# Ontology-Based Context Modeling

Reto Krummenacher, Thomas Strang

Digital Enterprise Research Institute, University of Innsbruck, Austria

**Abstract.** Ontologies are a widely accepted tool for the modeling of context information. We view the identification of the benefits and challenges of ontology-based models to be an important next step to further improve the usability of ontologies in context-aware applications. We outline a set of criteria with respect to ontology engineering and context modeling and discuss some recent achievements in the area of ontology-based context modeling in order to determine the important next steps necessary to fully exploit ontologies in pervasive computing.

## 1 Introduction

Ontologies – explicit formal specifications of the terms in a domain and the relations among them [17] – are widely accepted as instrument for the modeling of context information in pervasive computing applications. On the one hand its advantages compared to other traditional modeling approaches were recognized [32], while on the other the Semantic Web languages and tools have clearly gained maturity over the past years. In order to further improve the usability of ontologies for context-aware applications it is important to analyze the benefits and challenges. We outline a set of criteria with respect to context modeling and ontology engineering. The evaluation criteria shall help to investigate the recent achievements in the area of ontology-based context modeling and the same time depict the true benefits of ontology-based systems. This is considered to be an important step to fully exploit ontologies in pervasive computing.

In order to discuss the strengths and weaknesses of context models it is important to understand what context is and what role context and context modelling play in current systems. A widely accepted definition of context is [8]:

> *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 the application themselves.*

A system is then seen to be context-aware if it uses context to provide relevant information and services to the user, where relevancy depends on the users task. Given the increasing number of services and agents on the Internet and their self-determinism we argue that the definition should be extended to include the interaction between machines too (without human in the loop). Context-aware discovery, composition and negotiation of information and services are the dominant parameters of future systems [20, 31]. This additionally emphasizes the requirement for interoperability and machine-understandability of context information. Hence, it is not surprising that [6] calls ontologies a key requirement for the realization of the pervasive computing vision.

Traditionally context models are created top down: first the application and its functionality is defined, and then the necessary context ontologies developed. The ontologies are often only used to formalize taxonomies that represent the values and types of simple properties [4]. There is however clearly more to ontology-based modelling. A generic, reusable ontology has direct impact on the interoperability of context-aware systems and hence directly influences the speed of creation, integration and implementation of new applications; a well-designed model is a key accessor to context [32].

The paper is structured as follows: In Section 2 the advantages and challenges of ontology-based context modeling are discussed. Section 3 introduces a set of evaluation criteria taking into account ontology engineering and context modeling aspects. They provide the key factors to fully exploit the advantages of ontologies in context modeling. In Section 4 an overview of recent achievements in the concerned field is given and discussed with respect to the evaluation criteria. Section 5 concludes the paper.

## 2 Advantages and challenges

In [32] the authors concluded that the most promising assets for context modeling can be found in the ontology-based models. These representatives met the six requirements dominant in pervasive environments best: (1) distributed composition, (2) partial validation, (3) richness and quality of information, (4) incompleteness and ambiguity, (5) level of formality, and (6) applicability to existing environments. They clearly outperformed the analyzed key-value, markup scheme, graphical, logic-based, and object oriented models. In the meantime the use of ontologies for the formalization of context models has received even wider recognition and its advantages are acknowledged. Ontologies play a pivotal role not only for the Semantic Web, but in particular also in pervasive computing and next generation mobile communication systems [9, 12]. This is underpinned by the strong research movement and ongoing standardization efforts in the area of the Semantic Web that drives the languages and tools to more maturity.

Ontologies are a powerful tool to specify concepts and interrelationships. They provide formalizations to project real-life entities onto machine-understandable data constructs. In that way, ontologies provide a uniform way for specifying the model's concepts, subconcepts, relations, properties and facts, altogether providing the means for the sharing of contextual knowledge and information reuse. The contextual knowledge is interpreted and evaluated by use of ontology reasoning. This allows computers to determine the contextual compatibility, to compare contextual facts and to infer new and more complex context from core measurements, also counteracting the common problem of incompleteness and ambiguity.

These are core advantages of ontologies, however do not clearly differentiate the advantage over other approaches. Object-oriented models provide hierarchical class layering too, and hence allow for at least limited formalization of class and instance dependency models.[1] Therefore it is necessary to dig further into the needs of pervasive computing applications. The conclusions leading to improved user experience are generally based on data coming from diverse sets of information sources and sensors.

[1] [32] indicates that object-oriented models are the second most appropriate technique.

Taking the emerging globalism of applications and the increasing importance of the Web and Web services into account, it seems obvious that pervasive computing environments must more and more address the problem of data heterogeneity in the large and not on a per application basis. The same counts for the problems of ambiguity, quality and validity of contextual data. The formalization of concept and property dependencies allows inference engines to (partially) validate the models, as well as instance derivations. This clearly counteracts the aforementioned problems and emphasizes the fact that ontological models surpass the other approaches. Object-oriented models for example require low-level implementation agreement between applications to ensure interoperability and are thus not suited for knowledge sharing in open and dynamic environments [6].

The big challenge remains the right usage of the ontology tools and languages introduced below (Section 2.1). Formalizing data structures and terms, consistency checking, data mediation provide the desired measures to address the heterogeneity, ambiguity and quality-related issues. However, inappropriate use or miss-modeling may limit the impact and use of the context modeling ontology.

### 2.1 Ontologies: languages and functionalities

In principle we can differentiate two branches of formal languages to specify ontologies: First-Order Logic (FOL, [11]) and Logic Programming (LP, [24]). A prominent subset of FOL is referred to as Description Logics (DL [1]) with its standardized syntax OWL (Web Ontology Language, [26]). The strength of DL lies in subsumption reasoning and consistency checking and is often applied to classification tasks, i.e. for building taxonomies according to concept and relation definitions. For these tasks there are mature reasoners available, while there is a lack of support for efficient instance retrieval. Moreover, if the DL descriptions are combined with rules the knowledge base quickly enters full FOL and becomes undecidable. LP on the other hand is often used in tasks like query answering and consequence finding. This leads to the definition of rules, where a rule consists of a head and a body: $child(x) \leftarrow human(x) \wedge age(x) \leq 16$. This example rule declares a human with an age under 16 to be a child.

An important fact is that FOL usually adheres to the open world assumption, while LP is based on the closed world assumption and inherits a non-monotonic logic, i.e. the set of conclusions warranted on the basis of a given knowledge base does not increase (in fact, it can shrink) with the size of the knowledge base itself. This is in clear contrast to classical FOL, whose inferences can never be undone by new information. This contradicting traits make the integration of DL and LP problematic and is subject to ongoing research.

## 3 Evaluation criteria

In this section we present a set of evaluation criteria that not only consider the important features of context models, but that look at critical ontology engineering aspects too. The former are partially based on [32]. The ontology specific aspects were considered in the latter in order to evaluate the quality of the ontological support. These criteria can of course be generalized and are not context ontology engineering specific. We expect these success factors and guidelines to help improving the development of future ontology-based context models.

### 3.1 Context modeling criteria

The following criteria constitute the first part and consider aspects like uncertainty and quality of data, traceability and comparability of information and the applicability of the model. The criteria are constructed around questions that allow a more detailed look at the ways the context models address the respective issues.

**Applicability:** Traditionally the definition of models is conducted on a per task basis, hence for a given problem a respective model is developed. Context information results however from and is applied to very heterogeneous systems of devices and applications. A model that serves as a context encoding infrastructure should be flexible from an implementation point of view. This criterion considers the usability and applicability of the context model within existing infrastructures and various application domains. Does the model in any way restrict the domain of application?

**Comparability:** Context information is generally provided by a multitude of sensors and devices. Different measuring and coding systems used by different manufacturer result in a heterogeneous set of values describing the same entities. Hence, it is necessary to provide means to compare values with different units and encodings. Moreover high-order context often consists of non-countable values without obvious ordering. The question to answer is thus the means that a model provides in order to support comparability of diverse and non-countable information.

**Traceability:** In order to provide adequate control and interpretation of contextual information, it is necessary to determine the provenance and undergone manipulations of data. This becomes particularly eminent when using calculated context, where it is highly important to know the derivation rules applied. The interpretation of 'warm' is impossible, if the rule '$warm = temp > 21\,^{\circ}\mathrm{C}$' is not known. Thus, it is necessary to investigate to which extent the model, and how, provides means to record provenance and processing information.

**History, logging:** Often decisions depend on past events and facts. It is hence necessary to support the logging of past information. This allows moreover to trace the evolution of states and measures. Furthermore, logging is closely related to timestamping, an important tool for versioning of information. Comparing information on the basis of time provides tools to address ambiguity of contextual data: when sensor failures are detected or assumed past measures can fill the gap, or inconsistencies can be detected based on sudden and substantial changes. Does the model, and in which ways, address the issue of data logging and history records?

**Quality:** The quality of information delivered by a sensor varies over time. Is quality of information an issue that is directly integrated into the model? Are there means available to model precision, resolution or richness, possibly depending on the source of information?

**Satisfiability:** While quality is about the trustworthiness of accepted information, satisfiability deals with the conformance of measured or derived information to the defined model. A model should define the range a context value can take, or define a particular co-existence of values to be impossible [23]. Does the model adhere means to check the satisfiability of information context instances?

**Inference:** Low-order context is generally produced by sensors and combined in order to establish high-order context like situations, activities or procedures. Does the context model know, as integral part of it, a conceptualization of an derivation mechanisms or means to define high-order context types? In other words, are there tools defined that permit the definition of new contextual categories and facts on the basis of low-order context?

**Incompleteness** and **ambiguity** are also stated as critical in [32]: Sensor networks and mobile devices are connected in unstable and often unreliable networks. Thus the contextual information available at any point in time is usually non-deterministic. The ontologies and the value ranges defined therein provide means to address these issues by restricting the arbitrariness of contextual data. Moreover, the support for traceability, satisfiability and logging helps to detect and counteract system-related deficiencies.

### 3.2 Ontology engineering criteria

The second set of criteria is used to evaluate the ontologies and are influenced by [16, 18]. The considered facts touch issues like flexibility, extensibility and completeness of the ontology, consistency and granularity of the concepts and properties, as well as the language formalism applied.

**Reusability, standardization:** Applicability covered previously is different from ontology reusability. Increasing the reusability implies the maximization of using the ontology among several independent tasks, while usability rather means the maximization of applications using the ontology for the same or similar tasks. To what extend does the ontology allow reusability in other independent modeling tasks?

**Flexibility, extensibility:** This criterion refers to the possibility of adding new definitions to the ontology without altering the existing dependencies. How much effort and changes are needed to extend the ontology model? Does the model allow flexible and low-cost adjustments with respect to given applications?

**Genericity:** Ontologies are about the integration of knowledge and the relationship of resources. It is important that a generic and multi-functional backbone is provided for the modeling of information. Ontologies that are applicable across large sets of domains are referred to as upper ontologies and belong to the most general category of ontologies. Does the context ontology restrict the application domain? Does it provide an upper ontology for context modeling?

**Granularity:** This criterion highly related to the details of the concepts defined and the scope of their meaning. A fine-grained ontology defines concepts for closely related objects, while in contrast a coarse-grained model knows more general and distinguishable terms. Insofar granularity is related to the diversity and coverage of individual concepts. Upper ontologies are often coarse-grained, while application ontologies become more fine-grained.

**Consistency:** This criterion is about the existence of explicit or implicit contradictions in the represented ontological content. A good approach for a methodical evaluation of the criterion is presented in [18].

**Completeness:** According to [16] an ontology is complete if it (explicitly or implicitly) covers the intended domain. A ontology can thus be complete without covering all possible aspects, if its target domain is restrictive to some particular world. Does the ontology cover all relevant concepts, properties? Can the entities and their interactions be modeled?

**Redundancy:** The redundancy criterion investigates the existence of superfluous repetitions or overlapping definitions. Redundancy errors occur by explicit redefinition or by inference of information through other existing definitions. Short, redundancy is caused by the definition of two or more concepts or instances with the same formal definition, but different names [16].

**Readability:** This measure accounts the usage of intuitive labels to denominate the ontological entities. The importance of this criterion is limited to the understandability of humans and is hence of lower importance to context-aware computing.

**Scalability:** Ontology engineers distinguish three types of scalability: cognitive scalability which refers to the possibilities of humans to oversee and understand the ontology, engineering scalability which refers to the available tool support that is still quite limited for large scale ontologies, and reasoning scalability which refers to the difficulties of reasoning with large data sets.

**Language, formalism:** This criterion looks at the language used to model the ontology and the expressivity thereof (cf. Section 2.1). Possible languages are standard First-Order Logic and subsets thereof like Description Logic, as well as non-monotonic rule languages like investigated in Logic Programming. UML-based languages are excluded, as they do not have a model theoretic semantics.

In the next section we discuss an evaluation process and look at some of the criteria by means of an analysis of ontology-based context models.

## 4 Ontology-based context modeling: a first analysis

Analogous to the two categories of evaluation criteria there are two steps in developing or evaluating a context modeling ontology: the context model and its features, and the ontology development. We first consider the desired features of the context model.

A core requirement is definitively the flexibility of the model with respect to the application domain; this is mostly achieved by defining a generic core model that allows the definition of arbitrary context types and values. In ConOnto [20] a root concept *ContextView* provides an organizational reference point for declaring context information. Relevant entities are then described by at least one ContextView which itself is bound to *ContextFeature*s and *ContextEngagement*s. A similar approach comes from CoOL [33], where entities are characterized by *ContextInformation* instances which are defined and interlinked by use of the aspect-scale-context (ASC) model. ASC provides an umbrella vocabulary to transfer arbitrary context models and is therefore a strong approach with respect to the comparability criterion. In fact, relating different scales for the same context aspects and deriving and aggregating new scales from existing ones is one of the motivations for this ontology. mySAM [2] on the other hand introduces a model to define arbitrary context predicates.

Other ontologies address the genericity issue by means of upper ontologies. The most renowned representative is SOUPA, a very complete family of ontologies [6]. CONON defines 14 core classes to model *Person*, *Location*, *Activity* and *Computational Entities* [35]. CoDAMoS points in the same direction by defining an ontology around four concepts used to model *User*s, the *Environment*, *Platform*s and *Service*s [28]. Both approaches focus on the modeling of profiles for human users and applications, and might be limited with respect to future context-awareness tasks in service-service interaction models. Resembling ontologies were presented by [7, 9] in order to model devices, services and users; the latter targets the telecommunication industry. These approaches are to a big extent formalized markup scheme models (cf. [32]).

In summary, there are many models that satisfy the applicability and genericity criteria. We thus concentrate on the other criteria. With respect to comparability we already pointed to ASC-CoOL. Similar ideas were applied in [14], a model heavily influenced by the former. While quality of data is integrated by most models, traceability, recording of past data, and satisfiability are still too often neglected. Quality is bound to the model by means of quality classes [14, 35] or dedicated attributes: *quality*, *meanError* [33], *confidence* [22, 29] or *probability* [19, 21]. Some approaches moreover incorporate fuzzy inference or bayesian reasoning [20, 29]. Traceability is to our knowledge only explicitly addressed by CONON and [22]; The former uses *classifiedAs* to indicated the provenance of information: sensed, derived, aggregated, or deduced, while the latter knows the *source* attribute. Data logging is mostly provided by use of timestamps. GAIA makes moreover use of an external database for temporal queries [29], while [5] proposes to integrate a temporal vector space. Satisfiability at last is directly incorporated in ASC in form of so-called *memberCheck* operations. This is of particular interest when verifying non-countable values.

The context modeling criteria considered in the first step help to put in place and analyze the relevant terms and features. The ontology engineering process takes those terms as starting point to determine the required concepts, attributes and relations.

Many of the considered ontologies are reusing existing vocabularies, which increases the interoperability. Very popular is the agent description vocabulary FOAF.[2] Then again, only SOUPA seems to be regularly reused by other projects. This is certainly due to the fact that SOUPA is written in a very modular way by combining sub-ontologies for time, location, policies, persons (FOAF) and the MoGATU BDI ontology [27]. This enables the partial reuse of the context ontology, which makes integration much easier. Similar modularizations are at the basis of CoDAMoS that is built around four core concepts and CaMiDO which has chosen a 3-tier model: middleware, context, and application [3].

The reuse and extension of existing ontologies must be the general objective. Building whole ontologies from scratch has indeed two clear disadvantages: 1) it requires potentially large overhead in engineering, 2) it obviously decreases the interoperability with existing approaches [34].

Eventually, we shortly address the applied formalisms. OWL-DL seems to be a natural choice to model ontologies for its ensured decidability and as it is becoming a standard through W3C recommendation. CONON uses a straightforward extension

[2] http://www.foaf-project.org

to FOL to integrate user rules, while GAIA claims to use FOL [30]. Details of the implementation suggest however that they are using LP reasoning combined with DL reasoning in separate tasks. Also in [23] we find a DL+rules algorithm. Such approaches are however known to be problematic. LP was applied by [33].

A minority of the context modeling approaches rely on the less expressive Resource Description Framework (RDF, [25]). Noteworthy are the context rules approach of CAPNET [13] and the Context Description Framework (CDF, [21]). CAPNET defines RDF-based rules that have two properties: *Action* and *Condition*. The predicate in the *Action* statement indicates what shall be carried out if the condition statements are satisfied. CDF is a logic extension to RDF: it adds a TrueInContext statement to every RDF triple and considers contextual values as a container of RDF statements. Moreover CDF defines vocabularies to model significance and probability of truth.

**Table 1.** Summary of Initial Analysis

| **Criteria** | | **Criteria** | |
|---|---|---|---|
| **Genericity** | ASC, CAPNET, CDF, ConOnto, mySAM, (SOUPA) | **Quality** | ASC, CDF, CMF-VTT, CONON, GAIA, SCAFOS |
| **Traceability** | CMF-VTT, CONON | **Satisfiability** | ASC |
| **History** | ASC, CMF-VTT, GAIA | **Comparability** | ASC |

An alternative approach not coming from pervasive computing is the description and situation plug-in to DOLCE (DnS, [15]). DnS provides domain independent concepts and relations derived from linguistics, philosophy and mathematics and aims at the modeling of descriptions, situations and roles.

## 5 Conclusion

In this paper we set up tools to further improve the use of ontologies in pervasive computing. We are convinced that a state of the art analysis and a solid list of success factors are necessary to further improve ontology-based context models. The main contribution is a set of context modelling and ontology engineering criteria that shall help to evaluate existing approaches and more importantly that shall serve as support for future deployments. The paper does however not address the evaluation of context-aware systems and architectures. Furthermore we shortly considered some of the current achievements in the domain, together with a short description of an evaluation process. The mentioned approaches provide an overview to give an idea of the current contributions in the domain. The list is however – unfortunately – by no means complete. In fact it seems to be a general problem that the ontologies are not publicly available. The interoperability and applicability of ontologies depends not only on the success factors, but certainly on their distribution. A few good counter examples to this unfortunate trend are SOUPA[3] and ConOnto.[4]

[3] http://pervasive.semanticweb.org/soupa-2004-06.html
[4] http://www.site.uottawa.ca/˜mkhedr/contexto.html

Ontologies on the large should be the general aim. However, due to the clear scalability issues with current technologies, the reasoning tasks will continue for a while to be forced to be application domain specific, and thus reasoning will be based on application domain ontologies [10]. Keeping this current technological limitation in mind, we would like to emphasize again on the importance of good ontology engineering in order to ensure interoperability. With this paper we intend to initiate an context model integration process by highlighting the important features and critical aspects of ontology-based context modeling. Only if there is a solid set of well-designed context ontologies available full integration of contextual data can be guaranteed – a prerequisite for future context-aware applications. Moreover, integration and cross-fertilization efforts conducted today avoid the reestablishment and reduplication of modeling work once the reasoning technologies caught up with the requirements and dimensions of large scale mobile and pervasive systems.

## References


1. F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider. *The Description Logic Handbook*. Cambridge University Press, 2003.
2. Ph. Beaune, O. Boissier, and O. Bucur. Representing Context in an Agent Architecture for Context-Based Decision Making. In *Satellite Workshop on Context Representation and Reasoning*, Jul 2005.
3. N. Belhanafi, Ch. Taconet, and G. Bernard. CAMidO, A Context-Aware Middleware Based on Ontology Meta-Model. In *Workshop on Context Awareness for Proactive Systems*, pp. 93–103, Jun 2005.
4. O. Bucur, Ph. Beaune, and O. Boissier. Representing Context in an Agent Architecture for Context-Based Decision Making. In *Satellite Workshop on Context Representation and Reasoning*, Jul 2005.
5. S. Chan and Q. Jin. Enhancing Ontology-based Context Modeling with Temporal Vector Space for Ubiquitous Intelligence. In *20th Int'l Conf. on Advanced Information Networking and Applications*, pp. 669–674, Apr 2006.
6. H. Chen, F. Perich, T. Finin, and A. Joshi. SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In *1st Annual Int'l Conf. on Mobile and Ubiquitous Systems: Networking and Services*, Aug 2004.
7. E. Christopoulou and A. Kameas. GAS ontology: an ontology for collaboration among ubiquitous computing devices. *Int'l Journal of Human-Computer Studies*, 62(5):664–685, May 2005.
8. A.K. Dey. Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1):4–7, Feb 2001.
9. A.V. Zhdanova (ed.). Ontology Definition for the DCS and DCS Resource Description, User Rules. SPICE Project Deliverable D3.1, Sep 2006.
10. D. Fensel and F. van Harmelen. Unifying reasoning and search into something that scales up to frillions of triples. Technical Report DERI-TR-2007-01-11, Digital Enterprise Research Institute (DERI), Jan 2007.
11. M. Fitting. *First-order logic and automated theorem proving*. Springer, 1990.
12. P. Floréen, M. Przybilski, P. Nurmi, J. Koolwaaij, A. Tarlano, M. Wagner, M. Luther, F. Bataille, M. Boussard, B. Mrohs, and S.L. Lau. Towards a Context Management Framework for MobiLife. In *IST Mobile & Wireless Communications Summit*, Jun 2005.

13. J. Forstadius, O. Lassila, and T. Seppanen. RDF-Based Model for Context-Aware Reasoning in Rich Service Environment. In *Context Modeling and Reasoning Workshop*, pp. 15–19, Mar 2005.
14. F. Fuchs, I. Hochstatter, and M. Krause. A Metamodel Approach to Context Information. In *Context Modeling and Reasoning Workshop*, pp. 8–14, Mar 2005.
15. A. Gangemi and P. Mika. Understanding the Semantic Web through Descriptions and Situations. In *1st Int'l Conf. on Ontologies, Databases and Applications of SEmantics*, Nov 2003.
16. A. Gómez-Pérez. Evaluation of ontologies. *Int'l Journal of Intelligent Systems*, 16(3):391–409, Mar 2001.
17. T.R. Gruber. A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition*, 5(2):199–220, Jun 1993.
18. N. Guarino and C. Welty. Evaluating Ontological Decisions with Ontoclean. *Communications of the ACM*, 45(2):61–65, Feb 2002.
19. E. Katsiri and A. Mycroft. A first-order logic model for context-awareness in distributed sensor-driven systems. In *1st Int'l Workshop on Requirements and Solutions for Pervasive Software Infrastructures*, May 2006.
20. M. Khedr and A. Karmouch. Negotiation Context Information in Context-Aware Systems. *IEEE Intelligent Systems*, 19(6):21– 29, Nov/Dez 2004.
21. O. Khriyenko and V. Terziyan. Context Description Framework for the Semantic Web. In *Context Representation and Reasoning Workshop*, Jul 2005.
22. P. Korpipää, J. Mäntyjärvi, J. Kela, H. Keränen, and E.-J. Malm. Managing Context Information in Mobile Devices. *IEEE Pervasive Computing*, 2(3):42–51, Jul/Sep 2003.
23. O. Lassila and D. Khushraj. Contextualizing Applications via Semantic Middleware. In *2nd Ann. Int'l Conf. on Mobile and Ubiquitous Systems: Networking and Services*, Jul 2005.
24. J.W. Lloyd. *Foundations of Logic Programming*. Springer, 1987.
25. F. Manola and E. Miller. RDF Primer. W3C Recommendation, Feb 2004.
26. D.L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. W3C Recommendation, Feb 2004.
27. F. Perich. MoGATU BDI Ontology, Jan 2004.
28. D. Preuveneers, J. v.d.Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, and K. De Bosschere. Towards an Extensible Context Ontology for Ambient Intelligence. In *2nd European Symposium on Ambient Intelligence*, Nov 2004.
29. A. Ranganathan, J. Al-Muhtadi, and R.H. Campbell. Reasoning with Uncertain Context in Pervasive Computing Environments. *IEEE Pervasive Computing*, 3(2):62–70, Apr/Jun 2004.
30. A. Ranganathan and R.H. Campbell. An infrastructure for context-awareness based on first order logic. *Personal and Ubiquitous Computing*, 7(6):353–364, Dec 2003.
31. Th. Strang. *Service-Interoperabilität in Ubiquitous Computing Umgebungen*. PhD thesis, Ludwig-Maximilians-Universität München, Oct 2003.
32. Th. Strang and C. Linnhoff-Popien. A Context Modeling Survey. In *1st Int'l Workshop on Advanced Context Modelling, Reasoning and Management*, pp. 34–41, Sep 2004.
33. Th. Strang, C. Linnhoff-Popien, and K. Frank. CoOL: A Context Ontology Language to enable Contextual Interoperability. In *4th Int'l Conf. on Distributed Applications and Interoperable Systems*, pp. 236–247, Nov 2003.
34. X. Wang, J.S. Dong, C.Y. Chin, S.R. Hettiarachchi, and D. Zhang. Semantic Space: An Infrastructure for Smart Spaces. *IEEE Pervasive Computing*, 3(3):32–39, Jul/Sep 2004.
35. X.H. Wang, D.Q. Zhang, T. Gu, and H.K. Pung. Ontology-Based Context Modeling and Reasoning using OWL. In *Context Modeling and Reasoning Workshop*, pp. 18–22, Mar 2004.