Catalog of Invisibility Requirements for UbiComp and IoT Applications

Rainara M. Carvalho, Rossana M. C. Andrade[1]
Group of Computer Network, Software Engineer and systems (GREat), Computer Science Department (DC)
Federal University of Ceará
Fortaleza, Brazil
{rainaracarvalho, rossana}@great.ufc.br

Káthia M. Oliveira, Christophe Kolski
Laboratory of Industrial and Human Automation control, Mechanical Engineering and Computer Science
Université Polytechnique Hauts-de-France,
Valenciennes, LAMIH UMR CNRS 8201, France
{Kathia.Oliveira, Christophe.Kolski}@univ-valenciennes.fr

***Abstract*— A new set of Non-Functional Requirements (NFRs) have appeared with the advent of Ubiquitous Computing (UbiComp) and more recently Internet of Things (IoT). Invisibility is one of these NFRs that means the ability to hide technology from users. Although invisibility is long seen as an essential characteristic for achieving the goals of UbiComp, it has not been cataloged regarding its subcharacteristics and solutions, making its design and requirements specification in such applications a challenging task. Considering the Softgoal Interdependency Graph (SIG), which is a well-known format to catalog NFRs, this work aims at capturing subcharacteristics and solutions for Invisibility and cataloguing them in a SIG. Since there is no systematic approach on how to build SIGs, we also propose to systematize the definition of Invisibility SIG using the following well-defined research methods: snowballing, database search, grounded theory and questionnaires. As a result, we got an Invisibility SIG composed of two main subcharacteristics, twelve sub-subcharacteristics, ten general solutions and fifty-six specific solutions. This organized body of knowledge is useful for supporting software engineers to specify requirements and practical solutions for UbiComp and IoT applications. Furthermore, the proposed methodology used to capture and catalog requirements in a SIG can be reused for other NFRs.**

***Index Terms* — UbiComp, IoT, Interaction, Invisibility, NFR, Quality Characteristic, Catalog, Grounded Theory.**

I. Introduction

Several Ubiquitous Computing (UbiComp) ideas [1][2] have boosted what is nowadays called the Internet of Things (IoT), a collection of objects, smart and connected to the Internet [3]. For this work, we consider that, together, UbiComp and IoT create an environment full of smart objects which are interconnected and aware of the surrounding events [4].

Applications from this environment bring a new set of non-functional requirements (NFRs), most of them concerned with human-computer interaction (HCI) quality [5]. According to [6], one kind of NFR is the quality characteristics (*e.g.*, *Usability*, *Security*), which represents the expectations beyond the functionalities of the system. A recent study [7] about quality characteristics from UbiComp has identified *Invisibility* as a specific characteristic for interaction quality in UbiComp. This characteristic is concerned about the complete disappearance of the technology from user's mind [8], as envisaged by Mark Weiser in 1991 [1].

In fact, *Invisibility* is long seen as an essential characteristic for achieving the goals of UbiComp [8][9][10][11][12], which can also be taken to IoT systems [13][14]. However, this characteristic is not cataloged regarding its subcharacteristics and solutions about how to design and implement it. The knowledge is spread out among several studies. Software engineers must search for these sources or must rely on experts in the field, making its design and requirements specification in UbiComp and IoT applications a challenging task.

A well-known notation used for cataloging knowledge of quality characteristics is the Softgoal Interdependency Graph (SIG) [15][16][17]. SIGs treat a quality characteristic as a softgoal, which is a goal with no clear-cut criteria of satisfaction [17]. This is convenient for a characteristic as *Invisibility,* which is very hard or even impossible to be entirely achieved. Softgoals should be refined into sub softgoals (i.e., subcharacteristics), which should be refined into operationalizing softgoals (*i.e.*, solutions for achieving softgoals). Therefore, this work aims at capturing subcharacteristics and solutions for *Invisibility* characteristic and cataloging them in a SIG.

Existing studies that propose to build a SIG usually use literature review and experts to capture knowledge that is scattered among several sources [15][18][16][19][20]. However, most of them do not explain exactly how they arrive at the final graph [18][19] [20], and when they do it [15][16], they do not use a systematic approach to generate that knowledge. This lack of a well-defined methodology can increase potential misconceptions and bias in the final graph.

For that reason, we intend to systematize the definition of *Invisibility* SIG by using well-defined research methods. In addition to a literature review, we propose to use a method of qualitative data analysis, the Grounded Theory (GT) [21], to investigate the data obtained from literature. This method builds theory from data in a systematic way, by extracting and grouping concepts derived from data, building a knowledge pyramid, which can be suitable for defining a SIG. Besides, the SIG was complemented with specific solutions extracted from questionnaires sent to experienced developers of UbiComp and IoT applications.

[1] Researcher scholarship - DT Level 2, sponsored by CNPq (Brazil)

2332-6441/18/$31.00 ©2018 IEEE
DOI 10.1109/RE.2018.00019

The remainder of this paper is organized as follows. We present the state of the art about *Invisibility* and catalogs in Section II. In Section III, we describe the methodology and its main outcomes. Section IV presents the proposed SIG. Section V presents our contributions and threats to validity. Finally, Section VI concludes with a summary and further work.

II. State of The Art

A. Invisibility in UbiComp and IoT

Applications from UbiComp and IoT moved away from the desktop environment to be immersed in the user's everyday objects. They have new and different sensing possibilities as well as they can start an interaction with the user and can be present in various objects [22]. For example, air-conditioners can be controlled remotely or can act alone according to the environment context, doors can be automatically unlocked to authorized users [23]. Therefore, this kind of application changed the way users interact with technology and brought us a set of new characteristics [5].

A recent systematic mapping (SM) study identified 27 quality characteristics with great impact on the interaction in UbiComp applications [7]. A subset of these characteristics was identified as being very specific for the UbiComp environment: *Invisibility*, *Context-Awareness*, *Attention*, *Calmness* and *Mobility*. *Invisibility* is one of the most cited characteristics, only after *Usability* and *Context-Awareness*.

As a matter of fact, *Invisibility* has been one of the key issues and challenges to make UbiComp a reality for over ten years [8][9][10][11][12]. There are several definitions for this characteristic proposed by researchers [9][12][24][25][26][27][28][29][30]. They originate from early studies about UbiComp challenges and are mostly related to the ability of hiding systems from user's point of view[2].

Although *Invisibility* has been pointed out as a primary and essential characteristic for UbiComp [8][9][10][11][12] and, consequently, IoT applications [13][14], the literature lacks well-defined taxonomies or a body of organized knowledge about how to achieve it during software development.

The work of [12] is an attempt to characterize *Invisibility* by proposing software measures. However, they are not well defined in terms of description, measurement function and interpretation. Another recent study is from [5], which links *Invisibility* with concepts such as implicit inputs and proactivity when proposing software measures to evaluate these aspects.

In fact, the existing studies regarding *Invisibility* take an evaluation perspective in the sense that they propose measures to evaluate the degree to which an application achieves this characteristic [5][12][24][25]. It is essential to have this kind of approach in the software development, however, a body of knowledge about how to implement software considering this characteristic during the initial stages of requirements analysis is equally important [17].

[2] The term "*Invisibility*" used in this work is different from the term "*Transparency*" in [15], which is related to the rights of the users of seeing their data and understand the system. However, some UbiComp and IoT papers are using the two terms interchangeably. Then, we do not exclude studies that consider the two terms interchangeably, if their meaning is the same of ours, which came from UbiComp studies.

B. Catalogs of Requirements

According to [17], a body of knowledge can be stored into catalogs. Many studies have used the Softgoal Interdependencies Graphs (SIG) from the NFR Framework as a structure to catalog knowledge about a quality characteristic, such as Security [17], Usability [18] and Performance [31].

A SIG starts from a high-level NFR, called NFR softgoal, and drawn as a cloud in the SIG. Then, the NFR softgoal needs to be converted into something that can be implemented. Thus, they can be refined into more specific NFR softgoals (or subcharacteristics). These softgoals need to be refined until they reach solutions, which are called operationalizing softgoals and drawn as dark clouds in the graph. These operationalizations can even be refined into more specific ones (specific operationalizing softgoals). All these refinements can be made using the following contributions: *(i)* AND/OR contribution; *(ii)* Make/Help/Some+ for positive impact; or *(iii)* Break, Hurt, Some- for negative impact. Fig. 1 shows part of a SIG for Security.

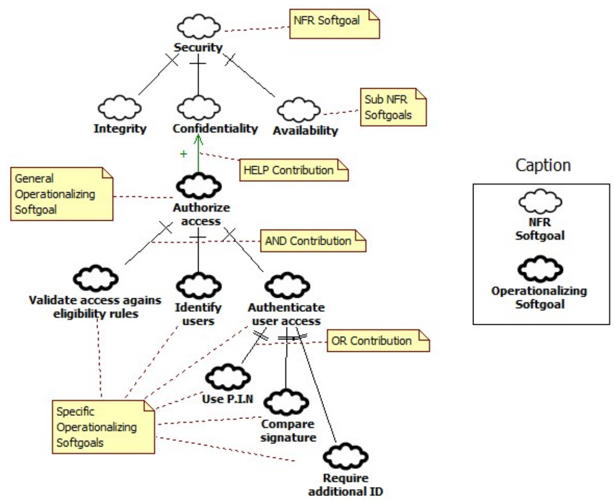


Fig. 1. Example of SIG for Security. Adapted from [17] .

We found three studies ([19][20][32]) that are catalogs for NFRs of ubiquitous computing. The first [20] presents a SIG for context-awareness. The second study [19] shows a SIG of *Mobility* requirement for ubiquitous computing middleware. The last one proposed a SIG for ubiquitous and pervasive ubiquitous that focus on the following characteristics: *Ubiquity*, *Pervasiveness*, *Mobility* and *User Satisfaction*. However, for the best of our knowledge, we did not find any study with a SIG for *Invisibility* for both Ubicomp and IoT.

Despite that, many studies that present NFRs catalogs in a SIG format do not describe precisely the steps towards the final graph [15][18][16][19]. We have found one study explaining how the authors captured and cataloged the knowledge about a not well characterized NFR, *Transparency* [15]. This study first starts with a literature review to capture subcharacteristics using principles of systematic reviews. However, they do not use a research method to analyze data that came from the literature, which is a threat to the validity of the SIG definition. Considering that a SIG (i.e., the catalog) is a reuse artifact [17], it is essential to use rigorous procedures to define it.

III. Defining The Invisibility SIG

Fig. 2 presents our methodology to define the *Invisibility* SIG, which is composed of three phases.

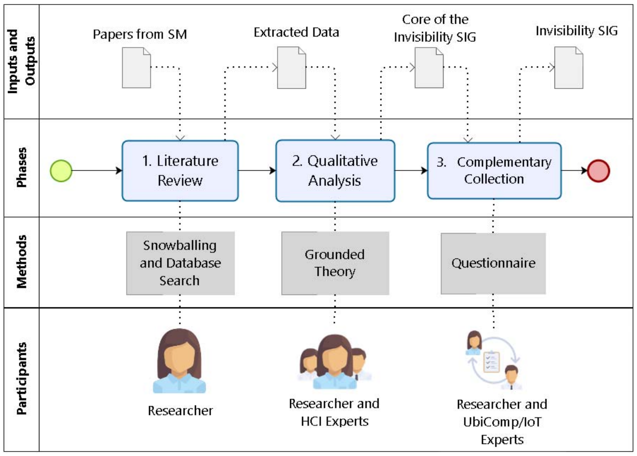


Fig. 2. Overview of the Methodology to define a SIG for *Invisibility*

In the first phase, a literature review is performed to capture existing data about *Invisibility* using snowballing and database search. Papers with *Invisibility* definitions from the previously explained SM [7] for UbiComp were used as input to perform the snowballing method. One of the authors of this paper (researcher in Fig.2) performed these tasks. At the end of this phase, a set of extracted data about *Invisibility* was identified. Then, a qualitative analysis was performed through the Grounded Theory (GT) method in the second phase. As an outcome of this method, we got a partial *Invisibility* SIG, with sub NFR softgoals and general operationalizing softgoals. This second phase was performed by the researcher of phase 1 and two experts in HCI, since *Invisibility* is highly related to an interaction perspective. Finally, at the third phase to gather more specialized operationalizing softgoals to complement the SIG, questionnaires were applied for developers in UbiComp and IoT applications development.

Each one of these phases and its outcomes are better explained in the next subsections.

A. Phase 1: Literature Review

When the knowledge of a quality characteristic is spread out in several sources, as happened with *Invisibility*, it is necessary to review the literature to capture that knowledge to start the decomposition of that characteristic.

To systematize this task, we suggest the use of a systematic literature review [33][34], so there is no bias in the collected data. In our case, there was already a SM study related to quality characteristics from UbiComp [7], we then decided to use a snowballing strategy to gather data from literature [35], which was performed in one iteration.

Snowballing refers to using the reference list of a paper (<u>backward snowballing</u>) or citations to that paper (<u>forward snowballing</u>) to identify additional papers [35]. Therefore, both strategies need a starting set of papers. Since the existing SM study has papers related to 27 characteristics, we selected only the papers citing *Invisibility*, which were 9 papers. Thus, we had a consistent and well-defined set of papers to consider as a starting set for the snowballing method.

Fig. 3. presents the snowballing steps, adapted from [35]. To perform the backward snowballing, we got all references from the 9 papers, which are 244 references in total. For the forward snowballing, we got all the citations of the 9 papers using Google Scholar, *i.e.*, 301 in total.

The first step of both snowballing strategies was applying the following basic criteria to filter references and citations: *(i)* Papers not written in English; *(ii)* Papers published before 1991 (year of the first paper about Ubicomp); and *(iii)* References and citations that were only web address of research groups, newspaper and/or companies. It was possible to reduce 244 references to 211 papers and 301 citations to 258 papers.

Then, the following criterion was applied in all steps of snowballing: *(i)* the study is not related to *invisible interaction* in UbiComp or IoT, which means we were looking for papers related to the interaction perspective, regardless of the research type (*i.e.*, solution proposal, experiment, case study).

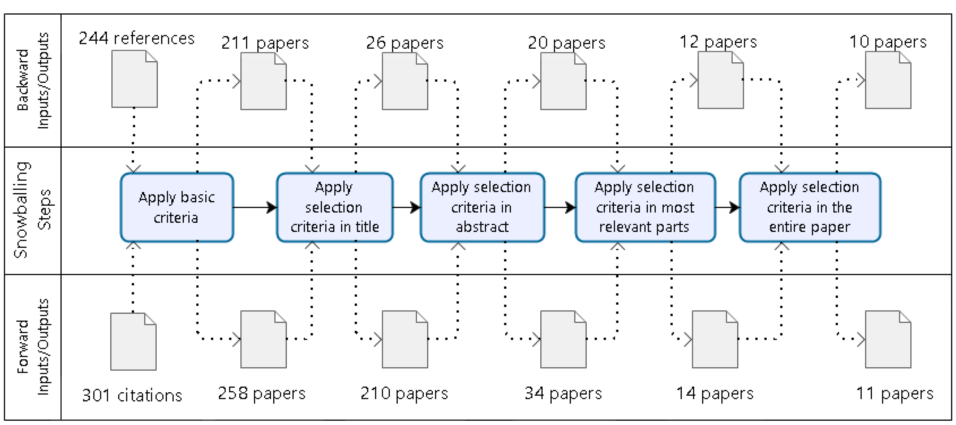


Fig. 3. Snowballing Steps

The second step consisted of applying the criterion in title reading, which resulted in 26 papers (backward) and 210 papers (forward). The third step consisted of applying it in the abstract reading, which reduced the set to 20 papers (backward) and 34 papers (forward). The fourth step consisted of reading the most relevant parts of the paper, resulting in 12 papers (backward) and 14 papers (forward). The last step was performed by applying the criteria during the full paper reading. This step resulted in 10 and 11 final papers, for backward and forward, respectively.

Additionally, a complementary search in a database for the IoT domain was performed to get more pieces of evidence in literature. We used Scopus library with the following search string: *(("internet of things" OR iot) AND (interaction) AND (transparency OR invisibility OR disappearance OR diffusion OR implicit OR "minimal user distraction" OR unobtrusive))*. This library was selected because of its broad coverage and stability [36]. Also, it covers major publishers in the requirements area [37].

This search resulted in 40 papers. Then, the same criteria from the snowballing procedure was applied in the title and abstract reading, which resulted in 5 papers. Finally, the most relevant parts of the paper were read and the criteria was applied, resulting in only 1 paper.

Therefore, this work resulted in 9 papers from a previous SM, 10 papers from backward snowballing, 11 papers from forward snowballing and 1 paper from the database search, resulting in 31 papers. They were extracted using a form with the following questions: *(a)* ***What are the Invisibility definitions?*** This question aims to collect all definitions available on literature about *Invisibility*; *(b)* ***How is Invisibility characterized?*** This question aims to collect existing subcharacteristics; and (c) ***How is Invisibility implemented?*** This question aims to identify any kind of solution used to implement *Invisibility*.

After this data extraction, we got a set of existing data about *Invisibility*. For example, regarding the definitions, we noticed that there are several definitions proposed by researchers. Some of them are:

- "*A system that requires* ***minimal human intervention*** *offers a reasonable approximation of invisibility*" [10];
- "*Invisibility refers to the* ***integration*** *of a system into the* ***user environment****...*" [12];
- *"The extent to which the system consists of* ***hidden components in the physical space*** *and interaction is performed through* ***natural interfaces****"* [24].

We can see that definitions differ from each other in some aspects and have similarities in others. The first definition talks about minimal user intervention. The last two address the concept by another respect: the user's physical environment. Still, in the latter, the concept has been linked with natural interfaces. The results from data extraction are even worse for the subcharacteristics, which lead us to conclude there is no consensus about a characterization of *Invisibility*.

This lack of consensus about a characteristic or NFR makes the development of a catalog a challenging task since it is hard to organize knowledge without a base. Then, there is a need to investigate deeply the data to understand how we can break down *Invisibility* into subcharacteristics, which was done through the GT method [21][38].

B. Phase 2: Qualitative Analysis

The data retrieved from the literature provided a significant amount of information to be analyzed, like the definitions related to *Invisibility*. The GT method comes as a strategy to understand this data and to extract and relate concepts with each other to create the SIG.

In general, GT is composed of the following steps: planning, data collection, coding and reporting results. The planning step aims to identify the area of interest and the research question that will drive the work. In our case, the area of interest is the *Invisibility* characteristic. Our research question is: "*How can the Invisibility characteristic in UbiComp and IoT applications be defined and refined into subcharacteristics and solutions?*".

The data collection step aims to collect the necessary data to answer the research question. The data gathered in the first phase of our methodology became the data used in the analysis. Finally, the coding step is the heart of GT. This step means extracting concepts from raw data and relating them to each other until reaching a core concept [21]. In our case, we would like to extract and relate concepts that characterize *Invisibility* in regards of achieving it in an application.

The coding process is performed in three tasks: open, axial and selective coding. Each one of them is better explained by showing what it means and its outcomes towards the definition of the SIG.

1) *Open Coding:* In this task, the researcher inspects the data to understand the essence of what is being expressed [21]. In our case, we inspected the data extracted from the papers, which were the answers of the extraction form. Then, the researcher created a conceptual name (code) to represent his/her understanding. Codes can represent a single word, a phrase or a whole paragraph. Some examples of codes from the open coding are given in Table 1.

TABLE 1. EXAMPLES OF CODES FROM OPEN CODING

Code	Text segments from the extracted data
Minimal User Involvement	"A system that requires minimal human intervention offers a reasonable approximation of invisibility." [10]
	"To achieve invisibility, system must keep the focus of user to the task while keeping computing invisible..." [39]
	"According to the 'invisibility' ideal, the user should not be bothered with details of the system's functioning..." [9]
Usage of Natural Interfaces	"…ubicomp applications should allow users to utilize the skills they have obtained from daily lives to interact with computers." [40]
	"PICT technology must be transparent to users, and must have interfaces that are intuitive for humans." [27]
	"If ubicomp applications support more natural human forms of communication, they will create more natural and expressively powerful means of interaction…" [27]

The MAXQDA12[3] tool was used to support open coding, which is a recommendation from [21]. First, all extracted data was imported into this tool. Then, this data was analyzed in depth, which enabled the researcher to codify texts segments by creating codes. Each information was compared with other information along the set of data for similarities and differences. Every time a similar information was found, the existing code could be used. This strategy is called "constant comparison", usually used for qualitative analysis [21].

In this work, all the matchings between text segments and codes were performed by one researcher. However, to minimize possible misinterpretation and to help with concepts related to user invisible interaction, two experts in HCI area, evaluated the suggested codes. One expert has 32 years of experience in HCI area and the other has 8 years. We followed the process showed in Fig. 4 for open coding.

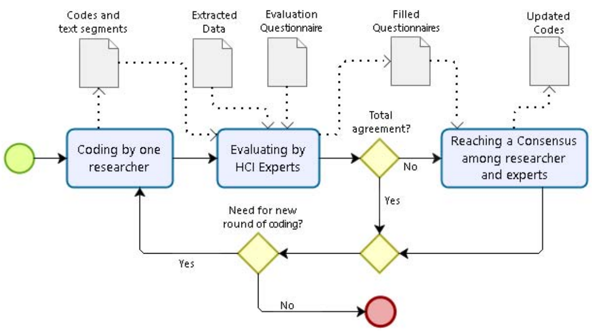
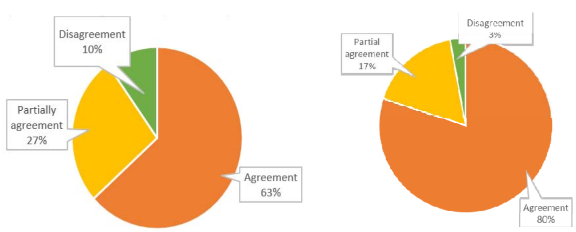


Fig. 4. Evaluation Process of the Open Coding

After coding by one researcher (first step), the suggested codes and text segments were sent to the HCI experts responsible for evaluating them (second step). The evaluation performed in this study consisted of a questionnaire where every match between the text segment and the suggested code would be evaluated and classified as: *(i)* agree, *(ii)* partially agree or *(iii)* disagree. Analyzing the results of the evaluation, if a total agreement was not obtained, a meeting was performed for discussion and consensus (third step). Finally, if the researcher noticed a need for more sessions of coding, the process can restart.

In our case, this process was carried out twice until reaching a total consensus. The evaluation rate among researchers is shown in Fig. 5 The agreement rate was 63% and 80% in the first and second time, respectively.



(a) First time **(b) Second time**

Fig. 5. Evaluation Rate among Researchers

[3] https://www.maxqda.com/

At the end of the open coding, after total consensus, this work resulted in 118 texts segments represented by 18 codes. The codes and the number of segments coded by them (called groundedness) are presented in Fig. 6. Therefore, codes with higher groundedness represent stronger concepts.

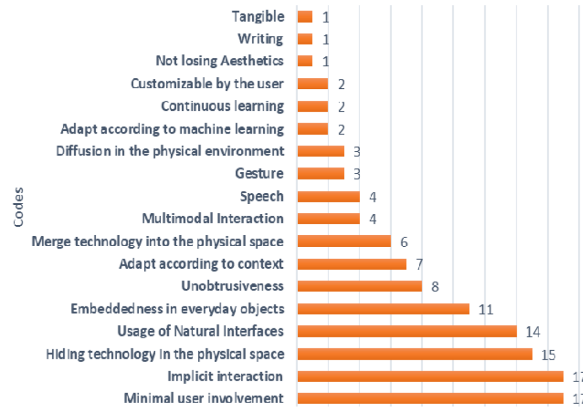


Fig. 6. Codes and its Amount of Coded Text Segments (groundedness)

We can see in Fig. 6 that "Minimal user involvement" and "Implicit interaction" are the codes with higher groundedness, followed by "Hiding technology in the physical space" and "Usage of natural interfaces".

Three codes ("tangible", "writing" and "not losing aesthetics") have the lowest groundedness. However, these codes were kept because they represent a low-level concept, which is part of another high-level concept.

All these codes can be directly represented as softgoals of the *Invisibility* SIG, either as NFR softgoal or operationalizing softgoal. Therefore, the main researcher classified each code according to the definition of NFR and operationalizing softgoals. For example, "gesture" is an operationalizing softgoal, since it operationalizes *Invisibility*. On the other hand, "implicit interaction" is still seen as an NFR softgoal, because it is an overall constraint in the system. Furthermore, they need to be related to each other and to the main softgoal *Invisibility*, which are the next tasks of GT.

2) Axial and Selective Coding: Axial coding is the process of relating concepts, or grouping them by creating categories (a high-level concept that represents a group of codes) [21]. These relations between concepts can be defined by the researcher, although there are already existing relations that can be reused, such as "is a", where a concept is a kind of another concept. In this work, the relations were created by analyzing implicit meaning between the codes and were based on the types of contributions from the SIG notation (AND, OR, MAKE, HELP, BREAK and HURT).

Fig. 7 presents examples of axial coding. The codes "tangible", "writing", "gesture" and "speech" are examples of "usage of natural interfaces". They were related with OR contribution, which means the softgoal "usage of natural interfaces" can be achieved by using any of these softgoals. Additionally, Fig. 7 shows the creation of a category called

"Natural Interaction", which embraces the codes "multimodal interaction" and "usage of natural interfaces", since they <u>*help*</u> to achieve more natural interaction.

At the end, we only used AND, OR and HELP contribution. AND contribution was used only when we were sure that all sub-softgoals were needed to achieve the parent. OR contribution was used when the sub-softgoals are examples of the same kind to achieve the parent softgoal. HELP was used when we knew about the positive support, but we were not certain about the AND/OR contribution.

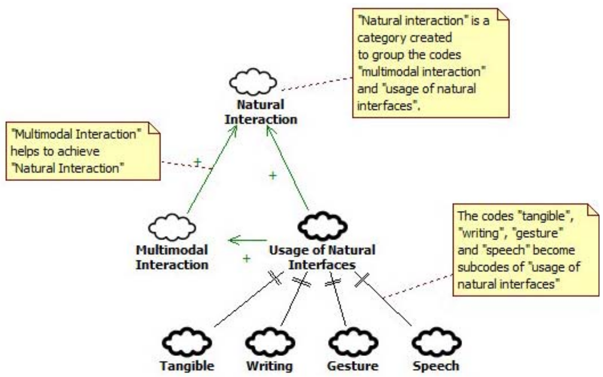


Fig. 7. Example of Axial Coding – Category of Codes

Furthermore, the researcher can group categories into new categories, creating a chain of categories. That is why axial coding is like putting together a series of interlinking blocks to build a knowledge pyramid [21]. Fig. 8 presents an example of a category of categories. The codes "customizable by the user" and "minimal user involvement" were grouped in a category called "Minimal Interaction", since they have the goal of minimizing the interaction. This category and the other category "Natural Interaction" were grouped into a new category called "*Invisibility from the usage point of view*", because they all are related to the user interaction.

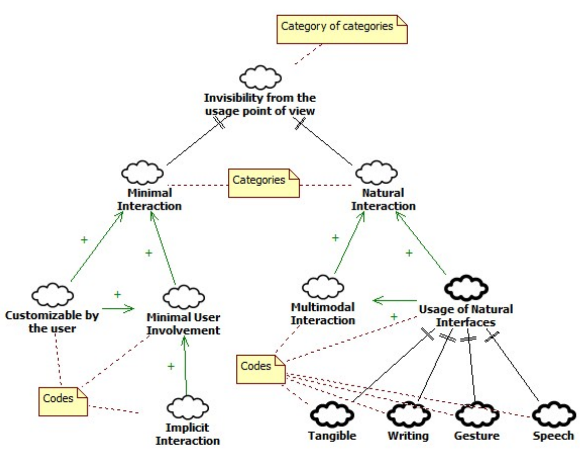


Fig. 8. Example of Axial Coding - Category of Categories

Finally, when all codes and categories can be related to a core category, it means the researcher is doing the *selective coding*. According to [21], the process of linking concepts (i.e., codes and categories) around a core category and refining and trimming the resulting theoretical construction is called Selective Coding.

The core category of this work is *Invisibility*, which is the main NFR softgoal we would like to achieve. Therefore, all codes and categories from open and axial coding were related in some way to this characteristic.

In our case, axial and selective coding were performed together by one researcher, the one who led the open coding. The researcher analyzed the codes to group them into categories, to make the necessary relations and to relate them to the core category, always with the effort to describe how the concepts are grounded in data. Then, meetings and discussions were held with the same two HCI experts, who participated in the open coding, to reach an agreement about the organization of the concepts.

During these meetings, the experts helped not only with the organization but also by complementing the *Invisibility* SIG by indicating additional softgoals. For example, the softgoal "Usage of Natural Interfaces" is composed by "tangible", "writing", "gesture" and "speech", however, these are not the only existing natural interfaces in literature. Then, the HCI experts suggested adding the following interfaces in the SIG: brain, body, eyes, breathing and haptic (See Fig. 9).

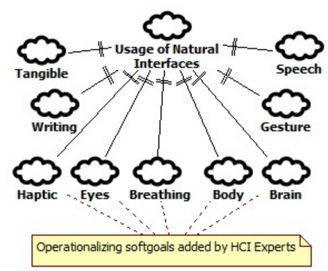


Fig. 9. Example of softgoals added by HCI experts

At the end of GT, it was possible to create a SIG of *Invisibility*. In fact, this knowledge became the core of the *Invisibility* SIG in which, with the proper arrangement, it can be expanded to include more softgoals.

C. Phase 3: Complementary Collection

As we explained in Section II, an operationalizing softgoal can be generic (broad), but it can be also refined into specific operationalizing softgoals [17], helping even more software developers and designers. Then, the *Invisibility* SIG generated from GT with HCI experts could be complemented with specific operationalizing softgoals accumulated from developers of ubiquitous and IoT applications. Although these specific softgoals could be extracted from literature, we decided to apply a questionnaire to seven UbiComp and IoT developers to collect data from their experience.

This questionnaire was designed to gather information for the softgoals at the lowest level of the SIG generated by the GT, consisting in 15 questions. For example, the developers should answer how they implement "Speech" from the "Usage of Natural Interfaces" softgoal. According to [17], solutions can be any technique, technology, data, strategy, operations or constraints that help design or code a software system with the requirement described [17]. Therefore, the developers could answer any solution about how to design or code that softgoal. It is worth noting that an experienced researcher in questionnaires development evaluated our questionnaire before being sent to the developers.

All seven developers answered the questionnaire. They have been working mainly with smart building and healthcare applications, for six years on average. Six of them have been working with programming and testing, and one has worked with architecture project.

After receiving all answers, an analysis by a researcher was performed to extract the common and most cited solutions. When an uncertainty about any of the solutions arose, the researchers contacted the developers to solve any doubt. An example of a solution from the developers is the Speech API from Google, which helps implementing "Speech" in Android applications, allowing conversion from speech to text and vice versa. Therefore, the *Invisibility* SIG could be concluded with specific operationalizing softgoals.

IV. The Invisibility SIG

The final SIG for *Invisibility* is presented in Fig. 10. The description of each softgoal is given in the next subsections. It is important to emphasize that all descriptions embrace what was extracted and analyzed from the GT with HCI experts. Although the SIG was defined following a bottom-up strategy, each softgoal will be presented in a top-down way. Then, firstly, the main NFR softgoal, *Invisibility,* is discussed in terms of a proposed definition based on and grounded in data. After that, each one of its sub-softgoals is explained.

A. Invisibility

The existing definitions of *Invisibility* are concerned mainly about the disappearance of the technology to let users focus on everyday tasks. However, as previously discussed, there are still differences between them. Through the GT analysis, it was possible to identify two groups of concepts regarding *Invisibility*: physical environment and workload during the usage of systems. Therefore, *Invisibility* can be related not only to devices in the user physical environment, but also with the minimization of the interaction workload. Then, based on and grounded in data, it was possible to propose the following definition:

"*Invisibility in UbiComp and IoT applications refers to either the merge of technology in user physical environment <u>or</u> the interaction workload decrease, both aiming to provide a greater user's focus on everyday tasks*."

Therefore, *Invisibility* can be considered either from physical view or usage view, which are its two subcharacteristics: *Invisibility from the usage point of view* (See Subsection B) and *Invisibility from the physical environment point of view* (See Subsection C).

B. Invisibility from the Usage Point of View

This subcharacteristic refers to decreasing the workload of the user interaction with the system. The workload reduction can be achieved in two ways: reducing interactions *or* designing an interaction more natural for the user. Thus, it is further refined into two others: (1) Minimal Interaction *or* (2) Natural Interaction.

1) ***Minimal Interaction****:* It refers to the system's ability to design tasks without them being entirely or constantly dependent on explicit user inputs. Minimal Interaction is *<u>helped</u>* by two other softgoals: (a) Customizable by the User or (b) Minimal User Involvement.

a) Customizable by the User: the system should, whenever possible, let the users make changes in the system according to their personal preferences, minimizing future interactions [11][12]. Table 2 presents two operationalizing softgoals that *<u>help</u>* this NFR softgoal: set warnings and set actions, both *<u>helped</u>* by rule-based mechanisms.

TABLE 2. Operationalizing "Customizable by The User"

General Op. Softgoals	**Set warnings**: the system can allow user to set warnings according to their preferences.
	Set actions: the system can allow user to set actions either in the system itself or in the physical environment.
Specific Operationalizing Softgoal ↓	
Rule-based mechanisms: mechanisms that can be implemented using rules such as If-This-Then-That.	

b) Minimal User Involvement: the system should, whenever possible, require less user interaction by decreasing the user inputs and actions. The following ones can *<u>help</u>* this softgoal: Minimize user's effort in tasks and Implicit Interaction.

Minimize user's effort in tasks is the ability to minimize the user's effort in tasks that cannot be excluded from the system. User access authentication is a general operationalizing softgoal, suggested by the UbiComp/IoT developers to *<u>help</u>* "Minimize user's effort". Table 3 presents its specific operationalizing softgoals.

TABLE 3. Operationalizing "Minimize User Actions"

General Op. Softgoal	**User Access Authenticatio**n - the ability to minimize the user's effort to access the system
Specific Operationalizing Softgoals ↓	
1. Integration with third-party services: using data from existing accounts minimize the need to type personal data to sign in, which can be implemented with Google Sign-in API [41] or Facebook API [42].	
2. Biometric techniques: automated techniques for identifying a human being based on the physiological or behavioral characteristics. For example, face or iris recognition.	
3. Login only once between devices: technique used to save passwords, so the user does not have to remember it when entering in the same service in a different device. One solution is the Smart Lock for Passwords API from Google [43].	

Implicit Interaction. It is the ability to interact with the user without his/her explicit command or awareness [44]. The general operationalizing softgoal used to *help* Implicit Interaction is "Adapt according to context". Table 4 presents its specific operationalizing softgoals.

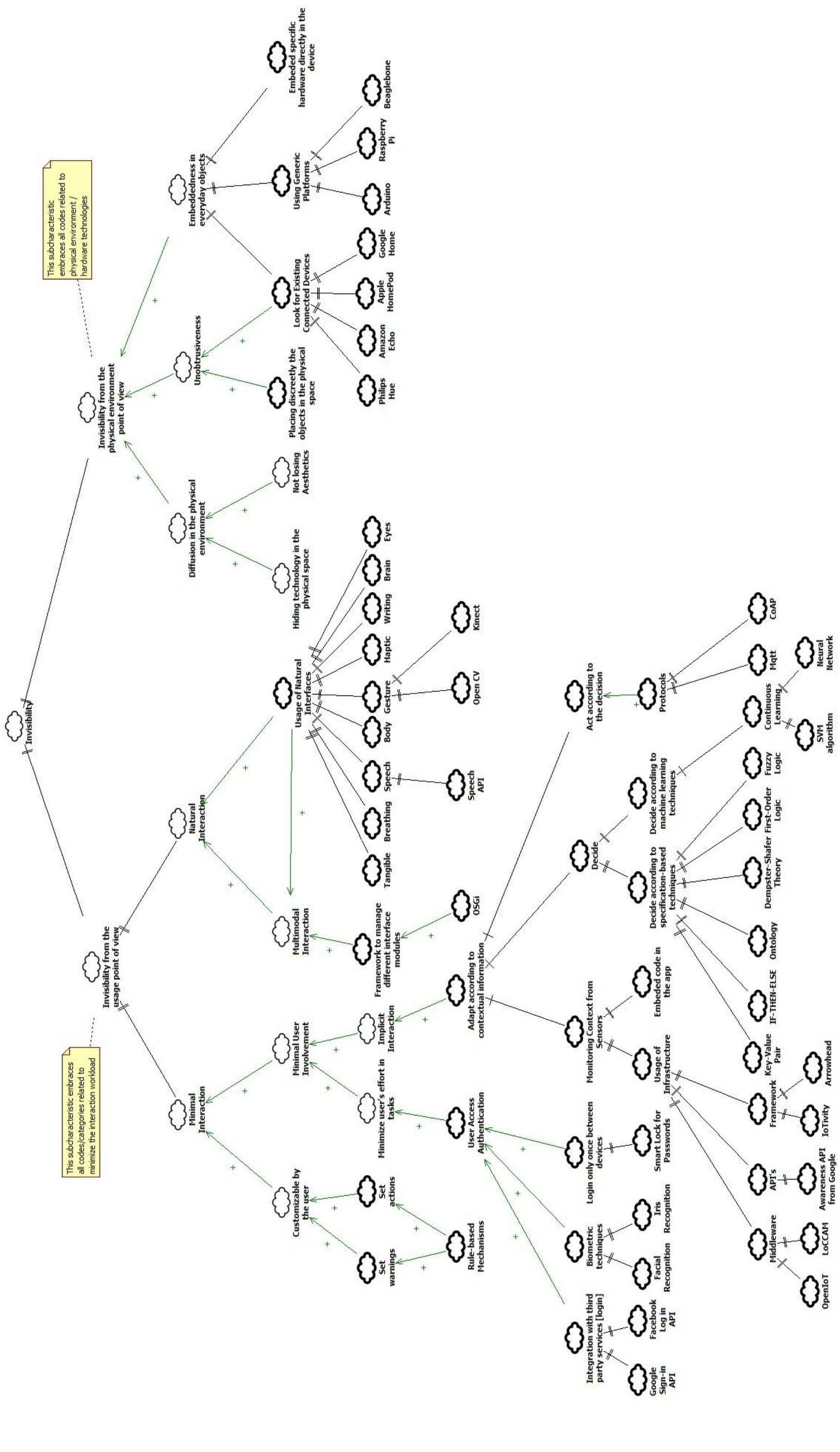


Fig. 10. The Invisibility SIG

TABLE 4. OPERATIONALIZING "IMPLICIT INTERACTION"

General Op. Softgoal	**Adapt According to the Context -** the ability of the system to adapt in response to contextual information [45]. This softgoals can be achieved by 3 softgoals: monitoring, deciding *and* acting.
Specific Operationalizing Softgoals ↓	
1. **Monitoring:** collecting data from sensors	
1.1 Usage of Infrastructure: Sensor data can be collected through context management infrastructures, which encapsulate data access. • **Middleware** (e.g. LoCCAM [46] *or* OpenIoT [47]) • **API** (e.g., Awareness from Google [48]) • **Framework** (e.g., IoTivity [49] *or* Arrowhead [50])	
1.2 Embeded code directly in the app: data collection can be performed inside the application code, without any supporting infrastructure.	
2. Deciding: once the data is collected, there two types of techniques to identify the current context situation: specification-based *or* machine learning-based techniques [51]	
2.1 Specification-based techniques: they are based on *a priori* expert knowledge [51]. Examples of these techniques are: • Key-value pair, If-then-else, Ontology, Dempster-Shafer Theory, First-Order Logic *or* Fuzzy Logic	
2.2 Machine learning-based techniques: they allow learning complex associations between situations and sensor data [51]. It is implemented by continuous learning with: SVM algorithm *or* Neural Network, for example.	
3. Acting: once a situation is detected, the action is automatically performed on the system.	
3.1 Protocols: the interaction with things can be performed through communication protocols, such as MqTT [52] *or* CoAP [53].	

2) ***Natural Interaction***: it refers to supporting more natural and expressively powerful means of interaction by using natural interfaces and letting the user switch between modes of interaction [40]. This support will significantly reduce the workload, since users are used to natural communication in their everyday lives. Two softgoals that can help natural interaction are: (a) Multimodal Interaction and (b) Usage of Natural Interfaces.

a) Multimodal Interaction: the system should support user alternating modes and switch modalities as needed during the changing conditions [40].

b) Usage of Natural Interfaces: This softgoal helps the system to achieve a more natural communication by using interfaces that are natural for the human [40][54]. Besides helping the Natural Interaction softgoal, it also *helps* Multimodal Interaction, by supporting different modalities.

Table 5 presents the operationalizing softgoals for Usage of Natural Interfaces and Multimodal Interaction.

C. Invisibility from the physical environment point of view

It refers to the merging of the technological infrastructure in the physical space to ubiquitously support their users [25]. This way, this softgoal is *helped* by three others: *Embeddedness in everyday objects, Diffusion in the physical environment* and *Unobtrusiveness* (See Fig. 10).

1) *Embeddedness in everyday objects:* It refers to augmenting everyday objects with sensors, processing and communication without compromising the primary functions [55]. There are three strategies (general operationalizing softgoals) to embed computational power into objects: *(i)* using generic platforms capable of embedding sensors and actuators, such as Arduino [56], Raspberry PI [57] *or* Beaglebone [58]; (ii) using devices already equipped with sensors and communication, such as Philip Hues [59], Amazon Echo [60], Apple Speaker [61] or Google Home [62]; *or* (iii) embedding specific hardware directly into objects, which requires specific knowledge about microelectronics.

TABLE 5. OPERATIONALIZING "NATURAL INTERACTION" AND "MULTIMODAL INTERACTION"

General Op. Softgoal	**Usage of Natural Interfaces:** the system should support more natural human forms of communication, which are referred as natural interfaces.
Specific Operationalizing Softgoals ↓	
Speech is the ability to interact by voice. Google Speech API [63] is one option to achieve this softgoals. This API converts speech into text and vice-versa.	
Gestures/Body/Eyes are other examples of natural interfaces. They can be implemented by OpenCV [64], library that provides an infrastructure for computer vision applications, *or* Kinect [65], a programming toolkit that includes rich APIs for raw sensor streams and natural user interfaces.	
Other interfaces are: **Tangible**, **Breathing**, **Haptic**, **Writing**, **Brain**	
General Op. Softgoal	**Framework to manage interface modules**: The application must be modular, allowing the management of each module (e.g., speech, writing, gestures, etc.) at runtime.
Specific Operationalizing Softgoal ↓	
OSGi: framework to life cycle management of dynamic components [66]	

2) *Diffusion in the physical environment*: The computational resources (e.g., data collection devices, communication technologies) should be diffused in the physical environment in order to promote the effective use of services, without compromising the physical space of the user [55]. Two NFR softgoals help in achieving it: (a) Hiding technology in the physical space and (b) Not losing Aesthetics. *Hiding technology in the physical space* is about hiding the hardware infrastructure (e.g., data collection devices, communication devices) from the user in a way that does not catch his/her attention. *Not losing Aesthetics* is related to the physical appearance of the space, which should not be changed.

3) *Unobtrusiveness:* the hardware resources should not be conspicuous or attracting attention. Even with little weight, knowing that a device is present in an environment increases the risk of invading user's personal space, causing discomfort [67]. Therefore, an operationalizing softgoal that can *help* it is:

a) Placing the objects discreetly in the physical space, if the hardware devices are not possible to be entirely hidden, they should be placed discreetly in the physical area. Therefore, places where the user does not need to perform actions, such as wall corners and roofs, would be the ideal.

Another operationalizing softgoal that can <u>*help*</u> Unobtrusiveness is the usage of existing connected devices from popular manufactures such as Philip Hues [59], since they already are specifically designed with sensors and actuators.

V. Discussion

This section presents the main contributions and threats to validity to this research.

A. Contributions

Through a methodology with three phases, it was possible to define a SIG for *Invisibility*[4]. Therefore, the contributions of this work have implications regarding both the SIG and the used methodology.

This SIG is a catalog of requirements for *Invisibility*. This catalog is then composed of subcharacteristics and solutions. Since there is a lack of well-defined taxonomy for *Invisibility*, this SIG contributes to filling in this gap in the literature, by defining an organized body of knowledge.

One may think that we have not shown a usage of this catalog for an application development, however, it is not our goal in this work to reuse this knowledge, but to define it through rigorous research procedures.

The reason why we highlight the methodology as a contribution is mainly because the existing studies do not explain in detail how they identify and relate the concepts (*i.e.*, subcharacteristics and solutions). When they do it through systematic review principles, they do not use a research method to analyze data to reach a set of well-defined concepts. The lack of systematic analysis leads to believe the SIGs can be biased and not grounded.

The usage of GT as a method to extract and relating concepts from the literature data fits perfectly into the structure of a SIG (NFR softgoal, sub softgoals, operationalizing softgoals), which is a pyramid of knowledge. Therefore, each concept extracted from GT can become a softgoal. This process leads to a SIG, based on and grounded in data, and to the possibility of refining the concept of a characteristic, as we achieved by refining the *Invisibility* definition.

Although GT is very suitable in defining a structure capable of being directly represented as a SIG, it may not bring specialized knowledge (*i.e.*, specific operationalizing softgoals). Therefore, it is important to apply another method to complement the SIG defined by GT. We choose to use questionnaires due to the availability of experienced developers. Their answers brought a significant amount of specific solutions, but despite that, some NFR softgoals (*e.g.*, hiding technology, aesthetics and unobtrusiveness) were not possible to be specialized by them, mainly because of their experience, which is more related to software development than to hardware issues. Nevertheless, the resulting SIG is flexible enough to be refined.

B. Threats to Validity

Although we used rigorous research methods, there still are threats to the validity of this work. The first one is related to not having considered all existing studies about *Invisibility*, which may lead to incomplete results. To mitigate it, we did not use only papers from a systematic mapping study [7], but also from two types of snowballing: backward and forward. Additionally, we performed a search in Scopus, an online database which indexes other important computer science online databases, such as IEEE [37][68].

Then, the coding process in GT could have had many biases, since only one researcher analyzed the data and extracted concepts. To mitigate this threat, two HCI experts took part of the process by evaluating all codes from the open coding and all relations from the axial and selective coding. Also, a concern regarding the SIG was about the description of each softgoal (*Invisibility* requirements) being grounded in data. Then, we based our descriptions only on the data that came from the papers and the experts.

Finally, the interpretation of the results from questionnaires was performed by only one researcher, which may lead to misinterpretation of the operationalizing softgoals. To mitigate this threat, every time a doubt or uncertainty about any information from the answers arose, the developers were contacted to clarify the information.

VI. Conclusion and Further work

Invisibility is one of new NFRs brought by the advance of UbiComp and IoT. Although it has been pointed out as an essential characteristic for UbiComp and IoT, the literature lacks a single body of knowledge about how to achieve these characteristics during software development. Therefore, the goal of this work aimed at capturing subcharacteristics and solutions for *Invisibility* and cataloging them in a SIG.

However, there is no systematic approach on how to build this kind of catalog. Therefore, we also proposed to systematize the definition of *Invisibility* SIG by using a set of well-defined research methods: snowballing, database search, grounded theory and questionnaires. As the main contribution, we got an *Invisibility* SIG composed of 2 subcharacteristics, 12 sub-subcharacteristics, 10 general solutions and 56 specific solutions.

Although different researchers may have a different understanding of the data through GT, leading to another interpretation of the concepts, the methodology used allows a systematization about how to define a SIG of NFR not well characterized in the literature. Additionally, the resulting catalog is flexible and can evolve to include other concepts.

As further work, we will expand the SIG by adding the conflicts (*i.e.*, negative relations) between the softgoals of *Invisibility* and other NFRs, such as *Usability*. Also, the resulting catalog will be evaluated regarding its usefulness.

[4] All files related to the outcomes of these phases are available on https://github.com/GREatResearches/InvisibilitySIG

References

[1] M. Weiser, "The Computer for the 21st Century," *Sci. Am. Spec. Issue Commun. Comput. Networks*, pp. 94–104, 1991.

[2] J.-P. Vasseur and A. Dunkels, *Interconnecting Smart Objects with IP*. Morgan Kaufmann, 2010.

[3] I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo, and D. P. Agrawal, "Choices for interaction with things on Internet and underlying issues," *J. Ad Hoc Networks*, vol. 28, pp. 68–90, 2015.

[4] C. Bodei, P. Degano, G.-L. Ferrari, L. Galletta, and G. Mezzetti, "Formalising security in ubiquitous and cloud scenarios," in *IFIP International Conference on Computer Information Systems and Industrial Management*, 2012.

[5] R. M. Carvalho, R. M. de C. Andrade, and K. M. de Oliveira, "AQUArIUM - A suite of software measures for HCI quality evaluation of ubiquitous mobile applications," *J. Syst. Softw.*, vol. 136, pp. 101–136, 2018.

[6] K. Wiegers and J. Beatty, *Software requirements*. Pearson Education, 2013.

[7] R. M. Carvalho, R. M. de C. Andrade, K. M. de Oliveira, I. de S. Santos, and C. I. M. Bezerra, "Quality characteristics and measures for human–computer interaction evaluation in ubiquitous systems," *Softw. Qual. J.*, vol. 25, no. 3, pp. 743–795, 2017.

[8] M. Satyanarayanan, "Pervasive computing: vision and challenges," *IEEE Pers. Commun.*, 2001.

[9] H. Karvonen and T. Kujala, "Designing and Evaluating Ubicomp Characteristics of Intelligent In-Car Systems," in *5th International Conference on Applied Human Factors and Ergonomics*, 2014, no. July, pp. 1726–1737.

[10] D. Saha and A. Mukherjee, "Pervasive computing: A paradigm for the 21st century," *IEEE Comput. Soc.*, vol. 36, no. 3, pp. 25–31, 2003.

[11] C. A. Da Costa, A. C. Yamin, and C. F. R. Geyer, "Toward a general software infrastructure for ubiquitous computing," *IEEE Pervasive Comput.*, vol. 7, no. 1, pp. 64–73, 2008.

[12] J. Scholtz and S. Consolvo, "Toward a Framework for Evaluating Ubiquitous Computing Applications," *IEEE Pervasive Comput.*, vol. 3, no. 2, pp. 82–88, 2004.

[13] R. M. Carvalho *et al.*, "Evaluating an IoT application using software measures," in *International Conference on Distributed, Ambient, and Pervasive Interactions*, 2017.

[14] R. M. C. Andrade, R. M. Carvalho, I. L. de Araújo, K. M. Oliveira, and M. E. F. Maia, "What changes from ubiquitous computing to internet of things in interaction evaluation?," in *International Conference on Distributed, Ambient, and Pervasive Interactions*, 2017.

[15] J. C. S. do Prado Leite and C. Cappelli, "Software transparency," *Bus. Inf. Syst. Eng.*, vol. 2, no. 3, pp. 127–139, 2010.

[16] M. Serrano and J. C. S. do Prado Leite, "Capturing transparency-related requirements patterns through argumentation," in *Requirements Patterns (RePa), 2011 First International Workshop on*, 2011, pp. 32–41.

[17] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-functional requirements in software engineering*, vol. 5. Springer Science & Business Media, 2000.

[18] L. M. Cysneiros, V. M. Werneck, and A. Kushniruk, "Reusable knowledge for satisficing usability requirements," *13th IEEE Int. Conf. Requir. Eng.*, 2005.

[19] M. E. F. Maia, L. S. Rocha, and R. M. C. Andrade, "Requirements and challenges for building service-oriented pervasive middleware," *Int. Conf. Pervasive Serv. - ICPS*, 2009.

[20] Herbet de Souza Cunha, "Desenvolvimento de Software Consciente com Base em Requisitos," PhD Thesis from Pontifícia Universidade Católica do Rio de Janeiro, Brazil. (in portuguese), 2014.

[21] H. S. Gorbin, M, Strauss, *Basics of Qualitative Research: Techniques and procedures for developing grounded theory.*, 3rd ed. Sage Publications, 2008.

[22] R. Poppe, R. Rienks, and B. Van Dijk, "Evaluating the Future of HCI: Challenges for the Evaluation of Emerging Applications," *Artifical Intell. Hum. Comput. Lect. Notes Comput. Sci.*, vol. 4451, pp. 234–250, 2007.

[23] D. Leung and D. Song, "Smart Locks : Lessons for Securing Commodity Internet of Things Devices," in *11th ACM on Asia Conference on Computer and Communications Security*, 2016, pp. 461–472.

[24] P. E. Kourouthanassis, G. M. Giaglis, and D. C. Karaiskos, "Delineating the degree of 'pervasiveness' in Pervasive Information Systems: An assessment framework and design implications," in *Pan-Hellenic Conference on Informatics*, 2008.

[25] D. Karaiskos, P. Kourouthanassis, and G. M. Giaglis, "Towards a Validated Construct for Information Systems Pervasiveness: An Exploratory Assessment," in *BLED Proceedings. Paper 12*, 2009.

[26] E. A. Kemp, A.-J. Thompson, and R. S. Johnson, "Interface Evaluation for Invisibility and Ubiquity: An Example from E-learning," in *Proceedings of the 9th ACM SIGCHI New Zealand Chapter's International Conference on Human-Computer Interaction: Design Centered HCI*, 2008, pp. 31–38.

[27] S. G. Thompson and B. Azvine, "No Pervasive Computing Without Intelligent Systems," *BT Technol. J.*, vol. 22, no. 3, pp. 39–49, 2004.

[28] H. Ryu, G. Y. Hong, and H. James, "Quality assessment technique for ubiquitous software and middleware," *Res. Lett. Inf. Math. Sci.*, vol. 9, pp. 13–87, 2006.

[29] I.-Y. Ko, H.-M. Koo, and A. Jimenez-Molina, "User-centric Web Services for Ubiquitous Computing," in *Advanced Techniques in Web Intelligence - I. Studies in Computational Intelligence*, Springer Berlin Heidelberg, 2010, pp. 167–189.

[30] R. M. Santos, K. M. Oliveira, R. M. C. Andrade, I. S. Santos, and E. R. R. Lima, "A Quality Model for Human-Computer Interaction Evaluation in Ubiquitous Systems," in *Latin American Conference on Human Computer Interaction*, 2013.

[31] Y. Yu, J. Mylopoulos, E. Yu, J. C. Leite, L. L. Liu, and E. D'Hollander, "Software Refactoring Guided By Multiple Soft-Goals," *Int. Work. REFactoring Achiev. Challenges, Eff. (REFACE),* no. June, pp. 876–880, 2003.

[32] M. Serrano, "Ubiquitous , Pervasive and Mobile Computing : A Reusable-Models-based Non-Functional Catalogue Objectives of Research," in *ER@ BR*, 2013.

[33] B. A. Kitchenham and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software Engineering," in *Technical Report. EBSE-2007-01, Keele University,* 2007.

[34] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Inf. Softw. Technol.*, vol. 64, pp. 1–18, 2015.

[35] C. Wohlin, "Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering," *18th Int. Conf. Eval. Assess. Softw. Eng. (EASE 2014)*, pp. 1–10, 2014.

[36] I. de Sousa Santos, R. M. de Castro Andrade, L. S. Rocha, S. Matalonga, K. M. de Oliveira, and G. H. Travassos, "Test Case Design for Context-Aware Applications: Are We There Yet?," *Inf. Softw. Technol.*, 2017.

[37] J. Horkoff *et al.*, "Goal-Oriented Requirements Engineering: A Systematic Literature Map," in *IEEE 24th International Requirements Engineering Conference*, 2016, pp. 106–115.

[38] K. Stol, P. Ralph, and B. Fitzgerald, "Grounded Theory in Software Engineering Research : A Critical Review and Guidelines," *Proc. 38th Int. Conf. Softw. Eng. - ICSE '16*, no. Aug, pp. 120–131, 2016.

[39] B. Abdulrazak and Y. Malik, "Review of challenges, requirements, and approaches of pervasive computing system evaluation," *IETE Tech. Rev.*, vol. 29, no. 6, p. 506, 2012.

[40] G. Yue, W., Wang, H., & Wang, "Designing Transparent Interaction for Ubiquitous Computing," *Int. Conf. Human-Computer Interact. Interact. Des. Usability. Lect. Notes Comput. Sci.*, vol. 4550, pp. 331–339, 2007.

[41] "Google Sign-In." [Online]. Available: https://developers.google.com/identity/sign-in/android/. [Accessed: 26-Feb-2018].

[42] "Facebook Login & Account Kit." [Online]. Available: https://developers.facebook.com/products/account-creation. [Accessed: 26-Feb-2018].

[43] "Smart Lock for Passwords on Android." [Online]. Available: https://developers.google.com/identity/smartlock-passwords/android/. [Accessed: 26-Feb-2018].

[44] P. Giner, C. Cetina, J. Fons, and V. Pelechano, "Implicit interaction design for pervasive workflows," *Pers. Ubiquitous Comput.*, vol. 15, no. 4, pp. 399–408, 2011.

[45] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, "Interaction Issues in Context- Aware Intelligent Environments," *Human-Computer Interact.*, vol. 16, no. 2, pp. 363–378, 2004.

[46] M. E. F. Maia, A. Fonteles, B. Neto, R. Gadelha, W. Viana, and R. M. C. Andrade, "LOCCAM - Loosely Coupled Context Acquisition Middleware," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 2013, pp. 534–541.

[47] "OpenIoT." [Online]. Available: http://www.openiot.eu/#. [Accessed: 26-Feb-2018].

[48] "Google Awareness API." [Online]. Available: https://developers.google.com/awareness/. [Accessed: 26-Feb-2018].

[49] "IoTIVITY." [Online]. Available: https://www.iotivity.org. [Accessed: 26-Feb-2018].

[50] "Arrowhead." [Online]. Available: http://www.arrowhead.eu/. [Accessed: 26-Feb-2018].

[51] J. Ye, S. Dobson, and S. McKeever, "Situation identification techniques in pervasive computing: A review," *Pervasive Mob. Comput.*, vol. 8, no. 1, pp. 36–66, 2012.

[52] "MQTT." [Online]. Available: http://mqtt.org/. [Accessed: 26-Feb-2018].

[53] "Constrained Application Protocol (CoAP)." [Online]. Available: http://coap.technology/. [Accessed: 26-Feb-2018].

[54] E. Kaasinen, T. Kymäläinen, M. Niemelä, T. Olsson, M. Kanerva, and V. Ikonen, "A User-Centric View of Intelligent Environments: User Expectations, User Experience and User Role in Building Intelligent Environments," *Computers*, vol. 2, no. 1, pp. 1–33, 2013.

[55] M. Beigl, H. Gellersen, and A. Schmidt, "MediaCups : Experience with Design and Use of Computer-Augmented Everyday Artefacts," *Comput. Networks*, vol. 35, no. 4, pp. 401–409, 1998.

[56] "Arduino." [Online]. Available: https://www.arduino.cc/. [Accessed: 26-Feb-2018].

[57] "Raspberry Pi." [Online]. Available: https://www.raspberrypi.org/. [Accessed: 26-Feb-2018].

[58] "BeagleBone." [Online]. Available: http://beagleboard.org/bone. [Accessed: 26-Feb-2018].

[59] "Philips Hue Lights." [Online]. Available: https://www.philips.co.in/c-m-li/hue. [Accessed: 26-Feb-2018].

[60] "Amazon Echo Plus." [Online]. Available: https://www.amazon.com/dp/B075RWFCHB/ref_=fs_ods_fs_aucc_sr. [Accessed: 26-Feb-2018].

[61] "Apple HomePod." [Online]. Available: https://www.apple.com/homepod/. [Accessed: 26-Feb-2018].

[62] "Google Home." [Online]. Available: https://store.google.com/gb/product/google_home. [Accessed: 26-Feb-2018].

[63] "Speech API." [Online]. Available: https://cloud.google.com/speech/. [Accessed: 26-Feb-2018].

[64] "Open Source Computer Vision Library - OpenCV." [Online]. Available: https://opencv.org/. [Accessed: 26-Feb-2018].

[65] "Kinect for Windows SDK beta." [Online]. Available: https://www.microsoft.com/en-us/research/project/kinect-for-windows-sdk-beta/. [Accessed: 26-Feb-2018].

[66] "he Dynamic Module System for Java - OSGi." [Online]. Available: https://www.osgi.org/. [Accessed: 26-Feb-2018].

[67] S. Moran and K. Nakata, "Analysing the factors affecting users in intelligent pervasive spaces," *Intell. Build. Int.*, vol. 2, no. 1, pp. 57–71, 2010.

[68] IEEE, "Is IEEE Xplore digital library content indexed in SCOPUS?" [Online]. Available: https://supportcenter.ieee.org/app/answers/detail/a_id/510/~/is-ieee-xplore-digital-library-content-indexed-in-scopus%3F. [Accessed: 26-Feb-2018].