Context modeling and reasoning

Key concepts for Pervasive computing

Daniela Nicklas
Institute of Parallel and Distributed Systems

Center of Collaborative Studies Nexus: Spatial World Models for Mobile, Context-aware Applications (SFB 627)
SFB 627 (Nexus): World Models for Mobile Context-Aware Systems

**Goals:**
- Methods to realize global world model
- Innovative context-aware
  - applications
  - mechanisms

**Start:** January 2003 at University of Stuttgart
- based on the preparatory work of DFG research group neXus
- 30+ research staff members, 9 research groups
- now in second funding period (2007-2010)
- Speaker: Kurt Rothermel, rothermel@informatik.uni-stuttgart.de
Nexus Overview

Communication

- Methods for Access Optimization
- Model Based Communication

Definition and Management of World Models

- Extensibility
- Federated Model Management
- Integration and Abstraction of Geographic Data

Model Presentation and Sensors

- Context Sensitive Model Presentation
- Generation of Model Information, Sensor Integration

Applications

- Campus Information System
- Smart Factory
- Navigation for Visually Handicapped
- ... (Industry Partners)

Application Development Research

- Process-based applications (context-aware workflows)
- higher level context
- context ontogies and semantic
- quality of context

Modeling and Applications

Security and Acceptability

Quality of Context
Overview

- Why?
  - Pervasive computing and context-aware applications
- What is context?
  - Definitions
  - Types
- Context modeling
  - Modeling approaches
  - Location models
- Context reasoning
- Application examples
Pervasive computing ...
Pervasive computing ... wer hats erfunden?

Marc Weiser, (1952 - 1999)

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it."

Pervasive computing ...

IEEE International Conference on Pervasive Computing and Communications (www.percom.org):

- all the time, everywhere services
- outcome of the tremendous advances in
  - wireless networks
  - mobile computing
  - sensor networks
  - distributed computing
  - agent technologies
Pervasive computing … also known as:

- Ubiquitous Computing
- Ambient Intelligence
- Context-aware Applications
- Sentient Computing
- Smart * (Room, Space, Factory, ...)
- Disappearing Computers
Context-aware applications
A clear trend

one computer, many users

one user, many computer

Situation work place  everyday situations
Poor device

Combination of context information
Context models

- **Physical world**
  - **Sensors (Fusion)**
    - **Update(id, value)**
  - **Application**
    - **State**
      - **Update(id, value)**
  - **Context model**
    - **Applications**
      - **query (filter)**
      - **event (condition)**
Typical context-aware applications

- **Navigation**
  - vehicles, pedestrians, multi-modal
- **Information systems**
  - tourist guides, support of field staff, ...
- **Communication services**
  - GeoCast, best network, ...
- **Edutainment**
  - Rallys, "mixed reality games"
- **Smart Environments**
  - smart room, building, factory
- **Remember services**
  - Do not forget; lifelog
- **Collaboration**
  - Rescue teams, medical applications, ...
Two classes of context-aware applications

mobile computing

"smart" environments
Query: How do I get to the closest hotel? (No single source for all information!)
 Examples: Smart Spaces
○ Easy Living (1998, MS Research)
○ Aware Home (2000, GeorgiaTech)
○ Sentient Computing (2001, AT&T Research)
○ Smart Factory (2003, Uni Stuttgart)
○ Future Store (2003, Metro)
○ MuSAMA (2006, Uni Rostock)
○ …
Four categories of context-based adaptation

- **Selection** of information and services
  - e.g., the nearest restaurants

- **Presentation** of information and services to a user
  - e.g., navigation: route visualization based on speed (text, map, or arrow)

- **Action**, automatic execution of a service for a user
  - e.g., choice of wireless communication or services within a smart room scenarios

- **Tagging** of context to information for later retrieval
  - e.g., Stick-E-Note, Virtual Information Towers, Forget Me Not, ...
Overview

- Why?
  - Pervasive computing and context-aware applications
- What is context?
  - Definitions
  - Types
- Context modeling
  - Modeling approaches
  - Location models
- Context reasoning
- Application examples
What is context?

Definitions
Context types
Context Definitions – Merriam Webster

- **context**

  *context*ere to weave together, from *com-* + *texere* to weave

  1: the parts of a discourse that surround a word or passage and can throw light on its meaning

  2: the interrelated conditions in which something exists or occurs
Context – Schilit, Adams, Want

- Three important aspects of context are
  - where you are,
  - who you are with, and
  - what resources are nearby.

"Context-aware systems adapt according to the location of use, the collection of nearby people, hosts, and accessible devices, as well as the changes to such things over time. A system with these capabilities can examine the computing environment and react to changes in the environment."

Context is the set of environmental states and settings that either determine an application’s behavior or in which an application event occurs and is interesting to the user.

- **Active context awareness**: an application automatically adapts to discovered context, by changing the application’s behaviour.
- **Passive context awareness**: an application presents the new or updated context to an interested user or makes the context persistent for the user to retrieve later.

---

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 applications themselves.

- 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.

A “working model for context”

Context describes a situation and the environment a device or user is in

Context is identified by a unique name

For each context a set of features is relevant

For each feature a range of values is determined
Context feature space – Schmidt, Gellersen, Beigl

Context definitions – summary

- A variety of definitions so far
- Context as concept well understood
- Common aspects of definitions
  - Context influences applications
    - Presentation
    - Processing
  - Context relates (also) to the physical world
  - Context involves the user
Context types

- Different characterizations:
  - content, source
  - primary/secondary
  - public/private
  - level of interpretation (lower/higher)
Context: content/source

- Geographic context (static)
  - map data, rooms, ...
- Dynamic context (sensed)
  - obtained from sensors: temperature, location, light, ...
- Information context (derived or static)
  - documents, virtual objects, ...
- Technical context (profiled or static)
  - available services, infrastructure, location and interfaces of sensors, device features, ...
- User context (profiled)
  - User's profile, current activity, ...

Italic terms are from Henricksen/Indulska's CML
Primary and secondary context

- Primary context
  - Location
  - Identity
  - Time
  - Type
- Secondary context: e.g.,
  - Email Addresses
  - Activity
  - An Entity’s state
- Primary context can be used as an index into context data

=> consequences for data management

Dey, Abowd; Rothermel, Becker et al.; Mitschang, Nicklas et al.
Characteristics of Context Data

- **low** update rate
- **high** update rate

- *primarily* usage for selection
- *rarely* usage for selection

- **primary context:**
  - location, id, time, type

- **secondary context:**
  - name, web site, temperature, activity, ...
Data Management

- **Specialized context servers:**
  - For static data: spatial databases (Spatial Model Server)
  - For dynamic data: sensor platforms, main memory services, ... (ContextCube)
  - For mobile data: Location Service (copes with high update rates)

- **Integrate legacy data / servers**
  - Wrappers for existing context management (AHSS)
  - Discovery, e.g. web robots (DCbot)
Specialized Context Servers

<table>
<thead>
<tr>
<th>Service</th>
<th>Num of objects</th>
<th>Access paths</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location Service</td>
<td>10,000</td>
<td>ID, pos</td>
</tr>
<tr>
<td>Spatial Model Server</td>
<td>100,000</td>
<td>ID, pos</td>
</tr>
<tr>
<td>Indoor Spatial Server</td>
<td>1,000</td>
<td>none</td>
</tr>
<tr>
<td>ContextCube</td>
<td>1</td>
<td>none</td>
</tr>
<tr>
<td>AwareHome Spatial Server</td>
<td>???</td>
<td>ID, pos</td>
</tr>
<tr>
<td>SensorContextServer</td>
<td>100</td>
<td>ID, pos</td>
</tr>
</tbody>
</table>

Public and private context

- Public context
  - is provided to the public, e.g., web sites, map data, weather information, ...
  - can be easily obtained by everyone
  - overlaps between applications in similar situations

- Private context
  - sensitive data, often user-related: exact position, preferences, behaviour patterns, ...
  - needs special protection, requires security, privacy and safety methods
  - if possible, keep locally on the device

- Differentiation depends on user, society, and legal situation (internationalization?)
Data, (lower) context and situation (higher context)

(attempt of a) differentiation:
- data: every bit of information the application gets, without interpretation. Typically: what the network sees
- context: information related to the situation of user and/or real world, needed to determine the information (lower context)
- situation: information that can be directly used by context-aware applications for adaptation (higher context)
## Example

<table>
<thead>
<tr>
<th>situation</th>
<th>user is accepting shopping recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>related context</td>
<td>near by: Neues Schloss, Königsbau, Art Gallery, shopping sites</td>
</tr>
<tr>
<td></td>
<td>opening hours</td>
</tr>
<tr>
<td>context</td>
<td>inside Stuttgart Schlossplatz</td>
</tr>
<tr>
<td></td>
<td>after work</td>
</tr>
<tr>
<td>data type</td>
<td>position of user</td>
</tr>
<tr>
<td></td>
<td>current time</td>
</tr>
<tr>
<td>data</td>
<td>WGS84 coordinates</td>
</tr>
<tr>
<td></td>
<td>GML Timestamp</td>
</tr>
<tr>
<td></td>
<td>9,175; 48,7826</td>
</tr>
<tr>
<td></td>
<td>2005-10-18T20:47:00.000</td>
</tr>
</tbody>
</table>

**Reasoning**

- **User Context**:
  - position of user: inside Stuttgart Schlossplatz
  - current time: after work

- **Related Context**:
  - near by: Neues Schloss, Königsbau, Art Gallery, shopping sites

- **Data**:
  - type: WGS84 coordinates
  - timestamp: 2005-10-18T20:47:00.000
Overview

- Why?
  - Pervasive computing and context-aware applications
- What is context?
  - Definitions
  - Types
- Context modeling
  - Modeling approaches
  - Location models
- Context reasoning
- Application examples
Context modeling

Modeling approaches
Location models
Context Modeling Approaches

- Key-Value Models
- Markup Scheme Models
- Graphical Models
- Object Oriented Models
- Logic Based Models
- Ontology Based Models
- Relational Models
- Hybrid Models

Key-Value-Pairs Models

- Early approach, e.g., Schilit
- Most simple category of models
- Not very efficient for more sophisticated structuring purposes
- Only exact matching

Markup Scheme Models

- "Shallow Context"
- Scheme implements model
- (Sometimes) hierarchical structured key-value-pairs
- Typical representatives: profiles
- e.g., CC/PP (based on RDF)
Markup Scheme models: CC/PP example

<ccpp:component>
  <rdf:Description
    rdf:about="http://www.example.com/profile#TerminalHardware">
    <rdf:type
      rdf:resource="http://www.example.com/schema#HardwarePlatform" />
    <ex:displayWidth>320</ex:displayWidth>
    <ex:displayHeight>200</ex:displayHeight>
  </rdf:Description>
</ccpp:component>
Graphical Models

- Particularly useful for structuring, but usually not used on instance level
- e.g., Context Modeling Language (extension of ORM) by Indulska et al.

Object Oriented Models

- **Intention:** encapsulation, reusability and heterogenous instance sets

- **Examples:**
  - Active Object Model
  - Nexus Augmented World Model (designed for context integration and schema evolution)


Schema Evolution: Nexus Augmented World Schema

- **Standard Class Schema**
  - Fundamental object classes
  - Needed by most applications
  - Ensures interoperability
  - Unique identifier for every object instance

- **Extended Class Schema**
  - For future applications
  - Objects inherit from Standard Class Schema
Ontology Based Models

- Ontology used as explicit specification of a shared conceptualization → strong in the field of normalization and formality
- Context is modeled as concept and facts
- Additional rules provide (often inefficient) reasoning
- Examples:
  - SOUPA, CoOL, CONON, CoBrA, ...


Relational Models

- Use "plain" database technology for mobile applications (mobile DB)
- Information is filtered according to context of the application
  → definition by view concepts
- based on a closed-world assumption

Hybrid approaches

- nowaday considered most beneficial by the community
- combines different modeling techniques for different purposes
- often along different levels of interpretation/semantics
Hybrid approaches

- Indulska et al.: CML distinguishes between facts (DB) and rules (inference engine)
- Bettini et al.: CC/PP and OWL
- Roussaki et al.: Ontology and geographic model (DB)
- Nexus: data, context, and situation
Location Models
Location Models

- **Geometric locations**: 2D and 3D geometric figures
  - "Arbitrary" areas but high modeling effort
- **Symbolic locations**: building/floor/room numbers, etc.
  - Intuitive to use, small modeling effort
- **Hybrid**: Combinations of symbolic and geometric locations, e.g.
  - GPS (geometric addressing) outdoor
  - ActiveBadge (symbolic addressing) indoor
  - ActiveBat (geometric addressing) in some rooms
- **Mobile target areas**, e.g. trains, ships, etc.
Geographic location models: Different coordinate system types

- Geographic Coordinate System
  - Spherical coordinates: latitude, longitude
  - Datum: specifies placement of ellipsoid

- Projected Coordinate System
  - Cartesian coordinates: northing, easting
  - Has dedicated geographic CS as starting point for projection

- Local Coordinate System
  - Cartesian coordinates: X, Y
  - Defined by transformation rule into another CS
Geographic location models: Transformation problem

Set $L$ of symbolic locations

Partial order $\leq$ defined by the spatial contains relationship, i.e. for two locations $l_1, l_2 \in L$ it holds $l_1 \leq l_2$, iff $l_2$ contains $l_1$.

Hierarchy is a lattice (more general than a location tree)
Symbolic location models: Example 2 – graph-based

- Locations modeled as nodes
- Edges represent connections
- Edges or Nodes weighted to indicate size/distance
(Brief) Discussion of Symbolic Location Models

- Lattice/tree based
  - Reflects spatial inclusion
  - Poorly reflects distances

- Graph based
  - Reflects distances
  - Does not reflect spatial inclusions

- Combinations required when nearest neighbor and range queries have to be processed
Hybrid Location Model

- Symbolic and geometric coordinates coexist
- Annotate locations in symbolic location model with geometric extensions
  - Complete annotation of locations in symbolic model (sub-space integration; Jian, Steenkiste)
  - Incomplete annotation of locations in symbolic model (inherited and approximated size; Dürr, Rothermel)
Location Models: Summary

- A variety of approaches possible and suitable
  - Depending on the application requirements
- Modeling effort vs. accuracy typical trade-offs
- Symbolic and geometric coordinates coexist
  - geo-coding offers restricted conversion
  - hybrid location models needed
Overview

- Why?
  - Pervasive computing and context-aware applications

- What is context?
  - Definitions
  - Types

- Context modeling
  - Modeling approaches
  - Location models

- Context reasoning

- Application examples
Context reasoning
Context reasoning

- Derivation of higher level context (e.g., sensor values) from lower level context

- Approaches:
  - by application ("hard-coded", or proprietary model)
  - probabilistic: e.g., Bayesian networks
  - rule-based: e.g., OWL DL
Rule-based Inference

- Logic defines conditions on which a concluding expression or fact may be derived from a set of other expressions or facts (reasoning)
  - context is defined as facts, expressions and rules
- High degree of formality
  - often based on ontologies
Rule-based inference – example

<table>
<thead>
<tr>
<th>DL Reasoning Rules</th>
<th>Explicit Context</th>
<th>Implicit Context</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>&lt;Room rdf:ID=&quot;Bedroom&quot;&gt; &lt;contains rdf:resource=&quot;#Wang&quot;/&gt; &lt;/Room&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Situation</th>
<th>Reasoning Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleeping</td>
<td>(?u locatedIn Bedroom) ^ (Bedroom lightLevel LOW)</td>
</tr>
<tr>
<td></td>
<td>^ (Bedroom drapeStatus CLOSED)</td>
</tr>
<tr>
<td></td>
<td>?u situation SLEEPING</td>
</tr>
<tr>
<td>Showering</td>
<td>(?u locatedIn Bathroom)</td>
</tr>
<tr>
<td></td>
<td>^ (WaterHeater locatedIn Bathroom)</td>
</tr>
<tr>
<td></td>
<td>^ (Bathroom doorStatus CLOSED)</td>
</tr>
<tr>
<td></td>
<td>^ (WaterHeater status ON)</td>
</tr>
<tr>
<td></td>
<td>?u situation SHOWERING</td>
</tr>
<tr>
<td>Cooking</td>
<td>(?u locatedIn Kitchen) ^ (ElectricOven locatedIn Kitchen)</td>
</tr>
<tr>
<td></td>
<td>^ (ElectricOven status ON)</td>
</tr>
<tr>
<td></td>
<td>?u situation COOKING</td>
</tr>
<tr>
<td>Watching-TV</td>
<td>(?u locatedIn LivingRoom)</td>
</tr>
<tr>
<td></td>
<td>^ (TVSet locatedIn LivingRoom)</td>
</tr>
<tr>
<td></td>
<td>^ (TVSet status ON)</td>
</tr>
<tr>
<td></td>
<td>?u situation WATCHING-TV</td>
</tr>
</tbody>
</table>

Rule-based inference – limitations

- Problem with high number of facts, classes, and/or rules
  - reduction approaches (hybrid modeling)


Bayesian networks

- directed acyclic graph
  - nodes: random variables representing various events
  - arcs: causal relationships
- training required
- useful for
  - performing probabilistic sensor fusion
  - higher-level context derivation
Bayesian network – example 1

http://de.wikipedia.org/wiki/Bayessches_Netz
Bayesian network – example 2

Application examples
StuPro (study project) SmartRoom

- Goal: use Nexus platform for smart room application, show benefits and drawbacks
- Project:
  - 11 graduate students
  - 1 year (∼4400h + seminar + lectures)
- Phase 1: Basic infrastructure
- Phase 2: Smart room applications
  - WindowWatcher
  - SmartMemory
  - ConferenceGuard
WindowWatcher

- Nexus lab in ground floor → windows should be closed when room is empty
- Light barrier observes windows
- Camera and motion detector observe presence of people
- When last person leaves the room, a warning is displayed on the door (steerable projector) and put on nearest loudspeaker
  - Person can defer for 5 minutes (e.g., toilet break)
  - After 5 minutes of no activity in the room: E-mail to room supervisor

Close the window!
SmartMemory

Pervasive game with flexible game environment:

- Game objects in context model: memory cards
- Steerable projector displays cards on the floor near the player
- Player uses high precision positioning system (Intersense) for game interaction (e.g., card selection)
- *Sound feedback by nearest loudspeaker*
ConferenceGuard

- Inside Nexus lab, there are a meeting area and student workplaces → conflicts during meetings (i.e., noise disturbance)
- Sensors for meeting detection: motion detector, camera, light barrier, microphone; meeting room schedule
- Sensor for student workplace: microphones (loudness)
- Actuator for student workplace: AmbientOrb (color changing lamp)
  - green: no meeting
  - yellow: meeting (work quietly)
  - blinking red: meeting and you are too loud
Context models

Applications

Update(id, value)

query (filter)

event (condition)

Context model

Sensors (Fusion)

Physical world

Application State

Update(id, value)
StuPro smartroom: infrastructure

Nexus context model

Stream information

Query (filter)

Register event

Update (commands)

Event notifications

Adaptors: Actuator-Sensor-Client (ASC)

Data objects representing sensor values and service descriptions (commands)

Smart room applications

WW  CG  SM

Sensors

Actuators

Steerable projector
AmbientOrb
Steerable camera
Light barrier
SmartIT

SmartITAmbientOrb

CG

WW

SM
Thank you!
Acknowledgements

I borrowed slides from:
- Christian Becker
- Frank Dürr
- Matthias Großmann
- Thomas Schwarz
CoMoRea '08

5th IEEE Workshop on Context Modeling and Reasoning (CoMoRea'08) @PerCom

http://www.informatik.uni-stuttgart.de/COMOREA

Hong Kong, 17 or 21 March 2008

Co-Chairs: Karen Henricksen, Daniela Nicklas


