UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Máster en Ingeniería Informática

Trabajo de Fin de Máster

Situational Context: Gestión distribuida de información contextual

Ángela Barriga Rodríguez

Febrero, 2017

UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Máster en Ingeniería Informática

Trabajo de Fin de Máster

Situational Context: Gestión distribuida de información contextual

Autor: Ángela Barriga Rodrǵuez

Fdo:

Directores: Juan Manuel Murillo Rodríguez y Javier Berrocal Olmeda

Fdo:

Tribunal Calificador

Presidente: José María Conejero Manzano

Fdo:

Secretario: José Manuel García Alonso

Fdo:

Vocal: Pedro José Clemente Martín

Fdo:

Resumen

A lo largo de este trabajo de fin de máster se presenta un proceso de investigación sobre sistemas conscientes del contexto, interacción humano-máquina e Internet de las Cosas con el fin de crear una definición apropiada de lo que es el contexto situacional.

Por otro lado y a partir de dicha definición, se introduce la propuesta y diseño de un marco teórico para crear entornos de Internet de las Cosas centrados en el usuario, guiado por diferentes corrientes ideológicas y psicológicas sobre la comunicación y actividad humana.

Asimismo, se plantea una arquitectura genérica basada en dicho marco, presentando qué componentes y servicios la componen y qué tecnologías reales podrían dar solución a su planteamiento, tales como librerías, paradigmas y modelos. El planteamiento de dicha arquitectura es lograr la gestión distribuida de información contextual desde dispositivos móviles.

Palabras Clave: Sistemas conscientes del contexto; Interacción humano-máquina; Internet de las Cosas; Dispositivos móviles

Abstract

Along this master's thesis, it is presented a research process and its results on Context-Aware systems, Human-Computer interaction and Internet of Things in order to present an appropriate definition of what situational context is.

On the other hand and based on this definition, it is introduced the proposal and design of a theoretical framework to create user-centric Internet of Things environments, guided by different ideological and psychological theories about human communication and activity.

Furthermore, a generic architecture based on this framework is built, presenting which are its components and services and which real technologies could give a real solution to its approach, such as libraries, paradigms and models. The path of this architecture is to achieve the distributed management of contextual information from mobile devices.

Keywords: Context-Aware systems; Human-Computer interaction; Internet of Things; Mobile devices

Contents

List of Figures

Chapter 1

Introduction

Humanity tends to define, think and tag time in eras, making it easier to understand under the vision of a lifespan. Looking back at how technology has impacted and changed society, there are three major eras, namely agricultural, industrial and informational [6]. Indeed, the third era could be seen as an extension of the previous one.

The industrial era started back in the decade of 1780, with the emergence of mechanization, water power and steam power (first industrial revolution). Then, mass production, assembly line and electricity appeared (second revolution). Nowadays, society is living the transition between the third revolution marked by the appearing of computer and automation to the fourth one, the establishment of cyber physical systems.

The rise of cyber physical systems comes from the development of different technologies over time within this informational era. Firstly came computer, then the Internet, mobile devices and reaching present time: the Internet of Things (IoT). This evolution has always been somehow intuitive, as each previous technology defined the next one.

However, there is a huge change in this last step of the informational era: the shift from mobile devices hegemony to the Internet of Things is radical because for the first time in history the primary users of technology will not be humans. This means that a smart, connected machine (or digitalized environment) could evolve itself by assimilating the most useful data from its users or environment and the most successful code from other machines. In other words: awareness, specifically context-awareness is the key to understand the 'IoT era'.

In general, IoT promotes a heightened level of awareness about the world, and a platform from which to monitor the reactions to the changing conditions that said awareness expose. And, like the advent of the Internet itself, IoT enables myriad applications ranging from the micro to the macro, and from the trivial to the critical. This myriad open endless possibilities to society for improving and growing, making everybody's lives easier.

Therefore, is of vital importance defining properly this context and how machines could work with it. And, although the primary users of this technology will not be humans, it will be controlled, developed and designed by them and its main purpose is to serve them. Accordingly, IoT design should always be guided by users' comfort without making the mistake of setting progress and technology above humanity.

Just as it happened before with computers, Internet and mobile devices, it seems intuitive that IoT revolves around its predecessor technology, namely smartphones, which in turn were based and guided by Internet. [7]

Hence, in a desirable future, there should be a generic IoT paradigm for designing architectures and systems, which could assure that all systems are guided by this human-centric idea, without losing the perspective of progress and evolution.

This thesis aims to propose a clear definition of context, in this occasion named 'Situational Context' and to introduce this previously mentioned desirable paradigm. Situational Context will provide a definition for human-centric context, based on the situation where the user is, defining it by its place in space and time and the elements existing in such environment.

The document follows the next structure: firstly, the general objectives of this thesis are stated. Then the state of art is exposed, showing the most relevant research lines within context-awareness and Internet of Things systems. It is followed by the section 'Material and Method' which explains the methodology followed during this work, all the theory studied from context-awareness, IoT, APIs to HCI psychology, the definition for Situational Context, its technical approach and some conclusions. Then comes the 'Results and Discussion' section, which contains the final scope of this project, the definition of components and services for the proposal, general results and a brief discussion about them. Finally, there is a 'Conclusions' section which states the general conclusions from this thesis, its future work and a personal reflection from the author.

Chapter 2

Objectives

The objectives of this thesis are:

- To create an appropriate definition of the concept 'Situational Context', making it complex enough to include all possibilities within this area.
- To define what would make a good IoT system, identifying what makes the user uncomfortable and what suits him better. Therefore, following a human-centric design. Also, to identify other guidelines to create a strong and secure ecosystem.
- To build a theoretical framework for designing IoT systems following the previous guidelines.
- To design a generic architecture for developing IoT environments, based on the previous theoretical framework and oriented to be managed from the smartphone.
- To propose accurate, real techniques and solutions for this architecture, able to add features and solve problems.

Chapter 3

State of the Art

Internet of Things is an ever growing field. There are an impressive number of proposals, brands, startups and researching efforts revolving around this topic and it is not surprising given its economic expectatives. However, within this growth comes a problem: the lack of standardization. Each company develop their own systems, both in software and hardware, so that users can only control an IoT device with the software of that same company. This diversity is translated directly in many different researching paths but finding a generic solution is not an issue easily solved, especially when it is applied to an user-centric IoT system; where its contextual information is a key element; due to its changing nature and the difficulty of its measurement.

Some generic proposals think in a big scale, for example in [8] which under the view of smart cities claims the urge to unify the design of IoT architectures. Smart cities make use of many different IoT areas (agriculture, energy, environment protection, health, home automation, etc.), so in order to reduce investments, top-down architectural principles need to be followed. This paper proposes a top-level generic IoT architecture particularly suited for the creation of smart cities.

Other reserach lines, try to generalize IoT development by using ontologies, web semantics, domain, markup and metalanguages. As domain-specific IoT applications use their own technologies and terms to describe sensors and their measurements, it is a difficult task to help users build generic IoT applications to combine several domains.[9] These techniques aim to create code generation tools so the IoT development could be automatized, as in [10], where it is exposed how context-awareness can be expressed at the programming language level with a basis on four main abstractions: context, adapters, adaptation commands, and adaptive behavior management policies. Web semantics are managed in some projects by an artificial intelligence library called Tensorflow [11], powered by Google.

Surveys as [12], can offer a wide vision on how there is no consensus even when trying to find a unified paradigm design. The main problem is the variety of situations where IoT can be applied, making easier to develop an specific architecture for each problem. There are many different solutions: three/four layers architectures, middlewares, publish-subscribe models, event-oriented systems... But most of them are oriented through traditional software development, are not aware of context and cannot adapt themselves to different situations, and what is most important, do not take into account the human factor.

This thesis will follow the path of [13]: 'Human-centric rather than Thing-centric'. Current IoT architectures are device or network oriented due to their operational significance in an IoT system. However, two key aspects that are often ignored are humans who are part of this ecosystem and the context within which interaction between people and things take place.

Human-centric systems are explored in papers as [14, 15] where is proposed the model of Internet of People (IoP). These papers state that the main objective in developing applications for the Internet of Things is to integrate technology into daily life. IoP proposes a manifesto for IoT development: 'Be

Social, Personalized, Proactive and Predictable'. They believe that following those four simple rules IoT development could be human friendly in all aspects. But what it is quite relevant about [15] is that their proposal is not only human-centric and take context into account, but it is also smartphone-centric.

Smartphones provide a great solution for distributed contextual information management, as they are a core element in almost any users' daily lives. Their capability is improving drastically each year, and they can process all that an IoT system needs without suffering a big impact on their performance or battery [16].

However, despite the great opportunities this platform can offer, researchers do not propose a generic architecture or a paradigm for managing IoT systems with smartphones, moreover, it is again hard to find consensus on how to work with contextual information when interacting with mobile platforms [17].

Some papers propose the same solutions as before for the creation of a generic architecture, such as ontologies [18] or middlewares [19, 20], but they do not take into account the contextual dimension.

Those who take it into account are usually centered in the data provided by the smartphone's GPS, namely location data [21, 22]. This type of data is quite simple to process and can provide many information from the user, as most aspects in daily life follow patterns and are determined by time and space. Other studies as [23], determine the context of the user depending on which artifacts or IoT devices are being used, being monitored through the smartphone.

Other lines emphasize in the lack of security, privacy and integrity that could have smartphone-centric systems [24] and provide some guidelines for developing secure and friendly systems. This topic was also explored by IoP and People as a Service, presented before [14, 15].

As a summary, the state of the art reviewed for this thesis was quite wide, as it covers many areas: IoT, Context-Awareness, generic architectures or paradigms for IoT development, Human-Computer Interaction (HCI) and mobile-centric IoT systems.

Despite the complexity and variety on IoT and contextual researching there seems to be some points in common. A desirable IoT scenario should be human-centric, protecting its privacy and security. Besides, it should act accordingly to context, which can be defined by time and space and also by the user's behaviour. And finally, should be able to adapt itself to distributed systems and different situations, which can be solved by centralizing the IoT system in a smartphone.

In conclusion, there are many proposals but a great part of them are old-fashioned and do not use contextual information. During the review of literature and its research there were not found any papers proposing a generic IoT system centered on the smartphone, able to act accordingly to context.

This review will guide the development of the thesis, helping to limit its research horizon and providing a solid base for start building the proposal 'Situational Context: Distributed Management of Contextual Information'.

Chapter 4

Material and Method

Content

During this section there will be exposed all the development phases of this thesis. The first subsection will explain which methodology was followed, after it there is a review of different researchs and information related to context-awareness and IoT; stressing which different lines of work exist and what challenges these topics present, as well as their business opportunities. Then some context-awareness APIs are reviewed, followed by the revision of different psychological theories related to HCI. Finally, it is presented the definition of Situational Context, its technical approach and what can be concluded from this section.

All the work done during these months is presented inside this section as the value of a thesis is not only its result but also the way travelled to reach it.

4.1 Methodology

This thesis was developed between march and december of 2016, making a total of ten months of work.

The first four months (from march to june) were focused on the review of literature, because as it was such a wide topic, it was necessary a really deep research in order to define what would be the best proposal to work on.

The next step, once delimited the scope of the thesis, was to define the theoretic framework that would be the basis of the work. This stage followed an incremental and iterative methodology, as the proposal was evaluated and redefined in each iteration, being perfected with the feedback from the director and co-director, until it covered all the desirables aspects for this work. As before, this phase occupied a period of four months, from july to september.

Then it came the design phase. Once the theoretical basis was clear, to conclude the work it was necessary to define an architectural design. It was followed the same methodology as before, the design was reviewed from week to week by the directors and it was corrected and developed further until there was a complete solution. This phase includes work from october to november, that is to say, two months.

Finally, december was dedicated to writing the thesis documentation.

All the work has been done in SPILab, under the supervision and with the help of Juan Manuel Murillo Rodriguez, Javier Berrocal Olmeda and Jose Garcia-Alonso.

4.2 From Context-Awareness to the Internet of Things

Initially the scope of this thesis was not fully defined, although it was clear the main purpose would be to define a generic architecture for IoT systems, aware of context. In order to do that, it was necessary to read as much as possible about IoT and context-awareness, without discarding old papers, as having a general vision on how this topic evolved during time would contribute on forming a complete and general vision that would help to build a better thesis. Therefore, this phase started with a research stage from which was selected papers written between 1991 and 2015.

This section means to give a general vision on how IoT, context-awareness and ubiquitous computing were defined and evolved. Also it will explore which different lines of work in IoT exists today, which challenges they have to overcome and what opportunities are opened for the future.

4.2.1 Definitions

Talking about IoT is inconceivable without taking into account Mark Weiser's work, who from the beginning of the 90s stated that future would be in ubiquitous computing and context-awareness. Weiser stated that 'The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.' [25] Indeed, he compared technology with writing. Writing was an skill quite rare in the past, very few people was able to read and write and what is more, writing was not easily found, existing only in sacred texts or manuscripts. However time changes, and nowadays everybody is able to read and write, writing is found everywhere (magazines, signals...) and it has become an automatized process. For Weiser, technology would walk the same path as writing, and his vision is becoming more true each day: 'The goal is to achieve the most effective kind of technology, that which is essentially invisible to the user... I call this future world 'Ubiquitous Computing' (Ubicomp)'. [26]

For a complete understanding of IoT it is also necessary to properly define what is context. According to [27] context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. Also, context would be defined by four categories: location, identity, activity and time, namely questions where, who, what and when.

Given that definition, context-awareness computation would be a system able to present information and services to users (understanding context as information), automatic execution of services and link context with information for its later retrieval. There appears a really interesting division between implicit and explicit context, being implicit context extractable directly from a user (e.g., a registration form, configuration settings) and explicit all information provided by environment (sensors, IoT devices, behaviour).

Internet of Things appeared as a concept for the first time in 1999, coined by Kevin Ashton [28]. He referred to a global network of devices connected by RFID technology.

Lately in 2001 ISTAG (Information Society Technology Advisory Group) came up with the term ambient intelligence. The term refers to electronic environments that are sensitive and responsive to the presence of people. [29]

Despite the different names given, IoT basis were clear back in early 2000s as many authors gave similar definitions. A context-awareness IoT system is an ubiquitous system able to understand the situation in which its user is and act accordingly to it.

4.2.2 Lines of Work

Analyzing the previous papers, and also others surveys like [30, 31, 32] one can see that there are almost infinite possibilities for working with IoT, appearing very different lines of work, although many of them go towards the same direction.

Most researched themes include domotic systems, gerontechnology and health applications, business systems, decision making systems, and automatization of processes.

Almost all proposals are decanted by distributed architectures able to adapt themselves to dynamic contexts. These architectures are made of interconnected devices which communicate with each other and with the centralized system. In conclusion, the tendency is to create pervasive systems (definition in [33]).

Some common technologies or tools used among the proposals are middlewares, publish-subscribe models, if-then models, logic rules, statistics tools, Markov chains, decision trees, neural networks and ontologies.

Many research efforts focus on behaviour prediction, automated learning, environment self-discovering, self-organization (as the number of devices and sensors grow each year more and more and they are very different), objective orientedness (non commanded) and knowledge extraction.

Some of these lines will be explored during this thesis especially environment self-discovering, self-organization and objective-orientedness.

4.2.3 Challenges

Despite the great opportunities that provide this kind of technology, there are many challenges that are yet to overcome. Provided that the implantation of these kind of systems into the daily live suppose a huge change in society, it is normal that it wakes some fears and worries. The main problem is that each scenario and application of IoT is quite different, as much as people and their activities are. Indeed, many systems are tailor-made to its area of application. Generalization is actually really hard, one of its causes is the absence of a generic architecture or paradigm design for IoT structures. Therefore, this is translated with long and intensive stages of analysis and study when designing them; which involve more budget and time in developing.

Speaking about budgets, obsolescence is a reality when talking about ubiquitous computing. Investment is hard when buying hardware, mainly sensors, as they evolve really quickly and sometimes in a matter of a couple of years they are old-fashioned, but there is no way to actually solve this. Indeed, during this thesis research, many papers talked about technologies that are not used anymore such as PDAs.

But while problems related to technology are difficult to deal with, are many other issues related to society misunderstanding and fears that should be the focus during next years.

Main people concerns are related to privacy invasion and security. It is necessary to develop systems that protect users data (as stated in [15]) and to create new laws and politics to assure the fulfillment of some good practices that provide quality and integrity for IoT developments. Companies should have a limit in how they can monetize users' data without their consent.

There is a common fear to technological alienation, mainly impulsed by social media and science fiction. Some automated systems might be seen as a threat to human development and evolution, as they provide the mechanism to avoid users dealing with routinary tasks. Some people perceive this as a loss of control and human capabilities. There is also a huge fear of technology leading to isolation instead of connectivity.

Besides, science fiction has put expectations too high, leading to disappointment when users see real IoT systems.

Another problem is that sometimes IoT tends to solve some needs that are not real needs, there will always be some situations that will be better resolved by a human being (at least with current technology).

As a summary, it is necessary to focus on the user, on the human component of this technology. This can be accomplished by analyzing real data of what the user needs, avoiding stereotypes. And given that the human factor comes first, caring its security and privacy is vital, developing transparent and user-friendly systems.

4.2.4 Business Opportunities

Internet of Things provide lot of opportunities, overall speaking in economic terms. According to [34], the prediction is that the world will have 50 billion connected devices by 2020. Keeping in mind that world population will be of approximately 7.6 billion people in that year this means there will be a media of 6,58 connected devices per person (today it is estimated to be around 3,5 devices). The numbers speak by themselves.

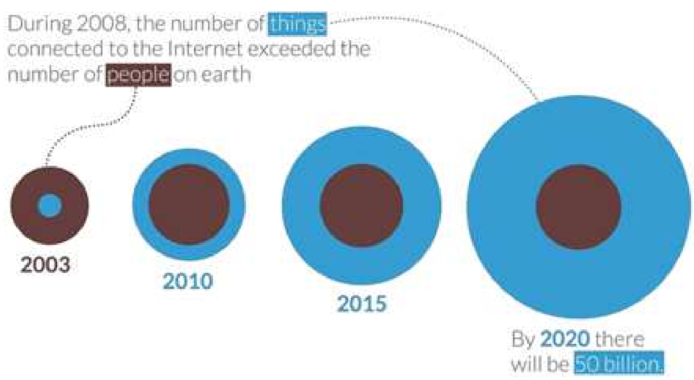

Figure 4.1: Relationship between population and Internet devices [1].

It is estimated that IoT market has a 4 trillion dollars revenue opportunity as there will be 4 billion people connected worldwide. From here to 2020 more

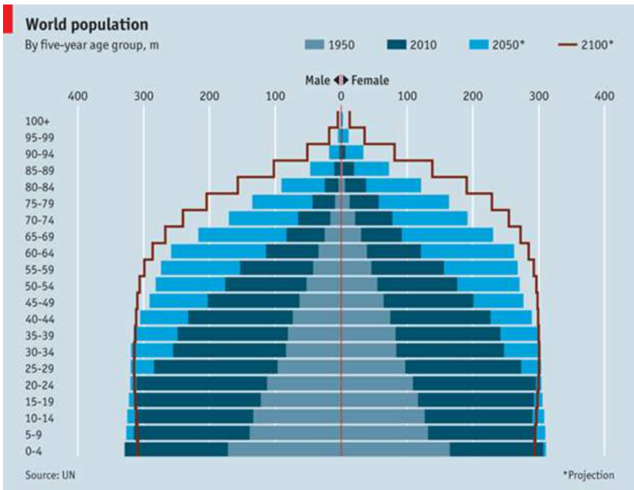

Figure 4.2: World's demopgraphic profile until 2100 [2].

than 25 million applications will be developed and people will create more than 50 trillion gigabytes of data. It is expected an annual growth of the 8 per cent in this sector. This growth can be seen in the graph 4.3.

Furthermore, one of the lines of work with more revenue expectations is gerontechnology, as from here to 2020 population over 60 years will be nearly the same as newborns (Figure 4.2).

Provided these astonishing numbers it is clear that IoT is the future and working on it can be really productive both on how it can contribute to society and in earnings.

4.2.5 Conclusions

From all of this research it is possible to extract some conclusions which will guide the development of this thesis.

IoT is mainly a multidisciplinary area, in order to create a good context-aware system is necessary to take into account not only engineers point of view but psychologists, sociologists and even philosophers. Also, it is key to consult experts in the area of application, for example, when working in a hospital to

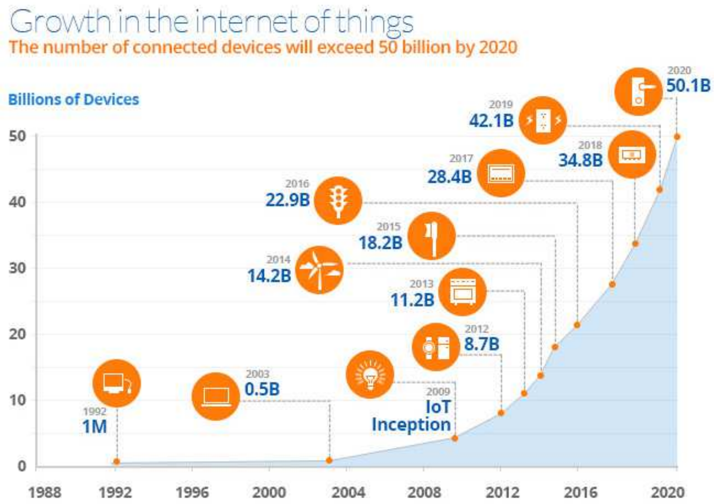

Figure 4.3: Expected growth in the Internet of Things [3].

consult doctors, nurses...

On other side, simplicity is the ultimate sophistication. Sometimes complex problems have simple solutions, and technology should be able to provide them avoiding an excessive complexity when not needed. Indeed there are many simple and automatic tasks in where IoT could be really useful.

Another important point when developing human-centric systems is their social contributions. IoT systems should help people to be connected and they should be not only physically pervasive but also socially.

Technology is not anymore just about satisfying needs but also about people and coexisting in their daily lives. Given that coexistence it is important to develop reliable, empathetic, secure and private systems.

In conclusion, IoT systems can be divided into three dimensions: user, environment and interaction, all three of them are connected being the key element the user.

4.3 Context Awareness API

After this initial research, it is decided to explore further the possibility of developing a context-awareness API for IoT systems. An API would suit all the objectives of this thesis, as it would provide a generic framework for creating IoT applications, always taking into account the context.

But first, in order to decide whether it is worth or not to follow this path, it is necessary to perform some research.

4.3.1 Literature

Published papers do not shed much light into this topic. There does not seem to be much research in this area and the ones that are found do not match exactly the aim of this work. Same problems appear again, many proposals do not research on current technologies or are too centered in developing IoT systems as if they were traditional software.

One example of a good proposal but not adapted to new technologies is [35]. It proposes a Java framework for IoT systems, and despite the proposal being good, it does not suit today's needs. It works mostly with RFID technology, keeping Bluetooth in a second place and not using Wi-Fi at all. Although some good ideas can be extracted from it, as how it catalogues context (person, place, thing, location, status and activity).

As literature does not offer a good picture of the topic, researching continues on the private side.

4.3.2 Google Awareness API

When searching about context awareness APIs, first results found are about Google Awareness API [4]. While it does not provide what this thesis is searching, it is a really useful tool that could be integrated in a bigger system given its nature and features.

In Google words: 'The Awareness API unifies 7 location and context signals in a single API, enabling developers to create powerful context-based features with minimal impact on system resources. Combine optimally processed context signals in new ways that were not previously possible, while letting the API manage system resources so your app doesn't have to.'

This API is divided into two APIs, namely Fence API and Snapshot API; both of them providing quite interesting concepts. (Figure 4.4).

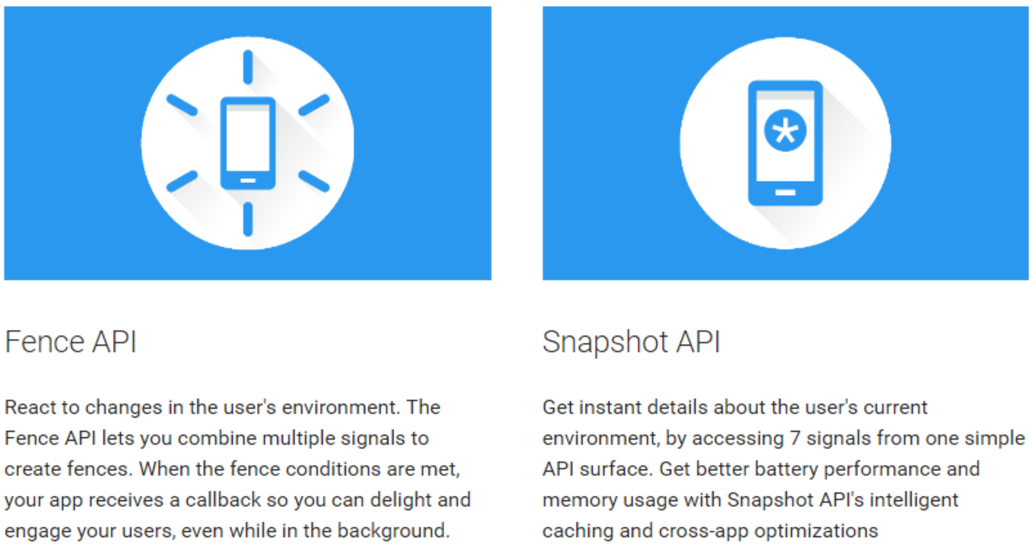

Figure 4.4: Google's Fence API and Snapshot [4]. API

On the one hand, Fence API is able to react to changes in the user's environment. The Fence API lets the developer combine multiple signals to create fences. When the fence conditions are met, the app receives a callback. For example, a fence could be the user is driving during a weekday in the morning (this usually means going to work) and the app could act accordingly to the user needs.

This fences idea is really interesting as it could provide a really useful solution to 'if-then' problems, being really intuitive for both developers and users.

On the other hand appears Snapshot API, which introduces another useful tool. In this occasion, Snapshot API can be used to get information about the user's current environment. Using the Snapshot API, a variety of context signals can be accessed [36]. These signals can be seen on the figure 4.5.

Context type	Example
Time	Current local time
Location	Latitude and longitude
Place	Place, including place type
Activity	Detected user activity (walking, running, biking)
Beacons	Nearby beacons (including namespace, type, and content)
Headphones	Are headphones plugged in?
Weather	Current weather conditions

Figure 4.5: Context signals on Snapshot API.

- Nearby beacons that you have registered.
- Headphone state (plugged in or not).
- Location, including latitude and longitude.
- Place where the user is currently located.
- Weather conditions in the user's current location.
- Detected user activity, such as walking or driving. (Figure 4.6).

This API provides the idea of getting a 'snapshot' of the user at any moment, giving the developer the ability to check current context easily.

int	IN_VEHICLE	The device is in a vehicle, such as a car.
int	ON_BICYCLE	The device is on a bicycle.
int	ON_FOOT	The device is on a user who is walking or running.
int	RUNNING	The device is on a user who is running.
int	STILL	The device is still (not moving).
int	TILTING	The device angle relative to gravity changed significantly.
int	UNKNOWN	Unable to detect the current activity.
int	WALKING	The device is on a user who is walking.

Figure 4.6: Activities on Snapshot API.

In resume, Google Awareness API might not be what this thesis is looking for, but it introduces two concepts that could be really useful for developing a generic IoT system: fences and snapshots.

4.3.3 Others

Besides Google Awareness API, there were not found any other APIs as interesting as this one. However, there are other projects and services worth checking.

One of them is Resonance [37]. Resonance is a service from a company called Atooma, specialized in artificial intelligence (AI). It provides an SDK able to get users' habits from the devices connected to their smartphones and the use of their apps. It combines user habits (past), current context (present) and gives predictions of what the user will need (future). In other words, it analyzes past and present with an AI layer in order to anticipate users' needs. From Resonance can be extracted the importance of time dimension and AI, but it is not a solution for this thesis problem, as it is a closed system (it only works with some devices).

There are many other IoT APIs, such as Amazon Web Services IoT [38] which does not take into account context. They just connect IoT devices and orchestrate them, combining different devices but offering no more than a bridge between them.

4.3.4 Conclusions

Whereas some interesting concepts were found during this phase, it does not feel like the right direction to choose. Literature on this topic is quite poor or do not adapt to the objectives, and although it is not impossible, developing an API that unifies all the objectives for this thesis is quite difficult given the time available to work on it.

Fences and snapshots are two concepts really interesting to work with, also it is even more clear now the importance of time dimension for analyzing user context (past, present and future in Resonance) and the opportunities that offer an AI layer.

So finally, creating an API is discarded, and it is decided to develop a generic proposal, an architecture for contextual and distributed IoT systems user, and therefore, smartphone centered. The concepts learnt during this phase will be used for creating this architecture.

4.4 Psychological approximation to Human-Computer Interaction

Since one of the aims of this thesis is to create a human-centric system, it is of vital importance to not forget this humanistic factor during the research stage.

As it was stated before, IoT is a multidisciplinary field, it would be a mistake to count only with technical information.

Therefore, research continues exploring some of the most famous psychological theories applied to Human-Computer Interaction (HCI). During this section some of these theories will be explained, namely Activity Theory, Situated Actions and Distributed Cognition. For each of them it will be explored how can they define a proper HCI system, and how can they be integrated in the designing of an IoT system.

4.4.1 Activity Theory

Activity Theory is a conceptual framework originating from the socio-cultural tradition in Russian psychology. The foundational concept of the framework is 'activity', which is understood as purposeful, transformative, and developing interaction between actors ('subjects') and the world ('objects'). The framework was originally developed by the Russian psychologist Aleksei Leontiev. A version of Activity Theory, based on Leontiev's framework, was proposed in the 1980s by the Finnish educational researcher Yrjo Engestrom. Currently, both Leontiev's and Engestrom's variants of Activity Theory, as well as their combinations, are being widely used interdisciplinarily, not only in psychology, but also in a range of other fields, including education, organizational learning, and cultural studies. [39]

Adopting an activity-theoretical perspective has an immediate implication for design: it suggests that the primary concern of designers of interactive systems should be supporting meaningful human activities in everyday contexts, rather than striving for logical consistency and technological sophistication. Currently many systems fail to comply with this, seemingly obvious, requirement. For instance, traditional desktop systems organize digital resources into formal categories (e.g., files, email messages, bookmarks...) rather than according to the relevance of a resource to the task at hand, and most systems provide limited support for task switching and interruptions.

Activity-centric computing is an approach to designing interactive systems according to which the top priority and an explicit aim in the design of digital artefacts and environments should be supporting meaningful human activities.

So in resume, the goal of Activity Theory is understanding the mental capabilities of a single individual. However, it rejects the isolated individuals as insufficient unit of analysis, analyzing the cultural and technical aspects of human actions.[40]

Activity Theory is most often used to describe actions in a socio-technical system through six related elements (Figure 4.7) of a conceptual system expanded by more nuanced theories, being its main three elements subject, instruments and objects:

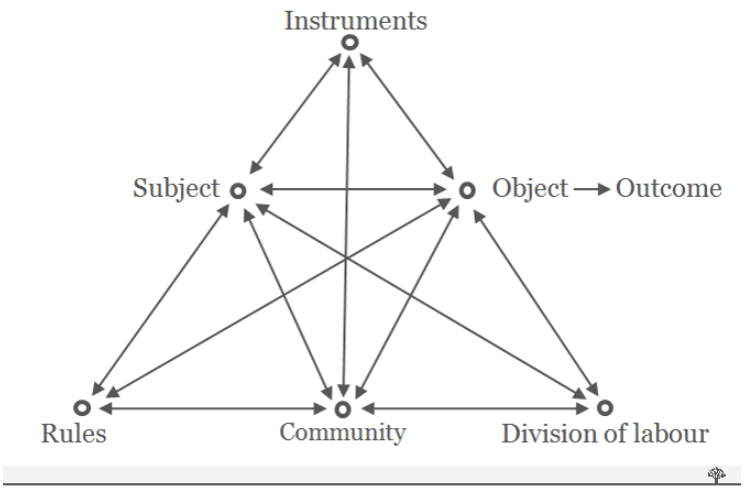

Figure 4.7: Elements of Activity Theory.

- Object-orientedness: the objective of the activity system. Object refers to the objectiveness of the reality; items are considered objective according to natural sciences but also have social and cultural properties. Intangible items can be objects too, as for example desires or goals.

- Subject or internalization: actors engaged in the activities; the traditional notion of mental processes.

- Community or externalization: social context; all actors involved in the activity system.

- Tools or tool mediation: the artifacts (or concepts) used by actors in the system. Tools influence actor-structure interactions, they change with accumulating experience. In addition to physical shape, the knowledge also evolves. Tools are influenced by culture, and their use is a way for the accumulation and transmission of social knowledge. Tools influence both the agents and the structure.

- Division of labor: social strata, hierarchical structure of activity, the division of activities among actors in the system.

- Rules - conventions, guidelines and rules regulating activities in the system. Activity Theory helps explain how social artifacts and social organization mediate social action.

So, how can Activity Theory actually help to build a good IoT system? A problem with intuitive, commonsense notions of activity is that they can be different for different people. In addition, they may be not specific enough (which is one of the main problem when developing a generic IoT framework or architecture). How to distinguish activities from non-activities? Can activities be broken down into smaller units? What role does technology play in human activity? To answer these and other similar questions HCI needs a more elaborated concept of activity. Such concept is offered by Activity Theory.

One of the main concepts that Activity Theory introduces is the division of needs into three dimensions, namely: motive, goals and conditions (Figure 4.8). This division can be very useful when defining what an user needs and how they can get it solved.

- Motives are the abstraction of a need, its inner intention (e.g., when

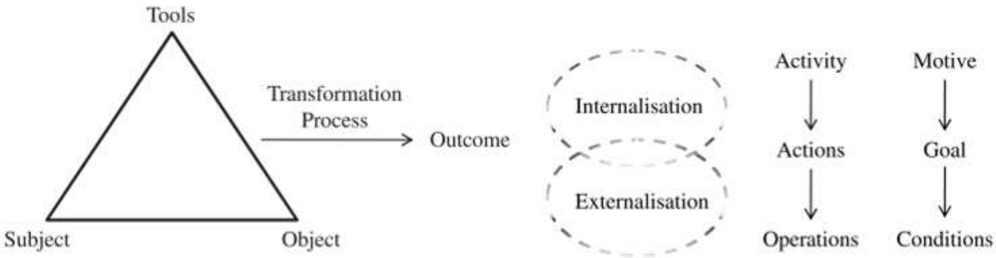

FIG. 1. The basic structure of activity theory.

TABLE 1. Description of the basic terms.

Activity: Not necessarily conscious, but may become conscious "WHY"	Governed by motive/ motives (Collective)
Actions: Conscious "WHAT"	Governed by goals (Individual or group)
Operations: Conscious when learned but can become unconscious or automatic in routine. "HOW"	Governed by conditions (Nonconscious)

Figure 4.8: Explanation of Activity Theory.

turning on the heating its motive could be having hosts at home and trying to make them more comfortable).

- Goals are the objective of a need (e.g., coming back to previous example, its objective would be simply stop being cold).
- Conditions are the snapshot of a need, the subconscious mechanism that wakes that need (e.g., for the previous example the condition would be temperature too low).

Another important concept from Activity Theory is mediation. Mediation states that a user perception of reality can change depending of the tools that user possesses. Imagine an adult and a teenager, the adult will probably have a car and a driving licence but the teenager will not. The perception of distance for the adult given that situation is quite different from the teenager's perception.

In conclusion, object-orientedness, mediation, need dimensions, division of labor can help a lot when building an IoT system human-centric. In following sections it will be seen how further.

4.4.2 Situated Actions

Situated Actions theory borns from Activity Theory, as an extension of its approach (Figure 4.9). This theory stresses the knowledgeability of actors (users) and how they use common-sense practices/procedures to produce, analyze and make sense of one another's actions and their local or situated circumstances. That is to say, understanding in order to do things. [41]

Rather than attempting to abstract action away from its circumstances and represent it as a rational plan, the approach is to study how people use their circumstances to achieve intelligent action. [42]

Within this theory appears three concepts that can be quite useful in defining a context-aware system: indexicality, ad hoc y mutual intelligibility.

Indexicality expresses the idea that communication and actions often depend for their meaning on a reference to things around users (e.g., pointing to a button on a photocopier and say 'try pressing there'. The phrase and gesture indexes the button on the photocopier in a way that makes it clear what 'there' refers to).

Ad Hoc is a latin phrase which means 'for this purpose'. The idea it captures is that of actions or solutions which are taken for a specific purpose, not for the general purpose. Its relevance with respect to the idea of Situated Actions is that in addition to general purpose plans, a lot of actions have this 'Ad Hoc' nature (e.g., as well as having a general idealised plan of how to use a photocopier, some actions might be improvised in response to specific events

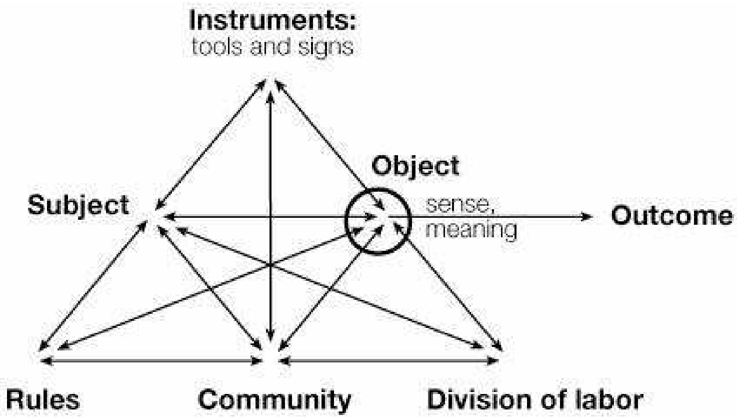

Figure 4.9: Explanation of Situated Actions.

- like when a paper-jam happens).

Finally, Mutual intelligibility captures the idea that when there are other people, a user can understand what they are doing and people can understand what the user is doing. Mutual (two people each) intelligibility (can understand). The relevance of this concept is that it is not possible to have this level of mutual intelligibility with a machine such as a photocopier, stressing the importance of human factor and sociability.

In summary, what can be extracted from Situated Actions theory is that human interactions is an instrumental interaction (this concept is linked with mediation concept introduced by Activity Theory), when users make actions these are guided by their objective, their functionality. Also, this theory stresses that human beings work by procedural interactions, which means all actions can be divided into a sequence of simpler actions, making them easier to process by artificial intelligence.

Some proposals take human actions as if they were predefined, taking for granted that human behaviour can be always predictable without exceptions. From the point of view of Situated Actions this is a huge mistake, as an excessive planification does not guide a user actual actions (as it was stated in the photocopier example).

To conclude, this theory supports the division of needs presented by Activity Theory, as indexicality and ad hoc represents conditions and goals respectively. The concept it exposes of human actions could be really useful when developing an IoT system, splitting actions into simpler actions and not taking behaviour as something predefined.

4.4.3 Distributed Cognition

As opposed to the previous theories, this one is much simpler, although its content is quite interesting too.

Distributed Cognition is a branch of cognitive science that proposes cognition and knowledge are not confined to an individual; rather, it is distributed across objects, individuals, artefacts, and tools in the environment (Figure 4.10). It can be distributed even through space and time. This theory is more accurately a useful descriptive framework that describes human work systems in informational and computational terms. It is useful for analyzing situations that involve problem-solving. As it helps provide an understanding of the role and function of representational media, therefore, it has implications for the design of technology in the mediation of the activity, because system designers will have a stronger, clearer model of the work. [43]

Distributed Cognition approach has three key components: [44]

- Embodiment of information that is embedded in representations of interaction.
- Coordination of enaction among embodied agents.
- Ecological contributions to a cognitive ecosystem.

How can all of this be extrapolated to an IoT system? In this occasion it is quite intuitive, for example, having an IoT room, with different sensors

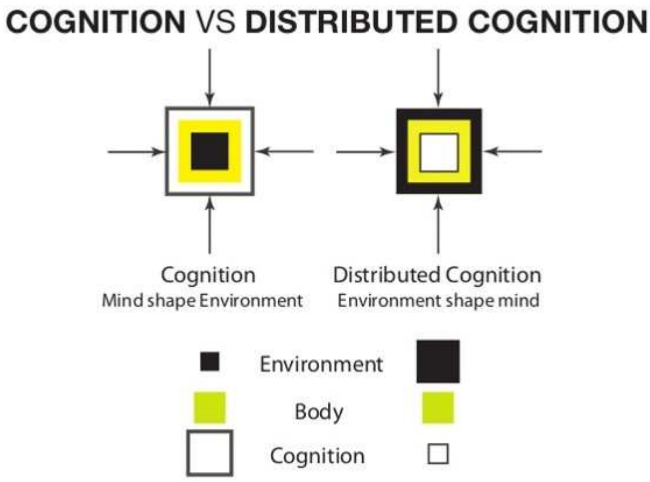

Figure 4.10: Explanation of Distributed Cognition.

and devices such as light, presence, noise, temperature and humidity sensors. All of these sensors can be taken as extensions of the user's cognition, as they provide him information of the room which he could not have gotten by himself. In other words, as the user would receive information of the room from these sensors, just as he would receive information from his own senses, they are part of his cognition.

This point of view can be particularly useful in the design of an IoT system, as it proposes indeed a human-centric approach where there is only one cognitive element, the user. Other interpretation of this theory might be, given a centralized system orchestrating sensors, it would also take into account the information coming directly from the user, as if being another sensor.

4.4.4 Control of the user on context-aware systems

This section aims to provide a brief reflection on which is the most suitable level of control that a user should have on a context-aware system.

On [45] , researchers define three levels of interactivity between a mobile computing device and its user: personalization, passive context-awareness and active context-awareness.

Personalization, sometimes also referred to as customization and tailoring, is a common feature of computing applications. Limiting the scope to mobile computing, it is exemplified by the settings in a mobile phone, where some studies expose that the majority of users use the default setting or change a small subset of the possible features. [46]

Active context-awareness describes applications that, on the basis of sensor data, change their content autonomously, where passive context-aware applications merely present the updated context to the user and let the user specify how the application should change, if at all. A simple example of an active context-aware application is the mobile phone that changes its time automatically when the phone enters a new time zone. In the corresponding passive context-aware application, the mobile phone prompts the user with information about the time zone change and lets the user choose whether the time should be updated or not.

They conducted an experimental case study comparing users' responses towards applications, representing these three levels of interactivity. After their study they found that people felt less in control when using context-aware applications than with personalizing applications. However, the preference for interactivity contradicted this result as users prefered context-awareness over personalization. One commonly observed factor is that even though potential users may disregard a technology a priori, they may adopt it anyway for various reasons.

As conclusion can be extracted that users are willing to accept a large degree of autonomy from applications as long as the application's usefulness is greater than the cost of limited control.

This kind of study can help developers and designers to overcome some of the challenges stated in 4.2 such as feeling loss of control, sense of invasion and decrease of human capabilities. The ideal would be creating a system worth losing some control for its features.

4.4.5 Conclusions

From all of these theories can be extracted quite really interesting conclusions, which were stated on each of their subsections, but in a more general matter the main idea extracted from all of them is that people must be the axis of the development of a good IoT system.

Actually, it is impossible to understand a context-aware application without counting with human psychology. Human life is variable, sociable and ever changing and cannot be understood guided by traditional software developments.

Indeed, context-awareness can have a great impact and benefits at psychological levels in some occasions, providing more independency and a better way to communicate with others and their environment to people with some difficulties (disabilities, isolation, sickness...).

In further sections the concepts from these theories will be applied in defining Situational Context concept.

4.5 Situational Context: Theoretical Framework

In this section the definition for Situational Context is detailed, which provides the theoretical basis that sustains this thesis' result.

First of all, the definition is centered on an individual user, as once the Situational Context is defined for one user it can be extrapolated to be interpreted in a social environment.

The starting point is choosing which elements are going to perform 'main roles' in the context. That is to say, which elements are going to define it. The leading role is, of course, pretty clear: the user, as a user define its own context by himself, with his behaviour, his environment and how he interacts in it.

Taking into account this interaction, Activity Theory can provide other components that define context, as it was stated in 4.4.1 that its three main parts are subject (user), instruments and objects. So, those will be the chosen ones, with slightly changes.

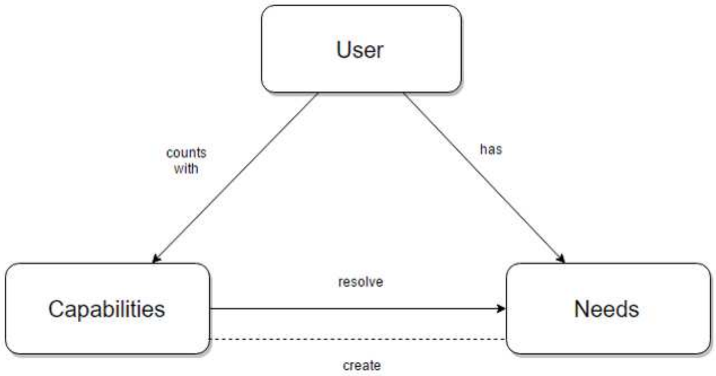

Figure 4.11: Elements of Situational Context.

Instruments will be tools users use to interact with their environment and satisfy their needs (objects). But, since Distributed Cognition stressed how tools are not just tools but an extension of an individual capabilities; instruments will be denominated capabilities in this proposal (Figure 4.11).

During Activity Theory, objects represent both tangible and intangible items in reality (e.g., a tangible item is a chair, and a integible one is the desire of a user to sit on it), these all present in users' life, both environment aspects and needs. In this occasion, objects will stick to its meaning of intangible needs.

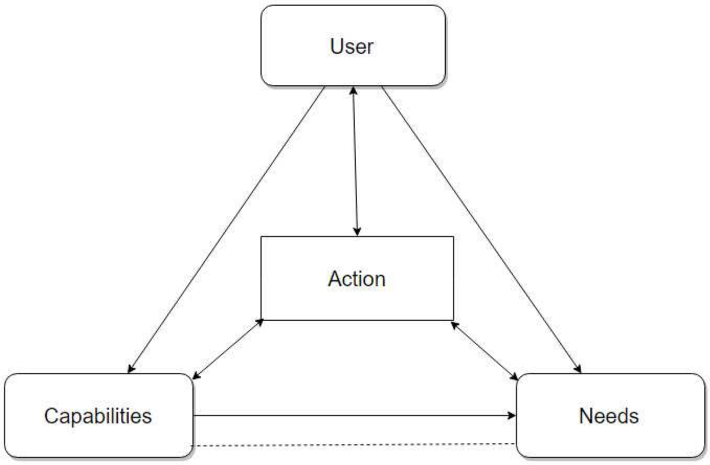

Figure 4.12: Actions as the center of Situational Context.

Therefore, a first definition would be a system where a user counts with capabilities and has needs, being these capabilities able to resolve needs but also creating them (mediation concept in 4.4.1, figure 4.13). And, of course, keeping in mind yet Activity Theory, all these elements would interact with each other by actions. A user performs actions using capabilities and taking care of needs, capabilities perform actions in order to solve users' needs and needs are resolved by actions (Figure 4.12).

As an example: there is the user A, in his environment there is a heating (capability). In a certain moment, A might get cold (need) and therefore would turn on the heating (action). The need of being cold is solved by the action turning on the heating, which is performed by the capability heating.

Focusing further into needs, they will be divided into the three dimensions presented by Activity Theory: conditions, objectives and motives (snapshots, goals and abstractions). Besides, actions could be defined using Indexicality and Ad hoc concepts (section 4.4.2), as the first one relates directly to conditions (snapshots) and the second one to goals (objectives).

As an example: for the need of driving from point A to point B, the

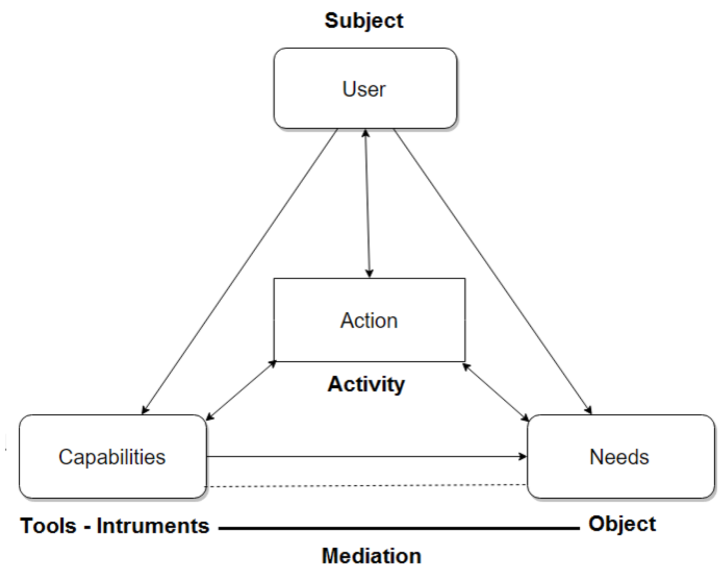

Figure 4.13: Situational Context inspired by Activity Theory.

condition could be a snapshot of a certain time of a daily day where the user enters his car, the goal would be driving to point B and the motive visiting his mother.

Returning to Situated Actions point of view, actions will be analyzed as a sequence of sub actions and never assuming users behaviour would always be the same.

The combination of capabilities and needs can define actions therefore, as an action comes from a capability in order to solve a need (e.g., A turns on (action) the heating (capability) for not being cold anymore (need)). (Figure 4.14).

Coming back to Distributed Cognition, in this theory cognition would be formed by the sum of users' internal capabilities and environment external capabilities (capabilities per se). This cognition would be aligned with needs, as its main focus would be solving them. And from this alignment would result actions.

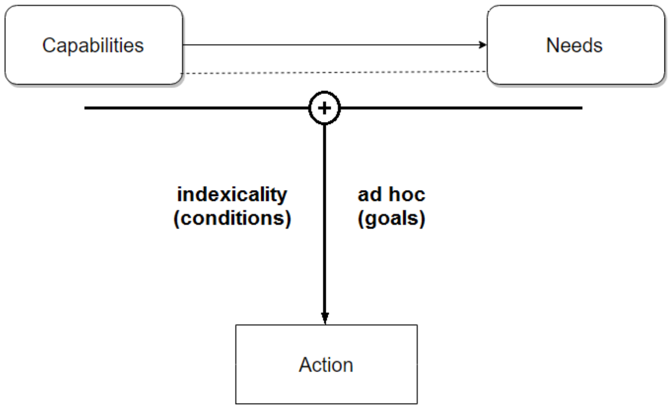

Figure 4.14: Situational Context inspired by Situated Actions.

Summarizing, Situational Context is understood as the union of the capabilities existing in the physical environment where the user is and the needs that this one has in a certain moment.

Needs will vary in time and space, that is, with situations, and so will capabilities. Therefore, it is of vital importance a close relationship between situation and context definition , due to its changing nature.

Users have capabilities and needs. Users are defined by their possibly extractable data from implicit and explicit context (presented in 4.2) enriching their profiles as the system knows more about them.

Users' capabilities will be given by all IoT devices surrounding them. Users have needs, understood as a situation in which they are and objectives to be accomplished (with different levels of abstraction).

Needs are splitted into three dimensions: conditions, objectives and motives; the latter two being defined by the first.

The conditions of an individual's need will be all those objectively analyzable factors that will lead to the attainment of an objective, either to modify them (physical needs, snapshots) or to perform an action influenced by them (fence).

Objectives are direct actions derived from conditions. It is in turn divided into a series of simple actions (e.g., stop being cold leading to turning on heating, driving from point A to B).

Motives are goal's abstraction, their ultimate why. They justify actions regarding users' desires and emotions not only their objective situation. (e.g.,, keeping guests comfortable during a visit, going to visit a relative).

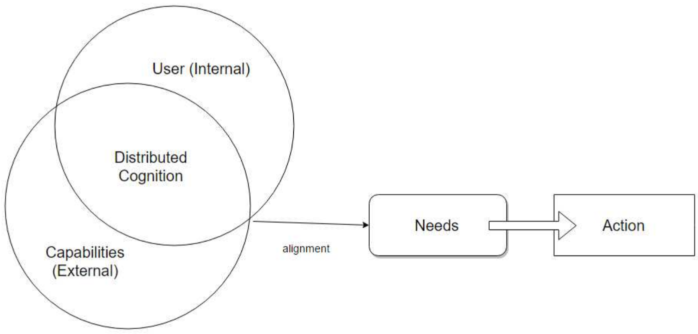

Figure 4.15: Situational Context inspired by Distributed Cognition.

Capabilities can define needs, both by snapshots produced by them and for the possibilities they open. (e.g., a user will not care about his heart rate if there is no sensor measuring it). Actions are activated to accomplish a goal using capabilities.

So, according to Socratic questioning, Situational Context would be defined by:

- Who: users with their profiles, obtained from implicit and explicit context.
- Where: place where users are and capabilities available.
- When: moment in time and its snapshot.

- What: activity and its objective.

- Why: motive, abstraction.

In image 4.16 a resume of this theoretical framework can be seen.

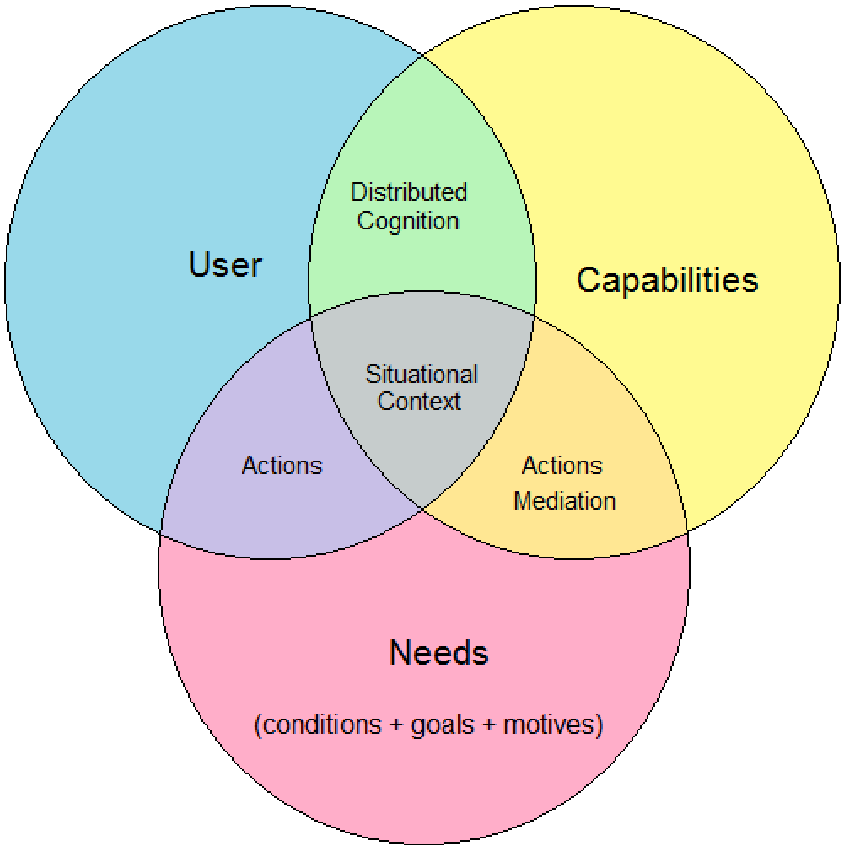

Figure 4.16: Situational Context: Theoretical Framework.

Once the theoretical framework has been settled, it is time to provide technical and real solutions to its content, in order to define further this thesis' proposal.

4.6 Technical Approach

This section aims to refine Situational Context definition, connecting its basis with real technology and tools that could make it work. It will be explored all dimensions of Situational Context, providing real tools for each of them, as well as the social dimension of the system.

4.6.1 Profiles

Profiles are a combination of data extracted from implicit and explicit context, therefore they are defined combining implicit and explicit profiles.

Implicit Profile

Implicit profiles can be built based on data extracted both from a registration form and configuration settings. These two forms could be filled the first time users entered the system, or modified any time they desired.

Also, users' smartphones could provide really valuable information of its use, which could be really useful when defining users' routines. This data can be extracted directly from the smartphone's OS using their APIs.

Social media data could also be a great source of information from the user, and can be easily obtained from each social media APIs.

Of course, users could decide at any moment which of this information they want to be analysed and it would never leave their smartphone, due privacy invasion issues.

Explicit Profile

Explicit information of users can be extracted from daily tasks they perform with their smartphone and with IoT devices: routines, repetitions and all data extracted from sensors surrounding users (including those in smartphones, as

GPS). It can also be obtained from usage data of capabilities (devices) and snapshots. Snapshots can be obtained from Google Awareness API [36].

This information can be very valuable as after a filtering and standardization process, pattern matching techniques could be applied in order to find behaviour patterns. For applying this it is necessary to get a minimum volume of data, so it would be necessary a 'trial' period when starting to use the system. Standardization and pattern matching can be managed by Tensorflow [11], which was stated before in the state of the art section 3. Once patterns are defined, the system would try to find correlations, as if-then cases.

Also, there might be some patterns entered manually on the system by the user or by default in order to perform recognition of basic tasks. Due to the changing nature of human behaviour, as stated in Situated Actions, some situations would require confirmation from users before being automated. The frequence of confirmations asked to users or whether they are necessary or no could be configurable by users at any moment.

Therefore, in order to define users' explicit profiles, the system would analyze this data, extracting patterns guided by time and space (places). These actions can be performed in a smartphone due to the nature of the contextual data which would be easily stored and managed, as mainly would be numerical data with small size.

Example

Ana is using her new IoT system. During weekdays, before arriving work, Ana follows a quite similar routine:

- Ana wakes up, using her smartphone alarm.
- Checks her social media.
- Starts playing music and make herself a coffee.

- Goes to the bathroom, takes a shower.
- Goes back to her bedroom, checks weather prediction on her smartphone and dresses up.
- Goes out, listens music with headphones.
- Heads for her car.
- Connects music via bluetooth to car's speakers.
- Goes to work, using the same route she always does.
- Parks in her place.

On the one hand, there would be Ana's implicit profile, which would contain all the information she provided in the registration form, her configuration settings, her ID in the system, information extracted from her social media and use of smartphone.

On the other hand, from this routine could be extracted many patterns, information for her explicit profile:

- Sleeping hours.
- Waking hours during weekdays.
- Use of apps.
- Turning on/off lights (assuming smart bulbs).
- Check of weather.
- Driving route.
- Music routine, music taste.
- Parking place.

Storage, classification and treatment

As stated before, contextual data would be easily stored and managed, as mainly would be numerical data with small size.

Regarding implicit profile, its information would be mainly static, needing an update just from time to time. This information would be composed by registration data, extracted data from social media or smartphone usage and configuration settings.

For explicit profiles, there would be some guidelines helping data classification, reducing its complexity. Firstly, all data would be classified by its time (as it is the dimension that guides most patterns in daily life). Classify information by place could be again really useful, especially when combined with time.

Other obvious but useful division would be between weekdays and weekends, because despite of its simpleness most people follow two different routines which are repeated over time.

With more volume of data the system would be able to classify it by events such as weather (raining, sunny day), activity (running, walking), place (home, work), etc.

4.6.2 Capabilities

It is necessary to connect all capabilities available (sensors, devices) with the centralized system in the smartphone. The system would need to know which devices are in the environment and which actions can perform each of them.

In order to transmit this information, due its simple nature, devices could connect with smartphones via Bluetooth [47]. Bluetooth provides a range long enough to work with devices within a house, it gives each device an unique device and thanks to Bluetooth Profiles is easy to get which actions can perform each device.

Bluetooth Profiles (Figure 4.17) are different for each type of device, such as GPS and speakers. Bluetooth manager, in this occasion the smartphone can get which profile have each device implemented.

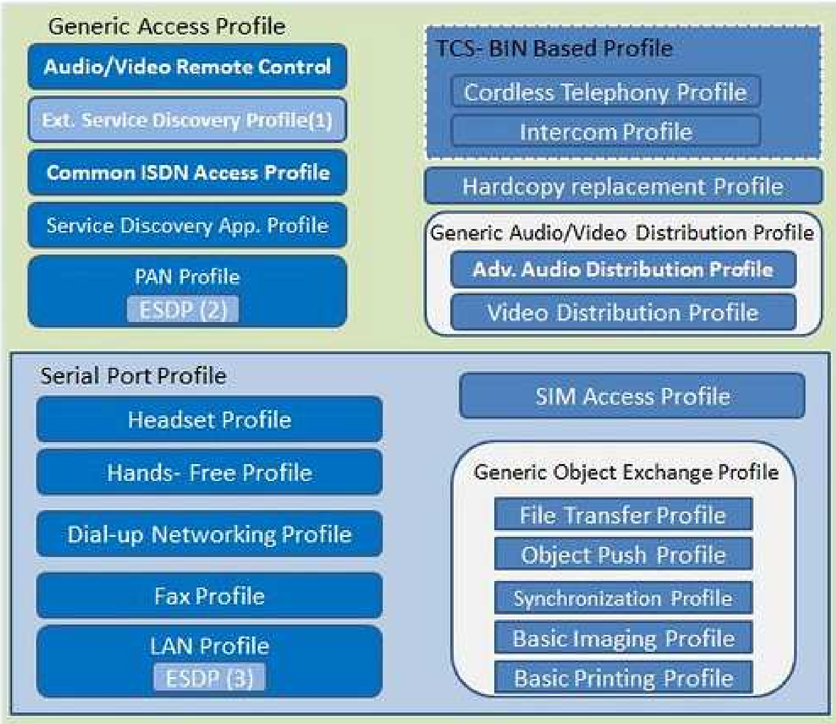

Figure 4.17: Bluetooth Profiles stack [5].

Returning to the previous example, Ana's smartphone would identify her speakers, light bulbs, heating and car as external capabilities. Within the smartphone would be other capabilities such as GPS, headphones, etc.

4.6.3 Needs

As stated in previous sections, needs would be divided into conditions, objectives and abstractions.

Conditions

Needs can be automated due their conditions. These conditions would be extracted from snapshots taken from Google Awareness API, learnt patterns or users' preferences.

Snapshots would be like a 'photograph' of user's context: weather conditions, location, place, activity, time, etc. This information can trigger some action if it is not aligned with user's preferences (if temperature is colder than the value introduced in the preferences, that would be automatically a need and therefore heating would be turned on). This behaviour would be the same with learnt patterns, automated or with previous authorization.

Objectives

Objectives are the purpose of a need and can be divided into sub actions to achieve it.

Here appears an opportunity to create a more automated IoT system. These sub actions could be aligned with capabilities using semantic and linguistic processing.

As an example, if the objective is turning on the heating or checking weather conditions using semantic processing with these words and capabilities, the system would be able to trigger the needed action of a capability, such as 'turning on heating' or 'display weather app'.

These could be done using The Skip-gram Model, which is supported again by Tensorflow. [48] Skip-gram is able to predict the 'context' of a given word, therefore could be able to contextualize the capability for each need.

In the last instance, motives could be introduced into system directly by users or, after a long usage of the system, when a lot of user information were available it could be processed with artificial intelligence in order to abstract the intention of an user when performing an action. As firstly it is not a vital dimension for adapting an IoT system to users' context, it will be discarded.

A resume of this technocal approach can be seen in 4.18.

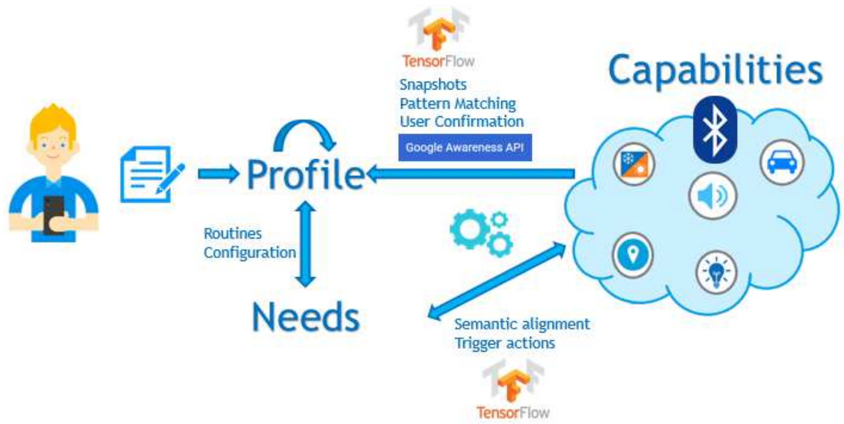

Figure 4.18: Situational Context: Technical Approach.

4.6.4 Social Integration

As it was introduced in previous sections, once an individual context is defined it is easier to integrate it in a social dimension.

Context is defined by users, capabilities and needs. In this section it will be explained how each of these factors adapt themselves in a social level:

Users would have a public and private profile. Private profile would only be visible by his own system. A system would have an owner user, being the others guests. Each capability would know which is his owner and the system would only be able to access to guests' public profiles. Capabilities could also be public or private, designed by the owner's preferences. They would be able to interact with public preferences or needs of guests. Needs' privacy would be configured by users.

Furthermore, in some places there would be one or more owners. These users would decide which capabilities are public to guests. In these situations there would be an owner profile that would include all owner users within an environment, this would work similarly to family accounts of Spotify, Netflix... Several users would work under the same owner profile, all of them being able to interact with the system at the same level (except for a possible parental control, which could be easily implemented as, even though all users would access under an owner account, each of them would have a different and unique ID).

As an example: a couple with their daughter, which is 12. The three of them can access the system of their house with their smartphones using a owner account which includes the three of them. However, each of them would access with their respective emails and passwords, as the three of them would be included into the owner account. The parents can configure a create an account for her daughter, restricting what she can do. All the owners can configure which elements are public to guests and which not. When a guest arrived the house, he would be able to interact with just the public capabilities.

In any system all users would be able to see others (if they wish so), making easy to interact between each other.

This social dimension of the IoT system will be explored further in next sections.

4.7 Conclusions

During this section has been stated how, using nowadays technology a good, generic human-centric IoT system can be built. This system is guided by human psychology theories, making it user friendly and respecting its privacy and security.

In next sections, this approach will be detailed in a more technical way by diagrams and designing its components and services.

Chapter 5

Results and Discussion

Content

With this section it is intended to show the final result of this thesis. Once all theoretical basis have been established, besides a technical way to make it true, it is necessary to show a specific design of its components and services, delimit its scope, as well as its functionalities.

To conclude this section, results will be validated according to all the literature reviewed and discussed properly, stating what was discarded and what can be improved.

5.1 Scope

The results of this thesis are the definition of Situational Context, as was stated in previous sections (section 4.5) and the design of a generic architecture, smartphone-centric, for distributed management of contextual information.

For this architecture has been proposed a theoretical framework (section 4.5), a technical approach (section 4.6) and the design of its components and services (section 5.2).

A system built upon this proposal could be able to manage all contextual information of an IoT environment, where there could be one or more users. The system would learn from users' routine and would be able to adapt itself to each user's needs. It would offer an easy way to manage IoT environments, centralizing it all in the smartphone and would respect users' privacy and security, keeping all their data inside smartphones, never going to third hands.

5.2 Components and Services

This section will follow an up-down description of the system, from the most general point of view and abstraction describing general nodes, then components and finally services and data exchange.

5.2.1 General nodes

The system is composed by two kinds of nodes, namely external and internal. External nodes are related to IoT devices, while internal nodes are those included in the smartphone, which will orchestrate external nodes.

These internal nodes can adapt two types of behaviour, namely owner mode and guest mode, depending on whether they are acting in their own environment or in another one. These modes will be explained further in next sections. They affect how components act inside internal nodes, activating or deactivating some of them and changing their behaviour.

5.2.2 Components and services

Inside an external node, there is always a component called 'IoT Device', which works as a generic component for all IoT devices. This one manages all the information an IoT device is able to send and receive. Its output is the device information, while its inputs are petitions coming from the smartphone, for actions or more information. Its main purpose is to identify devices and which actions can perform.

Next, internal nodes are explained by dividing them in owner and guest mode.

Owner Mode

In a situation with no guests, within this mode, six components remain active (see image 5.1 for more details). These are:

- Connectivity Manager: is the 'bridge' between internal and external nodes, as it receives and sends all information outside the smartphone. It sends as outputs petitions for the 'IoT Device' component, asking for actions or information, besides it sends raw usage data of IoT devices to the 'A.I. Scheduler'. Whilst as inputs it receives data from 'IoT Device' and requests for specific IoT actions from 'Context Manager'.

- Context Manager: it is the system's core. It is responsible for creating a contextual map with all IoT and actions available (and displaying it to users), creating action requests and triggering them, showing confirmation messages to users and remembering their preferences, and managing all information inside the system, specifically processing A.I. reports and triggering the corresponding behaviour. As output it sends action requests, queued in 'User Manager' (as there might be several owners),

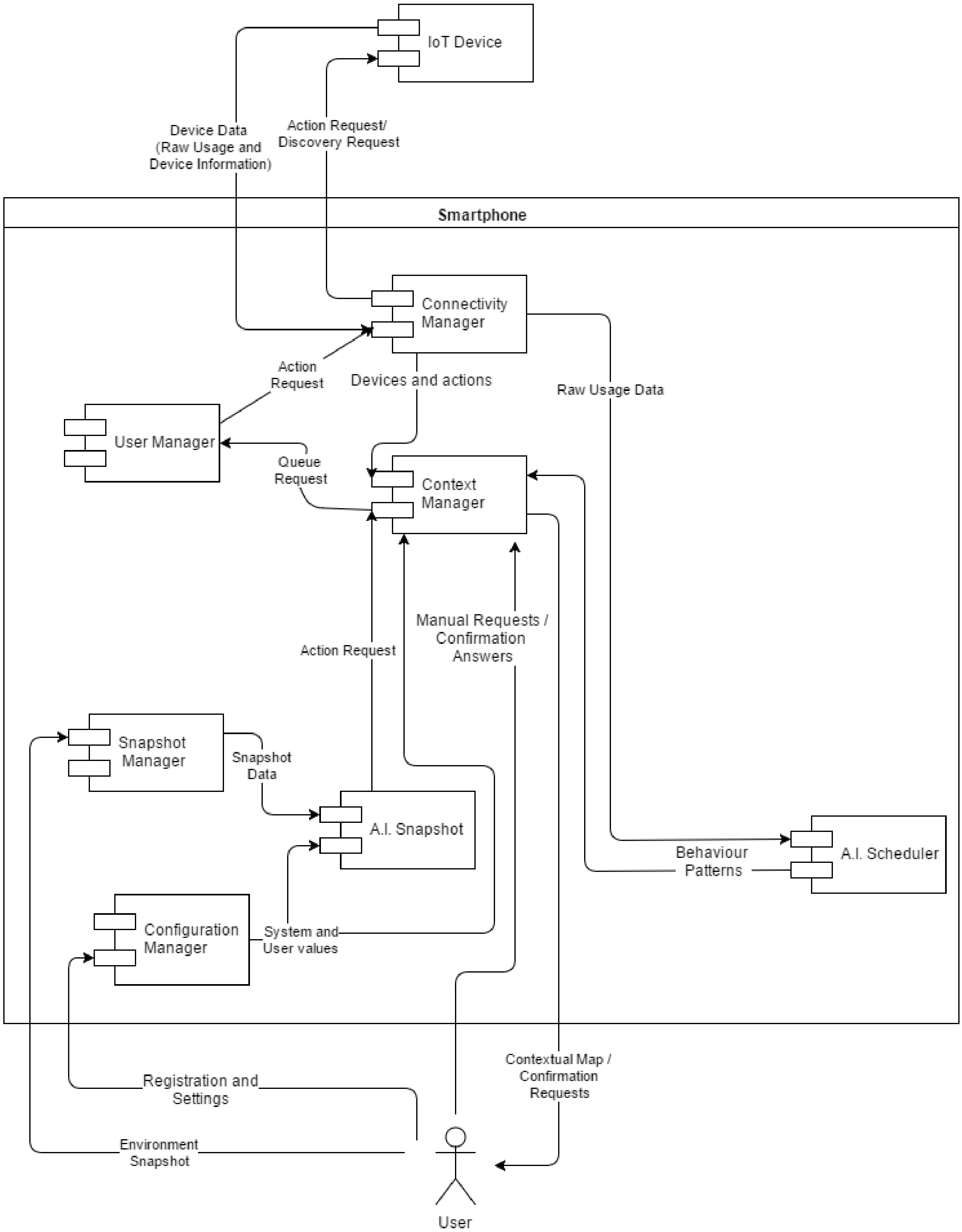

Figure 5.1: Situational Context: Components Diagram.

confirmations to users and displays the contextual map; while as inputs receive:

  - IoT devices and actions available from 'Connectivity Manager'
  - Manual requests from users.
  - Actions requested by 'A.I. Snapshot'.
  - Schedule from 'A.I. Scheduler', that is to say, usage patterns.
  - System values, coming from 'Configuration Manager'.
  - Answers from confirmations sent to users.

- A.I. Scheduler: its main function is finding patterns within usage data, so that user behaviour can be automatized. It receives as input usage data from 'Connectivity Manager' and once it has processed it, sends patterns to 'Context Manager'.

- A.I Snapshot: it is responsible for comparing snapshot data and system values, automatically detecting needs if some value is higher or lower than it should (light, temperature, volume...). It receives snapshots from 'Snapshot Manager' and system values from 'Configuration Manager'. Once it has compared them, sends the identified needs to 'Context Manager' in order to trigger whatever action is necessary.

- Configuration Manager: it displays configuration settings to users, so that they can modify them and insert new ones. These settings are saved and stored as system values which will guide the system behavior. Therefore, as input it receives data introduced by users and as output sends system values to 'Context Manager' and 'A.I. Snapshot'.

- Snapshot Manager: it works with Google Snapshot API, receiving data from the environment and sending it to further analysis.

- User Manager: which only purpose it to manage users petitions, avoiding hoarding. Once a petition is ready to be activated it is sent to 'Connectivity Manager' which will trigger the desired action in 'IoT Device'.

However, if a guest is detected, some things change in this behaviour, both services and components, as a new one is activated.

A guest will send a request to owner's 'Connectivity Manager', asking for permission to access the system. The guest's 'Connectivity Manager' will identify that it is in an unknown place and detect the owner's smartphone.

Once access has been granted, an ID will be assigned to each guest, whilst 'Connectivity Manager' sends a request to 'Context Manager' to send the public contextual map to guests.

Guests send petitions to owner's system thanks to this map and these petitions are handled by 'Context Manager' queuing them by their ID (so that an user could not monopolize the use of a device, despite the owner always having priority access), handling them according to a FIFO policy. Then, this queue is handled by 'User Manager'.

This behaviour is the same when there are more than one owner, except that all their information will be public for the system (different users under an owner account). All petitions will be queued and the system will know all petitions and owners' information.

Guest Mode

Once a user enters in guest mode (when it receives access to other users' system), most functionalities of the system are deactivated. See image 5.2 for more details.

When it receives the public contextual map from the owner, the guest's 'Context Manager' collect needs, system values and patterns and send them to the owner's 'Context Manager' so it can handle them normally, tagging this

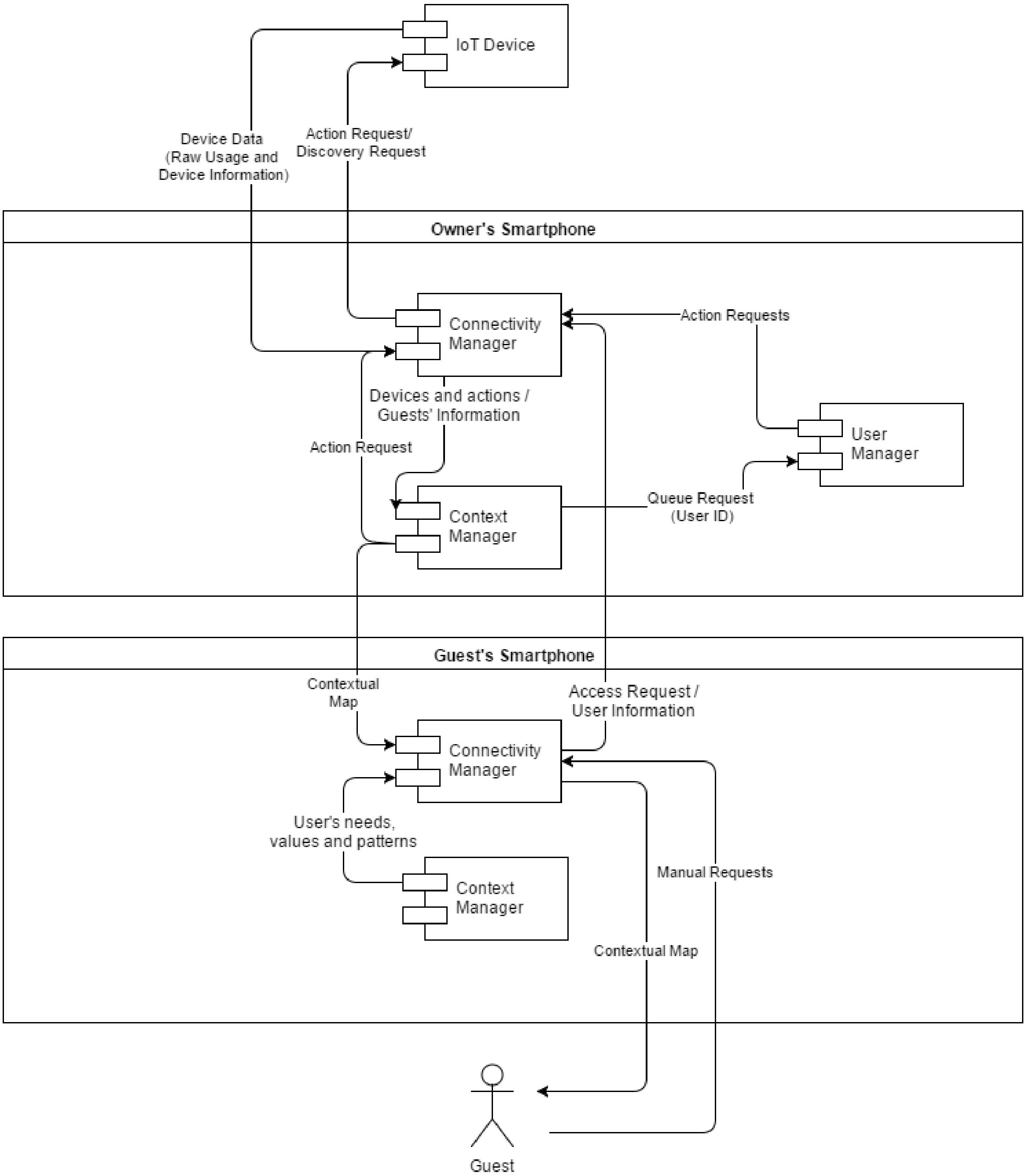

Figure 5.2: Situational Context: Interaction with guests.

information with the corresponding user ID. With the information received it will trigger actions, queuing them in 'User Manager'.

5.3 General Vision: Results

In resume, the result is a proposal for building IoT systems, making use of the smartphone and its capabilities as core of the system. This decision provides a cheap solution, affordable to all targets, as it is a tool almost everybody already own.

The present system is able to adapt itself to users' behaviour, automating some tasks, but asking users for confirmation when something is not clear enough. That way it keeps the artificial intelligence useful but user-friendly, therefore the user will not feel loss of control at any moment. Besides, users will be able to control all devices from their smartphones at any moment.

Automated processes include different aspects: discovering of behaviour patterns, differences between context and users' configuration and classification of tasks within space and time.

All the design process has been human-centric, trying to make the user feel safe and comfortable at any moment. All parameters in the system can be configured, allowing the user decide what information is public or private, which processes are automated, the frequence of confirmation asking and his desirable environment values (temperature, light, noise...).

It provides a generic architecture for IoT designs and despite it being a high-level design, some technical solutions have been proposed along this document (Google Awareness API, TensorFlow...).

5.4 Discussion

The results obtained during the development of this thesis agree with the objectives and the conclusions achieved during the review of literature.

Taking a look back at section 2, the objectives proposed for this thesis were:

- To create an appropriate definition of the concept 'Situational Context', making it complex enough to include all possibilities within this area.
- To define what would make a good IoT system, identifying what makes the user uncomfortable and what suits him better. Therefore, following a human-centric design. Also, to identify other guidelines to create a strong and secure ecosystem.
- To build a theoretical framework for designing IoT systems following the previous guidelines.
- To design a generic architecture for developing IoT environments, based on the previous theoretical framework and oriented to be managed from the smartphone.
- To propose accurate, real techniques and solutions for this architecture, able to add features and solve problems.

It has been created an original proposal, different to what can be found on literature, and influenced for what can be found on it though.

All the development has been human-centric, protecting interests, privacy and security of users, and taking into account psychological facts on Human-Computer Interaction (guided by the idea of IoT being a multidisciplinary field and that attending only to technical issues would be a huge mistake).

The system is ready to adapt itself to one or more users, promoting social interaction between them. Also, it is ready to act accordingly to context and distributed information, making users' lives easier.

Being a high-level design provides a solution to obsolescence and the fast way hardware gets old-fashioned. Smartphones are here to stay and although they may evolve, what this thesis proposes can be adapted easily to them. Changes and evolution of IoT devices are not a problem neither, as they interact with the system via communication, and protocols will not change that much in a medium future (many developments are yet using Bluetooth and Wi-Fi). Smartphones provide a cheap and non-intrusive solution, as users are used to them and do not need to buy a new device.

Technical solutions provided are based on a good theoretical basis, therefore if needs change with time, these can change too without affecting the architecture's design.

In conclusion, it has been described a theoretical framework that provides some strong and robust guidelines for developing an IoT system. Also, it has been proposed a generic architecture, smartphone-centric based on this framework.

Chapter 6

Conclusions

Content

In this final section conclusions extracted from the development of this thesis are stated, both personal and technical. Also, it includes which future work could be done following the results obtained.

Most technical conclusions were stated in the previous section 5.4, but it is of special importance stressing that all initial objectives have been covered, indeed over time.

Also, it has been conluded that when developing pervasive technology, all development should be human-centric, obtaining that way the best features from it, allowing to improve drastically daily lives.

Besides, it has been proved that using the smartphone as they key element of the system, allows to provide a cheaper solution than regular IoT systems, and avoids easier hardware obsolescence.

Following a generic prorposal like the one developed in this thesis would increase the growth of this technology, as it would be more accessible to everybody.

As a result this thesis presents an original work, a theoretical framework to develop IoT systems and an architecture proposal built over it. The proposal is robust, respectful to the user, generic, distributed, and can evolve easily towards new technologies.

6.1 Future Work

Internet of Things is a field where there are a lot to explore and develop yet. Little by little, society will get used to these systems, making their implantation easier and cheapening their components.

Sticking to this thesis' result, the obvious line to continue this project is implementing it. Due to the lack of time, and the adequate extension for a master's thesis, its result had to remain theoretical.

The ideal future work would be creating a couple of experimental scenarios with some IoT devices (speakers, heater, lights, air conditioner, car...), with different users: both owners and guests. These users would interact with the environment through the app developed.

Another interesting line would be to perfect the artificial intelligence of the system using deep learning techniques [49], allowing to even predict some behaviours. Regarding artificial intelligence, other interesting way that have been mentioned along the document but could not be introduced in the design, due its complexity, is semantic alignment (Section 4.6).

Semantic alignment techniques could provide the system with a huge independence, allowing it to deduce automatically which capabilities could resolve a need.

Introducing biometrics for users' identification and security could also be really interesting, including voice commands and gestures, making the system even more pervasive.

6.2 Personal Reflection

Throughout the realization of this work I have obtained a vast learning, both in a technical and personal dimension.

From the technical point of view not only my engineering knowledge have increased, but also I have learned a lot about researching, context-awareness, IoT, HCI and psychology. I also have learnt about how to develop a deep research from the most general basis until defining and targeting a concrete problem. Also, I have learnt about Tensorflow, Google Awareness API and other interesting tools.

During these months of work I have given practical use to the whole content that includes the degree in software engineering and the master's degree in computing engineering; remembering and securing them. This content goes from mathematical and theoretical bases, programming knowledge, protocols, detection of requirements and use cases, software design, project management skills, etc. Furthermore, I have not only put into practice all transversal competences that I acquired during my studies, but also I have improved and obtained others.

Following an agile development methodology has motivated me to improve my skills in teamwork, writing and speaking. Also, starting a new work completely from scratch and having to develop it until getting a satisfactory proposal has pushed me enormously to expand to the maximum my self-learning and problem-solving abilities.

Besides, the deep research previously mentioned and writing this document has helped me to improve my technical English. Indeed I challenged myself to write this whole document in English, in a way to get further knowledge from this thesis. Also, I have learnt to do a better use of my ability to planning and time management.

Personally I am very satisfied with the final result of the work, since the first moment it was very clear that it was essential to comply with all the objectives set out initially and delivering a good quality result.

Finally, a theoretical framework and an architectural design have been proposed, covering all objectives. Lastly, it should be stressed that selecting a work whose subject matter follows too many different research lines, has allowed me to have more freedom when it comes to innovating and has pushed me to work harder in designing and searching solutions by myself.

Bibliography

[1] Flexera Software. Big numbers: 50 billion connected devices (by 2020). http://blogs.flexerasoftware.com/elo/2014/12/big-numbers-50-billion-connected-devices-by-2020.html, 2017.

[2] La población mundial en 2100. http://sociologiac.net/2011/05/13/la-poblacion-mundial-en-2100/, 2017.

[3] Internet of things: why it is important? http://www.iotonlinestore.com/, 2017.

[4] Google Awareness API. https://developers.google.com/awareness/, 2017.

[5] Sagar Sapkota. Theory behind android- arduino communication via bluetooth — build circuit. http://www.buildcircuit.com/theory-behind-android-arduino-bluetooth-communication/, 2017.

[6] Technological Eras - TechEd. http://www.techedlearning.com/technological-eras-sc.html, 2017.

[7] Ernst Wittmann, Hadlee Simons, Hadlee Simons, Hadlee Simons, Hadlee Simons, Hadlee Simons, Hadlee Simons, Matthew Alexander, and Hadlee Simons. The internet of things is here, and it will revolve around the

smartphone - Memeburn. http://memeburn.com/2015/12/the-internet-of-things-is-here-and-it-will-revolve-around-the-smartphone/, 2017.

[8] Ivan Ganchev, Zhanlin Ji, and Máirtín O'Droma. A generic iot architecture for smart cities. In *Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014). 25th IET*, pages 196–199. IET, 2013.

[9] Amelie Gyrard, Soumya Kanti Datta, Christian Bonnet, and Karima Boudaoud. Standardizing generic cross-domain applications in internet of things. In *2014 IEEE Globecom Workshops (GC Wkshps)*, pages 589–594. IEEE, 2014.

[10] Iara Augustin, Adenauer C Yamin, Luciano Cavalheiro da Silva, Rodrigo Araújo Real, Gustavo Frainer, and Cláudio FR Geyer. Isamadapt: abstractions and tools for designing general-purpose pervasive applications. *Software: Practice and Experience*, 36(11-12):1231–1256, 2006.

[11] Vector representations of words. https://www.tensorflow.org/versions/r0.11/tutorials/word2vec/index.html#the-skip-gram-model, 2017.

[12] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[13] Prasant Misra, Yogesh Simmhan, and Jay Warrior. Towards a practical architecture for the next generation internet of things. *arXiv preprint arXiv:1502.00797*, 2015.

[14] Joaquin Guillen, Javier Miranda, Javier Berrocal, Jose Garcia-Alonso, Juan Manuel Murillo, and Carlos Canal. People as a service: a mobile-centric model for providing collective sociological profiles. *IEEE software*, 31(2):48–53, 2014.

[15] Javier Miranda, Niko Mäkitalo, Jose Garcia-Alonso, Javier Berrocal, Tommi Mikkonen, Carlos Canal, and Juan M Murillo. From the internet of things to the internet of people. *IEEE Internet Computing*, 19(2):40–47, 2015.

[16] Javier Berrocal, Jose Garcia-Alonso, Cristina Vicente-Chicote, Juan Hernández, Tommi Mikkonen, Carlos Canal, and Juan M Murillo. Early analysis of resource consumption patterns in mobile applications. *Pervasive and Mobile Computing*, 2016.

[17] Guanling Chen, David Kotz, et al. A survey of context-aware mobile computing research. Technical report, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.

[18] Panu Korpipää and Jani Mäntyjärvi. An ontology for mobile device sensor-based context awareness. In *International and Interdisciplinary Conference on Modeling and Using Context*, pages 451–458. Springer, 2003.

[19] Tao Gu, Hung Keng Pung, and Da Qing Zhang. A middleware for building context-aware mobile services. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 5, pages 2656–2660. IEEE, 2004.

[20] Patrick Fahy and Siobhan Clarke. Cass–a middleware for mobile context-aware applications. In *Workshop on context awareness, MobiSys*. Citeseer, 2004.

[21] Vasos Hadjioannou, Constandinos X Mavromoustakis, George Mastorakis, Evangelos K Markakis, Dimitra Valavani, and Evangelos Pallis. Context awareness location-based android application for tracking purposes in assisted living. In *Telecommunications and Multimedia (TEMU), 2016 International Conference on*, pages 1–7. IEEE, 2016.

[22] Gabriella Castelli, Marco Mamei, Alberto Rosi, and Franco Zambonelli. Extracting high-level information from location data: the w4 diary example. *Mobile Networks and Applications*, 14(1):107–119, 2009.

[23] Hans W Gellersen, Albercht Schmidt, and Michael Beigl. Multi-sensor context-awareness in mobile devices and smart artifacts. *Mobile Networks and Applications*, 7(5):341–351, 2002.

[24] Anind K Dey and Jonna Häkkilä. Context-awareness and mobile devices. *User interface design and evaluation for mobile technology*, 1:205–217, 2008.

[25] Mark Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.

[26] Mark Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7):75–84, 1993.

[27] Gregory D Abowd, Anind K Dey, Peter J Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *International Symposium on Handheld and Ubiquitous Computing*, pages 304–307. Springer, 1999.

[28] Kevin Ashton. That ?internet of things? thing. *RFiD Journal*, 22(7):97–114, 2009.

[29] Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten, and Jean-Claude Burgelman. Scenarios for ambient intelligence in 2010. *Office for official publications of the European Communities*, 2001.

[30] Fariba Sadri. Ambient intelligence: A survey. *ACM Computing Surveys (CSUR)*, 43(4):36, 2011.

[31] Bin Guo, Daqing Zhang, Zhiwen Yu, Yunji Liang, Zhu Wang, and Xingshe Zhou. From the internet of things to embedded intelligence. *World Wide Web*, 16(4):399–420, 2013.

[32] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1):414–454, 2014.

[33] What is pervasive computing? http://internetofthingsagenda.techtarget.com/definition/pervasive-computing-ubiquitous-computing, 2017.

[34] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[35] Jakob E Bardram. The java context awareness framework (jcaf)–a service infrastructure and programming framework for context-aware applications. In *International Conference on Pervasive Computing*, pages 98–115. Springer, 2005.

[36] Google Snapshot API.
https://developers.google.com/awareness/android-api/snapshot-api-overview,
2017.

[37] Atooma Team. Resonance — predicting human behaviours.
https://www.atooma.com, 2017.

[38] Amazon web servicies, Inc.
https://aws.amazon.com/es/documentation/iot/, 2017.

[39] Activity theory: The encyclopedia of human-computer interaction.
https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/activity-theory,
2017.

[40] Activity theory: Wikipedia.
https://en.wikipedia.org/wiki/Activity_theory, 2017.

[41] Situated action - hiowiki.
https://wiki.hiof.no/index.php/Situated_Action, 2017.

[42] Situated actions lecture summary.
http://jareddonovan.com/teaching/hci/situated-actions/, 2017.

[43] Distributed cognition - learning theories.
https://www.learning-theories.com/distributed-cognition-dcog.html,
2017.

[44] Distributed cognition: Wikipedia.
https://en.wikipedia.org/wiki/Distributed_cognition, 2017.

[45] Louise Barkhuus and Anind Dey. Is context-aware computing taking control away from the user? three levels of interactivity examined. In

International Conference on Ubiquitous Computing, pages 149–156. Springer, 2003.

[46] Stanley R Page, Todd J Johnsgard, Uhl Albert, and C Dennis Allen. User customization of a word processor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 340–346. ACM, 1996.

[47] Bluetooth basics - learn.sparkfun.com. https://learn.sparkfun.com/tutorials/bluetooth-basics/, 2017.

[48] Word2vec tutorial - the skip-gram model - chris mccormick. http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/, 2017.

[49] Deep learning. http://deeplearning.net/, 2017.