Application Development for Mobile and Ubiquitous Computing

4. Context Awareness

Dr. Ing. Thomas Springer
Technische Universität Dresden
Chair of Computer Networks
Application Development

Cross-Platform Development

- Java ME
- Android
- iOS
- .Net Compact Framework/Windows Phone 7
- OSGi

Mobile Middleware

- Disconnected Operations
- Mobile Databases
- Location-based Services
- Communication Mechanisms

Mobile Internet

Enabling Technologies and Challenges
Structure

- Context Definition
- Properties of Context
- Context Gathering
- Approaches for Context Abstraction
- Context Modelling
- Middleware for Context-Awareness
Motivation

- Mobility and location
- Computing environment, device and network resources
- Interaction capabilities and user attention
- Proactive, autonomous systems
- Adaptive system behaviour
- Personalized and simplified user interactions
- Situation/location dependent information
- Situation, social environment and activity

context-aware applications
control of adaptation processes
Context Definitions

- “... all but the explicit input and output of an application” [LiS00]

- “... that which surrounds, and gives meaning to, something else” [o.V.00]

- “Context is a subjective concept that is defined by the entity that perceives it" and „contextual states [...] are inherently associated with specific objects” [Pascoe]

- Context is an operational term: Something is context because of the way it is used in interpretation, not due to its inherent properties. [...] Features of the world become context through their use. [Winograd]
Hierarchical and Extensible Context Categories

- The categories of *human factors* and *physical environment* at the top level are further classified into *user, social environment, task and conditions, infrastructure, location* at the second level [Schmidt, Beigel, Gellersen].

- **Context-aware computing is the ability of a mobile user's applications to discover and react to changes in the environment they are situated in** [Schilit and Theimer, Xerox Parc]:.

- Three important aspects:
  - Where you are? (lighting, noise level, traffic conditions, temperature)
  - Who you are with? (user's profile, people nearby)
  - What resources are nearby (connectivity, printers, displays)
“Context could be generally described as the subset of physical and conceptual states of interest to a particular entity.” [Pascoe]

“Context is any information that can be used to characterise 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” [Anind K. Dey]

Contextual information is related to a certain entity
# Types of Contextual Information

<table>
<thead>
<tr>
<th>Contextual dimension</th>
<th>Respective contextual information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Physical context</strong></td>
<td>Location, time, temperature, light and noise intensity, nearby persons</td>
</tr>
<tr>
<td><strong>Technical context</strong></td>
<td>Network (bandwidth, latency, error rate), Device (input and output capabilities, memory, software support), available services, service preferences</td>
</tr>
<tr>
<td><strong>Personal context</strong></td>
<td>Address, phone number, payment, preferences, schedule, service preferences</td>
</tr>
<tr>
<td><strong>Social context</strong></td>
<td>Nearby persons, groups (teams) to which the user belongs</td>
</tr>
<tr>
<td><strong>Operational Context</strong></td>
<td>Roles, activities, to-do-items, content of the inbox of the user</td>
</tr>
</tbody>
</table>
Two Categories of Information

- **State Information**
  - Application actively requests required context (pull)
  - Access to actual and historical data
  - E.g. current location, device, etc.

- **Change Events**
  - Application registers for particular change events, Waits passively for the events
  - Context service notifies registered applications about changes of state (push)
  - E.g. location changes, access network changes

<table>
<thead>
<tr>
<th>state</th>
<th>event</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>information</strong></td>
<td>person X enters the room</td>
</tr>
<tr>
<td>current location of user, list of printers nearby</td>
<td>start communication with person x after she has entered the room</td>
</tr>
<tr>
<td><strong>action</strong></td>
<td>print a document using the closest printer</td>
</tr>
</tbody>
</table>

Dr. Thomas Springer

Application Development - 4. Context Awareness
Technology for Enabled Awareness (TEA), TEA II

- **Sensor placement**
  - Mobile device equipped with sensors

- **Application Scenario**
  - Sensing the situation of a mobile device and its user
    - Location and usage of mobile phone
  - Automatic selection of phone profile

- **Used Sensors**
  - **TEA I:**
    - photodiode, two accelerometers, passive IR sensor, temperature, pressure, CO gas, omni-directional microphone
  - **TEA II:**
    - two photodiodes, two microphones (miniature electrolet capsules usually used in mobile phones), a dual axis accelerometer (ADXL202 from Analog Devices), a digital temperature sensor (Dallas Semiconductor DS1820), touch sensor
    - board extensible with further sensors (slots available on the board)
4-layered model:

1. Sensor (physical, logical)
2. Cue (processing of one sensor output)
   - Contains Array of Values, Operations on these values, e.g. average, standard deviation
3. Context (describes disjunct situations together with probability)
   - Description of the current situation on an abstract level
   - Based on logic rules operating on cues
4. Application level (Situation and application specific)
   - Definition of actions based on scripts, triggered if a certain situation is entered or left with a given probability
Cues

- Computing method: statistical functions

- Average value (for values of single sensor over about 1 min., applied for light, acceleration, temperature, and pressure)

- Standard derivation (for values over about 1 min., applied for light, passive IR, acceleration)

- Base frequency (applied for light and acceleration, types of light (flickering), acceleration pattern for moving)

- First derivative (applied for passive IR (Infra Red) and acceleration)
- **Exclusive contexts**
  - Rule set for recognizing the situations that device is situated in Hand, Table and Suitcase based on the sensors light, and acceleration in two directions (X and Y)
  - Constants Dx, Dy, L, Xnormal, Ynormal, D, and Q defined based on observations
  - Location of mobile phone: hand/table/suitcase
  - Device usage: inside/outside, car/bus/train, stationary/moving car

| Hand(t):=- | standard deviation(accelX,t) > Dx, standard deviation(accelY,t) > Dy, % device is slightly moving in X and Y average(light,t)>L. % not totally dark |
| Table(t):- | abs(average(accelX,t)-Xnormal)<D, abs(average(accelY,t)-Ynormal)<D, % the device is level in X and Y quartile(accelX,t)<Q, quartile(accelY,t)<Q % the device is stationary average(light,t)>L. % not totally dark |
| Suitcase(t):- | average(light,t)<L. % it is totally dark |
Sensor placement
- Mobile device equipped with low-cost sensing elements

Application Scenario
- Environmental sound intensity controls volume of operating tones
- Font size, screen brightness and service content are adapted according to user activity and ambient light level

Used Sensors
- microphone, accelerometers, two channels for light, sensors for temperature, humidity, touch

Computed higher-level context
- Location {indoor, outdoor}
- SoundType {Car, Elevator, RockMusic, ClassicalMusic, TapWater, Speech, OtherSound}
Computing method

- 2 Phases
  - 1. Phase: abstraction of raw sensor data (comparable to cues)
  - 2. Phase: computing higher-level context using naive Bayesian networks

- First Phase:
  - Crisp limits: true-false labeling of sensed data (e.g. Silent, Moderate, Loud for environment sound intensity)
  - Fuzzy sets: overlapping ranges (e.g. 0.7/Silent, + 0.3/Moderate + 0/Loud)
Second Phase: naive Bayesian network
### Rule-based Reasoning

#### Use Case: Smart Home

#### Modeling:
- CONON Ontology (CONtext ONTology)
- OWL, RDF Triples

#### Reasoning:
- based on OWL light and RDF triples
- Rule-based Reasoner (simple pattern matching algorithms)

<table>
<thead>
<tr>
<th>Situation</th>
<th>Reasoning Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleeping</td>
<td>(?u locatedIn Bedroom) ^ (Bedroom lightLevel LOW) ^ (Bedroom drapeStatus CLOSED) ⇒ (?u situation SLEEPING)</td>
</tr>
<tr>
<td>Showering</td>
<td>(?u locatedIn Bathroom) ^ (WaterHeater locatedIn Bathroom) ^ (Bathroom doorStatus CLOSED) ^ (WaterHeater status ON) ⇒ (?u situation SHOWERING)</td>
</tr>
<tr>
<td>Cooking</td>
<td>(?u locatedIn Kitchen) ^ (ElectricOven locatedIn Kitchen) ^ (ElectricOven status ON) ⇒ (?u situation COOKING)</td>
</tr>
<tr>
<td>Watching TV</td>
<td>(?u locatedIn LivingRoom) ^ (TVSet locatedIn LivingRoom) ^ (TVSet status ON) ⇒ (?u situation WATCHINGTV)</td>
</tr>
<tr>
<td>Having Dinner</td>
<td>(?u locatedIn DiningRoom) ^ (?v locatedIn DiningRoom) ^ (?u owl:differentFrom ?v) ⇒ (?u situation HAVINGDINNER)</td>
</tr>
</tbody>
</table>
Usually distributed processes
Requires exchange/sharing of context
Context Sources are:

- sensor devices (e.g. GPS, temperature, light intensity, noise)
- databases (e.g. user DB of service/access provider)
- applications (e.g. scheduling app)
- user monitoring and input (service or platform GUI)
Context Capturing
Properties of Context Sources

- Heterogeneous
  - Technology – sensors, data bases, application data
  - Access – SQL, sensor driver, extraction code
  - Data representation – application dependent data structures, GPS coordinates
  - Semantics – equally named data with different meaning, differently named data with equal meaning
  - Granularity – user db contains comprehensive user information, GPS device provides GPS coordinates only
  - Level of abstraction – Sensor provides raw data, scheduling application provides high level date information
  - Owner – user, network provider, building owner
Context Capturing Properties of Context Sources

- Distributed
  - mobile device (location, schedule data, personal information)
  - Infrastructure (device/connectivity information, service profile)
  - multiple alternative sources

- Source architecture
  - centralized – user data base
  - partitioned – user data base (A-E, F-J, K-P, ...)
  - distributed – temperature for certain rooms

- Relevance
  - user specific – schedule, user preferences
  - application specific – buddy list
  - domain specific – patient data
  - generic – temperature, light intensity

- Users have their own configuration of context sources
Context Abstraction - Generalization

- **Interpretation**
  - e.g. raw sensor data to temperature in K

- **Alternatives**
  - e.g. choice between alternative values for one context value
  - e.g. dependent on different data qualities or availability

- **Merging**
  - e.g. location information based on GSM and signal strength of WLAN access point
  - combined to more exact location information

- **Derivation**
  - e.g. user logged in on local device and device is connected -> user online
Dr. Thomas Springer

Application Development - 4. Context Awareness

Context Abstraction - Levels of Fusion
Properties of Context Attributes

- **Time dependent**
  - Context represents dynamic information
  - Values change over time
  - Static information like date-of-birth can be interpreted as information with change frequency of zero

- **History**
  - Represents values at different points in time

- **Quality**
  - Uncertainty in measurements, application of heuristics, assumptions for derivation and interpretation

- **Incorrectness**
  - Due to unexact sensor information, measurement failures
  - Wrong assumptions for derivation and interpretation

- **Multiple sources**
  - Same information can be gathered in different ways
  - E.g. location of a person (GPS, Position of device, WLAN)
Properties of Context Attributes

- **Relevance**
  - capturing time
    - highest relevance at capturing time
    - decreases constantly
    - e.g. location of a mobile user
  - location
    - measured temperature
    - highest relevance at capturing place
    - decreases with distances
Properties of Contextual Information

- Multidimensional/Heterogeneous
  - Physical/technical context
  - Private/social/business
- Distributed
  - Contextual information occurs everywhere/all the time
- Imperfect
  - Incomplete
  - Inconsistent
- Unforeseeable
  - Any information can be relevant as context
Context Models

- Common understanding between system components
- Sharing of context information between applications/systems
- Model for application development
- Approaches
  - Context Profiles
  - Object Oriented Models
  - Ontology-based Models
Context Profiles

- Name/Value Pairs:
  - username = Thomas Springer, temperature = 21 C

- Profiles
  - triples – subject, predicate, object
  - <http://www.examplepage.de> - author – „Peter Müller“
  - RDF (Resource Description Framework)
  - CC/PP (Composite Capabilities/Preference Profiles)
  - CSCP Comprehensive Structured Context Profiles
Real-world objects modelled based on entities with attributes
Relations between attributes described by associations
Object Oriented Model - (Henricksen, Indulska)

- Real-world objects modelled based on entities with attributes
- Relations between attributes described by associations
Object Oriented Model – Quality
Ontology-based Context Models

- “An ontology is an explicit specification of a conceptualisation” [Gru93]
- describes real-world as formalized concepts
- consists of concepts and roles
- individuals as instances of concepts

- Web Ontology Language (OWL)
  - defined by W3C
  - based on RDF
  - 3 variants: OWL lite, OWL DL, OWL Full
CONON – CONtext ONtology

Ref: Infocom Research, Singapore (Zhang)
CONON – Domain-Specific Ontology

Legend:
- Upper Class
- Specific Class
- owl:Property
- rdfs:subClassOf
CONON- Quality Model

a) A Quality Ontology

Legend:
- owl:Class
- owl:Property
- rdfs:subClassOf

b) An Example Instance for Location Information

- Accuracy-Loc
- Resolution-Loc
- StandardError
- Resolution

- null
- float
- 0.79
- meter
- integer
- 50

Application Development - 4. Context Awareness
- Decoupling of applications and context sources
- Reuse of generic functionality
  - Gathering, abstraction, storage, transport, application access
  - Mediates contextual information between context sources and context users
- Server can provide high performance and storage capacity to process context
- High availability of information
- Central storage for user profiles, information available independent of terminal devices
- Central access control
- Limited scalability due centralized resource for high amount of clients
- Network connection and infrastructure required between mobile device and server for context access
Architecture: Peer-to-Peer

- Independent from network connection to certain server
- Support of ad-hoc scenarios
- High scalability due to distribution of context access
- Availability of information depends on device availability, limited availability of user profiles
- Limited performance and storage capacity on mobile devices for context processing
- No central control to context access
Basic Abstractions – Context Toolkit

- **Widgets:**
  - Sensor abstraction
  - Represent single context value
  - Encapsulate details of sensors and other sources
  - Current value, history, subscriptions

- **Aggregator**
  - Assign several context data (widgets, interpreters) to entities
  - Easier management

- **Interpreter**
  - Processing of sensor data
  - Generation of higher-level context
  - One or several input values
  - Can be used by Widgets, Aggregators, Interpreters and Applications
Sentient Computing

- System Building blocks are Sentient Objects (SO)
- Each SO represents context-aware functionality of gathering, processing or applying of contextual information
- Different types of objects
  - sensors
  - transformer
  - actuator
- Sentient objects contains internal representation of contexts as well as transformation code (e.g. inference mechanisms)
- Combination of sentient objects to context-aware applications
Main principles are:
• Decentralization
• Local handling of context sources
• Local access to context sources
• Abstract view on context sources (API)
Integration of Context Sources

a) ContextSourceAPI

<table>
<thead>
<tr>
<th>Wrapper Domain A</th>
<th>Wrapper Domain B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source specific</td>
<td>Access</td>
</tr>
</tbody>
</table>

- Database
- Sensor device
- File
- Application data

b) Local Source

c) Remote Source

d) Aggregated Source

- Context Source
- Context Source

e) Fusion Source

- Context Source
- Context Source
- Meta – model is the basis for context model and service implementation
- Derived from Topic Maps
- Enables the representation of arbitrary contextual information
Layered Context Model

- Integrated approach combining the aspects of modeling and runtime support
- Context model is shared between services, context service and sources
Structural context model (Domain-specific)

— Dr. Thomas Springer

Application Development - 4. Context Awareness

Slide 45
### ContextAwareness

<table>
<thead>
<tr>
<th>event type</th>
<th>relevant to class</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>change</td>
<td>ValueGroup, ContextValue, Entity, Association</td>
<td>an event is created if the value or at least one of the values of a ValueGroup or Entity change in any way</td>
</tr>
<tr>
<td>enter</td>
<td>ContextValue</td>
<td>an event is created if the value changes from any value different to that given value</td>
</tr>
<tr>
<td>leave</td>
<td>ContextValue</td>
<td>an event is created if a value changes from a given value to any other value</td>
</tr>
<tr>
<td>buildup</td>
<td>Association</td>
<td>an event is created if a new association is built up</td>
</tr>
<tr>
<td>destroy</td>
<td>Association</td>
<td>an event is generated if an existing association is destroyed</td>
</tr>
</tbody>
</table>

**Diagram Description**

- **ContextAttribute**
  - Type: String
  - +subscribeToEvent:boolean
  - +unsubscribeFromEvent:void

- **ValueGroup**
  - +ValueGroup
  - +getAttribute:ContextAttribute
  - +subscribeToEvent:boolean
  - +unsubscribeFromEvent:void

- **ContextValue**
  - -Unit: String
  - -Constraint: String
  - -Value: Object
  - -Dvalue: Object

- **Entity**
  - -Type: String
  - -AssociationList: Vector
  - +Entity
  - +getAttribute:ContextAttribute
  - +addAssociation:void
  - +getAssociation:Association
  - +subscribeToEvent:boolean
  - +unsubscribeFromEvent:void

- **Association**
  - -Type: String
  - +Association
  - +getAssociationEnd:Entity
  - +subscribeToEvent:boolean
  - +unsubscribeFromEvent:void

**Event Types and Relevant Classes**

- **change**
  - Relevant to: ValueGroup, ContextValue, Entity, Association
  - Description: an event is created if the value or at least one of the values of a ValueGroup or Entity change in any way

- **enter**
  - Relevant to: ContextValue
  - Description: an event is created if the value changes from any value different to that given value

- **leave**
  - Relevant to: ContextValue
  - Description: an event is created if a value changes from a given value to any other value

- **buildup**
  - Relevant to: Association
  - Description: an event is created if a new association is built up

- **destroy**
  - Relevant to: Association
  - Description: an event is generated if an existing association is destroyed

---

**Dr. Thomas Springer**

**Application Development - 4. Context Awareness**
Summary

- Context information enables awareness of execution environment
- Pre-requisite to control adaptation processes
  - Usually mapping between context information and control parameters/information necessary
- The major phases of context processing
  - Gathering
  - Abstraction
  - Decision making
- Context models allow sharing of context
- Context middleware provides general functionality to gather, process, store, distribute and access context information
References