Technische Universität Darmstadt

Telecooperation

Ubiquitous & Mobile Computing

Adaptability: Context

Dr. Erwin Aitenbichler

Why use context?

- Humans use context for adapting their behavior to the current situation (e.g. time of day, location, people they are with)

- Goal:
  - Applications, environments, … that reduce cognitive load of users

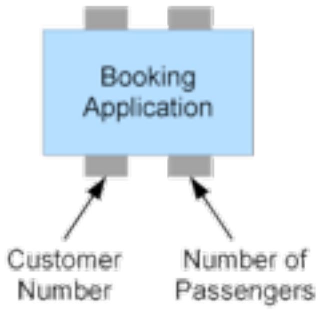
- How:
  - Proactivity
    - Setup environment according to user’s preferences or usage history
    - Auto-completion of forms (location, time in HEAG timetable)
    - Reminders
  - Search and filter information according to the user’s current needs
  - Avoid interrupting the user in inappropriate situations
  - Smart environments
    - Turn devices on/off, start applications, … depending on location, time, situation (lecture, meeting, home cinema, …)
    - Discover and use nearby interaction devices

What is context?

- Two prominent definitions in research literature:
- Context (Dey):
  - 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.
- Context (Schmidt/Gellersen):
  - **Abstrakte Beschreibung** der Situation oder von **signifikanten Merkmalen** der Situation in der lokalen Umgebung eines Benutzers.
  - System mit Kontextbezug:
    Ein System das die Fähigkeit hat, Aspekte der Umgebung als Kontext zu erfassen und diese für ein **situatives Verhalten** zu nutzen.

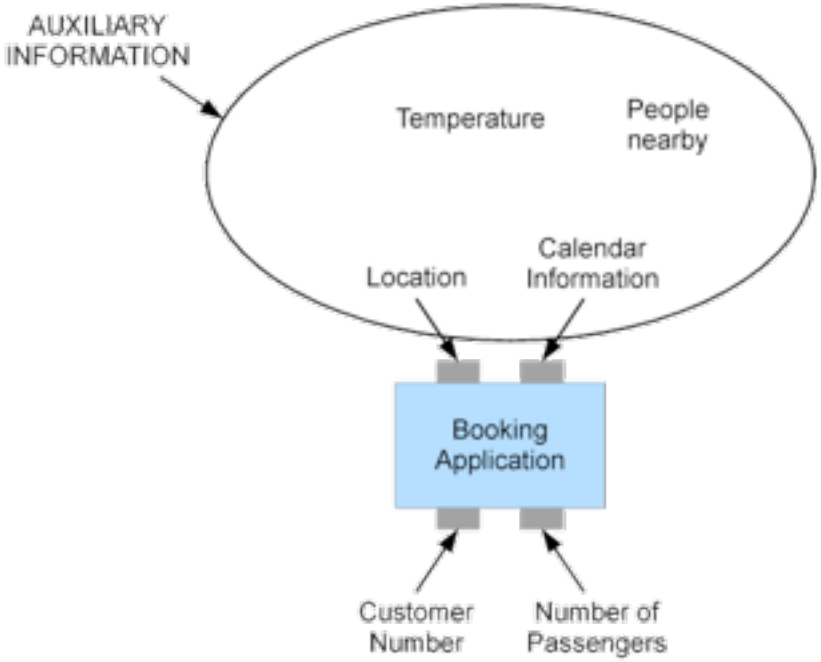
What is context?

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



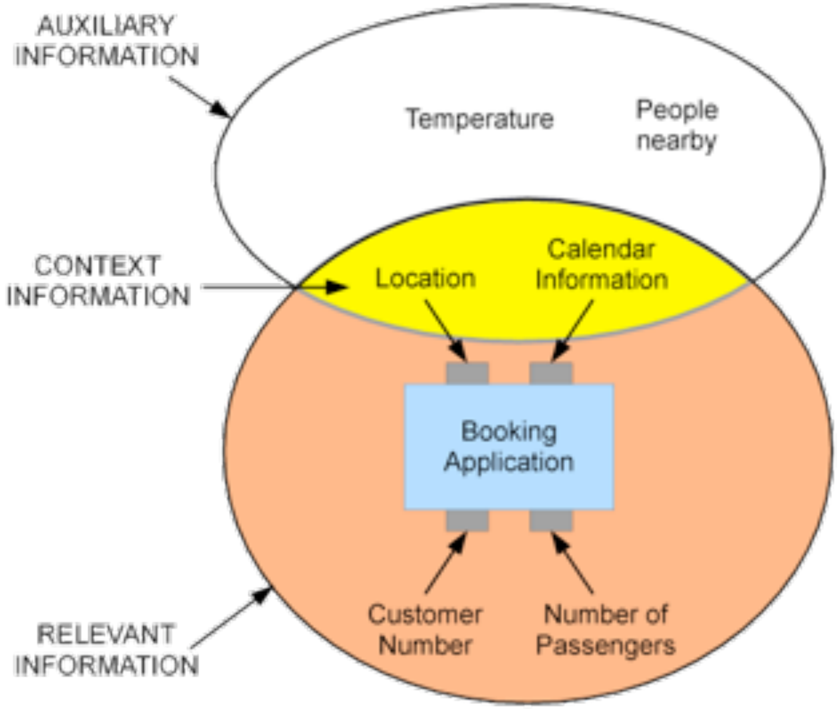
What is context?

- 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

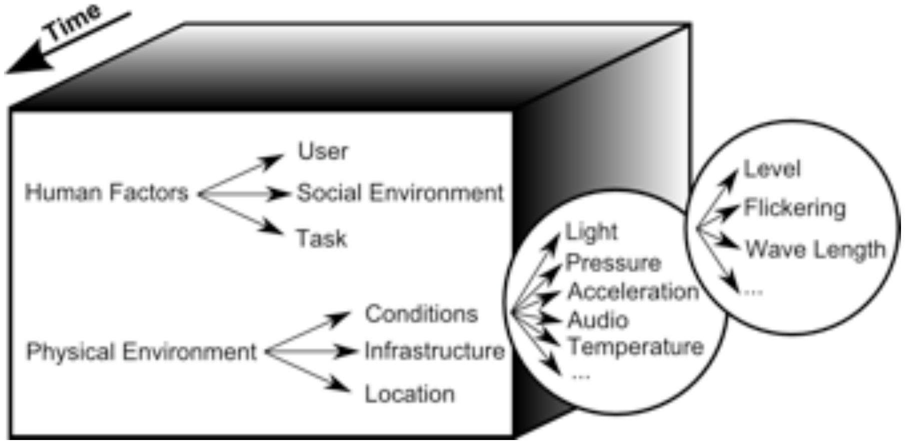


Definition

- Context (Mühlhäuser et al.):
  - Context characterizes the actual situation in which an application is used.
  - 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**).
- Context-aware system: system that adapts itself to context

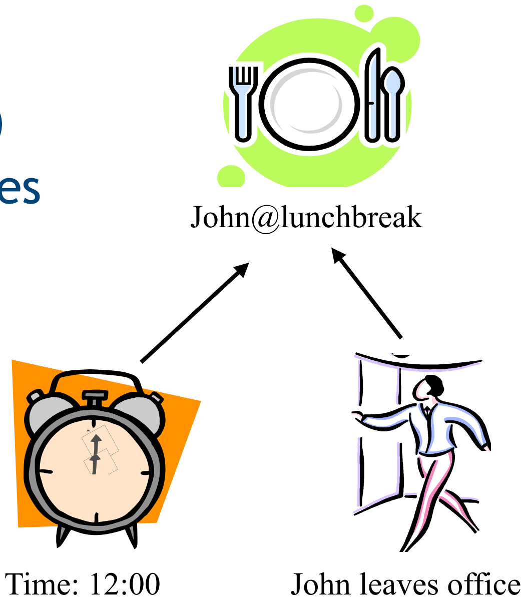
Examples

- 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
- Physical environment
  - location
    absolute or relative position
  - infrastructure
    surrounding resources for
    computation, communication
  - physical conditions
    noise, light, pressure



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 Models

- 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

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

Example:

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

Markup Scheme Model

- 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 device screen size defined in profiles
  - CC/PP Context Extension: Network Interfaces of devices, Location, DisconnectionStatus, …

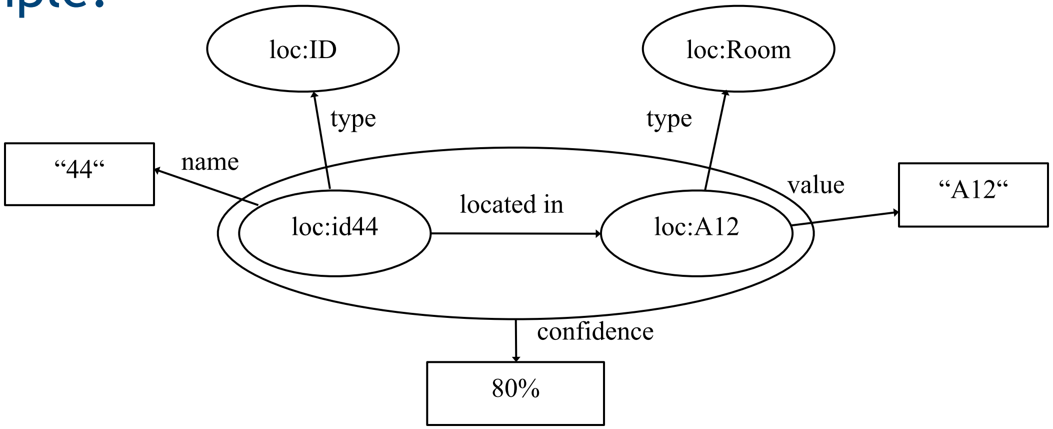
Example:

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

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

Example:



Object-oriented Model

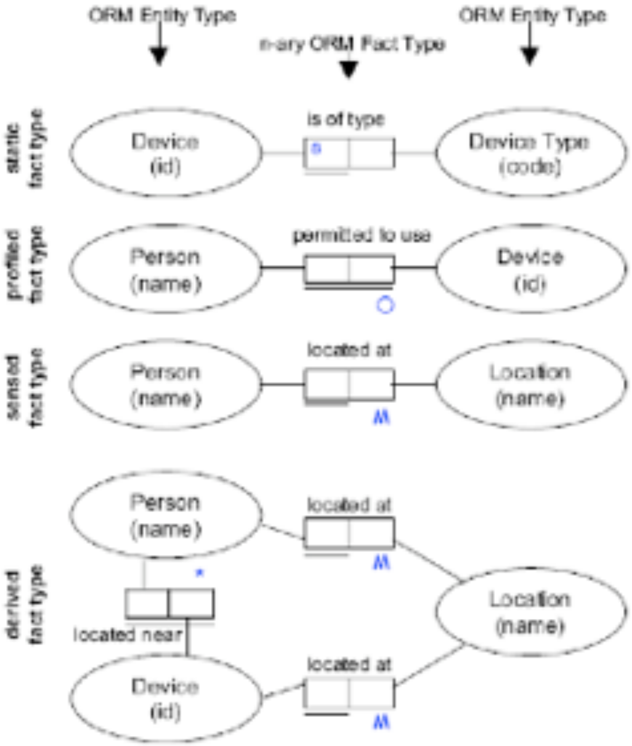
- Allows encapsulation and reuse of parts of the model
  - Entities and relations modeled as objects
  - Processing/reasoning done by “widgets”

Example:

Location

```
Room = A12
ID = 44
Confidence = 80%
```

- Representation of context, e.g., by Object-Role Modeling
  - “Typed” relation between classes → fact types
  - Instances are called facts



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

Example:

```
locatedAt(“44”, “A12”, 80%)
```

Dealing with Uncertainty

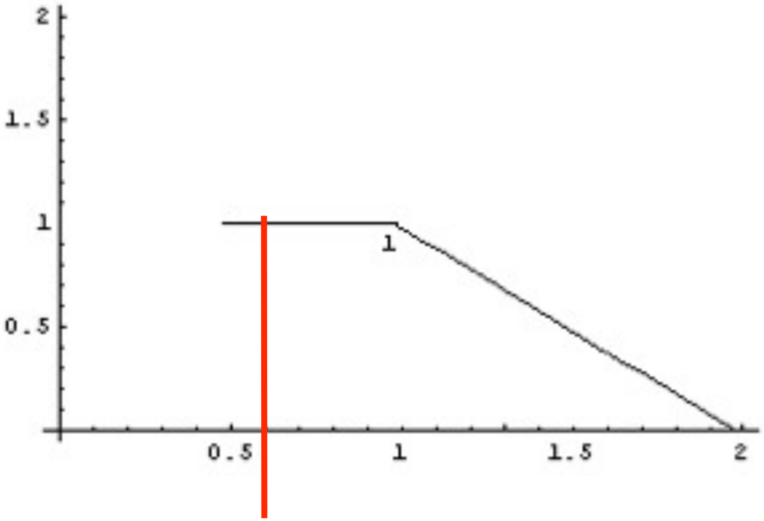
- Has to be handled in three areas:
  - Sensing context information
  - Inferring context information
  - Using context information
- How to determine uncertainty of sensed context
  - can be reported by sensor (e.g. biometric authentication devices give a measure for the confidence in reported data)
  - 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

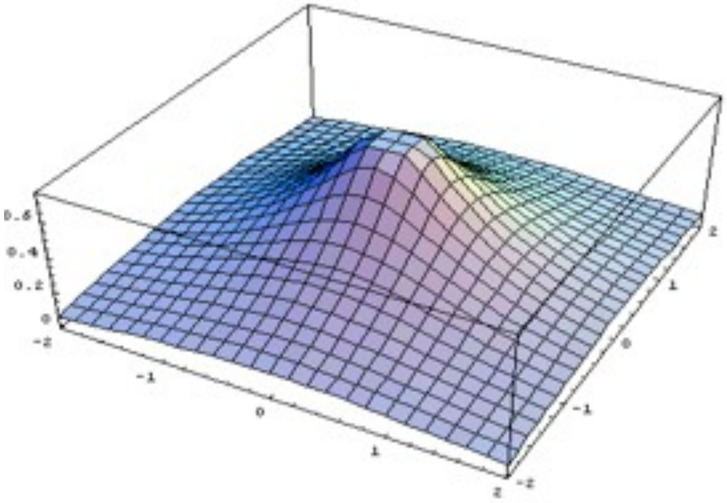
FuzzySpaces: Assumptions

- principle of location
  - context has place of origin
  - relevance is max. at origin, drops with distance towards 0, e.g.:
    - temperature measurement accurate at thermometer
    - temperature is similar “nearby”
  - multiple sensors -> nearest has maximum relevance
- principle of time
  - context has time of origin
  - then relevance is maximal
  - relevance drops with time towards 0
  - multiple sensors -> latest acquisition has maximum relevance
- principle of independency
  - context producer and consumer are independent
  - producers of (even the same) context exist independently
  - consumers of context exist independently
  - applications use context

FuzzySpaces

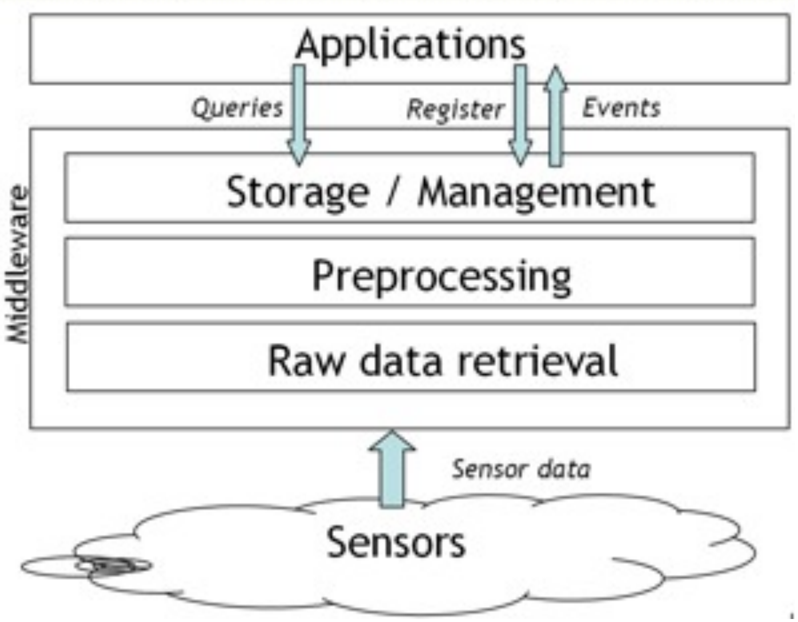
- Observations influenced the context model:
  - context C = (ID, description, unit, range of values, value, probability)
- Validity of context described by functions determining
  - temporal relevance
  - location relevance





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

- 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

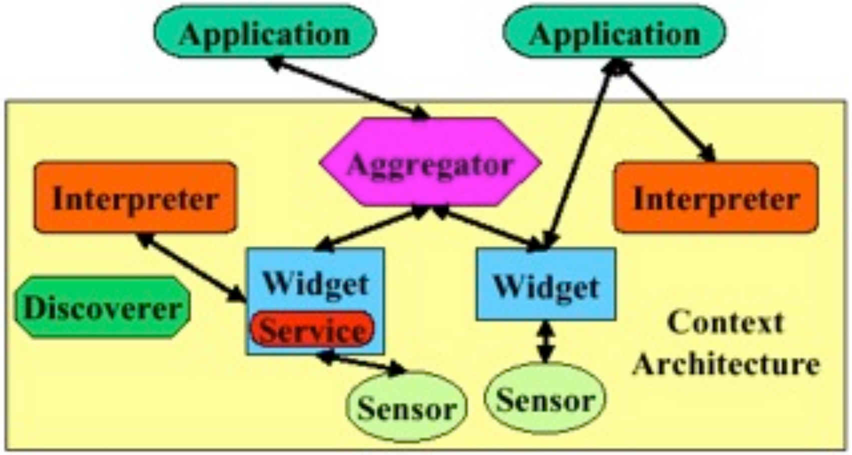
  Manages gathered data and offers public interface to the client applications

  Answers queries and notifies interested applications about events

  Stores the context history

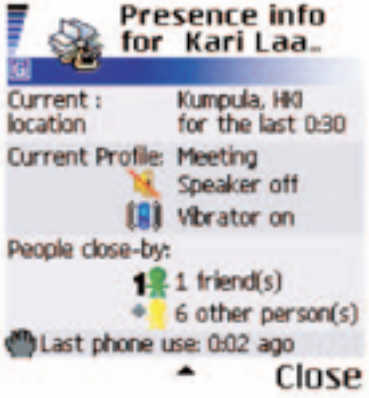
Example: Context Toolkit

- 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



Context-aware Applications

- **Presentation** of information and services to a user
  - Tourist Guides
  - ContextPhone: present information contacts (location, people nearby, phone use activity)
  - Google Mail displays advertisements according to email content



Presence info for Kari Laa..

Current : location — Kumpula, HKI for the last 0:30

Current Profile: Meeting
Speaker off
Vibrator on

People close-by:
1 friend(s)
6 other person(s)

Last phone use: 0:02 ago

- 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

Example Applications

- Google Mail displays advertisements according to the email content (http://mail.google.com)
- Use mobile phone to get traffic information for the current location without having to enter it explicitly
- One of the first context-aware applications was the Active Badge System [Harter 1994] to locate people in the office and forward calls to a nearby phone
- Most popular applications: location based services (e.g. GUIDE a travel guide system for Lancaster [Cheverest 2000])
- Memory-aids: provide user with reminders or information recorded from previous interaction that could be relevant in the current situation (e.g. Remembrance Agent [Rhodes 1997])

Context: Summary

- Context Definition
  - **context** is information that is **relevant** (for application) and **auxiliary**
- Context Feature Space
  - human factors, physical environment, …
- Context Categories
  - Sensed and inferred context
- Context Models
  - key-value, markup, ontology-based, object-oriented, logic-based
- Dealing with Uncertainty
  - FuzzySpaces
- Context Middleware
  - Context Toolkit
- Example Applications