to facilitate application-based reasoning
- Adapt the application with respect to the context
  - Passive: use the context to constrain user input or queries, or what information is presented to the user.
  - Active: adapt application to the context, proactively performing tasks (e.g. pulling data and/or presenting information to the user).

4. Context Management

TECHNISCHE UNIVERSITÄT DARMSTADT

- Discovery / Repository
  - Directory services enable context stores, sources, and users, to be registered and discovered
- Storage
  - Store context data in a repository (local or otherwise). May be historical, and used to further adapt the application or user profile in future
- Sharing of environmental and goal contexts
  - For distribution and access
- Access Control
  - Protect the privacy of any user-oriented data, or contextual information that could be used to identify a user

Context Models

TECHNISCHE UNIVERSITÄT DARMSTADT

- Context data must be represented in machine readable form to enable application to use it
- Context model defines exchange of context information
- Context model has to provide a useful set of attributes for each context data (type, value, timestamp, source…), ideally it addresses how to cope with incompleteness and ambiguity of context information

- Existing Context Models can be classified by means of the data structure they use for exchanging context information:
  - Key-Value Model
  - Markup Scheme Model
  - Ontology-based Model
  - Object-oriented Model
  - Logic-based Model

Key-Value Model

TECHNISCHE UNIVERSITÄT DARMSTADT

- Simplest model
- Describes context as a set of attributes
- Easy to manage
- Often used in service frameworks for describing
  the capability of a service

- Pros: Easy to manage and parse in embedded systems
- Cons: Uses Exact matches, lacks expressive structure, has weak formalisms

Example:

```
Room = A12
ID = 44
```

Markup Scheme Model

TECHNISCHE UNIVERSITÄT DARMSTADT

- Hierarchical structure
- Consists of markup tags with attributes and content
- Allows type and range checking for numerical values
- Typically used for modeling profiles, **e.g.** as extensions for CC/PP (Composite Capabilities / Preferences Profile);
  CC/PP defines profiles for mobile devices
  - Based upon the Resource Description Framework (RDF)
  - Capabilities of device like screen size defined in profiles
  - CC/PP Context Extension: Network Interfaces of devices, Location, DisconnectionStatus, …

- Pros: Distributed model, uses W3C standards, XML web services are becoming pervasive

- Cons: weak formalism

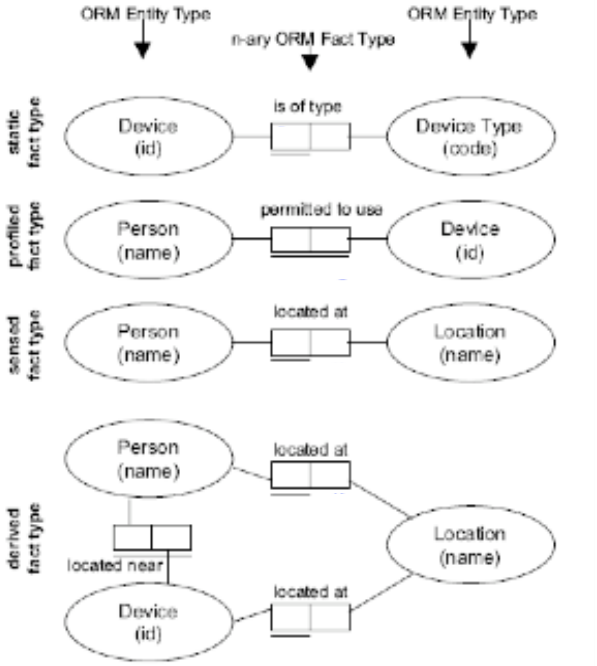
```
<Location confidence="80%">
   <Room>A12</Room>
   <ID>44</ID>
</Location>
```

Object-oriented Model

- Affords encapsulation and reuse of parts of the model
  - Entities and relations modeled as objects
  - Processing/reasoning done by “widgets”
- Representation of context, e.g., by Object-Role Modeling
  - “Typed” relation between classes → fact types
  - Instances are called facts
- Pros: Distributed OO is mature. Reuse supported through inheritance and composition
- Cons: handling incompleteness

Example:

Location
Room = A12 ID = 44 Confidence = 80%



Logic-based Model

- Formal system based on facts, expressions and rules
- Context information is added, updated, deleted from logical system
- Logical system infers new context information depending on the specified rules
- Mathematic properties useful for applications in the area of artificial intelligence
- Does not contain straightforward representation of quality meta-information

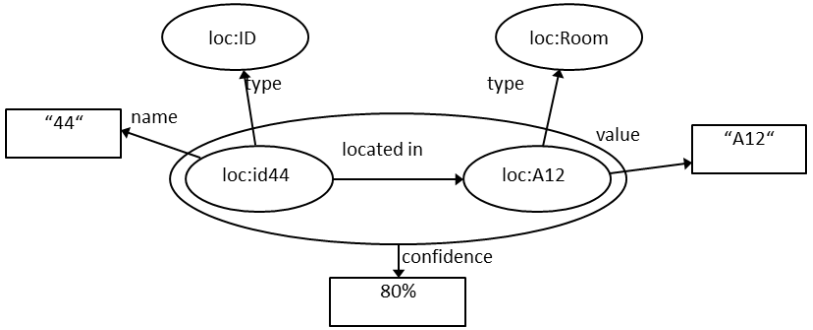
```
locatedAt("44", "A12", 80%)
```

- Pros: Strong formalism, expressive structuring
- Cons: handling uncertainty, heterogeneity and temporal elements is challenging. High complexity

Ontology-based Model

- Ontology consists of concepts, properties, relations and axioms
- Provides uniform way to specify a model's core concepts
- Facilitates sharing knowledge between different applications by defining a common vocabulary
  - Common upper ontology captures general model
  - Several domain or application-specific ontologies refine model

- Pros: expressive structuring, managing heterogeneity
- Cons: handling uncertainty, and scalability issues

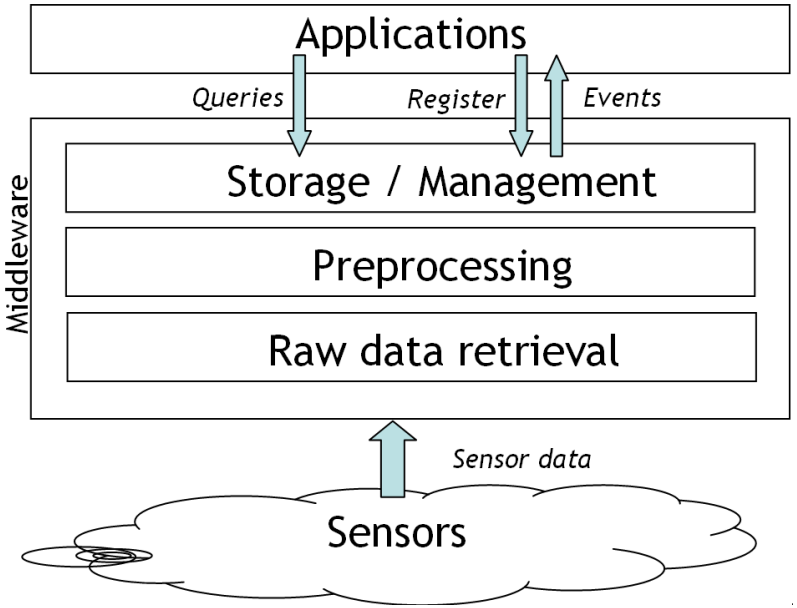


Context Architectures

- A basic context-aware model consists of four components:
  1. Current context capture (context determination)
  2. Goal context creation
  3. Adaptation of the current context to goal context
  4. Context management

Context Middleware

- Facilitates the development of context-aware applications by separating the detection and usage of context data ➔ use a reusable and extensible middleware for the detection
- Most middleware approaches use an architecture with the following layers



Context Middleware

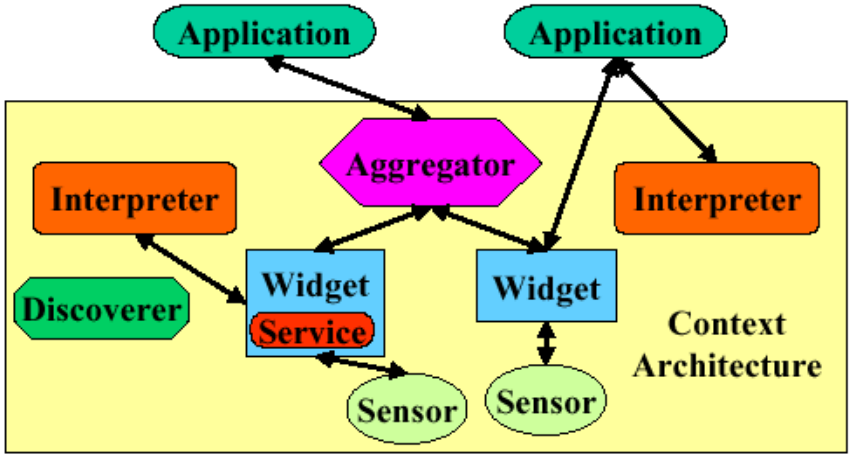
TECHNISCHE UNIVERSITÄT DARMSTADT

- Raw data retrieval
  - Uses drivers for querying physical sensors and APIs for querying virtual sensors
- Preprocessing

  Interpret and reason over context information by using:
  - Context aggregation / fusion: combine context values, cope with sensing conflicts
  - Context filtering: filter unnecessary data
  - Context interpretation: combine context data with static information (e.g. turn absolute coordinates into symbolic like “S202/A124”)
- Storage and Management
  - Manages gathered data and offers public interface to the client applications
  - Answers queries and notifies interested applications about events
  - Stores the context history

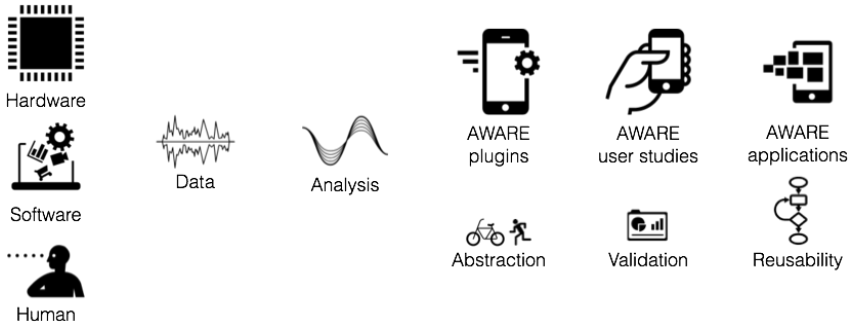
Example: Context Toolkit

- Developed by Dey et al. 2001 [Dey 2001b]
- Consists of context widgets and an infrastructure hosting the widgets
- Offers several software components for context acquisition to facilitate the software development:
  - *Context widgets:* collect context information from sensors
  - *Context services:* perform action on behalf of an application (e.g. sending an email)
  - *Context interpreters:* convert context between different representations
  - *Context aggregators:* combine data from several widgets and interpreters
  - *Discoverers:* maintain registry of available widgets



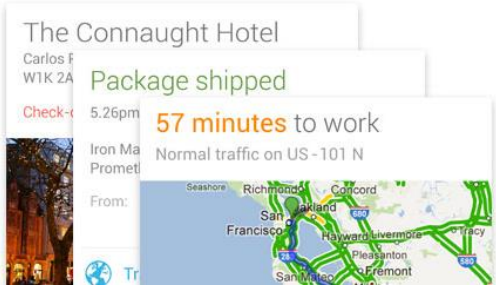
Example: Aware

- Android Mobile Context Instrumentation Framework (~2011)
  - Open Source Framework
  - www.awareframework.com
- Data collection from almost all Android sensors (Accessibility, API)
- Context creation and reusability based on Plugins
  - Presentation
  - Inference
- Context stored and shared
  - Broadcast – Android's BoradcastReceiver
  - Provider – pull from Content URI
  - Observer – push from ContentObserver
- Full-fledged AWARE applications



Context-aware Applications

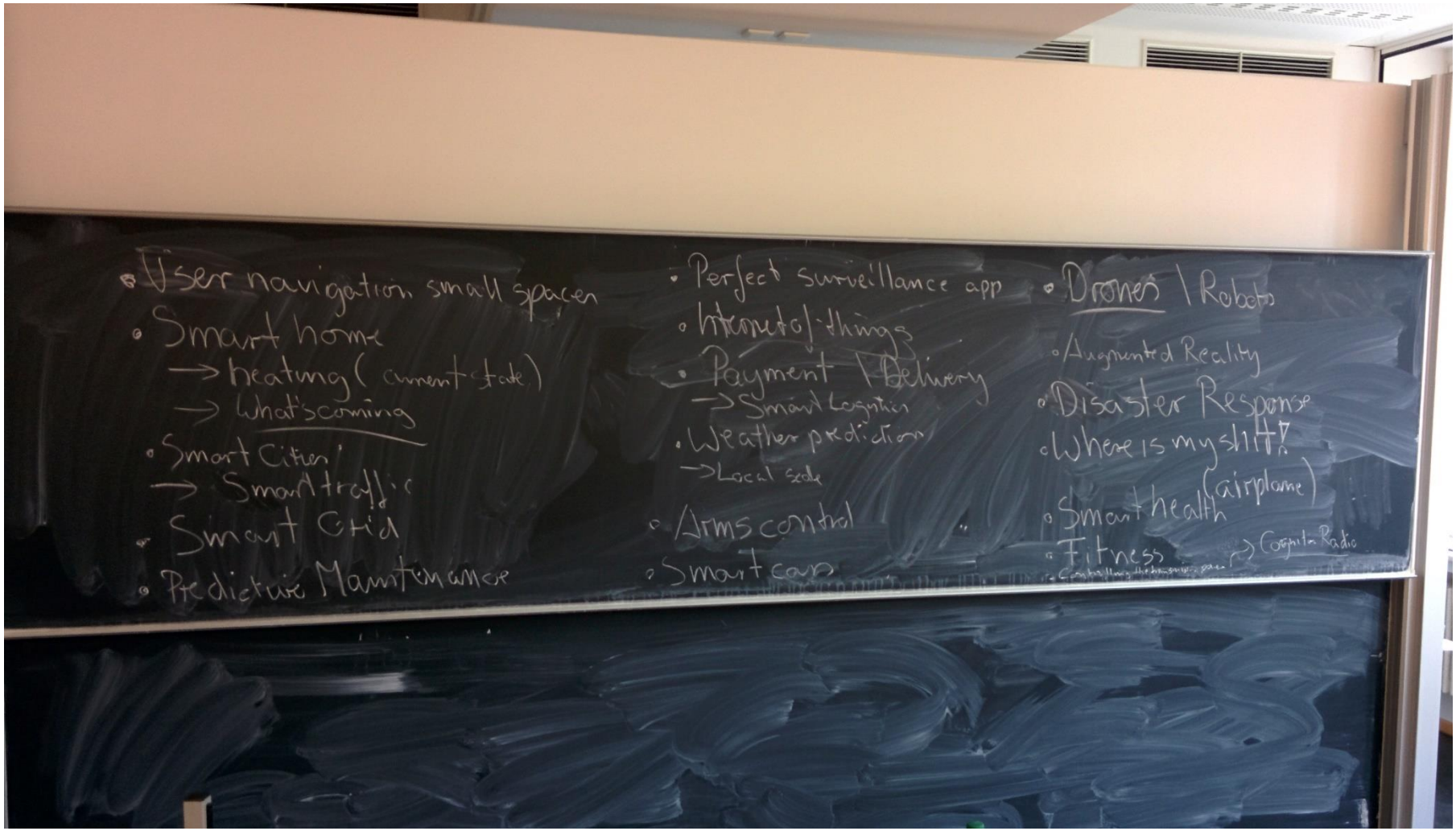
- **Presentation** of information and services to a user, e.g.:
  - Tourist Guides
  - ContextPhone: present information contacts (location, people nearby, phone use activity)
  - Google Mail / Facebook adverts
  - Google Now
- Automatic **execution** of a service for a user
  - PARCTAB System: Bind room resources to user on entering
- **Tagging** of context to information to support later retrieval
  - Add context at data acquisition time to improve later retrieval
  - Digital camera with GPS
  - CybreMinder: Reminder notes can be associated with location
- **Adaptation** of application's behavior and appearance
  - Automatically forward call to the phone in the vicinity of the user
  - Delay user interruption until an appropriate point in time
  - Highlight options/commands that best fit current needs



Context Architectures

- A basic context-aware model consists of four components:
  1. Current context capture (context determination)
  2. Goal context creation
  3. Adaptation of the current context to goal context
  4. Context management

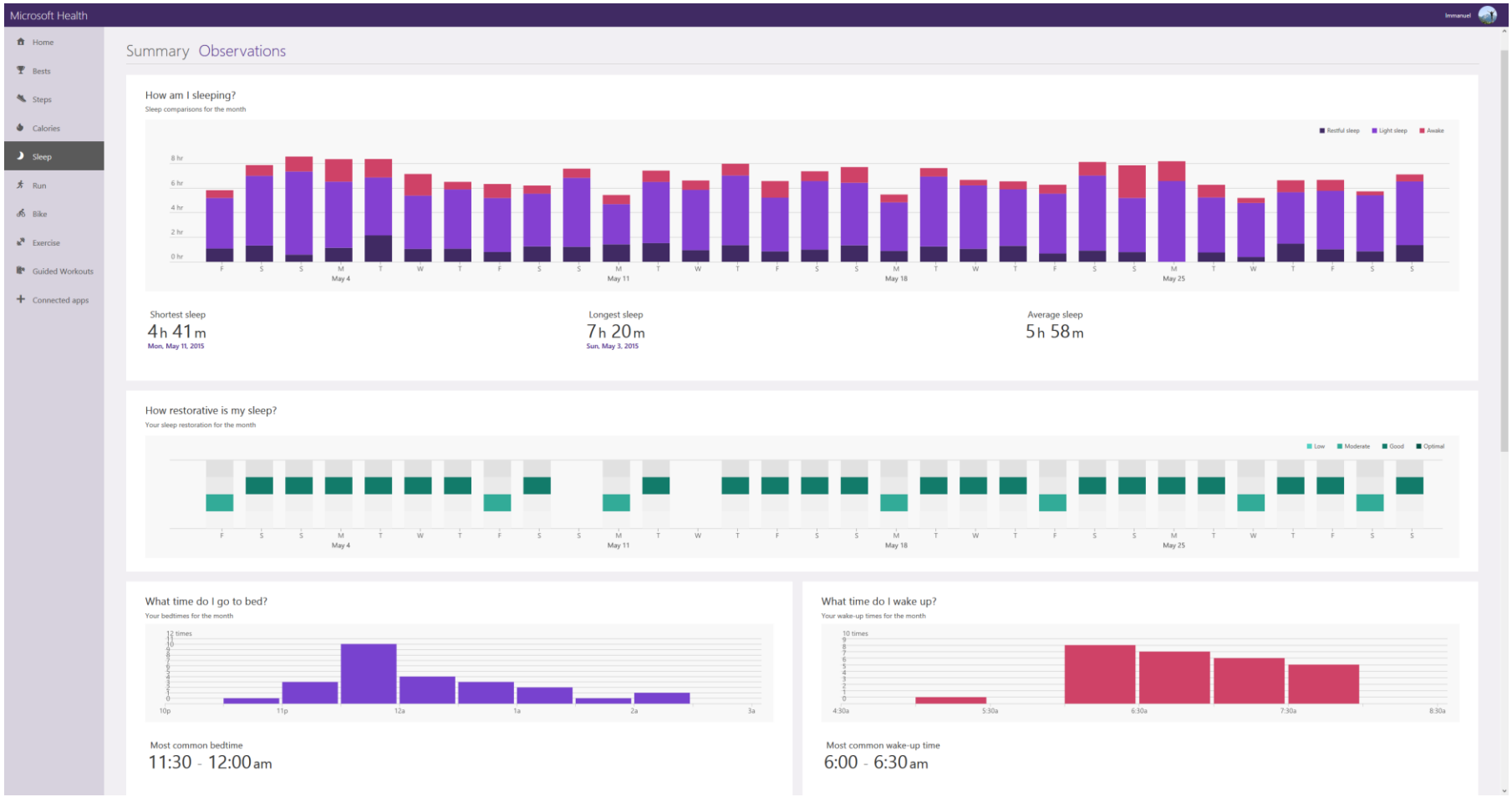
Examples


User navigation, small spaces
Smart home
→ heating (current state)
→ What's coming
Smart Cities
→ Smart traffic
Smart Grid
Predictive Maintenance
Perfect surveillance app
Internet of things
Payment / Delivery
→ Smart Logistics
Weather prediction
→ Local scale
Arms control
Smart cars
Drones / Robots
Augmented Reality
Disaster Response
Where is my shit?
Smart health (airplane)
Fitness
→ Cognitive Radio

Example: Robotics

Example: Autonomous Driving

Health

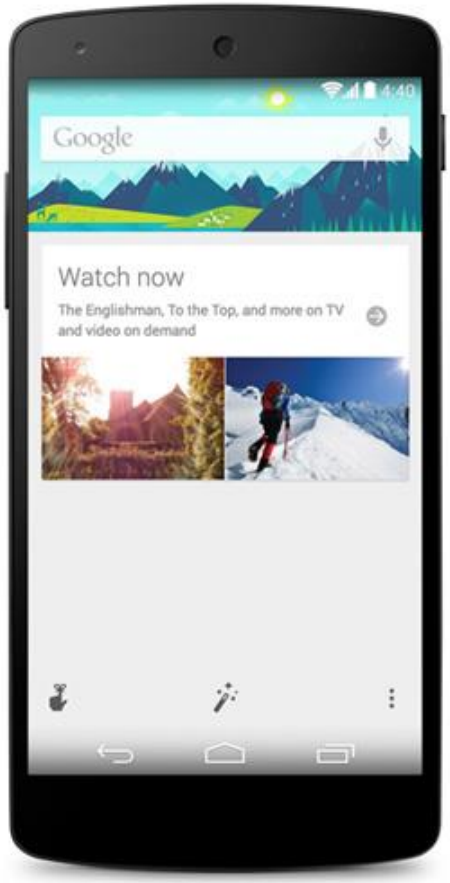

Microsoft Health
Immanuel
Home
Bests
Steps
Calories
Sleep
Run
Bike
Exercise
Guided Workouts
Connected apps
Summary Observations
How am I sleeping?
Sleep comparisons for the month
Restful sleep
Light sleep
Awake
8 hr
6 hr
4 hr
2 hr
0 hr
May 4
May 11
May 18
May 25
Shortest sleep
4h 41m
Mon, May 11, 2015
Longest sleep
7h 20m
Sun, May 3, 2015
Average sleep
5h 58m
How restorative is my sleep?
Your sleep restoration for the month
Low
Moderate
Good
Optimal
What time do I go to bed?
Your bedtimes for the month
12 times
10p
11p
12a
1a
2a
3a
Most common bedtime
11:30 - 12:00 am
What time do I wake up?
Your wake-up times for the month
10 times
4:30a
5:30a
6:30a
7:30a
8:30a
Most common wake-up time
6:00 - 6:30 am

Personal Assistance

TECHNISCHE UNIVERSITÄT DARMSTADT

Apple Siri

Google Now



Windows Cortana

Context: Summary

TECHNISCHE UNIVERSITÄT DARMSTADT

- Context Definitions
  - context is: the *situation* of usage of a (ICT based) system ☹
  - … situation means: **physical conditions** of operation a/o **human factors** 😐
  - **context** is information that is **relevant** (for application) and **auxiliary** 😐
- Context Categories
  - based on hard (physical world) or soft (digital world) sensors
  - sensed ('raw') or inferred ('higher level')
- Context Processing
  - signal processing for raw context (includes 'digital signals' for soft sensing)
  - further processing: 'widgets', rules&facts, machine learning, …
- Context Models
  - key-value, markup, ontology-based, object-oriented, logic-based
- Context Middleware
  - e.g., Context Toolkit
- Example Applications