

# Frameworks e Middleware para provisão de contexto

*Markus Endler*  
*LAC/PUC-Rio*

Adaptado da apresentação de Roy Campbell

## Agenda

- Challenges
- Classification of Architectures
- Abstract Layer Architecture
- Examples of Frameworks and Middleware for context-awareness
- Comparison

## Major Challenges

- How handle **heterogeneous and distributed sensors** in a uniform way?
- But what is the **relevant context information** in various situations?
- Mobility results in **continuous updates** of context information. How can we efficiently manage this?
- How can we **share** context?
- How do we reach a **common understanding** of implications and semantics of (shared) context information?
- How do we handle **uncertainty** of context information?
- How do we ensure **privacy** control and management of context information?
- How to handle resource restrictions (e.g. battery life)?
- How to deal with unreliable connectivity among devices/sensors?
- ...

3

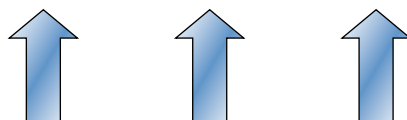
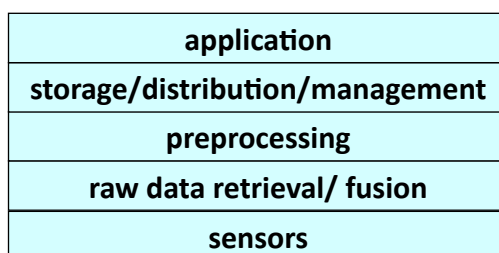
## Classification of Architectures (I)

- Direct sensor access
  - Tightly coupled
  - No extensibility
- Middleware
  - Hiding low-level sensing details
  - Extensible
- Context server/broker
  - Permit multiple clients access to remote context data sources
  - Relieve clients of resource intensive operations
  - Context server can perform global context reasoning
  - Has to consider appropriate protocols, network performance, quality of service parameters

## Classification of Architectures (II)

- Widgets (process-centric view)
  - Encapsulation
  - Exchangeable
  - Controlled by a widget manager
  - The tight coupled widget approach increases efficiency but is not robust to component failures
- Networked services (service-oriented model)
  - view all elements as software services
  - Resembles context server architecture
  - Not as efficient as a widget architecture due to complex network based components but provides robustness
- Blackboard model (data-centric view)
  - Processes post messages to a shared media, blackboard
  - Simplicity of adding new context sources
  - Easy configuration
  - Implemented as a centralized server
  - Lacks in communication efficiency (2 hops per communication are needed)

## Abstract Layer Architecture

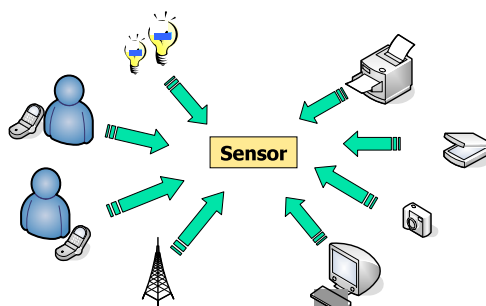


## Abstract Layer Architecture

- Sensors
  - Physical sensors
    - sensor, camera, microphone, accelerometer, GPS, thermometer, biosensors
  - Virtual sensors
    - From software: browsing an electronic calendar, a travel booking system, emails, mouse movements, keyboard input
  - Logical sensors
    - Combination of physical and virtual sensors with additional information from databases: analyzing logins at desktop pcs and a database mapping fixed devices to location information
- Raw data retrieval
  - Drivers and APIs
  - Query functionality (ex: getPosition())
  - Fusion
  - Exchangeable

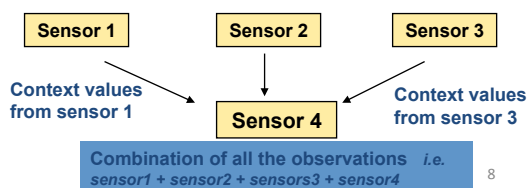
## Sensors and Raw data Retrieval

- Physical sensors
- Virtual sensors
- Logical sensors



### Sensor Fusion

- a more comprehensive view of the physical world



May 27, 10

## Abstract Layer Architecture

- Preprocessing
  - Reasoning and interpreting
  - Extraction and quantization operations
  - Aggregation or compositing
    - Statistical methods and training phase is required
  - Ex: not the exact GPS position of a person, but the name of the person
- Storage/Distribution/Management
  - Public interface to the client
  - Support for asynchronous access (e.g. Publish/Subscribe)
  - Synchronous (pull/polling) and asynchronous (push/subscription)
- Applications
  - Actual reaction on different events and context-instances is implemented

## Examples of Frameworks and Middleware for Context-awareness

- Systems
  - Context Toolkit
  - Context Managing Framework
  - CASS
  - Hydrogen
  - CORTEX
- Architecture
- Resource Discovery
- Sensing
- Context Model
- Context Processing
- Historical context data
- Security and Privacy



## Context Managing Framework

- Context Interpretation deduces semantically flavored descriptions of the entities' context
- It consists of a set of container components, which encapsulate communication, representation, privacy and security related functionalities.
- container components can subscribe to events (data) from other components.

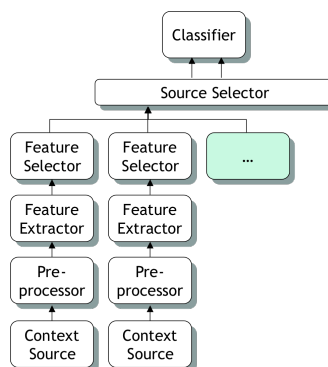


Fig. 2. A typical configuration of the Context Interpretation key function.

## Architecture: CASS (Context-awareness sub-structure)

- Centralized, server-based middleware supporting context-aware applications on hand-held devices.
- contains an Interpreter, a ContextRetriever, a RuleEngine and a SensorListener
- The SensorListener listens for update from local or remote sensor nodes and stores the information in database
- ContextRetriever is responsible for retrieving stored context data. Both may use an interpreter.

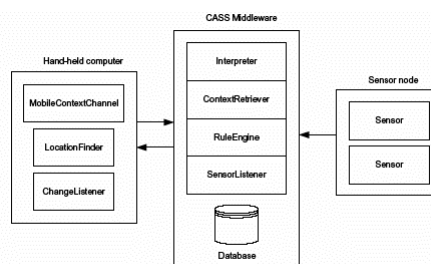
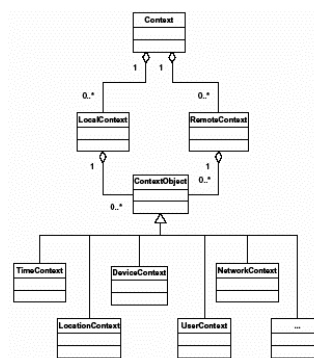


Fig. 3. Architecture of the CASS system

P Fahy, S Clarke, "CASS - Middleware for Mobile Context-Aware Applications", Mobisys 2004 Workshop on Context Awareness, 2004

## Architecture: Hydrogen

- Specializing in mobile devices
  - Remote context and local context
- Context sharing
  - In a peer-to-peer manner
- Object oriented approach
  - Superclass ContextObject



T Hofer, W Schwinger, M Pichler, G Leonhartsberger, "Context-Awareness on Mobile Devices - the Hydrogen Approach", System Sciences, 2003

## Architecture: Hydrogen

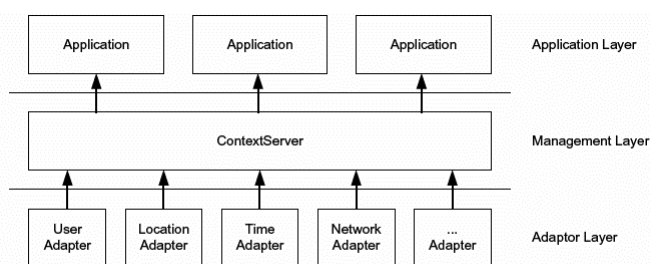


Fig. 5. Architecture of the Hydrogen project

- All located on the same device
- Context Server
  - Synchronous and asynchronous methods
- All inter-layer communication is based on a XML-protocol



## Architecture: CORTEX

(U. Lancaster, UK)

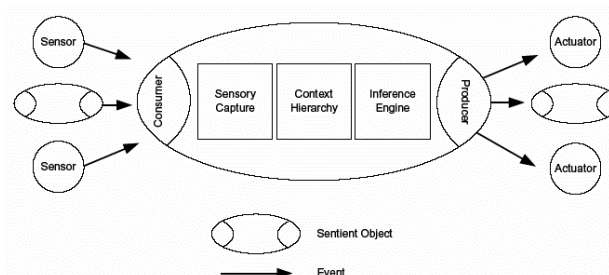
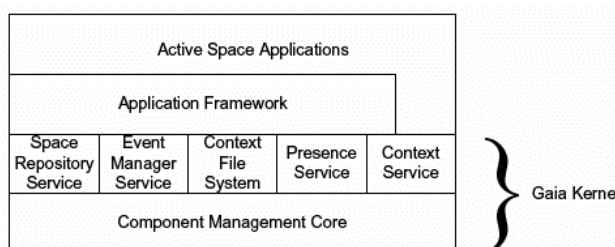


Fig. 6. The Sentient Object Model

- Middleware approach based on the Sentient Object Model. It was designed for the development of context-aware applications in an ad-hoc mobile environment
- Uses of STEAM
  - A location-aware event-based middleware service
- Supports a graphical development tool
  - Specify relevant sensors and actuators
  - Define fusion networks
  - Specify context hierarchy and production rules

## Architecture: Gaia (Urbana-Champaign, USA)

- Middleware infrastructure
  - Extends typical operating systems concepts to include context-awareness



R Manuel, K Christopher, "Gaia: A Middleware Infrastructure to Enable Active Spaces", - IEEE Pervasive Computing, 2002

## Other Frameworks and Middleware

- SOCAM (Service-oriented Context-Aware Middleware)
  - Centralized context interpreter
- CoBrA (Context Broker Architecture)
  - Agent based architecture
  - (Centralized) context broker
    - Context Knowledge Base, Context Inference Engine, Context Acquisition Module, Privacy Management Module
  - Broker federations
- Mobilis
  - Based on a centralized Broker (so far)
  - SIP-based Publish/Subscribe
  - Composable Context Providers
  - Context Providers can subscribe to remote providers

## Resource Discovery

- Discoverer [Context Toolkit]
  - A white page lookup (via names)
  - A yellow page lookup (via attributes)
- Service locating service [SOCAM]
- Registry component [Gaia]
- Pure p2p context-aware system only uses local built-in sensors [Hydrogen]

## Sensing

- “The separation of acquisition and use of context”
  - Context Widgets [Context Toolkit]
  - Sensor nodes [CASS]
  - Context providers [SOCAM]
  - Resource servers [Context Managing Framework]
  - Context acquisition components [CoBrA]
  - Context Providers [Mobilis]

## Context Model

- Attribute-value-tuples [Context Toolkit]
- Object-oriented context model [Hydrogen]
- Ontologies [SOCAM, CoBrA, Context Managing Framework]
- 4-ary predicates [Gaia]
  - (<ContextType>, <Subject>, <Relator>, <Object>)
  - Used for both representing context and forming inference rules

## Context Processing

- Context aggregators, context interpreters [Context Toolkit]
- Resource servers, context manager, context recognition services [Context Managing Framework]
- Context Reasoning Engine [SOCAM]
- Inference Engine [CoBrA]
- Inference engine and knowledge base [CASS]
- Sentient Objects [CORTEX]
- Context Service Module [Gaia]
  - First order logic: quantification, implication, conjunction, disjunction, and negation

## Historical context data

- A centralized high-resource storage component is needed
  - Database, SQL [Context Toolkit, CoBrA, CASS, SOCAM, CORTEX, Owl]
  - Context Knowledge Base [CoBrA, CASS]
- No persistent storage due to limited memory resources [Hydrogen]

## Security and Privacy

- Context ownership [Context Toolkit]
  - Mediated Widgets, Owner Permissions, a modified BaseObject and Authenticators
- Role Based Access Control (RBAC) [Owl]
- Rei, an own flexible policy language [CoBrA]

## Overview

	Sharing/ Discovery	Information Model	Context Composition	Inference/ Reasoning
Context Toolkit	P2P (needs discoverer)	Widget Abstraction	Yes	Not built-in
CMF	Central context Server	Resource Servers	No	Yes
CASS	Central context server	?	No	Yes
Hydrogen	P2P	Object based	No	No
CORTEX	Ad hoc mobile net	Object based (Sentinent Obj.)	Yes	Yes
Mobilis	Pub/Sub	CxP abstraction	Yes	No

## Other References

- Meyer, S. Rakotonirainy, A. A Survey of Research on Context-Aware Homes University of Queensland Brisbane, Australia.
- Fujii, A. Trends and Issues in Research on Context Awareness Technologies for a Ubiquitous Network Society. Information and Communications Research Unit, Japan.
- Loke, S. Context-Aware Pervasive Systems: Architectures for a New Breed of Applications. Auerbach Publications, Taylor & Francis Group, Boca Raton; New York. ISBN: 0-8493-7255-0, 2007 (Book)
- Poslad, S. Ubiquitous Computing: Smart Devices, Environments and Interactions John Wiley & Sons Ltd, ISBN: 978-0-470-03560-3, 2009 (Book)