Context Management for Proactive Adaptation in Pervasive Computing

Dipl.-Wirtsch.-Inf. Sebastian VanSyckel

Lehrstuhl Wirtschaftsinformatik II

Prof. Dr. Christian Becker

Universität Mannheim

Contents

Motivation

Goal: "[…] evolve reactive adaptation into proactive adaptation"

- Two facets:
  1. Shift application adaptation ahead of time
  2. Allow context adaptation to prevent suboptimal context

➔ In a nutshell, a context model for proactive adaptation has to provide the following functionality:

1. **Access** to context information and services **via a suitable abstraction**
2. **Dynamic integration** of context services, e.g. sensors and actuators
3. **Support for context prediction algorithms**

Definitions

Context is any information that can be used to characterize the situation of an entity. […][1]

A **context-aware** system provides context information to its applications, which thereby adapt their behavior and/ or user interaction according to the changes.[1]

Proactive adaptation describes a subject's adjustment to anticipated environmental conditions, in particular conditions categorized as the subjects context, or the precautionary influence thereof.

A **formal context model** is the abstract representation and provision of context, whereas a **context model implementation** is the realization of a formal context model as a piece of software.

[1] Following definitions by Shilit *et al.* 1994, Dey *et al.* 1999, Becker 2004

Requirements:
Formal Context Model 1/2

i. Abstraction of Context
   - Common label for each context entity in order to express and address context

ii. Location Relation
   - Distinct between local & distant/ relevant & irrelevant context
   - Address context at a specific location

iii. Temporal Relation
   - Distinct between past, present and future context
   - Address context at a specific point in time

Requirements:
Formal Context Model 2/2

iv. Information Quality Index
- Express probability of predictions, accuracy of sensors, etc.
- In general: reliability of a context information source

v. Defined Context Queries
- Standardized “language” for interaction with formal context model

Requirements:
Context Model Implementation 1/2

a. Persistent Context Storage
   - Some context prediction algorithms are based on context history/ patterns

b. Context Acquisition
   - In order to use context it must first be acquired

c. Context Prediction
   - 1st characteristic of proactive adaptation: application adaptation based on context anticipation

Requirements:
Context Model Implementation 2/2

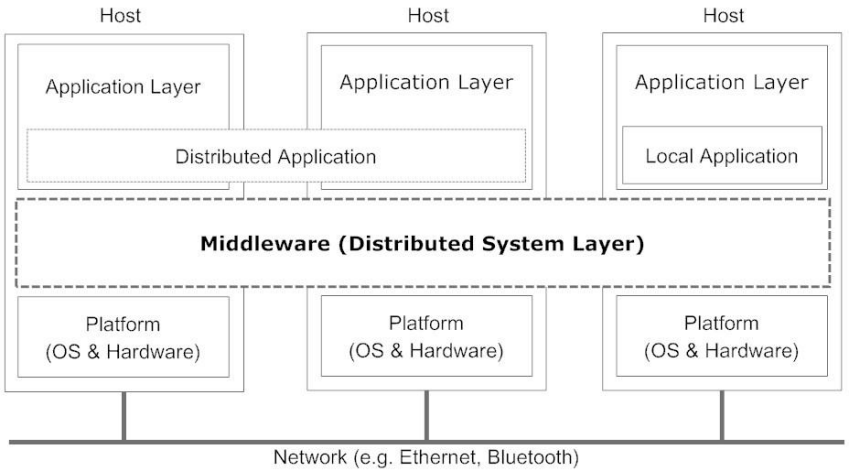
d. Context Adaptation
   - 2nd characteristic of proactive adaptation: precautionary context adaptation

e. Dynamic Environment Support
   - Devices in pervasive environments are highly dynamic, e.g. mobile phones

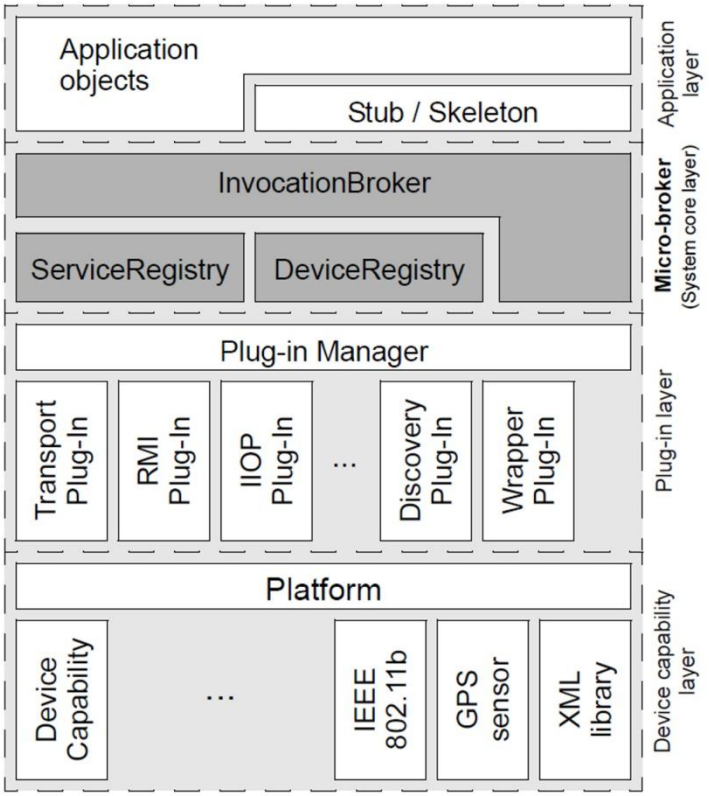
f. Uniform Access
   - Access any context information equally, regardless of type, source, location, etc.
   - ➔ A defined query issued to a central instance, i.e. the context model implementation

Related Work:
Middleware & BASE

Middleware



BASE[1]



Why Middleware/ BASE?

- Platform independence
- Abstraction via stubs and skeletons
- Lean
- **Dynamic environment support**

[1]Figure source: Christian Becker, Gregor Schiele, Holger Gubbels, and Kurt Rothermel. *Base - a micro-broker-based middleware for pervasive computing*. In Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on, pages 443-451, 2003.

Related Work:
Context Models & CoBrA

Context-aware Operational Life-cycle[1]:

1. Context Determination:
   - Sense context and convert result into processible data
2. User Context Acquisition:
   - Acquire *goal user context*, i.e. individual related information & policies
3. Context Processing:
   - Aggregate equal context
   - Compose package of relevant context
   - Application adaptation
4. Context Management:
   - Service discovery
   - Persistent storage
   - Knowledge sharing
   - Access control

[1] Stefan Poslad. *Ubiquitous Computing: Smart Devices, Environments and Interactions.* John Wiley and Sons Ltd., 2009, p. 222-225

Related Work:
Context Models & CoBrA

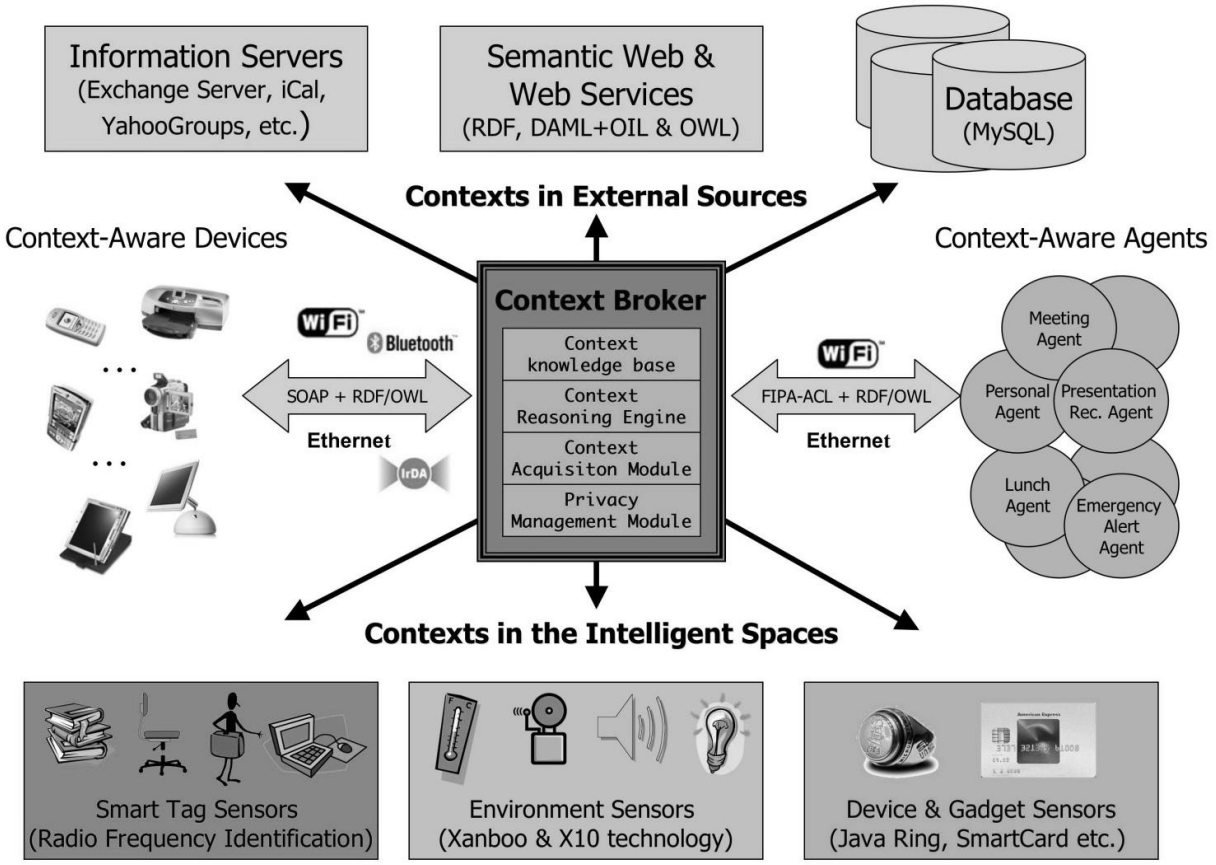
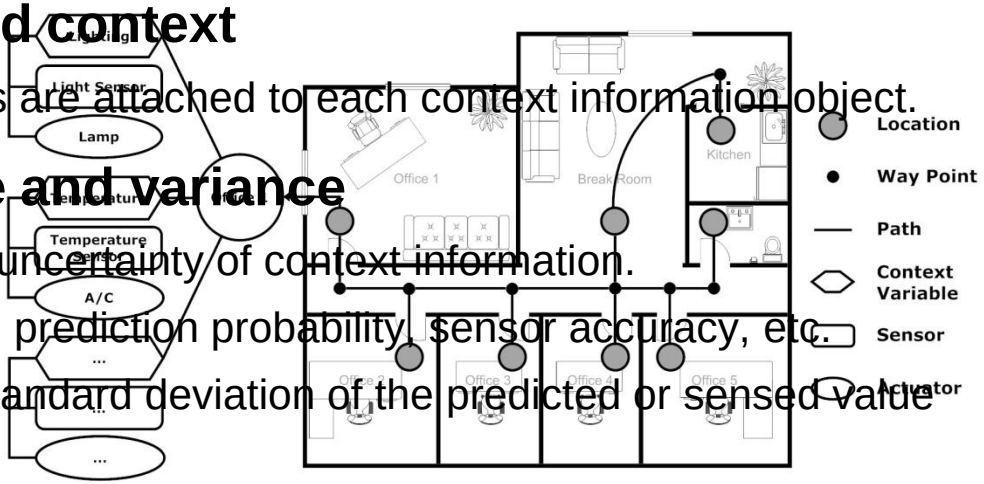


Figure source: Harry Lik Chen. *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*. PhD thesis, Department of Computer Science and Electrical Engineering, University of Maryland, 2004.

Context Management for Proactive Adaptation
Formal Context Model (Req. i.-iv.)

1. **Abstraction via context variables**
   - Each context entity has an assigned label for expressing and addressing purposes.
   - A label, a value and the value's data type form a *context information*.
2. **Location-based context**
   - Context is organized by location, i.e. each context information object is associated with a single location of the underlying location model.
   - For illustration purposes: A graph-based location model is "extended" by context.
3. **Time-related context**
   - Timestamps are attached to each context information object.
4. **Confidence and variance**
   - Denote the uncertainty of context information.
   - Confidence: prediction probability, sensor accuracy, etc.
   - Variance: standard deviation of the predicted or sensed value



Context Management for Proactive Adaptation
Minimal Set of Context Queries (Req. v.) 1/2

1. **Context Location Query**

   $Q^L = \{V, S, t_1, t_2\}$, $t_1 \leq t_2$ is a 4-tuple, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of context variables, $S = \{s_1, s_2, \ldots, s_n\}$ a set of context variable states, and t_1, t_2 denote a timeframe.

   - Returns a set of locations, incl. their respective confidence and variance values
   - Example: “Where is it bright and quiet between 9 am and 5 pm?”

2. **Context State Query**

   $Q^S = \{V, l, t_1, t_2\}$, $t_1 \leq t_2$ is a 4-tuple, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of context variables, l a location, and t_1, t_2 denote a timeframe.

   - Returns a set of configurations, incl. their respective confidence and variance values
   - Example: “How high is the noise level currently in the library?”

3. **Context Time Query**

   $Q^T = \{V, S, l\}$ is a 3-tuple, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of context variables, $S = \{s_1, s_2, \ldots, s_n\}$ a set of context variable states, and l a location.

   - Returns a set of timeframes, incl. their respective confidence and variance values
   - Example: “When will it be dark outside?”

Context Management for Proactive Adaptation
Minimal Set of Context Queries (Req. v.) 2/2

4. **Context Adaptation Capability Query**

$Q^{AC} = \{V, l, t_1, t_2\}$, $t_1 \leq t_2$ is a 4-tuple, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of context variables, l a location, and t_1, t_2 denote a timeframe.

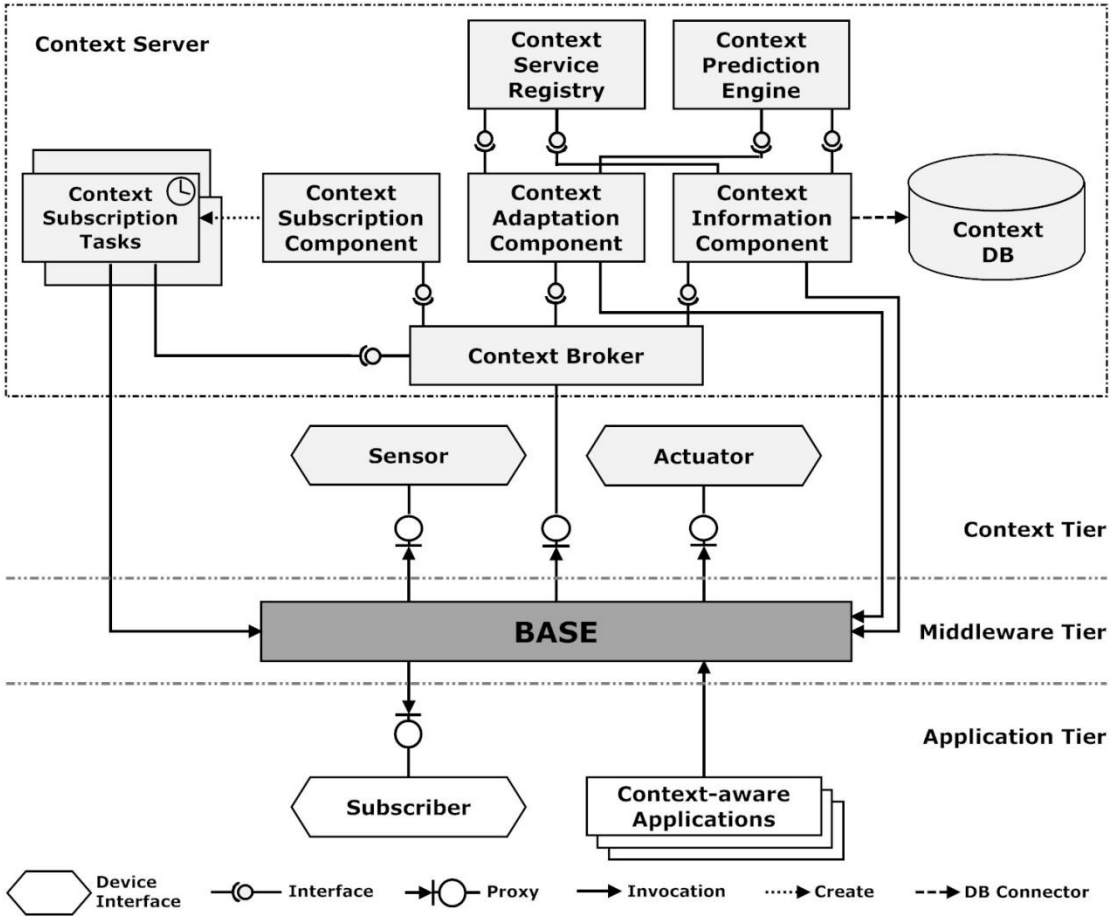
- Returns a set of adaptation capabilities, incl. their respective confidence and variance values
- Example: “Can the temperature be influenced in room 723?”

4. **Context Adaptation Instruction Query**

$Q^{AI} = \{V, S, l\}$ is a 3-tuple, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of context variables, $S = \{s_1, s_2, \ldots, s_n\}$ a set of context variable states, and l a location.

- Instructs the specified context adaptation
- No return value necessary, as adaptation can be checked via context state query
- Example: “Change the temperature in my office to 20° C!”

Context Management for Proactive Adaptation
Architecture



Implementation & Evaluation

- Deviating details of the (prototype) implementation:
  - Context broker API **extends set of context queries** by possibility to …
    - … query any context service instead of only actuators.
    - … subscribe to all queries, except adaptation instructions.
    - … cancel subscriptions.
    - … report context information.
  - Context prediction engine is basically empty container:
    - Defined interface
    - But: Only a set of dummy prediction algorithms for testing purposes
  - Although assumed externally available, prototype has very basic location model for testing purposes.
- **The prototype was evaluated using two test cases featuring context prediction, acquisition, subscription and adaptation.**

Conclusion & Outlook 1/2

- Recap: The two facets of proactive adaptation
  1. Shift application adaptation ahead of time
  2. Allow context adaptation to prevent suboptimal context
- Our approach offers context prediction and adaptation in a uniform fashion
  - ➔ **Suitable abstraction** via context variables & defined context queries
    - Location-based, time-related context
    - Context information quality expressed through confidence & variance
  - ➔ Centralized context broker provides **access to** …
    - … prediction engine (future context information, sensors and actuators)
    - … information component (past & present context information, present sensors)
    - … adaptation component (present actuators)
    - … subscription component (subscribe to all above; not required feature)

Conclusion & Outlook 2/2

Future work:

- Integration of …
  - … **context prediction algorithms**
  - … context reasoning to examine/ evaluate the **consistency of predictions**
- Development of …
  - … an **application model** for proactive adaptation
    - Which application configurations are possible for future context?
    - What is the cost/ benefit ratio for reconfigurations?
  - … **adaptation strategies** for proactive adaptation
    - Which series of context- and application adaptations are beneficial?
    - How can multiple proactive applications collaborate on future context- and application adaptations?

THANK YOU!
QUESTIONS?