

A Context Modeling Survey

Thomas Strang

German Aerospace Center (DLR)
Institute for Communications and Navigation
D-82230 Wessling/Oberpfaffenhofen, Germany
thomas.strang@dlr.de

Claudia Linnhoff-Popien

Ludwig-Maximilians-University Munich (LMU)
Institute for Informatics
D-80538 Munich, Germany
linnhoff@informatik.uni-muenchen.de

ABSTRACT

Context-awareness is one of the drivers of the ubiquitous computing paradigm, whereas a well designed model is a key accessor to the context in any context-aware system. This paper provides a survey of the the most relevant current approaches to modeling context for ubiquitous computing. Numerous approaches are reviewed, classified relative to their core elements and evaluated with respect to their appropriateness for ubiquitous computing.

1. INTRODUCTION

In the past a variety of context models were subject of research, because a well designed model is a key accessor to the context in any context-aware system. While early models mainly addressed the modeling of context with respect to one application or an application class, generic context models are of interest since many applications can benefit from these. While some models take the users current situation, e.g. “in a meeting” into account, others model the physical environment, i.e. locations. First steps towards a common understanding of context have been published, mostly with respect to location, identity, and time. The objective of most current research is to develop uniform context models, representation and query languages as well as reasoning algorithms that facilitate context sharing and interoperability of applications.

In this paper we want to make a survey of the most relevant current approaches to modeling context for ubiquitous computing. We start in section 2 with some specific requirements arising from ubiquitous computing. Section 3 will give a comprehensive introduction to the context modeling approaches under examination in section 4, before we summarize our paper with a conclusion in section 5.

2. FUNDAMENTALS

In the literature several definitions of the term *context* can be found [36, 35, 32, 37, 10, 16, 41]. A detailed discussion of the differences within these definitions is out of the scope of this paper but has some impact on the models introduced in the next section. A selection of some context-aware mobile computing research is for instance provided by [10, 39].

The concept of *context-awareness* increasingly gained im-

portance in the area of distributed systems since the 90’s, since it seemed to be a promising solution for a lot of problems which have been implied by the usage of mobile terminals in ever-changing environments. Context dependency is thus a major issue in recent research work in the area of ubiquitous computing systems which are specialisations of mobile, distributed systems as is outlined in the evolution chain in figure 1.

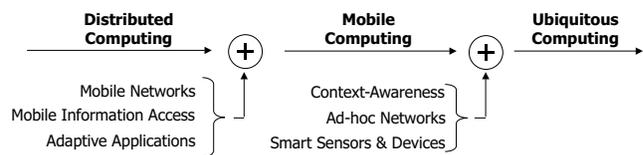


Figure 1: Evolution Chain

Ubiquitous computing systems make high demands on any context modeling approach in terms of:

1. *distributed composition (dc)*: Any ubiquitous computing system is a derivative of a distributed computing system (cf. figure 1) which lacks of a central instance being responsible for the creation, deployment and maintenance of data and services, in particular context descriptions. Instead, composition and administration of a context model and its data varies with notably high dynamics in terms of time, network topology and source.
2. *partial validation (pv)*: It is highly desirable to be able to partially validate contextual knowledge on structure as well as on instance level against a context model in use even if there is no single place or point in time where the contextual knowledge is available on one node as a result of distributed composition. This is particularly important because of the complexity of contextual interrelationships, which make any modeling intention error-prone.
3. *richness and quality of information (qua)*: The quality of a information delivered by a sensor varies over time, as well as the richness of information provided by different kinds of sensors characterizing an entity in an ubiquitous computing environment may differ. Thus a context model appropriate for usage in ubiquitous computing should inherently support quality and richness indication.
4. *incompleteness and ambiguity (inc)*: The set of contextual information available at any point in time characterizing relevant entities in ubiquitous computing environments is usually incomplete and/or ambiguous, in particular if this information is gathered from sensor net-

works. This should be covered by the model, for instance by interpolation of incomplete data on the instance level.

5. *level of formality (for)*: It is always a challenge to describe contextual facts and interrelationships in a *precise* and *traceable* manner. For instance, to perform the task “print document on printer near to me”, it is required to have a precise definition of terms used in the task, for instance what “near” means to “me”. It is highly desirable, that each participating party in an ubiquitous computing interaction shares the same interpretation of the data exchanged and the meaning “behind” it (so called *shared understanding*).
6. *applicability to existing environments (app)*: From the implementation perspective it is important that a context model must be applicable within existing the infrastructure of ubiquitous computing environments, e.g. a service framework such as Web Services.

The mentioned requirements are in particular important for any context modeling approach applied to an ubiquitous computing environment. Some of the requirements are addressed within a certain approach’s context model, some are addressed within the associated reasoning system, and some are not addressed at all within a certain approach.

3. MODELING APPROACHES

Throughout this section we will survey the most relevant context modeling approaches. These are classified by the scheme of data structures which are used to exchange contextual information in the respective system. (Obviously some of them may be classified in more than one category. In these cases they are listed in the most representative one.)

3.1 Key-Value Models

The model of key-value pairs is the most simple data structure for modeling contextual information.

Already Schilit et al. [35] used key-value pairs to model the context by providing the value of a context information (e.g. location information) to an application as an environment variable. The key-value modeling approach is frequently used in distributed service frameworks (e.g. Capeus [34]). In such frameworks, the services itself are usually described with a list of simple attributes in a key-value manner, and the employed service discovery procedure (e.g. SLP, Jini, . . . see [39]) operates an exact matching algorithm on these attributes.

In particular, key-value pairs are easy to manage, but lack capabilities for sophisticated structuring for enabling efficient context retrieval algorithms.

3.2 Markup Scheme Models

Common to all markup scheme modeling approaches is a hierarchical data structure consisting of markup tags with attributes and content. In particular, the content of the markup tags is usually recursively defined by other markup tags.

Typical representatives of this kind of context modeling approach are *profiles*. They usually base upon a serialization of a derivative of *Standard Generic Markup Language*

(*SGML*), the superclass of all markup languages such as the popular XML. Some of them are defined as extension to the *Composite Capabilities / Preferences Profile (CC/PP)* [44] and *User Agent Profile (UAProf)* [46] standards, which have the expressiveness reachable by RDF/S and a XML serialization. These kinds of context modeling approaches usually extend and complete the basic CC/PP and UAProf vocabulary and procedures to try to cover the higher dynamics and complexity of contextual information compared to static profiles.

An example of this approach are the *Comprehensive Structured Context Profiles (CSCP)* by Held et al. [23]. Unlike CC/PP, CSCP does not define any fixed hierarchy. It rather supports the full flexibility of RDF/S to express natural structures of profile information as required for contextual information. Attribute names are interpreted context-sensitively according to their position in the profile structure. Hence, unambiguous attribute naming across the whole profile as necessary with CC/PP is not required. Another drawback of CC/PP, the restricted overriding mechanism of default values only, replaced by a more flexible overriding and merging mechanism, allowing for instance to override and/or merge a whole profile subtree. See figure 2 for a CSCP profile example.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cscp="context-aware.org/CSCP/CSCPProfileSyntax#"
  xmlns:dev="context-aware.org/CSCP/DeviceProfileSyntax#"
  xmlns:net="context-aware.org/CSCP/NetworkProfileSyntax#"
  xmlns="context-aware.org/CSCP/SessionProfileSyntax#"
  <SessionProfile rdf:ID="Session">
    <cscp:defaults rdf:resource=
      "http://localContext/CSCPProfile/previous#Session"/>
    <device><dev:DeviceProfile>
      <dev:hardware><dev:Hardware>
        <dev:memory>9216</dev:memory>
      </dev:Hardware></dev:DeviceProfile>
    </device>
  </SessionProfile>
</rdf:RDF>
```

Figure 2: CSCP profile example

A similar approach to CSCP is the *CC/PP Context Extension* by Indulska et al. [27]. They extended the basic CC/PP and UAProf vocabulary by a number of component-attribute trees related to some aspects of context, e.g. concerning location, network characteristics, application requirements, session information as well as certain types of relations and dependencies. The authors concluded that their approach is capable of enabling context-awareness to applications and other parts of ubiquitous computing infrastructure. They already realized, that it is difficult and non-intuitive to capture complex contextual relationships and constraints due to the underlying CC/PP.

Another context modeling approach in the markup scheme category – which does not bear towards CC/PP – is the *Pervasive Profile Description Language (PPDL)* [14]. This XML-based language allows to account for contextual information and dependencies when defining interaction patterns on a limited scale. The number of evaluable contextual aspects and the comprehensiveness of the language itself seems to be relatively limited. Due to the fact that no design criteria and

only parts of the language are available to the public, the actual appropriateness of this context modeling approach remains unknown.

There are several other context modeling approaches in the markup scheme category. They are oftentimes either proprietary or limited to a small set of contextual aspects, or both. Examples affected by these limitations are, among others, the *context configuration* of Capra et al.'s reflective middleware [9] the *Centaurus Capability Markup Language (CCML)* [28], *ConteXtML* [33] or the note-tags of the *stick-e notes* system [7].

3.3 Graphical Models

A very well known general purpose modeling instrument is the *Unified Modeling Language (UML)* which has a strong graphical component (UML diagrams). Due to its generic structure, UML is also appropriate to model the context. This is shown for instance by Bauer in [5], where contextual aspects relevant to air traffic management are modeled as UML extensions.

Another example is the nicely designed graphics oriented context model introduced in [26] by Henricksen et al., which is a context extension to the Object-Role Modeling (ORM) approach [22] according some contextual classification and description properties [25]. In ORM, the basic modeling concept is the *fact*, and the modeling of a domain using ORM involves identifying appropriate fact types and the roles that entity types play in these. Henricksen extended ORM to allow fact types to be categorised, according to their persistence and source, either as *static* (facts that remain unchanged as long as the entities they describe persist) or as *dynamic*. The latter ones are further distinguished depending on the source of the facts as either *profiled*, *sensed* or *derived* types. Another quality indicator introduced by Henricksen is a history fact type to cover a time-aspect of the context. The last extension to ORM made by Henricksen for context modeling purposes are fact dependencies, which represent a special type of relationship between facts, where a change in one fact leads automatically to a change in another fact: the *dependsOn* relation. See figure 3 on the right for an example of Henricksen's notation.

This kind of approach is particularly applicable to derive an ER-model [12] from it, which is very useful as structuring instrument for a relational database in an information system based context management architecture such as the one described in [27].

3.4 Object Oriented Models

Common to object oriented context modeling approaches is the intention to employ the main benefits of any object oriented approach - namely encapsulation and reusability - to cover parts of the problems arising from the dynamics of the context in ubiquitous environments. The details of context processing is encapsulated on an object level and hence hidden to other components. Access to contextual information is provided through specified interfaces only.

A representative for this kind of approach are the *cues* [37] developed within the TEA project [1, 38]. The concept of

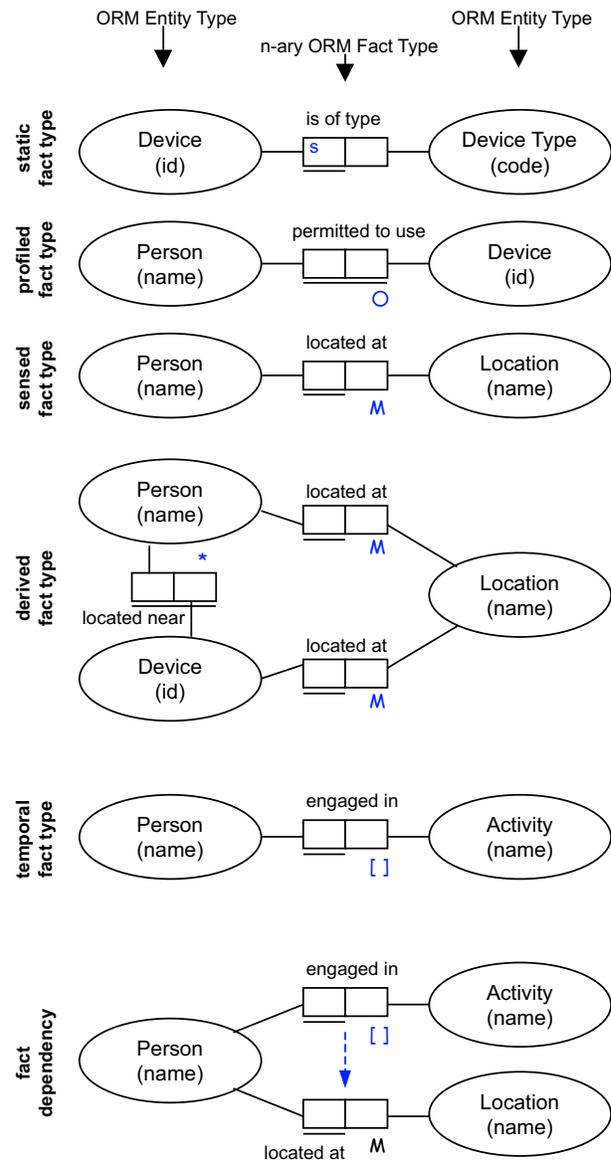


Figure 3: Contextual Extended ORM

cues provides an abstraction from physical and logical sensors. A cue is regarded as a function taking the value of a single physical or logical sensor up to a certain time as input and providing a symbolic or sub-symbolic output. A finite or infinite set of possible values is defined for each cue. The output of each cue depends on a single sensor, but different cues may be based on the same sensors. The context is modeled as an abstraction level on top of the available cues. Thus the cues are objects providing contextual information through their interfaces, hiding the details of determining the output values.

Another approach within the object category is the *Active Object Model* of the GUIDE project [13]. Again, the chosen approach has been primarily driven by the requirement of being able to manage a great variety of personal and environmental contextual information while maintaining sca-

lability. All the details of data collection and fusing (e.g. the context adaptive composition of HTML fragments) are encapsulated within the active objects and thus hidden to other components of the system.

The approach of Bouzy and Cazenave [6] followed a similar intention: They propose to use general object oriented mechanisms to represent contextual knowledge about temporal, goal, spatial and global contexts in computer Go (a 4000 years old game that is very famous in Japan, China and Korea). They justified their object oriented context modeling approach with its inheritance and reutilization capabilities, allowing “to define the smallest number of properties, functions and rules [...] in order to simplify knowledge representation in very complex domains and systems”.

3.5 Logic Based Models

A logic defines the conditions on which a concluding expression or fact may be derived (a process known as reasoning or inferencing) from a set of other expressions or facts. To describe these conditions in a set of rules a formal system is applied. In a logic based context model, the context is consequently defined as facts, expressions and rules. Usually contextual information is added to, updated in and deleted from a logic based system in terms of facts or inferred from the rules in the system respectively. Common to all logic based models is a high degree of formality.

One of the first logic based context modeling approaches has been researched and published as *Formalizing Context* in early 1993 by McCarthy and his group at Stanford [29, 30]. McCarthy introduced contexts as abstract mathematical entities with properties useful in artificial intelligence. He prevented emphatically to give a definition what context is. Instead he tried to give a formalization recipe which allows for simple axioms for common sense phenomena, e.g. axioms for static blocks worlds situations, to be *lifted* to context involving fewer assumptions, e.g. contexts in which situations change. Thus lifting rules, which relate the truth in one context to the truth in another context, are an important part of the model itself. The basic relation in this approach is $ist(c, p)$, which asserts that the it proposition p is true in the context c . This allows for formulas such as $c0: ist(context-of(“Sherlock Holmes stories”), “Holmes is a detective”)$, where $c0$ is considered to be an outer context. McCarthy’s model already supports the concept of inheritance.

The main focus of Giunchiglia’s approach, sometimes referred to as *Multicontext Systems*, is less on context modeling than on context reasoning [18, 17]. He take a context to be that specific subset of the complete state of an individual entity that is used for reasoning about a given goal; it is seen as a (partial) theory of the world which encodes an individual’s subjective perspective about it.

Another early representative of this kind of approach is the *Extended Situation Theory* by Akman and Surav [2]. As the name implies it extends the *Situation Theory* which has been proposed by Barwise and Perry [4]. Barwise and Perry tried to cover model-theoretic semantics of natural language in a formal logic system. Akman and Surav used and extended this system to model the context with situation types which

are ordinary situations and thus first-class objects of situation theory. The variety of different contexts is addressed in form of rules and presuppositions related to a particular point of view. They represent the facts related to a particular context with parameter-free expressions supported by the situation type which corresponds to the context. Confer figure 4 to see a short example of how the rules of a context are represented as constraints in their approach.

$$\begin{aligned} S_1 &= [\dot{s} \mid \dot{s} \models \ll \textit{bird}, \dot{a}, 1 \gg] \\ S_2 &= [\dot{s} \mid \dot{s} \models \ll \textit{flies}, \dot{a}, 1 \gg] \\ B &= \ll \textit{present}, \textit{air}, 1 \gg \wedge \ll \textit{penguin}, \dot{a}, 0 \gg \wedge \dots \\ C &= S_1 \Rightarrow S_2 \mid B \end{aligned}$$

Figure 4: Example from Extended Situation Theory

A similar approach is the *Sensed Context Model* proposed by Gray and Salber [19]. They use first-order predicate logic as a formal representation of contextual propositions and relations.

Another approach within this category is the multimedia system by Bacon et al. [3]. In this system the location as one aspect of the context is expressed as facts in a rule based system. The system itself is implemented in Prolog.

3.6 Ontology Based Models

Ontologies are a promising instrument to specify concepts and interrelations [43, 20]. They are particularly suitable to project parts of the information describing and being used in our daily life onto a data structure utilizable by computers.

One of the first approaches of modeling the context with ontologies has been proposed by Öztürk and Aamodt [31]. They analysed psychological studies on the difference between *recall* and *recognition* of several issues in combination with contextual information. From this examination they derived the necessity of normalizing and combining the knowledge from different domains. They proposed a context model based on ontologies due to their strengths in the field of normalization and formality.

Another approach within the ontology category has been proposed as the *Aspect-Scale-ContextInformation (ASC)* model [39]. Using ontologies provides an uniform way for specifying the model’s core concepts as well as an arbitrary amount of subconcepts and facts, altogether enabling contextual knowledge sharing and reuse in an ubiquitous computing system [15]. This contextual knowledge is evaluated using ontology reasoners. The model has been implemented applying selected ontology languages. These implementations build up the core of a non monolithic *Context Ontology Language (CoOL)*, which is supplemented by integration elements such as scheme extensions for Web Services and others [41, 40]. Beyond determination of service interoperability in terms of contextual compatibility and substitutability, this language is used to support context-awareness in distributed service frameworks for various applications. For instance a contextual motivated *non-carrier service handover* is presented as one of the applications [42].

The CONON context modeling approach by Wang et al. [21, 45] is based on the same idea of the ASC/CoOL approach, namely to develop a context model based on ontologies because of its knowledge sharing, logic inferencing and knowledge reuse capabilities. Wang et al. created an upper ontology which captures general features of basic contextual entities and a collection of domain specific ontologies and their features in each subdomain. The CANON ontologies are serialized in OWL-DL which has a semantic equivalence to well researched description logics. This allows for consistency checking and contextual reasoning using inference engines developed for description languages.

A promising emerging context modeling approach based on ontologies is the CoBrA system [11]. This system provides a set of ontological concepts to characterize entities such as persons, places or several other kinds of objects within their contexts. The CoBrA system uses a broker-centric agent architecture to provide runtime support for context-aware systems, particularly in *Intelligent Meeting Rooms*, a prevalent scenario of an ubiquitous computing environment.

4. EVALUATION

In the following the context modeling approaches surveyed in section 3 are evaluated concerning the requirements introduced in section 2. The design criteria of some of the approaches as well as the detailed specifications are unfortunately not available to the public. In these cases the assessment of the approach may be estimated with the properties of other approaches in the same category.

Beside inefficiencies in describing complex contextual information as mentioned in 3.1, it is common to all key-value models, that they are weak on the requirements 1 to 5. Distributed composition and the handling of incompleteness is possible on the instance level only. There is no scheme or at least range definitions available to check against, making partial validation a difficult task and any kind of matching algorithm error-prone at runtime. The simplicity of key-value pairs is an advance from the management and error risk perspective, but it is a drawback if quality meta-information or ambiguity shall be considered. Solely the applicability to existing ubiquitous computing environments is a strength of this kind of context modeling approach.

Markup scheme models (section 3.2) are strong concerning the partial validation requirement. There usually exists a scheme definition and a set of validation tools which can be used for type checking, even for complex types. Range checking is also possible to some degree for numerical values. But incompleteness and ambiguity have to be handled proprietary on the application level. If and how far the distributed composition requirement is met depends on the single approach. Standard CC/PP and UAProf have only restricted overriding and merging mechanisms which are required for distributed composition. This weakness is addressed within the CSCP approach by providing more flexible overriding and merging mechanisms. It is worth mentioning that Indulska et al. [27] as well as Butler [8] made negative experiences with CC/PP and UAProf based context models because of the constraints imposed by the XML serialization respectively the representation in RDF. Furthermore, they identified flaws in the de-

sign of CC/PP itself, for instance pertaining to the method of updating values or regarding the absence of relational constraints. Another drawback concerning distributed composition has to be tackled if Document Type Definitions (DTDs) are used on the markup structuring level - they do not provide overriding or merging. Quality meta-information may be added to contextual information at any level of the markup data. As far as visible, this is done to some degree in the CSCP approach, the CC/PP Context Extension approach as well as the PDDL approach. A comprehensive scheme definition is a step towards a high level formality and thus may be used to determine interoperability. Applicability to existing markup-centric infrastructures of an ubiquitous computing environment (e.g. Web Services) is a strength of this kind of context modeling approach.

The strengths of graphical models as described in 3.3 are definitely on the structure level. They are mainly used to describe the structure of contextual knowledge and derive some code (Bauer's approach) or an ER-model (Henricksen's approach) from the model, which is valuable in the sense of the applicability requirement. The distributed composition requirement has some constraints on the structure level, because the merging of model fragments is less efficient than the merging of instance data. Partial validation is possible. Incompleteness and ambiguity seem to be unconsidered by Bauer, but are addressed by Henricksen in a revised version of their model [24]. Most of the extensions made by Henricksen to ORM are quality labels so that quality meta-information may be considered to be intrinsic to that approach. The level of computer evaluable formality is usually relatively low for any graphical model. It is mainly used for human structuring purposes.

Object oriented context modeling approaches (section 3.4) are strong regarding the distributed composition requirement. New types of contextual information (classes) as well as new or updated instances (objects) may be handled in the system in a distributed fashion. Partial validation is possible, typically using a compiler on the structure level and a runtime environment on the instance level. The TEA approach is safe concerning the quality of information requirement, because the concept of cues contains a parameter describing the quality of the cue's output symbol. This is useful to handle incompleteness and ambiguity correctly. A higher level of formality is reached through the use of well-defined interfaces to access the object's content, but the invisibility as consequence of encapsulation is a little drawback concerning the formality requirement. Applicability to existing object oriented ubiquitous computing runtime environments is given, but has usually strong additional requirements on the resources of the computing devices – requirements which often cannot be fulfilled in ubiquitous computing systems.

Logic based context models (see 3.5) may be composed distributed, but partial validation is difficult to maintain. Their level of formality is extremely high, but without partial validation the specification of contextual knowledge within a logic based context model is very error-prone. None of the logic based models within our survey seem to fulfil the quality of information requirement, even if it should be easy to add quality meta-information. Incompleteness and ambigui-

ty seem to be addressed neither. Applicability to existing ubiquitous computing environments seems to be a major issue, because full logic reasoners are usually not available on ubiquitous computing devices.

Due to the similarities between the modeling instruments of ontologies (concepts, facts) and objects (classes, instances), ontology based context modeling approaches described in 3.6 are also strong regarding the distributed composition requirement. Partial validation is possible, and a comprehensive set of validation tools do exist. The ASC model seems to be the only model of the survey enabling not only data type validation, but also full data content validation by specifying ranges for the contextual information called *scales*. All ontology based context models inherit the strengths in the field of normalization and formality from ontologies. The ASC model and the CONON model inherently support quality meta-information and ambiguity, whereas the CoBrA approach seems not to do so, but could be easily extended in that way. Incompleteness is covered by all approaches in a similar way. Applicability to different existing ubiquitous computing environments is reached in the ASC model approach adopting integration elements of CoOL such as scheme extensions. The applicability of the CONON ontologies is without any further integration elements restricted to environments capable of handling OWL-DL for knowledge representation purposes. Due to its broker-centric agent architecture the CoBrA approach is particularly applicable to agent systems.

5. SUMMARY, CONCLUSION AND OUTLOOK

In the previous sections we derived some requirements on a context model which would be optimal for usage in an ubiquitous computing environment. Based on a survey of the most important existing context modeling approaches, we discussed the fulfillment of the different approaches with respect to these requirements. Table 1 summarizes the discussion of the appropriateness of the different context modeling approaches for ubiquitous computing and the identified particular requirements.

Approach - Requirem.	dc	pv	qua	inc	for	app
Key-Value Models	-	-	-	-	-	+
Markup Scheme Mod.	+	++	-	-	+	++
Graphical Models	-	-	+	-	+	+
Object Oriented Mod.	++	+	+	+	+	+
Logic Based Models	++	-	-	-	++	-
Ontology Based Mod.	++	++	+	+	++	+

Table 1: Appropriateness Indication

Due to our analysis we arrived at the conclusion that the most promising assets for context modeling for ubiquitous computing environments with respect to the requirements listed in section 2 can be found in the ontology category. The representatives of this category met the requirements best. However, this does not mean that the other approaches are unsuitable for ubiquitous computing environments.

As with all surveys, the list of context modeling approaches is comprehensive but incomplete. Further emerging approaches should be considered and evaluated in a similar way.

REFERENCES

- [1] Esprit project 26900: Technology for enabled awareness (tea), 1998.
- [2] AKMAN, V., AND SURAV, M. The use of situation theory in context modeling. *Computational Intelligence* 13, 3 (1997), 427–438.
- [3] BACON, J., BATES, J., AND HALLS, D. Location-oriented multimedia. *IEEE Personal Communications* 4, 5 (1997).
- [4] BARWISE, J., AND PERRY, J. *Situations and Attitudes*. MIT Press, 1983.
- [5] BAUER, J. Identification and Modeling of Contexts for Different Information Scenarios in Air Traffic, Mar. 2003. Diplomarbeit.
- [6] BOUZY, B., AND CAZENAVE, T. Using the Object Oriented Paradigm to Model Context in Computer Go. In *Proceedings of Context'97* (Rio, Brazil, 1997).
- [7] BROWN, P. J., BOVEY, J. D., AND CHEN, X. Context-aware Applications: from the Laboratory to the Marketplace. *IEEE Personal Communications* 4, 5 (October 1997), 58–64.
- [8] BUTLER, M. H. CC/PP and UAProf: Issues, improvements and future directions. In *Proceedings of W3C Delivery Context Workshop (DIWS 2002)* (Sophia-Antipolis/France, March 2002).
- [9] CAPRA, L., EMMERICH, W., AND MASCOLO, C. Reflective middleware solutions for context-aware applications, 2001.
- [10] CHEN, G., AND KOTZ, D. A survey of context-aware mobile computing research. Tech. Rep. TR2000-381, Dartmouth, November 2000.
- [11] CHEN, H., FININ, T., AND JOSHI, A. Using OWL in a Pervasive Computing Broker. In *Proceedings of Workshop on Ontologies in Open Agent Systems (AAMAS 2003)* (2003).
- [12] CHEN, P.-S. The entity-relationship model - toward a unified view of data. *ACM Transaction on Database Systems* 1, 1 (March 1976), 9–36.
- [13] CHEVERST, K., MITCHELL, K., AND DAVIES, N. Design of an object model for a context sensitive tourist GUIDE. *Computers and Graphics* 23, 6 (1999), 883–891.
- [14] CHTCHERBINA, E., AND FRANZ, M. Peer-to-peer coordination framework (p2pc): Enabler of mobile ad-hoc networking for medicine, business, and entertainment. In *Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet (SSGRR2003w)* (L'Aquila/Italy, January 2003).

- [15] DE BRUIJN, J. Using Ontologies - Enabling Knowledge Sharing and Reuse on the Semantic Web. Tech. Rep. Technical Report DERI-2003-10-29, Digital Enterprise Research Institute (DERI), Austria, October 2003.
- [16] DEY, A. K. Understanding and using context. *Personal and Ubiquitous Computing, Special issue on Situated Interaction and Ubiquitous Computing* 5, 1 (2001).
- [17] GHIDINI, C., AND GIUNCHIGLIA, F. Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence* 127, 2 (2001), 221–259.
- [18] GIUNCHIGLIA, F. Contextual reasoning. *Epistemologica - Special Issue on I Linguaggi e le Macchine* 16 (1993), 345–364. Also IRST-Technical Report 9211-20, IRST, Trento, Italy.
- [19] GRAY, P., AND SALBER, D. Modelling and Using Sensed Context Information in the design of Interactive Applications. In *LNCS 2254: Proceedings of 8th IFIP International Conference on Engineering for Human-Computer Interaction (EHCI 2001)* (Toronto/Canada, May 2001), M. R. Little and L. Nigay, Eds., Lecture Notes in Computer Science (LNCS), Springer, p. 317 ff.
- [20] GRUBER, T. G. A translation approach to portable ontologies. *Knowledge Acquisition* 5, 2 (1993), 199–220.
- [21] GU, T., WANG, X. H., PUNG, H. K., AND ZHANG, D. Q. Ontology Based Context Modeling and Reasoning using OWL. In *Proceedings of the 2004 Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS2004)* (San Diego, CA, USA, January 2004).
- [22] HALPIN, T. A. *Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design*. Morgan Kaufman Publishers, San Francisco, 2001.
- [23] HELD, A., BUCHHOLZ, S., AND SCHILL, A. Modeling of context information for pervasive computing applications. In *Proceedings of SCI 2002/ISAS 2002* (2002).
- [24] HENRICKSEN, K., AND INDULSKA, J. Modelling and Using Imperfect Context Information. In *Workshop Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communications (PerCom2004)* (Orlando, FL, USA, March 2004), pp. 33–37.
- [25] HENRICKSEN, K., INDULSKA, J., AND RAKOTONIRAINY, A. Modeling context information in pervasive computing systems. In *LNCS 2414: Proceedings of 1st International Conference on Pervasive Computing* (Zurich, Switzerland, 2002), F. Mattern and M. Naghshineh, Eds., Lecture Notes in Computer Science (LNCS), Springer, pp. 167–180.
- [26] HENRICKSEN, K., INDULSKA, J., AND RAKOTONIRAINY, A. Generating Context Management Infrastructure from High-Level Context Models. In *Industrial Track Proceedings of the 4th International Conference on Mobile Data Management (MDM2003)* (Melbourne/Australia, January 2003), pp. 1–6.
- [27] INDULSKA, J., ROBINSONA, R., RAKOTONIRAINY, A., AND HENRICKSEN, K. Experiences in using cc/pp in context-aware systems. In *LNCS 2574: Proceedings of the 4th International Conference on Mobile Data Management (MDM2003)* (Melbourne/Australia, January 2003), M.-S. Chen, P. K. Chrysanthis, M. Sloman, and A. Zaslavsky, Eds., Lecture Notes in Computer Science (LNCS), Springer, pp. 247–261.
- [28] KAGAL, L., KOROLEV, V., CHEN, H., JOSHI, A., AND FININ, T. Project centaurus: A framework for indoor mobile services.
- [29] MCCARTHY, J. Notes on formalizing contexts. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence* (San Mateo, California, 1993), R. Bajcsy, Ed., Morgan Kaufmann, pp. 555–560.
- [30] MCCARTHY, J., AND BUVAČ. Formalizing context (expanded notes). In *Working Papers of the AAAI Fall Symposium on Context in Knowledge Representation and Natural Language* (Menlo Park, California, 1997), S. Buvač and Ł. Iwańska, Eds., American Association for Artificial Intelligence, American Association for Artificial Intelligence, pp. 99–135.
- [31] ÖTZTÜRK, P., AND AAMODT, A. Towards a model of context for case-based diagnostic problem solving. In *Context-97; Proceedings of the interdisciplinary conference on modeling and using context* (Rio de Janeiro, February 1997), pp. 198–208.
- [32] PASCOE, J. Adding Generic Contextual Capabilities to Wearable Computers. In *2nd International Symposium on Wearable Computers (ISWC 1998)* (1998), pp. 92–99.
- [33] RYAN, N. ConteXtML: Exchanging Contextual Information between a Mobile Client and the FieldNote Server, August 1999.
- [34] SAMULOWITZ, M., MICHAHELLES, F., AND LINNHOF-POPIEN, C. Capeus: An architecture for context-aware selection and execution of services. In *New developments in distributed applications and interoperable systems* (Krakow, Poland, September 17-19 2001), Kluwer Academic Publishers, pp. 23–39.
- [35] SCHILIT, B. N., ADAMS, N. L., AND WANT, R. Context-aware computing applications. In *IEEE Workshop on Mobile Computing Systems and Applications* (Santa Cruz, CA, US, 1994).

- [36] SCHILIT, W. N. *A System Architecture for Context-Aware Mobile Computing*. PhD thesis, Columbia University, 1995.
- [37] SCHMIDT, A., BEIGL, M., AND GELLERSEN, H.-W. There is more to context than location. *Computers and Graphics* 23, 6 (1999), 893–901.
- [38] SCHMIDT, A., AND LAERHOVEN, K. V. How to Build Smart Appliances. *IEEE Personal Communications* (August 2001).
- [39] STRANG, T. *Service Interoperability in Ubiquitous Computing Environments*. PhD thesis, Ludwig-Maximilians-University Munich, Oct. 2003.
- [40] STRANG, T., LINNHOFF-POPIEN, C., AND FRANK, K. Applications of a Context Ontology Language. In *Proceedings of International Conference on Software, Telecommunications and Computer Networks (SoftCom2003)* (Split/Croatia, Venice/Italy, Ancona/Italy, Dubrovnik/Croatia, October 2003), D. Begusic and N. Rozic, Eds., Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Croatia, pp. 14–18.
- [41] STRANG, T., LINNHOFF-POPIEN, C., AND FRANK, K. CoOL: A Context Ontology Language to enable Contextual Interoperability. In *LNCS 2893: Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS2003)* (Paris/France, November 2003), J.-B. Stefani, I. Dameure, and D. Hagimont, Eds., vol. 2893 of *Lecture Notes in Computer Science (LNCS)*, Springer Verlag, pp. 236–247.
- [42] STRANG, T., LINNHOFF-POPIEN, C., AND ROECKL, M. Highlevel Service Handover through a Contextual Framework. In *Proceedings of 8th International Workshop on Mobile Multimedia Communications (MoMuC2003)* (Munich/Germany, October 2003), J. Kaefer and M. Zuendt, Eds., vol. 1, Center for Digital Technology and Management (CDTM), pp. 405–410.
- [43] USCHOLD, M., AND GRÜNINGER, M. Ontologies: Principles, methods, and applications. *Knowledge Engineering Review* 11, 2 (1996), 93–155.
- [44] W3C. Composite Capabilities / Preferences Profile (CC/PP). <http://www.w3.org/Mobile/CCPP>.
- [45] WANG, X. H., ZHANG, D. Q., GU, T., AND PUNG, H. K. Ontology Based Context Modeling and Reasoning using OWL. In *Workshop Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communications (PerCom2004)* (Orlando, FL, USA, March 2004), pp. 18–22.
- [46] WAPFORUM. User Agent Profile (UAPProf). <http://www.wapforum.org>.