Information and Software Technology 54 (2012) 759–785

Contents lists available at SciVerse ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Towards a framework to characterize ubiquitous software projects

Rodrigo Oliveira Spínola, Guilherme Horta Travassos *

Systems Engineering and Computer Science Program, COPPE, Federal University of Rio de Janeiro, P.O. Box 68511, Rio de Janeiro, Brazil

A R T I C L E   I N F O

Article history:
Received 19 July 2011
Received in revised form 29 December 2011
Accepted 25 January 2012
Available online 4 February 2012

Keywords:
Ubiquitous computing
Software projects characterization
Systematic review
Experimental software engineering

A B S T R A C T

Context: Ubiquitous Computing (or UbiComp) represents a paradigm in which information processing is thoroughly integrated into everyday objects and activities. From a Software Engineering point of view this development scenario brings new challenges in tailoring or building software processes, impacting current software technologies. However, it has not yet been explicitly shown how to characterize a software project with the perception of ubiquitous computing.
Objective: This paper presents a conceptual framework to support the characterization of ubiquitous software projects according to their ubiquity adherence level. It also intends to apply such characterization approach to some projects, aiming at observing their adherence with ubiquitous computing principles.
Method: To follow a research strategy based on systematic reviews and surveys to acquire UbiComp knowledge and organize a conceptual framework regarding ubiquitous computing, which can be used to characterize UbiComp software projects. Besides, to demonstrate its application by characterizing some software projects.
Results: Ubiquitous computing encapsulates at least 11 different high abstraction level characteristics represented by 123 functional and 45 restrictive factors. Based on this a checklist was organized to allow the characterization of ubiquitous software projects, which has been applied on 26 ubiquitous software projects from four different application domains (ambient intelligence, pervasive healthcare, U-learning, and urban space). No project demonstrated to support more than 65% of the characteristics set. Service omnipresence was observed in all of these projects. However, some characteristics, although identified as necessary in the checklist, were not identified in any of them.
Conclusion: There are characteristics that identify a software project as ubiquitous. However, a ubiquitous software project does not necessarily have to implement all of them. The application domain can influence the appearing of UbiComp characteristics in software projects, promoting an increase of their adherence to UbiComp and, thus, for additional software technologies to deal with these ubiquitous requirements.

1. Introduction

Many recent software projects have been characterized by a great number of requirements directly associated with the usual perception of ubiquitous computing. Ubiquitous Computing (or UbiComp) represents a paradigm in which information processing is thoroughly integrated into everyday objects and activities [121].

Moreover, UbiComp technologies not only enable new ways for acting and interacting, but also stimulate fundamental reassessments of the meaning of human action and interaction [46]. Lyytinen and Yoo [67] stated that radical improvements in microprocessor cost/performance ratios have pushed this process forward while drastically reducing computing-device format factors, allowing the use of these devices in ordinary environments such as classrooms, supermarkets, shopping centres, buildings, houses, airports, and so on. For instance, we can presently see some effort involved in the construction of ubiquitous applications in domain areas such as ambient intelligence, pervasive healthcare, ubiquitous learning, and urban spaces [1,44,66,36]. The development of those applications involve additional characteristics, such as context sensitivity, user experience capture, service omnipresence, alternative user interfaces and so on, which are usually not addressed in traditional software projects [105].

Along the last years, Computer Science researchers have made real progress in the UbiComp domain [120]. From a Software Engineering point of view, this development scenario can bring new challenges in tailoring or building software processes, impacting current methods, techniques, architectural styles, requirements gathering and verification [3,60,75].

Thus, we understand it is important to work on the development of Software Engineering technologies to support the development of this software category [102]. However, before proposing any Software Engineering technology, we consider an essential

* Corresponding author. Tel.: +55 21 2562 8712; fax: +55 21 2562 8676.
E-mail addresses: ros@cos.ufrj.br (R.O. Spínola), ght@cos.ufrj.br (G.H. Travassos).

0950-5849/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2012.01.009

point to address some investigation aiming at identifying what the ubiquitous computing characteristics are, and based on that, organize a body of knowledge on the UbiComp domain to support the development of this software category. More, it is also important to investigate how those UbiComp characteristics have been applied on current software projects. We believe this is an important step towards understanding how the ubiquity domain can affect the software development life cycle. That is what we have observed when dealing with some innovative software projects regarding e-science in Brazil, where one of the requirements explicitly mentioned the characteristic of ubiquity. From that moment, we have identified the need of understanding the impact of ubiquitous characteristics in the software project planning. However, no information could be easily found by us regarding this new software category in the technical literature.

Based on this context, this paper intends to present a Software Engineering point of view on ubiquitous computing, and, from that view, provide a checklist to characterize software projects according to their adherence level regarding ubiquity. Software projects presenting any adherence with the UbiComp domain are called ubiquitous software projects into this text. Additionally, the characterization of 26 ubiquitous software projects will be made and some insights on the distance between ubiquitous computing principles and software projects are discussed.

To reach this goal, we decided to follow an evidence-based research strategy to support our work in ubiquitous application development. The UbiComp body of knowledge organization followed the scientific approach shown in Fig. 1 [23]. It uses systematic reviews [8] (secondary studies) and surveys (primary studies) to acquire knowledge from the field. This research strategy follows Mary Shaw's suggestions [98] on what makes good research in Software Engineering: (1) Defining a research question (the UbiComp field characterization). (2) Identifying the correct research results (the set of UbiComp characteristics and their factors). (3) Validating the obtained results (surveying specialists).

We conducted two systematic reviews. The first one aimed at defining ubiquitous computing, identifying where it is currently being used and a definition of its main characteristics. The results allowed us to observe that, besides its definition, ubiquitous computing can be represented by 10 different characteristics. However, these characteristics were described on a high abstraction level, and it was not possible to use them to characterize ubiquitous software projects. Therefore, we made a second systematic review to identify functional and restrictive factors associated with these UbiComp features. It revealed 123 functional and 45 restrictive factors that can support the characterization of software projects regarding ubiquity.

After that, an initial evaluation was done to analyze the identified UbiComp characteristics as regards their applicability and scope in the context of software projects. This survey allowed us to make some improvements to the initial set of UbiComp characteristics. Finally, we did a second survey to analyze UbiComp characteristics as regards their pertinence and relevance when characterizing ubiquitous software projects. As a result, we identified 11 UbiComp features and their respective relevance levels when characterizing ubiquitous software projects. Fig. 2 (an instance of the research strategy shown in Fig. 1) shows a summary of the investigation activities performed in this research, and indicates the types of study and results obtained throughout their execution.

It is possible to observe in the technical literature that part of the body of knowledge organization activities, the two systematic reviews and the first survey, have already been summarized on some of our previous works [101,104]. However, this paper brings a depth, reviewed, updated and comprehensive discussion about the performed activities and their respective reached results. More specifically:

- On paper [104] there is a discussion about the importance of software engineering area on the ubiquitous computing domain. Additionally, that paper identified a preliminary set of ubiquitous computing characteristics and summarized how the domain knowledge about ubiquitous computing was organized presenting initially a resume of the two systematic reviews including research questions, keywords, paper sources, examples of search strings, and inclusion and exclusion criteria (they are going to be deeply presented, reviewed, and updated on Sections 2 and 3 of this paper). It also illustrated the first attempt for a checklist that could be used to support the characterization of software projects regarding ubiquity. That checklist mainly differs from the one presented in this paper (Section 6) because: (1) it did not include the steps (including the necessary formulas) used to characterize a ubiquitous software project evolved; and (2) it did not have the updated version of the body of knowledge considering the final list of UbiComp characteristics and their factors.
- On paper [101] there is a resume of the ubiquitous computing body of knowledge organization. Besides, that paper also presented a briefly description of the first evaluation performed on the organized body of knowledge. This evaluation was per-

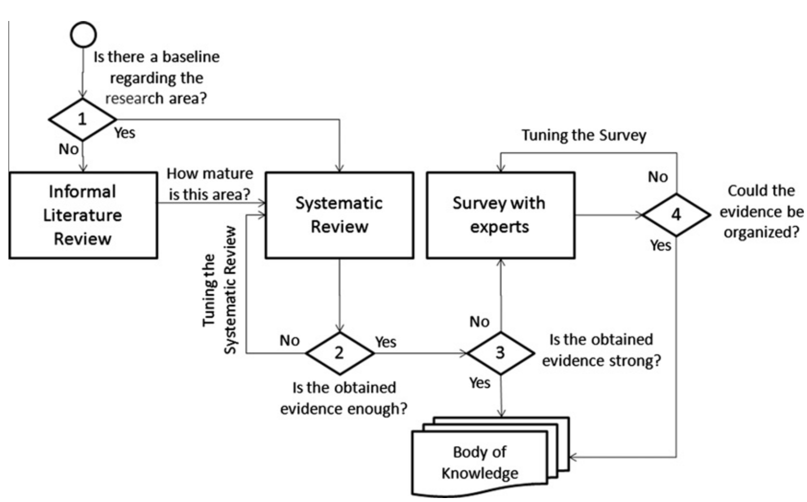


Fig. 1. The followed research strategy. Adapted from Ref. [23].

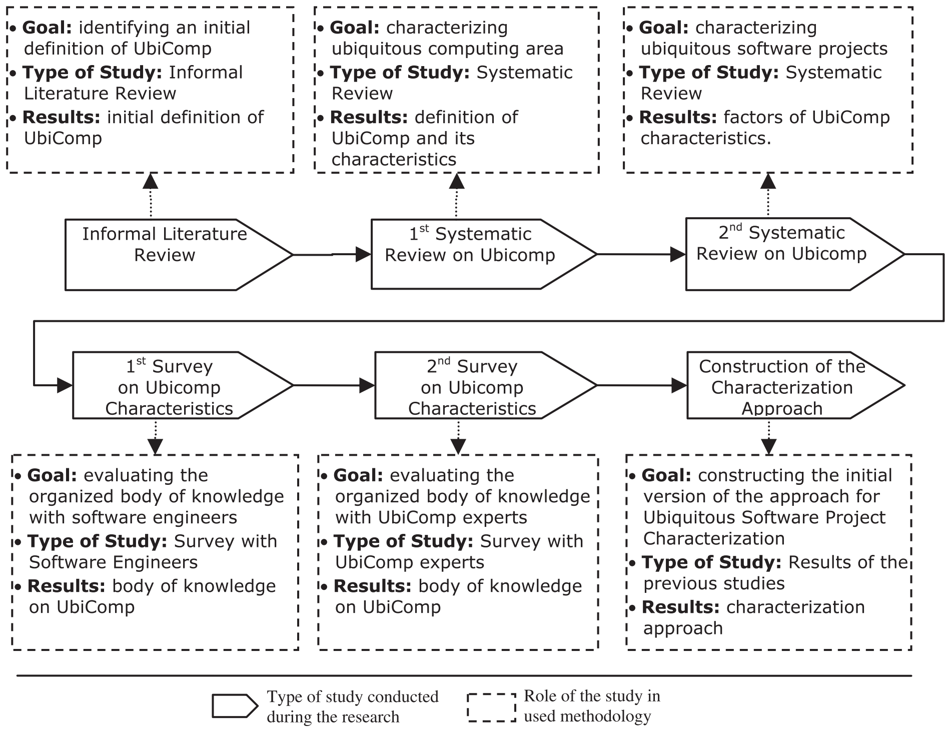


Fig. 2. Investigation activities.

formed through a survey. However, that survey (presented deeply on Section 4 of this paper) was not detailed nor deeply discussed once the main focus of the paper was to describe a proposal to support ubiquity requirements definition and verification activities.

Having organized the UbiComp body of knowledge, we used this set of information to define a checklist to support the characterization of software projects regarding ubiquity. We applied this checklist to characterize 26 different ubiquitous software applications, selected to represent four UbiComp domain areas: Ambient Intelligence [1], Pervasive Healthcare [122], Ubiquitous Learning (or U-learning) [66], and Urban spaces [36]. The importance in considering those areas justifies in the fact that: (i) They represent different perspectives on how to apply the concepts of ubiquitous computing in real software systems, and (ii) Each area brings specific challenges and requirements associated with the ubiquitous computing domain. Thus, we believe that this analysis can give us:

- directions on how far ubiquitous computing principles and ubiquitous software projects go, and;
- insights on the impact of application domain on the use of the UbiComp characteristics.

Apart from this introduction, this paper has seven more sections. Section 2 identifies the UbiComp characteristics. From those characteristics, Section 3 identifies functional and restrictive factors in each UbiComp characteristic. In the sequence, an initial evaluation of this UbiComp body of knowledge is shown in Section 4. Section 5 tackles the pertinence and relevance levels of each UbiComp characteristic shown. After that, a checklist-based approach to support the characterization of ubiquitous software projects and its use is provided in Section 6. Next, threats to validity regarding the performed and presented studies (systematic reviews and surveys) on this work are presented in Section 7. Finally, Section 8 presents the final considerations on this work.

2. Identifying ubiquitous computing characteristics

As shown in Fig. 2, the first step was an informal literature review. A search on ACM and IEEE digital libraries was done [104]. The results obtained at that moment were the theoretical foundation on UbiComp, allowing the first systematic review planning.

The steps towards a systematic review protocol include the definition of: the goal and research questions, search sources, search strings, inclusion/exclusion criteria, and strategies for classification and information to be extracted from each paper, summarized below [8].

The first systematic review goal was to characterize the UbiComp field. For this, the following research questions were defined [101]:

(i) What is ubiquitous computing?
(ii) How is ubiquitous computing being shown nowadays?
(iii) What characteristics define applications for ubiquitous computing?

The study objective was to make a characterization of the UbiComp field. There is no comparison between intervention and alternatives and no meta-analysis will be applied [78]. Therefore, this secondary study, although systematic, can be considered a *quasi*-systematic review [8].

To accomplish this *quasi*-systematic review the items below defined the main characteristics of its research protocol:

- **Keywords**: characteristic, characterization, concept, feature, definition, pervasive application, pervasive computing, pervasive software, pervasive system, requirement, ubiquitous application, ubiquitous computing, ubiquitous software, and ubiquitous system.

- **Paper sources**: Digital libraries have been chosen by convenience, since they were fully available for the researchers: ACM Digital Library, EI COMPENDEX, IEEE Portal, and INSPEC.
- **Search strings**:
  - **P0**
    - (ubiquitous computing <or> pervasive computing)<and> (definition <or> characterization <or> concept)
  - **P1**
    - ubiquitous application <or> ubiquitous system <or> ubiquitous software
    - pervasive application <or> pervasive system <or> pervasive software
  - **P2**
    - (ubiquitous computing <or> pervasive computing)<and> (feature <or> requirement <or> characteristic)
    - (ubiquitous application <or> ubiquitous system <or> ubiquitous software)<and>(feature <or> requirement <or> characteristic)
    - (pervasive application <or> pervasive system <or> pervasive software)<and>(feature <or> requirement <or> characteristic)
- **As the criteria for inclusion and exclusion of papers**:
  - Papers should be available on the internet;
  - Papers should be written in English;
  - Papers should provide a definition for ubiquitous computing (P0 only);
  - Papers should report current applications regarding ubiquitous computing concepts (P1 only);
  - Papers should report ubiquitous software projects (applications related to supporting software are not considered) (P1 only);
  - Papers should present characteristics associated with ubiquitous systems (P2 only).
- **Preliminary studies selection process**: two researchers analyzed the abstract and introduction of each publication returned and, based on the criteria for inclusion and exclusion of papers, the papers were selected or not to a more thorough analysis.
- **Information extraction**: after the selection process, the chosen papers were analyzed to extract information according to the corresponding research question. The information from each paper was organized on Tables to allow their analyzes on the next step of the systematic review execution. The extracted information was:
  - P0: definition of ubiquitous computing and ubiquitous systems;
  - P1: ubiquitous software projects where the concepts of UbiComp had been applied;
  - P2: the set of ubiquitous computing characteristics and their definitions.
- **Result analyzes**: after the information extraction activity, the tabulated results were analyzed to:
  - Prepare a definition for ubiquitous computing and ubiquitous systems. These definitions were organized from the set of definitions extracted from each paper analyzed;
  - Identify applications where UbiComp concepts were used, including their characteristics and use context.
  - Identify the basic characteristics that define ubiquitous systems. For this, the initial list of identified characteristics was reviewed to eliminate redundancies and achieve a final set of characteristics.

During this review, from 751 articles found, 41 technical papers were selected for information extraction based on the inclusion and exclusion criteria defined, in the beginning, in the systematic review protocol. Thus, only the papers that: are available on the internet, are written in English, provide a ubiquitous definition (P0 only), report current applications regarding ubiquitous computing concepts (P1 only), report software application (applications related to supporting software are not considered) and present characteristics associated with ubiquitous systems were considered. As a resulting of the application of such criteria, the following papers were selected: [2–7,9,11,18,24,25,28,33,39,43,45,48,54,59–61,63,67,80,82,91,96,100,106,107,109,112,119,121,122,127,124,130]. These papers allowed us to update the definition regarding ubiquitous computing and the identification of an **initial** set of characteristics that should be present in ubiquitous software projects.

Hence, from this 1st *quasi*-systematic review, we could see that ubiquitous computing is present when computational services or facilities are made available to people in such a way that the computer is no longer visible nor needed to be used as an essential tool to their access. The services or facilities can materialize themselves at any time or place, transparently, through the use of common daily devices. To make it happen it is necessary that systems that form this scenario take into consideration the following characteristics, which we call ubiquitous computing characteristics and their evolved definition (updated from [101]) can be found as follow:

- **Service omnipresence**: it makes users able to move around with the sensation of carrying computing services with them. For instance: an engineer manages several projects and needs to visit the development teams located in different sites. However, he also needs to monitor the other software projects' progresses to report their results for the organization. When the software engineer is visiting a specific development site, the local software development environment can connect with the other projects' environments. Thus, the engineer will have access to the software projects everywhere as he moves around.
- **Invisibility**: Ability to be present on a daily basis, using objects, weakening, from the user's point-of-view, the sensation of explicit use of a computer and enhancing the perception that objects or devices can provide services or some kind of 'intelligence'. For instance: an environment monitor that should be constantly monitoring some comfort variables and adjusting the air conditioning system or asking for maintenance without user intervention.
- **Context sensitivity**: it relates to mechanisms present in ubiquitous systems for collecting information from the environment where the system is being used. For instance: a system to control the intensity of light inside a classroom should be constantly monitoring the intensity of light to keep the room in the comfortable configuration for reading.
- **Adaptable behaviour**: it represents the dynamic capacity of self adaptation according the environment's limitations to the services that should be offered. For instance: by identifying the increasing of throughput to the point of harming the processors due their temperature, the high performance computer management system should command the increase of cooling to reduce the risk of processing failure.
- **Experience capture**: it makes the ubiquitous systems able to capture and register experiences for future use. For instance: a software for ambient intelligence can identify common user behaviours, for example: when arriving at work, the employee turns on the office light, computer, air conditioning system at 24 °C and notifies the team his presence at the office. The software can manage these activities as soon as it identifies the user arrives at office without repetitive user commands.
- **Service discovery**: it represents mechanisms to support proactive discovery of services by the ubiquitous system in accordance with the environment where it is being used, allowing the achievement of some desired target by the finding of new

services or information. For instance: a smartphone software based on the location of its user can identify the local restaurants serving the user's preferred food and discover those with available seats at the moment.
- **Function composition**: It represents the ability to create a service required by the user based on the existent basic services. For instance: a user needs to convert a XML file from one tool to another and this conversion service is not available in the computer. The software can identify the necessary services in other devices and makes them available for use.
- **Spontaneous interoperability**: it allows that according to its movement the ubiquitous system can change its partners during operation. For instance: a user is moving and the software, running on a smartphone, is executing a data-intensive process. During the moving, the software can interact with other devices in the environment for temporary allocation of information.
- **Heterogeneity of devices**: it makes able the software application to acquire mobility amongst heterogeneous devices. Thus, the software application could migrate amongst devices and adjust itself to each device. For instance: an email client that can be used in the workstation at the office or at the smartphone on the road.
- **Fault tolerance**: the facing of environment faults will lead to self adaptation. For instance: a positioning system detects a failure in the GPS module and starts to use the positioning of mobile transmission anthems to mark the location points.

We can see that the definition of ubiquitous computing represents the 'philosophy' of this computer paradigm. This way, it defines the ideal conditions where we can access computerized resources in an ubiquitous way. On the other hand, ubiquitous software projects have a well-defined scope and are strongly related to the different characteristics that form ubiquitous computing. It happens because, as ubiquity can be a property of a system, it can be achieved in full or in part. This variation relates to the fact that a particular software can implement or not all the functionalities that could represent the features of ubiquitous computing.

However, we can see that the identified UbiComp characteristics were described on a high abstraction level, making their use hard when dealing with concrete software projects. It would be more interesting to identify more precisely how each characteristic can be addressed in software projects. Therefore, it was necessary to go further, looking for more concrete functionalities associated with each UbiComp characteristic. This is the topic to be discussed in the next section.

3. Identifying the factors in ubiquitous computing characteristics

In order to have a more concrete view of how UbiComp characteristics could be found in ubiquitous software projects, a second *quasi*-systematic review was planned and executed. In this section a summary of the protocol of this review is made, again using the steps set in [8]. The goal of this second *quasi*-systematic review [104] was to answer the question:

(i) What are the functional and restrictive factors that characterize each ubiquitous computing characteristic?

To accomplish this second review, another protocol based on the first one was prepared. The items below define the main characteristics of this protocol:

- **Keywords**: adaptable behaviour or task dynamism, capture of experiences, characteristic, context sensitivity, device heterogeneity, fault tolerance, feature, functional requirement, functionality, quality requirement, invisibility, non-functional requirement, pervasive computing, service discovery, spontaneous interoperability, and ubiquitous computing.
- **Paper sources**: These digital libraries have been chosen by convenience, because they were fully available for the researchers: ACM Digital Library and IEEE Portal. On this research protocol we decided to reduce the number of digital libraries to be searched, due to the large amount of search strings used, what could make the adaptation of these different strings for each search engine harder and demanding a too long result analysis.
- **Search strings**:
  - P0: (ubiquitous computing <or> pervasive computing)<and>((functional requirement <or> functionality <or> feature <or> characteristic)<or>(no functional requirement <or> quality requirement))<and>(computer everywhere)
  - P1: (ubiquitous computing <or> pervasive computing)<and>((functional requirement <or> functionality <or> feature <or> characteristic)<or>(no functional requirement <or> quality requirement))<and>(invisibility)
  - P2: (ubiquitous computing <or> pervasive computing)<and>((functional requirement <or> functionality <or> feature <or> characteristic)<or>(no functional requirement <or> quality requirement))<and>(context awareness)
  - P3: (ubiquitous computing <or> pervasive computing)<and>((functional requirement <or> functionality <or> feature <or> characteristic)<or>(no functional requirement <or> quality requirement))<and>(adaptability)
  - P4: (ubiquitous computing <or> pervasive computing)<and>((functional requirement <or> functionality <or> feature <or> characteristic)<or>(no functional requirement <or> quality requirement))<and>(automated capture <or> experience capture)
  - P5: (ubiquitous computing <or> pervasive computing)<and>((functional requirement <or> functionality <or> feature <or> characteristic)<or>(no functional requirement <or> quality requirement))<and>(service discovery)
  - P6: (ubiquitous computing <or> pervasive computing)<and>((functional requirement <or> functionality <or> feature <or> characteristic)<or>(no functional requirement <or> quality requirement))<and>(service composition <or> functionality composition)
  - P7: (ubiquitous computing <or> pervasive computing)<and>((functional requirement <or> functionality <or> feature <or> characteristic)<or>(no functional requirement <or> quality requirement))<and>(service heterogeneity)
  - P8: (ubiquitous computing <or> pervasive computing)<and>((functional requirement <or> functionality <or> feature <or> characteristic)<or>(no functional requirement <or> quality requirement))<and>(spontaneous interoperability)
  - P9: (ubiquitous computing <or> pervasive computing)<and>((functional requirement <or>

functionality <or> feature <or> characteristic)<or>(no functional requirement <or> quality requirement))<and>(fault tolerance)

- **Criteria for the Inclusion and exclusion of papers:**
  - Papers should be available on the internet, should be written in English; and should provide functional and/or restrictive factors associated with each ubiquitous characteristic.
- **Preliminary studies selection process**: one researcher analyzed the abstract and introduction of each publication returned and, based on the criteria for inclusion and exclusion of papers, the papers were selected or not to a more thorough analysis. A second researcher was available to help on decision making in case of some doubt regarding the inclusion of any paper.
- **Information extraction**: after the selection process, the selected papers were analyzed to extract an initial list of factors associated with each UbiComp characteristic. The factors were organized in Tables to facilitate their analyzes on the result analysis activity. Thus, for each research question the following was extracted:
  - P0–9: list of factors associated with the UbiComp characteristic;
- **Result analyzes**: after the information extraction activity, the tabulated results were analyzed to define the functional and restrictive factors for each UbiComp characteristic. For this, these definitions were examined to eliminate duplication and to obtain the final set of factors. This activity was done in three steps:
  - identifying the presence of the ubiquitous characteristic;
  - identifying the factors of each characteristic, and;
  - grouping related factors in factor groups for each characteristic.

After protocol execution, from 599 identified papers, 59 scientific papers were selected for information extraction based on the inclusion and exclusion criteria defined in the first place in the review protocol [2,10,12,14–17,21,22,29,33,32,34,37,38,40–42,49–53,55,57,56,60,62,65,68,70–72,75,80,82–85,87–90,92–95,97,110,113,114,117,118,125,126,86,30,19]. They allowed the identification of 168 ubiquity factors that were organized into factor groups according to their definition and to the corresponding characteristic. These factors represent functionalities usually found in ubiquitous software projects for each UbiComp characteristic.

The analysis of the papers returned by the second *quasi*-systematic review was made in three steps [104]:

1. Identifying the presence of the ubiquitous characteristic.
2. Identifying the factors of each characteristic.
3. Grouping related factors in factor groups for each characteristic.

The results of the first and second steps are shown in Table 1. The first column shows the ubiquitous computing characteristics identified during the result analysis of the literature review shown in Section 2. The second and third columns show how many papers were found for each characteristic, in absolute and percentage values respectively. The fourth and fifth columns (functionality and restriction) show how many factors were found for each characteristic. Finally, the sixth column shows the percentage distribution of factors per characteristic.

According to the third step of the analyzes (grouping related factors in factors groups), an example of a factor/factor group/characteristic hierarchy for the 'Context sensitivity' characteristic is shown below:

Characteristic: Context sensitivity
Factor Group: Information capture
Factor: To capture user identity, location, effect, or activity.
Factor: To consider the time variable.
Factor Group: Context information management
Factor: To contextualize the information obtained.
Factor: To store the information captured.
Factor: To consider the semantics in the organization and capturing of context information.
Factor Group: Information sharing
Factor: To share context information with users and other devices.

Thus, as result of the two *quasi*-systematic review executions, it was possible to obtain:

- An updated definition of ubiquitous computing.
- An initial set of 10 UbiComp characteristics: service omnipresence, invisibility, context sensitivity, adaptable behaviour, experience capture, service discovery, function composition, spontaneous interoperability, heterogeneity of devices, and fault tolerance.
- A set of functional and restrictive factors associated with each UbiComp characteristic.

This set of concepts comprises the body of knowledge in ubiquitous computing that will be developed and used throughout this paper. These definitions reflect the technical knowledge on ubiquitous computing, as available in literature. However, its coverage may be limited by the scope defined in the used paper sources and search strings.

Besides that, according to Dybå and Dingsøyr, systematic reviews are only as good as the evidences they are based on [26]. Thus, the evaluation of the quality of the evidence obtained from systematic reviews is an important point in the process of conceiving a software technology. Dybå and Dingsøyr also point that systematic reviews whose results attained were not based on primary

Table 1
Characteristics – presence and factors. Evolved from Ref. [104].

Ubiquitous characteristic	Presence	% of 59	Functionality	Restriction	% of 168
Service omnipresence	28	47.5	9	1	6.0
Invisibility	26	44.0	8	2	6.0
Context sensitivity	56	94.9	22	8	17.9
Adaptable behaviour	52	88.1	24	8	19.0
Experience capture	11	18.6	7	0	4.2
Service discovery	28	47.5	13	13	15.5
Function composition	19	32.2	18	5	13.7
Spontaneous interoperability	21	35.6	10	2	7.1
Heterogeneity of devices	18	30.5	9	3	7.1
Fault tolerance	11	18.6	3	3	3.6
Total of factors			123	45	

studies can have the body of evidence classified as of very low strength. At this point, there are two possible paths to take:

- If the evidence has the strength required for the continuity of the research, it is possible to directly move to the structuring and organization of such evidence and knowledge as generated through a body of knowledge.
- Otherwise, it may be necessary to make an evaluation of the evidence obtained by a survey (primary study) with specialists in the domain area.

When the evidence obtained is conflicting or if there is no guarantee of its strength to enable its organization through a body of knowledge, an alternative would be to consult specialists in the domain area related to the research that is being made. At this point, surveys are good evaluation instruments of such evidence. Thus, a questionnaire should be organized and sent to the specialists.

Thus, as the systematic review results of this research were not based on primary studies, in order to have an initial evaluation from the point of view of other researchers about the organized set of characteristics and their factors, we considered to survey Software Engineering researchers who are working with the research and development of ubiquitous software projects. The survey planning and execution are described in the next section.

4. Initial survey

As described in [101], the main goal of this survey was to **analyze** the previously described ubiquity characteristics, their factors, and group of factors extracted from the technical literature **with the purpose of** characterizing **as regards** their applicability and scope **into the context of** software projects **from the point-of-view of** Software Engineering researchers working with the research and development of ubiquitous software projects.

This study intended to survey UbiComp researchers considering the following questions regarding the previously described set of ubiquity characteristics and corresponding functional and restrictive factors:

- Is there any additional ubiquity characteristic that is not present in the initial set that should be included?
- Is there any ubiquity characteristic present in the initial set that should be excluded?
- Is there any additional ubiquity characteristic factor group or factor that is not present in the initial set that should be included?
- Is there any ubiquity characteristic factor group or factor present in the initial set that should be excluded?
- Are the ubiquity characteristics and their associated factors and factor groups applicable to characterize software projects regarding ubiquity?

In this study, the following variables were defined:

- Variables related to the UbiComp characteristics:
  - C_{CI} = Initial set of UbiComp characteristics.
  - C_{IN} = Characteristics to be included in C_{CI}.
  - C_{EX} = Characteristics to be excluded from C_{CI}.
  - C_F = Final set of UbiComp characteristics.
- Variables related to the UbiComp characteristics factors groups:
  - GF_{CI} = Initial set of UbiComp characteristics factors groups.
  - GF_{IN} = Factors groups to be included in GF_{CI}.
  - GF_{EX} = Factors groups to be excluded from GF_{CI}.
  - GF_F = Final set of UbiComp characteristics factors groups.
- Variable related to the Applicability:
  - AP = Applicability.

Three null hypotheses were also defined for this study. The null hypotheses and their alternative hypotheses are:

- **Null hypothesis 1 (H0 1)**: The initial set of UbiComp characteristics is comprehensive, thus, there are no characteristics to be included nor excluded from C_{CI}.
  - **H0**: $C_{IN} = C_{EX} = \varnothing$; $C_F = C_{CI}$
- **Alternative Hypothesis (H1)**: The initial set is not comprehensive and there are UbiComp characteristics to be included in C_{CI}.
  - **H1**: $C_{IN} \neq \varnothing$; $C_F = C_{CI} + C_{IN}$
- **Alternative Hypothesis (H2)**: The initial set is not comprehensive and there are UbiComp characteristics to be excluded from C_{CI}.
  - **H2**: $C_{EX} \neq \varnothing$; $C_F = C_{CI} - C_{EX}$
- **Null hypothesis 2 (H0 2)**: The initial set of UbiComp characteristics factors groups is comprehensive, thus, there are no factors groups to be included nor excluded from GF_{CI}.
  - **H0 2**: $GF_{IN} = GF_{EX} = \varnothing$; $GF_{CI} = GF_F$
- **Alternative Hypothesis (H3)**: There are factors groups to be included in GF_{CI}.
  - **H3**: $GF_{IN} \neq \varnothing$; $GF_F = GF_{CI} + GF_{IN}$
- **Alternative Hypothesis (H4)**: There are factors groups to be excluded from GF_{CI}.
  - **H4**: $GF_{EX} \neq \varnothing$; $GF_F = GF_{CI} - GF_{EX}$
- **Null hypothesis 3 (H0 3)**: The initial set of UbiComp characteristics and their factors groups is not applicable to the characterization of ubiquitous software projects.
  - H0 3: AP = No
- **Alternative Hypothesis (H5)**: The initial set of UbiComp characteristics and their factors groups is applicable to the characterization of ubiquitous software projects.
  - H5: AP = Yes

4.1. Instrumentation and population planning

In this survey, Brazilian researchers were considered as population when planning and executing this study. Subjects were chosen through a search in the CNPq's (National Council for Scientific and Technological Development) Research Groups Search Directory[1] looking for those research groups which work with UbiComp area. All contact with subjects was done by email (about 60 invited UbiComp researchers), including the sending of questionnaire. The filling of the questionnaire was expected to happen in three steps:

(1) **Subject background and skills characterization**: In this step subjects were asked about their personal data (name, email), academic background, level of expertise in software project development (in years), and number of executed software projects per UbiComp characteristic.
(2) **Identification of ubiquity characteristics set completeness**: The subject can confirm those important ubiquity characteristics to characterize ubiquitous software projects, include or exclude characteristics in the initial set.
(3) **Identification of the ubiquity characteristic factors set completeness**: The subject can confirm those important factor group and factor, include or exclude factors in the ubiquity characteristic factor set.

[1] The CNPq (www.cnpq.br) is an agency of the Brazilian Science and Technology Department, which promotes scientific and technological research in Brazil. The Directory of Research Groups is a database with information on research groups' activity in the country. The information is continuously updated by the group leaders, researchers, students, and research leaders from the participating institutions.

4.2. Data analysis planning

For the data analysis activity, initially a weight will be attributed for each subject based on one's experience in the UbiComp domain. Thus, if a subject has high experience for a set of characteristics, for these characteristics one's answers will have a higher weight. After that definition, the answers of all subjects were analyzed for each characteristic, factor group, and factor.

The following criteria were used to support the data analyzes:

- Inclusion of characteristic/factor group/factor:
  - At least one researcher with high experience level for the characteristic + analysis of the researcher responsible for implementing the survey. (or)
  - At least two researchers with medium experience level for the characteristic + analysis of the researcher responsible for implementing the survey.
- Exclusion of characteristic/factor group/factor:
  - At least one researcher with high experience level for the characteristic + analysis of the researcher responsible for implementing the survey. (or)
  - At least two researchers with medium experience level for the characteristic + analysis of the researcher responsible for implementing the survey.

It is important to notice that the criterion 'analysis of the researcher responsible for implementing the survey' is needed because only this researcher has a broad view of the answers of all subjects. For instance, in the case of a subject with a high level of experience who suggests to exclude a characteristic while another subject, also with a high level of experience, points that a characteristic should be retained, the researcher heading the survey should decide if that characteristic should be excluded or maintained, based on the global scenario reported by the all study subjects.

4.3. Results

The survey execution resulted in 10 subjects (about 17% of invitations) answering the questionnaire (8 PhDs). Their individual characterization can be found in [101]. Table 2 summarizes the number of expert subjects for each UbiComp characteristic. The characteristics considered are: (SO) Service Omnipresence, (IN) Invisibility, (CS) Context Sensitivity, (AB) Adaptable Behaviour, (EC) Experience Capture, (SD) Service Discovery, (FC) Function Composition, (SI) Spontaneous Interoperability, (HD) Heterogeneity of Devices, and (FT) Fault Tolerance. The researchers' skill levels for each UbiComp characteristic were classified as:

- **H**igh: scholar who researches and has taken part in more than two software projects, considering the ubiquity characteristic.
- **M**edium: represents researchers that researches and has taken part of one or two software projects considering the ubiquity characteristic.
- **L**ow: represents researchers that just research the ubiquity characteristic.
- **N**one: represents researchers that neither research nor have taken part in a software project with the ubiquity characteristic.

Table 2
Number of experts for each ubiquity characteristic.

Level of expertise	SO	IN	CS	AB	EC	SD	FC	SI	DH	FT
High	6	1	6	6	4	1	1	1	2	0
Medium	2	5	1	3	2	5	6	5	3	2
Low	2	4	3	1	4	3	3	4	1	5
None	0	0	0	0	0	1	0	0	4	3

It is important to see that, except for the fault tolerance characteristic, all others have been analyzed by at least one researcher with high skill level.

For the analysis stage, each subject had a different weight according to one's background and skill level. After weight definition, the answers from all subjects were analyzed for each evaluated UbiComp characteristic, factor group, and factor.

Fig. 3 shows the organization of the initial set of ubiquity characteristics before and after survey execution. Before survey execution, 10 ubiquity characteristics were identified. The survey execution allowed us to see that these 10 characteristics can be structured considering two different perspectives: functional and restrictive. This new categorization seems to make sense as there are characteristics that are clearly related with non-functional software aspects. Moreover, the fault tolerance characteristic was included in the UbiComp restrictive characteristics group. Finally, 3 new characteristics related to non-functional aspects of software projects were identified:

- **Scalability**: indicates system ability to either handle growing amounts of work in a graceful manner or to be readily enlarged.
- **Quality of service**: indicates the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance during system execution.
- **Privacy and trust**: indicates system ability to keep the operations done by a given user confidential and ensure that this is not mocked within the system.

Besides that, some factors were included and others excluded from the initial set of factors that were evaluated. Thus, it was seen that Null Hypotheses 1, 2, and 3 (H0 1, H0 2, and H0 3) were refuted.

Despite the fact that the results contributed for the evolution of the body of knowledge on the organization of UbiComp, it is important to note that the population size was considered small and not representative considering a global scenario in UbiComp. Thus, the result of the survey could not be used as an evaluation

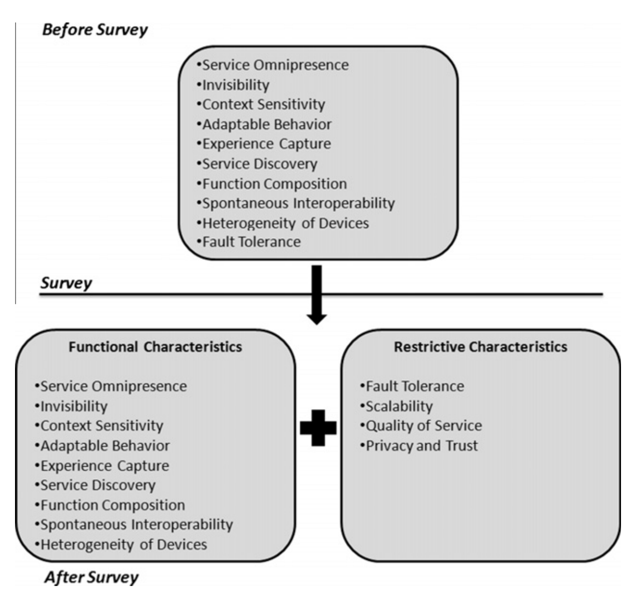


Fig. 3. Definition of UbiComp characteristics before and after survey execution.

of the body of knowledge, although their results are important to its evolution. Therefore, a second survey was done.

5. Evaluating the UbiComp body of knowledge

The goal of this study was **to analyze** ubiquity characteristics extracted from the technical literature **with the purpose of** characterization, **as regards** their adequacy and relevance when characterizing software projects regarding ubiquity, **from the point of view of** researchers on the UbiComp field, **in the context of** ubiquitous software projects.

Therefore, the following research questions were considered:

(i) Are the characteristics extracted from the technical literature adequate (or not) to characterize ubiquitous software projects?
(ii) Is there any additional characteristic to characterize a ubiquitous software project that could be considered?
(iii) What is the importance (relevance level) of each characteristic when characterizing ubiquitous software projects?

In this study, adequacy indicates if each characteristic is useful or not to describe or define a body of knowledge on UbiComp Characteristics. Relevance indicates how useful it is when characterizing a software project regarding ubiquity, that is, the weight of this characteristic on the characterization of ubiquitous software projects.

In this context, the following variables were defined for this study:

- Variables related to the pertinence of the UbiComp characteristics:
  - C_{CI} = Initial set of UbiComp characteristics.
  - C_{IN} = Characteristics to be included in C_{CI}.
  - C_{EX} = Characteristics to be excluded from C_{CI}.
  - C_F = Final set of UbiComp characteristics.
- Variable related to the relevance level of the UbiComp characteristics:
  - RE_i = relevance level of the UbiComp characteristic "*i*" considering the development of ubiquitous software projects, where "*i*" is a number from 1 to *n*, and n is the total number of UbiComp characteristics.

Two null hypotheses were also defined for this study, that are related to the pertinence and relevance levels of the UbiComp characteristics when characterizing software projects regarding ubiquity. The defined null hypotheses and their alternative hypotheses are:

- **Null hypothesis 1 (H0 1)**: The initial set of UbiComp characteristics is complete and, thus, there are no characteristics to be included or excluded from C_{CI}.
  - **H0**: $C_{IN} = C_{EX} = \varnothing$; $C_F = C_{CI}$
- **Alternative Hypothesis (H1)**: The initial set is not complete and there are UbiComp characteristics to be included in C_{CI}.
  - **H1**: $C_{IN} \neq \varnothing$; $C_F = C_{CI} + C_{IN}$
- **Alternative Hypothesis (H2)**: The initial set is not complete and there are UbiComp characteristics to be excluded from C_{CI}.
  - **H2**: $C_{EX} \neq \varnothing$; $C_F = C_{CI} - C_{EX}$
- **Null hypothesis 2 (H0 2)**: The UbiComp characteristics have the same relevance levels.
  - **H0 2**: $RE_1 = RE_2 = RE_3 = \cdots = RE_n$
- **Alternative Hypothesis (H3)**: There is at least one UbiComp characteristic that has the relevance level different from other characteristic.
  - **H3**: $RE_i \mid RE_i \neq RE_j$, $i \neq j$; (where "*i*" and "*j*" are numbers between 1 and *n*, and "*i*" ≠ "*j*")

5.1. Instrumentation and population planning

As instrumentation, an online questionnaire was developed and published in the Internet. It is filled in three steps:

(i) Subject characterization (e.g., personal data, academic degree, and experience level on ubiquitous software projects) (see Fig. 4).

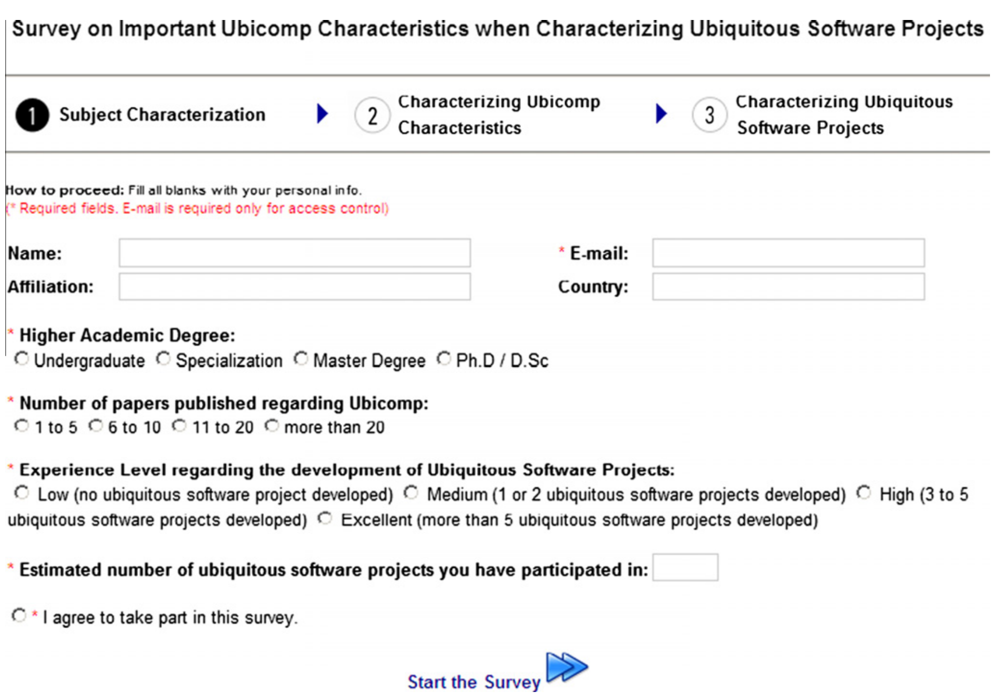


Fig. 4. Participant characterization.

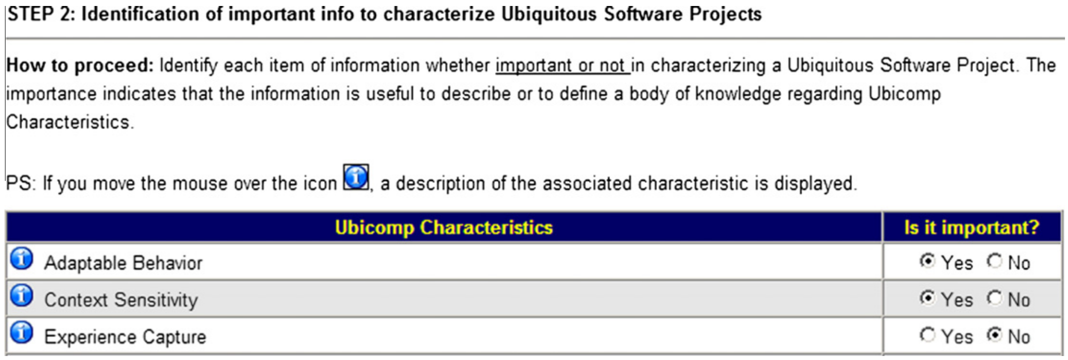


Fig. 5. Identification of characteristic pertinence.

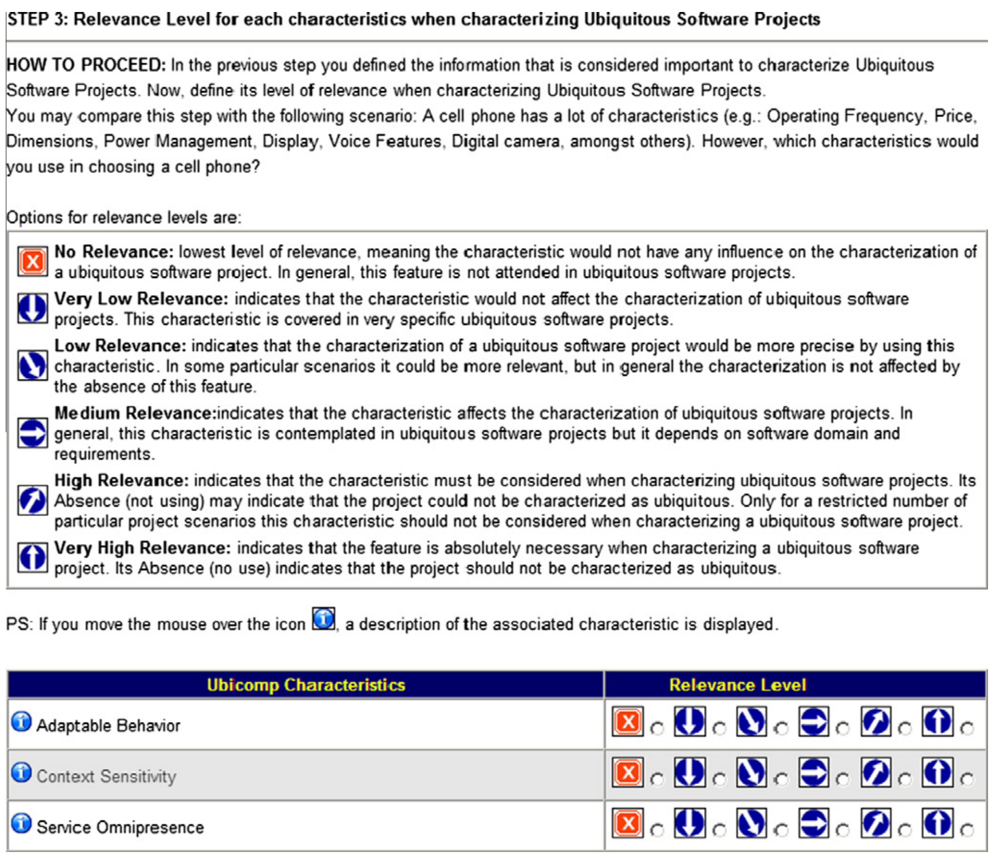


Fig. 6. Identification of the level of relevance for each characteristic.

(ii) Identification of those adequate/inadequate characteristics in characterizing ubiquitous software projects (see Fig. 5).

(iii) Definition of each characteristic's relevance level to support the characterization of ubiquitous software projects considering six relevance levels (Likert Scale) (see Fig. 6):

a. No Relevance (): lowest level of relevance, meaning the characteristic would have no influence in the characterization of an ubiquitous software project. In general, this feature is not met in ubiquitous software projects.

b. Very Low Relevance (): indicates that the characteristic would not affect the characterization of ubiquitous software projects. This characteristic is covered in very specific ubiquitous software projects.

c. Low Relevance (): indicates that the characterization of an ubiquitous software project would be more precise by using this characteristic. In some particular scenarios it could be more relevant, but in general the characterization is not affected by the absence of this feature.

d. Medium Relevance: (): indicates that the characteristic affects the characterization of ubiquitous software projects. In general, this characteristic is contemplated in ubiquitous software projects but it depends on software domain and requirements.

e. High Relevance (): indicates that the characteristic should be considered when characterizing ubiquitous software projects. Its Absence (no use) may indicate that the project could not be characterized as ubiquitous. Only for a restricted number of particular project scenarios this characteristic should not be considered when characterizing an ubiquitous software project.

f. Very High Relevance (◐): indicates that the feature is absolutely necessary when characterizing an ubiquitous software project. Its Absence (no use) indicates that the project should not be characterized as ubiquitous.

The population of this survey was represented by authors that published papers: (i) identified by two *quasi*-systematic reviews shown in Sections 2 and 3, or; (ii) in the proceedings of UBICOMP – one of the most important conferences in the area. These authors were contacted by email and were able to access a website with the questionnaire.

We assume this population can be representative in the context of UbiComp researchers, and the subjects answered the questionnaire using their background and experience in this field.

5.2. Data analysis planning

For the data analysis stage, it was necessary to define how the following variables could be calculated: subject weight, characteristic pertinence, and level of relevance of the characteristic [105]. The formula used to obtain the weights for subject 'x' is:

$$Weight(i) = f(i) + p(i) + e(i) + \frac{t(i)}{MedianTP},$$

where:

- *f(i)* is the higher level of academic degree;
- *p(i)* is the indicator for the number of papers regarding UbiComp published by the subject;
- *e(i)* is the subject's level of experience with the development of ubiquitous software projects;
- *t(i)* is the total number of ubiquitous software projects one participated in;
- *MedianTP* is the median of the total number of ubiquitous software projects considering the answers of all subjects.

To calculate the pertinence level of a ubiquity characteristic to characterize ubiquitous software projects, it is necessary to sum the answer of each subject multiplied by its respective weight [98]:

$$Pertinence(j) = \sum_{i=1}^{M}(Answer(i,j)) * Weight(i),$$

where:

- *Pertinence(j)* is the total value of the answers of all subjects (multiplied by their weights) about the adequacy of characteristic *j* to characterize ubiquitous software projects.
- *Answer(i,j)* is the indicator of being adequate (1) or not (0), defined by subject '*i*' for characteristic j.
- *Weight(i)* is the weight attributed for subject '*i*'.
- *M* is the amount of subjects in the survey.

According to [98], "a threshold of 50% of the maximum value that could be obtained for characteristic *j* in variable Pertinence(*j*) if all subjects answer YES regarding its adequacy to characterize ubiquitous software projects" can be used to support the decision of including (value greater than threshold) or not (value lower than threshold) the characteristic in the final set.

$$Threshold = 0,5 * \sum_{i=1}^{M} Weight(i)$$

Finally, to define the relevance level of each characteristic classified previously as adequate, it is necessary to sum the answers of each subject multiplied by its respective weight.

$$RLevel(j) = \sum_{i=1}^{N}(Scale(i,j) * Weight(i)),$$

where:

- *RLevel(j)* is the total value of the answers of all subjects (multiplied by their weights) for the characteristic *j*.
- *Scale(i,j)* is the scale of relevance level (0–5) defined by the subject *i* for the characteristic *j*.

After this, the UbiComp characteristics will be ranked from the highest level of relevance to the lowest. The most relevant characteristics are those that have a higher value for *RLevel(j)*.

5.3. Results

This survey was done considering a population of 280 subjects. Of this total, 31 researchers from different regions (North America, Asia and Europe) answered the questionnaire (about 11%); 22 of

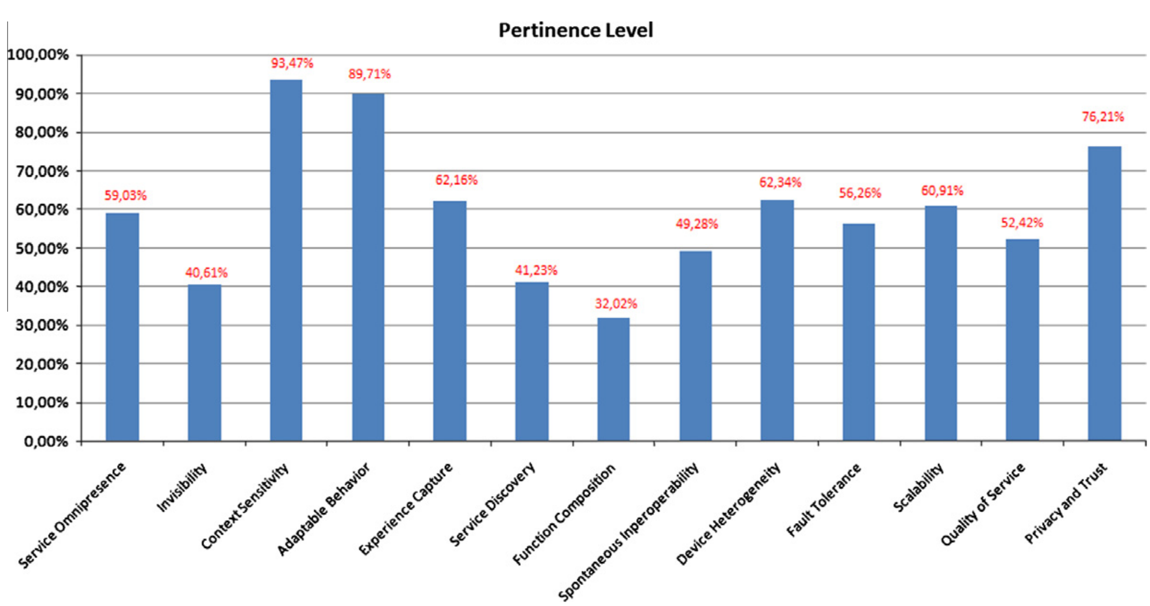


Fig. 7. Level of pertinence of the characteristics.

them hold a Ph.D., 7 are Masters, and 2 undergrads. On average, subjects had already participated in 7 ubiquitous software projects.

As a result, it was possible to identify the pertinence and relevance levels of each UbiComp characteristic.

5.3.1. Analysis of the level of pertinence of the characteristics

Having applied the formula to assess the pertinence of an ubiquitous computing characteristic, we obtained the results shown in Fig. 7. As it can be seen, the lower limit for a characteristic to be considered pertinent is 46.74%. This criterion was used as it is the midpoint for the pertinence scale level (ranging from 0% to 93.47%), following the defined formula for calculating the pertinence level. Thus, the characteristics context sensitivity, adaptive behaviour, service omnipresence, heterogeneity of devices, capture of experience, spontaneous interoperability, scalability, privacy and reliability, fault tolerance, and quality of service were considered pertinent. On the other hand, the characteristics service discovery, invisibility, and composition of functionality were discarded.

Besides, three researchers suggested the inclusion of a new characteristic:

- **Universal usability**: Associated to the fact that project usability is adhering to good usability standards, while considering different target user groups.

Thus, it is observed that the Null Hypothesis 1 (H0 1) was refuted because one UbiComp characteristic suggested by the participants was added and three UbiComp characteristics were excluded from the initial set.

5.3.2. Analysis of the level of relevance of the characteristics

After identifying characteristics considered relevant by the study's subjects, the next step was to define their relevance levels for the characterization of ubiquitous software projects. Thus, having applied the formula for calculating the relevance level shown in Section 5.2, we obtained the results shown in Fig. 8.

Thus, it is seen that Null Hypothesis 2 (H0 2) was also refuted, as there were characteristics with different relevance levels.

As a result, we identified that there are different relevance levels for the UbiComp characteristics when characterizing ubiquitous software projects.

Table 3 summarizes the results obtained:

- The rows marked in grey indicate the characteristics to be considered in the final set of ubiquity characteristics. This selection was done according to the inclusion criteria defined in the survey plan.
- Context sensitivity and adaptable behaviour are the most pertinent and relevant characteristics. It is also important to see that these characteristics have a supplementary relationship.
- We could see that there is a balance between functional (6/11) and restrictive (5/11) characteristics. This can indicate that non-functional characteristics are critical for this software category.

Based on those findings, we could elaborate on our interpretation of the UbiComp definition shown in Section 2 to: *Ubiquitous computing is present when computational services or facilities can materialize themselves at any time or place, transparently, through the use of common daily devices. To make it happen it is desirable that systems of this application category take the following characteristics into consideration*:

- **Functional**: *context sensitivity, adaptable behaviour, service omnipresence, heterogeneity of devices, experience capture, spontaneous interoperability*.

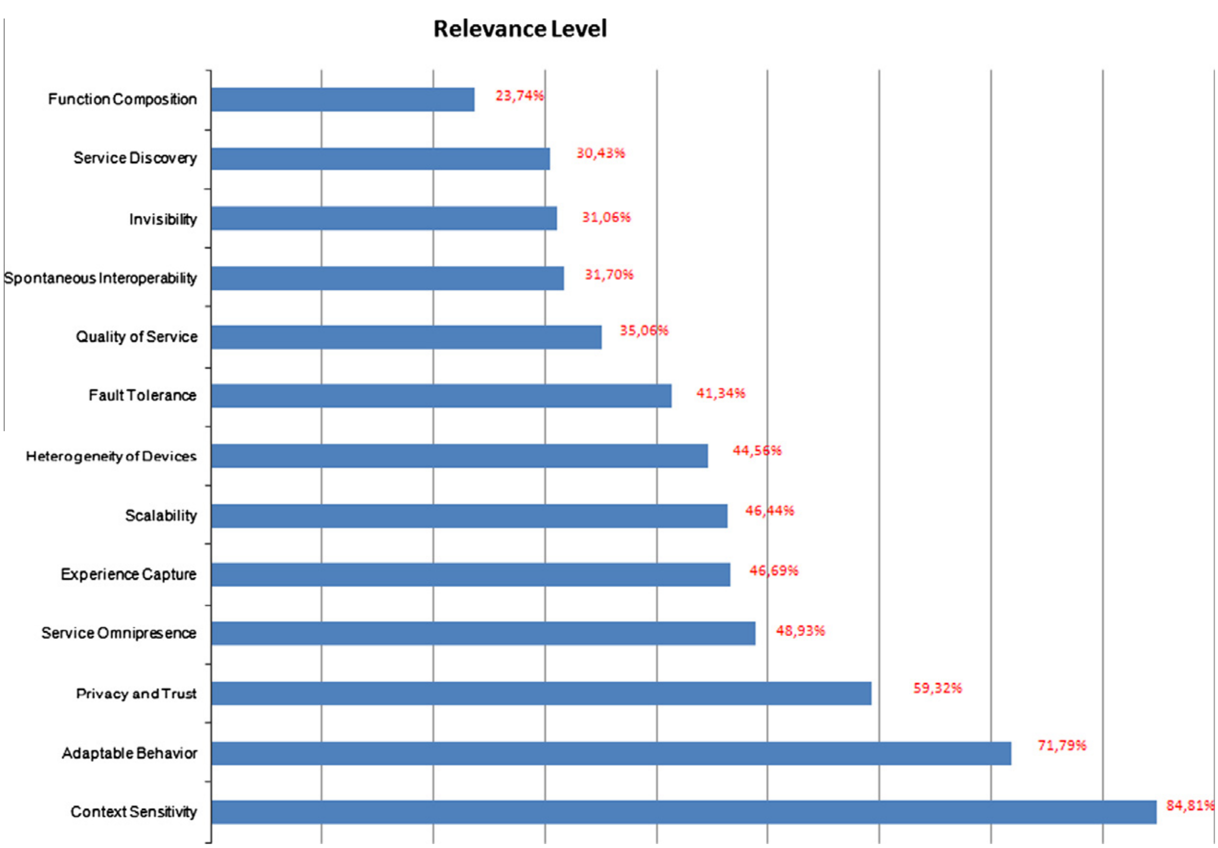


Fig. 8. Characteristics– relevance level.

Table 3
Pertinence and relevance level of characteristics to ubiquity software projects.

Characteristic	Pertinence level (%)	Rank	Relevance level (%)	Rank	Functional/restrictive
Context sensitivity	93.47	1	84.81	1	F
Adaptable behaviour	89.71	2	71.79	2	F
Privacy and trust	76.21	3	59.32	3	R
Heterogeneity of devices	62.34	4	44.56	7	F
Experience capture	62.16	5	46.69	5	F
Scalability	60.91	6	46.44	6	R
Service omnipresence	59.03	7	48.93	4	F
Fault tolerance	56.26	8	41.34	8	R
Quality of service	52.42	9	35.06	9	R
Spontaneous interoperability	49.28	11	31.70	11	F
Universal usability	–			–	R
Service discovery	41.23	12	30.43	14	F
Invisibility	40.61	13	31.06	12	F
Function composition	32.02	14	23.74	14	F

- **Restrictive**: *scalability, privacy and trust, fault tolerance, quality of service, and universal usability*.

These findings allowed us to elaborate on the body of knowledge on UbiComp and its characteristics. Thus, it was possible to organize a body of knowledge regarding UbiComp.

Despite the importance of organizing this body of knowledge, we could not as yet capture how the UbiComp characteristics were applied in practice on ubiquitous software projects. This information can help understanding how the UbiComp characteristics could influence the software project and obtaining insights about what ubiquity characteristics have been usually explored in practice.

Hence, the set of UbiComp characteristics and their factors were used to create a checklist to characterize ubiquitous software projects identified in the technical literature. The proposed checklist and its use is shown in the next Section.

6. Characterizing ubiquitous software projects

Ubiquitous software projects can display different levels of adherence to the ubiquity characteristics and their respective factors. These different adherence levels can be a consequence of the application domain and project requirements, for instance.

Therefore, considering the previously defined concepts can be used to characterize ubiquitous software projects, we have organized a set of procedures, including a checklist, to support the evaluation of the ubiquity adherence level of software projects. It is important to observe that its goal is not to define if a software project is more ubiquitous than others. This characterization can support the understanding of how the ubiquitous computing characteristics have been considered in practice. We believe this can be considered an important step towards to provide some hints and directions to new research trends on Software Engineering applied to ubiquitous software projects.

Basically, the characterization approach consists of four steps:

(1) Identifying the presence of the functional and restrictive factors of each **UbiComp functional characteristic**.
(2) Identifying the presence/absence of **UbiComp restrictive characteristics** based on the project's non-functional requirements. In this case, there are two possible values for the adherence level: 100% (presence) or 0% (absence).
(3) Assessing the adherence level of each UbiComp characteristic for the software project based on the presence/absence of each correspondent functional and restrictive factor.
(4) Representing the ubiquity adherence level for the system through the use of the values obtained in Step 3 to generate a graph.

The previously described steps can be error prone whether manually executed; therefore we developed a spreadsheet-based form to support the calculation of the adherence level for each

Characteristic	Characteristic Adherence Level	Factor Group	Factor	Status
Service omnipresence	70%	Mobility	User section management	✓
			To deal with the user's mobility	✗
			When moving the services, these should continue operand starting from the point that it processing was interrupted for the migration of the functionality	✓
			To support the mobility among domains and inside of a same domain	✗
		Service management	Each device should contain a container to allocate services	✓
			Each device should manage the services allocated in it container	✗
			To organize services according to the context	✓
		Service publish	To publish the existence of the service for other devices / applications	✓
			To maintain the registry of services published in cache to increase the performance in a new services publish	✓
		Restrictive Factor	Each service should be generic enough to continue executing while alterations happen in the environment	✓

Fig. 9. A checklist fragment to characterize ubiquitous software projects.

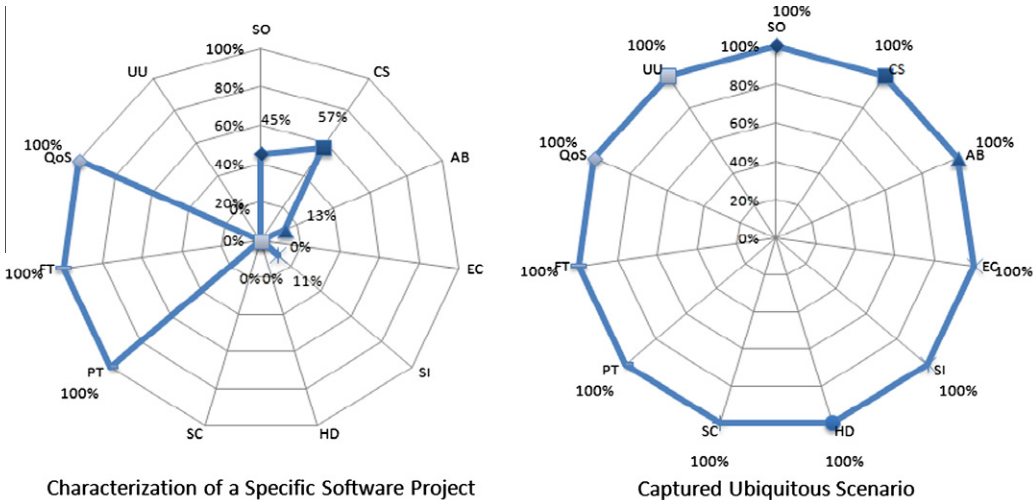


Fig. 10. Example of ubiquity characteristics and their adherence levels.

ubiquity characteristic. Fig. 9 shows a fragment of the proposed checklist that comprises the following columns:

- Characteristic: shows the final set of UbiComp characteristics shown in Section 5.
- Characteristic Adherence Level: shows the percentage of adherence based on the Status column. It is important to notice that each factor has the same weight and the adherence level is calculated as the average of the attended factors. The calculus is given by the expression bellow:

$$\text{Characteristic Adherence Level} = \frac{\sum \text{attended factors} \times 100}{\text{Number of factors}}$$

where:

- ○ Attended factors are the factors whose status value is 1 for a specific characteristic.
- ○ Number of factors is the total number of identified factors for a specific characteristic.
- ○ Factors: shows the functional and restrictive factors identified on the second review shown in Section 3.
- Factor Group: shows the factor groups identified on the second literature review shown in Section 3.
- Factors: shows the functional and restrictive factors identified on the second review shown in Section 3.
- Status: factor presence (1) or absence (0). The user provides this information.

Basically, as the user fills in the Status column, the Characteristic Adherence Level columns can be calculated for each ubiquitous computing characteristic. In the final step, the evaluated percentage values of the Characteristics Adherence Level columns are used to draw a graph representing the software project ubiquity adherence level. For instance, Fig. 10 (left graph) represents the observed ubiquity characteristics when applying this checklist to [116]. We can see that a real ubiquitous software project (left graph) can differ from an expected (right graph) full ubiquitous scenario (Sections 2, 3, and 5).

6.1. Applying the characterization checklist

This Section provides the results from applying the characterization approach to a set of 26 ubiquitous software projects found in technical literature that represent examples ranging from 2004 to 2010 where some technological evolution took place. Those software projects were selected in order to represent four UbiComp domain areas:

- Ambient intelligence: refers to electronic environments that are sensitive and responsive to the presence of people. The ambient intelligence area builds upon ubiquitous computing and is marked by systems and technologies that are [1]: embedded, context-aware, personalized, adaptive, and anticipatory.
- Pervasive Healthcare: may be defined from two perspectives: i) as the application of pervasive computing technologies for healthcare, and ii) as making healthcare available everywhere, anytime and to anyone. Thus, pervasive healthcare is closely related to Biomedical Engineering, Medical Information Systems, and Ubiquitous Computing [122].
- Ubiquitous learning (or U-learning): is equivalent to some form of simple mobile learning, e.g., learning environments that can be accessed in various contexts and situations. Besides the domains of e-Learning, U-learning may use more context awareness to provide most adaptive contents for learners [66].
- Urban spaces: computing devices are everywhere and our everyday life is undeniably linked to several of these. Mobile phones, PDAs, media players or laptops are the indispensable companions of the urban dweller. Beyond our controlled gadgets, myriads of devices require and expect our interaction in an increasingly networked urban environment. In order to describe these complex networked environments we use the notion of 'urban spaces' [36].

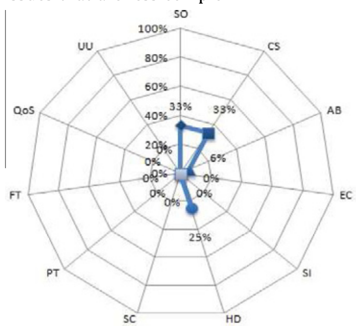
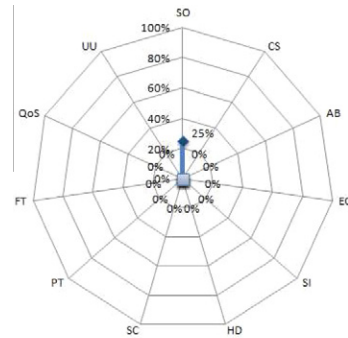
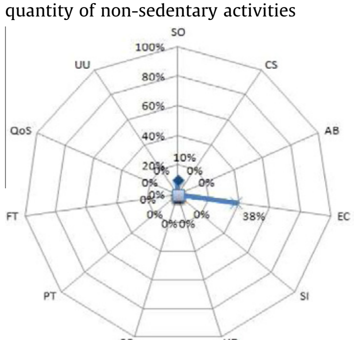
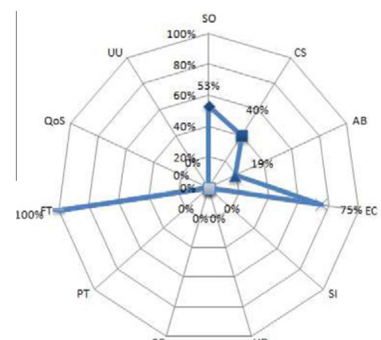
The importance in considering those areas lies in the fact that:

- They represent different perspectives of how to apply the concepts of ubiquitous computing in real software systems.
- Each area brings specific challenges and requirements associated with the ubiquitous computing domain.

Thus, we believe that this analysis will allow us to take a broad view of how far ubiquitous computing principles and ubiquitous software projects are each other in general and for each domain area.

For each one of the selected software projects, we conducted the characterization as exemplified in the previous Section. The results are shown on Table 4 (for ambient intelligence area), 5 (for

Table 4
Description of the analyzed software projects for Ambient Intelligence area.

Reference/year	Project 01: [5]/2004	Project 02: [31]/2005
Description	On this project, named U-Kitchen, some services in the context of a kitchen could be made available using concepts of ubiquitous computation. The scenario is made using some smart devices communicating to each other and sharing the context via a kitchen server. By adding a limited level of smartness to the appliances, ubiquitous services are developed that make the job for an autonomous ubiquitous system far more easier, and technical issues that are less complex	In this project, named GENIO, a real kitchen and sitting room have been built where users can command the home talking naturally. Possible actions are: reading emails, programming the washing machine, checking goods in the fridge, creating a shopping list, doing shopping with a PDA in the supermarket, activating the dishwasher, being guided on how to prepare an oven recipe, checking if the ingredients are in the larder, listening to some music stored at home, seeing photos, and so on
Ubiquity adherence level		
Reference/year	Project 03: [73]/2006	Project 04: [115]/2006
Description	Project ViTo functions as a universal remote control for a home entertainment system. The interface of this device, however, is designed in such a way that it may unobtrusively promote a reduction in the user's television viewing while encouraging an increase in the frequency and quantity of non-sedentary activities	This project proposes the use a context sensitive and proactive fuzzy control system for controlling the home environment. The designed control system is adaptive, and it can accommodate changing conditions of its dwellers. Thus, the system operates fully in the background and needs very little effort from its users
Ubiquitous adherence level		

(*continued on next page*)

Table 4 (*continued*)

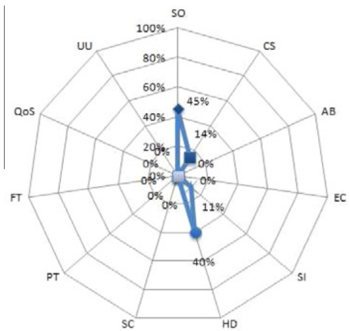
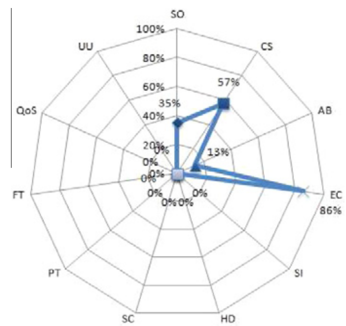
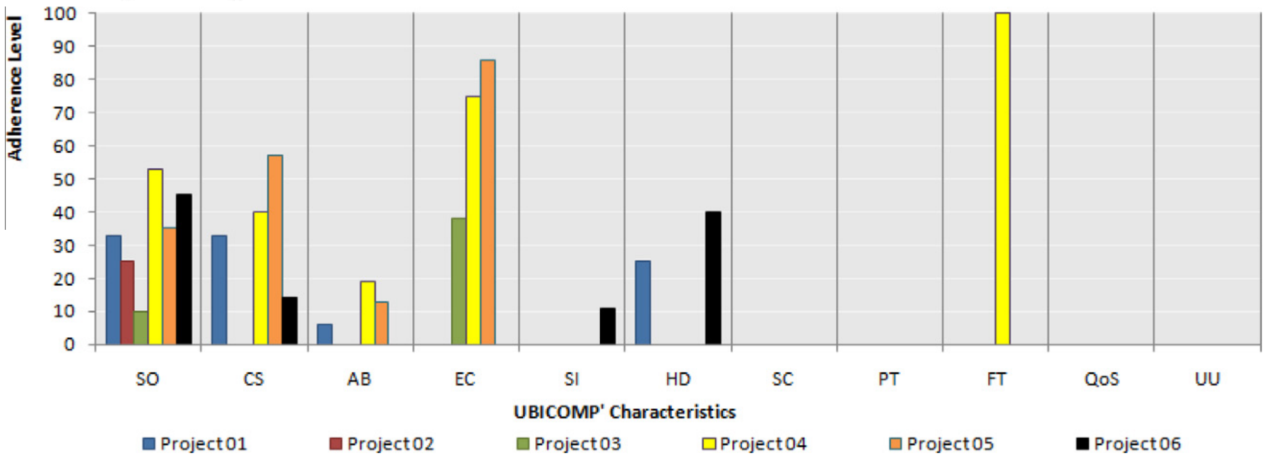
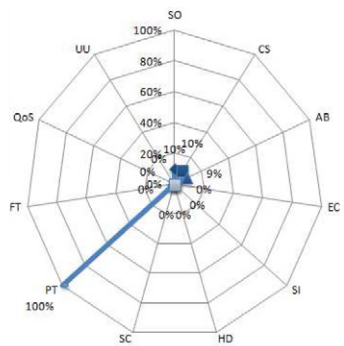
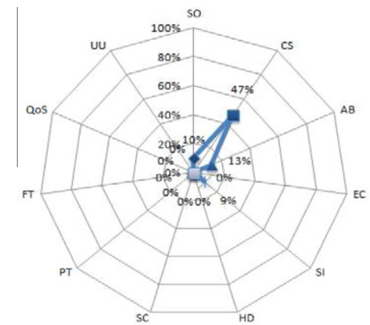
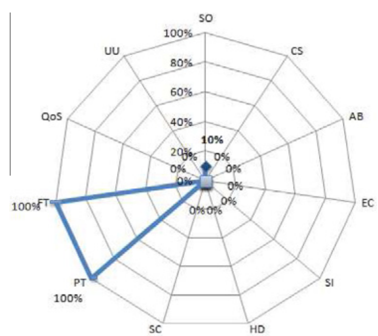
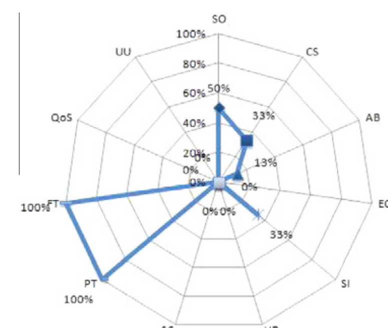
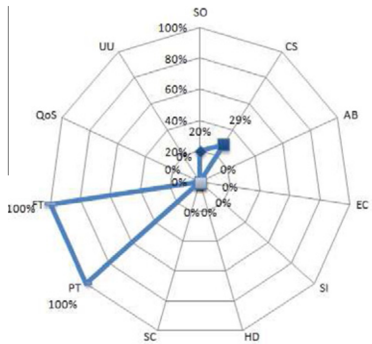
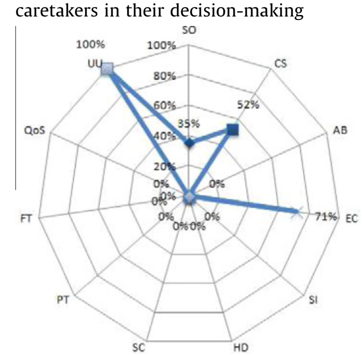
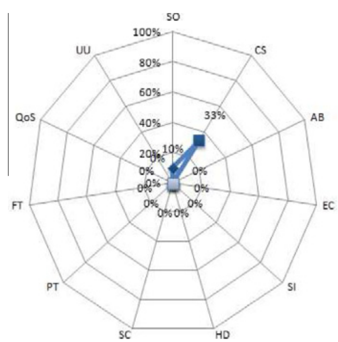
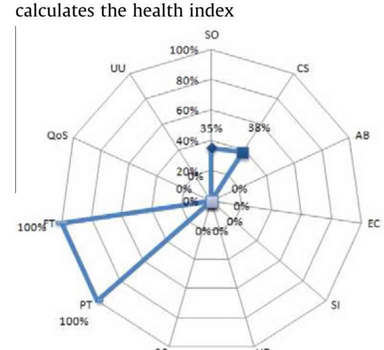
Reference/year	Project 01: [5]/2004	Project 02: [31]/2005
Reference/year	Project 05: [64]/2008	Project 06: [129]/2010
Description	This project proposes an agent-based management system for ubiquitous smart homes that provides an user-friendly interface, and can make use of add-on functionalities of the IMS such as voice chatting, SMS, and multimedia	On this project, a case-driven ambient intelligence system is proposed, aiming at sensing, predicting, reasoning, and acting in response to daily elderly person activities at home
Ubiquitous adherence level		
		

Table 5
Description of the analyzed software projects for pervasive healthcare area.

Reference/year	Project 01: [9]/2004	Project 02: [63]/2004
Description	This project's goal is to construct a software that provides easy access to mobile electronic registration of patients (EPR)	This software project intends to develop a set of applications regarding the concepts of the ubiquitous computing in the context of healthcare, environment control, and management
Ubiquity adherence level		
Reference/year	Project 03: [35]/2006	Project 04: [111]/2006
Description	In this project, the authors envisioned the design and implementation of a highly-automated AMI Teleconsultation & Monitoring System (AToMS) that relies on mobile communications to assist emergency teams and cardiologists to exchange information about an AMI patient and decide on its eligibility for receiving thrombolytics in a timely fashion, and to track patient evolution while one is being moved to the nearest available CCU	In this project, the authors propose a pervasive computing (PC) infrastructure for the delivery of medical services. This system is generic and can be easily adapted to meet some specific needs of a hospital or clinic.The system consists of three subsystems: (1) A pervasive patients' system (2) A subsystem for automated diagnostics and administration of patients' medication, and (3) an automated prescription subsystem
Ubiquitous adherence level		

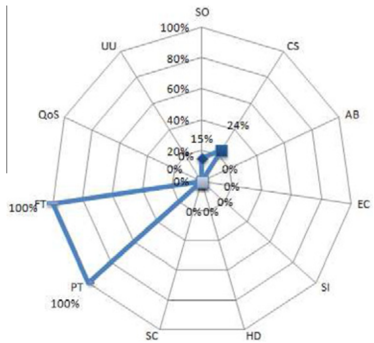
(*continued on next page*)

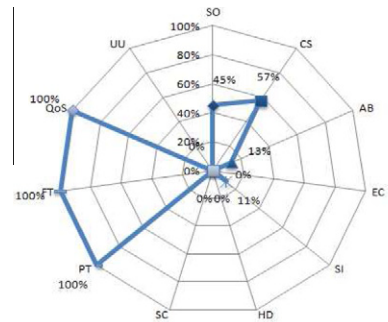
Table 5 (*continued*)

Reference/year	Project 01: [9]/2004	Project 02: [63]/2004
Reference/year	Project 05: [81]/2007	Project 06: [58]/2007
Description	In this project a system is proposed to support the control of Glucose levels. For the system implementation, they used the MIMOSA framework shown on the paper	In this project, a set of support systems is shown, for autistic children. The system consists of patient monitoring functionalities and, at the same time, it assists caretakers in their decision-making
Ubiquitous adherence level		
Reference/year	Project 07: [99]/2007	Project 08: [79]/2009
Description	Shin et al. (2007) suggest a healthcare monitoring system using unconstrained measurement devices with ubiquitous techniques. These measurement devices are developed into built-in type and sensor type. The first devices are built-in the households (bed, sofa, and toilet seat) and measure patient heart rate, breathing patterns, and estimate blood pressure. The second devices are placed in a kitchen, front door, and every room, to detect patient movements and activities. All digitized raw signals are sent to a hospital laboratory after an analysis process	Lee e Park (2009) developed a u-Healthcare aide system as a smart space that helps health management users to manage their health with precautionary measures. This system consists of three modules: Sensing, Management, and Analysis. The Sensing Module is in charge of measuring, transferring, and receiving vital data. The Management Module helps offering services to users, medical workers, and managers. This module supports inquiry and analysis and offers results, meal and exercise prescriptions, and medical pre-examinations. The Analysis Module analyzes and predicts diseases and calculates the health index
Ubiquitous adherence level		

Reference/year	Project 09: [108]/2009	Project 10: [116]/2010
Description	Su et al. (2009) developed an ubiquitous healthcare (u-Health) system platform that integrates wireless telecommunication, sensor network, and information technology to take care of patients having chronic diseases. The u-Health system provides not only telecare in the home setting, but also a health community network that can further integrate medical care and life care and emphasize patient humanity and dignity	In this project, the authors describe a policy-based architecture that uses wireless sensor devices, advanced network topologies and software agents to enable remote monitoring of patients and elderly people. Through those technologies the authors achieve continuous monitoring of a patient's condition and can proceed when necessary with proper actions

Ubiquitous adherence level





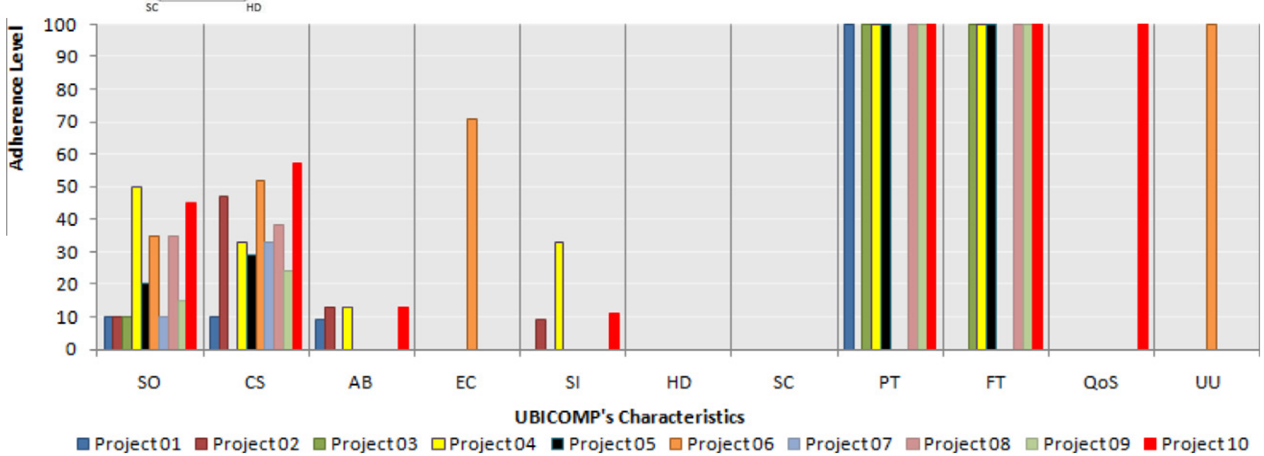
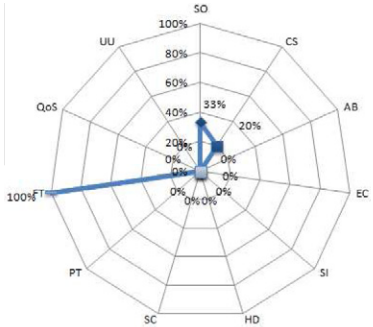
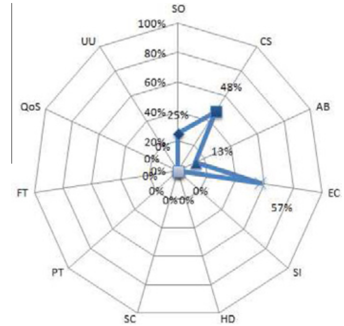
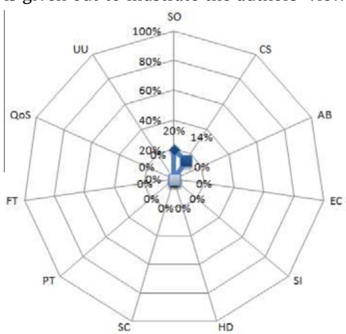
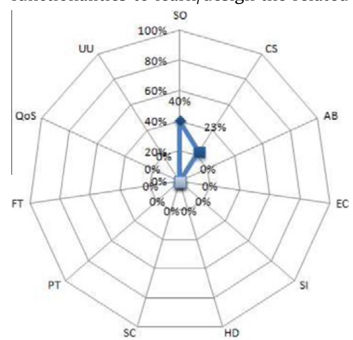


Table 6
Description of the analyzed software projects for U-learning area.

Reference/year	Project 01: [48]/2004	Project 02: [128]/2008
Description	The research described in this paper is the investigation of interactivity between learners and system in the context of remote access to educational field explorations. Thus, the proposed software project intends to develop software to support distance-based education	In this paper, the authors propose an ubiquitous learning system using co-learning to enhance the education effect. They created a virtual pet referred as ubiquitous pet that can progress with a user through its own progress attributes
Ubiquity adherence level		
Reference/year	Project 03: [47]/2008	Project 04: [123]/2009
Description	In this paper, the concept of Digital Artefacts and U-learning Digital Artefacts are given out. Two examples of U-learning Digital Artefacts, an ubiquitous schoolbag and a digital flash card, are shown. Moreover, an ubiquitous review system for the pupils, using digital artefacts, is given out to illustrate the authors' view on ubiquitous learning	Yang et al. (2009) proposed a Web-based outdoor experience game-based learning system by integrating the Geography Information System (GIS), Global Position System (GPS) and wireless connection technologies. Both learners and instructors could easily use the proposed system functionalities to learn/design the related outdoor experience course content
Ubiquitous adherence level		

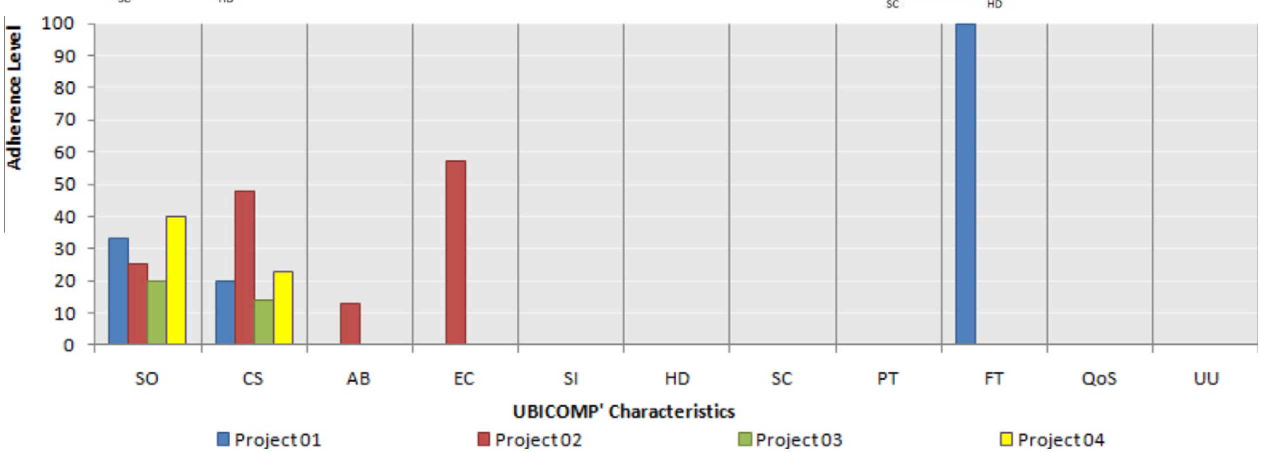
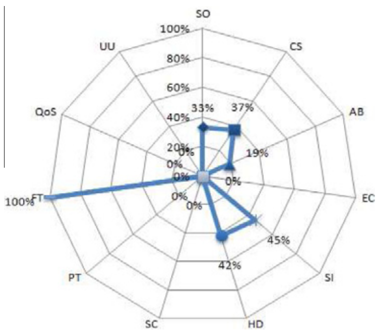
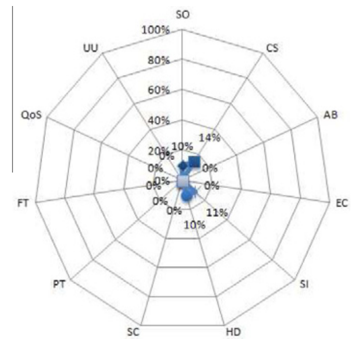
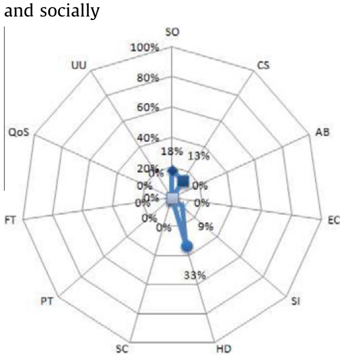
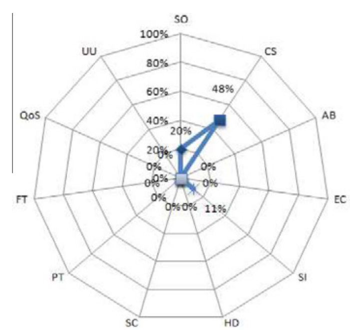
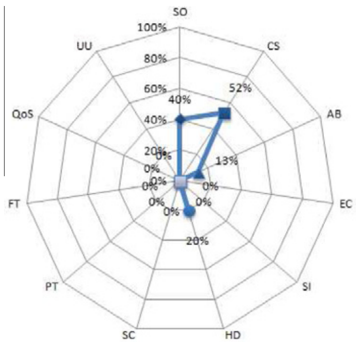
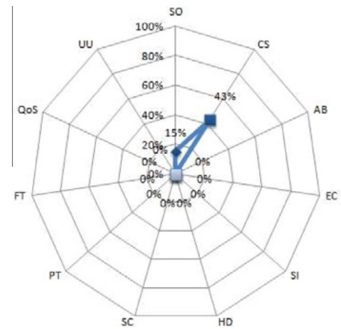


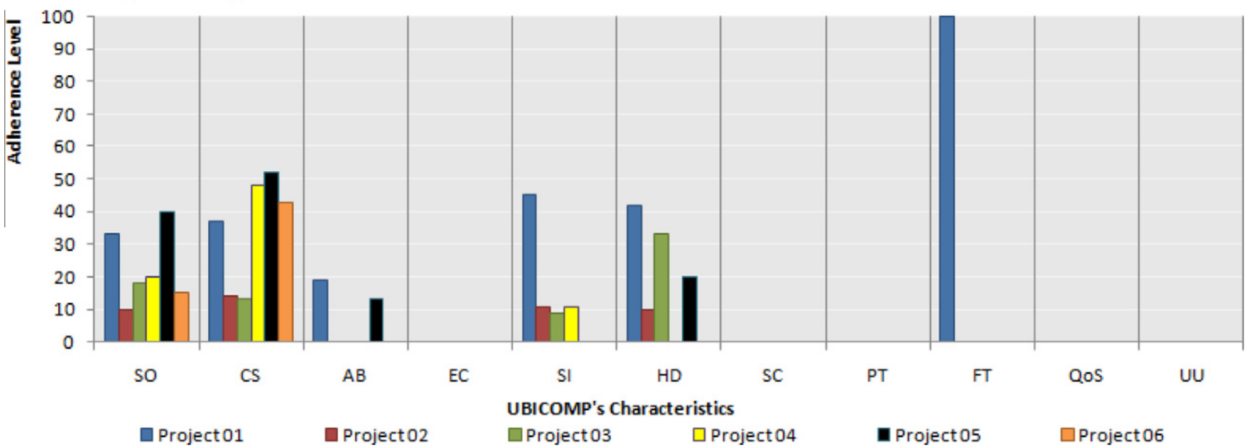
Table 7
Description of the analyzed software projects for urban space area.

Reference/year	Project 01: [130]/2005	Project 02: [20]/2005
Description	EZCab is an ubiquitous application that allows people to call to taxis that are close and available, using mobile phones or PDAs. The system discovers and calls available taxis	This paper presents a novel idea for a system known as Telelogs. Using the ubiquity of mobile devices, Telelogs functions as a service with which individuals in an urban environment can establish a better sense of community awareness
Ubiquity adherence level		
Reference/year	Project 03: [77]/2006	Project 04: [74]/2009
Description	To understand the city as a system encompassing physical and digital forms and their relationships with people's behaviours, this paper presents the developed, applied, and refined methods of observing, recording, modelling and analyzing the city, physically, digitally and socially	In this project, the authors introduce the PlaceAware application that can be used to enhance people's social interactions in urban areas
Ubiquitous adherence level		

(*continued on next page*)

Table 7 (*continued*)

Reference/year	Project 01: [130]/2005	Project 02: [20]/2005
Reference/year	Project 05: [13]/2009	Project 06: [69]/2009
Description	The authors present the general design of an architecture, based on software agents and oriented to the semantic Web, for the development and deployment of urban, ubiquitous services for citizens and tourist. The goal is to create a platform capable of providing personalized services based on recommendation algorithms, and user location, profile, and preferences	In this project, a new approach to monitor noise pollution involving citizens and built on the notions of participatory sensing and citizen science. We enable citizens to measure their personal exposure to noise in their everyday environment by using GPS-equipped mobile phones as noise sensors. The geo-localized measures and user-generated meta-data can be automatically sent and shared online with the public to contribute to the collective noise mapping of cities
Ubiquitous adherence level		



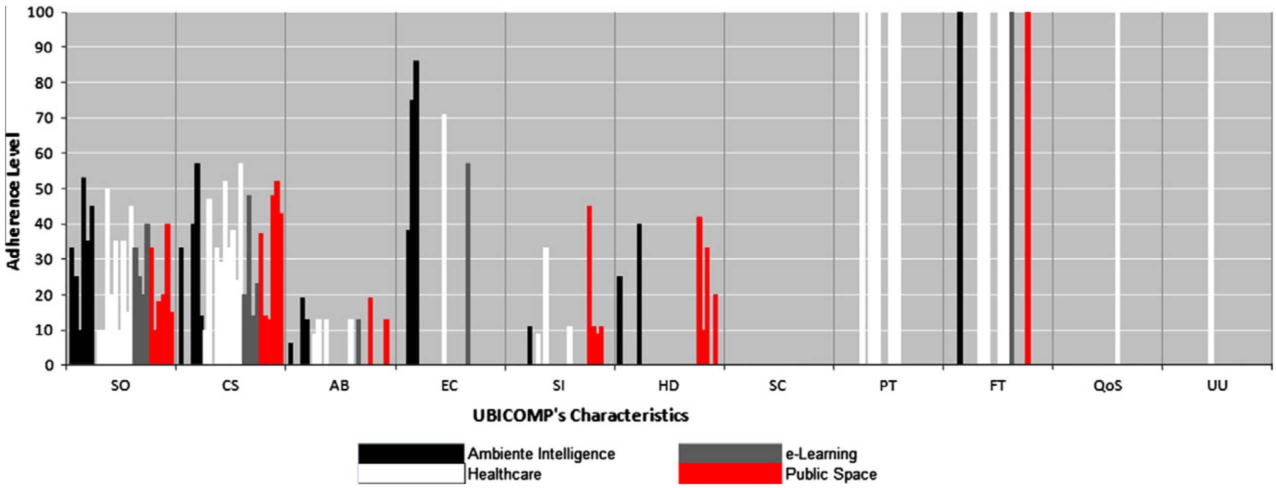


Fig. 11. View of all characterized ubiquitous software projects grouped by software project category.

pervasive healthcare area), 6 (for U-learning area), and 7 (for urban space area). For each graph, the following caption should be considered: **SO** – Service Omnipresence, **CS** – Context Sensitivity, **AB** – Adaptable Behaviour, **EC** – Experience Capture, **SI** – Spontaneous Interoperability, **HD** – Heterogeneity of Devices, **SC** – Scalability, **PT** – Privacy and Trust, **FT** – Fault Tolerance, **QoS** – Quality of Service, **UU** – Universal Usability. At the end of each Table, a graph is provided to summarize the data from all analyzed projects.

As it can be seen in Table 4, from an UbiComp perspective, the selected ambient intelligence systems are marked by the presence of service omnipresence, context sensitivity, adaptable behaviour, and experience capture of UbiComp characteristics. These results are aligned with the ambient intelligence systems' nature [1].

For the pervasive healthcare area, we could see, after observing the results of the characterization of 10 healthcare systems, as shown in Table 5, that this area is marked by the following UbiComp characteristics: service omnipresence, context sensitivity, adaptable behaviour, privacy and trust, and fault tolerance.

In Table 6 we can see that, from the UbiComp perspective, the selected U-learning systems are marked by the presence of service omnipresence and context sensitivity characteristics.

And finally, by observing the results of the characterization of 6 urban space systems shown in Table 7, we could notice that this software category is marked by the following UbiComp characteristics: service omnipresence, context sensitivity, adaptable behaviour, spontaneous interoperability, and heterogeneity of devices.

6.2. Discussion

In this subsection, we will put together the set of characterized ubiquitous software projects described in the prior Section and the survey results regarding the relevance and pertinence of each UbiComp characteristic, as described in Section 5, to analyze the distance between ubiquitous computing principles and ubiquitous software projects.

Fig. 11 shows all characterized ubiquitous software projects grouped by application domain areas in the same order as shown in Tables 4–7. Some interesting trends can be observed from the analyzed data:

- All analyzed software projects consider the Service Omnipresence UbiComp characteristic. This is an important indication that this characteristic should be considered in ubiquitous software projects.
- Almost all software projects consider the Context Sensitivity UbiComp characteristic. This fact is also aligned with the nature of ubiquitous computing in which systems are interacting with the environment (collecting data) in order to provide some service to the user.
- As a result of the two points above, the application domain area (ambient intelligence, pervasive healthcare, U-learning, urban space) does not seem to affect the presence of the characteristics service omnipresence and context sensitivity.
- For the other characteristics, the software project category affects clearly the presence/absence of some UbiComp characteristics. For instance:
  - Pervasive healthcare systems seem to be strongly associated with the following characteristics: service omnipresence, context sensitivity, privacy and trust, and fault tolerance.
  - Ambient intelligence systems are most of the time associated with service omnipresence, context sensitivity, and experience capture (due to the fact that in this UbiComp domain area it is important to understand user behaviour to provide customized services for their characteristics.
  - For urban space systems, besides service omnipresence and context sensitivity characteristics, this domain area is also associated with spontaneous interoperability and device heterogeneity characteristics. This connection makes sense as we usually have a large and heterogeneous number of devices where the system is deployed in urban space systems.
  - U-learning systems are generally marked by the presence of service omnipresence and context sensitivity characteristics.
- Apart from the fact of being considered pertinent, three restrictive characteristics were not considered or were considered in only one software project: scalability, quality of service, and universal usability. In our opinion, this behaviour could be explained by the fact that most of selected software projects were in an initial development stage, where only the system's functionalities had been clearly defined.

7. Threats to validity

Threats to validity are an important concern when performing primary and secondary studies. Throughout this paper four studies were presented. Therefore, some issues must be considered for each performed study. The two systematic reviews (secondary studies) will be discussed together because they hold similar threats to validity.

7.1. Identifying UbiComp characteristics (discussed on Section 2) and factors in UbiComp characteristics (discussed on Section 3)

The main limitations of these secondary studies are concerned with:

- **Publication selection bias**: publication bias refers to the issue that positive results are more likely to be published than negative ones. We can not deal with this. Our decision was to consider only papers published in journals or conference proceedings. Thus, we did not consider gray literature, unpublished results or not peer reviewed material. About the selections of publications, we chose the sources where papers concerned with ubiquitous computing are usually published.
- **Inaccuracy in data extraction and misclassification**: we attempted to alleviate the threats of inaccuracy in data extraction and misclassification by conducting the papers classifications with three reviewers and applying peer review on the extracted information.

7.2. Initial survey (discussed on Section 4)

Valid survey information comes from reliable and valid survey instruments and from the context in which the survey takes place [27]. With regards to the survey presented in Section 4, the following threats to validity can be considered:

- **Selection of participants**: to avoid bias on participant selection, the ideal condition is to have subjects that are randomly selected and associated to their activities. However, it requires vast resources that almost always are difficult and costly to implement. Thus, sometimes we need to limit the considered participants in the study population. On this initial survey, the study population was limited to academic researchers in Brazil, not including industry participants. This decision was taken due to time constraints. For this reason, the reached results were used just as initial evidence about the validity of knowledge organized until that time. Additionally, we considered to perform a more comprehensive survey involving participants from academia and industry to deal with this threat.
- **Instrumentation**: to avoid mistakes with the questionnaire, it was reviewed by three independent researchers that did not take part in the survey execution. Based on the points highlighted by the reviewers, the questionnaire was adjusted and its evolved version was used during the survey execution. Thus, we tried to reduce any possible misunderstanding regarding the questionnaire's contents.
- **Inaccuracy in data extraction**: we attempted to minimize the threats of inaccuracy in data extraction by conducting the data analyzes of the questionnaires with two reviewers (a researcher extracted and summarized the results, and the other one reviewed in detail the reported results).

Thus, as described on subsection 4.3 and based on the aforementioned threats to validity, despite the fact the population size was small and could be not representative considering the global UbiComp scenario, the results contributed for the evolution of the organized body of knowledge. Thus, the result of the survey was not used as definitive to evaluate the body of knowledge, although its results were important to an initial evolution. Therefore, a second survey was performed and its threats to validity are going to be discussed in the next subsection.

7.3. Evaluating the UbiComp body of knowledge (discussed on Section 5)

With regards to the survey presented in Section 5, the following threats to validity can be considered:

- **Selection of participants**: in order to reduce bias in the selection of participants, the researchers defined as criterion to include academic or industry researchers that published papers: (i) identified by the two *quasi*-systematic reviews (Sections 2 and 3), and; (ii) in the proceedings of UBICOMP conference. Despite the fact this criterion could not be considered unbiased, it allowed us to consider a more comprehensive number of participants in terms of academic and industrial researchers. Thus, we assumed this population could be somehow representative in the context of UbiComp. This was the only viable option for us at that time. Nevertheless, the number of participants that answered the questionnaire was not large (about 11%) enough. However, it usually occurs on survey executions. Thus, we believe that, although we could not completely eliminate this threat, the reached results could be used to support the evaluation of the organized body of knowledge concerned with UbiComp.
- **Instrumentation**: to avoid mistakes with the questions, an online questionnaire was developed and published in the Internet. Moreover, this questionnaire was also reviewed by three independent researchers (they were not considered as survey population). Based on the points highlighted by the reviewers, the questionnaire was adjusted and the new version was used during the survey execution. Thus, we tried to avoid any misunderstanding regarding the questionnaire contents.
- **Inaccuracy in data extraction**: for this survey, we also attempted to minimize the threats of inaccuracy in data extraction by conducting the data analyzes of the questionnaires with two reviewers (a researcher extracted and summarized the results, and the other one reviewed in details the reported results).

8. Conclusions

In this paper, the organization and evaluation of UbiComp characteristics and their factors with the use of a research strategy based on primary and secondary studies was described. From this point of view, this research strategy allowed us to reach some results:

(1) 1st and 2nd *quasi*-Systematic Reviews: a more recent definition for UbiComp and its characteristics, and identification of functional and restrictive factors for each UbiComp characteristic;
(2) Initial Survey: improvement of the body of knowledge considering functional and restrictive perspectives and 3 new characteristics;
(3) Body of Knowledge Evaluation: evaluation and improvement of the body of knowledge through the definition of pertinence and relevance of UbiComp characteristics.

As other engineering disciplines, software engineering needs to take into account different domains in which it is working. Different domains could require, for instance, different techniques, processes, and tools. Moreover, the development of software projects for specific domains can be considered a big challenge due to the difficulty on understanding and manipulating specific concepts and their relationships [76]. Thus, the organization of this body of knowledge can be considered an important step once domain knowledge can reveal concepts, descriptions, and relations that could be organized to show what need be analyzed on each software development activity.

Apart from that, this body of knowledge was structured into a checklist to support the characterization of ubiquitous software projects. This checklist was used to characterize 26 ubiquitous software projects from different domains: ambient intelligence, pervasive healthcare, U-learning, and urban spaces. This characterization allowed us to have some insights on how far ubiquitous computing principles and ubiquitous software projects are each other. Additionally, it was also possible to notice that the application domain area affects the presence/absence of UbiComp characteristics.

The applicability of the presented UbiComp body of knowledge can be exemplified by two software engineering technologies in the requirements engineering area that were developed and empirically evaluated: *Ubicheck* and *UbiVeri*. They represent checklist-based approaches to support respectively the requirements definition and verification in the UbiComp domain area [102,103]. However, it is important to note that the use of the organized body of knowledge is not limited for this type of software technology. We hope the organized body of knowledge can support the initial discussions towards dealing with some additional software engineering research challenges involving the management, planning, specification, designing, implementation, and testing of ubiquitous software projects, such as [102]:

- Process definition
  - What should be the activities added into the software development process to support working with ubiquitous computing characteristics?
- Project Planning
  - What are the risks associated with each ubiquitous computing characteristic?
  - How to reduce the risks associated with the development of ubiquitous software projects?
- Requirements
  - How to support the requirements elicitation in ubiquitous software projects?
  - What is the influence of an UbiComp characteristic in the activities regarding elicitation and specification of requirements?
  - Which are the most feasible approaches to support the verification of requirements concerned with UbiComp characteristics?
- Design
  - What are the impacts of ubiquitous computing characteristics in the software architecture?
  - How to assess the quality of the designed ubiquitous software architecture?
- Implementation
  - What coding technology or set of technologies should be used to implement an ubiquitous software project?
- Testing
  - How to choose the most feasible testing approaches to test such software project?

We believe that many other questions can be added by the readers to our initial list. Thus, we hope that the organization of the UbiComp body of knowledge and the characterization of ubiquitous software projects are an important step in order to provide some hints and directions to new research trends regarding Software Engineering applied to ubiquitous software projects.

Acknowledgements

Authors thank to CNPq (Grant 75459/2007-5), CAPES, and FAPERJ for their financial support. We would like to acknowledge the participation of Jobson Massolar da Silva and Felipe Rego Pinto from the Experimental Software Engineering group at COPPE/UFRJ in some activities regarding this research project. Our recognition to the survey participants which contributed to the results presented in this paper.

References

[1] Emile Aarts, Rick Harwig, Martin Schuurmans, Chapter "Ambient Intelligence" in The Invisible Future: The Seamless Integration Of Technology Into Everyday Life, McGraw-Hill Companies, 2001.
[2] G.D. Abowd, E.D. Mynatt, T. Rodden, The human experience, IEEE Pervasive Computing Archive 1 (1) (2002) 48–57.
[3] G.D. Abowd, Software engineering issues for ubiquitous computing, in: Proceedings of the 21st International Conference on Software Engineering, 1999, pp. 75–84.
[4] G. Acampora, V. Loia, Fuzzy control interoperability and scalability for adaptive domotic framework, IEEE Transactions on Industrial Informatics 1 (2) (2005).
[5] J.A. Ali, Y. Won-Sik, K. Jai-Hoon, C. We-Duke, U-kitchen: application scenario, in: Proceedings of the Second IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, 2004, pp. 169–171.
[6] A.A. Araya, Questioning ubiquitous computing, in: Proceedings of the ACM 23rd Annual Conference on Computer Science, 1995.
[7] G. Banavar, A. Bernstein, Software infrastructure and design challenges for ubiquitous computing applications, Communications of the ACM 45 (12) (2002) 92–96.
[8] J. Biolchini, P.G. Mian, A.C.C. Natali, G.H. Travassos, Systematic Review in Software Engineering. Technical Report ES 679/05, COPPE/UFRJ, 2005.
[9] C. Bossen, J.B. Jorgensen, Context-descriptive prototypes and their application to medicine administration, in: Proceedings of the 2004 Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques, 2004, pp. 297–306.
[10] A.H. Bunningen, L. Feng, P.M.G. Apers, Context for Ubiquitous Data Management, UDM 2005, 2005, pp. 17–24.
[11] H.E. Byun, K. Cheverst, Utilizing context history to provide dynamic adaptations, Applied Artificial Intelligence 18 (6) (2004) 533–548.
[12] C. Campo, M. Munoz, J.C. Perea, A. Marin, C. Garcia-Rubio, PDP and GSDL: a new service discovery middleware to support spontaneous interactions in pervasive systems, in: PerCom Workshops 2005, 2005, pp. 178–182.
[13] L. Ceccaroni, V. Codina, M. Palau, M. Pous, PaTac: Urban, ubiquitous, personalized services for citizens and tourists. Third International Conference on Digital Society, 2009. ICDS '09, 2009, pp. 7–12.
[14] H. Cervantes, R.S. Hall, Autonomous adaptation to dynamic availability using a service-oriented component model, in: 26th International Conference on Software Engineering (ICSE'04), 2004, pp. 614–623.
[15] O. Conclan, R. Power, S. Higel, D. O'Sullivan, K. Barrett, next generation context aware adaptive services, in: Proceedings of the 1st International Symposium on Information and Communication Technologies, Dublin, Ireland September 24–26, 2003, pp. 205–212.
[16] A. Corradi, R. Montanari, D. Tibaldi, Context-based access control for ubiquitous service provisioning, in: 28th Annual International Computer Software and Applications Conference (COMPSAC'04), 2004, pp. 444–451.
[17] A. Corradi, R. Montanari, D. Tibaldi, A. Toninelli, A context-centric security middleware for service provisioning in pervasive computing, in: SAINT 2005, 2005, pp. 421–429.
[18] P. Couder, M. Banâtre, Ambient computing applications: an experience with the SPREAD approach, in: 36th Hawaii International Conference on System Sciences, 2003.
[19] O. Davidyuk, J. Riekki, V.M. Rautio, J. Sun, Context-aware middleware for mobile multimedia applications, in: Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia (MUM '04), 2004.
[20] B. Davis, K. Karahalios, Telelogs: a social communication space for urban environments, in: Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services, 2005, pp. 231–234.
[21] P. Debaty, A toolkit to design adaptable user interfaces in ubiquitous computing environments, PerCom Workshops 2004, 2004, pp. 171–175.
[22] A.K. Dey, G.D. Abowd, A. Wood, CyberDesk: a framework for providing self-integrating context-aware services, Intelligent User Interfaces (1998) 47–54.
[23] A.C. Dias Neto, R. Spínola, G.H. Travassos, Developing software technologies through experimentation: experiences from the battlefield, in: Proc. of XIII

Congresso Iberoamericano en "Software Engineering" (CIBSE), Universidad del Azuay, Cuenca, Equador, vol. I, 2010, pp. 107–121.
[24] T. Disx, M.E. Papka, R. Stevens, UbiWorld: an environment integrating virtual reality, supercomputing, and design, in: Proceedings of the High Performance Distributed Computing (HPDC '96), 1996.
[25] T. Disx, M.E. Papka, R. Stevens, UbiWorld: an environment integrating virtual reality, supercomputing, and design, in: Proceedings of the 6th Heterogeneous Computing Workshop, HCW'97, Geneva, Switz, 1997.
[26] T. Dybå, T. Dingsøyr, Strength of evidence in systematic reviews in software engineering, in: ESEM'2008, Kaiserslautern, October 2008, pp. 178–187.
[27] A. Fink, J. Kosecoff, How To Conduct Surveys: A Step-by-Step Guide, second ed., Sage Publications, Inc., 1998.
[28] W. Fontijn, P. Boncz, AmbientDB: P2P data management middleware for ambient intelligence, in: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications, Workshops, PerCom 2004, 2004, pp. 203–207.
[29] K. Fujii, T. Suda, Dynamic Service Composition Using Semantic Information ICSOC'04, November 15–19, 2004, New York, New York, USA, 2004.
[30] K. Fujinami, T. Yamabe, T. Nakajima, "Take me with you!": a case study of context-aware application integrating cyber and physical spaces, in: Proceedings of the 2004 ACM Symposium on Applied Computing, 2004, pp. 1607–1614.
[31] A. Gárate, N. Herrasti, A. López, GENIO: an ambient intelligence application in home automation and entertainment environment, in: Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies, 2005, pp. 241–245.
[32] D. Garlan, V. Poladian, B. Schmerl, J.P. e Sousa, Task-based Self-adaptation, in: Proceedings of the ACM SIGSOFT 2004 Workshop on Self-Managing Systems (WOSS'04), Newport Beach, CA, October/November 2004.
[33] D. Garlan, D.P. Siewiorek, A. Smailagic, P. Steenkiste, Project aura: toward distraction-free pervasive computing, IEEE Pervasive Computing 1 (2) (2002) 22–31.
[34] M. Glesner, T. Hollstein, L.S. Indrusiak, P. Zipf, T. Pionteck, M. Petrov, H. Zimmer, T. Murgan, Reconfigurable platforms for ubiquitous computing, Conference on Computing Frontiers (2004) 377–389.
[35] A.T.A. Gomes, A. Ziviani, N.A. de Souza e Silva, R.A. Feijoo, Towards an ubiquitous healthcare system for acute myocardial infarction patients in Brazil, in: Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2006, PerCom Workshops 2006, 2006, pp. 582–589.
[36] A.M. Hadjiantonis, M. Charalambides, G. Pavlou, A policy-based approach for managing ubiquitous networks in urban spaces, in: IEEE International Conference on Communications, 2007, ICC '07, 2007, pp. 2089–2096.
[37] S. Hallsteinsen, E. Stav, J. Floch, Self-Adaptation for Everyday Systems WOSS'04 Oct 31-Nov 1, 2004 Newport Beach, CA, USA, 2004.
[38] A. Harrington, V. Cahill, Route profiling: putting context to work, in: SAC 2004, 2004.
[39] M. Hatala, R. Wakkary, L. Kalantari, Rules and ontologies in support of real-time ubiquitous application, Journal of Web Semantics (2005).
[40] T.D. Hodes, R.H. Katz, E. Servan-Schreiber, L.A. ROWE, Composable ad-hoc mobile services for universal interaction, Mobile Computing and Networking (1997) 1–12.
[41] D. Hong, D.K.W. Chiu, V.Y. Shen, Requirements elicitation for the design of context-aware applications in an ubiquitous environment, in: ICEC'05, August 15–17, 2005, Xi'an, China, 2005.
[42] A.C. Huang, C.L. Benjamin, J.J. Barton, A. Fox, Making computers disappear: appliance data services, in: ACM SIGMOBILE Seventh Annual International Conference on Mobile Computing and Networking (Mobicom), Rome Italy, 2001.
[43] W. Hurst, User interfaces for telepresentations—input devices, interaction concepts and design issues, Journal of Network and Computer Applications 23 (2000) 1–15.
[44] I. Korhonen, J.E. Bardram, Introduction to the special section on pervasive healthcare, IEEE Transactions on Information Technology in Biomedicine 8 (2004) 229–234.
[45] L.S. Indrusiak, F. Lubitz, R. Reis, M. Glesner, Ubiquitous access to reconfigurable hardware: application scenarios and implementation issues, in: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'03), 2003.
[46] L.M. Jessup, D. Robey, The relevance of social issues in ubiquitous computing environments, Communications of the ACM 45 (12) (2002) 88–91.
[47] L. Jing, X. Ye, Z. Cheng, An ubiquitous learning system based on the digital artifacts, in: First IEEE International Conference on Ubi-Media Computing, 2008, pp. 314–319.
[48] S. Joel, J.L. Arnott, N.A. Hine, H. Ingvarsson, R. Rentoul, S. Schofield, A framework for analyzing interactivity in a remote access field exploration system, SMC (3) (2004) 2669–2674.
[49] D. Johansen, H.D. Johansen, R. Renesse, Environment mobility: moving the desktop around, Middleware for Pervasive and Ad-hoc Computing (2004) 150–154.
[50] V.M. Johnson, R. Vernik, e-Ghosts: leaving virtual footprints in ubiquitous workspaces, in: A. Cockburn (Ed.), Proceedings of Fifth Australasian User Interface Conference (AUIC2004), Dunedin, New Zealand, CRPIT, vol. 28, ACS, 2004, pp. 111–116.
[51] R. Jose, A. Moreira, H. Rodrigues, N. Davies, The AROUND architecture for dynamic location-based services, Mobile Networks and Applications 8 (4) (2003) 377–387.
[52] L. Kagal, V. Korolev, S. Avancha, A. Joshi, T.W. Finin, Y. Yesha, Centaurus: an infrastructure for service management in ubiquitous computing environments, Wireless Networks 8 (6) (2002) 619–635.
[53] K. Kalapriya, S.K. Nandy, D. Srinivasan, R. Maheshwari, V. Satish, A framework for resource discovery in pervasive computing for mobile aware task execution, Conference on Computing Frontiers (2004) 70–77.
[54] H. Kawamichi, S. Sameshima, H. Kato, K. Kawano, A service selection method based on context types for an ubiquitous service system in a public space. In: Proceedings of the International Symposium on Applications and the Internet Workshops (Saint 2004Workshops), 2004, pp. 319–325.
[55] M. Keidl, A. Kemper, Towards context-aware adaptable web services, WWW (Alternate Track Papers & Posters), 2004, pp. 55–65.
[56] M. Kheder, A. Karmouch, Exploiting co-location history for efficient service selection in ubiquitous computing systems, in: The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, MobiQuitous, 2005, pp. 202–209.
[57] M. Khedr, A. Karmouch, Exploiting SIP and agents for smart context level agreements, in: IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, BC, Canada, August 2003.
[58] J.A. Kientz, G.R. Hayes, T.L. Westeyn, T. Starner, G.D. Abowd, Pervasive computing and autism: assisting caregivers of children with special needs, in: IEEE Pervasive Computing, Special Issue on Pervasive Healthcare, January 2007, pp. 28–35.
[59] T. Kindberg, J. Barton, G. Becker, D. Caswell, P. Debaty, G. Gopal, M. Frig, V. Krishnan, H. Morris, J. Schettino, B. Serra, M. Spasojevic, People, places, things: Web presence for the real world, in: Third IEEE Workshop on Mobile Computing Systems and Applications, 2000, pp. 19–28.
[60] T. Kindberg, A. e Fox, System software for ubiquitous computing, IEEE Pervasive Computing 1 (1) (2002) 70–81.
[61] P.S. Kumar, Q. Zeng, G. Singh, Constraining event flow for regulation in pervasive systems, in: Proceedings Third IEEE International Conference on Pervasive Computing and Communications, PerCom 2005, 2005, pp. 314–318.
[62] V. Kwan, F.C.M. Lau, C. Wang, Functionality adaptation: a context-aware service code adaptation for pervasive computing environments, Web Intelligence (2003) 358–364.
[63] S.H. Lee, T.C. Chung, System Architecture for Context-Aware Home Application, WSTFEUS 2004, 2004.
[64] S.J. Lee, Y.H. Kim, S.S. Kim, K.S. Ahn, A remote monitoring and control of home appliances on ubiquitous smart homes, in: Proceedings of the 1st International Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications, Innsbruck, Austria, 2008.
[65] D. Liu, J. Peng, K.H. Law, G. Wiederhold, Efficient integration of web services with distributed data flow and active mediation ICEC'04, in: Sixth International Conference on Electronic Commerce, 2004.
[66] Luyi Li, Yanlin Zheng, Hiroaki Ogata, Yoneo Yano, Ubiquitous computing in learning: toward a conceptual framework of ubiquitous learning environment, International Journal of Pervasive Computing and Communications 1 (3) (2005) 207–216.
[67] K. Lyytinen, Y. Yoo, Issues and challenges in ubiquitous computing, Communications of the ACM 45 (12) (2002) 63–65.
[68] Z. Maamar, S.K. Mostefaqui, H. Yahyaoui, A Web services composition approach based on software agents and context, SAC (2004) 1619–1623.
[69] N. Maisonneuve, M. Stevens, M.E. Niessen, P. Hanappe, L. Steels, Citizen noise pollution monitoring, in: Proceedings of the 10th Annual International Conference on Digital Government Research: Social Networks: Making Connections between Citizens, Data and Government, 2009, pp. 96–103.
[70] T. Mantoro, S. Johnson, Oleg Davidyuk, Jukka Riekki, Ville-Mikko Rautio, Junzhao Sun, Context-aware middleware for mobile multimedia applications, in: Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia, MUM '04 C., 2003. Location History in a Low-cost Context Awareness Environment. ACSW Frontiers 2003, 2004, pp. 153–158.
[71] P.G. McLean, A secure pervasive environment: Australasian Information Security Workshop 2003 (AISW2003), Adelaide, Australia, in: Conferences in Research and Practice in Information Technology, vol. 21, 2003.
[72] F. Michahelles, S. Antifakos, A. Schmidt, M. Beigl, B. Schiele, Towards distributed awareness – an artefact based approach, in: Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications, 2004.
[73] J. Nawyn, S. Intille, K. Larson, Embedding behaviour modification strategies into consumer electronic devices: a case study, in: Proceedings of the 8th International Conference on Ubiquitous Computing, 2006.
[74] T. Nguyen, S.W. Loke, T. Torabi, H. Lu, PlaceAware: a tool for enhancing social interactions in urban places, in: 10th International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN), 2009, pp. 143–147.
[75] E. Niemela, J. Latvakoski, Survey of requirements and solutions for ubiquitous software, in: Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia, October 27–29, 2004, College Park, Maryland, 2004, pp. 71–78.
[76] K.M. Oliveira, F. Zlot, A.R.C. Rocha, G.H. Travassos, C.G.M. Silva, C.S. Menezes, Domain oriented software development environment, Journal of Systems and Software 72 (2) (2004) 145–161.

[77] E. O'Neill, T. Kindberg, A.F. Schiecka, Gen, T. Jones, A. Penn, D.S. Fraser, Instrumenting the city: developing methods for observing and understanding the digital cityscape, in: Proceedings of the 8th International Conference on Ubiquitous Computing (UBICOMP), 2006.
[78] M. Pai, M. McCulloch, J.D. Gorman, et al., Systematic reviews and meta-analyses: an illustrated, step-by-step guide, The National Medical Journal of India 17 (2004) 2.
[79] P. Park, H.F. Lee, U-healthcare aide system for ubiquitous wellbeing lifecare smart space, in: International Conference on New Trends in Information and Service Science, 2009, NISS '09, 2009, pp. 781–783.
[80] V. Poladian, J.P. Sousa, D. Garlan, M. Shaw, Dynamic configuration of resource-aware services, in: Proceedings of the 26th International Conference on Software Engineering, ICSE 2004, 2004, pp. 604–613.
[81] J.M. Quero, C.L. Tarrida, J.J. Santana, V. Ermolov, I. Jantunen, H. Laine, J. Eichholz, Health Care Applications Based on Mobile Phone Centric Smart Sensor Network, in: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007, EMBS 2007, 2007, pp. 6298–6301.
[82] K. Raatikainen, H.B. Christensen, T. Nakajima, Application requirements for middleware for mobile and pervasive systems, ACM SIGMOBILE Mobile Computing and Communications Review 6 (4) (2002).
[83] M. Raento, A. Oulasvirta, R.e. Petit, H. Toivpnen, ContextPhone: a prototyping platform for context-aware mobile applications, IEEE Pervasive Computing 4 (2) (2005) 51–59.
[84] A. Ranganathan, S. Chetan, R. Cambell, Mobile polymorphic applications in ubiquitous computing environments, in: The First Annual International Conference on Mobile and Ubiquitous Systems, August 2004.
[85] D. Raptis, N. Tselios, N. Avouris, Context-based design of mobile applications for museums: a survey of existing practices, in: Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services (Salzburg, Austria, September 19–22, 2005), MobileHCI '05, vol. 111, ACM Press, New York, NY, 2005, pp. 153–160.
[86] O. Ratsimor, D. Chakraborty, A. Joshi, T. Finin, Allia: alliance based service discovery for ad-hoc environments, in: Proceedings of the 2nd International Workshop on Mobile Commerce, 2002.
[87] P. Repo, Facilitating user interface adaptation to mobile devices, in: Proceedings of the Third Nordic Conference on Human–Computer Interaction, October 23–27, 2004, Tampere, Finland, 2004, pp. 433–436.
[88] P. Repo, J. Riekki, Middleware support for implementing context-aware multimodal user interfaces, in: Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia, October 27–29, 2004, College Park, Maryland, 2004, pp. 221–227.
[89] M. Roman, N. Islam, Dynamically programmable and reconfigurable middleware services, in: Proceedings of the 5th ACM/IFIP/USENIX International Conference on Middleware, October 18–22, 2004, Toronto, Canada, 2004.
[90] M. Roman, B. Ziebart, R.H. Campbell, Dynamic application composition: customizing the behaviour of an active space, in: PerCom, 2003.
[91] L. Rosenthal, V. Stanford, NIST smart space: pervasive computing initiative, in: IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'00), 2000.
[92] S.M. Sadjadi, P.K. McKinley, B.H.C. Cheng, Transparent Shaping of Existing Software to Support Pervasive and Autonomic Computing DEAS 2005, May 21, 2005, St. Louis, Missouri, USA, 2005.
[93] A. Salovaara, A. Oulasvirta, Six modes of proactive resource management: a user-centric typology for proactive behaviours, in: NordiCHI '04, October 23–27, 2004, Tampere, Finland, 2004.
[94] S. Sameshima, K. Kawano, K. Takashio, H. Morikawa, M. Minam, Opportunities and issues relating to middleware technologies for context-aware service, in: IEEE, SICE Annual Conference in Fukui, August 4–6, 2003.
[95] I. Satoh, A mobile agent-based framework for location-based services, in: Proceedings of IEEE International Conference on Communications (ICC'2004), IEEE Communication Society, June 2004.
[96] I. Satoh, Mobile applications in ubiquitous computing environments, IEICE Transactions on Communications E88-B (3) (2005) 1026–1033.
[97] M. Sharmin, S. Ahmed, S.I. Ahamed, SAFE-RD (Secure, Adaptive, Fault Tolerant, and Efficient Resource Discovery) in a Pervasive Computing Environments, in: IEEE, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05), 2005.
[98] M. Shaw, What makes good research in software engineering? Shown at ETAPS 2002, Grenoble, France, International Journal of Software Tools for Technology Transfer 4 (1) (2002) 1–7.
[99] J.H. Shin, G.S. Chung, K.K. Kim, J.S. Kim, B.S. Hwang, K.S. Park, Ubiquitous house and unconstrained monitoring devices for home healthcare system, in: 6th International Special Topic Conference on Information Technology Applications in Biomedicine, 2007, ITAB 2007, 2007, pp. 201–204.
[100] F. Siegemund, C. Florkemeier, Interaction in pervasive computing settings using bluetooth-enabled active tags and passive RFID technology together with mobile phones, in: IEEE International Conference on Pervasive Computing and Communications (PerCom 2003), 2003, pp. 378–387.
[101] R. Spinola, F.R. Pinto, G.H. Travassos, Supporting requirements definition and quality assurance in ubiquitous software project, in: Leveraging Applications of Formal Methods, Verification and Validation Third International Symposium, ISoLA 2008, 2008, Porto Sani, 2008, vol. 17, 2008, pp. 587–603. doi: http://dx.doi.org/10.1007/978-3-540-88479-8_42.
[102] R. Spínola, F.R. Pinto, G.H. Travassos, UbiCheck: an approach to support requirements definition in the ubicomp domain, in: ACM Symposium on Applied Computing, 2010, Sierre, Switzerland, Proceedings of the 2010 ACM Symposium on Applied Computing, New York: ACM, 2010, pp. 306–310, doi: http://dx.doi.org/10.1145/1774088.1774152.
[103] R.O. Spínola, Supporting the Definition and Verification of Ubiquity Functional Requirements in Software Projects, PhD thesis, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, 2010.
[104] R.O. Spínola, J.L.M. Silva, G.H. Travassos, Checklist to characterize ubiquitous software projects, in: Proc. XXI Brazilian Symposium on Software Engineering, 2007.
[105] R.O. Spínola, G.H. Travassos, Characteristics of ubiquitous software projects: pertinence, relevance, and use, in: Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering (SEKE 2010), San Francisco Bay, CA, USA, 2010.
[106] V. Stanford, Pervasive computing puts food on the table, IEEE Pervasive Computing 2 (1) (2003) 9–14.
[107] N. Streitz, P. Nixon, The disappearing computer. communications of the ACM, Special issue on the Disappearing Computer, 2005.
[108] M. Su, H.W. Zhang, Y.J. Lin, Y.H. Su, S.J. Chen, H.S. Chen, Pilot study on a community-based ubiquitous healthcare system for current and retired university employees, in: IEEE International Conference on Communications Workshops, 2009, ICC Workshops 2009, 2009, pp. 1–5.
[109] D. Sullivan, D. Lewis, Semantically driven service interoperability for pervasive computing, in: Proceedings of the Third ACM International Workshop on Data Engineering for Wireless and Mobile Access, MobiDE 2003, San Diego, CA, United States, 2003.
[110] J. Sun, Z. Wu, A comprehensive context model for next generation ubiquitous computing applications, RTCSA (2005) 447–450.
[111] C. Tadj, M.D. Hina, A. Ramdane-Cherif, G. Ngantchaha, The LATIS pervasive patient subsystem: towards a pervasive healthcare system, in: International Symposium on Communications and Information Technologies, 2006, ISCIT '06, 2006, pp. 851–856.
[112] M. Tahti, V. Rauto, L. Arhippainen, Utilizing context-awareness in office-type working life, in: Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia, 2004.
[113] S.M. Thayer, P. Steenkiste, An architecture for the integration of physical and informational spaces, Personal and Ubiquitous Computing 7 (2) (2003) 82–90.
[114] K.N. Truong, G.D. Abowd, J.A. Brotherton, Personalizing the capture of public experiences, ACM Symposium on User Interface Software and Technology (1999) 121–130.
[115] A.M. Vainio, M. Valtonen, J. Vanhala, Learning and adaptive fuzzy control system for smart home, in: Proceedings of the AmI.d, September 20–22, 2006.
[116] D. Vassis, P. Belsis, C. Skourlas, G. Pantziou, Providing advanced remote medical treatment services through pervasive environments, Personal and Ubiquitous Computing 14 (6) (2010) 563–573.
[117] L. Veiga, P. Ferreira, PoliPer: policies for mobile and pervasive environments, Adaptive and Reflective Middleware (2004) 238–243.
[118] N. Venkatasubramanian, Safe 'composability' of middleware services, Communications of the ACM 45 (6) (2002) 49–52.
[119] Y. Wang, An approach to enhancing mobile applications with situation-awareness capabilities, in: Proceedings of the International Conference on Parallel Processing Workshops, 2004, pp. 472–479.
[120] R. Want, T. Pering, System challenges for ubiquitous & pervasive computing, in: Proceedings of the 27th International Conference on Software Engineering (ICSE), St. Louis, MO, 2005, pp. 9–14.
[121] M. Weiser, The Computer for the 21st Century, Scientific American, 1991, pp. 94–104.
[122] M. Weiser, Some computer science issues in ubiquitous computing, Communications of the ACM 36 (7) (1993).
[123] M.-J. Yang, J.-H. Chen, T.-H. Wang, L.R. Chao, T.K. Shih, To construct the outdoor experience game-based, in: Proceedings of the 2009 Workshop on Ambient Media Computing, Beijing, China, 2009, pp. 77–82.
[124] S.S. Yau, D. Huang, H. Gong, S. Seth, Development and runtime support for situation-aware application software in ubiquitous computing environments, in: Proceedings of the 28th Annual International Computer Software and Applications Conference, COMPSAC 2004, 2004, pp. 452–457.
[125] S.S. Yau, F. Karim, Context-sensitive middleware for real-time software in ubiquitous computing environments, ISORC (2001) 163–170.
[126] S.S. Yau, F. Karim, An energy-efficient object discovery protocol for context-sensitive middleware for ubiquitous computing, IEEE Transactions on Parallel and Distributed Systems 14 (11) (2003) 1074–1085.
[127] S.S. Yau, Y. Wang, F. Karim, Development of situation-aware application software for ubiquitous computing environments, in: Proceedings of the 26th Annual International Computer Software and Applications Conference, 2002.
[128] X. Ye, L. Jing, Z. Cheng, An ubiquitous learning system using co-learning ubiquitous pet for enhancing educational effect, in: IEEE 8th International Conference on Computer and Information Technology Workshops, 2008, CIT Workshops 2008, 2008, pp. 494–500.
[129] F. Zhou, J. Jiao, S. Chen, D. Zhang, A case-driven ambient intelligence system for elderly in-home assistance applications, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, PP (99), 2010, pp. 1–11.
[130] P. Zhou, T. Nadeem, P. Kang, C. Borcea, L. Iftode, EZCab: a cab booking application using short-range wireless communication, PerCom (2005) 27–38.