5th SASTech
Iran | Mashhad
May 12 - 17 | 2011
5th Symposium on Advances in Science & Technology

A Comparative Study on Context Modeling Approaches

F. Sahafipour[1], R. Javidan[2], S. Jafari[3], M. Kadivar[4]

[1]Department of Computer Engineering, Islamic Azad University – Arak Branch, Iran
[2]Assisstant professor, Islamic Azad University – Beyza Branch, Iran
[3]Depatment of Computer Engineering, Islamic Azad University– Mashhad Branch, Iran
[4]Assisstant professor, Shahrekord University, Shahrekord, Iran

Paper Reference Number: 20
Name of the Presenter: Foroogh Sahafipour

Abstract

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 applications themselves. Context-awareness is one of the drivers of the pervasive computing paradigm, whereas a well designed model is a key component to the context in any context-aware system. There is an inherent gap between the real-world and the world that can be perceived by computer systems, yielding uncertainty and ambiguity in system perceived context, with consequent effect on the performance of context-aware systems. A primary goal of context models is to enable context-awareness by performing some type of reasoning. Modelling context can be interpreted as a process of building a representation that supposedly embodies occurrences of real-life situations that can be reasoned about. The pervasive computing community increasingly understands that developing context-aware applications should be supported by adequate context information modeling techniques. These techniques reduce the complexity of context-aware applications. In this paper the requirements of context modeling is discussed. It is provides discussion on the most relevant current approaches to modeling context for pervasive computing. This discussion is followed by a comparison of current context modeling techniques.

Key words: Context, Context Awareness, Modeling, Pervasive Computing

1. Introduction

Recent advances in emerging computing technologies and communications evolved into ample pioneering initiatives, leading towards a world in which computing systems are distributed, mobile, intelligent and cooperative. Congruent with the evolution of computing systems is a new paradigm, often termed *pervasive* or *ubiquitous* computing, which leverages on the use of devices that can carry out computing in a relatively non-intrusive manner and ultimately support many aspects of work and everyday activities. Pervasive or ubiquitous computing has often been perceived as the next step in the evolution of distributed systems, in which computing components are not only distributed and mobile but also embedded in a large number of commonplace devices. Ubiquity of computing devices facilitates non-traditional uses and concerns research from a variety of disciplines, including distributed and

mobile computing, artificial intelligence, computer supported cooperative work, human-computer interaction, and more. Central to the notion of pervasive systems is the ability to become context-aware. In the context of pervasive environments, context-aware systems are those which can respond (possibly intelligently) to context information acquired by any type of sensor (either physical or virtual). This, in turn, enhances services provided to users (including service personalization), makes pervasive systems intelligent by reacting (and possibly pro-acting) to changing circumstances, and promotes adaptability and autonomy of systems, liberating users from avoidable interactions. Context-awareness in pervasive computing systems poses a major research challenge concerning a variety of issues originating from the underlying requirements of pervasive environments. To address challenges in context-aware computing, as a step towards realization of the vision of pervasive computing, research has started to investigate different aspects of context, including gathering, discovering, modeling, reasoning and managing context (Padovitz (2006)).
In this paper, a comparative study on context modeling aproachs is provided. In section 2 the concept of context is defined. In Section 3 context modeling requirements and approaches are discussed. Analysis of the paper is provided in section 4.

2. Defining Context in the Scope of Pervasive Computing

The term context is loaded with a wide variety of meanings. Various areas of computer science differ in their understandings of context, but even in the research community working on context-aware adaptive applications there is no consensus. Dey and Abowd (1999) present a survey of alternative views of context, typically defining context by synonym or example. Common examples of context are location, time, temperature, noise level, user activity, and a plethora of information related to the computing environment, including computing devices and their characteristics, network connectivity, communication bandwidth and so on.
Dey and Abowd also offers the following definition, which is now widely (but not universally) accepted in the field: 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 applications themselves.

3. Context Modeling

Context modeling aims to produce a formal or semi-formal description of the context information present in a context-aware system. By providing a uniform description of types of context information, as well as run-time instantiations of the types (context facts), context information from various sources can be easily combined, queried, and reasoned about. This promotes sharing and exchange of information between applications, and provides information representations that are straightforward for applications to process compared to other formats such as streams of raw sensor output.

3.1. Requirements for Context Modeling

In order to be broadly applicable to a variety of context-aware applications and to serve as useful abstractions for software engineering, context modeling approaches should meet the following requirements.

3.1.1. Support for Imperfect Context Information

A common problem in context-aware systems is the presence of imperfect context information. For example, problems with sensor-derived information can arise as a result of sensor failures, power shortage, noise in the environment, faulty sensor installation, or inaccuracy in the algorithms used to abstract context information from sensor outputs. Similarly, user- or application-supplied information can be subject to problems such as staleness.

Consequently, when modeling context, it is necessary to be able to represent:

- information that is incomplete;
- information that is imprecise;
- information that is ambiguous (for example, conflicting location reports provided by different location sensors);
- quality indicators (information source, timeliness, granularity, sensor accuracy, etc.) for information that may be imprecise or erroneous.

3.1.2. Support for Context Histories

Context-aware applications often require not only information about the current context, but also past or future contexts. Therefore, context modeling techniques must provide natural ways for modeling histories of information, and applications should be capable of querying and reasoning over these histories. Histories can be used to detect patterns in user behavior and predict future requirements.

3.1.3. Support for Software Engineering

A key role of context models is to simplify and introduce greater structure into the task of developing context-aware applications. The greatest benefit is often derived when a formal or semi-formal context model is introduced early in the software engineering life-cycle and refined incrementally over the life-cycle.

3.1.4. Support for Run-Time Querying and Reasoning

One of the most important forms of context model is the run-time model. Context models used for analysis and design purposes, run-time models must address representational issues - that is, how to represent information at run-time so that it can be efficiently stored in a repository, queried, and reasoned over to support decision making by context aware applications about how to react to context changes.

3.1.5. Support for Interoperability

Context-aware applications may be required to cooperate at run-time with components that were not known to the application designer, such as new applications or sensing hardware. If a given pair of modeling approaches differ in terms of their expressive power, complete transfer of information between them may not be possible. An important way to address interoperability is through standardization.

3.2. Context modeling approaches

In this section, the variety of context modeling techniques that are in use today is discussed. Then, there is an analysis about them.

3.2.1. Markup scheme approaches

One of the earliest modeling approaches built on the popularity of markup schemes such as XML (Bray et al., 2004). A well-known example that typifies this approach is CC/PP

(Composite Capability/Preference Profiles) (Klyne et al., 2004), which was standardized as a W3C recommendation during 1998-2004. CC/PP aims to support the transfer of simple context information and preferences - such as device characteristics and users' language preferences - from Web browsers to servers, in order to support dynamic adaptation of the Web pages returned by the servers to browsers. CC/PP supports little in the way of reasoning, as it is traditionally used only for representing simple types of information about which reasoning is not necessary. An example of an XML-encoded CC/PP profile is shown in Fig. 1.

```
<ccpp:component>
 <rdf:Description
  rdf:about="http://www.example.com/profile#TerminalHardware">
  <rdf:type
   rdf:resource="http://www.example.com/schema#HardwarePlatform" />
   <ex:displayWidth>320</ex:displayWidth>
   <ex:displayHeight>200</ex:displayHeight>
 </rdf:Description>
</ccpp:component>
```

Fig. 1: A mark up scheme model, CC/PP example.

In addition, a similar RDF-based proposal called Comprehensive Structured Context Profiles (CSCP) (Buchholz et al., 2004) was developed to provide greater expressive power than CC/PP. CSCP lifts some of the limitations imposed by CC/PP on the structure of the profiles, bringing the expressiveness closer to the original expressive power of RDF.

3.2.2. Ontology-based approaches

This section addresses context modeling approaches that are aligned with recent work in ontology language standardization - specifically, OWL (McGuinness and van Harmelen, 2004) and closely related precursors such as DAML+OIL (Horrocks, 2002). These modeling approaches are more sophisticated than the markup scheme approaches described in the previous section, in that they support additional concepts, such as set operators for defining classes (union, intersection, etc.), cardinality constraints on properties, and equivalence between pairs of classes or properties. Importantly, they are also based on logical formalisms that support reasoning. Like the markup scheme approaches, the ontology-based approaches focus on run-time context modeling, not software engineering issues like analysis of required context types, their quality, and other similar characteristics. One of the earliest ontology-based proposals, by Strang et al. (2003), was the Context Ontology Language (CoOL). CoOL differs from other work in ontology-based context modeling in that it introduces core context modeling constructs (aspects, scales, etc.) that are separate to the underlying ontology languages used (OWL and DAML+OIL). Wang et al. (2004) and Chen et al. (2005) take a different approach. They use the standard OWL constructs, but focus instead on creating extensible domain ontologies that define standard concepts/vocabularies that can be used for describing context. Wang et al. propose CONON ontology with the use of two levels of ontology to capture general concepts and domain-specific concepts. Fig. 2 (parts a and b) shows Wang et al. proposal method.

Wang et al. show that it is possible to use reasoning to derive high-level context (e.g., the current activity of the user) from lower-level context (e.g., location information derived from sensors). This requires the definition of appropriate rules. However, OWL does not support

such rules, so they are instead represented in a separate (non-standard) format. Further, Wang et al.'s original proposal does not address quality of context information. Their later work (Gu et al., 2004) introduces extensions for modeling the derivation of context information and relevant quality indicators. However, this work requires non-standard extensions to OWL, and therefore is not supported by OWL tools.

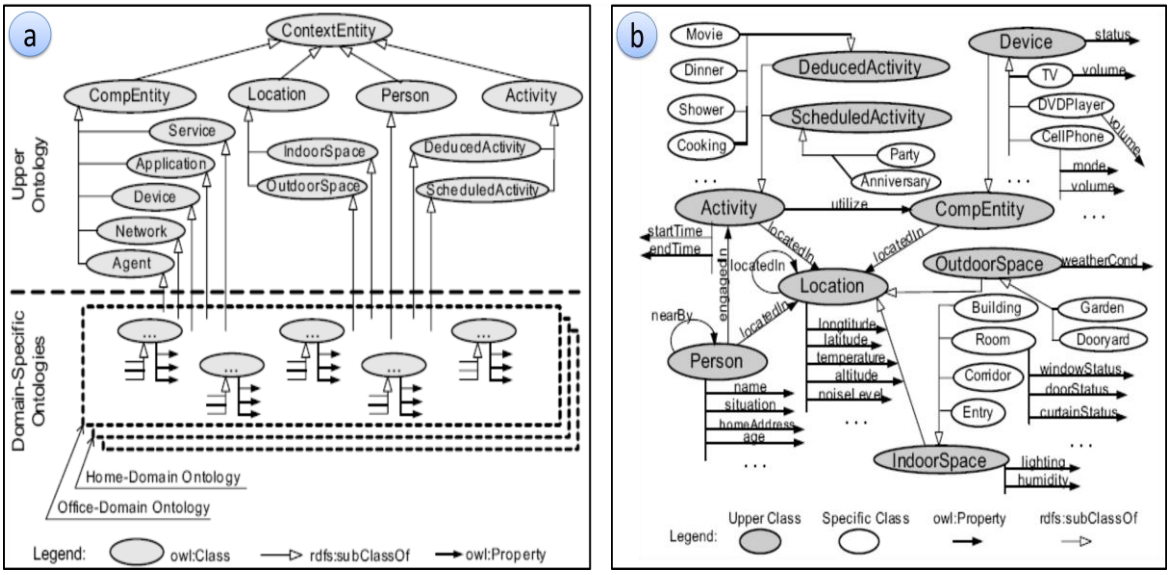

Fig. 2: CONON ontology. a) General concepts, Partial Definition of CONON upper ontology. b) Domain-specific concepts, Partial definition of a specific ontology for home domain.

The work of Chen et al. (2005) is similar to that of Wang et al. in its aims, but broader in scope. Chen et al. propose a set of OWL ontologies, collectively referred to as SOUPA, that address numerous modeling issues related to context-aware applications, including context modeling, modeling of concepts from the field of intelligent agents, such as roles, beliefs and intentions, and modeling of privacy policies for controlling access to sensitive information. SOUPA builds on a variety of well-known ontologies, such as Friend-Of-A-Friend (Brickley and Miller, 2005) and DAML-Time (Hobbs et al., 2002). SOUPA also incorporates an earlier set of ontologies called COBRA ONT, which were developed by Chen et al. (2004b) for modeling context information in smart meeting rooms.

Ranganathan and Campbell (2003) propose a very different approach that is based primarily on first order logic, but also makes use of simple ontologies expressed using DAML+OIL. They represent context information in terms of predicates, such as "Temperature(room 3231, "=", 98F)" and "Location(chris, entering, room 3231)" (Ranganathan and Campbell, 2003). The ontology definitions can be used to check the validity of predicates, and also as a basis for defining mappings between different predicates, in order to support interoperability. Context predicates can be combined to form complex logical expressions using the operators 'and', 'or' and 'not', and the universal and existential quantifiers.

3.2.3. The Context Modeling Language

Henricksen et al. propose a context modeling approach that supports incremental development of context models, beginning during the requirements analysis and design phases of the software engineering process, and continuing through to application deployment, execution and beyond (Henricksen et al., 2005a; Henricksen and Indulska, 2006). This approach builds

on the Object-Role Modeling (ORM) technique (Halpin, 2001), which is traditionally used for database modeling. ORM uses a graphical notation for creating a diagrammatic representation of relevant concepts and relationships between the concepts. In the terminology of ORM, concepts are known as entities and relationships as fact types. As ORM is designed for database modeling, not context modeling, it lacks powerful ways to describe relevant meta-attributes of context, such as sources of information and quality attributes. For this reason, Henricksen et al. extended ORM with a number of special-purpose context modeling constructs (Henricksen et al., 2005a), originally introduced in an earlier context modeling notation (Henricksen et al., 2002). This extended variant of ORM is known as the Context Modeling Language (CML). A simple example model specified using the CML notation is shown in Fig. 3.

ORM, the database modeling approach on which CML is based, is well established as a requirements analysis technique, and therefore much has been written about the process of constructing ORM models in cooperation with experts in the domain that is being modeled and/or intended database/application users. This process can be adapted to provide guidelines for analyzing context requirements and constructing a context model using CML. In addition, there is a straightforward mapping of ORM models to relational databases. This mapping procedure can be extended to allow mapping of CML models to run time context models stored in context repositories that take the form of enhanced relational databases supporting specialized context meta-attribute and constraints. The run-time models can be queried using either standard relational database query languages, or by evaluating pre-defined 'situations' expressed using a form of predicate logic. Situations provide basic support for evaluating ambiguous context using a three valued-logic (where situation expressions can be 'true', 'false' or 'possibly true'); in addition, situations can incorporate special forms of existential and universal quantification, in which variables are constrained by binding them according to a fact template, thereby ensuring efficient and safe evaluation.

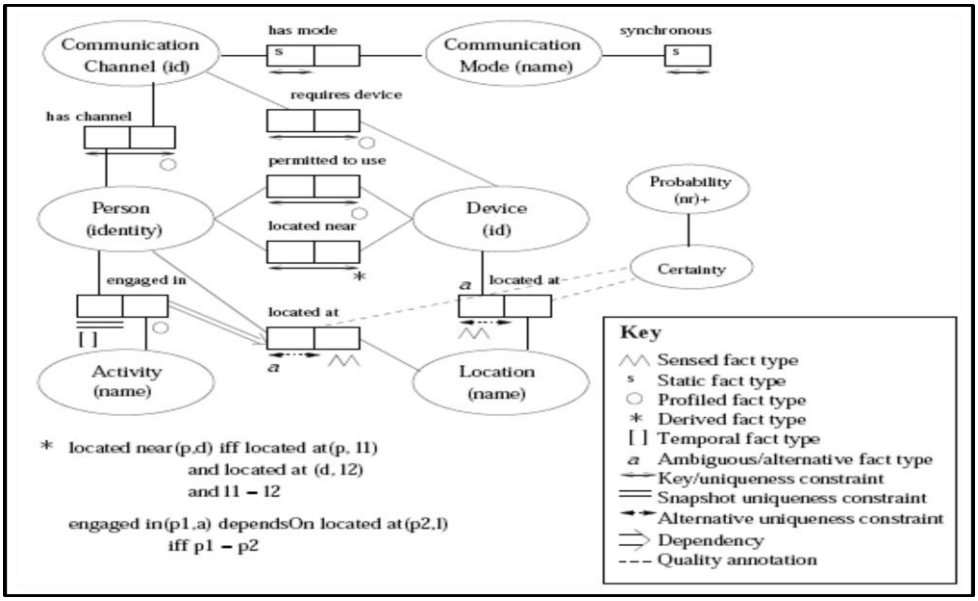

Fig. 3: An example CML model.

4. Conclusions of Simulation and Analysis

This paper of context modeling approaches is intended to be illustrative; it covers most of the well-known work in the area. Owing to the immaturity of the field of context-awareness, none of the modeling approaches has been widely adopted. CC/PP has the advantage of being standardized as a W3C Recommendation, but it is unsuitable for representing many types of information (Indulska et al., 2003), and its support for reasoning is limited. OWL-based modeling approaches are better able to support reasoning and are currently enjoying favor.

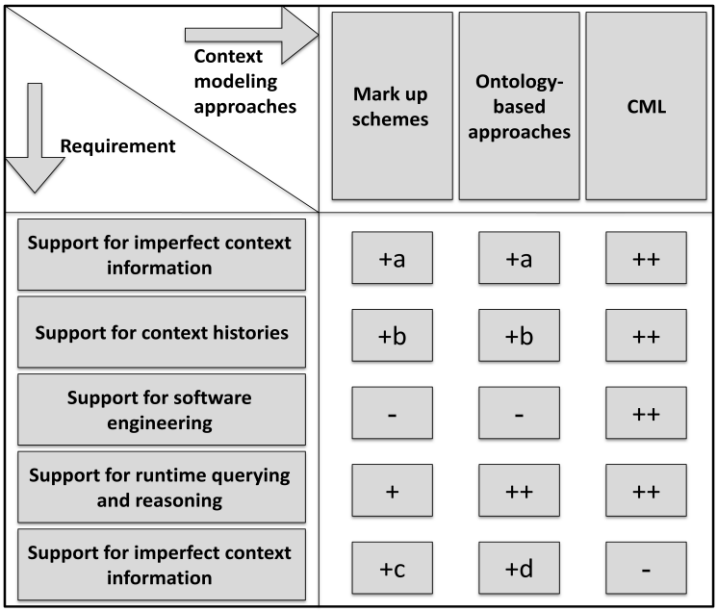

Fig. 4: An analysis of the context modeling approaches discussed in Sections 3.1
(Key: ++ : comprehensive support, + : partial support, - : no support , a: Imperfect information can usually be represented in some form (although rarely in a very natural way), but reasoning over imperfect information is not supported by conventional tools, b: Can be represented, but the majority of the approaches do not define natural concepts/vocabularies for doing so, c: Based on the use of standard vocabularies, d: Based on the use of standard vocabularies and defined mappings between concepts.)

However, these address run-time issues (representation of context information, reasoning and interoperability), not software engineering tasks such as requirements analysis and design.
CML offers the particular advantage that it supports the mapping of a requirements model to a run-time model. Fig. 4 presents a summary of the strengths of the various approaches. As none of the approaches is comprehensive in the sense that it addresses all of the requirements introduced in Section 3.1.

References

Bettini, C. et al. (2010). *A survey of context modelling and reasoning techniques.* Elsevier, Pervasive and Mobile Computing, 6 (2), pp 161_180.

Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., and Yergeau, F. (2004). Extensible Markup Language (XML 1.0) (third edition). W3C Recommendation, 4 February 2004.

Brickley, D. andMiller, L. (2005). FOAF vocabulary specification. Namespace Document, 27 July 2005.

Buchholz, S., Hamann, T., and Hubsch, G. (2004). Comprehensive Structured Context Profiles (CSCP): Design and experiences. In 1st Workshop on Context Modeling and Reasoning, PerCom'04 Workshop Proceedings, pages 43–47. IEEE Computer Society.

Chen, H., Finin, T., and Joshi, A. (2004). An ontology for contextaware pervasive computing environments. Knowledge Engineering Review, 18(3):197–207.

Chen, H., Finin, T., and Joshi, A. (2005). The SOUPA Ontology for Pervasive Computing. Ontologies for Agents: Theory and Experiences. Springer.

Dey, A. & Abowd, D. (1999). *Towards a Better Understanding of Context and Context Awareness*, Proceedings of the 1st international symposium on Handheld and Ubiquitous Comuting, Lecture Notes In Computer Science. 1707 pp. 304-307 Karlsruhe, Germany: Springer-Verlag.

Gu, T., Wang, X. H., Pung, K. K., and Zhang, D. Q. (2004). An ontologybased context model in intelligent environments. In Communication Net- works and Distributed Systems Modeling and Simulation Conference, SanDiego.

Halpin, T. A. (2001). Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design. Morgan Kaufman, San Francisco.

Henricksen, K., Indulska, J., and Rakotonirainy, A. (2002). Modeling context information in pervasive computing systems. In 1st International Conference on Pervasive Computing, volume 2414 of Lecture Notes in Computer Science, pages 167–180. Springer.

Henricksen, K., Indulska, J., and McFadden, T. (2005). Modeling context information with ORM. In OTM Federated Conferences Workshop on Object-Role Modeling, volume 3762 of Lecture Notes in Computer Science, pages 626–635. Springer.

Henricksen, K. and Indulska, J. (2006). Developing context-aware pervasive computing applications: Models and approach. Journal of Pervasive and Mobile Computing, 2(1):37–64.

Hobbs, J. R. et al. (2002). A DAML ontology of time, November 2002. http://www.cs.rochester.edu/ ferguson/daml/daml-time-nov2002.txt.

Horrocks, I. (2002). DAML+OIL: A description logic for the semantic web. IEEE Data Engineering Bulletin, 25(1):4–9.

Indulska, J., Robinson, R., Rakotonirainy, A., and Henricksen, K. (2003). Experiences in using CC/PP in context aware systems. In 4th Interna- tional Conference on Mobile Data Management, volume 2574 of Lecture Notes in Computer Science, pages 247–261. Springer.

Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M. H., and Tran, L. (2004). Composite Capability/Preference Profiles (CC/PP): Structure and vocabularies 1.0. W3C Recommendation, 15 January 2004.

McGuinness, D. L. and van Harmelen, F. (2004). OWL Web Ontology Language overview. W3C Recommendation, 10 February 2004.

Ranganathan, A. and Campbell, R. H. (2003). An infrastructure for context-awareness based on first-order logic. Personal and Ubiquitous Computing, 7(6):353–364.

Strang, T., Linnhoff-Popien, C., and Frank, K. (2003). CoOL: A Context Ontology Language to Enable Contextual Interoperability. In 4th International Conference on Distributed

Applications and Interoperable Systems, volume 2893 of Lecture Notes in Computer Science, pages 236–247. Springer

Wang, Z., Zhang, D., Gu, T., Dong, J., and Pung, H. K. (2004). Ontology based context modeling and reasoning using OWL. In Workshop on Context Modeling and Reasoning, PerCom'04 Workshop Proceedings, pages 18–22, Orlando. IEEE Computer Society.