Ontologies in Ubiquitous Computing

José R. Hilera[1], Francisco Ruiz[2]

[1] Department of Computer Science, University of Alcala, Spain.
jose.hilera@uah.es
[2] Department of Information Technologies and Systems, University of Castilla-La Mancha, Spain.
francisco.ruizg@uclm.es

Abstract. Ontologies are an explicit specification of a conceptualization, that is understood to be an abstract and simplified version of the world to be represented. In recent years, ontologies have been used in Ubiquitous Computing, especially for the development of context-aware applications. In this paper, we offer a taxonomy for classifying ontologies used in Ubiquitous Computing, in which two main categories are distinguished: *Domain ontologies*, created to represent and communicate agreed knowledge within some sub-domain of Ubiquitous Computing; and *Ontologies as software artifacts*, when ontologies play the role of an additional type of artifact in ubiquitous computing applications. The latter category is subdivided according with the moment in that ontologies are used: at development time or at run time. Also, we analyze and classify (based on this taxonomy) some recently published works.

Key Words. Ubiquitous Computing, Ontology, Knowledge Domain, Context-Aware Computing.

1. Introduction

Although a generally accepted meaning of the term “Ubiquitous Computing” does not exist, it is usual to consider it as a concept that refers to what Weiser has described as “methods of enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user” [1]. Since the end of the eighties of the last century, the evolution of Ubiquitous Computing has given place to new terms, like “Mobile Computing”, “Wearable Computing”, “Pervasive Computing”, “Context-Aware Computing” or “Tangible Computing”; presents in the lists of topics of the most important congresses and journals about the matter. Nevertheless, not all the researchers in this field agree in the meaning of these terms. For example, for many authors the terms "Pervasive Computing" and "Ubiquitous Computing" are synonymous while for others, as Lyytinen and Yoo [2], represent different concepts because the second implies a higher level of mobility that the first one.

This difference of interpretation is mainly due to the fact that it is a very new scientific discipline, and this situation affects also the researchers when searching and reading published works about this topic. In a similar way with other more consolidated fields, the existence of a model with the conceptualization of the common knowledge of this area would facilitate the classification and interpretation of new knowledge generated in the field. An ontology is a good tool, though not the only one, in order to specify this type of conceptualizations. This choice seems the most appropiated because most of the works are published and/or located across Internet, and precisely ontologies are a core element in the emergent "Semantic Web" paradigm, an evolution of the current Web (a "Syntactic Web" based on the syntactic analysis of indexed Web pages by searching engines), that will allow to annotate documents and resources in the Web with semantic information based on ontologies [3].

In the last years ontologies are being used for the development of ubiquitous computing applications, especially as software artifacts with the goal of modeling the information managed by such applications. It is habitual their use for facilitating the inter-operatibility among context-aware applications and the entities that may enter in the context at any time. For example, in the case of a user with a PDA visiting a museum, an entity could be a device, located in a room, that interacts with the application of the PDA when the user enters inside the room. This interaction must be based in common well-defined concepts ("museum", "picture", "sculpture", "author", etc.), and for it, as much the application as the context entity must use the same ontology, or alternatingly use a intermediate system or service of mapping.

Because every time is more frequent the combined use of ontologies and ubiquitous computing, in oerder to facilitate their analysis and reuse it is suitable to have a method to classify the proposals published in the literature. In this paper we present a simple taxonomy that can help in this effort.

The organization of the rest of the paper is: First, a brief introduction to ontologies in general is presented, and later the proposed taxonomy is described. Finally, a section of conclusions is included.

2. Ontologies

An ontology is "an explicit specification of a conceptualization " [4], that is to say, a formal representation of a knowledge domain. Usually an ontology consists of: i) *classes*, which represent the concepts of the domain (for example, in an ontology about the domain of Telecommunications, as in listing 1, a possible concept could be `"Phone"`); ii) *properties*, to establish relationships between the concepts (for example, a `"Phone"` concept could have as property the `"Company"` that makes it, being this one another concept of the ontology); iii) *instances*, with concrete examples associated with every concept (for example, `"Siemens"` could be an instance of the `"Company"`concept); and iv) *axioms*, which are restrictions applicable to certain elements of the ontology, necessary to specify completely the knowledge domain (for example, in the ontology about telecommunications, it could define a restriction to indicate that in this domain a `"Phone"` must have always, at least, a `"Company"`).

Ontologies can be stored using XML-based markup languages as OWL (Ontology Web Language) [5], which facilitates their reuse in different semantic platforms to annotate and search resources. These languages allow us to define tags in order to represent the different ontology elements. Listing 1 shows an extract of a OWL file containing the telecommunications example ontology that have been created using the *Protegé* tool (http:// protege.stanford.edu/). As you can observe, in this language the concepts are delimited by the `<Class>` tag, the properties by the `<ObjectProperty>` tag, the instances by the tag corresponding to the associate class (in the example, the class `<Company>` has as instance `"Siemens"`), and the axioms with tags like `<Restriction>` or `<subClassOf>` (this one is used in the example for representing that `"Cellphone"` is a type of `"Phone"`).

Listing 1. Content of a OWL file that includes a fragment of an ontology about Telecommunications.

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.owl-ontologies.com/unnamed.owl#"
    xml:base="http://www.owl-ontologies.com/unnamed.owl">

  <owl:Ontology rdf:about="Telecommunications"/>

  <owl:Class rdf:ID="Company"/>

  <owl:Class rdf:ID="Cellphone">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Phone"/>
    </rdfs:subClassOf>
  </owl:Class>

  <owl:Class rdf:about="#Phone">
    <rdfs:subClassOf rdf:resource="www.w3.org/2002/07/owl#Thing"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="company"/>
        </owl:onProperty>
        <owl:allValuesFrom rdf:resource="#Company"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

  <owl:ObjectProperty rdf:about="#comapany">
    <rdfs:domain rdf:resource="#Phone"/>
  </owl:ObjectProperty>

  <Cellphone rdf:ID="MobileC55">
    <company>
      <Company rdf:ID="Siemens"/>
    </company>
    <rdfs:seeAlso rdf:resource="http://www.telecosiemens.com/MobileC55.pdf"/>
  </Cellphone>

  <Phone rdf:ID="Gigaset3015Classic">
    <company rdf:resource="#Siemens"/>
    <rdfs:seeAlso
      rdf:resource="http://www.telecosiemens.com/Gigaset3015Classic.pdf"/>
  </Phone>

</rdf:RDF>
```

Today a main use of Ontologies is to support the Semantic Web [3], specially for annotating Web resources and facilitating the localization of this annotated resources when users formulate queries to semantic search engines. In this way, in the previous example of Telecommunications ontology have been included two annotations as instances of the "`Phone`" and "`Cellphone`" classes which correspondents to two documents ("`Gigaset3015Classic.pdf`" and "`MobileC55.pdf`", respectively) located in a hypothetical Web server ("`http://www.telecosiemens.com`").

3. A taxonomy of ontologies in Ubiquitous Computing

When attempting to establish a relationship between ontologies and Ubiquitous Computing, the former can be considering them another technique or artifact to be applied in ubiquitous applications development process, or merely another information resource. It is also possible to use this type of conceptual tool for the representation of Ubiquitous Computing domain knowledge. This should not be forgotten when establishing a taxonomy or classification of the possible combinations between both fields. Thus, at a basic level, we propose that the ontology taxonomy for Ubiquitous Computing be formed by the following two generic categories (see table 1)[1]:

- ***Ontologies of Domain***: describe knowledge of the Ubiquitous Computing domain, or some sub-domain of this discipline.
- ***Ontologies as Software Artifacts***: used as artifacts of diverse types, in the ubiquitous computing application development process, or during the execution of the application.

In the next sections, a brief description of the fundamental characteristics of ontologies belonging to these categories, and also the subcategories that we propose in both cases, are presented.

Table 1. Taxonomy of Ontologies in Ubiquitous Computing

<table>
<tr><th>Category</th><th colspan="2">Subcategories</th></tr>
<tr><td rowspan="2">Ontology of Ubiquitous Computing Domain</td><td colspan="2">Generic (All-domain)</td></tr>
<tr><td colspan="2">Specific (Sub-domain)</td></tr>
<tr><td rowspan="3">Ontology as Software Artifacts in Ubiquitous Computing Applications</td><td colspan="2">At development time</td></tr>
<tr><td rowspan="2">At run time</td><td>Ontology-driven applications</td></tr>
<tr><td>Ontology-aware applications</td></tr>
</table>

[1] This taxonomy is an adaptation of the most generic proposed by the same authors to classify ontologies in Software Engineering and Technology [6].

3.1 Ontologies of Ubiquitous Computing Domain

This generic category refers to the ontologies which main goal is to represent (at least partially) knowledge of a certain sub-domain within Ubiquitous Computing matter. The existence of a universal ontology to fully conceptualize this domain of knowledge would assist in the resource annotation and localization, for example, in the Semantic Web, and would avoid the ambiguities and inconsistencies which are commonly produced when computer science academics, researchers, and professionals use varying terms and concepts. The taxonomy of the "ontologies of domain" can be divided in Generic (all-domain) and Specific (sub-domain).

The "*Ubiquitous Conmputing generic ontologies*", also denominated as "Ubiquitous Computing all-domain ontologies", has the ambitious objective of modelling the complete Ubiquitous Computing body of knowledge. On the other hand, the "*Ubiquitous Computing specific ontologies*" only attempts to conceptualize one part (sub-domain) of this discipline, of interest for a determined goal, collective, or moment.

They do not exist proposals of ontologies that shape the whole knowledge domain of this discipline. Nevertheless, several ontologies of specific type have been published. One of them is SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications), which offer developers a shared ontology that combines many useful vocabularies from different consensus ontologies [7]. Their objective is to assist the ubiquitous and pervasive applications developers who are inexperienced in knowledge representation, in quickly begin building ontology-driven applications. SOUPA includes concepts such as *Agent* (to represent human users, with properties such as *Believes, Desires*, or *Intends*), *Action, Time, Device,* or *Location*.

Another specific ontology is CONON (CONtext ONtology), that provides an upper context ontology including general concepts to be common to all context-aware applications [8]. This ontology contains a set of upper-level concepts (*ContextEntity, Location, Person, Activity, IndoorSpace, Device*, etc.) and provide flexible extensibility to add specific concepts in different application domains (i.e., *CellPhone* can be a sub class of *Device*).

It is usually that ontologies like the previous ones are integrated with other ontologies widely accepted with the goal of knowledge reuse, one of the most important utilities of ontologies. An example of this type is "FIPA Device Ontology" [9], which can be used as reference to express the capabilities of different devices in an ubiquitous computing system. Some concepts of FIPA are: *Device*, *HardwareDescription*, *SoftwareDescription* and *ConnectionDescription*. Other one is GUMO (General User Model Ontology), a top level ontology for ubiquitous user modelling [10].

Ontologies of domain can be used in an hierarchical manner in order to create "ontologies as software artifacts", as it is described in the following section.

3.2 Ontologies as software artifacts in Ubiquitous Computing applications

In addition to the ontologies that conceptualize the knowledge of Ubiquitous Computing (sub)domains, there are other types of proposals that use ontologies as artifacts, with varying characteristics and functionalities, during the construction or functioning of ubiquitous computing software. Many authors have researched the usefulness of using ontologies in this way, even basing the software development process on this technology, and giving way to what Guarino [11] has termed "*Ontology-driven Information System development*". In fact, the World Wide Web Consortium (W3C), a main precursor in the use of ontology for Semantic Web, also endorses the use of ontologies for software development [12].

When it comes to proposing a taxonomy or classification of the ontologies that have been used as software artifacts in recent years, it seems reasonable to do so in function of the ontology´s use as an artifact (requirements specification, system conceptual modeling, etc.). Given that the software artifacts can be employed either at development or at run time, we have opted for the first-level classification proposed by Guarino [11], where analyzing the usefulness of ontologies in the Information Systems field distinguished between those artifacts used at system development and those used during system execution.

The first of these categories, that is, "**Ontologies as software artifacts at development time**" includes the ontologies used as artifacts in software development and maintenance, or in other complementary activities of the development: support activities, project management, knowledge reuse, etc. Using ontologies as knowledge's artifacts, facilitates the communication among project stakeholders and avoids ambiguities of natural language, as well as filters knowledge when defining requirements, models and metamodels of ubiquitous computing systems to be developed. Un ejemplo de este tipo is the GAS Ontology [13], that conceptualises the Gadgetware Architectural Style (GAS), which supports the composition of ubiquitous computing applications from everyday physical objects enhanced with sensing, acting, processing and communication abilities.

In the case of the category referred to as "**Ontologies as software artifacts at run time**", following the same reasoning as Guarino [11], we have determined two different situations:

a) Ontologies as architectural artifacts: When ontologies are part of the system software architecture, as an additional component, cooperating with the rest of the system at run time to attain the software objective. Se trata de ontology-driven software.
b) Ontologies as (information) resources: Are used by the software during run time for a specific purpose, as an information resource, normally remote, upon which the software operates, carrying out, for example, specific queries. It is the so-called ontology-aware software.

A lot of the proposals that exist in the literature can be included in the ontology-driven applications category, in which the software architecture is characterized by the use of one or more ontologies as central elements of the system. This is a knowledge-based system whose architecture consists mainly of a knowledge repository that is formed by an ontology and an inference engine acting on said repository. In [14], Cristopoulou et al. present an architecture for ubiquitous computing applications, and propose to integrate an ontology and an inference engine in the architecture. The basic goal of the ontology is to support a context management process based on a set of rules which determine how a decision should be made and how must be applied on existing knowledge represented by this ontology.

Ranganathan et al. [15] have integrated ontologies in a smart spaces framework called GAIA, using an ontology server with the following goals: the interoperability among different entities, the semantic discovery and matchmaking of the arrival and departure of mobile entities in the environment, and the context-awareness in ubiquitous computing systems. Power et al. have developed another work similar to the previous one [16]. This researchers have proposed an ontology-driven context service architecture to perform distributed open schema queries over heterogeneous context sources (entities) in ubiquitous computing environments.

4. Conclusions

In the last years a lot of works that propose to use ontologies in Ubiquitous Computing have been published, either as a way of representing the knowledge associated with this recent scientific field; or as software artifacts in ubiquitous computing applications, specially in those that operate within an extremely dynamic and heterogeneous environment and have to dynamically adapt to changes in their environment as a result of users or other context entities. The use of ontologies for the context definition, representation and management can improve their operation.

In this paper a taxonomy to classify ontologies in the Ubiquitous Computing field has been presented. Since the great number of different existing works published, this taxonomy can be useful to organize these sources. It is a general taxonomy that can be extended by means of the decomposition of the categories in other more specific subcategories. For example, the "Ontologies as software artifacts" category could be detailed in others sub-categories considering the different possible uses of ontologies in different activities of ubiquitous computing applications development projects: requirements specification, design, programming or maintenance of these applications.

References

1. Weiser, M.: Some Computer Science Issues in Ubiquitous Computing. Communications of the ACM, 36(7), pp. 74-84, 1993.
2. Lyytinen, K., Yoo, Y.: Issues and Challenges in Ubiquitous Computing. Communications of the ACM, 45(12), pp. 63-65, 2002.
3. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web, Scientific American, 284(5), pp. 34-43, 2001.
4. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5, pp. 199-200, 1993.
5. Web Ontology Language (OWL). World Wide Web Consortium, 2004. Available in http://www.w3.org/2004/OWL.
6. Ruiz, F., Hilera, J.R.: Using Ontologies in Software Engineering and Technology. In "Ontologies for Software Engineering and Software Technology", Springer, chapter 2, 2006.

7. Chen, H., Perich, F., Finin, T., Joshi, A.: SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. International Conference on Mobile and Ubiquitous Systems: Networking and Services, Boston, 22-25 August 2004, pp. 258-267.
8. Wang, X.H., Gu, T., Zhang, D.Q., Pung, H.K.: Ontology Based Context Modeling and Reasoning using OWL. Proc. 2nd IEEE Conf. Pervasive Computing and Communications, Workshop on Context Modeling and Reasoning, IEEE CS Press, 2004, pp. 18–22.
9. FIPA Device Ontology Specification. Foundation for Intelligent Physical Agents (FIPA), Geneva, Switzerland, 2001. http://www.fipa.org/specs/fipa00091/XC00091C.pdf.
10. Heckman, D., Schwartz, T., Brandherm, B., Von Wilamowitz-Moellendorff, M.: GUMO, the General User Model Ontology, Proc. Tenth International Conference on User Modeling (UM'2005), Edinburgh, Scotland, 24–29 July, 2005, pp.428–432.
11. Guarino, N.: Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, IOS Press, Amsterdam, 1998.
12. Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering, Draft 2006/02/11. World Wide Web Consortium. Available in http://www.w3.org/2001/sw/BestPractices/SE/ODA/.
13. Christopoulou E., Goumopoulos C., Zaharakis I., Kameas A.: An Ontology-based Conceptual Model for Composing Context-Aware Applications. Workshop on Advanced Context Modelling, Reasoning and Management, Sixth International Conference on Ubiquitous Computing (UbiComp), Nottingham, England, 7-10 September, 2004.
14. Chistopoulou, E., Goumopoulos, C., Kameas, A.: An ontology-based context management and reasoning process for UbiComp applications. Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence, Grenoble, France, October 2005, pp. 265-270.
15. Ranganathan, A., McGrath, R., Campbell, R., Mickunas, D.: Ontologies in a Pervasive Computing Environment. Workshop on Ontologies in Distributed Systems at IJCAI, Acapulco, Mexico, 2003.
16. Power, R., Levis, D., O'Sullivan, D., Conlan, O., Wade, V.: A Context Information Service using Ontology-Based Queries. Workshop on Advanced Context Modelling, Reasoning and Management, Sixth International Conference on Ubiquitous Computing, Nottingham, England, 7-10 September, 2004.