Data, Context, Situation

On the Usefulness of Semantic Layers for Designing Context-aware Systems

Prof. Dr. Daniela Nicklas

Department für Informatik
Juniorprofessur Datenbank- und Internettechnologien
Carl von Ossietzky Universität Oldenburg
Context
What is Context?

- Session “Context” @ ICOSOFT:

Context Modeling
Context Transition Determination
Software Development

Introduction
Motivation

The „Deyfinition“

"Any information that can be used to describe the situation of an entity” (Dey and Abowd)

Context Model

- Representation of situations
- More than just location
- Built on top of framework
- Connected to perspective
- Context model of Keen

The context of a software developer

Knowledge Provisioning


Gregor Grambow¹, Roy Oberhauser¹ and Manfred Reichert²

¹Computer Science Dept., Aalen University, Aalen, Germany
²Institute for Databases and Information Systems, Ulm University, Ulm, Germany
{gregor.grambow, roy.oberhauser}@htw-aalen.de, manfred.reichert@uni-ulm.de
What is Context?

- Session „Context“ @ ICSoft:

Enhancing Alternative and Augmentative Communication with Context Awareness Components

Lindsey Buehler and Sherrill David

The context of an user

Context (location) is used to determine most likely words (by categories)

→ Adaptation of the user interface (by highlighting)

Figure 1: The graphical user interface

Figure 2: The AAC Architecture.
What is Context?

from Wikipedia:

- **Context (computing):**
  - task context - “the minimal set of data used by this task that must be saved to allow a task interruption at a given date, and a continuation of this task at the point it has been interrupted and at an arbitrary future date”

- **Context awareness**
  - “deal with linking changes in the environment with computer systems, which are otherwise static”
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.

Why should I care?

- **Context in requirements engineering:**
  - during development time …
  - find out where your software is going to be used
  - find out whether it interacts with its context (does it have to be context-aware?)
  - derive additional requirements

- **Developing context-aware systems:**
  - during run time …
  - software finds out what the current context is
  - software adapts to context changes:
    - presentation
    - selection (of information/services)
    - action (do other things)
    - tagging (annotate information with context)

→ needs context management

…and this is a real challenge!
Overview

- Introduction
- Context Management
  - Characteristics of context
  - Anatomy of context-aware applications
  - Context management approaches
- Conclusion
Typical Context-Aware Systems

mobile computing

"smart" environments (AKA Internet of Things)

I can play your music

I tell you when I need water

If it’s too hot, I’ll help

I can give you information

I can adjust my settings

I know where I am and if I’m hot

Musikhochschule Stuttgart

dnicklas@acm.org
More examples … (of my current interest)

Application Areas

- Traffic monitoring
- Safe Off-shore Operations
- Vessel Traffic Management
- Smart Factory
- Logistics
- Driver assistant systems
- … and many more!

- Crew
- Congestion
- Snow
- Pedestrians
- Goods
- Trucks
- Parcels
- Obstacles
- Lanes
- Cars
- Roads
- Wind
- Cargo
- Processes
- Sea
- Ships
- Ground
- Products
- Workers
- Machines
Overview

- Introduction
- Context Management
  - Characteristics of context
  - Anatomy of context-aware applications
  - Context management approaches
- Conclusion
Context Management
How to get the context?

**situation**

user is accepting shopping recommendations

**related context**

near by: San Pietro, Musei Vaticani, Borgo A. Angelo, shopping sites

**context**

location of user

inside Piazza San Pietro

**data type**

WGS84 coordinates

**data**

9,175; 48,7826

**opening hours**

opening hours

user's timetable

afternoon

current time

**user's speed**

GML Timestamp

2005-10-18T20:47:00.000

**sensors**

GPS-Sensor

click on map

... system clock

gps clock

dnicklas@acm.org
Context types

- **Geographic context: map data**
  - Streets, buildings, landmarks, points of interest, ...
  - Data source: geographic information systems (GIS)
  - Stationary objects, rarely changing

- **Dynamic context: movement and change**
  - Persons, vehicles, traffic situation, weather, ...
  - Data source: sensors

- **Information context: digital world, "cyberspace"**
  - Web sites, documents, game objects, ...

- **Technical context: infrastructure**
  - Access networks, topology, services
    (printer, projector, ...)

- **User context**
  - Activities, plans, preferences
  - Highly sensitive!
Context models in smart environments

Technical context:
- service discovery
- object positioning
- object identification
- integration of new hardware
- remote control of entities
- context events

"smart" environments (AKA Internet of Things)

I can play your music
I tell you when I need water

I know where I am and if I’m hot
I can give you information
I can adjust my settings
If it’s too hot, I’ll help

Virtual Database of Things

 dnicklas@acm.org
Context characteristics

- Context information is heterogeneous
  - discrete / continuous
  - multi-dimensional
  - multi-media
  - spatial
  - sensed / static / profiled / derived

- Context information differ in
  - type
  - information quality
  - temporal characteristics
Context characteristics

- **Sensed Context: Low persistence**
  - may be inaccurate, unknown, or stale
  - source of errors
  - sensor failures
  - network disconnections
  - delays (in communication or processing)

- **Context information may differ in quality; may be**
  - unknown
  - ambiguous
  - imprecise
  - erroneous
Context characteristics

- Context information needs temporal meta data
  - past state (history)
  - current state
  - future state
  - changes in state over time

- Context has various dependencies
  - physical laws
  - ownership
    - who owns devices
    - which computers have a license to run
  - particular software
  - derivation rules for derived context
Local context model or shared

- **Local context model:**
  - context management for just one application
  - design of components up to the software developer, but it’s a good idea to separate concerns:
    - from sensors to data
    - from data to context
    - from context to situation

- **Shared context model:**
  - common context management for several applications
  - design of components depends on „sharability“
  - can save tremendous amount of development time if several applications work in similar/overlapping (by space / by content) contexts
Abstraction: Shared Context Model

Applications

(documents) context model

Continuous event (condition)

Update(id, value)

Query (filter)

Sensors (Fusion)

Physical world

Application state

dnicklas@acm.org
## Anatomy of a context-aware application

and „sharability“ of context information

<table>
<thead>
<tr>
<th>Application-specific</th>
<th>Application</th>
<th>what the user / other systems / the world see</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standards for aggregation methods</td>
<td>Adaptation</td>
<td>how the application's behaviour changes</td>
</tr>
<tr>
<td></td>
<td>Situation</td>
<td>in what cases does adaption happen</td>
</tr>
<tr>
<td>Easy to share</td>
<td>Context</td>
<td>any information that can be used to characterize the situation of an entity [&quot;Definition&quot;]</td>
</tr>
<tr>
<td>Enable domain-specific standards</td>
<td>Data</td>
<td>data (e.g., from sensors, content) used by the system to determine context information</td>
</tr>
</tbody>
</table>

[Image of table and diagram]

dnicklas@acm.org
Example: Smart Factory

- Context-awareness in production environment
- Detailed model of Smart Factory:
  - Locations of objects:
    - machines, installations, tools, work pieces, fixtures, etc.
  - State of objects:
    - e.g., mechanical wear of tools

Example Smart Factory: Maintenance Process

Objects in context model

machines
Smart Factory

?- Maintenance task

VTC with visibility area

tool

mobile agents

M1: exchange tool

M2: exchange tool

M3: handle emergency

ContextCast
Smart Factory – Machine Maintenance Processes

context event registration

context query

context decision

context event notification

start process on context event

answer context query

manage context event

manage context event

<context event>

event definition: (abrasion>90%) or (usage time>23)

context query for tool ID

generate maintenance task for worker

wait for notification

machine repaired

task handling time > 2h

order new spare part and generate problem report

Generate emergency task

Do routine work

Perform maintenance task

Do routine work
# Anatomy of the Smart Factory Applications

<table>
<thead>
<tr>
<th><strong>Tasks</strong> (offered/assigned)</th>
<th><strong>Application</strong></th>
<th>what the user / other systems / the world see</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Modeled in BPMN</strong> (with context extensions)</td>
<td><strong>Adaptation</strong></td>
<td>how the application's behaviour changes</td>
</tr>
<tr>
<td><strong>Context events</strong> (task notifications, task state)</td>
<td><strong>Situation</strong></td>
<td>in what cases does adaption happen</td>
</tr>
<tr>
<td><strong>Live digital factory</strong></td>
<td><strong>Context</strong></td>
<td>any information that can be used to characterize the situation of an entity [&quot;Deyfinition&quot;]</td>
</tr>
<tr>
<td><strong>Shop floor layout, locations, ...</strong></td>
<td><strong>Data</strong></td>
<td>data (e.g., from sensors, content) used by the system to determine context information</td>
</tr>
</tbody>
</table>

dnicklas@acm.org
Overview

- Introduction
- **Context Management**
  - Characteristics of context
  - Anatomy of context-aware applications
  - Context management approaches
- Conclusion
Context Modeling Approaches

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

... result in different management architectures

A quick round of typical Context Management Architectures

• CML (University of Queensland)
• Mileo (ISST Dortmund)
• CARE (University of Milan)
• Nexus (University of Stuttgart)
Context Management: the CML approach

• Jadwiga Indulska et al., University of Queensland / NICTA
Model-Based Context-Aware Applications

- **Context-Aware Application**
- **Context Management System**
  - **Abstract Context Models**
  - **Sensed Data**
- **Context Sources (Sensors)**
  - **Situation**
  - **Context**
  - **Data**

Layers:
- **Application Layer**
- **Middleware Layer**
- **Sensor Layer**
Fact Modelling: Example application model

Person (name)

has type

Device (ID)

has type

* person near(p, d₁) iff
using(p, d₁) or (using(p, d₂) and
(device near(d₂, d₁) or device near(d₁, d₂)))
current location(p, l) iff
person near(p, d) and contains(l, d)

Legend

○ Profiled information
* Derived information
<s> Sensed information
s Static information

Uniqueness/key constraint
----- Alternative uniqueness constraint
a Alternative fact type

Quality annotation
System architecture for CML based approach

Application 1  Application 2

Context Programming Toolkit

Context Database  Situation Database

Context Manager  Preference Manager

Context Processing

Sensors

Programming models:

• Situation Based Triggering
• Situation Based Branching  
  • Context dependent choice from set of alternatives  
  • Based on context situation and preferences  
  • API methods include rank, selectBest  
• Context handling logic moved out of the application  
• Support for evolvable applications
Context Management Examples: the MILEO – Context Server

Manfred Wojciechowski, Markus Wiedeler
Fraunhofer Institute for Software and Systems Engineering
Approach
Context Meta-Model
**Application Example**
Project 'Daily Care Journal'

**AAL Service**

**Formular „Eye Problems“**

**Event-Subscription**

**Context Server**

**Context**

**Action:**
Application specific logic

**Condition:**
Mapping between event and function

**Event:**
Context specific logic / context model

```java
SpaceQueryObject s = newSpaceQueryObject ("WatchingTV");
subscription = contextServiceInstance.SubscribeContextSpaceEvent
(“DCJ“, “AEDLCommunication“, s);
```

**API**
Context Management Examples: the CARE platform

A. Agostini, C. Bettini, D. Riboni
Data, Knowledge, and Web Engineering Laboratory
University of Milan, Italy
Modeling Non-Shallow Profile Data

• We represent “complex” data by means of OWL-DL ontologies, to allow:
  – Knowledge sharing among involved entities
  – Consistency checking of contextual data instances
  – Reasoning to derive additional contextual data (e.g., specific activity of the user)

• Profile data are mapped into CC/PP attributes
Architecture overview
Context Management Example: the Nexus Platform

• Kurt Rothermel et al, University of Stuttgart
Context Management by Nexus

A global federation over …

- 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 mangement (AHSS)
  - Discovery, e.g. web robots (DCbot)

- For updates: specialized event services
  - registration of spatial events
  - distributed observation

- For streamed, distributed context information: NexusDS
  - data stream management system with flexible operators
Nexus Architecture: Global Federation of Context Models

AWQL
AWML

Application

Situation Recognition

Context

Data Federation

Situation

Context Server

Sensors

Geo DB

Server

Server

Data

dnicklas@acm.org
Overview

- Introduction
- Context Management
  - Characteristics of context
  - Anatomy of context-aware applications
  - Context management approaches
- Conclusion

Data

Context

Situation
Conclusion

- To develop context-aware application, context modeling and management has to be solved first

- There has been many research in the pervasive computing community
  - however, less approaches for comprehensive software engineering approaches (see Indulska or Wojciechowski)

- The layers of data, context, and situation can help to cope with the complexity
  - develop applications that deal with (defined, named) *situations*
  - let the context management do the derivation of situations out of *context*
  - decouple from the derivation out of raw *data*
CoMoRea 2013: 10th IEEE Workshop on Context Modeling and Reasoning
San Diego, California, March 18-22, 2013
Thank you!

and also thanks to …

Bruno Antunes, Jadwiga Indulska, Daniele Riboni, Matthias Wieland, Manfred Wojciechowski for ideas and slides