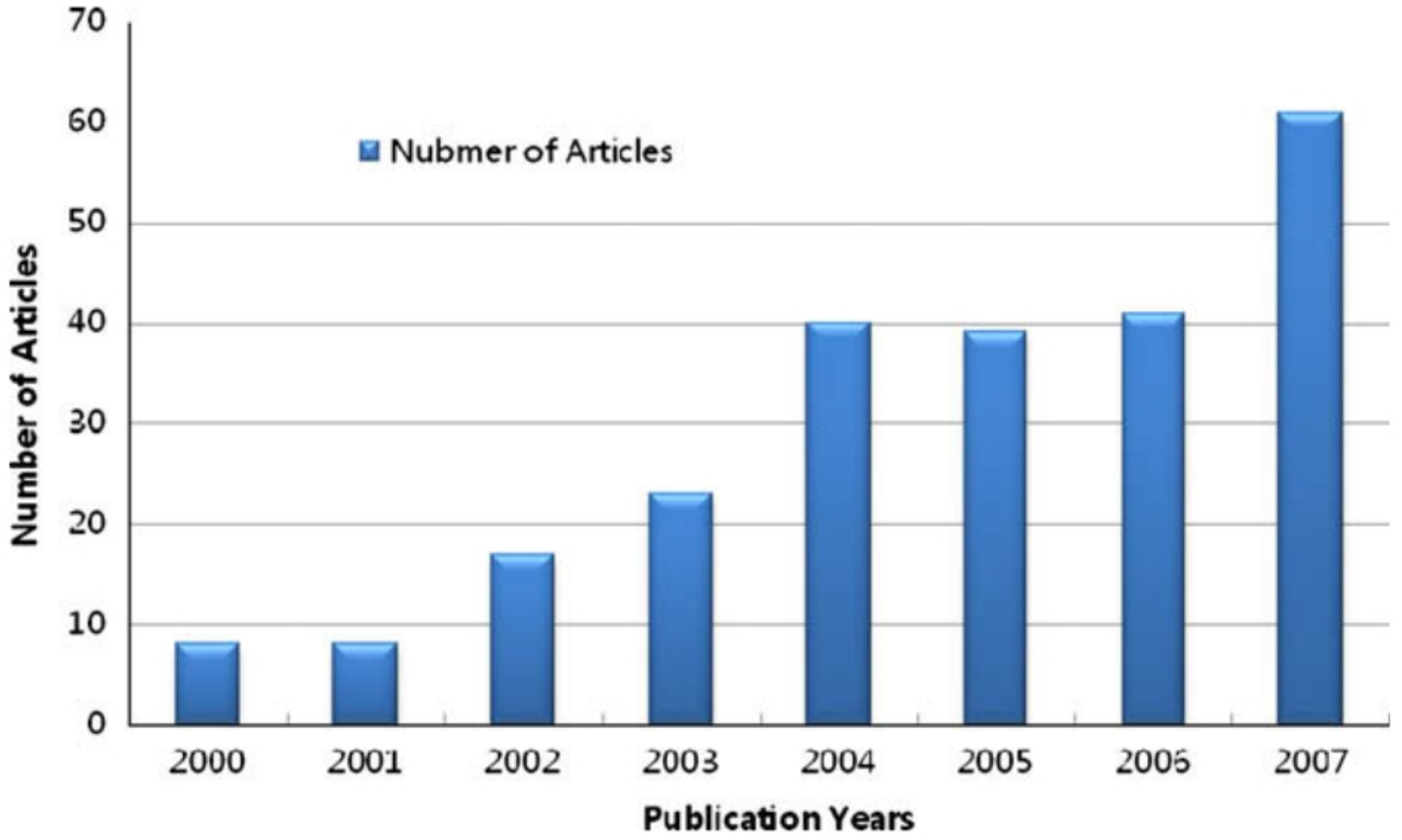
Context-Aware Systems

Michael Maynord
Feb. 24, 2014

- The precise definition of 'context' is contentious. Here we will be using 'context' as “any information that can be used to characterize the situation of entities (i.e., whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves.”

- Much work on context-aware systems, popularity is increasing.



- Research on context-aware systems can be conceived of as working over five layers:
  - User interface
  - Application
  - Middleware
  - Network
  - Concept and Research

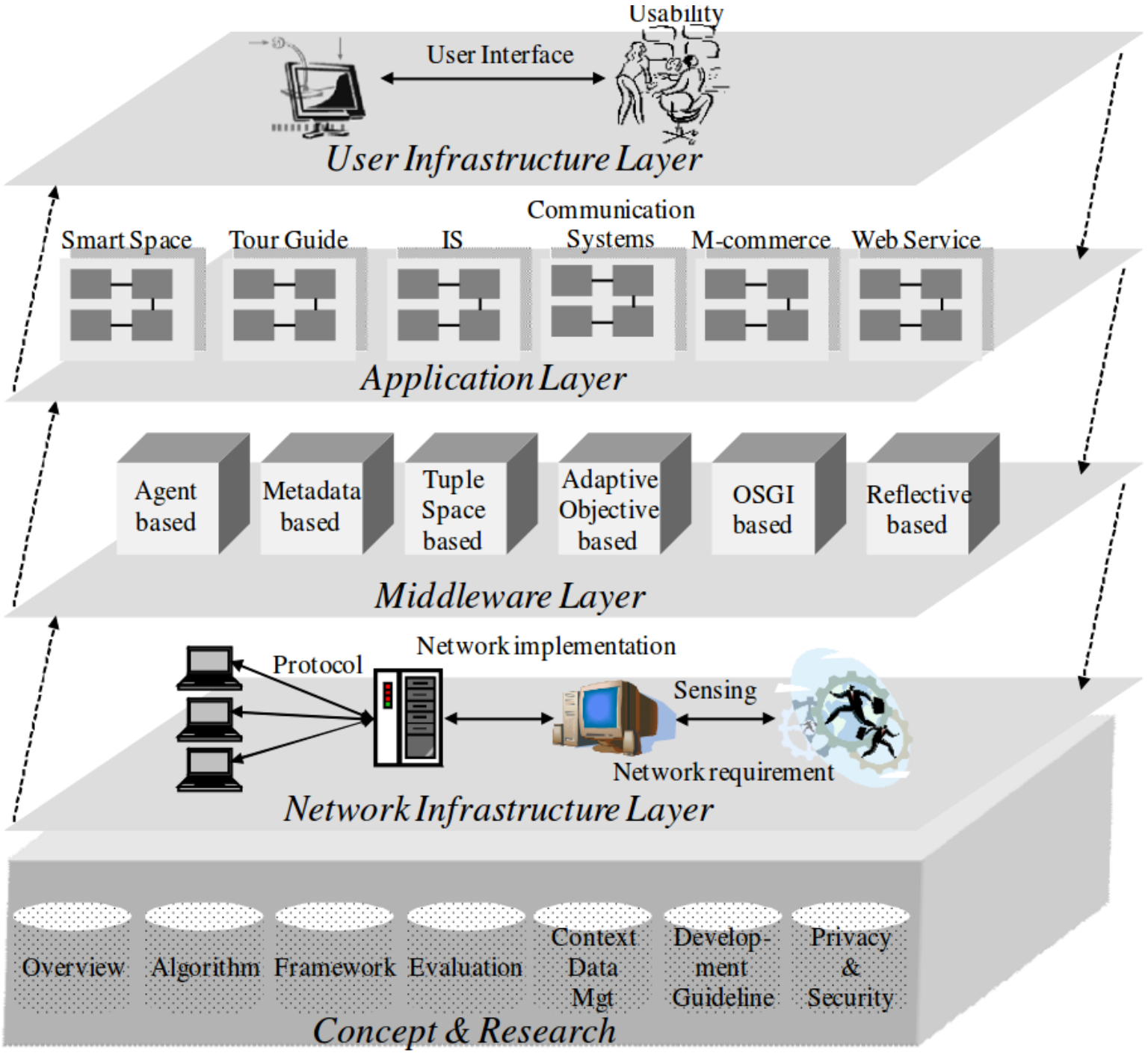


Fig. 4. Classification framework of context-aware systems.

To get an idea of where research is taking place (table with more detail in the paper):

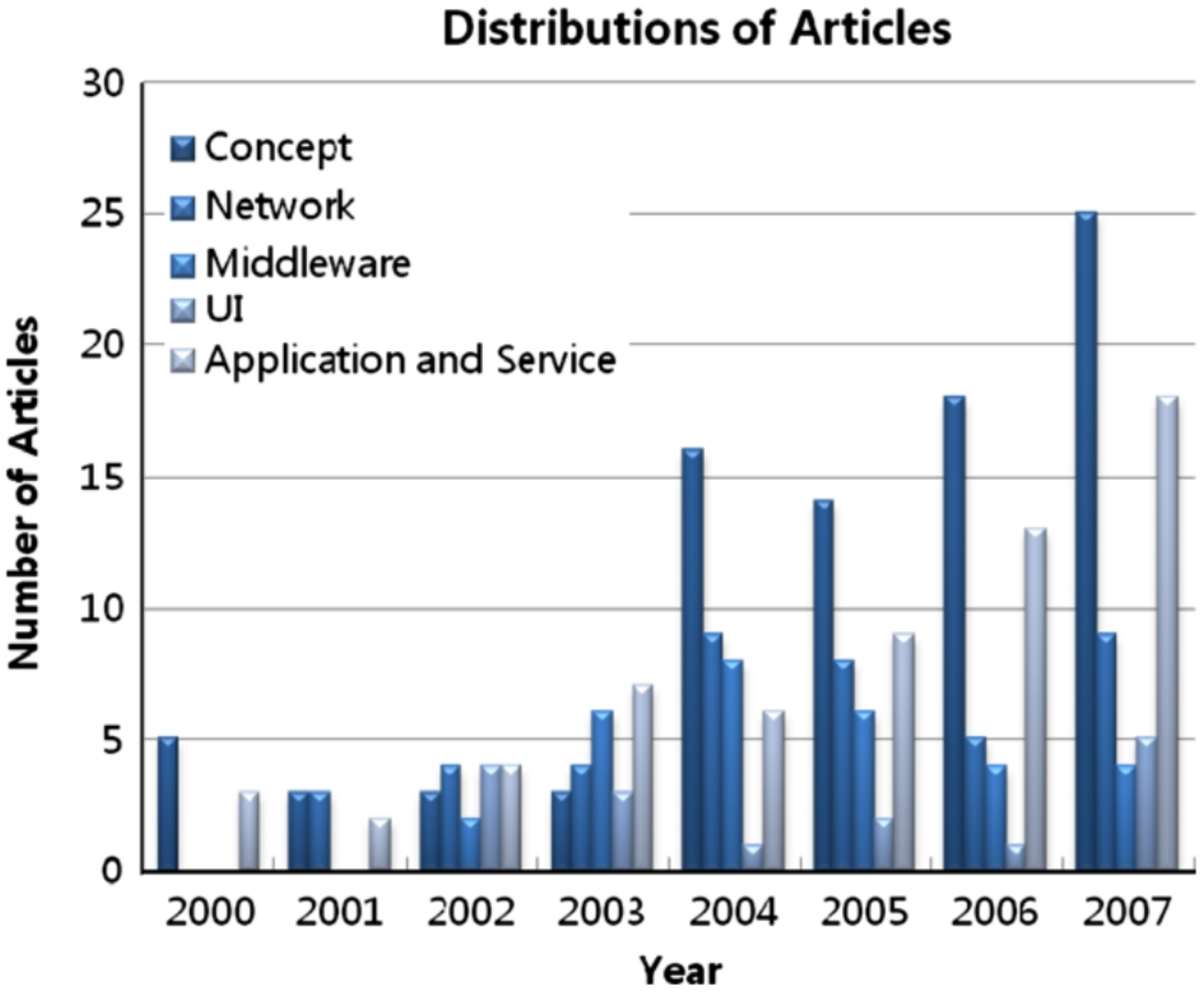


Fig. 5. Distribution of articles number by year and classification framework.

The scope of context-aware systems is typically very limited; in increasing scope the following questions are of concern:

- 1) How to extract and use the cognitive context in context-aware application?
  - Most context aware systems are presently aware only of physical context; this needs to expand this to internal state of the user.
- 2) What are design patterns of context-aware systems?
  - A compiled list of design patterns could help prevent the resolving of previously solved issues.
- 3) Which is the best inferring algorithm to extract user context and provide service to user?
  - Different contexts are amenable to different algorithms, a mapping between the two would be useful.
- 4) How to deal with concurrently enormous data, information and knowledge having different format to serve suitable service to users?
  - It is not clear in the general case how best to integrate different aspects of a context.
- 5) How to extract the best solution when the context of users is conflicted?
  - Sensors can give conflicting information. If a system may have multiple users with conflicting preferences.
- 6) How to reflect the preference of users for satisfying user needs?
  - A user's preference may be predicted from context and a user profile.
- 7) How to save users information in context-aware systems?
  - There are issues of security, privacy, and authentication.
- 8) How to evaluate performance of context-aware systems?
  - What does it mean for one context-aware system to be better than another?

Notes on context

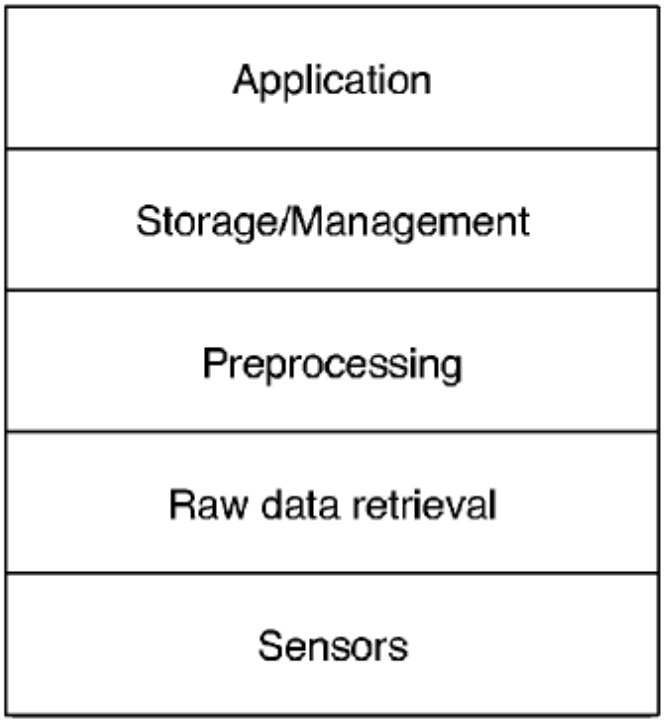
- Context has at least two dimensions:
  - Internal vs. External
  - Physical vs. Logical
- Context can be acquired from any of:
  - Direct sensors
  - Middleware infrastructure
  - Context server

Notes on context

- Context can be managed in at least three ways:
  - With widgets
  - As networked services
  - Using a blackboard model
- I key idea of many context-aware systems is the separation of detecting and using context.

Notes on context-aware systems

Figure 1 Layered conceptual framework for context-aware systems



Notes on context-aware systems

- Sensors:
  - Physical (gps, etc.)
  - Virtual (calendar, emails, etc.)
  - Logical
    - Composite sensors: combinations of physical and virtuals sensors, databases, etc.

Notes on context-aware systems

- Raw-data retrieval:
  - Drivers and APIs are used to interface with the sensors themselves.
- Preprocessing:
  - Not implemented in every system.
  - Abstracts over context atoms to give aggregate or composite information.
    - This could be left to the application, but including it in a context-aware framework has advantages.

Notes on context-aware systems

- Storage and management:
  - Organize data and provide access via a public interface.
    - Clients can access data synchronously (polling) or asynchronously (subscription).
- Application
  - The client is realized here.

Notes on contextual models

- Context models:
  - Key-Value
  - Markup scheme
  - Graphical model
  - Object oriented models
  - Logic based models
  - Ontologies

Notes on contextual models

- A contextual model should be:
  - Simple
  - Flexible and extensible
  - Generic
  - Expressive
- Context atoms should have at least:
  - Context type (temp., time, speed, etc.)
  - Context value
  - Timestampe
  - Source
  - Confidence

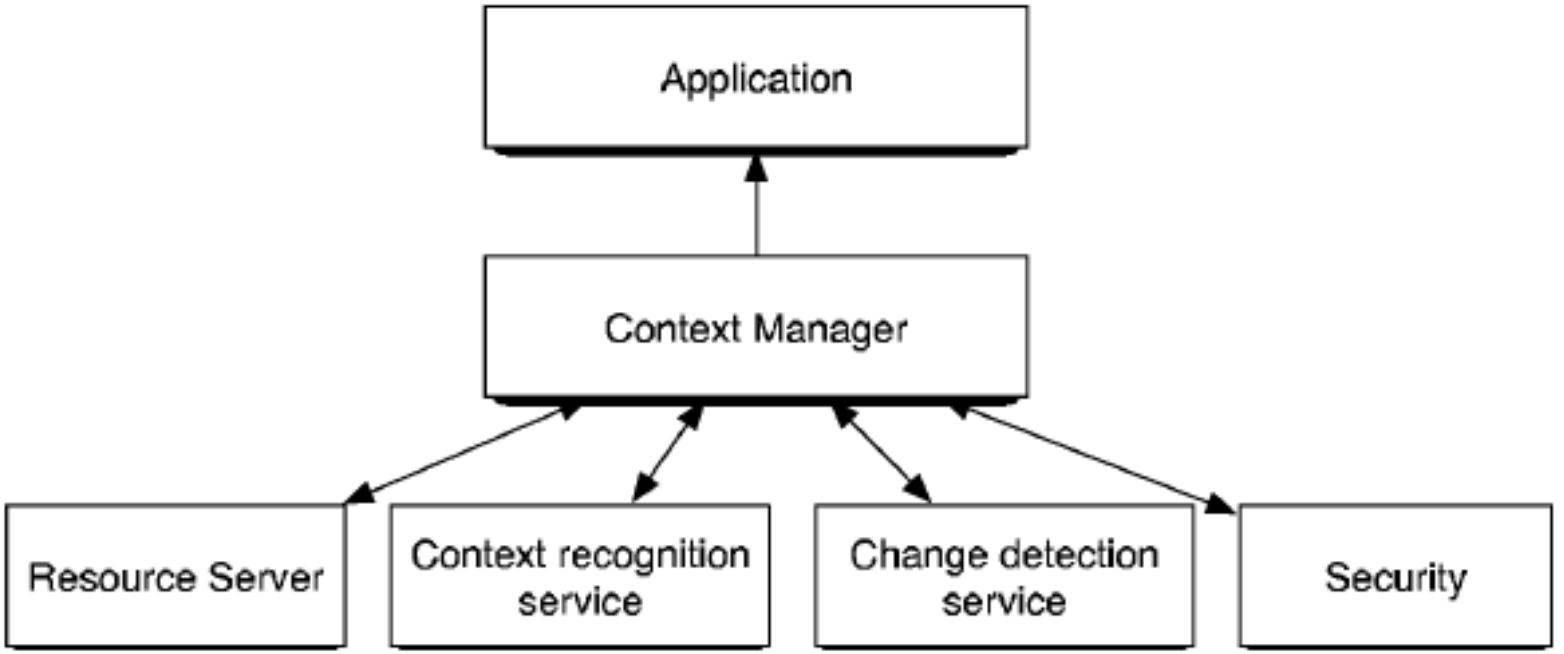
Context-aware architectures

Table 4 Summary of discussed approaches

Architecture	*Sensing*	*Context model*	*Context processing*	*Resource discovery*	*Historical context data*	*Security and privacy*	
CASS	Centralised middleware	Sensor nodes	Relational data model	Inference engine and knowledge base	n.a.	Available	n.a.
CoBra	Agent based	Context acquisition module	Ontologies (OWL)	Inference engine and knowledge base	n.a.	Available	Rei policy language
Context Management Framework	Blackboard based	Resource servers	Ontologies (RDF)	Context recognition service	Resource servers + subscription mechanism	n.a.	n.a.
Context Toolkit	Widget based	Context widgets	Attribute-value tuples	Context interpretation and aggregation	Discoverer component	Available	Context ownership
CORTEX	Sentient object model	Context component framework	Relational data model	Service discovery framework	Resource management component framework	Available	n.a.
Gaia	MVC (extended)	Context providers	4-ary predicates (DAML + OIL)	Context-service module (first-order logic)	Discovery service	Available	Supported (e.g., secure tracking, location privacy, access control)
Hydrogen	Three layered architecture	Adapters for various context types	Object-oriented	Interpretation and aggregation of raw data only	n.a.	n.a.	n.a.
SOCAM	Distributed with centralised server	Context providers	Ontologies (OWL)	Context reasoning engine	Service locating service	Available	n.a.

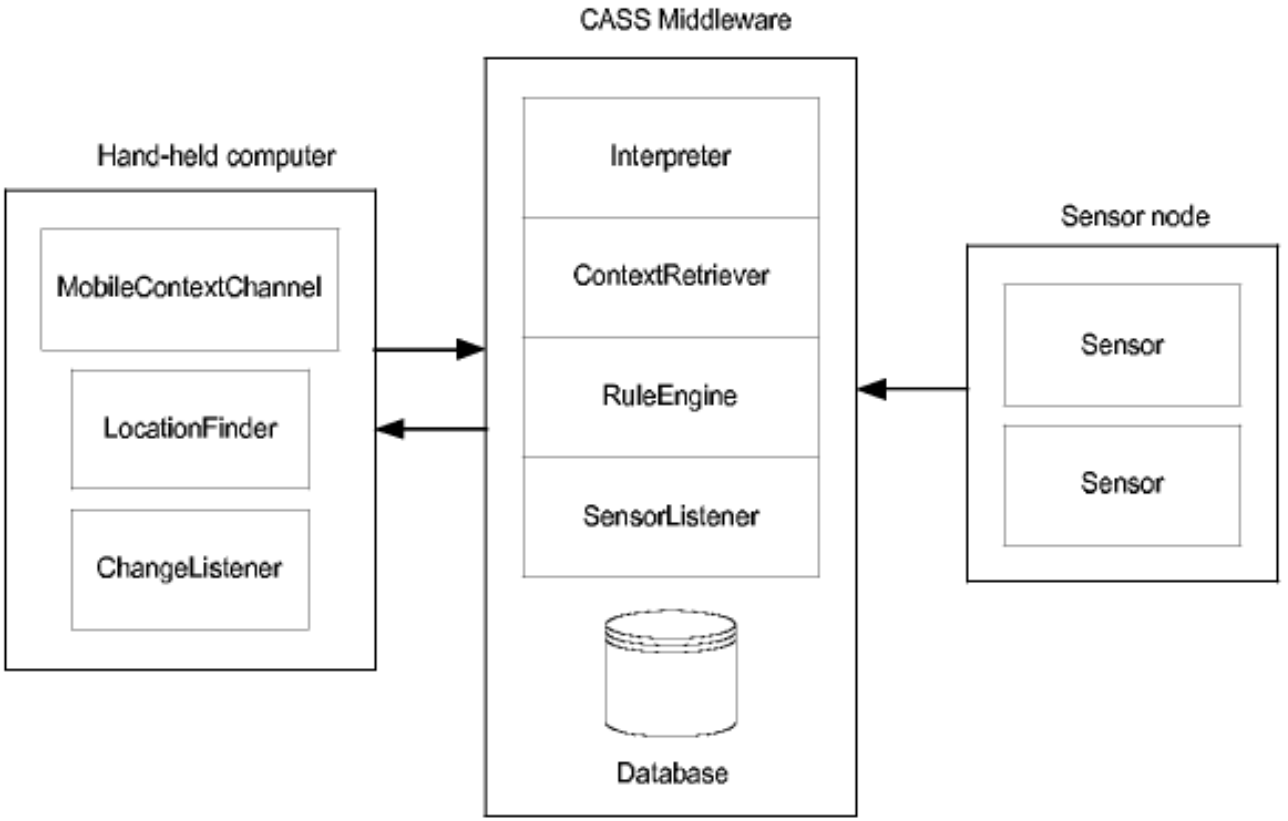
Context-aware architectures

Figure 2 *Context managing framework* architecture



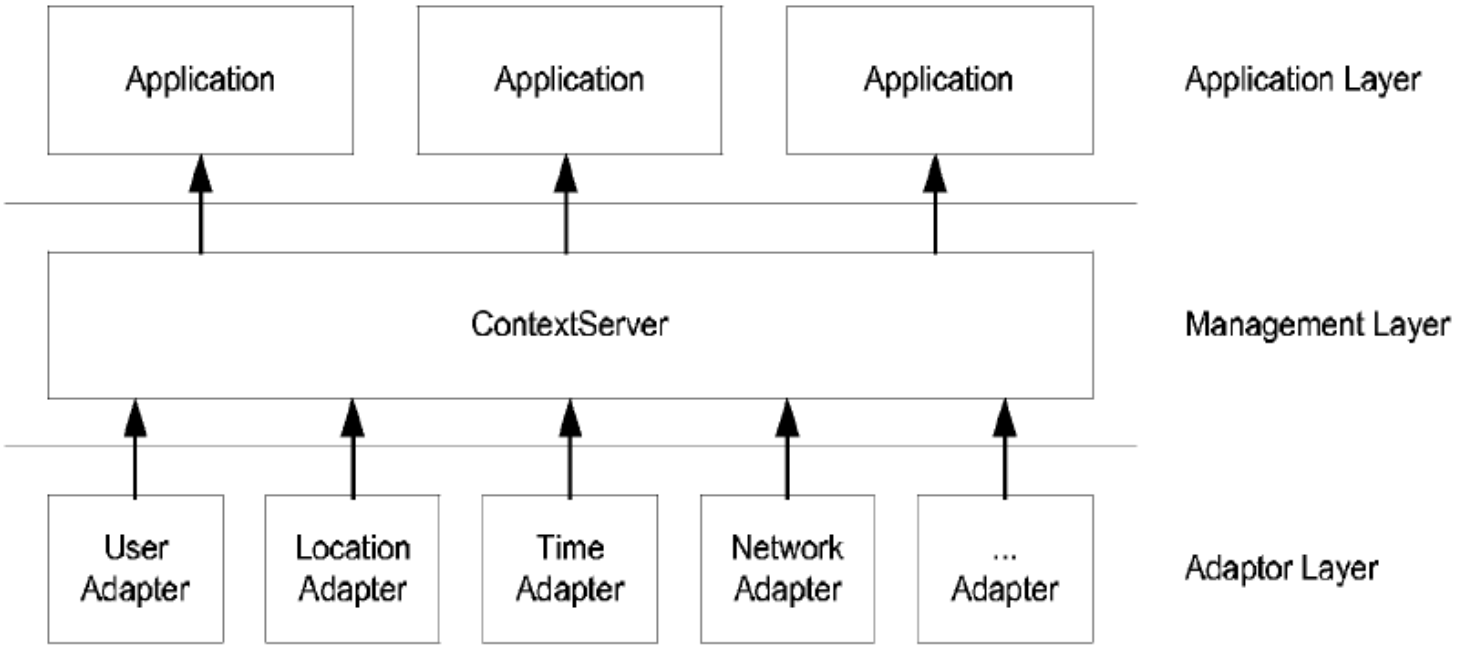
Context-aware architectures

Figure 3 Architecture of the CASS system



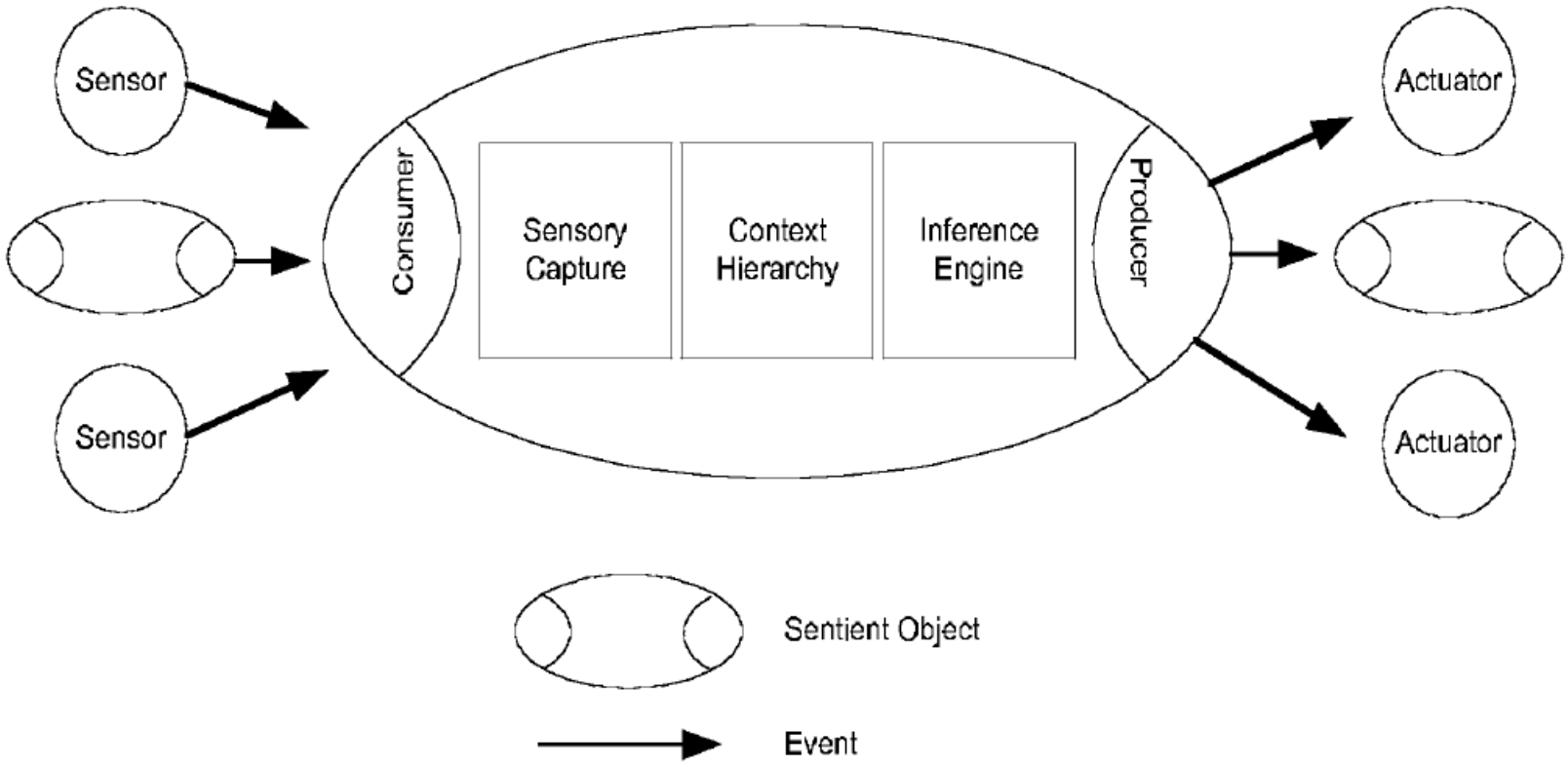
Context-aware architectures

Figure 5 Architecture of the hydrogen project



Context-aware architectures

Figure 6 The sentient object model



Discussion

- The form of an architecture is driven in part by the context acquisition method, and an architecture is considered reasonable if it adequately separates concerns of context acquisition and the user components.
- Encapsulation of sensors is an important theme.
- Ontologies are effective methods for representing context.
- Many architectures do not allow for adequate resource discovery, as there is an implicit assumption that context sources are stable and always available. This is not always a valid assumption.

Discussion

- Keeping a history of context enables learning and more adaptive behavior.
- Security and privacy concerns often don't receive the attention they deserve.
- A key feature of the architectures shown is a division of context acquisition and use, with a middleware layer in between.
- Different architectures use different encodings and methods for accessing context sources. Different system have different communication mechanisms and different context representations.
  - Standardization of formats and protocols could ease the development of context-aware services.

Papers

- Jong-yi Hong, Eui-ho Suh, and Sung-Jin Kim. 2009. Context-aware systems: A literature review and classification. Expert Syst. Appl. 36, 4 (May 2009), 8509-8522.
- Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. 2007. A survey on context-aware systems. Int. J. Ad Hoc Ubiquitous Comput. 2, 4 (June 2007), 263-277.